

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Liberty Mutual Insurance Company, Liberty Mutual Technology Group, Inc., Liberty Mutual Holding Company Inc., Liberty Mutual Group Inc., Liberty Mutual Plano LLC, Comparion Insurance Agency, LLC, Ironshore Holdings (U.S.) Inc.,

and

Comerica Incorporated

Petitioners

v.

Intellectual Ventures I LLC

Patent Owner

IPR2025-00201
U.S. Patent No. 7,949,785

**DECLARATION OF EREZ ZADOK IN SUPPORT OF PETITION FOR *INTER PARTES*
REVIEW OF
U.S. PATENT NO. 7,949,785**

Mail Stop Patent Board
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Table of Contents

I.	ASSIGNMENT.....	1
II.	BACKGROUND AND QUALIFICATIONS	1
III.	MATERIALS CONSIDERED	9
IV.	LEVEL OF ORDINARY SKILL IN THE ART	11
V.	RELEVANT LEGAL STANDARDS	12
VI.	SUMMARY OF OPINIONS	15
VII.	TECHNOLOGY OVERVIEW.....	15
A.	General Computer Operations	15
B.	Networking	17
1.	Networking Overview.....	17
2.	Network Protocols	20
3.	The Domain Name System, Hostnames, and IP Addresses.....	31
C.	Virtualization	38
VIII.	THE '844 PATENT	41
A.	Overview.....	41
B.	Prosecution History.....	47
IX.	THE CLAIMS.....	49
X.	OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES	53
A.	Mehta	53
B.	RFC-1383.....	54
C.	Johnson	Error! Bookmark not defined.
XI.	CLAIM CONSTRUCTION.....	57

XII.	GROUND 1: Mehta and RFC-1383 Renders Obvious Claims 1-3, 6-12, 22-69, 71-80, 82-86, and 89-90	63
A.	The Independent Claims	63
1.	Preamble: Virtual network system or manager.....	64
2.	“A virtual network defined by a domain name”	65
3.	Registration/distribution of virtual network address to each device which uniquely identifies the device	66
4.	Route director/routing.....	69
a.	The Encapsulated Packets in Claims [38.3], [48.3], and [62.4]....	70
5.	DNS server, request and responses	71
6.	A memory and a processor generally.....	81
B.	Motivation To Combine Mehta and RFC-1383	83
C.	The Dependent Claims.....	89
1.	Claims 2-3, 6-12, 22-29 are also obvious over the combination of Mehta and RFC-1383	89
a.	Claim 2: The system of claim 1 wherein each of the devices communicate with the virtual network manager and the other devices in the virtual network via an associated agent	90
b.	Claim 3: The system of claim 2 wherein the associated agent is installed on a proxy device which is a proxy agent for one or more of the devices in the virtual network with respect to the virtual network manager and the route director Error! Bookmark not defined.	
c.	Claim 6: The system of claim 1 wherein each of the devices include an agent configured to communicate with the virtual network manager and agents of the other devices in the virtual network	90
d.	Claim 7: The system of claim 1 wherein the virtual network manager is coupled to the public network and has an associated public network address	90
e.	Claim 8: The system of claim 1 wherein the route director is further configured to receive the encapsulated packets when routed to a public network address corresponding to the route director ..	91

f.	Claim 9: The system of claim 1 wherein each of the devices in the virtual network are further identified by at least one physical network address	91
g.	Claim 10: The system of claim 9 wherein the physical network addresses that are associated with the devices in the virtual network are dynamic	91
h.	Claim 11: The system of claim 9 wherein the physical network addresses that are associated with the devices in the virtual network are static	91
i.	Claim 12: The system of claim 9 wherein the physical network addresses that are associated with the devices in the virtual network are private network addresses, and wherein the virtual network addresses are unique in the virtual network so as not to conflict with the private network addresses.....	92
j.	Claim 22: The system of claim 1 wherein the virtual network manager is coupled to the public network and includes a public network address	93
k.	Claim 23: The system of claim 22 wherein the route director is coupled to the public network and includes a different public network address than the public network address.....	Error! Bookmark not defined.
l.	Claim 24: The system of claim 22 wherein the route director is coupled to a private network and includes a private network address.....	Error! Bookmark not defined.
m.	Claim 25: The system of claim 1 wherein the encapsulated packets further include virtual IP packets configured for communication of the data, the virtual IP packets being encrypted prior to being encapsulated.....	93
n.	Claim 26: The system of claim 1 wherein the encapsulated packets further include virtual IP packets configured for communication of the data, the virtual IP packets being encrypted prior to being encapsulated.....	93
o.	Claim 27: The system of claim 25 wherein the virtual IP packets are encrypted using DES or triple DES	Error! Bookmark not defined.
p.	Claim 28: The system of claim 1 wherein the virtual network manager includes a virtual community definition that is defined by	

	the domain name, and includes one or more of the devices that are registered in the virtual network	94
q.	Claim 29: The system of claim 1 wherein the route director is a public route director in the public network, the system further comprising a private route director in a private network configured to enable access to devices of the private network that are also in the virtual network from devices of the public network that are also in the virtual network	94
2.	Claims that Depend on Independent Claim 30	94
a.	Claim 31: The virtual network manager of claim 30 further comprising an additional network interface configured for data communication via a public network.....	94
b.	Claim 32: The virtual network manager of claim 31 wherein the network interface is a UDP port configured for data communication via the virtual network, and wherein the additional network interface is a TCP port configured for registration communications via the public network	95
c.	Claim 33: The virtual network manager of claim 30 wherein the network interface is a UDP port configured for data communication via the virtual network	95
d.	Claim 34: The virtual network manager of claim 30 wherein the register module is further configured to receive the registration request from the agent that is installed on the device for data communication via the virtual network	96
e.	Claim 35: The virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network	96
f.	Claim 36: The virtual network manager of claim 35 wherein the join module is further configured to provide virtual network addresses to the devices that are registered in the virtual network.....	97
g.	Claim 37: The virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network	97

3.	Similar or Identical Claims that Depend on Independent Claims 48 and 62	98
a.	Claim 49: The method of claim 48 further comprising providing the devices in the virtual network with a communication agent...	98
b.	Claim 63: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to provide the devices in the virtual network with a communication agent.....	98
c.	Claim 50: The method of claim 49 wherein said providing the devices in the virtual network with a communication agent includes providing a proxy agent... Error! Bookmark not defined.	
d.	Claim 64: One or more processor readable storage media devices as recited in claim 63 wherein the virtual network manager provides the as a proxy agent..... Error! Bookmark not defined.	
e.	Claim 52: The method of claim 48 further comprising defining a member set of the virtual network that includes one or more of the devices, and assigning the domain name that defines the virtual network	99
f.	Claim 65: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to define a member set of the virtual network that includes one or more of the devices, and the domain name that defines the virtual network	99
g.	Claim 54: The method of claim 48 further comprising assigning the virtual network address to the device as an IPV4 compliant address.....	99
h.	Claim 66: One or more processor readable storage media devices as recited in claim 62 wherein the virtual network address includes a non-routable IPV4 compliant address	99
i.	Claim 56: The method of claim 48 further comprising routing network traffic via the public network from a first device having a public network address to a second device having a different public network address, the network traffic being routed as data packets having a source address which is the virtual network address of the first device and a destination address which is the virtual network address of the second device, the data packets being encapsulated to include a second source address which is a public network	

address of the first device and a second destination address which is the different public network address of the second device **Error! Bookmark not defined.**

j. Claim 67: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to route network traffic via the public network from a first device having a public network address to a second device having a different public network address**Error! Bookmark not defined.**

k. Claim 57: The method of claim 48 further comprising routing network traffic from a first device in a private physical network having a private network address to a second device having a public network address, the network traffic being routed as data packets having a source address which is the virtual network address of the first device, a destination address which is the virtual network address of the second device, and a shim which includes the private network address, the data packets being encapsulated to include a second destination address which is the public network address of the second device..... 100

l. Claim 68: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to route network traffic from a first device in a private physical network having a private network address to a second device having a public network address**Error! Bookmark not defined.**

m. Claim 58: The method of claim 48 further comprising routing network traffic from a first device in a private physical network having a first private network address to a second device in a different private physical network having a second private network address, the network traffic being routed as encapsulated packets having a source address which is the virtual network address of the first device, a destination address which is the virtual network address of the second device, and a shim which includes the first and second private network addresses **Error! Bookmark not defined.**

n. Claim 69: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to route network traffic from a first device in a private physical network having a first private network address to a second device in a different private physical network having a second private network address.....**Error! Bookmark not defined.**

o.	Claim 59: The method of claim 58 wherein the first private network address and the second private network address are identical.....	Error! Bookmark not defined.
p.	Claim 71: One or more processor readable storage media devices as recited in claim 69 wherein the first private network address and the second private network address are identical	Error! Bookmark not defined.
q.	Claim 61: The method of claim 48 further comprising applying a group policy to the devices that are registered as the members of the virtual network	100
r.	Claim 74: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to apply a group policy to the devices that are registered as the members of the virtual network	100
4.	Additional Claims that Depend on Independent Claims 48 and 62.....	101
a.	Claim 51: The method of claim 48 further comprising defining at least two member sets of the virtual network that each include one or more of the devices, the at least two member sets having at least one different device.....	101
b.	Claim 53: The method of claim 48 further comprising defining at least two member sets of the virtual network that each include one or more of the devices, the at least two member sets having at least one different device.....	101
c.	Claim 55: The method of claim 54 wherein the IPV4 compliant address is non-routable	101
d.	Claim 60: The method of claim 48 further comprising: receiving a join status request from a first device in the virtual network as a query to determine the status of a second device in the virtual network; and responding to the join status request to indicate whether or not the second device is joined to the virtual network for data communication	101
e.	Claim 72: One or more processor readable storage media devices as recited in claim 67 wherein the network traffic is routed as encapsulated data packets	102
f.	Claim 73: One or more processor readable storage media devices as recited in claim 67 wherein the network traffic is routed as encrypted data traffic	102

5.	Claims that Depend on Independent Claims 75.....	102
a.	Claim 76: The system of claim 75 wherein the virtual network manager includes a member register module.....	102
b.	Claim 77: The system of claim 75 wherein the virtual network manager includes a member join module	102
c.	Claim 78: The system of claim 77 wherein the member join module provides a virtual network address to a device that is registered as a member of the network	103
d.	Claim 79: The system of claim 75 wherein the virtual network manager is further configured to maintain data on an association between at least one virtual network address with at least one device that is registered as a member of the network	103
e.	Claim 80: The system of claim 75 wherein the virtual network manager includes a DNS server for the virtual community network	103
f.	Claim 82: The system of claim 75 wherein the virtual network manager includes at least a second virtual community definition	103
g.	Claim 83: The system of claim 75 wherein the virtual network manager includes a member authenticator.....	103
h.	Claim 84: The system of claim 75 wherein the virtual network manager includes a DNS server configured to provide authoritative responses for DNS queries from devices in the virtual community	103
i.	Claim 85: The system of claim 75 further comprising at least one route director configured to communicate with the one or more devices in the user set	104
j.	Claim 86: The system of claim 75 wherein each device registered in the network is configured to communicate with the virtual network manager and other devices in the user set via at least one agent.....	104
k.	Claim 89: The system of claim 75 wherein communications between the one or more devices in the user set are encrypted ..	104
l.	Claim 90: The system of claim 89 wherein the virtual network manager is configured to provide a shared message to the one or	

more devices in the user set to establish encrypted communications
..... 105

XIII. GROUND 2: Mehta, Johnson, and RFC-1383 Render Obvious Claims 4-5, 13-21, 70, 81, and 87-88**Error! Bookmark not defined.**

A. Claims with NAT Limitations**Error! Bookmark not defined.**

1. Claim 4.0: The system of claim 2 wherein the virtual network manager is further configured to: receive a request from the agent associated with a device in a private network to register the device in the virtual network; and.....**Error! Bookmark not defined.**
2. Claim 4.1: The system of claim 2 wherein the virtual network manager is further configured to: receive a request from the agent associated with a device in a private network to register the device in the virtual network; and.....**Error! Bookmark not defined.**
3. Claim 4.2: the virtual network manager configured to] detect a presence of a NAT device via which the request is routed from the agent to the virtual network manager.....**Error! Bookmark not defined.**
4. Claim 5.0: The system of claim 4 wherein the virtual network manager is further configured to receive the request from the agent as a message, and**Error! Bookmark not defined.**
5. Claim 5.1: the virtual network manager configured to] detect the presence of the NAT device by comparing a field in a payload of the message with a source address in a header of the message **Error! Bookmark not defined.**
6. Claim 13.0: The system of claim 9 wherein the source device is coupled to a first private network and accesses a public network via a first NAT device configured to transmit the encapsulated packets, [and]..... **Error! Bookmark not defined.**
7. Claim 13.1: an encapsulated packet further including the physical network address of the source device as a private network address in the first private network**Error! Bookmark not defined.**
8. Claim 14: The system of claim 13 wherein the physical network address of the source device is dynamic**Error! Bookmark not defined.**
9. Claim 15: The system of claim 13 wherein the physical network address of the source device is static.....**Error! Bookmark not defined.**

10. Claim 16.0: The system of claim 13 wherein the destination device is coupled to a second private network and accesses the public network via a second NAT device, [and]**Error! Bookmark not defined.**
11. Claim 16.1: the encapsulated packet further including the physical network address of the destination device as a private network address in the second private network**Error! Bookmark not defined.**
12. Claim 17: The system of claim 16 wherein the physical network address of the destination device is dynamic**Error! Bookmark not defined.**
13. Claim 18: The system of claim 16 wherein the physical network address of the destination device is static.....**Error! Bookmark not defined.**
14. Claim 19: The system of claim 16 wherein said first private network and said second private network share at least one network address **Error! Bookmark not defined.**
15. Claim 20.0: The system of claim 1 wherein the route director is further configured to: provide a device-specific pseudo address assignment for the device in the virtual network when the device communicates with the route director via a NAT device; and.....**Error! Bookmark not defined.**
16. Claim 20.1: store an association of the device-specific pseudo address with a public network address of the NAT device and a port number of the NAT device**Error! Bookmark not defined.**
17. Claim 21: The system of claim 1 wherein the route director includes a translator for virtual network address information for a device in the virtual network that is implemented in a private network **Error! Bookmark not defined.**
18. Claim 70: One or more processor readable storage media devices as recited in claim 69 wherein the network traffic is routed to the first or second device via a NAT device.....**Error! Bookmark not defined.**
19. Claim 81: The system of claim 75 wherein the virtual network manager includes a NAT device detector for devices connecting with the virtual network manager behind a NAT device**Error! Bookmark not defined.**
20. Claim 87: The system of claim 75 wherein the user set includes at least a first device and a second device, and wherein at least one of said first device and said second device is coupled to a first private network and accesses a public network via a NAT device..... **Error! Bookmark not defined.**
21. Claim 88: The system of claim 87 wherein said first device is coupled to said first private network, and said second device is coupled to a second

private network and accesses the public network via a second NAT device
.....**Error! Bookmark not defined.**

B. Motivation to Combine Mehta, Johnson, and RFC-1383 **Error! Bookmark not defined.**

XIV. SECONDARY CONSIDERATIONS 106

XV. CONCLUSION..... 106

I, Erez Zadok, Ph.D., declare that:

I. ASSIGNMENT

1. I have been retained by Liberty Mutual Holding Company and Comerica Incorporated (together “Petitioners”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“PTO”).

2. My consulting company, Zadoks Consulting, LLC, is being compensated for my time at my standard consulting rate. I am also being reimbursed for expenses that I may incur during the course of this work.

3. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony, or the outcome of this or any other proceeding. I have no other interest in this proceeding.

4. I have been asked to consider whether certain references disclose or suggest the features recited in the claims of U.S. Patent No. 7,949,785 (“the ’785 patent”) (EX1001).¹ My opinions are set forth below.

II. BACKGROUND AND QUALIFICATIONS

5. I am a Professor in the Computer Science Department at Stony Brook University (part of the State University of New York (“SUNY”) system). I direct the File-systems and Storage Lab (FSL) at Stony Brook’s Computer Science Department. My research interests include file systems and storage systems, operating systems, transactional systems including database technologies, information technology and system administration, security/privacy and information

¹ Where appropriate, I refer to exhibits I understand will be attached to the petition for inter partes review of the ’785 patent (the “Petition”).

assurance, networking, energy efficiency, performance and benchmarking, virtualization, cloud systems, compilers, applied machine learning, and software engineering.

6. I studied at a professional high school in Israel, focusing on electrical engineering (“EE”), and graduated in 1982. I spent one more year at the high school’s college division, receiving a special Certified Technician’s degree in EE. I then went on to serve in the Israeli Defense Forces for three years (1983–1986). I received my Bachelor of Science degree in computer science (“CS”) in 1991, my Master’s degree in CS in 1994, and my PhD in CS in 2001—all from Columbia University in New York.

7. When I began my undergraduate studies at Columbia University, I also started working as a student assistant in the various campus-wide computer labs, eventually becoming an assistant to the head labs manager, who was managing all public computer labs on campus. During that time, I also became more involved with research within the CS Department at Columbia University, conducting research on operating systems, file and storage systems, distributed and networked systems, security, and other topics. I also assisted the CS department’s computer administrators in managing the department’s computers, which included storage, IT, networking, and cyber-security related duties.

8. In 1991, I joined Columbia University’s CS department as a full-time systems administrator, studying towards my MS degree part-time. My MS thesis topic is related to file system reliability, fault tolerance, replication, and failover in mobile networked storage systems using file virtualization. My main duties as a systems administrator involved installing, configuring, and managing many networked servers, proxies, and desktops running several operating systems, as well as network devices setup; this included many software and hardware upgrades, device upgrades, and BIOS firmware/chipset updates/upgrades. My duties also included

ensuring reliable, secure, authenticated access to networked systems/storage and licensed software, as well as software updates, security and bug fixes. Examples of servers and their protocols included email (SMTP), file transfer (FTP), domain names (DNS), network file systems (NFS), network news systems (NNTP), and Web (HTTP).

9. In 1994, I left my systems administrator position to pursue my doctoral studies at Columbia University. My PhD thesis topic was on versatile file system development using stackable (virtualized) file systems, with examples in the fields of security and encryption, efficiency, reliability, and failover. I continued to work part-time as a systems administrator at the CS department, and eventually I was asked to serve as manager to the entire information technology (“IT”) staff. From 1991 to 2001, I was also a member of the faculty-level Facilities Committee that oversaw all IT operations at the CS department.

10. As part of my PhD studies at Columbia, I collaborated on projects to develop advanced AI-like techniques to detect previously unknown viruses (a.k.a. “zero-day malware”), using data mining and rule-based detection. This work led to several highly cited papers (over 1,600 citations for one of the papers alone), and two patents. I also became a Teaching Assistant (TA) for a first-ever Computer Security course given at Columbia University’s CS department with Dr. Matt Blaze as instructor.

11. From 1990 to 1998, I consulted for SOS Corporation and HydraWEB Technologies, as a systems administrator and programmer, managing data storage use and backup/restore duties, databases, web servers, as well as information assurance and cyber-security (*e.g.*, malware protection, software licensing). From 1994 to 2000, I led projects at HydraWEB Technologies, and then became the Director of Software Development—overseeing the development of several products and appliances such as stateful firewalls and HTTP load-

balancers, utilizing network-virtualization and high-availability techniques. From 2009 to 2019, I have consulted for Packet General Networks, a startup specializing in secure, virtualized, network storage and applications' data security in the cloud.

12. In 2001, I joined the faculty of Stony Brook University, a position I have held since that time. In 2002, I joined the Operations Committee, which oversees the IT operations of the CS department at Stony Brook University. From 2006 to 2010, I was the Director of IT Operations of the CS department. My day-to-day duties included setting policies regarding computing, hiring and training new staff, assisting any staff with topics of my specialty, defining requirements for new software/hardware, and purchasing. From 2010 to 2015, I had served as the Co-Chair to the Operations Committee. From 2016 to 2019, I oversaw the IT Operations as the Chair of the Operations Committee. A significant component of these duties included defining and helping implement policies for data management, to ensure the security of users and their data, and data reliability and availability, while minimizing the inconvenience and performance impact to users. I personally helped setup and maintain an initial virtual-host infrastructure in the department. Since late 2019, I've been a member of the department's Executive Committee that also oversees all IT operations.

13. In 2017, I became the department's Graduate Academic Adviser, advising all Master students (over 400 annually on average) and many other graduate students on an assortment of academic matters. In August 2024, I took over as the department's Graduate Program Director, overseeing the entire graduate CS program (700-800 students annually on average).

14. Since 2001, I have personally configured and managed my own research lab's network. This includes setting up and configuring multiple storage systems (*e.g.*, NFS, CIFS/SMB, NAS), virtual and physical environments, applications such as database (*e.g.*, MySQL,

Postgresql), Web servers (*e.g.*, Apache), and mail servers; user access control (*e.g.*, NIS, LDAP), backups and restores, snapshot policies, and more. I've personally installed, configured, changed, replaced parts, and upgraded components in numerous devices including mobile devices, laptops, desktops, and servers, both physical and virtual.

15. Since 1995, I have taught courses on operating systems, storage and file systems, advanced systems programming in Unix/C, systems administration, data structures, data/software security, and more. My courses often use storage, file systems, distributed systems, and system/network security as key teaching principles and practical examples for assignments and projects. I have taught these concepts and techniques to my students, both to my direct advisees as well as in my courses. For example, in my graduate Operating Systems course, I often cover Linux's kernel mechanisms to protect users, applications, and data files, virtual file systems, as well as distributed storage systems (*e.g.*, NFS). And in the System Administration undergraduate course, I covered many topics such as networking, storage, backups, and configuring complex applications such as mail, web, and database servers.

16. My research often investigates computer systems from many angles: security, efficiency, energy use, scalability, reliability, portability, survivability, usability, ease-of-use, versatility, flexibility, and more. My research gives special attention to balancing five often-conflicting aspects of computer systems: performance, reliability, energy use, security, and ease-of-use.

17. Since joining Stony Brook University in 2001, my group in the File-systems and Storage Lab (FSL) has developed many file systems and operating system extensions; examples include a highly-secure cryptographic file system, a portable copy-on-write (COW) versioning file system, a tracing file system useful to detect intrusions, a replaying file system useful for forensics,

a snapshotting and sandboxing file system, a namespace unification file system (that uses stackable, virtualized, file-based COW), an anti-virus file system, an integrity-checking file system, a load balancing and replication/mirroring file system, network file system extensions for security and performance, distributed secure cloud-based storage systems, transactional key-value stores and file systems, OS-level embedded databases, a compiler to convert user-level C code to in-kernel efficient yet safe code, GCC plugins, stackable file system templates, and a Web-based backup system. Many of these projects used one form of virtualization or another (storage, network, host, etc.). I continue to maintain and release newer versions of some of these file systems and software.

18. I have published over 120 refereed publications (in ACM, IEEE, USENIX, and more). To date, my publications have been cited more than 10,000 times (as per Google Scholar as of November 7, 2024). My papers cover a wide range of related technologies such as file systems, storage systems, transactional systems, security, clouds and virtualization, performance benchmarking and optimization, energy efficiency, system administration, web systems, and more. I also published a book titled “Linux NFS and Automounter Administration” (Sybex, 2001), covering systems administration topics related to network storage and data security.

19. Some of my research has led to public software releases that have been used worldwide. I have publicly maintained the Amd Berkeley Automounter in a package called “am-utils” since 1992; this software helps administrators manage the multitude of file system mounts on dozens of different Unix systems, especially helping to automate access to multiple NFS/NAS storage volumes. Since 1997, I have maintained and released several stackable (virtualized) file system software projects for Linux, FreeBSD, and/or Sun Solaris, in a package called FiST. One of my stackable file system encryption projects, called Cryptfs, became the basis for IBM’s public

release of eCryptfs, now part of Linux. Packet General Networks, for whom I have provided consulting services between 2009 and 2019, licensed another encryption file system called Ncryptfs. Another popular file system released in 2003, called Unionfs, offers virtual namespace unification, transparent shadow copying (a.k.a. copy-on-write or COW), file system snapshotting (*e.g.*, useful for forensics and disaster recovery), and the ability to save disk space by sharing a read-only copy of data among several computers, among other features.

20. My research and teaching make extensive use of data security features. For example, each time I taught the graduate operating system course, the first homework assignment includes the creation of a new system call that performs new or added functionality, often for encrypting a file or verifying its integrity; many of my other assignments cover topics of user/process access control, anti-virus filtering, and more. Since 2001, over 1,000 graduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

21. Moreover, in an undergraduate course titled “Advanced Systems Programming in Unix/C,” I cover many topics of system security and vulnerabilities, such as the structure of UNIX processes, and memory segments such as the heap and stack. This course covers details of several hundred Linux system calls. Often, the first assignment for this course is to develop a tool to encrypt/decrypt files using advanced ciphers, use digital signatures to certify the cipher keys used, and reliably recover files in case of failures. Since 2001, several hundred undergraduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

22. In another undergraduate course, System Administration, I taught network configuration, security, and storage configuration and reliability. In a special topics course on

Storage Systems, I covered many topics such as data deduplication, RAID, transactional storage, storage hardware including modern Flash based ones, virtual storage, backup/restore, snapshots and continuous data protection (CDP), NAS and SAN, and NFS.

23. Overall, in addition to the aforementioned experience, my technical experience relevant to the challenged patent(s) at the time of the alleged invention(s) included the following. I configured, used, researched, developed, programmed, and/or used these technologies and systems: network services and protocols including HTTP, DNS, DHCP; network security systems including firewalls, VPN, and NAT; and virtualization at the storage, file system, and network layers.

24. My research has been supported by many federal and state grants as well as industry awards, including an NSF CAREER award, two IBM Faculty awards, two NetApp Faculty awards, a Western Digital award, a Facebook award, several Dell-EMC awards, and several equipment gifts. I received the 2008 SUNY Chancellor's Excellence in Teaching award, and the 2022 SUNY Chancellor's Award for Excellence in Scholarship and Creative Activities (both awards can be given only once in a lifetime). In 2021 I was named an ACM Distinguished Member for "Outstanding Scientific Contributions to Computing."

25. My service record to the community includes serving as the co-chair for the USENIX Annual Technical Conference in 2020 (ATC'20); serving as the co-chair for USENIX File and Storage Technologies (FAST'15) in 2015 and on the FAST Conference Steering Committee from 2015 to 2023; serving on the ACM HotStorage Steering Committee since 2021; and serving as the co-chair in 2012 and on the Steering Committee of the ACM SYSTOR conference since 2012. I have served as an Associate Editor to the ACM Transactions on Storage

(TOS) journal from 2009 to 2022; in 2022, I was named the Editor-in-Chief for ACM's TOS journal.

26. I am a named inventor on four patents, two titled "Systems and Methods for Detection of New Malicious Executables" (U.S. Patent No. 7,487,544, issued February 3, 2009; and U.S. Patent No. 7,979,907, issued July 12, 2011); and two more titled "Multi-Tier Caching," (U.S. Patent No. 9,355,109, issued May 31, 2016; and U.S. Patent 9,959,279, issued May 1, 2018).

27. I have been disclosed as a testifying expert in 19 cases (including *inter partes* review (IPR) proceedings) in the past four years. I have been deposed 13 times and testified in trial twice.

III. MATERIALS CONSIDERED

28. The opinions contained in this declaration are based on the documents I reviewed, my professional judgment, as well as my education, experience, and knowledge regarding electrical engineering, computer science, video and audio and communications sciences and technologies.

29. In forming my opinions expressed in this declaration, I reviewed the following materials:

Exhibit	Description
1001	U.S. Patent 7,949, 785 (filed 2003-31-03, granted 2011-24-05) ("785 patent")
1002	File History for U.S. Patent No. 7,949, 785 (Appl. No. 10/403,818) ("785 FH")
1003	Provisional Appl. No. 60/296, 902 (2001-06-08) ("902 Appl.")
1005	Dr. Erez Zadok Curriculum Vitae
1006	U.S. Patent Application Publication No. US2003/0028671 to Mehta et al. ("Mehta")

Exhibit	Description
1007	Network Working Group Request for Comment (RFC): 1383, entitled An Experiment in DNS Based IP Routing, by C. Huitema (“RFC-1383”)
1013	Microsoft Computer Dictionary, 3rd ed., 1997, excerpts (“Microsoft Computer Dictionary”)
1014	Andrew S. Tanenbaum, <i>Computer Networks</i> , 2nd ed., 1988, excerpts (“Tanenbaum”)
1015	W. Richard Stevens, <i>TCP/IP Illustrated Volume 1, The Protocols</i> , 1994, excerpts (“Stevens”)
1016	William R. Cheswick & Steven M. Bellovin, <i>Firewalls and Internet Security, Repelling the Wily Hacker</i> , 1994, excerpts (“Cheswick”)
1017	Domain Names - Concepts and Facilities, RFC 1034, IETF, 1987, https://www.rfc-editor.org/rfc/pdf/rfc1034.txt.pdf
1018	Domain Names - Implementation and Specification, RFC 1035, IETF, 1987, https://www.rfc-editor.org/rfc/pdf/rfc1035.txt.pdf
1019	IP Encapsulating Security Payload (ESP), RFC 2406, IETF, 1998, https://www.rfc-editor.org/rfc/pdf/rfc2406.txt.pdf
1020	Dynamic Host Configuration Protocol, RFC 2131, IETF, 1997, https://www.rfc-editor.org/rfc/pdf/rfc2131.txt.pdf
1021	Linux DNS Server Administration, Craig Hunt, 2000, excerpts (“Hunt”)
1022	<i>Intellectual Ventures v. Liberty Mutual</i> , No. 23-cv-525 (E.D. Tex.) – Infringement Contentions
1023	<i>Intellectual Ventures v. Comerica</i> , No. 23-cv-524 (E.D. Tex.) – Infringement Contentions

30. I have also reviewed any other materials I refer to in this declaration in support of my opinions.

31. All of the opinions contained in this declaration are based on the documents I reviewed and my knowledge, experience, and professional judgment. My opinions have also been

guided by my appreciation of how a person of ordinary skill in the art would have understood the claims and the specification of the '844 patent at the time of the alleged invention, which I have been asked to initially consider as no earlier than March 31, 2004. My opinions reflect how one of ordinary skill in the art would have understood the '785 patent, the prior art to the patent, and the state of the art at the time of the alleged invention.

32. Based on my experience and expertise, it is my opinion that certain references disclose and/or suggest all the features recited in claims 1-2, 6-12, 22, 25-26, 28-49, 51-55, 60-63, 65-66, 72-80, and 82-90 ("challenged claims") of the '785 patent, as I discuss in detail below.

IV. LEVEL OF ORDINARY SKILL IN THE ART

33. I have been informed and understand that, in the context of an invalidity analysis, a person having ordinary skill in the art ("POSA") is a hypothetical person who looks to prior art at the time of the invention. I further understand that the factors that may be considered in determining the level of ordinary skill include: (1) the problems encountered in the art; (2) the prior art solutions to the problems encountered in the art; (3) the rapidity of innovation; (4) the sophistication of the technology; and (5) the education level of active workers in the field. I understand that these factors need not all be considered for the analysis and that one or more of these factors may control.

34. I was asked to provide my opinion on the level of one of ordinary skill in the art with respect to the alleged invention of the '785 patent as of March 31, 2003. In my opinion a POSA would have had a bachelor's degree in computer science, computer engineering, or an equivalent degree. A POSA would also have had approximately two years of experience working in the fields of networking including network virtualization, security, and management. Additional experience might substitute for less education and vice versa.

35. As of March 31, 2003, I met, and in fact exceeded, the qualifications of a person of ordinary skill in the art. To be clear, all of my opinions in this declaration are from the perspective of one of ordinary skill in the art as I have defined it here during the relevant timeframe.

V. RELEVANT LEGAL STANDARDS

36. I am not an attorney and offer no legal opinions, but in the course of my work, I have had experience studying and analyzing patents and patent claims from the perspective of a person skilled in the art.

37. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to forming my opinions. My understanding of the law is as follows.

38. Petitioners' counsel has informed me that a published nonprovisional application is entitled to claim priority to the filing date of its provisional application (as opposed to the filing date of the published non-provisional application) if (1) the provisional application discloses the portions of the published non-provisional application relied upon as prior art, and (2) at least one claim of the published non-provisional claim is supported by the provisional application, *i.e.*, the provisional application must disclose the subject matter of the claim and enable a person of ordinary skill in the art to make and use the claim.

39. Petitioners' counsel has informed me that for the prior art to inherently disclose a claimed limitation, the prior art need not expressly disclose the limitation, so long as the claimed limitation necessarily flows from a disclosure in the prior art. I also understand that it is acceptable to examine evidence outside the prior art reference (extrinsic evidence) in determining whether a feature, while not expressly discussed in the reference, is necessarily present in that reference.

40. Petitioners' counsel has informed me that a patent claim can be considered to have been obvious to a person having ordinary skill in the art at the time the application was filed. I am

informed that this means that, even if all of the requirements of a claim are not found in a single prior art reference, the claim is not patentable if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a person having ordinary skill in the art at the time of the invention.

41. I have been informed by Petitioners' counsel that a determination of whether a claim would have been obvious should be based upon several factors, including, among others:

- the level of ordinary skill in the art at the time the application was filed;
- the scope and content of the prior art; and
- what differences, if any, existed between the claimed invention and the prior art.

42. I have been informed by Petitioners' counsel that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a person having ordinary skill in the art. Alternatively, I understand that the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to one having ordinary skill in the art. In determining whether a combination based on either a single reference or multiple references would have been obvious, it is appropriate to consider, among other factors:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- whether a person having ordinary skill in the art could implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by those skilled in the art;

- whether a person having ordinary skill would have recognized a reason to combine known elements in the manner described in the claim;
- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

43. I have been informed by Petitioners' counsel that one of ordinary skill in the art has ordinary creativity and is not an automaton. Petitioners' counsel has also informed me that in considering obviousness, obviousness may not be determined using the benefit of hindsight, including hindsight derived from the patent being considered.

44. I further understand that a hypothetical person of ordinary skill in the art is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent at issue. I have been informed that there are two criteria for determining whether prior art is analogous and thus can be considered prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the patentee's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the patentee is involved. I have also been informed that the field of endeavor of a patent is not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field. I have also been informed that a reference is reasonably pertinent if, even though it may be in a different field from that of the patentee's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to a patentee's attention in considering their problem.

45. I further understand that, in making a determination as to whether the claimed invention would have been obvious to a person of ordinary skill, the Board may consider certain objective factors if they are present, such as: commercial success of products practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results; copying; and praise by others in the field. These factors are generally referred to as “secondary considerations” or “objective indicia” of nonobviousness. I understand, however, that for such objective evidence to be relevant to the obviousness of a claim, there must be a causal relationship (called a “nexus”) between the claim and the evidence, and that this nexus must be based on a novel element of the claim rather than something in the prior art. I also understand that even when they are present, secondary considerations may be unable to overcome primary evidence of obviousness (such as motivation to combine with predictable results) that is sufficiently strong.

VI. SUMMARY OF OPINIONS

46. This Declaration explains the conclusions I have formed based on my analysis. These conclusions are summarized the table below:

Ground	Basis	Relied-On References	Claim(s)
1	Obviousness	Mehta and RFC-1383	1-2, 6-12, 22, 25-26, 28-49, 51-55, 60-63, 65-66, 72-80, and 82-90

VII. TECHNOLOGY OVERVIEW

47. In this section, I provide an overview of the background art as understood by a person of ordinary skill in the art as of March 31, 2023.

A. General Computer Operations

48. Computers are hardware and software devices that execute instructions. The hardware often includes a central processing unit (CPU) or other microprocessors, typically called

“processor,” volatile memory (*e.g.*, RAM (random-access memory), or DRAM (dynamic random-access memory)), non-volatile (persistent) storage (*e.g.*, a hard disk or flash drive), and input/output (I/O) devices (*e.g.*, network interface card, mouse, keyboard, display). RAM, hard disks, and flash drives are considered, generally, “storage” or “memory” devices because they can store and hold bits of information (whether persistently or not). Computers can have multiple storage devices. Processors are coupled to main memory using a memory bus and to peripherals such as hard disks and network interface cards using one or more I/O busses.

49. Memory and storage devices have considerable differences in their speeds, capacities, and costs. Non-volatile storage devices such as hard disks have the largest capacity, are least expensive per gigabyte, and are slowest (typically operating at millisecond speeds). Volatile DRAM devices are smaller, more expensive, and faster than hard disks (typically operating at microsecond speeds). CPUs’ own memory caches and operations are the smallest, most expensive, and fastest (typically operating at nanosecond speeds). Overall, CPUs are about 1,000× faster than DRAM, which in turn is about 1,000× faster than hard disks.

50. Computer programs are often written in a high-level human-readable language (*e.g.*, C, C++, Java, Perl, PHP, Python), then translated using a compiler or script processor to machine instructions understood by the CPU (*e.g.*, Intel or AMD processor). Machine instructions—generally called “software”—are stored in files on persistent media (*e.g.*, HDD or ROM), loaded into DRAM, and then loaded into the CPU where they can be executed. All modern computers operate in this manner, regardless of their purpose: a small laptop, a server in a data center, a mainframe computer, a personal workstation, a proxy device, a gateway, a firewall, a router, an appliance, a virtual machine, a handheld or mobile device, smart card, SD card, etc.

Computer systems may run multiple software programs, from background servers and daemons to user applications.

51. Users (including administrators) can login to one or more computers (*e.g.*, desktop, laptop, server) and execute one or more applications (*e.g.*, word processing, web browsing, backup tool). Some computers offer external interfaces and APIs to communicate with certain devices and especially peripherals. For example, both a USB storage device as well as a printer use a well-defined “protocol” to allow users to communicate with them through the main computer.

52. “IPC” or “interprocess communication” is “[t]he ability of one task or process to communicate with another in a multi-tasking operating system. Common methods include pipes, semaphores, shared memory, queues, signals, and mailboxes. *Acronym: IPC.*” EX1013 (Microsoft Computer Dictionary) at 261, 264.

B. Networking

1. Networking Overview

53. A Local Area Network (LAN) is typically a network at a local site where network latencies are short and physical distances between computers are small. A LAN and its computers are often protected from the rest of the internet using a firewall. EX1013 (Microsoft Computer Dictionary) at 276 (“**LAN** *n.* Acronym for **local area network**. A group of computers and other devices dispersed over a relatively limited area and connected by a communications link that enables any device to interact with any other on the network. LANs commonly include microcomputers and shared resources such as laser printers and large hard disks. The devices on a LAN are known as nodes, and the nodes are connected by cables through which messages are transmitted. *See also* baseband network, broadband network, bus network, collision detection,

communications protocol, contention, CSMA/CD, network, ring network, star network, token bus network, token passing, token ring network. *Compare* wide area network.”).

54. Another form of a local area network is one that operates inside a single computer. A “localhost” is “[t]he name that is used to represent the same computer on which a TCP/IP message originates. An IP packet sent to localhost has the IP address 127.0.0.1 and does not actually go out to the Internet. *See also* IP address, packet (definition 1), TCP/IP.” EX1013 (Microsoft Computer Dictionary) at 287. Often, when users/processes want to communicate with other users/processes, using TCP/IP communications on the same machine, they can use the localhost address: it is considered a trusted, high-speed internal network.

55. Conversely, a Wide-Area Network (WAN) refers to computers across the entire Internet, which are often distant from each other, exhibit longer latencies, and afford little protection. EX1013 (Microsoft Computer Dictionary) at 507 (“**wide area network** *n.* A communications network that connects geographically separated areas. *Acronym:* WAN.”).

56. A firewall typically intercepts all traffic between computers in a LAN and the WAN, monitors and controls access, and can filter out undesired traffic. Often, LAN computers are far more trusted than WAN computers; therefore, a firewall can be said to separate the trusted network (LAN or localhost) from the untrusted network (WAN). EX1013 (Microsoft Computer Dictionary) at 197 (“**firewall** *n.* A security system intended to protect an organization’s network against external threats, such as hackers, coming from another network, such as the Internet. A firewall prevents computers in the organization’s network from communicating directly with computers external to the network and vice versa. Instead, all communication is routed through a proxy server outside of the organization’s network, and the proxy server decides whether it is safe to let a particular message or file pass through to the organization’s network.”).

57. A server is a computer or software that provides a service to other computers, called clients. There are many servers on the Internet, for example, Web servers and Mail servers. EX1013 (Microsoft Computer Dictionary) at 430 (“**server** *n.* 1. On a local area network (LAN), a computer running administrative software that controls access to the network and its resources, such as printers and disk drives, and provides resources to computers functioning as workstations on the network. 2. On the Internet or other network, a computer or program that responds to commands from a client. For example, a file server may contain an archive of data or program files; when a client submits a request for a file, the server transfers a copy of the file to the client. *See also* client/server architecture. *Compare* client (definition 3).”

58. A client is typically an end-point computer or software, sometimes used by a user, that communicates across LAN and WAN networks; a client may contact servers to request some service. A user’s desktop, laptop, and handheld device are examples of clients. (Note that computer can be both a client and a server.) EX1013 (Microsoft Computer Dictionary) at 92 (“**client** *n.* 1. In object-oriented programming, a member of a class (group) that uses the services of another class to which it is not related. *See also* inheritance (definition 1). 2. A process, such as a program or task, that requests a service provided by another program—for example, a word processor that calls on a sort routine built into another program. The client process uses the requested service without having to ‘know’ any working details about the other program or the service itself. *Compare* child (definition 1), descendant (definition 2). 3. On a local area network or the Internet, a computer that accesses shared network resources provided by another computer (called a server). *See also* client/server architecture, server.”).

59. A client/server architecture describes how client computers interact with server computers. EX1013 (Microsoft Computer Dictionary) at 92 (“**client/server architecture** *n.* An

arrangement used on local area networks that makes use of distributed intelligence to treat both the server and the individual workstations as intelligent, programmable devices, thus exploiting the full computing power of each. This is done by splitting the processing of an application between two distinct components: a ‘front-end’ client and a ‘back-end’ server. The client component is a complete, stand-alone personal computer (not a ‘dumb’ terminal), and it offers the user its full range of power and features for running applications. The server component can be a personal computer, a minicomputer, or a mainframe that provides the traditional strengths offered by minicomputers and mainframes in a time-sharing environment: data management, information sharing between clients, and sophisticated network administration and security features. The client and server machines work together to accomplish the processing of the application being used. Not only does this increase the processing power available over older architectures but it also uses that power more efficiently. The client portion of the application is typically optimized for user interaction, whereas the server portion provides the centralized, multiuser functionality. *See also* distributed intelligence.”).

2. Network Protocols

60. To be useful, computer systems are connected using a worldwide network known as the Internet. Without being able to offer services over the network or access networked services, the Internet would not have become so pervasive in our lives. Indeed, one of the early pioneers in the field of networking, a then fledgling startup named Sun Microsystems (now acquired by Oracle), coined the visionary marketing phrase “the network *is* the computer.”

61. To ensure that networked systems could inter-operate, the community banded together starting in the late 1970’s and came up with several network standards. One of the most common and heavily used is the Open Systems Interconnection (OSI) model, administered by the

International Organization for Standardization (ISO). See generally EX1014 (“Tanenbaum”), pp. 6-13, Figure 1-7 (reproduced below):

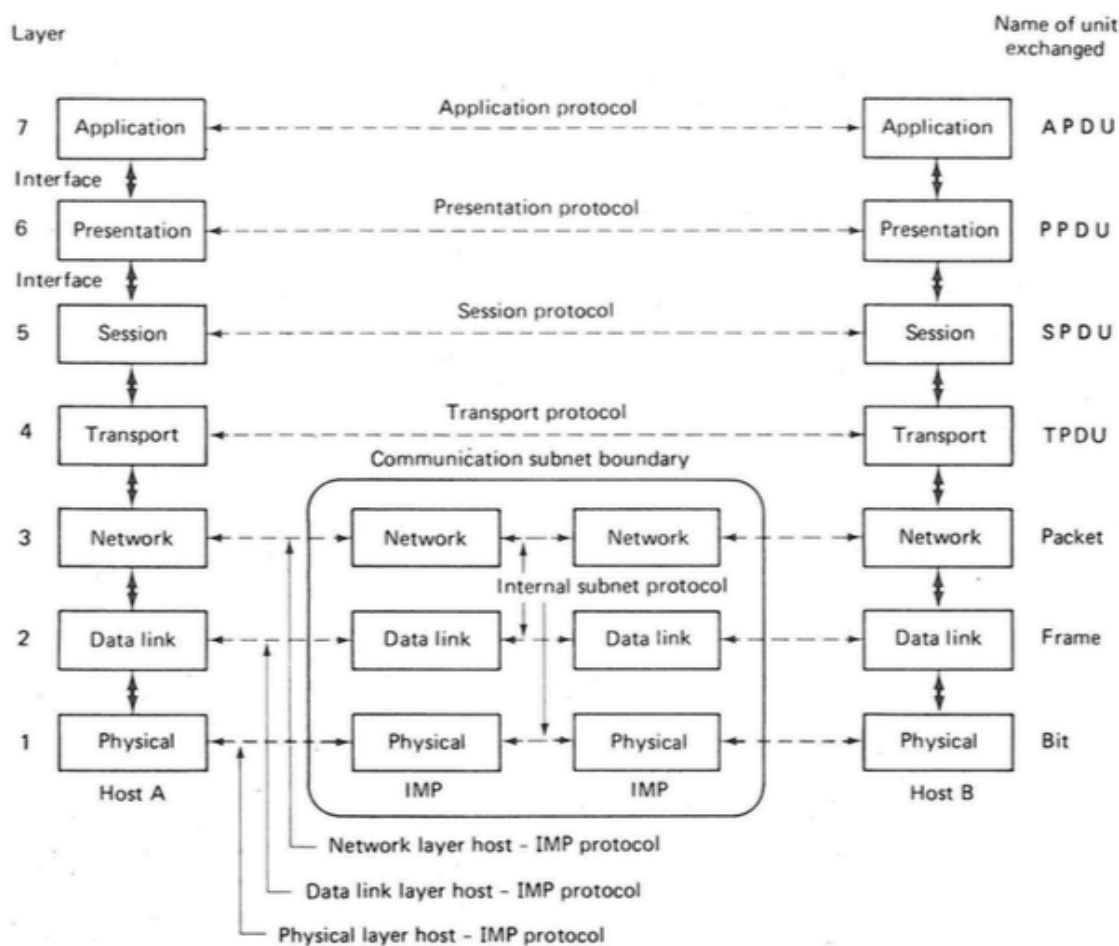


Fig. 1-7. The network architecture used in this book. It is based on the OSI model.

62. The OSI model defines the following seven logical abstraction layers:

63. *Layer 1: Physical Layer*, is the lowest abstraction layer, and defines how bits are communicated over physical media (e.g., wires). Examples here include twisted-pair cables with RJ45 connectors, used by Ethernet.

64. *Layer 2: Data Link Layer*, defines how data frames are transmitted reliably between two nodes over the physical layer. A popular example here is Ethernet.

65. *Layer 3: Network Layer*, defines how to communicate across multiple nodes, including how to route and address packets comprising multiple data frames. The most common example of this layer is the Internet Protocol (IP). This layer is where most traditional firewalls operate.

66. *Layer 4: Transport Layer*, defines how to transmit segments of data comprising multiple packets, and control the traffic (*e.g.*, ordering, acknowledgment, handling missing or duplicate packets, bandwidth control, timeouts and retransmissions, and more). The most popular examples of this layer are the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP).

67. *Layer 5: Session Layer*, defines continuous exchanges or communications among parties (*e.g.*, Remote Procedure Calls or RPCs).

68. *Layer 6: Presentation Layer*, defines data translations between a networked service and applications (*e.g.*, encryption, compression, encoding).

69. *Layer 7: Application Layer*, is the highest abstraction layer and directly corresponds to an application's use of the network. Examples include Web services (HTTP) and URL filtering, email (eSMTP) and anti-virus filtering, file transfer (FTP), and many more. This layer is where most application-level firewalls and filtering devices operate.

70. In a typical communication network, electrical signals arrive on the wire and are captured by a Network Interface Card (NIC). The NIC translates these signals into bits (1's and 0's) and passes these bits as short frames up to the Operating System (OS) for further processing. The frames are often checked for validity and passed on to a software module at the network layer. Again, the data is validated and processed further and passed up to the next layer up the chain.

71. Sometimes, data is deemed to be invalid (*e.g.*, doesn't pass a checksum check), or is a duplicate of data already seen, or perhaps was not intended for the receiving host. In such cases, the data packets are often "dropped" or discarded and no acknowledgments are sent to the sender. If all goes well, however, the data eventually reaches an application that can interpret it (*e.g.*, a Web browser displaying a JPEG image whose bits arrived from a remote Web server).

72. UDP (User Datagram Protocol) is a connectionless transport layer protocol providing no reliability. *See* EX1015 ("Stevens"), p. 12 ("UDP provides no reliability: it sends the datagrams that the application writes to the IP layer, but there is no guarantee that they ever reach their destination."). This means that the packets could be lost, reordered, or duplicated on their way to a destination host, and the sender would have no way of knowing this. Senders using UDP typically have to arrange other means to detect and retransmit lost packets.

73. TCP is a connection-oriented transport protocol, layered on top of the connectionless IP protocol. *See* EX1015 ("Stevens"), p. 14 ("Even though TCP and UDP use the same network layer (IP), TCP provides a totally different service to the application layer than UDP does. TCP provides a connection-oriented, reliable, byte stream service. The term connection-oriented means the two applications using TCP (normally considered a client and a server) must establish a TCP connection with each other before they can exchange data."). For example, TCP tracks all packets to ensure that lost packets are retransmitted and acknowledged, to ensure reliable connections between the two TCP endpoints.

74. TCP requires a 3-way handshake sequence before the connection can be established and data can flow in both directions. First, a sending host initiates communications by sending a SYN packet. Second, the recipient prepares for the communications by reserving a network slot and responds with a SYN-ACK message. Finally, the initiating host responds back with an ACK

message. At this stage both sides know for certain that bi-directional communication is possible. See EX1014 (“Tanenbaum”), p. 19 (“**The Transport Layer in the ARPANET (TCP) ... ARPANET grew into the ARPA Internet ... led to ... introduction of ... TCP (Transmission Control Protocol)**”) (emphasis in original), 20 (“The *SYN* bit is used to establish connections. The connection request has *SYN* = 1 and *ACK* = 0 to indicate that the piggyback acknowledgement field is not in use. The connection reply does bear an acknowledgement, so it has *SYN* = 1 and *ACK* = 1. In essence the *SYN* bit is used to denote *CONNECTION REQUEST* and *CONNECTION CONFIRM*, with the *ACK* bit used to distinguish between those two possibilities.”) (emphasis in original); see also EX1015 (“Stevens”), p. 19 (“**Connection Establishment Protocol ... To establish a TCP connection: 1. The requesting end (normally called the *client*) sends a SYN segment ... 2. The server responds with its own SYN segment ... The server also acknowledges the client’s SYN by ACKing the client’s ISN plus one ... 3. The client must acknowledge this SYN from the server by ACKing the server’s ISN plus one (segment 3). These three segments complete the connection establishment. This is often called the *three-way handshake*.**”) (emphasis in original), 20 (“The side that sends the first SYN is said to perform an *active open*. The other side, which receives this SYN and sends the next SYN, performs a *passive open*.”) (emphasis in original), Fig. 18.3 p. 20 (“Time line of connection establishment and connection termination.” below).



Figure 18.3 Time line of connection establishment and connection termination.

75. IP headers include the source and destination addresses. See EX1015 (“Stevens”), p. 7 (Figure 3.1, reproduced below):

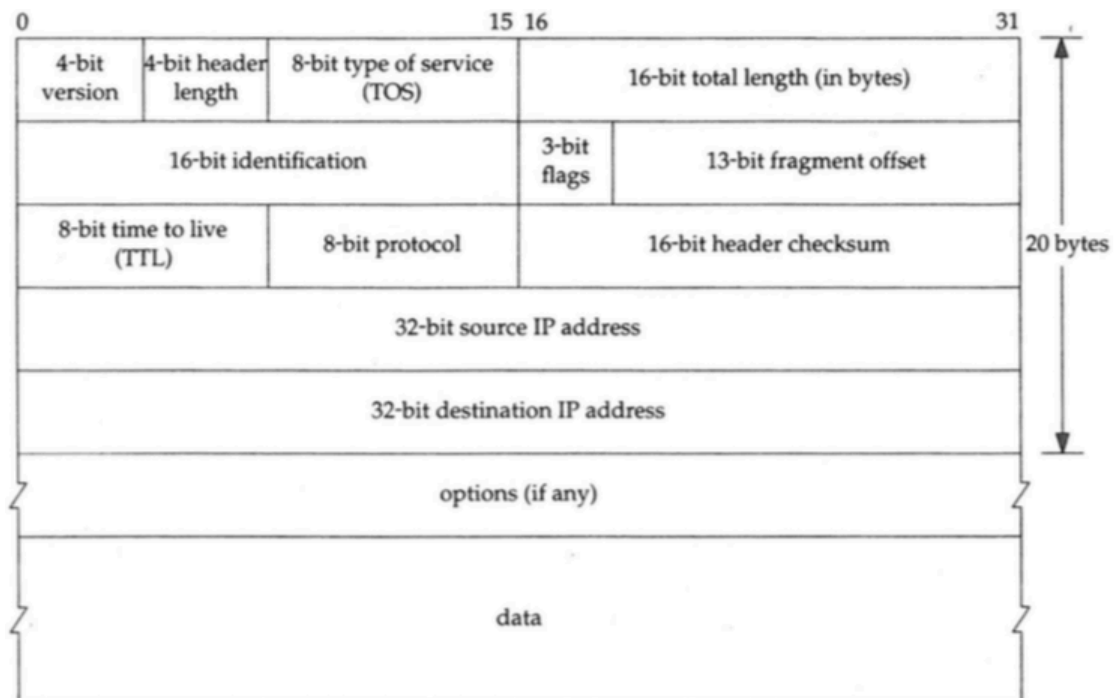


Figure 3.1 IP datagram, showing the fields in the IP header.

76. At the transport layer, UDP packets include the source and destination ports. See EX1015 (“Stevens”), p. 13 (Figure 11.2, reproduced below):

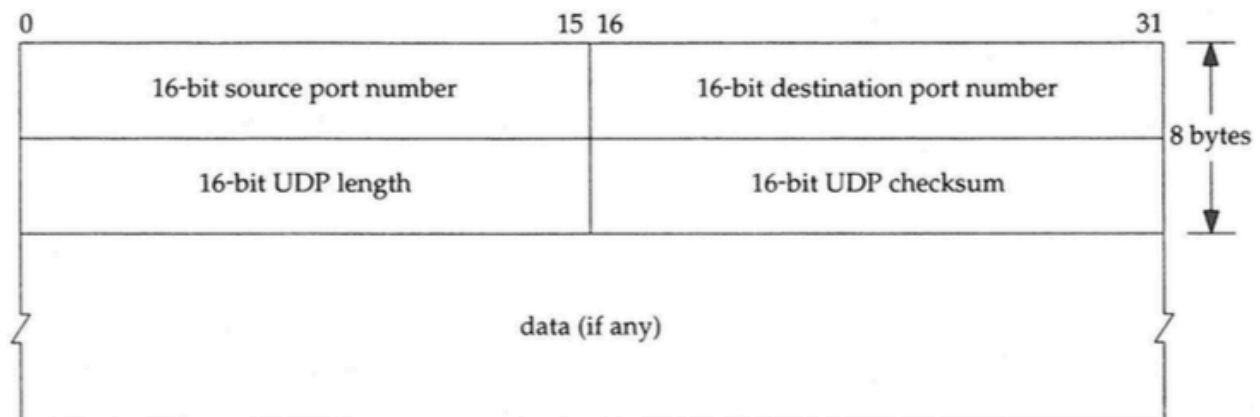


Figure 11.2 UDP header.

77. At the transport layer, TCP packets also include the source and destination ports. See EX1015 (“Stevens”), p. 16 (Figure 17.2, reproduced below):

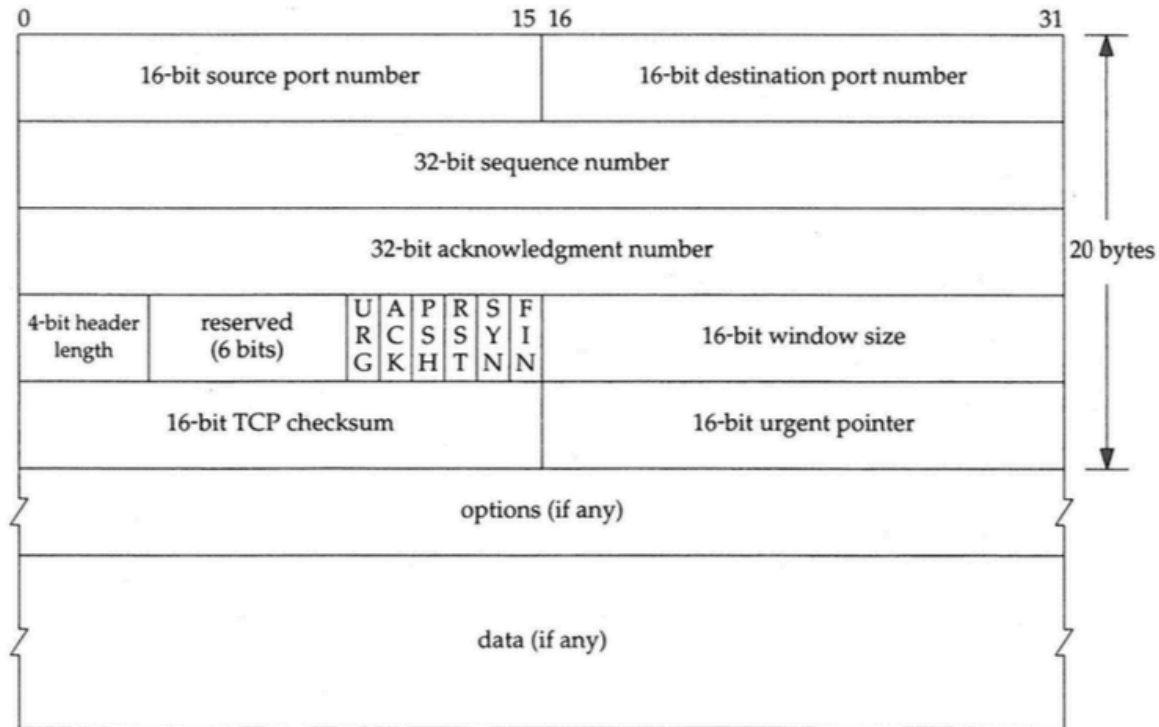


Figure 17.2 TCP header.

78. Therefore, TCP and UDP packets using IP include two IP addresses and two ports—for the source and destination ones.

79. ATM is a networking technology that establishes a full circuit between two endpoints, akin to the telephone system. ATM is thus said to be a circuit switched network. *See* EX1013 (Microsoft Computer Dictionary) at pp. 33-34 (“**ATM** *n.* 1. Acronym for **A**synchronous **T**ransfer **M**ode. A network technology capable of transmitting data, voice, video, and frame relay traffic in real time. Data, including frame relay data, is broken into packets containing 53 bytes each, which are switched between any two nodes in the system at rates ranging from 1.5 Mbps to 622 Mbps. ATM is defined in the broadband ISDN protocol at the levels corresponding to levels 1 and 2 of the ISO/OSI model. ...”).

80. A network flow can be thought of as a logical communication or a stream of bytes in a session between at least two entities. *See* EX1015 (“Stevens”), p. 17 (“If we consider the

stream of bytes flowing in one direction between two applications, TCP numbers each byte with a sequence number. ... This means that data can be flowing in each direction, independent of the other direction. Therefore, each end of a connection must maintain a sequence number of the data flowing in each direction.”). TCP provides implicit flow control, where both ends of a TCP connection can control their transmission rates. See EX1015 (“Stevens”), p. 15 (“TCP also provides flow control. Each end of a TCP connection has a finite amount of buffer space. A receiving TCP only allows the other end to send as much data as the receiver has buffers for. This prevents a fast host from taking all the buffers on a slower host.”), 18 (“TCP’s flow control is provided by each end advertising a *window size*.”). In the early 1990’s, TCP implementations advanced (in no small part thanks to Van Jacobson) to improve TCP’s reliability through flow control (*e.g.*, using “exponential back-off” during packet transmission timeouts, and “slow start” to gradually increase transmission rates).

81. Proxies, routers, bridges, and routers share much in common: they receive network packets on one end, process them, then transmit them on another end: their difference often boils down to which OSI network layer they operate at.

82. A “router” is “[a]n intermediary device on a communications network that expedites message delivery. On a single network linking many computers through a mesh of possible connections, a router receives transmitted messages and forwards them to their correct destinations over the most efficient available route. On an interconnected set of local area networks (LANs) using the same communications protocols, a router serves the somewhat different function of acting as a link between LANs, enabling messages to be sent from one to another. See also bridge, gateway.” EX1013 (Microsoft Computer Dictionary) at 415.

83. A “bridge” is “[a] device that connects networks using the same communications protocols so that information can be passed from one to the other. Compare gateway. 2. A device that connects two local area networks, whether or not they use the same protocols. A bridge operates at the ISO/OSI data-link layer. *See also* data-link layer. *Compare* router.” EX1013 (Microsoft Computer Dictionary) at 63

84. A “bridge router” is “[a] device that supports the functions of both a bridge and router. A bridge router links two segments of a local or wide area network, passing packets of data between the segments as necessary, and uses Level 2 addresses for routing. *Also called* Brouter. *See also* bridge (definition 2), router.” EX1013 (Microsoft Computer Dictionary) at 63.

85. A “gateway” is “[a] device that connects networks using different communications protocols so that information can be passed from one to the other. A gateway both transfers information and converts it to a form compatible with the protocols used by the receiving network. *Compare* bridge.” EX1013 (Microsoft Computer Dictionary) at 215.

86. A “proxy server” is “[a] firewall component that manages Internet traffic to and from a local area network (LAN) and can provide other features, such as document caching and access control. A proxy server can improve performance by supplying frequently requested data, such as a popular Web page, and can filter and discard requests that the owner does not consider appropriate, such as requests for unauthorized access to proprietary files. *See also* firewall.” EX1013 (Microsoft Computer Dictionary) at 387.

87. Over the years, many firewalls, proxies, routers, network devices, and security gateways were developed. Earlier ones tended to operate at one or a few of the lower OSI network layers and applied simpler stateless rules; subsequent firewalls were stateful and understood the legitimate life cycle of a protocol (*e.g.*, TCP’s 3-way handshaking). *See* EX1016 (“Cheswick”),

pp. 6-7 (“**Packet-Filtering Gateways...** Packet filters can provide a cheap and useful level of gateway security. Used by themselves, they are cheap: the filtering abilities come with the router software. Since you probably need a router to connect to the Internet in the first place, there is no extra charge.”), 7 (“Packet filters work by dropping packets based on their source or destination addresses or ports... Most security policies require finer control than this: they need to define access to specific services for hosts that are otherwise untrusted.”), 8 (“**Filtering Other Protocols...** Other protocols are layered on top of IP as well; depending on your environment, these may need to be filtered also. Of particular import is ICMP, the Internet Control Message Protocol.”), 9 (“Many advanced gateway designs rely in part on packet filtering. They are likely to work well, but pure packet filters leave us feeling uncomfortable.”).

88. As the field quickly matured, more advanced gateways began to operate at the application layer (*e.g.*, anti-virus, URL filtering). *See* EX1016 (“Cheswick”), p. 9 (“**Application-Level Gateways...** An application-level gateway represents the opposite extreme in firewall design. Rather than using a general-purpose mechanism to allow many different kinds of traffic to flow, special-purpose code can be used for each desired application. Although this seems wasteful, it is likely to be far more secure than any of the alternatives... Application gateways have another advantage that in some environments is quite critical: it is easy to log and control *all* incoming and outgoing traffic.”).

89. Principles of network quality of service (QoS, sometimes called traffic shaping) were well known in the 1980’s, including explicit bandwidth control and reservations. *See* EX1014 (“Tanenbaum”), p. 18 (“The *qos* parameter is actually two lists of values that determine the quality of service provided by the connection. The first list gives the goal—what the caller really wants. The second list gives the minimum values considered acceptable. If the network

service is unable to provide at least the minimum value for any parameter specified by either the caller (calling party) or the callee (called party), the connection establishment fails. The values that may be specified include throughput, delay, error rate, secrecy, and cost, among others.”).

3. The Domain Name System, Hostnames, and IP Addresses

90. As discussed in §VII.A above, every computer has an internal local hostname called “localhost” with an IP address 127.0.0.1. This IP address is accessible only internally on the same host. Computers, however, can have one or more external host names and IP addresses, used to reach (and be reached by) other hosts on the Internet.

91. A “host name” is “[t]he name of a specific server on a specific network within the Internet, leftmost in the complete host specification. For example, `www.microsoft.com` indicates the server called ‘www’ within the network at Microsoft Corporation.” (EX1013, Microsoft Computer Dictionary at 236.)

92. A “domain name” is “[a]n address of a network connection that identifies the owner of that address in a hierarchical format: `server.organization.type`. For example, `www.whitehouse.gov` identifies the Web server at the White House, which is part of the U.S. government.” (*Id.* at 158.)

93. A “top-level domain” is, “[i]n the domain-name system of Internet addresses, any of the broadest category of names, under which all domain names fit. Top-level domains for sites in the United States include `.com`, `.edu`, `.gov`, `.net`, and `.org`. *See also* DNS (definition 1), major geographic domain.” (*Id.* at 472.)

94. For example, “`www.microsoft.com`” is often called the “fully qualified host name” (FQHN) as it comprises several parts: the short host name “www” and the fully qualified domain name (FQDN) “microsoft.com,” which itself comprises a top-level domain (TLD) of “.com” and

a sub-domain “microsoft.com”. The entity Microsoft Corporation is delegated (by Internet-wide authorities) the right to define any sub-domain and hostname under “microsoft.com.”

95. Host names are intended for people to use because they are easier to remember. But internally, for computers to be able to communicate with each other, they must do so with a unique, publicly accessible 32-bit number (called an “IP address”) assigned to each host across the entire Internet. Host names can change and may not even be unique. This is akin to how every U.S. person has a name that is not unique and can change, but each is assigned a unique social security number at birth.

96. An “IP address” is “Short for Internet Protocol address. A 32-bit (4-byte) binary number that uniquely identifies a host (computer) connected to the Internet to other Internet hosts, for the purposes of communication through the transfer of packets. An IP address is expressed in ‘dotted quad’ format, consisting of the decimal values of its 4 bytes, separated with periods; for example, 127.0.0.1. The first 1, 2, or 3 bytes of the IP address, assigned by InterNIC Registration Services, identify the network the host is connected to; the remaining bits identify the host itself. The 32 bits of all 4 bytes together can signify almost 2^{32} , or roughly 4 billion, hosts. (A few small ranges within that set of numbers are not used.) *See also* host, InterNIC, IP, packet (definition 2). *Compare* domain name.” (*Id.* at 264.)

97. Managing the many domains, host names, and IP addresses on the Internet is accomplished with a massive, distributed network of “Domain Name System” servers that each maintain their own small view of the Internet’s naming.

98. “DNS” is an “1. Acronym for Domain Name System. The system by which hosts on the Internet have both domain name addresses (such as bluestem.prairienet.org) and IP addresses (such as 192.17.3.4). The domain name address is used by human users and is

automatically translated into the numerical IP address, which is used by the packet-routing software. *See also* domain name address, IP address. 2. Acronym for Domain Name Service. The Internet utility that implements the Domain Name System (see definition 1). DNS servers, also called *name servers*, maintain databases containing the addresses and are accessed transparently to the user.” (*Id.* at 155.)

99. A “DNS server” is “[a] computer that can answer Domain Name Service (DNS) queries. The DNS server keeps a database of host computers and their corresponding IP addresses. Presented with the name apex.com, for example, the DNS server would return the IP address of the hypothetical company Apex. *See also* DNS (definition 2), IP address.” (*Id.* at 155.)

100. The Domain Name System was so important to the very foundations of the Internet, that it was one of the earliest standards to be defined in the 1980s. Well before the WWW even existed, computers needed a way to identify each other and access other computers consistently.

101. “The DNS has three major components:” (1) “The DOMAIN NAME SPACE and RESOURCE RECORDS” which is a database that maps host names and IP addresses to each other, among other pieces of information; (2) “NAME SERVERS” that respond to queries from anywhere in the world to translate hostnames to IP addresses and vice versa, where “[a] name server may cache structure or set information about any part of the domain tree....” (EX1017, Domain Names Concepts, RFC 1034, at 6.)

102. The third part are software components used to perform these name/IP mapping, called “resolvers.” (3) “RESOLVERS are programs that extract information from name servers in response to client requests. Resolvers must be able to access at least one name server and use that name server’s information to answer a query directly, or pursue the query using referrals to

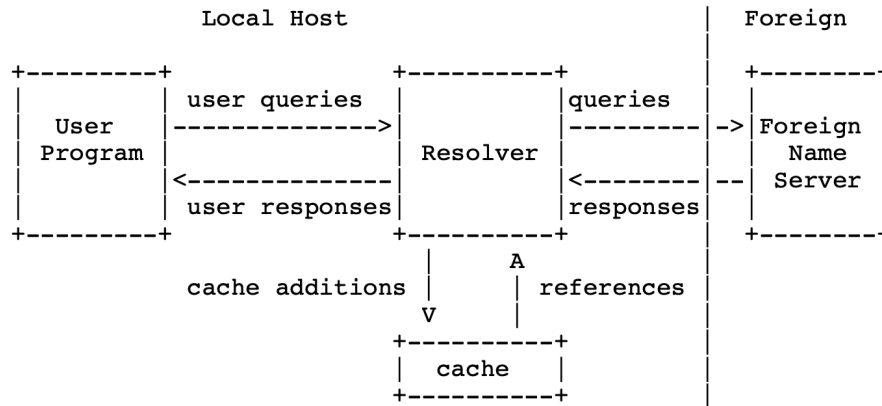
other name servers. A resolver will typically be a system routine that is directly accessible to user programs; hence no protocol is necessary between the resolver and the user program.” (*Id.* at 6.)

103. “Resolvers are programs that interface user programs to domain name servers. In the simplest case, a resolver receives a request from a user program (*e.g.*, mail programs, TELNET, FTP) in the form of a subroutine call, system call etc., and returns the desired information in a form compatible with the local host’s data formats. The resolver is located on the same machine as the program that requests the resolver’s services, but it may need to consult name servers on other hosts. Because a resolver may need to consult several name servers, or may have the requested information in a local cache, the amount of time that a resolver will take to complete can vary quite a bit, from milliseconds to several seconds.” (*Id.* at 29.)

104. Indeed, name resolution in the world-wide DNS system can be slow, as it may require querying multiple DNS servers all around the globe. Many such queries are performed every second on the Internet. Thus, DNS resolutions can be slow, and why caching is employed extensively. *See also* §**Error! Reference source not found.**

105. “A very important goal of the resolver is to eliminate network delay and name server load from most requests by answering them from its cache of prior results. It follows that caches which are shared by multiple processes, users, machines, etc., are more efficient than non-shared caches.” (EX1017, Domain Names Concepts, RFC 1034, at 29.) “7.4. Using the cache [] In general, we expect a resolver to cache all data which it receives in responses since it may be useful in answering future client requests.” (EX1018, Domain Names - Implementation and Specification, RFC 1035 at 47.)

106. The overall architecture of a DNS system, including clients, resolvers, caches, and DNS servers, can be seen in the following figure:



(*Id.* at 4.)

107. DNS describes the address ('A') records to define the IP address(es) of a named host. "A RDATA format ... Hosts that have multiple Internet addresses will have multiple A records. ... The RDATA section of an A line in a master file is an Internet address expressed as four decimal numbers separated by dots without any imbedded spaces (e.g., '10.2.0.52' or '192.0.5.6')." EX1018 (RFC 1035), pp. 20-21.

108. A host can have multiple records such as 'A' address records, and the DNS server will return all of them to a query. "In Listing 6.4, every host name has a unique address, and every address has a unique host name. This does not have to be the case, as illustrated in Listing 6.5. ... In Listing 6.5, owl is a multi-homed host. The name owl is assigned both of the host's addresses, so a query for owl returns both 172.16.5.15 and 172.16.7.32. The remote system's resolver can sort out which address it wants to use. (As you learned in Chapter 4, 'Configuring the Resolver,' the resolver sortlist option can be used to control which address is used.)" EX1021 (Hunt), p. 144, Listing 6.5 (reproduced below):

Listing 6.5 Address Records for Multi-Homed Hosts

owl	A	172.16.5.15
owl	A	172.16.7.32
owl5	A	172.16.5.15
owl7	A	172.16.7.32

109. DNS supports multiple records. *See generally*, EX1018 (RFC 1035, “Domain Names - Implementation and Specification”). One extensible record is the “text” (TXT) record, a flexible record into which any information can be placed and subsequently processed by other programs and scripts. “TXT RDATA format ... where: TXT-DATA [is] One or more <character-string>s. TXT RRs are used to hold descriptive text. The semantics of the text depends on the domain where it is found. ... **Text Record (TXT)** The Text record is used to define free-form information about the named object. Its format is simple ... Because of its free-form nature, the TXT record has been used over time for special purposes, such as providing input to locally developed scripts that collect domain information. An earlier version of BIND that runs on some Unix systems even uses these records to define security information.” EX1021 (Hunt), p. 366.

110. Putting it all together, consider a user using a Web browser to browse <http://www.google.com/some/index.html>. The user first types the URL into the browser. The browser then attempts to resolve the hostname [www.google.com](http://www.google.com/some/index.html) using the client’s DNS resolver libraries. These libraries parse the hostname into three parts, delimited by the ‘.’, and issue DNS queries to the right servers across the Internet. The resolver library consults and local host and site DNS caches for speedier resolution. Once the IP address is found, say 142.251.35.164, the Web browser attempts to open a TCP/IP connection to the Web server at that IP address, using default port 80, using the HTTP protocol. If successful, the browser will issue an HTTP command such

as “GET /some/index.html” to request the retrieval of the server’s content for the file “/some/index.html”. Once that content arrives back at the browser, it translates the content as HTML and displays it on the browser (*e.g.*, text, graphics, hyper-links, and more).

111. Web browsers may be configured to contact a local Web cache or proxy, rather than contacting the actual remote server (*e.g.*, www.google.com) directly. In that case, the Web proxy appears to the Web client as any compliant Web server. In turn, the proxy acts as a Web client and communicates with the actual remote Web server, who may be unaware that it is being contacted by a proxy on behalf of a user’s Web browser.

112. Hostnames on the internet can be assigned a fixed (static) IP address or get one assigned dynamically. “DHCP” is “[a]cronym for **D**ynamic **H**ost **C**onfiguration **P**rotocol. A TCP/IP protocol that enables a network connected to the Internet to assign a temporary IP address to a host automatically when the host connects to the network. *See also* IP address, TCP/IP. *Compare* dynamic SLIP.” EX1013 (Microsoft Computer Dictionary), pp. 142, 166.

113. The DHCP protocol supports multiple messages to allow a client to join and leave an IP network: DHCPDISCOVER lets a “Client broadcast to locate available servers”; DHCPOFFER is a message from “[s]erver to client in response to DHCPDISCOVER with offer of configuration parameters”; DHCPACK is an acknowledgement message from the “[s]erver to client with configuration parameters, including committed network address”; and DHCPRELEASE is a message from “[c]lient to server relinquishing network address and cancelling remaining lease.” EX1020 (RFC-2131), p. 14.

114. “Network Address Translation” or “NAT” is an “[a]cronym for **n**etwork **a**ddress **t**ranslation. The process of converting between IP addresses used within an intranet or other private network (called a stub domain) and Internet IP addresses. This approach makes it possible

to use a large number of addresses within the stub domain without depleting the limited number of available numeric Internet IP addresses. *See also* intranet, IP address.” EX1013 (Microsoft Computer Dictionary), pp. 323, 327.

115. On the Internet, IP and other packets can be wrapped inside other packets, enabling one protocol to be overlayed on top of other protocol. To “tunnel” is “[t]o encapsulate or wrap a packet or a message from one protocol in the packet for another. The wrapped packet is then transmitted over a network via the protocol of the wrapper. This method of packet transmission is used to avoid protocol restrictions. *See also* communications protocol, packet (definition 2).” EX1013 (Microsoft Computer Dictionary), p. 479.

116. The IETF standardized various forms of IP packet encapsulation, as they are often useful for security purposes (*e.g.*, “IP Security”). “The Encapsulating Security Payload (ESP) header is designed to provide a mix of security services in IPv4 and IPv6. ESP may be applied alone, in combination with the IP Authentication Header (AH) [KA97b], or in a nested fashion, *e.g.*, through the use of tunnel mode (see ‘Security Architecture for the Internet Protocol’ [KA97a], hereafter referred to as the Security Architecture document). Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host. ... The ESP header is inserted after the IP header and before the upper layer protocol header (transport mode) or before an encapsulated IP header (tunnel mode).” EX1019 (RFC 2406, IP Encapsulating Security Payload), p. 2.

C. Virtualization

117. Among computer scientists, there is an axiom stating that “you can solve almost any problem by adding another level of indirection.” Indeed, virtualization is an indirection technique that makes one thing seem like another; virtualization often presents a physical-world

artifact as one that appears as an illusion or “fake.” For something to be “virtual”, it is “[o]f or pertaining to a device, service, or sensory input that is perceived to be what it is not in actuality, usually as more ‘real’ or concrete than it actually is.” EX1013 (Microsoft Computer Dictionary), p. 497.

118. Indeed, the Microsoft Computer Dictionary includes 35 examples and definitions of virtualization. EX1013 (Microsoft Computer Dictionary), pp. 497-500.

119. In one example, a “virtual machine” is “[s]oftware that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola chip. *Acronym: VM.*” EX1013 (Microsoft Computer Dictionary), p. 498.

120. In another example, “virtual memory” is “[m]emory that appears to an application to be larger and more uniform than it is. Virtual memory may be partially simulated by secondary storage such as a hard disk. Applications access memory through virtual addresses, are translated (mapped) by special hardware and software onto physical addresses. *Acronym: VM. Also called disk memory. See also paging, segmentation.*” EX1013 (Microsoft Computer Dictionary), p. 498.

121. In yet another example, “virtual reality” is “[a] simulated 3-D environment that a user can experience and manipulate as if it were physical. The user sees the environment on display screens, possibly mounted in a special pair of goggles. Special input devices, such as gloves or suits fitted with motion sensors, detect the user’s actions. *Acronym: VR.*” EX1013 (Microsoft Computer Dictionary), p. 499.

122. Virtualization had been applied to storage systems, disk devices, and drivers. A “virtual device driver” is “[s]oftware in Windows 95 that manages a hardware or software system resource. If a resource retains information from one access to the next that affects the way it

behaves when accessed (for example, a disk controller with its status information and buffers), a virtual device driver must exist for it.” EX1013 (Microsoft Computer Dictionary), p. 498.

123. A “virtual disk” or “RAM disk” is “[s]hort for random access memory disk. A simulated disk drive whose data is actually stored in RAM memory. A special program allows the operating system to read from and write to the simulated device as if it were a disk drive. RAM disks are extremely fast, but they require that system memory be given up for their use. Also, RAM disks usually use volatile memory, so the data stored on them disappears when power is turned off. Many portables offer RAM disks that use battery-backed CMOS RAM to avoid this problem. *See also* CMOS RAM. *Compare* disk cache.” EX1013 (Microsoft Computer Dictionary), pp. 395-396, 498.

124. Virtualization is sometimes said to be “logical” representation of physical reality. “Logical” is “[o]f or pertaining to a conceptual piece of equipment or frame of reference, regardless of how it may be realized physically. *Compare* physical.” EX1013 (Microsoft Computer Dictionary), p. 288 (“logical” definition 2).

125. A “logical drive” or “logical device” is “[a] device named by the logic of a software system, regardless of its physical relationship to the system. For example, a single floppy disk drive can simultaneously be, to the MS-DOS operating system, both logical drive A and drive B.” EX1013 (Microsoft Computer Dictionary), p. 288.

126. Moreover, one can virtualize files and file systems. This was the subject of much of my graduate-level research and became part of my PhD Dissertation. *See* §II above and EX1005 (Zadok Curriculum Vitae).

127. Virtualization, of course, has been applied to networking as well. A “virtual network” is “[a] part of a network that appears to a user to be a network of its own. For example,

an Internet service provider can set up multiple domains on a single HTTP server so that each one can be addressed with its company's registered domain name. *See also* domain name, HTTP server (definition 1), ISP.” EX1013 (Microsoft Computed Dictionary), p. 499.

128. A “virtual LAN” is “[s]hort for **virtual local area network**. A local area network consisting of groups of hosts that are on physically different segments but that communicate as though they were on the same wire. *See also* LAN.” EX1013 (Microsoft Computed Dictionary), p. 498.

129. One of the most famous uses of a (secure) virtual network is a private one. A “virtual private network” (VPN) is “[a] set of nodes on a public network such as the Internet that communicate among themselves using encryption technology so that their messages are as safe from being intercepted and understood by unauthorized users as if the nodes were connected by private lines. 2. A wide area network formed of permanent virtual circuits (PVCs) on another network, especially a network using technologies such as ATM or frame relay. *Acronym*: VPN. *See also* ATM (definition 1), frame relay, PVC.” EX1013 (Microsoft Computed Dictionary), p. 499. A POSA would have understood definition 2 of VPN to be a form of an overlay network because it is a collection of individual virtual network segments.

VIII. THE '785 PATENT

A. Overview

130. The claims of the '785 patent relate to a “virtual network manager” or “virtual network system” that includes several key components: (1) a computing device or “network interface configured for data communication,” (2) a “register module configured to register devices in a virtual network,” (3) a router or route director, and (4) a “DNS server configured to receive a DNS request” from a first (source) device and return three addresses to the source, including “a

network address associated with a network route director, a private network address associated with a second [destination] device in the virtual network, and a virtual network address associated with the second [destination] device.” EX1001 at 36:37-56 (claim 30).

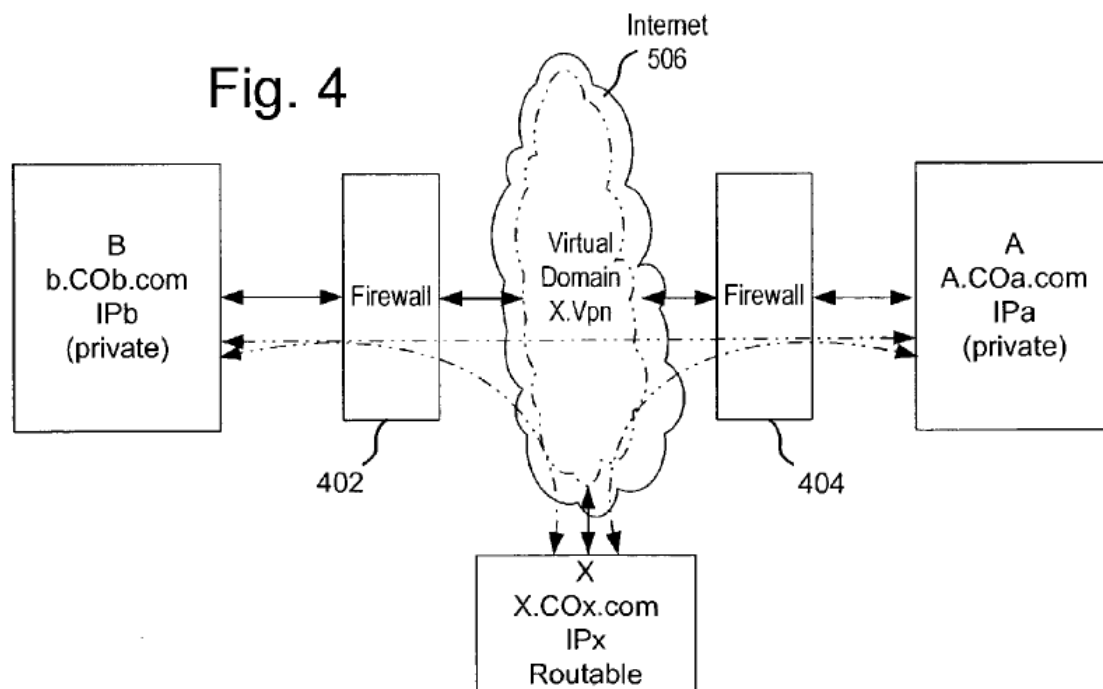
131. The ’785 patent identifies the so-called IP address depletion problem as one of the challenges it was trying to solve in order to “enable the rapid creation of secure means that allows local and remote specified entities [computer devices] to communicate and collaborate from any location via a standard Internet connection.” *Id.* at 1:32-37. The ’785 patent indicates that, due to its own popularity, the Internet is “running out of [IP] addresses” that can be assigned to each device, thereby limiting the ability for devices in private networks, for example, to communicate or be accessible via the Internet. *Id.* at 1:50-55. The ’785 patent also purports to address the “mobility problem,” involving devices being assigned constantly changing or dynamic IP addresses, which other devices on the Internet wishing to communicate with them might find difficult to track. *Id.* The ’785 patent thus describes “a need for a system that provides for local and remote entities to communicate and collaborate using the Internet, [a system that] can work with existing NAT devices and firewalls, and allows for devices to move to different physical networks.” *Id.* at 2:45-48.

132. In the “Description of the Related Art” section, the ’785 admits that NAT solutions were well known the art, and used to transparently translate between private and public IP addresses.

133. “One popular solution to the depleting address problem is Network Address Translation (NAT). This concept includes predefining a number of network addresses to be private addresses and public addresses. Public addresses are unique addresses that should only be used by one entity having access to the Internet. That is, no two entities on the Internet should have the

same public address. Private addresses are not unique and are typically used for entities not having direct access to the Internet. Private addresses are used in private networks. Private addresses can be used by more than one organization or network. NAT assumes that all of the machines on a network will not need to access the Internet at all times. Therefore, there is no need for each machine to have a public address. A local network can function with a small number of one or more public addresses assigned to a NAT device. The remainder of the machines on the network will be assigned private addresses. Since entities on the network have private addresses, the network is considered to be a private network.” EX1001, 1:36-2:7.

134. One embodiment of the 785 patent discloses a “secure Virtual Community Network or (‘VCN’),” described as “a private dynamic network which acts a private LAN for computing devices coupled to public networks or private networks.” *Id.* at 8:66-9:5, Fig.4.



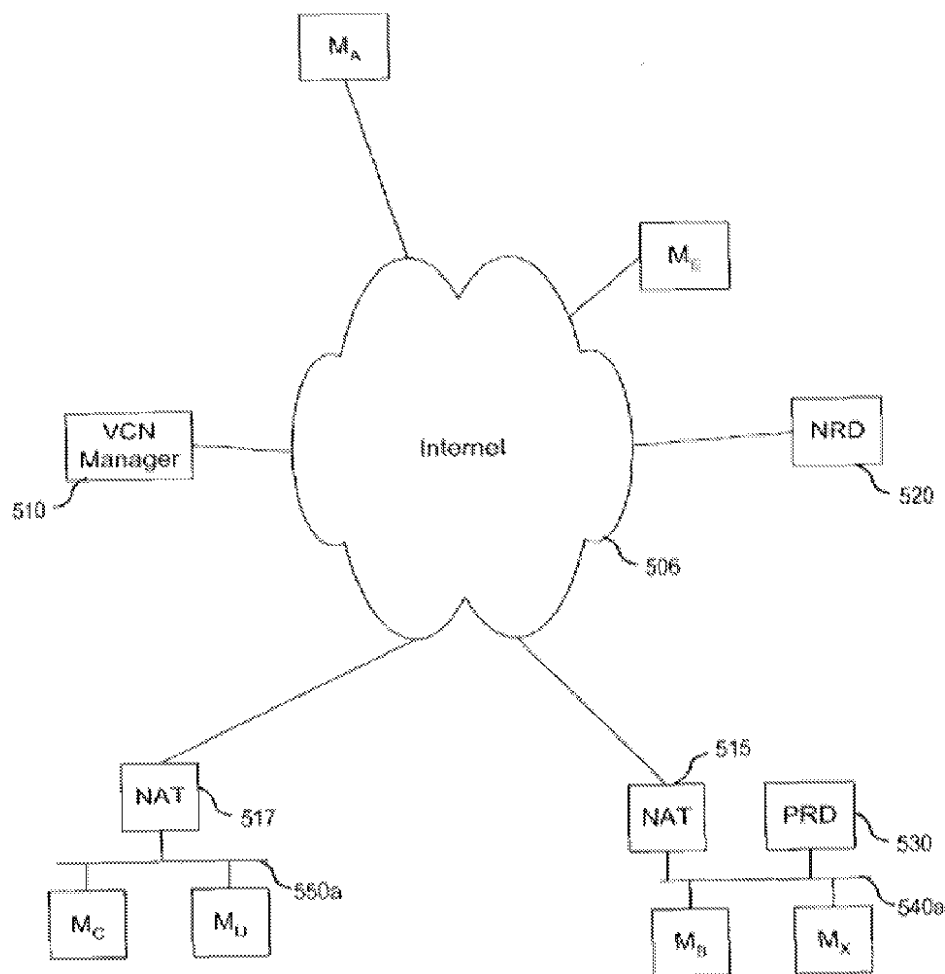
785 patent, Fig. 4.

135. Figure 4 shows an exemplary system or VCN, which comprises a “computer or device B (host name—b.Cob.com) in a first private domain and computer or device A (host

name—a.Coa.com) in a second private domain, both of which are coupled to the Internet by firewall devices **402, 404.**” *Id.* at 9:9-13. The firewall devices are Network Address Translation (“NAT”) devices. *Id.* at 9:13-15. Moreover, “[c]omputer or device X is coupled directly to the Internet and has a public IP address.” *Id.* at 9:16-17. “Devices A, B, and X can join the VCN, leave the VCN, or allows other devices in the VCN to communicate with them.” *Id.* at 9:18-23. “All members of the VCN are accessible as if they were part of a physical local network” (*e.g.*, all in the same private network)—allowing direct communication between devices within the virtual domain. *Id.* at 9:22-27.

136. The ’785 patent specifies that “hardware architecture for the machines, servers, or other devices used to implement the present invention should be well understood to one of ordinary skill in the art,” and outlines suitable hardware or software components, including “one or more processors, a memory, a mass storage device, a portable storage device, a first network interface, a second network interface, and I/O devices, in communication with each other.” *Id.* at 9:42-48. The ’785 patent acknowledges those are conventional computer components, noting that “the memory can be any conventional computer memory,” “[t]he choice of processor is not critical,” and “[t]he network interface can include a network card for connecting to an Ethernet or other type of LAN.” *Id.* at 9:48-58. The patent also discloses the network interface can include or be connected to a firewall. *Id.* The network interfaces are inside a computer and when the computer also functions as a “router,” it “includes two or more network interfaces.” *Id.* However, “in other embodiments, the computer may include only one network interface.” *Id.*

137. Another example of the system is illustrated in Figure 5A.



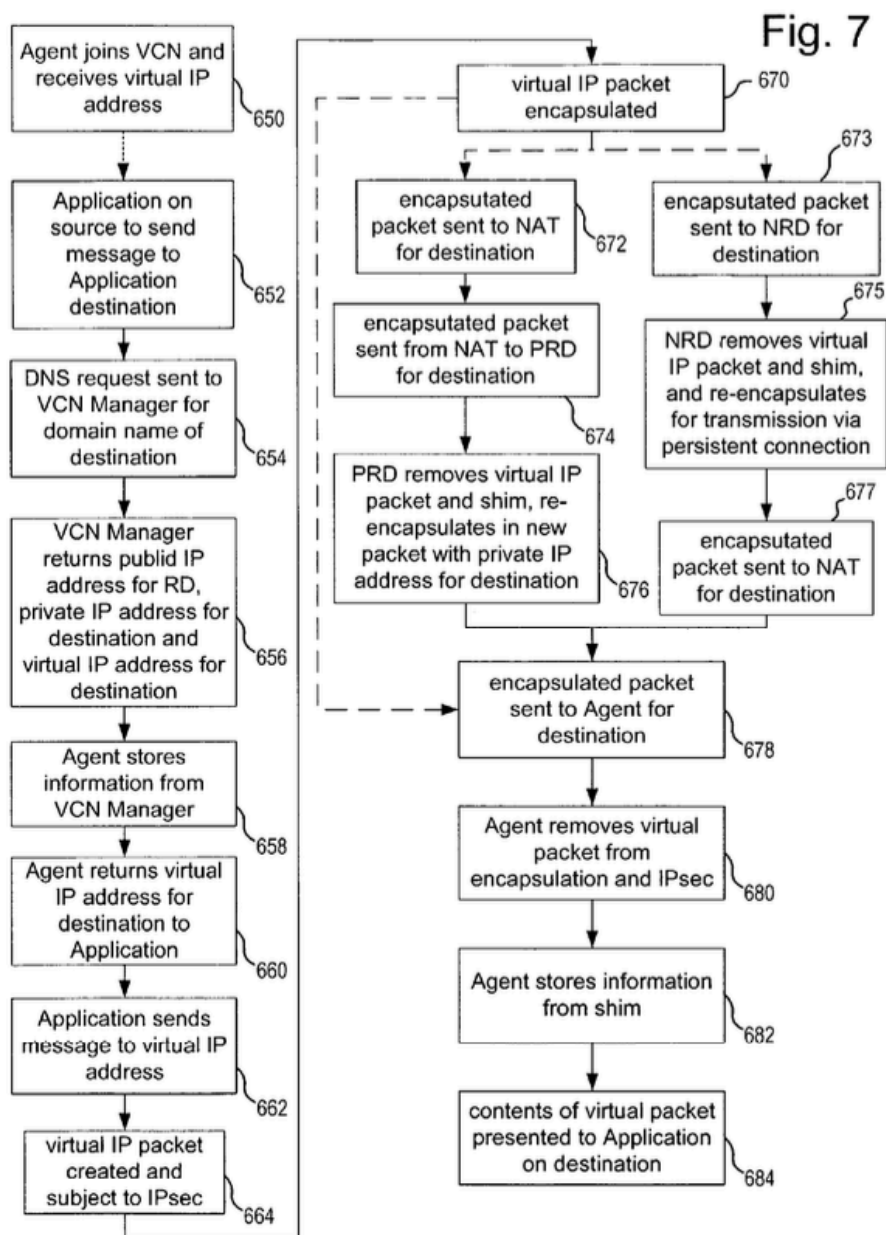
3

Id. at Fig. 5A.

138. In Figure 5A, the VCN includes a VCN Manager **510**, a Network Route Director **520** and/or Private Route Director **530**, and a number of user devices (*e.g.*, M_C , M_X , M_A , etc.). *Id.* at 10:24-27. The '785 patent explains that “[t]he VCN Manager **510** is a central server or server cluster providing management, connection, and security services for the VCN.” *Id.* at 10:66-11:1. The “various user machines, M_n , are coupled to the Internet **506**, as well as other devices.” *Id.* at 10:17-20. “In order to establish or participate in communication within the VCN, a device must register with and join the VCN, thereby becoming a member.” To join, “[t]he Agent will send a

message to the VCN Manager 510 and receive a virtual IP address in return.” *Id.* at 14:10-15.

This is shown in step 650 of Figure 7. *Id.*



785 patent, Fig. 7.

139. The '785 patent describes the virtual IP or network address as preferably another private network address that is chosen in a well-defined set of addresses (virtual address realm) such that each virtual network address occurs only once across the virtual network and thus is a

unique identifier of the device within the network address. EX1001, 12:50-54. To that end, the '785 Patent describes this virtual network address in open terms, adding that it can be a string, an IPV4, or IPV6 compliant address. *Id.*, 12:40-44.

140. A new member device wanting to send a message to another device in the VCN “will initiate a DNS request,” which is then sent to the VCN Manager **510**, as shown in step 654. *Id.* at 14:32-34. In response, the “VCN Manager 510 returns the public address of the Route Director for the destination, the private address for the destination device, and a virtual IP address for the destination device,” as shown in Step 656. *Id.* at 14:34-38. This information is then stored in a table, or any other data structure” for creating a virtual IP packet to be forwarded to the destination address. *Id.* at 14:48-15:15. When the destination wishes to respond to the source or the source wishes to send additional information, the process of Fig. 7 can be repeated, although the steps of performing the DNS request do not need to be performed again, since the information between the two devices has already been stored. *Id.* at 15:58-67.

141. Finally, I note that the '785 specification describes that the three addresses are not always returned, for instance, in the case where “the destination machine has a public address (and, thus does not use a routing director) then step **656** will only include returning the public IP address for the destination.” *Id.* at 14:40-44.

B. Prosecution History

142. The '785 patent application was filed on March 31, 2003 with a total of 109 claims. During prosecution, the examiner rejected the '785 patent claims on § 101, 102 and 103 grounds based on several prior art references. EX1002 ('785 FH), 145 (non-final rejection); 202 (final office action); 393 (non-final rejection); 613 (final office action); 689 (rejection).

143. Of particular relevance to allowance, in an office action dated November 22, 2006, the examiner rejected all claims as anticipated by U.S. Patent Publication No. 2003/0041136 (Cheline). *Id.* at 143-162. In response, the applicant amended the claims by adding specifically that the network traffic router “uses router data to route traffic between members of the virtual community,” and that “a first member of the virtual community communicates with another member in the virtual community by querying the DNS server to determine a virtual network address and a private network address for the other member, and to determine a public address of the network traffic router.” *Id.* at 172-187.

144. In its response, the applicant argued that “the client 102(1) of Cheline does not communicate with the DNS server 120 to obtain a virtual network address and private network address of another client, such as client 102(2), or to obtain a public network address of the service provider system 146, which is asserted by the Examiner to be a network traffic router as claimed.” *Id.* at 190-191.

145. On September 1, 2010, the examiner issued another office action, stating that claims 1-15, 17-24, 26-36, 41-43, and 93-109 were allowed, while claims 53-66 and 68-92 were rejected as anticipated by U.S. Patent No. 6,631,416 (Bendinelli). *Id.* at 613-618. The examiner’s reasons for allowance included the finding that “the prior art was not found to teach or suggest returning a public network address of a route director, a private network address for a destination device, and a virtual network address that corresponds to the destination device in response to a DNS request in the context of claims 1, 32, and 93,” and that “[s]pecifically, no examples were found in the prior art of record in which both a private address and a virtual address of a destination device would be returned.” *Id.* at 616-617. This determination by the examiner was critical to the claims of the ’785 patent being allowed. *Id.*

146. The applicant subsequently filed an amendment on October 21, 2010, revising some of the rejected claims (claims 53-55, 64, and 80) to include a similar DNS request and corresponding response with a triplet of addresses. *Id.* at 658-683.

147. Following another rejection by the examiner, this time rejecting claims 1-15, 17-24, 26-36, 41-43, and 80-92 as invalid under 35 U.S.C. § 101 while finding that the remaining claims contained allowable subject matter, the applicant filed another amendment clarifying that the recited virtual network manager and/or route director are implemented using a memory and a processor of a computing device to traverse the examiner's §101 rejection. *Id.* at 687-691. In response, the applicant filed an amendment on December 2, 2010, to clarify that the recited virtual network manager and/or route director are implemented using a memory and processor of a computing device. *Id.* at 723-750. A notice of allowance was then issued on January 14, 2011. *Id.* at 760-766.

IX. THE CLAIMS

148. The table below include identifiers for several representative independent claims and limitations of the '785 patent limitations, which are referenced throughout this Declaration.

Claims
[1.pre] A virtual network system, comprising:
[1.1] a virtual network manager implemented with a first device memory and a first device processor of a first computing device,
[1.2] the virtual network manager configured to register devices in a virtual network that is defined by a domain name,
[1.3] each device in the virtual network being identified to the other devices by a virtual network address that is unique for each device and not directly routable via a public network,

Claims
the virtual network manager further configured to distribute a virtual network address to a device when the device is registered in the virtual network;
[1.4] a route director implemented with a second device memory and a second device processor of a second computing device, the route director configured to communicate data between the devices that are registered in the virtual network, the data being communicated as encapsulated packets from a source device to a destination device, an encapsulated packet including a first virtual network address that corresponds to the source device and a second virtual network address that corresponds to the destination device; and
[1.5] the virtual network manager further configured to receive a DNS request from the source device, and return a public network address of the route director, a private network address for the destination device, and the second virtual network address that corresponds to the destination device.
[30.pre] A virtual network manager, comprising:
[30.1] a network interface configured for data communication via a virtual network that is defined by a domain name having an associated public network address;
[30.2] a memory and a processor to implement a register module configured to register devices in a virtual network, the register module further configured to:
[30.3] receive a registration request from an agent associated with a device; distribute a virtual network address to the device when the device is registered in the virtual network, the device being identified to other devices in the virtual network by the virtual network address; and
[30.4] a DNS server for the virtual network, the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a network route director, a private network address associated with a second device in the virtual network, and a virtual network address associated with the second device.
38. [38.pre] A virtual network system, comprising:
[38.1] a computing device that includes at least a memory and a processor configured to implement a network manager of a virtual network that is defined by a public domain name,

Claims
[38.2] the network manager configured to distribute virtual network addresses to devices that register as members in the virtual network, each device in the virtual network being identified to the other devices by a virtual network address associated with the device;
[38.3] a first virtual network agent associated with a first device that is registered as a member in the virtual network; at least a second virtual network agent associated with at least a second device that is registered as a member in the virtual network;
[38.4] a route director configured to route communications between the first device and the at least second device in the virtual network via the respective first and second virtual network agents, the communications configured for routing as encapsulated packets that include a first virtual network address that is not directly routable corresponding to the first device and a second virtual network address that is not directly routable corresponding to the at least second device; and
[38.5] the network manager includes a DNS server configured to provide authoritative responses for DNS queries in the virtual network, the DNS server further configured to receive a DNS query from the first device and return a network address of the route director, a network address of the second device, and the virtual network address of the second device.
[48.0] A computer-implemented method, comprising:
[48.1] receiving registration requests from devices that request to be registered as members of a
[48.2] virtual network that is defined by a domain name having an associated public network address in a public network,
[48.3] each of the devices having an associated private network address; distributing a virtual network address to a device to register the device as a member in the virtual network, each device in the virtual network being identified to the other devices by the virtual network address that is associated with the device;
[48.4] routing communications between the devices that are registered in the virtual network, the communications being routed as encapsulated packets from a source device to a destination device, an encapsulated packet including a first virtual network address that corresponds to the

Claims
source device and a second virtual network address that corresponds to the destination device; and
[48.5] transmitting a response to a DNS request received from one of the devices that are the members in the virtual network, the response to the DNS request including a public network address of a route director that registers the devices, a public network address of the destination device, and the second virtual network address that corresponds to the destination device.
[62.0] One or more processor readable storage media devices comprising
[62.1] processor readable code that, if executed by a computer device, implements a virtual network manager to: receive registration requests from devices that request to be registered as members of
[62.2] a virtual network that is defined by a domain name having an associated public network address in a public network,
[62.3] each of the devices having an associated private network address; distribute a virtual network address to a device to register the device as a member in the virtual network, each device in the virtual network being identified to the other devices by the virtual network address that is associated with the device;
[62.4] manage communications routed between the devices that are registered in the virtual network, the communications routed as encapsulated packets from a source device to a destination device, an encapsulated packet including a first virtual network address that corresponds to the source device and a second virtual network address that corresponds to the destination device; and
[62.5] transmit a response to a DNS request received from one of the devices that are the members in the virtual network, the response to the DNS request including a public network address of the virtual network manager, a public network address of the destination device, and the second virtual network address that corresponds to the destination device.
[75.0] A virtual network system, comprising:
[75.1] a computing device that includes at least a memory and a processor configured to implement a virtual network manager having a network interface coupled to a virtual network,
[75.2] the virtual network manager including at least one virtual community definition that is defined by a domain name having an associated public network address and
[75.3] a user set of one or more devices that are registered in the virtual network, each device in the virtual network being identified to the other devices by a virtual network address that is associated with the device, the virtual network manager configured to exchange virtual network information with the one or more devices of the user set, the virtual network being

Claims
accessible by devices in the user set and devices outside of the user set, and the virtual network manager further configured to
[75.4] receive a DNS request from a source device, and return a public network address of a route director, a private network address for a destination device, and a virtual network address that corresponds to the destination device.

X. OVERVIEW OF THE PRIMARY PRIOR ART REFERENCES

149. I first note that the prior art references analyzed in the grounds below are analogous to the '785 patent because they are within the same field of endeavor and/or reasonably pertinent to one or more problems addressed by the '785 patent. For example, all related to network routing between multiple private/public IP addresses.

A. Mehta

150. Mehta is the publication of US patent application to Mehta, entitled "Method and System for Two-way initiated Data Communication with Wireless Devices." EX1006 (Mehta), cover. Mehta describes a system designed to enable secure, bi-directional communication between public network devices and wireless devices on private networks. *Id.*, ¶¶[0011], [0015]. This system or method involves an Address Management Proxy System (AMPS), which allocates temporary public IP addresses to wireless devices to facilitate communication across public and private networks without exposing private IP addresses. The system in Mehta includes modified DNS/API servers, Address Proxy/Routers, and an Address Management Data Server to dynamically assign public IP addresses to private network devices as needed. *Id.*, ¶¶[0011]-[0015].

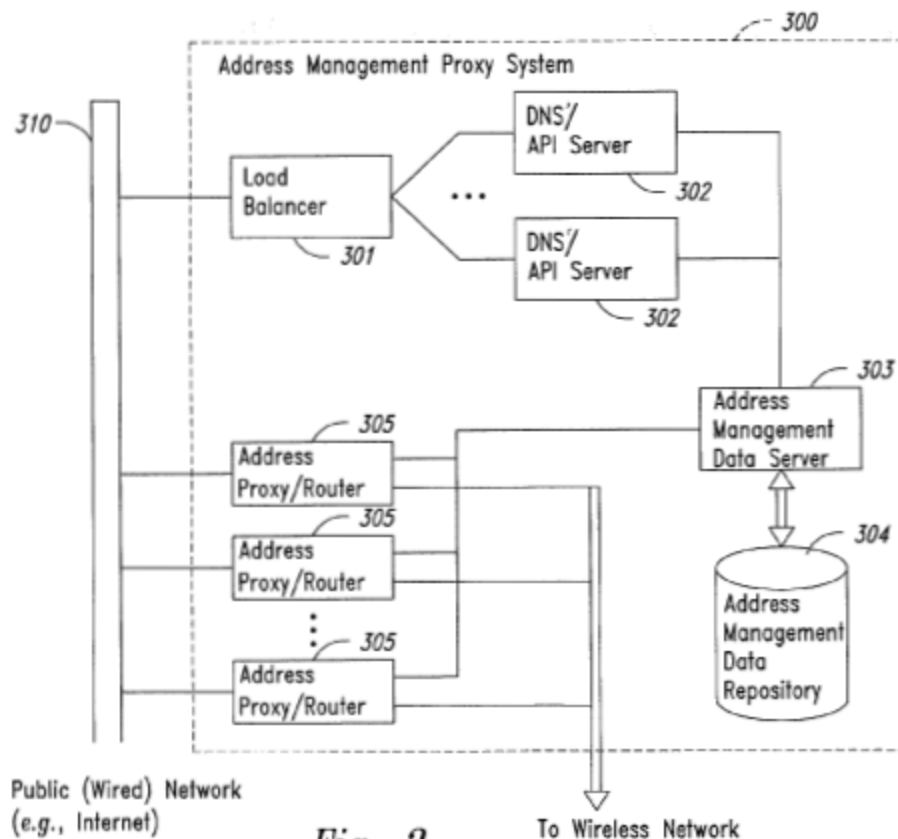


Fig. 3

151. AMPS uses techniques like Time to Live (TTL) for each IP mapping, ensuring temporary mappings that enhance security by automatically expiring connections. The system manages a pool of public addresses to allocate dynamically, using protocols like TCP/IP and UDP for seamless data exchange. AMPS can support various types of networks and devices beyond traditional IP networks, including ATM networks and other wireless technologies. Metha addresses the challenge of secure and efficient communication between disparate network types, enhancing connectivity options without compromising network security. *Id.*, ¶[0015].

B. RFC-1383

152. *Request For Comment 1383: An Experiment in DNS-Based IP Routing* by Christian Huitema outlines a proposal for utilizing Domain Name System (DNS) records for IP routing. The

approach described in the RFC seeks to help solve the issue of “routing explosion” on the internet caused by the increasing number of networks, resulting in larger routing tables, more data in route exchanges, and slower responses to routing events. EX1007 (RFC-1383) at 1.

153. In turn, RFC-1383 offered “a scheme that allows for simple routing,” and is “complementary with the classic hierarchical routing approach,” while providing “an easy to implement and low cost solution for ‘multi-homed’ domains.” *Id.*, 2. Specifically, the RFC-1383 solution is described as “a generalization of the ‘MX record’ scheme used for mail routing.” *Id.*

154. Similar to the aforementioned mail routing model that uses MX (Mail eXchange) records, the protocol introduces “RX records” for routing IP packets by associating domain names with preferred gateways by means of “source routing” or “tunneling,” or another form of encapsulation protocol. *Id.*, 4. To implement this routing, a modified DNS lookup is suggested: the source or an early relay makes the routing decision by querying the DNS for RX records, which list gateways with preference levels. This approach avoids hop-by-hop routing updates across the network and is scalable. *Id.*

155. RFC-1383 discusses mechanisms to use its DNS records. “Source directed routing, as described above, can be implemented through one of two techniques: source routing, or a form of encapsulation protocol. For the sake of simplicity, we will use source routing, as defined in [1]: we don’t have to define a particular tunnelling protocol, and we don’t have to require hosts to implement a particular encapsulation protocol.” *Id.*, 6. A POSA would have understood that when RFC-1383 returns one or more extra addresses in response to a DNS query, those could be easily used as the tunneling/encapsulation addresses to enable, for example, VPN and IPsec services. *See also* §VII.B above.

156. RFC-1383 describes a network interface integrated with a standard IP router, working alongside a DNS query manager. *Id.*, 8, 12. When an ICMP (Internet Control Message Protocol) message is received, the query manager updates the local routing table to ensure that any new packets bound for the specified destination are routed through the real-time forwarder. *Id.*, 8. Simultaneously, the query manager sends a DNS request to retrieve the RX records for the destination. *Id.*, 11. Upon receiving the response, it selects a gateway and provides this information to the real-time forwarder. *Id.*, 12.

157. When the real-time forwarder receives a packet, it checks if a gateway for the destination is available. *Id.*, 12-13. If so, it inserts the necessary source routing information and forwards the packet—either through the general IP routing program or directly to the network interface linked to the intermediate gateway.

158. Notably, RFC-1383 discloses (a) a way to store and return multiple IP addresses in response to a single query, and (b) identify an address of the destination in the domain part of the DNS request and response.

159. RFC-1383 proposes using “TXT” DNS records to store that information. “This [TXT] record is designed for easy general purpose extensions in the DNS, and its content is a text string. Each RX record will contain three fields:

- A record identifier, ‘RX,’ to distinguish it from other experimental uses of the ‘TXT’ record.
- A cost indicator, encoded on up to 3 numerical digits.
- An IP address, encoded as a text string following the ‘dot’ notation.” *Id.*, 11.

160. RFC-1383 shows an example record (id.):

domain	type	record	value
-			
*.27.32.192.in-addr.arpa	IP	TXT	RX, 10, 10.0.0.7

161. RFC-1383 explains the table shows: “that [] for all hosts whose IP address starts by the three octets ‘192.32.27’ the IP host ‘10.0.0.7’ can be used as a gateway, and that the preference value is 10.” EX1007 (RFC-1383), p. 12. The TXT records described in RFC-1383 are simply a domain field containing a destination address, and a string value comprising comma-delimited fields, including an IP address.

XI. Error! Reference source not found.VII.B aboveVII.B aboveVII.B aboveCLAIM CONSTRUCTION

162. For purposes of this declaration, I’ve been asked to assume that the time of the alleged invention is March 31, 2003.

A. Claim Construction Generally

163. I understand that, for purposes of my analysis in this inter partes review proceeding, the terms appearing in the patent claims should be interpreted according to their “ordinary and customary meaning” in view of the specification and prosecution history under the Phillips standard. See Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005); 37 C.F.R. § 42.100. In determining the ordinary and customary meaning, I understand that the words of a claim are first given their plain meaning that those words would have had to a POSA in the art at the time of the alleged invention. I also understand that the structure of the claims, the specification and file history may be used to better construe a claim. Moreover, I understand that even treatises and dictionaries may be used, albeit under limited circumstances, to determine the meaning attributed by a POSA in the art to a claim term.

B. Claim Construction Means Plus Function Terms

164. I understand that under the applicable version of 35 U.S.C. § 112 ¶6, “an element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.”

165. I understand that expressing an element in a claim in terms of a “means for ...” leads to a presumption that 35 U.S.C. § 112 ¶6 applies. I also understand that even if an element is not expressed using the term “means,” it may still be considered a means-plus-function limitation, if the term fails to “recite sufficiently definite structure” or else recites “function without reciting sufficient structure for performing that function.” That a claim term lacks the word “means” does not create a “strong” presumption that the claim term is not subject to 35 U.S.C. § 112 ¶6. The standard for determining whether the words of the claim are understood by POSA in the art to have a sufficiently definite meaning as the name for structure.

166. I understand that construing a claim limitation that is a means-plus-function term involves two steps. The first step is to identify the claimed function. The second step is to identify the structures that are described in the specification. I understand that the means-plus-function claim term is limited to those structures described in the specification that correspond to the claimed function, and their equivalents.

C. All Claims: “virtual network address”

167. All of the claims of the ’785 patent recite a “virtual network address” that is assigned to a device once it is registered in the virtual network. The ’785 patent explains a “virtual

network address” is a name or address that can be used to identify and send communications to other devices in the virtual network. See EX1001, 6:1-7, 31:24-43.

D. Claim 30, 34, 76 (“register module”)

168. Claims 30, 34, 76 (reproduced below) recite a “register module”² for performing certain functions. “Module” is a nonce word that substitutes for “means” for purposes of § 112, ¶6. See *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1350 (Fed. Cir. 2015) (en banc).

[30.2-30.4] a memory and a processor to implement] a register module configured to register devices in a virtual network, the register module further configured to: receive a registration request from an agent associated with a device; distribute a virtual network address to the device when the device is registered in the virtual network, the device being identified to other devices in the virtual network by the virtual network	Nonce word for “means” function without recitation of structure to perform it
[34] The virtual network manager of claim 30 wherein the register module is further configured to receive the registration request from the agent that is installed on the device for data communication via the virtual network.	

169. Means-plus-function claims are construed by identifying the claimed function and the corresponding structure in the specification for performing it. See *Williamson*, 792 F.3d at 1351. The chart below provides these constructions.

<u>Claims</u>	<u>Function</u>	<u>Structure in 785 Specification</u>
30	<u>“receiving registration request from device’s</u>	<u>“The Registration exchange between the member agent and the VCN Manager is shown in FIG. 12A.</u>

² In the litigation, Patent Owner asserts Claim 30 and reads the “registration module” very broadly against commonly and widely used open-source software known as Kubernetes. See EX1022 and EX1023 at Claim 30. Patent Owner contends this term is not a means-plus-function limitation.

	<u>agent and distributing a virtual network address to the device when it registers with the virtual network.”</u>	<u>When a Member Agent 565 attempts to register at step 1200, Member Agent 565 will send a registration request packet 1202 to the VCN Manager 510 to start registration. The packet is included in an HTTP wrapper as illustrated at 1204. The registration packet will carry the client Fully Qualified Domain Name (FQDN) and a Diffie-Hellman Key Exchange request to the VCN Manager. The packet further includes packet version information, type and length information, a user authenticator, the length of FQDN in octets, a VCN name offset, the member's FQDN in DNS format, the length of Diffie-Hellman value in octets, and the members initial Diffie-Hellman value.”</u> <u>EX1001, 18:18-29.</u>
34	<u>“receiving registration request from the agent installed on device for data communication via the virtual network.”</u>	<u>Id.</u>

E. Claims 35-37, 77-78: “join module”

170. Claims 35-37, 77, and 78 recite a “join module” performing the below described functions. See Williamson, 792 F.3d at 1350 (“Module” is a nonce word that substitutes for “means” for purposes of § 112, ¶(6.)). Claim set 25-37, discussed below, is illustrative.

[35] The virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network. [36] The virtual network manager of claim 35 wherein the join module is further configured to provide virtual network addresses to the devices that are registered in the virtual network. [37] The virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network. [77] The system of claim 75 wherein the virtual network manager includes a member join module.	Nonce word for “means” function without recitation of structure to perform it
--	---

[78] The system of claim 77 wherein the member join module provides a virtual network address to a device that is registered as a member of the network.

171. Means-plus-function claims are construed by identifying the claimed function and the corresponding structure in the specification for performing it. See Williamson, 792 F.3d at 1351. The chart below provides these constructions.

<u>Claims</u>	<u>Function</u>	<u>Structure in 785 Specification</u>
35	“receiving join and leave requests from the device or its agent indicating that the device is connected or will be disconnected from data communication, respectively”	The Member Join process is illustrated in FIG. 13 and the leave process in FIG. 14. <pre> sequenceDiagram participant VCN as VCN Manager 510 participant MA as Member Agent 565 Note over MA: Member Attempts to Authenticate 1302 MA->>VCN: Init Request 1304 VCN->>MA: Init Nack 1310 Note over VCN: 1306 If packet abnormal, Drop 1308 If member fails authentication, send Init Nack 1312 If member passes authentication, send Init Ack Agent Behind NAT? (Fig. 21) MA->>VCN: Init Ack 1314 Note over VCN: 1320 If packet abnormal, Drop 1322 If Join fails, then send Join Nack 1326 If Join succeeds, send Join Ack MA->>VCN: Join Request (or Radius Join) 1318 VCN->>MA: Join Nack 1324 MA->>VCN: Join Ack 1326 Note over MA: Member now Joined 1330 </pre> Fig. 13 510 VCN Manager 565 Member Agent 1304 Init Request 1310 Init Nack 1314 Init Ack 1318 Join Request (or Radius Join) 1324 Join Nack 1326 Join Ack 1302 Member Attempts to Authenticate 1306 If packet abnormal, Drop 1308 If member fails authentication, send Init Nack 1312 If member passes authentication, send Init Ack Agent Behind NAT? (Fig. 21) 1320 If packet abnormal, Drop 1322 If Join fails, then send Join Nack 1326 If Join succeeds, send Join Ack 1330 Member now Joined

		Fig. 14 <pre> sequenceDiagram participant VCNManager as VCN Manager 510 participant MemberAgent as Member Agent 565 Note over VCNManager: 1414 If packet abnormal, Drop Note over VCNManager: 1416 If member fails authentication or is not Joined, Send Leave Nack Note over VCNManager: 1420 If member passes authentication, was Joined and successfully left, Send Leave Ack MemberAgent->>VCNManager: 1412 Leave Request Note over MemberAgent: 1410 Member attempts to Leave VCNManager->>MemberAgent: 1418 Leave Nack VCNManager->>MemberAgent: 1430 Leave Ack Note over MemberAgent: 1440 Member successfully Leaves </pre> <i>See also EX1001, 20-21 (for description of figures).</i>
36, 78	“providing virtual network addresses to the devices that are registered in the virtual network”	<i>Id.</i>
37	“maintaining data to associate a virtual network address with a device in the virtual network.”	<i>Id.</i>
77	<i>N/A</i>	

F. [RESERVED]

172. [RESERVED]

G. [RESERVED]

173. [RESERVED]

H. Other Claims

174. No express construction of the other challenged claims is necessary in this IPR because the prior art renders these claims obvious under any reasonable construction.³

175. For purposes of this declaration, I've been asked to assume that the time of the alleged invention is March 31, 2003.

XII. GROUND 1: Claims 1-2, 6-12, 22, 25-26, 28-49, 51-55, 60-63, 65-66, 72-80, and 82-90 are obvious in view of Mehta alone or with RFC-1383

A. The Independent Claims

176. The six independent claims of the '785 patent (claims 1, 30, 38, 48, 62, and 75) recite substantially overlapping subject matter. For ease of discussion, I address the independent claims together. Furthermore, I group related limitations together.

177. These claims generally disclose a virtual network system or manager, comprising a virtual network with a public domain name, a registration feature or module to register devices on the network, a route director to route data traffic between the devices on the network and a DNS server to exchange network addresses of the devices on the network upon request. These limitations, as shown further below, would have been obvious to a POSA before the time of filing of the '785 patent, in light of the disclosures in Mehta and RFC-1383.

³ Petitioners do not concede no construction of the claims may ever be necessary or any other claim construction argument. Should any subsequent discussion of claim interpretation arise here or in the parallel district court proceeding, Petitioners reserve the right to propose constructions and respond to any construction advanced by Patent Owner, the Board, and/or the District Court.

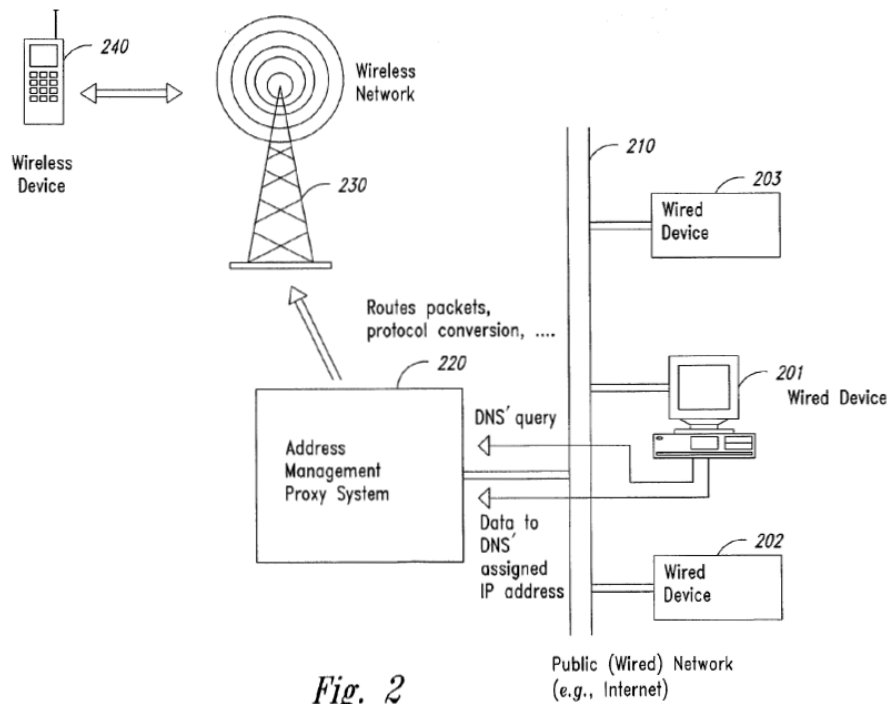
1. Preamble: Virtual network system or manager

Claim	Limitations
[1.0, 1.1]	A virtual network system, comprising a virtual network manager
[30.0]	A virtual network manager, comprising:
[38.0]	A virtual network system, comprising:
[48.0]	A computer-implemented method, comprising:
[62.0, 62.1]	One or more processor readable storage media devices comprising processor readable code that, if executed by a computer device, implements a virtual network manager
[75.0]	A virtual network system, comprising:

178. I note that Mehta discloses “a computer-readable memory medium containing instruction for controlling a processor in a computer system to establish a bi-directional [] virtual communication channel between a first and second device.”

179. I further note that the '785 patent explains that “[t]he VCN Manager 510 is a central server or server cluster providing management, connection, and security services for the VCN.” EX1001 at 10:66-11:1.

180. The preambles of the independent claims, to the extent limiting, recite a virtual network system or manager. Mehta discloses an Address Management Proxy System (“AMPS”) which is a virtual network system designed to manage communications between “a device on a private network [initiated] from a device on a public network to achieve virtual end-to-end connectivity.” EX1006 (Mehta) ¶[0002].



2. “A virtual network defined by a domain name”

Claim	Limitations
[1.2]	the virtual network manager configured to register devices in a virtual network that is defined by a domain name
[30.1]	a network interface configured for data communication via a virtual network that is defined by a domain name having an associated public network address
[38.1]	a network manager of a virtual network that is defined by a public domain name
[48.2]	a virtual network that is defined by a domain name having an associated public network address in a public network
[62.2]	a virtual network that is defined by a domain name having an associated public network address in a public network
[75.1]	a network interface coupled to a virtual network, the virtual network manager including at least one virtual community definition that is defined by a domain name having an associated public network address

181. Mehta explicitly discloses “[e]ach proxy/router machine has a preconfigured set of public network addresses, such as are typically configured by **network cards** [network interface]

inserted into the proxy/router machine.” EX1006 (Mehta) ¶[0055] (emphasis added). Network interfaces, which are points of interaction between a device (like a computer or server) and a network are also inherently disclosed in Figs. 1 and 2 of Mehta. Network interfaces can be either physical (e.g., Ethernet ports in LAN) or virtual (e.g., virtual network card, or NCI). *See also* §**Error! Reference source not found.**

182. Mehta discloses the AMPS (virtual network system/manager) enabling “devices and systems connected to a public internet.” EX1006 (Mehta) ¶[0024]. A POSA would have understood that for any system that purports to connect to a computer network, it must include devices (e.g., Ethernet or Wi-Fi Network Interface Cards or NICs, dialup modems, and the like). *See* §**Error! Reference source not found.** *See also id.* at ¶[0033], ¶[0055]. The AMPS has a “public network address” [*see* cl. 5] and includes a DNS or domain name server with a domain name for the network and associated public IP address. *Id.* at ¶[0012] (“the AMPS comprises one or more modified DNS’/API servers.”).

3. Registration/distribution of virtual network address to each device which uniquely identifies the device

Claim	Limitations
[1.2]	the virtual network manager configured to register devices in a virtual network that is defined by a domain name, each device in the virtual network being identified to the other devices by a virtual network address that is unique for each device and not directly routable via a public network, ⁴ the virtual network

⁴ The ’785 patent describes the virtual network address in broad terms, adding that it can be any address compliant with the IPv4 standard, provided that it “will not be routable to the member.” EX1001, 12:40-44 (“These virtual addresses may be any legal IPv4 addresses.”); *Id.* 12:48-50 (“In general, when choosing virtual addresses it is best to choose addresses that will not be routable to the member.”). Any POSA would understand “not directly routable via a public network” to mean

Claim	Limitations
	manager further configured to distribute a virtual network address to a device when the device is registered in the virtual network;
[30.2-30.4]	[a memory and a processor to implement] a register module configured to register devices in a virtual network , the register module further configured to: receive a registration request from an agent associated with a device; distribute a virtual network address to the device when the device is registered in the virtual network, the device being identified to other devices in the virtual network by the virtual network
[38.2]	the network manager configured to distribute virtual network addresses to devices that register as members in the virtual network ,
[48.1]	receiving registration requests from devices that request to be registered as members of a virtual network that is defined by a domain name having an associated public network address in a public network, each of the devices having an associated private network address; distributing a virtual network address to a device to register the device as a member in the virtual network,
[62.1]	receive registration requests from devices that request to be registered as members of a virtual network that is defined by a domain name having an associated

not directly routable via a public network **to the member**, because an address that is routable in the abstract is not necessarily directly routable with respect to a specific device. The temporary public addresses that Mehta discloses are from a pre-assigned pool of public addresses that are dynamically allocated to devices as required, and unusable when not allocated. EX1006, Abstract, (“[A] pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to [] network devices as required.”); EX1006, [0056] (“...returning the associated public address to the pool of unused public network addresses.”); *Id.* [0015] (“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection.”). Any POSA would understand that unused public network addresses are not directly routable to any destination device.

Claim	Limitations
	public network address in a public network, each of the devices having an associated private network address; distribute a virtual network address to a device to register the device as a member in the virtual network,
[75.3]	a user set of one or more devices that are registered in the virtual network, each device in the virtual network being identified to the other devices by a virtual network address that is associated with the device

183. I note that the language in claim 48.1, “each of the devices having an associated private network address,” to the extent limiting, is inherent in all of the independent claims. I discuss this further with the DNS limitation in §XII.A.5 below, which requires that DNS returns the private network address of a device.

184. Mehta discloses a process by which an embodiment of the invention “dynamically allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on” that uniquely identifies the device. *Id.* at ¶[0053]. “Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” *Id.* See also Fig. 3 (showing the Address Management Data Server 303 and Address Management Data Repository 304 where registration information for each device is stored). “The DNS’/API servers 302 use the Address Management Data Server 303 to assist in mapping a unique identifier [virtual network address] ... for a wireless device to a public address on public network 310.” *Id.* at ¶[0032].

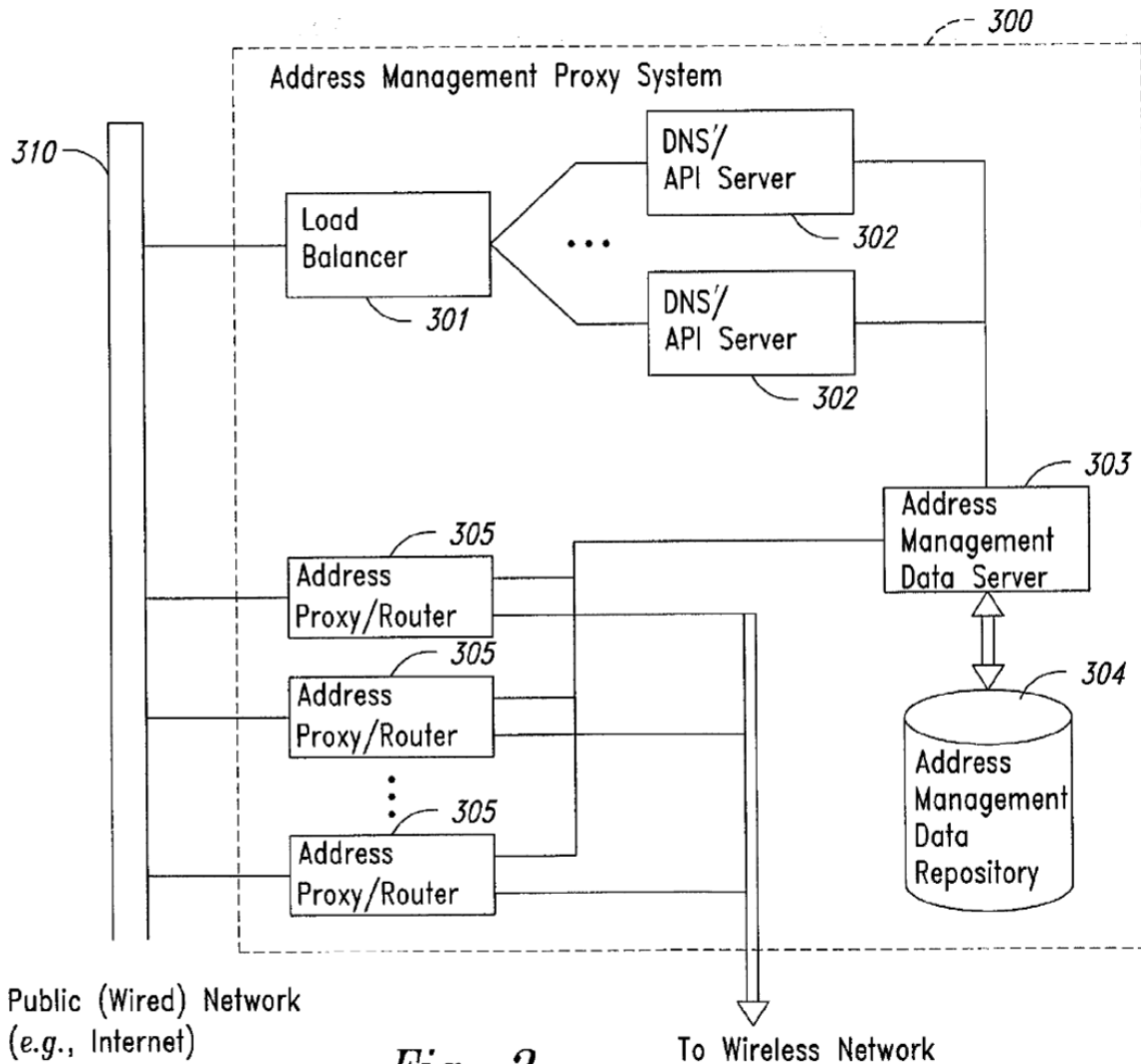


Fig. 3

4. Route director/routing

Claim	Limitations
[1.4]	a route director implemented with a second device memory and a second device processor of a second computing device, the route director configured to communicate data between the devices that are registered in the virtual network, the data being communicated as encapsulated packets from a source device to a destination device, an encapsulated packet including a first virtual network address that corresponds to the source device and a second virtual network address that corresponds to the destination device
[30.5]	the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a network route director

Claim	Limitations
[38.4]	a route director configured to route communications between the first device and the at least second device in the virtual network via the respective first and second virtual network agents, the communications configured for routing as encapsulated packets that include a first virtual network address that is not directly routable corresponding to the first device and a second virtual network address that is not directly routable corresponding to the at least second device;
[48.3]	routing communications between the devices that are registered in the virtual network, the communications being routed as encapsulated packets from a source device to a destination device, an encapsulated packet including a first virtual network address that corresponds to the source device and a second virtual network address that corresponds to the destination device
[62.4]	manage communications routed between the devices that are registered in the virtual network, the communications routed as encapsulated packets from a source device to a destination device, an encapsulated packet including a first virtual network address that corresponds to the source device and a second virtual network address that corresponds to the destination device
[75.4]	the virtual network manager further configured to receive a DNS request from a source device, and return a public network address of a route director

185. Mehta discloses a router, or route director (Address/Proxy/Router) for data communications between devices in the virtual network (AMPS). *Id.* at ¶[0012] (“the AMPS comprises one or more modified DNS’/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more Address Management Data Repositories, and optionally a load balancer.”) “The router (or routing server/service) contains a routing table, which determines to which machine (and optionally to which port) to send a datagram, given a destination IP address.” *Id.* at ¶[0008]. *See also id.* at ¶[0004], ¶[0010], ¶[0030] ¶[0032-0033], ¶[0036-0038], ¶[0048], ¶[0052], ¶[0055-0059].

a. The Encapsulated Packets in Claims [38.3], [48.3], and [62.4]

186. Furthermore, claim elements 38.3, 48.3 and 62.4 require that “the communications being routed in the virtual network as encapsulated packets” that include the virtual network addresses of the source and destination devices.

187. A POSA would have known that packet encapsulation was well understood and used by 2003. Virtual private networks (VPNs) existed, which utilize packet encapsulation (e.g., embedding one packet inside another). IP Security (IPSec) protocols existed to allow one to transmit secure, encrypted packets, embedded inside plain IP packets. *See* §VII.B above.

5. DNS server, request and responses

Claim	Limitations
[1.5]	the virtual network manager further configured to receive a DNS request from the source device, and return a public network address of the route director, a private network address for the destination device, and the second virtual network address that corresponds to the destination device
[30.5]	a DNS server for the virtual network, the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a network route director, a private network address associated with a second device in the virtual network, and a virtual network address associated with the second device
[38.4]	and the network manager includes a DNS server configured to provide authoritative responses for DNS queries in the virtual network, the DNS server further configured to receive a DNS query from the first device and return a network address of the route director, a network address of the second device, and the virtual network address of the second device
[48.5]	transmitting a response to a DNS request received from one of the devices that are the members in the virtual network, the response to the DNS request including a public network address of a route director that registers the devices, a public network address of the destination device, and the second virtual network address that corresponds to the destination device
[62.5]	transmit a response to a DNS request received from one of the devices that are the members in the virtual network, the response to the DNS request including a public network address of the virtual network manager, a public network address of the destination device, and the second virtual network address that corresponds to the destination device
[75.4]	the virtual network manager further configured to receive a DNS request from a source device, and return a public network address of a route director, a private network address for a destination device, and a virtual network address that corresponds to the destination device

188. In Mehta, each (wireless) device has a private IP address and a Unique ID assigned by a (e.g., cellular) carrier.

189. “The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier’s infrastructure. In some embodiments, the carrier infrastructure dynamically allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on.” EX1006 (Mehta), ¶[0053], Fig. 6 (reproduced below).

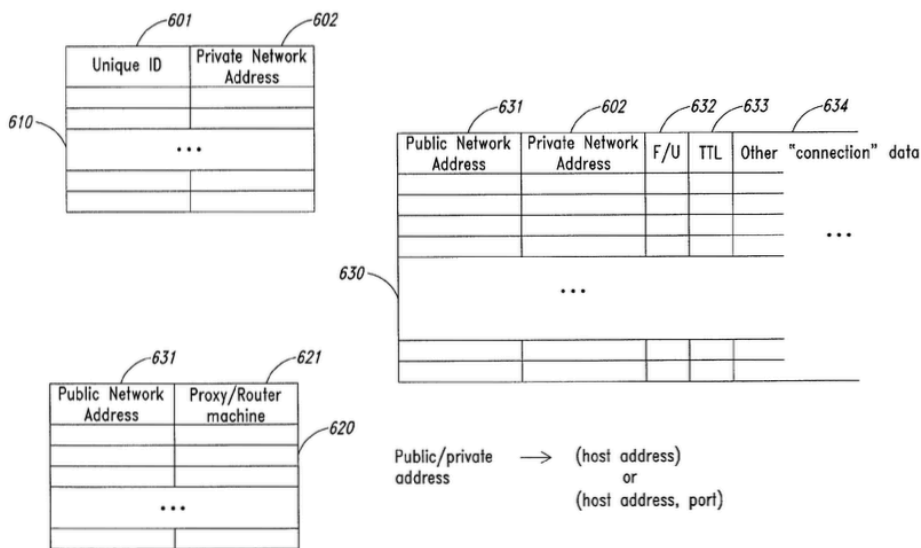


Fig. 6

190. For example, Table 610 may include Unique ID for “Alice’s iPhone” with Private Network Address 602 mapped to “192.1.2.3”.

191. Mehta has a set of Address Proxy/Router 305 machines, each of which can have multiple public addresses (facing the Internet) as well as multiple private address (facing the wireless network).

192. “Each proxy/router machine has a preconfigured set of public network addresses, such as are typically configured by network cards inserted into the proxy/router machine. These address[es] are allocated in a standard fashion through prior purchase or obtaining from an address authorizing authority, currently the Internet Corporation for Assigned Names and Numbers (ICANN). When a machine is inserted for use into the AMPS, indications of these addresses are stored in field 631.” EX1006 (Mehta), ¶[0055].

193. These proxy/router addresses are stored in tables 620 and 630 of Figure 6. In effect, there is a pool of available proxy/router machine addresses on standby, recorded in table 630. When a proxy/router address is not in use, or free, it is marked as (F)ree in flag 320 of table 630. When the entry is in use, it is marked as (U)sed in flag 320 of table 630.

194. “The public-to-private address table 630 comprises several fields/columns including a public network address 631, a private (wireless) network address 602, a flag 632 that specifies whether the public address stored in field 631 is free or is already used, a Time to Live (TTL) parameter 633, and other connection data 634. In one embodiment, the DNS’/API servers of the AMPS query table 630 to determine a public network address that corresponds to a designated private wireless network address or to allocate an unused public network address (as indicated in field 632) and map the determined unused public address to a private network address stored in field 602.” EX1006 (Mehta), ¶[0054].

195. Indeed, this pool of public addresses, while always available, is temporarily assigned as needed, for routing to a private wireless device, and using a time-to-live (TTL) expiration time (hence temporary).

196. “The AMPS allocates a public (routable) network address for temporary use by a requesting device on an external public network to communicate with a wireless device on a

private wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required.” EX1006 (Mehta), ¶[0011].

197. Figure 7 (reproduced below) illustrates all these steps.

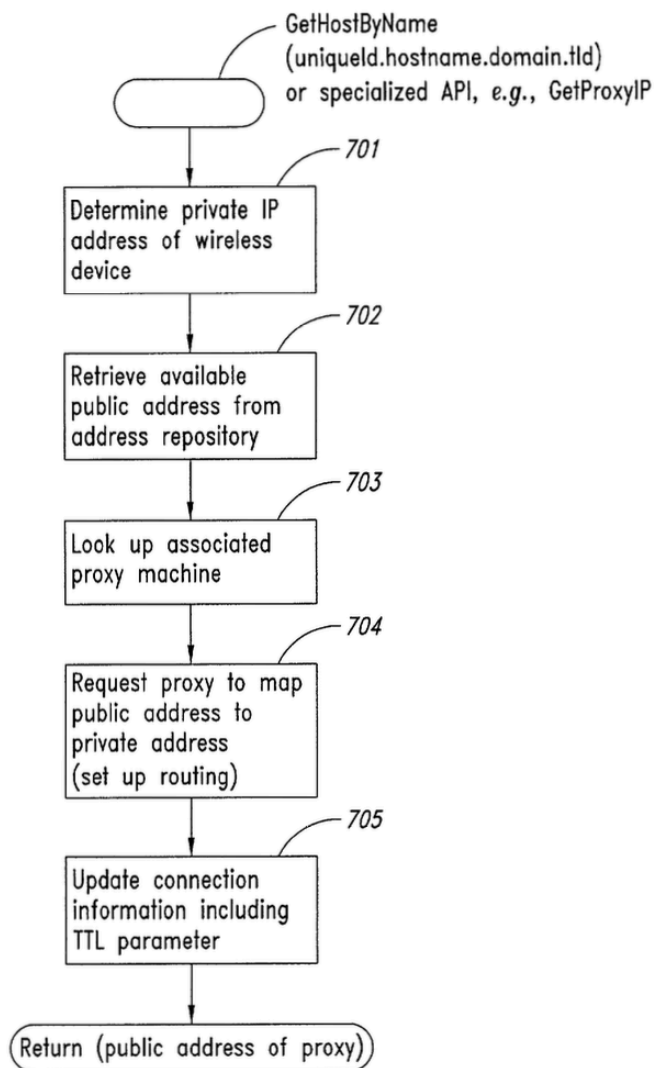


Fig. 7

198. Fig. 7 begins by responding to a DNS query `GetHostByName`. “FIG. 7 is an example flow diagram of an example routine provided by a DNS’/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier.” EX1006 (Mehta), ¶[0053].

199. This is clearly a DNS query. “In essence, this routine implements a DNS query or DNS-like query capability for the AMPS using a modified `GetHostByName` interface or a specialized API, for example `GetProxyIP`, as described with reference to FIG. 5.” *Id.*

200. This “routine dynamically allocates an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters).” *Id.* That is, one of the available public proxy/router addresses in the pool is being consumed temporarily for routing to a private wireless address.

201. “Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine.” *Id.* This is accomplished by consulting table 610.

202. “For example, the string parameter may use fields such as ‘uniqueID.hostname.domain.tld,’ which specifies a typical hierarchy of person/service on a machine named ‘hostname’ on a domain such as a company’s network on a top level domain such as ‘org.’ ‘com,’ ‘edu,’ etc.” *Id.*

203. The DNS server can request this from AMDS 303 which is connected to data repository 304. “One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6).” *Id.*

204. Next, “[i]n step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6).” *Id.* This step handles a case of routing to a device that had already been assigned a public IP address for routing to the device’s private IP address.

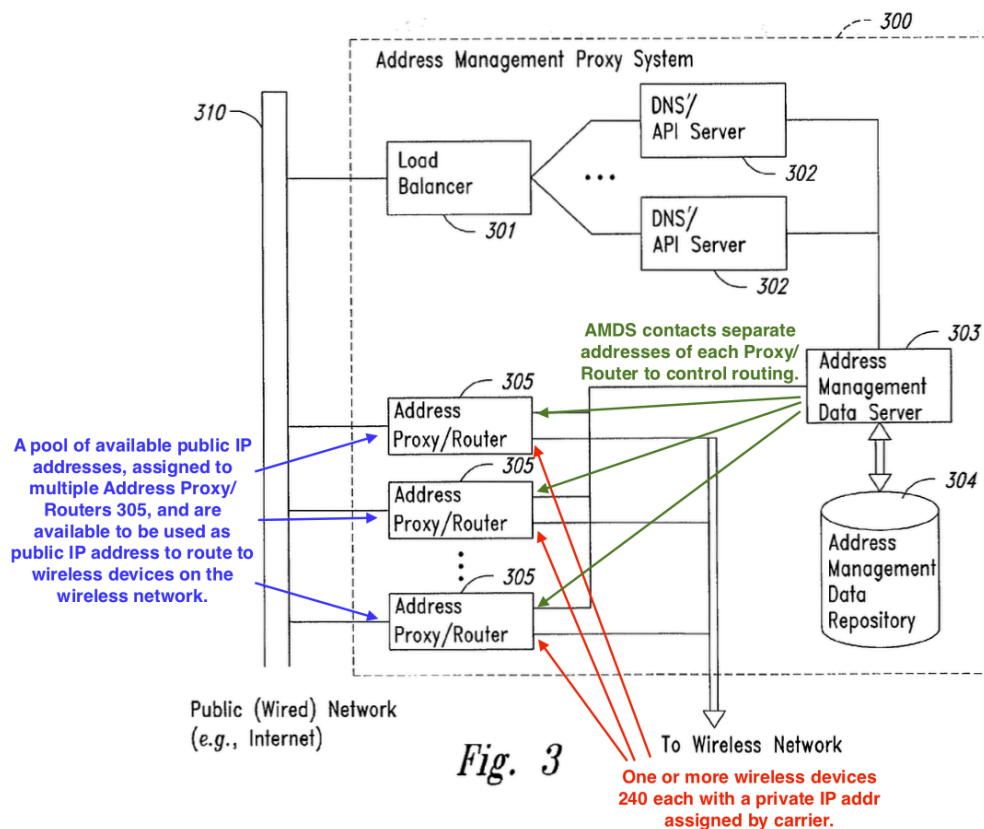
205. However, “[i]f a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address.” *Id.* That is, the AMPS will find an unused entry in table 630 that is marked (F)ree in flag 632.

206. “Appropriate tables in the data repository are then updated.” *Id.* Now that a free entry was selected, it must be marked as (U)sed in table 630 flag 632; and the private network address of the device from table 610 is copied in field 602 of table 630.

207. Now that the tables are updated, it is time to setup the actual device’s routing between its private and public addresses.

208. “In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6).” *Id.*

209. This step is to identify *how* to reach the Address Proxy/Router, because it has multiple IP addresses on both public and private networks. Fig. 3 shows that the Address Management Data Server 303 has a *direct* connection to each of the Address Proxy/Routers 305. I annotate this direct connect in Fig. 3 in green, reproduced further below.



210. A POSA would understand from this figure and Mehta’s disclosures that the AMDS 303 must deterministically contact each Address Proxy/Router 305 to control its routing: that cannot happen on the wireless private network (shown in red in my annotated Fig. 3 above) nor can it happen on the public side of Address Proxy/Routers 305 (shown in blue in annotated Fig. 3 above). The reason that the AMDS cannot use those private/public addresses of the Address Proxy/Router 305 is that those addresses and network interfaces are reserved for routing to/from wireless devices, rather than contacting the Address Proxy/Router 305 *directly* in order to control its state.

211. That is why, “[i]n step 704, the routine sends a request to the [“green” address of the] determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address.” *Id.*

212. Then, “[i]n step 705, the routine updates information in the data repository to include any other connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6).” *Id.*

213. Finally, “[o]nce all of the tables have been updated in both the proxy/router and in the data repository, the DNS’/API server returns the determined public network address of the associated Address Proxy/Router machine.” *Id.*

214. With this description of Mehta, I now address this DNS limitation (in reverse claimed order) next.

a. DNS address #3: a virtual network address associated with the second device.

215. This is one of the Address Proxy/Router 621 addresses, out of a pool of many “standby” ones, and is returned in the last step of Fig 7 above. This is also shown as the blue text and arrows in my annotated Fig. 3 above.

216. I note that this limitation only requires this address to be “associated with” the second device, but in this case, the address returned is actually the one given publicly and uniquely (albeit temporarily) to a wireless device on the wireless network.

b. DNS address #2: a private network address associated with a second device in the virtual network, and

217. This is the Private Network Address 602 in Tables 610 and 630 of Fig. 6 (shown above). As part of the steps of Fig. 7, the DNS’/API server has to query the Address Management Data Server 303 and it returns this private address from the Data Repository 304. This address is then further returned by AMDS 303 to DNS’/API Server 302. This is also shown as the red text and arrows in my annotated Fig. 3 above.

218. I note that this limitation only requires this address to be “associated with” the second device, but in this case, the address returned is actually the one assigned privately and uniquely to a wireless device on the wireless network (*e.g.*, by the carrier).

219. Mehta discloses returning the virtual network address (DNS address #3) and the network route director’s address (DNS address #1) in response to DNS queries. As these are public address, a POSA would have recognized them to be returned via DNS queries. *See also* §VII.B.3 above.

220. Mehta discloses that this private network address (DNS address #2) is returned internally from AMDR 304 to AMDS 303, and to DNS’API Server(s) 302. Mehta, however, does not describe specifically that this private network address is also returned as part of a DNS query.

221. However, a POSA would have been motivated to modify Mehta with RFC-1383 to add this step of returning the private IP address. As seen in RFC-1383, it already returns an “RX” record with private IP address “10.0.0.7.” EX1007 (RFC-1383), p. 11. *See also* §X.B above. In addition, a POSA would have been motivated to combine the systems and methods in Mehta with the DNS disclosure in RFC-1383 for the reasons discussed in §XII.B below.

222. In the alternative, Mehta also discloses a unique identifier that is a unique string name of the device which maps to the private network address assigned to the device. EX1006 (Mehta), ¶[0053]. Mehta also discloses that the unique identifier is used to “[determine] the private [] address of the wireless device designated by a string parameter passed as input to the [DNS query],” (EX1006 (Mehta), ¶[0057]) and any POSA would recognize that means the unique identifier (DNS address #2) is present in the DNS query and response.

223. In another alternative, Mehta also discloses that the string parameter “may use fields such as ‘uniqueID.hostname.domain.tld,’ which specifies a typical hierarchy of

person/service on a machine named ‘hostname’ on a domain such as a company’s network on a top level domain such as ‘org,’ ‘com,’ ‘edu,’ etc” and that “[o]ne skilled in the art will recognize that many other string parameter designations could be used.” EX1006 (Mehta), ¶[0057]. While Mehta, does not describe that the string parameter includes the private network address, a POSA would have been motivated to modify Mehta with RFC-1383 to add this step of including the private network address (DNS address #2) in the string parameter (“domain” field of RFC-1383), as the private network address maps to the unique identifier for the device, and is itself unique. EX1006 (Mehta), ¶[0052].

224. Mehta does not expressly disclose encapsulating IP addresses (*e.g.*, VPN, IPsec). Mehta, rather, discloses using Network Address Translation (NAT). EX1006 (Mehta), ¶[0026]. A POSA would have been motivated to improve on Mehta’s NAT techniques by using more modern IP encapsulation and tunneling techniques, such as those disclosed by RFC-1383. IP encapsulation, especially IPsec, would have also offered transparent packet encryption, an added security bonus. Moreover, by using RFC-1383, the Mehta invention would be improved by using IETF-standardized protocols, improving their interoperability world-wide.

c. DNS address #1: a network address associated with a network route director,

225. I note that this limitation again only requires the network address to be “associated” with the network route director.

226. The network route director is Mehta’s Address Proxy/Router 305 (any one of them). Mehta meets this limitation in two ways.

227. First, Mehta’s ADMS 303 necessarily contacts a unique address of each of the Address Proxy/Routers 305 as discussed above. This is shown in green text and arrows in my annotated Fig. 3 above. A POSA would have understood Fig. 3 and Mehta’s disclosures to clearly

mean that each Address Proxy/Router 305 must have a unique IP address for the AMDS to contact for route-control commands. As such, this address is “associated” with the network route director.

228. Second, in Mehta, each Address Proxy/Router 305 contains several standby public addresses as discussed above. Each of those addresses are clearly associated with the network route director (Mehta’s Address Proxy/Routers 305).

229. And, once one of those addresses are temporarily selected as the temporary routable public address of a wireless device, a POSA would have understood that this address *also* becomes the claimed “virtual network address associated with the second device” as discussed above.

6. A memory and a processor generally

230. Some of the independent claims in the ’785 patent recite generically a memory and processor as a component for the computing device which implements the virtual network manager, route director, or the register module of the virtual network manager, as shown in the table below.

Claim	Limitations
[1.1]	a virtual network manager implemented with a first device memory and a first device processor of a first computing device
[1.4]	a route director implemented with a second device memory and a second device processor of a second computing device,
[30.2]	a memory and a processor to implement a register module configured to register devices in a virtual network
[38.1]	a computing device that includes at least a memory and a processor configured to implement a network manager of a virtual network
[48]	<i>N/A</i>
[62.pre, 62.1]	One or more processor readable storage media devices comprising processor readable code that, if executed by a computer device, implements a virtual network manager

Claim	Limitations
[75.1]	a computing device that includes at least a memory and a processor configured to implement a virtual network manager having a network interface coupled to a virtual network

231. To the extent limiting, Mehta discloses various computing devices with memories and processors, including Computer System 400, Memory 401, and CPU 403, as shown in Figure 4 below. EX1006 (Mehta), ¶[0036]. To the extent that Mehta is found not to have disclosed a particular memory or processor (which are basic computing devices), it would have been obvious to a POSA to include such memory and processor to implement a routing director, register or join modules, or the virtual network system of the '785 patent. See §VII.A above.

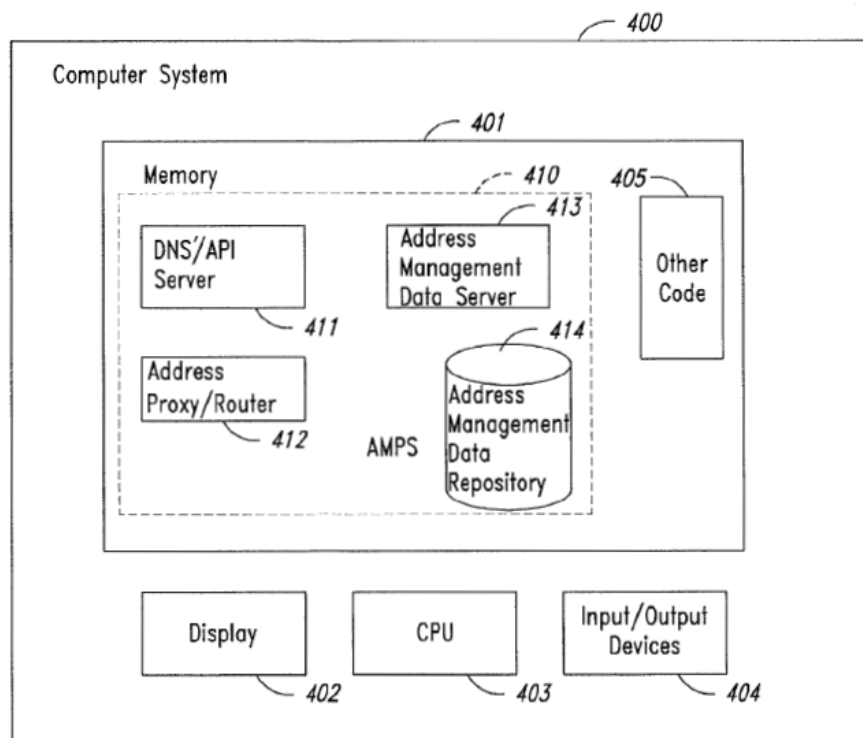


Fig. 4

B. Motivation To Combine Mehta and RFC-1383

232. A POSA would have been motivated to combine Mehta and RFC-1383 for at least the following reasons.

233. As an initial matter, the Domain Name System (DNS) is a foundational Internet technology, so much so that it was one of the very first IETF standards. Work on standardizing domain names began in 1981, with the publication of RFC-799, titled “Internet Name Domains”. *See* <https://datatracker.ietf.org/doc/html/rfc799>.

234. As I detail in §VII.B.3 above, several more RFCs ensued and provide extensive detail on DNS. Even the ’785 patent recognized the importance and ubiquity of DNS that it refers the reader for “[m]ore information about Dynamic DNS can be found in RFC 2136 incorporated herein by reference.” EX1001, 5:12-14. Indeed, without being able to name computer hosts and address them uniquely, the Internet would not even exist.

235. Similarly, Mehta briefly describes DNS. “In a typical TCP/IP (inter)network, machines are organized according to a naming hierarchy. One such hierarchy is the Domain Name System (‘DNS’), which specifies how a particular machine is connected to others. (The term DNS is sometimes used also to identify a server or service that implements the DNS protocol.)” EX1006 (Mehta), ¶[0009]. A POSA would recognize that DNS has been in use and well known for decades prior to 2003, and would have easily sought out additional DNS details in the various RFCs and textbooks written on the subject. *See* §VII.B.3 above.

236. Mehta discloses a modified DNS server that serves as a proxy/router for devices on a private network by maintaining a pool of addresses that are dynamically distributed among active devices. “The Address Management Proxy System achieves two way initiated bi-directional communication by implementing a modified DNS server and serving as a proxy/ router for devices

on the private wireless network as they interface to the public wired internetworking world. In summary, a pool of public addresses is maintained and dynamically distributed among active wireless devices as needed by the AMPS.” EX1006 (Mehta), ¶[0030].

237. Mehta also discloses a DNS query that returns (a) the device’s unique identifier as the domain part of the DNS query response and (b) a public address associated with the active device. “The DNS’/API servers 302 use the Address Management Data Server 303 to assist in mapping a unique identifier (e.g., string name) for a wireless device to a public address on public network 310.” EX1006 (Mehta), ¶[0032]. “FIG. 7 is an example flow diagram of an example routine provided by a DNS’/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier.” “Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as ‘uniqueID.hostname.domain.tld,’ ... One skilled in the art will recognize that many other string parameter designations could be used.” EX1006 (Mehta), ¶[0057].

238. Mehta also describes details of some of the well-known APIs users use to programmatically access DNS systems. For example, “TCP/IP and UDP/IP define tools/libraries for client systems to use to determine an IP address (a logical address on an internet) given a string name of a particular router/machine. One such tool is referred to as a DNS query and includes, for example, an API called ‘GetHostByName.’ GetHostByName returns an IP address given a string.” EX1006 (Mehta), ¶[0009]. (I personally used these programmable APIs and tools many times, *see* above)

239. A POSA would have also known that DNS queries can return multiple types of resource records and multiple entries per query. One example is returning multiple IP addresses for a single name. *See* §VII.B.3 above.

240. Mehta discloses a modified DNS server that can change what and how much is returned as a result of a DNS query. “According to one approach, the AMPS provides a modified DNS server that changes what is returned as a result of a call to a standard DNS query function, such as `GetHostByName`.” EX1006 (Mehta), ¶[0013].

241. Mehta then discloses being able to query information using alternative APIs, not necessarily the standard DNS ones. “In another approach, the AMPS implements a specialized API for a device on a public network to use to query for a public address of a designated wireless device. Using a specialized API implementation in conjunction with Internet Protocol, the AMPS can map private address to addresses that comprise only IP addresses, or can map to an (IP address, port) pair. This latter implementation can extend the usefulness of a particular public address space.” EX1006 (Mehta), ¶[0013].

242. A POSA would have appreciated the power that comes with modifying or extending APIs: new, more versatile functionality can be enabled easily.

243. Mehta further discloses returning the information using eXtended Markup Language (XML). A POSA would have known that data formatted as XML can be easily processed and parsed, and can be used to return an arbitrary number of items of any type.

244. “In one embodiment, a scripting language interface, such as XML, can be used instead of a specialized API (code interface) to interface to a specialized address mapping function. When using XML, the DNS'/API server accepts API calls as XML post events and returns XML formatted responses. Similar support for invoking this mapping function using other language

models and/or other programming languages is also contemplated and will operate with techniques of the present invention. An XML type interface minimizes the cost to a requesting device of integrating an API into its software.” EX1006 (Mehta), ¶¶[0045].

245. Mehta organizes all the information it needs into three tables, seen in Figure 6 (reproduced below). These tables list at least four pieces of information that can be associated with each host: a Unique ID 601, Private Network Address 602, Public Network Address 631, and Proxy/Router machine 621. Mehta also discloses auxiliary information such as F/U (flag) 632, TTL 633, and “Other ‘connection’ data” 634. *See generally* EX1006 (Mehta), ¶¶[0052]-[0056].

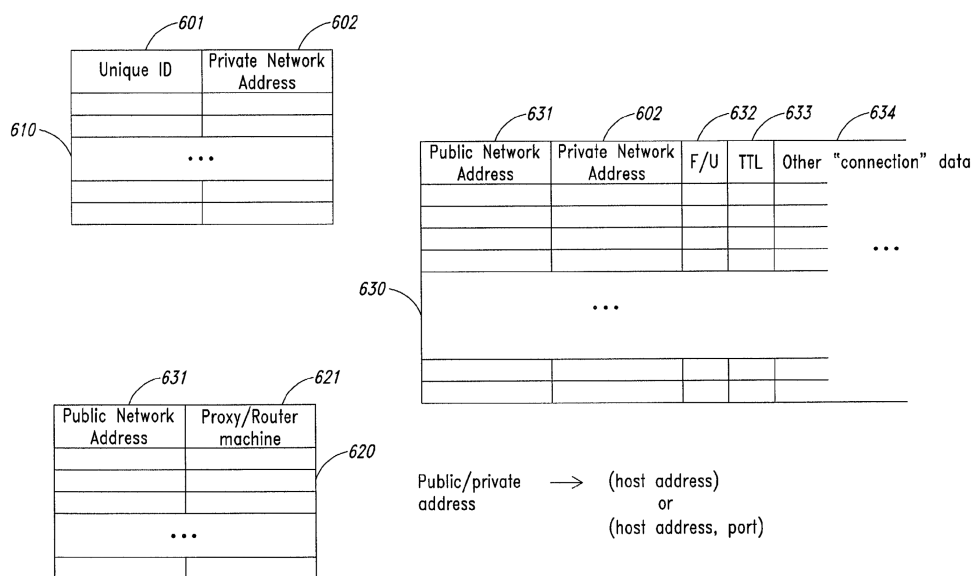


Fig. 6

246. Taking all these into account, a POSA would recognize that Mehta (a) discloses a modified DNS server that serves as a proxy/router by maintaining a pool of addresses that are assigned to active devices, (b) discloses a DNS query response that contains a device’s unique identifier and a public address associated with the device, and (c) discloses returning multiple pieces of data and (b) offers several methods for returning said data. In other words, Mehta describes a system that routes IP packets based on information stored in DNS.

247. A POSA implementing Mehta's system would have been motivated to seek out additional forms of returning Mehta's information. Mehta discloses changing the DNS server implementation to enable new functionality. Changing DNS code requires access to the DNS source code (which is generally opens source) as well as the ability to modify the code, test it, and deploy it. A POSA would have looked for ways to implement Mehta's solution with less effort, possibly utilizing DNS's existing mechanisms. Using an unmodified DNS server can allow one to experiment more quickly with a new technique; if those changes work well, one can then consider modifying the DNS server and/or exporting new APIs that can be more efficient.

248. Therefore, a POSA would have found RFC-1383, titled "An Experiment in DNS Based IP Routing." Just from this RFC's title alone, a POSA would have recognized its similarity and applicability to Mehta, because Mehta effectively routes IP packets based on information stored in DNS.

249. POSA would have recognized that RFC-1383 goes into more detail about the modified DNS server by disclosing a way to identify an address of the destination in the domain part of the DNS request and response, and to store and return multiple IP addresses in response to a single query. RFC-1383 proposes using "TXT" DNS records to store that information. A POSA would have understood that TXT records are a general purpose one that can store any arbitrary information. *See* §VII.B.3 above.

250. "This [TXT] record is designed for easy general purpose extensions in the DNS, and its content is a text string. The RX record will contain three fields: - A record identifier composed of the two characters 'RX'. This is used to disambiguate from other experimental uses of the 'TXT' record. - A cost indicator, encoded on up to 3 numerical digits. The corresponding

positive integer value should be less than 256, in order to preserve future evolutions. - An IP address, encoded as a text string following the ‘dot’ notation.” EX1007 (RFC-1383), p. 11.

251. RFC-1383 shows an example record:

domain	type	record	value
*.27.32.192.in-addr.arpa	IP	TXT	RX, 10, 10.0.0.7

252. RFC-1383 explains the table: “that means for all hosts whose IP address starts by the three octets ‘192.32.27’ the IP host ‘10.0.0.7’ can be used as a gateway, and that the preference value is 10.” EX1007 (RFC-1383), p. 12.

253. POSA would have appreciated that TXT records are just a domain field containing a destination address, and a string value comprising comma-delimited fields, including an IP address. A POSA would have found it trivial to use such TXT records to return any number of comma-delimited IP addresses, values, and any information disclosed in Mehta.

254. I note that even the ’785 patent contemplates using TXT records in very similar ways as disclosed by RFC-1383: a resource record with auxiliary information stored in TXT records. “The target member’s virtual IP address will be presented using an A-type resource record in the answer section of the normal DNS response packet. All the remaining information will be transferred using a TXT-type resource record in an additional section of the DNS Response.” EX1001, 22:53-57.

255. Parsing data strings using delimiters such as commas is a well-known technique often taught to first year computer science students. This would be well known to any user who used a spreadsheet file formatted as a comma-separated vector (CSV file). Indeed a “comma-delimited file” is “[a] data file consisting of fields and records, stored as text, in which the fields are separated from each other by commas. Use of comma-delimited files allows communication

between database systems that use different formats. If the data in a field contains a comma, the field is further surrounded with quotation marks.” EX1013 (Microsoft Computer Dictionary), pp. 100-101.

256. In sum, given how well-known DNS is, a POSA would have found it easy and desirable to modify Mehta with RFC-1383’s TXT record disclosures for routing IP packets using DNS because (a) RFC-1383 provides additional detail about modifications to the DNS server of Mehta that enable multiple addresses to be returned as part of the DNS request and response utilizing standard “TXT” DNS records, and (b) RFC-1383 enhances the unique identifier of Mehta with destination address information by including a destination address in the domain field of the DNS query response.

257. This change would have been simple and would not have required extensive experimentation or impose undue burden, because DNS is well known, standardized, and widely used. For the same reasons, the combination would have yielded predictable results. Lastly, a POSA would have had a high expectation of success because RFC-1383 merely provides details of one form of returning IP addresses out of several that Mehta describes.

C. The Dependent Claims

1. Claims 2, 6-12, 22, 25-26, and 28-29 are also obvious over the combination of Mehta and RFC-1383

258. Claims 2, 6-12, 22, 25-26, and 28-29 are dependents of independent Claim 1 and are obvious in light Mehta and RFC-1383 as shown below.

a. Claim 2: The system of claim 1 wherein each of the devices communicate with the virtual network manager and the other devices in the virtual network via an associated agent

259. Mehta discloses the system of claim 1 wherein each device communicates with the virtual manager (AMPS) and each other in the virtual network via an associated agent (client program).

260. Mehta discloses “client programs” capable of receiving and handling ... incoming communication packets,” including registration requests. *See* EX1006 (Mehta) ¶[0025]. It further explains that the devices or blocks within the AMPS “use standard interprocess communication mechanisms to communicate with each other,” which a POSA would understand to include agents, processes, or programs installed on the device for data communication. *Id.* at ¶[0036]; *see also id.* at ¶[0026], ¶[0053].

b. VII.B above Claim 6: The system of claim 1 wherein each of the devices include an agent configured to communicate with the virtual network manager and agents of the other devices in the virtual network

261. *See* Claim 2 §XII.C.1.a above.

c. Claim 7: The system of claim 1 wherein the virtual network manager is coupled to the public network and has an associated public network address

262. Mehta discloses the system of claim 1 wherein the virtual network manager is coupled to the public network and has a public network address.

263. Mehta discloses the AMPS (virtual network manager) enabling “devices and systems connected to a public internet.” EX1006 (Mehta) ¶[0024]. A POSA would have understood that for any system that purports to connect to a computer network, it must include devices (*e.g.*, Ethernet or Wi-Fi Network Interface Cards or NICs, dialup modems, and the like). *See* §**Error! Reference source not found.** *See also id.* at ¶[0033], ¶[0055]. The AMPS has a

“public network address” [*see* cl. 5] and includes a DNS or domain name server with a domain name for the network and associated public IP address. *Id.* at ¶[0012] (“the AMPS comprises one or more modified DNS’/API servers.”).

- d. Claim 8: The system of claim 1 wherein the route director is further configured to receive the encapsulated packets when routed to a public network address corresponding to the route director**

264. *See* discussion of “encapsulated packets” limitation in XII.A.4.a above. A POSA would have understood that encapsulating packets (e.g., using VPN or IPSec) was well known in the art. *See* §VII.B above.

- e. Claim 9: The system of claim 1 wherein each of the devices in the virtual network are further identified by at least one physical network address**

265. A POSA would have understood that every device on a network, whether physical or virtual, has a network address, else it cannot communicate. *See* §VII.B above

- f. Claim 10: The system of claim 9 wherein the physical network addresses that are associated with the devices in the virtual network are dynamic**

266. A POSA would have also understood that IP addresses could be assigned dynamically (e.g., via DHCP) or statically assigned. *See* §VII.B above.

- g. Claim 11: The system of claim 9 wherein the physical network addresses that are associated with the devices in the virtual network are static**

267. A POSA would have also understood that IP addresses could be assigned dynamically (e.g., via DHCP) or statically assigned. *See* §VII.B above.

- h. Claim 12: The system of claim 9 wherein the physical network addresses that are associated with the devices in the virtual network are private network addresses, and wherein the virtual network addresses are unique in the virtual network so as not to conflict with the private network addresses**

268. Mehta discloses the system of claim 9 wherein the physical network addresses of the devices are private network addresses and wherein the virtual network addresses are unique in the virtual network so as not to conflict with the private network addresses.

269. A POSA would have also understood that in any network domain where computers can communicate directly, IP addresses must be unique, else computers would not know where to send or receive packets uniquely. For example, if two computers claim to have the same IP address, a POSA would have known that their packet transmissions could overlap and corrupt each other (*e.g.*, no differently than if two people claimed to have the same social-security number). Similarly, a packet sent to such a computer would be inadvertently received by two computers and processed there, resulting in duplicate actions (*e.g.*, a shopping cart purchased twice). *See* §VII.B above.

270. Mehta discloses that the virtual network manager [AMPS] “dynamically allocat[es], using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on” that uniquely identifies the device. *Id.* at ¶[0053]. “Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” *Id.* *See also* Fig. 3 (showing the Address Management Data Server 303 and Address Management Data Repository 304 where registration information for each device is stored). “The DNS’/API servers 302 use the Address Management

Data Server 303 to assist in mapping a unique identifier [virtual network address] (e.g., string name)... for a wireless device to a public address on public network 310.” *Id.* at ¶[0032].

- i. Claim 22: The system of claim 1 wherein the virtual network manager is coupled to the public network and includes a public network address**

271. *See* Claim 7, §XII.C.1.c above.

- j. VII.B above VII.B above Claim 25: The system of claim 1 wherein the encapsulated packets further include virtual IP packets configured for communication of the data, the virtual IP packets being encrypted prior to being encapsulated**

272. A POSA would have known that IP Security (IPSec) protocols existed to allow one to transmit secure, encrypted packets, embedded (i.e., encapsulated) inside plain IP packets. *See* §VII.B above.

273. As an initial matter, a POSA would have known that any unencrypted networking connection can be encrypted using, for example, long deployed SSL and/or IPSec protocols. *See* §VII.B above.

274. Mehta discloses several cellular/wireless network protocols such as GPRS, GSM, and CMDA. EX1006 (Mehta), ¶[0027]. A POSA would have known that GSM uses network encryption. (See <https://en.wikipedia.org/wiki/GSM>, last viewed Nov. 16, 2024.)

275. *See* discussion of “encapsulated packets” limitation in §XII.A.4.a above.

- k. Claim 26: The system of claim 1 wherein the encapsulated packets further include virtual IP packets configured for communication of the data, the virtual IP packets being encrypted prior to being encapsulated**

276. A POSA would have known that IP Security (IPSec) protocols existed to allow one to transmit secure, encrypted packets, embedded (i.e., encapsulated) inside plain IP packets. *See* §VII.B above.

277. See discussion of “encapsulated packets” limitation in §XII.A.4.a above.

- l. VII.B above Claim 28: The system of claim 1 wherein the virtual network manager includes a virtual community definition that is defined by the domain name, and includes one or more of the devices that are registered in the virtual network**

278. See discussion related to Claim 1, §XII.A above, and Claim 7, §XII.C.1.c above.

- m. Claim 29: The system of claim 1 wherein the route director is a public route director in the public network, the system further comprising a private route director in a private network configured to enable access to devices of the private network that are also in the virtual network from devices of the public network that are also in the virtual network**

279. A POSA would have also understood that IP addresses could be assigned dynamically (*e.g.*, via DHCP) or statically assigned, and that DHCP servers can assign private and/or public addresses. See §VII.B above.

280. Moreover, a POSA “will recognize that various other approaches and combinations are possible.” *Id.* at ¶[0039].

2. Claims that Depend on Independent Claim 30

281. Claims 31-37 depend on Independent Claim 30 and generally recite additional limitations regarding network interface and/or the registration of member devices in the virtual network.

- a. Claim 31: The virtual network manager of claim 30 further comprising an additional network interface configured for data communication via a public network**

282. Mehta discloses the virtual network manager (AMPS) with an additional network interface (proxies/routers) for data communication via a public network (the internet). A POSA would recognize that proxy/router devices often contain two or more interfaces, as they process incoming packets on one end, and transmit them on another end. See §VII.B above.

283. “One skilled in the art [POSA] will recognize that these devices [in the AMPS] could be connected to another private or public network.” Mehta [[0030]. *See also id.* at Fig. 2, disclosing multiple proxies/routers (*e.g.*, network interfaces) for data communications.

b. Claim 32: The virtual network manager of claim 31 wherein the network interface is a UDP port configured for data communication via the virtual network, and wherein the additional network interface is a TCP port configured for registration communications via the public network

284. Mehta discloses a virtual network wherein the network interface can be a UDP port for data communication and wherein the additional network interface can be a TCP port configured for registration communications via the public network. A POSA would recognize that technologies that TCP can easily utilize UDP and vice versa. *See* §VII.B above.

285. “Ports are typically used to open virtual channels on a single physical address and potentially extend a single physical address to another approximately 65,000 connections.” *Id.* at ¶[0043], ¶[0042] (“Ports and their implementation are well known in the art.”) The AMPS “can be connected to another TCP/IP based or UDP/IP-based internet.” *Id.* at ¶[0008]. And “ports are specified preferably by the Address Proxy/Router [network interface]”. *Id.* at ¶[0043]. *See also id.* at ¶[006], ¶[009], 0010]–[0012], ¶[0014], cls. 107, 108.

286. Moreover, a POSA “will recognize that various other approaches and combinations are possible.” *Id.* at ¶[0039].

c. Claim 33: The virtual network manager of claim 30 wherein the network interface is a UDP port configured for data communication via the virtual network

287. *See* Claim 32, §XII.C.2.b above.

d. Claim 34: The virtual network manager of claim 30 wherein the register module is further configured to receive the registration request from the agent that is installed on the device for data communication via the virtual network

288. Mehta discloses a virtual network manager (AMPS) wherein the register module (blocks/components within the AMPS) is configured to receive the registration request (incoming communication packet) from the agent installed on the device (client program) for data communication.

289. Mehta discloses “client programs” capable of receiving and handling ... incoming communication packets,” including registration requests. *See* EX1006 (Mehta) ¶[0025]. It further explains that the devices or blocks/components within the AMPS (*e.g.*, the Address Management Data Repository [“AMDR”]) “use standard interprocess communication mechanisms to communicate with each other,” which a POSA would understand to include agents, processes, or programs installed on the device for data communication. *Id.* at ¶[0036]; *see also id.* at ¶[0026], ¶[0053].

e. Claim 35: The virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network

290. Mehta discloses virtual network manager (AMPS) with a join module configured to receive join and leave requests from the agent associated with the device (client program). A POSA would have known that protocols like DHCP support commands/messages to join and leave a network. *See* §VII.B above.

291. Mehta discloses devices “register[ing] and unregister[ing]” themselves with the AMPS (or the carrier infrastructure subsuming the AMPS), and storing device information in the AMDR, which can be “deleted dynamically as devices register and unregister. *Id.* at ¶[0053]. *See also id.* at ¶[0026], ¶[0036].

f. Claim 36: The virtual network manager of claim 35 wherein the join module is further configured to provide virtual network addresses to the devices that are registered in the virtual network

292. Mehta discloses a “Unique ID” and private wireless network address being assigned to devices when they register to the virtual network, and storing device information in the AMDR. *Id.* at ¶[0053]; *see also* ¶[0052].

293. Mehta discloses virtual network manager (AMPS) with a join module configured to receive join and leave requests from the agent associated with the device (client program). A POSA would have known that protocols like DHCP support commands/messages to join and leave a network. *See* §VII.B above.

294. Mehta discloses devices “register[ing] and unregister[ing]” themselves with the AMPS (or the carrier infrastructure subsuming the AMPS), and storing device information in the AMDR, which can be “deleted dynamically as devices register and unregister. *Id.* at ¶[0053]. *See also id.* at ¶[0026], ¶[0036].

g. Claim 37: The virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network

295. Mehta discloses the virtual network manager (AMPS) wherein the join module maintains data to associate a virtual network address (Unique ID) with a device in the virtual network.

296. Mehta's AMDR (join module and part of AMPS) stores and maintains device information, including addresses in mapping tables. *Id.* at ¶[0012] (The mapping of addresses associated with a device "is maintained and updated transparently by the AMPS using routing tables and other mapping data structures.") *See also* ¶[0032], ¶[0038], ¶[0057], ¶[0059], Fig. 6.

297. Mehta further maintains all sorts of data associated with a virtual network address, directly or indirectly. For example, Mehta's Table 6 shows several addresses as well as "F/U" (flags), "TTL", and "Other 'connection' data." *Id.*, Fig. 6, ¶[0052]-[0056]

3. Similar or Identical Claims that Depend on Independent Claims 48 and 62

298. Many of the claims that depend on independent claims 48 and 62 are identical or substantially similar. For ease of discussion, I discuss them together as follows, including where the specific limitations are disclosed in Mehta.

a. Claim 49: The method of claim 48 further comprising providing the devices in the virtual network with a communication agent

299. I note that the agent is described in the '785 patent as simply a "virtual network adapter interfacing with the device and applications on the devices to route traffic," *i.e.*, a virtual network card or interface. EX1001, 7:1-5.

b. Claim 63: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to provide the devices in the virtual network with a communication agent

300. Mehta discloses providing the devices in the virtual network with a communication agent (client program).

301. The "client programs" are for "receiving and handling ... incoming communication packets." *See* EX1006 (Mehta) ¶[0025].

302. Mehta further discloses that the devices or blocks within the AMPS “use standard interprocess communication mechanisms to communicate with each other,” which a POSA would understand to include agents or programs installed on the device for data communication. *Id.* at ¶[0036]; *see also id.* at ¶[0026], ¶[0053].

303. A POSA would have known that interprocess communication (IPC) was commonly used in operating systems by 2003. *See* §VII.A above.

- c. **Claim 52: The method of claim 48 further comprising defining a member set of the virtual network that includes one or more of the devices, and assigning the domain name that defines the virtual network**
- d. **Claim 65: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to define a member set of the virtual network that includes one or more of the devices, and the domain name that defines the virtual network**

304. Mehta discloses the virtual network that includes one or more devices, and the domain name that defines the virtual network. *See* §X.A above. *See also* discussion of “virtual network with a domain name” element in §XII.A.2 above.

- e. **Claim 54: The method of claim 48 further comprising assigning the virtual network address to the device as an IPV4 compliant address**
- f. **Claim 66: One or more processor readable storage media devices as recited in claim 62 wherein the virtual network address includes a non-routable IPV4 compliant address**

305. *See* EX1006 (Mehta) ¶[0007]. “IP addresses are often written in ‘dotted decimal’ notation, which specifies an address as ‘w.x.y.z’ (e.g., in a 32-bit address, each letter designates a byte of information). For example, an IP address such as ‘192.251.0.1’ may specify a host machine

at network ‘192,’ subnet ‘251,’ and domain ‘0.’” A POSA would have known that 32-bit IP addresses are IPv4 ones. *See* §VII.B above.

- g. Claim 57: The method of claim 48 further comprising routing network traffic from a first device in a private physical network having a private network address to a second device having a public network address, the network traffic being routed as data packets having a source address which is the virtual network address of the first device, a destination address which is the virtual network address of the second device, and a shim which includes the private network address, the data packets being encapsulated to include a second destination address which is the public network address of the second device**

306. *See* discussion of “encapsulated packets” limitation in §XII.A.4.a above.

- h. Claim 61: The method of claim 48 further comprising applying a group policy to the devices that are registered as the members of the virtual network**
- i. Claim 74: One or more processor readable storage media devices as recited in claim 62 further comprising processor readable code that, if executed, implements the virtual network manager to apply a group policy to the devices that are registered as the members of the virtual network**

307. Mehta discloses several parameters that can be used to enforce policies. “The public-to-private address table **630** comprises several fields/columns including a public network address **631**, a private (wireless) network address **602**, a flag **632** that specifies whether the public address stored in field **631** is free or is already used, a Time to Live (TTL) parameter **633**, and other connection data **634**.” EX1006 (Mehta) at ¶[0052]. For example, “once the **TTL** value indicates that the mapping has expired, the **AMPS** can destroy the mapping, and also any connection.” *Id.* at ¶[0015]. A POSA would have recognized at least the TTL field is controlling and enforcing a policy.

308. Moreover, a POSA “will recognize that various other approaches and combinations are possible.” *Id.* at ¶[0039].

4. Additional Claims that Depend on Independent Claims 48 and 62

309. Claims 51, 53, 55, 60, 72, and 73 are also obvious and unpatentable in light of Mehta and RFC-1383 as shown below.

- a. **Claim 51: The method of claim 48 further comprising defining at least two member sets of the virtual network that each include one or more of the devices, the at least two member sets having at least one different device**

310. See Claim 48, §XII.C.3 above.

311. Moreover, a POSA “will recognize that various other approaches and combinations are possible.” *Id.* at ¶[0039].

- b. **Claim 53: The method of claim 48 further comprising defining at least two member sets of the virtual network that each include one or more of the devices, the at least two member sets having at least one different device**

312. See Claim 48, §XII.C.3 above.

313. Moreover, a POSA “will recognize that various other approaches and combinations are possible.” *Id.* at ¶[0039].

- c. **Claim 55: The method of claim 54 wherein the IPV4 compliant address is non-routable**

314. See Claim 54, §XII.C.3.e above. Mehta discloses the method of claim 54 wherein the IPV4 compliant address is non-routable.

- d. **Claim 60: The method of claim 48 further comprising: receiving a join status request from a first device in the virtual network as a query to determine the status of a second device in the virtual network; and responding to the join status request to indicate whether or not the second device is joined to the virtual network for data communication**

315. See Claim 48, §XII.C.3 above; claim 35, §XII.C.2.e above.

- e. **Claim 72: One or more processor readable storage media devices as recited in claim 67 wherein the network traffic is routed as encapsulated data packets**

316. *See* Claim 48, §XII.C.3 above.

- f. **Claim 73: One or more processor readable storage media devices as recited in claim 67 wherein the network traffic is routed as encrypted data traffic**

317. A POSA would have known that IP Security (IPSec) protocols existed to allow one to transmit secure, encrypted packets, embedded (i.e., encapsulated) inside plain IP packets. *See* §VII.B above.

318. As an initial matter, a POSA would have known that any unencrypted networking connection can be encrypted using, for example, long deployed SSL and/or IPSec protocols. *See* §VII.B above.

319. Mehta discloses several cellular/wireless network protocols such as GPRS, GSM, and CMDA. EX1006 (Mehta), ¶[0027]. A POSA would have known that GSM uses network encryption. (See <https://en.wikipedia.org/wiki/GSM>, last viewed Nov. 16, 2024.)

5. Claims that Depend on Independent Claims 75

320. Claims 76-80, and 82-90 are obvious and unpatentable over Mehta and RFC-1383.

- a. **Claim 76: The system of claim 75 wherein the virtual network manager includes a member register module**

321. *See* Claim 75, §XII.C.5 above; and Claim 34, §XII.C.2.d above.

- b. **Claim 77: The system of claim 75 wherein the virtual network manager includes a member join module**

322. *See* Claim 75, §XII.C.5 above; and Claim 35, §XII.C.2.e above.

- c. **Claim 78: The system of claim 77 wherein the member join module provides a virtual network address to a device that is registered as a member of the network**

323. See Claim 36, §XII.C.2.f above.

- d. **Claim 79: The system of claim 75 wherein the virtual network manager is further configured to maintain data on an association between at least one virtual network address with at least one device that is registered as a member of the network**

324. See Claim 37, §XII.C.2.f above.

- e. **Claim 80: The system of claim 75 wherein the virtual network manager includes a DNS server for the virtual community network**

325. See Claim 75.4, §XII.A.4 aboveXII.A.5 above.

- f. **Claim 82: The system of claim 75 wherein the virtual network manager includes at least a second virtual community definition**

326. A POSA would have understood that computers and network systems/devices could easily define multiple subnets, multiple domains, and more. This is commonplace when configuring DNS and/or DHCP systems for example. See §VII.B above.

- g. **Claim 83: The system of claim 75 wherein the virtual network manager includes a member authenticator**

327. See Claim 35, §XII.C.2.e above.

- h. **Claim 84: The system of claim 75 wherein the virtual network manager includes a DNS server configured to provide authoritative responses for DNS queries from devices in the virtual community**

328. See Claim 75.4, §XII.A.4 aboveXII.A.5 above.

329. A POSA would have understood that DNS servers are designed to return authoritative responses, else the internet could not even be trusted. See §VII.B above.

- i. Claim 85: The system of claim 75 further comprising at least one route director configured to communicate with the one or more devices in the user set**

330. *See* Claim 75.4, §XII.A.4 aboveXII.A.5 above.

- j. Claim 86: The system of claim 75 wherein each device registered in the network is configured to communicate with the virtual network manager and other devices in the user set via at least one agent**

331. *See* Claim 75, §XII.C.5 above; and Claim 34, §XII.C.2.d above.

- k. Claim 87: The system of claim 75 wherein the user set includes at least a first device and a second device, and wherein at least one of said first device and said second device is coupled to a first private network and accesses a public network via a NAT device**

332. *See* Claim 75.

333. A POSA would have understood that a Network Address Translation (NAT) device enables devices on private networks to access public networks. A POSA would recognize that NAT has been in use and well known prior to 2003, and would have easily sought out additional NAT details in the various RFCs and textbooks written on the subject. *See* §VII.B above.

- l. Claim 88: The system of claim 87 wherein said first device is coupled to said first private network, and said second device is coupled to a second private network and accesses the public network via a second NAT device**

334. *See* Claim 87.XII.C.5 aboveXII.C.2.d above

- m. Claim 89: The system of claim 75 wherein communications between the one or more devices in the user set are encrypted**

335. *See* Claim 75, §XII.C.5 above.

336. A POSA would have known that IP Security (IPSec) protocols existed to allow one to transmit secure, encrypted packets, embedded (i.e., encapsulated) inside plain IP packets. *See* §VII.B above.

337. As an initial matter, a POSA would have known that any unencrypted networking connection can be encrypted using, for example, long deployed SSL and/or IPsec protocols. *See* §VII.B above.

338. Mehta discloses several cellular/wireless network protocols such as GPRS, GSM, and CMDA. EX1006 (Mehta), ¶[0027]. A POSA would have known that GSM uses network encryption. (See <https://en.wikipedia.org/wiki/GSM>, last viewed Nov. 16, 2024.)

n. Claim 90: The system of claim 89 wherein the virtual network manager is configured to provide a shared message to the one or more devices in the user set to establish encrypted communications

339. A POSA would have known that IP Security (IPsec) protocols existed to allow one to transmit secure, encrypted packets, embedded (i.e., encapsulated) inside plain IP packets. *See* §VII.B above.

340. As an initial matter, a POSA would have known that any unencrypted networking connection can be encrypted using, for example, long deployed SSL and/or IPsec protocols. *See* §VII.B above.

341. Mehta discloses several cellular/wireless network protocols such as GPRS, GSM, and CMDA. EX1006 (Mehta), ¶[0027]. A POSA would have known that GSM uses network encryption. (See <https://en.wikipedia.org/wiki/GSM>, last viewed Nov. 16, 2024.)

XIII. XII.C.2.d aboveError! Reference source not found.**Error! Reference source not found.**Error! Reference source not found.**Error! Reference source not found.****XII.A.4.a above**Error! Reference source not found.**Error! Reference source not found.****XII.A.4.a above**Error! Reference source not found.**VII.B above**Error! Reference source not found.**VII.B above**Error! Reference source not found.**XII.A.4.a above**Error! Reference source not found.**Error! Reference source not found.****XII.A.4.a above****VII.B above**Error! Reference source not found.**Error! Reference source not found.****Error! Reference source not found.****Error! Reference source not found.****Error! Reference source not found.****Error! Reference source not found.****XII.B above****VII.B above**Error! Reference source not found.**VII.B above****SECONDARY CONSIDERATIONS**

342. I have seen no evidence in either the '785 patent's prosecution history or elsewhere supporting any secondary considerations arguments, or evidence of nexus of such alleged evidence to the challenged claims. To the extent patent owner or its expert asserts the existence of any secondary considerations in its responses, I reserve the right to address any such evidence.

XIV. CONCLUSION

343. The findings and opinions set forth in this declaration are based on my work and examinations to date.

344. I may continue my examinations. I may also receive additional documentation and other factual evidence over the course of this IPR that will allow me to supplement and/or refine my opinions. I reserve the right to add to, alter, or delete my opinions and my declaration upon discovery of any additional information. I reserve the right to make such changes as may be deemed necessary.

345. In signing this declaration, I recognize that the declaration will be filed as evidence in an IPR before the PTAB. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

346. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code.

Date: November 20, 2024

A handwritten signature in black ink, appearing to read "Erez Zadok". The signature is written in a cursive, flowing style with some capitalization.

Erez Zadok, Ph.D.