

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

VONAGE HOLDINGS CORP., VONAGE AMERICA, INC., VONAGE
MARKETING LLC, and NETFLIX, Inc.,
Petitioner

v.

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

INTER PARTES REVIEW OF U.S. PATENT NO. 6,701,365
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. SUMMARY OF THE '365 PATENT.....	17
A. Summary of the Alleged Invention.....	17
2. Step 1: Processing Units Obtain Dynamically Assigned IP Addresses.....	18
3. Step 2: Processing Units Register Their IP Addresses and Identifiers with a Connection Server.....	19
4. Steps 3 & 4: First Processing Unit Sends Query to Connection Server, Which Returns IP Address of Second Processing Unit...20	20
5. Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communication with Second Processing Unit	21
B. Original Prosecution of the '365 patent.....	22
C. Prior <i>Ex Parte</i> Reexamination of the '365 patent.....	22
V. OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES	24
A. The Microsoft Manual (Ex. 1003)	24
2. Step 1: Processing Units Obtain Dynamically Assigned IP Addresses from DHCP Servers	26
3. Step 2: Processing Units Register Their IP Addresses and Identifiers with the WINS Server.....	28

4.	Step 3 & 4: First Processing Unit Sends Query to WINS Server and Receives the IP Address of the Second Processing Unit	32
5.	Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communication with Second Processing Unit	33
B.	NetBIOS (Ex. 1004)	34
2.	Step 1: Processing Units Have Assigned IP Addresses	35
3.	Step 2: Processing Units Register Their IP Addresses and Identifiers with the NBNS	36
4.	Steps 3 & 4: First Processing Unit Sends Query to the NBNS and Receives the IP Address of the Second Processing Unit	38
5.	Step 5: First Processing Unit Uses Received IP Address to Establish Point-to-Point Communications with Second Processing Unit	38
VI.	SPECIFIC GROUNDS FOR PETITION.....	39
A.	Claims 1-3 Would Have Been Obvious Over the Microsoft Manual and NetBIOS	39
1.	Level of Ordinary Skill in the Art	42
2.	One Skilled in the Art Would Have Been Motivated to Combine the Microsoft Manual and NetBIOS	43
3.	Claim 1 (Independent).....	44
4.	Claim 2 (Independent).....	57
5.	Claim 3 (Independent).....	58
B.	Claim Chart with Examples Showing Where Prior Art Teaches Claimed Features	64
VII.	CONCLUSION	67

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

1. I have been retained by Wilmer Cutler Pickering Hale and Dorr LLP and Kecker & Van Nest LLP, counsel for Petitioner Vonage Holding Corp., Vonage America, Inc., Vonage Marketing LLC, and Netflix, Inc., to submit this declaration in connection with Petitioner's Petition for *Inter Partes* Review of Claims 1-3 of U.S. Patent No. 6,701,365 ("the '365 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 '365 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 1015, 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 '365 patent (Ex. 1001) and the following other documents:

Exhibit	Description
1001	U.S. Patent No. 6,701,365 (“’365 patent”)
1002	Declaration of Dr. Bruce M. Maggs (“Maggs Decl.”)
1003	Microsoft Windows NT Server Version 3.5 (“Microsoft Manual”)
1004	Technical Standard: Protocols for X/Open PC Interworking: SMB, Version 2 (“NetBIOS”)
1005	Declaration of Jason Liss Regarding Documentation of Microsoft Windows NT Server Version 3.5
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)

Exhibit	Description
1011	Institution Decision in Sipnet EU S.R.O. v. Straight Path IP Group, Inc. (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)
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 ’365 Patent (Apr. 19, 2002 Amendment)
1021	Excerpt from File History for ’365 Patent (July 17, 2002 Request for Reconsideration)
1022	Excerpt from File History for ’365 Patent (Feb. 13, 2002 Office Action)
1023	Excerpt from File History for ’365 Patent (June 20, 2002 Office Action)
1024	Excerpt from File History for ’365 Patent (Oct. 8, 2002 Amendment)
1025	Excerpt from File History for ’365 Patent (Dec. 31, 2002 Notice of Allowance)
1026	Excerpt from Reexamination File History for ’365 Patent (Aug. 14, 2009 Office Action)

Exhibit	Description
1027	Excerpt from Reexamination File History for '365 Patent (May 11, 2010 Notice of Intent to Issue Reexamination Certificate)

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

III. THE BASICS OF NETWORK COMMUNICATION

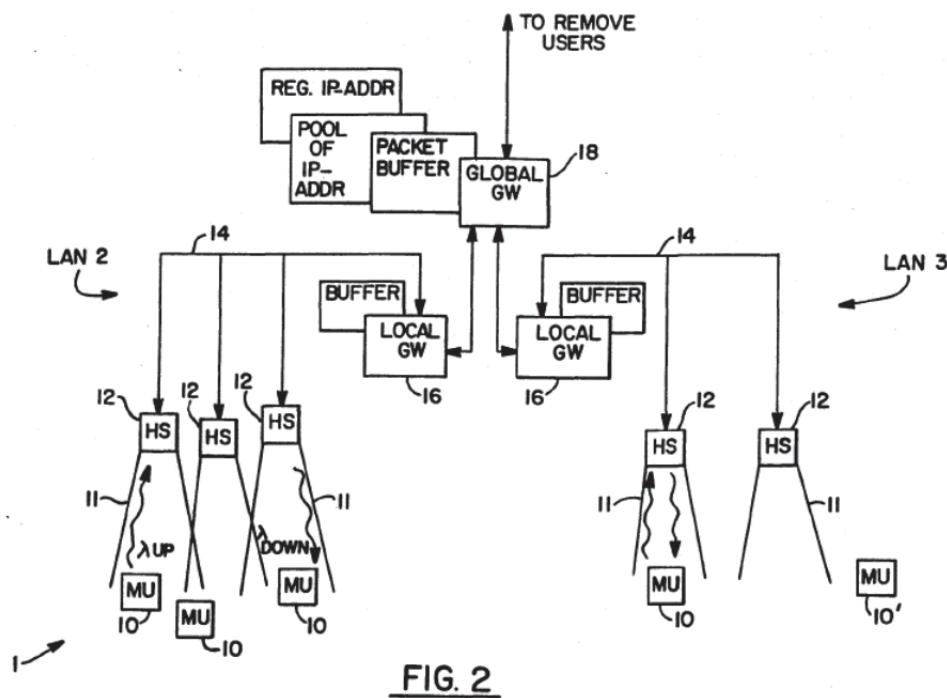
A. Computer Network Hardware Configurations

12. The '365 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 '365 patent was filed in June 1999, and when the application to which the '365 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 '365 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 '365 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), and the Post Office Protocol (POP). (Ex. 1001, 1:29-37 (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.*, 3:41-42 (discussing POP).)

16. As of the claimed priority date of the '365 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 '365 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, 1:42-48 (“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 '365 patent does not claim to invent a new way to assign static network addresses. (Ex. 1001, 1:55-59 (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 ’365 patent does not claim to invent a new way to assign addresses. (Ex. 1001, 1:48-54 (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.*, 1:60-62 (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

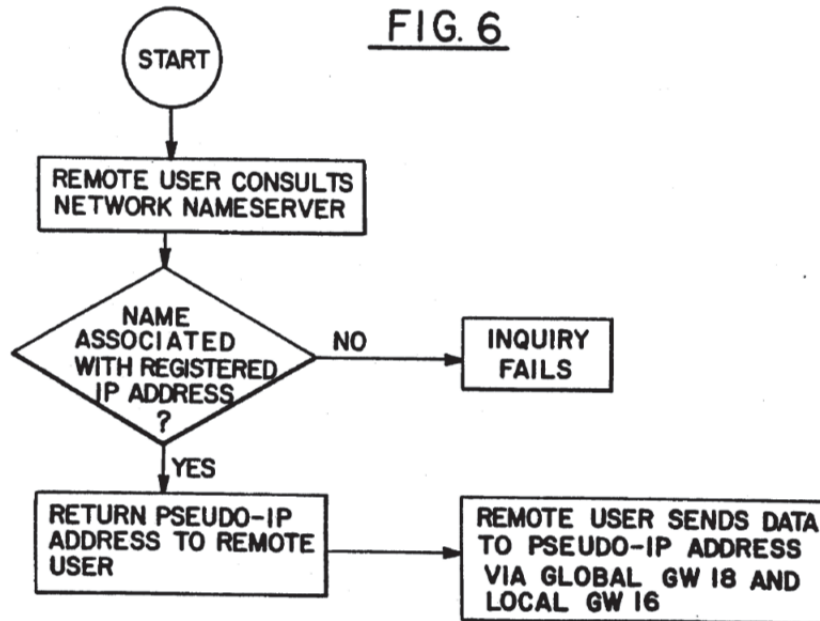
the network address of the second device. (Ex. 1019, 6:35-38, 7:5-7 (explaining that “[a]ll communication from a remote user to a mobile unit 10 employs the pseudo-IP address of the mobile unit 10,” and that if a remote user already knows the pseudo-IP address of a desired mobile unit, the remote user 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 ’365 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. SUMMARY OF THE '365 PATENT

A. Summary of the Alleged Invention

34. The '365 patent concedes that, in the prior art, a first processing unit could establish “point-to-point communications” with a second processing unit using the network address assigned to the second processing unit, “in a manner known in the art.” (Ex. 1001, 1:29-31 (“[D]evices interfacing to the Internet and other online services may communicate with each other upon establishing respective device addresses.”); *id.*, 1:55-57, 8:6-8, 8:30-33 (“Permanent IP addresses of users and devices accessing the Internet readily support point-to-point communications of voice and video signals over the Internet,” which “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”).) According to the '365 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). (*Id.*, 1:48-51, 1:60-63.)

35. The '365 patent claimed to solve that supposed “problem” through the basic lookup feature described in connection with Figure 8 below:

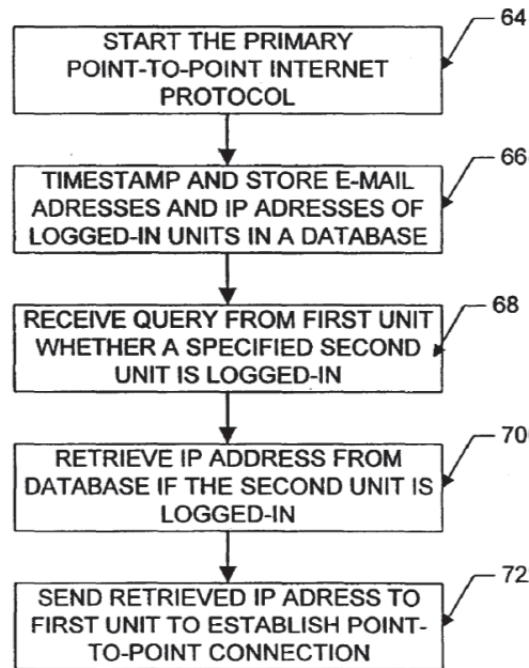


FIG. 8

(Ex. 1001, Fig. 8.)

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

36. When a “processing unit” (the term used in the specification of the ’365 patent) or “process” (the term used in the claims)¹ “logs on to the Internet ..., the respective [device] is provided a dynamically allocated IP address by a Connection Service Provider.” (Ex. 1001, 5:27-34.)

¹ For the sake of convenience, and because it is the term used in the specification of the ’365 patent, I use the term “processing unit” in Sections IV and V of this Declaration.

3. 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, 5:35-49; *id.*, 10:15-31 (“[C]onnection server 26 ... timestamp[s] and stor[es] E-mail and IP addresses of logged-in users and processing units in the database 34 in step 66.”); Ex. 1020 [4/19/02 Amendment] at 4 (“One embodiment utilizes a dedicated server which acts as a network address/information directory from which calling processes can obtain information. When a first process connects to the network, the process logs-on to the server and provides the server with the network protocol address under which the first process is currently operating.”); Ex. 1021 [7/17/02 Request] at 2.)

38. Connection server 26 keeps “relatively current” data concerning the “on-line status” of registered processing units; e.g., it may confirm that a registered processing unit remains online after “predetermined time periods, such as a default value of 24 hours.” (Ex. 1001, 5:50-55.) 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.*, 6:17-20.)

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

39. When the user of a first processing unit wishes to establish point-to-point communications with a second processing unit, the user of the first processing unit “sends a query, including the E-mail address [(or other identifier)] of the callee, to the connection server 26.” (Ex. 1001, 5:66-67.) The query can include, for instance, “the name or alias ... of a party to be called.” (*Id.*, 9:37-44; Ex. 1020 [4/19/02 Amendment] at 4 (“A second process, wishing to establish communications with the first process, connects to the server and request [sic] the network protocol address under which the first process is currently operating.”); Ex. 1021 [7/17/02 Request] at 2.)

40. Upon receiving the query, connection server 26 “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, 6:1-4.) “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.*, 6:4-8; *id.*, 10:42-45 (connection server “retriev[es] the IP address of the specified user from the database 34 in step 70 if the specified user is logged-in to the Internet”); Ex. 1020 [4/19/02 Amendment] at 4 (“Upon receipt of the network protocol address of the first process, the second

process establishes communications with the first process directly, without any intervention from the address/information server.”); Ex. 1021 [7/17/02 Request] at 2.)

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

41. After receiving the IP address of the second processing unit from connection server 26, “[t]he first [processing unit] may then directly establish point-to-point Internet communications with the [second processing unit] using the IP address of the [second processing unit].” (Ex. 1001, 6:8-11; *id.*, 10:45-47 (IP address received from connection server 26 enables “the first [processing unit] to establish point-to-point Internet communications with the [second processing unit].”); Ex. 1020 [4/19/02 Amendment] at 4 (“Upon receipt of the network protocol address of the first process, the second process establishes communications with the first process directly, without any intervention from the address/information server.”); Ex. 1021 [7/17/02 Request] at 2.)

42. The ’365 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, 8:7-8, 8:31-33 (emphases added).) The '365 patent also functions the same “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.*, 7:49-52.)

B. Original Prosecution of the '365 patent

43. During the original prosecution of the '365 patent, the Examiner initially rejected all claims, but then issued the '365 patent after the patentee amended the claims. (Ex. 1022 [2/13/02 Office Action]; Ex. 1023 [6/20/02 Office Action]; Ex. 1024 [10/8/02 Amendment]; Ex. 1025 [12/31/02 Notice of Allowance].) In forming my opinions I have not relied on any prior art cited during the original prosecution.

C. Prior *Ex Parte* Reexamination of the '365 patent

44. On February 23, 2009, a third party requested *ex parte* reexamination of the '365 patent. During that proceeding, the Examiner rejected all claims under 35 U.S.C. § 103 in light of NetBIOS and RFC 1531 (*i.e.*, the Dynamic Host Configuration Protocol (DHCP)). (Ex. 1026 [8/14/09 Office Action] at 3-5.) Although the Examiner agreed that NetBIOS describes the same address lookup functionality claimed in the '365 patent, he 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].” (Ex. 1027 [5/11/10 Notice] at 4.)

45. But the expert declaration failed to note that other prior art, including the Microsoft Manual, disclosed using dynamic addressing in a NetBIOS-type system. (*E.g.*, Ex. 1003 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.”); *id.* 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 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.”)).)

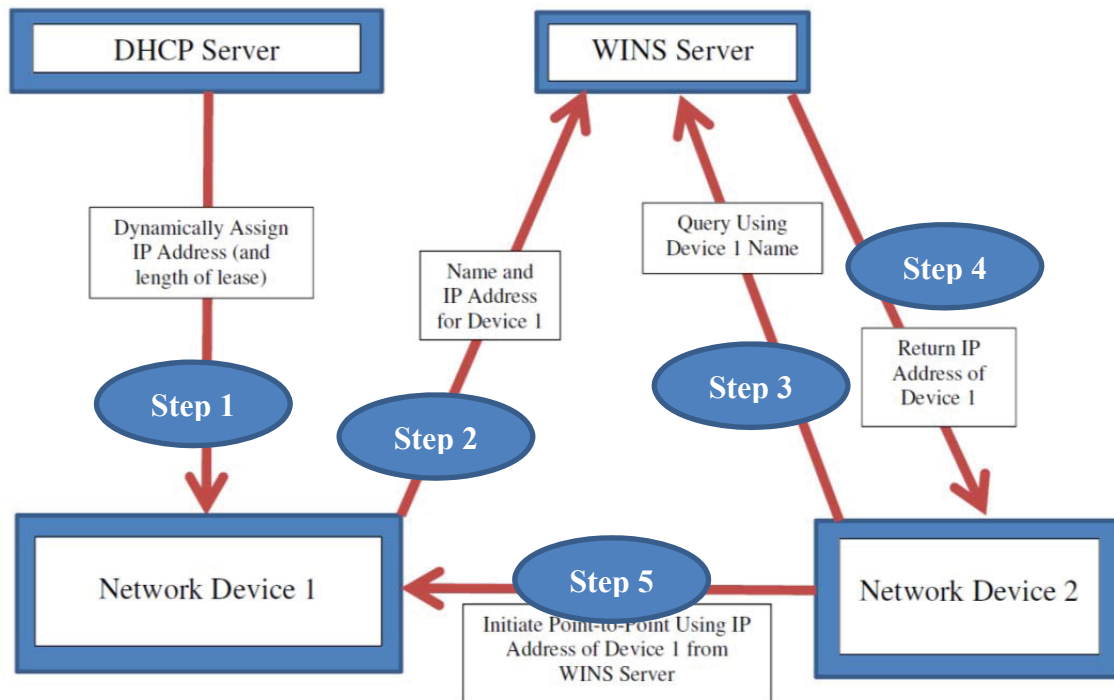
46. Relying on the flawed expert declaration, the Examiner confirmed the claims of the ’365 patent without amendment. (Ex. 1027 [5/11/10 Notice] at 6.)

V. OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES

A. The Microsoft Manual (Ex. 1003)

47. In 1994, Microsoft published and publicly distributed its Microsoft Windows NT Server TCP/IP manual (“Microsoft Manual”), 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.) As discussed below, the Microsoft Manual teaches how a processing unit (for example, a Windows 3.11 workstation running WINS client software) or process (for example, the Windows 3.11 operating system) can initiate

point-to-point communications with other processing units using the same five-step process described in the '365 patent.²



² The Microsoft Manual explains that the Windows NT operating system (v3.5) 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 compatible non-Microsoft networks); *id.* at 55-56 (outlining history and general operation of TCP/IP).)

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

48. The Microsoft Manual explains that each processing unit participating on a Windows NT network “must be assigned a unique IP address” (Ex. 1003 at 57-58), and that Windows NT networks can use a DHCP server to assign IP addresses “automatically” to processing units at start up—including dynamically (*id.* at 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 (explaining 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 software and configuration options, including for “a 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. This configuration information includes 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 processing unit] 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 Microsoft Manual makes clear that the lease period is set by the system administrator, and can either be “short” (e.g., “a few days”) or “long” (e.g., “two months”). (*Id.* at 115.)

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

51. The Microsoft Manual 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, the Microsoft Manual teaches that Windows NT is used to assign each device a “NetBIOS” name during “setup.” (*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. The Microsoft Manual further teaches that a Windows Internet Name Service (WINS) server consists of “a Windows NT Server computer running Microsoft TCP/IP and the [WINS] server software” and “contain[s] 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 (“In a Windows network, a WINS server can provide name registration

services” and is “[t]he server implemented under RFC 1001/1002 to provide name resolution services for NetBIOS computer names.”).³

53. The Microsoft Manual teaches that this mapping of names to IP addresses performed by the WINS server “provides a powerful new name resolution service for easy, centralized management of 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 (defining “WINS” as “[a] name resolution service that resolves Windows networking computer names to IP addresses in a routed environment” and 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”).)

54. The Microsoft Manual also explains that, in a Windows NT system, “WINS name resolution is enabled and configured automatically for a computer that is configured with DHCP.” (Ex. 1003 at 40.) In such a system, after the DHCP server dynamically assigns an IP address to a processing unit, the unit transmits its assigned IP address and device name to the WINS server as part of the system

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

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. The Microsoft Manual also teaches that the WINS 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.) Similarly, 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.) The Microsoft Manual also explains that, “[to] provide[] registration reliability through periodic reregistering of names with the WINS servers,” if a network device fails to re-register its name within the set “renewal time,” “the WINS server will mark the name as released and available for use.” (*Id.*)

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

56. The Microsoft Manual 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.”).)

57. The Microsoft Manual further teaches that a first processing unit—e.g., the Windows NT or Windows for Workgroups 3.11 operating system running on a first computer—can obtain the IP address of a second processing unit—e.g., a the Windows NT or Windows for Workgroups 3.11 operating system running on a second computer—by transmitting a query to the WINS server containing the name of the second processing unit—which causes the WINS server to return the IP

address of the second processing unit. To do so, the “client software” of the first processing unit sends “[a] *name query request*” to the WINS server, which returns the IP “address mapping” of the second processing unit to the first processing unit. (Ex. 1003 at 72-73 (“Any name-to-IP address mapping registered with a WINS server can be provided reliably as a response to a name query.”); *id.* at 67 (“[W]hen NT_PC1 wants to communicate with NT_PC2, it queries the WINS server for the address of NT_PC2 ... gets the appropriate address from the WINS server, [and] goes directly to NT_PC2”); *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”).)

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

58. 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 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 (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”); *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.”).)

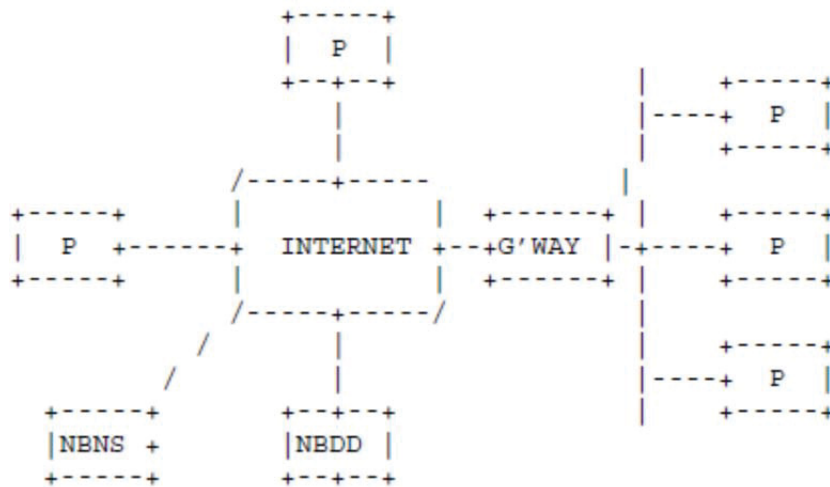
B. NetBIOS (Ex. 1004)

59. In March 1987, Network Working Group of the IETF published RFC 1001, which “describes the ideas and general methods used to provide NetBIOS on a TCP and UDP foundation,” and RFC 1002, which “contains the detailed packet formats and protocol specifications for NetBIOS-over-TCP.” (Ex. 1004 at 368, 374, 442.)⁴ In 1992, RFCs 1001 and 1002 were published as Appendices F and G in *Technical Standard – Protocols for X/Open PC Interworking: SMB, Version 2* (Ex. 1004).

60. NetBIOS teaches point-to-point communications between two or more nodes over a network (including LANs and the Internet), including between

⁴ Network Basic Input/Output System (NetBIOS) is a protocol first developed in the early 1980s that allows applications on different computers to communicate across a computer network, such as a local area network or the Internet. (Ex. 1004 at 374, 377.)

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



(Ex. 1004 at 389.)

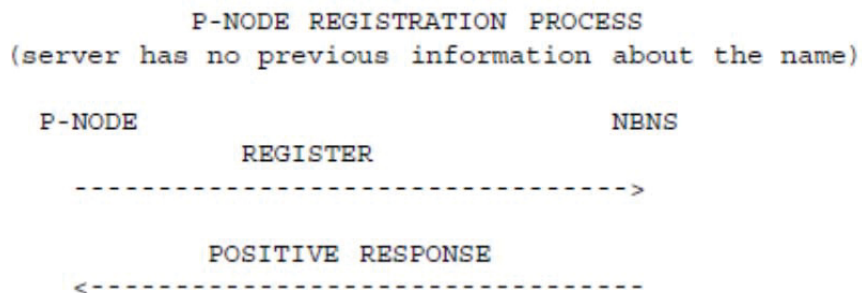
2. Step 1: Processing Units Have Assigned IP Addresses

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

the interface defined by NetBIOS running on a computer) must obtain an IP address to participate on the network.

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

62. A p-node with an IP address sends its name (“RR_NAME”) and assigned IP address (“NB_ADDRESS”) to the NBNS, and if the name is unique and accepted, the NBNS returns a “Positive Response” message confirming registration (as shown below):



(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 (noting “NAME REGISTRATION REQUEST” includes p-node’s name (“RR_NAME”) and IP address (“NB_ADDRESS”).)

63. The NBNS 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.”).) In this scenario, the node is treated as offline if

it fails to send a “name refresh message” during a “lifetime” period that the NBNS established 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 (“The 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 (“Each P or M node is 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.”).)

64. By contrast, “explicit release” occurs when “P nodes send a notification to their NBNS.” (Ex. 1004 at 395; *id.* at 378 (“An explicit name deletion function is specified, so that applications may remove a name.”).)

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

65. 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” to the NBNS that includes the name of the second p-node. (Ex. 1004 at 447, 458 (“NAME QUERY REQUEST” includes name of second p-node (“QUESTION_NAME”)); *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 has previously registered its name with the NBNS, the NBNS provides a “POSITIVE RESPONSE” to the first p-node that includes the IP address of the second p-node. (*Id.* at 459; *id.* at 407 (“An NBNS answers queries from a P node with a list of IP address and other information [for the queried name].”); *id.* at 422 (“To transmit a NetBIOS datagram, the datagram service must perform a name query operation to learn the IP address and the attributes of the destination NetBIOS name.”).)

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

66. 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) (“The NetBIOS session service begins after one or

more IP addresses have been found for the target name. These addresses may have been acquired using the NetBIOS name query transactions NetBIOS session service transactions, packets, and protocols are identical for all end-node types. They 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”); *id.* (each data packet includes the first and second p-node “IP address” and “NetBIOS name”); *id.* at 419 (explaining point-to-point communications if “a name query operation was successfully completed by the caller node for the listener’s name”).)

VI. SPECIFIC GROUNDS FOR PETITION

A. Claims 1-3 Would Have Been Obvious Over the Microsoft Manual and NetBIOS

67. 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.

68. 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 1-3 of the ’365 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 ’365 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 ’365 patent.

69. 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.

70. 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.

71. The following description demonstrates in detail that claims 1-3 would have been obvious under 35 U.S.C. § 103(a) over the combined teachings of the Microsoft Manual and NetBIOS.⁵ The discussion below contains non-limiting examples of how the prior art teaches each claimed feature and additional examples are set forth in the claim chart at Section VII(B) and elsewhere in the prior art.

1. Level of Ordinary Skill in the Art

72. 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

⁵ I have been informed and understand that in *inter partes* review, claims are construed according to their broadest reasonable interpretation. In forming my opinions I have applied the constructions set forth in Section VII of the '365 Patent Petition.

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.

73. In my opinion, a person of ordinary skill in the art related to the '365 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.

74. As described in more detail above, I was a person with at least ordinary skill in the art for the '365 patent at the time of its alleged invention.

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

75. The Microsoft Manual clearly discloses that NetBIOS (as defined in RFCs 1001 and 1002) is incorporated into, and 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, the Microsoft Manual refers to NetBIOS over one hundred times (*id.* at 5, 20, 22, 23, 28, 35, 40, 41, 44, 45, 53, 54, 64-67, 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, 274-77) and refers to NetBIOS protocols RFCs 1001 and 1002 ten times (*id.* at 22, 65, 66, 69, 102, 103, 167, 262, 265).

76. Because the Microsoft Manual (Ex. 1003) expressly teaches that its disclosed system should be 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 1 (Independent)

- a. **“A computer program product for use with a server operatively coupled over a computer network to a plurality of processes, the computer program product comprising a computer usable medium having program code embodied thereon the program code comprising:”**

77. To the extent that the preamble of claim 1 is limiting, the Microsoft Manual discloses a “computer program product ... comprising a computer useable medium having program code embodied thereon” that is “for use with a server.” (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.”).)

78. The Microsoft Manual further explains that such a product is used with a server (e.g., a WINS/NetBIOS server) coupled over a computer network (e.g., a LAN, WAN, or the Internet) to a plurality of processes (e.g., instances of the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on computers). (Ex. 1003 at 29 (Windows NT Server 3.5 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 24 (“Microsoft TCP/IP-32 for Windows for Workgroups 3.11 ... can be used to provide access for Windows for Workgroups computers to Windows NT, LAN Manager, and other TCP/IP systems. Microsoft TCP/IP-32 includes DHCP and WINS client software.”); *id.* at 118-19 (explaining use for “a small LAN” and “large enterprise network”); *see* Section V(A) (“The Microsoft Manual (Ex. 1003)”)).)

79. In addition, NetBIOS teaches that “NetBIOS on personal computers includes both a set of services and an exact program interface to those services.” (Ex. 1004 at 377; *id.* at 374 (“NetBIOS defines a software interface . . . [and] has generally been confined to personal computers to date. However, since larger

computers are often well suited to run certain NetBIOS applications, such as file servers, this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available.”); *see generally* Section V(B) (“NetBIOS (Ex. 1004)”.)

80. Thus, the prior art teaches a computer program product (e.g., Microsoft Windows NT Server 3.5) for use with a server (e.g., a WINS/NetBIOS server) operatively coupled over a network (e.g., a LAN, WAN, or the Internet) to a plurality of processes (e.g., instances of the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on computers). Also, the computer program product comprises a computer usable medium having computer program code embodied in the medium (e.g., the Windows NT Server 3.5 compact disc and memory in the server).

b. **“a. program code configured to receive the current network protocol address of one of the processes coupled to the network, the network protocol address being received by said one of the processes from an Internet access server;”**

(1) The Microsoft Manual and NetBIOS disclose “[a] network protocol address being received by said one of the processes from an Internet access server”

81. The Microsoft Manual discloses that every device participating 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 such as a Dynamic Host

Configuration Protocol (DHCP) server assigns IP addresses automatically to network processes—including dynamically assigned addresses. (*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 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.”); *id.* at 13, 22, 62, 63, 85-120 (describing DHCP software and configuration options).)

82. To do so, “[d]uring system startup,” a computer running WINS client software 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 then 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 acknowledgment, it enters a *bound* state and can now participate on the TCP/IP network and complete its system startup.” (*Id.* at 64; see Section V(A)(1) (“Processing Units Obtain Dynamically Assigned IP Addresses (from DHCP Servers)”).)

83. Similarly, NetBIOS discloses that p-nodes have “IP addresses ... obtained from the TCP service interface.” (Ex. 1004 at 416; *see* Section V(B)(1) (“Processing Units Have Assigned IP Addresses”).)

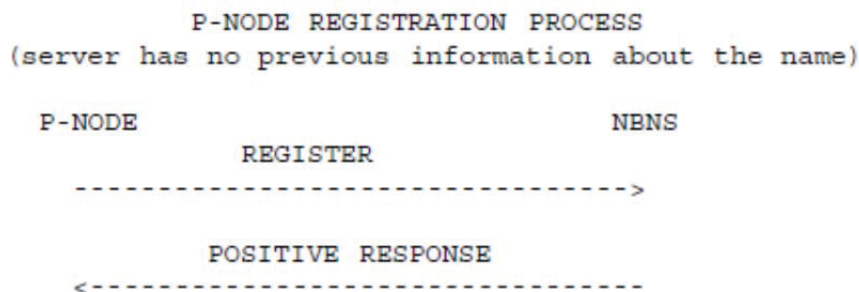
84. Accordingly, the prior art teaches a network protocol address (e.g., an IP address) being received by said one of the processes (e.g., the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on a computer registering with the DHCP server) from an Internet access server (e.g., the DHCP server).

(2) The Microsoft Manual and NetBIOS disclose “program code configured to receive the current network protocol address of one of the processes coupled to the network.”

85. As disclosed in the Microsoft Manual, after the DHCP server dynamically assigns an IP address to the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on the first computer, the WINS client transmits the 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 (“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

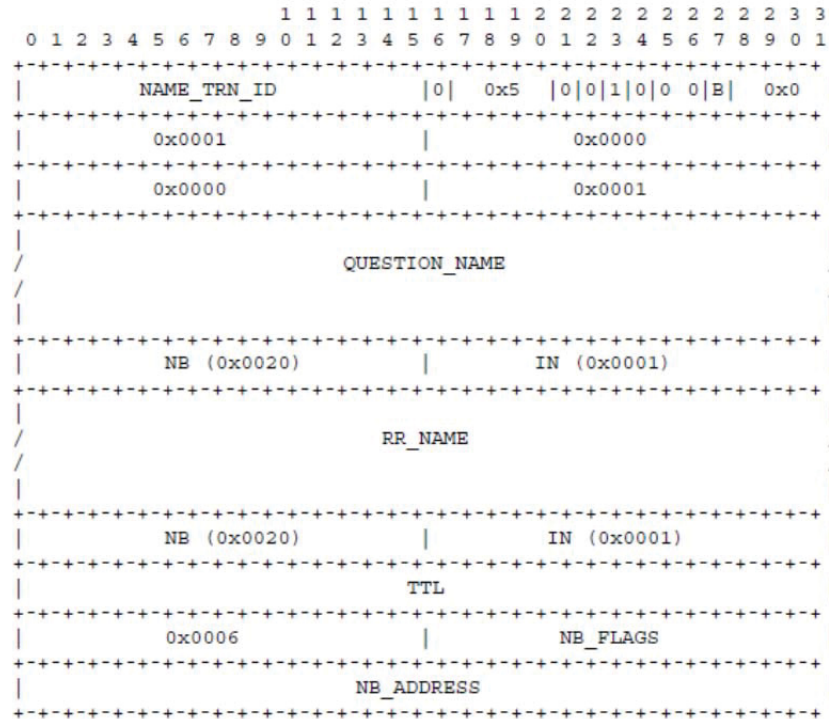
enhancements.”); *id.* at 69 (“WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment.”); *id.* at 72 (“During TCP/IP configuration, the computer’s name is registered with the WINS server, and the IP address of the WINS server”); *id.* at 74 (“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 (showing WINS database entry including “NetBIOS computer name” and “assigned Internet Protocol address”); *see* Section V(A)(2) (“Processing Units Register Their IP Addresses and Identifiers with the WINS Server”).)

86. NetBIOS specifies further 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”:



(Ex. 1004 at 402-03.) NetBIOS also specifies the format for 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



(Ex. 1004 at 448-50; *see* Section V(B)(2) (“Processing Units Register Their IP Addresses and Identifiers with the NBNS”).)

87. Therefore, the prior art discloses program code (e.g., Windows NT software running on the WINS/NetBIOS server) configured to receive the current network protocol address (e.g., IP address assigned by the DHCP server) of one of the processes (e.g., the the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on the first computer) coupled to the network.

c. “b. program code configured to receive an identifier

associated with said one process; and”

88. The Microsoft Manual 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, the disclosed Windows NT software is used to assign a “NetBIOS” name to each computer during “setup,” which the WINS client then 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.)

89. Accordingly, for these reasons, as well as the reason set forth in Section VI(A)(3)(b)(1)-(2) in connection with the preamble and limitation “a” of claim 1, the prior art teaches program code (e.g., Windows NT software running on the WINS/NetBIOS server) configured to receive an identifier (e.g., computer name) associated with said one process (e.g., the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on a computer).

d. **“c. program code configured to receive queries for one of the network protocol address and the associated identifier of said one of the processes from other processes over the computer network at the server, and to allow the establishment of a packet-based point-to-point communication between said one of the processes and one of said other processes.”**

(1) The Microsoft Manual and NetBIOS disclose “program code configured to receive queries for one of the network protocol address and the associated identifier of said one of the processes from other processes over the computer network at the server”

90. The Microsoft Manual teaches that: (1) WINS/NetBIOS server software “contain[s] a dynamic database mapping computer names to IP addresses” (Ex. 1003 at 65); (2) WINS client software running on the Windows NT 3.5 or Windows for Workgroups 3.11 operating system on a first computer can request the WINS/NetBIOS server to retrieve the IP address of the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on a second computer by sending the name of the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on the second computer to the WINS server in a “*name query request*” (*id.* at 73 (“If the name is found in the WINS database”)); and (3) the WINS/NetBIOS server will respond to the name query request by providing the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on the first computer with the IP address of the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on the second computer (if the queried

name is currently mapped 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”).) Thus, “when NT_PC1 [e.g., a first computer running the Windows NT operating system] wants to communicate with NT_PC2 [e.g., a second computer running the Windows NT operating system], it queries the WINS server for the address of NT_PC2 [e.g., of the first computer running the Windows NT operating system] ... [and] gets the appropriate address [e.g., of the first computer running the Windows NT operating system] from the WINS server” (*Id.* at 67; *see also* Section V(A)(2) (“Processing Units Register Their IP Addresses and Identifiers (with the WINS Server)”; Section V(A)(3) (“First Processing Unit Sends Query to WINS Server and Receives the IP Address of the Second Processing Unit”).)

91. NetBIOS similarly specifies that “[n]ame query transactions are initiated by end-nodes [e.g., the first process] to obtain the IP address(es) and other attributes associated with a NetBIOS name [of the second process].” (Ex. 1004 at 406). It also explains that a “NAME QUERY REQUEST” includes the desired name (e.g., “QUESTION_NAME”) of the first process. (Ex. 1004 at 458; *see*

Section V(B)(2) (“Processing Units Register Their IP Addresses and Identifiers With the NBNS”) and Section V(B)(3) (“First Processing Unit Sends Query to the NBNS and Receives the IP Address of the Second Processing Unit”).)

92. Accordingly, the prior art teaches program code configured to receive queries (e.g., Windows NT software running on a WINS/NetBIOS server, which is configured to receive name query requests) for one of the network protocol addresses (e.g., the IP address) and the associated identifier (e.g., the name) of said one of the processes (e.g., the Windows NT or Windows for Workgroups 3.11 operating system running on a first computer) from other processes (e.g. the Windows NT or Windows for Workgroups 3.11 operating system running on a second computer) over the computer network at the server (e.g., the WINS/NetBIOS server).

- (2) The Microsoft Manual and NetBIOS disclose “program code configured ... to allow the establishment of a packet-based point-to-point communication between said one of the processes and one of said other processes”

93. The Microsoft Manual also teaches how the Windows NT software is used to establish a point-to-point communication with another process over the computer network: “[i]n a p-node environment,” “when NT_PC1 [e.g., a first computer running the Windows NT operating system] wants to communicate with NT_PC2 [e.g., a second computer running the Windows NT operating system], 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.” (Ex. 1003 at 67; *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”).)

94. In addition, the Microsoft Manual explains that the disclosed system supports packet-based communication, for example via TCP/IP, which “is a protocol family that can be used to offer Windows networking capabilities.” (Ex. 1003 at 54.) “IP is a protocol that provides packet delivery for all other protocols within the TCP/IP family[, and] IP packets are not guaranteed to arrive at their destination, nor are they guaranteed to be received in the sequence in which they were sent.... [The] responsibility for the data contained within the IP packet (and the sequencing) is assured only by using higher-level protocols [such as TCP]. TCP supplies a reliable, connection-based protocol over (or encapsulated within) IP. TCP guarantees the delivery of packets This reliability makes TCP/IP the protocol of choice for session-based data transmission, client-server applications, and critical services such as electronic mail.” (*Id.* at 55; *see* Section VI(A)(3) (“First Processing Unit Sends Query to WINS Server and Receives the IP Address of the Second Processing Unit”) and Section V(A)(4) (“First Processing Unit Uses

Received IP Address to Establish Point-to-Point Communication with Second Processing Unit”).)

95. NetBIOS discloses that p-nodes (such as those implemented by WINS) are “[p]oint-to-point 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 called computer].” (*Id.* at 415-16.)

96. Accordingly, the Microsoft Manual and NetBIOS teach program code configured to allow the establishment of a packet-based point-to-point communication (e.g., via TCP/IP) between said one of the processes (e.g., the Windows NT or Windows for Workgroups operating system running on a first computer) and one of said other processes (e.g., the Windows NT or Windows for Workgroups operating system running on a second computer). Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 1 would have been in obvious to one of ordinary skill in the art in light of the Microsoft Manual and NetBIOS.

4. Claim 2 (Independent)

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

97. To the extent that the preamble of claim 2 is limiting, the Microsoft Manual and NetBIOS teach it because one of ordinary skill in the art would understand that a computer data signal embodied in a carrier wave would be created, among other times, during installation of the Windows NT or WINS client software from the CD-ROM on which it is provided. (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 receive a current network protocol address of a process coupled to a computer network, the network protocol address being received by the process from an Internet access server;”

98. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(b) in connection with claim 1, limitation “a.”

c. “b. program code configured to receive an identifier associated with said one process; and”

99. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(c)-(d) in connection with claim 1, limitations “b” and “c.”

- d. **“c. program code configured to receive queries for one of the network protocol address and the associated identifier of said one process from other processes over the computer network at a connection server, and to allow the establishment of a packet-based point-to-point communication between said one process and one of said other processes.”**

100. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(d) in connection with claim 1, limitation “c.”

101. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 2 would have been obvious to one of ordinary skill in the art in light of the Microsoft Manual and NetBIOS.

5. Claim 3 (Independent)

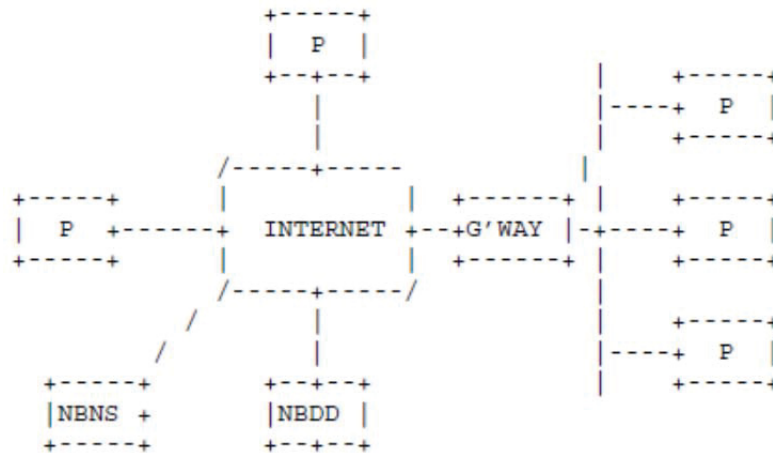
- a. **“In a computer system operatively coupled over a computer network to a plurality of processes, a method comprising the steps of:”**

102. To the extent that the preamble of claim 3 is limiting, the Microsoft Manual and NetBIOS teach it for the same reasons set forth above in Section VII(A)(3)(a) in connection with the preamble of claim 1.

103. In addition, the Microsoft Manual teaches that a WINS server is used in a networked computer system hardware environment. (*E.g.*, Ex. 1003 at 12 (“[The Microsoft 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 73 (showing computers “Corp01” and “Payroll” and WINS server).)

104. Similarly, NetBIOS discloses the NBNS operatively coupled over a computer network to a plurality of processes:



(Ex. 1004 at 389; *id.* at 374 (“The NetBIOS service has become the dominant mechanism for personal computer networking. NetBIOS provides a vendor independent interface for the IBM Personal Computer (PC) and compatible systems. . . . NetBIOS has generally been confined to personal computers to date. However, since larger computers are often well suited to run certain NetBIOS applications, such as file servers, this specification has been designed to allow an implementation to be built on virtually any type of system where the TCP/IP protocol suite is available”); *id.* at 375 (“NetBIOS is the foundation of a large body of existing applications. It is desirable to operate these applications on TCP networks and to

extend them beyond personal computers into larger hosts. To support these applications, NetBIOS on TCP must closely conform to the services offered by existing NetBIOS systems.”.)

- b. “a. receiving the current network protocol address of a process coupled to the network, the network protocol address being received by the process from an Internet access server;”**

105. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(b) in connection with claim 1, limitation “a.”

- c. “b. receiving an identifier associated with said one process;”**

106. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(c) in connection with claim 1, limitation “b.”

- d. “c. receiving a query for one of the network protocol address and the associated identifier of said one process from another of the processes over the computer network at a connection server; and”**

107. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(d) in connection with claim 1, limitation “c.”

- e. “d. providing one of the network protocol address and the associated identifier of said one process to a said another process over the computer network, if the said one process is connected to the computer network, and**

to allow the establishment of a packet-based point-to-point communication between said one process and one of said another processes.”

- (1) The Microsoft Manual and NetBIOS disclose “providing one of the network protocol address and the associated identifier of said one process to a said another process over the computer network, if the said one process is connected to the computer network”

108. The Microsoft Manual and NetBIOS teach this limitation, at least in part, for the reasons set forth above in Section VI(A)(3)(d) in connection with claim 1, limitation “c.”

109. Furthermore, the Microsoft Manual teaches that the WINS/NetBIOS server keeps its mapping of names to IP addresses relatively current by tracking which users are still connected to the network. 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.) Similarly, 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.) In addition, the Microsoft Manual describes that, if a network device fails to re-register its name within the set renewal time, “the WINS server will mark the name as released and available for use.” (*Id.*; see Section V(A)(1) (“Processing Units Obtain Dynamically Assigned IP Addresses from

DHCP Servers”) and Section V(A)(2) (“Processing Units Register Their IP Addresses and Identifiers with the WINS Server”).)

110. Similarly, NetBIOS teaches that “[n]ames 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 [e.g., a computer] fails to send a name refresh message to the NBNS before the lifetime expires.” (Ex. 1004 at 396; *id.* (“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. If the node defends the name then the node is allowed to continue possession. Otherwise it is assumed that the node has released the name.”); *id.* at 400-01.)

111. NetBIOS further specifies that if the queried name (contained in a “NAME QUERY REQUEST”) is found in the server directory (*i.e.*, if the queried node is “online”), the NBNS sends a “POSITIVE NAME QUERY RESPONSE” to the calling node. (Ex. 1004 at 395, 407-08, 413, 449, 459.) The “POSITIVE NAME QUERY RESPONSE” includes an “ADDR_ENTRY record ‘represent[ing] an owner of a name,’ and each ADDR_ENTRY record includes “NB_ADDRESS,” *i.e.*, the IP address corresponding to the name of the called computer. (*Id.*)

112. On the other hand, if the queried name is not found in the directory (*i.e.*, if the called node is “offline”), NetBIOS teaches that the NBNS returns a “NEGATIVE NAME QUERY RESPONSE” to the calling node. (Ex. 1004 at 395,

407, 413, 460.) The “NEGATIVE NAME QUERY RESPONSE” contains a “Name Error” signifying that “[t]he name requested does not exist” (*i.e.*, the device is not currently connected to the computer network). (Ex. 1004 at 460; *see* Section V(B)(1) (“Processing Units Have Assigned IP Addresses”) and Section V(B)(2) (“Processing Units Register Their IP Addresses and Identifiers with the NBNS”).)

113. Accordingly, the prior art teaches providing one of the network protocol address (e.g., the IP address) and the associated identifier (e.g., the name) of said one process (e.g., the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on a first computer) to a said another process (e.g., the Windows NT 3.5 or Windows for Workgroups 3.11 operating system running on a second computer) over the computer network (e.g., a TCP/IP network), if the said one process is connected to the computer network (e.g., is registered as online in the WINS/NetBIOS server, the WINS/NBNS server sends a POSITIVE NAME QUERY RESPONSE (with the first IP address) to the second process; and if the first process is not registered as online in the WINS/NetBIOS server, the WINS/NBNS server sends a NEGATIVE NAME QUERY RESPONSE to the second process).)

- (2) The Microsoft Manual and NetBIOS disclose “allow[ing] the establishment of a packet-based point-to-point communication between said one process and one of said another processes”

114. The Microsoft Manual and NetBIOS teach this limitation for the reasons set forth above in Section VI(A)(3)(d) in connection with claim 1, limitation “c.”

115. Therefore, based on my analysis and applying the legal standard summarized above, it is my opinion that claim 3 would have been in obvious to one of ordinary skill in the art in light of the Microsoft Manual and NetBIOS.

B. Claim Chart with Examples Showing Where Prior Art Teaches Claimed Features

<u>Claim Limitation</u>	<u>Prior Art</u>
1-P. “A computer program product for use with a server operatively coupled over a computer network to a plurality of processes, the computer program product comprising a computer usable medium having program code embodied thereon the program code comprising:”	Ex. 1003 at 12, 18-19, 23-24, 27-29, 55-56, 105, 118-19, 135; Ex. 1004 at 356, 359, 368, 373-74, 377, 384-85, 389, 442.
1-A. “a. program code configured to receive the current network protocol address of one of the processes coupled to the network, the network protocol address being received by said one of the processes from an Internet access server;”	Ex. 1003 at 13, 20-23, 40-41, 57-58, 62-67, 69, 72, 74-75, 80, 83-115, 118-19, 121, 151, 262, 265-66; Ex. 1004 at 365, 378, 394-96, 398, 400-03, 412-13, 416, 430-32, 448-50.
1-B. “b. program code configured to receive an identifier associated with said one process; and”	Ex. 1003 at 41, 65, 74, 121. <i>See also</i> 1-P, 1-A.
1-C. “c. program code configured to receive queries for one of the network protocol address and the	Ex. 1003 at 13, 20-22, 40-41, 54-55, 64-67, 69-70, 72-75,

<u>Claim Limitation</u>	<u>Prior Art</u>
associated identifier of said one of the processes from other processes over the computer network at the server, and to allow the establishment of a packet-based point-to-point communication between said one of the processes and one of said other processes.”	80, 121-22, 151, 262, 265-266; Ex. 1004 at 368, 374, 378, 383, 395-96, 400-03, 406-07, 412-13, 415-16, 422, 430-32, 438, 447, 458-59.
2-P. “A computer data signal embodied in a carrier wave comprising:”	Ex. 1003 at 135.
2-A. “a. program code configured to receive a current network protocol address of a process coupled to a computer network, the network protocol address being received by the process from an Internet access server;”	<i>See</i> 1-A.
2-B. “b. program code configured to receive an identifier associated with said one process; and”	<i>See</i> 1-B, 1-C.
2-C. “c. program code configured to receive queries for one of the network protocol address and the associated identifier of said one process from other processes over the computer network at a connection server, and to allow the establishment of a packet-based point-to-point communication between said one process and one of said other processes.”	<i>See</i> 1-C.
3-P. “In a computer system operatively coupled over a computer network to a plurality of processes, a method comprising the steps of:”	Ex. 1003 at 12, 73; Ex. 1004 at 374-75, 389. <i>See also</i> 1-P.
3-A. “a. receiving the current network protocol address of a process coupled to the network, the network protocol address being received by the process from an Internet access server;”	<i>See</i> 1-A.
3-B. “b. receiving an identifier associated with said one process;”	<i>See</i> 1-B.

<u>Claim Limitation</u>	<u>Prior Art</u>
3-C. “c. receiving a query for one of the network protocol address and the associated identifier of said one process from another of the processes over the computer network at a connection server; and”	<i>See</i> 1-C.
3-D. “d. providing one of the network protocol address and the associated identifier of said one process to a said another process over the computer network, if the said one process is connected to the computer network, and to allow the establishment of a packet-based point-to-point communication between said one process and one of said another processes.”	Ex. 1003 at 13, 23, 57-58, 62-64, 69, 75, 79, 83-115, 118-19; Ex. 1004 at 365, 394-96, 398, 400-01, 407-08, 413, 449, 459-60. <i>See also</i> 1-C.

VII. CONCLUSION

116. 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.

117. I declare under penalty of perjury that the foregoing is true and correct.

Executed on July 31, 2014 in Honolulu, Hawaii.

A handwritten signature in blue ink, appearing to read "Bruce M. Maggs", written in a cursive style.

Bruce M. Maggs, Ph.D.