

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LG ELECTRONICS, INC., *et al.*,
Petitioners

v.

STRAIGHT PATH IP GROUP, INC.
(FORMERLY KNOWN AS INNOVATIVE
COMMUNICATIONS
TECHNOLOGIES, INC.)
Patent Owner

INTER PARTES REVIEW OF U.S. PATENT NO. 6,131,121
Case IPR No.: *To Be Assigned*

DECLARATION OF BRUCE M. MAGGS, PH.D.

TABLE OF CONTENTS

	Page
I. PERSONAL AND PROFESSIONAL BACKGROUND	1
II. MATERIALS REVIEWED AND CONSIDERED	4
III. THE BASICS OF NETWORK COMMUNICATION	6
A. Computer Network Hardware Configurations	6
B. Network Protocols	8
C. Assigning Network Addresses to Devices	9
D. Mapping Names to IP Addresses	13
E. Looking Up the IP Address of a Network Device, Including Those with Dynamically Assigned Addresses	13
F. Point-to-Point Communications	16
IV. OVERVIEW OF THE '121 PATENT (EX. 1001)	17
A. Summary of the Alleged Invention	17
1. Step 1: Processing Units Obtain Dynamically Assigned IP Addresses	18
2. Step 2: Processing Units Register Their IP Addresses and Identifiers with a Connection Server	18
3. Steps 3 & 4: First Processing Unit Sends Query to Connection Server, Which Returns IP Address of Second Processing Unit	19
4. Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communication with Second Processing Unit	20
B. Original Prosecution of the '121 Patent	20
C. Prior <i>Ex Parte</i> Reexamination of the '121 Patent	21
D. The Sipnet <i>Inter Partes</i> Review for the '704 Patent	23
V. Overview of the Primary Prior Art References	23
A. WINS (Ex. 1003)	23
1. Step 1: Processing Units Obtain Dynamically Assigned IP Addresses from DHCP Servers	24

2.	Step 2: Processing Units Register Their IP Addresses and Identifiers with the WINS Server	27
3.	Steps 3 & 4: First Processing Unit Sends Query to WINS Server and Receives the IP Address of the Second Processing Unit	31
4.	Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communication with Second Processing Unit	32
B.	NetBIOS (Ex. 1004).....	32
1.	Step 1: Processing Units Have Assigned IP Addresses.....	33
2.	Step 2: Processing Units Register Their IP Addresses and Identifiers with the NBNS	34
3.	Steps 3 & 4: First Processing Unit Sends Query to the NBNS and Receives the IP Address of the Second Processing Unit	35
4.	Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communications with Second Processing Unit	36
VI.	SPECIFIC GROUNDS FOR PETITION.....	37
A.	Claims 3, 4, and 6-14 of the '121 Patent Would Have Been Obvious Over WINS and NetBIOS	37
1.	Level of Ordinary Skill in the Art.....	40
2.	One Skilled in the Art Would Have Been Motivated to Combine WINS and NetBIOS	41
3.	Claim 3 (Independent)	42
4.	Claim 4 (Depends From Claim 3).....	50
5.	Claim 6 (Independent)	55
6.	Claim 7 (Independent)	57
7.	Claim 8 (Independent)	59
8.	Claim 9 (Independent)	61
9.	Claim 10 (Independent)	63
10.	Claim 11 (Independent)	64
11.	Claim 12 (Independent)	65

12.	Claim 13 (Independent)	66
13.	Claim 14 (Independent)	68
B.	Examples Where Prior Art Teaches Claimed Features.....	69

I, Bruce M. Maggs, Ph.D., declare:

1. I have been retained by counsel for the Petitioners to submit this declaration in connection with Petitioners' Petition for *Inter Partes* Review of Claims 3, 4, and 6-14 of U.S. Patent No. 6,131,121 ("the '121 patent") (Ex. 1001). I am being compensated for my time at a rate of \$700 per hour, plus actual expenses. My compensation is not dependent in any way upon the outcome of this Petition.

I. PERSONAL AND PROFESSIONAL BACKGROUND

2. I am an expert in the field of computer systems and networking, including network communication protocols and database design. I have studied, taught, practiced, and researched in the field of Computer Science for approximately twenty-five years.

3. I received a Ph.D. in Computer Science from the Massachusetts Institute of Technology in 1989, a Master of Science degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 1986, and a Bachelor of Science degree in Computer Science from the Massachusetts Institute of Technology in 1985.

4. I have been a Professor of Computer Science at Duke University since July 2009, where I first served as a Visiting Professor, and then became a tenured full Professor in January 2010. On July 1, 2011, I became the Pelham Wilder Professor of Computer Science in the Trinity College of Arts and Sciences at Duke.

Prior to joining Duke, I was a full Professor of Computer Science at Carnegie Mellon University. I joined Carnegie Mellon as an Assistant Professor in January 1994, was promoted to Associate Professor in July 1997, was given tenure in July 1999, and was promoted to full Professor in 2004. From September 2007 through August 2008, I was a Visiting Professor in the Department of Computer Science at Duke University, and from September 1998 through January 1999, I was a Visiting Associate Professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology.

5. At Carnegie Mellon and Duke, I have taught a variety of courses related to the '121 patent. For example, at Carnegie Mellon, I taught undergraduate courses titled "Introduction to Computer Systems" and "Computer Networks." At Duke, I taught a graduate course on "Computer Networks and Distributed Systems." In these courses, students are asked to perform programming assignments such as building a web server, or building a web proxy. I have also taught related courses at Carnegie Mellon, such as a graduate course on "Basic Computer Systems" and an undergraduate course on "Operating System Design and Implementation."

6. I have had extensive experience in both industry and academia as it relates to the technical fields relevant here. I helped launch Akamai Technologies, a leading provider of services for accelerating content and business processes on the Internet. I retain a part-time role at Akamai as Vice President for Research. I also

worked as a research scientist at the NEC Research Institute, Inc. for approximately four years, where I conducted research on networking and parallel computing.

7. I have lectured and published extensively on computer systems and networking, including lectures and papers relating to content delivery over the Internet, improved network routing, database scalability and management, server reliability, the Domain Name System, source location, data management, and peer-to-peer networks.

8. Governmental agencies, such as the National Science Foundation and the Defense Advanced Research Projects Agency, and industrial grants from, for example, Sun Microsystems and NEC Research Institute, have provided funding for my research. My federally and corporately funded research has addressed areas such as computer networking and Internet protocol and system designs.

9. Additionally, I was elected to the Council of the Association for Computing Machinery, and have served as a member of the DARPA Information Science and Technology Study Group. I have also served three times on the program committee of the premier conference in computer networking, ACM SIGCOMM, served on both the Program and Steering Committees of the ACM Internet Measurement Conference, and chaired the first IEEE Workshop on Hot Topics in Web Systems and Technologies.

10. A copy of my curriculum vitae is attached as Exhibit 1017, which contains further details on my education, experience, publications, patents, and other qualifications to render an expert opinion in connection with this proceeding.

II. MATERIALS REVIEWED AND CONSIDERED

11. In connection with my work on this matter, I have reviewed the '121 patent (Ex. 1001) and the following other documents:

Exhibit	Description
1001	U.S. Patent No. 6,131,121 (“’121 patent”)
1002	Declaration of Bruce M. Maggs, Ph.D. (“Maggs Decl.”)
1003	Microsoft Windows NT Server Version 3.5 (“WINS”)
1004	Technical Standard: Protocols for X/Open PC Interworking: SMB, Version 2 (“NetBIOS”)
1005	Intentionally Left Blank
1006	Windows NT Server 3.5 TCP/IP Documentation [TCPIP.HLP]
1007	Windows NT Server Copyright Registration
1008	Windows NT Networking Guide
1009	Windows NT Networking Guide Copyright Registration
1010	Petition for Inter Partes Review of U.S. Patent No. 6,108,704, <i>Sipnet EU S.R.O. v. Straight Path IP Group, Inc.</i> (IPR No. 2013-00246) (April 11, 2013)
1011	Institution Decision in <i>Sipnet EU S.R.O. v. Straight Path IP Group, Inc.</i> (IPR No. 2013-00246) (Oct. 11, 2013)
1012	<i>Straight Path IP Group, Inc. v. Bandwidth.com, Inc. et al.</i> , No. 1:13-cv-0932 (E.D.V.A. Feb. 25, 2014) (Dkt. 107, Claim Construction Order)

Exhibit	Description
1013	IETF RFC 1541, October 1993 (“Dynamic Host Configuration Protocol”) (“DHCP”)
1014	IETF RFC 1034, November 1987 (“Domain Names - Concepts And Facilities”) (“Domain Names RFC 1034”)
1015	IETF RFC 1035, November 1987 (“Domain Names - Implementation And Specification”) (“Domain Names RFC 1035”)
1016	IETF RFC 791, September 1981 (“Internet Protocol”)
1017	Curriculum Vitae of Dr. Bruce M. Maggs
1018	Excerpt from File History for U.S. Patent No. No. 6,108,704 (December 2, 1997 Amendment)
1019	U.S. Patent No. 5,159,592 (“Perkins”)
1020	Excerpt from File History for ’121 Patent (Sept. 7, 1999 Amendment)
1021	Excerpt from Reexamination File History for ’121 Patent (May 17, 2010 Office Action)
1022	Excerpt from Reexamination File History for ’121 Patent (July 19, 2010 Amendment)
1023	Excerpt from File History for ’121 Patent (Aug. 23, 2000 Notice of Allowability)
1024	Excerpt from Reexamination for File History for ’121 Patent (Nov. 25, 2009 Response to Non-Final Rejection)
1025	Excerpt from Reexamination for File History for ’121 Patent (Aug. 16, 2010 Notice of Intent to Issue Reexamination Certificate)
1026	Excerpt from File History for ’121 Patent (March 4, 1999 Non-Final Rejection)
1027	Excerpt from Reexamination File History for ’121 Patent (August 25, 2009 Office Action)

Exhibit	Description
1028	Final Written Decision in <i>Sipnet EU S.R.O. v. Straight Path IP Group, Inc.</i> (IPR No. 2013-00246) (Oct. 9, 2014)

I also have relied on my academic and professional experience in reaching the opinions expressed in this declaration.

III. THE BASICS OF NETWORK COMMUNICATION

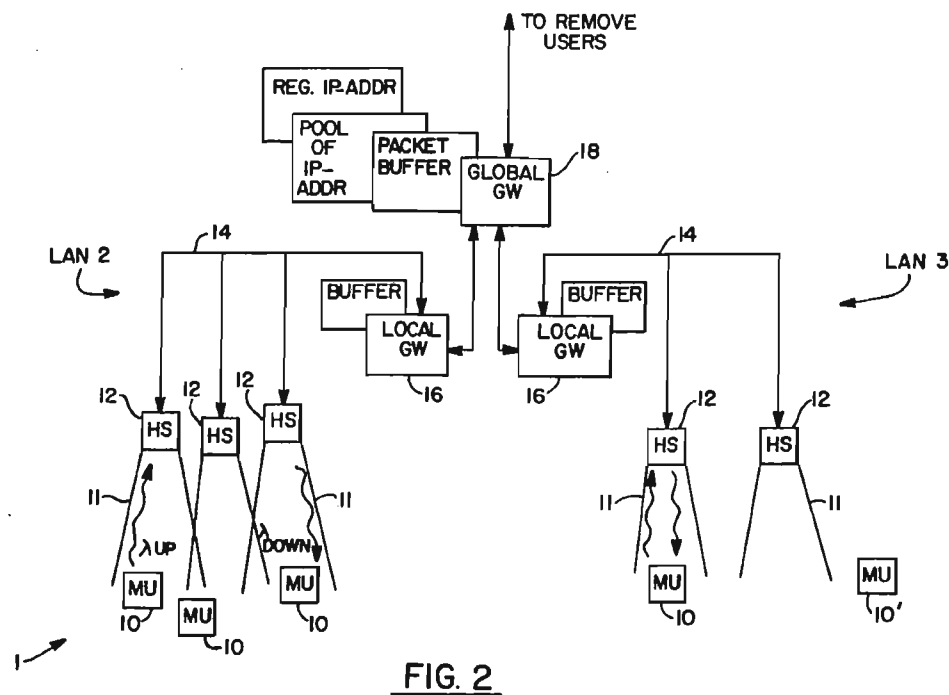
A. Computer Network Hardware Configurations

12. The '121 patent does not claim to invent a new type of networking technology or network hardware. Indeed, networking components such as computers, servers, routers, and gateways were all well known in the art when the '121 patent was filed in September 1996, and when the application to which the '121 patent claims priority was filed in September 1995.

13. There are many possible network configurations. One example is a “Local Area Network” or “LAN,” which interconnects computers within a limited geographic area such as a home, school, computer laboratory, or office building. Another type of network configuration is a “Wide Area Network” or “WAN,” which connects computers over a broad geographic area, such as across a city, a country, or internationally. The Internet is a widely known example of a WAN. A LAN can be connected to a WAN, such as the Internet, via a gateway that acts as an interface between the LAN and the WAN. These types of network configurations were well

known before September 1995, and the '121 patent does not claim to invent a new type of network configuration.

14. In addition, it also was well known before September 1995 how to couple one type of network (e.g., a LAN) with one or more other types of networks (e.g., a WAN or the Internet). For example, as shown below, in October 1990, U.S. Patent No. 5,159,592 to Perkins ("Perkins") (Ex. 1019) disclosed a communication area network 1 that includes one or more local area networks (LANs 3 and 4) that (via local gateway 16 and global gateway 18) are "coupled to remote network users who may be dispersed over a wide geographic area":



(Ex. 1019, 3:56-68, 4:21-27, Fig. 2.)

B. Network Protocols

15. There were many network protocols in existence long before September 1995. For example, the '121 patent references several prior art protocols, including the Transmission Control Protocol and Internet Protocol (TCP/IP), the Serial Line Internet Protocol (SLIP), the Point-to-Point Protocol (PPP), the Post Office Protocol (POP) and Simple Mail Transfer Protocol (SMTP). (Ex. 1001, 2:5-13 (explaining “devices interfacing to the Internet and other online services may communicate with each other” using the “Internet Protocol (IP)” or “Serial Line Internet Protocol or Point-to-Point Protocol (SLIP/PPP)”); *id.*, 5:7-9 (discussing POP and SMTP).)

16. As of the claimed priority date of the '121 patent, many network communication protocols had been standardized by the Internet Engineering Task Force (“IETF”), which codifies protocols in documents called “Requests for Comments” or “RFCs” that are widely distributed and used by engineers in designing networks and network products. The IETF first defined the Internet Protocol (“IP”) in RFC 791 (published in September 1981), which led to the “Internet Protocol” standard in 1981. (Ex. 1016.) The Internet Protocol forms the basis of the modern Internet and other computer networks. As I discuss further below, among other things, the Internet Protocol Suite provides mechanisms for network devices to identify themselves on a network (via a network address and/or

name), and to locate and communicate with other devices also participating on the network.

C. Assigning Network Addresses to Devices

17. As the '121 patent explains, the prior art Internet Protocol identifies devices participating on the network using a unique series of numbers, commonly represented as four values ranging from 0 to 255, separated by periods (e.g., 151.207.247.130). (Ex. 1001, 2:17-23 (“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.10, and each user may be allocated a successive IP address of XXX.XXX.XXX.11, XXX.XXX.XXX.12, etc.”).) As I discuss further below, the Internet Protocol provided a way for a networked device having one IP address to direct data to another networked device with a different IP address.

18. Some IP addresses are “static.” Assigning static addresses typically requires a user or network administrator to configure the device manually with the static address. The idea of assigning static network addresses was known before September 1995, and the '121 patent does not claim to invent a new way to assign static network addresses. (Ex. 1001, 2:30-35 (discussing prior art use of

“[p]ermanent IP addresses of users and devices accessing the Internet” and “dedicated IP addresses”).)

19. As the number of networked computers increased significantly during the 1980s, concerns increased about a shortage of available IP addresses. One way that network engineers addressed this issue was to assign “dynamic” IP addresses to devices – a process in which a host or server assigns an IP address to a first device, can re-assign that address to another device (e.g., after a certain period of time, or when the first device is turned off or moves outside of the network), and assigns a new IP address to the first device if the first device later seeks to resume participation on the network.

20. The idea of assigning IP addresses dynamically was well known before September 1995, and the ’121 patent does not claim to invent a new way to assign addresses. (Ex. 1001, 2:23-29 (explaining that, in the prior art, “temporary IP addresses may be reassigned or recycled to the users, for example, as each user is successively connected to an outside party” and that “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”); *id.*, 2:35-38 (discussing “the dynamic nature of temporary IP addresses of some [prior art] devices accessing the Internet”).)

21. One prior art example is the Dynamic Host Configuration Protocol (DHCP), which provided a framework for dynamically assigning IP addresses. (Ex.

1013.) Under DHCP, a device within a computer network acts as a DHCP server, and assigns IP addresses to devices joining the network. A device leaving the network releases its assigned IP address back to the DHCP server so that the server can reassign that address to another device that later joins the network. In some cases, a device periodically renews its address with the DHCP server (after a set period of time) so the DHCP server can distinguish between those devices still using their assigned addresses and those that are not. The DHCP server thus tracks which devices remain online, and reassigns addresses not in use to other devices. (Ex. 1013 at 11, 16, 35.)

22. Prior to September 1995, the computer network industry also developed other mechanisms that enabled devices, such as computers, to update a name resolution database as new network addresses were assigned to devices. For example, in October 1990, Perkins disclosed a network that dynamically assigned addresses to devices when they joined a network – by having the device transmit “a unique identifier, such as its serial number” to a global gateway, and having the global gateway dynamically assign the device a “pseudo-IP” network address “either on a temporary basis (one network session) or on a permanent, extended basis (several network sessions).” (Ex. 1019, 4:49-60, 4:63-64, 5:52-65 (“[A] permanent assignment is preferably not permanent in the sense that the mobile unit 10 would own the address for all time.... Preferably, the permanent assignment is

only sufficiently long so as to accomplish a specific task which may require a plurality of separate network sessions.”); *id.*, 5:12-18; Fig. 3 (showing assignment of pseudo-IP addresses transmitted from global gateway to mobile unit and local gateway).)

23. Perkins also disclosed that the local and global gateways could track which devices were currently “active” and “inactive” on the network. (*Id.*, 5:34-42 (explaining that if a registered mobile unit 10 goes inactive, “the mobile unit’s local gateway 16 [] notif[ies] the global gateway 18, via LAN 14, that the mobile unit 10 is no longer a member of the group of mobile units associated with the local gateway 16”); *id.*, 6:8-18, 6:60-64 (“If a mobile unit 10 intends to terminate incoming network service in an orderly manner it notifies the local gateway 16 via a header station 12. The local gateway 16 notifies the global gateway 18 that the mobile unit’s pseudo-IP address may be deallocated.”); *id.*, 5:5-9 (“A mobile unit 10 typically maintains its assigned pseudo-IP address until it is turned off, or until the network session is actively terminated.”); *id.*, claim 3 (de-assigning network addresses for mobile units that are “no longer active”); *id.*, claim 6 (re-assigning network addresses for a mobile unit that “is once more actively coupled to the wireless network”); *id.*, claim 11 (involving, for inactive devices, “deassigning the assigned network address at the network gateway”).)

D. Mapping Names to IP Addresses

24. Most users cannot easily remember, and prefer not to use, the lengthy numeric IP addresses associated with network devices. Therefore, since the 1980s, almost all network services have allowed users to map the text name of a particular device to the network address. For example, users can send an email by using the recipient's email address, rather than the numeric network address of the recipient's mail server. Likewise, the "Domain Name System" (DNS), which was developed in the 1980s, allows users to input an Internet domain name for a website rather than the numeric IP address of the website. (Ex. 1014-15.)

25. Systems that allow mapping names to IP addresses must include a mechanism to track which address is currently assigned to a specific name. Name servers that store names and corresponding IP addresses existed long before September 1995.

26. For example, Perkins explained that a global gateway (that assigned IP addresses to devices) could forward name/addressing information for devices to a "nameserver." (Ex. 1019, 7:7-20, Fig. 6, claim 12.)

E. Looking Up the IP Address of a Network Device, Including Those with Dynamically Assigned Addresses

27. In situations where the user of a first device knows the current address of a second device, the user of the first device can establish "point-to-point" communications with the second device by addressing communications directly to

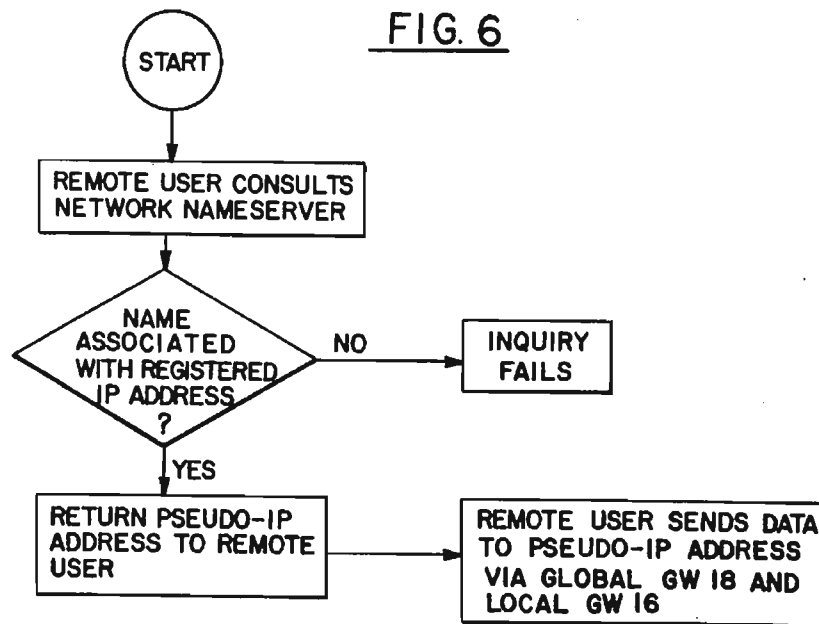
the network address of the second device. (Ex. 1019, 6:35-38, 7:5-7 (explaining that “[a]ll communication from a remote user to a mobile unit 10 employs the pseudo-IP address of the mobile unit 10,” and that if a remote user already knows the pseudo-IP address of a desired mobile unit, the remote user can establish point-to-point communications directly with the mobile unit by “employing known IP protocols”).)

28. However, if the user of the first device only knows the name of the second device, the user needs a way to look up the IP address of the second device by using its known name. This type of lookup system – including for dynamically assigned IP addresses – was well known in the prior art.

29. For example, in a DNS system, when a user enters a domain name (e.g., www.cnn.com) into their web browser, the browser sends a request to the DNS server, asking it to “resolve” that domain name to an IP address (sometimes referred to as “name resolution”). Using its stored list of name/IP address mappings, the DNS server looks up the IP address (e.g., 156.166.226.25) assigned to the queried domain name, and provides that numeric address to the browser (which the browser then uses to connect directly with the web site). (Ex. 1015 [RFC 1035, November 1987 (“Domain Names - Implementation And Specification”)].)

30. In a system involving the dynamic assignment of IP addresses, a look up mechanism must account for the fact that the address mapped to a device may change over time and must provide a means for keeping track of such changes.

31. Perkins explained in 1990 that if a user did not know the dynamically assigned (“pseudo-IP”) address of a device, the user could request the address by sending a name query to a “nameserver,” as depicted below:



(Ex. 1019, Fig. 6; *id.*, 7:7-17 (“When a remote user initiates a conversation with a mobile unit 10 the remote user typically consults a network nameserver configured to send requests for specified mobile unit 10 names to a specified mobile unit 10 global gateway 18.”).) If the nameserver determined that the queried name was associated with a registered dynamically assigned address, it returned that address to the user. (*Id.*, 7:17-20, Fig. 6, claim 12 (discussing steps involved “in response to a name inquiry to determine a network address that is associated with a name”).) In contrast to the “permanently stored” serial number, the pseudo-IP address of a

mobile unit may be “permanently assigned to that mobile unit” or “dynamically allocated.” (*Id.*, 5:57-65.)

F. Point-to-Point Communications

32. Once a user obtains the IP address of another device in response to a name query, the user can communicate to that device directly using the received IP address – what the ’121 patent calls “point-to-point” communications. This concept also was well known before September 1995.

33. For example, Perkins explained in 1990 that a remote user could establish point-to-point communications with the mobile unit 10 by using the pseudo-IP address received from the nameserver. (Ex. 1019, 7:37-39 (“If a remote user obtains the pseudo-IP address of a registered mobile unit 10, the remote user is enabled to send messages, such as mail, to the mobile unit 10 ...”); *id.*, 7:44-46 (describing “TCP session requests for the mobile unit 10 from the remote user”); *id.*, 8:15-19 (noting that, for point-to-point communications, “the remote user is enabled to deliver the mobile unit 10 packets directly to the mobile unit’s local gateway 16, without requiring the intervention of the global gateway 18”); *id.*, Fig. 6 (showing “Remote User Sends Data to Pseudo-IP Address Via Global GW 18 and Local GW 16”); *id.*, claims 1, 13, 19); *id.*, 7:54-56 (“A mobile unit 10 delivering a packet to a remote user employs conventional methods of network transmission and uses the IP address of the remote user.”).)

IV. OVERVIEW OF THE '121 PATENT (EX. 1001)

A. Summary of the Alleged Invention

34. The '121 patent concedes that, in the prior art, a first “processing unit” or “process”¹ could establish “point-to-point communications” with a second processing unit using the network address of the second processing unit “in a manner known in the art.” (Ex. 1001, 2:6-8 (“[D]evices interfacing to the Internet and other online services may communicate with each other upon establishing respective device addresses.”); *id.*, 2:32-34, 9:47-51, 10:5-9 (“[P]oint-to-point communications of voice and video signals over the Internet” “may be established as shown in FIGS. 3-4 in a manner known in the art” and “may be conducted in a manner known in the art between the first and second users through the Internet 24”).)

35. According to the '121 patent, however, point-to-point communication was “difficult to attain” between devices with “temporary IP addresses” (i.e., dynamically assigned IP addresses that “may be reassigned or recycled” over time). (Ex. 1001, 2:25-46, 2:58-61.) The '121 patent claimed to solve that supposed

¹ The specification of the '121 patent refers to both “processing units” and “processes.” For the sake of convenience, I will use “processing units” in Sections IV-V of this declaration.

“problem” through the basic lookup feature described in connection with Figure 8.
(*Id.*, Fig. 8.)

1. Step 1: Processing Units Obtain Dynamically Assigned IP Addresses

36. When a processing unit “logs on to the Internet ..., the [processing unit] is provided a dynamically allocated IP address by an Internet service provider.” (Ex. 1001, 5:51, 6:53-59, 13:2-5, 13:59-60, 15:27-32, 18:10-14, Fig. 13A.)

2. Step 2: Processing Units Register Their IP Addresses and Identifiers with a Connection Server

37. After receiving its address, a processing unit “automatically transmits ... its dynamically allocated IP address to the connection server 26,” which “stores these addresses in the database 34” (Ex. 1001, 6:53-66; *id.*, 12:9-12 (“[C]onnection server 26 ... timestamps and stores E-mail and IP addresses of logged-in users and processing units in the database 34.”); *id.*, 12:41-46 (connection server provides “directory and information related services.”); *id.*, 18:19-29 (“one-to-one mapping” of “identifier” and “current IP address” for each “client process”); *id.*, 22:50-23:40 (describing “start up”); Ex. 1020 [9/7/99 Amend.] at 19.)

38. Connection server 26 keeps “relatively current” data concerning the “on-line status” of registered processing units; e.g., it may confirm a registered processing unit remains online after “predetermined time periods, such as a default value of 24 hours....” (Ex. 1001, 7:8-13.) Alternatively, “[w]hen 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 offline.” (*Id.*, 7:43-46.)

3. Steps 3 & 4: First Processing Unit Sends Query to Connection Server, Which Returns IP Address of Second Processing Unit

39. To establish point-to-point communications with a second processing unit, the first processing unit “sends a query ... to the connection server 26” that includes the e-mail address or “name or alias ... of a party to be called.” (Ex. 1001, 7:24-29, 11:13-19, 11:59-63, 12:12-16; Ex. 1023 [9/7/99 Amend.] at 19-20.) Connection server 26 then “searches the database 34 to determine whether the [second processing unit] is logged-in by finding any stored information ... indicating that the [second processing unit] is active and on-line.” (Ex. 1001, 7:24-33 (“If the [second processing unit] is active and on-line ... the IP address of the [second processing unit] is retrieved from the database 34 and sent to the first [processing unit].”); *id.*, 12:17-20 (“Connection server 26 retrieves the IP address of the specified user from the database 34 in step 70 ... and sends the retrieved IP address to the first processing unit 12....”); *id.*, 18:30-19:9, 23:56-24:10 (server looks-up and returns requested IP addresses); *id.*, 23:45-52 (user inputs “callee party’s name or alias” “to obtain the current dynamically assigned Internet Protocol address of the prospective callee”); Ex. 1020 [9/7/99 Amend.] at 20.)

4. **Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communication with Second Processing Unit**

40. After receiving the IP address of the second processing unit from connection server 26, “[t]he first processing unit 12 may then directly establish the point-to-point Internet communications with the [second processing unit] using the IP address of the [second processing unit].” (Ex. 1001, 7:33-36; *id.*, 12:16-21 (received IP address “enable[s] first processing unit to establish point-to-point communications with specified second user.”).)

41. The ’121 patent does not claim to invent point-to-point communications, or even a new type of point-to-point communications. Rather, it admits the claimed point-to-point communications “may be established as shown in FIGS. 3-4 *in a manner known in the art*” and “may be conducted *in a manner known in the art* between the first and second users through the Internet 24.” (Ex. 1001, 9:47-51, 10:5-9 (emphases added).) The claimed invention also functions the same regardless of “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.”) (*Id.*, 9:25-28.)

B. Original Prosecution of the ’121 Patent

42. During prosecution of the ’121 patent, the Examiner rejected all claims, but later issued the ’121 patent after the patentee cancelled claims and amended and

added others. (Ex. 1026 [3/4/99 Non-Final Rejection]; Ex. 1020 [9/7/99 Amend.]; Ex. 1023 [8/23/00 Notice of Allowability].) In forming my opinions I have not relied on any prior art cited during that prosecution.

C. Prior *Ex Parte* Reexamination of the '121 Patent

43. In February 2009, a third party requested *ex parte* reexamination of claims 6-14 of the '121 patent. During that proceeding, the Examiner rejected claims 6-11 under 35 U.S.C. § 102(b) in light of NetBIOS, and rejected claims 12-14 under 35 U.S.C. § 103(a) in light of NetBIOS and RFC 1531 ("Dynamic Host Configuration Protocol" (DHCP)) (Ex. 1027 [8/25/09 Office Action].) The Examiner agreed that NetBIOS describes the same address lookup function claimed in the '121 patent, but was persuaded by an expert declaration arguing that "bringing dynamic addressing into a NetBIOS type system would create a new set of obstacles that would need to be solved" such that "one of ordinary skill in the art would [not] have been motivated to combine NetBIOS and [DHCP]." (*Id.*; Ex. 1021 [5/17/10 Office Action]; Ex. 1024 [11/25/09 Response].) During the reexamination, the Patentee amended claims 6-8 to require "dynamically assigned" network protocol addresses. (Ex. 1022 [7/19/10 Amend.] at 2-3.)

44. The expert declaration failed to note that prior art, including WINS, disclosed using dynamic addressing in a NetBIOS-type system. (*E.g.*, Ex. 1003 at 13 (DHCP servers allow "users [to] take advantage of dynamic IP address allocation

and management.”); *id.* at 62 (“DHCP offers dynamic configuration of IP addresses for computers.”); *id.* at 41 (“Microsoft Windows networking provides dynamic name resolution for NetBIOS computer names via WINS servers and NetBIOS over TCP/IP.”); *id.* at 65 (“WINS is a NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as p-node.”); *id.* at 66 (“When WINS servers are in place on the network, NetBIOS over TCP/IP resolves names on a client computer by communicating with the WINS server.”; “NetBIOS over TCP/IP is the session-layer network service that performs name-to-IP address mapping for name resolution. This section describes the modes of NetBIOS over TCP/IP, as defined in RFCs 1001 and 1002 to specify how NetBIOS should be implemented over TCP/IP.”); *id.* at 67 (“All computers register themselves with the WINS server, which is a NetBIOS Name Server (NBNS) with enhancements.”); *id.* at 69 (“[W]hen dynamic addressing through DHCP results in new IP addresses for computers that move between subnets, the changes are automatically updated in the WINS database,” which “is based on and is compatible with the protocols defined for [NetBIOS Name Server] in RFCs 1001/1002, so it is interoperable with any other implementation of these RFCs.”).)

45. Relying on the flawed declaration, the Examiner confirmed claims 12-14, and allowed amended versions of claims 6-11, explicitly limited to

“dynamically assigned” addresses. (Ex. 1021 [5/17/10 Office Action]; Ex. 1025 [8/16/10 Office Action]; Ex. 1022 [7/19/10 Amend.] at 2-3.)

D. The Sipnet *Inter Partes* Review for the ’704 Patent

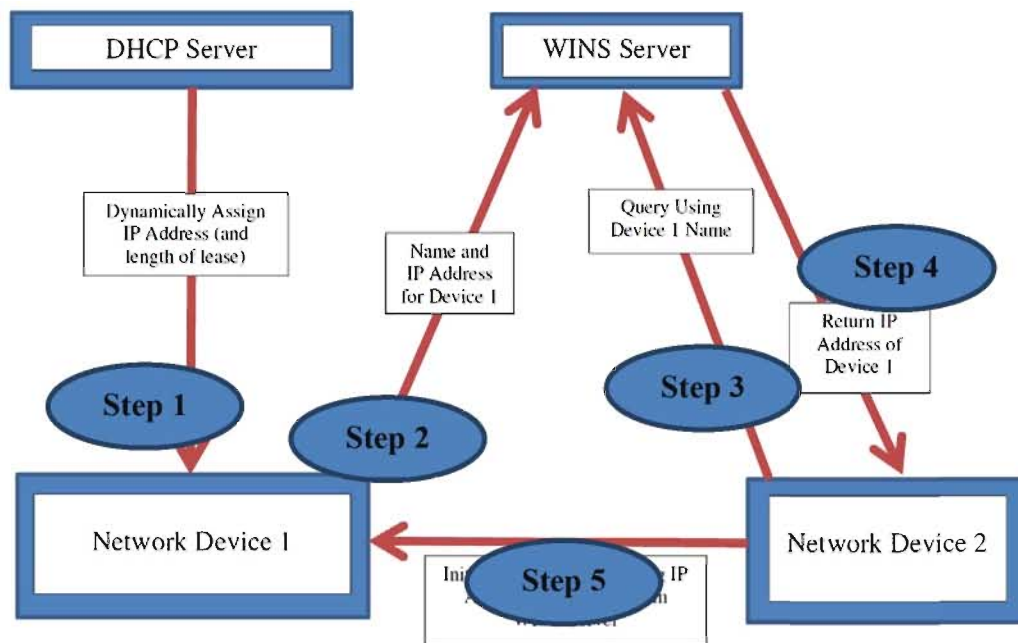
46. In April 2013, a third party (“Sipnet”) initiated an IPR seeking to cancel claims of the ’704 patent (parent of the ’121 patent), based on, *inter alia*, WINS and NetBIOS. On October 11, 2013, the Board instituted an IPR for all challenged claims and found a reasonable likelihood that WINS and NetBIOS each anticipated most claims and rendered the others obvious. (Ex. 1011 at 20-21.) On October 9, 2014, the Board determined that Sipnet had shown by a preponderance of the evidence that claims 1-7 and 32-42 of the ’704 patent are unpatentable. (Ex. 1028.)

V. Overview of the Primary Prior Art References

A. WINS (Ex. 1003)

47. In 1994, Microsoft published and publicly distributed WINS, which “describes how to install, configure, and troubleshoot Microsoft TCP/IP on a computer running the Microsoft Windows NT Workstation or Windows NT Server operating system,” including “the software to support new dynamic configuration and name resolution services.” (Ex. 1003 at 12.) WINS teaches how a processing unit (e.g., a computer running the Windows NT or Windows for Workgroups operating system) or process (e.g., the Windows NT or Windows for Workgroups operating system running on a computer) can initiate point-to-point communications

with other processing units or processes using the same lookup feature as in the '121 patent.²



1. **Step 1: Processing Units Obtain Dynamically Assigned IP Addresses from DHCP Servers**

48. WINS explains that each processing unit participating on a Windows NT network “must be assigned a unique IP address,” and that Windows NT

²WINS explains that the Windows NT operating system uses the TCP/IP protocol “to communicate with Windows NT systems, with devices that use other Microsoft networking products, and with non-Microsoft systems, such as UNIX.” (Ex. 1003 at 18, 19 (TCP/IP allows Windows NT to “connect [to] the Internet” and to “private internetworks”); *id.* at 27-28 (examples of non-Microsoft networks); *id.* at 55-56 (history and operation of TCP/IP).)

networks can use a DHCP server to assign IP addresses “automatically” to processing units at start up—including dynamically. (Ex. 1003 at 57-58, 62 (manually “[a]ssigning and maintaining IP address information can be an administrative burden” and “[DHCP] was established to relieve this administrative burden ... through centralized management of address allocation” and “dynamic configuration of IP addresses for computers.”); *id.* at 13 (“When DHCP servers are installed on the network, users can take advantage of dynamic IP address allocation and management.”); *id.* at 23 (Windows TCP/IP supports RFC 1541, the “Dynamic Host Configuration Protocol (DHCP)”); *id.* at 83 (“A [DHCP] server is a Windows NT Server computer running ... Microsoft TCP/IP and the DHCP-compatible server software” which “is defined in [RFCs] 1533, 1534, 1541, and 1542.”); *id.* at 85-120 (DHCP options, including for “small LAN” and “large enterprise network”); *id.* at 113 (“Allocation of IP addresses for distribution by DHCP servers can be done dynamically or manually Dynamic allocation allows a client to be assigned an IP address from the free address pool.”).)

49. “During system startup,” each processing unit wishing to participate on the network sends a “*discover message*” to all DHCP servers, each of which “responds with an *offer message* containing an IP address and valid configuration information for the client that sent the request.” (Ex. 1003 at 63.) After the processing unit chooses one of the DHCP server offers, “[t]he selected DHCP server

sends a *DHCP acknowledgment message* that contains the [IP] address first sent during the discovery stage, plus a valid lease for the address and the TCP/IP network configuration parameters for the client.” (*Id.* at 64; *id.* at 84 (“When a DHCP client computer is started, it communicates with a DHCP server to receive the required TCP/IP configuration information ... includ[ing] at least an IP address and submask plus the lease associated with the configuration.”).)

50. The registered processing unit can then participate on the network using the dynamically assigned IP address until either the lease expires or the processing unit leaves the network. (Ex. 1003 at 62 (Assigned IP address “is released automatically for a DHCP client computer that is removed from a subnet”); *id.* (“The system administrator controls how IP addresses are assigned by specifying *lease* durations, which specify how long a computer can use an assigned IP address before having to renew the lease with the DHCP server.”); *id.* at 63 (“DHCP uses a client-server model and is based on leases for IP addresses.”); *id.* at 64 (“As the lease approaches its expiration date, [the device] attempts to renew its lease with the DHCP server, and is assigned a new address if the current IP address lease cannot be renewed.”); *id.* at 113 (“[D]ynamically allocated addresses can be returned to the free address pool if the client computer is not being used, if it is moved to another subnet, or if its lease expires.”); *id.* at 114 (devices can renew their leases, or if that

fails, obtain a new IP address).) The lease period can either be “short” (e.g., “a few days”) or “long” (e.g., “two months”). (*Id.* at 115.)

2. Step 2: Processing Units Register Their IP Addresses and Identifiers with the WINS Server

51. WINS explains that, “[a]lthough TCP/IP uses IP addresses to identify and reach computers, users typically prefer to use computer names.” (Ex. 1003 at 41; *id.* at 65 (“Computers use IP addresses to identify each other, but users usually find it easier to work with computer names.”).) Thus, WINS teaches that Windows NT requires that each device have a “NetBIOS” name. (*Id.* at 64 (“Configuring Windows NT with TCP/IP requires the IP address and computer name, which are unique identifiers for the computer on the network.”); *id.* at 74 (“Name registration ensures that the computer’s name and IP address are unique for each device.”); *id.* at 262 (“In Windows NT, the computer name is set by choosing the Network icon in Control Panel, and it is a name of up to 15 uppercase characters”).)

52. WINS further teaches that Windows NT can use Windows Internet Name Service (WINS) servers—which consist of “a Windows NT Server computer running Microsoft TCP/IP and the [WINS] server software” and “contain a dynamic database mapping computer names to IP addresses” in accordance with “a NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as p-node.” (Ex. 1003 at 65, 121; *id.* at 151 (WINS database entry including “NetBIOS computer name” and “assigned Internet Protocol address”); *id.* at 20 (“Microsoft TCP/IP includes ...

NetBIOS for establishing logical names and sessions on the network” and “Windows Internet Name Service (WINS) for dynamically registering and querying computer names on an internetwork....”); *id.* at 21 (WINS and NetBIOS services are “[c]ore” services “Integrated with Windows NT”); *id.* at 22 (listing RFC 1001 and 1002 as “NetBIOS Service Protocols” supported by Windows NT); *id.* at 41 (“Microsoft Windows networking provides dynamic name resolution for NetBIOS computer names via WINS servers and NetBIOS over TCP/IP.”); *id.* at 66 (“NetBIOS over TCP/IP is the session-layer network service that performs name-to-IP address mapping for name resolution.”); *id.* at 67 (“The WINS server is responsible for knowing computer names and addresses and for ensuring no duplicate names exist on the network.”); *id.* at 69 (“The WINS protocol is based on and is compatible with the protocols defined for NBNS in RFCs 1001/1002”); *id.* at 265 (“[A] WINS server can provide name registration services” and is “implemented under RFC 1001/1002 to provide name resolution services for NetBIOS computer names.”).³

53. WINS teaches that this mapping of names to IP addresses “provides a powerful, new name resolution service for easy, centralized management of

³ WINS teaches that Windows NT also can be used with a Domain Name System (DNS) server, which (unlike WINS) uses “static configuration for computer name-to-IP address mapping.” (Ex. 1003 at 78, 80.)

computer name-to-IP address resolution in medium and large internetworks.” (Ex. 1003 at 13; *id.* at 69 (“WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment.”); *id.* at 266, 268 (“WINS” is “[a] name resolution service that resolves Windows networking computer names to IP addresses in a routed environment” and “p-node” is “[a] NetBIOS over TCP/IP mode that uses point-to-point communications with a name server to resolve computer names as addresses”).)

54. WINS also explains that “WINS name resolution is enabled and configured automatically for a computer that is configured with DHCP.” (Ex. 1003 at 40.) After the DHCP server dynamically assigns an IP address, the processing unit transmits its assigned IP address and device name to the WINS server as part of the system startup process. (*Id.* at 65 (“A mechanism must be available on a TCP/IP network to resolve names to IP addresses. To ensure that both name and address are unique, the Windows NT computer using TCP/IP registers its name and IP address on the network during system startup.”); *id.* at 66 (“When WINS servers are in place on the network, NetBIOS over TCP/IP resolves names on a client computer by communicating with the WINS server.”); *id.* at 67 (“In a p-node environment ... [a]ll computers register themselves with the WINS server, which is a NetBIOS Name Server (NBNS) with enhancements.”); *id.* at 69 (“WINS provides a distributed

database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment.”); *id.* at 72 (“During TCP/IP configuration, the computer’s name is registered with the WINS server, and the IP address of the WINS server”); *id.* at 74 (“The name registration request is sent directly to the WINS server to be added to the database. [If the name is not already in use] WINS accepts the entry and adds it to its local database . . . together with a timestamp, an incremental unique version number, and other information.”).)

55. WINS also teaches that the WINS/NetBIOS server keeps its mapping of names to IP addresses current to track which users are online. For example, it states that “when dynamic addressing through DHCP results in new IP addresses for computers that move between subnets, the changes are automatically updated in the WINS database.” (Ex. 1003 at 69.) It notes that “[w]henever a computer is shut down properly, it releases its name to the WINS server, which marks the related database entry as *released* ... because it knows that the old client is no longer using that name.” (*Id.* at 75.) And “[to] provide[] registration reliability through periodic reregistering of names with the WINS servers,” if a name is not re-registered within the set renewal time, “the WINS server will mark the name as released and available for use.” (*Id.*)

3. Steps 3 & 4: First Processing Unit Sends Query to WINS Server and Receives the IP Address of the Second Processing Unit

56. WINS explains that “WINS consists of two components: the WINS server, which handles name queries and registrations, and the client software, which queries for computer name resolution.” (Ex. 1003 at 69; *id.* at 70 (“WINS-enabled computers, including proxies, access the WINS server directly....”).) It further teaches a first processing unit (e.g., a first computer running the Windows NT or Windows for Workgroups operating system) obtains the IP address of a second processing unit (e.g., a second computer running the Windows NT or Windows for Workgroups operating system) by transmitting a “name query request” to the WINS/NetBIOS server, which returns the IP “address mapping” of the second processing unit to the first processing unit. (Ex. 1003 at 72-73 (“[N]ame-to-IP address mapping registered with a WINS server can be provided reliably as a response to a name query.”); *id.* at 67 (“[W]hen NT_PC1 wants to communicate with NT_PC2, it queries the WINS server for the address of NT_PC2 ... [and] gets the appropriate address from the WINS server.”); *id.* at 122 (WINS servers “support computer name registration and name resolution,” provide “dynamic name services” in “a NetBIOS namespace,” offer “[c]entralized management of the computer name database,” and “allow[] client computers to easily locate remote systems across local or wide area networks.”).)

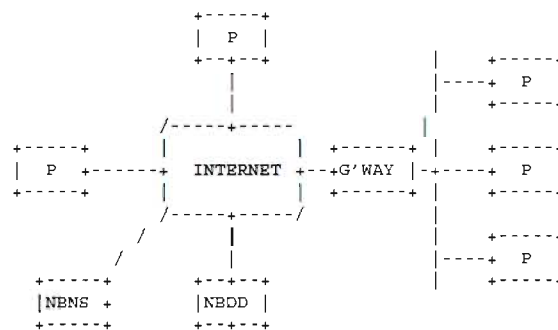
4. **Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communication with Second Processing Unit**

57. The first processing unit can then establish point-to-point communications with the second processing unit by using the IP address received from the WINS/NetBIOS server. (Ex. 1003 at 67 (“[W]hen NT_PC1 wants to communicate with NT_PC2, it queries the WINS server for the address of NT_PC2. When NT_PC1 gets the appropriate address from the WINS server, it goes directly to NT_PC2 without broadcasting.”); *id.* at 266 (“p-node” is “[a] NetBIOS over TCP/IP mode that uses point-to-point communications with a name server to resolve computer names as addresses.”); *id.* at 73 (“If the name is found in the WINS database, the client can establish a session based on the address mapping received from WINS.”); *id.* at 121 (“WINS servers maintain a database that maps computer names to IP addresses, allowing users to easily communicate with other computers while gaining all the benefits of TCP/IP.”).)

B. NetBIOS (Ex. 1004)

58. RFC 1001 “describes the ideas and general methods used to provide NetBIOS on a TCP and UDP foundation,” and RFC 1002 “contains the detailed packet formats and protocol specifications for NetBIOS-over-TCP.” (Ex. 1004 at 368, 374, 442.) In 1992, RFCs 1001 and 1002 were published as Appendices F and G to the book attached as Exhibit 1004 (“NetBIOS”).

59. NetBIOS teaches point-to-point communications between nodes over a network, including between “[p]oint-to-point (or ‘P’) nodes” that use a directory look-up service called a “NetBIOS Name Server” (NBNS). (Ex. 1004 at 384-85.) For example, the figure below shows “P”-nodes connected to the “Internet” (2 directly, and 3 through a gateway (“G’WAY”)), each of which can communicate with a NBNS that also is connected to the Internet:



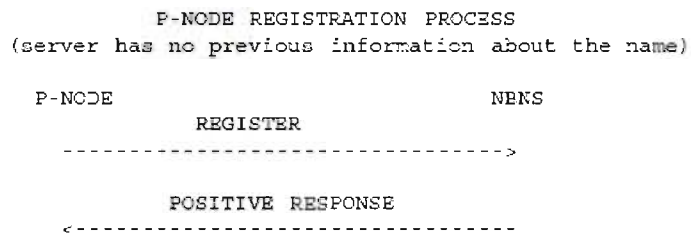
(Ex. 1004 at 389.)

1. Step 1: Processing Units Have Assigned IP Addresses

60. NetBIOS explains that “[e]very node has a permanent unique name,” and “[a]n IP address may be associated with only one instance of [a p-node].” (Ex. 1004 at 383, 394; *id.* at 416 (“The IP addresses are obtained from the TCP service interface.”); *id.* at 383 (p-node is a “[p]oint-to-point” node).) A person of ordinary skill in the art would understand from this disclosure that a processing unit (e.g., a computer running software defined by NetBIOS) or process (e.g., software implementing the interface defined by NetBIOS running on a computer) must obtain an IP address to participate on the network.

2. Step 2: Processing Units Register Their IP Addresses and Identifiers with the NBNS

61. NetBIOS teaches that a p-node sends its name (“RR_NAME”) and assigned IP address (“NB_ADDRESS”) to the NBNS, and if the name is unique and accepted, the NBNS confirms registration with a “Positive Response” message:



(Ex. 1004 at 402-03 (“[T]he end-node sends a NAME REGISTRATION REQUEST, [and] the NBNS responds with a POSITIVE NAME REGISTRATION RESPONSE”); *id.* at 430-32 (NAME REGISTRATION REQUEST includes p-node’s name and IP address).)

62. The NBNS also tracks which nodes remain online. For example, a node is deemed offline after a “[s]ilent release,” which “typically occurs when an end-node fails or is turned off.” (Ex. 1004 at 395; *id.* at 378 (“Implicit name deletion occurs when a station ceases operation. In the case of personal computers, implicit name deletion is a frequent occurrence.”).) A node is treated as offline if it fails to send a “name refresh message” during a “lifetime” period the NBNS establishes during name registration (which can be short or long, depending on the desired amount of network traffic). (*Id.* at 396 (“Names held by an NBNS are given a

lifetime during name registration. The NBNS will consider a name to have been silently released if the end-node fails to send a name refresh message to the NBNS before the lifetime expires. A refresh restarts the lifetime clock.”); *id.* at 400 (“NBNS may impose a ‘time-to-live’ on each name it registers” and a node “must periodically send a status report to the NBNS ... [to] restart the timers of both the NBNS and ... node The NBNS may consider that a node has released any name which has not been refreshed within some multiple of name’s time-to-live.”); *id.* at 401 (“NBNS is free to poll nodes ... to verify that their name status is the same as that registered in the NBNS.”); *id.* at 412-13 (p-nodes are “responsible for sending periodic NAME REFRESH REQUESTs ... [which] contain[] a single name that has been successfully registered by that node. The interval between such packets is negotiated between the end node and the NBNS server at the time that the name is initially claimed.”).) “[E]xplicit release” is when “P nodes send a notification to their NBNS.” (*Id.* at 395; *id.* at 378 (“An explicit name deletion function is specified, so that applications may remove a name.”).)

3. Steps 3 & 4: First Processing Unit Sends Query to the NBNS and Receives the IP Address of the Second Processing Unit

63. To establish point-to-point communications, a first p-node can learn the IP address of a second p-node by sending a “name query request” to the NBNS that includes the name (“QUESTION_NAME”) of the second p-node. (Ex. 1004 at 447, 458 (query includes name of second p-node); *id.* at 395 (“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 406 (“Name query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name.”).) If the second p-node is registered, the NBNS sends a POSITIVE RESPONSE to the first p-node that includes the IP address of the second p-node. (*Id.* at 459; *id.* at 407 (“NBNS answers queries from a P node with a list of IP address and other information [of the queried node]”); *id.* at 422 (“datagram service must perform a name query operation to learn the IP address and the attributes of the destination NetBIOS name”).)

4. Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communications with Second Processing Unit

64. After obtaining the second p-node’s IP address from the NBNS, the first p-node can establish “directed (point-to-point) communications” with the second p-node. (Ex. 1004 at 415) (IP addresses “found for the target name ... may have been acquired using the NetBIOS name query transactions NetBIOS session service transactions, packets, and protocols ... involve only directed (point-to-point) communications.”); *id.* at 383 (defining p-node as “[p]oint-to-point” node); *id.* at 416 (“An end-node begins establishment of a session to another node by somehow acquiring (perhaps using the name query transactions ...) the IP address of the node ...” and each data packet includes the first and second p-node “IP address”

and “NetBIOS name”); *id.* at 419 (point-to-point communications described if “a name query operation was successfully completed by the caller node for the listener’s name.”).)

VI. SPECIFIC GROUNDS FOR PETITION

A. Claims 3, 4, and 6-14 of the ’121 Patent Would Have Been Obvious Over WINS and NetBIOS

65. I have been informed and understand that a patent claim is invalid for being obvious if the differences between the patented subject matter and the prior art are such that the patented subject matter as a whole would have been obvious at the time the invention was made to a person of ordinary skill in the art. I am informed that this standard is set forth in 35 U.S.C. § 103(a), which has been reproduced in pertinent part below:

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 the subject matter pertains.

66. When considering the issue of obviousness, I understand that I am to do the following: (i) determine the scope and content of the prior art; (ii) ascertain the differences between the prior art and the claims at issue; (iii) resolve the level of

ordinary skill in the art; and (iv) consider objective evidence of non-obviousness (also known as “secondary considerations” of non-obviousness). Examples of evidence of secondary considerations of non-obviousness include evidence of commercial success, long-felt but unsolved needs, failure of others, and unexpected results. I am not aware of any secondary considerations that would rebut my opinion that claims 3, 4, and 6-14 of the ’121 patent would have been obvious (and I reserve the right to address any supposed secondary considerations to the extent Straight Path attempts to raise them in this proceeding). To the contrary, as explained below, the claimed invention of the ’121 patent did not solve a long-felt but unsolved need, fix the failure of others, or achieve unexpected results given that multiple prior art references had already addressed and solved the same issues in the same way. Nor am I aware of any evidence showing a nexus between the commercial success of any product and the claimed invention of the ’121 patent.

67. My understanding is that not all innovations are patentable. Even if a claimed product or method is not disclosed in its entirety in a single prior art reference, the patent claim is invalid if the invention would have been obvious to a person of ordinary skill in the art at the time of the invention. In particular, I understand that a patent claim is normally invalid as obvious if it would have been a matter of “ordinary innovation” within the relevant field to create the claimed product or method at the time of the invention.

68. I also understand that the following exemplary scenarios would support a conclusion that a claimed product or method would have been obvious:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of known technique to improve similar devices (methods, or products) in the same way;
- Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- “Obvious to try” – choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;
- Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

69. The following description demonstrates in detail that claims 3, 4, and 6-14 would have been obvious under 35 U.S.C. § 103(a) over the combined teachings of WINS and NetBIOS.⁴ The discussion below contains non-limiting examples of how the prior art teaches each claimed feature and additional examples are set forth elsewhere in the prior art.

1. Level of Ordinary Skill in the Art

70. I am informed that prior art references should be understood from the perspective of a person of ordinary skill in the art to which the patent is related at the time of the patent's priority date. I understand that a person of ordinary skill in the art is presumed to be aware of all pertinent prior art and the conventional wisdom in the art, and is a person of ordinary creativity. I have applied this standard throughout my declaration.

71. In my opinion, a person of ordinary skill in the art related to the '121 patent would generally have at least a Bachelor's of Science degree in Computer Science, or equivalent, and several years of relevant programming experience in networking and/or Voice over Internet Protocol. (Absent a formal degree, a person of ordinary skill in the art would have additional years of work experience.) Such a person would understand the prior art references described herein.

⁴ In forming my opinions I have applied the constructions set forth in Section VII of the '121 Patent Petition.

72. As described in more detail above, I was a person with at least ordinary skill in the art for the '121 patent at the time of its alleged invention.

2. One Skilled in the Art Would Have Been Motivated to Combine WINS and NetBIOS

73. WINS clearly discloses that NetBIOS (as defined in RFCs 1001 and 1002) is incorporated into, and is an integral part of, the disclosed WINS system. (E.g., Ex. 1003 at 65 (defining WINS as a “NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as p-node”); *id.* at 66 (section entitled “NetBIOS over TCP/IP and Name Resolution” describing “the modes of NetBIOS over TCP/IP, as defined in RFCs 1001 and 1002 to specify how NetBIOS should be implemented over TCP/IP”); *id.* at 69 (“WINS protocol is based on and is compatible with the protocols defined for NBNS in RFCs 1001/1002, so it is interoperable with any other implementations of these RFCs.”).) Indeed, WINS refers to NetBIOS over one hundred times (*id.* at 5, 20, 22, 23, 28, 35, 40, 41, 44, 45, 53, 54, 64, 65-67, 69, 77, 79, 80, 87, 102, 103, 122, 124, 126, 139, 142, 144, 147, 151, 164, 165, 171, 208, 209, 214, 224, 225, 262, 265, 270, 271, 273-77) and refers to NetBIOS protocols RFCs 1001 and 1002 ten times (*id.* at 22, 65, 66, 69, 102, 103, 167, 262, 265).) Because WINS (Ex. 1003) expressly teaches that it is combined with NetBIOS RFCs 1001 and 1002 (Ex. 1004), one skilled in the art would have known about and been motivated to combine the references.

3. **Claim 3 (Independent)**

- a. **“A computer program product for use with a computer system capable of executing a first process and communicating with other processes, a directory server process and a mail server process over a computer network, the computer program product comprising a computer usable medium having computer program code embodied in the medium, the program code comprising:”**

74. To the extent the preamble of claim 3 is limiting, WINS discloses a “computer program product ... comprising a computer usable medium having computer code embodied in the medium” that is “for use with a computer system.” (Ex. 1003 at 121 (“A WINS server is a Windows NT Server computer running Microsoft TCP/IP and the Windows Internet Name Service (WINS) server software.”); *id.* at 23 (referring to the “Windows NT Server 3.5 compact disc”); *id.* at 155 (“[T]he Windows NT Server installation source ... can be the Windows NT Server compact disc, the installation floppy disks, or a network directory that contains the master files for Windows NT Server.”).)

75. WINS further explains that such a computer system can execute a first process (e.g., the Windows NT or Windows for Workgroups operating system running on a first computer) and communicate with a second process (e.g., the Windows NT or Windows for Workgroups operating system running on a second computer), a directory server process (e.g., a WINS/NetBIOS server), and a mail server process (e.g., by using an “electronic mail[] client”) over a computer network

(e.g., a LAN, WAN, or the Internet). (Ex. 1003 at 29 (software includes “graphic connectivity utilities, terminal emulation software, Simple Mail Transfer Protocol (SMTP) and electronic mail clients, network printing utilities, SQL client applications, and corporate client-server applications”); *id.* at 76 (Windows NT computers can be used to “dial in to remotely access ... networks for services such as ... electronic mail”); *id.* at 118-19 (explaining use for “a small LAN” and “large enterprise network”); *see* Section V(A) (Overview).) It further teaches that “NetBIOS on personal computers includes both a set of services and an exact program interface to those services,” and that “NetBIOS defines a software interface ... designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available.” (Ex. 1004 at 374, 377; *see* Section V(B) (Overview of NetBIOS).)

b. **“A. program code configured to determine the currently assigned network protocol address of the first process upon connection to the computer network;”**

76. WINS discloses that each process on a Windows NT network “must be assigned a unique IP address” (Ex. 1003 at 57), and that software running on an Internet access server (e.g., a Dynamic Host Configuration Protocol (DHCP) server) dynamically assigns IP addresses to processes. (*Id.* at 83 (“A [DHCP] server is a Windows NT Server computer running Microsoft TCP/IP and the DHCP-compatible server software” as “defined in [RFCs] 1533, 1534, 1541, and

1542”); *id.* at 113 (“Allocation of IP addresses for distribution by DHCP servers can be done dynamically [which] allows a client to be assigned an IP address from the free address pool.”); *id.* at 13, 22, 23, 62, 63, 85-120 (DHCP software and configurations).)

77. “During system startup,” a computer running the Windows NT or Windows for Workgroups operating system wishing to participate on the network sends a “*discover message*” to multiple DHCP servers, each of which “responds with an *offer message* containing an IP address and valid configuration information for the client.” (Ex. 1003 at 63.) The computer selects the IP address offered by one of the DHCP servers, and “sends a *request message* that identifies the DHCP server for the selected configuration.” (*Id.*) “The selected DHCP server sends a *DHCP acknowledgment message* that contains the [IP] address [for the computer].... After the [computer] receives the acknowledgement, it enters a *bound* state and can now participate on the TCP/IP network and complete its system startup.” (*Id.* at 64; Section V(A)(1).) Similarly, NetBIOS discloses that p-nodes have “IP addresses ... obtained from the TCP service interface.” (Ex. 1004 at 416; Section V(B)(1).)

78. Thus, prior art teaches program code (e.g., the code underlying the Windows NT or Windows for Workgroups operating system) configured to determine the currently assigned network protocol address of the first process (e.g.,

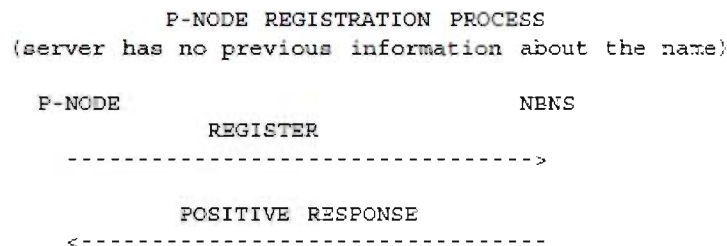
the IP address assigned by the DHCP server) upon connection to the network (e.g., at “system startup”).

- c. **“B. program code configured to establish a communication connection with the directory server process once the assigned network protocol [sic] of the first process is known;”**

79. WINS explains that, after the DHCP server dynamically assigns an IP address, the Windows NT or Windows for Workgroups operating system transmits its assigned IP address and name to the WINS server during startup. (Ex. 1003 at 65 (“A mechanism must be available on a TCP/IP network to resolve names to IP addresses. To ensure that both name and address are unique, the Windows NT computer using TCP/IP registers its name and IP address on the network during system startup.”); *id.* at 66 (“NetBIOS over TCP/IP resolves names on a client computer by communicating with the WINS server.”); *id.* at 67 (“In a p-node environment ... [a]ll computers register themselves with the WINS server, which is a NetBIOS Name Server (NBNS) with enhancements.”); *id.* at 69 (“WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment.”); *id.* at 72 (“During TCP/IP configuration, the computer’s name is registered with the WINS server, and the IP address of the WINS server”); *id.* at 74 (“Name registration ensures that the computer’s name and IP address are unique for each device.”); *id.* (“The name registration request is sent directly to the WINS server to be added to the

database.”); *id.* at 151 (database entry includes “NetBIOS computer name” and “assigned Internet Protocol address”); Section V(A)(2).)

80. NetBIOS specifies that the “Name Registration Request” includes “the IP address of the name’s owner [e.g., the p-node registering with the NBNS server].” (Ex. 1004 at 449-50.) “The diagram, below, shows the sequence of events when an end-node [i.e., a p-node] registers a name which is new to the NBNS”:



(*Id.* at 402-03.) NetBIOS also specifies the format for a node name registration (illustrated below), which requires a node to forward to the NBNS its name (“RR_NAME”) and assigned IP address (“NB_ADDRESS”):

4.2.2. NAME REGISTRATION REQUEST

```

      1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      NAME_TRN_ID      | 0 | 0x5 | 0 | 0 | 1 | 0 | 0 | 0 | B | 0x0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      0x0001      |      0x0000      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      0x0000      |      0x0001      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
|      QUESTION_NAME
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|      NB {0x0020}      |      IN {0x0001}      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
|      RR_NAME
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|      NB {0x0020}      |      IN {0x0001}      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
|      TTL
+-----+-----+-----+-----+-----+-----+-----+-----+
|      0x0006      |      NB_FLAGS
+-----+-----+-----+-----+-----+-----+-----+-----+
|
|      NB_ADDRESS
+-----+-----+-----+-----+-----+-----+-----+-----+

```

(*Id.* at 448-50; *see* Section V(B)(2).)

81. Therefore, the prior art discloses program code configured to establish a communication connection with the directory server process (e.g., software running on a first computer sends a name registration request to the WINS/NetBIOS server) once the assigned network protocol address of the first process is known (e.g., after the Windows NT or Windows for Workgroups operating system running on the first computer receives its IP address from the DHCP server during system startup).

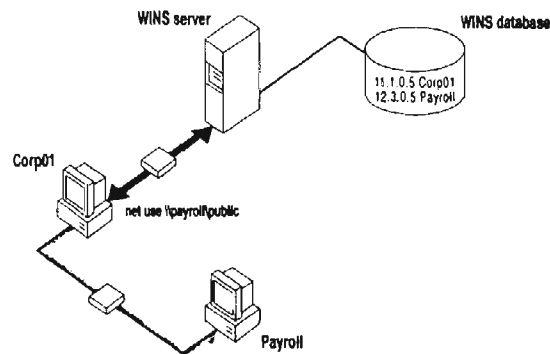
- d. **“C. program code configured to forward the assigned network protocol address of the first process to the directory server process upon establishing a communication connection with the directory server process; and”**

82. As set forth in Section VI(A)(3)(c) (claim 3[B]), the prior art teaches program code (e.g., the code underlying the Windows NT or Windows for Workgroups operating system) configured to forward the assigned network protocol address of the first process (e.g., the Windows NT or Windows for Workgroups operating system running on a first computer forwards a name registration request containing its IP address) to the directory server process (e.g., WINS/NetBIOS server) upon establishing a communication connection with the directory server process (e.g., during startup).

- e. **“D. program code configured to establish a point-to-point communication with another process over the computer network”**

83. WINS teaches how Windows NT software establishes a point-to-point communication with another process over the computer network: “[i]n a p-node environment,” “when NT_ PC1 [e.g., a computer running the Windows NT or Windows for Workgroups operating system] wants to communicate with NT_PC2 [e.g., another computer running the Windows NT or Windows for Workgroups operating system], it queries the WINS server for the address of NT_PC2 ... [and] gets the appropriate address from the WINS server” Ex. 1003 at 67 (emphasis

added); *id.* at 266 (defining “p-node” as “[a] NetBIOS over TCP/IP mode that uses point-to-point communications with a name server to resolve computer names as addresses”).) For example, WINS shows how “Corp01” obtains the IP address of “Payroll” using the WINS/NetBIOS server/database, and communicates with Payroll directly without using the WINS/NetBIOS Server:



(Ex. 1003 at 73 (“If the name is found in the WINS database, the client can establish a session based on the address mapping received from WINS.”); *see* Sections V(A)(3) and (A)(4).)

84. NetBIOS discloses that p-nodes (e.g., implemented by WINS) are “[p]oint-to-point (‘P’) nodes” (Ex. 1004 at 383), and that “[t]he NetBIOS session service,” which “begins after one or more IP addresses have been found for the target name,” involves “only directed (point-to-point) communications,” beginning with “a TCP connection [being] established with the remote party [e.g., with the second process].” (Ex. 1004 at 415-16; Sections V(B)(3) and (4).)

85. Thus, prior art teaches program code configured to establish a point-to-point communication with another process (e.g., the Windows NT or Windows for Workgroups operating system running on a first computer can establish point-to-point communication with the Windows NT or Windows for Workgroups operating system running on a second computer using the IP address received from the WINS/NetBIOS server). Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 3 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

4. **Claim 4 (Depends From Claim 3)**

- a. **“The computer program product of claim 3 wherein the program code configured to establish a point-to-point communication further comprises:”**

86. To the extent the preamble of claim 4 is limiting, WINS and NetBIOS teach it for the reasons they teach the limitations of claim 3, as discussed above in Section VI(A)(3).

- b. **“D.1 program code configured to transmit, from the first process to the directory server process, a query as to whether a second process is connected to the computer network; and”**

87. WINS explains that, “[a]lthough TCP/IP uses IP addresses to identify and reach computers, users typically prefer to use computer names.” (Ex. 1003 at 41; *id.* at 65 (“Computers use IP addresses to identify each other, but users usually

find it easier to work with computer names.”.) The Windows NT software assigns a “NetBIOS” name to each process during “setup,” which the process transmits to the WINS/NetBIOS server as a “name registration request” in accordance with “a NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as p-node.” (*Id.* at 65, 74, 121; *id.* at 67 (“All computers register themselves with the WINS server, which is a NetBIOS Name Server (NBNS) with enhancements.”); *id.* at 20 (“Microsoft TCP/IP includes ... NetBIOS for establishing logical names and sessions” and “[WINS] for dynamically registering and querying computer names on an internetwork”).)

88. WINS further teaches that (1) a WINS server “contain[s] a dynamic database mapping computer names to IP addresses” (Ex. 1003 at 65); (2) software running on a first process requests the WINS/NetBIOS server to retrieve the IP address of a second process by sending “[a] *name query request* . . . to the WINS server” (*id.* at 73); and (3) the WINS/NetBIOS server responds by providing the first process with the IP address of the second process if the name and IP address of the second process are registered in the database. (*Id.* at 73 (“Any name-to-IP address mapping registered with a WINS server can be provided reliably as a response to a name query.”); *id.* at 69 (“WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment.”); *id.* at 72 (“During TCP/IP configuration, the computer’s name is

registered with the WINS server, and the IP address of the WINS server....”); Section V(A)(2).

89. Thus, as explained in Section VI(A)(3)(e) (claim 3[D]), “when NT_PC1 wants to communicate with NT_PC2, it queries the WINS server for the address of NT_PC2 ... [and] gets the appropriate address [of NT_PC2] from the WINS server.” (Ex. 1003 at 67; Section V(A)(3).)

90. WINS also teaches that the WINS server keeps its mapping of names to IP addresses “relatively current” by tracking which users are still connected to the network. (Ex. 1003 at 69 (“[W]hen dynamic addressing through DHCP results in new IP addresses for computers that move between subnets, the changes are automatically updated in the WINS database.”); *id.* at 75 (“[To] provide[] registration reliability through periodic reregistering of names with the WINS servers,” if a process fails to re-register its name within the set renewal time, “the WINS server will mark the name as released and available for use.”); *see* Sections V(A)(1)-(3).)

91. Similarly, in NetBIOS “[n]ame query transactions are initiated by end-nodes to obtain the IP address(es) and other attributes associated with a NetBIOS name [of a second node]” (Ex. 1004 at 406), and that a “NAME QUERY REQUEST” includes the name (“QUESTION_NAME”) of the second node. (Ex. 1004 at 458.) “Names held by an NBNS are given a lifetime during name

registration,” and “[t]he NBNS will consider a name to have been silently released if the end-node fails to send a name refresh message to the NBNS before the lifetime expires.” (Ex. 1004 at 396 (“To detect whether a node has silently released its claim to a name, it is necessary on occasion to challenge that node’s current ownership,” and if that challenge fails “it is assumed that the node has released the name.”); *id.* at 400-01; *see* Sections V(B)(1)-(3).)

92. NetBIOS further specifies that if the queried name (contained in a “NAME QUERY REQUEST”) is registered with the server directory, the NBNS sends a “POSITIVE NAME QUERY RESPONSE” to the first node. (Ex. 1004 at 395, 407-08, 449, 459.) The “POSITIVE NAME QUERY RESPONSE” includes an “ADDR_ENTRY record ‘represent[ing] an owner of a name’ that includes an “NB ADDRESS” corresponding to the IP address of the name of the second computer. (*Id.*) If the queried name is not in the directory (i.e., the second process is not connected to the network), NetBIOS teaches that the NBNS returns a “NEGATIVE NAME QUERY RESPONSE” to the first process, which contains a “Name Error” signifying that “[t]he name requested does not exist” (i.e., the device is not currently connected to the network). (Ex. 1004 at 395, 407, 413, 460; *see* Sections V(B)(1) and (B)(2).)

93. The prior art thus teaches program code configured to transmit, from the first process (e.g., the Windows NT or Windows for Workgroups operating

system running on a first computer) to the directory server process (e.g., a WINS/NetBIOS server), a query (e.g., a WINS or NetBIOS Name Query Request) as to whether a second process (e.g., the Windows NT or Windows for Workgroups operating system running on a second computer) is connected to the computer network (e.g., if the name and IP address of the second computer is registered in the WINS/NetBIOS server database, the server returns a POSITIVE NAME QUERY RESPONSE with the current IP address of the second process, and if the second process is not registered, the server returns a NEGATIVE NAME QUERY RESPONSE).

c. **“D.2 program code configured to receive a network protocol address of the second process from the directory server process, when the second process is connected to the computer network”**

94. As set forth in Section VI(A)(4)(b) (claim 4[D.1]), WINS and NetBIOS teach program code configured to receive a network protocol address (e.g., IP address) of the second process (e.g., the Windows NT or Windows for Workgroups operating system running on a second computer) from the directory server process (e.g., WINS/NetBIOS server), when the second process is connected to the network (e.g., if the name and IP address of the second process is registered in the WINS/NetBIOS server database, the server informs the first process that the second process is connected to the network by providing a “POSITIVE NAME QUERY RESPONSE” with the current IP address of the second process).

95. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 4 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

5. **Claim 6 (Independent)**

- a. **“A computer program product for use with a computer system capable of executing a first process and connecting to other processes and a server process over a computer network, the computer program product comprising a computer usable medium having computer readable code means embodied in the medium comprising:”**

96. To the extent the preamble of claim 6 is limiting, WINS and NetBIOS disclose it for the same reasons set forth in Section VI(A)(3)(a) (claim 3[preamble]).

- b. **“A. program code configured to, following connection of the first process to the computer network, forward to the server process a dynamically assigned network protocol address at which the first process is connected to the computer network;”**

97. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) (claim 3[A-C]). WINS teaches the added requirement of claim 6 for a “dynamically assigned network protocol address” by explaining the DHCP server dynamically assigns IP addresses to processes when they connect to the network, and processes forward their dynamically assigned addresses to the WINS/NetBIOS server during startup. (*E.g.*, Ex. 1003 at 13 (“When DHCP servers are installed on the network, users can take advantage of dynamic IP address allocation and management”); *id.* at 62 (DHCP permits “centralized management of address

allocation” through “dynamic configuration of IP addresses for computers”); *id.* at 63 (process obtains network address from DHCP server “[d]uring system startup”); *id.* at 69 (“WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment.”); *id.* at 113 (“Allocation of IP addresses for distribution by DHCP servers can be done dynamically” which “allows a client to be assigned an IP address from the free address pool.”); *see* Sections V(A)(1) and (A)(2)). As discussed in Sections V(B)(1) and (B)(2), NetBIOS also teaches this feature.

98. WINS and NetBIOS thus teach program code configured to, following connection of the first process to the computer network (e.g., after a first computer running the Windows NT or Windows for Workgroups operating system registers with the DHCP server), forward to the server process (e.g., the WINS/NetBIOS server) a dynamically assigned network protocol address at which the first process is connected to the network (e.g., the IP address that the DHCP server assigned “during system startup”).

- c. **“B. program code configured to query the address server as to whether the second process is connected to the computer network;”**

99. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(b) (claim 4[D.1]).

- d. **“C. program code configured to receive a dynamically assigned network protocol address of the second process from the address server, when the second process is connected to the computer network; and”**

100. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(4)(c) and (5)(b) (claim 4[D.2]; claim 6[A] for “dynamically assigned” aspects of this limitation).

- e. **“D. program code configured to respond to the network protocol address of the second process, establish a point-to-point communication link with the second process over the computer network”**

101. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(3)(e) (claim 3[D]).

102. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 6 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

6. **Claim 7 (Independent)**

- a. **“A computer data signal embodied in a carrier wave comprising:”**

103. To the extent the preamble of claim 7 is limiting, WINS and NetBIOS teach it because one of ordinary skill would understand that a computer data signal embodied in a carrier wave would be created, among other times, during installation of Windows NT or WINS client software. (Ex. 1003 at 155 (“[T]he Windows NT Server installation source ... can be the Windows NT Server compact disc, the

installation floppy disks, or a network directory that contains the master files for Windows NT Server.”.)

- b. **“A. program code configured to, following connection of a first process to a computer network, forward to a server process a dynamically assigned network protocol address at which the first process is connected to the computer network;”**

104. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) (claim 3[A-C]) and VI(A)(5)(b) (claim 6[A]).

- c. **“B. program code configured to query the server process as to whether a second process is connected to the computer network;”**

105. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(4)(b) and (A)(5)(c) (claims 4[D.1] and 6[B]).

- d. **“C. program code configured to receive a dynamically assigned network protocol address of the second process from the server process, when the second process is connected to the computer network; and”**

106. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(4)(c) and (A)(5)(d) (claims 4[D.2] and 6[C]).

- e. **“D. program code, responsive to the network protocol address of the second process, and configured to establish a point-to-point communication connection with the second process over the computer network”**

107. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(3)(e) and (A)(5)(e) (claims 3[D] and 6[D]).

108. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 7 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

7. **Claim 8 (Independent)**

- a. **“An apparatus 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 directory database server process, the apparatus comprising:”**

109. To the extent the preamble of claim 8 is limiting, WINS and NetBIOS teach it for the reasons in Section VI(A)(3)(a) (claim 3[preamble]). In addition, WINS and NetBIOS teach that they are intended for use on an apparatus for use with a computer system. (Ex. 1003 at 12 (“[This manual] describes how to install, configure, and troubleshoot Microsoft TCP/IP on a computer running the Microsoft Windows NT Workstation or Windows NT Server operating system.”); *id.* at 18 (system “can be used to communicate with Windows NT systems, with devices that use other Microsoft networking products, and with non-Microsoft systems”); *id.* at 73 (computers “Corp01” and “Payroll” and WINS server); Ex. 1004 at 374 (NetBIOS intended “for the IBM Personal Computer (PC) and compatible systems” “file servers,” and “virtually any type of system where the TCP/IP protocol suite is available”); *id.* at 375 (NetBIOS used “on TCP networks,” “personal computers,” and “larger hosts”).)

- b. **“A. program logic configured to, following connection of the first process to the computer network forward to the address server a dynamically assigned network, protocol address at which the first process is connected to the computer network;”**

110. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) (claim 3[A-C]) and Section VI(A)(5)(b) (claim 6[A]).

- c. **“B. program logic configured to query the address server as to whether the second process is connected to the computer network;”**

111. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(4)(b) and (A)(5)(c) (claims 4[D.1] and 6[B]).

- d. **“C. program logic configured to receive a dynamically assigned network protocol address of the second process from the address server, when the second process is connected to the computer network; and”**

112. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(4)(c) and (A)(5)(d) (claims 4[D.2] and 6[C]).

- e. **“D. program logic configured to, in response to the network protocol address of the second process, establish a point-to-point communication link with the second process over the computer network”**

113. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(3)(e) (claim 3[D]), Section VI(A)(4) (claim 4), and Section VI(A)(5)(e) (claim 6[D]).

114. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 8 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

8. **Claim 9 (Independent)**

a. **“A computer data signal embodied in a carrier wave comprising:”**

115. To the extent the preamble of claim 9 is limiting, it is disclosed for the reasons in Section VI(A)(6)(a) (claim 7[preamble]).

b. **“a. program code configured to access a directory database, the database having a dynamically assigned network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network; and”**

116. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) (claim 3[A-C]), Sections VI(A)(4)(b)-(c) (claim 4[D.1-D.2]), and Section VI(A)(5)(b) (claim 6[A]). They teach program code configured to access a directory database (e.g., WINS client software accesses the WINS/NetBIOS database), the database having a dynamically assigned network protocol address (e.g., the WINS/NetBIOS server stores IP addresses dynamically assigned by the DHCP) for a selected plurality of processes (e.g., instances of the Windows NT or Windows for Workgroups operating system running on computers) having on-line status with respect to the computer network (e.g., the WINS/NetBIOS server

database tracks which processes are online, and informs a first process when a second process has online status with respect to the network by providing the current IP address of the second process and/or a POSITIVE NAME QUERY RESPONSE), the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network (e.g., “during system startup,” after the DHCP server dynamically assigns an IP address, the WINS client software forwards the IP address and name to the WINS/NetBIOS server).

- c. **“b. program code responsive to one of the dynamically assigned 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”**

117. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(3)(e) (claim 3[D]), Section VI(A)(4) (claim 4), and Section VI(A)(5)(e) (claim 6[D]). They teach program code responsive to one of the dynamically assigned network protocol addresses (e.g., program code responsive to the IP address that the DHCP server dynamically assigned) and configured to establish a point-to-point communication link over the network from the first process (e.g., the Windows NT or Windows for Workgroups operating system running on a first computer) to the second process (e.g., the Windows NT or Windows for

Workgroups operating system running on the second computer) over the computer network (e.g., using TCP/IP).

118. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 9 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

9. **Claim 10 (Independent)**

- a. **“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:”**

119. To the extent the preamble of claim 10 is limiting, WINS and NetBIOS teach it for the reasons in Section VI(A)(3)(e) (claim 3[D]).

- b. **“a. accessing a directory database, the database having a dynamically assigned network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network; and”**

120. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) (claim 3[A-C]), Section VI(A)(4) (claim 4), and Section VI(A)(5)(b) (claim 6[A]).

- c. **“b. in response to one of the dynamically assigned network protocol addresses, establish a point-to-point communication link from the first process to the second process over the computer network”**

121. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(e) (claim 3[D]), VI(A)(4)(c) (claim 4[D.2]), and VI(A)(5)(e) (claim 6[D]).

122. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 10 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

10. **Claim 11 (Independent)**

- a. **“An apparatus for use with a computer system, the computer system capable of executing a first process connectable over a computer network to a second process and a directory database server process, the apparatus comprising:”**

123. To the extent the preamble of claim 11 is limiting, WINS and NetBIOS teach it for the reasons in Section VI(A)(7)(a) (claim 8[preamble]).

- b. **“a. program logic configured to access a directory database, the database having a dynamically assigned network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network; and”**

124. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) (claim 3[A-C]), VI(A)(4) (claim 4), VI(A)(5)(b) (claim 6[A]), and VI(A)(8)(b) (claim 9[a]).

- c. **“b. program logic responsive to one of the dynamically assigned 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”**

125. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(e) (claim 3[D]), VI(A)(4)(c) (claim 4[D.2]), and VI(A)(5)(e) (claim 6[D]).

126. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 11 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

11. **Claim 12 (Independent)**

- a. **“A computer data signal embodied in a carrier wave comprising:”**

127. To the extent the preamble of claim 12 is limiting, it is disclosed for the reasons in Section VI(A)(6)(a) (claim 7[preamble]).

- b. **“program code for transmitting to a server process, a network protocol address received by a first process following connection to a computer network;”**

128. WINS and NetBIOS teach this limitation for the reasons in Sections (A)(3)(b)-(d) (claim 3[A-C]).

- c. **“program code for transmitting, to the server process, a query as to whether a second process is connected to the computer network;”**

129. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(b) (claim 4[D.1]).

- d. **“program code for receiving a network protocol address of the second process from the server process, when the second process is connected to the computer network; and”**

130. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(c) (claim 4[D.2]).

- e. **“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”**

131. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(e) (claim 3[D]) and Section VI(A)(4)(c) (claim 4, limitation D.2).

132. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 12 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

12. **Claim 13 (Independent)**

- a. **“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:”**

133. To the extent the preamble of claim 13 is limiting, WINS and NetBIOS teach it for the reasons in Section VI(A)(3)(a) (claim 3, preamble) and Section VI(A)(9)(a) (claim 10[preamble]).

- b. **“a. transmitting to the server a network protocol address received by the first process following connection to the computer network;”**

134. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(b)-(d) and (A)(5)(b) (claims 3[A-C] and 6[A]).

- c. **“b. transmitting, to the server, a query as to whether the second process is connected to the computer network;”**

135. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(b) (claim 4[D.1]).

- d. **“c. receiving a network protocol address of the second process from the server, when the second process is connected to the computer network; and”**

136. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(c) (claim 4[D.2]).

- e. **“d. 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”**

137. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(e) (claim 3[D]), VI(A)(4)(c) (claim 4[D.2]), and VI(A)(5)(e) (claim 6[D]).

138. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 13 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

13. **Claim 14 (Independent)**

- a. **“An apparatus for use with a computer system, the computer system capable of executing a first process and operatively connectable to a second process and a server process over a computer network, the apparatus comprising:”**

139. To the extent the preamble of claim 14 is limiting, WINS and NetBIOS teach it for the reasons in Section VI(A)(3)(a) (claim 3[preamble]).

- b. **“program logic configured to transmit to the server a network protocol address received by the first process following connection to the computer network;”**

140. WINS and NetBIOS teach this limitation for the reasons in Sections (A)(3)(b)-(d) (claim 3[A-C]).

- c. **“program logic configured to transmit, to the server, a query as to whether the second process is connected to the computer network;”**

141. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(b) (claim 4[D.1]).

- d. **“program logic configured to receive a network protocol address of the second process from the server, when the second process is connected to the computer network; and”**

142. WINS and NetBIOS teach this limitation for the reasons in Section VI(A)(4)(c) (claim 4[D.2]).

- e. **“program logic, responsive to the network protocol address of the second process, and configured to establish a point-to-point communication link between the first process and the second process over the computer network”**

143. WINS and NetBIOS teach this limitation for the reasons in Sections VI(A)(3)(e) (claim 3[D]), VI(A)(4)(c) (claim 4[D.2]), and VI(A)(5)(e) (claim 6[D]).

144. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 14 would have been obvious to one of ordinary skill in the art in light of WINS and NetBIOS.

B. Examples Where Prior Art Teaches Claimed Features

Claim Limitation	Prior Art
[3-P] A computer program product for use with a computer system capable of executing a first process and communicating with other processes, a directory server process and a mail server process over a computer network, the computer program product comprising a computer usable medium having computer program code embodied in the medium, the program code comprising:	Ex. 1003 at 10-11, 17-20, 23, 29, 118-19; Ex. 1004 at 374, 377
[3-A] A. program code configured to determine the currently assigned network protocol address of the first process upon connection to the computer network;	Ex. 1003 at 13, 22, 57-58, 62-64, 83-115, 118-19; Ex. 1004 at 383, 394, 402, 416, 448-50
[3-B] B. program code configured to establish a communication connection with the directory server process once the assigned network protocol of the first process is known;	Ex. 1003 at 13, 20-22, 40-41, 64-67, 69, 72, 74-75, 79, 121, 151, 262, 265-66; Ex. 1004 at 402-03, 448-50

Claim Limitation	Prior Art
[3-C] C. program code configured to forward the assigned network protocol address of the first process to the directory server process upon establishing a communication connection with the directory server process; and	<i>See</i> 3-B
[3-D] D. program code configured to establish a point-to-point communication with another process over the computer network.	Ex. 1003 at 67, 73, 121-22, 266; Ex. 1004 at 383, 395, 406-07, 415-16, 422, 447, 458
[4-P] The computer program product of claim 3 wherein the program code configured to establish a point-to-point communication further comprises:	<i>See</i> 3-P
[4-A] D.1 program code configured to transmit, from the first process to the directory server process, a query as to whether a second process is connected to the computer network; and	Ex. 1003 at 13, 20-22, 40-41, 64-67, 69, 72-75, 79, 121-22, 151, 262, 265-66; Ex. 1004 at 378, 395-96, 400-03, 406-08, 412-13, 422, 447-50, 458-60
[4-B] D.2 program code configured to receive a network protocol address of the second process from the directory server process, when the second process is connected to the computer network.	<i>See</i> 4-A
[6-P] A computer program product for use with a computer system capable of executing a first process and connecting to other processes and a server process over a computer network, the computer program product comprising a computer usable medium having computer readable code means embodied in the medium comprising:	<i>See</i> 3-P
[6-A] A. program code configured to, following connection of the first process to the computer network, forward to the server process a dynamically assigned network protocol address at which the first process is connected to the computer network;	<i>See</i> 3-A, 3-B
[6-B] B. program code configured to query the address server as to whether the second process is connected to the computer network;	<i>See</i> 4-A

Claim Limitation	Prior Art
[6-C] C. program code configured to receive a dynamically assigned network protocol address of the second process from the address server, when the second process is connected to the computer network; and	<i>See</i> 3-A, 3-B, 4-A
[6-D] D. program code configured to respond to the network protocol address of the second process, establish a point-to-point communication link with the second process over the computer network.	<i>See</i> 3-D
[7-P] A computer data signal embodied in a carrier wave comprising:	<i>See</i> 3-P
[7-A] A. program code configured to, following connection of a first process to a computer network, forward to a server process a dynamically assigned network protocol address at which the first process is connected to the computer network;	<i>See</i> 3-A, 3-B
[7-B] B. program code configured to query the server process as to whether a second process is connected to the computer network;	<i>See</i> 4-A
[7-C] C. program code configured to receive a dynamically assigned network protocol address of the second process from the server process, when the second process is connected to the computer network; and	<i>See</i> 3-A, 3-B, 4-A
[7-D] D. program code, responsive to the network protocol address of the second process, and configured to establish a point-to-point communication connection with the second process over the computer network.	<i>See</i> 3-D
[8-P] An apparatus 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 directory database server process, the apparatus comprising:	<i>See</i> 3-P; Ex. 1003 at 12, 73; Ex. 1004 at 375
[8-A] A. program logic configured to, following connection of the first process to the computer network forward to the address server a dynamically assigned network protocol address at which the first process is connected to the computer network;	<i>See</i> 3-A, 3-B
[8-B] B. program logic configured to query the address server as to whether the second process is connected to the computer network;	<i>See</i> 4-A
[8-C] C. program logic configured to receive a dynamically assigned network protocol address of the second process from the address server, when the second process is connected to the computer network, and	<i>See</i> 3-A, 3-B, 4-A

Claim Limitation	Prior Art
[8-D] D. program logic configured to, in response to the network protocol address of the second process, establish a point-to-point communication link with the second process over the computer network.	<i>See</i> 3-D, 4-A
[9-P] A computer data signal embodied in a carrier wave comprising:	<i>See</i> 7-P
[9-A] a. program code configured to access a directory database, the database having a dynamically assigned network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network; and	<i>See</i> 3-A, 3-B, 4-A
[9-B] b. program code responsive to one of the dynamically assigned 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.	<i>See</i> 3-D, 4-A
[10-P] 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:	<i>See</i> 3-D
[10-A] a. accessing a directory database, the database having a dynamically assigned network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network; and	<i>See</i> 3-A, 3-B, 4-A
[10-B] b. in response to one of the dynamically assigned network protocol addresses, establish a point-to-point communication link from the first process to the second process over the computer network.	<i>See</i> 3-D, 4-A
[11-P] An apparatus for use with a computer system, the computer system capable of executing a first process connectable over a computer network to a second process and a directory database server process, the apparatus comprising:	<i>See</i> 8-P

Claim Limitation	Prior Art
[11-A] a. program logic configured to access a directory database, the database having a dynamically assigned network protocol address for a selected plurality of processes having on-line status with respect to the computer network, the dynamically assigned network protocol address of each respective process forwarded to the database following connection to the computer network; and	<i>See 3-A, 3-B, 4-A</i>
[11-B] b. program logic responsive to one of the dynamically assigned 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.	<i>See 3-D, 4-A</i>
[12-P] A computer data signal embodied in a carrier wave comprising:	<i>See 7-P</i>
[12-A] program code for transmitting to a server process, a network protocol address received by a first process following connection to a computer network;	<i>See 3-A, 3-B</i>
[12-B] program code for transmitting, to the server process, a query as to whether a second process is connected to the computer network;	<i>See 4-A</i>
[12-C] program code for receiving a network protocol address of the second process from the server process, when the second process is connected to the computer network; and	<i>See 4-A</i>
[12-D] 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.	<i>See 3-D, 4-A</i>
[13-P] 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:	<i>See 3-D</i>
[13-A] a. transmitting to the server a network protocol address received by the first process following connection to the computer network;	<i>See 3-A, 3-B</i>
[13-B] b. transmitting, to the server, a query as to whether the second process is connected to the computer network;	<i>See 4-A</i>
[13-C] c. receiving a network protocol address of the second process from the server, when the second process is connected to the computer network; and	<i>See 4-A</i>

Claim Limitation	Prior Art
[13-D] d. 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.	<i>See</i> 3-D, 4-A
[14-P] An apparatus for use with a computer system, the computer system capable of executing a first process and operatively connectable to a second process and a server process over a computer network, the apparatus comprising:	<i>See</i> 8-P
[14-A] program logic configured to transmit to the server a network protocol address received by the first process following connection to the computer network;	<i>See</i> 3-A, 3-B
[14-B] program logic configured to transmit, to the server, a query as to whether the second process is connected to the computer network;	<i>See</i> 4-A
[14-C] program logic configured to receive a network protocol address of the second process from the server, when the second process is connected to the computer network; and	<i>See</i> 4-A
[14-D] program logic, responsive to the network protocol address of the second process, and configured to establish a point-to-point communication link between the first process and the second process over the computer network.	<i>See</i> 3-D, 4-A

145. I understand and have been warned that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. § 1001). I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under § 1001 of title 18 of the United States Code.

146. I declare under penalty of perjury that the foregoing is true and correct.
Executed on October 27, 2014 in Los Angeles, CA.

/s/ Bruce M. Maggs

Bruce M. Maggs, Ph.D.