

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

AMERICAN AIRLINES, INC.

AND

SOUTHWEST AIRLINES CO.

Petitioners

v.

INTELLECTUAL VENTURES I, LLC,

Patent Owner

---

IPR2025-00786

U.S. Patent No. 7,949,785 B2

**DECLARATION OF DR. GUEVARA NOUBIR  
IN SUPPORT OF PATENT OWNER'S PRELIMINARY RESPONSE**

## TABLE OF CONTENTS

I. INTRODUCTION AND SCOPE OF WORK .....	1
II. QUALIFICATIONS .....	2
III. RELEVANT LEGAL STANDARDS .....	5
IV. SPECIFICATION OF THE ‘785 PATENT .....	8
A. Overview.....	8
B. Terminology.....	11
C. Considered Claims of the ‘785 Patent.....	20
V. LEVEL OF A PERSON HAVING ORDINARY SKILL IN THE ART .....	26
VI. PRIOR ART RELIED ON IN THE PETITION .....	27
A. Caronni-I (‘941).....	27
B. Caronni-II.....	30
C. Hipp .....	33
D. RFC-1383 .....	36
VII. CARONNI-I IN COMBINATION WITH CARONNI-II AND HIPPI DOES NOT TEACH ‘785 CLAIMS 1, 30, 38, 48, 62, 75 .....	39
VIII. CARONNI-I IN COMBINATION WITH CARONNI-II AND RFC-1383 DOES NOT TEACH ‘785 CLAIMS 1, 30, 38, 48, 62, 75.....	52
IX. CONCLUSION .....	56
X. DECLARATION.....	57

I, Dr. Guevara Noubir, hereby declare:

## **I. INTRODUCTION AND SCOPE OF WORK**

1. I am making this declaration at the request of Intellectual Ventures I, LLC (“Intellectual Ventures”) in the matter of the *Inter Partes* Review (“IPR”) proceedings before the United States Patent and Trademark Office (“USPTO”) of U.S. Patent No. 7,949,785 (“the ‘785 Patent”) to provide expert technical opinions with respect to the ‘785 Patent.

2. I am being compensated for my work in this matter at my standard hourly consulting rate for such consulting services. My compensation is not dependent on, and in no way affects, the substance of my statements in this Declaration or the outcome of this case. I do not have any financial interest in the outcome of this proceeding.

3. In preparation of this Declaration, I considered the following materials, the subject matter of which are all within the scope of my education and professional experience:

- a. U.S. Patent No. 7,949,785 (Ex. 1001, “the ‘785 Patent”)
- b. The Petition for *Inter Partes* Review of U.S. Patent No. 7,949,785
- c. U.S. Patent No. 2005/6,970,941 B1 (Ex. 1003, “Caronni-I”)
- d. U.S. Patent No. 2010/US 7,814,228 B2 (Ex. 1004, “Caronni-II”)
- e. U.S. Patent No. 2004/US 6,766,371 B1 (Ex. 1005, “Hipp”)

- f. IETF RFC 1383, An Experiment in DNS Based IP Routing  
(December 1992) (Ex. 1006, “RFC-1383”)
- g. Expert Declaration of Dr. Erez Zadok (Ex. 1011)

4. I have been asked to review and provide my opinion as to whether Claims 1, 30, 38, 48, 62, and 75 of the ‘785 Patent are obvious over the combination of U.S. Patent No. 2005/6,970,941 B1 (hereinafter “Caronni-I”), with U.S. Patent No. 2010/US 7,814.228 B2 (hereinafter “Caronni-II”), and with U.S. Patent No. 2004/US 6,766,371 B1 (hereinafter “Hipp”), or over the combination of Caronni-I, Caronni-II, and RFC 1383, An Experiment in DNS Based IP Routing (December 1992) (hereinafter “RFC-1383”).

## II. QUALIFICATIONS

5. My name is Dr. Guevara Noubir. I am currently employed at Northeastern University as a Professor in the Khoury College of Computer Sciences.

6. Based on my qualifications, education, knowledge, expertise, experience, and work background, I believe I am qualified to offer opinions relating to the technology described in the ‘785 Patent. Exhibit 2009 is a copy of my current *curriculum vitae* (“CV”), detailing my education and experience. Additionally, the following overview of my background pertains to my qualifications for providing expert testimony in this matter.

7. I hold an M.S. in Computer Science (diplôme d'ingénieur) from the Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG) and Institut National Polytechnique de Grenoble (INPG), France, with a specialization in networks and real-time systems. I hold a Ph.D. in computer science from the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland. One of my primary fields of study in graduate school was algebraic methods for fault tolerance in the context of communications network protocols.

8. For three years, after my PhD, I worked as a senior research scientist in the Real-time Software and Networking Group at the Centre Suisse d'Electronique et de Microtechnique (CSEM) in Switzerland. My responsibilities included coordinating several European projects in wireless communications and network security.

9. I have been a Professor of computer science at Northeastern University since 2001, and a full Professor since 2011.

10. In my first full-time academic position (Assistant Professor) at Northeastern University, I taught in the areas of computer networks and wireless networks. I have taught courses on Internetworking and Wireless Networks since 2001. I introduced and began teaching courses in network security in 2003. In 2005, I began teaching a course in cryptography.

11. I have published and lectured extensively, primarily focusing in the areas of networks and wireless systems security, including new frontiers in networking, robustness and privacy in wireless and mobile systems, secure sharing of location information in wireless networks, preventing hacking of wireless networks, directing wireless security research, game theory in the wireless adversarial context, challenges to wireless security, heterogeneous wireless networks, wireless infrastructure, scalability and security of wireless networks, securing wireless sensor networks, and security challenges of multihop ad hoc networks.

12. I was named an IEEE Senior member in 2001 and have received numerous awards and honors for my research, including the US National Science Foundation CAREER Award in 2005, Google Faculty Research Award on Privacy in 2016, Northeastern University Excellence in Research and Creative Activity Award in 2018, best paper awards at ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec) in 2011 and 2018, and the IEEE Conference on Communications and Network Security best paper in 2016.

13. I led winning teams from Northeastern University in the DARPA Spectrum Collaboration Challenge (SC2) in 2017, 2018, and a finalist team in 2019.

14. I delivered keynote lectures at the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2019), and IEEE Conference on Communications and Network Security (CNS 2022).

15. I chaired the technical program committee of several academic conferences, including the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2015) and IEEE Conference on Communications and Network Security (CNS 2015). I served on the editorial boards of ACM Transactions on Privacy and Security, IEEE Transactions on Mobile Computing, Elsevier Journal on Computer Networks, and IEEE Transactions on Information Forensics and Security.

### **III. RELEVANT LEGAL STANDARDS**

16. The opinions I am expressing in this report involve the application of my knowledge and experience to evaluate certain references with respect to the '785 patent. My formal knowledge of patent law is no different than that of any lay person. Therefore, I have requested the attorneys from Volpe Koenig, who represent Intellectual Ventures, to provide me with guidance as to the applicable patent law in this matter. The paragraphs below express my understanding of how I must apply current principles related to patent validity to my analysis.

17. I understand that an IPR is a proceeding before the USPTO for evaluating the patentability of an issued patent's claims based on prior-art patents and printed publications.

18. I understand that the claims of a patent are reviewed from the point of view of a hypothetical person of ordinary skill in the art (“POSITA”) at the time of the filing of the patent. The “art” is the field of technology to which a patent relates.

19. I understand that in determining whether a patent claim is anticipated or obvious in view of the prior art, the Patent Office must apply the *Phillips* standard to construe the claims.

20. I understand that, in this proceeding, Petitioners have the burden of proving that the challenged claims of the ‘785 Patent are unpatentable by a preponderance of the evidence. I understand that “preponderance of the evidence” means that a fact or conclusion is more likely true than not true.

21. I understand that there are two ways in which prior art may render a patent claim unpatentable. First, the prior art can “anticipate” the claim. Second, the prior art can render the claim “obvious” to a POSITA. I understand that for an invention claimed in a patent to be patentable, it must not be anticipated and must not be obvious based on what was known before the invention was made. For purposes of this Declaration only, I have applied the definition of a POSITA proposed in the Petition.

22. I understand that a patent claim may be invalid as “anticipated” under 35 U.S.C. § 102 if each and every limitation of the claim as construed by the court is found, either expressly or inherently, in a single prior art reference, such as a

publication or patent, that predates the claimed subject matter. I also understand that a patent claim may be invalid as obvious under 35 U.S.C. § 103 if the differences between the subject matter claimed and the prior art are such that the claimed subject matter as a whole would have been obvious to a POSITA at the time the invention was made. I also understand that several factual inquiries underlie a determination of obviousness. These inquiries include (1) the scope and content of the prior art, (2) the level of ordinary skill in the field of the invention, (3) the differences between the claimed invention and the prior art, and (4) any objective evidence of non-obviousness (i.e., “secondary considerations”).

23. I also understand that, when a party alleges obviousness based on a combination of references, that party should identify a reason why a POSITA would have been motivated to combine the asserted references in the manner recited in the claims and to explain why a POSITA would have had a reasonable expectation of success in making such combinations.

24. I understand that an obviousness analysis permits the application of “common sense” in examining whether a claimed invention would have been obvious to a POSITA. For example, I understand that combining familiar elements according to known methods and in a predictable way may suggest obviousness when such a combination would yield nothing more than predictable results. I also understand a party asserting obviousness should still provide a specific motivation to combine the

references as recited in the claims and explain why one would have reasonably expected to succeed in doing so.

25. I understand that one indicator of non-obviousness is when a prior art reference leads in a different direction or discourages that combination, recommends steps that would not likely lead to the patent's result, or otherwise indicates that a seemingly inoperative device would be produced.

#### **IV. SPECIFICATION OF THE '785 PATENT**

##### **A. Overview**

26. The '785 Patent teaches how multiple devices can be operated and managed to form dynamic virtual communities and streamline communications even if some/all are not on the same physical network. It is noteworthy that a traditional VPN does not enable such communications as two devices might connect from arbitrary physical networks, changing their IP addresses; the IP addresses can change as they can change physical networks; such devices might pass through NAT boxes and, therefore have private IP addresses but are seen by other devices as using the global IP address of the NAT box. Ex. 1001, 4:58-5:10.

27. The '785 Patent introduces a comprehensive architecture, components, hardware and software to build a Virtual Community Network (VCN) with other components to support routing and learning Virtual IP addresses to reach other devices (using DNS names). It operates as an overlay but with various management

mechanisms and optimizations. It supports a range of connectivity/networking scenarios including NAT traversal, devices with public and private IP addresses, and Route Directors (PRD/NRDs). It integrates with the DNS protocols for the devices to determine how to reach each other. It supports secure tunnels; in particular encrypted ones (IPsec/IKE). It supports devices that are not aware of the VCN but are bridged through a Group Agent (GA). It supports defining virtual communities' membership including registration, and dynamic join/leave, authentication, and group policies. It supports multiple virtual communities.

28. Of particular importance to the '785 Patent is the domain name hierarchical structure. More specifically, the '785 patent describes a DNS-based hierarchical naming and addressing scheme for managing and routing communications within the VCN. Ex. 1001, FIG. 4, 9:18-35, 11:63-67. In this system, the VCN is defined by a domain name, and all participating elements -including nodes and route directors- are assigned fully qualified domain names (FQDNs) that reflect their position within the network's hierarchy. *Id.* at 9:18-35, 11:63-67, 5:66-6:7. Specifically, each node is identified by a DNS name, based on the specified X.VCN format. *Id.* at FIG. 4, 9:18-35. With such hierarchy, a POSITA would understand that VCN represents the top-level domain associated with the virtual community network, and X could identify a subdomain (e.g., group or route director), and member or endpoint identifiers could be added, such as member1.X.VCN. This approach

distinguishes the '785 patent from prior art that lacks such a domain-based hierarchy to delineate and manage relationships among distributed computing resources.

29. The '785 patent also outlines a structured process by which a source device within a VCN transmits a packet to a destination device, leveraging DNS-based addressing and route directors to manage path selection and endpoint resolution. The process begins when a source device determines that it needs to communicate with a destination node within the VCN. Rather than transmitting the packet directly via IP, the system first performs logical resolution based on hierarchical DNS naming conventions, enabling dynamic routing and policy enforcement. *Id.* at 13:66-14:46, FIG. 7. At step 654 of FIG. 7, an application on the source device that wants to send a message to the destination device sends a DNS request using the domain name for the destination device. *Id.* at 14:29-34, FIG. 4. The VCN manager then returns, in response to the DNS request, the three addresses that the source device might need for the entire transmission path between the source device and the destination, including the public address for the route director as well as the private/public and virtual IP addresses for the destination device. *Id.* at 14:34-40, FIG. 7.

30. This is not a conventional application of the Domain Name System (DNS) wherein a domain name is resolved to a single public IP address for basic connectivity. Rather, the '785 patent describes a novel and non-obvious use of DNS as a mechanism for distributing multiple pieces of addressing information required to

establish communication between devices operating within a shared virtual address realm, even when those devices reside in disparate physical address realms, such as behind different NAT devices. This approach leverages DNS not merely for endpoint resolution, but as a structured tool for managing multi-layered routing information essential for enabling inter-node communication in a dynamic environment.

## **B. Terminology**

31. **The Domain Name System (DNS)** The Domain Name System (DNS) is a distributed hierarchical system responsible for translating structured human-readable domain names into network addresses, primarily IP addresses. DNS allows users to conveniently access internet resources using **structured**, easy-to-remember names instead of numerical IP addresses. “Domains are administrative entities that provide decentralized management of host naming and addressing. The domain-naming system is distributed and hierarchical.” “Users also will appreciate shorter names. Most people agree that short names are easier to remember and type; most domain names registered so far are 12 characters or fewer.” Ex. 2012 Pages 1 and 3, Ex. 2010, Pages 2 and 6.

32. A POSITA would recognize that a domain name is a human-readable identifier that corresponds to a specific location on the Internet, such as a website or an online service. It acts as an alias for an IP address, the numerical identifier that computers use to locate and access network resources. Domain names simplify

navigation by enabling users to enter a recognizable name, such as example.com (or X.VCN in the '785 Patent), instead of a complex numerical IP address. They are organized hierarchically into components separated by dots, such as www.example.com, where “www” is a subdomain, “example” is a second-level domain, and “com” represents the top-level domain name. The DNS resolves these domain names into their corresponding IP addresses, facilitating communication between computers and the appropriate servers or devices. The same is true for the '785 Patent, where VCN is the top-level domain and “X” is the second-level domain. Devices in the VCN are also assigned domain names, which could be “member1.X.VCN,” which is a sub-domain of the VCN top-level domain. “A domain is identified by a domain name, and consists of that part of the domain name space that is at or below the domain name which specifies the domain. A domain is a subdomain of another domain if it is contained within that domain. This relationship can be tested by seeing if the subdomain’s name ends with the containing domain’s name. For example, A.B.C.D is a subdomain of B.C.D, C.D, D, and “.”” Ex. 1001, 1:45-50, 9:18-31; Ex. 2010, Page 8; Ex. 2011, Page 8.

33. RFC 1034 clarifies the specific DNS context, noting: “The terms ‘domain’ or ‘domain name’ are used in many contexts beyond the DNS described here. Very often, the term domain name is used to refer to a name with structure indicated by dots, but no relation to the DNS. This is particularly true in mail

addressing [Quarterman 86].” Ex. 2010, Page 2. Since all independent claims of ‘785 specifically require DNS usage, a POSITA would clearly understand that the discussion pertains explicitly to DNS.

34. According to RFC 1034 and RFC 1035, DNS has three major components:

**Domain Name Space and Resource Records:** This component defines a structured, hierarchical tree composed of nodes, each corresponding to resource records. As described in RFC 1034: “Each node has a label, which is zero to 63 octets in length. Brother nodes may not have the same label, although the same label can be used for nodes which are not brothers. One label is reserved, and that is the null (i.e., zero length) label used for the root.” A domain name consists of labels listed from left (most specific) to right (least specific, closer to the root). Ex. 2010, Page 7.

**Name Servers:** These are authoritative servers that store information about particular sections of the DNS tree. They respond to resolver queries by either providing the requested resource records or redirecting the resolver to another authoritative server closer to the domain queried. Ex. 2010, Page 6.

**Resolvers:** These client-side programs query name servers to obtain DNS information requested by client applications such as web browsers or email

clients. Resolvers convert domain names into IP addresses or retrieve other necessary resource records. Ex. 2010, Page 6.

35. Collectively, these components form a scalable and distributed system that efficiently translates and manages domain names across the Internet. RFC 1034 clearly states the hierarchical nature of domains: “A domain is identified by a domain name and consists of that part of the domain name space that is at or below the domain name which specifies the domain.” Ex. 2010, Page 8. For example, the domain example.com can have subdomains like example1.example.com.

36. Accordingly, a person of ordinary skill in the art would understand that, as used in the ‘785 patent, “DNS” refers to the standard Internet naming system that translates human-readable domain names (ASCII strings) into numerical IP addresses (Ex. 1001, 1:48–50). Within the disclosed architecture, DNS operates in two complementary capacities. First, public DNS associates the virtual community’s public domain name with the system’s publicly reachable endpoint (e.g., the virtual network manager or route director), enabling initial access from the Internet. Second, the virtual network manager implements an authoritative DNS service for the virtual network itself. In response to a member’s DNS query, that internal service returns overlay-specific addressing sufficient to establish the requested communication – namely, the route director’s address, the destination device’s private address, and the destination device’s virtual network (non-routable) address– as expressly recited in

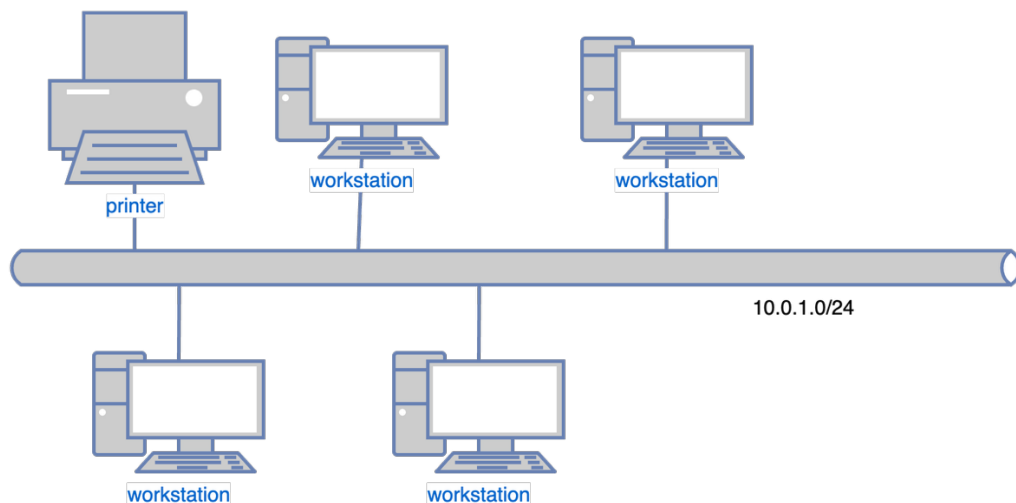
the claims (e.g., Ex. 1001, claims 16 and 44). Accordingly, in the context of the patent, “DNS” encompasses both conventional public DNS resolution of the community’s domain name and the manager’s authoritative resolution within the virtual network, which together provide name-to-address mapping and the additional routing information required by the claimed overlay.

37. **The Domain Name.** A POSITA would understand that, as used in the ‘785 patent, “a domain name” carries its plain, conventional networking meaning: an ASCII string identifier (typically a fully qualified domain name) used within the Domain Name System to identify Internet resources and to resolve to an IP address. The specification confirms this ordinary usage by explaining that the Internet “uses ASCII strings called domain names” and that DNS converts a domain name to an IP address (Ex. 1001, 1:47–48). Nothing in the intrinsic record provides a special or idiosyncratic definition, and the claim language uses “domain name” consistently with the DNS context. Accordingly, a skilled artisan would read “a domain name” to mean a DNS domain name, not an application-specific label or user handle.

38. **The Domain.** The ‘785 patent uses “a domain” in the DNS sense –as the naming scope that supplies the logical name of the virtual community network (VCN) and the DNS names of its member hosts. The specification expressly states that the VCN’s logical name “is a domain name,” and that this may be either a fully qualified domain name or a “virtual domain name,” i.e., one served within the system.

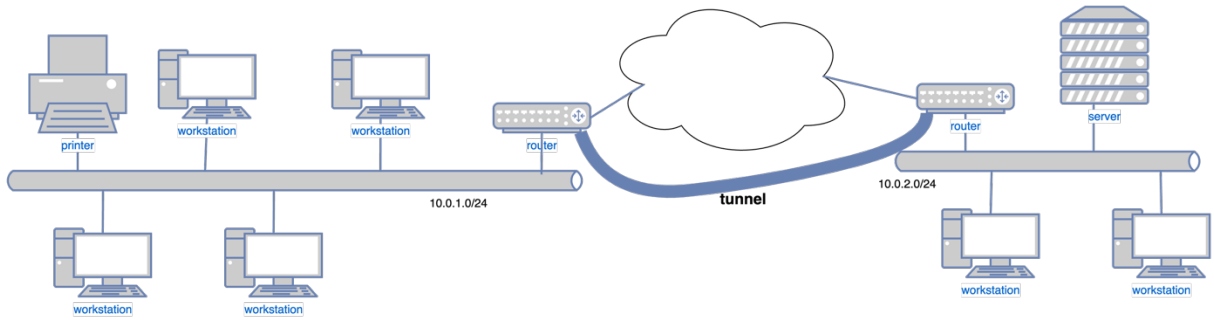
The VCN Manager is described as answering DNS for those virtual domain names and acting as their authoritative name server, reinforcing that “domain” here refers to the DNS namespace the system controls. Administrators set up the virtual domain and invite members by specifying each member’s DNS name, further confirming that member identity is grounded in DNS naming. The claims likewise tie the community definition to DNS by reciting a virtual network “defined by a domain name having an associated public network address.” Consistent with this usage, the patent discloses DNS lookups in which the VCN Manager returns not only the target member’s virtual IP address, but also overlay routing information (including the route director’s public address and the member’s private address), thereby using the domain as the governing DNS namespace for both community identification and member-to-member resolution.

39. **Private Networks.** A “Private Network” is a type of computer network consisting of nodes (which can be computer hosts, network switches or routers) and links (e.g., ethernet, wi-fi). It is private in the sense that it uses non-routable private IP addresses. Private IP ranges are defined by RFC 1918 (e.g., 10.0.0.0/8, 192.168.0.0/16). Private IP ranges are not routable (reachable) from the public internet. Figure 1, below, is an example of a Private Network with private IP addresses. Ex. 2013, at Pages 3-4 (Sections 2-3).



*Figure 1. An example of private network with private IP addresses.*

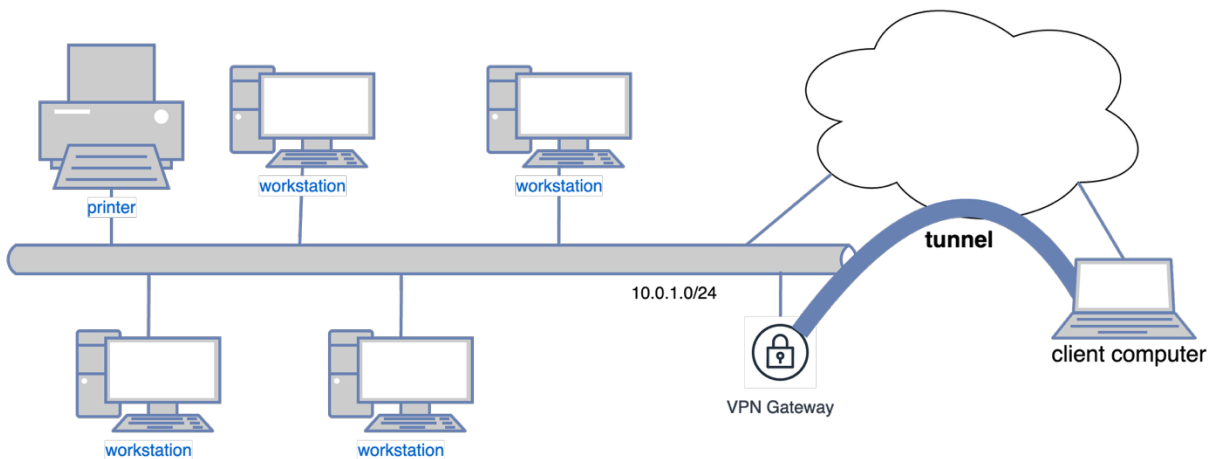
40. A “Virtual Private Network” (“VPN”) is a Private Network where some of the links are virtual, meaning they connect two nodes by tunneling traffic over other networks such as the Internet. This is done transparently to the nodes in the VPN. Figure 2, below, is an example of a VPN with two branches, each with private IP addresses, connected to a VPN tunnel. In the example shown in Figure 2, the two branches are “connected” via a VPN tunnel that terminates at a router on each branch. Ex. 2014, Pages 4-6. A POSITA would understand that this is a typical layout for a general purpose VPN with two branches.



*Figure 2. An example of virtual private network with two branches (each with private IP addresses) connected by a VPN tunnel.*

41. Often, the term “VPN” refers to the networking technology that enables the seamless connection of multiple computers that are not directly connected by a physical network. With this technology, computers on the VPN appear to be on the same network/subnet. Ex. 2014, Page 5.

42. Figure 3 is an example of a Private Network in which a VPN client, hosted on a client computer, initiates a VPN connection and establishes a VPN tunnel that terminates at a VPN Gateway on a network. Ex. 2014, Page 8. In the illustrated example, the VPN gateway is on a distributed network that includes a number of workstations and a printer. The client computer, in this example, could be operated by a network administrator performing maintenance on one of the workstations. In both cases, the VPN is initiated by a VPN client, hosted on a client computer, and terminates at a point on the network or branch (a router in Figure 2 and a VPN gateway in Figure 3).



*Figure 3. An example of private network with a client remotely connecting using a VPN tunnel.*

43. There are multiple flavors and protocols to build a VPN. Ex. 2014, at Pages 5 and 7. For example, the VPN client software can create a virtual network interface (on the client computer) whose IP address belongs to the VPN subnet. In this example, the VPN client encrypts and encapsulates IP packets sent over this interface (adding an IP header), then routes them over untrusted networks to the VPN server/gateway that decapsulates and decrypts the original IP packet sent over the virtual interface and injects them into the network (Figure 3). Another example is when a company has two branches and makes them appear on the same virtual private network by using VPN gateways establishing a VPN tunnel between the two branches (Figure 2). Such VPN can be established, for example, using IPsec in tunnel mode.

44. A VPN does not requires the existence of a domain or the connecting device to have a domain name. All a VPN enables is remote computers to act as if

they are physically connected via a wire when they are in fact connected via the Internet.

**C. Considered Claims of the '785 Patent**

45. **Claim 1:** A virtual network system, comprising:

a virtual network manager implemented with a first device memory and a first device processor of a first computing device, 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, the virtual network manager further configured to distribute a virtual network address to a device when the device is registered in the virtual network;

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

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.**

46. **Claim 30:** A virtual network manager, comprising:

a network interface configured for data communication via a virtual network that is **defined by a domain name** having an associated public network address;

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 address; and

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.**

47. **Claim 38:** A virtual network system, comprising:

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**, 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;

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;

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

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. **Claim 48:** A computer-implemented method, comprising:

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, each device in the virtual network being identified to the other devices by the virtual network address that is associated with the device;

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; and

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.**

49. **Claim 62:** One or more processor readable storage media devices comprising 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 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;

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;

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

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.**

50. **Claim 75:** A virtual network system, comprising:

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, 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 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 accessible by devices in the user set and devices outside of the user set, and 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.**

#### V. LEVEL OF A PERSON HAVING ORDINARY SKILL IN THE ART

51. I have been advised that the '785 patent application was filed on March 31, 2003, and issued on May 24, 2011.

52. I have been advised that a “person of ordinary skill in the art” (POSITA) is a hypothetical person to whom one could assign a routine task with reasonable confidence that the task would be successfully carried out. I have been advised that the relevant timeframe is prior to March 31, 2003. When I refer to “March 2003,” I am referring specifically to prior to March 31, 2003.

53. In my opinion, a person of ordinary skill in the art in March 2003 would have a bachelor’s degree in computer science or electrical engineering, and would have one or two years of experience in computer networking or equivalent educational and professional experience. I base this opinion on my experience in the networking

industry in general, and with computer networking specifically, in March 2003. At that time, those thinking about or developing improved systems or ways for secure communications networks would have at least this level of experience and knowledge.

## **VI. PRIOR ART RELIED ON IN THE PETITION**

### **A. Caronni-I ('941)**

54. Caronni-I presents a system and method for separating addresses from delivery schemes in virtual private networks (VPNs) to solve efficiency problems in enterprise network access. Traditional VPNs enable secure remote access to enterprise networks (“by using tunneling, remote device D<sub>1</sub> 108 may communicate and utilize the resources of the enterprise network 102 in a secure manner”), but suffer from inefficiencies. Ex. 1003, 2:41-57. The patent specifically identifies that, “[a]lthough VPNs alleviate the problem of geographic restrictiveness, they impose significant processing overhead when two remote devices communicate.” *Id.* at 2:47-49. This overhead occurs because traffic between remote devices must be routed through the enterprise network, requiring multiple encryption/decryption steps. *Id.* at 2:41-57.

55. The Caronni-I patent, therefore, aims to “provide addressing functionality that easily integrates and supports existing infrastructure services while at the same time allows for multiple delivery schemes” *Id.* at 3:8-11. The key technical approach consists of proposing a “Supernet”, meaning a virtual network where internal addresses are not associated with delivery schemes. *Id.* at 13:36-59, Claim 1.

This separation allows runtime changes to how packets are delivered (anycast, multicast, or unicast) without changing addresses *Id.* at 14:23-27, Claims 2-3.

56. The claims focus on this address/delivery separation. Claim 1 defines the core method where “the internal address is not associated with the delivery scheme.” Claim 2 enables “changing, at runtime, an association between the internal address and the delivery scheme.” Claims 4-6 require security mechanisms (authentication, encryption, access control). Claim 7 describes address mapping lookup and adding “the external address of the destination node to the packet.”

57. As illustrated in Figure 3, Caronni-I discloses a data processing system comprising multiple devices (elements 302–312) connected to the Internet (element 314). *Id.* at 4:64–66. These devices include nodes (elements 316, 318, 320, 322) that collectively form the “Supernet,” an overlay network built on top of standard Internet infrastructure. *Id.* at 4:66–5:3. The nodes operate within the devices themselves, enabling secure communication and shared use of the Supernet’s resources. *Id.* at 5:3–7. The system also includes a dedicated administrative node (e.g., element 306) that manages Supernet operations. *Id.* at 5:11–13.

58. Caronni-I further explains that the Supernet utilizes an internal address space that is separate from the public IP address space used by the Internet. *Id.* at 6:7–12. When a packet is transmitted from one Supernet node to another, it traverses the public Internet, and the system performs address translation between internal node

identifiers and public IP addresses. *Id.* at 6:12–16. This translation process is central to the Supernet architecture and is designed to be transparent to the user-level application. *Id.* at 6:6–7; 8:34–37.

59. Additional detail is provided in Figure 5, which depicts internal components of devices, such as element 302 and the administrative node 306. Each includes a CPU (elements 510, 512) capable of operating in both user mode and kernel mode. See *Id.* at 6:55–57. When operating in user mode, the CPU restricts direct access to hardware components; in contrast, kernel mode permits direct hardware manipulation. *Id.* at 6:57–60.

60. Notably, memory 504 of the administrative node includes both a TCP/IP stack (552) and a Virtual Address Resolution Protocol Daemon database (VARPDB, 551), both of which execute in kernel mode. *Id.* at 6:57–64. The VARPDB maintains mappings between internal Supernet addresses (referred to as “node IDs”) and external public network addresses (“real addresses”). *Id.* at 7:5–9. When a Supernet node initiates communication with another node, the inner IP layer 540 of the sending device generates a packet that includes a virtual source address (642), a virtual destination address (644), and associated data (654). *Id.* at 11:26–30.

61. The packet is passed to the SNSL layer (542) via a modified socket structure (step 806), which then queries the VARPDB to resolve the virtual addresses to real public addresses (step 808). *Id.* at 11:47–59. If necessary, the local VARPD

daemon will contact a remote VARPd server to complete the resolution. The address translation is performed transparently, without the involvement of the user-level application. *Id.* at 6:6–25; 8:31–37. This architectural choice is identified as a critical aspect of the Supernet’s security model.

## **B. Caronni-II**

62. Caronni-II describes a method to extend virtual network capabilities to bypass packet filtering and Network Address Translation (NAT) boxes. This is achieved through encapsulation of packets into higher-level protocols. As stated in the abstract: “Messages intended for a virtual destination address located on a network equipped with a device performing packet filtering, network address translation or a similar function on the edge of the network (an ‘edge device’), are encapsulated in higher level protocols prior to being sent to the edge device.” Ex. 1004, Abstract.

63. The Caronni-II patent identifies a problem with private networks: Virtual network messages cannot reach destinations behind firewalls, NAT boxes, or packet filtering devices because these edge devices block packets at the Network layer unless explicitly allowed by system administrators. *Id.* at 1:60-2:27. The key innovation in the Caronni-II patent, therefore, consists of encapsulating virtual network messages in higher-level protocols that can be configured by regular users rather than system administrators. The abstract explains: “Higher level protocol designations, including transport protocol designations accompanied by a port number

and application protocol designations, are retrieved from an extended virtual address registration.” *Id.* at Abstract. When messages arrive at the edge device, they are “passed up the Internet Protocol model stack to a higher layer. The higher layers of the edge device, such as the Transport layer and the Application layer, may be accessible and therefore configurable to a non-Systems Administrator thus allowing the message to reach the intended virtual destination address.” *Id.*

64. Caronni-II addresses NAT traversal by “encapsulating messages sent to a virtual destination address in higher level protocols” *Id.* at 3:50-51. The system uses an extended virtual address registration that includes “a real IP address, a port number, a transport protocol designation and an Application layer protocol designation.” *Id.* at Claim 12. When a destination is behind NAT, messages are sent to the edge device’s real IP address where “[t]he higher layers of the edge device, such as the Transport layer and the Application layer, may be accessible and therefore configurable to a non-Systems Administrator thus allowing the message to reach the intended virtual destination address.” *Id.* at 2:47-51. When both the originator and destination of a message are located behind NAT boxes, the patent specifies using a “reflecting agent being located outside a Network Address Translation (NAT) box that both the virtual address destination and virtual address said message is originating from are located behind” *Id.* at Claim 11.

65. In contrast to the Caronni-I Supernet architecture, the Caronni-II patent employs a distinct and flexible approach to virtual address resolution. Specifically, a process executing on the electronic device (element 52) registers its virtual address with a virtual address resolution facility (element 22). Ex. 1004, 4:54–56. This registration includes a corresponding real IP address associated with a physical device to which messages addressed to the virtual address should be delivered. *Id.* at 4:56–58.

66. The resolution information is stored in a Virtual Address Resolution Protocol (VARP) lookup table (element 28), which may be maintained at any location accessible over the network, allowing for a highly decentralized and distributed resolution system. *Id.* at 5:11–13. As shown in Figure 2 and described in the specification, the VARP table is structured with virtual addresses in the first column and their corresponding associations in the second column. *Id.* at 5:13–15. Each entry includes not only the virtual IP address (90) but also the real IP address (91), transport protocol designation (92), port number (93), and application-layer protocol (94). These associations are submitted at the time of virtual address registration. *Id.* at 5:18–24. In the case of electronic device 52, which is associated with an edge device (element 30), a process executing on the edge device may register the virtual address along with the real IP address of the edge device. As a result, messages directed to the

virtual destination are resolved using the real IP address of the edge device 30. *Id.* at 5:36–38.

67. The patent also addresses more complex network topologies involving NAT traversal. As shown in the embodiment of Figure 5, when two devices –each located behind separate NAT boxes– attempt to communicate within the same virtual network, a reflecting agent is introduced. *Id.* at Fig. 5, 8:1–4. In such cases, direct addressing between the devices is not possible, as the destination device’s address may not be publicly routable. *Id.* at 8:13–18. To facilitate communication in this context, the VARP table may include an entry for a reflecting agent located outside the NAT environment. *Id.* at 8:17–19. Accordingly, messages originating from a virtual address behind a NAT box are routed to the reflecting agent, which then relays (or “reflects”) the message to the destination behind a different NAT box. *Id.* at 8:19–23.

### **C. Hipp**

68. The Hipp patent describes Virtual Network Environments (VNEs) to isolate sets of applications from other applications on the same node or network, specifically for hosted application environments where multiple customers’ applications may run on shared infrastructure.

69. Hipp addresses security challenges of “leasable online computing infrastructure” where multiple customers share computing resources. Ex. 1005, 1:22-

37. “In hosted environment, one or more applications may be running on a shared computer or network at any given time. These applications may belong to the same customer/user or they may belong to different even competing customers/users.” *Id.* at 1:46-51. “Security measures are necessary to ensure that applications do not interfere with each other, either intentionally or unintentionally.” *Id.* at 1:53-55. The Hipp patent further notes: “A firewall is useful in separating a computer or group of computers in a network setting from computers beyond the firewall, but cannot separate or insulate computers behind the firewall from each other.” *Id.* at 1:56-60.

70. Hipp describes an application-level virtual network isolation method that enables multiple secure, isolated networks to coexist on the same physical infrastructure. “The Virtual Network Environment (VNE) of the present invention is defined by a collection of IP addresses. An application running within one VNE can communicate with another application in the same VNE. However, an application in one VNE cannot communicate with an application in another VNE (unless expressly permitted).” *Id.* at 2:8-13.

71. Unlike traditional firewalls that only protect network perimeters, Hipp’s VNE Framework operates at the kernel level to create Virtual Network Environments (VNEs) defined by IP address collections. Each VNE acts as an isolated network where applications can communicate freely within their VNE but are completely blocked from accessing applications in others. *Id.* at 4:2-3. The framework

transparently intercepts all network traffic, checking packet address destinations against VNE membership rules. *Id.* at 4:5-8. “The VNE framework 200 isolates an application within a VNE. Whenever an application running within the VNE communicates over a network connection, checks are made by the VNE framework 200 to ensure the remote address is either within the application’s VNE or is to an allowable destination. Any communication to an application in another VNE is not permitted (i.e. the packet is not sent and an error is returned).” *Id.* at 4:3-10.

“The VNE is specified at application run time. The VNE is transparent to the application and does not require any modifications to the application. The VNE is defined by subnet of addresses contained within the VNE. For example, all applications within the subnet 10.10.2.0 comprise a VNE. The subnet/netmask specifying such a VNE would be 10.10.2.0/255.255.255.0 and would include the addresses 10.10.2.0 through 10.10.2.255. In this example, an application with IP address 10.10.2.2 would be able to communicate with an application at address 10.10.2.60, but not at 10.10.0.1. Although using a subnet/netmask to specify the VNE is described herein for illustrative purposes, it is to be understood that other methods may be used to accomplish the same mechanism (e.g. an access control list).” *Id.* at 3:17-30.

This enables secure multi-tenancy at the application level rather than requiring physical separation, allowing service providers to dynamically place customer applications on any available hardware while maintaining complete isolation between different customers’ applications.

72. Hipp also describes that a virtual network identity is achievable as a by-product of the VNE: “Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up

the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application." *Id.* at 6:1-15.

73. "The virtual hostname resolves to the virtual IP address for both the applications registered with the VNI framework as well as those that are not registered. This may require configuration of a name service or OS host configuration files. For example, if an application instance used a virtual IP address of 10.10.0.1 and a virtual hostname of host1055, the standard hostname to IP address resolution mechanisms (e.g. DNS or the /etc/hosts file) would have to be preconfigured to resolve a query of host1055 to IP address 10.10.0.1." *Id.* at 6:15-26.

#### **D. RFC-1383**

74. RFC 1383 describes an approach to IP routing that leverages the Domain Name System (DNS) to scale routing. Specifically, it generalizes the concept of MX records to create routing-specific RX records. The goal is to reduce routing table complexity and support multi-homed networks without significantly altering

existing IP infrastructure. “The current proposal presents a scheme that allows for simple routing. It is complementary with the classic ‘hierarchical routing’ approach but provides an easy to implement and low cost solution for ‘multi-homed’ domains. The solution is a generalization of the ‘MX record’ scheme currently used for mail routing.” Ex. 1006, Page 2.

75. The primary problem addressed by the RFC-1383 publication is the “routing explosion” caused by the growth of routing tables in the Internet, which consumes increasing resources in routers. It notes that traditional hierarchical routing approaches, such as Classless Inter-Domain Routing (CIDR), are insufficient for efficiently managing routing information for multi-homed networks or those frequently changing topologies. A solution is sought that simplifies routing management, particularly for these complex scenarios, without imposing major structural changes. *Id.*

76. The RFC-1383 proposal uses reverse DNS lookup from in-addr.arpa domain to retrieve the IP addresses of gateways to reach a target IP address. In other words, given an IP address, it allows the retrieval of the IP address of a gateway that allows the requestor to reach the target IP address. To do so, it proposes the introduction of a DNS-based routing mechanism using “RX records,” analogous to existing MX records, enabling routers to query DNS to identify gateway IP addresses

dynamically. It also suggests the use of source routing or tunneling to reach the target IP address through the gateway IP address. *Id.* at Pages 2-3, 11-12.

77. “We propose to generalize this scheme for packet routing. Suppose a routing domain D, containing several networks, subnetwork and hosts, and connected to the Internet through a couple of IP gateways. These gateways are dual homed: they each have an address within the domain D -- say D1 and D2 -- and an address within the Internet -- say I1 and I2 --. These gateways also have a particularity: they retain information, and don’t try to announce to the Internet any reachability information on the networks contained within ‘D’. These networks however have been properly registered; a name server accessible from the Internet **contains the ‘in-addr.arpa’** records that enable reverse “address to name” lookup, and also contains the network level equivalent of ‘MX records’, say ‘RX records’. Given any host address Dx within D, one can get “RX records” pointing to the Internet addresses of the gateways, I1 and I2. A standard Internet router Ix **cannot in principle send a packet to the address Dx**: it does not have any corresponding routing information. However, if the said Internet router has been modified to exploit our scheme, **it will query the DNS with the name build up from ‘Dx’ in the ‘in-addr.arpa’ domain, obtain the RX records, and forward the packet towards I1 (or I2), using some form of ‘source routing’**. The gateway I1 (or I2) will receive the packet; its routing tables contain

information on the domain D and it can relay the packet to the host Dx.” *Id.* at Pages 2-3.

78. “An example of record would thus be:

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

which means 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.” *Id.* at Page 11.

79. It is important that note that **no virtual network is defined through a domain name**. Queries are made for IP addresses such as “\*.27.32.192.in-addr.arpa”.

## VII. CARONNI-I IN COMBINATION WITH CARONNI-II AND HIPP DOES NOT TEACH ‘785 CLAIMS 1, 30, 38, 48, 62, 75

80. Claims 1, 30, 38, 48, 62, and 75 of ‘785 all teach and explicitly state that the virtual community network “**is defined by a domain name**”. The DNS-defined structure of the Virtual Community Network (VCN) disclosed in the ‘785 Patent is central to the claimed invention. It provides the foundation for the hierarchical organization of network nodes and is critical to the process of address resolution. By leveraging DNS to define the VCN and delineate its subdomains (e.g., node1.X.VCN), the ‘785 Patent enables efficient, scalable, and secure communication between devices –whether connected directly to the Internet or located behind NAT

devices. This DNS-based hierarchy is not merely an addressing convention but a functional component of the network's operation and routing intelligence. None of the cited references discloses or suggests this architectural approach.

81. In my opinion, the '785 Patent discloses a Virtual Community Network ("VCN") architecture that is defined through a DNS-based naming scheme. Specifically, the VCN is identified by a domain name of the form X.VCN, and individual member devices are assigned sub-domains under that hierarchy during registration. Ex. 1001, 9:18–19, 28–31. For example, a member device may be assigned a domain name such as node node1.X.VCN, where "node1" is the device's unique identifier within the X.VCN domain. This hierarchical structure is not merely a naming convention but a functional mechanism for routing and address resolution within the virtual network.

82. In my opinion, the domain name X.VCN would be associated with public IP addresses, which enables devices –regardless of whether they reside within private networks (e.g., behind NAT devices) or are directly connected to the public Internet– to initiate communication with the VCN. Ex. 1001, 1:45-50. This structure allows seamless discovery and navigation to the VCN entry point using standard DNS mechanisms.

83. The '785 Patent describes domain names as ASCII strings that both identify members in human-readable form and can be resolved to the corresponding

IP address using a DNS server. Ex. 1001, 1:45–50. However, the use of domain names in the ‘785 Patent extends beyond conventional identification. In my opinion, the domain name reflects a hierarchical and logical organization of network entities within the virtual network. *See* ¶¶ 31-32 *supra*.

84. This DNS-based definition of the virtual network allows a virtual network manager to intercept communications destined for the VCN and to supply the source device with all necessary address information to complete delivery. This includes: (1) the address of the appropriate route director to handle routing decisions, (2) the public/private address of the destination device (private address if it resides behind a NAT), and (3) the virtual address of the destination device. Ex. 1001, 14:29-38.

85. Based on my review of Caronni-I, the Supernet is configured manually by a human administrator, who creates a configuration file that is stored on an administrative node and subsequently used by the SASD component when initializing or configuring the Supernet. Ex. 1003, 8:1–3. The configuration file in Caronni-I explicitly specifies several system-level parameters, including: “(1) the Supernet name, (2) all of the channels in the Supernet, (3) the nodes that communicate over each channel, Supernet, (4) the address of the KMS for each channel, [and] (5) the address of the VARPD that acts as the server for the Supernet . . . .” *Id.* at 8:3–7.

86. As further described in Caronni-I, the Virtual Address Resolution Protocol Daemon (VARPD) stores mappings in an associated VARPDDB between real IP addresses and Supernet IDs. *Id.* at 7:5–7, 23–25; 11:49–53. These Supernet IDs serve as internal identifiers for the nodes and specify, among other things, the channels over which particular processes are allowed to communicate. *Id.* at 11:3–4. To join a Supernet, Caronni-I discloses that a user must manually input the Supernet name, their user ID, and a password into their device. *Id.* at 9:66–10:2.

87. While Caronni-II adds the concept of bypass packet filtering and Network Address Translation boxes, both Caronni-I and Caronni-II lack any support, reference, or suggestion to the use of DNS. Caronni-I states: “The ‘node ID’ may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1).” *Id.* at 7:10-13. It also states: “To configure a Supernet, a system administrator creates a configuration file 558 that is used by SASD 540... This file may specify: (1) the Supernet name, (2) all of the channels in the Supernet, (3) the nodes that communicate over each channel...”. *Id.* at 8:1-5. The “Supernet name” mentioned in configuration is merely an administrative label in a config file, not a domain name used for network definition or operation. This is reiterated throughout the patent. For example, “[t]he first step performed is that the user invokes the SNlogin script and enters the Supernet

name, their user ID, their password, and a requested virtual address (step 702).” *Id.* at 9:67-10:3.

88. When Caronni-I discusses alternative addressing, it refers to non-IP protocols, not domain names: “Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6” *Id.* at 7:13-16. When referring to the Email Example, “other addresses (not IP) may be used for delivery within the Supernet. For example, an e-mail address may be used to deliver data in a Supernet. The sender node specifies an e-mail address as the delivery address. When the VARPD is queried... the VARPD provides the real IP address associated with the e-mail address.” *Id.* at 7:56-62. Even when using email addresses, the system translates to IP addresses, not domain names.

89. Hipp describes an application-level virtual network isolation method that enables multiple secure, isolated networks to coexist on the same physical infrastructure. It proposed a Virtual Network Environment (VNE) concept. “The Virtual Network Environment (VNE) of the present invention is defined by a collection of IP addresses.” Ex. 1005, 2:8-9.

90. Hipp refers to a “Virtual Network Identity.” However, this is a by-product of VNE membership, not what defines the VNE itself. The Virtual Network Definition (what creates the VNE) “is defined by a collection of IP addresses.” *Id.* at

2:8-9. “The VNE is defined by subnet of addresses contained within the VNE. For example, all applications within the subnet 10.10.2.0 comprise a VNE. The subnet/netmask specifying such a VNE would be 10.10.2.0/255.255.255.0 and would include the addresses 10.10.2.0 through 10.10.2.255.” *Id.* at 3:19-24. “The VNE is specified at application run time. The VNE is transparent to the application and does not require any modifications to the application. The VNE is defined by subnet of addresses contained within the VNE.” *Id.* at 3:17-20.

91. In Hipp, “subnet/netmask” is simply the conventional IP network specification used to define membership in a virtual network environment (VNE). Hipp says a VNE is “defined by [a] subnet of addresses,” and gives the concrete example 10.10.2.0/255.255.255.0, which denotes the block of IPs from 10.10.2.0 through 10.10.2.255; any application whose IP falls in that block is in the same VNE. The VNE parameters stored per process include both the “virtual subnet” and its “virtual mask,” as well as a “global virtual address subnet/mask” (e.g., 10.10.0.0/255.255.0.0) used to distinguish traffic to other VNEs versus the external Internet. Figures 6 and 7 make the usage explicit with the legends “WNET: VIRTUAL SUBNET ... VMASK: VIRTUAL SUBNET MASK ... GVNET ... GVMASK” and show the mask-based tests the framework performs to decide whether to allow, drop, or treat traffic as remote. Put plainly: Hipp uses the standard “network/mask” pair to do bitmask membership checks for VNE isolation and routing decisions.

92. IPv4 carved up the address space; the netmask is the rule that actually tells you where the network/host boundary is. In classful IPv4, the first few bits of an address fixed a default mask: Class A used /8 (255.0.0.0), Class B used /16 (255.255.0.0), and Class C used /24 (255.255.255.0). So, for example, any address beginning with 10.x.y.z is a Class A network by legacy convention, which implies a default mask of /8. A subnet is simply a network that uses a mask equal to or longer than the default mask for its class. You “borrow” host bits to create smaller networks. Concretely, 10.10.2.0/24 is a subnet carved out of the Class A block 10.0.0.0/8: the class would default to /8, but using /24 narrows the scope to 256 addresses (10.10.2.0–10.10.2.255) inside that larger classful space. Routers determine local vs. remote delivery by ANDing the destination with the mask; using a longer mask just changes where that boundary falls.

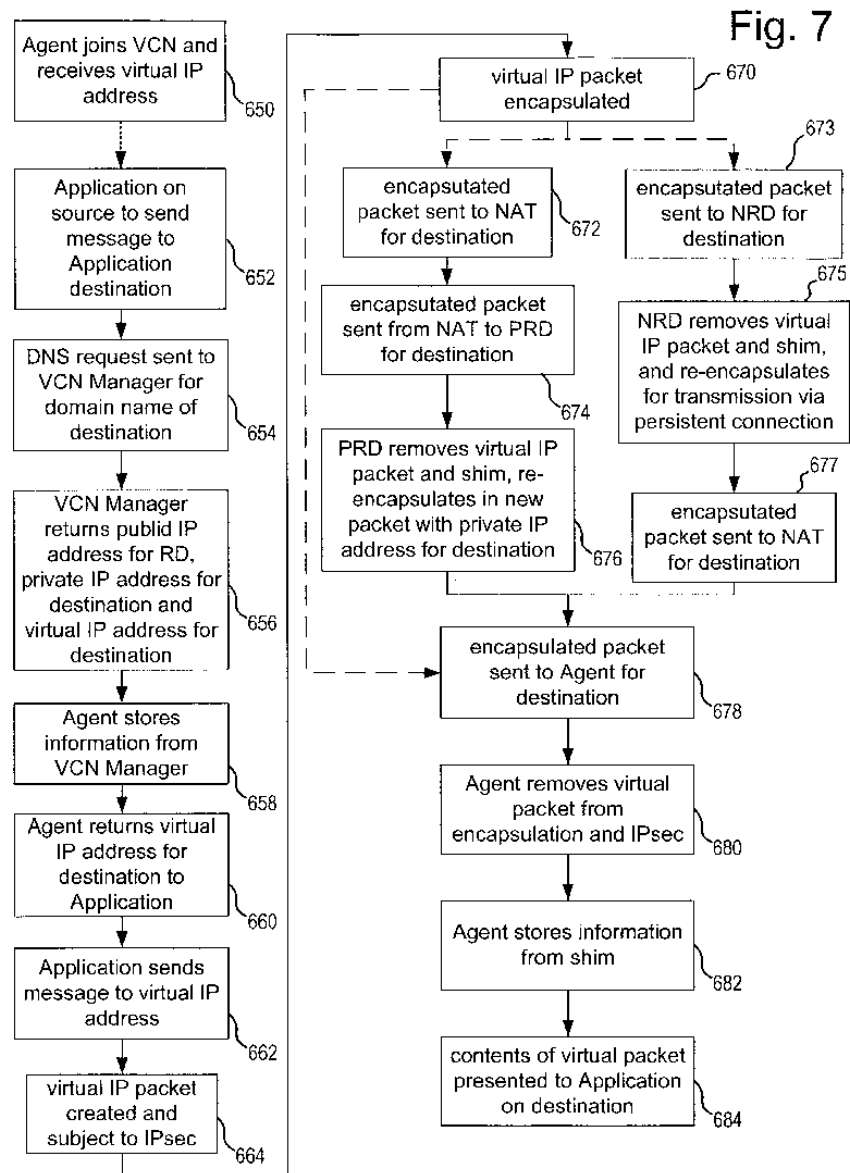
93. When Hipp refers to Virtual Network, it explicitly states: “Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on.” *Id.* at 6:1-6. This is what an application gets after joining a pre-existing VNE. It is the application’s identifier within the network.

94. In conclusion, Caronni-I combined with Caronni-II and Hipp do not teach a virtual community network that “**is defined by a domain name.**”

95. I also note that all independent claims 1, 30, 32, 48, 62, and 75 of ‘785 include elements that the DNS returns three addresses. They are unified in Fig. 7 as 656 and explained in the Patent as follows: “In step 656, 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. In the example described above, step 656 includes returning the public IP address for private Route Director 530, the private address for M<sub>B</sub> and a virtual IP address for M<sub>B</sub>. If the destination machine has a public IP address (and, thus, does not use a routing director) then step 656 will only include returning the public IP address for the destination. If the destination machine is in a private network that uses a Network Route Director (e.g., NRD 520), then step 656 will return the public IP address for the Network Route Director.” Ex. 1001, 14:34-46.

96. The ‘785 Patent provides a systematic solution to all scenarios. For instance, it considers both the case where the Route Director is inside the private network “Private Route Director (‘PRD’) 530 is an element that runs inside a private network enabling access to machines inside the private network from machines outside the private network.” *Id.* at 11:13-16. When it is located on the public Internet, “[t]he Network Route Director (NRD) 520 is a stand-alone unit that runs on the public

side of the Internet enabling Member Agents and Group Agents (discussed below) to be reached inside one or more private networks from the public network.” *Id.* at 11:9-145. This unified solution would not be obvious to a POSITA.



97. Caronni-I combined with Caronni-II and Hipp do not teach that any server provides a response with **both a Virtual and a Private IP address**, let alone all three addresses in response to a single query.

98. Based on my review of Caronni-I, the reference teaches an address translation mechanism that is intended to be transparent to the user-level operation of the nodes within the Supernet. To implement this, Caronni-I requires that an administrator, at the time of Supernet configuration, store a mapping between virtual node addresses and their corresponding public (real) IP addresses in a database known as the VARPDB. Ex. 1003, 7:3–9.

99. When a user-space application on a source device initiates communication with a destination device, the inner IP layer (element 540) receives a packet from the originating node. Caronni-I states: “The packet contains virtual source node address 642, virtual destination node address 644, and data 654.” *Id.* at 11:29–30. The inner IP layer then appends a Supernet ID to the packet using a modified socket structure. *Id.* at 11:37–38. This combination -comprising the virtual addresses and Supernet ID- is passed to the SNSL layer for address resolution and routing. “The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure.” *Id.* at 11:47–48.

100. The SNSL layer then accesses the VARPDB to retrieve the mapping information necessary to perform address translation. As Caronni-I explains: “The

SNSL layer then accesses the VARPDB to obtain the address mapping between virtual source address 642 and the source real address 614, as well as the virtual destination node address 644 and the destination real address 616.” *Id.* at 11:49–53. Given that the SNSL layer already has access to the virtual source and destination addresses, it is apparent that the purpose of querying the VARPDB is to obtain the real (physical) IP addresses corresponding to those virtual identifiers.

101. In my opinion, the only information being retrieved from the VARPDB at this stage are the **real IP addresses** needed for actual delivery. The SNSL layer is not retrieving virtual addresses –since it already possesses them– but instead performs a lookup using the virtual addresses as keys to obtain the physical routing information. This process is performed entirely in kernel mode, outside of the user application’s control or awareness. As a result, the source node continues to communicate using only the virtual addresses; it is completely unaware that address translation is taking place behind the scenes. *Id.* at 6:6–25, 8:31–37. Accordingly, it is my opinion that Caronni-I does not disclose sending a response, to a single query, that includes **both the virtual and (private/public) physical addresses** of the destination device.

102. Based on my review of Caronni-II, I understand it to disclose an electronic device that includes a virtual address resolution facility, described as “software used to register, store, and resolve virtual address information for processes and applications executing on the virtual network.” Ex. 1004 at 4:6–9. Caronni-II

further explains that a virtual address resolution protocol (VARP) lookup table stores the virtual addresses, and that these stored addresses are used “to resolve the virtual address into the real IP address of the physical device to which messages may be sent.” *Id.* at 4:14–16.

103. In my opinion, Caronni-II operates in a manner similar to Caronni-I with respect to address resolution. Specifically, when a query is made using the virtual address of the destination device, the system returns **only the real IP address** of that device (or of an associated edge device), not the virtual address itself. The source device already possesses the virtual address before making the query, and the purpose of the lookup is solely to retrieve the corresponding physical address.

104. While the VARP lookup table in Caronni-II can store the real IP address of a reflecting agent, it is my opinion that both Caronni-I and Caronni-II share the same fundamental approach: they rely on the virtual address of the destination device for the resolution query and return only the physical address in response. Ex. 1002 at 11:49–53; Ex. 1003 at 4:14–16.

105. Hipp is cited in the Petition only for its tangential mention of DNS, and not for any disclosure of returning a physical address, a virtual address, or both, in response to a single query. Therefore, in my opinion, there is no combination of Caronni-I, Caronni-II, and Hipp that results in returning all three required addresses—

the address of the reflecting agent, the real IP address of the destination, and the virtual address of the destination— in response to a single query.

106. It also bears repeating that Caronni-I and Caronni-II are silent with respect to domain names and DNS requests, and DNS, in Hipp, is used in the traditional sense—not to resolve three addresses, as in the ‘785 Patent. In my opinion, Caronni-II discloses a NAT traversal system in which client nodes interact with reflecting agents and coordination servers using custom application-layer signaling, not DNS queries. Ex. 1003 at 5:36–41, 6:54–64. Caronni-I, by contrast, teaches a virtual address resolution mechanism (VARPD) that resolves a virtual address to a single real address, without returning multiple address types or exposing the process to the application layer. Ex. 1002 at 8:55–64, 9:47–54. Petitioner appears to rely on Hipp to fill this gap by referencing its disclosure of DNS; however, Hipp merely describes conventional DNS hostname resolution to a single IP address (Ex. 1005 at 6:20–26) and does not teach or suggest returning multiple distinct addresses (such as public, private, and virtual) for a destination node, nor does it address NAT traversal, reflecting agents, or coordination servers.

107. Furthermore, Caronni-II’s virtual address resolution table may store the address of a NAT device rather than the actual address of the destination device. Ex. 1003 at 5:30–38. This represents selective disclosure, not a complete enumeration of all address types. Nowhere does Caronni-II suggest that a single query would return

all three of the addresses required by the '785 Patent (reflector, destination public/private, and virtual), nor does it teach that such a design is necessary or desirable. In my opinion, this means that even if Caronni-I and Caronni-II were combined, the result would not produce the claimed functionality of returning all three addresses in response to a single resolution request.

#### **VIII. CARONNI-I IN COMBINATION WITH CARONNI-II AND RFC-1383 DOES NOT TEACH '785 CLAIMS 1, 30, 38, 48, 62, 75**

108. Claims 1, 30, 38, 48, 62, and 75 of '785 all teach and explicitly state that the virtual community network **"is defined by a domain name."** I note that Ground 2 of the Petition is nearly identical to Ground 1 with the replacement of Hipp with RFC-1383. RFC-1383, however, in my opinion, does not resolve the shortcomings of the Caronni patents and Hipp.

109. RFC-1383 does not introduce or suggest the definition of a virtual community network by a domain name, and therefore does not address the shortcomings of Caronni-I and Caronni-II in teaching Claims 1, 30, 38, 48, 62, and 75 of patent '785. While it uses the DNS, its focus is on mapping an IP address to another IP address. It uses reverse DNS lookup from in-addr.arpa domain to retrieve the IP addresses of gateways to reach a target IP address. In other words, given an IP address, it allows the retrieval of the IP address of a gateway that allows to reach the target IP address.

110. RFC-1383 states: “We propose to generalize this scheme for packet routing. Suppose a routing domain D, containing several networks, subnetwork and hosts, and connected to the Internet through a couple of IP gateways. These gateways are dual homed: they each have an address within the domain D -- say D1 and D2 -- and an address within the Internet -- say I1 and I2 --. These gateways also have a particularity: they retain information, and don’t try to announce to the Internet any reachability information on the networks contained within ‘D’. These networks however have been properly registered; a name server accessible from the Internet **contains the ‘in-addr.arpa’ records** that enable reverse “address to name” lookup, and also contains the network level equivalent of ‘MX records’, say ‘RX records’. Given any host address Dx within D, one can get “RX records” pointing to the Internet addresses of the gateways, I1 and I2. A standard Internet router Ix **cannot in principle send a packet to the address Dx**: it does not have any corresponding routing information. However, if the said Internet router has been modified to exploit our scheme, **it will query the DNS with the name build up from ‘Dx’ in the ‘in-addr.arpa’ domain, obtain the RX records, and forward the packet towards I1 (or I2), using some form of ‘source routing’**. The gateway I1 (or I2) will receive the packet; its routing tables contain information on the domain D and it can relay the packet to the host Dx.” Ex. 1006, Pages 2-3.

111. “An example of record would thus be:

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

which means 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.”

Ex. 1006, Page 11.

112. It is important that note that **no virtual network is defined through a domain name**. Queries are made for IP addresses such as “\*.27.32.192.in-addr.arpa”.

113. In my review of RFC-1383 (Ex. 1006), I understand it to describe an experimental routing mechanism in which custom RX DNS records are used to identify a single gateway IP **address** for reaching otherwise unadvertised or fringe networks. Ex. 1006 at p. 11. These RX records function in a manner similar to MX records used in email routing, returning a **single gateway address** along with a cost-based preference value. *Id.* at pp. 2–4, 11–12. RFC-1383 discusses the use of DNS-based lookup of RX records by source or edge routers to identify an appropriate ingress point into a disconnected or multi-homed domain. *Id.* at pp. 11–13. However, in my opinion, each RX record in RFC-1383 returns **only one IP address**, and there is no disclosure of returning multiple addresses of different types, or public/private address mapping. The system is designed to enable routing into otherwise unreachable

domains, not to facilitate peer-to-peer communication between devices behind NATs using a DNS response that includes multiple address types. *Id.* at pp. 11–13.

114. Neither Caronni-I nor Caronni-II discloses the claimed DNS-based resolution behavior recited in the ‘785 Patent. In my review, Caronni-II describes an application-layer NAT traversal mechanism that uses reflecting agents and coordination nodes to exchange addressing information, but it does not use DNS to retrieve multiple address types in a single query. Ex. 1003 at 8:1–39. Caronni-I, by contrast, performs transparent address resolution via a Virtual Address Resolution Protocol Daemon (VARPD), resolving a virtual address to a single real address. See Ex. 1002 at 8:34–37; 6:6–7. In my opinion, Caronni-I does not expose multiple addresses to the querying node and does not utilize DNS. Neither reference teaches nor suggests that a single DNS request to an administrative node would return the three address types required by Claim 1.

115. Based on my analysis, RFC-1383 does not cure these deficiencies. It contains no disclosure of a DNS query returning a reflecting agent address, a public/private NAT address, and a virtual address together. In my opinion, there is also no basis to combine RFC-1383 with Caronni-I and Caronni-II to arrive at the claimed invention. These references operate in fundamentally different contexts – RFC 1383 in routing to unadvertised domains via gateways, Caronni-II in NAT traversal via coordination servers, and **Caronni-I in virtual address resolution via**

**kernel-based logic.** Ex. 1002 at 8:34–37, 6:6–7; Ex. 1003 at 8:1–39; Ex. 1006 at pp. 11–13. In my opinion, a person of ordinary skill in the art would not have been motivated to combine these disparate systems in a way that results in the single-query, multi-address DNS response recited in the ‘785 Patent. Any such combination would require using the ‘785 Patent as a roadmap and reflects an improper hindsight reconstruction rather than an obvious solution grounded in the prior art.

## **IX. CONCLUSION**

116. For the foregoing reasons, it is my technical opinion that Claims 1, 30, 38, 48, 62, and 75 of the ‘785 Patent would not have been obvious over the cited references and that the Petition fails to demonstrate a reasonable likelihood of prevailing on at least Grounds 1 and 2 of the Petition.

**X. DECLARATION**

117. I, Dr. Guevara Noubir, declare that all statements made herein on my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.h. I declare under penalty of perjury that the foregoing is true and correct.

Executed on August 11, 2025, in Boston, Massachusetts.

  
\_\_\_\_\_  
Dr. Guevara Noubir