

41. If a process communicates using the TCP or UDP protocol (a higher level protocol running on top of the IP protocol), each process communication channel (usually called a ‘socket’) is assigned a ‘port’ (a 16-bit number) that uniquely identifies that channel, and thus, the ‘process’ within a given ‘processing unit’. For a process communicating using the TCP or UDP protocol, the network protocol address consists of its processing unit IP address and the port number assigned to the process communication socket. 12.34.56.78:1111 is an example of such an address, where 12.34.56.78 is the processing unit IP address, and 1111 is the process port number.

42. Some TCP or UDP communication applications use fixed default port numbers. For example, Web (HTTP) servers use the TCP port 80 by default, and POP3 mail servers use the TCP port 110 by default. As a result, in order to instruct a Web browser to connect to a Web server running on a certain computer (‘processing unit’), one can specify just the IP address of that computer: if the port number is not specified, the Web browser will connect to the port number 80 by default. The Web URL <http://12.34.56.78> is the same as the <http://12.34.56.78:80> URL.

43. When an application has a default port number specified, a name server may contain a mapping of an application process name into the IP address of the processing unit (as opposed to a mapping into the full process address including the process port number). The name www.company.com can be mapped to the IP address 12.34.56.78, and when this name is used in the <http://www.company.com> URL, the Web browser will add the port 80 to the IP address 12.34.56.78.

44. If all processes registered with a name server use the same default port number, the name servers may store just the IP address of the registered processes (i.e. the network address of the ‘processing unit’). The preferred embodiment of the ‘704 Patent, the WebPhone™ product (‘704 Patent Col. 3, lines 6-10) used the fixed TCP port 21845. The ‘connection server’ of the WebPhone™ service could register either the full process network address (12.34.56.77:21845), or just the computer IP address (12.34.56.78), relying on the fact that the port number (21845) is fixed and it is known to the caller process (some other WebPhone™ application instance). It is likely that this was the cause of the terminology confusion mentioned above.

45. For NetBIOS-over-TCP, the process network address consists of the ‘processing unit’ IP address and the ‘process’ registered name (this can be shown as *somename AT 12.34.56.78*). When a datagram is sent to a NetBIOS ‘process’, it is delivered over the network to its ‘processing unit’ IP address, to the fixed UDP port number 138. The sent data packet contains the target ‘process’ name and the datagram itself. The NetBIOS-over-TCP software running on the target ‘processing unit’ receives all packets delivered to the port 138. It uses the supplied process name to deliver the datagram to the target ‘process’.

46. There can be confusion about the contents of NetBIOS Name Server records: do they contain only IP addresses of ‘processing units’, or do they contain full network addresses of the registered ‘processes’? In reality, they contain both. A record mapping the process “Jim Files” to the IP address 1.1.1.1 contains a mapping from a process name to its processing unit address. But because the process full network address is “Jim Files” AT 1.1.1.1, the same record also contains a mapping from a process name to the process network address: in NetBIOS-over-TCP, the process name itself is a part of the process network address.

47. **Conclusion.** The Patent Owner's Expert has made the following statements: "NetBIOS merely tracks the registration of a computer or node, and does not store any information relating to processes running on that computer or node" (Exhibit 2018, p.26 and p.34), and "WINS only tracks registration of a computer, and does not store information in its database relating to processes running on that computer" (Exhibit 2018, p.46), and "Lastly, the use of "PC1" and "PC2" further demonstrates that WINS merely coordinates queries regarding computers, not processes running on those computers" (Exhibit 2018, p.48), and "It [WINS] also does not store information regarding a process running on the computer" (Exhibit 2018, p.49), and "This "process," as discussed above, is a process or program running on a computer, not merely the associated computer itself. WINS only discloses the registration of a computer" (Exhibit 2018, p.55). All these statements are incorrect, as NetBIOS Name Servers (including WINS) contain registrations for process names. While many names may be mapped to the same IP address (many registered processes running on the same computer), each mapping in a NetBIOS Name Server includes the full process network address, as the registered name is a part of that address. I present an example of that mapping in Section 75.

IX. Dynamic Process Addresses: Not a limitation.

48. **Claim.** The Patent Owner and the Patent Owner's Expert stated repeatedly that a requirement for a process to receive (get assigned) a network protocol address "following connection to the network" is an important limitation of the '704 Patent Claims. "In the current inter partes review, the Board stated that "the limitation 'connection to the computer network' was given the narrow construction to require a dynamic element" in the reexamination proceedings, and then determined that the claims did not require dynamic addressing (Decision at 9-10). I do not agree with this statement and decision, for the dynamic element is incorporated into the '704 claims via the language "following connection to the computer network." (See Ex. 2005, Notice of Intent to Issue Ex Parte Reexamination Certificate at 2-3 (emphasis added)). The fact that a network protocol address is assigned to a process upon its connection to the network is, by definition, dynamic address allocation." (Mayer-Patel Declaration, Exhibit 2018, p.18).

49. As noted in Section VIII of this Declaration, the '704 Patent uses only the term 'process' in its Claims. The Patent Owner's Expert even tried to show that the '704 Patent Description is also talking about the 'process' (though it does not use

that term): “20. The specification also confirms that the claimed process is a computer program, rather than the computer itself: "The first processing unit 12 may operate the disclosed point-to-point Internet protocol by a computer program described hereinbelow in conjunction with FIG. 6, which may be implemented from compiled and/or interpreted source code in the C++ programming language and which may be downloaded to the first processing unit 12 from an external computer. The operating computer program may be stored in the memory 16, which may include about 8 MB RAM and/or a hard or fixed drive having about 8MB." ('704 Patent at 3:39-47 (emphasis added)). This disclosure corresponds with the common understanding of the term "process." To one of ordinary skill in this art, the term "process" as used in the claims conveys a process or computer program running on a computer, rather than the computer itself.” (Mayer-Patel Declaration, Exhibit 2018, p.10-11). I agree with the Patent Owner’s Expert that the ‘704 Patent Claims are using only the term ‘process’. The ’704 Patent Claims talk about the “process network protocol addresses,” and not about “process unit network protocol addresses” or “computer network protocol addresses.” The difference was explained in Section VIII of this Declaration.

50. The '704 Patent Claim 1 uses the phrase “a network protocol address received by the first process following connection to the computer network,” while the Claims 2,4, and 32 specify a process address received “following connection” to the network, and the Claims 33 and 38 specify a process address received “upon connection to the computer network.”

51. I agree with the Patent Owner’s Expert that a process is a running instance of an application, which is created anew every time when an application starts: “Q.If a program is started again, is a new process created? A. Generally that is true, yes.” (Mayer-Patel Deposition, Exhibit 1022, p.16:5-7). Such a dynamic entity gets its network address dynamically, too. Generically speaking, a process gets its network address when it “connects to the network”, i.e. when a process uses the Operating System controlling its ‘processing unit’ to prepare a network communication channel the process can use. The Patent Owner’s Expert provided an even more restricted definition: “That process would receive a network protocol address when it uses the operating system in order to make a connection to some other process on some other computer.” (Mayer-Patel Deposition, Exhibit 1022, p.22:15-19).

52. As the Patent Owner's Expert has confirmed, there is no way for a process to receive a "network protocol address" other than "upon/following connection to the computer network." All process network protocol addresses are assigned dynamically, after the process (a running instance of an application) has started and when it uses the operating system to "connect to the network." This is true for any process, whatever communication system or interface (NetBIOS, 'sockets', or any other) that process is using.

53. **Conclusion.** I cannot agree with the Patent Owner that "process receiving a network protocol address following a connection to the computer network" is a limitation not disclosed in NetBIOS, or that it is a limitation at all. I similarly cannot agree with the Patent Owner that "processes having dynamically assigned network protocol addresses" ('704 Patent, Claim 33) is a limitation not disclosed in NetBIOS, or that it is a limitation at all.

X. Dynamic Addressing of Processing Units

54. The Patent Owner and the Patent Owner's Expert pointed to the Description part of the '704 Patent which uses the term 'processing unit', as opposed

to the term 'process' used in the '704 Patent Claims. They refer to the Description sections which talk about dynamic assignment of network addresses to processing units: "27. The specification also explicitly incorporates dynamic addressing into the claimed system. The Background of the patent explains a problem in the prior art as the following: Permanent IP addresses [i.e., 'static' addresses] of users and devices accessing the Internet readily support point-to-point communications of voice and video signals over the Internet... Due to the dynamic nature of temporary IP addresses of some devices accessing the Internet, point-to-point communications in realtime of voice and video have been generally difficult to attain. ('704 Patent at 1:48-56). 28. The descriptions of the patented invention also explicitly describe dynamic, rather than static, addressing: "When either of the processing units logs on to the Internet via a dial-up connection, the respective unit is provided a dynamically allocated IP address by a connection service provider." ('704 Patent at 5:21-23). The figures of the patent also specify that the system utilizes dynamic addressing: "As shown in FIG. 1, the disclosed point-to-point Internet protocol and system operate when a callee processing unit does not have a fixed or predetermined IP address." ('704 Patent at 5:15-17). The Patent lastly provides the following conclusion: Realtime point-to-point communication of audio signals over the Internet, as well as video and

voicemail, may thus be established and supported without requiring permanent IP addresses to be assigned to either of the users or processing units. For the duration of the realtime point-to-point link, the relative permanence of the current IP addresses of the processing units is sufficient. ('704 Patent at 7:32-38).” (Mayer-Patel Declaration, Exhibit 2018, pp.19-20).

55. As noted in Section VIII of this Declaration, the term ‘processing unit’ is not used in any of the ‘704 Patent Claims, and in my opinion, the ‘processing unit network addresses’ are not relevant to the ‘704 Patent Claims, which discuss only the ‘process network protocol addresses’. Nonetheless, I have decided to review the Patent Owner’s and the Patent Owner Expert’s statements about dynamic address assignment to processing units being incompatible with the NetBIOS references.

56. **The ‘Dynamically Assigned IP address’ Terminology.** The NetBIOS was designed in early 1980s, and the NetBIOS-over-TCP standards (RFC1001/1002) were published in March 1987 (Exhibit 1003). At that time automated mechanisms to assign IP addresses to ‘processing units’ (devices and computers) had already emerged. The BOOTP protocol standardized in RFC951 was published in September

1985. At that time the term ‘dynamically assigned’ was not in use yet: while BOOTP implements what we now call ‘dynamically assigned IP addresses’ the word ‘dynamically’ is not used anywhere in the RFC951 document. Only when the BOOTP protocol eventually had evolved into the DHCP protocol (standardized in RFC1531 in October 1993), did the term ‘dynamically configured hosts’ begin to be introduced. Abbreviated (and technically incorrect¹) notion of ‘dynamically assigned IP address’ appeared even later than that. It is not surprising that the ‘dynamically assigned IP address’ notion does not appear in the RFC1001/RFC1002 documents published six years earlier, in March 1987.

57. **No limitation to static IP addresses.** The NetBIOS-over-TCP was designed with a clear vision that in its target environment (office networks) computer network addresses were not assigned permanently. The RFC1001/RFC1002 standard does not contain anything that would limit its usage to

¹ It is incorrect because dynamic network configuration requires assignment not only of a host IP address, but, as a minimum, of the IP network ‘mask’, and of the ‘default gateway’ IP address.

systems with permanently or ‘statically’ assigned IP addresses. This standard was designed to support ‘dynamically assigned addresses’, though these documents do not use this term because it was coined years later.

58. Clear Reference to the Dynamic nature of IP addresses. The Domain Name System (DNS) was standardized in the RFC882/883 documents published in November 1983. By the time the RFC1001/1002 documents were published in March 1987, DNS was already being used on the Internet. It was also used for large internal networks (universities, large corporations, etc). The Domain Name System was designed for environments where computers had permanently assigned IP addresses, and it was based on ‘static’ (manually modified) databases containing (among other things) mappings from names to IP addresses. The RFC1001 document described a NetBIOS Name Server (NBNS) that should contain similar mapping (from a process name to the processing unit IP address), and that document provided comparison between its NBNS and the already popular Domain Name System (DNS). When NetBIOS-over-TCP standards were developed, it was

recognized that these name-to-address mappings should be dynamic (i.e. that the IP address can change) and this is why the traditional DNS could not be used². The RFC1001 document reads: “This RFC, however, requires the NBNS to provide services beyond those provided by the current domain name system. An attempt has been made to coalesce all the additional services which are required into a set of transactions which follow domain name system styles of interaction and packet formats. Among the areas in which the domain name service must be extended before it may be used as an NBNS are:

- Dynamic addition of entries

- Dynamic update of entry data” (Exhibit 1003, p.386). As an entry data in NetBIOS Name server database contains a ‘process’ name and its ‘processing unit’ IP address, a requirement to support “dynamic update of entry data” indicates that the RFC1001 document anticipated the ‘dynamic’ nature of IP address allocation: record data for a given name contains only an IP address, and record data needs to be updated only if that IP address can change.

² The Domain Name System was later extended to support dynamic addressing, too. The resulting Dynamic DNS (DDNS) standards appeared in July of 1994.

59. The '704 Patent does not contain any mechanism specifically designed to support dynamic addressing, nor does it disclose any mechanism for dynamic allocation of IP addresses. The '704 'processes' (or 'processing units') connect to the network, register their current network addresses with the server, and later disconnect from the network. When they connect again, they may be using the same network address, or they may have a different network address assigned – the 'connection server' of the '704 patent simply registers the network address supplied. Similarly, a NetBIOS Name Server works equally well if a name was registered being mapped to one IP address, and later this name is re-registered being mapped to the same or to a different IP address.

60. The Microsoft WINS is an implementation of NetBIOS Name Server (NBNS). It was designed specifically to work in DHCP environments (i.e. in networks where IP addresses are assigned dynamically, using a DHCP server). WINS strictly follows the RFC1001/RFC1002 standards: "As described in the following section, WINS is a NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as p-node." (Exhibit 1004, p.65); "The 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.” (Exhibit 1004, p.69).

61. **Conclusion.** It is my opinion that not only the Microsoft WINS implementation, but the NetBIOS-over-TCP standards (RFC1001/1002) themselves fully anticipate configurations where IP addresses of ‘processing units’ are assigned on a non-permanent basis (‘dynamically’).

XI. Rebuttal to Additional Characterizations of NetBIOS

62. While most of the NetBIOS-related statements in the Expert Declaration (Mayer-Patel Declaration, Exhibit 2018) are repetitions of the same two statements (“on-line status vs. registration” and “dynamic address assignment”), there are several other statements that, in my opinion, should be reviewed and discussed.

63. The Patent Owner’s Expert claims “If a system is configured to utilize dynamic address allocation, each process is assigned a unique IP address from the server during network initialization.” (Mayer-Patel Declaration, Exhibit 2018, p.16). This statement is incorrect, as an IP address is assigned (permanently or dynamically) to an entire computer (‘processing unit’), and processes running on that computer unit share the

same IP address as the ‘computer part’ of their process network addresses. This is why a process network address should include (explicitly or implicitly) a ‘local part’ uniquely identifying that ‘process’ within the ‘processing unit’. It can be a ‘port number’ for processes using the TCP/UDP protocols, or it can be the registered name itself for a process using the NetBIOS-over-TCP protocol.

64. The Patent Owner’s Expert makes the following statement about NetBIOS: “35. In fact, once a node is registered, its status does not change in any way unless it is "challenged" by a new node requesting the same name registration as the original node.” (Mayer-Patel Declaration, Exhibit 2018, p.22). This is incorrect, as the NetBIOS Name Server may detect the change of the process on-line status when the process requests its ‘name release’, when the process or its ‘processing unit’ fail to refresh the registration before it times-out. See Sections 27 and 29 of this Declaration for the details.

65. The Patent Owner’s Expert makes the following statement about the record set (database) kept by NetBIOS Name Server (NBNS): “36. In the previous reexamination of the ’704 Patent, I noted that NetBIOS sets all names to "active," regardless

of the name's on-line status with the network. An active name simply refers to a name that has been registered and that has not yet been deregistered, independent of whether the associated computer is or is not on-line. The fields in the name flag merely denote whether the name is (1) active, or (2) in conflict, as pictured below: [...] (NetBIOS at 447).” (Mayer-Patel Declaration, Exhibit 2018, p. 23). The quoted NetBIOS record (“pictured below”) is taken from RFC1002, the document describing the formats of NetBIOS-over-TCP protocol messages, and the quoted part describes “4.2.18. NODE STATUS RESPONSE”, sent by a ‘processing unit’ (not a particular ‘process’) in response to NetBIOS NODE STATUS REQUEST. Firstly, the NODE STATUS REQUEST does not correspond to any of the operations described in the ‘704 Patent (it asks a ‘processing unit’ to report all registered names of all ‘processes’ running on that unit), although it can be used to implement more effective ‘active polling’ (the ‘704 Patent does not disclose ‘active polling’ at all, see Sections 31 and 35 of this Declaration). Secondly, the quoted NetBIOS record is part of the protocol, not a record of the NetBIOS Name Server internal database as the Patent Owner’s Expert implies, and the NODE STATUS RESPONSE record format does not define or restrict the content of that database.

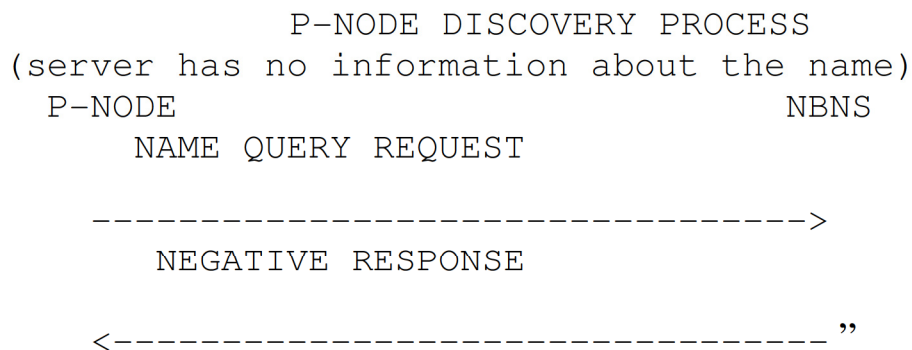
66. The Patent Owner's Expert makes the following statement: "I note that the NetBIOS system does not actually deregister any names. Instead, NetBIOS will allow the user of the computer to delete its own name from the NetBIOS database. According to NetBIOS, "The only valid user function against a marked name is DELETE NAME." (NetBIOS at 380)." (Mayer-Patel Declaration, Exhibit 2018, p.24). This statement was taken completely out of context, which is RFC1001 section 15.1.3.5 NAME CONFLICT. That section explains what should happen to a 'process' trying to register a NetBIOS name that is already in use with some other 'process', and to the NetBIOS software running on the 'processing unit' where the first 'process' runs. The quoted section says: "Notice that only those nodes that receive the name conflict message place a conflict mark next to a name. Logically, a marked name does not exist on that node", and then "The only valid user function against a marked name is DELETE NAME." The '704 Patent does not even try to discuss situations in which two different processes try to utilize the same name, so any discussion about duplicate names handling in NetBIOS is not relevant.

67. The Patent Owner's Expert later makes a similar statement: "39 Moreover, if the system does become aware that the node is not connected to the network (through a

name challenge, for example), the node will still remain "registered" with the system, merely marked as "in conflict." (Mayer-Patel Declaration, Exhibit 2018, p.24-25). Again, marking a NetBIOS record 'in conflict' takes place only when two processes try to use the same name – the condition completely ignored by the '704 Patent. Outside that condition, the NetBIOS Name Server does de-register records if their processes are no longer 'on-line', including the situations of active 'name challenge' polling: "A well-behaved NBNS, would, however, issue a challenge to-, or request a list of names from-, the non-reporting end-node before deleting its name(s). The absence of a response, or of the name in a response, will confirm the NBNS decision to delete a name." (Exhibit 1003, pp.400-401).

68. The Patent Owner's Expert claims that not only is NetBIOS Name Server unable to detect that a process is no longer 'on-line' (this incorrect statement was examined in Section VII above), but that "NetBIOS clearly delineates its protocol for when a computer or node is thought to be disconnected from the network, and it does not include the generation or transmission of an off-line message to the first process" (Mayer-Patel Declaration, Exhibit 2018, p.36). This is an incorrect statement. When a process tries to connect to other process by name, and the NetBIOS-over-TCP

software sends a NAME QUERY request to the NetBIOS Name Server, in a situation in which the Name Server does not have the requested record in its database, it generates and sends back a NEGATIVE RESPONSE message, in exactly the same way as the '704 Patent 'connection server' sends back its 'off-line message'. The RFC1001 document reads: "The following diagram shows what happens if the NBNS has no information about the name:



(Exhibit 1003, p.389). The RFC1002 section 4.2.14. NEGATIVE NAME QUERY RESPONSE specifies this 'off-line message' format (Exhibit 1003, p.460).

69. The Patent Owner's Expert then makes one more attempt to show that the NetBIOS-over-TCP does not teach an equivalent to 'off-line message' reporting of the '704 Patent (Sections 55-56 of the Expert Declaration). I agree with the Patent Owner's Expert when he says "I note that the "END-NODE CHALLENGE

REGISTRATION RESPONSE" transmitted by NBNS could not be interpreted as an "off-line message" generated and transmitted "to the first process"." (Mayer-Patel Declaration, Exhibit 2018, p.37), because he reviews a wrong RESPONSE. Here the Patent Owner's Expert reviews the process of NetBIOS registration and various messages the Name Server (NBNS) generates when it discovers that the name to be registered is already claimed by a different process. Firstly, the '704 Patent does not even try to disclose what should be done if two processes claim the same name. Secondly, the 'off-line message' as claimed in the '704 Patent Claim 7 is generated in response to "receiving a query from the first process to determine the on-line status of the second process" (parent Claim 4, B), i.e. in response to a NAME QUERY request (in NetBIOS-over-TCP terms), not in response to a NAME REGISTRATION request (in NetBIOS-over-TCP terms), claimed in the parent Claim 4, A. And, as shown in Section 68 of this Declaration, the NetBIOS-over-TCP Name Server does generate an 'off-line message' in response to a NAME QUERY request, in the form of NEGATIVE NAME QUERY RESPONSE.

70. The Patent Owner's Expert claims that "61. Further, NetBIOS does not delete an entry from its database. Claim 42 contains a step to "delete an entry from the

compilation”.” The Expert then immediately contradicts himself, saying that NetBIOS does delete names, but does it “incorrectly”: “Second, for a name to be deleted, it must be deleted by the registrant, not by the NetBIOS system. Petitioner cites the NetBIOS excerpts that "NetBIOS names may be released explicitly or silently by an end-node," and "P nodes send a notification to their NBNS." (Petition at 55). I note that both of these excerpts establish that the registrant deletes itself, as names are released "by an end-node," and the "P nodes send a notification" requesting the deletion of its name.” (Mayer-Patel Declaration, Exhibit 2018, p.42). Firstly, it is not true that a NetBIOS Name Server deletes names only in response to a registrant process request (in response to its NAME RELEASE request). When a NetBIOS Name Server discovers that a process is not ‘on-line’ any more, it deletes its name: “The absence of a response, or of the name in a response, will confirm the NBNS decision to delete a name.” (Exhibit 1003, pp.400-401). Secondly, the NAME RELEASE request sent by the registrant process perfectly matches the ‘704 Patent Claim 42. That process does not delete its registration completely by “itself”, as the Patent Owner’s Expert states: a ‘process’ running on a different ‘processing unit’ does not have direct access to the Name Server ‘processing unit’ and its database. Instead, a ‘process’ that desires to remove

its name (to “de-register”) uses the NetBIOS software to send the NAME RELEASE request to the NetBIOS Name Server. And the NetBIOS Name Server then deletes the process name from its database of active names (“releases the name”) and it sends back the POSITIVE NAME RELEASE RESPONSE. The RFC1002 document shows how a NetBIOS Name Server should process NAME RELEASE requests:

```
NAME RELEASE REQUEST:
/*
 * secure server may choose to disallow
 * a node from deleting a name
 */
update data base - remove entry;
send POSITIVE NAME RELEASE RESPONSE;
return;
```

(Exhibit 1003, p.502). Receiving a NAME RELEASE REQUEST is an example of “a predetermined event” specified in the ‘704 Patent Claim 42.

XII. Rebuttal to Additional Characterizations of Microsoft WINS

71. The Microsoft WINS product is an implementation of NetBIOS Name Server (NBNS). In this Section I address several statements the Patent Owner’s Expert has made about the Microsoft WINS product.

72. Microsoft WINS fully follows the RFC1001/RFC1002 standards: “As described in the following section, WINS is a NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as p-node.” (Exhibit 1004, p.65); “The 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.” (Exhibit 1004, p.69).

73. The Microsoft TCP/IP manual (Exhibit 1004) is a document designed to help system administrators to understand concepts and ideas behind a NetBIOS Name Server (WINS), dynamic IP address allocation (DHCP), FTP, SNMP, and many other networking protocols and methods. This document can be used to illustrate some features of those protocols and methods. On the other hand, the Microsoft TCP/IP manual is not a formal programming guide, nor is it a detailed description of product internals. This document does not define everything a particular protocol or product can do: an absence of a technical detail in such a manual does not mean that this detail is not implemented in the product.

74. The Patent Owner's Expert claims: "65. Registration (or the "claim[ing of] the particular IP address") occurs during system startup. As the WINS reference states, "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." (WINS at 49). "A computer typically registers itself when it starts." (WINS at 50)." (Mayer-Patel Declaration, Exhibit 2018, p.44). Microsoft Windows NT allows a user to assign a 'name' to the computer. When Windows NT starts, several system processes are started automatically. Among them the Workstation Service and, optionally, the Messenger Service. These processes are registering themselves with a NetBIOS Name Server (such as WINS) using the computer name (up to 15 characters) and a 1-byte suffix. The Messenger Service process also registers itself using the computer user name. While technically these are processes (started automatically), for an average user or system administrator (the target audience of the Microsoft TCP/IP manual) it looks like the "computer itself" has registered these names. On the same page, in the next section of his Declaration, the Patent Owner's Expert confirms that these services are just processes that start and stop: "When a computer finishes with a particular name