REQUEST FOR EX PARTE REEXAMINATION TRANSMITTAL FORM

Address to:
Mail Stop Ex Parte Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

1. **[X]** This is a request for ex parte reexamination pursuant to 37 CFR 1.510 of patent number 6,108,704 issued August 22, 2000. The request is made by:

   [ ] patent owner

   [X] third party requester.

2. **[X]** The name and address of the person requesting reexamination is:
   EDWIN H. TAYLOR
   BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
   1279 OAKMEAD PARKWAY,
   SUNNYVALE, CALIFORNIA 94085

3. **[ ]** a. A check in the amount of $2,520 is enclosed to cover the reexamination fee, 37 CFR 1.20(c)(i);

   [X] b. The Commissioner is hereby authorized to charge the fee as set forth in 37 CFR 1.20(c)(i) to Deposit Account No. 02-2666; or

   [ ] c. Payment by credit card. Form PTO-2038 is attached.

4. Any refund should be made by
   [ ] check or
   [X] credit to Deposit Account No. 02-2666.
   37 CFR 1.26. If payment is made by credit card, refund must be to credit card account.

5. **[X]** A copy of the patent to be reexamined having a double column format on one side of a separate paper is enclosed. 37 CFR 1.510(b)(4).
   <u>EXHIBIT A</u>

6. **[ ]** CD-ROM or CD-R in duplicate, Computer Program (Appendix) or large table

7. [ ] Nucleotide and/or Amino Acid Sequence Submission
   If applicable, all of the following are necessary

   a. [ ] Computer Readable Form (CRF)

   b. [ ] Specification Sequence Listing on:

      i [ ] CD-ROM (2 copies) or CD-R (2 copies); **or**

      ii [ ] paper

   c. [ ] Statements verifying identify of above copies

8. **[ ]** A copy of any disclaimer, certificate of correction or reexamination certificate issued in the patent is included.

9. **[X]** Reexamination of claims(s) 1-7 and 10-44 is requested.

10. **[X]** A copy of every patent or printed publication relied upon is submitted herewith including a listing thereof on Form PTO-1449 or PTO/SB/08.

11. [ ] An English language translation of all necessary and pertinent non-English language patents and/or printed publications is included.

12. [X] The attached detailed request includes at least the following items:.

a. [X] A statement identifying each substantial new question of patentability based on prior patents and printed publications 37 CFR 1.510(b)(1)

b. [X] An identification of every claim for which examination is requested, and a detailed explanation of the pertinency and manner of applying the cited art to every claim for which reexamination is requested. 37 CFR 1.51(b)(2)

13. [ ] A proposed amendment is included (only where the patent owner is the requester). 37 CFR 1.510(e)

14. [X] a. It is certified that a copy of this request (if filed by other than the patent owner) has been served in its entirety on the patent owner as provided in 37 CFR 1.33(c).
The name and address of the party served and the date of service are:
JEFFREY S. GINSBERG, ESQ.
KENYON & KENYON
ONE BROADWAY
NEW YORK, NY 10004

Date of Service: February 17, 2009 ; or

[ ] b. A duplicate copy is enclosed since service on patent owner was not possible.

15. [x] Correspondence Address: Direct all communication about the reexamination to:

[x] Customer Number: 8791
Type Customer Number here
Place Customer Number Bar Code Label here

OR

[ ] Firm or Individual Name: Edwin H. Taylor

Address (line 1) 1279 Oakmead Parkway

Address (line 2)

City Sunnyvale State California Zip 94085-4040 Country USA

Telephone (408) 720-8300 Fax (408) 720-8383

16. [X] The patent is currently the subject of the following concurrent proceeding(s):

[ ] a. Copending reissue Application No.

[ ] b. Copending reexamination Control No.

[ ] c. Copending Interference No.

[X] d. Copending litigation styled:

Net2Phone, Inc. v. eBay, Inc., Skype Technologies SA, and Skype, Inc., Case No. 06-2469, United States District Court for the District of New Jersey.

WARNING: Information on this form may become public. Credit Card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

/ET/
Authorized Signature

/02-17-2009/
Date

[ ] For Patent Owner Requester

[X] For Third Party Requester

Based on Form PTO/SB/57 (02-01) as modified by BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP on 05/09/03

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Request for Ex Parte Reexamination	§	REQUEST FOR EX PARTE REEXAMINATION
	§	
U.S. Patent No. 6,108,704	§	
	§	
Issued: August 22, 2000	§	
	§	Attorney Docket No.: 03801.G184
For: Point-to-Point Internet Protocol	§	
	§	
Requester: Skype, Inc.	§	Customer No.: 08791

<u>REQUEST FOR EX PARTE REEXAMINATION UNDER 35 U.S.C. § 302</u>

Mail Stop *Ex Parte* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Pursuant to the provisions of 35 U.S.C. §§ 302-307, the undersigned hereby requests an *ex parte* reexamination of claims 1-7 and 10-44 of United States Patent No. 6,108,704 ("the '704 patent," Exhibit A) which issued on August 22, 2000 to Glenn W. Hutton et al. resulting from a patent application filed on September 25, 1995. The Requester hereby asserts that claims 1-7 and 10-44 of the '704 patent are unpatentable over prior art references not before the Patent and Trademark Office (PTO) during prosecution of the '704 patent.

TABLE OF CONTENTS

I. PENDING LITIGATION

The '704 patent is the subject of pending litigation, Net2Phone, Inc. v. eBay, Inc., Skype Technologies SA, and Skype, Inc., Case No. 06-2469, instituted by the current assignee, Net2Phone, Inc., in the United States District Court for the District of New Jersey. Net2Phone alleges that Skype Technologies SA, Skype, Inc. and eBay Inc. infringe claims 1, 2, 4-7, 11, 22, 32-44 of the '704 patent. The parties have submitted their claim construction briefs and a Markman hearing is currently scheduled for March 2, 2009. The Court has not yet set a schedule for summary judgment proceedings. No trial date has been set. Skype, Inc., plans to file a motion to stay the above-entitled litigation pending reexamination on the grounds that a stay of litigation at this time will permit the Court and parties to benefit from the PTO's guidance on issues of patentability and to avoid further costly legal proceedings that would otherwise burden the Court and parties. Claim Construction Briefs submitted by the parties to the pending litigation are set forth in Exhibits S-X.

II. LISTING OF PRIOR ART PATENTS AND PRINTED PUBLICATIONS

In accordance with 37 C.F.R. §§ 1.510(b)(1) and (b)(2), reexamination of claims 1-7 and 10-44 of the '704 patent is requested in view of the following references:

Exhibit B — The Open Group, Technical Standard, Protocols for X/Open PC Interworking SMB, Version 2, (1992) ("**NetBIOS**"), which **published as a single publication** containing:[1] (a) Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concept and Methods, RFC 1001 (March 1987) ("RFC 1001"); and (b) Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications, RFC 1002 (March 1987) ("RFC 1002").

Exhibit C — Etherphone: Collected Papers 1987-1988 (May 1989) (collectively referred to herein as "**Etherphone**"). These papers, which **published together as a single publication**, include the following:[2]

[1] NetBIOS published as a single reference with RFC 1001 and RFC 2002.
[2] The five papers comprising this reference were published together as set forth on the first page of this reference. Thus, all five papers are a single reference.

a. Polle T. Zellweger, et al., *An Overview of the Etherphone System and its Applications*, IEEE CONFERENCE ON COMPUTER WORKSTATIONS (March 1988), 160-168 (hereinafter "Zellweger 1").

b. Daniel C. Swinehart, *Telephone Management in the Etherphone System*, PROCEEDINGS OF THE IEEE/IEICE GLOBAL TELECOMMUNICATIONS CONFERENCE (November 1987), 1176-1180 (hereinafter "Swinehart 1").

c. Douglas B. Terry and Daniel C. Swinehart, *Managing Stored Voice in the Etherphone System*, ACM TRANSACTIONS ON COMPUTER SYSTEMS 6(1) (February 1988), 3-27 (hereinafter "Terry").

d. Daniel C. Swinehart, *System Support Requirements for Multi-media Workstations*, PROCEEDINGS OF THE SPEECHTECH '88 CONFERENCE (April 1988), 82-83 (hereinafter "Swinehart 2").

e. Polle T. Zellweger, *Active Paths through Multimedia Documents*, DOCUMENT MANIPULATION AND TYPOGRAPHY, J.C. AN VILET (ED.), CAMBRIDGE UNIVERSITY PRESS (1988) (hereinafter "Zellweger 2").

Exhibit D Vin, Harrick M., et al., *Multimedia Conferencing in the Etherphone Environment*, IEEE COMPUTER SOCIETY (October 1991) ("**Vin**"); and

Exhibit E Droms, R., Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("**RFC 1531**")

Exhibit F Pinard, et al., U.S. Patent No. 5,533,110 ("Pinard")

Exhibit G VocalChat User's Guide, Version 2.0 (1994) ("User's Guide")

Exhibit H VocalChat Readme File, Version 2.02 (June, 1994) ("Readme")

Exhibit I VocalChat 1.01 Networking Information (March 6, 1994) ("VocalChat Networking")

Exhibit J VocalChat Information, Version 2.02 (July 18, 1994) ("Help File")

Exhibit K VocalChat Troubleshooting Help File, Version 2.02 (July 18, 1994) ("Troubleshooting Help File")

III. OVERVIEW OF THE '704 PATENT

Before providing detailed explanations of the pertinency and manner of applying the cited prior art to the claims, presented here is an overview of the '704 patent and its prosecution history. The '704 patent issued on August 22, 2000, and includes 44 claims, of which claims 1, 2, 4, 10, 21, 32, 33, 38, 43, and 44 are independent.

A. Subject Matter of the '704 Patent

The '704 patent describes two different techniques for locating computer processes on a network. Referring to Figure 1 of the '704 patent (reproduced below), one technique relies on a "connection server" (26) to locate processes and a second technique relies on a "mail server" (28) to locate processes.[3] According to the first technique, each computer (referred to as a "processing unit" in the '704 patent) registers its IP addresses with the connection server (26). The IP address of each "online" computer is stored within a database (34) on the connection server. As described in the '704 patent (referring to Figure 1):

> Upon the first user initiating the point-to-point Internet protocol when the first user is logged on to 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 a connection service provider, is processed by the connection server 26 to be established in the database 34 as an active on-line party.[4]

[3] The first technique is referred to as the "primary point-to-point Internet protocol" and the second technique is referred to as the "secondary point-to-point internet protocol." *See*, *e.g.,* '704 patent, Col. 5, line 55 - Col. 6, line 29.
[4] '704 patent, Col. 5, lines 25-38.

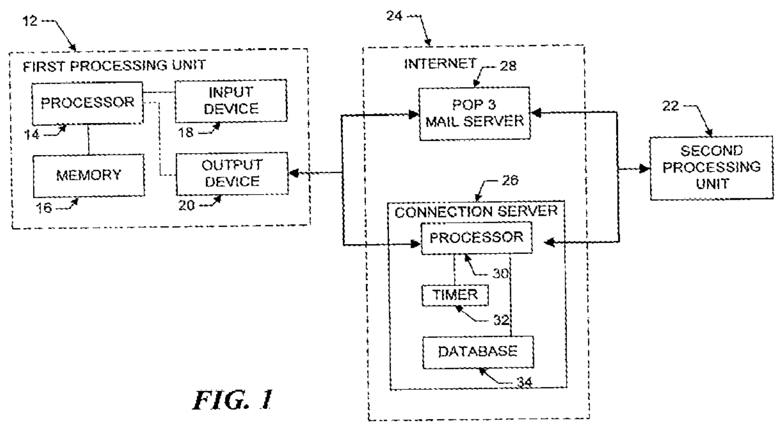


FIG. 1

In order to initiate a connection with the second computer (22) on the network, the first computer (12) retrieves the current IP address of the second computer from the connection server (26). Once the first computer knows the IP address of the second computer, it can establish a point-to-point connection with the second computer. As described in the '704 patent:

> 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.[5]

The second technique for locating computers on a network (the "secondary point-to-point Internet protocol") utilizes the email server (28) illustrated in Figure 1. The second technique is used "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."[6] Using the second technique, the first computer (12) transmits an email message which includes the IP address of the first user and a session number (referred to as a "<ConnectRequest>" message).[7] After receiving the email message from the mail server, the second computer (22) uses the IP address and session number to establish a point-to-point connection with the first computer (12). As described in the '704 patent:

---

[5] '704 patent, Col. 5, lines 55-67.
[6] '704 patent, Col. 6, lines 20-23.
[7] '704 patent, Col. 6, lines 31-36.

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.[8]

While the independent claims of the ‘704 patent are not expressly limited to a particular protocol standard, the embodiments described in the ‘704 patent utilize the TCP/IP protocol.[9] Thus, the focus of the ‘704 patent is a central repository of IP addresses which is queried to locate computers on a network.

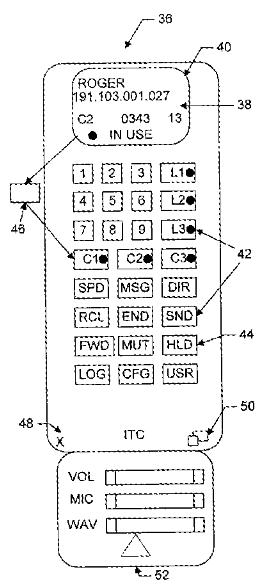


FIG. 5

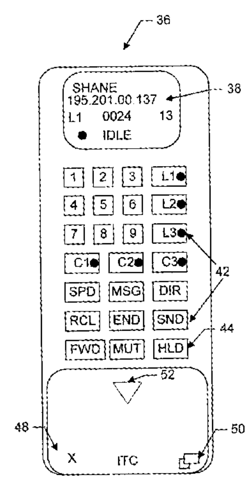


FIG. 6

The ‘704 patent also describes a graphical user interface (“GUI”) for managing calls on a computer. The GUI, illustrated in Figures 5 and 6 of the ‘704 patent (reproduced above), includes a status area (38) which is used to indicate . . .

. . . 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”,

[8] ‘704 patent, Col. 7, lines 10-25.

[9] *See, e.g.*, ‘704 patent, Col. 5, lines 8-14 and 22-38.

> 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."[10]

Figures 5 and 6 also illustrate a set of graphical icons (42) which are "configured to substantially simulate a telephone handset or a cellular telephone interface."[11] The icons provide functions typically found on a telephone such as speed dial (SPD), hold (HLD), send (i.e., initiate call) (SND), end call (END), mute (MUT). Icons are also provided to indicate individual telephone "lines" (L1-L3) and "conference lines" (C1-C3). An active call may be transferred to a different line by "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."[12]

B. Prosecution History of the '704 Patent

The application which resulted in the '704 patent was filed on September 25, 1995. The '704 patent application[13] initially included a total of 20 claims. This was extended to 53 claims via a preliminary amendment.[14]

The PTO mailed a first Office Action on June 2, 1997, rejecting all 53 claims under 35 U.S.C. § 103 as being unpatentable in view of several prior art references, including Civanlar, U.S. Patent No. 5,581,552 ("Civanlar"). As noted by the Examiner, Civanlar discloses "a communication protocol in which the requesting node sends a request for communication with another node through an address server, which contains an address database, to obtain the address and routing information necessary to complete the communication."[15]

The Applicants filed an Amendment and Response on December 2, 1997 in which they added claims 54-68 and distinguished the invention from the cited references as follows:

> Applicant's invention provides techniques for determining the current dynamically assigned network protocal [sic] address of a user process connected

---

[10] '704 patent, Col. 8, lines 42-50.
[11] '704 patent, Col. 8, lines 58-60.
[12] '704 patent, Col. 9, lines 36-42.
[13] Application serial no. 08/533,11 resulting in the '704 patent is referred to as the "'704 application."
[14] *See* Preliminary Amendment (April 5, 1996).
[15] Office Action (June 2, 1997), page 3.

to the network. The first technique utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.[16]

The Applicants also submitted a Declaration of Prior Invention under 37 U.S.C. § 1.131, stating "to overcome the rejection of all claims under 35 U.S.C. § 103 as being unpatentable over Civanlar, et al. in view of Morgan et al. and/or further in view of December et al . . . In light of the declaration and accompanying exhibits, all rejections based on the Civanlar et al. reference are deemed moot."[17]

The PTO mailed a second Office Action on April 14, 1998 indicating a restriction requirement under 35 U.S.C. § 121. The Applicants mailed a response to the restriction requirement on August 11, 1998, electing a group of claims (Group 1).

An Office Action mailed on October 28, 1998 rejected all pending claims under 35 U.S.C. § 102(e) and 35 U.S.C. § 103. The Applicants mailed an Amendment and Response on March 1, 1999, cancelling claims 1-4 and 6-11 and amending numerous claims. The Applicants again distinguished the alleged "invention" over the cited prior art, stating:

> Applicants' invention solves a fundamental problem associated with the Internet. . . The problem is: How can a global network user be located if he/she has no permanent network address?
>
> Applicants have disclosed a solution to the above-described problem. The solution utilizes a client/server system. In the disclosed system, a client process contacts a dedicated address directory server and forwards to the server the network protocol address to which it has been assigned upon connection to the computer network, along with other identification information. The dedicated address directory server maintains a compilation or list of entries, each of which contain a process identifier and the corresponding network protocol address forwarded to the server by the process itself. Other processes wishing to contact a desired target process simply

[16] Office Action Response (December 2, 1997), page 8.

[17] Amendment and Response (December 2, 1997), page 7. We question the sufficiency of the support document, "webph.doc," submitted with the Declaration of Prior Invention. However, because the prior art we rely upon predates the priority date of this document, we did not perform a detailed analysis related to this document.

query the address directory server to determine whether the target process is on-line and the current network protocol address at which the target process is located.[18]

In addition, with respect to independent Claims 10 and 21 of the '704 patent (and associated dependent claims), the Applicants argued that these claims were "directed to a method for establishing a point-to-point communication link with the user interface of a client process by associating elements representing a communication line and various processes."[19]

A Notice of Allowability was mailed on May 25, 1999, allowing Claims 21, 23-24, 26-64, 66, and 67 (which issued as Claims 1-44). The '704 patent issued on August 22, 2000.

IV. SUBSTANTIAL NEW QUESTION (SNQ) OF PATENTABILITY AS REQUIRED BY 37 C.F.R 1.510 (b)(1)

The following section provides a list of the SNQs and detailed explanation of the prior art references relied upon in the present request for the SNQ, including references not previously considered by the PTO.

A. SNQs Raised by NetBIOS

A SNQ as to Claims 1-7 and 32-44 is raised by NetBIOS. NetBIOS anticipates all of the limitations of these claims, including teachings of Civanlar, which were used by the Examiner in a § 103 rejection against the instant claims. This reference was removed from consideration due to the acceptance of a 37 C.F.R. § 1.131 declaration which swore behind the date of Civanlar. While Civanlar shows an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications, the question of patentability was removed when the reference was antedated. *See, e.g.*, Civanlar, col. 3, lines 1-4 ("The address server contains an address data base for performing address resolution, i.e., translation, between the at least two addresses of each ELAN end-point in response to requests for such translations.").

NetBIOS, which was not cited or discussed in the prosecution of the '704 patent, presents a SNQ of patentability because it, like Civanlar, discloses an address server (referred to as a "NetBIOS Name Server" or "NBNS") with an address database for storing network protocol addresses usable by network nodes to establish point-to-point communications, as discussed in

---

[18] Amendment and Response (March 1, 1999), page 14.
[19] Office Action response (March 1, 1999), page 17.

detail below. *See, e.g.*, NetBIOS at 367 (describing how the NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."). *See also id.* at 388 ("Name query transactions are initiated by endnodes to obtain the IP address(es) and other attributes associated with a NetBIOS name.").

Recall also that, during prosecution of the '704 patent, the Applicants argued that the claimed invention . . .

> . . . utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.

Office Action Response (December 2, 1997), page 8. This is precisely the manner in which nodes in NetBIOS register their own IP addresses with the NBNS and query the NBNS for the IP addresses of other nodes. *See, e.g., id. at* 397 ("The NetBIOS session service begins after one or more IP addresses have been found for the target name. . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>.") (emphasis added).

For all of the foregoing reasons, NetBIOS would be considered important in deciding the question of patentability for all independent claims of the '704 patent and, accordingly, presents a SNQ of patentability, particularly with respect to Claims 1-7 and 32-44.

B. SNQs Raised by NetBIOS in view of RFC 1531

A SNQ as to Claims 1-7 and 32-44 is raised by NetBIOS in view of RFC 1531. NetBIOS and RFC 1531 were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because NetBIOS discloses an address server (referred to as a "NetBIOS Name Server" or "NBNS") for storing network protocol addresses usable by network nodes to establish point-to-point communications (as described above), and RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a

Dynamic Host Configuration Protocol (DHCP) server. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). As argued by Applicants in the prosecution of the '704 patent:

> Applicant's invention provides techniques for determining the current dynamically assigned network protocol address of a user process connected to the network. The first technique utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.

Office Action Response (December 2, 1997), page 8. As described above, these features are all described explicitly in NetBIOS except for "dynamically assigned" network protocol addresses. Consequently, NetBIOS in view of RFC 1531 presents a SNQ of patentability for all claims which require the dynamic assignment of network protocol addresses, as discussed in detail below. Such combination would be considered important in deciding the question of patentability and accordingly present a SNQ of patentability, particularly with respect to Claim 33 which explicitly requires "locating processes having dynamically assigned network protocol addresses" and with respect to various other claims which require that the network protocol address is received "following connection to the computer network." *See, e.g.*, Claims 1, 4, and 38.

C. SNQs Raised by NetBIOS in view of Pinard

A SNQ as to Claims 10-17, 19-28, and 30-31 is raised by NetBIOS in view of Pinard. NetBIOS and Pinard were not cited or discussed alone, or in combination in the prosecution of the '704 patent. Recall that during prosecution of the '704 patent, the Applicants argued that these claims were "directed to a method for establishing a point-to-point communication link with the user interface of a client process by associating elements representing a communication line and various processes." Office Action response (March 1, 1999), page 17. Consequently, the combination of NetBIOS and Pinard presents a substantial new question of patentability because NetBIOS discloses an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications and Pinard discloses graphical elements representing communication lines and callees that may be clicked and dragged to establish and terminate calls, set up conference calls, and place calls on hold, as discussed in detail below. *See, e.g.*, Pinard, Figures 2-16 and associated text. Such combination would be considered important in deciding the question of patentability and accordingly present a substantial new question of patentability, particularly with respect to Claims 10-17, 19-28, 30-31.

D. SNQs Raised by NetBIOS in view of Pinard and further in view of VocalChat User's Guide

A SNQ as to Claims 18 and 29 is raised by NetBIOS in view of Pinard and further in view of VocalChat. NetBIOS, Pinard, and VocalChat were not cited or discussed alone, or in combination, in the prosecution of the '704 patent. The combination presents a substantial new question of patentability because the combination discloses all of the features from NetBIOS and Pinard described above and, in addition, the VocalChat User's Guide describes a "communication line on mute status" as recited in Claims 18 and 29. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57. Such combination would be considered important in deciding the question of patentability and accordingly would present a substantial new question of patentability, particularly with respect to Claims 18 and 29.

E. SNQs Raised by Etherphone

A SNQ as to Claims 1-2, 4-7, 10-12, 14, 19-23, 25, and 30-44 is raised by Etherphone. Etherphone anticipates all of the limitations of these claims, including teachings of Civanlar which were used in a § 103 rejection against the instant claims. This reference was removed from consideration due to the acceptance of a 37 C.F.R. § 1.131 declaration which swore behind the date of Civanlar. While Civanlar shows an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications, the question of patentability was removed when the reference was antedated. *See, e.g.*, Civanlar, col. 3, lines 1-

4 ("The address server contains an address data base for performing address resolution, i.e., translation, between the at least two addresses of each ELAN end-point in response to requests for such translations.").

Etherphone, which was not cited or discussed in the prosecution of the '704 patent, presents a SNQ of patentability because it, like Civanlar, discloses an address server (referred to as a "Voice Control Server" or "Telephone Control Server") for storing network protocol addresses usable by network nodes to establish point-to-point communications, as discussed in detail below. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").

In addition, during prosecution of the '704 patent, the Applicants argued that the claimed invention . . .

> . . . utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.

Office Action Response (December 2, 1997), page 8. This is precisely the manner in which Etherphones/workstations described in Etherphone register their own IP addresses with the Voice Control Server and query the Voice Control Server for the IP addresses of other Etherphones/workstations. *See, e.g.,* Swinehart 1, page 4 (after receiving the IP address of a callee's Etherphone from the Voice Control Server, "voice datagrams are transmitted directly among the participants, bypassing the control server."). *See also id.* ("The *telephone control server* controls voice conversations, implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities . . . It uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues."); Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does.").

For the foregoing reasons, Etherphone would be considered important in deciding the question of patentability and accordingly would present a SNQ of patentability, particularly with respect to Claims 1-2, 4-7, 10-12, 14, 19-23, 25, and 30-44.

F. SNQ Raised by Etherphone in view of NetBIOS

A SNQ as to Claim 3 is raised by Etherphone in view of NetBIOS. Etherphone and NetBIOS were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because Etherphone discloses an address server (referred to as a "Voice Control Server" or "Telephone Control Server") for storing network protocol addresses usable by network nodes to establish point-to-point communications, as mentioned above, and NetBIOS discloses "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory" as recited in Claim 3. *See, e.g.*, NetBIOS at 382 ("[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure."). Such combination would be considered important in deciding the question of patentability and accordingly would present a SNQ of patentability, particularly with respect to Claim 3.

G. SNQ Raised by Etherphone in view of Vin

A SNQ as to Claim 32 is raised by Etherphone in view of Vin. Etherphone and Vin were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because Etherphone discloses an address server (referred to as a "Voice Control Server" or "Telephone Control Server") for storing network protocol addresses usable by network nodes to establish point-to-point communications, as mentioned above, and Vin discloses using TCP/IP as the network protocol in an Etherphone system. *See, e.g.*, Vin, page 77, Figure 5 (illustrating a "protocol stack and format" used in an

Etherphone system which includes internet protocol (IP) packets). Such combination would be considered important in deciding the question of patentability and accordingly would present a SNQ of patentability, particularly with respect to Claim 32, which requires the "Internet protocol."

H. SNQs Raised by Etherphone in view of Vin and further in view of RFC 1531

A SNQ as to Claim 33 is raised by Etherphone in view of NetBIOS and further in view of RFC 1531. Etherphone, Vin, and RFC 1531were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because Etherphone discloses an address server (referred to as a "Voice Control Server" or "Telephone Control Server") for storing network protocol addresses usable by network nodes to establish point-to-point communications, Vin discloses the use of TCP/IP as the network protocol in an Etherphone system, as mentioned above, and RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a Dynamic Host Configuration Protocol (DHCP) server. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). As argued by Applicants in the prosecution of the '704 patent:

> Applicant's invention provides techniques for determining the current dynamically assigned network protocol address of a user process connected to the network. The first technique utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.

Office Action Response (December 2, 1997), page 8. Etherphone, Vin and RFC 1531, in combination, disclose all of these features. Consequently, such combination would be considered important in deciding the question of patentability and accordingly presents a SNQ of patentability, particularly with respect to Claim 33 which requires "dynamically assigned network protocol addresses."

I. SNQs Raised by Etherphone in view of Pinard

A SNQ as to Claims 10-17, 19-28, and 30-31 is raised by Etherphone in view of Pinard. Etherphone and Pinard were not cited or discussed alone, or in combination in the prosecution of the '704 patent. Recall that during prosecution of the '704 patent, the Applicants argued that these claims were "directed to a method for establishing a point-to-point communication link with the user interface of a client process by associating elements representing a communication line and various processes." Office Action response (March 1, 1999), page 17. The combination presents a SNQ of patentability because Etherphone discloses an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications and Pinard discloses graphical elements representing communication lines and callees that may be clicked and dragged to establish and terminate calls, set up conference calls, and place calls on hold, as discussed in detail below. *See, e.g.*, Pinard, Figures 2-16 and associated text. Such combination would be considered important in deciding the question of patentability and accordingly present a SNQ of patentability, particularly with respect to Claims 10, 12-17, 21, and 24-28.

J. SNQ Raised by Etherphone in view of Pinard and further in view of VocalChat User's Guide

A SNQ as to Claims 18 and 29 is raised by Etherphone in view of Pinard and further in view of VocalChat. Etherphone, Pinard, and VocalChat were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because Etherphone discloses an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications; Pinard discloses graphical elements representing communication lines and callees that may be clicked and dragged to establish and terminate calls, set up conference calls, and place calls on hold (as described above); and VocalChat discloses a "communication line on mute status" as recited in Claims 18 and 29. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57. Such combination would be considered important in deciding the question of patentability and accordingly presents a SNQ of patentability, particularly with respect to Claims 18 and 29.

K. SNQs Raised by VocalChat User's Guide in view of VocalChat Readme, and further in view of VocalChat Networking, and further in view of VocalChat Help File, and further in view of VocalChat Troubleshooting Help File (collectively "VocalChat" or "the VocalChat references")

A SNQ as to Claims 1-2, 4, 7, 10-11, 19-22, 30-42 is raised by VocalChat User's Guide in view of Readme, and further in view of VocalChat Networking, Help File, and Troubleshooting Help File (collectively referred to as "VocalChat"). A strong motivation to combine all of these references exists because they all describe the same VocalChat system. VocalChat anticipates all of the limitations of these claims, including teachings of Civanlar, which were used in a103 rejection against the instant claims. This reference was removed from consideration due to the acceptance of a 37 C.F.R. § 1.131 declaration which swore behind the date of Civanlar. While Civanlar shows an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications, the question of patentability was removed when the reference was antedated. *See, e.g.*, Civanlar, col. 3, lines 1-4 ("The address server contains an address data base for performing address resolution, i.e., translation, between the at least two addresses of each ELAN end-point in response to requests for such translations.").

VocalChat User's Guide, Readme, VocalChat Networking, Help File, and Troubleshooting Help File, which were not cited or discussed in the prosecution of the '704 patent, present a SNQ of patentability because they, like Civanlar, disclose an address server (referred to as a "Post Office") for storing network protocol addresses usable by network nodes to establish point-to-point communications, as discussed in detail below. *See, e.g.*, Readme, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."). In TCP/IP implementations, the Post Office directory includes a "Connection List" file (CONNLIST.VC) containing the unique usernames and IP addresses of connected VocalChat users. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). In earlier implementations, the Connection List file was called the "USERS" File. *See, e.g.*, VocalChat Network Information, page 10 ("When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address.")

Recall also that, during prosecution of the '704 patent, the Applicants argued that the claimed invention . . .

> . . . utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.

Office Action Response (December 2, 1997), page 8. This is precisely the manner in which VocalChat clients register their own IP addresses with the Post Office server and query the Post Office Server for the IP addresses of other VocalChat clients. *See, e.g.*, Help File, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses and a user name should be entered in the Setup for each user."). *See also* VocalChat Network Information, page 10 ("VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address.").

For the foregoing reasons, VocalChat User's Guide in view of Readme, and further in view of VocalChat Networking, Help File, and Troubleshooting Help File would be considered important in deciding the question of patentability and accordingly would present a SNQ of patentability, particularly with respect to Claims 1-2, 4, 7, 10-11, 19-22, 30-42.

L. SNQs Raised by the VocalChat References in view of RFC 1531

A SNQ as to Claims 1-2, 4, 7, 10-11, 19-22, and 30-42 is raised by VocalChat User's Guide in view of Readme, and further in view of VocalChat Networking, Help File, and Troubleshooting Help File, and further in view of RFC 1531. These prior art references were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because VocalChat User's Guide, Readme, VocalChat Networking, Help File, and Troubleshooting Help File disclose an address server (referred to as a "Post Office" server) for storing network protocol addresses usable by network nodes to establish point-to-point communications over a TCP/IP network (as described above), and RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a Dynamic Host Configuration Protocol (DHCP) server. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). In addition, as argued by Applicants in the prosecution of the '704 patent:

> Applicant's invention provides techniques for determining the current dynamically assigned network protocol address of a user process connected to the network. The first technique utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the networks, the process logs-on to the server and provides the server with the network protocal [sic] address under which the process is currently operating. A second process wishing to establish communications with the first process, connects to the server and request the network protocal [sic] address under which the first process is currently operating. Upon receipt of the network protocal [sic] address of the first process, the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.

Office Action Response (December 2, 1997), page 8. As described above, these features are all described explicitly in the VocalChat references except for "dynamically assigned" network protocol addresses. Consequently, the VocalChat references in view of RFC 1531 present a SNQ of patentability for all claims which require the dynamic assignment of network protocol addresses, as discussed in detail below. Such combination would be considered important in deciding the question of patentability and accordingly present a SNQ of patentability, particularly with respect to Claim 33 which explicitly requires "locating processes having dynamically assigned network protocol addresses" and with respect to various other claims which require that the network protocol address is received "following connection to the computer network." *See, e.g.*, Claims 1, 4, and 38.

M. SNQ Raised by the VocalChat References in view of NetBIOS

A SNQ as to Claim 3 is raised by VocalChat in view of NetBIOS. VocalChat and NetBIOS were not cited or discussed alone, or in combination in the prosecution of the '704 patent. The combination presents a SNQ of patentability because VocalChat discloses an address server (referred to as a "Post Office" server) for storing network protocol addresses usable by network nodes to establish point-to-point communications, as mentioned above, and NetBIOS discloses "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory" as recited in Claim 3. *See, e.g.*, NetBIOS at 382 ("[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure."). Such combination would be considered important in deciding the question of patentability and accordingly would present a SNQ of patentability, particularly with respect to Claim 3.

N. SNQs Raised by the VocalChat References in view of Pinard

A SNQ as to Claims 12-18 and 23-29 is raised by VocalChat User's Guide in view of Readme, and further in view of VocalChat Networking, Help File, and Troubleshooting Help File (collectively referred to as "VocalChat" or "the VocalChat references"), and further in view of Pinard. These references were not cited or discussed alone, or in combination in the prosecution of the '704 patent. Recall that during prosecution of the '704 patent, the Applicants argued that these claims were "directed to a method for establishing a point-to-point communication link with the user interface of a client process by associating elements representing a communication line and various processes." Office Action response (March 1, 1999), page 17. The combination presents a SNQ of patentability because VocalChat User's Guide, Readme, VocalChat Networking, Help File, and Troubleshooting Help File disclose an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications and Pinard discloses graphical elements representing communication lines and callees that may be clicked and dragged to establish and terminate calls, set up conference calls, and place calls on hold, as discussed in detail below. *See, e.g.*, Pinard, Figures 2-16 and associated text. Such combination would be considered important in deciding the question of patentability and accordingly present a SNQ of patentability, particularly with respect to Claims 12-18 and 23-29.

V. OVERVIEW OF THE PRIOR ART REFERENCES PRESENTING A SNQ OF PATENTABILITY

A. NetBIOS

The Network Basic Input/Output System, known as NetBIOS, was originally developed for IBM's PC-Network in the early 1980s. In March 1987, the NetBIOS Working Group of the Internet Engineering Task Force released Request for Comments 1001 (“RFC 1001”), titled “Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concept and Methods,” and Request for Comments 1002 (“RFC 1002”), titled “Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications.” Both of these documents were republished in NetBIOS in 1992 (Exhibit B).[20]

NetBIOS is a software interface which allows applications on different computers to communicate within a computer network, such as a local area network or the Internet. “NetBIOS applications employ NetBIOS mechanisms to locate resources, establish connections, send and receive data with an application peer, and terminate connections.” NetBIOS at 359. NetBIOS “defines a proposed standard protocol to support NetBIOS services in a TCP/IP environment. Both local network and Internet operation are supported.” *Id.* at 350.

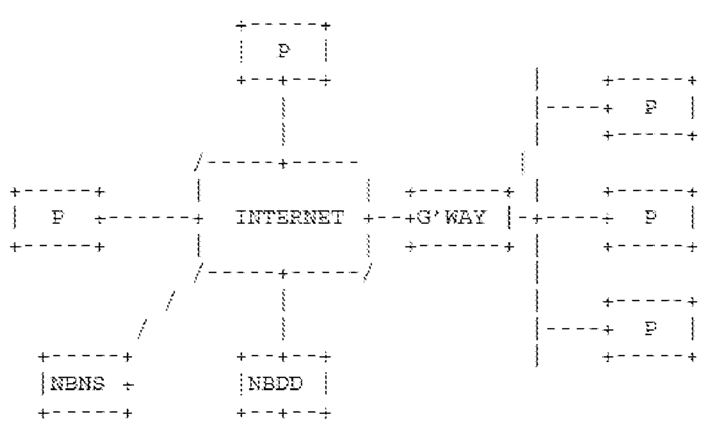


Figure From NetBIOS, page 371 (RFC 1001, page 22)

[20] The pages of NetBIOS are numbered consecutively including RFC 1001 and RFC 1002. When referencing a page in NetBIOS the consecutively numbered pages of RFC 1001 and RFC 1002 are used.

As illustrated in the figure above, NetBIOS enables point-to-point communications between two or more “point-to-point” nodes (also referred to as “P nodes”). The point-to-point connections are established over the Internet or a local area network (LAN). A NetBIOS Name Server (“NBNS”) coupled to the point-to-point nodes over the Internet (or other network) provide a dedicated directory service for associating node names with IP addresses. In operation, NetBIOS point-to-point nodes register distinguishing names and corresponding IP addresses with the NBNS. When a node makes a point-to-point connection it first “queries” the NBNS to obtain the current IP address of other nodes. Having obtained the target node's IP address from the NBNS, the originating node can establish “directed (point-to-point) communications.” *Id.* at 397 (“The NetBIOS session service begins after one or more IP addresses have been found for the target name. . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>.”) (emphasis added). In the December 2, 1997 Office Action Response on page 8, the Patent Owner indicated, among other things, that “the second process establishes communications with the first process directly, without any intervention [sic] from the address/information server.” Accordingly, the examiner would have considered NetBIOS important in determining patentability.

“Name query transactions are initiated by endnodes to obtain the IP address(es) and other attributes associated with a NetBIOS name.” *Id.* at 388. *See also id.* at 376 (describing how “[e]very node has a permanent unique name.”); *id.* at 377 (describing how NetBIOS point-to-point nodes perform “name resolution” by “ask[ing]” the NBNS for the IP address corresponding to a NetBIOS end-node identified by name); *id.* (“Name query (also known as ‘resolution’ or ‘discovery’) is the procedure by which the IP address(es) associated with a NetBIOS name are discovered.”); *id.* at 388 (“Name query transactions are initiated by end nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name.”); *id.* at 389 (“An NBNS answers queries from a P [point-to-point] node with a list of IP address and other information” for the queried name.).

This is a comprehensive reference for Internet and LAN point-to-point connections as will be seen by the citation to NetBIOS in the text below and in the claim charts of Exhibit P. As discussed above, Civanlar was utilized in a § 103 rejection against the instant claims. This reference was removed from consideration due to the acceptance of a 37 C.F.R. § 1.131 declaration which swore behind the date of Civanlar. While this reference shows an address server for storing network protocol addresses usable by network nodes to establish point-to-point communications, the question of patentability was removed when the reference was antedated. Accordingly, NetBIOS, which was not cited or discussed in the prosecution of the '704 patent, presents a SNQ of patentability because it, like Civanlar, discloses an NBNS with an address database for storing network protocol addresses usable by network nodes to establish point-to-point communications, as discussed in detail below. *See, e.g.*, NetBIOS at 367 (describing how the NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."). Consequently, NetBIOS provides a SNQ of patentability as the question of patentability based on Civanlar was removed.

Additionally, NetBIOS was not cited during the prosecution of the '704 patent and, as shown above, would have been considered important to an examiner in deciding patentability of the '704 patent, accordingly, it presents a substantially new question of patentability.

B. Etherphone

The Etherphone system (Exhibit C) "consists of microprocessor-based electronic telephones, a centralized switching server, a voice file server, and workstation programs to support voice communications and voice recording services. From a workstation, a user can place and receive telephone calls, maintain private telephone directories, and manage a database of voice messages."[21]

Figure 1 of Terry (reproduced below) provides an architectural overview of the Etherphone system. A Voice Control Server (sometimes called a "Telephone Control Server") registers the network addresses of workstations/Etherphones of users and provides the network addresses to requesting workstations/Etherphones upon request to establish calls between users. As described in Etherphone:

> The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.[22]

[21] Zellweger 2, page 11.
[22] Swinehart 1, page 4.

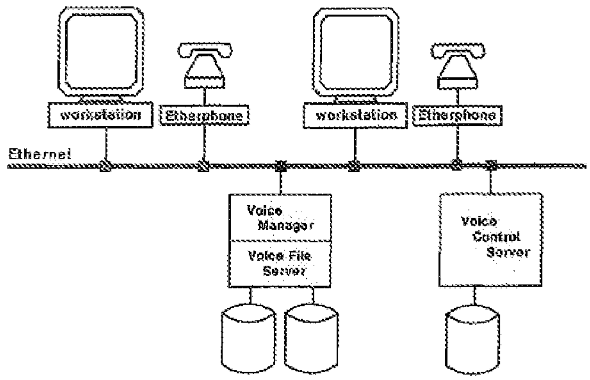


Figure 1. A simple Etherphone system environment.

Thus, after determining the current network addresses for a call, the workstation/Etherphone of the calling user and the workstation/Etherphone of a called user establish a point-to-point ("direct") connection over the network, bypassing the Voice Control Server.

In addition, the Voice Control Server associates user identifiers with each network protocol address. For example, a user may log in to any workstation. Calls to that user will then be directed to that workstation and its associated Etherphone. As described:[23]

> The *telephone control server* controls voice conversations, implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues.[24]

Etherphone also describes a graphical user interface (GUI) to provide various telephony functions. One example of the GUI is provided in Figure 3 of Zellweger 1 (reproduced below). As described in Zellweger 1:

> A variety of convenient workstation dialing methods are provided: a user can . . . select names or numbers from anywhere on the [Etherphone telephone

---

[23] Swinehart 1, page 4 (underline emphasis added). *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does.").

[24] Swinehart 1, page 4 (underline emphasis added). *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does.").

management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad.[25]

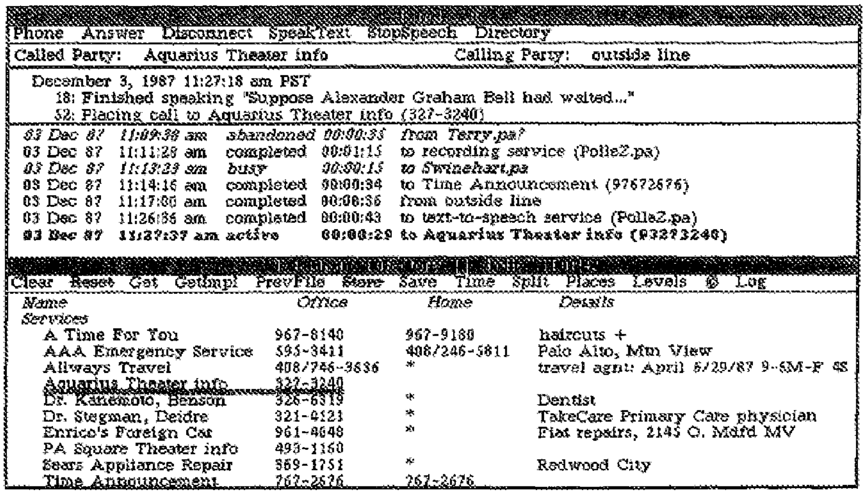


Figure 3 From Zellweger 1

In addition, the Etherphone GUI uses icons to represent callers and telephone lines. For example, Figure 4 of Zellweger 1 (reproduced below) includes a rolodex graphic to represent callees (upper left) and a telephone graphic (upper middle) and a graphic of a person talking on the phone (upper right) to represent telephone lines. When a call is placed to a callee in the rolodex, the name of the callee is associated with the active telephone line graphic (upper right). For example, the active telephone line graphic includes name of the callee and an image of a user speaking on the telephone.

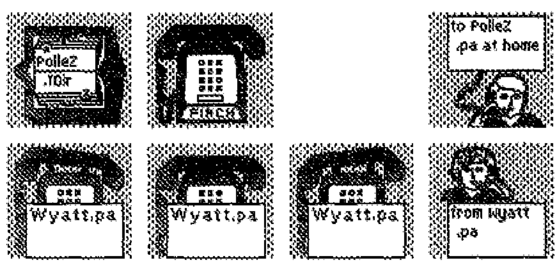


Figure 4. Etherphone system icons. The two icons at the upper left show a closed personal telephone directory and a Finch icon at rest. The icon at the upper right shows an outgoing call to Polle Zellweger's home (username PolleZ.pa). The four bottom icons show several stages of an incoming call from Doug Wyatt: the three left icons of the group are animated during ringing, while the right conversation icon is used after the call has been answered.

[25] Zellweger 1, page 4.

The Etherphone system also supports conference calling and call waiting. For example, Swinehart 1 describes how conference calls may be scheduled with other participants ("negotiated conference calls"). *See, e.g.*, Swinehart 1, page 3. In addition, using the Etherphone system, a user may receive and answer a call while already on an existing call. *See, e.g.*, Swinehart 1, page 2 (describing how users can place and receive other calls during a "background call").

The Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. At least one implementation of the Etherphone system used the Internet Protocol (IP) to support network communications. *See, e.g.*, Vin, page 77, Figure 5 (illustrating a "protocol stack and format" which includes internet protocol (IP) packets).

Consequently, Etherphone, which was not cited or discussed in the prosecution of the '704 patent, presents a SNQ of patentability because it, like Civanlar (which was antedated with a 37 C.F.R. § 1.131 declaration), discloses an address server (referred to as a "Voice Control Server" or "Telephone Control Server") for storing network protocol addresses usable by network nodes to establish point-to-point communications, as discussed in detail below. In view of the above, a reasonable examiner would consider Etherphone to be an important reference in deciding patentability. Accordingly, Etherphone presents a SNQ of patentability.

C. VocalChat User's Guide, VocalChat Readme File, VocalChat Networking Information, VocalChat Help File, VocalChat Troubleshooting Help File (collectively referred to as "VocalChat" or "the VocalChat references")

As mentioned above, the VocalChat system is described in VocalChat User's Guide, VocalChat Readme File, VocalChat Networking Information, VocalChat Help File, and VocalChat Troubleshooting Help File. As stated in the declaration of Alon Cohen, one of the co-founders of VocalTec, Ltd., included as **Exhibit L** with this reexamination:

1. VocalChat 1.01 Networking Information ("Networking Information"), attached as **Exhibit I** (referred to as "Exhibit A" in the declaration), was publicly distributed in 1994 as part of the VocalChat version 1.01 software, which was commercially released and on sale to the general public in 1994. The VocalChat version 1.01 software was sold as a boxed product, which included an electronic copy of the VocalChat 1.01 Networking Information document.

2. VocalChat 2.0 User's Guide ("User's Guide"), attached as **Exhibit G** (referred to as "Exhibit B" in the declaration), was publicly distributed in 1994 as part of the VocalChat version 2.0 software, which was commercially released and on sale to the general public in 1994. The VocalChat version 2.0 software was sold as a boxed product, which included a printed copy of the VocalChat 2.0 User's Guide.

3. The VocalChat Readme File ("Readme"), attached as **Exhibit H** (referred to as "Exhibit C" in the declaration), the VocalChat Troubleshooting Help File ("Troubleshooting Help File"), attached as **Exhibit K** (referred to as "Exhibit D" in the declaration), and VocalChat Information ("Help File"), attached as **Exhibit J** (referred to as "Exhibit E" in the declaration), are true and correct print outs of VocalChat version 2.02's README.TXT, TROUBLE.HLP, and INFO.HLP files, respectively. Electronic copies of these documents were publicly distributed in 1994 as part of the VocalChat version 2.02 software, which was commercially released and on sale to the general public as a boxed product in 1994.

VocalChat is a software-based telephone executed on a personal computer which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks.[26] In particular, as illustrated in the figures on pages 4 and 5 of the VocalChat User's Guide (reproduced below), computers with VocalChat installed connect directly to a Post Office directory on a server to register their current network protocol addresses, query the Post Office directory for the network protocol addresses of other on-line computers, and establish point-to-point communications with each other using the retrieved network protocol addresses.[27]

[26] *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers)

[27] *See, e.g.*, Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory.").

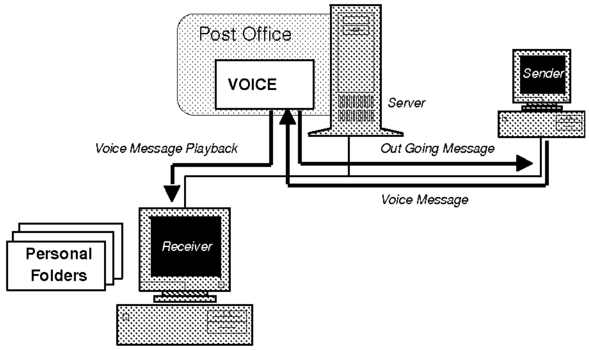


Figure from Page 5 of User's Guide

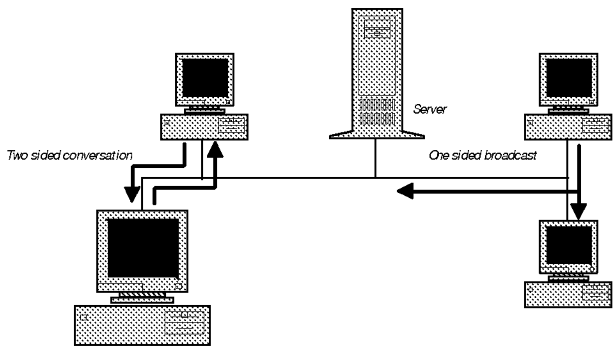


Figure from Page 4 of User's Guide

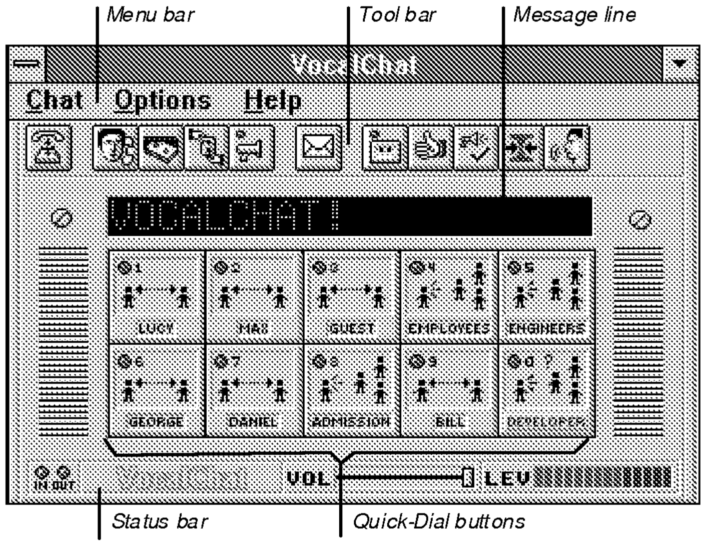
VocalChat may be implemented over a variety of network protocols including TCP/IP and NetBIOS. In a TCP/IP implementation, the Post Office directory includes a "Connection List" file (CONNLIST.VC) which contains the unique usernames and IP addresses of connected VocalChat users.[28] In the initial versions of VocalChat (versions 1.x), the Connection List file is called a "USERS file."[29] When the VocalChat client starts it transmits a user's unique username and IP address to the Connection List file.[30] User information maintained in the Connection List file is then made available to other VocalChat users, thereby enabling those users to locate and communicate with other VocalChat users.

---

[28] *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). For the purpose of this reexamination we have converted the VocalChat Help File (Exhibit J) and Troubleshooting Help File (Exhibit K) into PDF files and added page numbers to simplify navigation. Aside from the addition of page numbering, the content of the help file and troubleshooting help file has not been modified in any manner.

[29] See, e.g., VocalChat Network Information, page 10 ("When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address.")

[30] See, e.g., Help File, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses and a user name should be entered in the Setup for each user."). *See also* VocalChat Network Information, page 10.



As illustrated above, VocalChat also provides a graphical user interface (GUI) to establish point-to-point calls over the network. The user interface displays various buttons and interface elements, including a Call button, Quick Dial buttons, and an "Idle" icon, representing a temporarily disabled communication line, and a volume slider. Additionally, VocalChat includes a user interface window known as the "User List" to display a list of on-line users.[31] A VocalChat user may browse the list to find someone to call. Clicking on a username followed by the Call button establishes a point-to-point call with the selected user.[32] The VocalChat software queries the central directory database to determine whether the callee is on-line and, if so, the callee's network protocol address is returned to the VocalChat software (an IP address in the TCP/IP implementation).[33] The VocalChat software establishes a point-to-point call using the IP address.

Consequently, these VocalChat references, which were not cited or discussed in the prosecution of the '704 patent, present a SNQ of patentability because they, like Civanlar, show an address server (referred to as a "Post Office") for storing network protocol addresses usable by network nodes to establish point-to-point communications, as discussed in detail below. In view of the above a reasonable examiner would consider the VocalChat references to be

[31] *See, e.g.*, User's Guide, page 14 (illustrating an Address Book User List).
[32] *Id.*
[33] *See, e.g.*, Help File, page 22 (describing how VocalChat retrieves network addresses from the connection list file in a TCP/IP implementation).

important in deciding patentability. Accordingly, these VocalChat references present a SNQ of patentability.

D. RFC 1531

RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a Dynamic Host Configuration Protocol (DHCP) server. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). There are various benefits to using dynamic IP address assignment. For example, dynamically assigning IP addresses allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). Given that Claim 33 of the '704 patent explicitly requires "locating processes having dynamically assigned network protocol addresses" and that various other claims require the assignment of a network protocol address "following connection to the computer network" (see, e.g., Claim 1, 4, and 38), RFC 1531 would be considered important in deciding the question of patentability and accordingly presents a SNQ of patentability.

E. Vin

Vin is another prior art reference describing the Etherphone system which was published separately from Etherphone: Collected Papers 1987-1988 (May 1989) (collectively referred to herein as "Etherphone"). Vin describes many of the same features of the Etherphone system described in Etherphone and, in addition, describes how the Etherphone system may be used on a TCP/IP network. *See, e.g.*, Vin, page 77, Figure 5 (illustrating a "protocol stack and format" used in an Etherphone system which includes internet protocol (IP) packets). Consequently, Vin would be considered important in deciding the question of patentability and accordingly presents a SNQ of patentability, particularly with respect to Claim 32, which requires "Internet protocol" addresses.

F. Pinard

Pinard (Exhibit F) entitled "Human Machine Interface for Telephone Feature Invocation," issued on July 2, 1996 from an application filed on November 29, 1994. Pinard discloses that "[t]he ability to display icons on a computer display and to invoke commands by dragging an icon to another has long been known" in the prior art. (Col. 3, lines 15-17.) Pinard applies such a graphical user interface to the field of telephony. (Col. 1, lines 5-7.)

In Pinard "a method of providing information to a user unambiguously as to which persons are parties to a call" is described. (Col. 1, lines 55-57.) Specifically, Pinard shows how such information is represented graphically: "icons representing a subscriber's line associated with a local subscriber, the status of the line and [sic] associated with particular other subscribers to which calls are made or received are displayed in a manner that provides full information as to their status and the status of any call in progress, whether on line or being held, and whether it is a conference call or not." (Col. 2, lines 47-54.)

Also described is how call functions or processes can be displayed and then invoked using the graphical user interface: "The state of the call can be changed merely by dragging icons to particular locations on the display." (Col. 2, lines 54-55.) Specifically, Pinard describes:

> A method for calls to be made between parties, to be placed on hold, to be dropped from hold, to be conferenced or to be dropped from a conference with clear indication to the user which of the parties to any call are being dealt with. (Col. 1, lines 57-61.)

In Pinard with the graphical user interface on a personal computer, one has the ability to dial out and make and receive calls via a local area network (LAN). *See, e.g.*, Col. 1, line 64 - Col. 2, line 8; Col. 2, lines 38-41; Col. 3, lines 55-60; Col. 4; lines 1-3; and Figure 1. Pinard teaches that the described graphical user interface "can be used with any system in which a telephony application on a personal computer or [a] personal computer in conjunction with a server operates." (Col. 2, lines 41-45.) *See also* Col. 1, lines 60-62.

As show in Figure 2 below, Pinard discloses a personal computer with a display 11 running a telephone application software program. *See* Col. 4, lines 10-11. The program creates an icon 13 representing the caller ("Debbie" in the example) as well as an icon 15 representing a call setup process. (Col. 4, lines 11-18.)

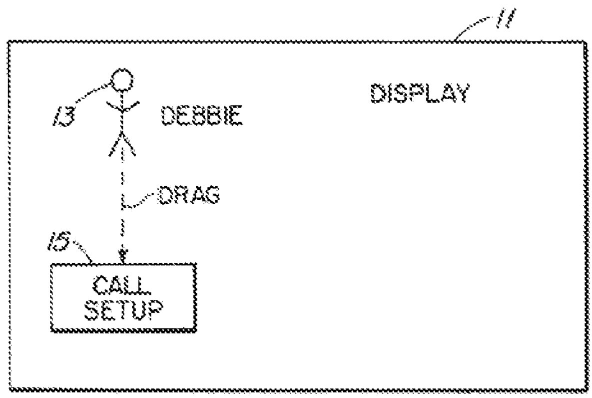


FIG. 2

Pinard discloses that the program uses the graphical user interface to permit the caller to place a call. (Col. 2, line 59 – Col. 3, line 9; Col. 3, lines10-14.) Figure 2 above and Figure 3 below are illustrative. By dragging the caller icon 13 ("Debbie") onto the call set up icon 15, the caller instructs the program that an outgoing call is to be made. (Col. 4, lines 19-21.) The program creates icons ("images of the faces of the persons listed in the directory") in directory 17 representing potential callees. *See, e.g.,* Col. 4, lines 22-31. When the user drags an icon from the directory 17 onto the call setup icon 15, the program retrieves and dials the corresponding callee's telephone number. *See, e.g.,* Col. 4, lines 38-48.

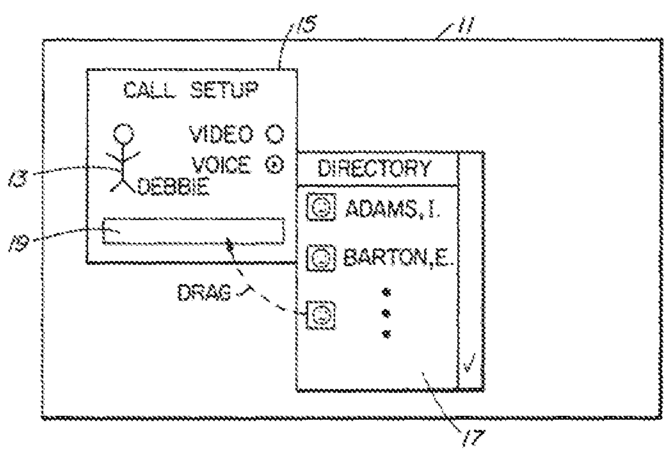


FIG. 3

As show in Figure 4 below, when caller "Debbie" and callee "John" are telephonically connected, the call setup icon 15 is transformed into a call icon 23, within whose borders the caller icon (stick figure labeled "Debbie") and the callee icon 21 ("John") are located. (Col. 4, lines 43-55.) This defined boundary signifies that a call is in progress. The program also creates a new call setup icon 24. (Col. 4, lines 50-41.) The program allows the caller to terminate a call by dragging the callee icon 21 ("John") into a trash basket icon 26. *See, e.g.,* Col. 5, lines 1-4.

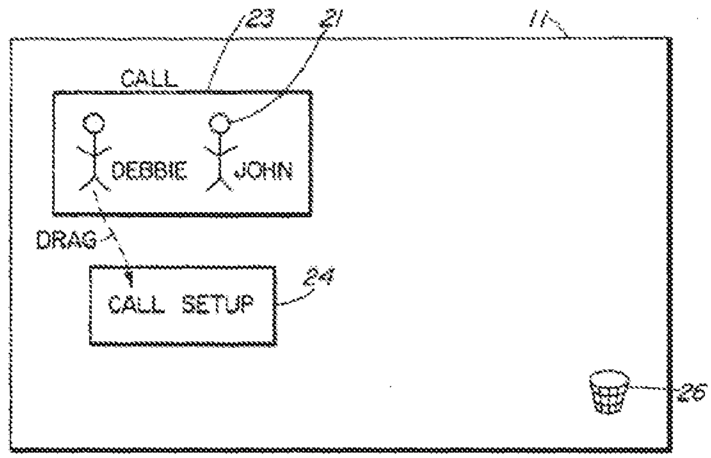


FIG. 4

The program of Pinard uses the graphical user interface to permit a caller to call a first callee on a first phone line and a second callee on a second phone line. *See, e.g.,* Col. 2, line 59 – Col. 3, line 9; Col. 3, lines 10-14. The program also uses a graphical interface to permit a caller to place on hold a first callee on one phone line and speak to a second callee on a second phone line (or vice versa). *See, e.g.*, Col. 2, line 59 – Col. 3, line 9; Col. 3, lines 10-14.

As shown in Figure 6 below, caller icon 13A ("Debbie") is ghosted in the first call icon 23 to indicate that the first phone line over which the first caller ("John," represented by icon 21) is connected is on hold. *See* Col. 5, lines 5-35; Col. 5, line 45 – Col. 6, line 5. Caller icon 13 ("Debbie") is solid in the second call icon 29 to indicate that the second phone line over which the second caller ("Mary," represented by icon 28) is connected is active. *See* Col. 5, lines 5-35; Col. 5, line 45 – Col. 6, line 5. Moreover, in the example of Figure 6, Debbie moves John from the first line (represented by call icon 23) to the second line (represented by call icon 29) by clicking and dragging John's icon 21, thereby creating a conference call between Debbie, Mary, and John. *See, e.g.*, Col. 5, lines 36-40.

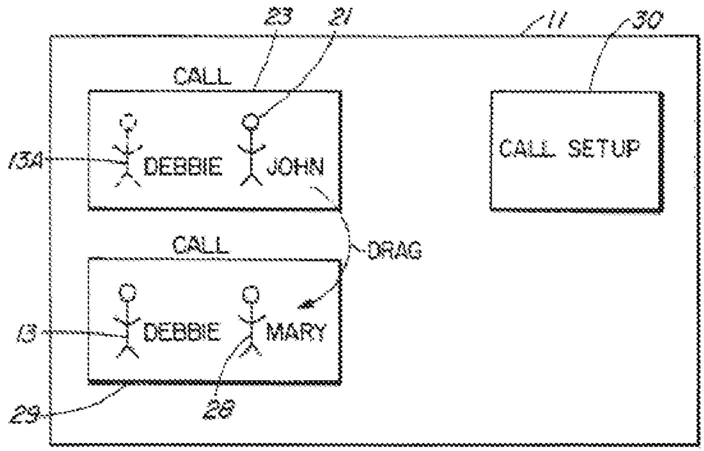


FIG. 6

A hard hold icon 39 of Pinard, shown below in Figure 12 allows the caller ("Debbie") to drag a callee icon 28 ("Mary") to the hard hold icon 39. This places the callee ("Mary") on hold. Other callers (represented by icons 41) may also be placed on hold. (Col. 6, lines 36-53.)

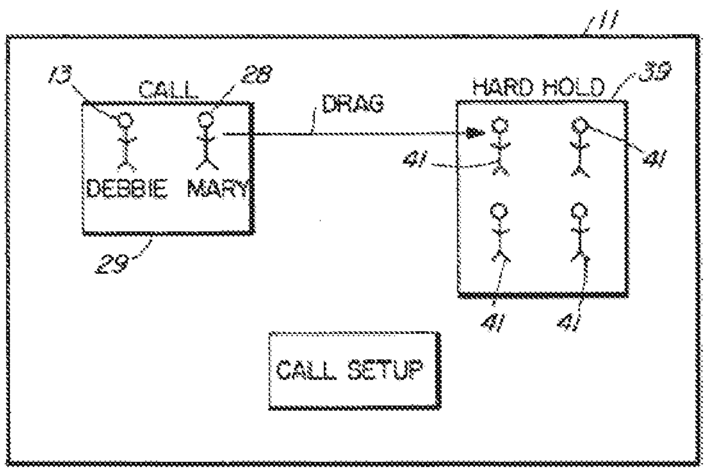


FIG. 12

Pinard discloses that the program uses the graphical user interface to permit a caller to conference a first callee and a second callee onto a single conference call. (Col. 2, line 59 – Col. 3, line 9; Col. 3, lines 10-14.) As shown in Figure 6 above, callee icon 21 ("John") is dragged onto call icon 29. As shown in Figure 7 below, this results in a conference call represented by conference icon 32 in whose borders caller icon 13 ("Debbie"), first callee icon 21 ("John"), and second callee icon 28 ("Mary") are located. (Col. 5, lines 36-44.)

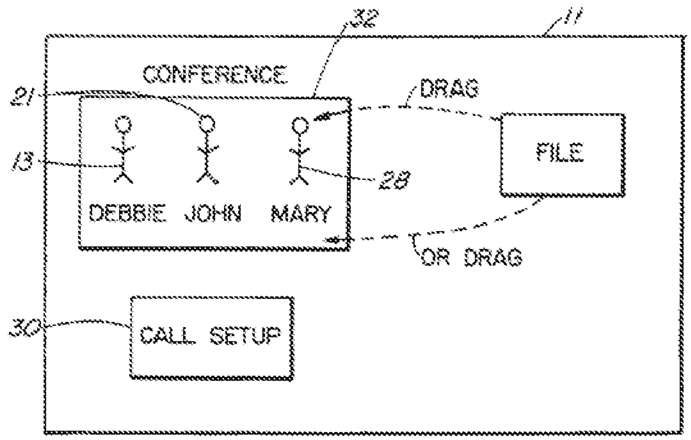


FIG. 7

Finally, Pinard teaches that "[u]sing similar principles, a person skilled in the art will now be able to provide unambiguous other features, such as call pickup, redial, speed call, callback, etc." (Col. 7, lines 48-52.) These and other features were well known to those of ordinary skill in the art insofar as they were prevalent on prior art telephones.

In summary, Pinard discloses graphical elements representing communication lines and callees that may be clicked and dragged to establish and terminate calls, set up conference calls, and place calls on hold, as recited in Claims 12-18 and 23-29 of the '704 patent. In view of the above, a reasonable examiner would consider Pinard to be an important reference in deciding patentability. Additionally, Pinard was not cited as a reference or discussed in the prosecution of the '704 Patent. Accordingly, Pinard presents a SNQ of patentability.

VI. DETAILED EXPLANATION OF THE PERTINENCY AND MANNER OF APPLYING THE PRIOR ART REFERENCES TO EVERY CLAIM FOR WHICH REEXAMINATION IS REQUESTED

As required under 37 C.F.R. § 1.510(b)(2), a detailed explanation of the pertinency and manner of applying the prior art references to the claims is provided. The following analysis is directed to prior art which was not cited during the prosecution of the claims of the '704 patent. Additional explanation of the pertinency and manner of applying the prior art references to the claims is provided in the claim charts at Exhibits M-O of this Request.

A. NetBIOS

1. <u>Anticipation Rejections</u>

¶ 1. The quotation of 35 U.S.C. §102 (b) forms the basis for the anticipation rejections which follow:

> A person shall be entitled to a patent unless…
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States.

¶ 2. Claims 1-7 and 32-44 are anticipated by Protocols for X/Open PC Interworking SMB, Version 2, THE OPEN GROUP (1992) ("NetBIOS"), which includes Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concept and Methods, RFC 1001 (March 1987) ("RFC 1001") and Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications, RFC 1002 (March 1987) ("RFC 1002").

¶ 3. During the Net2Phone Litigation, Net2Phone attempted to distinguish the claims of the '704 patent over NetBIOS. The court has yet to render an opinion on these arguments. As set forth in **Exhibit P** submitted with this reexamination, these arguments fail to distinguish the claims of the '704 patent over NetBIOS for a variety of reasons.

¶ 4. Neither NetBIOS nor its included RFC 1001/1002 were cited during the prosecution of the '704 patent. As delineated below there is a SNQ of patentability raised by NetBIOS. Below first the independent claims are set forth along with a discussion concerning the relevancy of NetBIOS to the SNQ of patentability. Then the dependent claims are set forth.

INDEPENDENT CLAIM 1

The preamble of Claim 1 reads, in pertinent part: "A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network . . ."

¶ 5. NetBIOS discloses a computer program product for use with a computer system which executes a "first process" and is operatively connectable to a "second process" and a server over a computer network. That NetBIOS discloses a computer program product for use with a computer system can be seen from 356 ("The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems."). In addition, NetBIOS describes that the computer systems (or "nodes") execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS discloses a "server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates a NetBIOS Name Server ("NBNS") coupled to point-to-point nodes ("P nodes") over the Internet.

¶ 6. In the pending litigation, Net2Phone argued that the term "server" should be defined broadly. Plaintiff Net2Phone, Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), page 3. More specifically, Net2Phone argued:

> Consistent with the use of the term 'server' in the specification, the claims do not refer to any specific server configuration. They simply require a 'server' (also referred to as a 'connection server,' 'address server,' or 'server process'). There is nothing in any of the claims that require that the server be in the form of a single computer with a centralized database, as defendants contend.

Id., page 4. Similarly, Net2Phone argued that "[a] server in a 'client/server system' can be implemented in any number of ways, from one to multiple computers, in one location or many, and from a single large computer acting as the server to a network of personal computers." Plaintiff Net2Phone Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007) (Exhibit W), page 7 (citing to the declaration of Professor Larry L. Peterson). Thus, under Net2Phone's interpretation, a "server" is not limited to any particular hardware or software configuration.

¶ 7. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Oct. 19, 2007) (Exhibit X), pages 2-9. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a "server." Under any interpretation, the NBNS described in NetBIOS is a "server."

Claim 1 requires "a computer usable medium having program code embodied in the medium."

¶ 8. NetBIOS applications are loaded into random access memory, which is a computer usable medium, and executed by a computer processor. NetBIOS describes that "[o]ne of the first implementations [of NetBIOS] was for personal computers running the PC-DOS and MS-DOS operating systems. It is possible to implement NetBIOS within other operating systems, or as processes which are, themselves, simply application programs as far as the host operating system is concerned." *id* at 359. NetBIOS further discloses that "typical use of NetBIOS is among independently-operated personal computers." *Id.* at 360.

Claim 1 further requires "program code for transmitting to the server a network protocol address received by the first process following connection to the computer network."

¶ 9. NetBIOS discloses program code executed on a node that transmits its name and IP address to the NBNS. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *id.* 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *id.* 431. *See also* NetBIOS at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name.");

id. at 461-464 (disclosing program code for the P-node name registration process) and *id.* 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes.

Claim 1 also requires "program code for transmitting, to the server, a query as to whether the second process is connected to the computer network."

¶ 10. As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *id.* 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id.* *See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446.

Claim 1 further requires "program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network."

¶ 11. NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. NetBIOS at 389. The NetBIOS Name Server maintains entries only for "active" (on-line) nodes, i.e., nodes that have an on-line status with respect to the computer network. *Id.* at 446 (each entry "represents an active name"). Thus, if the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE").

¶ 12. NetBIOS discloses a number of mechanisms to track the online status of nodes. Entries for off-line nodes are "removed" through the use of log-out messages and timers. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *id.* 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include a "Name Refresh" mechanism, whereby each point-to-point node "is responsible for sending periodic Name Request Requests" to the Name Server, which allows the Name Server "to detect if a P [] node has 'silently' gone down, so that names held by that node can be purged from the data base." *Id.* at 394; *see id.* at 378 ("The NBNS will consider a name to have been silently released if the end-node fails to send a name refresh message" prior to the expiration of a predetermined interval.). Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Thus, by design, only logged-in nodes are registered with the NBNS. *See, e.g., id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g., id.* at 389.

¶ 13. In Claim Construction Briefs filed in the pending litigation, the patentee argued that the term

> ‘connected’ means ‘logged on,’ and *vice versa* . . . To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect. While Net2Phone’s invention endeavors to identify accurately who is on line, it is not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user’s Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person’s status— *e.g.*, setting a default value of two hours, after which the server assumes that a party has gone off-line if it has not heard from her. *See* ’704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, “the on-line status information stored in the database is ***relatively*** *current*.” *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term “connected” (or “on-line”) is going to be modified at all, it should be modified to say “*relatively* currently connected,” because that is what the patents actually say.

Plaintiff Net2Phone Inc.’s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), pages 24-25. Thus, under Net2Phone’s interpretation, the information retained in the “server” as to which processes are “connected to the computer network” or “online” may be imperfect. As described above, while the server “endeavors to identify accurately who is on line, it is not possible to achieve perfection.” *Id.* NetBIOS employs similar techniques as NBNS entries for off-line nodes are removed through the use of log-out messages and timers.

¶ 14. Once again, the requestor of the present reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Exhibit X), pages 12-14. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the ‘704 patent which require a process to be “connected to” the computer network or “on-line.” Under any interpretation, a first NetBIOS process receives the network protocol address of a second NetBIOS process from the NBNS when the second NetBIOS process is “connected to the computer network.”

Claim 1 also requires "program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network."

¶ 15. Once the node seeking to initiate the communication has obtained from the NBNS the IP address for the node to receive the communication, a point-to-point communication is established between the nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>." NetBIOS at 397 (emphasis added). *See also id.* at 401:

> This first diagram shows the sequence of network events used to successfully establish a session without retargeting by the listener. The TCP connection is first established with the well-known NetBIOS session service TCP port, SSN_SRVC_TCP_PORT. The caller then sends a SESSION REQUEST packet over the TCP connection requesting a session with the listener. The SESSION REQUEST contains the caller's name and the listener's name. The listener responds with a POSITIVE SESSION RESPONSE informing the caller this TCP connection is accepted as the connection for the data transfer phase of the session.

See also id. at 398-400 ("16.1: Overview of NetBIOS Session Service"), 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are fullduplex, sequenced, and reliable. Data is organized into messages."). In sum, NetBIOS discloses all of the elements of, and hence anticipates, claim 1 of the '704 Patent.

INDEPENDENT CLAIM 2

Claim 2 claims "[a]n apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising: a processor; a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network."

¶ 16. NetBIOS describes a NBNS which is coupled to M and P nodes over the Internet (or other network). See, e.g., NetBIOS, page 371 (illustrating a NBNS) coupled to point-to-point nodes ("P nodes") over the Internet). The NBNS executes software using a processor and is inherently coupled to the Internet (or other network) via a network interface. See, e.g., id., page 359 ("[o]ne of the first implementations [of NetBIOS] was for personal computers running the PC-DOS and MS-DOS operating systems. It is possible to implement NetBIOS within other operating systems, or as processes which are, themselves, simply application programs as far as the host operating system is concerned."); id. at 357 ("NetBIOS is the foundation of a large body of existing applications.").

Claim 2 requires "a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of processes, each network protocol address stored in the memory following connection of a respective process to the computer network."

¶ 17. As described above, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). This registration must inherently be stored in a "memory." A NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." NetBIOS, page 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby stores in a memory a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. The P and M nodes execute software which is inherently a computer-implemented "process." *See, e.g., id.* at 356 ("NetBIOS defines a software interface . . . .); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications.").

Claim 2 also requires "means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process."

¶ 18. In Claim Construction Briefs filed in the pending litigation, the patentee argued as follows with respect to the term "on-line status":

> To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect. While Net2Phone's invention endeavors to identify accurately who is on line, it is

> not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user's Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person's status— *e.g.*, setting a default value of two hours, after which the server assumes that a party has gone off-line if it has not heard from her. *See* '704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, "the on-line status information stored in the database is ***relatively*** *current*." *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term "connected" (or "on-line") is going to be modified at all, it should be modified to say "*relatively* currently connected," because that is what the patents actually say.

Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007), pages 24-25. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc, pages 12-14.

¶ 19. NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. NetBIOS at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Thus, by design, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g., id.* at 389.

INDEPENDENT CLAIM 4

Claim 4 claims "A method for enabling point-to-point communication between a first process and a second process over a computer network."

¶ 20. NetBIOS describes a method for enabling point-to-point communication between a first process (on a first node) and a second process (on a second node) over a computer network. As discussed above, the NBNS is used to resolve IP addresses of point-to-point end-nodes to facilitate point-to-point communications between such nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name. . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>." *Id.* at 397 (emphasis added).

Claim 4 requires "receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network."

¶ 21. NetBIOS describes the NBNS receiving and storing the names and IP addresses of processes in its memory. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See* NetBIOS at page 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Consequently, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claim 4 also requires "receiving a query from the first process to determine the on-line status of the second process."

¶ 22. As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *Id.* at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id. See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446.

Claim 4 also requires "determining the on-line status of the second process."

¶ 23. The requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. For instance, when the NBNS receives a query for a target node's IP address, it performs a look search in its directory database for the target's current IP address. *Id.* at 389 (The NBNS "answers queries from a P node with a list of IP address and other information for" the target name.). The NBNS determines an end-node with the target name is currently registered in its database, and hence is deemed to be on-line. *See id.* at 376 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). As described above, the NBNS employs various mechanisms for determining the on-line status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database.

Finally, Claim 4 requires "transmitting an indication of the on-line status of the second process to the first process over the computer network."

¶ 24. If the end-node with the target name is currently registered in the NBNS database, the NBNS responds with a positive name query response. *See, e.g., id.* at 389 (The NBNS "answers queries from a P node with a list of IP address and other information for" the target name.); *id.* at 440 ("NAME QUERY REQUEST"), *id.* at 441 ("POSITIVE NAME QUERY RESPONSE"), *id.* at 464-465 ("P-Node Find Name Procedure"). A positive name query response includes the IP address for the target end-node, *id.* at 441, which is an indication that the target node has a positive on-line status. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). A negative name query response from the NBNS may include a message that "[t]he name requested does not exist" in the NBNS database, which is an indication that the target node has an off-line status. *See, e.g., id.* at 442; *see also id.* at 484.

INDEPENDENT CLAIM 32

Claim 32 recites "[a] method of locating a process over a computer network comprising the steps of: a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network."

¶ 25. NetBIOS describes the NBNS maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See* NetBIOS, page 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to an NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); NetBIOS at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. NetBIOS further discloses that this list may be accessible over the Internet. For example, the Figure on page 371 of NetBIOS clearly shows a NBNS and multiple client nodes connected over the "INTERNET." In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." NetBIOS at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g., id.* at 389.

Claim 32 also requires "in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process."

¶ 26. A NBNS "answers queries from a P node with a list of IP address and other information for" the target name. NetBIOS at 389. If the NBNS has an entry in its database for the target name identified by the requesting node, the NBNS returns a positive name query response to the requesting node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 389. *See also id.* at 441 ("POSITIVE NAME QUERY RESPONSE").

INDEPENDENT CLAIM 33

Claim 33 claims "[a] method for locating processes having dynamically assigned network protocol addresses over a computer network."

¶ 27. Because IP addresses were known to be dynamically assigned on TCP/IP networks such as the Internet, this feature is inherent in NetBIOS. *See, e.g.*, RFC 1531, Dynamic Host Configuration Protocol (1993), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). One of ordinary skill in the art would understand that the use and implementation of a NBNS enables locating point-to-point nodes that have dynamically assigned network addresses. Alternatively, it would have been obvious to combine NetBIOS with other references such as RFC 1531 which describe the use of dynamically assigned IP addresses.

Claim 33 also requires "maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network."

¶ 28. NetBIOS describes the NBNS maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a

NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); NetBIOS at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Thus, by design, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claim 33 also requires "in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process."

¶ 29. A NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has an entry in its database for the target name identified by the requesting node, the NBNS returns a positive name query response to the requesting node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 389. *See also id.* at 441 ("POSITIVE NAME QUERY RESPONSE").

INDEPENDENT CLAIM 38

Claim 38 claims "[a] computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium."

¶ 30. NetBIOS discloses a computer program product for use with a computer system. *See id.* at 356 ("The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems."). In addition, NetBIOS describes that the computer systems (or "nodes") execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS also discloses a "server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates the NBNS coupled to point-to-point nodes ("P nodes") over the Internet. NetBIOS applications may be loaded into random access memory, which is a computer usable medium, and executed by a computer processor. NetBIOS describes that "[o]ne of the first implementations [of NetBIOS] was for personal computers running the PC-DOS and MS-DOS operating systems. It is possible to implement NetBIOS within other operating systems, or as processes which are, themselves, simply application programs as far as the host operating system is concerned." *Id.* at 359.

NetBIOS further discloses that "typical use of NetBIOS is among independently-operated personal computers." *Id.* at 360.

Claim 38 requires "program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network."

¶ 31. NetBIOS describes the NBNS maintaining a network accessible compilation of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to the NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claim 38 further requires "program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process."

¶ 32. The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has an entry in its database for the target name identified by the requesting node, the NBNS returns a positive name query response to the requesting node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 389. *See also id.* at 441 ("POSITIVE NAME QUERY RESPONSE").

INDEPENDENT CLAIM 43

Claim 43 claims "[a] computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein."

¶ 33. NetBIOS discloses a computer program product for use with a computer system which executes a "first process" and is operatively connectable to a "second process" and a server over a computer network. First, NetBIOS discloses a computer program product for use with a computer system. *See id.* at 356 ("The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems."). In addition, NetBIOS describes that the computer systems (or "nodes") execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS also discloses a "server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates the NBNS coupled to point-to-point nodes ("P nodes") over the Internet.

Claim 43 further requires "program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network."

¶ 34. NetBIOS describes a "directory database" for storing network addresses of on-line processes. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." NetBIOS *id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a directory database of names and corresponding IP addresses of point-to-point and mixed end-nodes.

Claim 43 also requires "program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network."

¶ 35. Once the node seeking to initiate the communication has obtained from the NBNS the IP address for the node to receive the communication, a point-to-point communication is established between the nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>." *Id.* at 397 (emphasis added). *See also id.* at 401:

> This first diagram shows the sequence of network events used to successfully establish a session without retargeting by the listener. The TCP connection is first established with the well-known NetBIOS session service TCP port, SSN_SRVC_TCP_PORT. The caller then sends a SESSION REQUEST packet over the TCP connection requesting a session with the listener. The SESSION REQUEST contains the caller's name and the listener's name. The listener responds with a POSITIVE SESSION RESPONSE informing the caller this TCP connection is accepted as the connection for the data transfer phase of the session.

See also id. at 398-400 ("16.1: Overview of NetBIOS Session Service"), 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are fullduplex, sequenced, and reliable. Data is organized into messages."). In sum, NetBIOS discloses all of the elements of, and hence anticipates, claim 43 of the '704 Patent.

INDEPENDENT CLAIM 44

The preamble of Claim 44 reads: "In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of."

¶ 36. NetBIOS describes that the various network "nodes" execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS also discloses an "address server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of

NetBIOS illustrates the NBNS coupled to point-to-point nodes ("P nodes") over the Internet. The NBNS is an "address server" because it stores names and IP addresses of nodes. *See, e.g., id.* at 367 describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."

Claim 44 also requires "following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network."

¶ 37. In NetBIOS, each node forwards its IP address to the NBNS following connection to the computer network. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). More specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 describing how the NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names.

Claim 44 also requires "querying the address server as to whether the second process is connected to the computer network."

¶ 38. As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *Id.* at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id. See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446.

Claim 44 further requires "receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network."

¶ 39. NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Consequently, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claim 44 further requires "in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network."

¶ 40. Once the node seeking to initiate the communication has obtained from the NBNS the IP address for the node to receive the communication, a point-to-point communication is established between the nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>." *Id.* at 397 (emphasis added). *See also id.* at 401:

> This first diagram shows the sequence of network events used to successfully establish a session without retargeting by the listener. The TCP connection is first established with the well-known NetBIOS session service TCP port, SSN_SRVC_TCP_PORT. The caller then sends a SESSION REQUEST packet over the TCP connection requesting a session with the listener. The SESSION REQUEST contains the caller's name and the listener's name. The listener responds with a POSITIVE SESSION RESPONSE informing the caller this TCP connection is accepted as the connection for the data transfer phase of the session.

See also id. at 398-400 ("16.1: Overview of NetBIOS Session Service"); 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are fullduplex, sequenced, and reliable. Data is organized into messages.").

DEPENDENT CLAIMS 3, 5-7, 14, 34-37, AND 39-42

Claim 3 of the '704 patent requires "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory."

¶ 41. The NBNS includes a timer for time-stamping name/IP address entries. For example, "[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure." *Id.* at 382. Similarly, as described in NetBIOS:

> If an end-node holds any names that have finite time-to-live values, then that node must periodically send a status report to the NBNS. Each name is reported using the NAME REFRESH REQUEST packet. These status reports restart the timers of both the NBNS and the reporting node. However, the only timers which are

restarted are those associated with the name found in the status report. Timers on other names are not affected. *Id.*

Claim 5 of the '704 patent requires "searching the computer memory for an entry relating the second process; and retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process."

¶ 42. These features are inherent in the NetBIOS. In order for the NBNS to identify the IP address of a process in response to a query, it must inherently search its memory for an entry related to the process. For example, NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claim 6 of the '704 patent requires "transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network."

¶ 43. NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). For these reasons, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claim 7 requires "generating an off-line message when the second process is determined in step C to have a negative on-line status with respect to the computer network; and transmitting the off-line message to the first process."

¶ 44. The requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389 (illustrating a negative response "if the NBNS has no information about the name."). This negative response is an off-line message, which is generated when a node is determined to have a negative on-line status. As mentioned above, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder.").

Claim 34 requires "modifying the compilation of entries."

¶ 45. NBNSs periodically modify their compilation of name/address entries in response to various conditions. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS." NetBIOS at 377. Upon going off-line, the node sends a log-out message to the NBNS, which then deletes the node's name/address entry from the compilation, thereby modifying its compilation of entries. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). The compilation of entries may also be modified when an end node does not send a refresh message to its NBNS within a determined period of time, which may result in the deletion of its name/address entry from the NBNS's compilation of entries. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "NAME CHALLENGE" operation), 380 (describing "NODE STATUS REQUEST" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."); 384 (describing "OVERWRITE" operation).

Claim 35 requires "adding an entry to the compilation upon the occurrence of a predetermined event."

¶ 46. NBNS add entries to their compilation of name/address entries in response to various circumstances. For example, NetBIOS discloses that registrations adds the names (distinguishing identifiers) and current IP addresses of end-nodes to the NBNS' compilation of name/address entries. *Id.* at 385. *See also* id. at 431-432; id. at 367 (The NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 461-464 (P-node name registration process) and 480-482 (NBNS incoming packet processing for name registration). Registration thereby adds names and corresponding IP addresses of end-nodes to the NBNS' compilation of name/address entries.

Claim 36 requires that "the predetermined event comprises notification by a user process of an assigned network protocol address."

¶ 47. User processes executed on P and M nodes "notify" the NBNS of their assigned names and IP addresses. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). More specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes.

Claim 37 requires "deleting an entry from the compilation upon the occurrence of a predetermined event."

¶ 48. The NBNSs periodically delete a name/address entry from their compilation upon the occurrence of a predetermined event. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS." *Id.* at 377. Upon going off-line, the node sends a log-out message to the NBNS. Upon the occurrence of this predetermined event, the NBNS deletes the node's name/address entry from the compilation of entries. NetBIOS also discloses that a name/address entry may be deleted if an end node does not send a "refresh" message to its NBNS within a predetermined period of time. *See id.* at 448, 452-453, 464-465. *See also id.* at 448 (describing "name challenge" operation), 450 (describing "Node Status Request" operation), 451 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 453 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS.").

Claim 39 requires "program code configured to modify the compilation of entries."

¶ 49. NBNSs periodically modify their compilation of name/address entries in response to various conditions. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS." *Id.* at 377.

Upon going off-line, the node sends a log-out message to the NBNS, which then deletes the node's name/address entry from the compilation, thereby modifying its compilation of entries. *See also id.* at 393-394 (describing “NAME RELEASE TRANSACTIONS”). The compilation of entries may also be modified when an end node does not send a refresh message to its NBNS within a determined period of time, which may result in the deletion of its name/address entry from the NBNS’s compilation of entries. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing “NAME CHALLENGE” operation), 380 (describing “NODE STATUS REQUEST” operation), 381 (“15.1.7 CONSISTENCY OF THE NBNS DATA BASE”), 383 (“A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS.”); 384 (describing “OVERWRITE” operation).

Claim 40 requires “program code configured to add an entry to the compilation upon the occurrence of a predetermined event.”

¶ 50. The NBNSs add entries to their compilation of name/address entries in response to various circumstances. For example, NetBIOS discloses that registration adds the names (distinguishing identifiers) and current IP addresses of end-nodes to the NBNS’ compilation of name/address entries. *Id.* at 385. *See also* id. at 431-432; id. at 367 (NBNS acts as a “‘bulletin board’ on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names.”); *id.* at 461-464 (P-node name registration process) and 480-482 (NBNS incoming packet processing for name registration). Registration thereby adds names and corresponding IP addresses of end-nodes to the NBNS’ compilation of name/address entries.

Claim 41 requires that the “predetermined event comprises notification by a process of an assigned network protocol address.”

¶ 51. User processes executed on P and M nodes “notify” the NBNS of their assigned names and IP addresses. For example, to engage in NetBIOS communications, a point-to-point (“P”) or mixed (“M”) node must register with a NBNS by transmitting a notice of the end node’s name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the “P-NODE REGISTRATION PROCESS”). More specifically, a NetBIOS “Name Registration Request” sent by an M or P node to a NBNS includes the field “NB_ADDRESS,” which is the “IP address of the name's owner.” *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); NetBIOS *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes.

Claim 42 requires "program code configured to delete an entry from the compilation upon the occurrence of a predetermined event."

¶ 52. The NBNSs periodically delete a name/address entry from their compilation upon the occurrence of a predetermined event. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS." *Id.* at 377. That is, upon going off-line, the node sends a log-out message to the NBNS. Upon the occurrence of this predetermined event, the NBNS deletes the node's name/address entry from the compilation of entries. NetBIOS also discloses that a name/address entry may be deleted if an end node does not send a "refresh" message to its NBNS within a predetermined period of time. *See id.* at 448, 452-453, 464-465. *See also id.* at 448 (describing "name challenge" operation), 450 (describing "Node Status Request" operation), 451 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 453 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS.").

2. <u>Obviousness Rejections</u>

¶ 53. The following is a quotation of 35 U.S.C. §103 (a) which forms the basis for the following obviousness rejections:

> 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.

(i) NetBIOS in view of RFC 1531

¶ 54. Claim 33 explicitly states that the network protocol address of the client computer system is "dynamically assigned." *See* Claim 33 ("A method for locating processes having dynamically assigned network protocol addresses over a computer network"). Other independent claims state, more generally, that the network protocol address is assigned or transmitted to the database following the connection of the computer to the computer network. *See* Claim 1 ("transmitting to the server a network protocol address received by the first process following connection to the computer network"); Claim 2 ("each network protocol address stored in the memory following connection of a respective process to the computer network"); Claim 4 ("the network protocol addresses received following connection of the respective process to the computer network"); Claim 32 ("the Internet Protocol address added to the list following connection of the process to the computer network"); Claim 38 ("the network protocol address of the corresponding process assigned to the process upon connection to the computer network"); Claim 43 ("the network protocol address of each respective process forwarded to the database following connection to the computer network"); Claim 44 ("following connection of the first process to the computer network forwarding to the address server a network protocol address").

¶ 55. As described above, the NetBIOS reference inherently describes these features. For example, on many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in the NetBIOS reference.

¶ 56. Alternatively, Claims 1-7 and 32-44 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over the NetBIOS reference in view of RFC 1531, which describes how TCP/IP addresses were dynamically assigned. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks).

(ii) Motivation to Combine NetBIOS with RFC 1531

¶ 57. A motivation to combine the NetBIOS reference with RFC 1531 exists because the NetBIOS reference describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine the NetBIOS reference with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of the NetBIOS reference would frequently have their IP addresses dynamically assigned.

(iii) NetBIOS in view of Pinard

INDEPENDENT CLAIM 10

The preamble to Claim 10 reads: "In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network . . ."

¶ 58. As discussed above, NetBIOS discloses a method of establishing a point-to-point communication link between nodes over a computer network such as a local network or the Internet. "NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only directed (point-to-point) communications." *Id.* at 397 (emphasis added). *See also id.* at 361 (describing how a call to a named callee process is used to "[i]nitiate a session with a process that is listening under the specified name. The calling entity must indicate both a calling name (properly registered to the caller) and a called name) (emphasis added); *id.* at 359 ("NetBIOS applications employ NetBIOS mechanisms to locate resources, establish connections, send and receive data with an application peer, and terminate connections."); *id.* at 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are full-duplex, sequenced, and reliable. Data is organized into messages."). Applications which utilize NetBIOS application services inherently include "user interfaces." For example, "NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems." *Id.* at 356. The IBM PC included a text-based user interface known as PC-DOS. *See, e.g.*, id. (it is expected that on computers operating under the PC-DOS and MS-DOS operating systems that the existing NetBIOS interface will be preserved by implementers).

Claim 10 requires "providing a user interface element representing a first communication line."

¶ 59. Pinard discloses a user interface element representing a first communication line. For example, Figure 6 of Pinard illustrates a first call icon 23 which represents a first communication line and a second call icon 29 which represents a second communication line. In the example shown in Figure 6, the first call icon 23 represents a telephone call between "Debbie" and "John" and the second call icon 29 represents a telephone call between "Debbie" and "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 10 also requires "providing a user interface element representing a first callee process."

¶ 60. Pinard describes "a user interface element representing a first callee process." In the example shown in Figure 6 of Pinard, a first user interface element 21 is shown for the callee named "John" and a second user interface element is shown for the callee named "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 10 further requires "establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line."

¶ 61. As described above, NetBIOS describes establishing a point-to-point communication link between nodes. *See, e.g.*, NetBIOS at 397 ("NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only</u>

directed (point-to-point) communications.") (emphasis added). Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

INDEPENDENT CLAIM 21

Claim 21 claims "[a] computer program product for use with a computer system comprising: a computer usable medium having program code embodied in the medium."

¶ 62. NetBIOS is implemented in software which is program code stored on a computer usable medium. *See, e.g.,* NetBIOS at 356 ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available.").

Claim 21 requires "establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network."

¶ 63. As discussed above, NetBIOS discloses a method of establishing a point-to-point communication link between a caller process and a callee process over a computer network such as a local network or the Internet. "NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only directed (point-to-point) communications." NetBIOS at 397 (emphasis added). *See also id.* at 361 (describing how a call to a named callee process is used to "[i]nitiate a session with a process that is listening under the specified name. The calling entity must indicate both a calling name (properly registered to the caller) and a called name) (emphasis added); *id.* at 359 ("NetBIOS applications employ NetBIOS mechanisms to locate resources, establish connections, send and receive data with an application peer, and terminate connections."); *id.* at 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are full-duplex, sequenced, and reliable. Data is organized into messages."). NetBIOS applications include "user interfaces," for example, "NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems." *Id.* at 356. The IBM PC included a text-based user interface known as PC-DOS. *See, e.g.*, id. (it is expected that on computers operating under the PC-DOS and MS-DOS operating systems that the existing NetBIOS interface will be preserved by implementers).

Claim 21 also requires "program code for generating an element representing a first communication line."

¶ 64. Pinard discloses program code for generating an element representing a first communication line. For example, Figure 6 of Pinard illustrates a first call icon 23 which represents a first communication line and a second call icon 29 which represents a second communication line. In the example shown in Figure 6, the first call icon 23 represents a telephone call between "Debbie" and "John" and the second call icon 29 represents a telephone call between "Debbie" and "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 21 also requires "program code for generating an element representing a first callee process."

¶ 65. Pinard describes program code for generating an element representing a first callee process. In the example shown in Figure 6 of Pinard, a first user interface element 21 is shown for the callee named "John" and a second user interface element is shown for the callee named "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 21 also requires "program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process."

¶ 66. As described above, NetBIOS describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, NetBIOS at 397

("NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>.") (emphasis added). Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

DEPENDENT CLAIMS 11-20 AND 22-31

Claim 11 requires "querying the server as to the on-line status of the first callee process; and receiving a network protocol address of the first callee process over the computer network from the server."

¶ 67. As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *Id.* at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id. See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446. If the end-node with the target name is currently registered in the NBNS database, the NBNS responds with a positive name query response. *See, e.g., id.* at 389 (The NBNS "answers queries from a P node with a list of IP address and other information for" the target name.).

Claim 12 requires "providing an element representing a second communication line."

¶ 68. The graphical user interface described in Pinard provides an element representing a second communication line. For example, call icons 23 and 29 representing two communication lines are shown in Figure 6 of Pinard. *See* Pinard, Col. 5, lines 31-40, Figure 6 ("Now there are clearly two calls in progress . . .").

Claims 13 and 24 require "terminating the point-to-point communication link from the caller process to the first callee process, in response to the user disassociating the element representing the first callee process from the element representing the first communication line" and "program code, responsive to the user disassociating the element representing the first callee process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process," respectively.

¶ 69. Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.

Claim 13 and 24 further require "establishing a different point-to-point communication link from the caller process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line" and "program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process," respectively.

¶ 70. In Figure 6 of Pinard, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, Col. 4, lines 22-31.

> **Claims 14 and 25 require "providing a user interface element representing a second callee process; and . . . establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line" and "program code for generating an element representing a second callee process; and program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process," respectively.**

¶ 71. In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." See Pinard, Col. 5, lines 31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

> **Claims 15 and 26 require "removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line" and "program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link," respectively.**

¶ 72. In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, Col. 6, lines 14-15.

> **Claims 16 and 27 require "providing a user interface element representing a communication line having a temporarily disabled status" and "program code for generating an element representing a communication line having a temporarily disabled status," respectively.**

¶ 73. Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status.

For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 16 and 27 also require "temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status" and "program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process," respectively.

¶ 74. In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 17 and 28 require that "the element provided in step D represents a communication line on hold status" and "the communication line having a temporarily disabled status comprises a communication line on hold status," respectively.

¶ 75. In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claim 19 requires "wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface."

¶ 76. Pinard discloses a graphical user interface on a visual display which allows the caller to control the operation of the telephone. *See, e.g.*, Pinard, Figures 2-16 and Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claim 20 requires "wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface."

¶ 77. As described above, Pinard discloses that a point-to-point communication link is established in response to a user associating a graphic element representing a callee process with a graphic element representing a communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

Claim 22 requires "program code for querying the server as to the on-line status of the first callee process; and program code for receiving a network protocol address of the first callee process over the computer network from the server."

¶ 78. As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." NetBIOS at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id.* *See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446. If the end-node with the target name is currently registered in the NBNS database, the NBNS responds with a positive name query response. *See, e.g., id.* at 389 (The NBNS "answers queries from a P node with a list of IP address and other information for" the target name.).

Claim 23 requires "program code for generating an element representing a second communication line."

¶ 79. Pinard describes an element representing a second communication line. For example, Figure 6 of Pinard illustrates a first element (23) representing a first communication line and a second element (29) representing a second communication line.

Claim 30 requires that the "computer system further comprises a visual display and the user interface comprises a graphic user interface."

¶ 80. Figures 2-16 of Pinard illustrate a graphical user interface for managing telephone calls (which is inherently rendered on a "visual display").

Claim 31 requires that "the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises: program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process."

¶ 81. As described above, Pinard discloses that a point-to-point communication link is established in response to a user associating a graphic element representing the first callee process with a graphic element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

(iv) Motivation to Combine NetBIOS and Pinard

¶ 82. A motivation to combine NetBIOS and Pinard exists considering the problem sought to be solved. The Pinard reference relates to the field of computer-implemented telephony, and in particular to a computer-implemented method of indicating the status of various calls, to a user. *See* Pinard, Col. 1, lines 5-7. Pinard explicitly states that the invention "can be used with any system in which a . . . personal computer in conjunction with a server operates." Pinard, col. 2, lines 43-46. Given that NetBIOS describes networking software executed on personal computers (such as IBM PCs), and in particular describes a system in which a "personal computer" operates in conjunction with a "server" (i.e., a NBNS), one of ordinary skill in the art would have recognized that the particular design choices reflected in the graphical user interface of Pinard could readily be implemented within the context of the systems described in NetBIOS.

(v) NetBIOS in view of Pinard and Further in View of VocalChat User's Guide

¶ 83. Claim 18 and 29 require that "the element provided in step D represents a communication line on mute status" and "the communication line having a temporarily disabled status comprises a communication line on mute status," respectively. As described above, NetBIOS and Pinard describe all of the elements of Claim 18 and 29 except for a "communication line on mute status." VocalChat User's Guide describes a "communication line on mute status." As described in the User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.

(vi) Motivation to Combine VocalChat User's Guide with NetBIOS and Pinard

¶ 84. A motivation to combine the VocalChat User's Guide with NetBIOS and Pinard exists considering the problem sought to be solved. All three references relate to the field of communications over a computer network, and the VocalChat User's Guide and Pinard relate to the use of a computer system to implement telephony features. *See, e.g.,* Pinard, Col. 1, lines 5-7. One of ordinary skill in the art would have recognized the need for a "mute" function to enable users to mute the audio of a call as needed.

B. Etherphone

1. **Anticipation Rejections**

¶ 85. The quotation of 35 U.S.C. §102 (b) forms the basis for the anticipation rejections which follow:

> A person shall be entitled to a patent unless…
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States.

¶ 86. Claims 1-2, 4-7, 10-12, 14, 17-23, 25, and 30-44 are anticipated by *Etherphone: Collected Papers 1987-1988* (May 1989) (hereinafter "Etherphone") which published, **as a single publication**, with the following papers:

a. Polle T. Zellweger, et al., *An Overview of the Etherphone System and its Applications*, IEEE CONFERENCE ON COMPUTER WORKSTATIONS (March 1988), 160-168 (hereinafter "Zellweger 1").

b. Daniel C. Swinehart, *Telephone Management in the Etherphone System*, PROCEEDINGS OF THE IEEE/IEICE GLOBAL TELECOMMUNICATIONS CONFERENCE (November 1987), 1176-1180 (hereinafter "Swinehart 1").

c. Douglas B. Terry and Daniel C. Swinehart, *Managing Stored Voice in the Etherphone System*, ACM TRANSACTIONS ON COMPUTER SYSTEMS 6(1) (February 1988), 3-27 (hereinafter "Terry").

d. Daniel C. Swinehart, *System Support Requirements for Multi-media Workstations*, PROCEEDINGS OF THE SPEECHTECH '88 CONFERENCE (April 1988), 82-83 (hereinafter "Swinehart 2").

e. Polle T. Zellweger, *Active Paths through Multimedia Documents*, DOCUMENT MANIPULATION AND TYPOGRAPHY, J.C. AN VILET (ED.), CAMBRIDGE UNIVERSITY PRESS (1988) (hereinafter "Zellweger 2").

These papers were published together and form a single reference.

¶ 87. During the Net2Phone Litigation mentioned above, Net2Phone attempted to distinguish the claims of the '704 patent over Etherphone. The court has yet to render an opinion on these arguments. As set forth in **Exhibit Q**, these arguments fail to distinguish the claims of the '704 patent over Etherphone for a variety of reasons.

¶ 88. Etherphone was not cited during the prosecution of the '704 patent. As delineated below there is a SNQ of patentability raised by Etherphone. Below first the independent claims are set forth along with a discussion concerning the relevancy of Etherphone to the SNQ of patentability. Then the dependent claims are set forth.

INDEPENDENT CLAIM 1

The preamble of Claim 1 reads, in pertinent part: "A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network . . ."

¶ 89. Etherphone discloses a computer program product for use with a computer system which executes a "first process" and is operatively connectable to a "second process" and a server over a computer network. For example, the Etherphone system is:

> …based on a hardware architecture that uses microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server, this system has been used for applications such as directory-based call placement, call logging, call filtering, and automatic call forwarding. (Zellweger 1, page 1.) *See also id.*, Figure 1 (illustrating Etherphones, computer workstations and servers communicating over an Ethernet network).

The system components shown in Figure 1 of Zellweger 1 provides communication "between two or more parties (Etherphones, servers, and so on)." *Id.*, page 3.

¶ 90. Claim 1 requires "a computer usable medium having program code embodied in the medium." The functionality of the Etherphone system is implemented in software, which is inherently stored on a computer usable medium. As described in Swinehart 2, the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services." Swinehart 2, page 1. *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet which also supports a voice file server and a voice synthesis server . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").

¶ 91. In the pending litigation, Net2Phone argued that the term "server" should be defined broadly. Plaintiff Net2Phone, Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), page 3. More specifically, Net2Phone argued:

> Consistent with the use of the term 'server' in the specification, the claims do not refer to any specific server configuration. They simply require a 'server' (also referred to as a 'connection server,' 'address server,' or 'server process'). There is nothing in any of the claims that require that the server be in the form of a single computer with a centralized database, as defendants contend.

Id., page 4. Similarly, Net2Phone argued that "[a] server in a 'client/server system' can be implemented in any number of ways, from one to multiple computers, in one location or many, and from a single large computer acting as the server to a network of personal computers." Plaintiff Net2Phone Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007) (Exhibit W), page 7 (citing to the declaration of Professor Larry L. Peterson). Thus, under Net2Phone's interpretation, a "server" is not limited to any particular hardware or software configuration.

¶ 92. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Oct. 19, 2007) (Exhibit X), pages 2-9. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a "server." Under any interpretation, the Voice Control Server described in Etherphone is a "server."

Claim 1 further requires "program code for transmitting to the server a network protocol address received by the first process following connection to the computer network."

¶ 93. Etherphone processes notify the Voice Control Server (sometimes referred to as a "Telephone Control Server") of their network addresses in order to receive calls from other Etherphone processes. As described in Swinehart 1:

> The telephone control server controls voice conversations, implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it <u>stores personal preference information about each user</u> that allows it to support advanced features such as ring motifs and subdued ringing without involving workstation programs. It uses <u>dynamic information linking users to workstations</u> in order to provide calls to individuals rather than fixed locations and the registration of visitors in the offices of their colleagues.

> Finally, and most importantly, the telephone control server provides a set of network protocols that workstations use to participate in the operation of the system.

Swinehart 1, page 4 (emphasis added). Consequently, when a user logs in to a workstation, the user's identity and the network address of the workstation are sent to the Voice Control Server to identify the user's current location. In fact, any time a computer transmits a data packet over an Ethernet network, the data packet must include the network address of the transmitting computer system (i.e., so that the receiving system knows the source of the data packet). *See id.* ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server."); Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does."). The network protocol address of the first process is "received" following connection to the computer network. For example, an Etherphone process (a running instance of an Etherphone application) will be assigned a network protocol address after the workstation or Etherphone on which it is running connects to the computer network – hence, following connection to the computer network. This is the case regardless of whether the workstation or Etherphone on which the given Etherphone process is running has a static network protocol address or, instead, a dynamically assigned network protocol address. The network protocol address is then transmitted to the Voice Control Server so that other Etherphone processes can locate the Etherphone process.

Claim 1 also requires "program code for transmitting, to the server, a query as to whether the second process is connected to the computer network."

¶ 94. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Moreover, as mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Accordingly, when a first user at a first

Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting.").

Claim 1 further requires "program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network."

¶ 95. As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. Because of this, a first Etherphone or workstation attempting to connect to a second Etherphone or workstation receives the network address of the second Etherphone or workstation from the Voice Control Server.

¶ 96. In Claim Construction Briefs filed in the pending litigation, the patentee argued that the term

> 'connected' means 'logged on,' and *vice versa* . . . To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect. While Net2Phone's invention endeavors to identify accurately who is on line, it is not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user's Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person's status— *e.g.*, setting a default value of two hours, after which the server assumes that a party has gone off-line if it has not heard from her. *See* '704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, "the on-line status information stored in the database is ***relatively*** *current*." *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term "connected" (or "on-line") is going to be modified at all, it should be modified to say "*relatively* currently connected," because that is what the patents actually say.

Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), pages 24-25. Thus, under Net2Phone's interpretation, the information retained in the "server" as to which processes are "connected to the computer network" or "online" may be imperfect. As described above, while the server "endeavors to identify accurately who is on line, it is not possible to achieve perfection." *Id.*

¶ 97. Once again, the requestor of the present reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Exhibit X), pages 12-14. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a process to be "connected to" the computer network or "on-line." Under any interpretation, a first Etherphone process receives the network protocol address of a second Etherphone process from the Voice Control Server when the second Etherphone process is "connected to the computer network." Given that the Voice Control Server must track the location of individual on-line users, it is capable of determining "on-line" status with at least the same level of precision described in the '066 patent. *See, e.g.,* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting.").

Claim 1 also requires "program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network."

¶ 98. As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server</u>." (emphasis added). For these reasons, after retrieving a network address of a callee device from the Voice Control Server, a workstation or Etherphone communicates directly over a point-to-point communication link with the callee device identified by the network address. *See also* Zellweger, page 2 ("Etherphones digitize, packetize, and encrypt telephone-quality voice (64 kilobits/second, with silence suppression) and send it to each other directly over an Ethernet . . ."); Swinehart 2, page 1 ("Etherphones digitize and encrypt telephone quality audio and transmit it in packet form directly over an Ethernet.").

INDEPENDENT CLAIM *2*

Claim 2 claims "[a]n apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising: a processor; a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network."

¶ 99. The Etherphone system includes a Voice Control Server which enables point-to-point communication between workstation and Etherphone processes. As described in Swinehart 1:

> The *telephone control server* controls voice conversations, implements the stand-alone behavior of telephone instruments, and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of *visitors* in the offices of their colleagues.

Swinehart 1, page 4 (emphasis in original). The Voice Control Server includes a processor and a network interface for connecting the Voice Control Server to the computer network. *See, e.g.*, Zellweger 1, Figure 1 (illustrating the Voice Control Server coupled to a 1.5 Mbit/sec Ethernet network).

Claim 2 requires "a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of processes, each network protocol address stored in the memory following connection of a respective process to the computer network."

¶ 100. Etherphone describes this limitation. In particular, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a memory. The network addresses are stored in the memory following the connection of the processes to the computer network. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Consequently, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user.

Claim 2 also requires "means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process."

¶ 101. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Moreover, as mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting."). Consequently, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. The query will then return the current location of the user to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").

INDEPENDENT CLAIM 4

Claim 4 claims "A method for enabling point-to-point communication between a first process and a second process over a computer network . . ."

¶ 102. Etherphone describes a method for enabling point-to-point communication between a first process and a second process over a computer network. For example, after receiving a network addresses of a first process, a second process establishes a point-to-point communication connection with the first process. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among</u>

the participants, bypassing the control server.") (emphasis added). The "participants" all communicate with the system via software processes executed on computer workstations or Etherphones. *See, e.g.*, Swinehart 2, page 1 (describing how the capabilities of the Etherphone system "are presented to application programmers as program packages and network services."). *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").

Claim 4 requires "receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network."

¶ 103. As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently receive and store the network protocol addresses in a computer memory. The network addresses are stored in the memory following the connection of the processes to the computer network. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. When a user logs in to a workstation (e.g., as a "visitor"), the user is assigned an "on-line status."

Claim 4 also requires "receiving a query from the first process to determine the on-line status of the second process."

¶ 104. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Moreover, as mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting."). Consequently, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. The query will then return the current location of the user to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing how a user "turns to his workstation and registers Karmen as a visitor" and also describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").

Claim 4 also requires "determining the on-line status of the second process."

¶ 105. See response to the previous claim element. As described above, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. *See* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."). Swinehart 1 describes different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing how a user "turns to his workstation and registers Karmen as a visitor" and also describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").

Finally, Claim 4 requires "transmitting an indication of the on-line status of the second process to the first process over the computer network."

¶ 106. The Voice Control Server will connect a first user to a second user if the second user is "online." *See, e.g.*, Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."). In addition, if the second user is offline or does not wish to receive calls, an indication is sent to the first user that the second user is unavailable. *See, e.g.*, Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").

INDEPENDENT CLAIM 10

Claim 10 claims "a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network . . ."

¶ 107. As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server</u>." (emphasis added). Swinehart 1, page 2. Accordingly, after retrieving a network address of a callee device from the Voice Control Server, a workstation or Etherphone communicates directly over a point-to-point communication link with the callee device identified by the network address. *See also* Zellweger, page 2 ("Etherphones digitize, packetize, and encrypt telephone-quality voice (64 kilobits/second, with silence suppression) and send it to each other directly over an Ethernet . . ."); Swinehart 2, page 1 ("Etherphones digitize and encrypt telephone quality audio and transmit it in packet form directly over an Ethernet.").

Claim 10 further claims "the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network . . ."

¶ 108. The workstations described in Etherphone include a graphical user interface (GUI). *See, e.g.*, Figures 1-10 of Swinehart 1 (illustrating various GUI features presented on the workstation display). *See also* Zellweger 1, Figures 3-4 (illustrating "telephone management windows" (Figure 3) and icons representing callers, callees and telephone lines (Figure 4)). The workstations may be Apple Macintoshes or Xerox 6085s. *See* Swinehart 1, page 1. The workstations are operatively connectable to the callee process and a server over the computer network. As previously described, "[t]he telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 2.

Claim 10 requires "providing a user interface element representing a first communication line."

¶ 109. Etherphone discloses this limitation. For example, Figure 3 of Zellweger 1 depicts the Etherphone telephone management windows, including Phone and Answer buttons, a conversation log, and a portion of a personal telephone directory, which is a set of speed-dialing buttons. As described in Zellweger 1, "[a] variety of convenient workstation dialing methods are provided: a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad." Zellweger 1, page 4. In addition, Figure 4 of Zellweger 1 illustrates telephone icons representing telephone lines and icons with graphical images of a caller/callee which represent active telephone lines. As such, the Etherphone telephone management windows provide a "user interface element representing a first communication line."

Claim 10 also requires "providing a user interface element representing a first callee process."

¶ 110. Etherphone discloses user interface elements in the form of speed-dial buttons which represent frequently called callees. As described in Zellweger 1, the GUI provides "browsable lists of names and associated telephone numbers as speed-dialing buttons." Zellweger 1, page 4. *See also* Zellweger 1, Figure. 3 (depicting portion of a personal telephone directory, which is a set of speed-dial buttons). As another example, in Zellweger 1, Figure 4, the top left user interface icon represents a personal telephone directory in the form of a graphical rolodex.

Claim 10 further requires "establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line."

¶ 111. First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, the top row of Figure 4 of Zellweger 1 shows a series of graphical icons used for placing a call including a personal telephone directory, a telephone, and a picture of a user on the phone (to indicate a call is in process). In this example, the personal telephone directory, displayed as a graphical rolodex, includes a plurality of graphical elements representing callees (i.e., with a separate card in the rolodex for each callee). The icon of the telephone and the icon with the picture of a user talking on the phone represents a telephone communication line. As described in Zellweger 1, "[a]n active conversation is represented as a conversation between two people with a superimposed indication of the other party's name (also shown in Figure 4)." Zellweger 1, pages 4-5. Thus, when the user makes a call, the name from the graphical rolodex (PolleZ in the example) is "associated with" the graphical element representing the communication line (the image with the user talking on the phone).

¶ 112. Alternatively, as set forth below, Claim 10 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard.

INDEPENDENT CLAIM 21

Claim 21 claims "[a] computer program product for use with a computer system comprising: a computer usable medium having program code embodied in the medium."

¶ 113. The functionality of the Etherphone system is implemented in software, which is inherently stored on a computer usable medium. As described in Swinehart 2, the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services." Swinehart 2, page 1. *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").

Claim 21 requires "establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network."

¶ 114. First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process over a computer network. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the caller process (i.e., the software executed on the caller's machine) includes a user interface. *See, e.g.*, Zellweger 1, page 4, Figures 3-4 ("a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad."). Finally, Etherphone describes "being operatively connectable to the callee process and a server over the computer network." *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").

Claim 21 also requires "program code for generating an element representing a first communication line."

¶ 115. Etherphone describes a software-based graphical user interface with a graphical element representing a communication line. For example, Figure 4 of Zellweger 1 illustrates a graphical telephone icon that represents both outgoing calls (top row) and incoming calls (bottom row) over a telephone line, and graphical icons representing active calls (top and bottom right) established over the telephone line.

Claim 21 also requires "program code for generating an element representing a first callee process."

¶ 116. Etherphone describe a software-based graphical user interface with a graphical element representing a "callee process" (if the callee receives calls at an Etherphone or workstation). As described in Zellweger 1, the GUI provides "browsable lists of names and associated telephone numbers as speed-dialing buttons. Zellweger 1, page 4. *See also* Zellweger 1, Figure. 3 (depicting portion of a personal telephone directory, which is a set of speed-dial buttons). As another example, in Zellweger 1, Figure 4, the top left user interface icon represents a personal telephone directory in the form of a graphical rolodex.

> **Claim 21 also requires "program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process."**

¶ 117. First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, the top row of Figure 4 of Zellweger 1 shows a series of graphical icons used for placing a call including a personal telephone directory, a telephone, and a picture of a user on the phone (to indicate a call is in process). In this example, the personal telephone directory, displayed as a graphical rolodex, includes a plurality of graphical elements representing callees (i.e., with a separate card in the rolodex for each callee). The icon of the telephone and the icon with the picture of a user talking on the phone represents a telephone communication line. As described in Zellweger 1, "[a]n active conversation is represented as a conversation between two people with a superimposed indication of the other party's name (also shown in Figure 4)." Zellweger 1, pages 4-5. Thus, when the user makes a call, the name from the graphical rolodex (PolleZ in the example) is "associated with" the graphical element representing the communication line (the image with the user talking on the phone).

¶ 118. Alternatively, as set forth below, Claim 21 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard.

INDEPENDENT CLAIM 32

Claim 32 claims "[a] method of locating a process over a computer network comprising the steps of: a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network."

¶ 119. Etherphone describes these limitations. As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Therefore, the Telephone Control Server (also referred to as the Voice Control Server) stores a list of network addresses which are made available to workstations and Etherphones. In addition, the Voice Control Server associates different user identifiers with each network protocol address. For example, a user may log in to any workstation and, thereafter, calls to that user will be directed to that workstation and its associated Etherphone. As described in Swinehart 1:

> The *telephone control server* controls voice conversations, implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues.

Swinehart 1, page 4 (underline emphasis added). The network addresses may be Internet protocol addresses. For example, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. *See also* Terry, Abstract ("the voice manager stores voice on a special voice file server that is accessible via the local internet."). Moreover, another Etherphone reference, Vin, explicitly describes using the Internet Protocol (IP) within the Etherphone system. *See, e.g.*, Vin, page 77, Figure 5 (Exhibit D of this Request) (illustrating a "protocol stack and format" which includes internet protocol (IP) packets). Vin may be combined with Etherphone under 35 U.S.C. § 102. *See* MPEP 2131.01 (stating that a §102 rejection over multiple references is proper when the extra references are cited to explain the meaning of a term used in the primary reference). In this case, Vin is used to define the complete meaning of the term "Voice Transmission Protocol" used in Etherphone. In any case, it would have been obvious to combine Vin with Etherphone because they both describe the same Etherphone system. See obviousness section below.

Claim 32 also requires "in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process."

¶ 120. As mentioned above, when a first user attempts to call a second user from a workstation and Etherphone, the Voice Control Server provides the current network protocol address of the second user to the requesting process executed on the workstation/ Etherphone of the first user. Using the network address, the requesting process then initiates a communication session with the workstation and Etherphone of the second user. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").

INDEPENDENT CLAIM 33

Claim 33 claims "[a] method for locating processes having dynamically assigned network protocol addresses over a computer network."

¶ 121. As discussed above, Etherphone discloses a method of locating network protocol addresses over a computer network. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). The network protocol addresses may be Internet protocol addresses. For example, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. *See also* Terry, Abstract ("the voice manager stores voice on a special voice file server that is <u>accessible via the local internet</u>."). Moreover, another Etherphone reference, Vin, explicitly describes using the Internet Protocol (IP) within the Etherphone system. *See, e.g.*, Harrick M. Vin, et al., *Multimedia Conferencing in the Etherphone Environment*, IEEE COMPUTER SOCIETY (Oct. 1991), page 77, Figure 5 (Exhibit D of this request) (illustrating a "protocol stack and format" which includes internet protocol (IP) packets). Vin may be combined with Etherphone under 35 U.S.C. § 102. *See* MPEP 2131.01 (stating that a §102 rejection over multiple references is proper when the extra references are cited to explain the meaning of a term used in the primary reference). In this case, Vin is used to define the complete meaning of the term "Voice Transmission Protocol" used in Etherphone. In any case, it would have been obvious to combine Vin with Etherphone because they both describe the same Etherphone system. Because dynamically assigning IP addresses was known, this feature is inherent in Etherphone or at least obvious in light of the numerous references describing dynamic IP address assignment. *See, e.g.*, RFC 1531, Dynamic Host Configuration Protocol (1993), Section 2.2 (describing the "dynamic allocation of network addresses").

Claim 33 also requires "maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network."

¶ 122. The Voice Control Server maintains "a compilation of entries . . . comprising a network protocol address and a corresponding identifier." For example, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a "computer memory." The network addresses are stored in the memory with corresponding user identifiers. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. Since the user is connecting via a workstation and Etherphone, the user identifier identifies the current software process through which the user is interacting with the system.

Claim 33 also requires "in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process."

¶ 123. As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.

INDEPENDENT CLAIM 38

Claim 38 claims "[a] computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium."

¶ 124. The Etherphone system is implemented with software executed on a plurality of computing devices, including servers, workstations, and Etherphones. As described in Zellweger 1, the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ." Zellweger 1, page 1. *See also* Swinehart 2, page 1 (describing how the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services."); Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment."). The computer systems are operatively connectable over a computer network to computer processes. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Remote procedure calls are inherently directed to "computer processes."

Claim 38 requires "program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network."

¶ 125. The Voice Control Server maintains "a compilation of entries . . . comprising a network protocol address and a corresponding identifier or a process connected to the computer network." For example, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a "computer memory." The network addresses are stored in the memory with corresponding user identifiers. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. Since the user is connecting via a workstation and Etherphone, the user identifier identifies the current software "process" through which the user is interacting with the system.

Claim 38 further requires "program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process."

¶ 126. As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.

INDEPENDENT CLAIM 43

Claim 43 claims "[a] computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein."

¶ 127. The Etherphone system is implemented with software executed on a plurality of computing devices, including servers, workstations, and Etherphones. As described in Zellweger 1, the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ." Zellweger 1, page 1. *See also* Swinehart 2, page 1 (describing how the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services."); Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment."). The computer systems are operatively connectable over a computer network to computer processes and server processes. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Remote procedure calls are inherently directed to "computer processes."

Claim 43 further requires "program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network."

¶ 128. Etherphone describes a directory database for storing network addresses of on-line processes. For example, the Voice Control Server (also referred to as a "Telephone Control Server") stores network addresses for processes executed on each workstation and Etherphone. As described in Zellweger 1: "Users can place calls by specifying a name, a number, or other attributes of the called party. A system directory database for local Xerox employees (about 1000 entries) is stored on the Voice Control Server." Zellweger 1, page 4 (emphasis added). *See also* Swinehart 1, page 4 ("The Telephone Control Server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). The network addresses are sent to the Voice Control Server following connection to the computer network. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server . . . uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues."). Thus, when a user logs in to a workstation, the network protocol address of the workstation and the identity of the user are sent to the Voice Control Server.

Claim 43 also requires "program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network."

¶ 129. After receiving the network protocol address from the Voice Control Server, the caller's workstation and Etherphone will establish a point-to-point connection with the callee's workstation and Etherphone. *See, e.g.*, Swinehart 1, page 4 (describing how after receiving a network address from the Voice Control Server, "voice datagrams are transmitted directly among the participants, bypassing the control server.").

INDEPENDENT CLAIM 44

The preamble of Claim 44 reads: "In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of."

¶ 130. Etherphone describes a first computer process operatively coupled to a second process and an address server and further describe a method in the first process for establishing point-to-point communication with the second process. For example, in the Etherphone system, a first process executed on a first Etherphone or workstation contacts the Voice Control Server to learn the address of a second process executed on a second Etherphone or workstation. *See e.g.*, Swinehart 1, page 4 ("The Telephone Control Server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). The first process then uses the network address to establish point-to-point communication with the second process. *See, e.g., id.* ("Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").

Claim 44 also requires "following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network."

¶ 131. In the Etherphone system, when a user logs in to a particular workstation, the user's identity and the network protocol address of the workstation is forwarded to the Voice Control Server. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server . . . uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues.").

Claim 44 also requires "querying the address server as to whether the second process is connected to the computer network."

¶ 132. As mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. Moreover, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone.

See also Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting.").

Claim 44 further requires "receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network."

¶ 133. As mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.

Claim 44 further requires "in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network."

¶ 134. As described in Swinehart 1, after receiving a network address from the Voice Control Server, "voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4.

DEPENDENT CLAIMS 5-7, 11-12, 14, 19-20, 22-23, 25, 30-31, 34-37, 39-42

Claim 5 of the '704 patent requires "searching the computer memory for an entry relating the second process; and retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process."

¶ 135. These features are inherent in the Etherphone system. In order for the Voice Control Server to manage "voice switching by sending to each Etherphone or service the network addresses of the other participants" (Swinehart 1, page 4) it must inherently search the server memory for the network addresses related to the other workstation and Etherphone processes. In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").

Claim 6 of the '704 patent requires "transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network."

¶ 136. In the Etherphone system, if a user is logged in and "online" then the Voice Control Server transmits the network address of the user's process (executed on the Etherphone or workstation) to the requesting process. *See, e.g.*, Swinehart 1, page 4 ("The *telephone control server* . . . uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of *visitors* in the offices of their colleagues.") (emphasis in original).

Claim 7 requires "generating an off-line message when the second process is determined in step C to have a negative on-line status with respect to the computer network; and transmitting the off-line message to the first process."

¶ 137. As described above, the Etherphone system includes a "do-not-disturb" option in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant." Swinehart 1, page 4.

Claim 11 requires "querying the server as to the on-line status of the first callee process; and receiving a network protocol address of the first callee process over the computer network from the server."

¶ 138. The Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2.

Claim 12 claims "providing an element representing a second communication line."

¶ 139. The Etherphone system inherently provides an element representing a second communication line. For example, using the Etherphone system, a user may receive and answer a call while already on an existing call. *See, e.g.*, Swinehart 1, page 2 (describing how users can place and receive other calls during a "background call"). Thus, multiple sets of graphical icons such as the ones shown in Figure 4 of Zellweger 1 were inherently displayed in the Etherphone system (e.g., graphical cards in a rolodex to represent callee processes, telephones to represent communication lines, and users talking on telephones to represent a callee process associated with a particular communication line). Alternatively, as set forth below, Claim 12 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard.

Claim 14 requires "providing a user interface element representing a second callee process; and establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line."

¶ 140. Etherphone describes this limitation. For example, Figure 8 of Swinehart 1 illustrates four user interface elements representing four different callee processes (four different users). The four callee processes are each associated with a graphical element representing a communication line – i.e., a telephone graphic and graphical window representing a teleconference (titled "conference at 3PM re: Budget"). *See* Swinehart 1, page 3.

Claim 19 requires "wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface."

¶ 141. Etherphone illustrates a visual display which allows the caller to control the operation of the Etherphone. *See, e.g.*, Figure 3 of Zellweger 1 (illustrating a "workstation telephone management windows") and Figures 1-10 of Swinehart 1 (illustrating a series of windows for controlling an Etherphone).

Claim 20 requires "wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface."

¶ 142. As described in Swinehart 1, to establish a call, user's can "select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad." Zellweger 1, page 4. As described above, a call from one workstation/Etherphone to another workstation/Etherphone comprises a point-to-point link.

Claim 22 requires "program code for querying the server as to the on-line status of the first callee process; and program code for receiving a network protocol address of the first callee process over the computer network from the server."

¶ 143. As mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. Moreover, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."). In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").

Claim 23 requires "program code for generating an element representing a second communication line."

¶ 144. The Etherphone system is inherently capable of providing an element representing a second communication line. For example, the Etherphone system is capable of conference calling. *See, e.g.*, Swinehart 1, page 3 (describing "negotiated conference calls"). In addition, using the Etherphone system, a user may receive and answer a call while already on an existing call. *See, e.g.*, Swinehart 1, page 2 (describing how users can place and receive other calls during a "background call"). Thus, multiple sets of graphical icons such as the ones shown in Figure 4 of Zellweger 1 were inherently displayed in the Etherphone system (e.g., graphical cards in a rolodex to represent callee processes, telephones to represent communication lines, and users talking on telephones to represent a callee process associated with a particular communication line).

Claim 25 requires "program code for generating an element representing a second callee process; and program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process."

¶ 145. Etherphone describes this limitation. For example, Figure 8 of Swinehart 1 illustrates four user interface elements representing four different callee processes (four different users). The four callee processes are each associated with a graphical element representing a communication line – i.e., a telephone graphic and graphical window representing a teleconference (titled "conference at 3PM re: Budget"). *See* Swinehart 1, page 3.

Claim 30 requires that the "computer system further comprises a visual display and the user interface comprises a graphic user interface."

¶ 146. Etherphone illustrates a visual display which allows the caller to control the operation of the Etherphone. *See, e.g.*, Figure 3 of Zellweger 1 (illustrating a "workstation telephone management windows") and Figures 1-10 of Swinehart 1 (illustrating a series of windows for controlling an Etherphone).

Claim 31 requires that "the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises: program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process."

¶ 147. As described above, Etherphone illustrates a visual display which allows the caller to control the operation of the Etherphone. *See, e.g.*, Figure 3 of Zellweger 1 (illustrating a "workstation telephone management windows") and Figures 1-10 of Swinehart 1 (illustrating a series of windows for controlling an Etherphone). Figure 3 of Zellweger 1, for example, shows a "phone" button representing a first communication line and several other graphical elements representing callee processes. As described in Zellweger 1, "[a] variety of convenient workstation dialing methods are provided: a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad." Zellweger 1, page 4. In addition, the top row of Figure 4 of Zellweger 1 shows a series of graphical icons used for placing a call including a personal telephone directory, a telephone, and a picture of a user on the phone (to indicate a call is in process). In this example, the personal telephone directory, displayed as a graphical rolodex, includes a plurality of graphical elements representing callees (i.e., with a separate card in the rolodex for each callee). The icon of the telephone and the icon with the picture of a user talking on the phone represents a telephone communication line. As described in Zellweger 1, "[a]n active conversation is represented as a conversation between two people with a superimposed indication of the other party's name (also shown in Figure 4)." Zellweger 1, pages 4-5.

Claim 34 requires "modifying the compilation of entries."

¶ 148. Etherphone discloses modifying the entries stored on the Voice Control Server as users log-in and log-out of workstations. As described in Swinehart 1:

> The *telephone control server* <u>controls voice conversations,</u> implements the stand-alone behavior of telephone instruments and <u>coordinates the activities of workstations and adjacent telephones</u> in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It <u>uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues</u>.

Swinehart 1, page 4 (underline emphasis added). Thus, the entries stored on the Voice Control Server are updated dynamically "in order to provide calls to individuals rather than fixed locations."

Claim 35 requires "adding an entry to the compilation upon the occurrence of a predetermined event."

¶ 149. Etherphone inherently discloses this limitation. The predetermined event may include, for example, adding a new workstation and Etherphone to the network, powering on an existing Etherphone or workstation, adding a new Voice Control Server, and/or logging in a user to the system from a new workstation/Etherphone. Each of these events may require adding an entry to the Voice Control Server. *See, e.g.,* Swinehart 1, page 4 (describing how the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). In order to "manage" voice switching by sending network addresses of logged in participants, the Voice Control Server must be capable of adding entries to its database in response to "predetermined events."

Claim 36 requires that "the predetermined event comprises notification by a user process of an assigned network protocol address."

¶ 150. Etherphone discloses this limitation. For example, when a user logs in to a workstation/Etherphone, the identity of the user and the network address of the workstation/Etherphone is transmitted to the voice control server so that the user can be located by other users. As mentioned above, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of visitors in the offices of their colleagues." Swinehart 1, page 4.

Claim 37 requires "deleting an entry from the compilation upon the occurrence of a predetermined event."

¶ 151. Etherphone inherently describes this limitation. As described in Zellweger:

> If an Etherphone user logs in at a workstation, his calls can be automatically forwarded to the adjacent Etherphone. An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does. <u>Each visit request cancels any earlier requests</u>. The common problem of forgetting to cancel forwarding is eased by ringing both Etherphones during visiting.

Zellweger, page 5 (emphasis added). *See also* Swinehart, page 2 (describing how after Karmen leaves Lee's office "an additional call to Karmen . . . reminds Lee to terminate the visiting arrangement."). In these examples, "terminating" or "cancelling" the user's visiting status inherently requires deleting the association between the visiting user and the network address of the workstation/Etherphone.

Claim 39 requires "program code configured to modify the compilation of entries."

¶ 152. Etherphone discloses modifying the entries stored on the Voice Control Server as users log-in and log-out of workstations. As described in Swinehart 1:

> The *telephone control server* <u>controls voice conversations,</u> implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It <u>uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues</u>.

Swinehart 1, page 4 (underline emphasis added). *See also* Swinehart, page 2 (describing how after Karmen leaves Lee's office "an additional call to Karmen . . . reminds Lee to terminate the visiting arrangement."). Thus, the entries stored on the Voice Control Server are updated dynamically "in order to provide calls to individuals rather than fixed locations" and to "terminate" old, outdated entries.

Claim 40 requires "program code configured to add an entry to the compilation upon the occurrence of a predetermined event."

¶ 153. Etherphone inherently discloses this limitation. The predetermined event may include, for example, adding a new workstation and Etherphone to the network, powering on an existing Etherphone or workstation, adding a new Voice Control Server, and/or logging in a user to the system from a new workstation/Etherphone. Each of these events may require adding an entry to the Voice Control Server. *See, e.g.,* Swinehart 1, page 4 (describing how the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). In order to "manage" voice switching by sending network addresses of logged in participants, the Voice Control Server must be capable of adding entries to its database in response to "predetermined events."

Claim 41 requires that the "predetermined event comprises notification by a process of an assigned network protocol address."

¶ 154. Etherphone inherently discloses this limitation. For example, when a user logs in to a workstation/Etherphone, the identity of the user and the network address of the workstation/Etherphone is transmitted to the voice control server so that the user can be located by other users. As mentioned above, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of visitors in the offices of their colleagues." Swinehart 1, page 4.

Claim 42 requires "program code configured to delete an entry from the compilation upon the occurrence of a predetermined event."

¶ 155. Etherphone inherently describes this limitation. As described in Zellweger:

> If an Etherphone user logs in at a workstation, his calls can be automatically forwarded to the adjacent Etherphone. An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does. <u>Each visit request cancels any earlier requests</u>. The common problem of forgetting to cancel forwarding is eased by ringing both Etherphones during visiting.

Zellweger, page 5 (emphasis added). *See also* Swinehart, page 2 (describing how after Karmen leaves Lee's office "an additional call to Karmen . . . reminds Lee to terminate the visiting arrangement."). In these examples, "terminating" or "cancelling" the user's visiting status inherently requires deleting the association between the visiting user and the network address of the workstation/Etherphone.

2. <u>Obviousness Rejections</u>

¶ 156. The following is a quotation of 35 U.S.C. §103 (a) which forms the basis for the following obviousness rejections:

> 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.

(i) Etherphone in view NetBIOS

¶ 157. Claim 3 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of NetBIOS.

Claim 3 of the '704 patent requires "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory."

¶ 158. Etherphone does not explicitly describe a timer for time stamping network protocol address entries stored on the Voice Control Server. However, time stamping was a well known technique at the time the application which resulted in the '704 patent was filed. For example, the NBNS described in NetBIOS includes a timer for time-stamping name/IP address entries. As described in NetBIOS, "[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure." NetBIOS at 382. Similarly, as described in NetBIOS:

> If an end-node holds any names that have finite time-to-live values, then that node must periodically send a status report to the NBNS. Each name is reported using the NAME REFRESH REQUEST packet. These status reports restart the timers of both the NBNS and the reporting node. However, the only timers which are restarted are those associated with the name found in the status report. Timers on other names are not affected. *Id.*

(ii) Motivation to Combine Etherphone With NetBIOS

¶ 159. The motivation to combine Etherphone with NetBIOS exists due to the problem being solved. For example, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include the capability of time stamping network protocol address entries at the Etherphone Voice Control Server to determine the length of time that a user has been online at a particular workstation/Etherphone and/or to remove stale entries.

(iii) Etherphone in view of Vin

¶ 160. As described above, Claim 32 is anticipated by Etherphone. Alternatively, Claim 32 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Harrick M. Vin, et al., *Multimedia Conferencing in the Etherphone Environment*, IEEE COMPUTER SOCIETY (October 1991) ("Vin").

Claim 32 claims "[a] method of locating a process over a computer network comprising the steps of: a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to

the Internet, the Internet Protocol address added to the list following connection of the process to the computer network."

¶ 161. As described in Etherphone: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, the Telephone Control Server (also referred to as the Voice Control Server) stores a list of network addresses which are made available to workstations and Etherphones. In addition, the Voice Control Server associates different user identifiers with each network protocol address. For example, a user may log in to any workstation and, thereafter, calls to that user will be directed to that workstation and its associated Etherphone. As described in Swinehart 1:

> The *telephone control server* <u>controls voice conversations,</u> implements the stand-alone behavior of telephone instruments and <u>coordinates the activities of workstations and adjacent telephones</u> in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It <u>uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues</u>.

Swinehart 1, page 4 (underline emphasis added). The network addresses may be Internet protocol addresses. For example, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. *See also* Terry, Abstract ("the voice manager stores voice on a special voice file server that is <u>accessible via the local internet</u>."). While the Etherphone papers do not explicitly describe using Internet Protocol (IP) addresses, another Etherphone reference, Vin, explicitly describes using IP addresses within the Etherphone system. *See, e.g.*, Vin, page 77, Figure 5 (illustrating a "protocol stack and format" used in an Etherphone system which includes internet protocol (IP) packets).

Claim 32 also requires "in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process."

¶ 162. As mentioned above, when a first user attempts to call a second user from a workstation and Etherphone, the Voice Control Server provides the current network protocol address of the second user to the requesting process executed on the workstation/ Etherphone of the first user. Using the network address, the requesting process then initiates a communication session with the workstation and Etherphone of the second user. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").

(iv) Motivation to Combine Etherphone and Vin

¶ 163. A motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine one reference describing the Etherphone system (Etherphone) with another reference describing the same Etherphone system (Vin). Moreover, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. Thus, it would have been obvious to a person of ordinary skill in the art at the time of the invention to combine Etherphone with other references (such as Vin) which describe the use of IP addresses.

(v) Etherphone in view of Vin and Further in View of RFC 1531

¶ 164. Claim 33 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Vin and further in view of Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531") .

Claim 33 claims "[a] method for locating processes having dynamically assigned network protocol addresses over a computer network."

¶ 165. As discussed above, Etherphone discloses a method of locating network protocol addresses over a computer network. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). While Etherphone does not explicitly state that the network protocol addresses may be Internet protocol addresses, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. *See also* Terry, Abstract ("the voice manager stores voice on a special voice file server that is <u>accessible via the local internet</u>."). Vin explicitly describes using the Internet Protocol (IP) within the Etherphone system. *See, e.g.*, Vin, page 77, Figure 5 (illustrating a "protocol stack and format" used in an Etherphone system which includes internet protocol (IP) packets). In addition, as described in

RFC 1531, IP addresses were known to be dynamically assigned. *See, e.g.*, RFC 1531, Section 2.2 (describing the "dynamic allocation of network addresses").

Claim 33 also requires "maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network."

¶ 166. The Voice Control Server maintains "a compilation of entries . . . comprising a network protocol address and a corresponding identifier." For example, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a "computer memory." The network addresses are stored in the memory with corresponding user identifiers. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. Since the user is connecting via a workstation and Etherphone, the user identifier identifies the current software process through which the user is interacting with the system.

Claim 33 also requires "in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process."

¶ 167. As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.

¶ 168. The other independent claims of the '704 patent state, more generally, that the network protocol address is assigned or transmitted to the database following the connection of the computer to the computer network. *See* Claim 1 ("transmitting to the server a network protocol address received by the first process following connection to the computer network"); Claim 2 ("each network protocol address stored in the memory following connection of a respective process to the computer network”); Claim 4 (“the network protocol addresses received following connection of the respective process to the computer network”); Claim 32 (“the Internet Protocol address added to the list following connection of the process to the computer network”); Claim 38 (“the network protocol address of the corresponding process assigned to the process upon connection to the computer network”); Claim 43 (“the network protocol address of each respective process forwarded to the database following connection to the computer network”); Claim 44 (“following connection of the first process to the computer network forwarding to the address server a network protocol address”).

¶ 169. As described above, Etherphone and Vin inherently describes these features. For example, on many networks, including the TCP/IP networks, network addresses are assigned “following connection to the computer network.” *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) (“RFC 1531”), Section 2.2 (describing the “dynamic allocation of network addresses” on TCP/IP networks). Thus, in at least some instances, the computer system on which NetBIOS was used received IP addresses dynamically, after connecting to the computer network. Consequently, dynamic address assignment is inherent in the NetBIOS reference.

¶ 170. Alternatively, Claims 1-2, 4-7, 10-12, 14, 19-23, 25, and 30-44 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Vin and further in view of RFC 1531, which describes how TCP/IP addresses were dynamically assigned. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) (“RFC 1531”), Section 2.2 (describing the “dynamic allocation of network addresses” on TCP/IP networks).

(vi) Motivation to Combine Etherphone, Vin, and RFC 1531

¶ 171. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction).

(vii) Etherphone in view of Pinard

¶ 172. Claims 10-17, 19-28, and 30-31should be rejected under 35 U.S.C. § 103 as being unpatentable over Etherphone in view of Pinard.

INDEPENDENT CLAIM 10

¶ 173. Etherphone anticipates Claim 10 for the reasons stated above. Alternatively, Claim 10 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard.

Claim 10 claims "a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network . . ."

¶ 174. As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server</u>." (emphasis added). Swinehart 1, page 2. Thus, after retrieving a network address of a callee device from the Voice Control Server, a workstation or Etherphone communicates directly over a point-to-point communication link with the callee device identified by the network address. *See also* Zellweger, page 2 ("Etherphones digitize, packetize, and encrypt telephone-quality voice (64 kilobits/second, with silence suppression) and send it to each other directly over an Ethernet . . ."); Swinehart 2, page 1 ("Etherphones digitize and encrypt telephone quality audio and transmit it in packet form directly over an Ethernet.").

Claim 10 further claims "the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network . . ."

¶ 175. The workstations described in Etherphone includes a graphical user interface (GUI). *See, e.g.*, Figures 1-10 of Swinehart 1 (illustrating various GUI features presented on the workstation display). *See also* Zellweger 1, Figures 3-4 (illustrating "telephone management windows" (Figure 3) and icons representing callers, callees and telephone lines (Figure 4)). The workstations may be Apple Macintoshes or Xerox 6085s. *See* Swinehart 1, page 1. The workstations are operatively connectable to the callee process and a server over the computer network. As previously described, "[t]he telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 2.

Claim 10 requires "providing a user interface element representing a first communication line."

¶ 176. Pinard discloses a user interface element representing a first communication line. For example, Figure 6 of Pinard illustrates a first call icon 23 which represents a first communication line and a second call icon 29 which represents a second communication line. In the example shown in Figure 6, the first call icon 23 represents a telephone call between "Debbie" and "John" and the second call icon 29 represents a telephone call between "Debbie" and "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 10 also requires "providing a user interface element representing a first callee process."

¶ 177. Pinard describes "a user interface element representing a first callee process." In the example shown in Figure 6 of Pinard, a first user interface element 21 is shown for the callee named "John" and a second user interface element is shown for the callee named "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 10 further requires "establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line."

¶ 178. As described above, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

INDEPENDENT CLAIM 21

Claim 21 claims "[a] computer program product for use with a computer system comprising: a computer usable medium having program code embodied in the medium."

¶ 179. The functionality of the Etherphone system is implemented in software, which is inherently stored on a computer usable medium. As described in Swinehart 2, the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services." Swinehart 2, page 1. *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").

Claim 21 requires "establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network."

¶ 180. First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process over a computer network. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the caller process (i.e., the software executed on the caller's machine) includes a user interface. *See, e.g.*, Zellweger 1, page 4, Figures 3-4 ("a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad."). Finally, Etherphone describes "being operatively connectable to the callee process and a server over the computer network." *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").

Claim 21 also requires "program code for generating an element representing a first communication line."

¶ 181. Pinard discloses program code for generating an element representing a first communication line. For example, Figure 6 of Pinard illustrates a first call icon 23 which represents a first communication line and a second call icon 29 which represents a second communication line. In the example shown in Figure 6, the first call icon 23 represents a telephone call between "Debbie" and "John" and the second call icon 29 represents a telephone call between "Debbie" and "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 21 also requires "program code for generating an element representing a first callee process."

¶ 182. Pinard describes program code for generating an element representing a first callee process. In the example shown in Figure 6 of Pinard, a first user interface element 21 is shown for the callee named "John" and a second user interface element is shown for the callee named "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.

Claim 21 also requires "program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process."

¶ 183. As described above, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

DEPENDENT CLAIMS 11-17, 19-20, 22-28, AND 30-31

Claims 11 and 22 require "querying the server as to the on-line status of the first callee process; and receiving a network protocol address of the first callee process over the computer network from the server" and "program code for querying the server as to the on-line status of the first callee process; and program code for receiving a network protocol address of the first callee process over the computer network from the server," respectively.

¶ 184. The Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2.

Claim 12 and 24 require "providing an element representing a second communication line" and "program code for generating an element representing a second communication line," respectively.

¶ 185. The graphical user interface described in Pinard provides an element representing a second communication line. For example, call icons 23 and 29 representing two communication lines are shown in Figure 6 of Pinard. *See* Pinard, Col. 5, lines 31-40, Figure 6 ("Now there are clearly two calls in progress . . .").

Claims 13 and 24 require "terminating the point-to-point communication link from the caller process to the first callee process, in response to the user

disassociating the element representing the first callee process from the element representing the first communication line" and "program code, responsive to the user disassociating the element representing the first callee process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process," respectively.

¶ 186. Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.

Claim 13 and 24 further require "establishing a different point-to-point communication link from the caller process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line" and "program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process," respectively.

¶ 187. In Figure 6, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, Col. 4, lines 22-31.

Claims 14 and 25 require "providing a user interface element representing a second callee process; and . . . establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line" and "program code for generating an element representing a second callee process; and program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process," respectively.

¶ 188. In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and

"Debbie." See Pinard, Col. 5, lines 31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

Claims 15 and 26 require "removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line" and "program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link," respectively.

¶ 189. In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, Col. 6, lines 14-15.

Claims 16 and 27 require "providing a user interface element representing a communication line having a temporarily disabled status" and "program code for generating an element representing a communication line having a temporarily disabled status," respectively.

¶ 190. Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 16 and 27 also require "temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status" and "program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process," respectively.

¶ 191. In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 17 and 28 require that "the element provided in step D represents a communication line on hold status" and "the communication line having a temporarily disabled status comprises a communication line on hold status," respectively.

¶ 192. In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 19 and 30 require "wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface" and "wherein the computer system further comprises a visual display and the user interface comprises a graphic user interface," respectively.

¶ 193. Pinard discloses a graphical user interface on a visual display which allows the caller to control the operation of the telephone. *See, e.g.*, Pinard, Figures 2-16 and Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 20 and 31 requires "wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface" and "program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process," respectively.

¶ 194. As described above, Pinard discloses that a point-to-point communication link is established in response to a user associating a graphic element representing a callee process with a graphic element representing a communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15.

Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

(viii) Motivation to Combine Etherphone and Pinard

¶ 195. A motivation to combine Etherphone and Pinard exists due to the problem to be solved. The Pinard reference relates to the field of telephony, and in particular to a method of indicating the status of various calls, to a user. *See* Pinard, Col. 1, lines 5-7. Indeed, the graphical user interface described in Pinard could be used in any system that operates a telephony application on a personal computer or on a personal computer in conjunction with a server. *See* Pinard, Col. 1, lines 60-62; Col. 2, lines 41-45. One of ordinary skill in the art would have recognized that the particular design choices reflected in the graphical user interface of Pinard could readily be implemented within the context of the Etherphone system. In fact, Etherphone discloses a graphical user interface with similar features to those described in Pinard.

(ix) Etherphone in view of Pinard and Further in View of VocalChat

Claim 18 and 29 require that "the element provided in step D represents a communication line on mute status" and "the communication line having a temporarily disabled status comprises a communication line on mute status," respectively.

¶ 196. As described above, Etherphone and Pinard describe all of the elements of Claim 18 and 29 except for a "communication line on mute status. However, VocalChat describes a "communication line on mute status." As described in the User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.

(x) Motivation to Combine VocalChat with Etherphone and Pinard

¶ 197. A motivation to combine VocalChat with Etherphone and Pinard exists due to the problem to be solved. All three references relate to the field of telephony, and in particular to the use of computer system to implement telephony features. *See, e.g.,* Pinard, Col. 1, lines 5-7. One of ordinary skill in the art would have recognized the need for a "mute" function to enable users to mute the audio of a call as needed. In fact, the Etherphone system included a mute function, although it was not explicitly described in Etherphone. *See, e.g.*, Vin, page 73 ("Additional accelerators and features, such as manually toggling the Recv-only condition to mute the conversation, are available via mouse clicks on conversation log entries.").

C. VocalChat User's Guide in view of VocalChat Readme, and further in view of VocalChat Networking, and further in view of VocalChat Help File, and further in view of VocalChat Troubleshooting Help File

¶ 198. The following is a quotation of 35 U.S.C. §103 (a) which forms the basis for the following obviousness rejections:

> 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.

¶ 199. Claims 1-2, 4, 7, 10-11, 19-22, 30-42 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over VocalChat User's Guide, Version 2.0 (1994) ("User's Guide"), in view of VocalChat Readme File, Version 2.02 (June, 1994) ("Readme"), in view of VocalChat 1.01 Networking Information ("VocalChat Networking"), in view of VocalChat Information, Version 2.02 (July 18, 1994) ("Help File"), and further in view of VocalChat Troubleshooting Help File, Version 2.02 (July 18, 1994) ("Troubleshooting Help File"). These prior art publications are collectively referred to herein as "the VocalChat references" or "VocalChat."

¶ 200. As stated in the declaration of Alon Cohen, one of the co-founders of VocalTec, Ltd., included as **Exhibit L**: (1) VocalChat 1.01 Networking Information ("Networking Information"), attached as **Exhibit I** (referred to as Exhibit A in the declaration), was publicly distributed in 1994 as part of the VocalChat version 1.01 software, which was commercially released and on sale to the general public in 1994; (2) VocalChat 2.0 User's Guide ("User's Guide"), attached as **Exhibit G** (referred to as Exhibit B in the declaration), was publicly distributed in 1994 as part of the VocalChat version 2.0 software, which was commercially released and on sale to the general public in 1994; and (3) The VocalChat Readme File ("Readme"), attached as **Exhibit H** (referred to as Exhibit C in the declaration), the VocalChat Troubleshooting Help File ("Troubleshooting Help File"), attached as **Exhibit K** (referred to as Exhibit D in the declaration), and VocalChat Information ("Help File"), attached as **Exhibit J** (referred to as Exhibit E in the declaration), are true and correct print-outs of VocalChat version 2.02's README.TXT, TROUBLE.HLP, and INFO.HLP files, respectively. In sum, electronic copies of all of these documents were publicly distributed in 1994 as part of the various VocalChat software releases.

1. Motivation to Combine the VocalChat References Under 35 U.S.C. § 103

¶ 201. A strong motivation to combine the VocalChat references under 35 U.S.C. § 103 exists because they all describe the same VocalChat system. The fact that some of the references describe different versions of the VocalChat system does not alter the fact that it would have been obvious to combine these references because all of the references share numerous common features (e.g., a central server to store addresses and VocalChat client software) which interoperate in the same basic manner.

¶ 202. During the Net2Phone Litigation, Net2Phone attempted to distinguish the claims of the '704 patent over the VocalChat References. The court has yet to render an opinion on these arguments. As set forth in **Exhibit R**, these arguments fail to distinguish the claims of the '704 patent over the combined VocalChat References for a variety of reasons.

¶ 203. These VocalChat references were not cited or discussed alone, or in combination, during the prosecution of the '704 patent. As delineated below, there is a SNQ of patentability raised by VocalChat. Below first the independent claims are set forth along with a discussion concerning the relevancy of VocalChat to the SNQ of patentability. Then the dependent claims are set forth.

INDEPENDENT CLAIM 1

The preamble of Claim 1 reads, in pertinent part: "A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network . . ."

¶ 204. VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols.").

¶ 205. In the pending litigation, Net2Phone argued that the term "server" should be defined broadly. Plaintiff Net2Phone, Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), page 3. More specifically, Net2Phone argued:

> Consistent with the use of the term 'server' in the specification, the claims do not refer to any specific server configuration. They simply require a 'server' (also referred to as a 'connection server,' 'address server,' or 'server process'). There is nothing in any of the claims that require that the server be in the form of a single computer with a centralized database, as defendants contend.

Id., page 4. Similarly, Net2Phone argued that "[a] server in a 'client/server system' can be implemented in any number of ways, from one to multiple computers, in one location or many, and from a single large computer acting as the server to a network of personal computers." Plaintiff Net2Phone Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007) (Exhibit W), page 7 (citing to the declaration of Professor Larry L. Peterson). Thus, under Net2Phone's interpretation, a "server" is not limited to any particular hardware or software configuration.

¶ 206. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Oct. 19, 2007) (Exhibit X), pages 2-9. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a "server." Under any interpretation, the "post office" server in VocalChat is a "server."

Claim 1 requires "a computer usable medium having program code embodied in the medium ..."

¶ 207. As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A").

Claim 1 also requires "program code for transmitting to the server a network protocol address received by the first process following connection to the computer network."

¶ 208. As illustrated in the figure on page 5 of the VocalChat User's Guide (reproduced above), computers with VocalChat installed connect directly to a server to register their current network protocol addresses. In the initial VocalChat implementations (versions 1.x) each VocalChat client transmits its name and network protocol address to a USERS file stored on the server. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users.
>
> Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*.

VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" stored within a 'Post-Office' directory. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory.").

¶ 209. Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server**

Installation is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

¶ 210. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network.

Claim 1 further requires "program code for transmitting, to the server, a query as to whether the second process is connected to the computer network."

¶ 211. VocalChat employs different techniques for locating users based on the underlying network protocol. As described in the Help File:

> Method of determining users address:
>
> | | |
> |---|---|
> | Netware | Get Users information from Netware 2.x/3.x bindery |
> | WinWorkgroups | Get users information from Windows for Workgroups. |
> | Generic User | VocalChats files for users information. (See Generic network, below). |

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is

> entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2. *See also* Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"); page 2 ("When working with the IPX and TCP/IP protocols, the network addresses of the different workstations are required for VocalChat to be able to access the network users."). With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 1 also requires "program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network."

¶ 212. When TCP/IP is used, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, as described above, with TCP/IP, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that VocalChat clients "received" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*; *see also id.*, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses").

¶ 213. In Claim Construction Briefs filed in the pending litigation, the patentee argued that the term

> 'connected' means 'logged on,' and *vice versa* . . . To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect. While Net2Phone's invention endeavors to identify accurately who is on line, it is not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user's Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person's status— *e.g.*, setting a default value of two hours, after which the server assumes that a party has gone off-line if it has not heard from her. *See* '704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, "the on-line status information stored in the database is ***relatively*** *current*." *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term "connected" (or "on-line") is going to be modified at all, it should be modified to say "*relatively* currently connected," because that is what the patents actually say.

Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), pages 24-25. Thus, under Net2Phone's interpretation, the information retained in the "server" as to which processes are "connected to the computer network" or "online" may be imperfect. As described above, while the server "endeavors to identify accurately who is on line, it is not possible to achieve perfection." *Id.* As described above, VocalChat employs similar techniques as address entries for off-line VocalChat processes are removed through the use of log-out messages.

¶ 214. Once again, the requestor of the present reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Exhibit X), pages 12-14. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a process to be "connected to" the computer network or "on-line." Under any interpretation, a first VocalChat process receives the network protocol address of a second VocalChat process from the "post office" server when the second VocalChat process is "connected to the computer network."

> **Claim 1 also requires "program code, responsive to the network protocol address of the second process, for establishing a point-to-point**

communication link between the first process and the second process over the computer network."

¶ 215. VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

INDEPENDENT CLAIM 2

Claim 2 claims "[a]n apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising: a processor; a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network."

¶ 216. VocalChat discloses the preamble of claim 2. For example, VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols."). Inherently, the personal computers and the server included a processor, for processing program code, and a network interface connected to the network. The VocalChat software installed on each computer system comprises a computer-implemented "process."

Claim 2 requires "a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of processes, each network protocol address stored in the memory following connection of a respective process to the computer network."

¶ 217. Computer systems with VocalChat installed and the central server inherently included "a memory, operatively coupled to the processor" in the form of a RAM and a hard drive. *See, e.g.*, User Guide, page 7 (describing minimum system requirements of 4 MB RAM).

In the initial VocalChat implementations (versions 1.x) each VocalChat client transmits its name and network protocol address to a USERS file stored on a server. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users.
>
> <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*.

VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8("the Setup program crates a Connection List File which is used to identify and access users").

¶ 218. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network in combination with VocalChat.

> **Claim 2 also requires "means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process."**

¶ 219. In a TCP/IP implementation, the server on which the Connection List/USERS file is located transmits the network protocol address of a second VocalChat client (second process) to a first VocalChat client (first process) upon request. As described in the Help File:

> Method of determining users address:
>
> | | |
> |---|---|
> | Netware | Get Users information from Netware 2.x/3.x bindery |
> | WinWorkgroups | Get users information from Windows for Workgroups. |
> | Generic User | VocalChats files for users information. (See Generic network, below). |

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2. With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

INDEPENDENT CLAIM 4

The preamble of Claim 4 reads, in pertinent part: "A method for enabling point-to-point communication between a first process and a second process over a computer network."

¶ 220. As mentioned above, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

Claim 4 requires "receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network."

¶ 221. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a <u>shared USERS file with the names of logged in users</u>.
>
> <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file.*

VocalChat Network Information, page 10 (underline emphasis added); *see also* Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared

CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

¶ 222. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network in combination with VocalChat.

Claim 4 also requires "receiving a query from the first process to determine the on-line status of the second process."

¶ 223. In a TCP/IP implementation, the server on which the Connection List/USERS file was located received queries from VocalChat clients (first processes) to determine the on-line status of other VocalChat clients (second processes). As described in the Help File:

> Method of determining users address:
>
> | | |
> |---|---|
> | Netware | Get Users information from Netware 2.x/3.x bindery |
> | WinWorkgroups | Get users information from Windows for Workgroups. |
> | Generic User | VocalChats files for users information. (See Generic network, below). |

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2. With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 4 next requires "determining the on-line status of the second process."

¶ 224. As described with respect to the previous claim element, in a TCP/IP implementation, the server on which the Connection List/USERS file is located transmits the network protocol address of a second VocalChat client (second process) to a first VocalChat client (first process) upon request. As described in the Help File:

Method of determining users address:

Netware	Get Users information from Netware 2.x/3.x bindery
WinWorkgroups	Get users information from Windows for Workgroups.
Generic User	VocalChats files for users information. (See Generic network, below).

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) determining the on-line status of other users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2. With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Additionally, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 4 finally requires "transmitting an indication of the on-line status of the second process to the first process over the computer network."

¶ 225. As described above, with TCP/IP, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, with TCP/IP, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that the server "transmits" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*; *see also id.*, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses").

INDEPENDENT CLAIM 10

The preamble of Claim 10 reads, in pertinent part: "In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network."

¶ 226. As discussed above, VocalChat clients connect to a server to locate and establish point-to-point connections with other VocalChat clients over a network. For example, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file" which is stored on a server. Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

Claim 10 also requires "providing a user interface element representing a first communication line."

¶ 227. A VocalChat user makes a point-to-point call to another user by using the VocalChat "Call" button, which is a user interface element representing a first communication line. *See, e.g.*, User Guide, page 14 ("Select Call from the Chat menu, or click on the tool bar Call button"). In addition, the VocalChat graphical user interface (GUI) includes a plurality of Quick Dial buttons. *See* User Guide, page 12. Depending on the implementation, either the Call button or the Quick Dial button comprises an "element representing a first communication line."

Claim 10 further requires "providing a user interface element representing a first callee process."

¶ 228. The VocalChat GUI displayed the names of potential callees in a dialog box. *See, e.g.*, Help File, page 14 ("just select a user from the user list, and choose "OK"). Callees are also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 ("Setting a Quick Dial Button"). Depending on the implementation, either the callee names listed within the dialog box or the Quick Dial buttons comprise "a user interface element representing a first callee process."

Claim 10 also requires "establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line."

¶ 229. As mentioned above, a VocalChat user makes a point-to-point call to another user with the Call button or a Quick Dial Button representing a frequently called callee. *See* Help File, page 14 (describing use of the Call button) and 20 (describing use of the Quick Dial buttons). Selecting the Call button opens a dialog box displaying a list of connected VocalChat users. A caller then clicks on a user's name in the list and then clicks the OK button to establish a point-to-point communication link. *See, e.g.,* Help File, page 14. In this example, the graphical representation of the user in the list is an "element representing the first callee process" and the OK button is an "element representing a first communication line." Alternatively, a user can associate any VocalChat user with a Quick Dial button by right-clicking on a Quick Dial button, which presents the user with the VocalChat users list. *See* Help File, page 20. After the user selects a user name from the list, that user is associated with the quick dial button. *See* Help File, page 21 ("From the user list, choose the user name that you want the button to hold."). The caller then places a call to the callee by selecting the Quick Dial button. VocalChat also assigns Quick Dial buttons automatically ("When you call a user with the Call command, a vacant button changes to hold the user's name if one does not hold it already."). In these examples, the graphical representation of the user in the list is an "element representing the first callee process" and the quick dial button is an "element representing a first communication line." In both cases, the element representing the callee process is associated with an element representing a communication line.

INDEPENDENT CLAIM 21

Claim 21 reads, in pertinent part: "A computer program product for use with a computer system comprising."

¶ 230. The techniques described in VocalChat are implemented in software, which is a "computer program product." In particular, VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols.").

Claim 21 requires "a computer usable medium having program code embodied in the medium for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network."

¶ 231. As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A"). As discussed above, VocalChat clients connect to a server to locate and establish point-to-point connections with other VocalChat clients over a network. For example, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file" which is stored on a server. Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

Claim 21 also requires "program code for generating an element representing a first communication line."

¶ 232. A VocalChat user makes a point-to-point call to another user by using the VocalChat "Call" button, which is a user interface element representing a first communication line. *See, e.g.*, User Guide, page 14 ("Select Call from the Chat menu, or click on the tool bar Call button"). In addition, the VocalChat graphical user interface (GUI) included a plurality of Quick Dial buttons. *See* User Guide, page 12. Depending on the implementation, either the Call button or the Quick Dial button comprises an "element representing a first communication line."

Claim 21 also requires "program code for generating an element representing a first callee process."

¶ 233. The VocalChat GUI displayed the names of potential callees in a dialog box. *See, e.g.*, Help File, page 14 ("just select a user from the user list, and choose "OK"). Callees were also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 ("Setting a Quick Dial Button"). Depending on the implementation, either the callee names listed within the dialog box or the Quick Dial buttons comprise "a user interface element representing a first callee process."

Claim 21 also requires "program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process."

¶ 234. As mentioned above, a VocalChat user makes a point-to-point call to another user with the Call button or a Quick Dial Button representing a frequently called callee. *See* Help File, page 14 (describing use of the Call button) and 20 (describing use of the Quick Dial buttons). Selecting the Call button opens a dialog box which displays a list of connected VocalChat users. A caller then clicks on a user's name in the list and then clicks the OK button to establish a point-to-point communication link. *See*, *e.g.,* Help File, page 14. In this example, the graphical representation of the user in the list is an "element representing the first callee process" and the OK button is an "element representing a first communication line." Alternatively, a user can associate any VocalChat user with a Quick Dial button by right-clicking on a Quick Dial button, which presents the user with the VocalChat users list. *See* Help File, page 20. Once the user selects a user name from the list, that user was associated with the quick dial button. *See* Help File, page 21 ("From the user list, choose the user name that you want the button to hold."). The caller may then place a call to the callee by selecting the Quick Dial button. VocalChat also assigns Quick Dial buttons automatically ("When you call a user with the Call command, a vacant button changes to hold the user's name if one does not hold it already."). In these examples, the graphical representation of the user in the list is an "element representing the first callee process" and the quick dial button is an "element representing a first communication line." In both cases, the element representing the callee process is associated with an element representing a communication line.

INDEPENDENT CLAIM 32

> **Claim 32 recites: "A method of locating a process over a computer network comprising the steps of: maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network."**

¶ 235. In the initial VocalChat implementations (versions 1.x) each VocalChat client transmits its name and network protocol address to a USERS file stored on a server. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users.
>
> Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file.*

VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), refer to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

¶ 236. While VocalChat does not explicitly describe a server with stored names and addresses is accessible over "the Internet," it describes the use of TCP/IP, which is the protocol used on the Internet. Thus, VocalChat inherently describes that the list of users and network addresses is accessible over the Internet. Moreover, the Internet is a type of Wide Area Network (WAN) and VocalChat describes a WAN implementation. For example, the Help File describes that "[o]ver a WAN . . . it is advisable to create local copy of the executables and DLLs, and reference only the Post Office over the low-speed [WAN] connection." Help File, page 4.

¶ 237. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network.

Claim 32 further requires "in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process."

¶ 238. As discussed above, a VocalChat caller sends the directory server a query identifying a particular callee. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server identifies an entry in the directory corresponding to the identified callee (the Connection List file in a TCP/IP implementation), and, if the callee is connected, provides the corresponding IP address associated with that callee in the directory to the caller.

INDEPENDENT CLAIM 33

The preamble of Claim 33 reads, in pertinent part: "A method for locating processes having dynamically assigned network protocol addresses over a computer network."

¶ 239. As described above, VocalChat clients rely on a central server to locate the network addresses of other VocalChat clients. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users.
>
> Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file.*

VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

¶ 240. On many networks, including TCP/IP networks, network addresses are assigned dynamically "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network.

> **Claim 33 also requires "maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network."**

¶ 241. The server on which the Connection List/USERS file is maintained inherently includes a computer memory. Moreover, the server stores a network accessible compilation of entries including a network protocol address and a name (corresponding identifier) of a process connected to the computer network. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users.
>
> Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file.*

VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help

File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

Claim 33 also requires that "in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process."

¶ 242. As discussed above, a VocalChat caller sends the directory server a query identifying a particular callee. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server identifies an entry in the directory corresponding to the identified callee (the Connection List file in a TCP/IP implementation), and, if the callee is connected, provides the corresponding IP address that is associated with that callee in the directory to the caller.

INDEPENDENT CLAIM 38

Claim 38 recites, in pertinent part: "A computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium the program code."

¶ 243. VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols."). As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A").

Claim 38 requires "program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network."

¶ 244. The server on which the Connection List/USERS file is maintained inherently includes a computer memory. Moreover, the server stores a network accessible compilation of entries including a network protocol address and a name (corresponding identifier) of a process connected to the computer network. As described in VocalChat Network Information:

> When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users</u>.
>
> <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the <u>user name is kept in the file</u>. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file.*

VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own computers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), refers to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

¶ 245. On many networks, including TCP/IP networks, network addresses are assigned dynamically "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network.

Claim 38 also requires "program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process."

¶ 246. As discussed above, a VocalChat caller sends the directory server a query identifying a particular callee. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server identifies an entry in the directory corresponding to the identified callee (the Connection List file in a TCP/IP implementation), and, if the callee is connected, provides the corresponding IP address that is associated with that callee in the directory to the caller.

INDEPENDENT CLAIM 43

Claim 43 recites, in pertinent part: "A computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein."

¶ 247. VocalChat is a software-based telephone executed on personal computers which connected to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols."). As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A").

> **Claim 43 requires "program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network."**

¶ 248. In a TCP/IP implementation, the server on which the Connection List/USERS file is located transmits the network protocol address of a second VocalChat client (second process) to a first VocalChat client (first process) upon request. As described in the Help File:

> Method of determining users address:
>
> | | |
> |---|---|
> | Netware | Get Users information from Netware 2.x/3.x bindery |
> | WinWorkgroups | Get users information from Windows for Workgroups. |
> | Generic User | VocalChats files for users information. (See Generic network, below). |

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2 (underline emphasis added). Thus, With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file."

VocalChat Network Information, page 10 (emphasis added). Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 43 also requires "program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network."

¶ 249. VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

INDEPENDENT CLAIM 44

Claim 44 recites, in pertinent part: "In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes."

¶ 250. VocalChat is a software-based telephone executed on personal computers which connected to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols.").

Claim 44 requires "following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network."

¶ 251. As discussed above, VocalChat clients connect to a server to locate and establish point-to-point connections with other VocalChat clients over a network. For example, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the

[Connection List] file" which is stored on a server. Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons."). Because the server stores network addresses of logged in clients, it is an "address server."

¶ 252. Inherently, a VocalChat client transmits its network protocol address "following connection of the [VocalChat client] to the computer network." Moreover, on many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network.

Claim 44 also requires "querying the address server as to whether the second process is connected to the computer network."

¶ 253. In a TCP/IP implementation, the server on which the Connection List/USERS file is located receives queries from VocalChat clients (first processes) to determine the on-line status of other VocalChat clients (second processes). As described in the Help File:

> Method of determining users address:
>
> | | |
> |---|---|
> | Netware | Get Users information from Netware 2.x/3.x bindery |
> | WinWorkgroups | Get users information from Windows for Workgroups. |
> | Generic User | VocalChats files for users information. (See Generic network, below). |

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2. Thus, with NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 44 also requires "receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network."

¶ 254. As described above, with TCP/IP, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, with TCP/IP, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that the server "transmits" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*; *see also id.*, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses").

Claim 44 further requires "in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network."

¶ 255. VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

DEPENDENT CLAIMS 5-7, 11, 19-20, 22, 30-31, 34-37, 39-42

Claim 5 of the '704 patent requires "searching the computer memory for an entry relating the second process; and retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process."

¶ 256. VocalChat inherently describes "searching the computer memory for an entry relating the second process." For example, as described above, VocalChat used a server to store names and network addresses of on-line users and to provide those network address to VocalChat clients upon request. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server was inherently capable of "searching the computer memory for an entry relating to the second process (i.e., a name and address of a VocalChat client).

Claim 6 of the '704 patent requires "transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network."

¶ 257. In a TCP/IP implementation, the server for the Connection List/USERS file receives queries from VocalChat clients (first processes) to determine the on-line status of other VocalChat clients (second processes). As described in the Help File:

> Method of determining users address:
> Netware Get Users information from Netware 2.x/3.x bindery
> WinWorkgroups Get users information from Windows for Workgroups.
> Generic User VocalChats files for users information. (See Generic

network, below).

Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

> When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network.

Help File, page 2. With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

> **Claim 7 requires "generating an off-line message when the second process is determined in step C to have a negative on-line status with respect to the computer network; and transmitting the off-line message to the first process."**

¶ 258. First, as described above, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction is made between on-line and off-line users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users." Moreover, various types of off-line messages are provided to indicate the unavailability of VocalChat users. *See, e.g.*, Troubleshooting Help File, page 2 (describing that when a user's name in the "New Users" dialog box, one of the causes may be that "[t]he 'Show only Logged-in' check-box is checked, and the person is not currently logged-in."). Consequently, the server inherently transmits "off-line messages" to the VocalChat clients to distinguish between online and offline users.

Claim 11 requires "querying the server as to the on-line status of the first callee process; and receiving a network protocol address of the first callee process over the computer network from the server."

¶ 259. As described above, with NetWare, the VocalChat client queries existing NetWare Bindery services to locate "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services to locate online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). *See, e.g.*, Help File, page 2. Regardless of protocol, the query determines the online status of the callee process (the VocalChat client of a callee). For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 19 requires "wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface."

¶ 260. The VocalChat client has a graphical user interface that is a "visual display." *See, e.g.*, User Guide, page 11 (illustrating the primary VocalChat GUI including a Call button, a volume slider and a plurality of Quick Dial buttons).

Claim 20 requires "wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface."

¶ 261. A VocalChat user makes a point-to-point call to another user with the Call button or a Quick Dial Button representing a frequently called callee. *See* User Guide, page 14 ("just select a user from the user list, and choose "OK"). Callees were also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 ("Setting a Quick Dial Button").

Claim 22 requires "program code for querying the server as to the on-line status of the first callee process; and program code for receiving a network protocol address of the first callee process over the computer network from the server."

¶ 262. As described above, with NetWare, the VocalChat client queries existing NetWare Bindery services to locate "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services to locate online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). *See, e.g.*, Help File, page 2. Regardless of protocol, the query determines the online status of the callee process (the VocalChat client of a callee). For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."

Claim 30 requires that the "computer system further comprises a visual display and the user interface comprises a graphic user interface."

¶ 263. The VocalChat client has a graphical user interface that is a "visual display." *See, e.g.*, User Guide, page 11 (illustrating the primary VocalChat GUI including a Call button, a volume slider and a plurality of Quick Dial buttons).

Claim 31 requires that "the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further

comprises: program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process."

¶ 264. A VocalChat user makes a point-to-point call to another user with the Call button or a Quick Dial Button representing a frequently called callee. *See* User Guide, page 14 ("just select a user from the user list, and choose "OK"). Callees were also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 ("Setting a Quick Dial Button").

Claim 34 requires "modifying the compilation of entries."

¶ 265. The compilation of entries stored on the server (e.g., in the Connection List/USERS file) is modified as new users install VocalChat software and as existing users log in and log off. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Later VocalChat implementations (e.g., version 2.02), refer to the USERS file as a "Connection List" file, which is modified in the same manner as the USERS file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory.").

Claim 35 requires "adding an entry to the compilation upon the occurrence of a predetermined event."

¶ 266. An entry is added to the compilation of entries within the Connection List/USERS file when a user first sets up VocalChat or when the user logs in to VocalChat. *See, e.g.*, VocalChat Network Information, page 10 ("Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address"); Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses . . . the user name for each user, is entered when performing the User Installation in the Setup program"); page 4 (user installation "is used to . . . add the user to the Address Book"); page 10 (user name "will be used by VocalChat to identify you and will appear in the VocalChat Address Book and in the Connection List file").

Claim 36 requires that "the predetermined event comprises notification by a user process of an assigned network protocol address."

¶ 267. When a user logs in, or when a computer with VocalChat is turned on, the network address of the VocalChat client is sent to the Connection List/USERS file. For example, as described in VocalChat Network Information, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address." VocalChat Network Information, page 10.

Claim 37 requires "deleting an entry from the compilation upon the occurrence of a predetermined event."

¶ 268. VocalChat inherently discloses this limitation. Any database containing entries, such as the one used on the server containing the Connection List/USERS file, is inherently capable of deleting entries upon request from an end user and/or a network administrator. Moreover, when a user logs off the system, the user's network address is deleted from the list of "on-line" users. *See, e.g.,* VocalChat Network Information, page 10 ("VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file.") (emphasis added).

Claim 39 requires "program code configured to modify the compilation of entries."

¶ 269. The compilation of entries stored on the server (e.g., in the Connection List/USERS file) is modified as new users install VocalChat software and as existing users log in and log off. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Later VocalChat implementations (e.g., version 2.02), refer to the USERS file as a "Connection List" file, which is modified in the same manner as the USERS file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory.").

Claim 40 requires "program code configured to add an entry to the compilation upon the occurrence of a predetermined event."

¶ 270. An entry is added to the compilation of entries within the Connection List/USERS file when a user first sets up VocalChat or when the user logs on to VocalChat. *See, e.g.*, VocalChat Network Information, page 10 ("Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address"); Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses . . . the user name for each user, is entered when performing the User Installation in the Setup program"); page 4 (user installation "is used to . . . add the user to the Address Book"); page 10 (user name "will be used by VocalChat to identify you and will appear in the VocalChat Address Book and in the Connection List file").

Claim 41 requires that the "predetermined event comprises notification by a process of an assigned network protocol address."

¶ 271. When a user logs in, or when a computer with VocalChat is turned on, the network address of the VocalChat client is sent to the Connection List/USERS file. For example, as described in VocalChat Network Information, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address." VocalChat Network Information, page 10.

Claim 42 requires "program code configured to delete an entry from the compilation upon the occurrence of a predetermined event."

¶ 272. VocalChat inherently discloses this limitation. Any database containing entries, such as the one used on the server containing the Connection List/USERS file, is inherently capable of deleting entries upon request from an end user and/or a network administrator. Moreover, when a user logs off the system, the user's network address is deleted from the list of "on-line" users. *See, e.g.,* VocalChat Network Information, page 10 ("VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. <u>When exiting VocalChat, the address is removed</u>, but the user name is kept in the file.") (emphasis added).

2. The VocalChat references further in view of RFC 1531

¶ 273. Claim 1-2, 4, 7, 10-11, 19-22, 30-42 should be rejected under 35 U.S.C. § 103 as being unpatentable over the VocalChat references and further in view of RFC 1531.

¶ 274. Claim 33 states that the network protocol address of the client computer system is "dynamically assigned." *See* Claim 33 ("A method for locating processes having dynamically assigned network protocol addresses over a computer network"). Other independent claims state, more generally, that the network protocol address is assigned or transmitted to the database following the connection of the computer to the computer network. *See* Claim 1 ("transmitting to the server a network protocol address received by the first process following connection to the computer network"); Claim 2 ("each network protocol address stored in the memory following connection of a respective process to the computer network"); Claim 4 ("the network protocol addresses received following connection of the respective process to the computer network"); Claim 32 ("the Internet Protocol address added to the list following connection of the process to the computer network"); Claim 38 ("the network protocol address of the corresponding process assigned to the process upon connection to the computer network"); Claim 43 ("the network protocol address of each respective process forwarded to the database following connection to the computer network"); Claim 44 ("following connection of the first process to the computer network forwarding to the address server a network protocol address").

¶ 275. As described above, VocalChat inherently describes these features. By way of example, on many networks, including the TCP/IP networks of VocalChat, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). For this reason, in at least some instances, the VocalChat computer system dynamically receives its IP address, following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system.

¶ 276. Alternatively, a SNQ of patentability of Claims 1-2, 4, 7, 10-11, 19-22, 30-42 is raised under 35 U.S.C. § 103 based on the VocalChat references in view of RFC 1531, which describes how TCP/IP addresses are dynamically assigned. *See, e.g.*, Dynamic Host

Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks).

3. Motivation to Combine the VocalChat references with RFC 1531

¶ 277. A motivation to combine VocalChat with RFC 1531 exists because VocalChat describes the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that VocalChat software would be installed and executed on personal computers that would frequently have their IP addresses dynamically assigned.

4. The VocalChat References in view of NetBIOS

¶ 278. Claim 3 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over VocalChat references in view of NetBIOS.

Claim 3 of the '704 patent requires "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory."

¶ 279. VocalChat does not describe a timer for time stamping network protocol address. However, time stamping was a well known technique at the time the application which resulted in the '704 patent was filed. For example, the NetBIOS Name Server ("NBNS") described in NetBIOS includes a timer for time-stamping name/IP address entries. As described in NetBIOS, "[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure." NetBIOS at 382. Similarly, as described in NetBIOS:

> If an end-node holds any names that have finite time-to-live values, then that node must periodically send a status report to the NBNS. Each name is reported using the NAME REFRESH REQUEST packet. These status reports restart the timers of both the NBNS and the reporting node. However, the only timers which are restarted are those associated with the name found in the status report. Timers on other names are not affected. *Id.*

5. Motivation to Combine the VocalChat References with NetBIOS

¶ 280. A motivation to combine VocalChat with NetBIOS explicitly exists within VocalChat. For example, NetBIOS is one of the network protocols explicitly supported by VocalChat. *See, e.g.*, Help File, page 2 ("When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, <u>a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses</u>.") (underline emphasis added). Thus, one of ordinary skill in the art would have been motivated to combine VocalChat with NetBIOS, because VocalChat explicitly states that NetBIOS may be used as the underlying network protocol.

6. The VocalChat References in view of Pinard

¶ 281. Claims 12-18 and 23-29 should be rejected under 35 U.S.C. § 103 as being unpatentable over the VocalChat references in view of Pinard.

CLAIMS 12-18 & 23-29

Claim 12 and 23 requires "providing an element representing a second communication line" and "program code for generating an element representing a second communication line," respectively.

¶ 282. The graphical user interface described in Pinard provides an element representing a second communication line. For example, call icons 23 and 29 representing two communication lines are shown in Figure 6 of Pinard. See Pinard, 5:31-40, Figure 6 ("Now there are clearly two calls in progress . . .").

Claims 13 and 24 require "terminating the point-to-point communication link from the caller process to the first callee process, in response to the user disassociating the element representing the first callee process from the element representing the first communication line" and "program code, responsive to the user disassociating the element representing the first callee

process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process," respectively.

¶ 283. Figure 6 of Pinard illustrates how a call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.

Claim 13 and 24 further require "establishing a different point-to-point communication link from the caller process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line" and "program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process," respectively.

¶ 284. In Figure 6, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, Col. 4, lines 22-31.

Claims 14 and 25 require "providing a user interface element representing a second callee process; and . . . establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line" and "program code for generating an element representing a second callee process; and program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process," respectively.

¶ 285. In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." See Pinard, Col. 5, lines 31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").

Claims 15 and 26 require "removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line" and "program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link," respectively.

¶ 286. In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, Col. 6, lines 14-15.

Claims 16 and 27 require "providing a user interface element representing a communication line having a temporarily disabled status" and "program code for generating an element representing a communication line having a temporarily disabled status," respectively.

¶ 287. Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 16 and 27 also require "temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status" and "program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process," respectively.

¶ 288. In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims 17 and 28 require that "the element provided in step D represents a communication line on hold status" and "the communication line having a temporarily disabled status comprises a communication line on hold status," respectively.

¶ 289. In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claim 18 and 29 require that "the element provided in step D represents a communication line on mute status" and "the communication line having a temporarily disabled status comprises a communication line on mute status," respectively.

¶ 290. VocalChat describes a "communication line on mute status." As described in the User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.

7. Motivation to Combine VocalChat and Pinard

¶ 291. A motivation to combine VocalChat and Pinard exists due to the problem to be solved. Like VocalChat, Pinard relates to the field of computer-implemented telephony, and in particular to a method of indicating the status of various calls, to a user. *See* Pinard, Col. 1, lines 5-7. Indeed, the graphical user interface described in Pinard could be used in any system that operates a telephony application on a personal computer or on a personal computer in conjunction with a server. *See* Pinard, Col. 1, lines 60-62; Col. 2, lines 41-45. One of ordinary skill in the art would have recognized that the particular design choices reflected in the graphical user interface of Pinard could readily be implemented within the context of the network telephony system described in VocalChat. In fact, as described above, VocalChat discloses a graphical user interface with some similar features to those described in Pinard.

VII. LIST OF EXHIBITS

Exhibit A **U.S. Patent No. 6,108,704 issued to Hutton et al. ("the '704 patent")**

Exhibit B **Protocols for X/Open PC Interworking SMB, Version 2, THE OPEN GROUP (1992) ("NetBIOS"), which <u>published as a single document</u> with:**

- Protocol Standard for a NetBIOS Service on a TCP/UDP Transport:

Concept and Methods, RFC 1001 (March 1987) ("RFC 1001"); and

- Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications, RFC 1002 (March 1987) ("RFC 1002").

Exhibit C **Etherphone: Collected Papers 1987-1988. The papers <u>published together as a single document</u> include:**

- Polle T. Zellweger, et al., *An Overview of the Etherphone System and its Applications*, IEEE CONFERENCE ON COMPUTER WORKSTATIONS (March 1988), 160-168 (hereinafter "Zellweger 1").

- Daniel C. Swinehart, *Telephone Management in the Etherphone System*, PROCEEDINGS OF THE IEEE/IEICE GLOBAL TELECOMMUNICATIONS CONFERENCE (November 1987), 1176-1180 (hereinafter "Swinehart 1").

- Douglas B. Terry and Daniel C. Swinehart, *Managing Stored Voice in the Etherphone System*, ACM TRANSACTIONS ON COMPUTER SYSTEMS 6(1) (February 1988), 3-27 (hereinafter "Terry").

- Daniel C. Swinehart, *System Support Requirements for Multi-media Workstations*, PROCEEDINGS OF THE SPEECHTECH '88 CONFERENCE (April 1988), 82-83 (hereinafter "Swinehart 2").

- Polle T. Zellweger, *Active Paths through Multimedia Documents*, DOCUMENT MANIPULATION AND TYPOGRAPHY, J.C. AN VILET (ED.), CAMBRIDGE UNIVERSITY PRESS (1988) (hereinafter "Zellweger 2")

Exhibit D **Harrick M. Vin, et al., *Multimedia Conferencing in the Etherphone Environment*, IEEE COMPUTER SOCIETY (October 1991) ("Vin")**

Exhibit E **Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531")**

Exhibit F **Pinard, et al., U.S. Patent No. 5,533,110 ("Pinard")**

Exhibit G **VocalChat User's Guide, Version 2.0 (1994) ("User's Guide")**

Exhibit H **VocalChat Readme File, Version 2.02 (June, 1994) ("Readme")**

Exhibit I **VocalChat 1.01 Networking Information ("VocalChat Networking")**

Exhibit J **VocalChat Information (July 18, 1994) ("Help File")**

Exhibit K **VocalChat Troubleshooting Help File (July 18, 1994) ("Troubleshooting Help File")**

Exhibit L **Declaration of VocalTec, Ltd., co-founder Alon Cohen**

Exhibit M **Claim Chart for NetBIOS**

Exhibit N **Claim Chart for Etherphone**

Exhibit O **Claim Chart for VocalChat**

Exhibit P **Comments on arguments made by Net2Phone's expert to distinguish over NetBIOS**

Exhibit Q **Comments on arguments made by Net2Phone's expert to distinguish over Etherphone**

Exhibit R **Comments on arguments made by Net2Phone's expert to distinguish over VocalChat**

Exhibit S **Plaintiff Net2Phone's Opening Claim Construction Brief (Oct. 18, 2007)**

Exhibit T **Reformatted Opening Claim Construction Brief of Skype Technologies, SA, Skype, Inc., and Ebay Inc (Oct 18, 2007)**

Exhibit U **Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007)**

Exhibit V **Reformatted Responsive Claim Construction Brief of Skype Technologies SA, Skype, Inc., and EBay Inc (Oct. 18, 2007)**

Exhibit W **Plaintiff Net2Phone, Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007)**

Exhibit X **Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc., and EBay Inc. (Oct. 19, 2007)**

VIII. CONCLUSION

For the reasons set forth above, it is clear that a SNQ of patentability is raised in connection with claims 1-7 and 10-44 of the '704 patent by this Request for *Ex Parte* Reexamination since claims 1-7 and 10-44 are anticipated and/or rendered obvious in view of the above-listed prior art references. Therefore, it is requested that this request for reexamination be granted and claims 1-7 and 10-44 all be finally rejected.

As identified in the attached Certificate of Service and in accordance with 37 CFR §§1.33(c) and 1.915(b)(6), a copy of the present request, in its entirety, is being served to the address of the attorney or agent of record.

Please direct all correspondence in this matter to the undersigned.

Respectfully submitted,

_/ET/____
Edwin Taylor
Registration No. 25,129

Dated:___/02-17-2009/

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
1279 Oakmead Parkway
Sunnyvale, California 94085-4040
Telephone: 408/720-8300
Facsimile: 408/720-8383
Attorney Docket No.: 03801.G164

CERTIFICATE OF SERVICE

The undersigned certifies that copies of the following:

(1) Request for *Ex Parte* Reexamination Transmittal Form; and

(2) Request for *Ex Parte* Reexamination including Exhibits A through X

were served on

JEFFREY S. GINSBERG, ESQ.
KENYON & KENYON
ONE BROADWAY
NEW YORK, NY 10004

_/ET/________________

Edwin H. Taylor

Dated: /02-17-2009/

the attorney of record for the assignee of USP 6,108,704 in accordance with 37 CFR § 1.915(b)(6), on the 17 day of February, 2008.

Exhibit A

U.S. Patent No. 6,108,704 issued to Hutton, et al. ("the '704 patent")

Customer No.: 8791

Blakely, Sokoloff, Taylor & Zafman, LLP
Sunnyvale, California 94085-4040
Telephone (408) 720-8300
Fax (408) 720-8383

US006108704A

United States Patent [19]

Hutton et al.

[11] **Patent Number: 6,108,704**

[45] **Date of Patent: Aug. 22, 2000**

[54] **POINT-TO-POINT INTERNET PROTOCOL**

[75] Inventors: **Glenn W. Hutton**, Miami; **Shane D. Mattaway**, Boca Raton; **Craig B. Strickland**, Tamarac, all of Fla.

[73] Assignee: **NetSpeak Corporation**, Boca Raton, Fla.

[21] Appl. No.: **08/533,115**

[22] Filed: **Sep. 25, 1995**

[51] **Int. Cl.**[7] **G06F 13/38**; G06F 15/17

[52] **U.S. Cl.** **709/227**; 709/204

[58] **Field of Search** 395/200.01, 200.02, 395/200.09, 200.11, 200.15, 200.34, 200.35, 200.47, 200.48, 200.57, 200.58, 200.75; 709/204, 205, 217, 218, 227, 228, 235

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,095,480	3/1992	Fenner .
5,150,360	9/1992	Perlman et al. .
5,166,931	11/1992	Riddle .
5,204,669	4/1993	Dorfe et al. .
5,224,095	6/1993	Woest et al. .
5,291,554	3/1994	Morales .
5,309,433	5/1994	Cidon et al. .
5,309,437	5/1994	Perlman et al. .
5,321,813	6/1994	McMillen et al. .
5,357,571	10/1994	Banwart .
5,400,335	3/1995	Yamada .
5,425,028	6/1995	Britton et al. .
5,430,709	7/1995	Galloway .
5,430,727	7/1995	Callon .

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

A2 0445402	11/1991	European Pat. Off. .
A2 0556012	8/1993	European Pat. Off. .
WO 9219054	10/1992	WIPO .

OTHER PUBLICATIONS

December & Randall, "The World Wide Web Unleased, "Samw Publishing, Indianapolis, IN, Dec. 1994, pp. 3–24.

Heylighen, "WorldWideWeb: a distributed hypermedia paradigm for global networking," IEEE/INSPEC Database Updates and Additionss (1960–19950 Doc.# 134618: Proceedings SHARE Spring Conference, pp. 355–368, Apr. 1994.

Internetworking with TCP/IP, vol. 1, Second Edition, Principles, Protocols, and Architecture, by Douglas E. Comer.; 1991; table of contents, pp. 1–3, 17–19, 311–333.

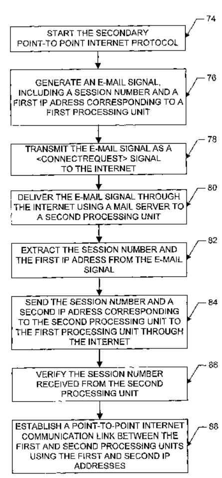
(List continued on next page.)

Primary Examiner—Mark H. Rinehart
Attorney, Agent, or Firm—Kudirka & Jobse, LLP

[57] **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.

44 Claims, 6 Drawing Sheets



U.S. PATENT DOCUMENTS

5,434,797	7/1995	Barris .
5,442,633	8/1995	Perkins et al. .
5,452,296	9/1995	Shimizu .
5,455,854	10/1995	Dilts et al. .
5,457,683	10/1995	Robins .
5,463,625	10/1995	Yasrebi .
5,469,500	11/1995	Satter et al. .
5,479,411	12/1995	Klein .
5,517,494	5/1996	Green .
5,524,110	6/1996	Danneels et al. .
5,524,254	6/1996	Morgan et al. 395/500
5,526,489	6/1996	Nilakantan et al. .
5,533,110	7/1996	Pinard et al. .
5,544,303	8/1996	Mraoteaux et al. .
5,546,582	8/1996	Brockmeyer et al. .
5,581,552	12/1996	Civanlar et al. 370/396
5,608,786	3/1997	Gordon .
5,740,231	4/1998	Cohn et al. .

OTHER PUBLICATIONS

VocalTec Internet Phone (TM) Version 2.5, www.cox.smu.edu/class/mis6386/people/stort/phone25.exe.

Weinberg, Netscape Conference and Cooltalk Meeting Room, www.q5.com.

Gull, Re: Getting IP address of PPP–connected Mac, <jgull–030495100535001@pm012–11.dialip.mich.net>.

Gull, Re: Internet Phone for Mac?, >jgull–1704950116450001@pm049–28.dialip.mich.net>.

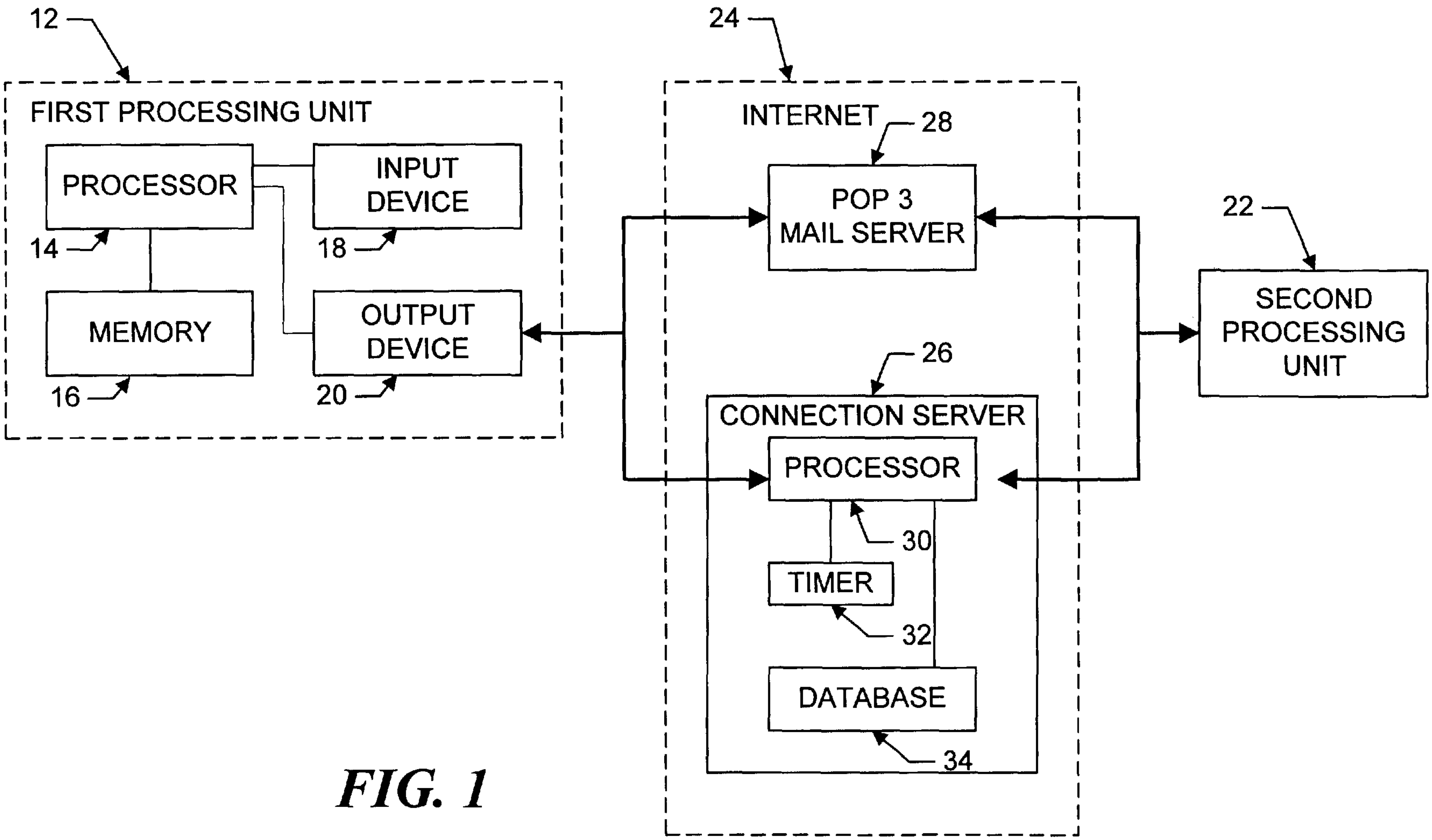

12
FIRST PROCESSING UNIT
PROCESSOR
14
INPUT DEVICE
18
MEMORY
16
OUTPUT DEVICE
20
24
INTERNET
28
POP 3 MAIL SERVER
26
CONNECTION SERVER
PROCESSOR
30
TIMER
32
DATABASE
34
22
SECOND PROCESSING UNIT

FIG. 1

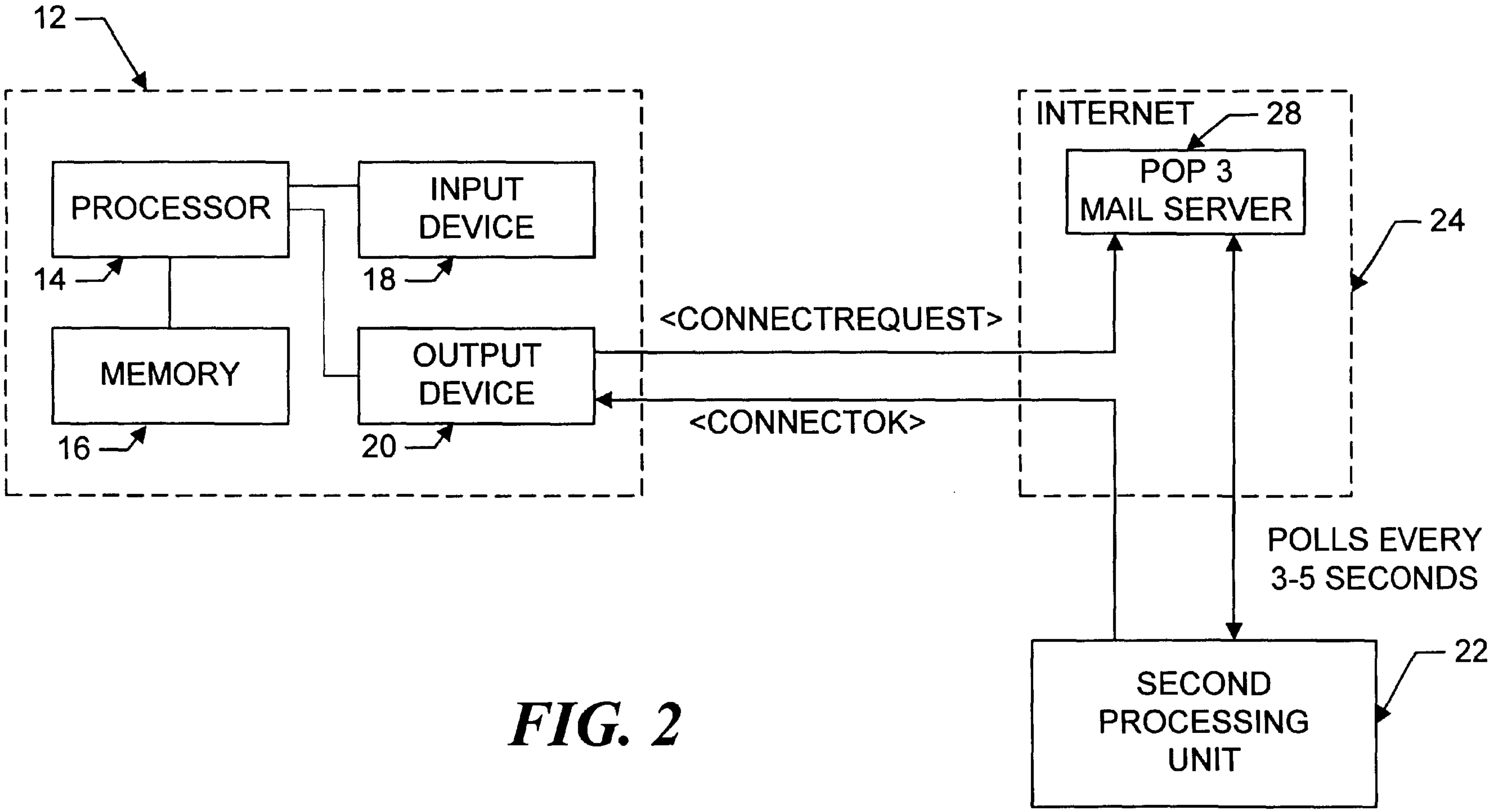

12
PROCESSOR
14
INPUT DEVICE
18
MEMORY
16
OUTPUT DEVICE
20
<CONNECTREQUEST>
<CONNECTOK>
INTERNET
28
POP 3 MAIL SERVER
24
POLLS EVERY 3-5 SECONDS
SECOND PROCESSING UNIT
22

FIG. 2

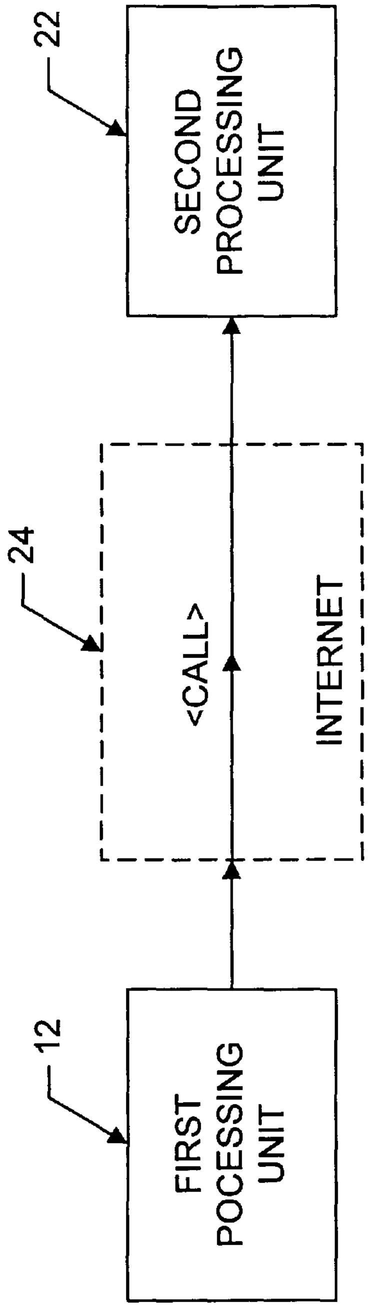

12
FIRST POCESSING UNIT
24
<CALL>
INTERNET
22
SECOND PROCESSING UNIT

FIG. 3

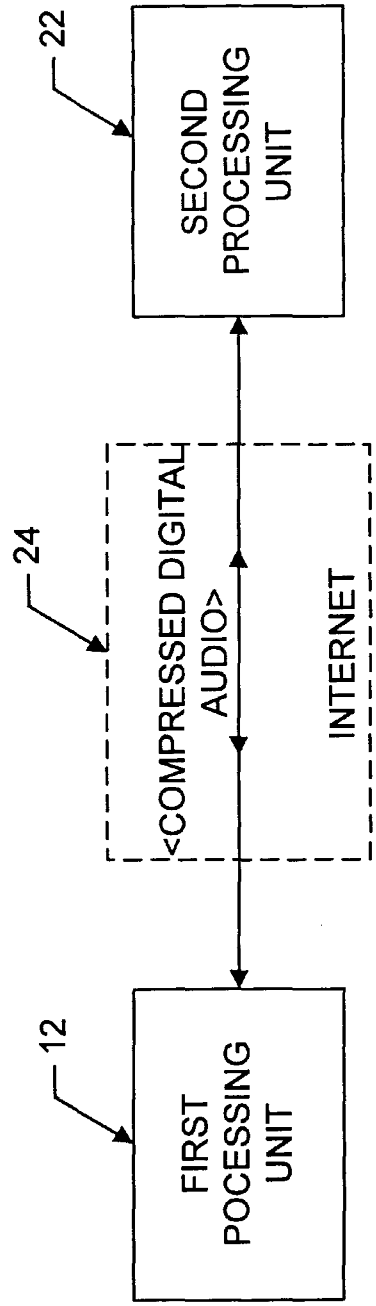

12
FIRST POCESSING UNIT
24
<COMPRESSED DIGITAL AUDIO>
INTERNET
22
SECOND PROCESSING UNIT

FIG. 4

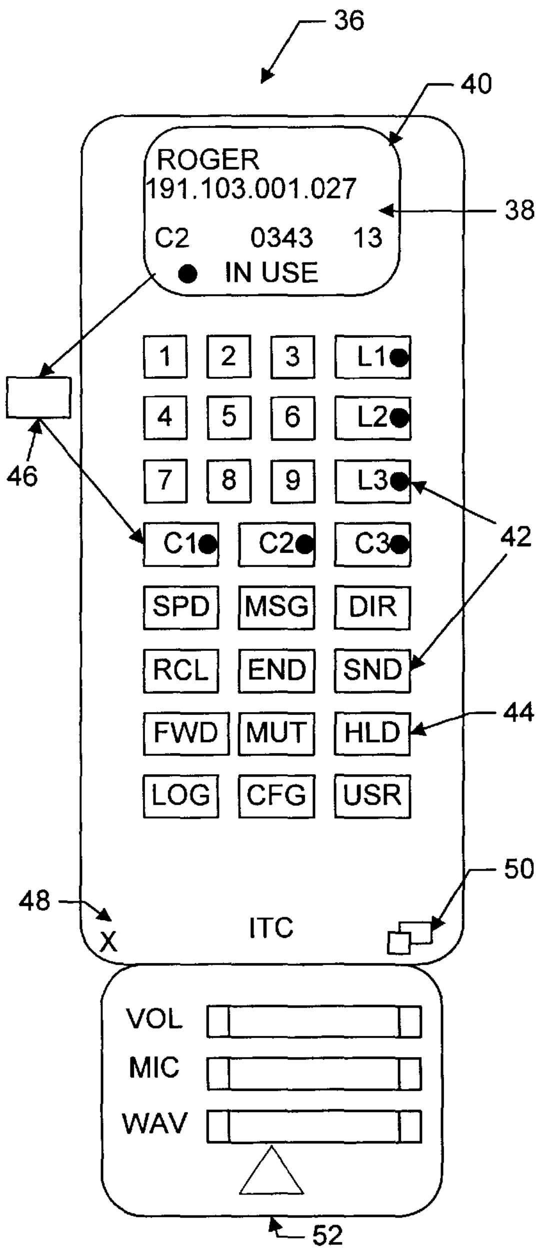

36
40
ROGER
191.103.001.027
38
C2 0343 13
IN USE
1 2 3 L1
4 5 6 L2
46
7 8 9 L3
C1 C2 C3
42
SPD MSG DIR
RCL END SND
FWD MUT HLD
44
LOG CFG USR
50
48
X
ITC
VOL
MIC
WAV
52

FIG. 5

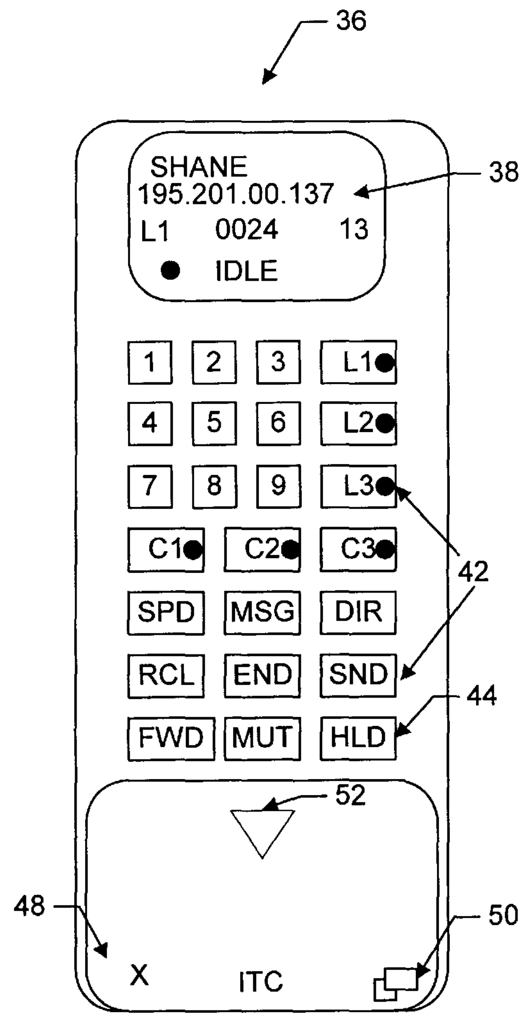

36
SHANE
38
195.201.00.137
L1 0024 13
IDLE
1 2 3 L1
4 5 6 L2
7 8 9 L3
C1 C2 C3
42
SPD MSG DIR
RCL END SND
44
FWD MUT HLD
52
48
50
X
ITC

FIG. 6

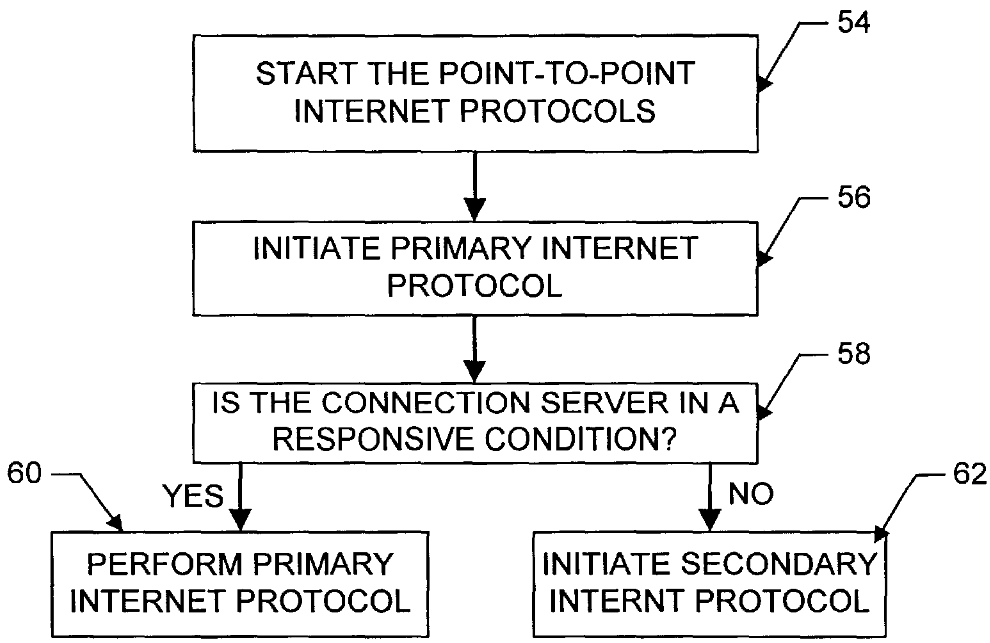

54
START THE POINT-TO-POINT INTERNET PROTOCOLS
56
INITIATE PRIMARY INTERNET PROTOCOL
58
IS THE CONNECTION SERVER IN A RESPONSIVE CONDITION?
60
YES
NO
62
PERFORM PRIMARY INTERNET PROTOCOL
INITIATE SECONDARY INTERNT PROTOCOL

FIG. 7

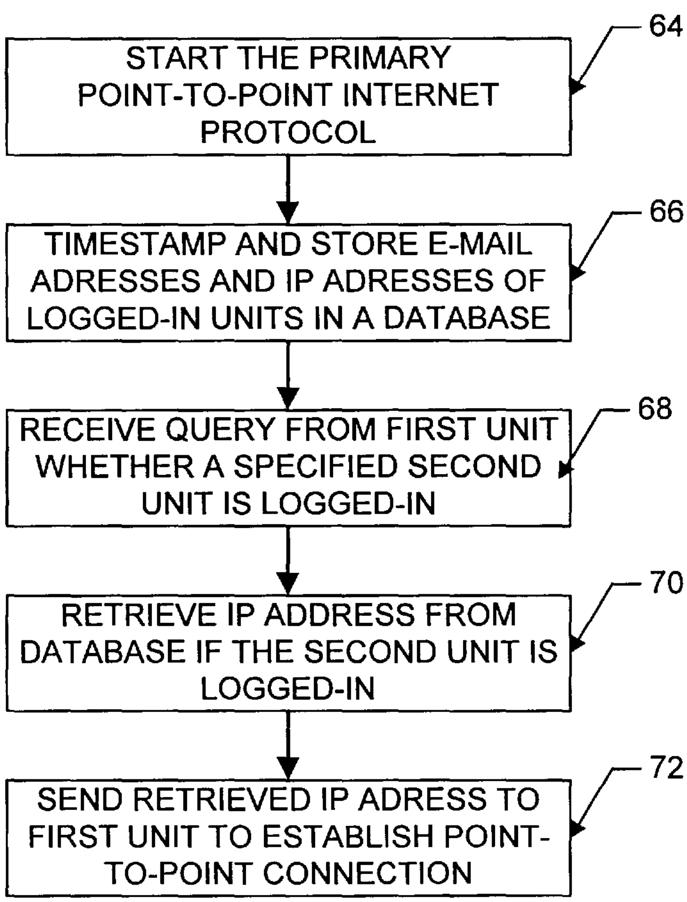

64
START THE PRIMARY POINT-TO-POINT INTERNET PROTOCOL
66
TIMESTAMP AND STORE E-MAIL ADRESSES AND IP ADRESSES OF LOGGED-IN UNITS IN A DATABASE
68
RECEIVE QUERY FROM FIRST UNIT WHETHER A SPECIFIED SECOND UNIT IS LOGGED-IN
70
RETRIEVE IP ADDRESS FROM DATABASE IF THE SECOND UNIT IS LOGGED-IN
72
SEND RETRIEVED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

FIG. 8

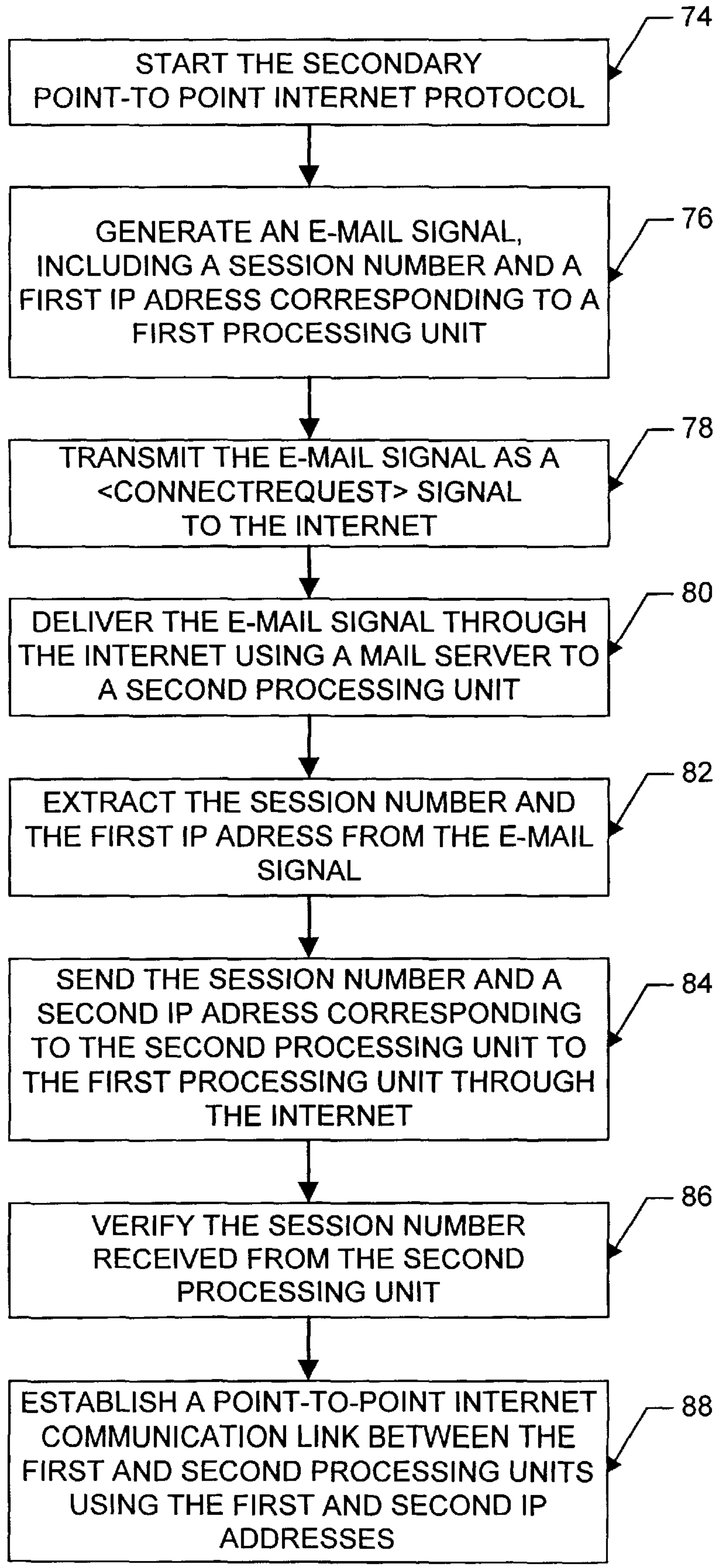

74
START THE SECONDARY POINT-TO POINT INTERNET PROTOCOL
76
GENERATE AN E-MAIL SIGNAL, INCLUDING A SESSION NUMBER AND A FIRST IP ADRESS CORRESPONDING TO A FIRST PROCESSING UNIT
78
TRANSMIT THE E-MAIL SIGNAL AS A <CONNECTREQUEST> SIGNAL TO THE INTERNET
80
DELIVER THE E-MAIL SIGNAL THROUGH THE INTERNET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

FIG. 9

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.

Generally, devices interfacing the to 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 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", and each user may be allocated a successive IP address of XXX.XXX.XXX.10, XXX.XXX.XXX.11, 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™ unit, available from NetSpeak Corporation, Boca Raton, Fla. 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. The processing units **12**, **22** may be operatively interconnected through the Internet **24** to 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 Veiw, Calif. 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 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 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 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** 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, Was. 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, Fla. 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 touch-screen 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 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 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 a connection service provider.

Upon the first user initiating the point-to-point Internet protocol when the first user is logged on to 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 a connection service provider, 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 N^{TH} 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 N^{TH} 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 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 **649-2** 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.

What is claimed is:

1. A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network, the computer program product comprising:

a computer usable medium having program code embodied in the medium, the program code comprising:

program code for transmitting to the server a network protocol address received by the first process following connection to the computer network;

program code for transmitting, to the server, a query as to whether the second process is connected to the computer network;

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

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

2. An apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising:

a processor;

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

a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of processes, each network protocol address stored in the memory following connection of a respective process to the computer network;

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

3. The computer server apparatus of claim **2** further comprising a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory.

4. A method for enabling point-to-point communication between a first process and a second process over a computer network, the method comprising the steps of:

A. receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network;

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

C. determining the on-line status of the second process; and

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

5. The method of claim **4** wherein step C further comprises the steps of:

c.1 searching the computer memory for an entry relating the second process; and

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

6. The method of claim **4** wherein step D further comprises the steps of:

d.1 transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network.

7. The method of claim **4** wherein step D further comprises the steps of:

d.1 generating an off-line message when the second process 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 process.

8. The method of claim **4** further comprising the steps of:

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

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

9. The method of claim **8** 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 process.

10. In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the method comprising the steps of:

A. providing a user interface element representing a first communication line;

B. providing a user interface element representing a first callee process; and

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

11. The method of claim **10** wherein step C further comprises the steps of:

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

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

12. The method of claim **10** further comprising the step of:

D. providing an element representing a second communication line.

13. The method of claim **12** further comprising the steps of:

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

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

14. The method of claim **10** further comprising the steps of:

D. providing a user interface element representing a second callee process; and

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

15. The method of claim **10** further comprising the step of:

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

16. The method of claim **10** further comprising the steps of:

D. providing a user interface element representing a communication line having a temporarily disabled status; and

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

17. The method of claim **16** wherein the element provided in step D represents a communication line on hold status.

18. The method of claim **17** wherein the element provided in step D represents a communication line on mute status.

19. The method of claim **10** wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface.

20. The method of claim **19** wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface.

21. A computer program product for use with a computer system comprising:

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

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

program code for generating an element representing a first callee process;

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

22. The computer program product of claim **21** wherein the program code for establishing a point-to-point communication link further comprises:

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

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

23. A computer program product of claim **21** further comprising:

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

24. The computer program product of claim **23** further comprising:

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

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

25. The computer program product of claim **21** further comprising:

program code for generating an element representing a second callee process; and

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

26. The computer program product of claim **25** further comprising:

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

27. The computer program product of claim **21** further comprising:

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

program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process.

28. The computer program product of claim **27** wherein the communication line having a temporarily disabled status comprises a communication line on hold status.

29. The computer program product of claim **27** wherein the communication line having a temporarily disabled status comprises a communication line on mute status.

30. A computer program product of claim **21** wherein the computer system further comprises a visual display and the user interface comprises a graphic user interface.

31. The computer program product of claim **30** wherein the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises:

program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process.

32. A method of locating a process over a computer network comprising the steps of:

a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network; and

b. in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process.

33. A method for locating processes having dynamically assigned network protocol addresses over a computer network, the method comprising the steps of:

a. maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and

b. in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process.

34. The method of claim **33** further comprising the step of:

c. modifying the compilation of entries.

35. The method of claim **34** wherein step c further comprises:

c.1 adding an entry to the compilation upon the occurrence of a predetermined event.

36. The method of claim **35** wherein the predetermined event comprises notification by a user process of an assigned network protocol address.

37. The method of claim **34** wherein step c further comprises:

c.1 deleting an entry from the compilation upon the occurrence of a predetermined event.

38. A computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium the program code comprising:

a. program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and

b. program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process.

39. The computer program product of claim **38** further comprising:

c. program code configured to modify the compilation of entries.

40. The computer program product of claim **39** wherein program code configured to modify comprises:

c.1 program code configured to add an entry to the compilation upon the occurrence of a predetermined event.

41. The computer program product of claim **40** wherein the predetermined event comprises notification by a process of an assigned network protocol address.

42. The computer program product of claim **38** wherein step c further comprises:

c.1 program code configured to delete an entry from the compilation upon the occurrence of a predetermined event.

43. A computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein, the program code comprising:

a. program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network; and

b. program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network.

44. In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of:

A. following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network;

B. querying the address server as to whether the second process is connected to the computer network;

C. receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network; and

D. in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network.

* * * * *

Exhibit M

Claim chart for Claims 1-7 and 10-44 of the '704 patent:

- **NetBIOS (Claims 1-7 and 32-44) under 35 U.S.C. § 102(b)**
- **NetBIOS in view of RFC 1531 (Claims 1-7 and 32-44) under 35 U.S.C. § 103(a)**
- **NetBIOS in view of Pinard (Claims 10-17, 19-28, and 30-31) under 35 U.S.C. § 103(a)**
- **NetBIOS in view of Pinard and further in view of VocalChat User's Guide (Claims 18 and 29) under 35 U.S.C. § 103(a)**

Blakely, Sokoloff, Taylor & Zafman, LLP
Sunnyvale, California 94085-4040
Telephone (408) 720-8300
Fax (408) 720-8383

Exhibit M
Claim Chart – NetBIOS

Claims	Prior Art and Relevant Statute
1. A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network, the computer program product comprising:	**35 U.S.C. § 102 – NetBIOS** NetBIOS discloses a computer program product for use with a computer system which executes a "first process" and is operatively connectable to a "second process" and a server over a computer network. That NetBIOS discloses a computer program product for use with a computer system can be seen from 356 ("The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems."). In addition, NetBIOS describes that the computer systems (or "nodes") execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS discloses a "server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates a NetBIOS Name Server ("NBNS") coupled to point-to-point nodes ("P nodes") over the Internet. In the pending litigation, Net2Phone argued that the term "server" should be defined broadly. Plaintiff Net2Phone, Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), page 3. More specifically, Net2Phone argued: > Consistent with the use of the term 'server' in the specification, the claims do not refer to any specific server configuration. They simply require a 'server' (also referred to as a 'connection server,' 'address server,' or 'server process'). There is nothing in any of the claims that require that the server be in the form of a single computer with a centralized database, as defendants contend. *Id.*, page 4. Similarly, Net2Phone argued that "[a] server in a 'client/server system' can be implemented in any number of ways, from one to multiple computers, in one location or many, and from a single large computer acting as the server to a network of personal computers." Plaintiff Net2Phone Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007) (Exhibit W), page 7 (citing to the declaration of Professor Larry L. Peterson). Thus, under Net2Phone's interpretation, a "server" is not limited to any

Claims	Prior Art and Relevant Statute
	particular hardware or software configuration. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Oct. 19, 2007) (Exhibit X), pages 2-9. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a "server." Under any interpretation, the NBNS described in NetBIOS is a "server."
a computer usable medium having program code embodied in the medium, the program code comprising:	NetBIOS applications are loaded into random access memory, which is a computer usable medium, and executed by a computer processor. NetBIOS describes that "[o]ne of the first implementations [of NetBIOS] was for personal computers running the PC-DOS and MS-DOS operating systems. It is possible to implement NetBIOS within other operating systems, or as processes which are, themselves, simply application programs as far as the host operating system is concerned." *id* at 359. NetBIOS further discloses that "typical use of NetBIOS is among independently-operated personal computers." *Id.* at 360.
program code for transmitting to the server a network protocol address received by the first process following connection to the computer network;	NetBIOS discloses program code executed on a node that transmits its name and IP address to the NBNS. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *id.* 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NetBIOS Name Server includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *id.* 431. *See also* NetBIOS at 367 (describing how the NetBIOS Name Server may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and *id.* 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. On many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2

Claims	Prior Art and Relevant Statute
	(describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in NetBIOS. **35 U.S.C. § 103 – NetBIOS in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine NetBIOS with RFC 1531 exists because NetBIOS describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine NetBIOS with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of NetBIOS would frequently have their IP addresses dynamically assigned.
program code for transmitting, to the server, a query as to whether the second process is connected to the computer network;	As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *id.* 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NetBios Name Server for the IP address and other information of the target node with whom they wish to communicate. *Id. See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NetBIOS Name Server "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446.

Claims	Prior Art and Relevant Statute
program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network; and	NetBIOS states that the NetBIOS Name Server "answers queries from a P node with a list of IP address and other information for" the target name. *id.* 389. If the NetBIOS Name Server has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *id.* 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Thus, by design, only logged-in nodes are registered with the NetBIOS Name Server. *See, e.g., id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NetBIOS Name Server only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g., id.* at 389. In Claim Construction Briefs filed in the pending litigation, the patentee argued that the term 'connected' means 'logged on,' and *vice versa* . . . To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect.

Claims	Prior Art and Relevant Statute
	While Net2Phone's invention endeavors to identify accurately who is on line, it is not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user's Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person's status— *e.g.*, setting a default value of two hours, after which the server assumes that a party has gone off-line if it has not heard from her. *See* '704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, "the on-line status information stored in the database is ***relatively*** *current*." *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term "connected" (or "on-line") is going to be modified at all, it should be modified to say "*relatively* currently connected," because that is what the patents actually say. Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), pages 24-25. Thus, under Net2Phone's interpretation, the information retained in the "server" as to which processes are "connected to the computer network" or "online" may be imperfect. As described above, while the server "endeavors to identify accurately who is on line, it is not possible to achieve perfection." *Id.* As described above, NetBIOS employs similar techniques as NBNS entries for off-line nodes are removed through the use of log-out messages and timers. Once again, the requestor of the present reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Exhibit X), pages 12-14. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a process to be "connected to" the computer network or "on-line." Under any interpretation, a first NetBIOS process receives the network protocol address of a second NetBIOS process from the NBNS when the second NetBIOS process is "connected to the computer network."
program code, responsive to the network protocol address of the second process, for establishing a point-to-point	Once the node seeking to initiate the communication has obtained from the NetBIOS Name Server the IP address for the node to receive the communication, a point-to-point communication is

Claims	Prior Art and Relevant Statute
communication link between the first process and the second process over the computer network.	established between the nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only directed (point-to-point) communications." NetBIOS at 397 (emphasis added). *See also id.* at 401: This first diagram shows the sequence of network events used to successfully establish a session without retargeting by the listener. The TCP connection is first established with the well-known NetBIOS session service TCP port, SSN_SRVC_TCP_PORT. The caller then sends a SESSION REQUEST packet over the TCP connection requesting a session with the listener. The SESSION REQUEST contains the caller's name and the listener's name. The listener responds with a POSITIVE SESSION RESPONSE informing the caller this TCP connection is accepted as the connection for the data transfer phase of the session. *See also id.* at 398-400 ("16.1: Overview of NetBIOS Session Service"), 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are fullduplex, sequenced, and reliable. Data is organized into messages."). In sum, NetBIOS discloses all of the elements of, and hence anticipates, claim 1 of the '704 Patent.
2. An apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising:	**35 U.S.C. § 102 – NetBIOS** NetBIOS describes a NBNS which is coupled to M and P nodes over the Internet (or other network). *See, e.g.*, NetBIOS, page 371 (illustrating a NBNS) coupled to point-to-point nodes ("P nodes") over the Internet).
a processor; a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network;	The NBNS executes software using a processor and is inherently coupled to the Internet (or other network) via a network interface. *See, e.g.*, *id.*, page 359 ("[o]ne of the first implementations [of NetBIOS] was for personal computers running the PC-DOS and MS-DOS operating systems. It is possible to implement NetBIOS within other operating systems, or as processes which are, themselves, simply application programs as far as the host operating system is concerned."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications.").
a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of	As described above, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a

Claims	Prior Art and Relevant Statute
processes, each network protocol address stored in the memory following connection of a respective process to the computer network;	distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). This registration must inherently be stored in a "memory." A NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." NetBIOS, page 431. *See also id.* at 367 (describing how the NetBIOS Name Server may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby stores in a memory a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. The P and M nodes execute software which is inherently a computer-implemented "process." *See, e.g., id.* at 356 ("NetBIOS defines a software interface . . . .); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). On many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in NetBIOS. **35 U.S.C. § 103 – NetBIOS in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine NetBIOS with RFC 1531 exists because NetBIOS describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine NetBIOS with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the

Claims	Prior Art and Relevant Statute
	alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of NetBIOS would frequently have their IP addresses dynamically assigned.
means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process.	NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Thus, by design, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.

Claims	Prior Art and Relevant Statute
3. The computer server apparatus of claim 2 further comprising a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory.	**35 U.S.C. § 102 – NetBIOS** The NBNS includes a timer for time-stamping name/IP address entries. For example, "[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure." *Id.* at 382. Similarly, as described in NetBIOS: If an end-node holds any names that have finite time-to-live values, then that node must periodically send a status report to the NBNS. Each name is reported using the NAME REFRESH REQUEST packet. These status reports restart the timers of both the NBNS and the reporting node. However, the only timers which are restarted are those associated with the name found in the status report. Timers on other names are not affected. *Id.*
4. A method for enabling point-to-point communication between a first process and a second process over a computer network, the method comprising the steps of:	**35 U.S.C. § 102 – NetBIOS** NetBIOS describes a method for enabling point-to-point communication between a first process (on a first node) and a second process (on a second node) over a computer network. As discussed above, the NBNS is used to resolve IP addresses of point-to-point end-nodes to facilitate point-to-point communications between such nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name. . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>." *Id.* at 397 (emphasis added).
A. receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network;	NetBIOS describes the NBNS receiving and storing the names and IP addresses of processes in its memory. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See* NetBIOS at page 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other

Claims	Prior Art and Relevant Statute
	attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Consequently, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NetBIOS Name Server only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389. On many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in NetBIOS.

Claims	Prior Art and Relevant Statute
	35 U.S.C. § 103 – NetBIOS in view of RFC 1531 Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine NetBIOS with RFC 1531 exists because NetBIOS describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine NetBIOS with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of NetBIOS would frequently have their IP addresses dynamically assigned.
B. receiving a query from the first process to determine the on-line status of the second process;	As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *Id.* at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id. See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446.
C. determining the on-line status of the second process; and	the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. For instance, when the NBNS receives a query for a target node's IP address, it performs a look search in its directory database for the target's current IP address. *Id.* at 389

Claims	Prior Art and Relevant Statute
	(The NetBIOS Name Server "answers queries from a P node with a list of IP address and other information for" the target name.). The NetBIOS Name Server determines an end-node with the target name is currently registered in its database, and hence is deemed to be on-line. *See id.* at 376 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). As described above, the NBNS employs various mechanisms for determining the on-line status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database.
D. transmitting an indication of the on-line status of the second process to the first process over the computer network.	If the end-node with the target name is currently registered in the NBNS database, the NBNS responds with a positive name query response. *See, e.g., id.* at 389 (The NetBIOS Name Server "answers queries from a P node with a list of IP address and other information for" the target name.); *id.* at 440 ("NAME QUERY REQUEST"), *id.* at 441 ("POSITIVE NAME QUERY RESPONSE"), *id.* at 464-465 ("P-Node Find Name Procedure"). A positive name query response includes the IP address for the target end-node, *id.* at 441, which is an indication that the target node has a positive on-line status. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). A negative name query response from the NetBIOS Name Server may include a message that "[t]he name requested does not exist" in the NBNS database, which is an indication that the target node has an off-line status. *See, e.g., id.* at 442; *see also id.* at 484.
5. The method of claim 4 wherein step C further comprises the steps of:	**35 U.S.C. § 102 – NetBIOS**
c.1 searching the computer memory for an entry relating the second process; and c.2 retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process.	These features are inherent in the NetBIOS. In order for the NBNS server to identify the IP address of a process in response to a query, it must inherently search its memory for an entry related to the process. For example, NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP

Claims	Prior Art and Relevant Statute
	address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.
6. The method of claim 4 wherein step D further comprises the steps of:	**35 U.S.C. § 102 – NetBIOS**
d.1 transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network.	NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For

Claims	Prior Art and Relevant Statute
	example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). For these reasons, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.
7. The method of claim 4 wherein step D further comprises the steps of:	**35 U.S.C. § 102 – NetBIOS**
d.1 generating an off-line message when the second process 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 process.	The requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389 (illustrating a negative response "if the NBNS has no information about the name."). As mentioned above, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release

Claims	Prior Art and Relevant Statute
	typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder.").
10. In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the method comprising the steps of:	**35 U.S.C. § 103 – NetBIOS in view of Pinard** As discussed above, NetBIOS discloses a method of establishing a point-to-point communication link between nodes over a computer network such as a local network or the Internet. "NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>." *Id.* at 397 (emphasis added). *See also id.* at 361 (describing how a call to a named callee process is used to "[i]nitiate a session with a process that is listening under the specified name. The calling entity must indicate both a <u>calling name (properly registered to the caller) and a called name</u>) (emphasis added); *id.* at 359 ("NetBIOS applications employ NetBIOS mechanisms to locate resources, establish connections, send and receive data with an application peer, and terminate connections."); *id.* at 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are full-duplex, sequenced, and reliable. Data is organized into messages."). Applications which utilize NetBIOS application services inherently include "user interfaces." For example,

Claims	Prior Art and Relevant Statute
	"NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems." *Id.* at 356. The IBM PC included a text-based user interface known as PC-DOS. *See, e.g.*, id. (it is expected that on computers operating under the PC-DOS and MS-DOS operating systems that the existing NetBIOS interface will be preserved by implementers).
A. providing a user interface element representing a first communication line;	Pinard discloses a user interface element representing a first communication line. For example, Figure 6 of Pinard illustrates a first call icon 23 which represents a first communication line and a second call icon 29 which represents a second communication line. In the example shown in Figure 6, the first call icon 23 represents a telephone call between "Debbie" and "John" and the second call icon 29 represents a telephone call between "Debbie" and "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.
B. providing a user interface element representing a first callee process; and	Pinard describes "a user interface element representing a first callee process." In the example shown in Figure 6 of Pinard, a first user interface element 21 is shown for the callee named "John" and a second user interface element is shown for the callee named "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.
C. establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line.	As described above, NetBIOS describes establishing a point-to-point communication link between nodes. *See, e.g.*, NetBIOS at 397 ("NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>.") (emphasis added). Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
11. The method of claim 10 wherein step	**35 U.S.C. § 103 – NetBIOS in view of Pinard**

Claims	Prior Art and Relevant Statute
C further comprises the steps of:	
c.1 querying the server as to the on-line status of the first callee process; and c.2 receiving a network protocol address of the first callee process over the computer network from the server.	As disclosed in NetBIOS, an end-node sends a "query" to the NetBIOS Name Server to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NetBIOS Name Server. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *Id.* at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NetBios Name Server for the IP address and other information of the target node with whom they wish to communicate. *Id. See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NetBIOS Name Server "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446. If the end-node with the target name is currently registered in the NBNS database, the NBNS responds with a positive name query response. *See, e.g., id.* at 389 (The NBNS "answers queries from a P node with a list of IP address and other information for" the target name.).
12. The method of claim 10 further comprising the step of:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
D. providing an element representing a second communication line.	The graphical user interface described in Pinard provides an element representing a second communication line. For example, call icons 23 and 29 representing two communication lines are shown in Figure 6 of Pinard. *See* Pinard, Col. 5, lines 31-40, Figure 6 ("Now there are clearly two calls in progress . . .").
13. The method of claim 12 further comprising the steps of:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
E. terminating the point-to-point communication link from the caller process to the first callee process, in response to the user disassociating the element representing the first callee process from the element representing the first communication line; and	Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.

Claims	Prior Art and Relevant Statute
F. establishing a different point-to-point communication link from the caller process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line.	In Figure 6 of Pinard, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, Col. 4, lines 22-31.
14. The method of claim 10 further comprising the steps of:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
D. providing a user interface element representing a second callee process; and	In Figure 6 of Pinard, the user interface element for "John" 21 represents a second callee process.
E. establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line.	In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." See Pinard, Col. 5, lines 31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
15. The method of claim 10 further comprising the step of:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
F. removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line.	In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, Col. 6, lines 14-15.
16. The method of claim 10 further comprising the steps of:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
D. providing a user interface element	Pinard describes a user interface element representing a

Claims	Prior Art and Relevant Statute
representing a communication line having a temporarily disabled status; and	communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
E. temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status.	In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
17. The method of claim 16 wherein the element provided in step D represents a communication line on hold status.	**35 U.S.C. § 103 – NetBIOS in view of Pinard** In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
18. The method of claim 17 wherein the element provided in step D represents a communication line on mute status.	**35 U.S.C. § 103 – NetBIOS in view of Pinard and further in view of VocalChat User's Guide** The VocalChat User's Guide describes a "communication line on mute status" as recited in Claim 18. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.
19. The method of claim 10 wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface.	**35 U.S.C. § 103 – NetBIOS in view of Pinard** Pinard discloses a graphical user interface on a visual display which allows the caller to control the operation of the telephone. *See, e.g.*, Pinard, Figures 2-16 and Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
20. The method of claim 19 wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface.	**35 U.S.C. § 103 – NetBIOS in view of Pinard** As described above, Pinard discloses that a point-to-point communication link is established in response to a user associating a graphic element representing a callee process with a graphic element representing a communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a

Claims	Prior Art and Relevant Statute
	callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
21. A computer program product for use with a computer system comprising: a computer usable medium having program code embodied in the medium for	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the medium further comprising:	As discussed above, NetBIOS discloses a method of establishing a point-to-point communication link between a caller process and a callee process over a computer network such as a local network or the Internet. "NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>." NetBIOS at 397 (emphasis added). *See also id.* at 361 (describing how a call to a named callee process is used to "[i]nitiate a session with a process that is listening under the specified name. The calling entity must indicate both a <u>calling name (properly registered to the caller) and a called name</u>) (emphasis added); *id.* at 359 ("NetBIOS applications employ NetBIOS mechanisms to locate resources, establish connections, send and receive data with an application peer, and terminate connections."); *id.* at 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are full-duplex, sequenced, and reliable. Data is organized into messages."). NetBIOS applications include "user interfaces," for example, "NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems." *Id.* at 356. The IBM PC included a text-based user interface known as PC-DOS. *See, e.g.*, id. (it is expected that on computers operating under the PC-DOS and MS-DOS operating systems that the existing NetBIOS interface will be preserved by implementers).
program code for generating an element representing a first communication line;	Pinard discloses program code for generating an element representing a first communication line. For example, Figure 6 of Pinard illustrates a first call icon 23 which represents a first communication line and a second call icon 29 which represents a

Claims	Prior Art and Relevant Statute
	second communication line. In the example shown in Figure 6, the first call icon 23 represents a telephone call between "Debbie" and "John" and the second call icon 29 represents a telephone call between "Debbie" and "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.
program code for generating an element representing a first callee process;	Pinard describes program code for generating an element representing a first callee process. In the example shown in Figure 6 of Pinard, a first user interface element 21 is shown for the callee named "John" and a second user interface element is shown for the callee named "Mary." *See, e.g.*, Pinard, Col. 5, lines 23-30.
program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process.	As described above, NetBIOS describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, NetBIOS at 397 ("NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve <u>only directed (point-to-point) communications</u>.") (emphasis added). Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
22. The computer program product of claim 21 wherein the program code for establishing a point-to-point communication link further comprises:	**35 U.S.C. § 102 – NetBIOS**
program code for querying the server as to the on-line status of the first callee process; and program code for receiving a network protocol address of the first callee process over the computer network from the server.	As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." NetBIOS at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NetBios Name Server for the IP

Claims	Prior Art and Relevant Statute
	address and other information of the target node with whom they wish to communicate. *Id.* *See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446. If the end-node with the target name is currently registered in the NBNS database, the NBNS responds with a positive name query response. *See, e.g., id.* at 389 (The NBNS "answers queries from a P node with a list of IP address and other information for" the target name.).
23. A computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
program code for generating an element representing a second communication line.	Pinard describes an element representing a second communication line. For example, Figure 6 of Pinard illustrates a first element (23) representing a first communication line and a second element (29) representing a second communication line.
24. The computer program product of claim 23 further comprising:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
program code, responsive to the user disassociating the element representing the first callee process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process; and	Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.
program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process.	In Figure 6 of Pinard, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, Col. 4, lines 22-31.

Claims	Prior Art and Relevant Statute
25. The computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
program code for generating an element representing a second callee process; and	In Figure 6 of Pinard, the user interface element for "John" 21 represents a second callee process.
program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process.	In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." See Pinard, Col. 5, lines 31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
26. The computer program product of claim 25 further comprising:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link.	In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, Col. 6, lines 14-15.
27. The computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – NetBIOS in view of Pinard**
program code for generating an element representing a communication line having a temporarily disabled status; and	Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling	In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").

Claims	Prior Art and Relevant Statute
the point-to-point communication link between the caller process and the first callee process.	
28. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on hold status.	**35 U.S.C. § 103 – NetBIOS in view of Pinard** In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, Col. 6, lines 36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
29. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on mute status.	**35 U.S.C. § 103 – NetBIOS in view of Pinard and further in view of VocalChat User's Guide** The VocalChat User's Guide describes a "communication line on mute status" as recited in Claim 29. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.
30. A computer program product of claim 21 wherein the computer system further comprises a visual display and the user interface comprises a graphic user interface.	**35 U.S.C. § 103 – NetBIOS in view of Pinard** Figures 2-16 of Pinard illustrate a graphical user interface for managing telephone calls (which is inherently rendered on a "visual display").
31. The computer program product of claim 30 wherein the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises: program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process.	**35 U.S.C. § 103 – NetBIOS in view of Pinard** As described above, Pinard discloses that a point-to-point communication link is established in response to a user associating a graphic element representing the first callee process with a graphic element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, Col. 4, lines 38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, Col. 5, lines 36-37 ("Now

Claims	Prior Art and Relevant Statute
	to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
32. A method of locating a process over a computer network comprising the steps of:	**35 U.S.C. § 102 – NetBIOS**
a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network; and	NetBIOS describes the NBNS maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See* NetBIOS, page 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NetBIOS Name Server includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); NetBIOS at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." NetBIOS at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to

Claims	Prior Art and Relevant Statute
	their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389. On many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "following connection" to the computer network. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in NetBIOS. **35 U.S.C. § 103 – NetBIOS in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine NetBIOS with RFC 1531 exists because NetBIOS describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine NetBIOS with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of NetBIOS would frequently have their IP addresses dynamically assigned.

Claims	Prior Art and Relevant Statute
b. in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process.	A NBNS "answers queries from a P node with a list of IP address and other information for" the target name. NetBIOS at 389. If the NBNS has an entry in its database for the target name identified by the requesting node, the NBNS returns a positive name query response to the requesting node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 389. *See also id.* at 441 ("POSITIVE NAME QUERY RESPONSE").
33. A method for locating processes having dynamically assigned network protocol addresses over a computer network, the method comprising the steps of:	**35 U.S.C. § 102 – NetBIOS** Because IP addresses were known to be dynamically assigned on TCP/IP networks such as the Internet, this feature is inherent in NetBIOS. *See, e.g.*, RFC 1531, Dynamic Host Configuration Protocol (1993), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). One of ordinary skill in the art would understand that the use and implementation of a NBNS enables locating point-to-point nodes that have dynamically assigned network addresses. Alternatively, it would have been obvious to combine NetBIOS with other references such as RFC 1531 which describe the use of dynamically assigned IP addresses.
a. maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and	NetBIOS describes the NBNS maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NetBIOS Name Server includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NetBIOS Name Server may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); NetBIOS at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an

Claims	Prior Art and Relevant Statute
	endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Thus, by design, only logged-in nodes are registered with the NetBIOS Name Server. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389. On many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "upon connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in NetBIOS. **35 U.S.C. § 103 – NetBIOS in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine NetBIOS with RFC 1531 exists because NetBIOS describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary

Claims	Prior Art and Relevant Statute
	skill in the art would have been motivated to combine NetBIOS with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of NetBIOS would frequently have their IP addresses dynamically assigned.
b. in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process.	A NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NetBIOS Name Server has an entry in its database for the target name identified by the requesting node, the NBNS returns a positive name query response to the requesting node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 389. *See also id.* at 441 ("POSITIVE NAME QUERY RESPONSE").
34. The method of claim 33 further comprising the step of:	**35 U.S.C. § 102 – NetBIOS**
c. modifying the compilation of entries.	NBNSs periodically modify their compilation of name/address entries in response to various conditions. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." NetBIOS at 377. Upon going off-line, the node sends a log-out message to the NBNS, which then deletes the node's name/address entry from the compilation, thereby modifying its compilation of entries. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). The compilation of entries may also be modified when an end node does not send a refresh message to its NBNS within a determined period of time, which may result in the deletion of its name/address entry from the NBNS's compilation of entries. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "NAME CHALLENGE" operation), 380 (describing "NODE STATUS REQUEST" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."); 384 (describing "OVERWRITE" operation).

Claims	Prior Art and Relevant Statute
35. The method of claim 34 wherein step c further comprises:	**35 U.S.C. § 102 – NetBIOS**
c.1 adding an entry to the compilation upon the occurrence of a predetermined event.	NBNS add entries to their compilation of name/address entries in response to various circumstances. For example, NetBIOS discloses that registrations adds the names (distinguishing identifiers) and current IP addresses of end-nodes to the NBNS' compilation of name/address entries. *Id.* at 385. *See also* id. at 431-432; id. at 367 (The NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 461-464 (P-node name registration process) and 480-482 (NBNS incoming packet processing for name registration). Registration thereby adds names and corresponding IP addresses of end-nodes to the NBNS' compilation of name/address entries.
36. The method of claim 35 wherein the predetermined event comprises notification by a user process of an assigned network protocol address.	**35 U.S.C. § 102 – NetBIOS** User processes executed on P and M nodes "notify" the NBNS of their assigned names and IP addresses. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). More specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes.
37. The method of claim 34 wherein step c further comprises:	**35 U.S.C. § 102 – NetBIOS**

Claims	Prior Art and Relevant Statute
c.1 deleting an entry from the compilation upon the occurrence of a predetermined event.	The NBNS periodically delete a name/address entry from their compilation upon the occurrence of a predetermined event. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* at 377. Upon going off-line, the node sends a log-out message to the NBNS. Upon the occurrence of this predetermined event, the NBNS deletes the node's name/address entry from the compilation of entries. NetBIOS also discloses that a name/address entry may be deleted if an end node does not send a "refresh" message to its NBNS within a predetermined period of time. *See id.* at 448, 452-453, 464-465. *See also id.* at 448 (describing "name challenge" operation), 450 (describing "Node Status Request" operation), 451 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 453 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS.").
38. A computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium the program code comprising:	**35 U.S.C. § 102 – NetBIOS** NetBIOS discloses a computer program product for use with a computer system. *See id.* at 356 ("The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems."). In addition, NetBIOS describes that the computer systems (or "nodes") execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS also discloses a "server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates the NBNS coupled to point-to-point nodes ("P nodes") over the Internet. NetBIOS applications may be loaded into random access memory, which is a computer usable medium, and executed by a computer processor. NetBIOS describes that "[o]ne of the first implementations [of NetBIOS] was for personal computers running the PC-DOS and MS-DOS operating systems. It is possible to implement NetBIOS within other operating systems, or as processes which are, themselves, simply application programs as far as the host operating system is concerned." *Id.* at 359. NetBIOS further discloses that "typical use of NetBIOS is among independently-operated personal computers." *Id.* at 360.

Claims	Prior Art and Relevant Statute
a. program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and	NetBIOS describes the NBNS maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to the NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NetBIOS Name Server may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes. In addition, NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the node sends a "log-out" message to the NetBIOS Name Server, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or

Claims	Prior Art and Relevant Statute
	NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Therefore, only logged-in nodes are registered with the NetBIOS Name Server. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389. On many networks, including the TCP/IP networks described in NetBIOS, network addresses are assigned "upon connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer systems on which NetBIOS was used received IP addresses following connection to the computer network. Consequently, dynamic address assignment is inherent in NetBIOS. **35 U.S.C. § 103 – NetBIOS in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine NetBIOS with RFC 1531 exists because NetBIOS describes NetBIOS operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine NetBIOS with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, one of skill in the art would have understood at the time of the alleged invention of the '704 patent that personal computers connected to the Internet as described in RFC 1001/1002 of NetBIOS would frequently have their IP addresses dynamically assigned.
b. program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process.	The NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NetBIOS Name Server has an entry in its database for the target name identified by the requesting node, the NBNS returns a positive name query response to the requesting node. *Id.* at 390. The NBNS's positive name query response includes the IP address

Claims	Prior Art and Relevant Statute
	for the target node. *Id.* at 389. *See also id.* at 441 ("POSITIVE NAME QUERY RESPONSE").
39. The computer program product of claim 38 further comprising:	**35 U.S.C. § 102 – NetBIOS**
c. program code configured to modify the compilation of entries.	NBNSs periodically modify their compilation of name/address entries in response to various conditions. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* at 377. Upon going off-line, the node sends a log-out message to the NBNS, which then deletes the node's name/address entry from the compilation, thereby modifying its compilation of entries. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). The compilation of entries may also be modified when an end node does not send a refresh message to its NBNS within a determined period of time, which may result in the deletion of its name/address entry from the NBNS's compilation of entries. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "NAME CHALLENGE" operation), 380 (describing "NODE STATUS REQUEST" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."); 384 (describing "OVERWRITE" operation).
40. The computer program product of claim 39 wherein program code configured to modify comprises:	**35 U.S.C. § 102 – NetBIOS**
c.1 program code configured to add an entry to the compilation upon the occurrence of a predetermined event.	The NBNSs add entries to their compilation of name/address entries in response to various circumstances. For example, NetBIOS discloses that registration adds the names (distinguishing identifiers) and current IP addresses of end-nodes to the NBNS' compilation of name/address entries. *Id.* at 385. *See also* id. at 431-432; id. at 367 (NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 461-464 (P-node name registration process) and 480-482 (NBNS incoming packet processing for name registration). Registration thereby adds names and corresponding IP addresses of end-nodes to the NBNS' compilation of name/address entries.

Claims	Prior Art and Relevant Statute
41. The computer program product of claim 40 wherein the predetermined event comprises notification by a process of an assigned network protocol address.	**35 U.S.C. § 102 – NetBIOS** User processes executed on P and M nodes "notify" the NBNS of their assigned names and IP addresses. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). More specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NetBIOS Name Server includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 (describing how the NetBIOS Name Server may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); NetBIOS *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a list of names and corresponding IP addresses of point-to-point and mixed end-nodes.
42. The computer program product of claim 38 wherein step c further comprises:	**35 U.S.C. § 102 – NetBIOS**
c.1 program code configured to delete an entry from the compilation upon the occurrence of a predetermined event.	The NBNSs periodically delete a name/address entry from their compilation upon the occurrence of a predetermined event. For example, NetBIOS discloses an "explicit name release" mechanism. For point-to-point nodes, this involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* at 377. That is, upon going off-line, the node sends a log-out message to the NBNS. Upon the occurrence of this predetermined event, the NBNS deletes the node's name/address entry from the compilation of entries. NetBIOS also discloses that a name/address entry may be deleted if an end node does not send a "refresh" message to its NBNS within a predetermined period of time. *See id.* at 448, 452-453, 464-465. *See also id.* at 448 (describing "name challenge" operation), 450 (describing "Node Status Request" operation), 451 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 453 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS.").

Claims	Prior Art and Relevant Statute
43. A computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein, the program code comprising:	**35 U.S.C. § 102 – NetBIOS** NetBIOS discloses a computer program product for use with a computer system which executes a "first process" and is operatively connectable to a "second process" and a server over a computer network. First, NetBIOS discloses a computer program product for use with a computer system. *See id.* at 356 ("The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems."). In addition, NetBIOS describes that the computer systems (or "nodes") execute software, which is a computer-implemented "process." *See id.* ("NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available."); *id.* at 357 ("NetBIOS is the foundation of a large body of existing applications."). Finally, NetBIOS also discloses a "server" to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates the NBNS coupled to point-to-point nodes ("P nodes") over the Internet.
a. program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network; and	For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). Specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NBNS includes the field "NB_ADDRESS," which is the "IP address of the name's owner." NetBIOS *id.* at 431. *See also id.* at 367 (describing how the NetBIOS Name Server may act as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names."); *id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."); *id.* at 461-464 (disclosing program code for the P-node name registration process) and 480-482 (disclosing program code for NBNS incoming packet processing for name registration). The NBNS thereby contains a directory database of names and corresponding IP addresses of point-to-point and mixed end-nodes.
b. program code responsive to one of the network protocol addresses and configured to establish a point-to-point	Once the node seeking to initiate the communication has obtained from the NBNS the IP address for the node to receive the communication, a point-to-point communication is established

<table>
<tr><th>Claims</th><th>Prior Art and Relevant Statute</th></tr>
<tr><td>communication link from the first process to the second process over the computer network.</td><td>between the nodes. “The NetBIOS session service begins after one or more IP addresses have been found for the target name . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>.” Id. at 397 (emphasis added). See also id. at 401:

This first diagram shows the sequence of network events used to successfully establish a session without retargeting by the listener. The TCP connection is first established with the well-known NetBIOS session service TCP port, SSN_SRVC_TCP_PORT. The caller then sends a SESSION REQUEST packet over the TCP connection requesting a session with the listener. The SESSION REQUEST contains the caller's name and the listener's name. The listener responds with a POSITIVE SESSION RESPONSE informing the caller this TCP connection is accepted as the connection for the data transfer phase of the session.

See also id. at 398-400 (“16.1: Overview of NetBIOS Session Service”), 361 (“A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are fullduplex, sequenced, and reliable. Data is organized into messages.”). In sum, NetBIOS discloses all of the elements of, and hence anticipates, claim 43 of the '704 Patent.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>44. In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of:</td><td>35 U.S.C. § 102 – NetBIOS

NetBIOS describes that the various network “nodes” execute software, which is a computer-implemented “process.” See id. (“NetBIOS defines a software interface . . . . NetBIOS has generally been confined to personal computers to date. However, . . . this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available.”); id. at 357 (“NetBIOS is the foundation of a large body of existing applications.”). Finally, NetBIOS also discloses an “address server” to which all processes are operatively coupled over a network. For example, the figure on page 371 of NetBIOS illustrates the NBNS coupled to point-to-point nodes (“P nodes”) over the Internet. The NBNS is an “address server” because it stores names and IP addresses of nodes. See, e.g., id. at 367 describing how the NBNS may act as a “‘bulletin board’ on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names.”</td></tr>
</table>

Claims	Prior Art and Relevant Statute
A. following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network;	In NetBIOS, each node forwards its IP address to the NBNS following connection to the computer network. For example, to engage in NetBIOS communications, a point-to-point ("P") or mixed ("M") node must register with a NBNS by transmitting a notice of the end node's name (a distinguishing identifier) and current IP address to the NBNS. *See id.* at 385 (illustrating the "P-NODE REGISTRATION PROCESS"). More specifically, a NetBIOS "Name Registration Request" sent by an M or P node to a NetBIOS Name Server includes the field "NB_ADDRESS," which is the "IP address of the name's owner." *Id.* at 431. *See also id.* at 367 describing how the NBNS acts as a "'bulletin board' on which name/address information is freely posted (and removed) by P and M nodes without validation by the NBNS. Alternatively, the NBNS may elect to completely manage and validate names.
B. querying the address server as to whether the second process is connected to the computer network;	As disclosed in NetBIOS, an end-node sends a "query" to the NBNS to determine whether another end-node with the target name is currently logged onto the computer network, and hence is registered with the NBNS. "Name query (also known as 'resolution' or 'discovery') is the procedure by which the IP address(es) associated with a NetBIOS name are discovered." *Id.* at 377. NetBIOS point-to-point nodes "perform name resolution" by "ask[ing]" the NBNS for the IP address and other information of the target node with whom they wish to communicate. *Id.* *See also id.* at 388 ("Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name."). The NetBIOS Name Server "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. *See also id.* at 440 (RFC 1002 describing "Name Query Request"); *id.* at 464-465 (describing "P-Node Find Name Procedure"). "Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder." *Id.* at 446.
C. receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network; and	NetBIOS states that the NBNS "answers queries from a P node with a list of IP address and other information for" the target name. *Id.* at 389. If the NBNS has a record of the unique target name in its database, it returns a positive name query response to the requesting end-node. *Id.* at 390. The NBNS's positive name query response includes the IP address for the target node. *Id.* at 441 (describing a "POSITIVE NAME QUERY RESPONSE"). NetBIOS discloses a number of mechanisms to track the online status of nodes. For example, "NetBIOS names may be released explicitly or silently by an endnode. Silent release typically occurs when an end-node fails or is turned off." *Id.* at 377. For point-to-point nodes, the "explicit name release" involves "send[ing] a notification to their NBNS [NetBIOS Name Server]." *Id.* That is, upon going off-line, the

Claims	Prior Art and Relevant Statute
	node sends a "log-out" message to the NBNS, which then deletes the node's name/address entry from its database. *See also id.* at 393-394 (describing "NAME RELEASE TRANSACTIONS"). NetBIOS also discloses mechanisms designed to detect "silent" releases, *i.e.*, when a nodes goes off-line without sending an explicit log-out message to the NBNS. *Id.* at 360 ("An explicit name deletion function is specified, so that applications may remove a name. Implicit name deletion occurs when a station ceases operation."). These mechanisms include the refresh mechanism discussed above. Nodes which do not send a refresh message to their NBNS within a determined period of time are deemed to have gone off-line and their name/address entry is deleted from the NBNS. *Id.* at 378, 382-383, 394-395. *See also id.* at 378 (describing "name challenge" operation), 380 (describing "Node Status Request" operation), 381 ("15.1.7 CONSISTENCY OF THE NBNS DATA BASE"), 383 ("A very cautious NBNS is free to poll nodes (by sending NAME QUERY REQUEST or NODE STATUS REQUEST packets) to verify that their name status is the same as that registered in the NBNS."). Consequently, only logged-in nodes are registered with the NBNS. *See id.* at 446 ("Each NODE_NAME entry represents an active name in the same NetBIOS scope as the requesting name in the local name table of the responder."). In sum, the requesting node receives the target node's IP address from the NBNS only if the target node is currently logged in; otherwise, the NBNS responds to the requesting node's name query request with a negative response. *See, e.g.*, *id.* at 389.
D. in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network.	Once the node seeking to initiate the communication has obtained from the NBNS the IP address for the node to receive the communication, a point-to-point communication is established between the nodes. "The NetBIOS session service begins after one or more IP addresses have been found for the target name . . . NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They involve only <u>directed (point-to-point) communications</u>." *Id.* at 397 (emphasis added). *See also id.* at 401: This first diagram shows the sequence of network events used to successfully establish a session without retargeting by the listener. The TCP connection is first established with the well-known NetBIOS session service TCP port, SSN_SRVC_TCP_PORT. The caller then sends a SESSION REQUEST packet over the TCP connection requesting a session with the listener. The SESSION REQUEST contains the caller's name and the listener's name. The listener responds with a POSITIVE SESSION RESPONSE informing the

Claims	Prior Art and Relevant Statute
	caller this TCP connection is accepted as the connection for the data transfer phase of the session. *See also id.* at 398-400 ("16.1: Overview of NetBIOS Session Service"), 361 ("A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are fullduplex, sequenced, and reliable. Data is organized into messages.").

Exhibit N

Claim chart for Claims 1-7 and 10-44 of the '704 patent:

- **Etherphone (Claims 1-2, 4-7, 10-12, 14, 19-23, 25, 30-44) under 35 U.S.C. § 102(b)**
- **Etherphone in view of NetBIOS (Claim 3) under 35 U.S.C. § 103(a)**
- **Etherphone in view of Vin (Claim 32) under 35 U.S.C. § 103(a)**
- **Etherphone in view of Vin and further in view of RFC 1531 (Claim 33) under 35 U.S.C. § 103(a)**
- **Etherphone in view of Pinard (Claims 10-17, 19-28, 30-31) under 35 U.S.C. § 103(a)**
- **Etherphone in view of Pinard and further in view of VocalChat User's Guide(Claims 18 and 29) under 35 U.S.C. § 103(a)**

Customer No.: 8791

Blakely, Sokoloff, Taylor & Zafman, LLP
Sunnyvale, California 94085-4040
Telephone (408) 720-8300
Fax (408) 720-8383

Exhibit N
Claim Chart – Etherphone

Claims	Prior Art and Relevant Statute
1. A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network, the computer program product comprising:	**35 U.S.C. § 102 – Etherphone** Etherphone discloses a computer program product for use with a computer system which executes a "first process" and is operatively connectable to a "second process" and a server over a computer network. For example, the Etherphone system is "based on a hardware architecture that uses microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server, this system has been used for applications such as directory-based call placement, call logging, call filtering, and automatic call forwarding." Zellweger 1, page 1. *See also id.*, Figure 1 (illustrating Etherphones, computer workstations and servers communicating over an Ethernet network). The system components shown in Figure 1 of Zellweger 1 provides communication "between two or more parties (Etherphones, servers, and so on)." *Id.*, page 3. In the pending litigation, Net2Phone argued that the term "server" should be defined broadly. Plaintiff Net2Phone, Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), page 3. More specifically, Net2Phone argued: > Consistent with the use of the term 'server' in the specification, the claims do not refer to any specific server configuration. They simply require a 'server' (also referred to as a 'connection server,' 'address server,' or 'server process'). There is nothing in any of the claims that require that the server be in the form of a single computer with a centralized database, as defendants contend. *Id.*, page 4. Similarly, Net2Phone argued that "[a] server in a 'client/server system' can be implemented in any number of ways, from one to multiple computers, in one location or many, and from a single large computer acting as the server to a network of personal computers." Plaintiff Net2Phone Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007) (Exhibit W), page 7 (citing to the declaration of Professor Larry L. Peterson). Thus, under Net2Phone's interpretation, a "server" is not limited to any particular hardware or software configuration. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Oct. 19, 2007) (Exhibit X), pages 2-9. For the sake of

Claims	Prior Art and Relevant Statute
	brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a "server." Under any interpretation, the Voice Control Server described in Etherphone is a "server."
a computer usable medium having program code embodied in the medium, the program code comprising:	The functionality of the Etherphone system is implemented in software, which is inherently stored on a computer usable medium. As described in Swinehart 2, the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services." Swinehart 2, page 1. *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").
program code for transmitting to the server a network protocol address received by the first process following connection to the computer network;	Etherphone processes notify the Voice Control Server (sometimes referred to as a "Telephone Control Server") of their network addresses in order to receive calls from other Etherphone processes. As described in Swinehart 1: The telephone control server controls voice conversations, implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it <u>stores personal preference information about each user</u> that allows it to support advanced features such as ring motifs and subdued ringing without involving workstation programs. It uses <u>dynamic information linking users to workstations</u> in order to provide calls to individuals rather than fixed locations and the registration of visitors in the offices of their colleagues. Finally, and most importantly, the telephone control server provides a set of <u>network protocols that workstations use</u> to participate in the operation of the system. Swinehart 1, page 4 (emphasis added). Thus, when a user logs in to a workstation, the user's identity and the network address of the workstation are sent to the Voice Control Server to identify the user's current location. In fact, any time a computer transmits a data packet over an Ethernet network, the data packet must include the network address of the transmitting computer system (i.e., so that the receiving system knows the source of the data packet). *See id.* ("The telephone control server manages voice switching by sending to each Etherphone or service <u>the network addresses of the</u>

Claims	Prior Art and Relevant Statute
	<u>other participants</u>. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.") (emphasis added); Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does."). The network protocol address of the first process is "received" following connection to the computer network. For example, an Etherphone process (a running instance of an Etherphone application) will be assigned a network protocol address after the workstation or Etherphone on which it is running connects to the computer network – hence, following connection to the computer network. This is the case regardless of whether the workstation or Etherphone on which the given Etherphone process is running has a static network protocol address or, instead, a dynamically assigned network protocol address. The network protocol address is then transmitted to the Voice Control Server so that other Etherphone processes can locate the Etherphone process. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Given that Etherphone was implemented using TCP/IP as described in Vin, in at least some instances, the computer system on which Etherphone was used received IP addresses dynamically, after connecting to the computer network. Consequently, dynamic address assignment is inherent in the Etherphone. **35 U.S.C. § 103 – Etherphone in view of Vin and further in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on an Etherphone network. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is

Claims	Prior Art and Relevant Statute
	no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction).
program code for transmitting, to the server, a query as to whether the second process is connected to the computer network;	As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Moreover, as mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting.").
program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network; and	As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. Thus, a first Etherphone or workstation attempting to connect to a second Etherphone or workstation receives the network address of the second Etherphone or workstation from the Voice Control Server. In Claim Construction Briefs filed in the pending litigation, the patentee argued that the term > 'connected' means 'logged on,' and *vice versa* . . . To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect. While Net2Phone's invention endeavors to identify accurately who is on line, it is not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user's Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person's status— *e.g.*, setting a default value of two hours,

Claims	Prior Art and Relevant Statute
	after which the server assumes that a party has gone off-line if it has not heard from her. *See* '704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, "the on-line status information stored in the database is ***relatively*** *current*." *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term "connected" (or "on-line") is going to be modified at all, it should be modified to say "*relatively* currently connected," because that is what the patents actually say. Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), pages 24-25. Thus, under Net2Phone's interpretation, the information retained in the "server" as to which processes are "connected to the computer network" or "online" may be imperfect. As described above, while the server "endeavors to identify accurately who is on line, it is not possible to achieve perfection." *Id.* Once again, the requestor of the present reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Exhibit X), pages 12-14. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a process to be "connected to" the computer network or "on-line." Under any interpretation, a first Etherphone process receives the network protocol address of a second Etherphone process from the Voice Control Server when the second Etherphone process is "connected to the computer network."
program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network.	As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server</u>." (emphasis added). Thus, after retrieving a network address of a callee device from the Voice Control Server, a workstation or Etherphone communicates directly over a point-to-point communication link with the callee device identified by the network address. *See also* Zellweger, page 2 ("Etherphones digitize, packetize, and encrypt telephone-quality voice (64 kilobits/second, with silence suppression) and send it to each other directly over an Ethernet . . ."); Swinehart 2, page 1 ("Etherphones digitize and encrypt telephone quality audio and transmit it in packet form directly over an Ethernet.").

Claims	Prior Art and Relevant Statute
2. An apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising:	**35 U.S.C. § 102 – Etherphone** The Etherphone system includes a Voice Control Server which enables point-to-point communication between workstation and Etherphone processes. As described in Swinehart 1: The *telephone control server* controls voice conversations, implements the stand-alone behavior of telephone instruments, and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of *visitors* in the offices of their colleagues. Swinehart 1, page 4 (emphasis in original). The Voice Control Server includes a processor and a network interface for connecting the Voice Control Server to the computer network. *See, e.g.*, Zellweger 1, Figure 1 (illustrating the Voice Control Server coupled to a 1.5 Mbit/sec Ethernet network).
a processor;	The Voice Control Server inherently includes a processor for processing software.
a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network;	Given that the Voice Control Server is connected to an Ethernet network, it inherently includes a "network interface."
a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of processes, each network protocol address stored in the memory following connection of a respective process to the computer network;	Etherphone describes this limitation. In particular, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a memory. The network addresses are stored in the memory following the connection of the processes to the computer network. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of

Claims	Prior Art and Relevant Statute
	network addresses" on TCP/IP networks). Given that Etherphone was implemented using TCP/IP as described in Vin, in at least some instances, the computer system on which Etherphone was used received IP addresses dynamically, after connecting to the computer network. Consequently, dynamic address assignment is inherent in the Etherphone. **35 U.S.C. § 103 – Etherphone in view of Vin and further in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on an Etherphone network. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction).
means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process.	As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Moreover, as mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting."). Consequently, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. The query will then return the current location of the user to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users

Claims	Prior Art and Relevant Statute
	including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").
3. The computer server apparatus of claim 2 further comprising a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory.	**35 U.S.C. § 103 – Etherphone in view of NetBIOS** Claim 3 should be rejected under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of NetBIOS. Claim 3 of the '704 patent requires "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory." Etherphone does not explicitly describe a timer for time stamping network protocol address entries stored on the Voice Control Server. However, time stamping was a well known technique at the time the application which resulted in the '704 patent was filed. For example, the NetBIOS Name Server described in NetBIOS includes a timer for time-stamping name/IP address entries. As described in NetBIOS, "[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure." NetBIOS reference at 382. Similarly, as described in NetBIOS: If an end-node holds any names that have finite time-to-live values, then that node must periodically send a status report to the NBNS. Each name is reported using the NAME REFRESH REQUEST packet. These status reports restart the timers of both the NBNS and the reporting node. However, the only timers which are restarted are those associated with the name found in the status report. Timers on other names are not affected. *Id.*
4. A method for enabling point-to-point communication between a first process and a second process over a computer network, the method comprising the steps of:	**35 U.S.C. § 102 – Etherphone** Etherphone describes a method for enabling point-to-point communication between a first process and a second process over a computer network. For example, after receiving a network addresses of a first process from the Voice Control Server, a second process establishes a point-to-point communication connection with the first process. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server</u>.") (emphasis added). The "participants" all communicate with the system via software processes executed on computer workstations or

Claims	Prior Art and Relevant Statute
	Etherphones. *See, e.g.*, Swinehart 2, page 1 (describing how the capabilities of the Etherphone system "are presented to application programmers as program packages and network services."). *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").
A. receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network;	As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently receive and store the network protocol addresses in a computer memory. The network addresses are stored in the memory following the connection of the processes to the computer network. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. When a user logs in to a workstation (e.g., as a "visitor"), the user is assigned an "on-line status." On many networks, including TCP/IP networks, network addresses are assigned "following connection" to the computer network. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Given that Etherphone was implemented using TCP/IP as described in Vin, in at least some instances, the computer system on which Etherphone was used received IP addresses dynamically, after connecting to the computer network. Consequently, dynamic address assignment is inherent in the Etherphone. **35 U.S.C. § 103 – Etherphone in view of Vin and further in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on an Etherphone network. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an

Claims	Prior Art and Relevant Statute
	Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction).
B. receiving a query from the first process to determine the on-line status of the second process;	As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Moreover, as mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting."). Consequently, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. The query will then return the current location of the user to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing how a user "turns to his workstation and registers Karmen as a visitor" and also describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").
C. determining the on-line status of the second process; and	See response to the previous claim element. As described above, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. *See* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."). Swinehart 1 describes different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing how a user "turns to his

Claims	Prior Art and Relevant Statute
	workstation and registers Karmen as a visitor" and also describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").
D. transmitting an indication of the on-line status of the second process to the first process over the computer network.	The Voice Control Server will connect a first user to a second user if the second user is "online." *See, e.g.*, Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."). In addition, if the second user is offline or does not wish to receive calls, an indication is sent to the first user that the second user is unavailable. *See, e.g.*, Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").
5. The method of claim 4 wherein step C further comprises the steps of:	**35 U.S.C. § 102 – Etherphone**
c.1 searching the computer memory for an entry relating the second process; and	These features are inherent in the Etherphone system. In order for the Voice Control Server to manage "voice switching by sending to each Etherphone or service the network addresses of the other participants" (Swinehart 1, page 4) it must inherently search the server memory for the network addresses related to the other workstation and Etherphone processes.
c.2 retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process.	In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular Etherphone, the user's online status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").
6. The method of claim 4 wherein step D further comprises the steps of:	**35 U.S.C. § 102 – Etherphone**
d.1 transmitting the network protocol address of the second process to the first	In the Etherphone system, if a user is logged in and "online" then the Voice Control Server transmits the network address of the

Claims	Prior Art and Relevant Statute
process when the second process is determined in step C to have a positive on-line status with respect to the computer network.	user's process (executed on the Etherphone or workstation) to the requesting process. *See, e.g.*, Swinehart 1, page 4 ("The *telephone control server* . . . uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of *visitors* in the offices of their colleagues.") (emphasis in original).
7. The method of claim 4 wherein step D further comprises the steps of:	**35 U.S.C. § 102 – Etherphone**
d.1 generating an off-line message when the second process is determined in step C to have a negative on-line status with respect to the computer network; and	As described above, the Etherphone system includes a "do-not-disturb" option in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant." Swinehart 1, page 2.
d.2 transmitting the off-line message to the first process.	The do not disturb message was transmitted to the first process (the caller) after an attempt to contact the second process (the callee). *See* Swinehart 1, page 4.
10. In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the method comprising the steps of:	**35 U.S.C. § 102 or § 103 – Etherphone or Etherphone in view Pinard** As described in Swinehart 1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. <u>Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server</u>." (emphasis added). Swinehart 1, page 2. Thus, after retrieving a network address of a callee device from the Voice Control Server, a workstation or Etherphone communicates directly over a point-to-point communication link with the callee device identified by the network address. *See also* Zellweger, page 2 ("Etherphones digitize, packetize, and encrypt telephone-quality voice (64 kilobits/second, with silence suppression) and send it to each other directly over an Ethernet."); Swinehart 2, page 1 ("Etherphones digitize and encrypt telephone quality audio and transmit it in packet form directly over an Ethernet."). The workstations described in Etherphone include a graphical user interface (GUI). *See, e.g.*, Figures 1-10 of Swinehart 1 (illustrating various GUI features presented on the workstation display). *See also* Zellweger 1, Figures 3-4 (illustrating "telephone management windows" (Figure 3) and icons representing callers, callees and telephone lines (Figure 4)). The workstations may be Apple

Claims	Prior Art and Relevant Statute
	Macintoshes or Xerox 6085s. *See* Swinehart 1, page 1. The workstations are operatively connectable to the callee process and a server over the computer network. As previously described, "[t]he telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 2.
A. providing a user interface element representing a first communication line;	Etherphone discloses this limitation. For example, Figure 3 of Zellweger 1 depicts the Etherphone telephone management windows, including Phone and Answer buttons, a conversation log, and a portion of a personal telephone directory, which is a set of speed-dialing buttons. As described in Zellweger 1, "[a] variety of convenient workstation dialing methods are provided: a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad." Zellweger 1, page 4. In addition, Figure 4 of Zellweger 1 illustrates telephone icons representing telephone lines and and icons with graphical images of a caller/callee which represent active telephone lines. As such, the Etherphone telephone management windows provide a "user interface element representing a first communication line."
B. providing a user interface element representing a first callee process; and	Etherphone discloses user interface elements in the form of speed-dial buttons which represent frequently called callees. As described in Zellweger 1, the GUI provides "browsable lists of names and associated telephone numbers as speed-dialing buttons." Zellweger 1, page 4. *See also* Zellweger 1, Figure. 3 (depicting portion of a personal telephone directory, which is a set of speed-dial buttons). As another example, in Zellweger 1, Figure 4, the top left user interface icon represents a personal telephone directory in the form of a graphical rolodex.
C. establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line.	First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, the top row

Claims	Prior Art and Relevant Statute
	of Figure 4 of Zellweger 1 shows a series of graphical icons used for placing a call including a personal telephone directory, a telephone, and a picture of a user on the phone (to indicate a call is in process). In this example, the personal telephone directory, displayed as a graphical rolodex, includes a plurality of graphical elements representing callees (i.e., with a separate card in the rolodex for each callee). The icon of the telephone and the icon with the picture of a user talking on the phone represents a telephone communication line. As described in Zellweger 1, "[a]n active conversation is represented as a conversation between two people with a superimposed indication of the other party's name (also shown in Figure 4)." Zellweger 1, pages 4-5. Thus, when the user makes a call, the name from the graphical rolodex (PolleZ in the example) is "associated with" the graphical element representing the communication line (the image with the user talking on the phone). Alternatively, Claim 10 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard. Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, 4:38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, 5:36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
11. The method of claim 10 wherein step C further comprises the steps of:	**35 U.S.C. § 102 – Etherphone**
c.1 querying the server as to the on-line status of the first callee process; and	The Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular Etherphone, the user's online

Claims	Prior Art and Relevant Statute
	status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2.
c.2 receiving a network protocol address of the first callee process over the computer network from the server.	The Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.
12. The method of claim 10 further comprising the step of:	**35 U.S.C. § 102 – Etherphone**
D. providing an element representing a second communication line.	The Etherphone system inherently provides an element representing a second communication line. For example, using the Etherphone system, a user may receive and answer a call while already on an existing call. *See, e.g.*, Swinehart 1, page 2 (describing how users can place and receive other calls during a "background call"). Thus, multiple sets of graphical icons such as the ones shown in Figure 4 of Zellweger 1 were inherently displayed in the Etherphone system (e.g., graphical cards in a rolodex to represent callee processes, telephones to represent communication lines, and users talking on telephones to represent a callee process associated with a particular communication line). Alternatively, as set forth below, Claim 12 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard.
13. The method of claim 12 further comprising the steps of:	**35 U.S.C. § 103 – Etherphone in view Pinard**
E. terminating the point-to-point communication link from the caller process to the first callee process, in response to the user disassociating the element representing the first callee process from the element representing the first communication line; and	Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.
F. establishing a different point-to-point communication link from the caller	In Figure 6 of Pinard, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the

Claims	Prior Art and Relevant Statute
process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line.	call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, 4:22-31.
14. The method of claim 10 further comprising the steps of:	**35 U.S.C. § 102 or § 103 – Etherphone or Etherphone in view Pinard**
D. providing a user interface element representing a second callee process; and	Etherphone describes this limitation. For example, Figure 8 of Swinehart 1 illustrates four user interface elements representing four different callee processes (four different users).
E. establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line.	The four callee processes are each associated with a graphical element representing a communication line – i.e., a telephone graphic and graphical window representing a teleconference (titled "conference at 3PM re: Budget"). *See* Swinehart 1, page 3. Alternatively, Claim 14 is obvious under 35 USC § 103 in view of Pinard. In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." See Pinard, 5:31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
15. The method of claim 10 further comprising the step of:	**35 U.S.C. § 103 – Etherphone in view Pinard**
F. removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line.	In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, 6:14-15.

Claims	Prior Art and Relevant Statute
16. The method of claim 10 further comprising the steps of:	**35 U.S.C. § 103 – Etherphone in view Pinard**
D. providing a user interface element representing a communication line having a temporarily disabled status; and	Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
E. temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status.	In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
17. The method of claim 16 wherein the element provided in step D represents a communication line on hold status.	**35 U.S.C. § 103 – Etherphone in view Pinard** In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
18. The method of claim 17 wherein the element provided in step D represents a communication line on mute status.	**35 U.S.C. § 103 – Etherphone in view Pinard and further in view of VocalChat** The VocalChat User's Guide describes a "communication line on mute status" as recited in Claim 18. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.
19. The method of claim 10 wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface.	**35 U.S.C. § 102 – Etherphone** Etherphone illustrates a visual display which allows the caller to control the operation of the Etherphone. *See, e.g.*, Figure 3 of Zellweger 1 (illustrating a "workstation telephone management

Claims	Prior Art and Relevant Statute
	windows") and Figures 1-10 of Swinehart 1 (illustrating a series of windows for controlling an Etherphone).
20. The method of claim 19 wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface.	As described in Swinehart 1, to establish a call, user's can "select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad." Zellweger 1, page 4. As described above, a call from one workstation/Etherphone to another workstation/Etherphone comprises a point-to-point link
21. A computer program product for use with a computer system comprising: a computer usable medium having program code embodied in the medium for	**35 U.S.C. § 102 or § 103 – Etherphone or Etherphone in view Pinard** The functionality of the Etherphone system is implemented in software, which is inherently stored on a computer usable medium. As described in Swinehart 2, the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services." Swinehart 2, page 1. *See also* Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); *id.*, page 1 (describing how the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment.").
establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the medium further comprising:	First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process over a computer network. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the caller process (i.e., the software executed on the caller's machine) includes a user interface. *See, e.g.*, Zellweger 1, page 4, Figures 3-4 ("a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad."). Finally, Etherphone describes "being

Claims	Prior Art and Relevant Statute
	operatively connectable to the callee process and a server over the computer network." *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.") (emphasis added).
program code for generating an element representing a first communication line;	Etherphone describes a software-based graphical user interface with a graphical element representing a communication line. For example, Figure 4 of Zellweger 1 illustrates a graphical telephone icon that represents both outgoing calls (top row) and incoming calls (bottom row) over a telephone line, and graphical icons representing active calls (top and bottom right) established over the telephone line.
program code for generating an element representing a first callee process;	Etherphone describes a software-based graphical user interface with a graphical element representing a "callee process" (if the callee receives calls at an Etherphone or workstation). As described in Zellweger 1, the GUI provides "browsable lists of names and associated telephone numbers as speed-dialing buttons. Zellweger 1, page 4. *See also* Zellweger 1, Figure. 3 (depicting portion of a personal telephone directory, which is a set of speed-dial buttons). As another example, in Zellweger 1, Figure 4, the top left user interface icon represents a personal telephone directory in the form of a graphical rolodex.
program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process.	First, Etherphone describes establishing a point-to-point communication link between a caller process and a callee process. *See, e.g.*, Swinehart 1, page 2 ("voice datagrams are transmitted directly among the participants, bypassing the control server"). Second, Etherphone discloses that the point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, the top row of Figure 4 of Zellweger 1 shows a series of graphical icons used for placing a call including a personal telephone directory, a telephone, and a picture of a user on the phone (to indicate a call is in process). In this example, the personal telephone directory, displayed as a graphical rolodex, includes a plurality of graphical elements representing callees (i.e., with a separate card in the rolodex for each callee). The icon of the telephone and the icon with the picture of a user talking on the phone represents a telephone communication line. As described in Zellweger 1, "[a]n active conversation is represented as a conversation between two people with a superimposed indication of the other party's name (also shown in Figure 4)." Zellweger 1, pages 4-5. Thus, when the

Claims	Prior Art and Relevant Statute
	user makes a call, the name from the graphical rolodex (PolleZ in the example) is "associated with" the graphical element representing the communication line (the image with the user talking on the phone). Alternatively, Claim 21 is invalid under 35 U.S.C. § 103(a) as being unpatentable over Etherphone in view of Pinard. In particular Pinard. Pinard discloses that a point-to-point communication link is established in response to a user associating an element representing the first callee process with the element representing a first communication line. For example, Figure 3 of Pinard illustrates clicking and dragging an icon representing a callee from a directory 17 into a call setup icon 15. Once the callee answers the call, the call setup icon 15 becomes a call icon 23 as illustrated in Figure 4 of Pinard. *See, e.g.*, Pinard, 4:38-51 (describing how "[t]he user can then drag the icon or the name of the person to be called into the call setup icon . . . As soon as John answers the call, the application software program changes the call setup icon to a call icon designated as 23, and establishes a new call setup icon 24 spaced from the icon 23."). Similarly, Figure 6 illustrates how a point-to-point communication link may be established by clicking and dragging a callee icon 21 into an existing call icon 29. *See* Pinard, 5:36-37 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
22. The computer program product of claim 21 wherein the program code for establishing a point-to-point communication link further comprises:	**35 U.S.C. § 102 – Etherphone**
program code for querying the server as to the on-line status of the first callee process; and	As mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. Moreover, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."). In addition, the Voice Control Server maintains an "on-line status" for each process. For example, if a user is logged in to a particular

Claims	Prior Art and Relevant Statute
	Etherphone, the user's online status is "online" and associated with that Etherphone. A query (in the form of a remote procedure call) will then return the current location of the user's process to the requesting process (executed on another Etherphone or workstation). Swinehart 1 describes various different types of "on-line status" for users including "visiting" a workstation or Etherphone and "offline." *See* Swinehart 1, page 2 (describing a "do-not-disturb option" in which "internal callers were given an on-screen explanation for being turned away, while outside callers were routed to an attendant").
program code for receiving a network protocol address of the first callee process over the computer network from the server.	*See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants.)
23. A computer program product of claim 21 further comprising:	**35 U.S.C. § 102 – Etherphone**
program code for generating an element representing a second communication line.	The Etherphone system is inherently capable of providing an element representing a second communication line. For example, the Etherphone system is capable of conference calling. *See, e.g.*, Swinehart 1, page 3 (describing "negotiated conference calls"). In addition, using the Etherphone system, a user may receive and answer a call while already on an existing call. *See, e.g.*, Swinehart 1, page 2 (describing how users can place and receive other calls during a "background call"). Thus, multiple sets of graphical icons such as the ones shown in Figure 4 of Zellweger 1 were inherently displayed in the Etherphone system (e.g., graphical cards in a rolodex to represent callee processes, telephones to represent communication lines, and users talking on telephones to represent a callee process associated with a particular communication line).
24. The computer program product of claim 23 further comprising:	**35 U.S.C. § 103 – Etherphone in view Pinard**
program code, responsive to the user disassociating the element representing the first callee process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process; and	Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.

Claims	Prior Art and Relevant Statute
program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process.	In Figure 6, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, 4:22-31.
25. The computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – Etherphone in view Pinard**
program code for generating an element representing a second callee process; and	Etherphone describes this limitation. For example, Figure 8 of Swinehart 1 illustrates four user interface elements representing four different callee processes (four different users). The four callee processes are each associated with a graphical element representing a communication line – i.e., a telephone graphic and graphical window representing a teleconference (titled "conference at 3PM re: Budget"). *See* Swinehart 1, page 3.
program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process.	In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." See Pinard, 5:31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
26. The computer program product of claim 25 further comprising:	**35 U.S.C. § 103 – Etherphone in view Pinard**
program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link.	In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, 6:14-15.

Claims	Prior Art and Relevant Statute
27. The computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – Etherphone in view Pinard**
program code for generating an element representing a communication line having a temporarily disabled status; and	Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process.	In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
28. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on hold status.	**35 U.S.C. § 103 – Etherphone in view Pinard** In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
29. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on mute status.	**35 U.S.C. § 103 – Etherphone in view Pinard and further in view of VocalChat** The VocalChat User's Guide describes a "communication line on mute status" as recited in Claim 18. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.

Claims	Prior Art and Relevant Statute
30. A computer program product of claim 21 wherein the computer system further comprises a visual display and the user interface comprises a graphic user interface.	**35 U.S.C. § 102 – Etherphone** Etherphone illustrates a visual display which allows the caller to control the operation of the Etherphone. *See, e.g.*, Figure 3 of Zellweger 1 (illustrating a "workstation telephone management windows") and Figures 1-10 of Swinehart 1 (illustrating a series of windows for controlling an Etherphone).
31. The computer program product of claim 30 wherein the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises: program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process.	**35 U.S.C. § 102 – Etherphone** As described above, Etherphone illustrates a visual display which allows the caller to control the operation of the Etherphone. *See, e.g.*, Figure 3 of Zellweger 1 (illustrating a "workstation telephone management windows") and Figures 1-10 of Swinehart 1 (illustrating a series of windows for controlling an Etherphone). Figure 3 of Zellweger 1, for example, shows a "phone" button representing a first communication line and several other graphical elements representing callee processes. As described in Zellweger 1, "[a] variety of convenient workstation dialing methods are provided: a user can . . . select names or numbers from anywhere on the [Etherphone telephone management windows], use either of two directory tools that present browsable lists of names and associated telephone numbers as speed-dialing buttons, or redial any previously-made call by clicking on its conversation log entry. Calls can also be placed by name or number from the telephone keypad." Zellweger 1, page 4. In addition, the top row of Figure 4 of Zellweger 1 shows a series of graphical icons used for placing a call including a personal telephone directory, a telephone, and a picture of a user on the phone (to indicate a call is in process). In this example, the personal telephone directory, displayed as a graphical rolodex, includes a plurality of graphical elements representing callees (i.e., with a separate card in the rolodex for each callee). The icon of the telephone and the icon with the picture of a user talking on the phone represents a telephone communication line. As described in Zellweger 1, "[a]n active conversation is represented as a conversation between two people with a superimposed indication of the other party's name (also shown in Figure 4)." Zellweger 1, pages 4-5.
32. A method of locating a process over a computer network comprising the steps of:	**35 U.S.C. § 102 – Etherphone**
a. maintaining an Internet accessible list	Etherphone describes these limitations. As described in Swinehart

Claims	Prior Art and Relevant Statute
having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network; and	1: "The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4. Thus, the Telephone Control Server (also referred to as the Voice Control Server) stores a list of network addresses which are made available to workstations and Etherphones. In addition, the Voice Control Server associates different user identifiers with each network protocol address. For example, a user may log in to any workstation and, thereafter, calls to that user will be directed to that workstation and its associated Etherphone. As described in Swinehart 1: The *telephone control server* <u>controls voice conversations,</u> implements the stand-alone behavior of telephone instruments and <u>coordinates the activities of workstations and adjacent telephones</u> in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It <u>uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues</u>. Swinehart 1, page 4 (underline emphasis added). The network addresses may be Internet protocol addresses. For example, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. *See also* Terry, Abstract ("the voice manager stores voice on a special voice file server that is <u>accessible via the local internet</u>."). Moreover, other Etherphone references explicitly describe using the Internet Protocol (IP) within the Etherphone system. *See, e.g.*, Vin, page 77, Figure 5 (illustrating a "protocol stack and format" which includes internet protocol (IP) packets). Vin may be combined with Etherphone under 35 U.S.C. § 102. *See* MPEP 2131.01 (stating that a §102 rejection over multiple references is proper when the extra references are cited to explain the meaning of a term used in the primary reference). In this case, Vin is used to define the complete meaning of the term "Voice Transmission Protocol" used in Etherphone. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Given that Etherphone was implemented using TCP/IP as described in Vin, in at least some instances, the computer system on which Etherphone was used received IP addresses dynamically, after connecting to the computer

Claims	Prior Art and Relevant Statute
	network. Consequently, dynamic address assignment is inherent in the Etherphone. **35 U.S.C. § 103 – Etherphone in view of Vin and further in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on an Etherphone network. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction).
b. in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process.	As mentioned above, when a first user attempts to call a second user from a workstation and Etherphone, the Voice Control Server provides the current network protocol address of the second user to the requesting process executed on the workstation/ Etherphone of the first user. Using the network address, the requesting process then initiates a communication session with the workstation and Etherphone of the second user. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants. Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").
33. A method for locating processes having dynamically assigned network protocol addresses over a computer network, the method comprising the steps of:	**35 U.S.C. § 102 – Etherphone** As discussed above, Etherphone discloses a method of locating network protocol addresses over a computer network. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). The network protocol addresses may be Internet protocol addresses. For example, the Etherphone system was intended for use in "multiple networks and communication protocols." Terry, page 3. *See also* Terry, Abstract ("the voice manager stores voice on a special voice file server that is <u>accessible via the local internet</u>."). Moreover, other Etherphone references explicitly describe using the Internet Protocol (IP) within the Etherphone system. *See, e.g.*, Harrick M. Vin, et al.,

Claims	Prior Art and Relevant Statute
	Multimedia Conferencing in the Etherphone Environment, IEEE COMPUTER SOCIETY (Oct. 1991), page 77, Figure 5 (illustrating a "protocol stack and format" which includes internet protocol (IP) packets). Vin may be combined with Etherphone under 35 U.S.C. § 102. *See* MPEP 2131.01 (stating that a §102 rejection over multiple references is proper when the extra references are cited to explain the meaning of a term used in the primary reference). In this case, Vin is used to define the complete meaning of the term "Voice Transmission Protocol" used in Etherphone. In any case, it would have been obvious to combine Vin with Etherphone because they all describe the Etherphone system. On many networks, including TCP/IP networks, network addresses are dynamically assigned. *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Given that Etherphone was implemented using TCP/IP as described in Vin, in at least some instances, the computer system on which Etherphone was used received IP addresses dynamically, after connecting to the computer network. Consequently, dynamic address assignment is inherent in the Etherphone. **35 U.S.C. § 103 – Etherphone in view of Vin and further in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on an Etherphone network. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction)..
a. maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the	The Voice Control Server maintains "a compilation of entries . . . comprising a network protocol address and a corresponding identifier." For example, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a "computer memory." The network addresses are stored in the

Claims	Prior Art and Relevant Statute
process upon connection to the computer network; and	memory with corresponding user identifiers. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id*. Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. Since the user is connecting via a workstation and Etherphone, the user identifier identifies the current software process through which the user is interacting with the system.
b. in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process.	Claim 33 also requires "in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process." As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.
34. The method of claim 33 further comprising the step of:	**35 U.S.C. § 102 – Etherphone**
c. modifying the compilation of entries.	Etherphone discloses modifying the entries stored on the Voice Control Server as users log-in and log-out of workstations. As described in Swinehart 1: The *telephone control server* <u>controls voice conversations,</u> implements the stand-alone behavior of telephone instruments and <u>coordinates the activities of workstations and adjacent telephones</u> in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced features such as *ring motifs* and *subdued ringing* without involving workstation programs. It <u>uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues</u>. Swinehart 1, page 4 (underline emphasis added). Thus, the entries stored on the Voice Control Server are updated dynamically "in order to provide calls to individuals rather than fixed locations."
35. The method of claim 34 wherein step c further comprises:	**35 U.S.C. § 102 – Etherphone**

Claims	Prior Art and Relevant Statute
c.1 adding an entry to the compilation upon the occurrence of a predetermined event.	Etherphone inherently discloses this limitation. The predetermined event may include, for example, adding a new workstation and Etherphone to the network, powering on an existing Etherphone or workstation, adding a new Voice Control Server, and/or logging in a user to the system from a new workstation/Etherphone. Each of these events may require adding an entry to the Voice Control Server. *See, e.g.*, Swinehart 1, page 4 (describing how the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). In order to "manage" voice switching by sending network addresses of logged in participants, the Voice Control Server must be capable of adding entries to its database in response to "predetermined events."
36. The method of claim 35 wherein the predetermined event comprises notification by a user process of an assigned network protocol address.	Etherphone discloses this limitation. For example, when a user logs in to a workstation/Etherphone, the identity of the user and the network address of the workstation/Etherphone is transmitted to the voice control server so that the user can be located by other users. As mentioned above, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of visitors in the offices of their colleagues." Swinehart 1, page 4.
37. The method of claim 34 wherein step c further comprises:	**35 U.S.C. § 102 – Etherphone**
c.1 deleting an entry from the compilation upon the occurrence of a predetermined event.	Etherphone inherently describes this limitation. As described in Zellweger: > If an Etherphone user logs in at a workstation, his calls can be automatically forwarded to the adjacent Etherphone. An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does. <u>Each visit request cancels any earlier requests</u>. The common problem of forgetting to cancel forwarding is eased by ringing both Etherphones during visiting. Zellweger, page 5 (emphasis added). *See also* Swinehart, page 2 (describing how after Karmen leaves Lee's office "an additional call to Karmen . . . reminds Lee to terminate the visiting arrangement."). In these examples, "terminating" or "cancelling" the user's visiting status inherently requires deleting the association

Claims	Prior Art and Relevant Statute
	between the visiting user and the network address of the workstation/Etherphone.
38. A computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium the program code comprising:	**35 U.S.C. § 102 – Etherphone** The Etherphone system is implemented with software executed on a plurality of computing devices, including servers, workstations, and Etherphones. As described in Zellweger 1, the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ." Zellweger 1, page 1. *See also* Swinehart 2, page 1 (describing how the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services."); Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment."). The computer systems are operatively connectable over a computer network to computer processes. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Remote procedure calls are inherently directed to "computer processes."
a. program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and	The Voice Control Server maintains "a compilation of entries . . . comprising a network protocol address and a corresponding identifier or a process connected to the computer network." For example, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. In order to send the network addresses of the other participants, the Voice Control Server must inherently store the network protocol addresses in a "computer memory." The network addresses are stored in the memory with corresponding user identifiers. For example, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues." *Id.* Thus, if a user logs in to any workstation, the identity of that user and the associated network address must be stored in a memory of the Voice Control Server so that other users can locate the user. Since the user is connecting via a workstation and Etherphone, the user identifier identifies the current software "process" through which the user is interacting with the system.

Claims	Prior Art and Relevant Statute
	On many networks, including TCP/IP networks, network addresses are assigned "upon connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Given that Etherphone was implemented using TCP/IP as described in Vin, in at least some instances, the computer system on which Etherphone was used received IP addresses dynamically, after connecting to the computer network. Consequently, dynamic address assignment is inherent in the Etherphone. **35 U.S.C. § 103 – Etherphone in view of Vin and further in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on an Etherphone network. As mentioned above, a motivation to combine Etherphone and Vin exists because they both describe the same Etherphone system. In addition, a motivation to combine these references with RFC 1531 exists due to the problem to be solved. In particular, Vin describes the use of IP addresses within an Etherphone system and RFC 1531 describes techniques for dynamically assigning IP addresses. One of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers (e.g., workstations and Etherphones) and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction).
b. program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process.	As described above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.
39. The computer program product of claim 38 further comprising:	**35 U.S.C. § 102 – Etherphone**
c. program code configured to modify the compilation of entries.	Etherphone discloses modifying the entries stored on the Voice Control Server as users log-in and log-out of workstations. As described in Swinehart 1: The *telephone control server* <u>controls voice conversations,</u> implements the stand-alone behavior of telephone instruments and coordinates the activities of workstations and adjacent telephones in their implementation of the various voice capabilities. In addition, it stores personal preference information about each user that allows it to support advanced

Claims	Prior Art and Relevant Statute
	features such as *ring motifs* and *subdued ringing* without involving workstation programs. It <u>uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of *visitors* in the offices of their colleagues</u>. Swinehart 1, page 4 (underline emphasis added). *See also* Swinehart, page 2 (describing how after Karmen leaves Lee's office "an additional call to Karmen . . . reminds Lee to terminate the visiting arrangement."). Thus, the entries stored on the Voice Control Server are updated dynamically "in order to provide calls to individuals rather than fixed locations" and to "terminate" old, outdated entries.
40. The computer program product of claim 39 wherein program code configured to modify comprises:	**35 U.S.C. § 102 – Etherphone**
c.1 program code configured to add an entry to the compilation upon the occurrence of a predetermined event.	Etherphone inherently discloses this limitation. The predetermined event may include, for example, adding a new workstation and Etherphone to the network, powering on an existing Etherphone or workstation, adding a new Voice Control Server, and/or logging in a user to the system from a new workstation/Etherphone. Each of these events may require adding an entry to the Voice Control Server. *See, e.g.,* Swinehart 1, page 4 (describing how the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). In order to "manage" voice switching by sending network addresses of logged in participants, the Voice Control Server must be capable of adding entries to its database in response to "predetermined events."
41. The computer program product of claim 40 wherein the predetermined event comprises notification by a process of an assigned network protocol address.	Etherphone inherently discloses this limitation. For example, when a user logs in to a workstation/Etherphone, the identity of the user and the network address of the workstation/Etherphone is transmitted to the voice control server so that the user can be located by other users. As mentioned above, the Voice Control Server "uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations and the registration of visitors in the offices of their colleagues." Swinehart 1, page 4.
42. The computer program product of claim 38 wherein step c further comprises:	**35 U.S.C. § 102 – Etherphone**

Claims	Prior Art and Relevant Statute
c.1 program code configured to delete an entry from the compilation upon the occurrence of a predetermined event.	Etherphone inherently describes this limitation. As described in Zellweger: If an Etherphone user logs in at a workstation, his calls can be automatically forwarded to the adjacent Etherphone. An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting. Registering with the destination location allows users to travel more freely than forwarding calls from the home location does. <u>Each visit request cancels any earlier requests</u>. The common problem of forgetting to cancel forwarding is eased by ringing both Etherphones during visiting. Zellweger, page 5 (emphasis added). *See also* Swinehart, page 2 (describing how after Karmen leaves Lee's office "an additional call to Karmen . . . reminds Lee to terminate the visiting arrangement."). In these examples, "terminating" or "cancelling" the user's visiting status inherently requires deleting the association between the visiting user and the network address of the workstation/Etherphone.
43. A computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein, the program code comprising:	**35 U.S.C. § 102 – Etherphone** The Etherphone system is implemented with software executed on a plurality of computing devices, including servers, workstations, and Etherphones. As described in Zellweger 1, the Etherphone system uses "microprocessor-controlled telephones to transmit voice over an Ethernet that also supports a voice file server and a voice synthesis server . . ." Zellweger 1, page 1. *See also* Swinehart 2, page 1 (describing how the capabilities provided by the Etherphone system "are presented to application programmers as program packages and network services."); Zellweger 1, page 2 ("Etherphone software is written in C"); *id.* ("Centralized server software limited the necessary size and speed of the Etherphone processor, and thus its cost . . ."); Terry, page 4 ("The server software and the initial workstation software was developed in the Cedar programming environment."). The computer systems are operatively connectable over a computer network to computer processes and server processes. As described in Zellweger 1, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Remote procedure calls are inherently directed to "computer processes."
a. program code configured to access a directory database, the database having a network protocol address for a selected	Etherphone describes a directory database for storing network addresses of on-line processes. For example, the Voice Control Server (also referred to as a "Telephone Control Server") stores

Claims	Prior Art and Relevant Statute
plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network; and	network addresses for processes executed on each workstation and Etherphone. As described in Zellweger 1: "Users can place calls by specifying a name, a number, or other attributes of the called party. A system directory database for local Xerox employees (about 1000 entries) is stored on the Voice Control Server." Zellweger 1, page 4 (emphasis added). *See also* Swinehart 1, page 4 ("The Telephone Control Server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). The network addresses are sent to the Voice Control Server following connection to the computer network. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server . . . uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues."). Thus, when a user logs in to a workstation, the network protocol address of the workstation and the identity of the user are sent to the Voice Control Server.
b. program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network.	After receiving the network protocol address from the Voice Control Server, the caller's workstation and Etherphone will establish a point-to-point connection with the callee's workstation and Etherphone. See, e.g., Swinehart 1, page 4 (describing how after receiving a network address from the Voice Control Server, "voice datagrams are transmitted directly among the participants, bypassing the control server.").
44. In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of:	**35 U.S.C. § 102 – Etherphone** Etherphone describes a first computer process operatively coupled to a second process and an address server and further describe a method in the first process for establishing point-to-point communication with the second process. For example, in the Etherphone system, a first process executed on a first Etherphone or workstation contacts the Voice Control Server to learn the address of a second process executed on a second Etherphone or workstation. *See e.g.,* Swinehart 1, page 4 ("The Telephone Control Server manages voice switching by sending to each Etherphone or service the network addresses of the other participants."). The first process then uses the network address to establish point-to-point communication with the second process. *See, e.g., id.* ("Thereafter, voice datagrams are transmitted directly among the participants, bypassing the control server.").
A. following connection of the first process to the computer network	In the Etherphone system, when a user logs in to a particular workstation, the user's identity and the network protocol address of

Claims	Prior Art and Relevant Statute
forwarding to the address server a network protocol address at which the first process is connected to the computer network;	the workstation is forwarded to the Voice Control Server. *See, e.g.*, Swinehart 1, page 4 ("The telephone control server . . . uses dynamic information linking users to workstations in order to provide calls to individuals rather than fixed locations, and the registration of visitors in the offices of their colleagues.").
B. querying the address server as to whether the second process is connected to the computer network;	As mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4. Moreover, "conversations are established between two or more parties (Etherphones, servers, and so on) by performing remote procedure calls to the Voice Control Server." Zellweger 1, page 3. Thus, when a first user at a first Etherphone (a first "process") calls a second user at a second Etherphone (a second "process"), the first Etherphone transmits a query in the form of a remote procedure call to determine the location of the second Etherphone. *See also* Swinehart 1, page 2 ("Calls are to individuals, not locations . . . Logging in tells the telephone system where Karmen is."); Zellweger 1, page 5 ("An additional feature, called visiting, allows him to register his presence with a second workstation or Etherphone, such as during a meeting.").
C. receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network; and	As mentioned above, the Voice Control Server "manages voice switching by sending to each Etherphone or service the network addresses of the other participants." Swinehart 1, page 4.
D. in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network.	As described in Swinehart 1, after receiving a network address from the Voice Control Server, "voice datagrams are transmitted directly among the participants, bypassing the control server." Swinehart 1, page 4.

Exhibit O

Claim chart for Claims 1-7 and 10-44 of the '704 patent:

- **VocalChat User's Guide in view of VocalChat Readme, and further in view of VocalChat Networking, and further in view of VocalChat Help File, and further in view of VocalChat Troubleshooting Help File (collectively "VocalChat" or "VocalChat references") (Claims 1-2, 4, 7, 10-11, 19-22, 30-42) under 35 U.S.C. § 103(a)**
- **VocalChat References in view of RFC 1531 (Claims 1-2, 4, 7, 10-11, 19-22, 30-42) under 35 U.S.C. § 103(a)**
- **VocalChat References in view of NetBIOS (Claim 3) under 35 U.S.C. § 103(a)**
- **VocalChat References in view of Pinard (Claims 12-18 and 23-29) under 35 U.S.C. § 103(a)**

Blakely, Sokoloff, Taylor & Zafman, LLP
Sunnyvale, California 94085-4040
Telephone (408) 720-8300
Fax (408) 720-8383

Exhibit O
Claim Chart – VocalChat

Claims	Prior Art and Relevant Statute
1. A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network, the computer program product comprising:	**35 U.S.C. § 103 – VocalChat** When TCP/IP is used, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, as described above, with TCP/IP, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that VocalChat clients "received" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*; *see also id.*, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses[.]"). In the pending litigation, Net2Phone argued that the term "server" should be defined broadly. Plaintiff Net2Phone, Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), page 3. More specifically, Net2Phone argued: > Consistent with the use of the term 'server' in the specification, the claims do not refer to any specific server configuration. They simply require a 'server' (also referred to as a 'connection server,' 'address server,' or 'server process'). There is nothing in any of the claims that require that the server be in the form of a single computer with a centralized database, as defendants contend. *Id.*, page 4. Similarly, Net2Phone argued that "[a] server in a 'client/server system' can be implemented in any number of ways, from one to multiple computers, in one location or many, and from a single large computer acting as the server to a network of personal computers." Plaintiff Net2Phone Inc.'s Reply Brief on Claim Construction (Oct. 19, 2007) (Exhibit W), page 7 (citing to the declaration of Professor Larry L. Peterson). Thus, under Net2Phone's interpretation, a "server" is not limited to any particular hardware or software configuration. It should be noted, however, that the requestor of the present Reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Oct. 19, 2007) (Exhibit X), pages 2-9. For the sake of brevity, these interpretations are not repeated below with respect to the other claims of the '704 patent which require a "server." Under any interpretation, the "post office" server in VocalChat is a

Claims	Prior Art and Relevant Statute
	"server."
a computer usable medium having program code embodied in the medium, the program code comprising:	As software, VocalChat is inherently stored as program code on a computer-usable medium. See, e.g., Readme, page 1 (listing the VocalChat files copied during installation). See also VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A").
program code for transmitting to the server a network protocol address received by the first process following connection to the computer network;	As illustrated in the figure on page 5 of the VocalChat User's Guide (reproduced above), computers with VocalChat installed connect directly to a server to register their current network protocol addresses. In the initial VocalChat implementations (versions 1.x) each VocalChat client transmitted its name and network protocol address to a USERS file stored on the server. As described in VocalChat Network Information: When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*. VocalChat Network Information, page 10 (underline emphasis added). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file which was stored within a 'Post-Office' directory. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. Regardless of the file name, the Connection List/USERS file was stored on a server so that it could be accessed by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory.").

Claims	Prior Art and Relevant Statute
	On many networks, including TCP/IP networks, network addresses were assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, the computer system on which VocalChat was executed received its IP address dynamically, after connecting to the computer network. Consequently, dynamic address assignment was inherent in the VocalChat system. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
program code for transmitting, to the server, a query as to whether the second process is connected to the computer network;	VocalChat employed different techniques for locating users based on the underlying network protocol. As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. GenericUser VocalChats files for users information. (See Generic network, below). Help File, page 26. Thus, when any protocol other than Netware or Windows for Workgroups was used (such as TCP/IP or NetBIOS), a "generic" method was used in which the VocalChat client queried VocalChat files (the Connection List/USERS files) to locate users on the network. As described in greater detail in the Help File: When **NetWare** is used, VocalChat uses the

Claims	Prior Art and Relevant Statute
	NetWare Bindery services to get the list of servers, known users and groups on each server, currently logged-in users, and the addresses of specific users. When **Windows for Workgroups** is used, VocalChat uses the Window for Workgroups users services to get the list of workgroups and computers (there is no need for user addresses here). When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. *See also* Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Thus, when NetWare was used, the VocalChat client queried existing NetWare Bindery services to locate "currently logged-in users;" when Windows for Workgroups was used, the VocalChat client queried the Windows for Workgroups services to locate online users; and when other protocols, such as TCP/IP and NetBIOS were used, the VocalChat client queried the shared Connection List file (CONNLIST.VC). Regardless of which protocol was used, the query determined whether the second process (the VocalChat clients of other users) were is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction was made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieved a list of "currently logged in users."
program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network; and	When TCP/IP is used, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, as described above, when TCP/IP is used, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that VocalChat clients "received"

Claims	Prior Art and Relevant Statute
	the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.* In Claim Construction Briefs filed in the pending litigation, the patentee argued that the term > 'connected' means 'logged on,' and *vice versa* . . . To the extent defendants are trying to suggest that the claims require perfect information about who is on line at a given moment, that is simply incorrect. While Net2Phone's invention endeavors to identify accurately who is on line, it is not possible to achieve perfection. For example, it takes some time (albeit minimal) for the signal that a user has gone off-line to be communicated to the server, or a user's Internet connection may get interrupted before she can send an off-line message (and thus the server, for a time, assumes she is on-line, when in fact she is not). *See* Strickland Dep. at 140:7-141:7 (Ex. 21). Recognizing these issues, the patents explain that the server may use timestamps to update a person's status— *e.g.*, setting a default value of two hours, after which the server assumes that a party has gone off-line if it has not heard from her. *See* '704 patent, col. 5, ll. 39-44 (Ex. 2). In this respect, the patents explain, "the on-line status information stored in the database is ***relatively*** *current*." *Id.* at col. 5, ll. 42-43 (emphasis added). While Net2Phone believes that the claim language is clear, if the term "connected" (or "on-line") is going to be modified at all, it should be modified to say "*relatively* currently connected," because that is what the patents actually say. Plaintiff Net2Phone Inc.'s Response Brief on Claim Construction (Oct. 18, 2007) (Exhibit U), pages 24-25. Thus, under Net2Phone's interpretation, the information retained in the "server" as to which processes are "connected to the computer network" or "online" may be imperfect. As described above, while the server "endeavors to identify accurately who is on line, it is not possible to achieve perfection." *Id.* As described above, VocalChat employs similar techniques as address entries for off-line VocalChat processes are removed through the use of log-out messages. Once again, the requestor of the present reexamination does not agree with this interpretation, and has stated as such in the pending litigation. *See, e.g.*, Reply Claim Construction Brief of Skype Technologies SA, Skype, Inc. and EBay Inc (Exhibit X), pages 12-14. For the sake of brevity, these interpretations are not repeated

Claims	Prior Art and Relevant Statute
	below with respect to the other claims of the '704 patent which require a process to be "connected to" the computer network or "on-line." Under any interpretation, a first VocalChat process receives the network protocol address of a second VocalChat process from the "post office" server when the second VocalChat process is "connected to the computer network."
program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network.	VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").
2. An apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising:	**35 U.S.C. § 103 – VocalChat** VocalChat discloses the preamble of claim 2. For example, VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols."). Inherently, the personal computers and the server included a processor, for processing program code, and a network interface connected to the network. The VocalChat software installed on each computer system comprises a computer-implemented "process."
a processor;	Logically, the personal computers and the server included a processor for processing program code and a network interface for connecting to the network.
a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network;	Logically, the personal computers and the server included a processor for processing program code and a network interface for connecting to the network.
a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of	Computer systems with VocalChat installed and the central server inherently included "a memory, operatively coupled to the processor" in the form of a RAM and a hard drive. *See, e.g.*, User

Claims	Prior Art and Relevant Statute
processes, each network protocol address stored in the memory following connection of a respective process to the computer network;	Guide, page 7 (describing minimum system requirements of 4 MB RAM). In the initial VocalChat implementations (versions 1.x) each VocalChat client transmits its name and network protocol address to a USERS file stored on a server. As described in VocalChat Network Information: > When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. > Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*. VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own comptuers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8("the Setup program crates a Connection List File which is used to identify and access users"). On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP

Claims	Prior Art and Relevant Statute
	address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network in combination with VocalChat. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process.	In a TCP/IP implementation, the server on which the Connection List/USERS file is located transmits the network protocol address of a second VocalChat client (second process) to a first VocalChat client (first process) upon request. As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. Generic User VocalChats files for users information. (See Generic network, below). Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File: When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold

Claims	Prior Art and Relevant Statute
	user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."
3. The computer server apparatus of claim 2 further comprising a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory.	**35 U.S.C. § 103 – VocalChat in view of NetBIOS** Claim 3 of the '704 patent requires "a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory." The VocalChat references do not describe a timer for time stamping network protocol address. However, time stamping was a well known technique at the time the application which resulted in the '704 patent was filed. For example, the NBNS described in NetBIOS includes a timer for time-stamping name/IP address entries. As described in NetBIOS, "[t]he NBNS may impose a 'time-to-live' on each name it registers. The registering node is made aware of this time value during the name registration procedure." NetBIOS at 382. Similarly, as described in NetBIOS: If an end-node holds any names that have finite time-to-live values, then that node must periodically send a status report to the NBNS. Each name is reported using the NAME REFRESH REQUEST packet. These status reports restart the timers of both the NBNS and the reporting node. However, the only timers which are restarted are those associated

Claims	Prior Art and Relevant Statute
	with the name found in the status report. Timers on other names are not affected. *Id.* A motivation to combine the VocalChat references with NetBIOS explicitly exists within the VocalChat references. For example, NetBIOS is one of the network protocols explicitly supported by VocalChat. See, e.g., Help File, page 2 ("When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, <u>a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses</u>." Thus, one of ordinary skill in the art would have been motivated to combine VocalChat with NetBIOS, because the VocalChat references explicitly state that NetBIOS may be used as the underlying network protocol.
4. A method for enabling point-to-point communication between a first process and a second process over a computer network, the method comprising the steps of:	**35 U.S.C. § 103 – VocalChat** As mentioned above, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").
A. receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network;	As described in VocalChat Network Information: When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a <u>shared USERS file with the names of logged in users</u>. <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*.[1] VocalChat Network Information, page 10 (underline emphasis

[1] VocalChat Network Information, page 10 (underline emphasis added).

Claims	Prior Art and Relevant Statute
	added); *see also* Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own comptuers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users"). On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network in combination with VocalChat. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an

Claims	Prior Art and Relevant Statute
	address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
B. receiving a query from the first process to determine the on-line status of the second process;	In a TCP/IP implementation, the server on which the Connection List/USERS file was located received queries from VocalChat clients (first processes) to determine the on-line status of other VocalChat clients (second processes). As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. Generic User VocalChats files for users information. (See Generic network, below). Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File: When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting

Claims	Prior Art and Relevant Statute
	VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."
C. determining the on-line status of the second process; and	As described above, with TCP/IP, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, with TCP/IP, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that the server "transmits" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*; *see also id.*, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses[.]").
D. transmitting an indication of the on-line status of the second process to the first process over the computer network.	As described above, when TCP/IP is used, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, when TCP/IP is used, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that the server "transmitted" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*
5. The method of claim 4 wherein step C further comprises the steps of:	**35 U.S.C. § 103 – VocalChat**
c.1 searching the computer memory for an entry relating the second process; and c.2 retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process.	The VocalChat references inherently describe "searching the computer memory for an entry relating the second process." For example, as described above, VocalChat used a server to store names and network addresses of on-line users and to provide those network address to VocalChat clients upon request. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server was inherenly capable of "searching the computer memory for an entry relating to the second process (i.e., a name and address of a VocalChat client).
6. The method of claim 4 wherein step D	**35 U.S.C. § 103 – VocalChat**

Claims	Prior Art and Relevant Statute
further comprises the steps of:	
d.1 transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network.	In a TCP/IP implementation, the server on which the Connection List/USERS file was located received queries from VocalChat clients (first processes) to determine the on-line status of other VocalChat clients (second processes). As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. Generic User VocalChats files for users information. (See Generic network, below). Help File, page 26. Thus, when any protocol other than Netware or Windows for Workgroups was used (such as TCP/IP or NetBIOS), a "generic" method was used in which the VocalChat client queried VocalChat files (the Connection List/USERS files) to locate users on the network. As described in greater detail in the Help File: When **NetWare** is used, VocalChat uses the NetWare Bindery services to get the list of servers, known users and groups on each server, currently logged-in users, and the addresses of specific users. When **Windows for Workgroups** is used, VocalChat uses the Window for Workgroups users services to get the list of workgroups and computers (there is no need for user addresses here). When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. Thus, when NetWare was used, the VocalChat client queried existing NetWare Bindery services to locate "currently logged-in users;" when Windows for Workgroups was used, the VocalChat client queried the Windows for Workgroups services to locate online users; and when other protocols, such as TCP/IP and NetBIOS were used, the VocalChat client queried the shared Connection List file (CONNLIST.VC). Regardless of which

Claims	Prior Art and Relevant Statute
	protocol was used, the query determined whether the second process (the VocalChat clients of other users) was connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction was made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieved a list of "currently logged in users."
7. The method of claim 4 wherein step D further comprises the steps of:	**35 U.S.C. § 103 – VocalChat**
d.1 generating an off-line message when the second process 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 process.	First, as described above, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction was made between on-line and off-line users. Similarly, as described above, in the NetWare implementation, the query retrieved a list of "currently logged in users." Moreover, various types of off-line messages were provided to indicate the unavailability of VocalChat users. *See, e.g.*, Troubleshooting Help File, page 2 (describing that when a user's name in the "New Users" dialog box, one of the causes may be that "[t]he 'Show only Logged-in' check-box is checked, and the person is not currently logged-in."). Consequently, the server inherently transmitted "off-line messages" to the VocalChat clients to distinguish between online and offline users.
10. In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the method comprising the steps of:	**35 U.S.C. § 103 VocalChat** As discussed above, VocalChat clients connect to a server to locate and establish point-to-point connections with other VocalChat clients over a network. For example, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file" which is stored on a server. Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with

Claims	Prior Art and Relevant Statute
	other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.”).
A. providing a user interface element representing a first communication line;	A VocalChat user makes a point-to-point call to another user by using the VocalChat “Call” button, which is a user interface element representing a first communication line. *See, e.g.*, User Guide, page 14 (“Select Call from the Chat menu, or click on the tool bar Call button”). In addition, the VocalChat graphical user interface (GUI) includes a plurality of Quick Dial buttons. *See* User Guide, page 12. Depending on the implementation, either the Call button or the Quick Dial button comprises an “element representing a first communication line.”
B. providing a user interface element representing a first callee process; and	The VocalChat GUI displayed the names of potential callees in a dialog box. *See, e.g.*, Help File, page 14 (“just select a user from the user list, and choose “OK”). Callees are also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 (“Setting a Quick Dial Button”). Depending on the implementation, either the callee names listed within the dialog box or the Quick Dial buttons comprise “a user interface element representing a first callee process.”
C. establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line.	As mentioned above, a VocalChat user makes a point-to-point call to another user with the Call button or a Quick Dial Button representing a frequently called callee. *See* Help File, page 14 (describing use of the Call button) and 20 (describing use of the Quick Dial buttons). Selecting the Call button opens a dialog box displaying a list of connected VocalChat users. A caller then clicks on a user’s name in the list and then clicks the OK button to establish a point-to-point communication link. *See*, *e.g.*, Help File, page 14. In this example, the graphical representation of the user in the list is an “element representing the first callee process” and the OK button is an “element representing a first communication line.” Alternatively, a user can associate any VocalChat user with a Quick Dial button by right-clicking on a Quick Dial button, which presents the user with the VocalChat users list. *See* Help File, page 20. After the user selects a user name from the list, that user is associated with the quick dial button. *See* Help File, page 21 (“From the user list, choose the user name that you want the button to hold.”). The caller then places a call to the callee by selecting the Quick Dial button. VocalChat also assigns Quick Dial buttons automatically (“When you call a user with the Call command, a vacant button changes to hold the user’s name if one does not hold

Claims	Prior Art and Relevant Statute
	it already.”). In these examples, the graphical representation of the user in the list is an “element representing the first callee process” and the quick dial button is an “element representing a first communication line.” In both cases, the element representing the callee process is associated with an element representing a communication line.
11. The method of claim 10 wherein step C further comprises the steps of:	**35 U.S.C. § 103 – the VocalChat References**
c.1 querying the server as to the on-line status of the first callee process; and c.2 receiving a network protocol address of the first callee process over the computer network from the server.	As described above, when NetWare was used, the VocalChat client queried existing NetWare Bindery services to locate “currently logged-in users;” when Windows for Workgroups was used, the VocalChat client queried the Windows for Workgroups services to locate online users; and when other protocols, such as TCP/IP and NetBIOS were used, the VocalChat client queried the shared Connection List file (CONNLIST.VC). *See, e.g.*, Help File, page 2. Regardless of which protocol was used, the query determined the online status of the callee process (the VocalChat client of a callee). For example, “[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file.” VocalChat Network Information, page 10 (emphasis added). Thus, a distinction was made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieved a list of “currently logged in users.”
12. The method of claim 10 further comprising the step of:	**35 U.S.C. § 103 – VocalChat in view Pinard**
D. providing an element representing a second communication line.	The graphical user interface described in Pinard provides an element representing a second communication line. For example, call icons 23 and 29 representing two communication lines are shown in Figure 6 of Pinard. *See* Pinard, 5:31-40, Figure 6 (“Now there are clearly two calls in progress . . .”).
13. The method of claim 12 further comprising the steps of:	**35 U.S.C. § 103 – VocalChat in view Pinard**

Claims	Prior Art and Relevant Statute
E. terminating the point-to-point communication link from the caller process to the first callee process, in response to the user disassociating the element representing the first callee process from the element representing the first communication line; and	Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.
F. establishing a different point-to-point communication link from the caller process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line.	In Figure 6 of Pinard, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, 4:22-31.
14. The method of claim 10 further comprising the steps of:	**35 U.S.C. § 103 – VocalChat in view Pinard**
D. providing a user interface element representing a second callee process; and	In Figure 6 of Pinard, the user interface element for "John" 21 is a "second callee process." *See* Pinard, 5:31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
E. establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line.	In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." *See* Pinard, 5:31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.").
15. The method of claim 10 further comprising the step of:	**35 U.S.C. § 103 – VocalChat in view Pinard**
F. removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line.	In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, 6:14-15.

Claims	Prior Art and Relevant Statute
16. The method of claim 10 further comprising the steps of:	**35 U.S.C. § 103 – VocalChat in view Pinard**
D. providing a user interface element representing a communication line having a temporarily disabled status; and	Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
E. temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status.	In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
17. The method of claim 16 wherein the element provided in step D represents a communication line on hold status.	**35 U.S.C. § 103 – VocalChat in view Pinard** In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
18. The method of claim 17 wherein the element provided in step D represents a communication line on mute status.	**35 U.S.C. § 103 – VocalChat in view of Pinard** The VocalChat User's Guide describes a "communication line on mute status." As described in the User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.
19. The method of claim 10 wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface.	**35 U.S.C. § 103 – VocalChat** The VocalChat client included a graphical user interface capable of being displayed on a "visual display." *See, e.g.*, User Guide, page 11 (illustrating the primary VocalChat GUI including a Call button, a volume slider and a plurality of Quick Dial buttons).

Claims	Prior Art and Relevant Statute
20. The method of claim 19 wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface.	**35 U.S.C. § 103 – VocalChat** A VocalChat user could make a point-to-point call to another user by using the Call button or a Quick Dial Button representing a frequently called callee. *See* User Guide, page 14 ("just select a user from the user list, and choose "OK"). Callees were also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 ("Setting a Quick Dial Button").
21. A computer program product for use with a computer system comprising:	**35 U.S.C. § 103 – VocalChat**
a computer usable medium having program code embodied in the medium for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the medium further comprising:	The techniques described in VocalChat are implemented in software, which is a "computer program product." In particular, VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols.").
program code for generating an element representing a first communication line;	As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A"). As discussed above, VocalChat clients connect to a server to locate and establish point-to-point connections with other VocalChat clients over a network. For example, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file" which is stored on a server. Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

Claims	Prior Art and Relevant Statute
program code for generating an element representing a first callee process;	A VocalChat user makes a point-to-point call to another user by using the VocalChat "Call" button, which is a user interface element representing a first communication line. *See, e.g.*, User Guide, page 14 ("Select Call from the Chat menu, or click on the tool bar Call button"). In addition, the VocalChat graphical user interface (GUI) included a plurality of Quick Dial buttons. *See* User Guide, page 12. Depending on the implementation, either the Call button or the Quick Dial button comprises an "element representing a first communication line."
program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process.	The VocalChat GUI displayed the names of potential callees in a dialog box. *See, e.g.*, Help File, page 14 ("just select a user from the user list, and choose "OK"). Callees were also represented as Quick Dial buttons. *See* Help File, pages 11, 20-21 ("Setting a Quick Dial Button"). Depending on the implementation, either the callee names listed within the dialog box or the Quick Dial buttons comprise "a user interface element representing a first callee process." As mentioned above, a VocalChat user makes a point-to-point call to another user with the Call button or a Quick Dial Button representing a frequently called callee. *See* Help File, page 14 (describing use of the Call button) and 20 (describing use of the Quick Dial buttons). Selecting the Call button opens a dialog box which displays a list of connected VocalChat users. A caller then clicks on a user's name in the list and then clicks the OK button to establish a point-to-point communication link. *See*, *e.g.,* Help File, page 14. In this example, the graphical representation of the user in the list is an "element representing the first callee process" and the OK button is an "element representing a first communication line." Alternatively, a user can associate any VocalChat user with a Quick Dial button by right-clicking on a Quick Dial button, which presents the user with the VocalChat users list. *See* Help File, page 20. Once the user selects a user name from the list, that user was associated with the quick dial button. *See* Help File, page 21 ("From the user list, choose the user name that you want the button to hold."). The caller may then place a call to the callee by selecting the Quick Dial button. VocalChat also assigns Quick Dial buttons automatically ("When you call a user with the Call command, a vacant button changes to hold the user's name if one does not hold it already."). In these examples, the graphical representation of the user in the list is an "element representing the first callee process" and the quick dial button is an "element representing a first communication line." In both cases, the element representing the callee process is associated with an element representing a communication line.

Claims	Prior Art and Relevant Statute
22. The computer program product of claim 21 wherein the program code for establishing a point-to-point communication link further comprises:	**35 U.S.C. § 103 – VocalChat**
program code for querying the server as to the on-line status of the first callee process; and	As described above, when NetWare was used, the VocalChat client queried existing NetWare Bindery services to locate "currently logged-in users;" when Windows for Workgroups was used, the VocalChat client queried the Windows for Workgroups services to locate online users; and when other protocols, such as TCP/IP and NetBIOS were used, the VocalChat client queried the shared Connection List file (CONNLIST.VC). *See, e.g.*, Help File, page 2. Regardless of which protocol was used, the query determined the online status of the callee process (the VocalChat client of a callee). For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction was made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieved a list of "currently logged in users."
program code for receiving a network protocol address of the first callee process over the computer network from the server.	As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. Generic User VocalChats files for users information. (See Generic network, below). Help File, page 26. Thus, when any protocol other than Netware or Windows for Workgroups was used (such as TCP/IP or NetBIOS), a "generic" method was used in which the VocalChat client queried VocalChat files (the Connection List/USERS files) to locate users on the network. As described in greater detail in the Help File: When **NetWare** is used, VocalChat uses the NetWare Bindery services to get the list of servers, known users and groups on each server, currently logged-in users, and the addresses of specific users. When **Windows for Workgroups** is used, VocalChat uses the Window for Workgroups users services to get the list of workgroups and computers

Claims	Prior Art and Relevant Statute
	(there is no need for user addresses here). When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. Thus, when NetWare was used, the VocalChat client queried existing NetWare Bindery services to locate "currently logged-in users;" when Windows for Workgroups was used, the VocalChat client queried the Windows for Workgroups services to locate online users; and when other protocols, such as TCP/IP and NetBIOS were used, the VocalChat client queried the shared Connection List file (CONNLIST.VC). Regardless of which protocol was used, the query determined whether the second process (the VocalChat clients of other users) was connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction was made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieved a list of "currently logged in users."
23. A computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – VocalChat in view Pinard**
program code for generating an element representing a second communication line.	The graphical user interface (GUI) in Pinard clearly shows an element representing a second communication line. See, e.g., Pinard, Figure 6 (illustrating a first call icon (23) representing a first communication line and a second call icon (29) representing a second communication line.)
24. The computer program product of claim 23 further comprising:	**35 U.S.C. § 103 – VocalChat in view Pinard**
program code, responsive to the user disassociating the element representing	Figure 6 of Pinard illustrates how the call represented by call icon 23 is terminated by dragging the user icon for "John" 21 out of the

Claims	Prior Art and Relevant Statute
the first callee process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process; and	call icon 23. Similarly, Figure 11 illustrates how a call is terminated by dragging the user icon to a "waste basket" icon 26.
program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process.	In Figure 6, the callee process icon for "John" 21 is dragged from call icon 23 to call icon 29, thereby terminating the call represented by call icon 23 and establishing a different link with the callee process represented by icon 21 (in this case a conference call with "John," "Mary," and "Debbie"). Moreover, once a callee is removed from a call by clicking and dragging the callee's icon, a new call can always be established with the callee by dragging the callee's icon to a call setup icon. *See, e.g.*, Pinard, Figure 3 (showing a callee icon dragged from a directory to a call setup icon to establish a call). *See also* Pinard, 4:22-31.
25. The computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – VocalChat in view Pinard**
program code for generating an element representing a second callee process; and	In Figure 6 of Pinard, the user interface element for "John" 21 is "an element representing a second callee process."
program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process.	In Figure 6 of Pinard, the user interface element for "John" 21 is dragged from call icon 23 to call icon 29, thereby creating a conference call between "John," "Mary," and "Debbie." *See* Pinard, 5:31-44 ("Now to conference all parties, the user Debbie merely drags the John icon to the call icon 29.")
26. The computer program product of claim 25 further comprising:	**35 U.S.C. § 103 – VocalChat in view Pinard**
program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link.	In Pinard, any callee process can be removed from a conference call by dragging the element representing the callee process from the conference call icon. For example, Figure 8 of Pinard shows the user icon "Debbie" removed from conference call represented by call icon 32, thereby "breaking Debbie's line from the conference." Pinard, 6:14-15.

Claims	Prior Art and Relevant Statute
27. The computer program product of claim 21 further comprising:	**35 U.S.C. § 103 – VocalChat in view Pinard**
program code for generating an element representing a communication line having a temporarily disabled status; and	Examples of a "temporarily disabled status" provided in the '704 patent include "line on hold" and "line on mute." *See, e.g.*, '704 patent, Claims 17 and 18. Pinard describes a user interface element representing a communication line having a temporarily disabled status. For example, Figure 12 illustrates a "hard hold" icon 39 to which user icons representing callers/callees 41 may be dragged to put the callers/callees on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process.	In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
28. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on hold status.	**35 U.S.C. § 103 – VocalChat in view Pinard** In Pinard, in response to an icon of a caller/callee 41 being moved into the hard hold icon 39, the caller/callee is placed on hold. *See, e.g.*, Pinard, 6:36-53 ("To place Mary on hard hold, Debbie drags Mary's icon 28 to the hard hold icon 39.").
29. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on mute status.	**35 U.S.C. § 103 – VocalChat in view Pinard** The VocalChat User's Guide describes a "communication line on mute status" as recited in Claim 29. As described in the VocalChat User's Guide, "Manual Activation can also be used like the MUTE option in many phones: it lets you talk without being heard on the other user's system." User's Guide, page 57.
30. A computer program product of claim 21 wherein the computer system further comprises a visual display and the user interface comprises a graphic user interface.	**35 U.S.C. § 103 – VocalChat** The VocalChat client included a graphical user interface capable of being displayed on a "visual display." *See, e.g.*, User Guide, page 11 (illustrating the primary VocalChat GUI including a Call button, a volume slider and a plurality of Quick Dial buttons).

Claims	Prior Art and Relevant Statute
31. The computer program product of claim 30 wherein the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises:	**35 U.S.C. § 103 – VocalChat** The elements in VocalChat were graphic elements. *See, e.g.*, User's Guide, page 11, 14, 20-21 (illustrating various features of the VocalChat GUI).
program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process.	A VocalChat user could make a point-to-point call to another user by using the Call button or a Quick Dial Button representing a frequently called callee. *See* User Guide, page 14 ("just select a user from the user list, and choose "OK"). Callees were also represented as Quick Dial buttons. *See* User's Guide, pages 11, 20-21 ("Setting a Quick Dial Button").
32. A method of locating a process over a computer network comprising the steps of:	**35 U.S.C. § 103 – VocalChat**
a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network; and	In the initial VocalChat implementations (versions 1.x) each VocalChat client transmits its name and network protocol address to a USERS file stored on a server. As described in VocalChat Network Information: When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the <u>names of logged in users</u>. <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the <u>user name is kept in the file</u>. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*. VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own comptuers host name and

Claims	Prior Art and Relevant Statute
	IP address."). Later VocalChat implementations (e.g., version 2.02), refer to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users"). While VocalChat does not explicitly describe a server with stored names and addresses is accessible over "the Internet," it describes the use of TCP/IP, which is the protocol used on the Internet. Thus, VocalChat inherently describes that the list of users and network addresses is accessible over the Internet. Moreover, the Internet is a type of Wide Area Network (WAN) and VocalChat describes a WAN implementation. For example, the Help File describes that "[o]ver a WAN . . . it is advisable to create local copy of the executables and DLLs, and reference only the Post Office over the low-speed [WAN] connection." Help File, page 4. On many networks, including TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a

Claims	Prior Art and Relevant Statute
	well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
b. in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process.	As discussed above, a VocalChat caller sends the directory server a query identifying a particular callee. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server identifies an entry in the directory corresponding to the identified callee (the Connection List file in a TCP/IP implementation), and, if the callee is connected, provides the corresponding IP address associated with that callee in the directory to the caller.
33. A method for locating processes having dynamically assigned network protocol addresses over a computer network, the method comprising the steps of:	**35 U.S.C. § 103 – VocalChat** As described above, VocalChat clients rely on a central server to locate the network addresses of other VocalChat clients. As described in VocalChat Network Information: > When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. > <u>Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address</u>. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*. VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the

Claims	Prior Art and Relevant Statute
	TCP/IP software to recognize your own comptuers host name and IP address.”). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a “Connection List” file. *See, e.g.*, Help File, page 2 (“a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory.”). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 (“**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users.”). *See also* Readme File, page 2 (“VocalChat creates a central directory on the network, shared by all users called ‘Post-Office.’ All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory.”); Help File, page 8 (“the Setup program crates a Connection List File which is used to identify and access users”). On many networks, including TCP/IP networks, network addresses are assigned dynamically “following connection to the computer network.” *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) (“RFC 1531”), Section 2.2 (describing the “dynamic allocation of network addresses” on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for “automatic reuse of an address that is no longer needed by the host to which it was

Claims	Prior Art and Relevant Statute
	assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
a. maintaining, in a computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and	The server on which the Connection List/USERS file is maintained inherently includes a computer memory. Moreover, the server stores a network accessible compilation of entries including a network protocol address and a name (corresponding identifier) of a process connected to the computer network. As described in VocalChat Network Information: When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*. VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own comptuers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users").

Claims	Prior Art and Relevant Statute
b. in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process.	As discussed above, a VocalChat caller sends the directory server a query identifying a particular callee. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server identifies an entry in the directory corresponding to the identified callee (the Connection List file in a TCP/IP implementation), and, if the callee is connected, provides the corresponding IP address that is associated with that callee in the directory to the caller.
34. The method of claim 33 further comprising the step of:	**35 U.S.C. § 103 – VocalChat**
c. modifying the compilation of entries.	The compilation of entries stored on the server (e.g., in the Connection List/USERS file) was modified as new users installed VocalChat software and as existing users logged in and logged off. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file, which was modified in the same manner as the USERS file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory.").
35. The method of claim 34 wherein step c further comprises:	**35 U.S.C. § 103 – VocalChat**
c.1 adding an entry to the compilation upon the occurrence of a predetermined event.	An entry is added to the compilation of entries within the Connection List/USERS file when a user first sets up VocalChat or when the user logs in to VocalChat. *See, e.g.*, VocalChat Network Information, page 10 ("Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address"); Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses . . . the user name for each user, is entered when performing the User Installation in the Setup program"); page 4 (user installation "is used to . . . add the user to the Address Book"); page 10 (user name "will be used by VocalChat to identify you and will appear in the VocalChat Address Book and in the Connection List file").

Claims	Prior Art and Relevant Statute
36. The method of claim 35 wherein the predetermined event comprises notification by a user process of an assigned network protocol address.	**35 U.S.C. § 103 – VocalChat** When a user logs in, or when a computer with VocalChat is turned on, the network address of the VocalChat client is sent to the Connection List/USERS file. For example, as described in VocalChat Network Information, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address." VocalChat Network Information, page 10.
37. The method of claim 34 wherein step c further comprises:	**35 U.S.C. § 103 – VocalChat**
c.1 deleting an entry from the compilation upon the occurrence of a predetermined event.	VocalChat inherently discloses this limitation. Any database containing entries, such as the one used on the server containing the Connection List/USERS file, is inherently capable of deleting entries upon request from an end user and/or a network administrator. Moreover, when a user logged off the system, the user's network address was deleted from the list of "on-line" users. *See, e.g.,* VocalChat Network Information, page 10 ("VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. <u>When exiting VocalChat, the address is removed</u>, but the user name is kept in the file.") (emphasis added).
38. A computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium the program code comprising:	**35 U.S.C. § 103 – VocalChat** VocalChat is a software-based telephone executed on personal computers which connects to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols."). As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A").

Claims	Prior Art and Relevant Statute
a. program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and	The server on which the Connection List/USERS file is maintained inherently includes a computer memory. Moreover, the server stores a network accessible compilation of entries including a network protocol address and a name (corresponding identifier) of a process connected to the computer network. As described in VocalChat Network Information: > When the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. > Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file. Thus other users can add this user's name as a Quick Dial button even if the user is not running VocalChat at the moment. However, in order for VocalChat to work properly, *all users must have access to the same USERS file, and all must have read/write access to that file*. VocalChat Network Information, page 10 (underline emphasis added); Troubleshooting Help File, page 28 ("VocalChat needs the TCP/IP software to recognize your own comptuers host name and IP address."). Later VocalChat implementations (e.g., version 2.02), refers to the USERS file as a "Connection List" file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory."). Regardless of the file name, the Connection List/USERS file is stored on a server for access by VocalChat clients. *See, e.g.*, VocalChat Network Information, page 2 ("**Server Installation** is used to install the VocalChat program files on the network, for use by the different network users."). *See also* Readme File, page 2 ("VocalChat creates a central directory on the network, shared by all users called 'Post-Office.' All users must use the same Post-Office, otherwise they won't be able to communicate or leave messages to each other. This means that all users must be attached to one file-server which will be used for the Post-Office, and all have write permission for the Post-Office directory."); Help File, page 8 ("the Setup program crates a Connection List File which is used to identify and access users"). On many networks, including TCP/IP networks, network addresses are assigned dynamically "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks).

Claims	Prior Art and Relevant Statute
	Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
b. program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process.	As discussed above, a VocalChat caller sends the directory server a query identifying a particular callee. *See, e.g.*, Help File, page 22 (VocalChat "will use the CONNLIST.VC files to get network addresses"); page 8 ("the Setup program creates a Connection List file which is used to identify and access users"). Consequently, the server identifies an entry in the directory corresponding to the identified callee (the Connection List file in a TCP/IP implementation), and, if the callee is connected, provides the corresponding IP address that is associated with that callee in the directory to the caller.
39. The computer program product of claim 38 further comprising:	
c. program code configured to modify the compilation of entries.	**35 U.S.C. § 103 – VocalChat** The compilation of entries stored on the server (e.g., in the Connection List/USERS file) was modified as new users installed VocalChat software and as existing users logged in and logged off.

Claims	Prior Art and Relevant Statute
	For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Later VocalChat implementations (e.g., version 2.02), referred to the USERS file as a "Connection List" file, which was modified in the same manner as the USERS file. *See, e.g.*, Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory.").
40. The computer program product of claim 39 wherein program code configured to modify comprises:	**35 U.S.C. § 103 – VocalChat**
c.1 program code configured to add an entry to the compilation upon the occurrence of a predetermined event.	An entry is added to the compilation of entries within the Connection List/USERS file when a user first sets up VocalChat or when the user logs on to VocalChat. *See, e.g.*, VocalChat Network Information, page 10 ("Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address"); Help File, page 2 ("a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses . . . the user name for each user, is entered when performing the User Installation in the Setup program"); page 4 (user installation "is used to . . . add the user to the Address Book"); page 10 (user name "will be used by VocalChat to identify you and will appear in the VocalChat Address Book and in the Connection List file").
41. The computer program product of claim 40 wherein the predetermined event comprises notification by a process of an assigned network protocol address.	**35 U.S.C. § 103 – VocalChat** When a user logs in, or when a computer with VocalChat is turned on, the network address of the VocalChat client is sent to the Connection List/USERS file. For example, as described in VocalChat Network Information, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address." VocalChat Network Information, page 10.
42. The computer program product of claim 38 wherein step c further comprises:	**35 U.S.C. § 103 – VocalChat**
c.1 program code configured to delete an	VocalChat inherently discloses this limitation. Any database

Claims	Prior Art and Relevant Statute
entry from the compilation upon the occurrence of a predetermined event.	containing entries, such as the one used on the server containing the Connection List/USERS file, is inherently capable of deleting entries upon request from an end user and/or a network administrator. Moreover, when a user logged off the system, the user's network address was deleted from the list of "on-line" users. *See, e.g.,* VocalChat Network Information, page 10 ("VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file.") (emphasis added).
43. A computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein, the program code comprising:	**35 U.S.C. § 103 – VocalChat** VocalChat is a software-based telephone executed on personal computers which connected to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols."). As software, VocalChat is inherently stored as program code on a computer-usable medium. *See, e.g.*, Readme, page 1 (listing the VocalChat files copied during installation). *See also* VocalChat User's Guide, page 8 (describing how VocalChat is installed by inserting "the VocalChat Disk in drive A").
a. program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network; and	In a TCP/IP implementation, the server on which the Connection List/USERS file is located transmits the network protocol address of a second VocalChat client (second process) to a first VocalChat client (first process) upon request. As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. Generic User VocalChats files for users information. (See Generic network, below). Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File:

Claims	Prior Art and Relevant Statute
	When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. Thus, With NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10 (emphasis added). Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users." **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is

Claims	Prior Art and Relevant Statute
	a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
b. program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network.	VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").
44. In a first computer process operatively coupled over a computer network to a second process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of:	**35 U.S.C. § 103 – VocalChat** VocalChat is a software-based telephone executed on personal computers which connected to a central server to locate other personal computers on a variety of computer networks, including TCP/IP, NetBIOS, and IPX networks. *See, e.g.*, VocalChat User's Guide, page 5 (illustrating a central server with a "post office" to enable communication between computers). *See also id.*, pages 7-8 (describing minimum personal computer requirements as a "386SX or higher IBM-compatible computer"); Readme, page 1 (listing the VocalChat files copied during installation); Help File, page 2 ("VocalChat can work with IPX, NetBIOS and TCP/IP network protocols.").
A. following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network;	As discussed above, VocalChat clients connect to a server to locate and establish point-to-point connections with other VocalChat clients over a network. For example, VocalChat discloses that "[u]ser-to-user access is facilitated automatically through the [Connection List] file" which is stored on a server. Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons."). Because the server stores network addresses of logged in clients, it is an "address server." Inherently, a VocalChat client transmits its network protocol address "following connection of the [VocalChat client] to the computer network." Moreover, on many networks, including

Claims	Prior Art and Relevant Statute
	TCP/IP networks, network addresses are assigned "following connection to the computer network." *See, e.g.*, Dynamic Host Configuration Protocol, RFC 1531 (Oct. 1993) ("RFC 1531"), Section 2.2 (describing the "dynamic allocation of network addresses" on TCP/IP networks). Thus, in at least some instances, a computer system executing VocalChat receives its IP address following connection to the computer network. Consequently, dynamic address assignment is inherent in the VocalChat system. Alternatively, as set forth below, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. **35 U.S.C. § 103 – VocalChat References in view of RFC 1531** Alternatively, it would have been obvious to one of ordinary skill in the art to use dynamic address assignment on a TCP/IP network. A motivation to combine the VocalChat references with RFC 1531 exists because the VocalChat references describe the VocalChat software operating on a TCP/IP network and RFC 1531 describes a well known technique for dynamically assigning IP addresses within a TCP/IP network. One of ordinary skill in the art would have been motivated to combine VocalChat with RFC 1531 to realize the benefits associated with dynamic IP address assignment. For example, one of ordinary skill in the art would have been motivated to use dynamic IP address assignment because it eliminates the burdensome task of manually assigning IP addresses for all networked computers and allows for "automatic reuse of an address that is no longer needed by the host to which it was assigned." RFC 1531, page 2 (Section 1, Introduction). In fact, it is a virtual certainty that VocalChat software was executed on personal computers which had IP addresses dynamically assigned.
B. querying the address server as to whether the second process is connected to the computer network;	In a TCP/IP implementation, the server on which the Connection List/USERS file is located receives queries from VocalChat clients (first processes) to determine the on-line status of other VocalChat clients (second processes). As described in the Help File: Method of determining users address: Netware Get Users information from Netware 2.x/3.x bindery WinWorkgroups Get users information from Windows for Workgroups. Generic User VocalChats files for users information. (See Generic network, below). Help File, page 26. With any protocol other than Netware or Windows for Workgroups (such as TCP/IP or NetBIOS), a "generic" method is used where the VocalChat client queries

Claims	Prior Art and Relevant Statute
	VocalChat files (the Connection List/USERS files) locating users on the network. As described in greater detail in the Help File: When **NetBIOS** or **IPX** are used, but not with NetWare or Window for Workgroups, or when TCP/IP is used, a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses. This file is placed in the Post Office directory. In this case, the user name for each user, is entered when performing the **User Installation** in the Setup program. You should make sure that this name is not used by any other user on the network. Help File, page 2. Thus, with NetWare, the VocalChat client queries existing NetWare Bindery services locating "currently logged-in users;" with Windows for Workgroups, the VocalChat client queries the Windows for Workgroups services locating online users; and with other protocols, such as TCP/IP and NetBIOS, the VocalChat client queries the shared Connection List file (CONNLIST.VC). Regardless of protocol, the query determines whether the second process (the VocalChat client of another user) is connected to the computer network. For example, "[w]hen the network used is not NetWare or Windows for Workgroups, VocalChat maintains a shared USERS file with the names of logged in users. Each time a user loads VocalChat, its entry in the USERS file is updated with its IPX/NetBIOS address. When exiting VocalChat, the address is removed, but the user name is kept in the file." VocalChat Network Information, page 10. Thus, a distinction is made between logged in users and logged out users. Similarly, as described above, in the NetWare implementation, the query retrieves a list of "currently logged in users."
C. receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network; and	As described above, with TCP/IP, "the network addresses of the different workstations are required for VocalChat to be able to access the network users." Help File, page 2. Moreover, with TCP/IP, "a shared CONNLIST.VC file is used by the different running copies of VocalChat to hold user names and addresses." The ability to establish a call merely by identifying a unique user name demonstrates that the server "transmits" the network addresses of VocalChat callees from the server's directory database. Indeed, the directory database is used to "get . . . the addresses of specific users." *Id.*; *see also id.*, page 22 ("VocalChat will use the CONNLIST.VC files to get network addresses[.]").
D. in response to the network protocol	VocalChat discloses that "[u]ser-to-user access is facilitated

Claims	Prior Art and Relevant Statute
address of the second process, establishing a point-to-point communication link with the second process over the computer network.	automatically through the [Connection List] file." Help File, page 17. VocalChat also discloses "the peer-to-peer nature of Windows for Workgroups, which VocalChat "uses . . . for user services." *Id.* In fact, VocalChat is a voice over computer network product for use on various networks that "enables communication between" VocalChat users. *Id.* at 8. *See also* User Guide, page 2 ("Talk with other users over the network, and broadcast to network users or groups. Access network users with the Address Book and Quick-Dial buttons.").

Exhibit P

Comments on Arguments Made by Net2Phone's Expert to Distinguish Over NetBios

---

Customer No.: 8791

Blakely, Sokoloff, Taylor & Zafman, LLP
Sunnyvale, California 94085-4040
Telephone (408) 720-8300
Fax (408) 720-8383

EXHIBIT P
COMMENTS ON ARGUMENTS MADE BY NET2PHONE'S EXPERT TO DISTINGUISH OVER NETBIOS

During the Net2Phone Litigation, Net2Phone attempted to distinguish the claims of the '704 patent over NetBIOS. The court has yet to render an opinion on these arguments. As set forth below, these arguments fail to distinguish the claims of the '704 patent over NetBIOS for a variety of reasons.

A. Net2Phone argued that NetBIOS does not disclose "Processes" because, according to Net2Phone, NetBIOS describes a set of software components executed at the operating system level.

This argument fails for at least two reasons. First, the claims of the '704 patent are not limited to software processes at the application layer (i.e., outside of the operating system). No meaningful distinction can be made between processes executed as applications and processes executed within an operating system. In both cases, they are "processes" as recited in the claims of the '704 patent.

Second, NetBIOS, in fact, describes that the NetBIOS software may be implemented either as an application or within an operating system. For example, Appendix B of NetBIOS provides "Implementation Models" for implementing NetBIOS as applications and within an operating system. *See, e.g.*, NetBIOS at 411 (describing a "Combined Service and Application Model" in which "[t]he NetBIOS service and application are both contained within a single process" as well as a "Common Kernel Element Model" in which "[t]he NetBIOS Service is part of the operating system (perhaps as a device driver or a front-end processor).)."

Thus, Net2Phone's assertion that NetBIOS does not describe "processes" as claimed in the '704 patent is incorrect.

B. Net2Phone argued that NetBIOS describes a system designed to work on Local Area Networks (LANs) and Wide Area Networks (WANs) but not on the Internet.

This argument is both untrue and irrelevant. First, NetBIOS explicitly describes that NetBIOS name servers (NBNS) and clients may be connected over the Internet. For example, the Figure on page 371 of NetBIOS clearly shows a NBNS and multiple client nodes connected over the "INTERNET." In addition, as recited on the first page of RFC 1001:

> This RFC defines a proposed standard protocol to support NetBIOS services in a TCP/IP environment. Both local network and internet operation are supported. Various node types are defined to accommodate local and internet topologies and to allow operation with or without the use of IP broadcast.

NetBIOS *id.* at 350. *See also id.* at 355 ("This RFC specifies a proposed standard for the Internet community. Since this topic is new to the Internet community, discussions and suggestions are specifically requested."); *id.* at 358 (due to the need to "ALLOW INTERNET OPERATION . . .[t]he proposed standard recognizes the need for NetBIOS operation across a set of networks interconnected by network (IP) level relays (gateways)."). NetBIOS also describes that it is interoperable with the Internet's Domain Name System. *See, e.g., id.* at 368 ("The NBNS design attempts to align itself with the Domain Name System in a number of ways. First, the NetBIOS names are encoded in a form acceptable to the domain name system."). *See also id.* at 367 ("A single NBNS may be implemented as a distributed entity, such as the Domain Name Service.").

Second, even the LAN embodiments described in NetBIOS anticipate the claims of the '704 patent. Specifically, a single NBNS on a LAN with two or more NetBIOS clients connected to the LAN communicate using the Internet Protocol, the same protocol used on the Internet. *See, e.g., id.* at 1 ("This RFC defines a proposed standard protocol to support NetBIOS services in a TCP/IP environment."). Thus, the NBNS performs the same basic operations on a

LAN using TCP/IP as it does on "the Internet" (i.e., responding to queries for IP addresses from caller nodes to identify the locations of callee nodes).

Thus, Net2Phone's assertion that the NetBIOS reference does not operate on the Internet is both incorrect and irrelevant.

C. Net2Phone Argued That the NetBIOS protocol does not provide for permanently-assigned unique identifiers.

Once again, this assertion is both irrelevant and untrue. It is irrelevant because the claims of the '704 patent do not require "permanently-assigned unique identifiers." Claims 32, 33 and 38 simply require "an identifier" to identify a process on the network, not a "unique" or "permanent" identifier.

The assertion is untrue because the NetBIOS names are permanently-assigned unique identifiers. As stated unambiguously in NetBIOS, "[e]very node has a permanent unique name." *Id.* at 376. Net2Phone calls this statement "aspirational" because two nodes may improperly acquire the same name, resulting in an error condition. As described in NetBIOS, "[a] unique name should be held by only one station at a time. However, duplicates ('name conflicts') may arise due to errors." NetBIOS also describes how name conflicts are resolved, thereby ensuring that each node is identified by a unique name. *See, e.g.*, *id.* at 380 (if a node improperly acquires a name which is already in use "can no longer be used by that node for any session establishment or sending or receiving datagrams."). *See also id.* ("The only valid user function against a [name with a conflict] is DELETE NAME. Any other user NetBIOS function returns immediately with an error code of 'NAME CONFLICT'."). Consequently, using name conflict detection and resolution techniques, NetBIOS ensures that each node has a permanent unique name. *See id.* at 379 (section entitled "Name Conflicts").

D. Net2Phone Argued That a Name Query to an NBNS was limited to the "Scope" associated with that NBNS and, as such, a NetBIOS network would

> **be "implemented in a series of nonintersecting network fragments, not in a large single network such as the Internet with a global server designed to facilitate communications among all of the client processes connected to that network."**

Once again, this assertion is both irrelevant and untrue. It is irrelevant because any network which operates using NetBIOS, including a single LAN within a single "scope," includes all of the limitations found in the claims. The particular size of the network is irrelevant.

The assertion is untrue because NetBIOS naming was structured so as to be compatible with the Internet's Domain Name Service (DNS). As described in NetBIOS:

> The NBNS design attempts to align itself with the Domain Name System in a number of ways.
>
> First, the NetBIOS names are encoded in a form acceptable to the domain name system.
>
> Second, a scope identifier is appended to each NetBIOS name. This identifier meets the restricted character set of the domain system and has a leading period. This makes the NetBIOS name, in conjunction with its scope identifier, a valid domain system name.
>
> Third, the negotiated responsibility mechanisms permit the NBNS to be used as a simple bulletin board on which are posted (name, address) pairs. This parallels the existing domain sytem query service.

NetBIOS, page 368. *See also id.* at 367 ("A single NBNS may be implemented as a distributed entity, such as the Domain Name Service."); 426 ("The NetBIOS Name Service packets follow the packet structure defined in the Domain Name Service (DNS)."); *id.* at 502 ("[the NetBIOS SCOPE_ID] is expressed as a character string meeting the requirements of the domain name system and without a leading or trailing 'dot'. An implementation may elect to make this a single global value for the node or allow it to be specified with each separate NetBIOS name (thus permitting cross-scope references."). Consequently, NetBIOS describes that NetBIOS names may be used across network “Scopes” and, in fact, may be integrated within the Internet using the Domain Name Service.

Substitute for Form 1449/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Application Number	Reexamination of 6,108,704
				Filing Date	Herewith
				First Named Inventor:	
				Art Unit	
				Examiner Name	
Sheet	1	**of**	1	Attorney Docket Number	003801.G184

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (If known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	Exhibit F	US-	5,533,110	07-02-1996	Pinard, Deborah L., et al.	

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published	T[2]
	Exhibit B	The Open Group, Technical Standard, Protocols for X/Open PC Interworking: SMB, Version 2, 1992, pages ii-xvi and pages 1-516.	
	Exhibit C	ZELLWEGER, POLLE T., et al., Etherphone: Collected Papers 1987-1988, Xerox Corporation, May 1989.	
	Exhibit D	VIN, HERRICK M., et al, Multimedia Conferencing in the Etherphone Environment, October 1991, pages 69-79.	
	Exhibit E	DROMS, R., Dynamic Host Configuration Protocol, RFC 1531, Bucknell University, October 1993, pages 1-39.	
	Exhibit G	VocalChat User's Guide Version 2.0, Vocaltec, 1994, pages 1-77.	
	Exhibit H	README, VocalChat Version 2.02 & VocalChat WAN Version 2.02, Vocaltec, June 1994, pages 1-3.	
	Exhibit I	VocalChat 1.01 Network Information, Vocaltec, 1994, pages 1-10.	
	Exhibit J	VocalChat Information, Vocaltec, 1994, pages 1-31.	
	Exhibit K	VocalChat Troubleshooting, Vocaltec, 1994, pages 1-101.	

Examiner Signature		Date Considered	

*Examiner: Initial if reference 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.

[1]Applicant's unique citation designation number (optional). [2]Applicant is to place a check mark here if English Translation is attached.

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	Point-to-Point Internet Protocol
First Named Inventor/Applicant Name:	Glenn W. Hutton
Filer:	Edwin Taylor/carla vignola
Attorney Docket Number:	3801G184

Filed as Large Entity

ex parte reexam Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Request for ex parte reexamination	1812	1	2520	2520
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
		Total in USD ($)		**2520**

Electronic Acknowledgement Receipt

EFS ID:	4809885
Application Number:	90010416
International Application Number:	
Confirmation Number:	1061
Title of Invention:	Point-to-Point Internet Protocol
First Named Inventor/Applicant Name:	Glenn W. Hutton
Customer Number:	08791
Filer:	Edwin Taylor/carla vignola
Filer Authorized By:	Edwin Taylor
Attorney Docket Number:	3801G184
Receipt Date:	17-FEB-2009
Filing Date:	
Time Stamp:	20:20:11
Application Type:	Reexam (Third Party)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$2520
RAM confirmation Number	5779
Deposit Account	022666
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Miscellaneous Incoming Letter	3801G184_REQUEST_REEXAMINATIONTransmittal_filed_2-17-09.pdf	26904 b84f4bd27d186d5ee96ed5b923d5bc69cbc71384	no	2
Warnings:					
Information:					
2		3801G184_ExParteReexam_filed_2-17-09.pdf	1317032 795c63f360c6de7acf0097ef7962dd007f6f0c19	yes	172

Multipart Description/PDF files in .zip description

Document Description	Start	End
Receipt of Original Ex Parte Reexam Request	1	171
Reexam Certificate of Service	172	172

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
Warnings:					
Information:					
3	Copy of patent for which reexamination is requested	3801G184_ExhibitA.PDF	1192985 94a0fc23c4559fcaa00d56f4ad3f541b67893462	no	17
Warnings:					
Information:					
4	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitB.pdf	1801369 c103526ca50919aa753ce30046ef708a5699c151	no	535
Warnings:					
Information:					
5	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitD.pdf	2178857 3a7030b24403c468e3286f76c887e2c584cb0679	no	12
Warnings:					
Information:					
6	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitE.pdf	145608 0a4112ad4d35e657d3d08118acee8c042b39d37e	no	36
Warnings:					
Information:					
7	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitF.pdf	713931 2958ef3c79157e849eaf5055b9f7d5e2ac251d53	no	15

Warnings:					
Information:					
8	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitH.pdf	51633 2181524ea493d3c20f587900895c6e1bc79edd13	no	4
Warnings:					
Information:					
9	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitC.pdf	13352064 590cc954eb96c4f8fd535f53e57ec07edbae507e	no	66
Warnings:					
Information:					
10	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitG.pdf	504025 fa60810c5dd34389e88917c3aca6fac650ca848e	no	84
Warnings:					
Information:					
11	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitI.pdf	1451797 c1520efa3eb179f7624023dc2397172fc87306aa	no	11
Warnings:					
Information:					
12	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitK.PDF	2807133 7dfbdfe997c62a29a8d3301336006eb6e3049d6d	no	102
Warnings:					
Information:					
13	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitL.pdf	387428 6abdc0142e45ceab63e0eef5a5de5732e9ddaf97	no	3
Warnings:					
Information:					
14	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitM.pdf	283503 d91be25395f5475292100e3d041ac0126ac7f4be	no	41
Warnings:					
Information:					
15	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitN.pdf	252985 107ad95ea6dcc321d1312ab1b73b28369dc5aa66	no	36
Warnings:					
Information:					
16	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitO.pdf	280581 b04fd638172[illegible]0cfa6	no	42

Warnings:					
Information:					
17	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitP.pdf	66253 6a7d0ccc88476b7fc1c9989f2bebbabbe85eabc9	no	6
Warnings:					
Information:					
18	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitJ.pdf	202590 118ce9e35be24f5c76e8ee4e4ed928f748286c0c	no	32
Warnings:					
Information:					
19	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitQ.pdf	67055 0b8e18e06da86b6c39ffdbc796f4130397e55ff4	no	4
Warnings:					
Information:					
20	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitR.pdf	72783 8e196549b2da336a3beb210e306d028e79c36d2d	no	5
Warnings:					
Information:					
21	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitT.PDF	3107593 acc55ef61770d090d92db7c4d8c80568c1c729eb	no	38
Warnings:					
Information:					
22	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitV.PDF	2956851 8f7fec64984e8484b2274e3dfb4321c62bc6466a	no	36
Warnings:					
Information:					
23	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitS.PDF	302208 9994dc2772591b9f14bfd0da6e1360b8d1c6a8be	no	39
Warnings:					
Information:					
24	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitU.PDF	272809 6833d6a4bcc778d65dbb8c29ff9ad044c7ac7b43	no	39
Warnings:					
Information:					
25	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitX.pdf	146362 8e1bf1026e1[illegible]9d15	no	20

Warnings:					
Information:					
26	Reexam - Affidavit/Decl/Exhibit Filed by 3rd Party	3801G184_ExhibitW.PDF	140011 0d7c0a6dc5234a0a36a09df267b05939aa1b5095	no	18
Warnings:					
Information:					
27	Reexam - Info Disclosure Statement Filed by 3rd Party	3801G184_1449_filed_2-17-09.pdf	40834 224393d9b480d4b72e6c129e2ac9e91d2cfcb552	no	2
Warnings:					
Information:					
28	Fee Worksheet (PTO-06)	fee-info.pdf	29274 b2bed7f3c03f11b0c73ef3a9bb7024ab2d194610	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	34152458	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Application Number	Application/Control No.	Applicant(s)/Patent under Reexamination	
	90/010,416	6108704	
	Examiner	**Art Unit**	
		3992	

U.S. Patent and Trademark Office

Part of Paper No. 20090224

Index of Claims

	Application/Control No.	Applicant(s)/Patent under Reexamination	
	90/010,416	6108704	
	Examiner	Art Unit	
		3992	

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

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

Claim		Date								
Final	Original									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	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									

Claim		Date								
Final	Original									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

Issue Classification	Application/Control No. 90/010,416	Applicant(s)/Patent under Reexamination 6108704
	Examiner	Art Unit 3992

ISSUE CLASSIFICATION

ORIGINAL	
CLASS	SUBCLASS
709	227

CROSS REFERENCES

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			

INTERNATIONAL CLASSIFICATION

CLAIMED				NON-CLAIMED			
			/				/
			/				/
			/				/
			/				/
			/				/
			/				/
			/				/

(Assistant Examiner) (Date)

(Legal Instruments Examiner) (Date)

(Primary Examiner) (Date)

Total Claims Allowed:

O.G. Print Claim(s)	O.G. Print Fig.

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		31		61		91		121		151		181
	2		32		62		92		122		152		182
	3		33		63		93		123		153		183
	4		34		64		94		124		154		184
	5		35		65		95		125		155		185
	6		36		66		96		126		156		186
	7		37		67		97		127		157		187
	8		38		68		98		128		158		188
	9		39		69		99		129		159		189
	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
	15		45		75		105		135		165		195
	16		46		76		106		136		166		196
	17		47		77		107		137		167		197
	18		48		78		108		138		168		198
	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

Reexamination	Application/Control No. 90/010,416	Applicant(s)/Patent Under Reexamination 6108704
	Certificate Date	**Certificate Number**

Requester Correspondence Address:	☐ **Patent Owner**	☒ **Third Party**

Edwin H. Taylor
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040

LITIGATION REVIEW ☐	(examiner initials)	(date)

Case Name	Director Initials

COPENDING OFFICE PROCEEDINGS

TYPE OF PROCEEDING	**NUMBER**
1.	
2.	
3.	
4.	

U.S. Patent and Trademark Office DOC. CODE **RXFILJKT**

Search Notes	Application/Control No. 90/010,416	Applicant(s)/Patent under Reexamination 6108704	
	Examiner	Art Unit 3992	

SEARCHED

Class	Subclass	Date	Examiner

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 1061

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
90/010,416	02/17/2009 RULE	709	3992	3801G184

APPLICANTS

6108704, Residence Not Provided;
NET2PHONE, INC.(OWNER), Newark, NJ;
Edwin H. Taylor(3RD PTY REQ), Sunnyvale, CA;
Edwin H. Taylor, Sunnyvale, CA

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

This application is a REX of 08/533,115 09/25/1995 PAT 6,108,704

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

	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☐ no 35 USC 119 (a-d) conditions met ☐ yes ☐ no ☐ Met after Allowance Verified and Acknowledged ______ Examiner's Signature Initials			44	10

ADDRESS

Jeffrey S. Ginsberg, ESQ.
KENYON & KENYON
One Broadway
Neew York, NY10004

TITLE

Point-to-Point Internet Protocol

FILING FEE RECEIVED		
2520	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

Litigation Search Report CRU 3999

Reexam Control No 90/010,416

To: Examiner Location: CRU Art Unit: 3999 Date: 02/18/2009 Case Serial Number: 90/010,416	From: Renee M. Preston Location: CRU 3999 MDW 7C84 Phone: (571) 272-1607 Renee.preston@uspto.gov

Search Notes

U.S. Patent No. 6,108,704

1) I performed a KeyCite Search in Westlaw, which retrieves all history on the patent including any litigation.

2) I performed a search on the patent in Lexis CourtLink for any open dockets or closed cases.

3) I performed a search in Lexis in the Federal Courts and Administrative Materials databases for any cases found.

4) I performed a search in Lexis in the IP Journal and Periodicals database for any articles on the patent.

5) I performed a search in Lexis in the news databases for any articles about the patent or any articles about litigation on this patent.

Litigation was found

Closed: No

US District Court Civil Docket
U.S. District – New Jersey
(Newark)
2:06cv2469
Net2phone, Inc v. Ebay, Inc et al

Westlaw.

Date of Printing: FEB 18,2009

KEYCITE

C US PAT 6108704 POINT-TO-POINT INTERNET PROTOCOL, Assignee: NetSpeak Corporation (Aug 22, 2000)

History

Direct History

=> 1 **POINT-TO-POINT INTERNET PROTOCOL,** US PAT 6108704, 2000 WL 1193732 (U.S. PTO Utility Aug 22, 2000) (NO. 533115)

Patent Family

2 COMPUTER PROGRAM FOR ENABLING POINT-TO-POINT COMMUNICATION IN COMPUTER NETWORK, ESTABLISHES POINT-TO-POINT COMMUNICATION LINK BETWEEN PROCESSES OVER COMPUTER NETWORK, Derwent World Patents Legal 2000-685834

Assignments

3 ACTION: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS). NUMBER OF PAGES: 034, (DATE RECORDED: Dec 09, 2005)

4 ACTION: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS). NUMBER OF PAGES: 032, (DATE RECORDED: Oct 28, 2005)

5 ACTION: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS). NUMBER OF PAGES: 033, (DATE RECORDED: Oct 28, 2005)

.. Action: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS). Number of Pages: 032, (DATE RECORDED: Sep 12, 2005)

7 ACTION: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS). NUMBER OF PAGES: 004, (DATE RECORDED: Jun 07, 1999)

8 Action: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS). Number of Pages: 004, (DATE RECORDED: Feb 22, 1999)

9 ASSIGNEE(S): INTERNET TELEPHONE COMPANY, (DATE RECORDED: May 30, 1996)

10 ASSIGNEE(S): NETSPEAK CORPORATION, (DATE RECORDED: May 30, 1996)

11 ASSIGNEE(S): INTERNET TELEPHONE COMPANY, (DATE RECORDED: Jan 08, 1996)

Docket Summaries

12 "NET2PHONE, INC. v. EBAY, INC. ET AL", (D.N.J. Jun 01, 2006) (NO. 2:06CV02469), (35 USC 271 PATENT INFRINGEMENT)

Prior Art (Coverage Begins 1976)

C 13 ASYNCHRONOUS TRANSFER MODE COMMUNICATION SYSTEM, US PAT 5452296Assignee: NEC Corporation, (U.S. PTO Utility 1995)

C 14 AUDIO COMMUNICATION SYSTEM FOR A COMPUTER NETWORK, US PAT 5434797 (U.S. PTO Utility 1995)

C 15 AUTOMATIC STATION IDENTIFICATION WHERE FUNCTION MODULES AUTOMATIC-ALLY INITIALIZE, US PAT 5204669Assignee: DataCard Corporation, (U.S. PTO Utility 1993)

C 16 BRIDGE-LIKE INTERNET PROTOCOL ROUTER, US PAT 5309437Assignee: Digital Equipment Corporation, (U.S. PTO Utility 1994)

C 17 COMMUNICATIONS NETWORK DYNAMIC ADDRESSING ARRANGEMENT, US PAT 5166931Assignee: AT&T Bell Laboratories, (U.S. PTO Utility 1992)

C 18 COMMUNICATIONS SYSTEM FOR AN ISDN AND A LAN, AND AN ISDN-LAN CONNECTION TERMINAL, US PAT 5400335Assignee: Ricoh Company, Ltd., (U.S. PTO Utility 1995)

C 19 CONFERENCING OVER MULTIPLE TRANSPORTS, US PAT 5524110Assignee: Intel Corporation, (U.S. PTO Utility 1996)

C 20 EXTENSION OF TWO PHASE COMMIT PROTOCOL TO DISTRIBUTED PARTICIPANTS, US PAT 5546582Assignee: International Business Machines, (U.S. PTO Utility 1996)

C 21 HIGH PERFORMANCE MACHINE FOR SWITCHED COMMUNICATIONS IN A HETEROGENEOUS DATA PROCESSING NETWORK GATEWAY, US PAT 5463625Assignee: International Business Machines, (U.S. PTO Utility 1995)

C 22 HUMAN MACHINE INTERFACE FOR TELEPHONE FEATURE INVOCATION, US PAT 5533110Assignee: Mitel Corporation, (U.S. PTO Utility 1996)

C 23 LINK AND DISCOVERY PROTOCOLS FOR A RING INTERCONNECT ARCHITECTURE, US PAT 5457683Assignee: Apple Computer, Inc., (U.S. PTO Utility 1995)

C 24 MESSAGE ROUTING SYSTEM FOR SHARED COMMUNICATION MEDIA NETWORKS, US PAT 5095480 (U.S. PTO Utility 1992)

C 25 METHOD AND APPARATUS FOR DELIVERING CALLING SERVICES, US PAT 5469500Assignee: Voiceplex Corporation, (U.S. PTO Utility 1995)

C 26 METHOD AND SYSTEM OF MULTICAST ROUTING FOR GROUPS WITH A SINGLE TRANSMITTER, US PAT 5517494Assignee: Apple Computer, Inc., (U.S. PTO Utility 1996)

C 27 METHOD FOR CONFIGURING AND OPERATING A TELECOMMUNICATION APPARATUS, US PAT 5544303Assignee: International Business Machines, (U.S. PTO Utility 1996)

C 28 METHOD FOR POINT-TO-POINT COMMUNICATIONS WITHIN SECURE COMMUNICATION SYSTEMS, US PAT 5357571Assignee: Motorola, Inc., (U.S. PTO Utility 1994)

C 29 METHODS AND APPARATUS FOR ROUTING PACKETS IN PACKET TRANSMISSION NETWORKS, US PAT 5309433Assignee: International Business Machines Corp., (U.S. PTO Utility 1994)

30 MULTI-MEDIA INTEGRATED MESSAGE ARRANGEMENT, US PAT 5479411Assignee: AT&T Corp., (U.S. PTO Utility 1995)

31 MULTIMEDIA SERVER, US PAT 5581552Assignee: AT&T, (U.S. PTO Utility 1996)

32 MULTIPLE PROTOCOL ROUTING, US PAT 5430727Assignee: Digital Equipment Corporation, (U.S. PTO Utility 1995)

33 NETWORK-BASED MULTIMEDIA COMMUNICATIONS AND DIRECTORY SYSTEM AND METHOD OF OPERATION, US PAT 5740231Assignee: Octel Communications Corporation, (U.S. PTO Utility 1998)

34 NETWORK CONTROL SYSTEM AND METHOD, US PAT 5224095Assignee: Johnson Service Company, (U.S. PTO Utility 1993)

35 NETWORK MONITORING METHOD AND APPARATUS, US PAT 5430709Assignee: Hewlett-Packard Company, (U.S. PTO Utility 1995)

36 OBJECT-ORIENTED TELEPHONY SYSTEM, US PAT 5455854Assignee: Taligent, Inc., (U.S. PTO Utility 1995)

37 PROTOCOL SELECTION AND ADDRESS RESOLUTION FOR PROGRAMS RUNNING IN HETEROGENEOUS NETWORKS, US PAT 5425028Assignee: International Business Machines, (U.S. PTO Utility 1995)

38 RECONFIGURABLE, FAULT TOLERANT, MULTISTAGE INTERCONNECT NETWORK AND PROTOCOL, US PAT 5321813Assignee: Teradata Corporation, (U.S. PTO Utility 1994)

39 SCHEME FOR INTERLOCKING LINE CARD TO AN ADDRESS RECOGNITION ENGINE TO SUPPORT PLURALITY OF ROUTING AND BRIDGING PROTOCOLS BY USING NETWORK INFORMATION LOOK-UP DATABASE, US PAT 5524254Assignee: Digital Equipment Corporation, (U.S. PTO Utility 1996)

40 SHARED-PRICE CUSTOM VIDEO RENTALS VIA INTERACTIVE TV, US PAT 5291554Assignee: TV Answer, Inc., (U.S. PTO Utility 1994)

41 SHORTCUT NETWORK LAYER ROUTING FOR MOBILE HOSTS, US PAT 5442633Assignee: International Business Machines, (U.S. PTO Utility 1995)

42 SYSTEM FOR REVERSE ADDRESS RESOLUTION FOR REMOTE NETWORK DEVICE INDEPENDENT OF ITS PHYSICAL ADDRESS, US PAT 5526489Assignee: 3Com Corporation, (U.S. PTO Utility 1996)

43 UNIFIED MESSAGING SYSTEM AND METHOD, US PAT 5608786Assignee: Alphanet Telecom Inc., (U.S. PTO Utility 1997)

44 UTILIZATION OF REDUNDANT LINKS IN BRIDGED NETWORKS, US PAT 5150360Assignee: Digital Equipment Corporation, (U.S. PTO Utility 1992)

US District Court Civil Docket

U.S. District - New Jersey
(Newark)

2:06cv2469

Net2phone, Inc v. Ebay, Inc et al

This case was retrieved from the court on Tuesday, February 17, 2009

Date Filed:	**06/01/2006**	**Class Code:**	**PRETRL, RULE16, SCHEDO**
Assigned To:	**Judge Katharine S Hayden**	**Closed:**	**No**
Referred To:	**Magistrate Judge Patty Shwartz**	**Statute:**	**35:271**
Nature of suit:	**Patent (830)**	**Jury Demand:**	**Both**
Cause:	**Patent Infringement**	**Demand Amount:**	**$0**
Lead Docket:	**None**	**NOS Description:**	**Patent**
Other Docket:	**None**		
Jurisdiction:	**Federal Question**		

Litigants	Attorneys
Howard S Jonas Movant	Joseph P Lasala [COR LD NTC] McElroy, Deutsch, Mulvaney & Carpenter, LLP 1300 Mount Kemble Avenue PO Box 2075 Morristown , NJ 07962-2075 USA (973) 993-8100 Email: Jlasala@mdmc-Law.com
Ronald J Hedges Mediator	Ronald J Hedges [COR LD NTC] Nixon Peabody 437 Madison Avenue New York , NY 10022 USA 212-940-3786 Email: RHEDGES@NIXONPEABODY.COM
Ronald J Hedges Special Master	Ronald J Hedges [COR LD NTC] Nixon Peabody 437 Madison Avenue New York , NY 10022 USA 212-940-3786 Email: RHEDGES@NIXONPEABODY.COM
Net2phone, Inc Plaintiff	Joseph P Lasala [COR LD NTC] McElroy, Deutsch, Mulvaney & Carpenter, LLP 1300 Mount Kemble Avenue

PO Box 2075
Morristown , NJ 07962-2075
USA
(973) 993-8100
Email: Jlasala@mdmc-Law.com

William F O'Connor , Jr
[COR LD NTC]
McElroy, Deutsch Mulvaney & Carpenter, LLP
1300 Mount Kemble Avenue
PO Box 2075
Morristown , NJ 07962-2075
USA
973 993-8100
Email: WOCONNOR@MDMC-Law.com

Maria A Savio
[COR LD NTC]
Gottlieb Rackman & Reisman
270 Madison Avenue
New York , NY 10016
USA
212-684-3900
Email: Msavio@grr.com

Ebay, Inc
Defendant

Kathleen M Fennelly
[COR LD NTC]
Graham, Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: KFENNELLY@GRAHAMCURTIN.COM

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: TCURTIN@GRAHAMCURTIN.COM

George C Jones
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962
USA
(973) 292-1700
Fax: (973) 292-1767
Email: GJONES@GRAHAMCURTIN.COM

Skype Technologies SA
Defendant

Kathleen M Fennelly
[COR LD NTC]
Graham, Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA

(973) 292-1700
Fax: (973) 292-1767
Email: KFENNELLY@GRAHAMCURTIN.COM

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: TCURTIN@GRAHAMCURTIN.COM

George C Jones
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962
USA
(973) 292-1700
Fax: (973) 292-1767
Email: GJONES@GRAHAMCURTIN.COM

Skype, Inc
Defendant

Kathleen M Fennelly
[COR LD NTC]
Graham, Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: KFENNELLY@GRAHAMCURTIN.COM

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: TCURTIN@GRAHAMCURTIN.COM

George C Jones
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962
USA
(973) 292-1700
Fax: (973) 292-1767
Email: GJONES@GRAHAMCURTIN.COM

John Does 1-10
Defendant

Ebay, Inc
Counter Claimant

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA

	4 Headquarters Plaza PO Box 1991 Morristown , NJ 07962-1991 USA (973) 292-1700 Fax: (973) 292-1767 Email: TCURTIN@GRAHAMCURTIN.COM
	Kathleen M Fennelly [COR LD NTC] Graham, Curtin, PA 4 Headquarters Plaza PO Box 1991 Morristown , NJ 07962-1991 USA (973) 292-1700 Fax: (973) 292-1767 Email: KFENNELLY@GRAHAMCURTIN.COM
Skype Technologies SA Counter Claimant	Thomas R Curtin [COR LD NTC] Graham Curtin, PA 4 Headquarters Plaza PO Box 1991 Morristown , NJ 07962-1991 USA (973) 292-1700 Fax: (973) 292-1767 Email: TCURTIN@GRAHAMCURTIN.COM
	Kathleen M Fennelly [COR LD NTC] Graham, Curtin, PA 4 Headquarters Plaza PO Box 1991 Morristown , NJ 07962-1991 USA (973) 292-1700 Fax: (973) 292-1767 Email: KFENNELLY@GRAHAMCURTIN.COM
Skype, Inc Counter Claimant	Thomas R Curtin [COR LD NTC] Graham Curtin, PA 4 Headquarters Plaza PO Box 1991 Morristown , NJ 07962-1991 USA (973) 292-1700 Fax: (973) 292-1767 Email: TCURTIN@GRAHAMCURTIN.COM
	Kathleen M Fennelly [COR LD NTC] Graham, Curtin, PA 4 Headquarters Plaza PO Box 1991 Morristown , NJ 07962-1991 USA (973) 292-1700 Fax: (973) 292-1767 Email: KFENNELLY@GRAHAMCURTIN.COM
Net2phone, Inc	Joseph P Lasala

Counter Defendant

[COR LD NTC]
McElroy, Deutsch, Mulvaney & Carpenter, LLP
1300 Mount Kemble Avenue
PO Box 2075
Morristown , NJ 07962-2075
USA
(973) 993-8100
Email: Jlasala@mdmc-Law.com

Ebay, Inc
Counter Claimant

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: TCURTIN@GRAHAMCURTIN.COM

Kathleen M Fennelly
[COR LD NTC]
Graham, Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: KFENNELLY@GRAHAMCURTIN.COM

Skype Technologies SA
Counter Claimant

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: TCURTIN@GRAHAMCURTIN.COM

Kathleen M Fennelly
[COR LD NTC]
Graham, Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: KFENNELLY@GRAHAMCURTIN.COM

Skype, Inc
Counter Claimant

Thomas R Curtin
[COR LD NTC]
Graham Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: TCURTIN@GRAHAMCURTIN.COM

Kathleen M Fennelly
[COR LD NTC]
Graham, Curtin, PA
4 Headquarters Plaza
PO Box 1991
Morristown , NJ 07962-1991
USA
(973) 292-1700
Fax: (973) 292-1767
Email: KFENNELLY@GRAHAMCURTIN.COM

Net2phone, Inc
Counter Defendant

Date	#	Proceeding Text
06/01/2006	1	COMPLAINT against EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC., JOHN DOES 1-10 (Filing fee $ 350 receipt number 987250.) JURY DEMAND, filed by NET2PHONE, INC.. (Attachments: # 1 Exhibit A# 2 7.1)(lm2,) (Entered: 06/02/2006)
06/02/2006	--	Summons Issued as to SKYPE TECHNOLOGIES SA, SKYPE, INC..Days Due - 20. (counsel picked up 6/2/06) (lm2,) (Entered: 06/02/2006)
06/02/2006	--	Summons Issued as to EBAY, INC..Days Due - 20. (counsel picked up 6/2/06) (lm2,) (Entered: 06/02/2006)
06/07/2006	2	AMENDED COMPLAINT against all defendants all defendants., filed by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 06/07/2006)
06/09/2006	3	AMENDED COMPLAINT against EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC., JOHN DOES 1-10, filed by NET2PHONE, INC.. (Attachments: # 1 Exhibit A to amended complaint) (LASALA, JOSEPH) (Entered: 06/09/2006)
06/21/2006	4	MOTION for Leave to Appear Pro Hac Vice of Allen Rubenstein and Steven Stern by NET2PHONE, INC.. (Attachments: # 1 Affidavit of Joseph P. LaSala# 2 Affidavit of Steven Stern# 3 Affidavit of Allen Rubenstein# 4 Text of Proposed Order # 5 Certificate of Service)(LASALA, JOSEPH) (Entered: 06/21/2006)
06/21/2006	--	Set Deadlines as to 4 MOTION for Leave to Appear Pro Hac Vice of Allen Rubenstein and Steven Stern. Motion Hearing set for 7/24/2006 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 06/22/2006)
06/26/2006	5	NOTICE of Appearance by MARIA A. SAVIO on behalf of all plaintiffs (SAVIO, MARIA) (Entered: 06/26/2006)
06/26/2006	6	ORDER granting 4 Motion for Allen I. Rubenstein & Steven Stern to Appear Pro Hac Vice . Signed by Judge S. D. Wigenton on 06/22/06. (nr,) Modified on 6/27/2006 (nr,). (Entered: 06/27/2006)
06/28/2006	7	AMENDED COMPLAINT against all defendants all defendants., filed by NET2PHONE, INC.. (Attachments: # 1)(LASALA, JOSEPH) (Entered: 06/28/2006)
07/11/2006	8	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit of J. LaSala# 2 Affidavit of J. Alan Galbraith# 3 Affidavit of B. Sullivan# 4 Affidavit of M. Stern# 5 Text of Proposed Order # 6 Certificate of Service)(LASALA, JOSEPH) (Entered: 07/11/2006)
07/11/2006	--	Set Deadlines as to 8 MOTION for Leave to Appear Pro Hac Vice. Motion Hearing set for 9/11/2006 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 07/27/2006)
07/25/2006	9	AFFIDAVIT of Service for Summons and Second Amended Complaint served on Carla McCreight on behalf of Ebay on 7/13/06, filed by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 07/25/2006)
07/25/2006	10	AFFIDAVIT of Service for Summons, Complaint, Exhibit and First Amended Complaint served on Skype, Inc. on 6/12/06, filed by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 07/25/2006)
07/25/2006	11	AFFIDAVIT of Service for Summons, Complaint, Exhibit, First Amended Complaint served on Ebay on 6/12/06, filed by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 07/25/2006)
07/25/2006	12	AFFIDAVIT of Service for Summons, Second Amended Complaint served on Carla McCreight on

		behalf of Skype, Inc. on 7/13/06, filed by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 07/25/2006)
08/08/2006	13	ORDER granting 8 Motion for Brendan V. Sullivan, J. Alan Galbraith and Michael K. Stern to Appear Pro Hac Vice . Signed by Judge Patty Shwartz on 08/08/06. (nr,) (Entered: 08/08/2006)
08/15/2006	14	ORDER directing plainhtiff to move for default and default judgment by September 4, 2006. Signed by Judge Katharine S. Hayden on 8/15/06. (RG,) (Entered: 08/16/2006)
08/17/2006	15	NOTICE of Appearance by THOMAS R. CURTIN on behalf of EBAY, INC., SKYPE, INC. (CURTIN, THOMAS) (Entered: 08/17/2006)
08/17/2006	16	STIPULATION AND ORDER extending deft's time to move or otherwise respond to second amended complt.. Signed by Judge Madeline C. Arleo on 08/14/06. (nr,) (Entered: 08/18/2006)
09/15/2006	17	ANSWER to Amended Complaint, COUNTERCLAIM against all plaintiffs by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Statement Rule 7.1 Disclosure Statement# 2 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 09/15/2006)
09/20/2006	18	ANSWER to Counterclaim by NET2PHONE, INC.. (Attachments: # 1)(LASALA, JOSEPH) (Entered: 09/20/2006)
09/22/2006	19	Notice of Request by Pro Hac Vice to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 09/22/2006)
09/22/2006	20	Notice of Request by Pro Hac Vice to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 09/22/2006)
09/22/2006	21	Notice of Request by Pro Hac Vice to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 09/22/2006)
09/22/2006	22	Notice of Request by Pro Hac Vice to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 09/22/2006)
09/22/2006	23	Notice of Request by Pro Hac Vice to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 09/22/2006)
10/06/2006	24	ORDER granting application for pro hac vice admission of Andrei Iancu, Morgan Chu, Ted M. Sichelman and Michelle E. Armond . Signed by Judge Patty Shwartz on 10/03/06. (nr,) (Entered: 10/10/2006)
10/11/2006	25	SCHEDULING ORDER setting Scheduling Conference for 10/27/2006 11:30 AM in Newark - Courtroom 10 before Magistrate Judge Patty Shwartz. Signed by Judge Patty Shwartz on 10/11/06. (aa,) (Entered: 10/11/2006)
10/12/2006	--	Pro Hac Vice fee: $ 600.00, receipt number 200341554 re Andrei Iancu, Morgan Chu, Sichel Man, Michelle Armond (nr,) (Entered: 10/13/2006)
10/16/2006	26	Notice of Request by Pro Hac Vice Andrei Iancu, Esq. to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 10/16/2006)
10/16/2006	27	Notice of Request by Pro Hac Vice Morgan Chu, Esq. to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 10/16/2006)
10/16/2006	28	Notice of Request by Pro Hac Vice Michelle E. Armond, Esq. to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 10/16/2006)
10/16/2006	29	Notice of Request by Pro Hac Vice Ted M. Sichelman, Esq. to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 10/16/2006)
10/27/2006	--	Minute Entry for proceedings held before Judge Patty Shwartz : Scheduling Conference held on 10/27/2006. (aa,) (Entered: 10/30/2006)
10/30/2006	30	PRETRIAL SCHEDULING ORDER: Settlement Conference set for 4/12/2007 01:00 PM before Magistrate Judge Patty Shwartz. Telephone Conference set for 12/4/2006 04:00 PM before Magistrate Judge Patty Shwartz. Final Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz. Discovery due by 12/31/2007. Proposed Pretrial Order due by 6/10/2008.. Signed by Judge Patty Shwartz on 10/27/2006. (nr,) (Entered: 10/31/2006)
11/02/2006	31	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 11/02/2006)
11/03/2006	32	CORRECTED SCHEDULING ORDER: Settlement Conference set for 4/12/2007 01:00 PM before Magistrate Judge Patty Shwartz. Telephone Conference set for 12/4/2006 04:00 PM before Magistrate Judge Patty Shwartz. Final Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz. Amended Pleadings due by 3/8/2007.. Signed by Judge Patty Shwartz on 11/2/2006. (mn,) (Entered: 11/03/2006)

11/03/2006	33	Notice of Request by Pro Hac Vice Andrei Iancu, Esq. to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 11/03/2006)
11/20/2006	34	Letter from Joseph P. LaSala, Esq. enclosing Plaintiff's Preliminary Identification of Allegedly Infringing Products List and Certification of service. (LASALA, JOSEPH) (Entered: 11/20/2006)
11/30/2006	35	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 11/30/2006)
12/01/2006	36	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 12/01/2006)
12/04/2006	--	Minute Entry for proceedings held before Judge Patty Shwartz : Telephone Conference held on 12/4/2006. (aa,) (Entered: 12/08/2006)
12/08/2006	37	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 12/08/2006)
12/11/2006	38	Letter from Joseph P. La Sala, Esq. in lieu of formal motion regarding form of protective order re Telephone Conference. (Attachments: # 1 Exhibit 1# 2 Exhibit 2# 3 Exhibit 3# 4 Certificate of Service)(LASALA, JOSEPH) (Entered: 12/11/2006)
01/04/2007	39	DISCOVERY CONFIDENTIALITY ORDER ON INFORMAL APPLICATION . Signed by Judge Patty Shwartz on 12/29/06. (dc,) (Entered: 01/04/2007)
01/17/2007	40	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 01/17/2007)
01/17/2007	41	ORDER on informal application denying request to impose a patent prosecution bar. Signed by Judge Patty Shwartz on 1/12/2007. (mn,) (Entered: 01/17/2007)
01/19/2007	42	ORDER ON INFORMAL APPL. that the issue raised in the 1/16/07 letter is deemed resolved by the Order dated 1/12/07.. Signed by Judge Patty Shwartz on 1/17/07. (DD,) (Entered: 01/19/2007)
01/22/2007	43	TRANSCRIPT of Proceedings held on December 29, 2006 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 01/23/2007)
01/30/2007	44	ORDER on informal application mooting the need to file a response to the letter seeking reconsideration of the Discovery Confidentiality order . Signed by Judge Patty Shwartz on 01/30/2007. (nr,) (Entered: 01/31/2007)
02/05/2007	45	NOTICE by TED M. SICHELMAN, MICHELLE E. ARMOND, EBAY, INC., EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC. re 24 Order NOTICE OF WITHDRAWAL OF ADMISSIONS PRO HAC VICE OF TED M. SICHELMAN, ESQ., AND MICHELLE E. ARMOND, ESQ. (FENNELLY, KATHLEEN) (Entered: 02/05/2007)
02/06/2007	46	ORDER ON INFORMAL APPLICATION granting pro hac vice admission of counsel etc. Signed by Judge Patty Shwartz on 2/6/07. (cs,) (Entered: 02/09/2007)
02/06/2007	47	DECLARATION of Kathleen N. Fennelly, Esq.in support of pro hac vice admission on behalf of eBay, Inc and Skype, Inc. (cs,) (Entered: 02/09/2007)
02/06/2007	48	DECLARATION of Alan J. Heinrich re admission pro hac vice on behalf of dfts., e-Bay Inc. and Skype, Inc. (cs,) (Entered: 02/09/2007)
02/06/2007	49	DECLARATION of Eric Vandevelde re admission pro hac vice on behalf of dfts., EBAY, INC., SKYPE, INC.. (cs,) (Entered: 02/09/2007)
02/06/2007	50	DECLARATION of Andrew D. Weiss re admission pro hac vice on behalf of dfts., EBAY, INC., SKYPE, INC.. (cs,) (Entered: 02/09/2007)
02/27/2007	--	Minute Entry for proceedings held before Judge Patty Shwartz : Telephone Conference held on 2/27/2007. (aa,) (Entered: 03/05/2007)
02/28/2007	51	ORDER on informal application granting request to extend the deadline to commence foreign evidence collection and to file motions to amend the pleadings, SCHEDULING ORDER: Settlement Conference set for 4/12/2007 01:00 PM before Magistrate Judge Patty Shwartz. Telephone Conference set for 6/18/2007 03:00 PM before Magistrate Judge Patty Shwartz. Final Pretrial Conference set for 6/17/2007 01:00 PM before Magistrate Judge Patty Shwartz. Discovery due by 12/31/2007.. Signed by Judge Patty Shwartz on 02/27/2007. (nr,) (Entered: 03/01/2007)
03/02/2007	--	Pro Hac Vice fee: $ 450.00, receipt number 200344847 alan J. Heinrich, Andrew D. Weiss and Eric Vandevelde (nr,) (Entered: 03/05/2007)
03/05/2007	52	ORDER on informal application granting request to extend the deadline to submit a porposed confidentiality order and clarify the timing for serving interrogatories; SCHEDULING ORDER: Settlement Conference set for 4/12/2007 01:00 PM before Magistrate Judge Patty Shwartz. Telephone Conference set for 2/27/2007 03:00 PM before Magistrate Judge Patty Shwartz. Final

		Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz. Discovery due by 12/31/2007. Proposed Pretrial Order due by 6/10/2008.. Signed by Judge Patty Shwartz on 12/04/2006. (nr,) (Entered: 03/05/2007)
03/08/2007	53	Notice of Request by Pro Hac Vice Alan J. Heinrich to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 03/08/2007)
03/08/2007	54	Notice of Request by Pro Hac Vice Andrew D. Weiss to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 03/08/2007)
03/08/2007	55	Notice of Request by Pro Hac Vice Eric Vandevelde to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 03/08/2007)
03/14/2007	56	ORDER on Informal application that defts' request to bar Professor Bhattacharjee from being designated as an expert witness in this case is denied w/out prejudice, etc.. Signed by Judge Patty Shwartz on 3/14/07. (jd,) (Entered: 03/14/2007)
03/26/2007	57	SCHEDULING LETTER ORDER: Settlement Conference set for 6/18/2007 10:00 AM before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 03/26/2007. (nr,) (Entered: 03/27/2007)
03/28/2007	58	Letter from Thomas R. Curtin, Esq.. (CURTIN, THOMAS) (Entered: 03/28/2007)
03/29/2007	59	ORDER denying pltf's request to extend certain deadlines. Signed by Judge Patty Shwartz on 03/29/2007. (nr,) (Entered: 03/30/2007)
03/30/2007	60	NOTICE by NET2PHONE, INC. of Claims Identification (Attachments: # 1)(LASALA, JOSEPH) (Entered: 03/30/2007)
04/10/2007	61	Order on informal application granting request to extend pretrial deadlines; & THIRD AMENDED SCHEDULING ORDER: Settlement Conference set for 6/18/2007 10:00 AM before Magistrate Judge Patty Shwartz., Telephone Conference set for 9/25/2007 03:00 PM before Magistrate Judge Patty Shwartz., Final Pretrial Conference set for 6/17/2007 01:00 PM before Magistrate Judge Patty Shwartz.,Discovery due by 12/31/2007.. Signed by Judge Patty Shwartz on 04/09/2007. (nr,) (Entered: 04/11/2007)
04/13/2007	62	MOTION for Leave to Appear Pro Hac Vice on Behalf of Bruce R. Genderson, Esq., Nicholas J. Boyle, Esq., Kevin Hardy, Esq. and Hannah M. Stott-Bumsted, Esq. by NET2PHONE, INC.. (Attachments: # 1 Affidavit of Joseph P. La Sala, Esq.# 2 Affidavit of Kevin Hardy, Esq.# 3 Affidavit of Nicholas J. Boyle, Esq.# 4 Affidavit of Hannah M. Stott-Bumsted, Esq.# 5 Affidavit of Bruce R. Genderson, Esq.# 6 Text of Proposed Order # 7 Certificate of Service)(LASALA, JOSEPH) (Entered: 04/13/2007)
04/13/2007	63	ORDER on informal application granting the parties request to extend deadline to raise disputes regarding the designation of Dr. Bhattacharjee as an expert. Signed by Judge Patty Shwartz on 04/12/2007. (nr,) (Entered: 04/16/2007)
04/16/2007	64	ORDER granting 62 Motion for Bruce R. Genderson, Nicholas J. Boyle, Kevin Hardy, and Hannah M. Stott-Bumsted to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 04/16/2007. (nr,) (Entered: 04/17/2007)
04/18/2007	65	Notice of Request by Pro Hac Vice Kevin Hardy, Esq. to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 04/18/2007)
04/18/2007	66	Notice of Request by Pro Hac Vice Hannah M. Stott-Bumsted, Esq. to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 04/18/2007)
04/18/2007	67	Notice of Request by Pro Hac Vice Nicholas J. Boyle, Esq. to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 04/18/2007)
04/18/2007	68	Notice of Request by Pro Hac Vice Bruce Genderson, Esq. to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 04/18/2007)
04/19/2007	--	Pro Hac Vice fee: $ 600.00, receipt number 1441831,1441843,1441852,1441856 re Kevin Hardy, Hannah M. Scott Brumsted, Nicholas J. Boyle, Bruce Genderson (nr,) (Entered: 04/19/2007)
04/20/2007	69	MOTION to Withdraw Pro Hac Vice Admission of Michael K. Stern, Esq. by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 04/20/2007)
04/20/2007	70	First MOTION for Issuance of Letters Rogatory by NET2PHONE, INC.. (Attachments: # 1 Exhibit A - Part 1# 2 Exhibit A - Part 2# 3 Exhibit B - Part 1# 4 Exhibit B - Part 2# 5 Exhibit C - Part 1# 6 Exhibit C - Part 2# 7 Exhibit D - Part 1# 8 Exhibit D - Part 2# 9 Exhibit E - Part 1# 10 Exhibit E - Part 2# 11 Exhibit F - Part 1# 12 Exhibit F - Part 2# 13 Text of Proposed Order # 14 Certificate of Service)(LASALA, JOSEPH) (Entered: 04/20/2007)
04/20/2007	--	Set Deadlines as to 70 First MOTION for Issuance of Letters Rogatory. Motion Hearing set for

		5/28/2007 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 04/24/2007)
04/24/2007	--	Pro Hac Vice fee: $ 600, receipt number 200344948 re Bruce R. Genderson, Nicholas J. Boyle, Hannah M. Stott-Bumsted, Kevin Hardy (nr,) (Entered: 04/24/2007)
04/27/2007	71	Letter from Joseph La Sala re 70 First MOTION for Issuance of Letters Rogatory. (LASALA, JOSEPH) (Entered: 04/27/2007)
04/27/2007	72	Amended MOTION for Issuance of Letters Rogatory by NET2PHONE, INC.. (Attachments: # 1 Exhibit A - part 1# 2 Exhibit A - part 2# 3 Exhibit B# 4 Exhibit C# 5 Exhibit D# 6 Exhibit E# 7 Exhibit F# 8 Text of Proposed Order # 9 Certificate of Service)(LASALA, JOSEPH) (Entered: 04/27/2007)
04/27/2007	--	Set Deadlines as to 72 Amended MOTION for Issuance of Letters Rogatory. Motion Hearing set for 5/28/2007 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 04/30/2007)
04/30/2007	73	NOTICE by NET2PHONE, INC. of Amended Identification of Infringing Products (Attachments: # 1)(LASALA, JOSEPH) (Entered: 04/30/2007)
05/01/2007	75	ORDER withdrawing pltf's request for issuance of letters rogatory (Docket No. 71). Signed by Judge Patty Shwartz on 04/27/2007. (nr,) (Entered: 05/02/2007)
05/02/2007	74	Notice of Request by Pro Hac Vice Steven Stern, Esq. to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 05/02/2007)
05/07/2007	76	ORDER overruling dfts' objection to the production of responsive discovery based upon a parivate agreement that contains a confidentiality clause. Any and all such responsive information shall be produced no later than 5/26/07 subject to the Discovery Confidentiality Order. Signed by Judge Patty Shwartz on 5/7/07. (cs,) (Entered: 05/07/2007)
05/07/2007	77	MOTION for Leave to File Third Amended Complaint by NET2PHONE, INC.. (Attachments: # 1 Exhibit 1-Third Amended Complaint# 2 Text of Proposed Order # 3 Certificate of Service) (LASALA, JOSEPH) (Entered: 05/07/2007)
05/07/2007	78	ORDER granting 72 Motion for Issuance of Letters Rogatory. Signed by Judge Patty Shwartz on 05/07/20047. (nr,) (Entered: 05/09/2007)
05/07/2007	--	Set Deadlines as to 77 MOTION for Leave to File Third Amended Complaint. Motion Hearing set for 6/11/2007 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 05/09/2007)
05/07/2007	--	Minute Entry for proceedings held before Judge Patty Shwartz : Telephone Conference held on 5/7/2007. (aa,) (Entered: 06/01/2007)
05/11/2007	79	Third MOTION for Issuance of Letters Rogatory by NET2PHONE, INC.. (Attachments: # 1 Exhibit A# 2 Exhibit B# 3 Exhibit C# 4 Exhibit D# 5 Exhibit E# 6 Exhibit F# 7 Text of Proposed Order # 8 Certificate of Service)(LASALA, JOSEPH) (Entered: 05/11/2007)
05/11/2007	--	Set Deadlines as to 79 Third MOTION for Issuance of Letters Rogatory. Motion Hearing set for 6/11/2007 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 05/15/2007)
05/14/2007	80	TRANSCRIPT of Proceedings held on May 7, 2007 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 05/15/2007)
05/21/2007	81	ORDER granting 79 Motion for Issuance of Letters Rogatory. Signed by Judge Patty Shwartz on 05/18/2007. (nr,) (Entered: 05/21/2007)
05/22/2007	--	Letters Rogatory issued re 81 Order on Motion for Issuance of Letters Rogatory. (mn,) (Entered: 05/22/2007)
05/29/2007	82	Letter from Thomas R. Curtin, Esq., re 77 MOTION for Leave to File Third Amended Complaint. (CURTIN, THOMAS) (Entered: 05/29/2007)
05/30/2007	83	MOTION for Leave to Appear Pro Hac Vice on behalf of Michael D. Hurwitz, Esq. by NET2PHONE, INC.. (Attachments: # 1 Affidavit of Jospeh P. La Sala# 2 Affidavit of Michael D. Hurwitz# 3 Text of Proposed Order # 4 Certificate of Service)(LASALA, JOSEPH) (Entered: 05/30/2007)
05/30/2007	85	ORDER on informal application granting the request to extend deadlines concerning the

		invalidity disclosures and infringement contentions to address the newly asserted patent; FOURTH AMENDED PRETRIAL SCHEDULING ORDER: Settlement Conference set for 6/18/2007 10:00 PM before Magistrate Judge Patty Shwartz., Telephone Conference set for 9/25/2007 03:00 PM before Magistrate Judge Patty Shwartz., Final Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz.,Discovery due by 12/31/2007.. Signed by Judge Patty Shwartz on 05/30/2007. (nr,) (Entered: 05/31/2007)
05/30/2007	--	Set Deadlines as to 83 MOTION for Leave to Appear Pro Hac Vice on behalf of Michael D. Hurwitz, Esq.. Motion Hearing set for 6/25/2007 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 05/31/2007)
05/31/2007	84	NOTICE by NET2PHONE, INC. of Withdrawal of Admission Pro Hac Vice of J. Alan Galbraith, Esq. (LASALA, JOSEPH) (Entered: 05/31/2007)
05/31/2007	86	ORDER granting 77 Motion for Leave to File third amended complt.. Signed by Judge Patty Shwartz on 05/30/2007. (nr,) (Entered: 05/31/2007)
05/31/2007	87	ORDER granting 83 Motion for Michael D. Hurwitz to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 05/31/2007. (nr,) (Entered: 06/01/2007)
06/04/2007	88	AMENDED COMPLAINT against all defendants all defendants., filed by NET2PHONE, INC.. (Attachments: # 1)(LASALA, JOSEPH) (Entered: 06/04/2007)
06/12/2007	89	Notice of Request by Pro Hac Vice Michael D. Hurwitz, Esq. to receive Notices of Electronic Filings. (LASALA, JOSEPH) (Entered: 06/12/2007)
06/12/2007	--	Pro Hac Vice fee: $ 150, receipt number 200345674 re Michael Hurwitz (nr,) (Entered: 06/12/2007)
06/22/2007	90	LETTER ORDER: resetting Settlement Conference set for 9/6/2007 11:00 AM before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 06/22/2007. (nr,) (Entered: 06/25/2007)
06/25/2007	91	ANSWER to Amended Complaint, COUNTERCLAIM against all plaintiffs by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Certificate of Service)(CURTIN, THOMAS) (Entered: 06/25/2007)
06/25/2007	92	Letter from Joseph P. La Sala, Esq. Regarding Joint Request to Change Scheduling Order. (LASALA, JOSEPH) (Entered: 06/25/2007)
06/25/2007	93	CONSENT ORDER extending defts' time to answer to 06/25/2007. Signed by Judge Patty Shwartz on 06/22/2007. (nr,) (Entered: 06/26/2007)
06/25/2007	94	ORDER on informal application granting request to extend deadline to raise unresolved discovery, FIFTH AMENDED PRETRIAL SCHEDULING ORDER: Settlement Conference set for 9/6/2007 11:30 AM before Magistrate Judge Patty Shwartz., Telephone Conference set for 9/25/2007 03:00 PM, 12/4/2007 AT 3:00P.M. & 4/29/2008 AT 3:00P.M. before Magistrate Judge Patty Shwartz., Final Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz.,Discovery due by 12/31/2007.. Signed by Judge Patty Shwartz on 06/24/2007. (nr,) (Entered: 06/26/2007)
06/27/2007	95	LETTER ORDER: Settlement Conference set for 9/6/2007 11:00 AM before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 06/26/2007. (nr,) (Entered: 06/27/2007)
08/06/2007	96	ORDER on informal application granting the request to extend deadline to submit Markman briefs; SCHEDULING ORDER: Settlement Conference set for 9/6/2007 11:30 AM before Magistrate Judge Patty Shwartz., Telephone Conference set for 9/25/2007 03:00 PM before Magistrate Judge Patty Shwartz., Final Pretrial Conference set for 6/17/2007 01:00 PM before Magistrate Judge Patty Shwartz.,Discovery due by 12/31/2007.. Signed by Judge Patty Shwartz on 08/06/2007. (nr,) (Entered: 08/07/2007)
08/30/2007	97	Platiniff's Net2Phone, Inc's Opening Claim Construction MEMORANDUM by NET2PHONE, INC.. (Attachments: # 1 Declaration of Kevin Hardy# 2 Exhibit 1# 3 Exhibit 2# 4 Exhibit 3# 5 Exhibit 4# 6 Exhibit 5# 7 Exhibit 6# 8 Exhibit 7# 9 Exhibit 8# 10 Exhibit 9# 11 Exhibit 10# 12 Exhibit 11# 13 Exhibit 12# 14 Exhibit 13# 15 Exhibit 14# 16 Exhibit 15# 17 Exhibit 16# 18 Exhibit 17# 19 Certificate of Service)(LASALA, JOSEPH) Modified on 10/11/2007 (rg,). (Entered: 08/30/2007)
08/30/2007	98	Declaration of Alan J. Heinrich in support of Opening Claim Construction MEMORANDUM of SKYBE Tech, SKYPE, Inc. and EBAY by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit A# 2 Exhibit B# 3 Exhibit C# 4 Exhibit D# 5 Exhibit E# 6 Exhibit F# 7 Exhibit G# 8 Exhibit H# 9 Exhibit I# 10 Exhibit J# 11 Exhibit K# 12 Exhibit L# 13 Exhibit M# 14 Exhibit N# 15 Exhibit O# 16 Exhibit P# 17 Exhibit Q# 18 Exhibit R# 19 Exhibit S# 20 Exhibit T# 21 Exhibit U# 22 Exhibit V# 23 Exhibit W# 24 Brief Skype's Opening Claim Construction

		Brief# 25 Appendix A# 26 Appendix B# 27 Certificate of Service)(CURTIN, THOMAS) Modified on 10/11/2007 (rg,). (Entered: 08/30/2007)
09/07/2007	99	LETTER ORDER rescheduling Settlement Conference set for 10/29/2007 11:00 AM before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 09/07/2007. (nr,) (Entered: 09/10/2007)
09/11/2007	100	ORDER on informal application granting request that defts. produce hardware, compilers and codes needed to establish a "test". Signed by Judge Patty Shwartz on 09/11/2007. (nr,) (Entered: 09/12/2007)
09/17/2007	101	TRANSCRIPT of Proceedings held on September 11,2 007 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 09/18/2007)
09/20/2007	102	Letter from Thomas R. Curtin, Esq.. (FENNELLY, KATHLEEN) (Entered: 09/20/2007)
09/21/2007	103	Letter from Joseph P. La Sala, Esq. regarding discovery disputes. (LASALA, JOSEPH) (Entered: 09/21/2007)
09/24/2007	--	CLERK'S QUALITY CONTROL MESSAGE: ERIC VANDEVELDE, does not have a correct e-mail address listed with the court and is not receiving his/her notices of electronic filing in this case. Pursuant to local rule 10.1 and court procedures, counsel and unrepresented parties are required to notify the court of any mailing or e-mail address changes. The court has deleted the invalid e-mail address. Attorneys should review the ECF link on our web site for information on maintaining your account and unrepresented parties, or those attorneys without access to maintaining their account, should notice the Clerk. (mem,) (Entered: 09/24/2007)
09/25/2007	--	Text Minute Entry for proceedings held before Judge Patty Shwartz : Telephone Conference held on 9/25/2007. (aa,) (Entered: 09/28/2007)
09/27/2007	104	ORDER on informal application directing the pltf. to submit no later than Oct. 15, 2007 the nonprivileged documents referred to in the Sept. 20, 2007 submissions. Signed by Judge Patty Shwartz on 09/25/2007. (nr,) (Entered: 09/28/2007)
10/02/2007	105	NOTICE by ERIC VANDEVELDE, EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC. re 55 Notice of Pro Hac Vice to Receive NEF Withdrawl of Pro Hac Vice Admission and Request for Electronic Notification (FENNELLY, KATHLEEN) (Entered: 10/02/2007)
10/04/2007	106	Plaintiff Net2Phone Inc's Response MEMORANDUM on Claim Construction by NET2PHONE, INC.. (Attachments: # 1 Declaration of Kevin Hard (Second)# 2 Exhibit 18# 3 Exhibit 19# 4 Exhibit 20# 5 Exhibit 21# 6 Exhibit 22# 7 Exhibit 23# 8 Exhibit 24# 9 Exhibit 25# 10 Exhibit 26# 11 Exhibit 27# 12 Exhibit 28# 13 Exhibit 29# 14 Certificate of Service)(LASALA, JOSEPH) Modified on 10/11/2007 (rg,). (Entered: 10/04/2007)
10/04/2007	107	Responsive claim Construction MEMORANDUM of SKYPE Tech, SKYPE and EBAY by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of David Johnson# 2 Exhibit 1# 3 Exhibit 2# 4 Declaration of Alan Heinrich# 5 Exhibit A# 6 Exhibit A part 2# 7 Exhibit B# 8 Exhibit C# 9 Exhibit D# 10 Exhibit E# 11 Exhibit F# 12 Certificate of Service) (CURTIN, THOMAS) Modified on 10/11/2007 (rg,). (Entered: 10/04/2007)
10/11/2007	108	Minute Entry for proceedings held before Judge Katharine S. Hayden : Status Conference held on 10/11/2007. (rg,) (Entered: 10/12/2007)
10/11/2007	111	AMENDED Minute Entry for proceedings held before Judge Katharine S. Hayden : Status Conference held on 10/11/2007. (rg,) Additional attachment(s) added on 10/19/2007 (rg,). (Entered: 10/18/2007)
10/15/2007	109	Letter from Joseph P. La Sala, Esq.. (LASALA, JOSEPH) (Entered: 10/15/2007)
10/15/2007	110	ORDER on informal application directing he parties o produce the supplemental responses to the document demands to include documents that came into existence between April 1, 2007 and Aug. 1, 2007. Signed by Judge Patty Shwartz on 10/13/2007. (nr,) (Entered: 10/16/2007)
10/18/2007	112	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 97 Pretrial Memorandum, Supplemental Memorandum Relating to Entry 97. (Attachments: # 1 Certificate of Service for Supplemental Memorandum Relating to Entry 97)(LASALA, JOSEPH) (Entered: 10/18/2007)
10/18/2007	113	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 106 Pretrial Memorandum, Supplemental Memorandum Relating to Entry 106. (Attachments: # 1 Certificate of Service for Supplemental Memorandum Relating to Entry 106)(LASALA, JOSEPH) (Entered: 10/18/2007)
10/18/2007	114	AMENDED DOCUMENT by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. Amendment to 98 Pretrial Memorandum,, Supplemental Memorandum Relating to Entry 98, Attachment 24. (Attachments: # 1 Appendix A to Reformatted Opening Brief# 2 Appendix B to Reformatted

		Opening Brief# 3 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/18/2007)
10/18/2007	115	AMENDED DOCUMENT by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. Amendment to 107 Pretrial Memorandum, Supplemental Memorandum Relating to Entry 107. (Attachments: # 1 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/18/2007)
10/19/2007	116	PRETRIAL MEMORANDUM by NET2PHONE, INC.. (Attachments: # 1 Declaration of Kevin Hardy (Third)# 2 Exhibit 30 to Third Declaration of Kevin Hardy# 3 Exhibit 31 to Third Declaration of Kevin Hardy# 4 Exhibit 32 to Third Declaration of Kevin Hardy# 5 Exhibit 33 to Third Declaration of Kevin Hardy# 6 Declaration of Professor Larry L. Peterson# 7 Exhibit 1 to Peterson Declaration# 8 Exhibit 2 to Peterson Declaration# 9 Exhibit 3 to Peterson Declaration# 10 Exhibit 4 to Peterson Declaration# 11 Exhibit 5 to Peterson Declaration# 12 Exhibit 6 to Peterson Declaration# 13 Exhibit 7 to Peterson Declaration# 14 Exhibit 8 to Peterson Declaration# 15 Exhibit 9 to Peterson Declaration# 16 Exhibit 10 to Peterson Declaration# 17 Exhibit 11 to Peterson Declaration# 18 Exhibit 12 to Peterson Declaration# 19 Exhibit 13 to Peterson Declaration# 20 Exhibit 14 to Peterson Declaration# 21 Exhibit 15 to Peterson Declaration# 22 Exhibit 16 to Peterson Declaration# 23 Exhibit 17 to Peterson Declaration# 24 Certificate of Service)(LASALA, JOSEPH) (Entered: 10/19/2007)
10/19/2007	117	PRETRIAL MEMORANDUM by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Appendix to Reply Claim Construction Brief# 2 Declaration of Alan Heinrich# 3 Exhibit A to Heinrich Dec.# 4 Exhibit B to Heinrich Dec.# 5 Exhibit D to Heinrich Dec.# 6 Exhibit E to Heinrich Dec.# 7 Exhibit H to Heinrich Dec.# 8 Exhibit I to Heinrich Dec.# 9 Exhibit J to Heinrich Dec.# 10 Exhibit K to Heinrich Dec.# 11 Exhibit L to Heinrich Dec.# 12 Exhibit M to Heinrich Dec.# 13 Exhibit C to Heinrich Dec.# 14 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/19/2007)
10/19/2007	118	PRETRIAL MEMORANDUM by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit F# 2 Exhibit G)(FENNELLY, KATHLEEN) (Entered: 10/19/2007)
10/22/2007	119	MOTION to Seal by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief # 2 Declaration of Kathleen N. Fennelly# 3 Text of Proposed Order # 4 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/22/2007)
10/22/2007	--	Set Deadlines as to 119 MOTION to Seal. Motion Hearing set for 11/26/2007 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 10/23/2007)
10/25/2007	120	ORDER ON INFORMAL APPLICATION directing all parties to produce the supplemental responses to the document demands to include documents that came into existence between April 1, 2007 and August 1, 2007 no later than October 29, 2007, etc. Signed by Judge Patty Shwartz on 10/25/07. (aa,) (Entered: 10/26/2007)
10/29/2007	--	Minute Entry for proceedings held before Judge Patty Shwartz : Settlement Conference held on 10/29/2007. (aa,) (Entered: 11/05/2007)
10/30/2007	121	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit Affidavit of Joseph P. La Sala# 2 Affidavit Affidavit of Scott K. Dasovich, Esq.# 3 Text of Proposed Order Proposed Form of Order Pro Hac Vice Dasovich# 4 Certificate of Service Cert of Filing and Service Dasovich)(LASALA, JOSEPH) (Entered: 10/30/2007)
10/30/2007	--	Set Deadlines as to 121 MOTION for Leave to Appear Pro Hac Vice. Motion Hearing set for 11/26/2007 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 11/01/2007)
10/30/2007	122	ORDER granting in part and denying in part 119 Motion to Seal. Signed by Judge Patty Shwartz on 10/25/2007. (nr,) (Entered: 11/02/2007)
11/02/2007	123	MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC..Responses due by 11/12/2007 (Attachments: # 1 Brief in Support of Motion to Strike# 2 Exhibit A to Brief in Support of Motion to Strike (Decl.)# 3 Exhibit 1 to Ex. A to Brief# 4 Exhibit 2 to Ex. A to Brief# 5 Exhibit 3 to Ex. A to Brief# 6 Exhibit 4 to Ex. A to Brief# 7 Exhibit 5 to Ex. A to Brief# 8 Exhibit 6 to Ex. A to Brief# 9 Exhibit 7 to Ex. A to Brief# 10 Exhibit 8 to Ex. A to Brief# 11 Exhibit 9 to Ex. A to Brief# 12 Exhibit 10 to Ex. A to Brief# 13 Exhibit 11 to Ex. A to Brief# 14 Exhibit 12 to Ex. A to Brief# 15 Text of Proposed Order # 16 Certificate of Service)(CURTIN, THOMAS) (Entered: 11/02/2007)
11/02/2007	124	NOTICE by NET2PHONE, INC. of Filing Redacted Documents (Attachments: # 1)(LASALA, JOSEPH) (Entered: 11/02/2007)
11/02/2007	--	Set Deadlines as to 123 MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to

		Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief. Motion Hearing set for 11/26/2007 10:00 AM before Judge Katharine S. Hayden. (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (nr,) (Entered: 11/05/2007)
11/06/2007	125	AFFIDAVIT of Joseph P. La Sala, Esq. re 121 MOTION for Leave to Appear Pro Hac Vice Amended Affidavit by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 11/06/2007)
11/06/2007	126	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 121 MOTION for Leave to Appear Pro Hac Vice Affidavit Scott K. Dasovich, Esq.. (LASALA, JOSEPH) (Entered: 11/06/2007)
11/07/2007	127	ORDER on informal application overruling objection to producing Niklas Zennstrom for deposition; deposition will be completed no later than 12/20/2007. Signed by Judge Patty Shwartz on 11/05/2007. (nr,) (Entered: 11/08/2007)
11/09/2007	128	ORDER granting 121 Motion for Scott K. Dasovich to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 11/07/2007. (nr,) (Entered: 11/09/2007)
11/13/2007	129	BRIEF in Opposition re 123 MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief filed by NET2PHONE, INC.. (Attachments: # 1 # 2)(LASALA, JOSEPH) (Entered: 11/13/2007)
11/13/2007	130	Notice of Request by Pro Hac Vice Scott K. Dasovich, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 1736940.) (LASALA, JOSEPH) (Entered: 11/13/2007)
11/14/2007	--	Pro Hac Vice fee: $ 150, receipt number 1736940 re Scott K. Dasovich (nr,) (Entered: 11/14/2007)
11/16/2007	131	REPLY to Response to Motion re 123 MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief MOTION to Strike 116 Pretrial Memorandum,,,, Specifically the Declaration of Larry Peterson submitted in Support of Reply Claim Constructin Brief Reply Brief in Support of Motion to Strike filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Certificate of Service)(CURTIN, THOMAS) (Entered: 11/16/2007)
11/16/2007	--	Minute Entry for proceedings held before Judge Patty Shwartz : Settlement Conference held on 11/16/2007. (drc,) (Entered: 12/03/2007)
11/19/2007	132	ORDER on informal application granting request to extend deadlines; SEVENTH AMENDED SCHEDULING ORDER: Telephone Conference set for 12/4/2007 03:00 PM before Magistrate Judge Patty Shwartz., Final Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz.,Proposed Pretrial Order due by 6/10/2008.. Signed by Judge Patty Shwartz on 11/16/2007. (nr,) (Entered: 11/19/2007)
11/19/2007	133	ORDER granting application for pro hac vice admission of Mark M. Kuo and Benjamin T. Wang for pro hac vice admission. Signed by Judge Patty Shwartz on 11/16/2007. (nr,) (Entered: 11/19/2007)
11/19/2007	137	DECLARATION of Minh Z. Kuo in support of pro hac vice admission (Attachments: # 1 Decl. of Kathleen N. Fennelly# 2 Decl. of Benjamin T. Wang)(nr,) (Entered: 11/26/2007)
11/20/2007	136	ORDER on informal application requesting to correct order of 11/19/2007, and SEVENTH AMENDED SCHEDULING ORDER: Telephone Conference set for 12/4/2007 03:00 PM before Magistrate Judge Patty Shwartz., Final Pretrial Conference set for 6/17/2008 01:00 PM before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 11/20/2007. (mn,) (Entered: 11/26/2007)
11/21/2007	134	Letter from Thomas R. Curtin, Esq. (CURTIN, THOMAS) (Entered: 11/21/2007)
11/24/2007	135	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 11/24/2007)
11/27/2007	138	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit

		JPLS# 2 Affidavit Robert J. Shaughnessy# 3 Certificate of Service # 4 Text of Proposed Order)(LASALA, JOSEPH) (Entered: 11/27/2007)
11/27/2007	139	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit JPLS# 2 Affidavit Thomas G. Hentoff# 3 Certificate of Service # 4 Text of Proposed Order to appear pro hac vice)(LASALA, JOSEPH) (Entered: 11/27/2007)
11/27/2007	140	LETTER ORDER Setting a Telephone Conference for 11/29/2007 11:00 AM before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 11/26/07. (cs,) (Entered: 11/28/2007)
11/27/2007	--	Set Deadlines as to 139 MOTION for Leave to Appear Pro Hac Vice, 138 MOTION for Leave to Appear Pro Hac Vice. Motion Hearing set for 12/24/2007 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 11/29/2007)
11/28/2007	142	STIPULATION AND ORDER for issuance of depositions for foreign residents. Signed by Judge Patty Shwartz on 11/28/2007. (nr,) (Entered: 11/29/2007)
11/28/2007	143	ORDER granting 138 Motion for Robert J. Shaughnessy and Thomas G. Hentoff to Appear Pro Hac Vice; granting 139 Motion for Leave to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 11/28/2007. (nr,) (Entered: 11/30/2007)
11/29/2007	141	TRANSCRIPT of Proceedings held on October 25, 2007 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 11/29/2007)
11/29/2007	--	Minute Entry for proceedings held before Judge Patty Shwartz : Telephone Status Conference held on 11/29/2007. (drc,) (Entered: 12/03/2007)
11/30/2007	144	NOTICE by NET2PHONE, INC. of Withdrawal of Admission Pro Hac Vice of Michael D. Hurwitz, Esq. (Attachments: # 1 Certificate of Service)(LASALA, JOSEPH) (Entered: 11/30/2007)
11/30/2007	145	ORDER on informal application directing the pltf. to report to the Court its position concerning whether or not it would agree to have a special Master recview all privilege documents and limit any appeal to legal decisions made concerning the pre-sale documents and limit such appeal to one level of appeal or agreed to have a judicial officer review only a limited sampling of the documents and to waive any appeal of the decisions about the rulings, etc.. Signed by Judge Patty Shwartz on 11/29/2007. (nr,) (Entered: 11/30/2007)
12/10/2007	146	ORDER appointing Ronald J. Hedges as the special master; scheduling a telephone conference with the Special Master for 1/8/2008 at 1:00p.m.; Hearing set for 1/10/2008 10:00 AM & 1/15/2008 at 10:00a.m. before Magistrate Judge Patty Shwartz.. Signed by Judge Patty Shwartz on 12/07/2007. (nr,) (Entered: 12/10/2007)
12/10/2007	147	MOTION for Leave to Appear Pro Hac Vice on behalf of Steven R. Ruby, Esq. by NET2PHONE, INC.. (Attachments: # 1 Affidavit of Joseph P. La Sala in suppoprt of motion# 2 Affidavit of Steven R. Ruby, Esq.# 3 Text of Proposed Order # 4 Certification of Service)(LASALA, JOSEPH) (Entered: 12/10/2007)
12/10/2007	148	Notice of Request by Pro Hac Vice Robert J. Shaughnessy, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 1773820.) (LASALA, JOSEPH) (Entered: 12/10/2007)
12/10/2007	149	Letter from Thomas R. Curtin, Esq.. (CURTIN, THOMAS) (Entered: 12/10/2007)
12/10/2007	--	Set Deadlines as to 147 MOTION for Leave to Appear Pro Hac Vice on behalf of Steven R. Ruby, Esq.. Motion Hearing set for 1/14/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 12/11/2007)
12/11/2007	150	Notice of Request by Pro Hac Vice Thomas G. Hentoff, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 1776164.) (LASALA, JOSEPH) (Entered: 12/11/2007)
12/12/2007	151	ORDER granting 147 Motion for Steven R. Ruby to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 12/11/2007. (nr,) (Entered: 12/12/2007)
12/17/2007	152	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit # 2 Affidavit # 3 Certificate of Service # 4)(LASALA, JOSEPH) (Entered: 12/17/2007)
12/17/2007	--	Set Deadlines as to 152 MOTION for Leave to Appear Pro Hac Vice. Motion Hearing set for 1/14/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 12/18/2007)

12/18/2007	153	ORDER granting 152 Motion for Stephen D. Andrews to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 12/18/2007. (nr,) (Entered: 12/19/2007)
12/19/2007	--	Pro Hac Vice fee: $ 300.00, receipt number 200349723 re Marko Kuo & Benjamin Wang (nr,) (Entered: 12/19/2007)
12/19/2007	154	AFFIDAVIT of Ronald J. Hedges by RONALD J. HEDGES. (HEDGES, RONALD) (Entered: 12/19/2007)
12/19/2007	155	Letter from Thomas R. Curtin, Esq., to Hon. Ronald Hedges forwarding Skype Privilege Log Submission. (FENNELLY, KATHLEEN) (Entered: 12/19/2007)
12/20/2007	156	Notice of Request by Pro Hac Vice Marko Kuo to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 12/20/2007)
12/20/2007	157	Notice of Request by Pro Hac Vice Benjamin Wang to receive Notices of Electronic Filings. (FENNELLY, KATHLEEN) (Entered: 12/20/2007)
01/04/2008	158	ORDER on informal application granting Skype's request for deposition of Mr. Oberg. Signed by Judge Patty Shwartz on 01/02/2008. (nr,) (Entered: 01/07/2008)
01/10/2008	159	ORDER granting application for pro hac vice admission of Perry M. Goldberg. Signed by Judge Patty Shwartz on 01/07/2008. (nr,) (Entered: 01/10/2008)
01/11/2008	160	Notice of Request by Pro Hac Vice Steven R. Ruby, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 1813268.) (LASALA, JOSEPH) (Entered: 01/11/2008)
01/11/2008	161	MOTION for Leave to Appear Pro Hac Vice on Behalf of Amy Mason Saharia, Esq. by NET2PHONE, INC.. (Attachments: # 1 Affidavit # 2 Affidavit of Amy Saharia# 3 Text of Proposed Order # 4 Certificate of Service)(LASALA, JOSEPH) (Entered: 01/11/2008)
01/11/2008	--	Set Deadlines as to 161 MOTION for Leave to Appear Pro Hac Vice on Behalf of Amy Mason Saharia, Esq.. Motion Hearing set for 2/4/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 01/14/2008)
01/14/2008	162	TRANSCRIPT of Proceedings held on January 2, 2008 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 01/14/2008)
01/14/2008	163	ORDER on informal application withdrawing the telephone conference schedule for 1/11/2008. Signed by Judge Patty Shwartz on 01/14/2008. (nr,) (Entered: 01/14/2008)
01/14/2008	164	STATEMENT Attaching Revised Net2Phone and IDT Privilege Log by NET2PHONE, INC.. (Attachments: # 1 Exhibit A)(LASALA, JOSEPH) (Entered: 01/14/2008)
01/17/2008	165	DECLARATION of PERRY M. GOLDBERG in support of application for pro hac vice admission (nr,) (Entered: 01/18/2008)
01/17/2008	166	DECLARATION of Kathleen N. Fennelly in support of application for pro hac vice admission (nr,) (Entered: 01/18/2008)
01/18/2008	167	ORDER on informal application granting application for pro hac vice admission of Perry M. Goldberg. Signed by Judge Patty Shwartz on 01/07/2008. (nr,) (Entered: 01/18/2008)
01/18/2008	168	AFFIDAVIT of Joseph P. La Sala in Compliance with Court Order by NET2PHONE, INC.. (Attachments: # 1 Certificate of Service)(LASALA, JOSEPH) (Entered: 01/18/2008)
01/22/2008	169	ORDER granting 161 Motion for Amy Mason Sharia to Appear Pro Hac Vice. Signed by Judge Patty Shwartz on 01/18/2008. (nr,) (Entered: 01/23/2008)
01/22/2008	170	ORDER on informal application advising the parties that if they do not resolve the prior art issue by 1/24/2008 at 5:00p.m. the parties shall then submit their positions concerning the prior art issue via joint letter protocol and be prepared to discuss the issue during the the telephone conference schedule for 1/25/2008 at 5:00p.m.. Signed by Judge Patty Shwartz on 01/18/2008. (nr,) (Entered: 01/23/2008)
01/24/2008	171	AFFIDAVIT of of Compliance by Andrew D. Weiss by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (FENNELLY, KATHLEEN) (Entered: 01/24/2008)
01/24/2008	172	Letter from Thomas R. Curtin, Esq. re: Certifications of Compliance. (CURTIN, THOMAS) (Entered: 01/24/2008)
01/24/2008	173	Letter from Kathleen N. Fennelly, Esq., Requesting Extension of Joint Letter Deadline. (FENNELLY, KATHLEEN) (Entered: 01/24/2008)
01/28/2008	174	ORDER on informal application regarding production of documents and directing that depositions

		be completed before the close of fact discovery, etc.. Signed by Magistrate Judge Patty Shwartz on 01/25/2008. (nr,) (Entered: 01/29/2008)
01/30/2008	175	Notice of Request by Pro Hac Vice Amy Mason Saharia, Esq. referred to in the Order Granting Pro Hac Vice as Amy Mason Sharia to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000001840295.) (LASALA, JOSEPH) (Entered: 01/30/2008)
01/30/2008	176	MOTION for Reconsideration re 174 Order on Oral Motion by NET2PHONE, INC.. (Attachments: # 1 Brief, # 2 Certification of Counsel, # 3 Exhibit A, B & C, # 4 Exhibit D, E & F, # 5 Exhibit G, H & I, # 6 Text of Proposed Order, # 7 Certificate of Service)(LASALA, JOSEPH) (Entered: 01/30/2008)
01/30/2008	--	Pro Hac Vice fee: $ 150, receipt number 1840295 re Amy Mason Sharia (nr,) (Entered: 01/31/2008)
01/30/2008	--	Set Deadlines as to 176 MOTION for Reconsideration re 174 Order on Oral Motion MOTION for Reconsideration re 174 Order on Oral Motion. Motion Hearing set for 3/3/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION SHALL BE DECIDED ON THE PAPER UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 02/01/2008)
02/01/2008	178	ORDER On Informal Application for the deposition of Mr. Cohen to be completed no later than 3/14/08 in either N.J. or California; deposition of the other Vocal Tech shall take place on 2/6/08 and for the resumed deposition of Mr. Oberg shall take place in London during the week of 2/4/08 etc.. Signed by Magistrate Judge Patty Shwartz on 1/31/08(cs,) (Entered: 02/04/2008)
02/01/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Telephone Conference held on 2/1/2008. (aa,) (Entered: 02/25/2008)
02/04/2008	177	MOTION to Quash Subpoena by HOWARD S. JONAS. (Attachments: # 1 Brief, # 2 Certification of Counsel with Exhibits, # 3 Text of Proposed Order, # 4 Certificate of Service)(LASALA, JOSEPH) (Entered: 02/04/2008)
02/04/2008	179	TRANSCRIPT of Proceedings held on September 25, 2007 and January 25, 2008 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 02/05/2008)
02/04/2008	--	Set Deadlines as to 177 MOTION to Quash Subpoena. Motion Hearing set for 3/3/2008 10:00 AM before Judge Katharine S. Hayden. (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT)(nr,) (Entered: 02/07/2008)
02/05/2008	180	STIPULATION (JOINT) TO RESCHEDULE NET2PHONE'S DEPOSITION OF MR. LIOR HARAMATY AND PROPOSED ORDER by NET2PHONE, INC., EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Text of Proposed Order)(FENNELLY, KATHLEEN) (Entered: 02/05/2008)
02/08/2008	181	Letter from Kathleen N. Fennelly, Esq., Requesting Extension of Time to Oppose Motion to Quash Jonas Subpoena re 177 MOTION to Quash Subpoena. (FENNELLY, KATHLEEN) (Entered: 02/08/2008)
02/08/2008	182	STIPULATION Joint Stipulation re:Notice of Deposition of M. Whitman & H. Jonas by NET2PHONE, INC.. (Attachments: # 1 Text of Proposed Order)(LASALA, JOSEPH) (Entered: 02/08/2008)
02/08/2008	184	LETTER ORDER granting Skype's request to extend the deadline to oppose pltf's motion to quash to 2/13/2008. Signed by Magistrate Judge Patty Shwartz on 02/08/2008. (nr,) (Entered: 02/13/2008)
02/08/2008	185	ORDER rescheduling Mr. Haramaty's deposition from Feb. 6, 2008 to March 6, 2008. Signed by Magistrate Judge Patty Shwartz on 02/05/2008. (nr,) (Entered: 02/13/2008)
02/11/2008	183	BRIEF in Opposition re 176 MOTION for Reconsideration re 174 Order on Oral Motion MOTION for Reconsideration re 174 Order on Oral Motion filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of Benjamin T. Wang in Support of Opposition to Motion for Reconsideration, # 2 Certificate of Service)(CURTIN, THOMAS) (Entered: 02/11/2008)
02/14/2008	189	ORDER denying 176 Motion for Reconsideration. Signed by Magistrate Judge Patty Shwartz on 02/14/2008. (nr,) Modified on 2/15/2008 (nr,). (Entered: 02/15/2008)
02/15/2008	187	MOTION to Seal by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief In Support of Motion to Seal, # 2 Text of Proposed Order to Seal, # 3 Declaration of Kathleen N. Fennelly in Support of Motion to Seal, # 4 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 02/15/2008)

02/15/2008	188	Letter from Kathleen N. Fennelly. (CURTIN, THOMAS) (Entered: 02/15/2008)
02/15/2008	190	ORDER terminating/deleting document No. 186 from this docket; terminating 187 Motion to Seal. Signed by Judge Katharine S. Hayden on 02/15/2008. (nr,) (Entered: 02/15/2008)
02/15/2008	191	ORDER on informal application regarding notice of depositions of Margaret Whitman and subpoena for deposition of Howard Jonas. Signed by Magistrate Judge Patty Shwartz on 02/08/2008. (nr,) (Entered: 02/15/2008)
02/19/2008	192	CERTIFICATION in Opposition re 177 MOTION to Quash Subpoena (Including Only Exhibits Not Subject to Motion to Seal) filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit 1 to Kuo Certification, # 2 Exhibit 2 to Kuo Certification, # 3 Exhibit 4 to Kuo Certification, # 4 Exhibit 5 to Kuo Certification, # 5 Exhibit 7 to Kuo Certification, # 6 Exhibit 8 to Kuo Certification, # 7 Exhibit 9 to Kuo Certification, # 8 Exhibit 10 to Kuo Certification, # 9 Exhibit 13 to Kuo Certification)(FENNELLY, KATHLEEN) (Entered: 02/19/2008)
02/19/2008	193	BRIEF in Opposition re 177 MOTION to Quash Subpoena filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Certification of Marko Kuo, # 2 Exhibit 3 to Kuo Certification, # 3 Exhibit 6 to Kuo Certification, # 4 Exhibit 11 to Kuo Certification, # 5 Exhibit 12 to Kuo Certification, # 6 Exhibit 14 to Kuo Certification, # 7 Exhibit 15 to Kuo Certification, # 8 Exhibit 16 to Kuo Certification, # 9 Exhibit 17 to Kuo Certification, # 10 Exhibit 18 to Kuo Certification, # 11 Exhibit 19 to Kuo Certification)(FENNELLY, KATHLEEN) (Entered: 02/19/2008)
02/19/2008	194	MOTION to Seal by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief In Support of Motion to Seal, # 2 Declaration of Kathleen N. Fennelly In Support of Motion to Seal, # 3 Text of Proposed Order, # 4 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 02/19/2008)
02/22/2008	195	ORDER granting in part and denying in part (187) Motion to seal & 194 Motion to Seal. Signed by Magistrate Judge Patty Shwartz on 02/21/2008. (nr,) (Entered: 02/25/2008)
02/22/2008	198	ORDER denying 177 Motion to Quash and directing Howard Jonas to appear for a deposition lasting no longer than three and one-half hours, etc.. Signed by Magistrate Judge Patty Shwartz on 02/21/2008. (nr,) (Entered: 02/26/2008)
02/25/2008	196	REPLY to Response to Motion re 177 MOTION to Quash Subpoena filed by HOWARD S. JONAS. (Attachments: # 1 Certification, # 2 Certificate of Service)(LASALA, JOSEPH) (Entered: 02/25/2008)
02/25/2008	197	MOTION to Seal Document 196 Reply to Response to Motion by HOWARD S. JONAS. (Attachments: # 1 Brief in Support of Motion to Seal, # 2 Text of Proposed Order, # 3 Declaration, # 4 Certificate of Service)(LASALA, JOSEPH) (Entered: 02/25/2008)
02/26/2008	199	ORDER finding as moot 197 Motion to Seal Document; striking reply brief and certfication. Signed by Magistrate Judge Patty Shwartz on 02/26/2008. (nr,) (Entered: 02/27/2008)
02/27/2008	200	BRIEF In Opposition to Motion to Quash filed by SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Certification of Marko Kuo, # 2 Exhibit 1 to Certification of Marko Kuo, # 3 Exhibit 2 to Certification of Marko Kuo, # 4 Exhibit 3 to Certification of Marko Kuo, # 5 Exhibit 4 to Certification of Marko Kuo, # 6 Exhibit 5 to Certification of Marko Kuo, # 7 Exhibit 6 to Certification of Marko Kuo, # 8 Exhibit 7 to Certification of Marko Kuo, # 9 Exhibit 8 to Certification of Marko Kuo, # 10 Exhibit 9 to Certification of Marko Kuo, # 11 Exhibit 10 to Certification of Marko Kuo, # 12 Exhibit 11 to Certification of Marko Kuo, # 13 Exhibit 12 to Certification of Marko Kuo, # 14 Exhibit 13 to Certification of Marko Kuo, # 15 Exhibit 14 to Certification of Marko Kuo, # 16 Exhibit 15 to Certification of Marko Kuo, # 17 Exhibit 16 to Certification of Marko Kuo, # 18 Exhibit 17 to Certification of Marko Kuo, # 19 Exhibit 18 to Certification of Marko Kuo, # 20 Exhibit 19 to Certification of Marko Kuo, # 21 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 02/27/2008)
02/27/2008	201	BRIEF filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit 14 to Certification of Marko Kuo, # 2 Exhibit 16 to Certification of Marko Kuo, # 3 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 02/27/2008)
02/29/2008	--	Pro Hac Vice fee: $ 150, receipt number 200350736 re Perry M. Goldberg (nr,) (Entered: 02/29/2008)
02/29/2008	202	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 02/29/2008)
02/29/2008	--	CLERKS QUALITY CONTROL MESSAGE - The Brief Doc. #201 submitted by K. FENNELLY on 2/27/2008 did not contain a proper electronic signature (s/). PLEASE RESUBMIT THE DOCUMENT WITH THE PROPER ELECTRONIC SIGNATURE (s/ Attorneys Name.) This submission will remain on the docket unless otherwise ordered by the court. (nr,) (Entered: 02/29/2008)
03/04/2008	203	Notice of Request by Pro Hac Vice Perry Goldberg to receive Notices of Electronic Filings.

		(FENNELLY, KATHLEEN) (Entered: 03/04/2008)
03/04/2008	204	BRIEF In Opposition to Motion to Quash Jonas Subpoena (re-filed under seal with proper signature) filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (FENNELLY, KATHLEEN) (Entered: 03/04/2008)
03/05/2008	205	ORDER granting the application for a protective order to preclude the deposition of Margaret Whitman. Signed by Magistrate Judge Patty Shwartz on 03/05/2008. (nr,) (Entered: 03/07/2008)
03/18/2008	206	Letter from Joseph La Sala. (LASALA, JOSEPH) (Entered: 03/18/2008)
03/20/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Telephone Conference held on 3/20/2008. (aa,) (Entered: 03/20/2008)
03/24/2008	207	Letter from Thomas R. Curtin. (CURTIN, THOMAS) (Entered: 03/24/2008)
03/24/2008	208	Letter from Joseph La Sala, Esq. re 207 Letter. (LASALA, JOSEPH) (Entered: 03/24/2008)
03/25/2008	209	Letter from Thomas R. Curtin re 208 Letter. (Attachments: # 1 Exhibit A to March 25 Letter, # 2 Exhibit B to March 25 Letter, # 3 Exhibit C to March 25 Letter, # 4 Exhibit D to March 25 Letter)(CURTIN, THOMAS) (Entered: 03/25/2008)
03/25/2008	210	ORDER on informal application granting the request to adjust the pretrial schedule; SCHEDULING ORDER: Telephone Conference set for 5/8/2008 03:00 PM before Magistrate Judge Patty Shwartz., Proposed Pretrial Order due by 10/28/2008., Final Pretrial Conference set for 11/7/2008 01:00 PM before Magistrate Judge Patty Shwartz.. Signed by Magistrate Judge Patty Shwartz on 03/24/2008. (nr,) (Entered: 03/27/2008)
03/28/2008	211	ORDER on informal application granting request to extend pretrial schedule; EIGHTH AMENDED PRETRIAL SCHEDULING ORDER: Telephone Conference set for 5/8/2008 03:00 PM before Magistrate Judge Patty Shwartz., Proposed Pretrial Order due by 10/28/2008., Final Pretrial Conference set for 11/7/2008 10:00 AM before Magistrate Judge Patty Shwartz.. Signed by Magistrate Judge Patty Shwartz on 03/27/2008. (nr,) (Entered: 03/31/2008)
03/31/2008	212	TRANSCRIPT of Proceedings held on February 1, 2008 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 04/01/2008)
04/17/2008	213	Letter from Joseph P. La Sala. (Attachments: # 1 Text of Proposed Order Order Appointing Mediator)(LASALA, JOSEPH) (Entered: 04/17/2008)
04/18/2008	214	Letter from Joseph P. La Sala. (Attachments: # 1 Text of Proposed Order Consent Order Appointing Mediator)(LASALA, JOSEPH) (Entered: 04/18/2008)
04/21/2008	215	Letter from Ronald J. Hedges, Special Master. (Attachments: # 1 Findings of Fact and Conclusions of Law)(HEDGES, RONALD) (Entered: 04/21/2008)
04/22/2008	216	ORDER on informal application denying pltf's application to preclude the deft. from relying on VocalTec's prior art and for the appointment of the Special Master; etc.. Signed by Magistrate Judge Patty Shwartz on 04/22/2008. (nr,) (Entered: 04/24/2008)
04/24/2008	217	Order Appointing Mediator, RONALD J. HEDGES rep by RONALD J. HEDGES appointed.. Signed by Magistrate Judge Patty Shwartz on 04/18/2008. (nr,) (Entered: 04/25/2008)
05/02/2008	--	The telephone status conference set for 5/8/2008 has been adjourned until 5/16/08 at 11:00 AM in Newark before Magistrate Judge Patty Shwartz. Plaintiff's attorney shall initiate the conference call. Signed by Magistrate Judge Patty Shwartz on 5/2/08. (drc,) (Entered: 05/02/2008)
05/02/2008	218	ORDER on informal application directing that any objections to any report of the Special Master shall be filed with the undersigned in accordance with the deadlines set forth in the FRCP. Signed by Magistrate Judge Patty Shwartz on 04/23/2008. (nr,) (Entered: 05/05/2008)
05/05/2008	219	NOTICE by NET2PHONE, INC. re 215 Letter Net2Phone Inc.'s Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated April 21, 2008 (Attachments: # 1 Brief, # 2 Text of Proposed Order, # 3 Certificate of Service)(LASALA, JOSEPH) (Entered: 05/05/2008)
05/05/2008	220	NOTICE by NET2PHONE, INC. re 219 Notice (Other), Notice (Other) Declaration by Hannah Stott-Bumsted concerning documents submitted for in camera review (LASALA, JOSEPH) (Entered: 05/05/2008)
05/05/2008	221	NOTICE by NET2PHONE, INC. re 219 Notice (Other), Notice (Other) Declaration of Hannah Stott-Bumsted concerning attached Exhibits (Attachments: # 1 Exhibit Ex. 1, # 2 Exhibit Ex. 2, # 3 Exhibit Ex. 3, # 4 Exhibit Ex. 4, # 5 Exhibit Ex. 5.1, # 6 Exhibit Ex. 5.2, # 7 Exhibit Ex. 6,

		# 8 Exhibit Ex. 7, # 9 Exhibit Ex. 8, # 10 Exhibit Ex. 9.1, # 11 Exhibit Ex. 9.2, # 12 Exhibit Ex. 10, # 13 Exhibit Ex. 11, # 14 Exhibit Ex. 12, # 15 Exhibit Ex. 13, # 16 Exhibit Ex. 14, # 17 Exhibit Ex. 15, # 18 Exhibit Ex. 16, # 19 Exhibit Ex. 17, # 20 Exhibit Ex. 18, # 21 Exhibit Ex. 19, # 22 Exhibit Ex. 20, # 23 Exhibit Ex. 21, # 24 Exhibit Ex. 22, # 25 Exhibit Ex. 23, # 26 Exhibit Ex. 24, # 27 Exhibit Ex. 25, # 28 Exhibit Ex. 26)(LASALA, JOSEPH) (Entered: 05/05/2008)
05/06/2008	222	MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) by NET2PHONE, INC.. (Attachments: # 1 Brief, # 2 Text of Proposed Order, # 3 Declaration Declaration of Hannah Stott-Bumsted concerning Exhibits to Motion to Seal, # 4 Exhibit Ex. 1, # 5 Exhibit Ex. 2B, # 6 Exhibit Ex. 3B, # 7 Exhibit Ex. 4B, # 8 Exhibit Ex. 5B, # 9 Exhibit Ex. 6B, # 10 Exhibit Ex. 7B, # 11 Exhibit Ex. 8B, # 12 Exhibit Ex. 9B, # 13 Exhibit Ex. 10B, # 14 Exhibit Ex. 12B, # 15 Exhibit Ex. 13B, # 16 Exhibit Ex. 14B, # 17 Exhibit Ex. 15B, # 18 Exhibit Ex. 16B, # 19 Exhibit 17B, # 20 Exhibit Ex. 18B, # 21 Certificate of Service)(LASALA, JOSEPH) (Entered: 05/06/2008)
05/06/2008	223	NOTICE by NET2PHONE, INC. re 222 MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) Declaration of Hannah Stott-Bumsted concerning Exhibits in Support of Motion to Seal (Attachments: # 1 Exhibit Ex. 2A, # 2 Exhibit Ex. 3A, # 3 Exhibit Ex. 4A, # 4 Exhibit 5A, # 5 Exhibit Ex. 6A, # 6 Exhibit Ex. 7A, # 7 Exhibit Ex. 8A, # 8 Exhibit 9A, # 9 Exhibit 10A, # 10 Exhibit Ex. 11, # 11 Exhibit Ex. 12A, # 12 Exhibit Ex. 13A, # 13 Exhibit Ex. 14A, # 14 Exhibit Ex. 15A, # 15 Exhibit Ex. 16A, # 16 Exhibit Ex. 17A, # 17 Exhibit Ex. 18A)(LASALA, JOSEPH) (Entered: 05/06/2008)
05/06/2008	--	CLERKS QUALITY CONTROL MESSAGE - The Motions & Declarations -Doc. Nos. 219, 220, 221, & 223 filed by JOSEPH LASALA on 5/5/2008 & 5/6/2008 was submitted incorrectly as NOTICES. PLEASE RESUBMIT THE Motions & Declarations using the correct events. This submission will remain on the docket unless otherwise ordered by the court. (nr,) (Entered: 05/06/2008)
05/06/2008	224	MOTION Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated April 21, 2008 re 222 MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other), 220 Notice (Other), 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 215 Letter, 219 Notice (Other), Notice (Other) by NET2PHONE, INC.. (Attachments: # 1 Brief Redacted brief, # 2 Text of Proposed Order, # 3 Declaration Decl. of Hannah Stott-Bumsted concerning documents submitted for in camera review, # 4 Declaration Redacted Decl. of Hannah Stott-Bumsted concerning attached Exhibits, # 5 Exhibit Ex. 2, # 6 Exhibit Ex. 5.1, # 7 Exhibit Ex. 5.2, # 8 Exhibit Ex. 7, # 9 Exhibit Ex. 8, # 10 Exhibit Ex. 9.1, # 11 Exhibit Ex. 9.2, # 12 Exhibit Ex. 10, # 13 Exhibit Ex. 13, # 14 Exhibit Ex. 19, # 15 Exhibit Ex. 20, # 16 Exhibit Ex. 21, # 17 Exhibit Ex. 24, # 18 Exhibit Ex. 26, # 19 Certificate of Service)(LASALA, JOSEPH) (Entered: 05/06/2008)
05/06/2008	225	BRIEF in Support of Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated April 21, 2008 filed by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 05/06/2008)
05/06/2008	226	DECLARATION of Hannah Stott-Bumsted in Support of Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated April 21, 2008 by NET2PHONE, INC.. (Attachments: # 1 Exhibit Ex. 1, # 2 Exhibit Ex. 3, # 3 Exhibit Ex. 4, # 4 Exhibit Ex. 6, # 5 Exhibit Ex. 11, # 6 Exhibit Ex. 12, # 7 Exhibit Ex. 14, # 8 Exhibit Ex. 15, # 9 Exhibit Ex. 16, # 10 Exhibit Ex. 17, # 11 Exhibit Ex. 18, # 12 Exhibit Ex. 22, # 13 Exhibit Ex. 23, # 14 Exhibit Ex. 25)(LASALA, JOSEPH) (Entered: 05/06/2008)
05/06/2008	227	DECLARATION of Hannah Stott-Bumsted re 222 MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other), 223 Notice (Other), Notice (Other), Notice (Other), Notice (Other) in support of Motion to Seal by NET2PHONE, INC.. (Attachments: # 1 Exhibit Ex. 2A, # 2 Exhibit Ex. 3A, # 3 Exhibit Ex. 4A, # 4 Exhibit Ex. 5A, # 5 Exhibit Ex. 6A, # 6 Exhibit Ex. 7A, # 7 Exhibit Ex. 8A, # 8 Exhibit Ex. 9A, # 9 Exhibit Ex. 10A, # 10 Exhibit Ex. 11, # 11 Exhibit Ex. 12A, # 12 Exhibit Ex. 13A, # 13 Exhibit Ex. 14A, # 14 Exhibit Ex. 15A, # 15 Exhibit Ex. 16A, # 16 Exhibit Ex. 17A, # 17 Exhibit Ex. 18A)(LASALA, JOSEPH) (Entered: 05/06/2008)
05/06/2008	--	Set Deadlines as to 222 MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice

		(Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other). Motion Hearing set for 6/2/2008 10:00 AM before Judge Katharine S. Hayden. (PLEASE BE ADVISED THAT THIS MOTION WILL DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT)(nr,) (Entered: 05/12/2008)
05/06/2008	--	Set Deadlines as to 224 Motion Motion Hearing set for 6/2/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 05/12/2008)
05/16/2008	230	ORDER on informal application scheduling a telephone conference on May 21. 2008 at 5:00p.m. to address pltf's application to strike the Vocal Tec evidence; documents identified by pltf. as confidential shall be made available for inspection to Professor Maggs, etc.. Signed by Magistrate Judge Patty Shwartz on 05/16/2008. (nr,) (Entered: 05/20/2008)
05/16/2008	231	ORDER on informal application regarding the special master; Telephone Conference set for 5/19/2008 06:00 PM with the Special Master. Signed by Magistrate Judge Patty Shwartz on 05/16/2008. (nr,) (Entered: 05/20/2008)
05/16/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Telephone Conference held on 5/16/2008. (aa,) (Entered: 05/28/2008)
05/19/2008	228	APPLICATION/PETITION for by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Text of Proposed Order (Unopposed) To Extend Time for eBay and Skype to Respond to Rule 53 (f) Objections, # 2 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 05/19/2008)
05/19/2008	229	RESPONSE in Opposition re 222 MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) MOTION to Seal Document 221 Notice (Other), Notice (Other), Notice (Other), Notice (Other), 219 Notice (Other), Notice (Other) filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 05/19/2008)
05/21/2008	232	RESPONSE in Opposition to Net2Phone's Objections to the Report of the Special Master filed by SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief, # 2 Declaration of Andrew D. Weiss, Esq., # 3 Exhibit A, # 4 Exhibit B, # 5 Exhibit C, # 6 Exhibit D, # 7 Exhibit E, # 8 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 05/21/2008)
05/21/2008	233	RESPONSE in Opposition to Net2Phone's Objections to the Report of the Special Master filed by SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief, # 2 Declaration of Andrew D. Weiss, Esq., # 3 Exhibit A, # 4 Errata B, # 5 Exhibit C, # 6 Exhibit D, # 7 Exhibit E, # 8 Certificate of Service)(FENNELLY, KATHLEEN) (Entered: 05/21/2008)
05/21/2008	234	ORDER on informal application advising that absent a request to reschedule same by May 28, 2008 there shall be an evidentiary hrg. concerning the pltf's request to strike the Vocal Tec documents on June 27, 2008 at 9:30a.m.; etc.. Signed by Magistrate Judge Patty Shwartz on 05/21/2008. (nr,) (Entered: 05/27/2008)
05/21/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Telephone Conference held on 5/21/2008. (aa,) (Entered: 05/28/2008)
05/23/2008	235	ORDER extending deadline to respond to Net2phone's objection to May 21, 2008;. Signed by Magistrate Judge Patty Shwartz on 05/20/2008. (nr,) (Entered: 05/27/2008)
06/02/2008	236	ORDER on informal application granting pltf's request to submit reply brief. Signed by Magistrate Judge Patty Shwartz on 06/02/2008. (nr,) (Entered: 06/03/2008)
06/02/2008	237	ORDER on informal application denying pltf's request to modify the order dated May 6, 2008. Signed by Magistrate Judge Patty Shwartz on 06/02/2008. (nr,) (Entered: 06/04/2008)
06/05/2008	238	REPLY to Response to Motion Rule 53 Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated April 21, 2008 filed by NET2PHONE, INC.. (Attachments: # 1 Declaration Declaration of Steven R. Ruby, # 2 Exhibit Ex. 27, # 3 Exhibit Ex. 28, # 4 Exhibit Ex. 29, # 5 Exhibit Ex. 30, # 6 Exhibit Ex. 31, # 7 Certificate of Service)(LASALA, JOSEPH) (Entered: 06/05/2008)
06/05/2008	239	REPLY to Response to Motion Rule 53 Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated April 21, 2008 filed by NET2PHONE, INC.. (Attachments: # 1 Declaration Declaration of Steven R. Ruby, # 2 Exhibit Redacted Ex. 27, # 3 Exhibit Redacted Ex. 28, # 4 Exhibit Redacted Ex. 29, # 5 Exhibit Redacted Ex. 30, # 6 Exhibit Redacted Ex. 31, # 7 Certificate of Service)(LASALA, JOSEPH) (Entered: 06/05/2008)

06/12/2008	240	NOTICE of Appearance by JOSEPH P. LASALA on behalf of NET2PHONE, INC. (Attachments: # 1 Certificate of Service)(LASALA, JOSEPH) (Entered: 06/12/2008)
06/26/2008	241	Letter from Thomas R. Curtin, Esq.. (CURTIN, THOMAS) (Entered: 06/26/2008)
06/26/2008	242	Letter from Joseph P. La Sala. (Attachments: # 1 Certificate of Service)(LASALA, JOSEPH) (Entered: 06/26/2008)
06/26/2008	243	OPINION. Signed by Magistrate Judge Patty Shwartz on 06/25/2008. (nr,) (Entered: 06/27/2008)
06/26/2008	244	ORDER affirming the Special Master's report in its entirety; denying 222 Motion to Seal Document. Signed by Magistrate Judge Patty Shwartz on 06/25/2008. (nr,) (Entered: 06/27/2008)
06/27/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Evidentiary Hearing held on 6/27/2008. (Court Reporter Margaret Vollmuth.) (aa,) (Entered: 07/02/2008)
06/30/2008	245	ORDER administratively terminating 123 Motion to Strike. Signed by Judge Katharine S. Hayden on 6/30/08. (rg,) (Entered: 06/30/2008)
06/30/2008	246	ORDER on informal application granting request for an extension of time to produce the documents that are the subject of the Special Master's report and Order dated June 25, 2008. Signed by Magistrate Judge Patty Shwartz on 06/30/2008. (nr,) (Entered: 07/01/2008)
07/03/2008	247	ORDER on informal application denying pltf's request under Fed. R. Civ. P. that the Court preclude reliance on the Vocal Tec prior art and directing the pltf. to issue a supplemental expert report that address the Vocal Tec prior art. Signed by Magistrate Judge Patty Shwartz on 07/03/2008. (nr,) (Entered: 07/08/2008)
07/10/2008	248	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit Joseph P. La Sala pro hac vice Russell Shay Glass, # 2 Affidavit of Russell Shay Glass, # 3 Certificate of Service, # 4 Text of Proposed Order)(LASALA, JOSEPH) (Entered: 07/10/2008)
07/10/2008	249	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit, # 2 Affidavit, # 3 Certificate of Service, # 4 Text of Proposed Order)(LASALA, JOSEPH) (Entered: 07/10/2008)
07/10/2008	250	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit JPLS in support of motion for an order admitting Victor Aronoff Kubli, Esq. Pro Hac Vice, # 2 Affidavit of Victor Aronoff Kubli, Esq., # 3 Certificate of Service, # 4 Text of Proposed Order) (LASALA, JOSEPH) (Entered: 07/10/2008)
07/11/2008	252	ORDER granting 250 Motion for Russell Shay Class, Sarah Brashears Macatee and Victor Aronoff Kubli to Appear Pro Hac Vice. Signed by Magistrate Judge Patty Shwartz on 07/11/2008. (nr,) (Entered: 07/15/2008)
07/14/2008	251	TRANSCRIPT of Proceedings held on 7/3/08 before Judge Shwartz. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (jgb) (Entered: 07/15/2008)
07/22/2008	253	APPEAL OF MAGISTRATE JUDGE DECISION to District Court by NET2PHONE, INC. re 247 Order on Oral Motion, (Attachments: # 1 Brief, # 2 Text of Proposed Order, # 3 Certificate of Service) (LASALA, JOSEPH) (Entered: 07/22/2008)
07/22/2008	254	DECLARATION re 253 APPEAL OF MAGISTRATE JUDGE DECISION to District Court by NET2PHONE, INC. re 247 Order on Oral Motion, by NET2PHONE, INC.. (Attachments: # 1 Exhibit, # 2 Exhibit, # 3 Exhibit, # 4 Exhibit, # 5 Exhibit, # 6 Exhibit, # 7 Exhibit, # 8 Exhibit, # 9 Exhibit, # 10 Exhibit, # 11 Exhibit, # 12 Exhibit, # 13 Exhibit, # 14 Exhibit, # 15 Exhibit, # 16 Exhibit, # 17 Exhibit, # 18 Exhibit, # 19 Exhibit, # 20 Exhibit, # 21 Exhibit, # 22 Exhibit, # 23 Exhibit, # 24 Exhibit, # 25 Exhibit, # 26 Exhibit, # 27 Exhibit, # 28 Exhibit, # 29 Exhibit) (LASALA, JOSEPH) (Entered: 07/23/2008)
07/22/2008	--	Set Deadlines as to 253 APPEAL OF MAGISTRATE JUDGE DECISION to District Court by NET2PHONE, INC. re 247 Order on Oral Motion,. Motion Hearing set for 8/18/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 07/23/2008)
07/24/2008	255	MOTION for Leave to File Amended Reply to Amended Counterclaim by NET2PHONE, INC.. (Attachments: # 1 Exhibit A)(LASALA, JOSEPH) (Entered: 07/24/2008)
07/24/2008	256	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit JPLS in support of motion to admit Alan M. Fisch, Esq. and Coke Morgan Stewart, Esq. pro hac vice, # 2 Affidavit Alan M. Fisch, Esq., # 3 Affidavit of Coke Morgan Stewart, Esq., # 4

		Certificate of Service Certification of Service and Filing, # 5 Text of Proposed Order Proposed Order)(LASALA, JOSEPH) (Entered: 07/24/2008)
07/24/2008	257	NOTICE of Appearance by JOSEPH P. LASALA on behalf of NET2PHONE, INC. (Attachments: # 1 Certificate of Service Certification of Service and Filing)(LASALA, JOSEPH) (Entered: 07/24/2008)
07/24/2008	--	Set Deadlines as to 255 MOTION for Leave to File Amended Reply to Amended Counterclaim. Motion Hearing set for 8/18/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWIS NOTIFIED BY THE COURT) (Entered: 07/25/2008)
07/24/2008	--	Set Deadlines as to 256 MOTION for Leave to Appear Pro Hac Vice. Motion Hearing set for 8/18/2008 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 07/25/2008)
07/25/2008	258	NOTICE by NET2PHONE, INC. of Withdrawal of Appearance (LASALA, JOSEPH) (Entered: 07/25/2008)
07/25/2008	259	Letter from Kathleen Fennelly Re: Motion for Leave to File Amended Reply re 255 MOTION for Leave to File Amended Reply to Amended Counterclaim. (FENNELLY, KATHLEEN) (Entered: 07/25/2008)
07/28/2008	260	ORDER granting 255 Motion for Leave to File an amended reply to the counterclaim. Signed by Magistrate Judge Patty Shwartz on 07/25/2008. (nr,) (Entered: 07/29/2008)
07/30/2008	261	Letter from Kathleen N. Fennelly Requesting Extension of Time to Reply to Magistrate Appeal re Set/Reset Motion and R&R Deadlines/Hearings, 254 Declaration,,. (FENNELLY, KATHLEEN) (Entered: 07/30/2008)
07/31/2008	262	ORDER granting defendant's letter request dated July 30, 2008. Signed by Judge Katharine S. Hayden on 7/31/08. (rg,) (Entered: 08/01/2008)
08/04/2008	263	AFFIDAVIT of Coke Morgan Stewart, Esq. re 256 MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit Amended Affidavit of Alan M. Fisch, Esq., # 2 Certificate of Service of JPLS for Amended Affidavit of Alan M. Fisch and Coke Morgan Stewart) (LASALA, JOSEPH) (Entered: 08/04/2008)
08/04/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Telephone Conference held on 8/4/2008. (aa,) (Entered: 08/05/2008)
08/04/2008	268	ORDER granting 256 Motion for Coke Morgan Stewart and Alan M. Fisch to Appear Pro Hac Vice on behalf of plaintiff. Signed by Magistrate Judge Patty Shwartz on 8/4/08. (cs,) (Entered: 08/07/2008)
08/05/2008	264	ORDER on informal application for production of documents and directing defts. to reopen the depositions of Messrs. Jonas, DiGiorgio, Alroy, Skelton, and Greenstein, etc.. Signed by Magistrate Judge Patty Shwartz on 08/04/2008. (nr,) (Entered: 08/06/2008)
08/07/2008	265	Notice of Request by Pro Hac Vice Russell Shay Glass to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002128665.) (LASALA, JOSEPH) (Entered: 08/07/2008)
08/07/2008	266	Notice of Request by Pro Hac Vice Sarah Brashears Macatee to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002128741.) (LASALA, JOSEPH) (Entered: 08/07/2008)
08/07/2008	267	Notice of Request by Pro Hac Vice Victor Aronoff Kubli to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002128807.) (LASALA, JOSEPH) (Entered: 08/07/2008)
08/07/2008	269	MOTION for Leave to Appear Pro Hac Vice by NET2PHONE, INC.. (Attachments: # 1 Affidavit JPLS in support of motion to admit Alan M. Grayson pro hac vice, # 2 Affidavit Alan M Grayson in support of motion pro hac vice, # 3 Certificate of Service JPLS for pro hac vice Alan M. Grayson, # 4 Text of Proposed Order)(LASALA, JOSEPH) (Entered: 08/07/2008)
08/07/2008	--	Set Deadlines as to 269 MOTION for Leave to Appear Pro Hac Vice. Motion Hearing set for 9/2/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COYRT) (Entered: 08/08/2008)
08/08/2008	--	Pro Hac Vice fee: $ 450, receipt number 2128665,2128741,2128807 re Russell Shay Glass, Sarah Brachears Macatee & Victor Aronoff Kubli (nr,) (Entered: 08/08/2008)
08/08/2008	270	STIPULATION re 264 Order on Oral Motion Joint Proposed Order Amending Pretrial Schedule by

		NET2PHONE, INC., EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (FENNELLY, KATHLEEN) (Entered: 08/08/2008)
08/08/2008	271	ORDER granting 269 Motion for Alan Mark Grayson to Appear Pro Hac Vice. Signed by Magistrate Judge Patty Shwartz on 08/08/2008. (nr,) (Entered: 08/11/2008)
08/12/2008	272	ORDER amending pretrial schedule. Signed by Magistrate Judge Patty Shwartz on 08/08/2008. (nr,) (Entered: 08/13/2008)
08/15/2008	273	Notice of Request by Pro Hac Vice Alan M. Grayson to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002140931.) (LASALA, JOSEPH) (Entered: 08/15/2008)
08/15/2008	274	Notice of Request by Pro Hac Vice Coke Morgan Stewart to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002141440.) (LASALA, JOSEPH) (Entered: 08/15/2008)
08/15/2008	275	Notice of Request by Pro Hac Vice Alan M. Fisch to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002141576.) (LASALA, JOSEPH) (Entered: 08/15/2008)
08/18/2008	276	BRIEF Skype's Opposition to Net2Phone's Rule 72 Objection to the Order of Magistrate Judge Shwartz filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of Benjamin Wang, # 2 Exhibit 1 to Wang Declaration, # 3 Exhibit 2 to Wang Declaration, # 4 Exhibit 3 to Wang Declaration, # 5 Exhibit 4 to Wang Declaration, # 6 Exhibit 5 to Wang Declaration, # 7 Exhibit 6 to Wang Declaration, # 8 Exhibit 7 to Wang Declaration, # 9 Exhibit 8 to Wang Declaration, # 10 Exhibit 9 to Wang Declaration, # 11 Exhibit 10 to Wang Declaration, # 12 Exhibit 11 to Wang Declaration, # 13 Exhibit 12 to Wang Declaration, # 14 Errata 13 to Wang Declaration, # 15 Exhibit 14 to Wang Declaration, # 16 Exhibit 15 to Wang Declaration, # 17 Exhibit 16 to Wang Declaration, # 18 Exhibit 17 to Wang Declaration, # 19 Exhibit 18 to Wang Declaration, # 20 Exhibit 19 to Wang Declaration, # 21 Exhibit 20 to Wang Declaration, # 22 Exhibit 21 to Wang Declaration, # 23 Exhibit 22 to Wang Declaration, # 24 Exhibit 23 to Wang Declaration, # 25 Exhibit 24 to Wang Declaration, # 26 Exhibit 25 to Wang Declaration, # 27 Exhibit 26 to Wang Declaration, # 28 Exhibit 27 to Wang Declaration, # 29 Errata 28 to Wang Declaration, # 30 Exhibit 29 to Wang Declaration, # 31 Exhibit 30 to Wang Declaration, # 32 Exhibit 31 to Wang Declaration, # 33 Exhibit 32 to Wang Declaration, # 34 Exhibit 33 to Wang Declaration, # 35 Exhibit 34 to Wang Declaration, # 36 Certificate of Service)(CURTIN, THOMAS) (Entered: 08/18/2008)
08/20/2008	277	Letter from Kathleen N. Fennelly Re: Relaxation of Brief Page Limits. (FENNELLY, KATHLEEN) (Entered: 08/20/2008)
08/21/2008	278	Letter from Joseph P. La Sala to Judge Hayden re Summary Judgment Page Extension. (LASALA, JOSEPH) (Entered: 08/21/2008)
08/22/2008	279	ORDER denying letter request dated August 20, 2008 by defendants eBay,Inc. and Skype which requested leave to file an over-length brief. Signed by Judge Katharine S. Hayden on 8/22/08. (rg,) (Entered: 08/22/2008)
08/26/2008	--	Pro Hac Vice fee: $ 450., receipt number 2140931,2141440,2141 576 re Alan Grayson, Coke Morgan Stewart & Alan M. Fisch (nr,) (Entered: 08/26/2008)
08/29/2008	280	Letter from Joseph P. La Sala. (Attachments: # 1 Exhibit Exhibits A through E to Joint Letter, # 2 Exhibit Exhibit F - Part I, # 3 Exhibit Exhibit F - Part 2, # 4 Exhibit Exhibit F - Part 3, # 5 Exhibit Exhibits 1 through 3)(LASALA, JOSEPH) (Entered: 08/29/2008)
09/08/2008	281	ORDER on informal application overruling the efforts to limit the pltfs' expert access to the source code or test environment; denying pltf's request to modify the terms of access; and mooting deft's request to compel pltf. to provide its portion about Mr. Derwin's deposition. Signed by Magistrate Judge Patty Shwartz on 09/05/2008. (nr,) (Entered: 09/09/2008)
09/08/2008	282	ORDER on informal application granting request to continue Mr. Derwin's deposition; granting request to compel the production of documents from Mr. Derwin. Signed by Magistrate Judge Patty Shwartz on 09/05/2008. (nr,) (Entered: 09/09/2008)
09/10/2008	283	Letter from Joseph P. La Sala to Judge Shwartz. (LASALA, JOSEPH) (Entered: 09/10/2008)
09/15/2008	284	TRANSCRIPT of Proceedings held on September 5, 2008 before Judge Shwartz. Court Reporter/Recorder: King Transcription Services. PLEASE NOTE: The complete transcript of these proceedings is maintained in paper format on file in the Clerks Office. To request copies of this transcript, contact the Official Court Reporter or Transcription Service who prepared the transcript. (ji,) (Entered: 09/15/2008)
09/16/2008	285	ORDER on informal application granting requesting to extend deadline to disclose supplemental expert reports; directing that all summary judgment motions be filed no later than 10/10/2008

		and setting briefing schedule. Signed by Magistrate Judge Patty Shwartz on 09/15/2008. (nr,) (Entered: 09/17/2008)
09/17/2008	286	MOTION to Seal by NET2PHONE, INC.. (Attachments: # 1 Brief Brief in Support of Notice of Motion to Seal, # 2 Certification of JPLS in support of Motion to Seal, # 3 Exhibit to Certification of JPLS in support of motion to seal, # 4 Exhibit #2 to Certification of JPLS in support of motion to seal, # 5 Text of Proposed Order Propose Order to Seal, # 6 Certificate of Service of Motion to Seal)(LASALA, JOSEPH) (Entered: 09/17/2008)
09/17/2008	287	NOTICE by NET2PHONE, INC. Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated September 4, 2008 (Attachments: # 1 Certification of JPLS in support of Rule 53(f) objections and motion to modify findings of fact and conclusions of law of Special Master dated September 17, 2008, # 2 Exhibit # 1, # 3 Exhibit #2, # 4 Brief Redacted Brief in support of plaintiff's objectin to the Special Master's Ruling on Privilege Log Entry 9072, # 5 Text of Proposed Order, # 6 Certificate of Service of JPLS)(LASALA, JOSEPH) (Entered: 09/17/2008)
09/17/2008	288	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 286 MOTION to Seal Amended Certification of JPLS to add Electronic Signature. (LASALA, JOSEPH) (Entered: 09/17/2008)
09/17/2008	--	Set Deadlines as to 289 MOTION to Seal. Motion Hearing set for 10/20/2008 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 09/22/2008)
09/18/2008	--	CLERK'S NOTE: document #246 was filed without motion. Counsel to file Motion to Seal (only). See doc #246 for supporting papers. (jd,) (Entered: 09/18/2008)
09/18/2008	--	CLERK'S NOTE: Please be advised the correct document #is 286 (motion to seal) not doc #246 (jd,) (Entered: 09/18/2008)
09/18/2008	289	MOTION to Seal by NET2PHONE, INC.. (LASALA, JOSEPH) (Entered: 09/18/2008)
09/19/2008	290	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 287 Notice (Other), Notice (Other), Notice (Other) Redacted Brief in Support of Plaintiff's Objection to the Special Master's Ruling on Privilege Log Entry 9072. (LASALA, JOSEPH) (Entered: 09/19/2008)
09/19/2008	291	Letter from Joseph P. La Sala enclosing Proposed Order to Seal. (Attachments: # 1 Proposed Order to Seal Exhibit F attached to Docket Entry #280 in its entirety, # 2 Certificate of Service and Filing)(LASALA, JOSEPH) (Entered: 09/19/2008)
09/22/2008	292	ORDER on informal application sealing Exhibit F in it's entirety. Signed by Magistrate Judge Patty Shwartz on 09/19/2008. (nr,) (Entered: 09/22/2008)
09/29/2008	--	CLERKS QUALITY CONTROL MESSAGE - The Motion Doc. 287 filed by JOSEPH LASALA on 9/17/2008 was submitted incorrectly as a Notice. PLEASE RESUBMIT THE MOTION USING MOTION. This submission will remain on the docket unless otherwise ordered by the court. (nr,) (Entered: 09/29/2008)
09/29/2008	293	MOTION Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated September 4, 2008 (refiling of docket entry #287 pursuant to Clerk's Quality Control Message of 9/29/08) by NET2PHONE, INC.. (Attachments: # 1 Certification of Joseph La Sala, # 2 Certification of Joseph La Sala - Exhibit 1, # 3 Certification of Joseph La Sala - Exhibit 2, # 4 Brief (redacted) in Support of Plaintiff's Objections to the Special Master's Ruling on Privilege Log Entry 9072, # 5 Text of Proposed Order, # 6 Certificate of Service) (LASALA, JOSEPH) (Entered: 09/29/2008)
09/29/2008	--	Set Deadlines as to 293 MOTION Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated September 4, 2008 (refiling of docket entry #287 pursuant to Clerk's Quality Control Message of 9/29/08) MOTION Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of Special Master Dated September 4, 2008. Motion Hearing set for 11/3/2008 10:00 AM before Judge Katharine S. Hayden. (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT)(nr,) (Entered: 10/01/2008)
10/03/2008	294	ORDER on informal application denying request to extend the deadlines associated with the preparation of the joint proposed final pretrial order; granting request to extend deadline to serve supplemental expert reports and to file motion for summary judgment. Signed by Magistrate Judge Patty Shwartz on 10/03/2008. (nr,) (Entered: 10/06/2008)
10/07/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Telephone Conference held on 10/7/2008. (aa,) (Entered: 10/08/2008)
10/08/2008	295	Letter from Thomas R. Curtin and Joseph LaSala requesting conference call. (CURTIN, THOMAS) (Entered: 10/08/2008)
10/08/2008	296	ORDER issuing a expedited briefing schedule. Signed by Judge Katharine S. Hayden on 10/8/08.

		(rg,) (Entered: 10/08/2008)
10/09/2008	297	MOTION for Leave to File Supplemental Claim Construction Brief by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief, # 2 Declaration of Benjamin Wang, # 3 Exhibit A to Wang Declaration (Redacted), # 4 Exhibit B to Wang Declaration (Redacted), # 5 Exhibit C to Wang Declaration (Redacted), # 6 Exhibit E to Wang Declaration (Redacted), # 7 Exhibit E to Wang Declaration, # 8 Exhibit F to Wang Declaration, # 9 Exhibit G to Wang Declaration (Redacted), # 10 Exhibit H to Wang Declaration (Redacted), # 11 Exhibit I to Wang Declaration, # 12 Exhibit J to Wang Declaration, # 13 Exhibit K to Wang Declaration, # 14 Exhibit L to Wang Declaration, # 15 Exhibit M to Wang Declaration, # 16 Exhibit N to Wang Declaration, # 17 Exhibit O to Wang Declaration, # 18 Text of Proposed Order, # 19 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/09/2008)
10/09/2008	298	Exhibit to 297 Motion for Leave to File,,, by SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit B to Wang Declaration, # 2 Exhibit C to Wang Declaration, # 3 Exhibit D to Wang Declaration, # 4 Exhibit G to Wang Declaration, # 5 Exhibit H to Wang Declaration, # 6 Exhibit O to Wang Declaration, # 7 Brief in Support of Motion for Leave to File Supplemental Claim Construction Brief)(CURTIN, THOMAS) (Entered: 10/09/2008)
10/09/2008	299	MOTION to Seal by SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief in Support of Motion to Seal, # 2 Declaration of Kathleen N. Fennelly in Support of Motion to Seal, # 3 Text of Proposed Order Including Proposed Findings of Fact and Conclusions of Law, # 4 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/09/2008)
10/09/2008	--	Set Deadlines as to 299 MOTION to Seal, 297 MOTION for Leave to File Supplemental Claim Construction Brief. Motion Hearing set for 11/3/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 10/10/2008)
10/10/2008	300	MOTION for Reconsideration re 294 Order on Oral Motion, Rule 72 Objection to the Order of Magistrate Judge Patty Shwartz Regarding Extension of Filing Deadlines for the Final Joint Pretrial Order by NET2PHONE, INC.. (Attachments: # 1 Brief in Support of Objection, # 2 Certificate of Service, # 3 Text of Proposed Order, # 4 Text of Proposed Order for Alternate Relief)(LASALA, JOSEPH) (Entered: 10/10/2008)
10/10/2008	301	MOTION for Reconsideration re 294 Order on Oral Motion, 296 Order by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of Alan J. Heinrich, # 2 Exhibit 1 to Heinrich Decl., # 3 Exhibit 2 to Heinrich Decl., # 4 Exhibit 3 to Heinrich Decl., # 5 Exhibit 4 to Heinrich Decl., # 6 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/10/2008)
10/10/2008	302	Exhibit to 301 Motion for Reconsideration, by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (CURTIN, THOMAS) (Entered: 10/11/2008)
10/10/2008	304	ORDER on informal application denying request for leave to serve a subpoena upon Kenyon & Kenyon for documents; granting request to reopen the deposition of Mr. Skelton, etc.. Signed by Magistrate Judge Patty Shwartz on 10/10/2008. (nr,) (Entered: 10/14/2008)
10/10/2008	--	Set Deadlines as to 300 MOTION for Reconsideration re 294 Order on Oral Motion, Rule 72 Objection to the Order of Magistrate Judge Patty Shwartz Regarding Extension of Filing Deadlines for the Final Joint Pretrial Order MOTION for Reconsideration re 294 Order on Oral Motion, Rule 72 Objection to the Order of Magistrate Judge Patty Shwartz Regarding Extension of Filing Deadlines for the Final Joint Pretrial Order. Motion Hearing set for 11/3/2008 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWIS NOTIFIED BY THE COURT) (Entered: 10/14/2008)
10/11/2008	303	MOTION to Seal Document 302 Exhibit (to Document) by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief, # 2 Declaration of Kathleen N. Fennelly, # 3 Text of Proposed Order, # 4 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/11/2008)
10/14/2008	--	CLERKS QUALITY CONTROL MESSAGE - The BRIEF DOC. #301 filed by T. CURTIN on 10/10/2008 was submitted incorrectly as a MOTION. PLEASE RESUBMIT THE BRIEF USING RESPONSES AND REPLIES. This submission will remain on the docket unless otherwise ordered by the court. (nr,) (Entered: 10/14/2008)
10/15/2008	305	BRIEF Defendants' Memorandum in Support of Parties' Expedited Appeal of Magistrate Judge Shwartz's October 3, 2008, Order filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (CURTIN, THOMAS) (Entered: 10/15/2008)
10/20/2008	306	RESPONSE in Opposition re 297 MOTION for Leave to File Supplemental Claim Construction Brief filed by NET2PHONE, INC.. (Attachments: # 1 Certificate of Service, # 2 Text of Proposed Order, # 3 Text of Proposed Order Alternate Proposed Order)(LASALA, JOSEPH) (Entered: 10/20/2008)

10/20/2008	307	BRIEF Response to Plaintiff's Objection to the Special Master's Ruling on Privilege Log Entry 9072 filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of Marko Kuo, # 2 Exhibit 1 to Kuo Decl., # 3 Exhibit 2 to Kuo Decl., # 4 Exhibit 3 to Kuo Decl. (Redacted), # 5 Exhibit 4 to Kuo Decl. (Redacted), # 6 Exhibit 5 to Kuo Decl. (Redacted), # 7 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/20/2008)
10/20/2008	308	BRIEF Skype's Resopnse to Plaintiff's Objection to the Special Master's Ruling on Privilege Log Entry 9072 filed by SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of Marko Kuo, # 2 Exhibit 1 to Kuo Decl., # 3 Exhibit 2 to Kuo Decl., # 4 Exhibit 3 to Kuo Decl. (Redacted), # 5 Exhibit 4 to Kuo Decl. (Redacted), # 6 Exhibit 5 to Kuo Decl. (Redacted), # 7 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/20/2008)
10/20/2008	309	BRIEF filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief in Support of Skype's Response to Plaintiff's Objection to the Special Master's Ruling on Privilege Log Entry 9072 (Unredacted), # 2 Exhibit 3 to Kuo Decl. (Unredacted), # 3 Exhibit 4 to Kuo Decl. (Unredacted), # 4 Exhibit 5 to Kuo Decl. (Unredacted))(CURTIN, THOMAS) (Entered: 10/20/2008)
10/20/2008	310	MOTION to Seal by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief in Support of Motion to Seal, # 2 Declaration of Kathleen N. Fennelly in Support of Motion to Seal, # 3 Text of Proposed Order to Seal Including Proposed Findings of Fact and Conclusions of Law, # 4 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/20/2008)
10/22/2008	311	ORDER granting in part and denying in part 299 Motion to Seal certain portions of exhibits and directing the parties to submit by 10/30/2008 a redacted version of these documents. Signed by Magistrate Judge Patty Shwartz on 10/21/2008. (nr,) (Entered: 10/22/2008)
10/27/2008	312	MOTION to Seal by NET2PHONE, INC.. (Attachments: # 1 Declaration J. LaSala, # 2 Brief motion to seal, # 3 Text of Proposed Order order to seal, # 4 Certificate of Service J. LaSala)(LASALA, JOSEPH) (Entered: 10/27/2008)
10/27/2008	313	REPLY BRIEF to Opposition to Motion re 253 APPEAL OF MAGISTRATE JUDGE DECISION to District Court by NET2PHONE, INC. re 247 Order on Oral Motion,, 312 MOTION to Seal filed by NET2PHONE, INC.. (Attachments: # 1 Brief Plaintiff's Reply in Further Support of its Objection to the Special Master's Ruling on Privilege Log No: 9072, # 2 Exhibit 1, # 3 Exhibit 2, # 4 Exhibit 3, # 5 Certificate of Service J. LaSala)(LASALA, JOSEPH) (Entered: 10/27/2008)
10/27/2008	314	BRIEF filed by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Declaration of Benjamin Wang, # 2 Index A to Wang. Dec. (Redacted), # 3 Exhibit b, # 4 Exhibit C (Redacted), # 5 Exhibit D, # 6 Exhibit E (Redacted), # 7 Exhibit F (Redacted), # 8 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/27/2008)
10/27/2008	--	Set Deadlines as to 312 MOTION to Seal. Motion set for 12/1/2008 10:00 AM before Judge Katharine S. Hayden. (nr,)(PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 10/28/2008)
10/27/2008	317	OPINION & ORDER that the order of Magistrate Judge Shwartz dated 10/3/08 denying the parties' request to delay filing of certain portions of the joint proposed Final Pretrial Order is AFFIRMED; scheduling deatlines set forth by Magistrate Judge Shwartz shall remain intact, etc.. Signed by Judge Katharine S. Hayden on 10/27/08. (rg,) (Entered: 10/29/2008)
10/28/2008	315	DECLARATION of Benjamin Wang in Support of Skype's Reply Brief in Support of Motion to File Supplemental Claim Construction Brief (With Unredacted Copies of Ex. A, C, E and F Attached) (Subject to Motion to Seal) re 314 Brief, by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit A to Wang Decl. (Unredacted), # 2 Exhibit B to Wang Decl., # 3 Exhibit C to Wang Decl. (Unredacted), # 4 Exhibit D to Wang Decl., # 5 Exhibit E to Wang Decl. (Unredacted), # 6 Exhibit F to Wang Decl. (Unredacted))(CURTIN, THOMAS) (Entered: 10/28/2008)
10/28/2008	316	MOTION to Seal Exhibits A, C, F and G to Declaration of Benjamin Wang In Support of Reply Brief in Support of Motion to File Supplemental Claim Construction Brief by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Brief, # 2 Declaration of Kathleen N. Fennelly, # 3 Text of Proposed Order, # 4 Certificate of Service)(CURTIN, THOMAS) (Entered: 10/28/2008)
10/28/2008	--	Set Deadlines as to 316 MOTION to Seal Exhibits A, C, F and G to Declaration of Benjamin Wang In Support of Reply Brief in Support of Motion to File Supplemental Claim Construction Brief. Motion set for 12/1/2008 10:00 AM before Judge Katharine S. Hayden. (nr,) (PLEASE BE ADVISED THAT THIS MOTION WILL BE DECIDED ON THE PAPERS UNLESS OTHERWISE NOTIFIED BY THE COURT) (Entered: 10/30/2008)
10/29/2008	318	ORDER that the unredacted version of Exhibit 1 to the Heinrich Declaration remain under seal. Signed by Judge Katharine S. Hayden on 10/21/2008. (nr,) (Entered: 10/29/2008)

10/30/2008	319	DECLARATION of Benjamin Wang In Support of Motion to File Supplemental Claim Construction Brief (With Redacted Exhibits as Per D.E. 311)e re 297 MOTION for Leave to File Supplemental Claim Construction Brief, 311 Order on Motion to Seal by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit A to Wang Dec., # 2 Exhibit B to Wang Decl., # 3 Exhibit C to Wang Decl. (Redacted Pursuant to D.E. 311), # 4 Exhibit D to Wang Decl. (Redacted Pursuant to D.E. 311), # 5 Exhibit E to Wang Decl., # 6 Exhibit F to Wang Decl., # 7 Exhibit G to Wang Decl. (Redacted Pursuant to D.E. 311), # 8 Exhibit H to Wang Decl. (Redacted Pursuant to D.E. 311), # 9 Exhibit I to Wang Decl., # 10 Exhibit J to Wang Decl., # 11 Exhibit K to Wang Decl., # 12 Exhibit L to Wang Decl., # 13 Exhibit M to Wang Decl., # 14 Exhibit N to Wang Decl., # 15 Exhibit O to Wang Decl. (Redacted Pursuant to D.E. 311)) (FENNELLY, KATHLEEN) (Entered: 10/30/2008)
10/31/2008	324	Transcript of Proceedings held on October 10, 2008, before Judge Shwartz. Court Reporter/Transcriber King Transcription. (ji,) Modified on 11/7/2008 (ji,). (Entered: 11/06/2008)
11/03/2008	320	ORDER granting in part and denying in part 316 Motion to Seal certain documents; directing the deft. to file on public docket a version of these items redacted no later than 11/7/2008. Signed by Magistrate Judge Patty Shwartz on 10/31/2008. (nr,) (Entered: 11/05/2008)
11/05/2008	321	NOTICE of Appearance by GEORGE C. JONES on behalf of EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC. (JONES, GEORGE) (Entered: 11/05/2008)
11/05/2008	322	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 260 Order on Motion for Leave to File. (Attachments: # 1 Certificate of Service)(LASALA, JOSEPH) (Entered: 11/05/2008)
11/05/2008	323	AMENDED DOCUMENT by NET2PHONE, INC.. Amendment to 322 Amended Document Amended Certification of Service. (LASALA, JOSEPH) (Entered: 11/05/2008)
11/05/2008	325	ORDER granting in part and denying in part 310 Motion to Seal; denying 312 Motion to Seal; denying 289 Motion to Seal; denying 293 Motion objection to Special Master's ruling and adopting the Special Master's decision; directing the parties to produce no later than 11/10/2008 a redacted version of these documents. Signed by Magistrate Judge Patty Shwartz on 11/03/2008. (nr,) (Entered: 11/06/2008)
11/06/2008	326	DECLARATION of Benjamin Wang in Support of Skype's Reply Brief in Support of Motion to File Supplemental Claim Construction Brief (with redacted and unredacted exhibits per D.E. 320) re 314 Brief, 320 Order on Motion to Seal, 297 MOTION for Leave to File Supplemental Claim Construction Brief by EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC.. (Attachments: # 1 Exhibit A to Wang Declaration (Unredacted per D.E. 320), # 2 Exhibit B to Wang Declaration, # 3 Exhibit C to Wang Declaration (Unredacted per D.E. 320), # 4 Exhibit D to Wang Declaration, # 5 Exhibit E to Wang Declaration (Redacted per D.E. 320), # 6 Exhibit F to Wang Declaration (Redacted per D.E. 320))(JONES, GEORGE) (Entered: 11/06/2008)
11/07/2008	--	Minute Entry for proceedings held before Magistrate Judge Patty Shwartz: Final Pretrial Conference held on the record on 11/7/2008. (CD #S08-23.) (aa,) (Entered: 11/30/2008)
11/10/2008	327	MOTION for Leave to Appear Pro Hac Vice on Behalf of Joseph M. Drayton, Esq., Vandana Koelsch, Esq., Kevin Jakel, Esq. and Gillian T. DiFilippo, Eqs. by NET2PHONE, INC.. (Attachments: # 1 Affidavit of Joseph M. Drayton, Esq., # 2 Affidavit of Vandana Koelsch, Esq., # 3 Affidavit of Kevin Jakel, Esq., # 4 Affidavit of Gillian T. DiFilippo, Esq., # 5 Affidavit of Joseph P. La Sala, Esq., # 6 Text of Proposed Order, # 7 Certificate of Service)(LASALA, JOSEPH) (Entered: 11/10/2008)
11/10/2008	328	ORDER on informal application directing the parties to submit their revisions to the portions of the revised joint proposed final pretrial order; Proposed Pretrial Order due on 12/9/2008. granting request to depose Mr. Skelton and granting request to allow Professor Maggs and Professor Johnson to supplement expert reports Signed by Magistrate Judge Patty Shwartz on 11/7/2008. (nr,) (Entered: 11/13/2008)
11/12/2008	329	ORDER granting 327 Motion for Joseph M. Drayton, Vandana Koelsch, Kevin Jakel, and Gillian T. DiFilippo to Appear Pro Hac Vice. Signed by Magistrate Judge Patty Shwartz on 11/12/2008. (nr,) Modified on 11/14/2008 (nr,). (Entered: 11/14/2008)
11/14/2008	330	Notice of Request by Pro Hac Vice Joseph M. Drayton, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 03120000000002271582.) (LASALA, JOSEPH) (Entered: 11/14/2008)
11/14/2008	331	Notice of Request by Pro Hac Vice Vandana Koelsch, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 03120000000002271590.) (LASALA, JOSEPH) (Entered: 11/14/2008)
11/14/2008	332	Notice of Request by Pro Hac Vice Kevin Jakel, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 03120000000002271599.) (LASALA, JOSEPH) (Entered:

		11/14/2008)
11/14/2008	333	Notice of Request by Pro Hac Vice Gillian T. DiFilippo, Esq. to receive Notices of Electronic Filings. (Pro Hac Vice fee $ 150 receipt number 0312000000002271610.) (LASALA, JOSEPH) (Entered: 11/14/2008)
11/17/2008	334	Transcript of Proceedings (Volume I) held on 11/3/2008, before Judge Patty Shwartz. Court Reporter/Transcriber King Transcription Services, Telephone number 973 237-6080. NOTICE REGARDING REDACTION OF TRANSCRIPTS: The parties have seven (7) calendar days to file with the Court a Notice of Intent to Request Redaction of this Transcript. If no such notice is filed, the transcript will be made remotely available in electronic format to the public without redaction after ninety(90) calendar days. The redaction policy is located on our website at www.njd.uscourts.gov. Transcripts may be viewed at the court public terminal or purchased through the Court Reporter/Transcriber before the deadline for release of transcript restriction. After that date it may be obtained through PACER. Redaction Request due 12/8/2008. Redacted Transcript Deadline set for 12/18/2008. Release of Transcript Restriction set for 2/15/2009. (mn,) (Entered: 11/17/2008)
11/24/2008	335	Letter from Kathleen N. Fennelly (Joint Letter) Regarding Markman Hearing Schedule. (FENNELLY, KATHLEEN) (Entered: 11/24/2008)
12/03/2008	336	SCHEDULING ORDER: Status Conference in person set for 12/10/2008 10:00 AM before Judge Katharine S. Hayden.. Signed by Judge Katharine S. Hayden on 12/3/08. (rg,) (Entered: 12/03/2008)
12/10/2008	--	Minute Entry for proceedings held before Judge Katharine S. Hayden: Status Conference held on 12/10/2008; scheduling order to issue. (rg,) (Entered: 12/10/2008)
12/12/2008	337	ORDER setting a Markman Hearing for 3/2/2009 10:00 AM.. Signed by Judge Katharine S. Hayden on 12/10/2008. (nr,) (Entered: 12/12/2008)
01/05/2009	338	NOTICE by NET2PHONE, INC. of change of firm name from Grayson & Kubli, P.C. to Kubli & Associates, P.C. (Attachments: # 1 Certificate of Service)(LASALA, JOSEPH) (Entered: 01/05/2009)
01/05/2009	339	NOTICE by NET2PHONE, INC. re 273 Notice of Pro Hac Vice to Receive NEF Notice of Withdrawal of Pro Hac Vice Attorney Alan M. Grayson, Esq. (Attachments: # 1 Certificate of Service) (LASALA, JOSEPH) (Entered: 01/05/2009)
01/09/2009	340	Letter from Joseph P. La Sala, Esq. regarding change of contact information for pro hac vice attorneys re 338 Notice (Other). (LASALA, JOSEPH) (Entered: 01/09/2009)
01/14/2009	341	FINAL PRETRIAL ORDER. Signed by Magistrate Judge Patty Shwartz on 12/11/2008. (Attachments: # 1 Cont. of Pretrial Order, # 2 Cont. of Pretrial Order, # 3 Cont. of Pretrial Order, # 4 Cont. of Pretrial Order, # 5 Cont. of Pretrial Order)(nr,) (Entered: 01/14/2009)
02/06/2009	342	ORDER on informal application granting deft's request to strike the Oct. 10,2009 "supplemental" report of Kevin Jaffay and "rebuttal" reports of Matthew Lynde and Samrat Bhattacharjee. Signed by Magistrate Judge Patty Shwartz on 02/05/2009. (nr,) (Entered: 02/09/2009)

Source: Command Searching > **Utility, Design and Plant Patents**
Terms: **patno= 6108704** (Edit Search | Suggest Terms for My Search)

533115 (08) 6108704 August 22, 2000

UNITED STATES PATENT AND TRADEMARK OFFICE GRANTED PATENT

6108704

Get Drawing Sheet 1 of 6
Access PDF of Official Patent *
Order Patent File History / Wrapper from REEDFAX®
Link to Claims Section

August 22, 2000

Point-to-point internet protocol

REEXAM-LITIGATE:

NOTICE OF LITIGATION

NET2PHONE, Inc v. Ebay, Inc et al, Filed June 1, 2006, D.C. New Jersey, Doc. No. 2:06cv2469

INVENTOR: Hutton, Glenn W. - Miami, Florida, United States (US)Mattaway, Shane D. - Boca Raton, Florida, United States (US); Strickland, Craig B. - Tamarac, Florida, United States (US)

APPL-NO: 533115 (08)

FILED-DATE: September 25, 1995

GRANTED-DATE: August 22, 2000

ASSIGNEE-PRE-ISSUE: January 8, 1996 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., INTERNET TELEPHONE COMPANY SUITE 305 1 SOUTH OCEAN BOULEVARDBOCA RATON, FLORIDA, 33432, Reel and Frame Number: 008295/0167
May 30, 1996 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., INTERNET TELEPHONE COMPANY 1 SOUTH OCEAN BOULEVARD, SUITE 305BOCA RATON, FLORIDA, 33432, Reel and Frame Number: 007981/0020
May 30, 1996 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., NETSPEAK CORPORATION STE. 104 902 CLINT MOORE ROADBOCA RATON, FLORIDA, 33437, Reel and Frame Number: 007981/0053
February 22, 1999 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., NETSPEAK CORPORATION BOCA RATON FLORIDA 33487, Reel and Frame Number: 009792/0568
June 7, 1999 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., NETSPEAK CORPORATION 902 CLINT MOORE ROAD, SUITE 104BOCA RATON, FLORIDA,

33487, Reel and Frame Number: 010012/0953

ASSIGNEE-AT-ISSUE: NetSpeak Corporation, Boca Raton, Florida, United States (US), United States company or corporation (02)

ASSIGNEE-AFTER-ISSUE: September 12, 2005 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., VOIP TECHNOLOGY HOLDINGS, LLC 520 BROAD STREET, 8TH FLOOR NEWARK NEW JERSEY 07102, Reel and Frame Number: 016522/0205
October 28, 2005 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., NET2PHONE, INC. 520 BROAD STREET, 8TH FLOOR NEWARK NEW JERSEY 07102, Reel and Frame Number: 016945/0858
October 28, 2005 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., NET2PHONE, INC. 520 BROAD STREET, 8TH FLOOR NEWARK NEW JERSEY 07102, Reel and Frame Number: 016945/0890
December 9, 2005 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., NET2PHONE, INC. 520 BROAD STREET, 8TH FLOOR NEWARK NEW JERSEY 07102, Reel and Frame Number: 017105/0240

LEGAL-REP: Kudirka & Jobse, LLP

PUB-TYPE: August 22, 2000 - Utility Patent having no previously published pre-grant publication (A)

PUB-COUNTRY: United States (US)

US-MAIN-CL: 709#227

US-ADDL-CL: 709#204

CL: 709

IPC-MAIN-CL: [7] G06F 013#38

IPC-ADDL-CL: [7] G06F 015#17

PRIM-EXMR: Rinehart; Mark H.

REF-CITED:
5095480March, 1992FennerUnited States (US)
5150360September, 1992Perlman et al.United States (US)
5166931November, 1992RiddleUnited States (US)
5204669April, 1993Dorfe et al.United States (US)
5224095June, 1993Woest et al.United States (US)
5291554March, 1994MoralesUnited States (US)
5309433May, 1994Cidon et al.United States (US)
5309437May, 1994Perlman et al.United States (US)
5321813June, 1994McMillen et al.United States (US)
5357571October, 1994BanwartUnited States (US)
5400335March, 1995YamadaUnited States (US)
5425028June, 1995Britton et al.United States (US)
5430709July, 1995GallowayUnited States (US)
5430727July, 1995CallonUnited States (US)
5434797July, 1995BarrisUnited States (US)
5442633August, 1995Perkins et al.United States (US)
5452296September, 1995ShimizuUnited States (US)
5455854October, 1995Dilts et al.United States (US)

5457683October, 1995RobinsUnited States (US)
5463625October, 1995YasrebiUnited States (US)
5469500November, 1995Satter et al.United States (US)
5479411December, 1995KleinUnited States (US)
5517494May, 1996GreenUnited States (US)
5524110June, 1996Danneels et al.United States (US)
5524254June, 1996Morgan et al.United States (US)395#500
5526489June, 1996Nilakantan et al.United States (US)
5533110July, 1996Pinard et al.United States (US)
5544303August, 1996Mraoteaux et al.United States (US)
5546582August, 1996Brockmeyer et al.United States (US)
5581552December, 1996Civanlar et al.United States (US)370#396
5608786March, 1997GordonUnited States (US)
5740231April, 1998Cohn et al.United States (US)
A 2 0445402November, 1991European Patent Office (EP)
A 2 0556012August, 1993European Patent Office (EP)
WO 9219054October, 1992World Intellectual Property Organization (WO)

NON-PATENT LITERATURE: December & Randall, "The World Wide Web Unleased,"Samw Publishing, Indianapolis, IN, Dec. 1994, pp. 3-24.
Gull, Re: Getting IP address of PPP-connected Mac, .
Gull, Re: Internet Phone for Mac?,>jgull-1704950116450001@pm049-28.dialip.mich.net>.
Heylighen, "WorldWideWeb: a distributed hypermedia paradigm for global networking," IEEE/INSPEC Database Updates and Additionss (1960-19950 Doc.# 134618: Proceedings SHARE Spring Conference, pp. 355-368, Apr. 1994.
Internetworking with TCP/IP, vol. 1, Second Edition, Principles, Protocols, and Architecture, by Douglas E. Comer.; 1991; table of contents, pp. 1-3, 17-19, 311-333.
VocalTec Internet Phone (TM) Version 2.5, www.cox.smu.edu/class/mis6386/people/stort/phone25.exe.
Weinberg, Netscape Conference and Cooltalk Meeting Room, www.q5.com.

CORE TERMS: processing, internet, user, point-to-point, protocol, e-mail, server, icon, input, callee, database, on-line, message, display, processor, session, screen, send, temporary, stored, memory, alternatively, computer, transmitting, secondary, mouse, mail, host, nnn, conjunction

ENGLISH-ABST:

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.

NO-OF-CLAIMS: 44

EXMPL-CLAIM: 1

NO-OF-FIGURES: 9

NO-DRWNG-PP: 6

SUMMARY:

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.TM., COMPUSERVE.RTM., 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.TM., 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.

Generally, devices interfacing the to 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 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", and each user may be allocated a successive IP address of XXX.XXX.XXX.10, XXX.XXX.XXX.11, 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.

DRWDESC:

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.

DETDESC:

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.TM. unit, available from NetSpeak Corporation, Boca Raton, Fla. 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. The processing units 12, 22 may be operatively interconnected through the Internet 24 to 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 Veiw, Calif. 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 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 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 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 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.TM. 3.1 available from MICROSOFT.TM. Corporation Redmond, Was. 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, Fla. 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 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 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 a connection service provider.

Upon the first user initiating the point-to-point Internet protocol when the first user is logged on to 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 a connection service provider, 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 N.sup.TH 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 N.sup.TH 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 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 649-2 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.

ENGLISH-CLAIMS:
Return to Top of Patent

1. A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network, the computer program product comprising: a computer usable medium having program code embodied in the medium, the program code comprising: program code for transmitting to the server a network protocol address received by the first process following connection to the computer network; program code for transmitting, to the server, a query as to whether the second process is connected to the computer network; program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network; and program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network.

2. An apparatus for enabling point-to-point communications between a first and a second process over a computer network, the apparatus comprising: a processor; a network interface, operatively coupled to the processor, for connecting the apparatus to the computer network; a memory, operatively coupled to the processor, for storing a network protocol address for selected of a plurality of processes, each network protocol address stored in the

memory following connection of a respective process to the computer network; means, responsive to a query from the first process, for determining the on-line status of the second process and for transmitting a network protocol address of the second process to the first process in response to a positive determination of the on-line status of the second process.

3. The computer server apparatus of claim 2 further comprising a timer, operatively coupled to the processor, for time stamping the network protocol addresses stored in the memory.

4. A method for enabling point-to-point communication between a first process and a second process over a computer network, the method comprising the steps of: A. receiving and storing into a computer memory a respective network protocol address for selected of a plurality of processes that have an on-line status with respect to the computer network, each of the network protocol addresses received following connection of the respective process to the computer network; B. receiving a query from the first process to determine the on-line status of the second process; C. determining the on-line status of the second process; and D. transmitting an indication of the on-line status of the second process to the first process over the computer network.

5. The method of claim 4 wherein step C further comprises the steps of: c.1 searching the computer memory for an entry relating the second process; and c.2 retrieving a network protocol address of the second process in response to a positive determination of the on-line status of the second process.

6. The method of claim 4 wherein step D further comprises the steps of: d.1 transmitting the network protocol address of the second process to the first process when the second process is determined in step C to have a positive on-line status with respect to the computer network.

7. The method of claim 4 wherein step D further comprises the steps of: d.1 generating an off-line message when the second process 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 process.

8. The method of claim 4 further comprising the steps of: E. receiving an E-mail signal comprising a first network protocol address from the first process; and F. transmitting the E-mail signal over the computer network to the second process.

9. The method of claim 8 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 process.

10. In a computer system, a method for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the method comprising the steps of: A. providing a user interface element representing a first communication line; B. providing a user interface element representing a first callee process; and C. establishing a point-to-point communication link from the caller process to the first callee process, in response to a user associating the element representing the first callee process with the element representing the first communication line.

11. The method of claim 10 wherein step C further comprises the steps of: c.1 querying the server as to the on-line status of the first callee process process; and c.2 receiving a network protocol address of the first callee process over the computer network from the server.

12. The method of claim 10 further comprising the step of: D. providing an element representing a second communication line.

13. The method of claim 12 further comprising the steps of: E. terminating the point-to-point communication link from the caller process to the first callee process, in response to the user disassociating the element representing the first callee process from the element representing the first communication line; and F. establishing a different point-to-point communication link from the caller process to the first callee process, in response to the user associating the element representing the first callee process with the element representing the second communication line.

14. The method of claim 10 further comprising the steps of: D. providing a user interface element representing a second callee process; and E. establishing a conference point-to-point communication link between the caller process and the first and second callee process, in response to the user associating the element representing the second callee process with the element representing the first communication line.

15. The method of claim 10 further comprising the step of: F. removing the second callee process from the conference point-to-point communication link in response to the user disassociating the element representing the second callee process from the element representing the first communication line.

16. The method of claim 10 further comprising the steps of: D. providing a user interface element representing a communication line having a temporarily disabled status; and E. temporarily disabling a point-to-point communication link between the caller process and the first callee process, in response to the user associating the element representing the first callee process with the element representing the communication line having a temporarily disabled status.

17. The method of claim 16 wherein the element provided in step D represents a communication line on hold status.

18. The method of claim 17 wherein the element provided in step D represents a communication line on mute status.

19. The method of claim 10 wherein the caller process further comprises a visual display and the user interface comprises a graphic user interface.

20. The method of claim 19 wherein the steps of establishing a point-to-point link as described in step C is performed in response to manipulation of the graphic elements on the graphic user interface.

21. A computer program product for use with a computer system comprising: a computer usable medium having program code embodied in the medium for establishing a point-to-point communication link from a caller process to a callee process over a computer network, the caller process having a user interface and being operatively connectable to the callee process and a server over the computer network, the medium further comprising: program code for generating an element representing a first communication line; program code for generating an element representing a first callee process; program code, responsive to a user associating the element representing the first callee process with the element representing the first communication line, for establishing a point-to-point communication link from the caller process to the first callee process.

22. The computer program product of claim 21 wherein the program code for establishing a point-to-point communication link further comprises: program code for querying the server as to the on-line status of the first callee process; and program code for receiving a network protocol address of the first callee process over the computer network from the server.

23. A computer program product of claim 21 further comprising: program code for generating an element representing a second communication line.

24. The computer program product of claim 23 further comprising: program code, responsive to the user disassociating the element representing the first callee process from the element representing the first communication line, for terminating the point-to-point communication link from the caller process to the first callee process; and program code responsive to the user associating the element representing the first callee process with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller process to the first callee process.

25. The computer program product of claim 21 further comprising: program code for generating an element representing a second callee process; and program code means, responsive to the user associating the element representing the second callee process with the element representing the first communication line, for establishing a conference communication link between the caller process and the first and second callee process.

26. The computer program product of claim 25 further comprising: program code, responsive to the user disassociating the element representing the second callee process from the element representing the first communication line, for removing the second callee process from the conference communication link.

27. The computer program product of claim 21 further comprising: program code for generating an element representing a communication line having a temporarily disabled status; and program code, responsive association of the element representing the first callee process with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller process and the first callee process.

28. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on hold status.

29. The computer program product of claim 27 wherein the communication line having a temporarily disabled status comprises a communication line on mute status.

30. A computer program product of claim 21 wherein the computer system further comprises a visual display and the user interface comprises a graphic user interface.

31. The computer program product of claim 30 wherein the element representing the first communication line and the element representing the first callee process are graphic elements and wherein the program code for establishing a point-to-point communication link from the caller process to the first callee process further comprises: program code, responsive to manipulation of the graphic elements on the graphic user interface, for establishing the point-to-point communication link from the caller process to the first callee process.

32. A method of locating a process over a computer network comprising the steps of: a. maintaining an Internet accessible list having a plurality of selected entries, each entry comprising an identifier and a corresponding Internet protocol address of a process currently connected to the Internet, the Internet Protocol address added to the list following connection of the process to the computer network; and b. in response to identification of one of the list entries by a requesting process, providing one of the identifier and the corresponding Internet protocol address of the identified entry to the requesting process.

33. A method for locating processes having dynamically assigned network protocol addresses over a computer network, the method comprising the steps of: a. maintaining, in a computer

memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and b. in response to identification of one of the entries by a requesting process providing one of the identifier and the network protocol address to the requesting process.

34. The method of claim 33 further comprising the step of: c. modifying the compilation of entries.

35. The method of claim 34 wherein step c further comprises: c.1 adding an entry to the compilation upon the occurrence of a predetermined event.

36. The method of claim 35 wherein the predetermined event comprises notification by a user process of an assigned network protocol address.

37. The method of claim 34 wherein step c further comprises: c.1 deleting an entry from the compilation upon the occurrence of a predetermined event.

38. A computer program product for use with a computer system having a memory and being operatively connectable over a computer network to one or more computer processes, the computer program product comprising a computer usable medium having program code embodied in the medium the program code comprising: a. program code configured to maintain, in the computer memory, a network accessible compilation of entries, selected of the entries comprising a network protocol address and a corresponding identifier of a process connected to the computer network, the network protocol address of the corresponding process assigned to the process upon connection to the computer network; and b. program code responsive to identification of one of the entries by a requesting process and configured to provide one of the identifier and the network protocol address to the requesting process.

39. The computer program product of claim 38 further comprising: c. program code configured to modify the compilation of entries.

40. The computer program product of claim 39 wherein program code configured to modify comprises: c.1 program code configured to add an entry to the compilation upon the occurrence of a predetermined event.

41. The computer program product of claim 40 wherein the predetermined event comprises notification by a process of an assigned network protocol address.

42. The computer program product of claim 38 wherein step c further comprises: c.1 program code configured to delete an entry from the compilation upon the occurrence of a predetermined event.

43. A computer program product for use with a computer system, the computer system executing a first process operatively coupled over a computer network to a second process and a server process, the computer program product comprising a computer usable medium having computer readable program code embodied therein, the program code comprising: a. program code configured to access a directory database, the database having a network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the network protocol address of each respective process forwarded to the database following connection to the computer network; and b. program code responsive to one of the network protocol addresses and configured to establish a point-to-point communication link from the first process to the second process over the computer network.

44. In a first computer process operatively coupled over a computer network to a second

process and an address server, a method of establishing a point-to-point communication between the first and second processes comprising the steps of: A. following connection of the first process to the computer network forwarding to the address server a network protocol address at which the first process is connected to the computer network; B. querying the address server as to whether the second process is connected to the computer network; C. receiving a network protocol address of the second process from the address server, when the second process is connected to the computer network; and D. in response to the network protocol address of the second process, establishing a point-to-point communication link with the second process over the computer network.

LOAD-DATE: December 3, 2007

Command Searching > **Patent Cases from Federal Courts and Administrative Materials**

Search

Enter Search Terms

(●) Terms and Connectors ( ) Natural Language ( ) Easy Search™

6108704 or 6,108,704

Suggest terms for my search

Check spelling

Restrict by Document Segment

Select a document segment, enter search terms for the segment, then click Add.

Select a Segment | Add

Note: Segment availability differs between sources. Segments may not be applied consistently across sources.

Restrict by Date

(●) No Date Restrictions ( ) From To Date formats...

Search Connectors

and	and	w/p	in same paragraph
or	or	w/seg	in same segment
w/N	within N words	w/s	in same sentence
pre/N	precedes by N words	and not	and not

- More Connectors & Commands...

How Do I...?

- Combine sources?
- Restrict by date?
- Restrict by document segment?
- Use wildcards as placeholders for one or more chara term?

View Tutorials

Source: Command Searching > **Patent Cases from Federal Courts and Administrative Materials**
Terms: **6108704 or 6,108,704** (Edit Search | Suggest Terms for My Search)

*2008 U.S. Dist. LEXIS 50451, **

NET2PHONE, INC., Plaintiff v. EBAY, INC., et al., Defendants

Civil Action 06-2469 (KSH)

UNITED STATES DISTRICT COURT FOR THE DISTRICT OF NEW JERSEY

2008 U.S. Dist. LEXIS 50451

June 25, 2008, Decided
June 26, 2008, Filed

NOTICE: NOT FOR PUBLICATION

SUBSEQUENT HISTORY: Later proceeding at Net2Phone, Inc. v. eBay, Inc., 2008 U.S. Dist. LEXIS 87521 (D.N.J., Oct. 27, 2008)

CORE TERMS: patent, privileged, log, disclosure, common interest, handwriting, declaration, legal interests, legal advice, email, attorney-client, untimely, waived, seal, de novo review, deadline, in-house, shareholder, valuation, privileged communication, monetization, anticipation of litigation, tender offer, confidential, negotiation, partial, embody, confidentiality, handwritten, conclusions of law

COUNSEL: [*1] For **HOWARD S. JONAS, Movant: JOSEPH P. LA SALA**, *LEAD ATTORNEY,* MCELROY, DEUTSCH, MULVANEY & CARPENTER, LLP, MORRISTOWN, NJ.

For **RONALD J. HEDGES, Mediator, Special Master: RONALD J. HEDGES**, *LEAD ATTORNEY,* NIXON PEABODY, NEW YORK, NY.

For **NET2PHONE, INC., Plaintiff: JOSEPH P. LA SALA**, **WILLIAM F. O'CONNOR, JR.**, *LEAD ATTORNEYS,* MCELROY, DEUTSCH, MULVANEY & CARPENTER, LLP, MORRISTOWN, NJ; **MARIA A. SAVIO**, GOTTLIEB RACKMAN & REISMAN, NEW YORK, NY.

For **EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC., Defendants: KATHLEEN M. FENNELLY**, **THOMAS R. CURTIN**, *LEAD ATTORNEYS,* GRAHAM, CURTIN, PA, MORRISTOWN, NJ.

For **EBAY, INC., SKYPE TECHNOLOGIES SA, SKYPE, INC., Counter Claimants: THOMAS R. CURTIN**, *LEAD ATTORNEY,* **KATHLEEN M. FENNELLY**, GRAHAM, CURTIN, PA, MORRISTOWN, NJ.

For **NET2PHONE, INC., Counter Defendant: JOSEPH P. LA SALA**, *LEAD ATTORNEY*, MCELROY, DEUTSCH, MULVANEY & CARPENTER, LLP, MORRISTOWN, NJ.

JUDGES: Patty Shwartz, United States Magistrate Judge.

OPINION BY: Patty Shwartz

OPINION

SHWARTZ, Magistrate Judge

This matter having come before the Court as a result of objections to the Report of the Special Master dated April 21, 2008 and the motion to seal documents submitted in connection with the objections. For the reasons set forth **[*2]** herein, the objections are overruled, the Report is adopted, and the motion to seal is denied.

I. PROCEDURAL HISTORY

On June 1, 2006, plaintiff Net2Phone, Inc. ("Net2Phone" or "plaintiff") filed a Complaint against defendants eBay, Inc. ("eBay"), Skype, Inc., Skype Technologies SA ("Skype"), and John Does 1-10 (collectively "defendants") alleging patent infringement [1] and violations of 35 U.S.C. § 271. See Compl. at P 1. Plaintiff filed its First Amended Complaint on June 7, 2006 [2] and followed with its Second Amended Complaint on June 28, 2006 adding additional patents to the lawsuit ("patents-in-suit"). [3] Defendants filed an Answer and Counterclaim denying infringement, validity, and enforceability of the patents-in-suit on September 15, 2006, see Docket Entry No. 17, and plaintiff filed a Response to the Counterclaim on September 20, 2006. See Docket Entry No. 18. Discovery has proceeded in accordance with various scheduling orders.

FOOTNOTES

1 The initial Complaint alleged violation of US. Patent No. **6,108,704**. See Compl. at P 14-22.

2 First Amended Complaint was re-filed on June 9, 2006. See Docket Entry No. 3.

3 In the Second Amended Complaint added alleged violations of U.S. Patent Nos. 6,701,365, **[*3]** 6,009,469, 6,131,121 and 6,226,678. See Second Am. Compl. at PP 23-62.

By way of letters dated November 14, 2007 and November 16, 2007, the parties advised the Court that plaintiff had designated over 1,000 documents as privileged or protected from disclosure under the work product rule and that defendant intended to challenge the majority of the plaintiff's designations. See Order Appointing Special Master at 1. After considering the parties submissions dated November 20, 2007, November 24, 2007, November 30, 2007, and December 3, 2007, the Court concluded that based on the volume of challenges and the likelihood that in camera inspection may be needed, the appointment of a Special Master under Fed. R. Civ. P. 53 to resolve the disputes was warranted. See id. at 1-2. The parties concurred in this assessment and, by way of Order dated December 7, 2007, the Court appointed Ronald J. Hedges as Special Master ("Special Master"), Docket Entry No. 146, and, by agreement of the parties, the parties agreed to limit any review of his findings to one level of appeal. Order Appointing Special Master at P 7.

The Special Master conducted five days of hearings, [4] reviewed documents in camera, and **[*4]** heard arguments concerning "hundreds of documents" as to which Net2Phone asserted privilege. Findings of Fact and Conclusions of Law of Special Master ("Report") dated April 21, 2008 at 3. In addition to these hearings, the Special Master considered the submissions made pursuant to the Order of September 27, 2007, including defendants' submission dated November 14, 2007 and plaintiff's submission dated November 16, 2007. See id. At the January 10, 2008 hearing, the Special Master also gave the plaintiff an opportunity to submit additional evidence to support its privilege claims even though the original submission deadline had passed. See Report at P 5. He advised that no further submissions would be allowed after January 15, 2008. See id. The plaintiff and defendants made additional submissions on April 8, 2008 and April 10, 2008 respectively, which the Special Master declined to consider because they were untimely. See id.

FOOTNOTES

4 These hearings took place on January 10, January 15, February 6, February 7, and March 24, 2008.

On April 21, 2008, the Special Master filed his report wherein he noted that plaintiff eventually produced 4,667 documents that it had previously withheld as privileged. **[*5]** Id. at 4. The Special Master also found other documents should be produced. Among other things, the Special Master concluded that, during an approximate one-year period, plaintiff and IDT "did not have any identity (or even similarity) of legal interests." Id. at 14. Accordingly, he found the "common interest" doctrine inapplicable to communications between the plaintiff and IDT and found that documents involving communications between plaintiff and IDT during the time period should be produced. Id. at 16.

The Special Master also concluded that the plaintiff and GE did not have the identical legal interest required for asserting the attorney-client privilege and their communications about a potential financing arrangement were not protected from disclosure. See id. at 22.

As to the valuation and infringement analyses, the plaintiff conceded that it voluntarily disclosed patent analyses and valuations of the patents-in-suit. Id. at 17. Accordingly, the Special Master concluded that the plaintiff waived the privilege over communications (other than those with trial counsel) concerning the following subjects: "(1) whether Skype infringes the NetSpeak patents; (2) whether the NetSpeak patents **[*6]** are easy to design around; (3) whether the NetSpeak patents are valid; (4) whether Vonage infringes the NetSpeak patents; (5) whether the PacketCable Specs require use of the NetSpeak patents; and (6) the value of the NetSpeak patents." Id. at 20. The Special Master observed that while the parties may disagree as to particular documents that fall within these subjects, such disagreement is "not yet ripe for judicial review." Id. at 21.

As to the disputes regarding privilege log entries 2623, 2629, 2632, 2633, 2634, 2645, 9062, 1861, 1864, 1870, 3814, 1870, 3814, 1142, 1332, 1333, 1337, 1840, 2783, 4562, 8832-33, 9061, 9073 and 4382, the Special Master concluded that the plaintiff failed to sustain its burden to show that the communications are entitled to protection under the attorney-client privilege or the work-product rule, or that any existing privilege has not been waived. [5] Id. at 5-12.

FOOTNOTES

5 The plaintiff did not appeal the Special Master's rulings regarding log entries 4638-39,

4675, 3893 and 1766.

On May 6, 2008, the plaintiff filed its Rule 53(f) Objections and Motion to Modify Findings of Fact and Conclusions of Law of the Special Master. See Docket Entry No. 224. Defendant filed **[*7]** a Response in Opposition to Net2Phone's Objections to the Report of the Special Master on May 21, 2008, see Docket Entry Nos. 232, 233, and plaintiff filed a reply on June 5, 2008. See Docket Entry Nos. 238, 239.

The plaintiff also filed a motion to seal certain documents submitted in connection with its objections. Plaintiff argues that good cause exists to seal the documents because they: (1) disclose a confidential arrangement between IDT and a third party to engage in joint patent enforcement, (2) disclose the name of a competitor against whom IDT was contemplating litigation, (3) contain confidential opinion of counsel on issues of patent infringement, (4) embody certain information that is designated for attorney's eyes only, and (5) contain the confidential agreement between Net2Phone and IDT. The plaintiff asserts that the fact that the Special Master's findings have been posted on the docket does not preclude the plaintiff from seeking to seal the information. In opposition, defendants argue that the plaintiff's motion to seal should be denied because: (1) the plaintiff's request is untimely, (2) none of the information it seeks to seal contains confidential information, and **[*8]** (3) the plaintiff has previously disclosed the information it now seeks to be sealed.

II. DISCUSSION

A. STANDARD OF REVIEW

The plaintiff argues that: (1) the Court must conduct a de novo review of the Special Master's findings of fact and conclusions of law, and (2) that the Court may review new evidence.

In response, the defendants argue that: (1) the Court should apply the "abuse of discretion" standard when reviewing the Special Master's procedural rulings regarding the discovery process, (2) the Court should show deference to the Special Master's findings on these matters, and (3) even if a de novo standard of review is appropriate, that standard does not permit plaintiff to present new arguments and evidence because allowing it to do so would undermine the purpose of the proceedings before the Special Master and would run counter to the interest of "fair and timely resolution of the issues."

In reply, plaintiff argues that Skype misstates the standard of review because: (1) Fed. R. Civ. P. 53 states that a Special Master's findings of fact and conclusions of law are to be reviewed de novo, and (2) pursuant to Rule 53, the Court may consider new arguments, documents, and evidence as **[*9]** part of its de novo review. [6]

FOOTNOTES

6 Plaintiff denies that they are attempting to raise new arguments, but points out that defendants are raising a whole new set of arguments related to preclusion.

Under Federal Rules of Civil Procedure 53(a)(1)(C), the Court may appoint a Special Master to "address pretrial and posttrial matters." Fed. R. Civ. P. 53(a)(1)(C). The Special Master must report his findings to the court that appointed him and serve a copy of his findings on each party. Fed. R. Civ. P. 53(e). The parties may appeal both the substantive and procedural findings of the Special Master. Fed. R. Civ. P. 53(f)(2); see, e.g., Commissariat A L'Energie Atomique v. Samsung Electronics Co., 245 F.R.D. 177, 179 (D.Del. 2007).

Here, the plaintiff appeals the Special Master's findings of fact, conclusions of law, and procedural decisions. The Special Master made rulings concerning the application of the attorney-client privilege. In this circuit, "the applicability of a privilege is a factual question" and "determining the scope of a privilege is a question of law." In re Bevill, Bresler, & Schulman Asset Management. Corp., 805 F.2d 120, 124 (3d Cir. 1986) (citing U.S. v. Liebman, 742 F.2d 807, 809 (3d Cir. 1984)). **[*10]** Objections to the Special Master's findings of fact and conclusions of law are reviewed de novo. Fed. R. Civ. P. 53(f)(3) and (4); see, e.g., Wachtel v. Guardian Life Ins. Co., Civ. Nos. 01-4183, 03.-1801, 2006 U.S. Dist. LEXIS 28879, 2006 WL 1320031, at *3 (D.N.J. May 11, 2006); accord In re Intel Corp. Mircoprocessor Antitrust Litigation, Civ. No. 05-485, 562 F. Supp. 2d 606, 2008 U.S. Dist. LEXIS 39642, 2008 WL 2156751, at *1 (D.Del May 14, 2008).

In conducting a de novo review of the Special Master's finding of facts and conclusions of law, the Court is mindful that a

> [d]e novo review . . . does not necessarily mean a review that includes the submission of new evidence, particularly when evidentiary proceedings previously occurred before the Special Master. When a record on review "is sufficiently developed the district court may, *in its discretion,* merely conduct a de novo review" of the decision, making its own independent determination. Although de novo review refers to the review based on the record below plus any additional evidence received by the reviewing court, it also refers to review of the decision based only on the record below. The plain language of Rule 53 shows that the review of a Special Master's decision requires the court to make a de novo **[*11]** determination, not conduct a de novo hearing. Rule 53 is similar to 28 U.S.C. § 636(b)(1)(C), when a district court reviews the recommendations of a magistrate judge, the district judge "may accept, reject, or modify" the findings made by the magistrate and "may receive further evidence." Unlike a de novo hearing, "a de novo determination requires the district judge to 'consider the record which has been developed before the magistrate [judge] and make his own determination on the basis of that record, without being bound to adopt the findings and conclusions of the magistrate [judge].'"

Commissariat a l'Energie Atomique v. Samsung Electronics Co., 245 F.R.D. 177, 179 (D. Del. 2007). "The phrase 'de novo determination' . . . means an independent determination of a controversy that accords no deference to any prior resolution of the same controversy." United States v. Raddatz, 447 U.S. 667, 690, 100 S. Ct. 2406, 65 L. Ed. 2d 424 (1980) (Stewart, J., dissenting)(citing United States v. First City Nat'l Bank, 386 U.S. 361, 368, 87 S. Ct. 1088, 18 L. Ed. 2d 151 (1967)). This, however, does not require the reviewing court to hear new arguments. In fact, courts generally "exclud[e] evidence of new arguments on objections . . . [because] [s]ystematic efficiencies **[*12]** would be frustrated and the [Special Master's] role reduced to a mere dress rehearser. . . . In addition, it would be fundamentally unfair to permit a litigant to set its case in motion before the [Special Master] . . . and -- having received an unfavorable recommendation -- shift gears before the [reviewing] judge." Dunkin' Donuts Franchised Restaurants LLC v. Mehta, Civ. No. 07-0423, 2007 U.S. Dist. LEXIS 67112, 2007 WL 2688710, at *1-2 (W.D.Pa. 2007) (citing Paterson-Leitch Co., Inc. v. Massachusetts Municipal Wholesale Electric Co., 840 F.2d 985, 991 (1st Cir. 1988)). [7] For these reasons, in an appeal of a Special Master's decision, the parties "cannot raise entirely new arguments for the first time on an objection to a Special Master's Report." World Triathalon Corp. v. Dunbar, 539 F. Supp. 2d 1270, 1278 n. 13 (D.Hawaii 2008) (citing Convolve, Inc. v. Compaq Computer Corp., Civ. No. 00-5141, 2004 U.S. Dist. LEXIS 17502, 2004 WL 1944834, at *1 (S.D.N.Y. Sept. 1, 2004)).

FOOTNOTES

7 Generally, on appeal "[t]he matter of what questions may be taken up and resolved for the first time on appeal is one left primarily to the discretion of the [reviewing court], to be exercised on the facts of individual cases." Singleton v. Wulff, 428 U.S. 106, 121, 96 S. Ct. 2868, 49 L. Ed. 2d 826 (1976). **[*13]** Appellate courts generally require exceptional circumstances in order to hear issues not presented in the court below. Harris Corp. v. Ericsson, Inc., 417 F.3d 1241, 1266 (Fed. Cir. 2005)

With respect to considering additional evidence, the reviewing Court has the discretion to consider additional facts or hear evidence itself if it is needed to make a de novo determination. See Raddatz, 447 U.S. at 692.

Challenges to the Special Master's rulings on procedural matters are only reviewed for an abuse of discretion. Fed. R. Civ. P. 53(f)(5); see, e.g., Wachtel, 2006 U.S. Dist. LEXIS 28879, 2006 WL 1320031, at *3; accord Gunter v. Ridgewood Energy Corp., 223 F.3d 190, 196-97 (3d Cir. 2000). Among other procedural rulings, the Special Master set deadlines for the presentation of submissions and evidence and precluded evidence not timely submitted.

i. Special Master's refusal to consider untimely submissions

The plaintiff asks the Court to consider evidence that it failed to present by the deadline that the Special Master had set for the presentation of new evidence. In effect, the plaintiff is asking the Court to overrule a procedural ruling of the Special Master. Here, the Special Master permitted the parties to submit **[*14]** certifications in support of their respective positions by January 10, 2008. The Special Master extended this deadline to January 15, 2008,.Report at 5, and he notified the parties that he would not consider any submissions after this date. Hr'g Tr. 119:22-120:8, Jan. 10, 2008; Hr'g Tr. 52:16-18, 136:10-20, Jan. 15, 2008. Despite this warning, the plaintiff attempted to submit additional materials on April 8, 2008. The Special Master's decision not to consider plaintiff's untimely submissions was not an abuse of his discretion. First, the plaintiff has not provided the Court with any reason why its submissions were so untimely. Second, the plaintiff has not presented any evidence that it sought an extension of the deadline for submissions from the Special Master. Third, the additional materials contain information that was in plaintiff's control before the deadline passed, namely evidence known to its witnesses about documents in existence years before the January 15 deadline. Fourth, the record indicates that the plaintiff missed the Special Master's deadline to submit additional materials by approximately four months, not a few days. It would be nearly impossible for the Special **[*15]** Master to conduct the privilege review and meet the Court imposed deadline to file a report if the parties did not comport with the deadlines for submissions. As such, the Special Master's refusal to consider submissions that were presented more than four months past his deadline is not an abuse of discretion. Thus, the Court will not disturb any of his rulings based upon his decision not to consider the late filed materials.

Moreover, the Court declines to consider any factual materials not timely presented to the Special Master. To allow this without any explanation as to why these materials were not timely presented would render the proceedings before the Special Master nothing more than a moot court exercise. The appointment of a Special Master was made to expedite the resolution of privilege disputes. To this end, the Special Master set deadlines for submissions so he could have a complete record on which to render his rulings. Like all evidentiary proceedings, at some point, the record must be closed. Allowing unending augmentation would mean that the decision-maker would never have the complete record upon which to render a final decision and the adverse party would be deprived **[*16]** of an opportunity to confront the new evidence. Here, the parties had ample opportunity to present evidence. They were on notice of the deadline and the consequences of noncompliance. Hr'g Tr. 52:16-20, Jan. 15, 2008. There is no reason to allow further augmentation of the record with evidence clearly available to the plaintiff at the time the Special Master set the original January 15, 2008 deadline. "[T]o do so in this situation," where there has been no explanation provided for the failure to comply, "would emasculate the purpose of the Special Master and Rule 53." Commissariat, 245 F.R.D. at 180.

For these reasons, the Court declines to consider evidence not timely presented to the Special Master.

ii. Consideration of New Arguments

As to the assertion that the plaintiff has presented new arguments, the Court declines to parse the submissions that were presented to the Special Master and those that are presented to this Court to determine whether new cases are now being presented because all of the arguments raised to the Special Master and to this Court (with the exception of the need to find unfairness before requiring disclosure of documents embodying topics for which there has **[*17]** been a waiver) embrace the same legal theories concerning the applicability of the privilege, the common interest doctrine, and waiver. To the extent additional cases are presented, the Court finds that there is no prejudice to any party if the Court considers these cases because each side has had an opportunity to address them. To the extent that the "unfairness" component of the waiver analysis was not argued, consideration of this issue is required but it does not lead to conclusions different from those the Special Master reached. See Convolve, 2004 U.S. Dist. LEXIS 17502, 2004 WL 1944834, at * 1.

The Court turns to its de novo review of the Special Master's findings of fact and conclusions of law.

B. Attorney Client Privilege

As a preliminary matter, the Court notes that because jurisdiction is based upon the presence of a federal question, the federal common law of privilege governs this matter. See Fed. R. Evid. 501; Harding v. Dana Transport, Inc., 914 F. Supp 1084, 1090 (D.N.J. 1996)(citing Wm. T. Thompson Co. v. General Nutrition Corp., Inc., 671 F.2d 100, 103 (3d Cir. 1982)).

The purpose of the attorney-client privilege is to encourage "full and frank communication between attorney and their clients." Upjohn Co. v. United States, 449 U.S. 383, 389, 101 S. Ct. 677, 66 L. Ed. 2d 584 (1981); **[*18]** Westinghouse Electric Corp. v. Republic of the Philippines, 951 F.2d 1414, 1423 (3d Cir. 1991). Because the attorney-client privilege obstructs the truth-finding process, however, it is construed narrowly and "protects only those disclosures -- necessary to obtain informed legal advice -- which might not have been made absent the privilege." Westinghouse, 951 F.2d at 1423-24 (quoting Fisher v. United States, 425 U.S. 391, 403, 96 S. Ct. 1569, 48 L. Ed. 2d 39 (1976)(emphasis in original)); Harding, 914 F. Supp at 1091 (stating "because the privilege obstructs the search for the truth and because its benefits are, at best, indirect and speculative, it must be strictly confined within the narrowest possible limits consistent with logic of its principle")(citations and internal quotations omitted).

The Court of Appeals for the Third Circuit states the traditional elements of the attorney client privilege as follows:

> (1) the asserted holder of the privilege is or sought to become a client;
>
> (2) the person to whom the communication was made
>
> > (a) is a member of the bar of a court, or his or her subordinate, and

(b) in connection with this communication is acting as a lawyer;

(3) the communication relates to a fact of which the attorney **[*19]** was informed

(a) by his client

(b) without the presence of strangers

(c) for the purpose of securing primarily either

(i) an opinion of law or

(ii) legal services or

(iii) assistance in some legal proceeding, and

(d) not for the purpose of committing a crime or tort; and

(4) the privilege has been

(a) claimed and

(b) not waived by the client.

Montgomery County v. MicroVote Corp., 175 F.3d 296, 301 (3d Cir. 1999); Rhone-Poulenc Rorer Inc. v. Home Indem. Co., 32 F.3d 851, 862 (3d Cir. 1994). A party asserting the privilege must show "(1) that it submitted confidential information to a lawyer, . . . (2) that it did so with the reasonable belief that the lawyer was acting as the parties' attorney," Montgomery Acad. v. Kohn, 50 F. Supp. 2d 344, 350 (D.N.J. 1999), and (3) the purpose of the communications was to secure legal, as opposed to business, advice. In re Ford Motor Co., 110 F.3d 954, 965 (3d Cir. 1997). It is, therefore, "vital to a claim of privilege that the communications between client and attorney were made in confidence and have been maintained in confidence." In re Howard Indus., Inc., 67 B.R. 291, 293 (Bankr. D.N.J. 1986) (quoting In re Horowitz, 482 F.2d 72, 81-82 (2d Cir. 1973)); **[*20]** see Republic of Philippines v. Westinghouse Electric Corp., 132 F.R.D. 384, 388 (D.N.J. 1990) (stating "a litigant who wishes to assert confidentiality must maintain genuine confidentiality")(citations omitted and emphasis in original). A party may waive the attorney-client privilege through various actions including purposeful disclosure, partial disclosure, and careless disclosure. Edna Epstein, The Attorney-Client Privilege and the Work-Product Doctrine 292-309 (American Bar Association 2001). Under the doctrine of waiver, when "[c]onduct touches a certain point of disclosure, fairness requires that the privilege shall cease whether he intended that result or not." 8 Wigmore, Evidence § 2327 at 636 (1961). Accordingly, a client generally waives the privilege if he or she voluntarily discloses the privileged communication to a third party, Westinghouse, 951 F.2d at 1424; In re Diet Drugs Prods. Liab. Lit., MDL No. 1203, 2001 U.S. Dist. LEXIS 5494, at *13 (E.D. Pa. April 19, 2001), or fails to take reasonable measures to ensure the confidentiality of communications with counsel. See Kaufman v. SunGard Invest. Sys., Civ. No. 05-1236, 2006 U.S. Dist. LEXIS 28149, 2006 WL 1307882, at *3 (D.N.J. May 9, 2006); Smithkline Beecham Corp. v Apotex Corp., 232 F.R.D. 467, 479 (E.D. Pa. 2005)(stating **[*21]** that mass dissemination of purportedly confidential communications can destroy an assertion of the privilege). [8]

FOOTNOTES

8 Similarly, the work-product privilege precludes disclosure of "materials prepared by an attorney, or an attorney's agent, in anticipation of or for litigation," as well as "[a]n attorney's mental impressions, conclusions, opinions or legal theories." In re Diet Drugs, 2001 U.S. Dist. LEXIS 5494, at *11 (citing In re Ford Motor Co. v. Kelly, 110 F.3d 954, 967 (3d Cir. 1997); see also U.S. v. Ernstoff, 183 F.R.D. 148, 153 (D.N.J. 1998); Fed. R. Civ. P. 26(b)(3)). The work product privilege may be waived and "[t]he predicate of the waiver inquiry in the work-product context . . . [is] whether the material was disclosed to an adversary." Maldonado v. New Jersey ex rel. Administrative Office of Courts-Probation Division, 225 F.R.D. 120, 131-32 (D.N.J. 2004). "The essential question with respect to waiver of the work-product privilege by disclosure is whether the material has been kept away from adversaries." Id. (citing Nicholas v. Wyndham Int'l, Inc., Civ. No. 01-147, 2003 WL 23198845, at *3-4, 2003 U.S. Dist. LEXIS 24086, at *9 (D.V.I. May 19, 2003)). "The party seeking to **[*22]** obtain protected work product bears the burden of proving that the protection has been waived." Hatco Corp. v. W.R. Grace & Co.-Conn., Civ. No. 89-1031, 1991 U.S. Dist. LEXIS 6479, 1991 WL 83126, at *7 (D.N.J. May 10, 1991). A showing of disclosure to a third party does not result in a waiver of the work product protection if the parties have common interests. Id.

C. Common Interest Privilege

Here, defendants assert that plaintiff did not share the attorney-client privilege with either IDT or GE and therefore their communications with the plaintiff are not protected from disclosure. Plaintiff argues that, at the time of the communications, it had a common interest with these third parties and their communications are privileged. The "common interest privilege is an extension of the attorney-client privilege and work product doctrine," Block Drug Company, Inc. v. Sedona Laboratories, Inc., Civ. No. 06-350, 2007 U.S. Dist. LEXIS 29028, at *3 (D. Del. Apr. 19, 2007), and thus is "an exception to the general rule that the [] privilege is waived upon disclosure of privileged information to a third party." Katz v. AT&T Corp., 191 F.R.D. 433, 436 (E.D. Pa., 2000) (citing In re The Regents of the University of California, 101 F.3d 1386, 1390 (Fed. Cir. 1996)).

Under **[*23]** the common interest doctrine, "although an attorney actually represents only one party, there is no waiver of the attorney-client privilege by disclosure of privileged communications to third parties with a 'community of interest.'" [9] Pittston Co. v. Allianz Ins. Co., 143 F.R.D. 66, 69 (D.N.J. 1992). Parties have a "community of interest" where they "have an identical legal interest with respect to the subject matter of a communication between an attorney and client concerning legal advice. . . . The key consideration is that the nature of the interest be identical, not similar, and be legal, not solely commercial." Id. (citing Duplan Corp. v. Deering Milliken Inc., 397 F. Supp. 1146, 1172 (S.D.S.C. 1974); In re The Regents of Univ. of Cal., 101 F.3d at 1390; In re Diet Drugs, 2001 U.S. Dist. LEXIS 5494, at *14 (stating that the doctrine preserves a privilege where persons or companies "share a common legal interest in a legal issue or exchange privileged communications with one another"). For the doctrine to apply, the parties must have "an identical legal interest with respect to the subject matter of the communication . . . ." Id.; **[*24]** Grider v. Keystone Health Plan Central, Inc., Civ. No. 05-MC-40, 2005 U.S. Dist. LEXIS 44069, at *21 (M.D. Pa. July 28, 2005). Thus, under "the common interest doctrine . . .'parties with shared interest in actual or potential litigation against a common adversary may share privileged information without waiving their right to assert the privilege.'" [10] Katz, 191 F.R.D. at 437 (quoting Thompson v. Glenmede Trust Co., Civ. No. 92-5233, 1995 U.S. Dist. LEXIS 18780, at *15 (E.D. Pa. Dec. 18, 1995)); see also Hewlett-Packard Co. v. Bausch & Lomb, Inc., 115

F.R.D. 308, 309-10 (N.D. Cal. 1987)(applying the doctrine where the communication is in anticipation of a joint litigation). The doctrine, however, does not apply where the third party's interest "'does not appear to be that of a potential co-defendant in a possible . . . action' . . . but rather [is] that of an 'adverse [party], negotiating at arm's length a business transaction between themselves.'" Nidec Corp. v. Victor Comp. Of Japan, Civ. No. 05-0686, 2007 U.S. Dist. LEXIS 48841, at *13-14 (N.D. Cal. July 3, 2007)(quoting SCM Corp. v. Xerox Corp., 70 F.R.D. 508, 512-13 (D. Conn. 1976)). In short, to assert the common interest doctrine, plaintiff must show: (1) the material is privileged, **[*25]** Grider, 2005 U.S. Dist. LEXIS 44069, at *20 (stating that "[t]he common interest privilege 'does not create an independent privilege, but depends upon a proper showing of the other elements of' . . . [a] recognized privilege before it will apply"), (2) "the parties had an identical legal and not solely commercial interest," In re The Regents of Univ. of Cal., 101 F.3d at 1390; Katz, 191 F.R.D. at 438, and (3) the communication was designed to further the shared legal interest. Nidec Corp., 2007 U.S. Dist. LEXIS 48841, at *10-11. Here, the Court will assume, without deciding, the first prong is met and the materials plaintiff seeks to withhold would be privileged if disclosed between an attorney and a client. As such, the Court turns to consider whether or not the remaining two prongs are met.

FOOTNOTES

9 Remington Arms Co. v. Liberty Mutual Insurance Co., 142 F.R.D. 408, 418 (D. Del. 1992) (declining to apply the common interest doctrine because "the rationale which supports the common interest exception to the attorney-client privilege simply doesn't apply if the attorney never represented the party seeking the allegedly privileged materials."); see also Pittston Co., 143 F.R.D. at 70.

10 Of course, **[*26]** "[e]ven if there were a common legal interest, the common interest exception requires that the communication at issue be 'designed to further that [legal] effort.'" Nidec Corp. v. Victor Comp. Of Japan, Civ. No. 05-0686, 2007 U.S. Dist. LEXIS 48841, at *15 (N.D. Cal. July 3, 2007)(quoting United States v. Bergonzi, 216 F.R.D. 487, 495 (N.D. Cal. 2003) (alteration and emphasis in original)).

I. Communications between Net2Phone and IDT

Plaintiff argues that the common interest privilege protects communications between plaintiff and IDT because: (1) plaintiff and IDT were closely affiliated companies with identical legal interests in preserving plaintiff's intellectual property, (2) the limited adversity between plaintiff and IDT as to a tender offer did not waive privilege on all issues, and (3) the common interest privilege between plaintiff and IDT arose separately from the IP Agreement.

The defendants argue that the common interest privilege does not apply to IDT and plaintiff between April 4, 2005 and March 13, 2006 because: (1) plaintiff and IDT were separate, publicly-traded corporations, (2) they did not share a common interest because their communications were not between a parent **[*27]** corporation and its wholly-owned subsidiary, (3) IDT did not then have any interest in the patents-in-suit, (4) the lack of adversity between the two corporations on certain topics does not necessarily imply they shared identical legal interests in those areas, (5) there is no evidence that either party to the communications considered the relationship between them to be that of attorney-client, (6) after the termination of the Intellectual Property Legal Services Agreement and before the acquisition, there was no existing legal relationship between the plaintiff and IDT that would permit application of the common interest privilege as reflected by plaintiff's use of its own counsel to enforce its patent portfolio, instead of IDT's attorneys, (7) IDT and plaintiff never entered into an agreement or placed any confidentiality restrictions on each other, (8) the information was disclosed for commercial purposes, and not to form a joint defense, and

(9) the plaintiff has failed to show that joint legal activity between plaintiff and IDT was likely.

Here, a de novo review shows that the Special Master's conclusion that the communications between plaintiff and IDT between the dates April **[*28]** 4, 2005 and March 12, 2006 are not subject to the common interest privilege doctrine is correct. First, there was no common legal interest between plaintiff and IDT during the dates of April 4, 2005 and March 12, 2006. See Stott-Bumsted Dec. Ex. 5 ("S.E.C. Schedule 14D-9") at 28-35. [11] April 4, 2005 is the date plaintiff's termination of its contractual relationship with IDT became effective. Id. at 28. March 13, 2006 is the date that IDT's acquisition of Net2Phone became effective. Thus, from April 4, 2005 to March 13, 2006, the parties were and functioned as separate, publically traded companies. See id. at 28-35. Indeed, after April 4, 2005, the independent committee of plaintiff affirmed that it was "not aware of any other arrangement that gave IDT any interest in the Netspeak patents, other than as an indirect interest as a shareholder of Net2Phone. Accordingly, we understand that no agreement exists between IDT and Net2Phone that gives IDT an interest in the Netspeak patents." Id. at 29. Second, from April 4, 2005 to March 13, 2006, IDT and Net2Phone had adverse interests because IDT and Net2Phone were negotiating the price IDT would pay for Net2Phone's shares. See id. Indeed, **[*29]** the plaintiff concedes that IDT was adverse to Net2Phone, but argues that it was only on the issue of the price IDT would pay for Net2Phone shares. The fact that the parties were adverse in the price per share for Net2Phone illustrates that the parties were indeed separate entities negotiating at arms length in a commercial transaction. Corning, Inc. v. SRU Biosystems, LLC, 223 F.R.D. 189, 190 (D.Del. 2004); SCM Corp., 70 F.R.D. at 525, Nidec Corp., 2007 U.S. Dist. LEXIS 48841, at *13-14 (citing cases); see also Katz, 191 F.R.D. at 438 (no common interest between parties before reaching licensing agreement because such negotiations do show an identity of legal interests). Third, the relationship between plaintiff and IDT was that of a corporation and its controlling shareholder. Simply because in-house counsel enforced the corporation's patents, which would benefit its shareholders, does not mean that they shared a legal interest. Put differently, a legal interest cannot arise simply because a company acts in a way that advances the economic interests of its majority shareholder. A logical extension of plaintiff's argument would expand the application of the common interest doctrine **[*30]** to cover all business transactions where a company acted in the interest of its majority shareholder. While shareholders and the corporation may share an interest in commercial success, this shared economic interest is not a legal interest. Moreover, IDT's direct contractual interest in the plaintiff's patents ended when the intellectual property agreement ended. In short, during this period, these legally separate entities had no mutual obligation and were engaging in negotiations to change their commercial relationship and they then shared no common legal interest. Their separate interests on legal issues is demonstrated by the representations to the SEC that plaintiff retained counsel for services that IDT had formerly provided. S.E.C. Schedule 14D at 13, 28-29. Finally, there is no indication that the communications associated with the tender offer were disclosed to further a common legal strategy or joint interest in pending or anticipated litigation. Rather, the information was shared to further a commercial transaction between legally separate entities. Nidec Corp., 2007 U.S. Dist. LEXIS 48841, at *15-16.

FOOTNOTES

11 Exhibit 5 is a Schedule 14D-9 Solicitation/Recommendation Statement filed **[*31]** by IDT with the Securities and Exchange Commission detailing to its shareholders the tender offer for the outstanding Net2Phone shares.

The cases plaintiff embraces do not change this result. Plaintiff's reliance on In re Teleglobe Communications Corp., 493 F.3d 345 (3d Cir. 2007), for the proposition that parent and subsidiary corporations are joint clients and thus afforded the common interest privilege is unpersuasive because: (1) the Teleglobe court applied Delaware state law, rather than Federal common law; and (2) the communications in Teleglobe were between parent-subsidiary and not between the corporation and its majority shareholder. For these reasons, Teleglobe does not change the analysis.

In addition, Hewlett-Packard Co., 115 F.R.D. at 310, does not advance the plaintiff's position as the facts in Hewlett-Packard Co. are distinguishable. In finding a common interest privilege between two parties, the court in Hewlett-Packard Co. observed that the defendant and its prospective business partner shared information pursuant to a confidentiality agreement. Moreover, the court observed that each faced litigation from the same plaintiff, and "[i]n such a lawsuit[,] defendant would **[*32]** be defending its marketing of the product in the years preceding the sale and GEC would be defending its marketing of exactly the same product in the years following the sale. Thus, at the time defendant and GEC were negotiating it seemed quite likely that defendant and GEC would be sued by plaintiff and that in that litigation defendant and GEC would be identically aligned, fighting to protect interests distinguished only by the time frame in which the marketing took place." Id. For these reasons, the court opined that the defendant and GEC would likely pursue a joint defense in defending the patent claims. Id. Here, there is nothing to show that plaintiff and IDT shared information under a confidentiality agreement nor is there evidence that at the time of the communication IDT and plaintiff faced the prospect of imminent litigation or a common adversary.

For these reasons, and on a de novo review, the Court finds that IDT and Net2Phone did not have a common legal interest during the period April 4, 2005 through March 13, 2006 and the Special Master's conclusions are adopted.

II. Communications between Net2Phone and GE

At some point during 2005, IDT and GE contemplated partnering to **[*33]** enforce IDT's patent portfolio through litigation or licensing. See Stott-Bumsted Dec. Ex. 17 (Email from James DiGiorgio, senior counsel to IDT, to David Greenblatt and "Ldiaz"). The transaction was to be structured as a loan from GE to IDT in the amount of a hundred million dollars. Id. The loan was to be repaid from the proceeds of the licencing/enforcement of the NetSpeak patent portfolio. Id. Any licencing/enforcement revenue above a hundred million dollars would be shared by GE and IDT on a pre-defined basis. Id.

The plaintiff argues that the communications between IDT and GE about joint enforcement of the NetSpeak patents are subject to the common interest privilege because the contemplated relationship between GE and IDT was not limited to a commercial transaction but rather involved both parties having an identical legal interest in the enforcement of the patents at issue. Finally, plaintiff argues that the fact that GE and IDT did not consummate their negotiations is irrelevant because the appropriate standard is whether or not the parties contemplated joint legal action, which GE and IDT did. Plaintiff also points out that the parties intended to keep their communications **[*34]** confidential, which is a hallmark of the attorney-client privilege.

The defendant argues that the common interest privilege does not apply to the IDT-GE communications because: (1) the parties did not have a common legal interest, (2) the agreement contemplated between the parties did not move past the negotiation stage, (3) the parties did not execute a confidentiality agreement, and (4) the cases the plaintiff relies on are distinguishable.

Here, it is undisputed that IDT and GE had discussed an agreement where GE proposed to partner with IDT to enforce the patents through litigation or licensing. Id. This proposed business arrangement was to be configured as a loan, in which GE would lend IDT money which would be re-paid with the proceeds from any fruitful litigation or licensing agreements.

Id. The interest here was commercial not legal. First, the arrangement between the parties was a proposed financing arrangement between independent entities. At the time of the negotiations, their interest was commercial and their communications during the negotiations were to further that interest and not a legal position. Second, at the time of the negotiations, GE was not a licensee, potential **[*35]** licensee, or owner of the patent. Third, although GE maintained the information received and shared with IDT in confidence, Stott-Bumsted Dec. Ex. 18 at P 5, the plaintiff has failed to show that there was a strict confidentiality agreement to do so or that their negotiations were conducted to advance a legal rather than a commercial interest.

Moreover, and as discussed above, Hewlett-Packard Co., 115 F.R.D. at 310, does not advance the plaintiff's position. Here, plaintiff and GE did not face the prospect of imminent litigation. There was neither a threat of impending legal action against them nor was there a common adversary. Rather, GE and plaintiff were negotiating a business transaction whereby GE would loan plaintiff money that would be repaid through patent enforcement actions or licensing of patents. Had the agreement come to pass, then communications to further the enforcement activity may have been protectable but the purpose of the communications during the negotiations were to entice a third-party to loan plaintiff money and not to further a then-shared legal interest. For these reasons, the common interest doctrine does not cover the communications between plaintiff and **[*36]** GE and the conclusions of the Special Master are adopted.

Having determined that communications between IDT and plaintiff during the period April 4, 2006 and March 13, 2006 and the communications between plaintiff and GE are not privileged, the documents reflecting these communications must be disclosed.

III. Waiver

The Court next considers whether or not disclosure to these entities and disclosures to plaintiff's shareholders waives the privilege asserted over communications concerning the same topics. Generally, privileged material disclosed to a third party waives the privilege. Westinghouse Electric Corp., 951 F.2d at 1425; Bulow v. Bulow, 828 F.2d 94, 103 (2d Cir. 1987). The Third Circuit has identified two distinct forms of "limited" waiver: selective waiver and partial waiver. Westinghouse, 951 F.2d at 1423 n.7. Selective waiver "permits the client who has disclosed privileged communications to one party to continue asserting the privilege against other parties" whereas partial waiver "permits a client who has disclosed a portion of privileged communications to continue asserting the privilege as to the remaining portions of the same communications." Id. (citations omitted). While **[*37]** fairness is not a consideration in selective waiver cases, it is a "central element" of a court's determination where partial waiver is invoked. Harding, 914 F.Supp at 1092; see also Westinghouse, 951 F.2d at 1426 (stating that "[g]enerally, the 'fairness doctrine' is invoked in partial (as opposed to selective) disclosure cases"). With a partial waiver, "the privilege is waived only as to the communication actually disclosed unless a partial waiver would be unfair to the party's adversary." Westinghouse, 951 F.2d at 1426 n.13; Wachtel, 2006 U.S. Dist. LEXIS 27591, 2006 WL 1286188, at *1 n. 2; In re Intel Corp., 2008 WL 2310288, at *10; In re Linerboard Antitrust Litigation, 237 F.R.D. 373, 388 (D.Pa. 2006). The fairness component seeks to "prevent prejudice to a party and distortion of the judicial process that may be caused by the privilege holder's selective disclosures . . . of otherwise privileged information." In re Intel Corp., 2008 WL 2310288, at *10 (citations omitted). A waiver can occur when a party attempts to use the communication in a litigation or where the party "makes factual assertions, the truth of which can only be assessed by examination of the privileged communications." Id. at 11. As the **[*38]** Intel court observed concerning disclosure of a report about document production, by disclosing summaries of a report, it "placed the accuracy and validity of the information contained in these summaries at issue." Id. at 12. The Intel court reasoned that to conclude otherwise would enable Intel to assert facts as a sword and shield the adversary from challenging the accuracy of the assertions.

The plaintiff objects to the Special Master's conclusion that it waived privilege on the subjects of whether: (1) Skype infringes the NetSpeak patents, (2) NetSpeak patents are easy to design around, (3) NetSpeak patents are valid, (4) Vonage infringes the NetSpeak patents, (5) PacketCable Specs require the use of the NetSpeak patents, and (6) the value of the NetSpeak patents. The plaintiff argues that: (1) disclosure of a few communications between the parties should not result in subject matter waiver in the aforementioned topics; (2) according to the Court of Appeals for the Third Circuit, privilege is waived only as to those communications actually disclosed unless a partial waiver would be unfair to the party's adversary, and there has been no showing of unfairness nor prejudice here, (3) **[*39]** the plaintiff is willing to alleviate any fear of future prejudice to the defendants by agreeing not to "affirmatively rely on any of the partial disclosures at issue at any future stage in litigation," Pl. Reply Br. at 8, and (4) the subject matter waiver defendants demand would unfairly prejudice the plaintiff in other litigations because the scope of the Special Master's conclusions were not necessarily limited to the patents at issue in the present litigation.

The defendants argue that the Special Master correctly interpreted the scope of the subject matter waiver regarding the analyses and valuations of patents-in-suit. The defendants assert that: (1) they need not show prejudice under Third Circuit law and, in any event, defendants have been prejudiced by the plaintiff's withholding of documents because the plaintiff has selectively disclosed documents to support its argument while withholding others on the same subject that may contradict its position, (2) the Special Master correctly imposed the appropriate limitation on the subject matter waiver when he determined that the waiver covers all documents except communications with trial counsel, and (3) that the plaintiff's complaint **[*40]** regarding temporal limitations to the waiver should be rejected because it is untimely.

A de novo review of the scope of the waivers that resulted from disclosures of the Monetization Plan, summaries of the CRA Report, and opinions of Douglas Derwin demonstrates that the Special Master's conclusions are correct.

1. Monetization Plan and CRA Report [12]

FOOTNOTES

12 Attached as Exhibit A to the Declaration of Hannah Stott-Bumsted, dated May 5, 2008.

According to the plaintiff, IDT's and plaintiff's attorneys prepared and presented a Monetization Plan to shareholders, which discussed, among other things, the implementation of a licensing and patent sale strategy. The plan was also used during the tender offer negotiation to provide information about the value of the patents. During the tender offer negotiations, plaintiff's counsel also obtained a report from CRA International concerning the value of the patents. The report's conclusion was disclosed to the shareholders because the valuation was material to their decision about tendering their shares but the analysis was not disclosed.

A de novo review shows that the plaintiff waived its assertion of privilege concerning valuation. The plaintiff affirmatively **[*41]** disclosed valuation information when it advanced its interest. As to the Monetization Plan, it was publicly disclosed in IDT's 14-D9 Securities and Exchange filing in connection with its tender offer. S.E.C. Schedule 14D-9 at 28. Similarly, the conclusions in the CRA Report were disclosed to shareholders and referred to in IDT's 14-D9 Securities and Exchanged filing. Id. at 31, 44-45. There is no dispute that the topic of valuation was widely disseminated. Thus, at a minimum, the plaintiff engaged in a selective waiver when it disclosed the Monetization Report and the conclusions in the CRA report to its shareholders and others with an interest in the tender offer. Fairness is not a consideration in selective waiver cases, and thus the Special Master was correct in declining to undertake a fairness analysis.

Moreover, there is at least a partial waiver as it relates to the CRA report. Plaintiff disclosed the conclusions but not the analysis or reasoning for the conclusions. If viewed as a partial waiver, the Court must consider whether or not it would be unfair to the defendant to allow the plaintiff to withhold the remainder of the report. The Court finds that it would be unfair to **[*42]** allow the plaintiff to withhold the analysis portion of the report.

First, the plaintiff seeks to rely on valuation to make arguments concerning damages. Even if the Monetization Plan or CRA Report are not offered affirmatively in evidence, plaintiff embraced them in the context of a commercial event and defendants should have an opportunity to investigate the bases for valuations contained in these documents and challenge the plaintiff's present valuation position with them. It would be unfair to allow plaintiff's to take one position in one context to advance its commercial purposes and preclude defendants from seeing if it took a different position in litigation. The defendants should be able to impeach plaintiff with its own statements or those it embraced on this topic. See V. Mane Fils S.A. v. Int'l Flavors and Fragrances, 249 F.R.D. 152, 2008 WL 619207, at *3 (D.N.J. 2008). Second, the plaintiff's representation that they do not intend to use these documents is insufficient. Plaintiff has not abandoned a desire to offer evidence about the value of its patents and thus the subject to which the documents relate is still present in this case and should be documents that **[*43]** defendants can examine. For all of these reasons, the Special Master's subject-matter waiver finding regarding valuation is correct.

2. Patent Opinions

IDT and plaintiff disclosed patent opinions outside the attorney-client relationship. According to the plaintiff, IDT retained Doug Derwin to evaluate a lawsuit against Vonage for infringement of the NetSpeak patents. In an October 2005 email, Doug Derwin disclosed his analysis to third parties about whether or not certain products infringe on plaintiff's patents, see Stott-Bumsted Dec. Ex. 4 at 1-2 (Email from Ely D. Tendler, Chief Legal Officer for IDT, to Doug Derwin and Abbe L. Dienstag) and provided an opinion that NetSpeak patents are easy to design around. Id. at 2. Derwin's views were then discussed by various third parties at a meeting in October 2005 consisting of legal counsel, consultants and technical personnel for both plaintiff and IDT at a time when plaintiff and IDT had no legal relationship beyond IDT's ownership of some of plaintiff's shares and at a time they had no shared legal interests. S.E.C. Schedule 14D at 31. Patent opinions were also embodied in the Monetization Plan. Specifically, the Monetization Plan included **[*44]** information pertaining to whether the NetSpeak patents are valid and whether the PacketCable Specs require use of the NetSpeak patents. S.E.C. Schedule 14D at 29. Plaintiff contends these disclosures were made to "an affiliate" in the context of a tender offer and the disclosure does not waive the privilege over other communications on the same subject and the Special Master's order directing disclosure was wrong because there was no finding of prejudice from nondisclosure of other communications on this subject.

When Mr. Derwin made his disclosures, the plaintiff and IDT did not share a common interest. Moreover, Mr. Derwin was acting only on behalf of IDT when he made his disclosures. Thus, by announcing his patent infringement opinions beyond IDT, the privilege has been waived on these subjects. Similarly, the Monetization Plan, which includes comments about the Net2Speak patents, was widely disseminated. Allowing plaintiff to withhold other communications on this subject would be unfair to the defendants. Defendants should be able to counter plaintiff's attempts to undermine Mr. Derwin's opinion and be confronted with their own views as announced in the Monetization Plan, particularly **[*45]** if the plaintiff attempts to distance itself from these opinions in this litigation about patents involving a similar technology.

Based upon these disclosures and the prejudice to the defendants by limiting the disclosure to the actual communication plaintiff conveyed to IDT and plaintiff's shareholders, and upon consideration of the subjects implicated by these disclosures, the Court finds that the Special Master correctly found that the plaintiff waived the privilege to the following topics: (1) whether Skype infringes the NetSpeak patents; (2) whether the NetSpeak patents are easy to design around; (3) whether the NetSpeak patents are valid; (4) whether Vonage infringes the NetSpeak patents; (5) whether the PacketCable Specs require use of the NetSpeak patents; and (6) the value of the NetSpeak patents. The absence of temporal limits to the scope of the waiver is consistent with the fact that certain of the disclosures do not have temporal limits. The Court notes that the Special Master imposed a temporal limitation on communication between IDT and the plaintiff and this reflects he was mindful of the applicability of such limits when appropriate.

For all of these reasons, the Court **[*46]** overrules that plaintiff's privilege assertion over responsive documents embodying: (1) communications between plaintiff and IDT during the period April 4, 2005 through March 16, 2006; (2) communications between plaintiff and GE; and (3) communications (except those with trial counsel) falling within the following categories: (a) whether Skype infringes the NetSpeak patents; (b) whether the NetSpeak patents are easy to design around; (c) whether the NetSpeak patents are valid; (d) whether Vonage infringes the NetSpeak patents; (e) whether the PacketCable Specs require use of the NetSpeak patents; and (f) the value of the NetSpeak patents. The plaintiff shall produce the withheld documents no later than June 30, 2008.

D. Specific Documents

The plaintiff also objects to the Special Master's privilege rulings concerning specific documents. The plaintiff bears the burden to prove that any document that does not appear privileged on its face is in fact privileged material. To this end, it must present evidence about the identity of the author of the document and the reason for its creation. For the reasons set forth herein, plaintiff has not met its burden.

i. Entry 2623 (Exhibit D) [13]

FOOTNOTES

13 The **[*47]** Exhibits are attached to the Declaration of Hannah Stott-Bumsted, dated May 5, 2008.

The Special Master concluded that the handwriting on log entry 2623 is not subject to privilege. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, the substance of the writing indicates that it contains legal impressions. The plaintiff asserts that the two sentences handwritten at the top of the first page reflects a legal comparison of the subject matter of the underlying document with another patent. The defendants argue that the Special Master reviewed the handwritten notes and did not err when he concluded that the privilege did not apply to the handwriting on entry 2623 because plaintiff failed to sustain its burden of proof and was correct in refusing to consider the Declaration of Joseph John, a senior technical director at during the relevant period, because it was submitted more than two months after the deadline.

The privilege log describes the document as embodying work product [14] and lists the author as Oblon Spivek. The plaintiff concedes, however, that it does not know the identity

[*48] of the author of the handwriting. Pl. Br. at 24-25. Moreover, its effort to prove the identity circumstantially through the Declaration of Joseph John dated March 13, 2008 fails because it was submitted approximately two months after the Special Master's January 15, 2008 deadline. Stott-Bumsted Dec. Ex. 19 ("Joseph John's Dec."). Because it was not timely submitted, the Court will not consider it. Moreover, the record silent as to whether or not the document was prepared in anticipation of litigation and for no other purpose, which is critical to sustaining the assertion of work product. Without the identity of the author and the purpose for which the writings were made, the plaintiff cannot establish that the writings on these documents are privileged or protected by the work product rule.

FOOTNOTES

14 Rule 26 (b)(3) of the Federal Rules of Civil Procedure provides, in relevant part:

> (A) Documents and Tangible Things. Ordinarily, a party may not discover documents and tangible things that are prepared in anticipation of litigation or for trial by or for another party or its representative (including the other party's attorney, consultant, surety, indemnitor, insurer, or agent). But, subject to **[*49]** Rule 26(b)(4), those materials may be discovered if:
>
> (i) they are otherwise discoverable under Rule 26(b)(1); and
>
> (ii) the party shows that it has substantial need for the materials to prepare its case and cannot, without undue hardship, obtain their substantial equivalent by other means.
>
> (B) Protection Against Disclosure. If the court orders discovery of those materials, it must protect against disclosure of the mental impressions, conclusions, opinions, or legal theories of a party's attorney or other representative concerning the litigation.

Fed. R. Civ. P. 26(b)(3). Rule 26(b)(3) essentially establishes "two tiers of protection: first, work prepared in anticipation of litigation by an attorney or his agent is discoverable only upon a showing of need and hardship; second, 'core' or 'opinion' work product that encompasses the 'mental impressions, conclusions, opinion, or legal theories of an attorney or other representative of a party concerning the litigation' is 'generally afforded near absolute protection from discovery.'" In re Cendant Corp. Sec. Litig., 343 F.3d 658, 663 (3d Cir. 2003) (quoting United States v. Nobles, 422 U.S. 225, 238-239, 95 S. Ct. 2160, 45 L. Ed. 2d 141 (1975)). As discussed by District Judge Stanley Chessler **[*50]** in In re Gabapentin Patent Litigation, 214 F.R.D. 178 (D.N.J. 2003),

> Courts generally, and in this Circuit in particular, have applied what amounts to a two part test for ascertaining whether the documents (or things) at issue should be protected under the . . . work product privilege. The first prong of the inquiry is the "reasonable anticipation" test, which requires that the court determine at what point in time litigation could reasonably have been anticipated. Whether a particular document was prepared in "anticipation of litigation" is . . . . incapable of precise definition. In general, though, a party must show more than a remote prospect, an inchoate possibility, or a likely chance of litigation. Rather, a party must show that there existed an identifiable specific claim of impending litigation when the materials were prepared. The mere involvement of, . . . or investigation by an attorney does not, in itself, evidence the "anticipation of litigation." Neither will the mere fact that litigation actually occurred establish that the documents prepared before the litigation were created in anticipation thereof.

> This Circuit has imposed an additional requirement beyond that embodied **[*51]** in the reasonable anticipation test. Thus, the second prong of the test is whether the material [was] produced because of the litigation and for no other purpose. In order to determine whether a document satisfies this standard, the proper inquiry is whether in light of the nature of the document and the factual situation in the particular case, the document can fairly be said to have been prepared or obtained because of the prospect of litigation. Documents created for other purposes that prove useful in subsequent litigation are not . . . work product; similarly, documents that are routinely prepared in the ordinary course of business are outside the scope of work product protection. Even where reasonable anticipation of litigation is established, whether the document comes within the purview of work product privilege still depends primarily on the reason or purpose for the document's production. Finally, the articulable claim likely to lead to litigation must pertain to this particular party, not the world in general.

In re Gabapentin Patent Litigation, 214 F.R.D. at 183-184 (citations and quotations omitted)(emphasis added).

ii. Entry 2629 (Exhibit E)

The Special Master concluded **[*52]** that the handwriting on log entry 2629 is not subject to privilege. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor the purpose of it, the substance of the writing indicates that it contains legal conclusions about priority date. The defendants argue that the Special Master reviewed the handwriting and did not err when he concluded that the privilege did not apply to the handwritten entries on 2629 because plaintiff failed to sustain its burden of proof and Mr. John's Declaration is untimely and insufficient.

According to the log, plaintiff has asserted that the handwriting is protected by the attorney-client privilege. Although the log states the author is "Joe John," no timely submitted evidence establishes the identity of the author of the handwriting. Without proof of the identity of the author and the purpose for which the writings were made, the plaintiff cannot establish that the handwriting is privileged.

iii. Entry 2632 (Exhibit F)

The Special Master concluded that the handwriting on log entry 2632 is not subject to privilege. Plaintiff states that the Special Master erred in this **[*53]** ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, the substance of the writing indicates that it contains legal conclusions about prior art. The defendants argue that the Special Master reviewed the handwriting and did not err when he concluded that the privilege did not apply to the handwritten entries on 2632 because plaintiff failed to sustain its burden of proof and Mr. John's Declaration is untimely and insufficient.

According to the log, plaintiff asserts that the handwriting on the document is privileged and protected work product. The timely presented record, however, does not reflect that the notes were made in anticipation of litigation nor does it establish the author of the handwriting. Moreover, although the log identifies Mr. Spivek as the author, the plaintiff concedes that it does not know the identity of the author for handwriting on entry 2632. See id. Without the identity of the author and the purpose for which the writings were made, the plaintiff cannot establish that the writing on the documents are privileged or protected work product.

iv. Entry 2633 (Exhibit G)

The Special Master concluded that the handwriting **[*54]** on log entry 2633 is not subject to privilege. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could not identify the author of the writing nor its purpose, the substance of the writing indicates that it contains legal conclusions about the patent's priority date. The defendants argue that the Special Master reviewed the handwriting and did not err when he concluded that the privilege did not apply to the handwritten entries on 2633 because plaintiff failed to meet its burden of proof and Mr. John's Declaration is untimely and insufficient.

According to the log, plaintiff asserts that the document contains handwriting protected by the attorney-client privilege and work product rule. Although the log identifies Mr. Spivek as the author, plaintiff concedes that it does not know the identity of the author for writings on entry 2633. See id. **[*55]** Moreover, the record is silent as to whether the notations were made in anticipation of litigation. Without the identity of the author and the purpose for which the writings were made, the plaintiff cannot establish that the writings on these documents are privileged or protected by the work product rule.

v. Entry 2634 (Exhibit H)

The Special Master concluded that the handwriting on log entry 2634 is not subject to privilege. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, the substance of the writing indicates that it contains legal conclusions about prior art. The defendants argue that the Special Master reviewed the handwriting and did not err when he concluded that the privilege did not apply to the handwritten entries on 2634 because plaintiff failed to sustain its burden of proof and Mr. John's Declaration is untimely and insufficient.

According to the log, plaintiff asserts that the handwriting on the document is protected by the attorney-client privilege and work product rule. Although Mr. Spivek is listed on the log as the author, plaintiff concedes that it does not **[*56]** know the identity of the author for writings on entry 2634. See id. Moreover, the record is silent as to whether the notations were made in anticipation of litigation and for no other purpose. Without the identity of the author and the purpose for which the writings were created, the plaintiff cannot establish that the writings on these documents are privileged or protected by the work product rule.

vi. Entry 2645 (Exhibit 1)

The Special Master concluded that the handwriting on log entry 2645 is not privileged. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, the substance of the writing indicates that it contains legal conclusions that compare the patent's claim to certain technology. The defendants argue that the Special Master reviewed the handwriting and did not err when he concluded that the privilege did not apply to the handwritten entries on 2645 because the plaintiff failed to carry its burden of proof and Mr. John's Declaration is untimely and insufficient.

According to the log, plaintiff asserts that the document is protected by the attorney-client privilege. The record, **[*57]** however, does not establish the identity of the author and the log merely asserts that the author is in-house counsel. Plaintiff concedes that it does not know the identity of the author, see id., and without the identity of the author and the purpose for which the writings were made, the plaintiff cannot establish that the writings on these documents are privileged. Moreover, the writing embodies a series of questions about a product and does not appear privileged on its face. Thus, the plaintiff has failed to meet is burden to sustain the privilege.

vii. Entry 9062 (Exhibit J)

The Special Master concluded that the writings on log entry 9062 are not privileged. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, the substance of the writing indicates that it contains legal conclusions. The defendants argue that the Special Master reviewed the handwriting and did not err when he concluded that the privilege did not apply to the handwritten entries on 9062 and properly declined to consider Mr. John's Declaration as it was untimely and insufficient.

According to the log, plaintiff **[*58]** asserts that the document constitutes work product prepared in anticipation of litigation. The record, however, does not show that it was created for this purpose nor does it establish the author of the handwritten notations. Although the log identifies Michael Casey as the author of the document, the plaintiff concedes that it does not know the identity of the author for handwriting on entry 9062. See Pl. Br. at 24-25. Moreover, entry 9062 does not include handwritten words but rather embodies underlines of words in the text of a published patent. Without the identity of the author and the purpose for which the lines were made, the plaintiff cannot establish that the writings on these documents are privileged.

viii. Entry 1861 (Exhibit K)

The Special Master concluded that log entry 1861 is an undated document without an identified author and nothing on its face shows that it is privileged. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, there is information within the document that suggests that it was drafted by one of IDT's in-house lawyers and the substance indicates that **[*59]** the document contains legal advice "on the most likely terms of a" sales transaction. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 1861 because plaintiff failed to produce evidence to support its claims that the document is privileged.

According to the log, entry 1861 is an undated outline that plaintiff asserts embodies work product and privileged communications with Jim DiGorgio about a VOIP patent. In 2005, Mr. DiGiorgio was senior counsel for IDT. Nonetheless, there is no showing that the outline was prepared in anticipation of litigation, no showing that when it was shared with Net2Phone that the plaintiff and IDT had a shared legal interest, and no showing it was authored by an attorney. In fact, according to the log, the author is listed as "NetSpeak Corp." and the plaintiff has conceded that it does not know the identity of the author. See id. at 25. Moreover, the section titled "most likely" does not contain any legal analysis but rather it contains terms of a financial agreement that may be reached. Without the identity of the author and the purpose for which the document was created, the plaintiff cannot **[*60]** establish that the document is privileged.

ix. Entry 1864 (Exhibit L)

The Special Master concluded that log entry 1864 is an undated document without an identified author that does not embody privileged communications. Plaintiff states that the Special Master erred in this ruling because, although the plaintiff could neither identify the author of the writing nor its purpose, the substance of the writing indicates that it was prepared by a member of IDT's in-house legal team and contains legal advice "on the most likely terms of a" sales transaction. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 1864 because plaintiff failed to produce evidence to support its claim that the document is privileged.

According to the log, entry 1864 is identified as having been authored by "IDT" and received by David Greenblatt, an IDT employee, Declaration of Hannah Stotts-Bumstead, dated May 5, 2008, at Ex. 23, that plaintiff asserts embodies work product and privileged communications with Jim DiGiorgio about the corporation's patent portfolio. Again, plaintiff conceded that it does not know the identity of the author of entry 1864. **[*61]** See id. Moreover, there is nothing in the record to show it was prepared in anticipation of litigation and for no other purpose. Lastly, the section titled "most likely" does not contain any privileged material on its face. Instead, it contains terms of a potential financial agreement, including time frame and payment method. Absent proof of the identity of the author and the purpose for which it was created, the plaintiff cannot establish that the document is privileged.

x. Entry 1870 (Exhibit M)

The Special Master concluded that the certification of IDT's in-house counsel attesting to the legal nature of log entry 1870 was insufficient to sustain the privilege assertion because in-house counsel could not identify who marked up the document, when, or why. Plaintiff argues that entry 1870 is a draft of IDT's tender offer that was drafted by IDT's inside and outside lawyers for filing with the Securities and Exchange Commission ["SEC"] and the Special Master erred in failing to credit the declaration of IDT's in-house counsel that asserts that the edits were those of lawyers. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to **[*62]** entry 1870 because the plaintiff provided no evidence as to who marked up the document, when or why.

According to the log, entry 1870 is a draft SEC filing allegedly reflecting communications with the law firm of Kramer Levin. The log describes the author as "Net2Phone, Inc. NtoP acquisition IDT Corporation." The plaintiff, however, did not present evidence that establishes the author and conceded that it does not know the identity of the author. See id. at 25-26. The plaintiff submitted the Certification of Dov Schwell, Senior Vice President for IDT, dated March 18, 2008, to support its contention that these documents reflect confidential attorney-client communications, Stott-Bumsted Dec. Ex. 20, but it was untimely and will not be considered. Thus, the plaintiff failed to submit timely evidence that identifies the author of these edits. Without the identity of the author of the markings, the plaintiff cannot establish that they are privileged.

xi. Entry 3814 (Exhibit N)

The Special Master concluded that the certification of IDT's in-house counsel attesting to the legal nature of log entry 3814 was insufficient to sustain the privilege assertion because it embodies a communication among **[*63]** non-lawyers and the document is not clearly privileged on its face and the declaration submitted did not identify the author of the markings on the document. Plaintiff argues that entry 3814 is a draft of IDT's tender offer that was drafted by IDT's inside and outside lawyers for filing with the SEC and that the Special Master erred in failing to credit the certification of IDT's in-house counsel who attested to the legal nature of entry 3814 even though he could not identify its author. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 3814 because plaintiff's untimely declaration did not identify if counsel made any of the marks.

According to the log, the draft of the SEC filing is dated November 10, 2005 and the author is identified as a person at "semdd.com" and the recipient is a person at IDT. The plaintiff has conceded that it does not know the identity of the author of the markings on entry 3814. See Pl. Br. at 25-26. Moreover, at the time of these communications, IDT and plaintiff did not share a common legal interest and when the document was shared between them, it lost any privilege status. For these reasons, **[*64]** the privilege is not applicable.

xii. Entry 1142 (Exhibit O)

The Special Master concluded that plaintiff failed to present evidence to show that log entry 1142, an email from IDT's in house counsel Jim DiGiorgio to David Greenblatt about the VOIP patents, was privileged and there is no way to tell if it was a privileged communication on its face. Plaintiff argues that, because the email was written by its in house counsel, the only plausible interpretation is that in-house counsel is "proposing a meeting at which he will render legal advice concerning legal action," Pl. Br. at 26, and a corporation's declaration is not needed to establish the privilege. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 1142 and plaintiff presented no evidence to support its clam of privilege despite having had an opportunity to do so.

Although the log asserts that the email contains legal advice, the face of the document does not support this description and plaintiff has presented no evidence to show that it was associated with an effort to secure legal advice. The mere fact it was from an attorney, without showing its purpose, is insufficient **[*65]** to sustain the privilege since the privilege applies only to communications engaged in for the purpose of securing or providing legal advice. As such, the Special Master's ruling will not be disturbed.

xiii. Entries 1332 & 1333 (Exhibit P)

The Special Master concluded that the plaintiff did not submit evidence that shows log entries 1332-33 are privileged and he could not determine from their face that they embody privileged communications. The plaintiff contends that the portion of the document summarizing a meeting between Binyamin Bauman, a nonlawyer, and the Chairman of IDT's Board of Directors concerning patent enforcement embodies a request for legal advice and is privileged. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entries 1332-33 because there is nothing to show that it involves a discussion between non-lawyers reflecting advice of counsel.

According to the log, entries 1332-33 are March 28, 2006 emails from Binyamin Bauman to David Lando, Net2Phone employees, Stott-Bumsted Dec. at Ex. 23, that contain communications with in-house counsel regarding the patent portfolio. The emails, however, were not exchanged **[*66]** between attorneys and their contents do not reflect legal advice. Moreover, the plaintiff has submitted no evidence to supports its claim of privilege regarding these documents. Thus, the Special Master's conclusion that the documents are not privileged is correct.

xiv. Entry 1337 (Exhibit Q)

The Special Master concluded that entry 1337, a February 21, 2006 email from Philip Florenzo, an attorney in private practice, to David Lando, his client at Net2Phone, see id., forwarding slides prepared by Joseph John, a senior technical advisor in IDT's in-house Intellectual Property group, contains material that is purely factual and is thus not protected by privilege and plaintiff did not timely submit other evidence to establish its claim of privilege. In addition, the Special Master noted that the document was shared with IDT at a time that plaintiff and IDT did not share a common legal interest. The plaintiff asserts that the communication of these facts was for obtaining legal advice and is protected by the attorney-client privilege and that the Declaration of Joseph John explained this was the purpose. The defendants argue that the Special Master did not err when he concluded that the privilege **[*67]** did not apply to entry 1337 because he correctly disregarded Mr. John's Declaration and correctly acknowledged that, even if the document were privileged, that privilege had been waived.

According to the log, the plaintiff asserts that the email and attachment reflects legal advice from in-house patent counsel. Even if this were established, the contents actually emanated from an IDT employee during the period before the tender offer had occurred. As stated previously, the common interest doctrine does not protect communications with IDT during this period. Moreover, the only evidence to support the privileged assertion comes in from the untimely submission of the Declaration of Joseph John. See Stott-Bumsted Dec. at 20. Since that evidence is precluded, the plaintiff has failed to timely submit competent evidence to support its privilege claims. As such, the Special Master did not err in concluding that no privilege attached to these documents.

xv. Entries 1840 (Exhibit R)

The Special Master concluded that plaintiff produced no evidence that establishes log entry 1840, a draft of a 2005 operation plan with handwriting, is protected by privilege and nothing on the face of the document **[*68]** reveals that it is privileged. Entry 1845 (Exhibit S) is a similar document without handwriting. The plaintiff argues that entry 1840 is a draft presentation by IDT's in-house legal group and contains legal advice and legal services provided in 2005. Plaintiff asserts that the face of the document demonstrates its legal nature, and thus it was not required to provide a declaration to establish that it is privileged. Plaintiff also notes that the Special Master sustained the privilege concerning a similar document. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 1840 because plaintiff did not present an affidavit concerning the privilege and did not bring to the attention of the Special Master the similarity between 1840 and 1845, even though he allowed plaintiff to move for reconsideration.

According to the log, this document is described as a "presentation" that the IDT Phoenix Group authored. Plaintiff describes it as a privileged communication with Mr. DiGiorgio about patents. Despite this description, the plaintiff has failed to disclose the actual author of the document or handwriting on entry 1840, and indeed **[*69]** conceded that it does not know the identity of the author of the handwritings on entry 1840. See Pl. Br. at 27. Moreover, the document is titled "2005 Operating Plan" and does not contain any legal advice. Without the identity of the author or proof that it was created to obtain or convey legal advice, the plaintiff cannot establish that the document is privileged.

xvi. Entry 2783 (Exhibit T)

The Special Master concluded that log entry 2783, an April 2, 2003 email from Anthony Tobey, a member of IDT's information technology staff, to a person associated with IXtelecom, is not privileged because it is between non-lawyers, nothing on its face shows it is privileged, and plaintiff submitted no evidence to establish it is privileged. The plaintiff argues that the first sentence of the email is privileged on its face because it conveys legal advice received from IDT's legal department and thus there is no need for a certification to establish it as privilege. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 2783 because it involves a communication between non-lawyers and plaintiff failed to supply evidence to show that the **[*70]** entry was privileged.

According to the log, plaintiff describes the email as reflecting privileged communications between IDT's business and in-house counsel about the information-technology policy. Despite this entry, the plaintiff concedes that the communication is between non-lawyers. See id. at 28. The plaintiff did not submit any evidence that shows the information relayed between the non-attorneys is legal advice. Moreover, a review of this document shows that it is a group email about the employees' access to external file sharing networks and does not contain legal advice. Thus, the plaintiff has not established that the document is privileged.

xvii. Entry 4562 (Exhibit U)

According to the log, entry 4562 is an email and attachment from Jim DiGiorgio to Luis Diaz that relates to monetizing intellectual property. The Special Master concluded that the plaintiff has waived the privilege relating to the Monetization Plan by its disclosure of documents about the same subject. The plaintiff argues that entry 4562 is an email between two attorneys that contains both non-privileged and privileged information and that the privileged portion should be disclosed but the remainder should **[*71]** be shielded because it embodies legal advice to IDT about intellectual property and does not address the Monetization Plan. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 4562 because this document is within the scope of the waiver and plaintiff concedes the document should have been produced. Pl. Br. at 28.

A review of the document shows that the email merely forwards the attachment and the attachment is a document addressing monetization. For the reasons set forth herein, to the extent a privilege covered this subject, it has been waived. As such, the Special Master's conclusions will not be disturbed here.

xviii. Entries 8832 & 8833 (Exhibit C) (which are contained in entry 8034)

According to the log, entries 8832 and 8833 are January 24, 2005 emails among Jim DiGiorgio, Peter Emanuel, a GE lawyer, and Laurence Rosenberg, a member of GE's Technology Group, which were shared with seven GE staff members about a European patent. Although the Declaration of Kenneth Glick, an attorney for GE, reflects that GE maintained the confidentiality of its internal discussions with counsel and the information it received and shared **[*72]** with IDT, see Stott-Bumsted Dec. Ex.18 (attaching the Declaration of Kenneth Glick dated Jan. 14, 2008), the Special Master found that plaintiff failed to timely identify one of the recipients, Ed Howard. Thus, the Special Master concluded that the plaintiff failed to meet its burden of showing based on timely submitted evidence that this email was privileged. The plaintiff argues that the Special Master erred in his conclusion because, in light of Glick's Declaration, it was apparent that the email was confidential and it had timely produced evidence that shows Mr. Howard was a Net2Phone lawyer. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply because the plaintiff failed to meet its burden of proof and because the communications between IDT and GE are not privileged.

The Court concludes that even though the plaintiff timely identified Ed Howard, the contents were shared with GE personnel and, for the reasons already discussed, these communications are not privileged. Thus, the record before the Court shows that the plaintiff failed to meet its burden to withhold the document on privilege grounds.

xix. Entry 9061 (Exhibit C)

For **[*73]** the same reasons, the privilege assertion over entry 9061, which is represented to be duplicated in 8834 is overruled.

xx. Entry 9073 (Exhibit V) (redacted versions of already produced materials)

Document entry 9073 is an email dated January 20, 2005 from Arthur Dubroff at Net2Phone to Claude Pupkin and Glenn Williams and copies were provided to Lione Alroy, Michael Pastor, Mitch Silverman, Ken Kaplan and Nicholas Day. These individuals are associated with Net2Phone or IDT. The Special Master concluded that privilege did not apply to the redacted portions of entry 9073 because the plaintiff did not submit any proof the privilege applied and nothing from the face of the document indicated it was privileged. The plaintiff argues that the portions of one paragraph are privileged because it contains communications from Arthur Dubroff to Net2Phone attorneys about actions to be taken with respect to Net2Phone's intellectual property. The defendants argue that the Special Master did not err when he concluded that the privilege did not apply to entry 9073 because the plaintiff failed to meet its burden of proof.

As to entry 9073, this Court finds that the plaintiff fails to meet its burden of **[*74]** establishing that this document is privileged with timely produced evidence and therefore the privilege assertion is overruled. Moreover, a review of the document reveals that it does not contain or seek legal advice but rather pertains to valuation of assets. Furthermore, as stated previously, even if it were privileged, the privilege about valuation has been waived and because this document discusses valuation it must be disclosed.

xxi. Entry 4382 (Exhibit W)

The Special Master concluded that privilege did not apply to log entry 4382, a June 18, 2004 email and attachment from Pat Gartner to Luis Diaz, an IDT attorney, because it forwarded as an attachment drawings created by a non-lawyer that do not appear privileged on their face. Moreover, although the log states the email and attachment discuss obtaining legal advice about intellectual property, the Special Master concluded that the Declaration of Luis Diaz did not address this document, no timely evidence was adduced to support the privilege claim, and he refused to consider the untimely submissions purported to support plaintiff's assertion of privilege. The plaintiff argues that, because the document was sent to an attorney, **[*75]** it is only plausible to conclude that it was sent in connection with a request for legal advice about an intellectual property matter and the declaration of Mr. Gartner supports this conclusion. The defendants argue that the Special Master did not err when he declined to consider the untimely evidence and in any event, it did not address the document and the plaintiff did not prove that the privilege applies to entry 4382. The defendants also argue that any argument that is nonresponsive to any discovery demand cannot be a basis to object to the privilege ruling because this is not an issue ripe for resolution in this context.

This Court finds that the timely submission of Luis Diaz's Declaration does not satisfy its burden because Mr. Diaz's Declaration does not address this document. Moreover, the Court will not consider plaintiff's untimely submissions. Finally, a review of the document reflects it embodies factual information and does not on its face reflect legal advice. As such, the plaintiff has failed to meet its burden to show that the document is covered by the attorney-client privilege.

xxii. Additional Entries

Neither party's submission addresses the Special Master's decision **[*76]** regarding privilege to log entries 4638-39, 4675, 3893, and 1766. As such, the Court will deem any objections thereto waived and the Court will not address the Special Master's decision on these documents.

E. Motion to Seal

The plaintiff seeks to seal certain documents submitted in connection with its objections to the Special Master's Report. Plaintiff has not demonstrated that these documents warrant sealing. First, the Special Master's Report was publicly filed without opposition by the plaintiff and discloses information the plaintiff now seeks to seal. Second, the communications the plaintiff's seeks to seal relate to matters that occurred several years ago and thus the need to seal what may have been confidential information no longer exists as there is no showing that there would be present harm from disclosure. Lastly, to the extent the request to seal is made to preserve the confidential nature of alleged privileged documents, the assertion of privilege has been overruled and the need for confidentiality for this purpose is moot. Thus, plaintiff's motion to seal the documents in connection with the objection is denied.

III. Conclusion

For the foregoing reasons, the Special Master's **[*77]** findings of fact, conclusions of law, and procedural determinations are affirmed in their entirety and plaintiff's motion to seal is denied. The plaintiff shall produce the withheld documents no later than June 30, 2008.

/s/ Patty Shwartz

United States Magistrate Judge

Date: June 25, 2008

ORDER

This matter having come before the Court on the plaintiff's objections to the Special Master's Report issued on April 21, 2008 and plaintiff's motion to seal the documents submitted in connection with its objections;

and the Court having considered the parties submissions;

and the Court having decided this motion without oral argument pursuant to Fed. R. Civ. P. 78 and L. Civ. R. 78.1;

and for the reasons set forth in the Opinion dated June 25, 2008;

IT IS ON THIS 25th day of June, 2008,

ORDERED that the Special Master's Report is affirmed in its entirety and the objections [Docket No. 219] are overruled;

IT IS FURTHER ORDERED that the plaintiff shall produce the documents consistent with the Special Master's Report no later than **June 30, 2008;** and

IT IS FURTHER ORDERED that the plaintiff's motion to seal [Docket No. 222] is denied.

/s/ Patty Shwartz

UNITED STATES MAGISTRATE JUDGE

Command Searching > **Patent, Trademark & Copyright Periodicals, Combined**

Search

Enter Search Terms

◉ Terms and Connectors ○ Natural Language ○ Easy Search™

6108704 or 6,108,704

Suggest terms for my search

Check spelling

Restrict by Document Segment

Select a document segment, enter search terms for the segment, then click Add.

Select a Segment | Add

Note: Segment availability differs between sources. Segments may not be applied consistently across sources.

Restrict by Date

◉ No Date Restrictions ○ From To Date formats...

Search Connectors

and	and	w/p	in same paragraph
or	or	w/seg	in same segment
w/N	within N words	w/s	in same sentence
pre/N	precedes by N words	and not	and not

More Connectors & Commands...

How Do I...?

- Combine sources?
- Restrict by date?
- Restrict by document segment?
- Use wildcards as placeholders for one or more chara term?

View Tutorials

No Documents Found

No documents were found for your search terms

"6108704 or 6,108,704"

Click "Save this search as an Alert" to schedule your search to run in the future.

- OR -

Click "Edit Search" to return to the search form and modify your search.

Suggestions:

- Check for spelling errors .
- Remove some search terms.
- Use more common search terms, such as those listed in "Suggested Words and Concepts"
- Use a less restrictive date range.

Save this Search as an Alert   Edit Search

Command Searching > **News, All (English, Full Text)**

Search

Enter Search Terms

◉ Terms and Connectors ○ Natural Language ○ Easy Search™

6108704 or 6,108,704

Suggest terms for my search

Check spelling

Restrict by Document Segment

Select a document segment, enter search terms for the segment, then click Add.

Select a Segment | Add

Note: Segment availability differs between sources. Segments may not be applied consistently across sources.

Restrict by Date

◉ No Date Restrictions ○ From To Date formats...

Search Connectors

and	and	w/p	in same paragraph
or	or	w/seg	in same segment
w/N	within N words	w/s	in same sentence
pre/N	precedes by N words	and not	and not

More Connectors & Commands...

How Do I...?

- Combine sources?
- Restrict by date?
- Restrict by document segment?
- Use wildcards as placeholders for one or more chara term?

View Tutorials

Source: Command Searching > **News, All (English, Full Text)**
Terms: **6108704 or 6,108,704** (Edit Search | Suggest Terms for My Search)

Select for FOCUS™ or Delivery

☐ 1. vnunet.com, 6 June 2006 Tuesday, 167 words, Skype under fire over Net2Phone patent, Matt Chapman
... 2000. US Patent **6,108,704** covers the technology that allows ...

☐ 2. News Release, February 27, 1996, 6108704, 211 words, New Teradyne Third-Party Supplier Program for Programming & Fixture Houses

Source: Command Searching > **News, All (English, Full Text)**
Terms: **6108704 or 6,108,704** (Edit Search | Suggest Terms for My Search)
View: Cite
Date/Time: Wednesday, February 18, 2009 - 1:35 PM EST

Skype under fire over Net2Phone patent vnunet.com 6 June 2006 Tuesday

vnunet.com

6 June 2006 Tuesday

LENGTH: 167 words

HEADLINE: Skype under fire over Net2Phone patent

BYLINE: Matt Chapman

HIGHLIGHT:

Lawsuit seeks damages and injunction

BODY:

VoIP company Net2Phone has filed a lawsuit against eBay, the owner of Skype, claiming that its internet phone software has infringed Net2Phone's patents.

The lawsuit, filed at the US District Court in New Jersey, claims that Skype's software infringes a point-to-point IP patent that Net2Phone was awarded in 2000.

US Patent **6,108,704** covers the technology that allows VoIP software to swap IP addresses and create a direct connection.

It is thought that VoIP services from telecoms companies will be safe from similar lawsuits as they use exchanges to create the direct connections between users.

Net2Phone's lawsuit asks for compensation for the infringement of the patent, and asks the court to issue an injunction banning the Skype service.

EBay bought Skype for around $2.6bn in September last year and the VoIP service signed up its 100 millionth user last month.

The lawsuit follows an unrelated action filed against eBay by StreamCast last month, which questioned the legality of the Skype acquisition.

LOAD-DATE: 10 July 2006

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

REEXAM CONTROL NUMBER	FILING OR 371 (c) DATE	PATENT NUMBER
90/010,416	02/17/2009	6108704

CONFIRMATION NO. 1061

REEXAM ASSIGNMENT NOTICE

Jeffrey S. Ginsberg, ESQ.
KENYON & KENYON
One Broadway
Neew York, NY 10004

OC000000034702625

Date Mailed: 02/24/2009

NOTICE OF ASSIGNMENT OF REEXAMINATION REQUEST

The above-identified request for reexamination has been assigned to Art Unit 3992. All future correspondence to the proceeding should be identified by the control number listed above and directed to the assigned Art Unit.

A copy of this Notice is being sent to the latest attorney or agent of record in the patent file or to all owners of record. (See 37 CFR 1.33(c)). If the addressee is not, or does not represent, the current owner, he or she is required to forward all communications regarding this proceeding to the current owner(s). An attorney or agent receiving this communication who does not represent the current owner(s) may wish to seek to withdraw pursuant to 37 CFR 1.36 in order to avoid receiving future communications. If the address of the current owner(s) is unknown, this communication should be returned within the request to withdraw pursuant to Section 1.36.

cc: Third Party Requester(if any)
Edwin H. Taylor
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040

/jcmcdougald/

Legal Instruments Examiner
Central Reexamination Unit 571-272-7705; FAX No. 571-273-9900

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

REEXAM CONTROL NUMBER	FILING OR 371 (c) DATE	PATENT NUMBER
90/010,416	02/17/2009	6108704

CONFIRMATION NO. 1061

REEXAMINATION REQUEST NOTICE

Edwin H. Taylor
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040

OC000000034702624

Date Mailed: 02/24/2009

NOTICE OF REEXAMINATION REQUEST FILING DATE

(Third Party Requester)

Requester is hereby notified that the filing date of the request for reexamination is 02/17/2009, the date that the filing requirements of 37 CFR § 1.510 were received.

A decision on the request for reexamination will be mailed within three months from the filing date of the request for reexamination. (See 37 CFR 1.515(a)).

A copy of the Notice is being sent to the person identified by the requester as the patent owner. Further patent owner correspondence will be the latest attorney or agent of record in the patent file. (See 37 CFR 1.33). Any paper filed should include a reference to the present request for reexamination (by Reexamination Control Number).

cc: Patent Owner
Jeffrey S. Ginsberg, ESQ.
KENYON & KENYON
One Broadway
Neew York, NY 10004

/jcmcdougald/

Legal Instruments Examiner
Central Reexamination Unit 571-272-7705; FAX No. 571-273-9900

Patent Assignment Abstract of Title

Total Assignments: 9

Application #: 08533115 **Filing Dt:** 09/25/1995 **Patent #:** 6108704 **Issue Dt:** 08/22/2000
PCT #: NONE **Publication #:** NONE **Pub Dt:**
Inventors: SHANE D. MATTAWAY, CRAIG B. STRICKLAND, GLENN W. HUTTON
Title: POINT-TO-POINT INTERNET PROTOCOL

Assignment: 1

Reel/Frame: 007981 / 0020 **Received:** 06/14/1996 **Recorded:** 05/30/1996 **Mailed:** 12/26/1996 **Pages:** 3
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).
Assignor: HUTTON, GLEN W. **Exec Dt:** 11/27/1995
Assignee: INTERNET TELEPHONE COMPANY
1 SOUTH OCEAN BOULEVARD, SUITE 305
BOCA RATON, FLORIDA 33432
Correspondent: BOOKSTEIN & KUDIRKA, P.C.
BRUCE D. JOBSE, ESQ.
ONE BEACON STREET
BOSTON, MASSACHUSETTS 02108

Assignment: 2

Reel/Frame: 008295 / 0167 **Received:** 01/15/1997 **Recorded:** 01/08/1996 **Mailed:** 03/17/1997 **Pages:** 4
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).
Assignor: HUTTON, GLENN W. **Exec Dt:** 11/27/1995
Assignee: INTERNET TELEPHONE COMPANY
SUITE 305
1 SOUTH OCEAN BOULEVARD
BOCA RATON, FLORIDA 33432
Correspondent: BOOKSTEIN & KUDIRKA, P.C.
BRUCE D. JOBSE
ONE BEACON STREET
BOSTON, MA 02108

Assignment: 3

Reel/Frame: 007981 / 0053 **Received:** 06/14/1996 **Recorded:** 05/30/1996 **Mailed:** 09/04/1996 **Pages:** 4
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).
Assignor: INTERNET TELEPHONE COMPANY **Exec Dt:** 05/14/1996
Assignee: NETSPEAK CORPORATION
STE. 104
902 CLINT MOORE ROAD
BOCA RATON, FLORIDA 33437
Correspondent: BOOKSTEIN & KUDIRKA, P.C.
BRUCE D. JOBSE
ONE BEACON STREET
BOSTON, MA 02108

Assignment: 4

Reel/Frame: 009792 / 0568 **Received:** 02/25/1999 **Recorded:** 02/22/1999 **Mailed:** 05/24/1999 **Pages:** 4
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).
Assignor: STRICKLAND, CRAIG B. **Exec Dt:** 02/18/1999
Assignee: NETSPEAK CORPORATION
902 CLINT MOORE ROAD, SUITE 104
BOCA RATON, FLORIDA 33487
Correspondent: KUDIRKA & JOBSE, LLP
BRUCE D. JOBSE, ESQ.
TWO CENTER PLAZA
BOSTON, MA 02108

Assignment: 5

Reel/Frame: 010012 / 0953 **Received:** 06/14/1999 **Recorded:** 06/07/1999 **Mailed:** 08/12/1999 **Pages:** 4

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: MATTAWAY, SHANE D. **Exec Dt:** 03/02/1999

Assignee: NETSPEAK CORPORATION
902 CLINT MOORE ROAD, SUITE 104
BOCA RATON, FLORIDA 33487

Correspondent: KUDIRKA & JOBSE, LLP
BRUCE D. JOBSE, ESQ.
TWO CENTER PLAZA
BOSTON, MA 02108

Assignment: 6

Reel/Frame: 016522 / 0205 **Received:** 09/12/2005 **Recorded:** 09/12/2005 **Mailed:** 09/13/2005 **Pages:** 32

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: NETSPEAK CORPORATION **Exec Dt:** 03/25/2004

Assignee: VOIP TECHNOLOGY HOLDINGS, LLC
520 BROAD STREET, 8TH FLOOR
ATTN: NET2PHONE LEGAL DEPARTMENT
NEWARK, NEW JERSEY 07102

Correspondent: DENNEMEYER & CO., INC.
6990 COLUMBIA GATEWAY DRIVE, STE 360
ATTN: MICHELLE CHERVENAK
COLUMBIA, MD 21046

Assignment: 7

Reel/Frame: 016945 / 0858 **Received:** 10/28/2005 **Recorded:** 10/28/2005 **Mailed:** 12/30/2005 **Pages:** 32

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: VOIP TECHNOLOGY HOLDINGS, LLC **Exec Dt:** 10/06/2005

Assignee: NET2PHONE, INC.
520 BROAD STREET, 8TH FLOOR
NEWARK, NEW JERSEY 07102

Correspondent: DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

Assignment: 8

Reel/Frame: 016945 / 0890 **Received:** 10/28/2005 **Recorded:** 10/28/2005 **Mailed:** 12/30/2005 **Pages:** 33

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: NETSPEAK CORPORATION **Exec Dt:** 10/06/2005

Assignee: NET2PHONE, INC.
520 BROAD STREET, 8TH FLOOR
NEWARK, NEW JERSEY 07102

Correspondent: DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

Assignment: 9

Reel/Frame: 017105 / 0240 **Received:** 12/09/2005 **Recorded:** 12/09/2005 **Mailed:** 02/02/2006 **Pages:** 34

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: VOIP TECHNOLOGY HOLDINGS, LLC **Exec Dt:** 10/06/2005

Assignee: NET2PHONE, INC.
520 BROAD STREET, 8TH FLOOR
NEWARK, NEW JERSEY 07102

Correspondent: MICHAEL R. CASEY, PH.D.
DAVIDSON BERQUIST JACKSON & GOWDEY, LLP
4300 WILSON BOULEVARD, 7TH FLOOR
ARLINGTON, VIRGINIA 22203

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/010,416	02/17/2009	6108704	3801G184	1061

7590 03/11/2009

Jeffrey S. Ginsberg, ESQ.
KENYON & KENYON
One Broadway
New York, NY 10004

EXAMINER	

ART UNIT	PAPER NUMBER

DATE MAILED: 03/11/2009

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

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

1279 OAKMEAD PARKWAY

SUNNYVALE, CA 94085-4040

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/010,416*.

PATENT NO. *6108704*.

ART UNIT *3992*.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

PTOL-465 (Rev.07-04)

Order Granting / Denying Request For Ex Parte Reexamination	Control No. 90/010,416	Patent Under Reexamination 6108704	
	Examiner ALEXANDER J. KOSOWSKI	Art Unit 3992	

--*The MAILING DATE of this communication appears on the cover sheet with the correspondence address*--

The request for *ex parte* reexamination filed 17 February 2009 has been considered and a determination has been made. An identification of the claims, the references relied upon, and the rationale supporting the determination are attached.

Attachments: a)☐ PTO-892, b)☒ PTO/SB/08, c)☐ Other: _____

1. ☒ The request for *ex parte* reexamination is GRANTED.

RESPONSE TIMES ARE SET AS FOLLOWS:

For Patent Owner's Statement (Optional): TWO MONTHS from the mailing date of this communication (37 CFR 1.530 (b)). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**

For Requester's Reply (optional): TWO MONTHS from the **date of service** of any timely filed Patent Owner's Statement (37 CFR 1.535). **NO EXTENSION OF THIS TIME PERIOD IS PERMITTED.** If Patent Owner does not file a timely statement under 37 CFR 1.530(b), then no reply by requester is permitted.

2. ☐ The request for *ex parte* reexamination is DENIED.

This decision is not appealable (35 U.S.C. 303(c)). Requester may seek review by petition to the Commissioner under 37 CFR 1.181 within ONE MONTH from the mailing date of this communication (37 CFR 1.515(c)). **EXTENSION OF TIME TO FILE SUCH A PETITION UNDER 37 CFR 1.181 ARE AVAILABLE ONLY BY PETITION TO SUSPEND OR WAIVE THE REGULATIONS UNDER 37 CFR 1.183.**

In due course, a refund under 37 CFR 1.26 (c) will be made to requester:

a) ☐ by Treasury check or,

b) ☐ by credit to Deposit Account No. _____, or

c) ☐ by credit to a credit card account, unless otherwise notified (35 U.S.C. 303(c)).

cc:Requester (if third party requester)

DECISION

1) A substantial new question of patentability affecting claims 1-7 and 10-44 of United States Patent Number 6,108,704 (Hutton et al) is raised by the request for *ex parte* reexamination filed 02/17/09.

Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that *ex parte* reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.550(a)). Extensions of time in *ex parte* reexamination proceedings are provided for in 37 CFR 1.550(c).

References Cited in the Request

NetBIOS (Protocols for X/Open PC Interworking SMB...)

Etherphone (Collected Papers 1987-1988...)

Vin (Multimedia Conferencing in the Etherphone Environment...)

RFC 1531 (Dynamic Host Configuration Protocol...)

Pinard (U.S. Pat 5,533,110)

Vocalchat (5 submitted Vocalchat references, exhibits G-K in the Request)

Identification of Every Claim for Which Reexamination is Requested

2) The six sets of references cited above are discussed regarding claims 1-7 and 10-44 of the Hutton patent. Requestor has proposed at least 14 possible combinations of rejections for the requested claims. Pages 12-24 of the Request detail out proposed substantial new questions of patentability in light of the combination of the six sets of references cited above.

Prosecution History

3) The Hutton patent was assigned serial number 08/533,115. During prosecution, the application was allowed with no specific reasons for allowance after a series of amendments to the claims, in addition to the filing of an affidavit under 37 U.S.C. 1.131 antedating a primary reference. None of the references in the currently filed request except Pinard were previously discussed by the examiner or applied to claims 1-7 or 10-44 in the prosecution history of the Hutton patent. The Pinard reference was initialed as considered by the original examiner in an IDS submitted along with an office action mailed 6/2/97. However, the Pinard reference was not discussed or actively used in any office action during the prosecution of the Hutton application.

Substantial New Question of Patentability

4) For purposes of determination, independent claim 1 is a representative claim. The italicized sections of claim 1 below are utilized by the examiner to show how specific teachings of the proposed references create a substantial new question of patentability.

Claim 1: A computer program product for use with a computer system, the computer system executing a first process and operatively connectable to a second process and a server over a computer network, the computer program product comprising:

a computer usable medium having program code embodied in the medium, the program code comprising:

program code for transmitting to the server a network protocol address received by the first process following connection to the computer network;

program code for transmitting, to the server, a query as to whether the second process is connected to the computer network;

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

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

NetBIOS and RFC 1531

5) The NetBIOS reference discloses an address server with an address database for storing network protocol addresses usable by network nodes to establish point-to-point communications. RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a DHCP server.

The Request shows that NetBIOS and RFC 1531 in combination teach *program code for transmitting to the server a network protocol address received by the first process following connection to the computer network* (NetBIOS, pg. 385 and RFC, Section 2.2)*;*

program code for transmitting, to the server, a query as to whether the second process is connected to the computer network (NetBIOS, pg. 377 and 388)*;*

program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network (NetBIOS, pg. 389-394)*; and* *program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network* (NetBIOS, pg. 397-400).

The NetBIOS and RFC 1531 references were not previously discussed by the examiner nor applied to claims 1-7 and 10-44 in the prior examination of the patent as discussed above.

It is agreed that the consideration of NetBIOS and RFC 1531 raises an SNQ as to claims 1-7 and 10-44 of the Hutton patent as pointed out above. There is a substantial likelihood that a reasonable examiner would consider these teachings important in deciding whether or not these claims are patentable.

Accordingly, NetBIOS and RFC 1531 raise a substantial new question of claims 1-7 and 10-44, which question has not been decided in a previous examination of the Hutton patent nor was there a final holding of invalidity by the Federal Courts regarding the Hutton patent.

Etherphone, Vin and RFC 1531

6) The Etherphone reference discloses an address server with an address database for storing network protocol addresses usable by network nodes to establish point-to-point communications. RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a DHCP server. Vin discloses TCP/IP as the network protocol in an Etherphone system.

The Request shows that Etherphone in combination with Vin and RFC 1531 teach *program code for transmitting to the server a network protocol address received by the first process following connection to the computer network* (Etherphone, Swinehart 1, pgs. 2-4, Vin pg. 77, and RFC, Section 2.2);

program code for transmitting, to the server, a query as to whether the second process is connected to the computer network (Etherphone, Zellweger pgs. 1-3 and Swinehart 1, pg. 4);

program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network (Etherphone, Swinehart 1, pg. 4); *and*

program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network (Etherphone, Swinehart 1, pg. 4 and Zellweger, pg. 2).

The Etherphone, Vin and RFC 1531 references were not previously discussed by the examiner nor applied to claims 1-7 and 10-44 in the prior examination of the patent as discussed above.

It is agreed that the consideration of Etherphone, Vin and RFC 1531 raises an SNQ as to claims 1-7 and 10-44 of the Hutton patent as pointed out above. There is a substantial likelihood that a reasonable examiner would consider these teachings important in deciding whether or not these claims are patentable.

Accordingly, Etherphone, Vin and RFC 1531 raise a substantial new question of claims 1-7 and 10-44, which question has not been decided in a previous examination of the Hutton patent nor was there a final holding of invalidity by the Federal Courts regarding the Hutton patent.

VocalChat and RFC 1531

7) The VocalChat reference disclose an address server with an address database for storing network protocol addresses usable by network nodes to establish point-to-point communications.

RFC 1531 discloses how TCP/IP addresses are assigned dynamically by a DHCP server.

The Request shows that the VocalChat references and RFC 1531 in combination teach *program code for transmitting to the server a network protocol address received by the first process following connection to the computer network* (VocalChat ReadMe, Page 2, Help File page 2, RFC 1531, section 2.2);

program code for transmitting, to the server, a query as to whether the second process is connected to the computer network (VocalChat Help File, pages 2, 22 and 26);

program code for receiving a network protocol address of the second process from the server, when the second process is connected to the computer network (VocalChat Help File, page 2); *and*

program code, responsive to the network protocol address of the second process, for establishing a point-to-point communication link between the first process and the second process over the computer network (VocalChat Help File, page 17, User Guide, page 2).

The VocalChat and RFC 1531 references were not previously discussed by the examiner nor applied to claims 1-7 and 10-44 in the prior examination of the patent as discussed above.

It is agreed that the consideration of VocalChat and RFC 1531 raises an SNQ as to claims 1-7 and 10-44 of the Hutton patent as pointed out above. There is a substantial likelihood that a reasonable examiner would consider these teachings important in deciding whether or not these claims are patentable.

Accordingly, VocalChat and RFC 1531 raise a substantial new question of claims 1-7 and 10-44, which question has not been decided in a previous examination of the Hutton patent nor was there a final holding of invalidity by the Federal Courts regarding the Hutton patent.

Pinard

8) Pinard is cited by Requester as supporting the primary references in alternative obviousness rejections, as well as proposed teachings for dependent claims in Hutton. Examiner agrees that many of the claims in Hutton, particularly independent claims 10 and 21, as mapped out in the Request, appear to be read on by the combination of References listed above with Pinard.

The Pinard reference was not previously discussed by the examiner nor applied to claims 1-7 and 10-44 in the prior examination of the patent as discussed above.

It is agreed that the consideration of Pinard in combination with the references above raises an SNQ as to claims 1-7 and 10-44 of the Hutton patent as pointed out above. There is a substantial likelihood that a reasonable examiner would consider these teachings important in deciding whether or not these claims are patentable.

Accordingly, Pinard raises a substantial new question of claims 1-7 and 10-44, which question has not been decided in a previous examination of the Hutton patent nor was there a final holding of invalidity by the Federal Courts regarding the Hutton patent.

Scope of Reexamination

9) Claims 1-7 and 10-44 will be reexamined as requested in the Request.

Conclusion

Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.550(a)). Extension of time in *ex parte* reexamination proceedings are provided for in 37 CFR 1.550(c).

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving Patent No. 5,337,753 throughout the course of this reexamination proceeding. The third party requester is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

All correspondence relating to this ex parte reexamination proceeding should be directed as follows:

By U.S. Postal Service Mail to:

Mail Stop Ex Parte Reexam
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to:

(571) 273-9900
Central Reexamination Unit

By hand to:

Customer Service Window
Randolph Building
401 Dulany St.
Alexandria, VA 22314

By EFS-Web:

Registered users of EFS-Web may alternatively submit such correspondence via the electronic filing system EFS-Web, at

https://sportal.uspto.gov/authenticate/authenticateuserlocalepf.html

EFS-Web offers the benefit of quick submission to the particular area of the Office that needs to act on the correspondence. Also, EFS-Web submissions are "soft scanned" (i.e., electronically uploaded) directly into the official file for the reexamination proceeding, which offers parties the opportunity to review the content of their submissions after the "soft scanning" process is complete.

Any inquiry concerning this communication or earlier communications from the Reexamination Legal Advisor or Examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

/Alexander J Kosowski/

Primary Examiner, Art Unit 3992

7A
ESK

Substitute for Form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(use as many sheets as necessary)

Complete if Known	
Application Number	Reexamination of 6,108,704
Filing Date	Herewith
First Named Inventor:	
Art Unit	
Examiner Name	
Attorney Docket Number	003801.G184

Sheet 1 **of** 1

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
ABK	Exhibit F	US-	5,533,110	07-02-1996	Pinard, Deborah L., et al.	

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published	T[2]
ABK	Exhibit B	The Open Group, Technical Standard, Protocols for X/Open PC Interworking: SMB, Version 2, 1992, pages ii-xvi and pages 1-516.	
ABK	Exhibit C	ZELLWEGER, POLLE T., et al., Etherphone: Collected Papers 1987-1988, Xerox Corporation, May 1989.	
ABK	Exhibit D	VIN, HERRICK M., et al, Multimedia Conferencing in the Etherphone Environment, October 1991, pages 69-79.	
ABK	Exhibit E	DROMS, R., Dynamic Host Configuration Protocol, RFC 1531, Bucknell University, October 1993, pages 1-39.	
ABK	Exhibit G	VocalChat User's Guide Version 2.0, Vocaltec, 1994, pages 1-77.	
ABK	Exhibit H	README, VocalChat Version 2.02 & VocalChat WAN Version 2.02, Vocaltec, June 1994, pages 1-3.	
ABK	Exhibit I	VocalChat 1.01 Network Information, Vocaltec, 1994, pages 1-10.	
ABK	Exhibit J	VocalChat Information, Vocaltec, 1994, pages 1-31.	
ABK	Exhibit K	VocalChat Troubleshooting, Vocaltec, 1994, pages 1-101.	

Examiner Signature	[signature]	Date Considered	3/9/09

*Examiner: Initial if reference 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.

[1]Applicant's unique citation designation number (optional). [2]Applicant is to place a check mark here if English Translation is attached.

This collection of information is required by 37 CFR 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SENT FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.**

Based on Form PTO/SB/08A (08-03) as modified by BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP on 09/10/03.

Search Notes	Application/Control No. 90/010,416	Applicant(s)/Patent under Reexamination 6108704	
	Examiner ALEXANDER J. KOSOWSKI	Art Unit 3992	

SEARCHED

Class	Subclass	Date	Examiner

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
Reviewed proposed submitted prior art	3/9/2009	AJK

Reexamination	**Application/Control No.** 90/010,416	**Applicant(s)/Patent Under Reexamination** 6108704
	Certificate Date	**Certificate Number**

Requester Correspondence Address:	☐ Patent Owner	☒ Third Party

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

LITIGATION REVIEW ☒	**AJK** (examiner initials)	**3/9/09** (date)

Case Name	Director Initials
OPEN: 2:06cv2469 Net2phone, Inc. v. Ebay, Inc et al	Eric Keasel for G-M

COPENDING OFFICE PROCEEDINGS	
TYPE OF PROCEEDING	**NUMBER**
1. no copending office proceedings	
2.	
3.	
4.	

U.S. Patent and Trademark Office DOC. CODE **RXFILJKT**

OIPE MAR 13 2009 PATENT & TRADEMARK OFFICE

Re-Exam

STATEMENT UNDER 37 CFR 3.73(B)

Applicant / Patent Owner: Net2Phone, Inc. Docket No. 2655-0188

Control No. 90/010,416 Filed / Issued Date: 08/22/2000

Entitled: POINT-TO-POINT INTERNET PROTOCOL

Assignee: Net2Phone, Inc. A corporation
(Name of assignee) (Type of Assignee: corporation, partnership, university, government agency, etc.)

States that it is:

1. ☒ the assignee of the entire right, title, and interest; or
2. ☐ an assignee of less than the entire right, title and interest.
(The extent (by percentage) of its ownership interest is %)

in the patent application / patent identified above by virtue of either:

A. ☐ An assignment from the inventor(s) of the patent application / patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel , Frame , or for which a copy thereof is attached.

OR

B. ☒ A chain of title from the inventor(s), of the patent application / patent identified above, to the current assignee shown below:

1.	From: HUTTON, Glen W. To: Internet Telephone Company The document was recorded in the United States Patent and Trademark Office at Reel 007981 Frame 0020, or for which a copy thereof is attached.
2.	From: HUTTON, Glenn W. To: Internet Telephone Company The document was recorded in the United States Patent and Trademark Office at Reel 008295 Frame 0167, or for which a copy thereof is attached.
3.	From: Internet Telephone Company To: Netspeak Corporation The document was recorded in the United States Patent and Trademark Office at Reel 007981 Frame 0053, or for which a copy thereof is attached.

☒ Additional documents in the chain of title are listed on a supplemental sheet.

☒ Copies of assignments or other documents in the chain of title are attached.

As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[Note: A separate copy (*i.e.*, a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, if the assignment is to be recorded in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Michael R. Casey
Signature

3/13/09
Date

Michael R. Casey, Ph. D.
Printed or Typed Name

703-894-6400
Telephone Number

Title: Attorney, Registration No. 40,294

STATEMENT UNDER 37 CFR 3.73(B)
Continued

4.	From: STRICKLAND, Craig B. To: Netspeak Corporation The document was recorded in the United States Patent and Trademark Office at Reel 009792 Frame 0568, or for which a copy thereof is attached.
5.	From: MATTAWAY, Shane D. To: Netspeak Corporation The document was recorded in the United States Patent and Trademark Office at Reel 010012 Frame 0953, or for which a copy thereof is attached.
6.	From: Netspeak Corporation To: VOIP Technology Holdings, LLC The document was recorded in the United States Patent and Trademark Office at Reel 016522 Frame 0205, or for which a copy thereof is attached.
7.	From: VOIP Technology Holdings, LLC To: Net2Phone, Inc. The document was recorded in the United States Patent and Trademark Office at Reel 016945 Frame 0858, or for which a copy thereof is attached.
8.	From: Netspeak Corporation To: Net2Phone, Inc. The document was recorded in the United States Patent and Trademark Office at Reel 016945 Frame 0890, or for which a copy thereof is attached.
9.	From: VOIP Technology Holdings, LLC To: Net2Phone, Inc. The document was recorded in the United States Patent and Trademark Office at Reel 017105 Frame 0240, or for which a copy thereof is attached.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

POWER OF ATTORNEY,
CORRESPONDENCE ADDRESS
AND REVOCATION OF PRIOR POWERS

Hon. Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Revocation: I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

Power of Attorney: I hereby appoint the practitioners associated with customer number **42624**, individually and collectively, as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

I authorize Davidson Berquist Jackson & Gowdey, LLP to delete names/numbers of persons no longer with the Firm and to act and rely on instructions from and communicate directly with the entity who first sent this case to them and by whom I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Davidson Berquist Jackson & Gowdey, LLP in writing to the contrary.

Correspondence Address: Please recognize or change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to the address associated with Customer Number **42624**.

Assignee Name and Address:

Net2Phone, Inc.
520 Broad Street, 8th Floor
Newark, New Jersey 07102

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record The individual whose signature and title is supplied below is authorized to act on behalf of the assignee			
Signature		Date	3/12/09
Name	JAMES RAANAN	Telephone	973 438 3253
Title	EVP & DIRECTOR		

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a copy of this Power of Attorney and Statement Under 37 CFR 3.73 (B) is being served via First Class Mail on 03/13/09, upon the following:

Edwin H. Taylor
Blakely, Sokoloff, Taylor & Zafman, LLP
1279 Oakmead Parkway
Sunnyvale, California 94085

Daniel L. Jackson

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Ex Parte Reexamination of:	
U.S. Patent No. 6,108,704	Examiner: Kosowski, Alexander J
Issued: August 22, 2000	Art Unit: 3992
Application No. 90/010,416	Confirmation No. 1061
Filed: February 17, 2009	
For: Point-to-Point Internet Protocol	
Requester: Skype, Inc.	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Appraisal of Litigation Proceedings
Pursuant to MPEP §§ 2207, 2282, and 2286

Dear Sir:

Pursuant to MPEP §§ 2207, 2282, and 2286, the Requestor hereby submits copies of a Court order setting forth a hearing date of July 1, 2009 for oral arguments on the Motion to Stay in the pending litigation (Case 2:06-cv-02469-KSH-PS NET2PHONE, INC. v. EBAY, INC. et al).

Respectfully submitted,

/Thomas C. Webster/
Thomas C. Webster
Registration No. 46,154

Dated: 6-8-09

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
1279 Oakmead Parkway
Sunnyvale, California 94085-4040
Telephone: 408/720-8300
Facsimile: 408/720-8383
Attorney Docket No.: R001E

From: njdefiling@njd.uscourts.gov
Sent: Tuesday, April 28, 2009 2:46 PM
To: ecfhelp@njd.uscourts.gov
Subject: Activity in Case 2:06-cv-02469-KSH-PS NET2PHONE, INC. v. EBAY, INC. et al Order

U.S. District Court

District of New Jersey [LIVE]

Notice of Electronic Filing

The following transaction was entered on 4/28/2009 at 5:45 PM EDT and filed on 4/28/2009

Case Name: NET2PHONE, INC. v. EBAY, INC. et al
Case Number: 2:06-cv-2469
Filer:
Document Number: No document attached

Docket Text:
TEXT ORDER: Counsel are advised that the previously scheduled 5/13/09 oral argument on the pending motion to stay is rescheduled to 7/1/09 at 10:00 a.m. entered by Judge Katharine S. Hayden on 4/28/09. (rg,)

2:06-cv-2469 Notice has been electronically mailed to:

ALAN J. HEINRICH AHeinrich@Irell.com

ALAN M FISCH AFisch@kayescholer.com

ALLEN I. RUBENSTEIN arubenstein@grr.com

ANDREI IANCU aiancu@irell.com

BENJAMIN WANG bwang@irell.com

COKE MORGAN STEWART CStewart@kayescholer.com

GEORGE C. JONES gjones@GrahamCurtin.com, jcostine@GrahamCurtin.com

GILLIAN T. DIFILIPPO gdifilippo@kayescholer.com

JOSEPH M. DRAYTON jdrayton@kayescholer.com

JOSEPH P. LASALA jlasala@mdmc-law.com, sgrisez@mdmc-law.com

KATHLEEN M. FENNELLY kfennelly@grahamcurtin.com, llett@grahamcurtin.com

KEVIN JAKEL kjakel@kayescholer.com

MARIA A. SAVIO msavio@grr.com

MARKO KUO mkuo@irell.com

MORGAN CHU mchu@irell.com

PERRY GOLDBERG pgoldberg@irell.com

STEVEN STERN sstern@grr.com

STEVEN STERN sstern@grr.com

THOMAS R. CURTIN tcurtin@grahamcurtin.com, llett@grahamcurtin.com

VANDANA KOELSCH vkoelsch@kayescholer.com

VICTOR KUBLISH kubli@graysonlaw.net

WILLIAM F. O'CONNOR , JR woconnor@mdmc-law.com, wfo73@yahoo.com

2:06-cv-2469 Notice will not be electronically mailed to::

CERTIFICATE OF SERVICE FOR CONTROL NO: 90/010,416

The undersigned certifies that copies of the following:

(1) Appraisal of Litigation Proceedings; and

(2) Court Order setting forth a hearing date.

were served on

Michael R. Casey, Ph.D.
Davidson Berquist Jackson & Gowdey, LLP
4300 Wilson Blvd. – 7th Floor
Arlington, VA 22203

/Thomas C. Webster/
Thomas C. Webster
Reg. No. 46,154

Dated: /06-08-2009/

the attorney of record for the assignee of USP 6,108,704in accordance with 37 CFR § 1.915(b)(6), on the 8 day of June, 2009.

Electronic Acknowledgement Receipt

EFS ID:	5477355
Application Number:	90010416
International Application Number:	
Confirmation Number:	1061
Title of Invention:	Point-to-Point Internet Protocol
First Named Inventor/Applicant Name:	6108704
Customer Number:	42624
Filer:	Thomas Webster/Janece Shannon
Filer Authorized By:	Thomas Webster
Attorney Docket Number:	2655-0188
Receipt Date:	08-JUN-2009
Filing Date:	17-FEB-2009
Time Stamp:	21:17:29
Application Type:	Reexam (Third Party)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Reexam Miscellaneous Incoming Letter	R001E_Apprisal_6_8_09.pdf	26572 cd7550ab9f67a77bf036c31192fd0973484d20d8	no	1

Warnings:

Information:

2	Reexam Miscellaneous Incoming Letter	4_8_09_Order_6_8_09.pdf	39418 d5de53e0fac111d817dedc057816591a1adc1090	no	2
Warnings:					
Information:					
3	Reexam Certificate of Service	3801G184_CertOfService.pdf	15663 64acc0f8933bfd2a1000e0f05a12a0dff4605cf5	no	1
Warnings:					
Information:					
		Total Files Size (in bytes):	81653		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

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).

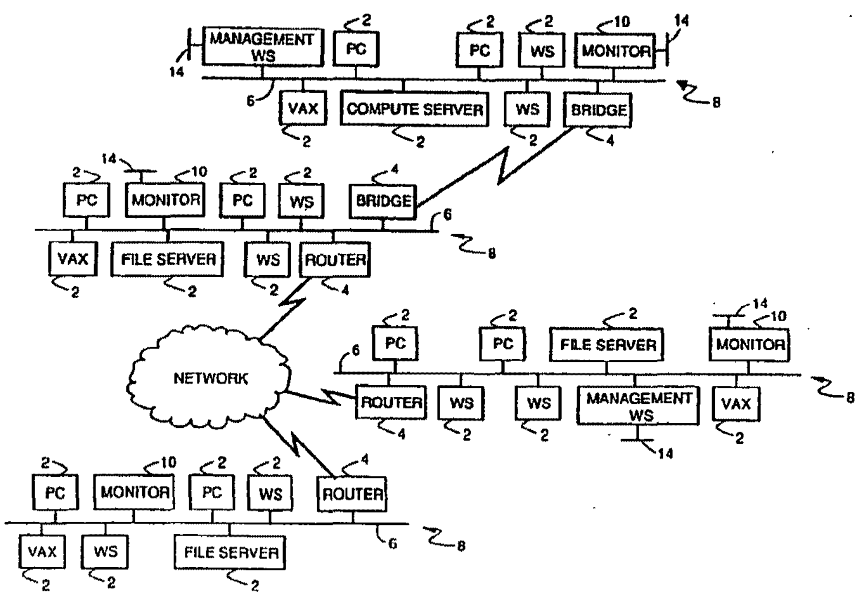
(72) Inventors: FERDINAND, Engel ; 21 Joseph Road, Northborough, MA 01532 (US). JONES, Kendall, S. ; 90 Boulder Road, Newton Center, MA 02159 (US). ROBERTSON, Kary ; 398 North Road, Bedford, MA 01739 (US). THOMPSON, David, M. ; 5127 243rd Road, Redmond, WA 98053 (US). WHITE, Gerard ; 133 Massapoag Road, 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	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	RU	Russian Federation
CG	Congo	KP	Democratic People's Republic of Korea	SD	Sudan
CH	Switzerland			SE	Sweden
CI	Côte d'Ivoire	KR	Republic of Korea	SN	Senegal
CM	Cameroon	LI	Liechtenstein	SU	Soviet Union
CS	Czechoslovakia	LK	Sri Lanka	TD	Chad
DE	Germany	LU	Luxembourg	TG	Togo
DK	Denmark	MC	Monaco	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 algorithm 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...).

<u>The Network Monitor and the Management Workstation</u>:

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.

<u>The Protocol Tree</u>:

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.

The Packet:

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.

<u>The Statistics Module (STATS)</u>:

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.

<u>The Data Structures in STATS</u>

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.

<u>The Monitor/Workstation Interface</u>:

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-