

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,

Petitioner

IPR2020-01678

U.S. Patent No. 8,451,844

PETITION FOR *INTER PARTES* REVIEW

UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104

I. INTRODUCTION.....	11
II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8	11
A. Real Party-in-Interest.....	11
B. Related Matters.....	11
C. Lead and Back-up Counsel and Service Information	12
III. GROUNDS FOR STANDING	12
IV. NOTE	13
V. SUMMARY OF THE '844 PATENT	13
VI. PROSECUTION HISTORY	14
VII. LEVEL OF ORDINARY SKILL IN THE ART	15
VIII. CLAIM CONSTRUCTION.....	15
A. “domain” (All claims)	15
B. “Means for” Limitations.....	17
1. “means for constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number”	18
2. “means for generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet”	19
3. “means for routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address”	20
4. “means for extracting the IPv4 destination address and the destination port number from the IPv6 destination address”	21
5. “means for generating an IPv4 data packet from the IPv6 data packet”	22

6.	“means for routing the IPv4 data packet to its IPv4 destination” ..	22
7.	“means for advertising to the IPv4 domain IPv4 addresses of network equipments connected to the IPv6 domain”	23
8.	“means for advertising in the IPv6 domain IPv6 addresses constructed from IPv4 addresses of the IPv4 domain”	23
9.	“means for obtaining an IPv4 address, a range of port numbers, and a range of authorized IPv6 addresses”	24
IX. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF		25
X. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE		25
A.	Challenged Claims and Statutory Grounds for Challenge	25
B.	Discretionary denial under 35 U.S.C. § 325(d) is not appropriate.....	27
C.	Discretionary denial under the <i>Fintiv</i> factors is not appropriate.....	28
1.	Whether evidence exists that a stay may be granted if a proceeding is instituted	28
2.	Proximity of the Court’s trial date to the Board’s projected statutory deadline for a final written decision.....	29
3.	Investment in the parallel proceeding by the Court and the parties	30
4.	Overlap between issues raised in the petition and in the parallel proceeding	30
5.	Whether the petitioner and the defendant in the parallel proceeding are the same party.....	31
6.	Other circumstances that impact the Board’s exercise of discretion, including the merits.....	31
D.	Discretionary denial under <i>General Plastic</i> is not appropriate	31

E.	Ground 1: Claims 1-6, 8 and 10 are obvious over Ananda, Wetterwald, and RFC4380	32
1.	Ananda	32
2.	Wetterwald	35
3.	RFC4380	35
4.	Reasons to combine Ananda, Wetterwald, and RFC4380	36
5.	Claim 1	36
6.	Claim 2	58
7.	Claim 3	60
8.	Claim 4	70
9.	Claim 5	71
10.	Claim 6	76
11.	Claim 8	82
12.	Claim 10	83
F.	Ground 2: Claim 7 is obvious over Ananda, Wetterwald, RFC4380, and RFC4271	84
1.	RFC4271	84
2.	Reasons to combine	84
3.	Claim 7	84
G.	Ground 3: Claim 9 is obvious over Ananda, Wetterwald, RFC4380, and RFC1918	87
1.	RFC1918	87

2. Reasons to combine Ananda, Wetterwald, RFC4380, and RFC1918 88

3. Claim 9 88

XI. CONCLUSION 90

CERTIFICATE OF WORD COUNT 92

CERTIFICATE OF SERVICE 93

PETITIONER’S EXHIBIT LIST

Ex.1001	U.S. Patent No. 8,451,844 to Mohamed Boucadair et al. (“’844 patent”)
Ex.1002	Prosecution History of U.S. Patent No. 8,451,844 (“’844 Prosecution History”)
Ex.1003	Declaration of Dr. Paul Min under 37 C.F.R. § 1.68
Ex.1004	<i>Curriculum Vitae</i> of Dr. Min
Ex.1005	U.S. Patent Application Publication No. 2004/0107287 to Ananda et al. (“Ananda”)
Ex.1006	Definition of “Domain,” Newton, Harry, “Newton’s Telecom Dictionary,” Sixteenth Expanded Edition, July 2000 (“Newton”)
Ex.1007	Y. Rekhter et al., “Address Allocation for Private Internets,” Internet Engineering Task Force (“IETF”) Network Working Group, Request for Comments: 1918 (Feb. 1996), available at https://tools.ietf.org/html/rfc1918 (“RFC1918”)
Ex.1008	U.S. Patent No. 7,639,686 to Wetterwald et al. (“Wetterwald”)
Ex.1009	C. Huitema, “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs),” Internet Engineering Task Force Network Working Group, Request for Comments: 4380 (February 2006), available at https://tools.ietf.org/html/rfc4380 (“RFC4380” or “Teredo”)
Ex.1010	U.S. Patent Application Publication No. 2005/0286553 to Wetterwald et al. (“’533 publication”).
Ex.1011	Reserved.
Ex.1012	“Internet Protocol – DARPA Internet Program Protocol Specification,” IETF Network Working Group, Request for Comments: 791 (J. Postel ed., Sep. 1981), available at https://tools.ietf.org/html/rfc791 (“RFC791”)
Ex.1013	R. Hinden et al., “IP Version 6 Addressing Architecture,” IETF Network Working Group, Request for Comments: 1884 (Dec. 1995), available at https://tools.ietf.org/html/rfc1884 (“RFC1884”)
Ex.1014	O. Jacobsen et al., “A Glossary of Networking Terms,” IETF Network Working Group, Request for Comments: 1208 (Mar. 1991), available at https://tools.ietf.org/html/rfc1208 (“RFC1208”)

Ex.1015	“Internet Users’ Glossary,” IETF Network Working Group, Request for Comments: 1983 (G. Malkin ed., Aug. 1996), available at https://tools.ietf.org/html/rfc1983 (“RFC1983”)
Ex.1016	U.S. Patent Application Publication No. 2006/0212572 to Afek et al. (“Afek”)
Ex.1017	European Patent No. EP1295454 to Soppera
Ex.1018	U.S. Patent No. 6,792,546 to Shanklin et al.
Ex.1019	Monarch Networking Solution LLC’s and Defendant Cisco Systems, Inc.’s Joint Claim Constructions and Prehearing Statement (P.R. 4-3), <i>Monarch Networking Sols. LLC v. Cisco Sys., Inc. et al.</i> , 2:20-cv-00015 (E.D. Tex. filed September 4, 2020)
Ex.1020	Plaintiff’s Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, <i>Monarch Networking Sols. LLC v. Cisco Sys., Inc. et al.</i> , 2:20-cv-00015 (E.D. Tex. served May 4, 2020)
Ex.1021	Reserved.
Ex.1022	R. Hinden et al., “IP Version 6 Addressing Architecture,” IETF Network Working Group, Request for Comments: 2373 (July 1998), available at https://tools.ietf.org/html/rfc2373 (“RFC2373”)
Ex.1023	R. Hinden et al., “Internet Protocol Version 6 (IPv6) Addressing Architecture,” Request for Comments: 3513 (Apr. 2003), available at https://tools.ietf.org/html/rfc3513 (“RFC3513”)
Ex.1024	R. Hinden et al., “IP Version 6 Addressing Architecture,” IETF Network Working Group, Request for Comments: 4291 (Feb. 2006), available at https://tools.ietf.org/html/rfc4291 (“RFC4291”)
Ex.1025	U.S. Patent Application Publication No. 2006/0227792 to Wetterwald et al.
Ex.1026	Amended Docket Control Order, <i>Snik LLC v. Samsung Elecs. Co., Ltd.</i> , No. 2:19-cv-387-JRG, ECF No. 52 (E.D. Tex.) (Gilstrap)
Ex.1027	Docket Control Order, <i>Trioptima AB v. Quantile Techs. Ltd.</i> , No. 2:19-cv-390-JRG, ECF No. 41 (E.D. Tex.) (Gilstrap)
Ex.1028	Second Amended Docket Control Order, <i>GREE, Inc. v. Supercell Oy</i> , No. 2:19-cv-00413-JRG, ECF No. 61 (E.D. Tex.) (Gilstrap)
Ex.1029	Docket Control Order, <i>SpaceTime3D, Inc. v. Samsung Elecs. Co., Ltd.</i> , No. 2:19-cv-00372-JRG, ECF No. 26 (E.D. Tex.) (Gilstrap)

Ex.1030	Third Amended Docket Control Order, <i>Blitzsafe Texas, LLC v. PACCAR Inc.</i> , No. 2:19-cv-00393-JRG-RSP, ECF No. 48 (E.D. Tex.) (Payne)
Ex.1031	Docket Control Order, <i>Blitzsafe Texas, LLC v. Navistar, Inc.</i> , No. 2:19-cv-00403-JRG, ECF No. 29 (E.D. Tex.) (Gilstrap)
Ex.1032	Docket Control Order, <i>Luminati Networks Ltd. v. TESO LT, UAB, et al.</i> , No. 2:19-cv-0395-JRG, ECF No. 65 (E.D. Tex.) (Gilstrap)
Ex.1033	Docket Control Order, <i>Gloria Toussaint, et al. v. Michael Smalling, et al.</i> , No. 2:20-cv-0013-JRG-RSP, ECF No. 8 (E.D. Tex.) (Payne)
Ex.1034	Reserved.
Ex.1035	Reserved.
Ex.1036	R. Droms, “Dynamic Host Configuration Protocol,” IETF Network Working Group, Request for Comments: 2131 (March 1997), available at https://tools.ietf.org/html/rfc2131 (“RFC2131”)
Ex.1037	U.S. Patent No. 8,089,968 to Filsfils et al. (“Filsfils”)
Ex.1038	U.S. Patent No. 7,274,704 to Ould-Brahim et al. (“Ould-Brahim”)
Ex.1039	U.S. Patent No. 6,728,782 to D’Souza et al. (“D’Souza”)
Ex.1040	Docket Control Order, <i>Monarch Network Sols. LLC v. Cisco Sys., Inc.</i> , No. 2:20-cv-0015-JRG, ECF. No. 51 (E.D. Tex.) (Gilstrap)
Ex.1041	RFC Editor Webpage, www.rfc-editor.org , as captured by the Internet Archive Wayback Machine on Apr. 30, 2008, available at https://web.archive.org/web/20080430211008/http://www.rfc-editor.org/
Ex.1042	S. Bradner, “The Internet Standards Process – Revision 3,” Request for Comments: 2026 (Oct. 1996), available at https://tools.ietf.org/html/rfc2026 (“RFC2026”).
Ex.1043	Declaration of Dr. Sylvia Hall-Ellis under 37 C.F.R. § 1.68
Ex.1044	<i>Curriculum Vitae</i> of Dr. Hall-Ellis

Ex.1045	“Transmission Control Protocol – DARPA Internet Program Protocol Specification,” IETF Network Working Group, Request for Comments: 793 (J. Postel ed., Sep. 1981), available at https://tools.ietf.org/html/rfc793 (“RFC793”)
Ex.1046	Y.Rekhter, <i>et al.</i> , “A Border Gateway Protocol 4 (BGP-4),” IETF Network Working Group, Request for Comments: 4271 (Jan. 2006), available at https://tools.ietf.org/html/rfc4271 (“RFC4271”).
Ex.1047	U.S. Patent No. 7,403,995 to Mace et al.
Ex.1048	U.S. Patent No. 7,277,453 to Chin et al.
Ex.1049	U.S. Patent No. 7,764,691 to Sehagal et al.
Ex.1050	U.S. Patent No. 7,720,976 to Chu et al.
Ex.1051	U.S. Patent No. 7,729,347 to Rosberg
Ex.1052	U.S. Patent No. 7,133,405 to Graham et al.
Ex.1053	U.S. Patent No. 6,917,618 to Thubert et al.
Ex.1054	Canadian Patent Application Publication No. CA2446579A1
Ex.1055	PCT Publication No. WO 01/27766
Ex.1056	Reserved.
Ex.1057	Reserved.
Ex.1058	U.S. Patent No. 7,080,151 to Borella et al.
Ex.1059	U.S. Patent No. 7,095,738 to Desanti
Ex.1060	U.S. Patent No. 8,045,558 to Droms et al.
Ex.1061	U.S. Patent No. 7,729,312 to Suzuki et al.

Ex.1062	U.S. Patent No. 7,961,725 to Nagarajan et al.
Ex.1063	U.S. Patent No. 7,813,843 to Patel et al.
Ex.1064	U.S. Patent No. 8,688,834 to Fujimoto et al.
Ex.1065	U.S. Patent No. 7,568,040 to Townsley et al.
Ex.1066	U.S. Patent No. 7,738,882 to Dutta et al.
Ex.1067	U.S. Patent No. 7,543,065 to Barkley et al.
Ex.1068	U.S. Patent No. 7,640,338 to Huitema et al.
Ex.1069	U.S. Patent No. 7,502,474 to Kaniz et al.
Ex.1070	U.S. Patent No. 8,245,304 to Chen et al.
Ex.1071	U.S. Patent No. 7,804,848 to Mirtorabi et al.

I. INTRODUCTION

Pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, Cisco Systems, Inc. (“Petitioner”) respectfully requests that the Board review and cancel as unpatentable under 35 U.S.C. § 103(a) claims 1-10 (the “Challenged Claims”) of U.S. Patent No. 8,451,844 (the “’844 patent,” Ex.1001).¹

The ’844 patent describes encapsulating Internet Protocol version 4 (IPv4) data packets for routing over an Internet Protocol version 6 (IPv6) network. Such techniques were known in prior art that the Examiner did not evaluate during prosecution of the ’844 patent. Petitioner asks the Board to institute review and cancel the Challenged Claims as unpatentable.

II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

A. Real Party-in-Interest

Under 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real party-in-interest is Cisco Systems, Inc.

B. Related Matters

As of the filing date of this Petition, and to the best knowledge of Petitioner, the ’844 patent is or has been involved in the following case:

¹ The pre-AIA statutory framework applies to the ’844 patent.

Case Caption	Filed	Number	Court
<i>Monarch Networking Solutions LLC v. Cisco Systems, Inc. and Charter Communications, Inc.</i>	Jan. 21, 2020	2:20-cv-00015	E.D. Tex.

C. Lead and Back-up Counsel and Service Information

<u>Lead Counsel</u> Theodore M. Foster HAYNES AND BOONE, LLP 2323 Victory Ave. Suite 700 Dallas, TX 75219	Phone: (972) 739-8649 Fax: (214) 200-0853 ipr.theo.foster@haynesboone.com USPTO Reg. No. 57,456
<u>Backup Counsel</u> David L. McCombs HAYNES AND BOONE, LLP 2323 Victory Ave. Suite 700 Dallas, TX 75219 Jonathan R. Bowser HAYNES AND BOONE, LLP 2323 Victory Ave. Suite 700 Dallas, TX 75219	Phone: (214) 651-5533 Fax: (214) 200-0853 david.mccombs.ipr@haynesboone.com USPTO Reg. No. 32,271 Phone: (202) 654-4503 Fax: (214) 200-0853 jon.bowser.ipr@haynesboone.com USPTO Reg. No. 54,574

Please address all correspondence in this proceeding to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

III. GROUNDS FOR STANDING

Petitioner certifies that the '844 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the patent

claims. 37 C.F.R. § 42.8(b)(1).

IV. NOTE

Petitioner cites to exhibits' original page numbers, unless otherwise noted. Emphasis in quoted material has been added throughout, unless otherwise noted.

V. SUMMARY OF THE '844 PATENT

The '844 patent is directed to an Internet Protocol (IP) telecommunications network for transmitting data packets from a source terminal identified by a source address to a destination terminal identified by a destination address. Ex.1001, 1:17-21. There existed two incompatible versions of Internet Protocol: IPv4² with 32-bit addresses and IPv6³ with 128-bit addresses. Ex.1001, 6:21-24; Ex.1012, 7 (IPv4); Ex.1013, 3 (IPv6); Ex.1003, ¶ 31.

The '844 patent describes “routing an IPv4 data packet coming from an IPv4 in an IPv6 domain,” where “said data packet includes an IPv4 destination address

² The IPv4 protocol was standardized by the Internet Engineering Task Force (IETF) in RFC791 in September 1981. Ex.1012; *see* Ex.1043, ¶¶ 49-52 (citing Exs.1051-1052).

³ The IPv6 protocol was standardized by the IETF in RFC1884 in December 1995 and revised in later years. Ex.1013; *see* Ex.1043, ¶¶ 53-56 (citing Exs.1048, 1053); *see also* Ex.1022-1024; Ex.1043, ¶¶ 69-80 (citing Exs.1058-1063).

and a destination port number.” Ex.1001, 2:61-3:11; Ex.1003, ¶ 32. The described approach includes “IPv6 domain access equipment to construct an IPv6 destination address routable in the IPv6 domain from the IPv4 destination address and the destination port and to generate an IPv6 data packet containing the constructed destination address” and “the payload carried by the IPv4 data packet.” Ex.1001, 3:9-18.

VI. PROSECUTION HISTORY

The ’844 patent was filed as U.S. Application No. 13/001,850 (“’850 Application”) on December 29, 2010. The ’850 Application claims priority to French Patent Application No. 0854398, filed on June 30, 2008. No priority date analysis is unnecessary, however, as the relied-upon prior art predates June 2008. Petitioner reserves the right to challenge the priority date of the ’844 patent in this or another forum where the issue becomes relevant.

Only one Office Action was issued during prosecution of the ’844 patent. Ex.1002, 37-40. The Office Action did not include any prior art rejections.

Following minor claim amendments, the Examiner allowed claims 1-10 with a statement of reasons for allowance emphasizing the step of “constructing an IPv6 destination address by concatenating an operator prefix, IPv4 destination address, and the destination port number” (corresponding to limitations [1.1], [3.0.2], [5.1], [6.2.2], and [10.0.2], respectively). Ex.1002, 16.

The cited prior art teaches these and all other features of the Challenged Claims.

VII. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) in June 2008 would have had a working knowledge of the packet switching art that is pertinent to the ’844 patent, including computer networking, IPv4, IPv6, packet encapsulation, and packet routing. A POSITA would have had a bachelor’s degree in electrical engineering, computer engineering, computer science, or a related subject, and three years of work experience in computer networking, including experience with the IPv4 and IPv6 protocols, or a related field. Lack of professional experience can be remedied by additional education, and vice versa. Ex.1003, ¶¶ 21-22.

VIII. CLAIM CONSTRUCTION

The Challenged Claims are construed “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b).

A. “*domain*” (All claims)

Petitioner does not believe that “*domain*” requires construction. To the extent that Patent Owner argues, or the Board determines, that construction is required, Petitioner submits that an appropriate construction is “network.”

Ex.1003, ¶ 43. The ’844 patent suggests that “domain” is essentially a synonym for

“network.” See, e.g., Ex.1001, 1:30-31, 1:40-44; Ex.1003, ¶ 44. However, Petitioner’s detailed claim analysis below shows how the claims are unpatentable under multiple constructions that Patent Owner might advance.

Various technical dictionaries offer define “domain” as:

- “the part of a computer network in which the data processing resources are under common control...” (Ex.1006, 196);
- “an administrative partition of a complex distributed system” (Ex.1014, 6; see also Ex.1014, 16 (defining “subnetwork”));
- “a heavily overused term” whose meaning depends on context, including referring to “collection of hosts and routers, and the interconnecting network(s), managed by a single administrative authority” or to the “routers exchanging routing information within” such a collection. Ex.1015, 17, 4, 48; Ex.1003, ¶ 45; see also Ex.1043, ¶¶ 61-64 (citing Exs.1054-1055).

The term “domain” should not be construed to require “common control” or a “single administrative authority” because the claims themselves identify the point of commonality for each “domain” by reciting “IPv4 domain” or “IPv6 domain.” Thus, terminals within an “IPv4 domain” communicate using the IPv4 protocol, while terminals within an “IPv6 domain” communicate using the IPv6 protocol. Ex.1003, ¶ 46. Even so, this Petition shows that features such as “common control”

would have been obvious. Ex.1003, ¶ 46.

In summary, no construction is believed necessary for “*domain*,” but to the extent a construction is adopted, the Board should construe “*domain*” as a “network.”

B. “Means for” Limitations

Claims 5-7, 9, and 10 recite “*means for*” limitations. Petitioner identifies the function and structure for each of the “*means*” terms below, in compliance with 37 C.F.R. § 42.104(b)(3), but in co-pending litigation Petitioner contends that in some cases the disclosed structure is insufficient under 35 U.S.C. § 112, rendering certain terms indefinite.⁴

Many of the “*means*” terms refer to structures shown in Figure 6a, described as “**a device [25] of the invention for receiving [a data packet] from an IPv4 domain in an IPv6 domain.**” Ex.1001, 6:4-6, 12:40-41. The device 25 includes “hardware elements conventionally found in a standard computer or a dedicated router,” including a **processor 25₁** and a **memory 25₅** that “stores the computer program of the invention.” Ex.1001, 12:53-55. The memory 25₅ also “contain[s] a database in which is stored an address translation table (NAT table) for translating

⁴ Patent Owner acknowledges that these terms should be construed under 35 U.S.C. § 112 ¶ 6. Ex.1019, 14-20.

addresses between the [IPv6] domain 1 and its [IPv4] home network 3.” Ex.1001, 12:48-50, 7:4-8. The ’844 patent describes a similar device 26 “**for sending [a data packet] from an IPv6 domain to an IPv4 destination address.**” Ex.1001, 6:7-9; *see also* Fig. 6b.

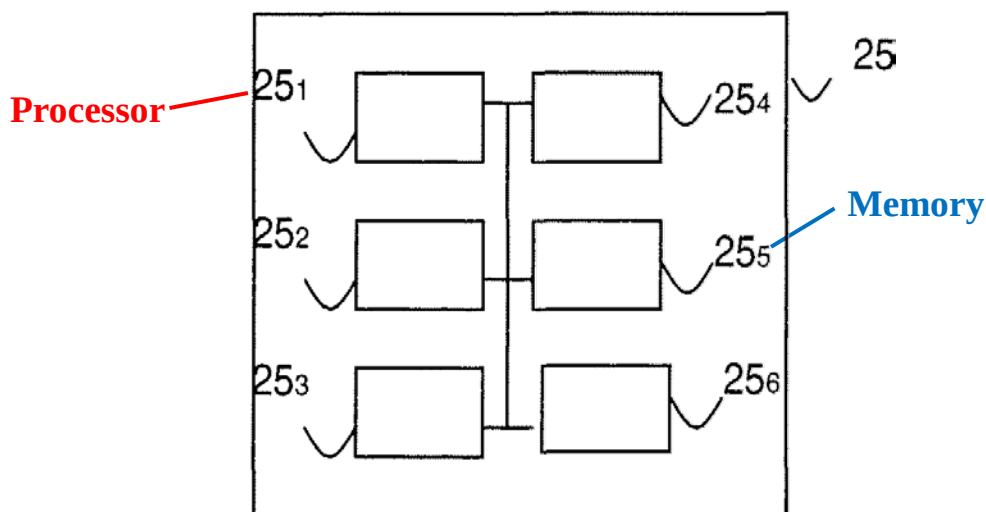


FIG. 6a

Ex.1001, Fig. 6a (annotated).

With reference to Figure 1, the ’844 patent describes that both the home gateway 20 and the access node 40 include devices 25 and 26. Ex.1001, 12:38-40, 12:58-60, 13:11-22.

1. “***means for constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number***”

Function: constructing an IPv6 destination address by concatenating an IPv6

prefix, an IPv4 destination address, and a destination port number.

Structure: the processor 25₁ of the device 25 and an address translation table (NAT table) of memory 25₅ within device 25 found within home gateway 20 or access node 40, and equivalents. Ex.1001, 12:38-57, Fig. 3 (step PA1).

Explanation: The '844 patent explains that the processor 25₁ executes computer program instructions stored in the memory 25₅ of the device 25 for constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number. Ex.1001, 10:25-30, 12:38-57, Fig. 3 (step PA1). The '844 patent also explains that memory 25₅ stores a NAT table “for translating addresses between the [IPv6] domain 1 and [the IPv4] network 3.” Ex.1001, 12:48-50, 7:4-8, Fig. 1.

2. “means for generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet”

Function: generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet.

Structure: the processor 25₁ of the device 25 within home gateway 20 or access node 40 executing computer program instructions stored in memory 25₅ of the device 25, including instructions for generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet, and equivalents. Ex.1001, 3:11-18, 10:38-53, 12:38-57, Fig. 3 (step PA2).

Explanation: The '844 patent explains that the processor 25₁ executes the computer program instructions stored in the memory 25₅ of the device 25 for generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet. Ex.1001, 5:23-28, 12:53-57.

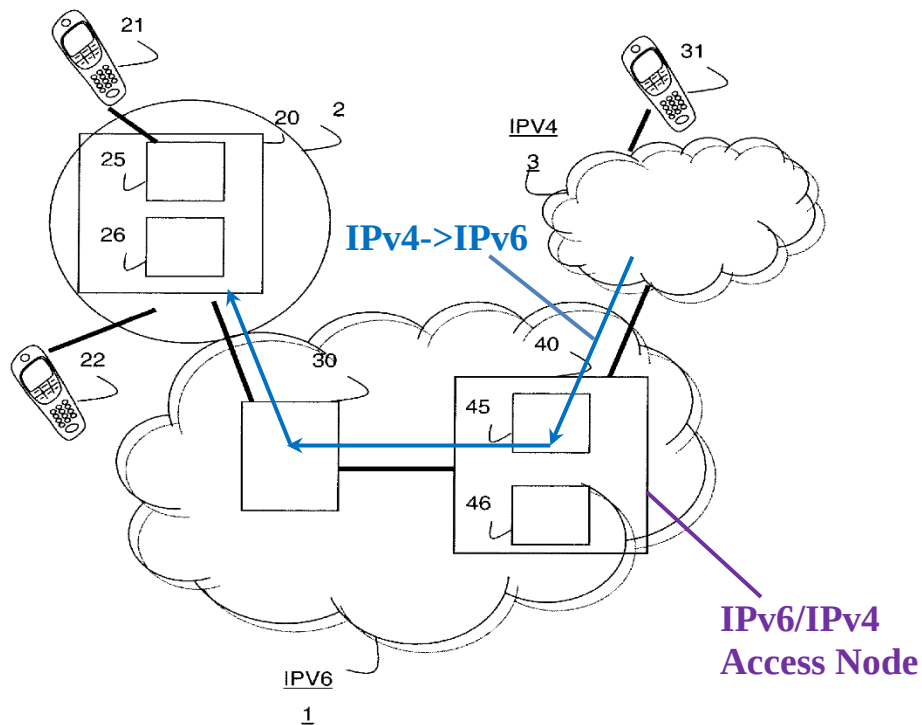
3. “means for routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address”

Function: routing the IPv6 data packet toward its destination.

Structure: IPv6 routing core of access node 40, and equivalents. Ex.1001, 10:59-11:2, Fig. 3 (step PA3).

Explanation: The '844 patent explains that the IPv6 data packet generated using the IPv6 constructed destination address is routed in the IPv6 domain 1. Ex.1001, 10:59-61, Fig. 3 (step PA3). The IPv6 packet “therefore enters the IPv6 routing core of the access node ... 40” and “is routed to the interface of the access node ... 40 that serves the corresponding IPv6 route to the home gateway 20,” e.g., via “telecommunications means 25₄” of device 25 within access node 40, as shown in annotated Figure 1 below.⁵ Ex.1001, 10:61-11:2, 12:38-46, 7:4-8.

⁵ Device 25 of access node 40 is mislabeled as “45” in Figure 1 of the '844 patent.



Ex.1001, Fig. 1 (annotated).

4. “means for extracting the IPv4 destination address and the destination port number from the IPv6 destination address”

Function: extracting the IPv4 destination address and port number from the IPv6 destination address.

Structure: the processor 26₁ of the device 26 and an address translation table (NAT table) of memory 26₅ within device 26 found within home gateway 20 or access node 40, and equivalents. Ex.1001, 12:29-34, 12:58-13:10.

Explanation: The '844 patent explains how device 26₁ within home gateway 20 or access node 40 (which includes processor 26₁ and memory 26₅) extracts an entire IPv4 packet (which would include a destination IPv4 address and destination

port number) from a received IPv6 packet addressed to a destination IPv6 address. Ex.1001, 12:29-34, 12:58-13:10. The '844 patent also explains that memory 26₅ stores a NAT table for translating between IPv6 and IPv4 addresses. Ex.1001, 12:67-13:3, 7:4-8.

5. “means for generating an IPv4 data packet from the IPv6 data packet”

Function: generating an IPv4 data packet from the IPv6 data packet, said packet including said IPv4 destination address and said destination port number.

Structure: the processor 26₁ of the device 26 within home gateway 20 or access node 40 executing computer program instructions stored in the memory 26₅ of the device 26, including instructions for generating an IPv4 data packet from the IPv6 data packet, and equivalents. Ex.1001, 13:8-10, 11:25-31, Fig. 5 (step PR1).

Explanation: The '844 patent explains how the processor 26₁ of the device 26 generates an IPv4 data packet from the IPv6 data packet, e.g., by “de-encapsulating the IPv4 packet contained in the [IPv6] packet.” Ex.1001, 11:25-31.

6. “means for routing the IPv4 data packet to its IPv4 destination”

Function: routing the IPv4 data packet to its IPv4 destination.

Structure: a telecommunication means 26₄ of the device 26 within the home gateway 20 or access node 40, and equivalents. Ex.1001, 12:58-65, 11:25-41, Fig. 5 (step PR3).

Explanation: The '844 patent explains that the processor 26₁ of the device 26 executes computer program instructions stored in the memory 26₅ for routing the IPv4 data packet toward the destination IPv4 domain using a telecommunication means 26₄ of the device 26. Ex.1001, 11:25-41, 12:58-13:10.

7. “means for advertising to the IPv4 domain IPv4 addresses of network equipments connected to the IPv6 domain”

Function: advertising, to devices in the IPv4 domain, the IPv4 addresses of network equipment connected to the IPv6 domain

Structure: the processor 26₁ of the device 26 within home gateway 20 or access node 40 executing computer program instructions stored in the memory 26₅ of the device 26, including instructions for advertising, to devices in the IPv4 domain, the IPv4 addresses of network equipment connected to the IPv6 domain. Ex.1001, 13:61-14:26.

Explanation: The '844 patent describes advertising routes via IGP and BGP. Ex.1001, 14:1-4, 14:24-25. The '844 patent further explains that advertisements from a device can “indicate to the other equipments ... that the data packets comprising the advertised addresses must be sent to it.” Ex.1001, 4:48-62.

8. “means for advertising in the IPv6 domain IPv6 addresses constructed from IPv4 addresses of the IPv4 domain”

Function: advertising, to devices in the IPv6 domain, the IPv6 addresses constructed from IPv4 addresses of network equipment of the IPv4 domain

Structure: the processor 26₁ of the device 26 within home gateway 20 or access node 40 executing computer program instructions stored in the memory 26₅ of the device 26, including instructions for advertising, to devices in the IPv6 domain, the IPv6 addresses of network equipment of the IPv4 domain. Ex.1001, 4:31-58, 12:38-13:15, 14:1-26.

Explanation: The '844 patent describes advertising routes via IGP and BGP. Ex.1001, 14:1-4, 14:24-25, 8:39-44; Fig. 1. The '844 patent further explains that advertisements from a device can “indicate to the other equipments ... that the data packets comprising the advertised addresses must be sent to it.” Ex.1001, 4:48-62.

9. “means for obtaining an IPv4 address, a range of port numbers, and a range of authorized IPv6 addresses”

Function: obtaining an IPv4 address, a range of port numbers, and a range of authorized IPv6 addresses.

Structure: the home gateway 20 or access router 30 serving home gateway 20. Ex.1001, 5:4-9, 8:39-66.

Explanation: The specification states that home gateway 20 obtains the recited information. Ex.1001, 5:4-9. “[A]ccess router 30 obtains the IPv6 prefixes... in a manner that is entirely conventional.” Ex.1001, 8:45-57. The '844 patent further explains that a constructed IPv6 address prefix “combines the bits of the IPv4 public address... and the range of authorized port numbers.” Ex.1001,

8:58-66; *see also* Ex.1001, 13:57-67.

IX. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims in view of the analysis below.

X. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Challenged Claims and Statutory Grounds for Challenge

Ground	Claims	Basis
#1	1-6, 8, 10	35 U.S.C. § 103 over Ananda, Wetterwald, and RFC4380
#2	7	35 U.S.C. § 103 over Ananda, Wetterwald, RFC4380, and RFC4271
#3	9	35 U.S.C. § 103 over Ananda, Wetterwald, RFC4380, and RFC1918

U.S. Publication 2004/0107287 to Ananda (“**Ananda**,” Ex.1005) published on June 3, 2004 and is prior art under 35 U.S.C. §§ 102(a), (b), and (e).

U.S. Patent 7,639,686 to Wetterwald (“**Wetterwald**,” Ex.1008) was filed on April 7, 2005 and issued on December 29, 2009. Wetterwald is prior art under 35 U.S.C. § 102(e). Wetterwald was also the subject of a pre-grant publication, U.S. Pub. No. 2006/0227792 on October 12, 2006, and is therefore prior art under 35 U.S.C. § 102(a), (b). *See* Ex.1025.

“Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)” by C. Huitema was published by the Internet Engineering Task Force as Request for Comments 4380 (“**RFC4380**,” Ex.1009) in February 2006. IETF publications, including RFC4380, were publicly available and well-known documents to those of skill in the art. Ex.1043, ¶¶ 45-48 (public accessibility of RFC4380) (citing Exs.1049-1050); Ex.1003, ¶ 24; Ex.1042, 6, 8, 26; *see* Ex.1043, ¶¶ 85-88 (public accessibility of RFC2026, citing Exs.1066-1067); *see also* U.S. Patent No. 7,764,691 (Ex.1049) (filed on March 15, 2007, uses “Teredo” in the title of the invention, references RFC4380 and Teredo at length throughout the specification (e.g., 2:38-3:2, 4:2-15, 4:40-46, Abstract), and provides an online address for Teredo). RFC4380 is prior art under 35 U.S.C. § 102(a) and (b).

“A Border Gateway Protocol 4 (BGP-4)” by Yakov Rehkter, *et al.*, was published the IETF as Request for Comments 4271 (“**RFC4271**,” Ex.1046) in February 1996. IETF publications, including RFC4271, were publicly available and well-known documents to those of skill in the art. Ex.1043, ¶ 93-96 (citing Exs.1070-1071); Ex.1003, ¶ 24; Ex.1042, 6, 8, 26. RFC4271 is prior art under 35 U.S.C. § 102(a) and (b).

“Address Allocation for Private Internets” by Yakov Rehkter was published by the IETF as Request for Comments 1918 (“**RFC1918**,” Ex.1007) in February 1996. IETF publications, including RFC1918, were publicly available and well-

known documents to those of skill in the art. Ex.1043, ¶ 41-44 (citing Exs.1047-1048); Ex.1003, ¶ 24; Ex.1042, 6, 8, 26. RFC1918 is prior art under 35 U.S.C. § 102(a) and (b).

Dr. Min and this Petition cite to additional documents published by the IETF. To the extent the Board determines that these IETF documents must qualify as prior art “printed publications,” the documents do so qualify. Ex.1003, ¶ 24 (citing Exs.1041-1042). The ’844 patent itself acknowledges that IETF documents were accessible, available, and known to the public. *See, e.g.*, Ex.1001, 6:41-53, 6:62-64, 8:55-57 (citing RFC2462, RFC3315, and RFC3633), 2:1-6 (discussing IETF reports). The Board has repeatedly found that IETF documents, including RFCs, are “printed publications.” *See, e.g.*, IPR2017-00337, Paper 31 at 46-47 (Requests for Comments are “precisely the type of documents whose main purpose is for public disclosure”); IPR2018-00130, Paper 11 at 30-33 (finding multiple RFCs to be printed publications).

B. Discretionary denial under 35 U.S.C. § 325(d) is not appropriate

Ananda (Ex.1005), Wetterwald (Ex.1008), and RFC4380 (Ex.1009) were not considered or cited during prosecution of the ’844 patent. There is no basis for denying this petition “because, the same or substantially the same prior art or arguments previously were presented to the Office.” 35 U.S.C. § 325(d).

The Examiner cited a document related to Wetterwald (Ex.1010, “’553

publication”) but did not discuss it or use it in any rejection. Ex.1002, 41.

Petitioner’s analysis below references Wetterwald’s disclosure of IPv4 address translation (e.g., Ex.1008, 6:38-47) and supporting structures including translation resource 53 and NAT table 54 (e.g., Ex.1008, 6:38-7:36). There is no indication that the Examiner considered any corresponding disclosure of ’533 publication. Thus, to the extent Wetterwald’s disclosure was before the Examiner, the Examiner overlooked its relevance.

C. Discretionary denial under the *Fintiv* factors is not appropriate

The Board balances the six factors in considering denial under § 314. These factors strongly favor institution. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

1. Whether evidence exists that a stay may be granted if a proceeding is instituted

If an IPR is instituted, a stay would be appropriate under the legal standard in the Eastern District of Texas. *See Cypress Lake Software, Inc. v. Samsung Elecs. Am., Inc.*, No. 6:18-cv-00030 (E.D. Tex. Aug. 28, 2019) (order granting stay). Patent Owner waited for years after the ’844 patent issued on May 28, 2013, before filing its litigation Complaint. Ex.1001; Ex.1020. Further, Petitioner diligently reviewed prior art and prepared this Petition after Patent Owner identified asserted claims on May 4, 2020 and proposed claim constructions on September 4, 2020.

Exs.1019-1020, 2; Ex.1019. A stay pending IPR would be appropriate.

However, given that a motion to stay has not yet been filed, the Board should not infer the outcome of that motion. *Intel Corp. v. VLSI Tech. LLC*, IPR2020-00158, Paper 16, at 7 (PTAB May 20, 2020); *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 15, at 12 (PTAB May 13, 2020). Thus, this factor is neutral on discretionary denial.

2. Proximity of the Court's trial date to the Board's projected statutory deadline for a final written decision

The district court's schedule currently calls for a claim construction hearing to be held on November 20, 2020 and for jury selection for trial to begin May 3, 2021. *See* Ex.1040. A Final Written Decision in this proceeding is likely to issue in or around April 2022. While jury selection is currently scheduled before the expected Final Written Decision, the scheduled May 3, 2021 trial date is uncertain, as the district court has eight other cases scheduled for trial that same day. *See* Ex.1026, 1 (trial date scheduled for May 3, 2021); Ex.1027, 1 (same); Ex.1028, 1 (same); Ex.1029, 1 (same); Ex.1030, 1 (same); Ex.1031, 1 (same); Ex.1032, 1 (same); Ex.1033, 1 (same). It is thus a fair possibility that the scheduled May 3, 2021 trial date involving Petitioner and Patent Owner will be moved back closer to, or even after, the expected Final Written Decision date. Because of the uncertainty of the current trial date, this factor weighs against discretionary denial.

See Sand Revolution II, LLC v. Continental Intermodal Grp.—Trucking LLC, IPR2020-01393, Paper 24 (June 16, 2020) (panel granting institution on rehearing based on finding that uncertainty over district court’s trial date weighed against discretionary denial).

3. Investment in the parallel proceeding by the Court and the parties

The litigation is in its early stages, and the investment in it has been minimal. As mentioned above, a claim construction hearing is currently scheduled for November 20, 2020. *See* Ex.1040, 3. The early stage of the litigation weighs against discretionary denial.

4. Overlap between issues raised in the petition and in the parallel proceeding

The prior art addressed in the Petition is also part of Petitioner’s invalidity contentions in the litigation. Instituting a proceeding will allow the Board to address the art, and the issues will be narrowed in the litigation due to the estoppel provisions of 35 U.S.C. § 315(e)(2). To reduce overlap further, if the Board institutes trial, Petitioner will cease asserting in the district court litigation the combination of references on which trial is instituted for the claims on which trial is instituted, while trial is instituted. The most efficient course is for the Board to address the issues before it, and the district court to address only the issues that would remain. Accordingly, this factor weighs against discretionary denial.

5. Whether the petitioner and the defendant in the parallel proceeding are the same party

Petitioner is a defendant in the litigation. Ex.1040. That is true of most Petitioners in IPR proceedings. While the *Fintiv* case indicates that a difference between the district court defendant and the petitioner may weigh against discretionary denial, nothing within the *Fintiv* case suggests that the same party between the proceedings weighs in favor of discretionary denial. Accordingly, this factor should not be a basis for denying institution.

6. Other circumstances that impact the Board's exercise of discretion, including the merits

As discussed in detail below, the prior art presented in this Petition renders the Challenged Claims unpatentable as obvious. The merits of Petitioner's arguments are strong, and this factor weighs against discretionary denial.

As such, because the *Fintiv* factors are either neutral or weigh against discretionary denial, and because this Petition was filed more than three months before the statutory bar date, institution should not be denied on discretionary factors.

D. Discretionary denial under *General Plastic* is not appropriate

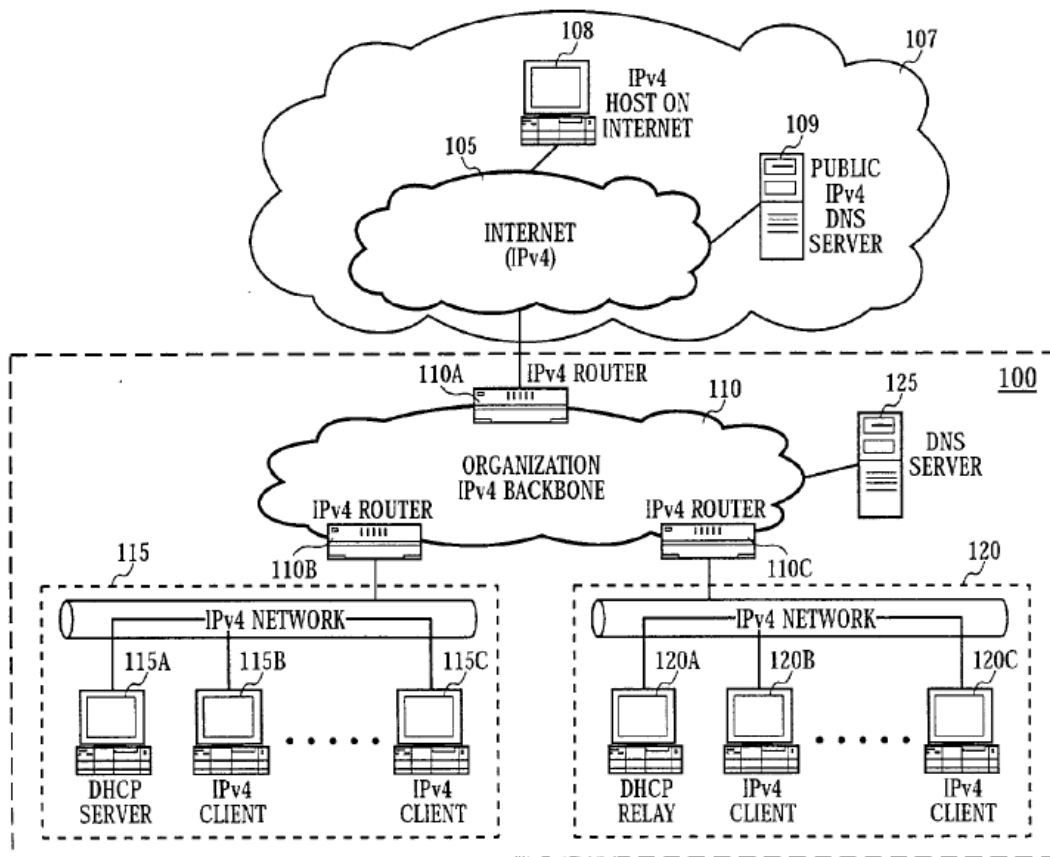
The '844 patent has not been challenged in any prior IPR petition, so none of *General Plastic* discretionary institution factors apply to this Petition. See *General Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at

16 (PTAB Sept. 6, 2016) (Section II.B.4.i. precedential).

E. Ground 1: Claims 1-6, 8 and 10 are obvious over Ananda, Wetterwald, and RFC4380

1. Ananda

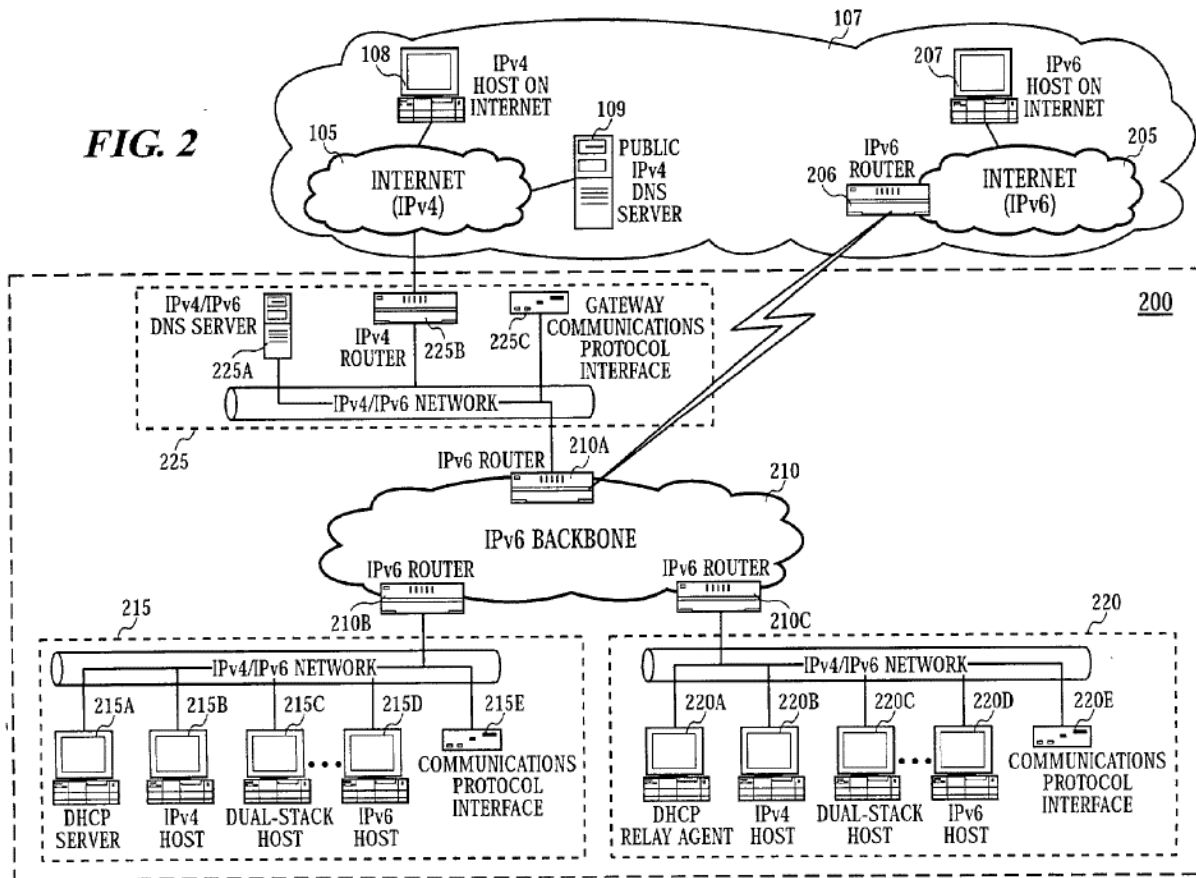
Like the '844 patent, Ananda describes enabling IPv4 devices to communicate across IPv6 networks. Ex.1005, ¶¶ 6-9, 17. With reference to Figure 1 below, Ananda describes transitioning portions of a conventional IPv4 network 100 (depicted in Figure 1 below) to use IPv6. Before the transition, network 100 includes a private IPv4 backbone network 110 and two subnetworks (“subnets”) 115, 120. Ex.1005, ¶¶ 65, 67.



Ex.1005, Fig. 1.

Ananda discloses that the network 100 is a private IPv4 network, and thus the IPv4 subnets 115, 120 are also private networks. Ex.1005, ¶ 67.

Ananda discloses transitioning IPv4 network 100 to an IPv6 network by replacing the IPv4 routers 110A-110C in Figure 1 with IPv6 routers 210A-210C and upgrading the IPv4 backbone 110 to an IPv6 backbone 210, shown in Figure 2. Ex.1005, ¶ 73, 75-77, Fig. 2.



Ex.1005, Fig. 2.

Subnet 215 (né 115) becomes a mixed IPv4/IPv6 network by using a communications protocol interface (CPI) 215E that can communicate according to both the IPv4 and IPv6 protocols. Ex.1005, ¶ 78. For IPv4-only devices on subnet 215, the CPI 215E provides packet translation to IPv6 so that the IPv4-only devices can continue to communicate with devices reachable over the IPv6 backbone. Ex.1005, ¶ 78, 82. The IPv4 hosts 215B, 220B may continue to have private IPv4 addresses. Ex.1005, ¶¶ 83-84.

Like the '844 patent, Ananda discloses facilitating communications between

the IPv4 hosts by encapsulating an IPv4 packet in an IPv6 data packet carrying an IPv6 destination address, as discussed further in the claim analysis below. Ex.1005, ¶¶ 103, 105, 137-138.

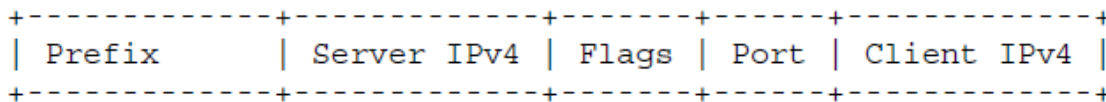
2. Wetterwald

Like the '844 patent and Ananda, Wetterwald facilitates communications between IPv4 nodes and IPv6 nodes across an IPv6 network. Ex.1008, 1:8-14. Wetterwald describes an address assignment scheme that allows an enterprise to assign private IPv4 addresses to IPv4 hosts connected to a private IPv4 network, thus limiting the number of globally-unique IPv4 addresses required for hosts connecting to a public IPv4 network, e.g., the Internet. Ex.1008, 4:56-5:4. In particular, Wetterwald explains that concerns arise in the event that an IPv6 node is coupled to a private IPv4 network having a Network Address Translator (NAT). NATs perform a Layer-3 translation of IP-Addresses, so that public Internet addresses map to private IP addresses, as described in detail by RFC1918 (Ex.1007). This mapping has allowed enterprises to map a large number of private addresses to a limited number of public addresses, thus limiting the number of public addresses required by Internet users. Ex.1008, 1:36-44; Ex.1003, ¶ 84.

3. RFC4380

Like the '844 patent, Ananda and Wetterwald, RFC4380 facilitates communications between IPv4 nodes and IPv6 nodes across an IPv6 network.

Ex.1009, 1. RFC4380 discloses a Teredo relay for constructing an IPv6 packet using an IPv6 prefix together with the IPv4 address of a Teredo server, a port number (referred to as a UDP port), and an IPv4 address of the original IPv4 packet, as shown in the diagram below. Ex.1009, 11 (§ 4).



Ex.1009, 11 (§ 4).

4. Reasons to combine Ananda, Wetterwald, and RFC4380

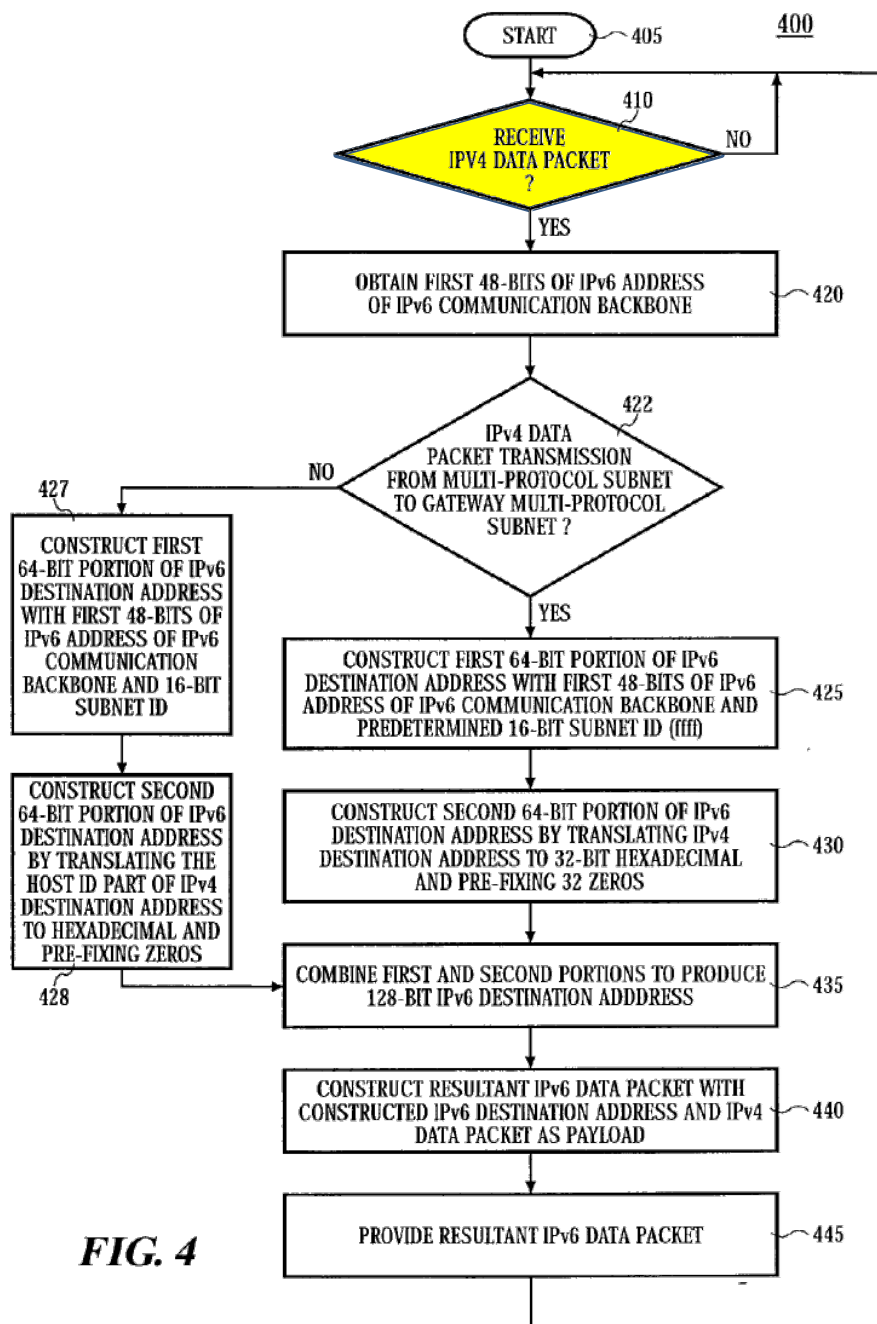
An explanation of the multiple reasons to combine the teachings of Ananda, Wetterwald, and RFC4380 are provided in the analysis below.

5. Claim 1

[1.0.1] A method of receiving a data packet from an IPv4 domain in an IPv6 domain,

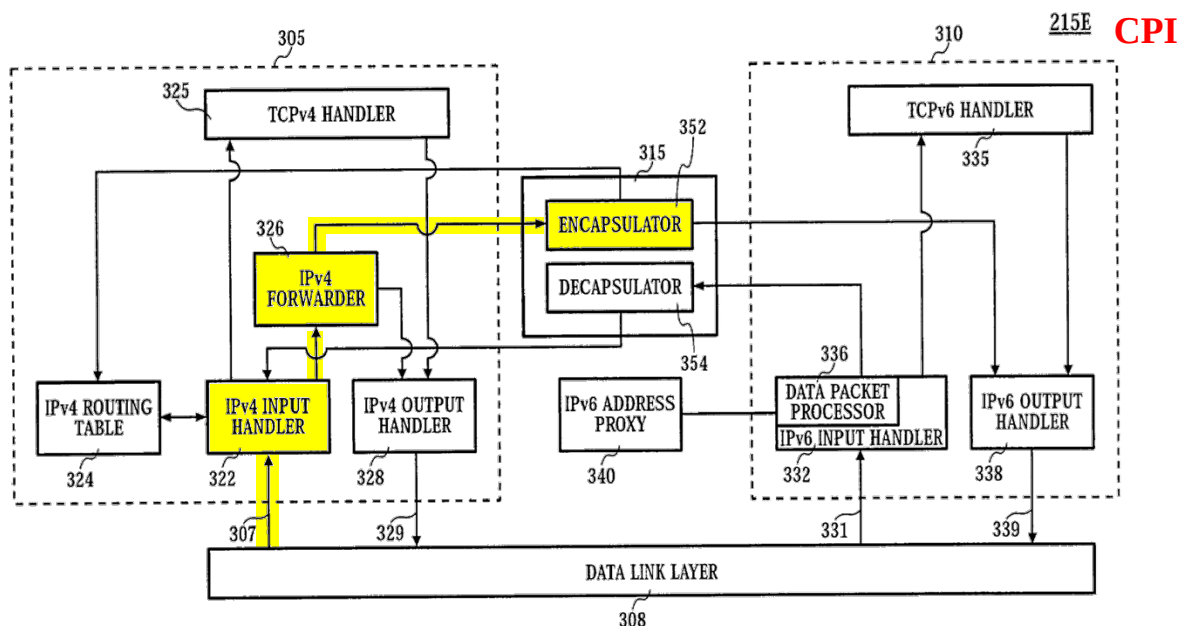
Ananda discloses “a method in a communication protocol interface (CPI),” that includes “**receiving a data packet.**” Ex.1005, ¶¶ 35, 36. For example, Ananda teaches that CPIs 215E, 220E of a communication network 200 transport IPv4 data packets encapsulated in IPv6 data packets across an IPv6 communication backbone 210 (“*IPv6 domain*”), thereby allowing IPv4 hosts 215B, 220B in multi-protocol subnets 215, 220 to communicate with each other. Ex.1005, Abstract, ¶ 63. With reference to Figure 4, Ananda teaches an encapsulation process 400

performed by an encapsulator 352 of the CPI 215E (see Figure 3), which starts in step 410 when the CPI 215E “receives an IPv4 data packet,” e.g., from the IPv4 host 215B in subnet 215 (“from an IPv4 domain”). Ex.1005, ¶ 103.



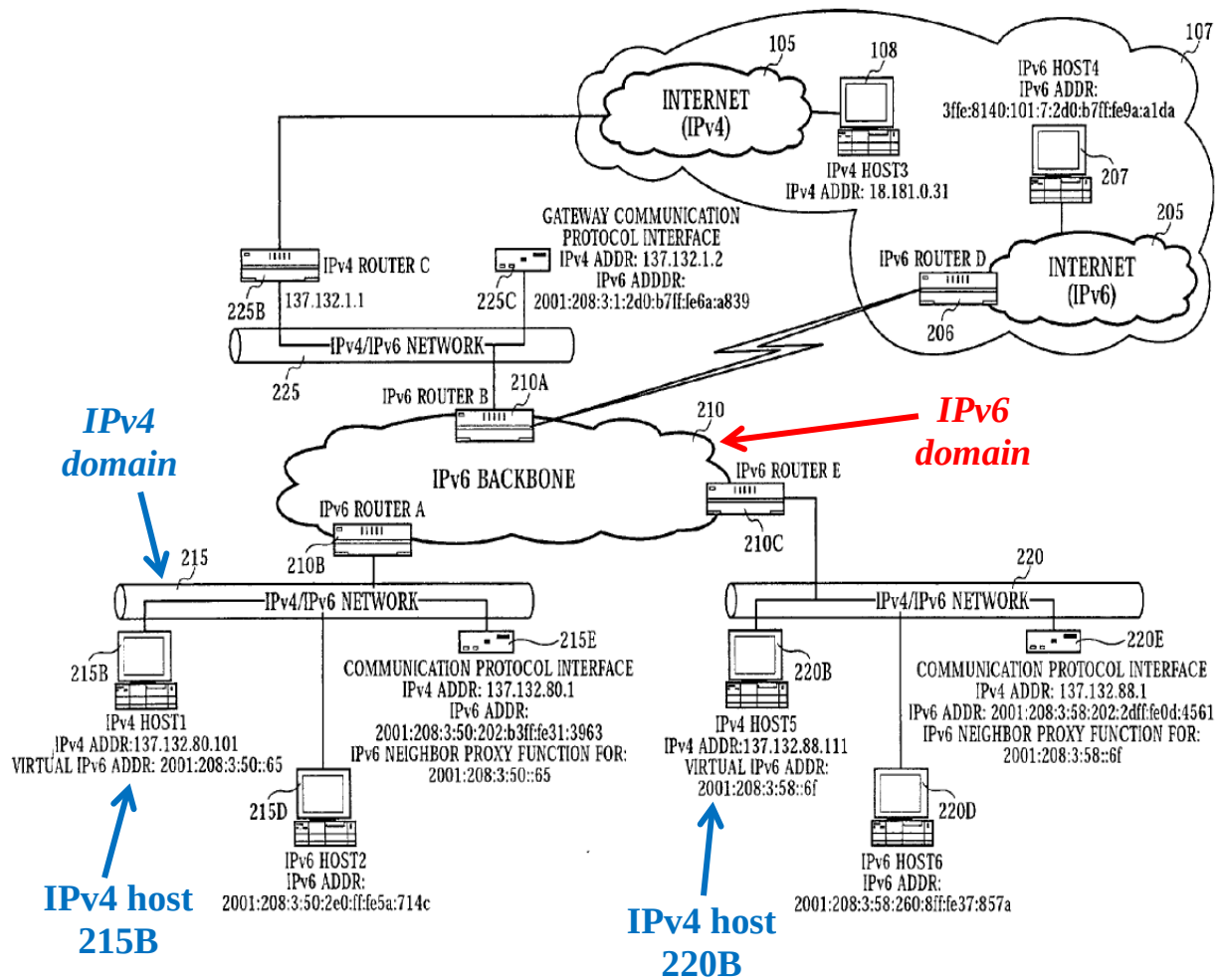
Ex.1003, ¶ 83; Ex.1005, Fig. 4 (annotated).

With reference to annotated Figure 3 below, Ananda describes that the CPI 215E includes an IPv4 input handler 322 “coupled to input 307 for **receiving an IPv4 data packet from** a data link layer 308 of **the multi-protocol subnet 215**” (i.e., *from an IPv4 domain*) and that “encapsulator 352 has an input that is coupled to the IPv4 forwarder 326 to receive an IPv4 data packet” (i.e., the IPv4 packet received by the IPv4 input handler 322). Ex.1005, ¶¶ 90, 91, 99.

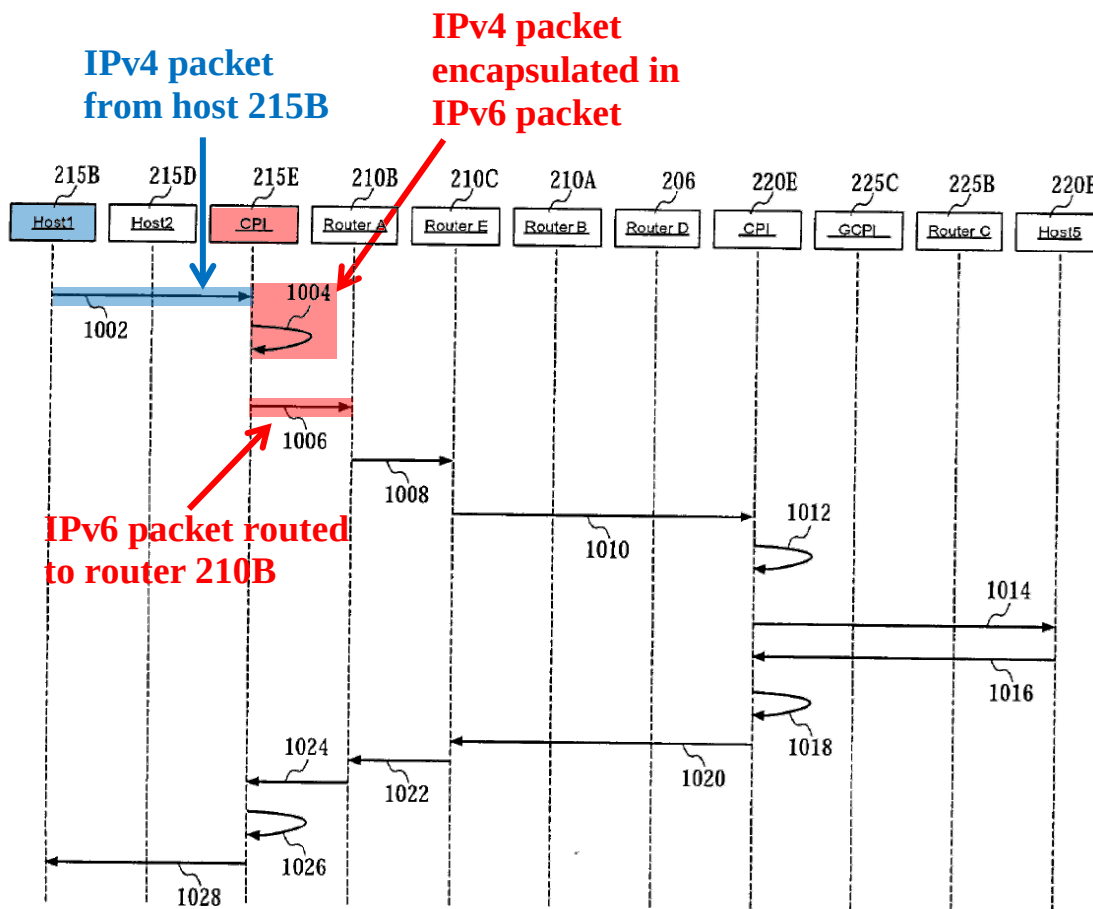


Ex.1003, ¶ 84; Ex.1005, Fig. 3 (annotated).

Further, as shown in annotated Figures 8 and 10 below, Ananda describes the sequence of actions that occur when the CPI 215E receives an IPv4 data packet that IPv4 host 215B in subnet 215 (“*IPv4 domain*”) wants to send to IPv4 host 220 in subnet 220 via IPv6 communication backbone 210 (“*IPv6 domain*”). Ex.1005, ¶¶ 136, 137.



Ex.1003, ¶ 85; Ex.1005, Fig. 8 (annotated).



Ex.1003, ¶ 85; Ex.1005, Fig. 10 (annotated).

As shown in annotated Figure 10 above, Ananda teaches that the encapsulator 352 of the CPI 215E encapsulates the IPv4 data packet received from the IPv4 host 215B in subnet 215 (“*IPv4 domain*”). Ex.1005, ¶¶ 138, 139. The CPI 215E, which is assigned an IPv6 address and communicates with IPv6 router 210B to route IPv6 data packets across the IPv6 communication backbone 210 according to the IPv6 protocol, is “*in an IPv6 domain*” of Ananda’s communication network. Ex.1003, ¶ 86.

While petitioner does not believe that the terms “*IPv4 domain*” and “*IPv6*

domain” require claim construction, it would have been obvious to a POSITA that each multi-protocol subnet, which “advantageously supports transparent operation of both IPv4 and IPv6” in Ananda’s communication network, includes an *IPv4 domain* and an *IPv6 domain* under any reasonable construction of these terms.

Ex.1005, ¶ 64; Ex.1003, ¶ 87. For example, to the extent that “domain” is interpreted to refer to an administrative partition of a network, Ananda states that each subnet “is a portion of a network, which may be a physically independent network segment, which shares a network address with other portions of the network.” Ex.1005, ¶ 66, 76, 78-79, 82 (“a subnet can comprise purely IPv4 hosts”). As each “multi-protocol subnet has a mixture of IPv4, IPv6, and dual-stack hosts i.e. IPv4 and IPv6 hosts,” a POSITA would have understood that the IPv4 portion of multi-protocol subnet 215 (including the IPv4 hosts therein) is an *IPv4 domain* and that the IPv6 portion of multi-protocol subnet 215 (including CPI 215E) is an *IPv6 domain*. See, e.g., Ex.1005, ¶¶ 63, 76, 78-79; Ex.1003, ¶ 87.

If *domain* is interpreted to mean a part of a computer network in which data processing resources are under common control, Ananda further explains how each multi-protocol subnet includes a CPI and an IPv6 router that controls network traffic flowing to and from the individual subnet. Ex.1005, ¶¶ 78-79. The IPv4 and IPv6 devices in subnet 215 are also under the common control of CPI 215E, which enables multi-protocol communication within the subnet 215, and CPI 215E and

IPv6 router 210B exchange IPv6 packets with the IPv6 network 210 on behalf of the devices in the subnet 215 (e.g., IPv4 host 215B and IPv6 host 215D). Ex.1005, ¶¶ 78, 137 (step 1002 in Fig. 10), 149-150 (steps 1026 and 1028 in Fig. 10).

Communications to and from subnet 220 are facilitated separately by IPv6 router 210C and CPE 220E. See Ex.1005, ¶¶ 143-144 (steps 1014 and 1016 in Fig. 10).

Therefore, a POSITA would have understood that the IPv4 hosts in the IPv4 portion of subnet 215, which communicate using the IPv4 protocol, are under common control and that the collection of IPv6 hosts, which communicate using the IPv6 protocol in the IPv6 portion of subnet 215, are under common control. Thus, the IPv4 portion of subnet 215 is an *IPv4 domain* and the IPv6 portion of subnet 215 is an *IPv6 domain*. Ex.1003, ¶ 88.

Finally, as a third example, to the extent that *domain* is interpreted to mean a collection of hosts, routers, and interconnecting network(s) managed by a single administrative authority, Ananda teaches that a subnet “is a portion of a network, which may be a **physically independent network segment**, which shares a network address with other portions of the network.” Ex.1005, ¶ 66. Because a subnet can include purely IPv4 devices (e.g., IPv4 host 215B in subnet 215), which support only IPv4 and are assigned IPv4 addresses (Ex.1005, ¶¶ 82-83), it would have been obvious to a POSITA for the collection of IPv4 devices in an IPv4 portion of each multi-protocol subnet to be physically independent and thus,

managed by an independent administrative authority. Ex.1003, ¶ 89. Likewise, it would have been obvious to a POSITA for IPv6 hosts (e.g., IPv6 host 215D in subnet 215) in an IPv6 portion of each multi-protocol subnet, which “support IPv6 only” (Ex.1005, ¶ 77) and are assigned IPv6 addresses (Ex.1005, ¶ 83), to be physically independent and thus, managed by an independent administrative authority. Ex.1003, ¶ 89. Thus, the collection of IPv4 hosts in the IPv4 portion of subnet 215 is an *IPv4 domain* and the collection of IPv6 hosts in the IPv6 portion of subnet 215 is an *IPv6 domain*. Ex.1003, ¶ 89.

Accordingly, Ananda renders obvious limitation [1.0.1]. Ex.1003, ¶ 90.

[1.0.2] said data packet comprising an IPv4 destination address and a destination port number, said method comprising the following steps:

Ananda teaches that the IPv4 input handler 322 of the CPI 215E “**determines the IPv4 destination address from the received IPv4 data packet,** and accesses the IPv4 routing table 324 to determine where the IPv4 data packet should be routed.” Ex.1005, ¶ 90. Therefore, a POSITA would have understood that the IPv4 data packet received by the CPI 215E in Ananda “*compris[es] an IPv4 destination address.*” Ex.1003, ¶ 92-93; *see also* Ex.1005, Figs. 4A-4B. Annotated Figure 4 below shows the IPv4 destination address from the IPv4 data packet received in step 410 is used in steps 428 and 430 to construct a “second 64-bit portion” of an IPv6 destination address.

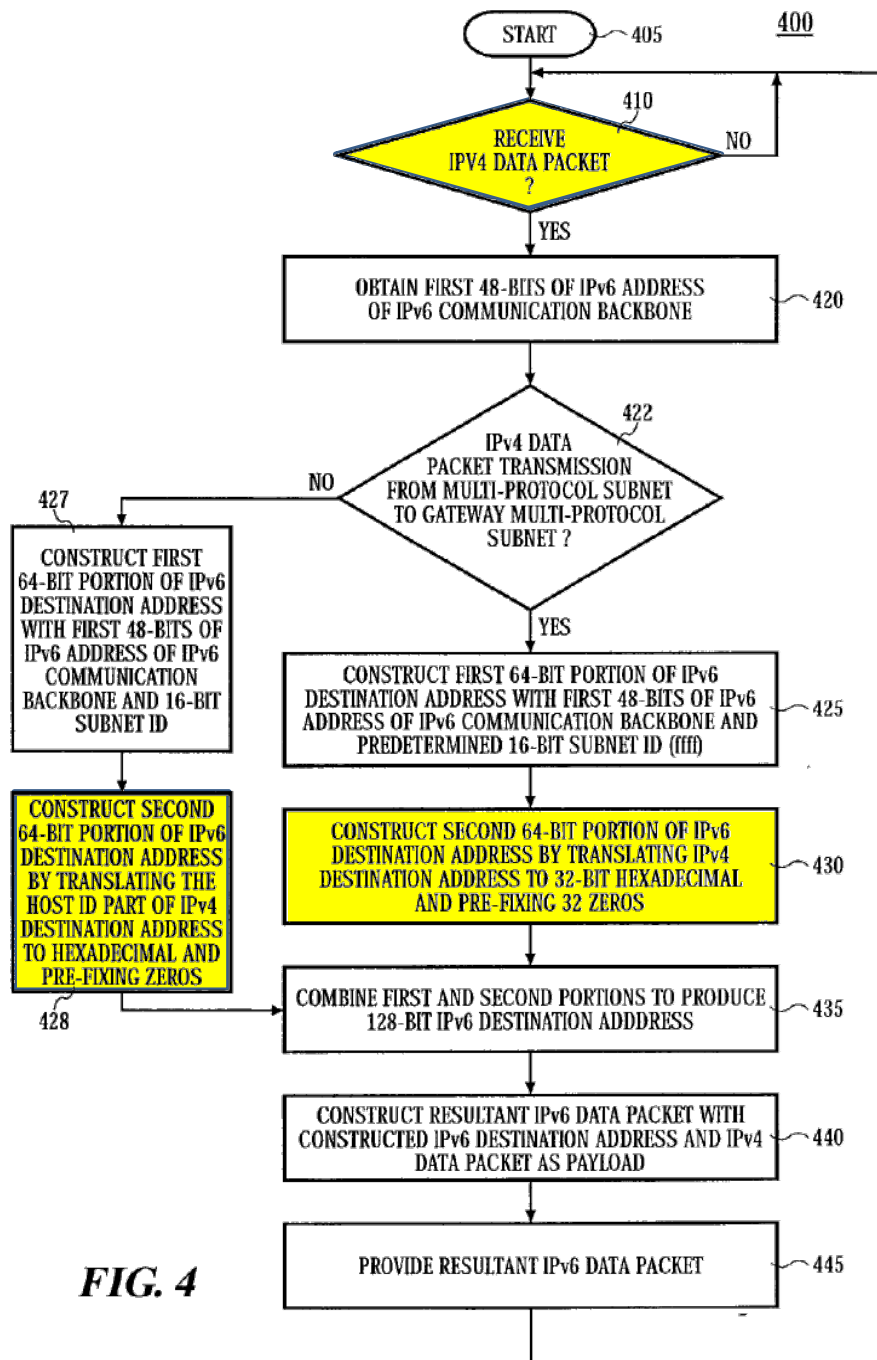


FIG. 4

Ex.1003, ¶ 92; Ex.1005, Fig. 4 (annotated).

Second, Ananda describes IP packets as containing data formatted according to higher-layer protocols such as Transmission Control Protocol (TCP). See

Ex.1005, ¶¶ 90, 95 (describing TCPv4 and TCPv6 handlers). A POSITA would have known that a TCP-formatted packet would include a TCP destination port number. Ex.1003, ¶ 94; Ex.1045, 15; *see also* Ex.1043, ¶¶ 89-92 (citing Exs.1068-1069).

It was well known for IPv4 data packets to include destination addresses and port numbers. Ex.1003, ¶ 95. For example, Wetterwald teaches that a clusterhead 32 in the IPv6 access network 14 includes “an IPv4 interface 40...for sending and receiving IPv4 packets onto the public IPv4 network 16 using public addresses” and “a translator resource 50” for performing “IPv6-to-IPv4 translations and vice versa” and “IPv4 address translations between private and public IPv4 addresses **(and PAT-based port translations for address reuse)**.” Ex.1008, 6:38-47.

Wetterwald further describes that an “IPv4 NAT/PAT translation resource 53” analyzes a packet’s destination by matching “**the destination address 112 and the destination TCP/UDP port (e.g., ‘1’) in the payload 102’ [of the packet 120].**” Ex.1008, 7:7-30. Wetterwald uses the TCP/UDP port number “as a 16-bit reference for identifying the data flow” between source and destination nodes in the network. Ex.1008, 7:7-36; *see also* RFC793 (Ex.1045), § 3.1 at 15 (showing a “TCP Header Format” with a 16-bit “Source Port” field for the “source port number” and a 16-bit “Destination Port” field for the “destination port number”). Ex.1003, ¶ 96.

Because destination TCP and UDP port numbers were commonly included in IPv4 packets, as described in Wetterwald, it would have been obvious to a POSITA for IPv4 packets in Ananda's network to include a TCP or UDP destination port number. Ex.1003, ¶ 97.

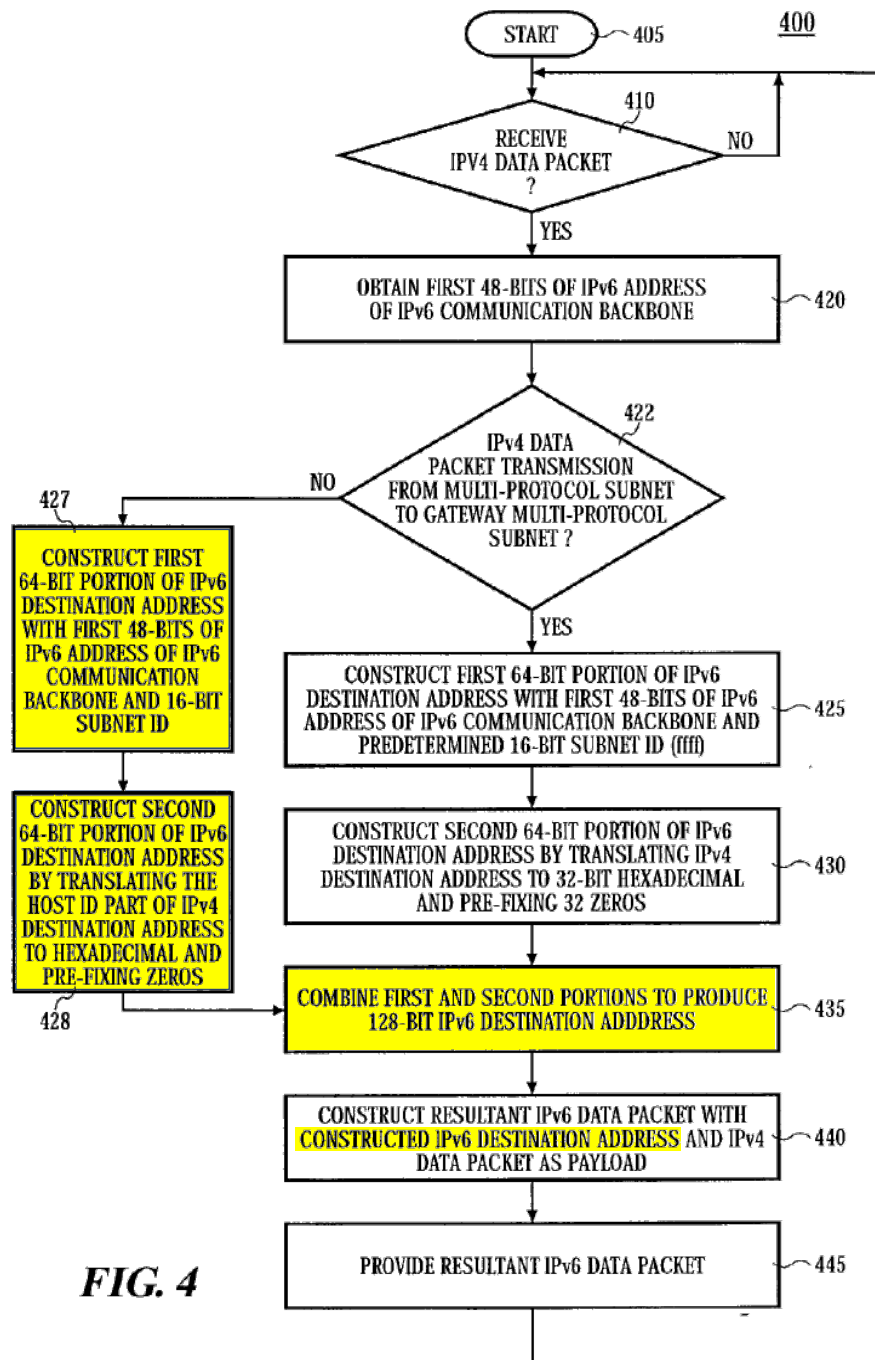
Thus, Ananda and Wetterwald render obvious limitation [1.0.2].

[1.1] constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number;

The prior art renders obvious limitation [1.1]. Ex.1003, ¶¶ 99-111.

First, Ananda describes a “**constructed IPv6 destination address**” in Figure 4. Ex.1005, Fig. 4, step 440. Figure 4 illustrates an encapsulation process 400 that is performed by the encapsulator 352 of CPI 215E (see Figure 3) of subnet 215, when the CPI 215E receives an IPv4 packet from IPv4 host 215B (step 410). Ex.1005, ¶¶ 103, 99 (describing encapsulator 352). Ananda discloses that the CPI 215E supports dynamic encapsulation of IPv4 data packets, resulting in IPv6 data packets, where “the resultant IPv6 or outer encapsulation layer destination address is advantageously derived and **constructed from the destination address of the encapsulated IPv4 data packet**, dynamically.” Ex.1005, ¶ 116; Ex.1003, ¶ 100.

Annotated Figure 4 below shows Ananda's two IPv6 address construction techniques in steps 427, 428, 425, 430, and 435.



Ex.1003, ¶ 101; Ex.1005, Fig. 4 (annotated).

Ananda explains that in step 427 or 425, a first portion of the constructed IPv6 address is formed from a 48-bit IPv6 prefix associated with the IPv6 communication backbone (“*an operator prefix*”).

Ananda explains that in step 428 or 430, a second portion of the constructed IPv6 address is formed from the original IPv4 destination address (“*said IPv4 destination address*”).

Wetterwald further teaches that it was known, when constructing an IPv6 destination address, to include IPv6 extension headers to identify the source and destination port numbers:

If desired, the translator 42 also may **add IPv6 option headers related to the layer 4 TCP/UDP source and destination ports**, as described in detail in RFC2460.

Ex.1008, 8:24-28.

It would have been well-known to a POSITA that TCP/UDP source and destination port numbers are each just 16 bits long. See Ex.1045, § 3.1 at 15 (showing “TCP Header Format” with a 16-bit “Source Port” field for the “source port number” and a 16-bit “Destination Port” field for the “destination port number”). Ex.1003, ¶ 95. Together, both source and destination port numbers are therefore 32 bits long. Since Ananda’s IPv6 address construction technique uses only a portion of the 64 bits available in the second half of an IPv6 address, it would have been obvious to a POSITA that the TCP/UDP source port number,

destination port number, or both, could be advantageously incorporated into the constructed IPv6 address itself. Ex.1003, ¶ 95. For example, Ananda references in steps 428 and 430 that it prefixes “zeros” to fill out the unused and unallocated portions of the IPv6 address. Ex.1005, ¶¶ 104, 107. By including port information in the IPv6 address itself, a POSITA would have recognized that the port information could be made available in the IPv6 header without adding an option header (as referenced by Wetterwald) that would increase the overall IPv6 packet size. Ex.1003, ¶ 105.

It was known in the art to include a port number in a constructed IPv6 address. The Teredo protocol (RFC4380) provided connectivity and integration between IPv4 and IPv6 networks using a “Teredo IPv6 address” that includes “the UDP port and IPv4 address at which a client can be reached.” Ex.1009, 5.

A POSITA would have been motivated to modify Ananda’s IPv6 address construction technique to incorporate a destination port number in the IPv6 address, as taught by Wetterwald and RFC4380, for the same reasons that the Teredo protocol incorporates a port number in an IPv6 address. Ex.1003, ¶ 107. For example, Teredo protocol uses the port number included in the IPv6 address to detect potentially fraudulent (“spoofed”) traffic:

- 3) If the source IPv6 address is a Teredo address, the client compares **the mapped IPv4 address and mapped port in the**

source address with the source IPv4 address and source port of the packet....

6) In the other cases, the packet may be accepted, but **the client should be conscious that the source address may be spoofed;** before processing the packet, the client should perform the "direct IPv6 connectivity test" described in Section 5.2.9.

Ex.1009, 22-23.

By comparing the destination port number identified in an IPv4 packet with the destination port number incorporated into an IPv6 destination address, the recipient of Ananda's encapsulated IPv6 packet (e.g., CPI 220E) could potentially detect spoofed traffic. It was well-known in the art that spoofed packets were potentially dangerous and problematic. Ex.1003, ¶ 108; *see, e.g.*, Ex.1016, ¶ 3 (describing the use of "spoofing" in network attacks); Ex.1017, ¶ 3 ("Attackers hide their location using incorrect or 'spoofed' IP source addresses"); Ex.1018, 1:57-61 ("unauthorized access is associated with 'IP spoofing'"). Thus, the desire to protect Ananda's mixed IPv4 and IPv6 network against spoofed traffic would have motivated a POSITA to incorporate an IPv4 destination port in Ananda's constructed IPv6 destination address. Ex.1003, ¶ 108.

A POSITA would have had an expectation of success in combining the technical features of Ananda, Wetterwald, and RFC4380 because there would have been ample room (e.g., available bits) in Ananda's constructed IPv6 address to include a 16-bit destination port number. Ex.1003, ¶ 109. As noted earlier, Ananda

teaches prefixing zeros to the IPv4 address (which is 32 bits in length) to create a 64-bit “second portion” of the constructed IPv6 address. Hence, Ananda’s constructed IPv6 address includes 32 unallocated bits that Ananda suggests filling with zeros. A destination port number is just 16 bits, so a POSITA would have expected success in including the port number as a substitute for some of the zero bits in Ananda’s constructed IPv6 address. Ex.1003, ¶ 109.

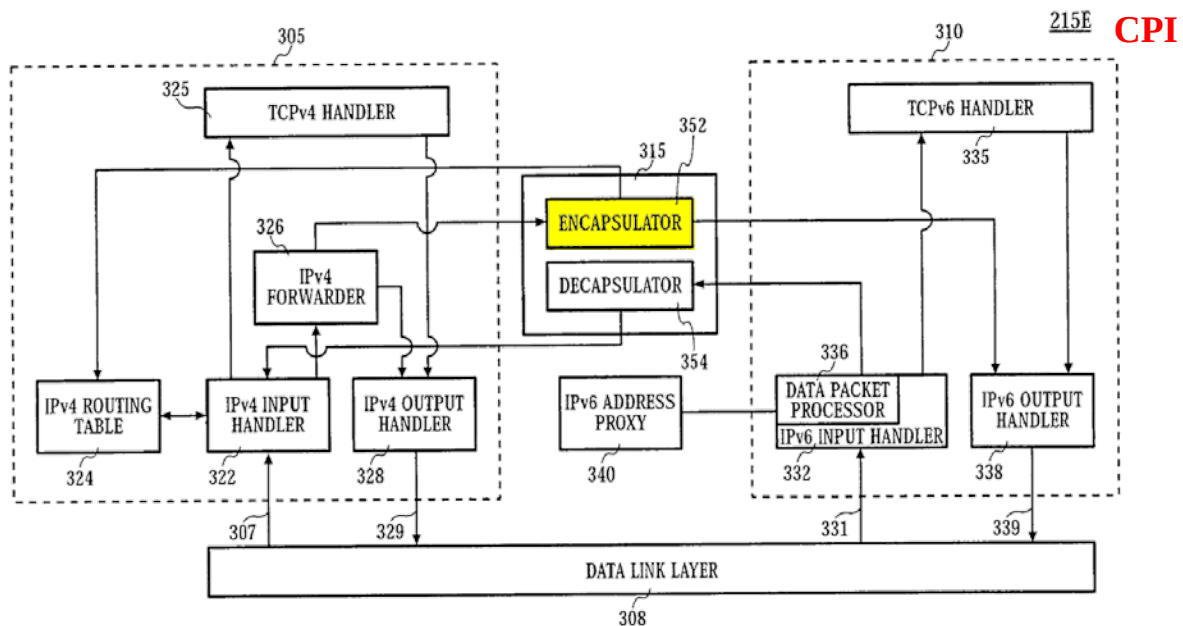
Accordingly, a POSITA would have been motivated to modify Ananda’s technique of constructing an IPv6 destination address by concatenating an IPv6 prefix with an IPv4 destination address, as taught by Wetterwald, and with a destination port number, as taught by Wetterwald and RFC4380. Ex.1003, ¶ 110. This modification would have been nothing more than the combination of prior art elements (constructing an IPv6 destination address with an operator prefix and an IPv4 destination address, per Ananda, and with a destination port number, per Wetterwald and RFC4380) according to known methods (concatenation) to yield the predictable result of constructing an IPv6 destination address with IPv6 extension headers to identify destination port numbers, as taught by Wetterwald and RFC4380. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 401 (2007); Ex.1003, ¶ 110.

Thus, Ananda, Wetterwald, and RFC4380 render obvious limitation [1.1]. Ex.1003, ¶ 111.

[1.2] generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet; and

As discussed above for limitation [1.1], Ananda's Figure 4 illustrates an encapsulation process 400 that is performed by the encapsulator 352 of CPI 215E (see Figure 3), when the CPI 215E receives an IPv4 data packet. Ex.1005, ¶¶ 103, 99 (describing encapsulator 352).

The encapsulator 352 “has an output... to **provide a resultant IPv6 data packet, which comprises the IPv4 data packet encapsulated therein.**” Ex.1005, ¶ 99.

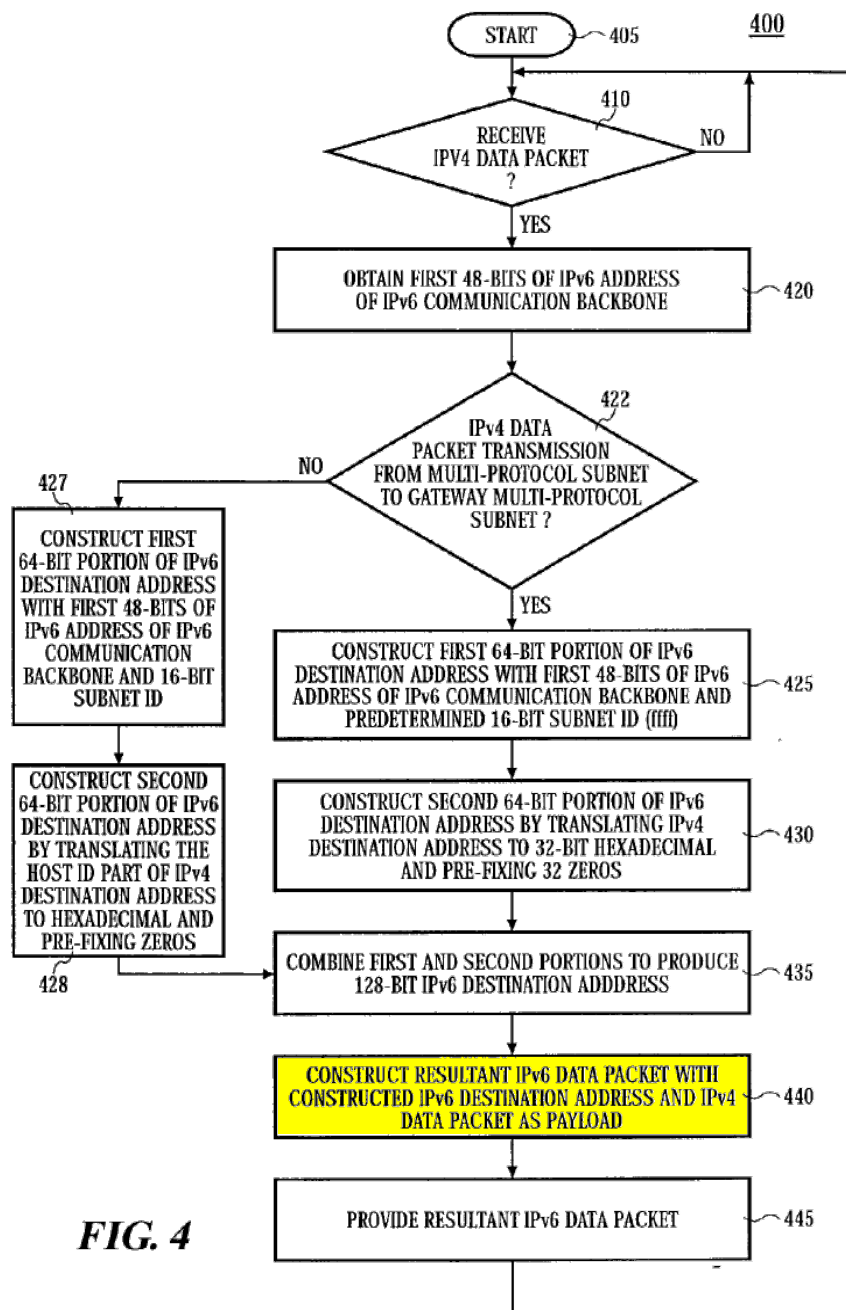


Ex.1003, ¶ 114; Ex.1005, Fig. 3 (annotated).

The output IPv6 packet includes a “resultant IPv6 or outer encapsulation layer destination address [that] is advantageously derived and constructed from the

destination address of the encapsulated IPv4 data packet, dynamically.” Ex.1005, ¶ 116.

Further, as shown in annotated Figure 4 below, Ananda teaches constructing an IPv6 destination address (step 435) and then constructing (“*generating*”) a resultant IPv6 data packet (step 440) “with **the constructed IPv6 destination address and the complete IPv4 data packet** as the payload.” Ex.1005, ¶ 105. Therefore, the resultant IPv6 data packet in Ananda’s encapsulation process is generated *from the IPv6 constructed destination address and the received IPv4 packet*.



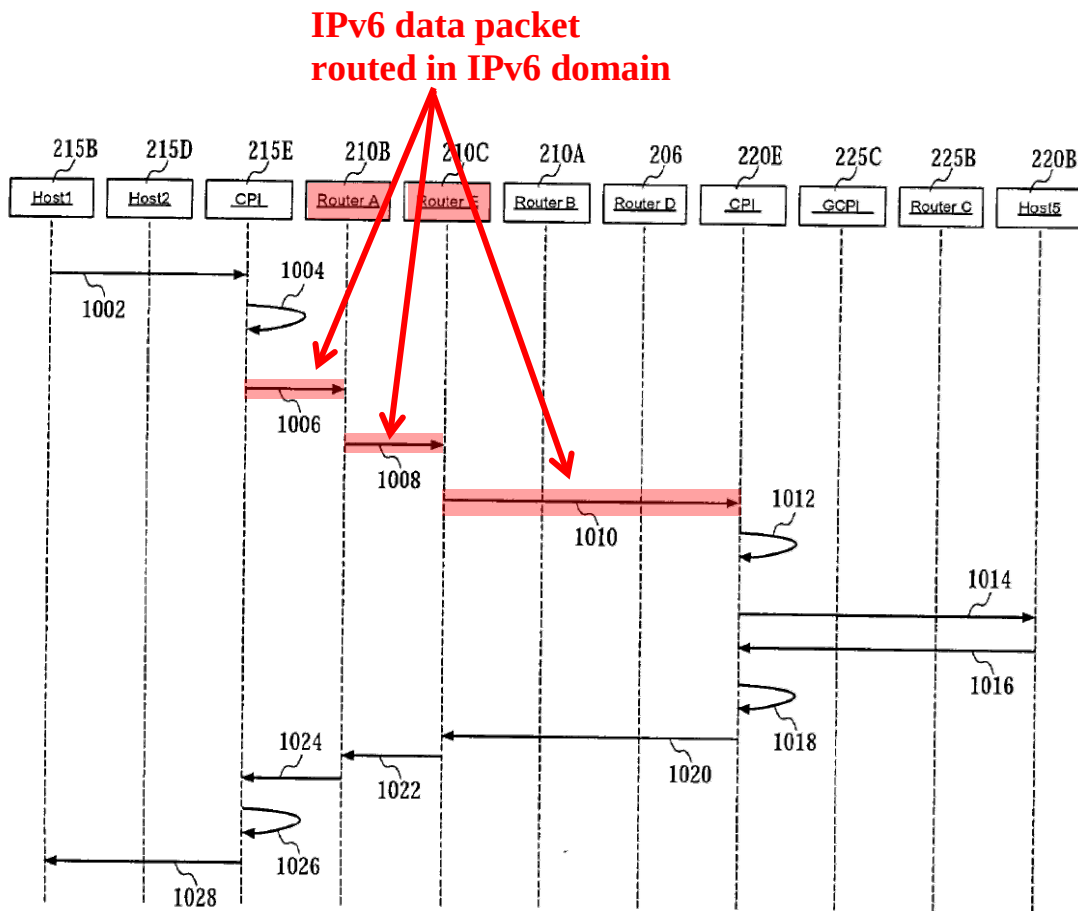
Ex.1003, ¶ 116; Ex.1005, Fig. 4 (annotated).

Thus, Ananda renders obvious limitation [1.2]. Ex.1003, ¶ 113-117.

[1.3.1] routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address,

Continuing from the discussion of limitation [1.2], Ananda teaches that CPI 215E supports “dynamic encapsulation of IPv4 data packets in resultant IPv6 data packets” that are “**routed through IPv6 infrastructure.**” Ex.1005, ¶ 116.

As shown in annotated Figure 10 below, Ananda teaches that after the encapsulator 352 of the CPI 215E “encapsulates the IPv4 data packet and produces a resultant IPv6 data packet... having an IPv6 destination address 2001:208:3:58::6f, which is the IPv6 virtual address of the IPv4 host [220B]” (step 1004), the resultant IPv6 data packet is routed to IPv6 router 210B (step 1006). Ex.1005, ¶¶ 138-139. Next, router 210B routes the IPv6 data packet to router 210C (step 1008) in the IPv6 backbone 210 (“*in the IPv6 domain*”), which in turn routes the IPv6 data packet to the CPI 220E (step 1010). Ex.1005, ¶¶ 140, 141.



Ex.1003, ¶ 120; Ex.1005, Fig. 10 (annotated).

Thus, Ananda renders obvious limitation [1.3.1]. Ex.1003, ¶ 119-121.

[1.3.2] said constructed address belonging to a range of IPv6 addresses routable to an interconnection equipment of the IPv6 domain with the IPv4 destination address.

As discussed above for limitation [1.3.1], Ananda's CPI 215E constructs an IPv6 destination address that the IPv6 backbone will route to CPI 220E. Ex.1005, ¶¶ 105, 116. Ananda contemplates that there may be multiple IPv4 hosts on subnet 220. Ex.1005, ¶¶ 82. It would have been obvious to a POSITA that IPv6 packets

on the IPv6 backbone that are directed to any of the IPv4 hosts on subnet 220 would have constructed IPv6 addresses that are routed to CPI 220E so that CPI 220E could decapsulate those packets and deliver them to their ultimate IPv4 destinations. Ex.1005, ¶¶ 82, 142; Ex.1003, ¶ 123. Thus, the constructed IPv6 destination addresses that cause packets to be routed over the IPv6 backbone to CPI 220E are a “*range of IPv6 addresses routable to an interconnection equipment.*” Ex.1003, ¶ 123. As discussed previously at [1.3.1], CPI 220E includes IPv6 communication connectivity and is therefore “*of the IPv6 domain.*”

Finally, as discussed at [1.1], the prior art renders obvious constructing a destination IPv6 address that includes the destination IPv4 address and a destination port number. Since TCP and UDP port numbers are 16 bits long, there are 2^{16} , or 65536, different port numbers possible for a given destination IPv4 address. Ex.1003, ¶ 124. For the reasons discussed at [1.3.1], packets with an IPv6 destination address including any of these potential port numbers would be routed over the IPv6 backbone to the CPI responsible for providing IPv6-to-IPv4 connectivity for the IPv4 destination (e.g., CPI 220E). Thus, by including the port number in the destination IPv6 address (for the reasons discussed above at [1.1]), the prior art renders obvious that the destination IPv6 address is in “*a range of IPv6 addresses routable... with the IPv4 destination address.*” Ex.1003, ¶ 124.

Thus, the prior art renders obvious limitation [1.3.2]. Ex.1003, ¶ 123-124.

6. Claim 2

[2.0] A non-transitory computer program product that is at least one of downloadable from a communications network and stored on a computer-readable medium and executable by a microprocessor,

As discussed above in association with limitation [1.0.1], Ananda teaches implementing “a method in a communication protocol interface (CPI) for communicating a variety of data packets that comply with a corresponding variety of communication protocols on a communication network.” Ex.1005, ¶ 35. CPIs 215E, 220E are described as including “an IPv4 processor for processing IPv4 specific communication, an IPv6 processor for processing IPv6 specific communication, and a protocol converter for processing communication between the IPv4 and IPv6 processors.” Ex.1005, ¶¶ 19, 63.

Additionally, Ananda discloses that a processor of the CPI uses a *memory* for “storing” information associated with the CPI. Ex.1005, claim 6. The CPI’s memory is a “*computer-readable medium*.” Ex.1003, ¶ 128.

Ananda also teaches a CPI “implemented as software modules within a destination host, a stand-alone unit coupled to a subnet and an IPv6 network, be incorporated in a data-switching unit, or be incorporated in a network interface card (NIC).” Ex.1005, ¶ 118. It would have been obvious to a POSITA for such software modules to be implemented within a host or other device on Ananda’s network as a computer program (or “*computer program product*”), stored on the

device's memory (in the form of a *computer-readable medium*), and executable by a processor of the device. Ex.1003, ¶ 129.

Wetterwald similarly teaches “executable code stored” on “a computer readable medium” or “propagated via a computer readable transmission medium.” Ex.1008, 7:55-62, 6:38-42.

It would have been obvious to a POSITA that the executable code stored in the computer-readable medium taught by Wetterwald is a “*non-transitory computer program product ... executable by a microprocessor.*” Ex.1003, ¶ 131. Furthermore, a POSITA would have found it obvious to implement Ananda's CPI as software modules of a *non-transitory computer program product* stored on a computer-readable medium, as taught by Wetterwald, and executable by the IPv4 or IPv6 processor (*microprocessor*) of the CPI. Ex.1003, 131.

Thus, Ananda and Wetterwald render obvious limitation [2.0].

[2.1] comprising program code instructions for executing the method according to claim 1 of receiving in an IPv6 domain a data packet from an IPv4 domain.

First, as discussed for limitation [2.0], Ananda and Wetterwald render obvious “[a] *non-transitory computer program product that is ... stored on a computer-readable medium and executable by a microprocessor.*” See Ex.1005, ¶¶ 19, 35, 63.

Second, Wetterwald discloses a method for sending and receiving IPv4

packets between IPv4 hosts via an IPv6 network. Wetterwald teaches that the steps of the method “can be implemented... as executable code stored on a computer readable medium.” Ex.1008, 7:55-62, 6:38-42. It would have been obvious to a POSITA that the executable code stored on the computer-readable medium in Wetterwald would include “*program code instructions for executing the method*” of Wetterwald or the similar method of Ananda’s CPI. Ex.1003, ¶ 135.

Thus, Ananda and Wetterwald render obvious limitation [2.1].

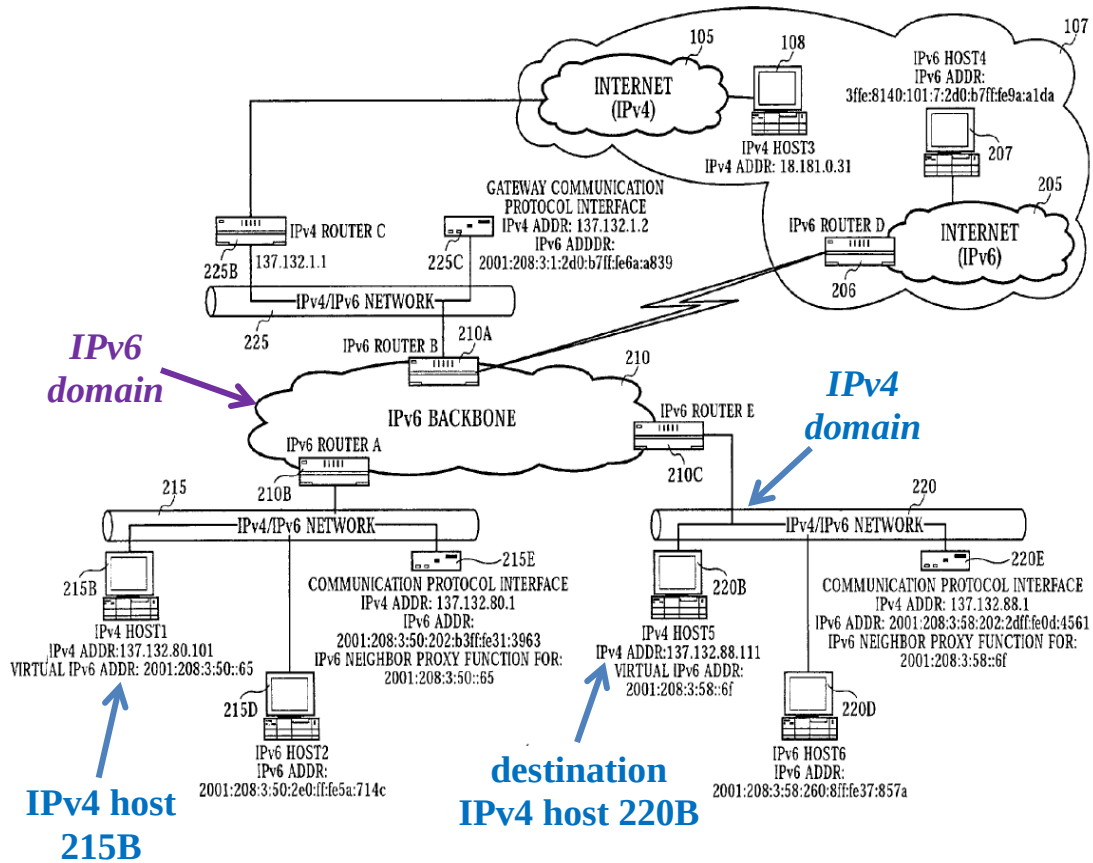
7. Claim 3

[3.0.1] A method of sending a data packet from an IPv6 domain to a destination address of an IPv4 domain,

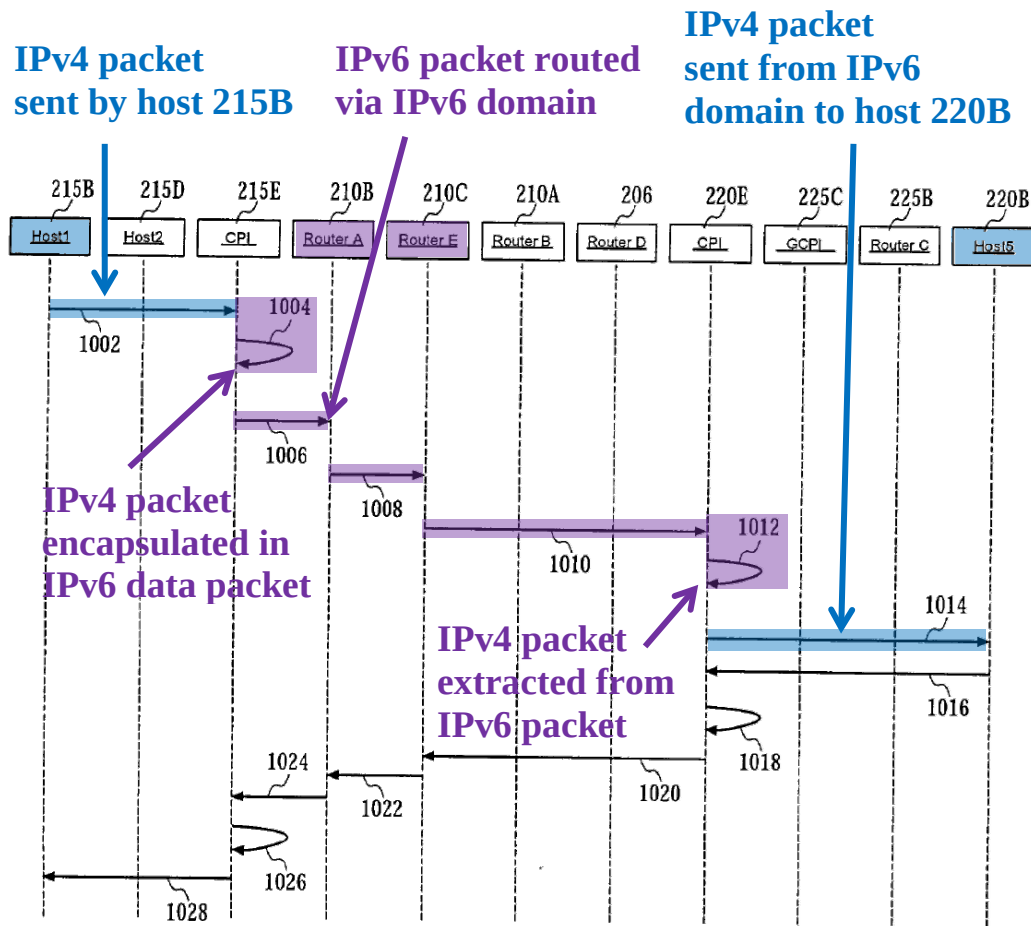
Ananda discloses “a **method** in a communication protocol interface (CPI) for communicating a variety of data packets that comply with a corresponding variety of communication protocols on a communication network.” Ex.1005, ¶ 35. For example, Ananda teaches a communication network 200 with CPIs 215E, 220E for transporting (“*sending*”) IPv4 data packets encapsulated in IPv6 data packets (each “*a data packet*”) across an IPv6 communication backbone 210 (“*an IPv6 domain*”), thereby allowing the IPv4 hosts 215B, 220B in multi-protocol subnets 215, 220 to communicate with each other. Ex.1005, Abstract, ¶ 78.

With reference to Figures 8 and 10, Ananda describes the sequence of

actions that occur when the IPv4 host 215B of subnet 215 wants to send an IPv4 data packet to the IPv4 host 220B of subnet 220. Ex.1005, ¶¶ 136, 137. As shown in annotated Figures 8 and 10 below, the IPv4 data packet sent by the IPv4 host 215B is encapsulated (step 1004) in an IPv6 data packet by the CPI 215E and the resultant IPv6 data packet is sent (steps 1006, 1008, 1010) from the CPI 215E via routers 210B and 210C of the IPv6 communication backbone 210 (*“from an IPv6 domain”*) to the CPI 220E for delivery (steps 1012, 1014) to the destination IPv4 host 220B having an IPv4 address 137.132.88.111 (*“to a destination address ”*) of subnet 220 (*“IPv4 domain”*). Ex.1005, ¶¶ 137-143.



Ex.1003, ¶ 138; Ex.1005, Fig. 8 (annotated).



Ex.1003, ¶ 138; Ex.1005, Fig. 10 (annotated).

Thus, Ananda renders obvious limitation [3.0.1]. Ex.1003, ¶ 137-139.

[3.0.2] said packet comprising an IPv6 destination address of the packet constructed by concatenating an operator prefix, the IPv4 destination address, and a destination port number,

As discussed above at [1.1] and [1.2], the prior art renders obvious generating a IPv6 packet with “an IPv6 destination address [constructed] by concatenating an operator prefix, said IPv4 destination address, and the destination port number.” As discussed at [1.3.1], such IPv6 packets are routed

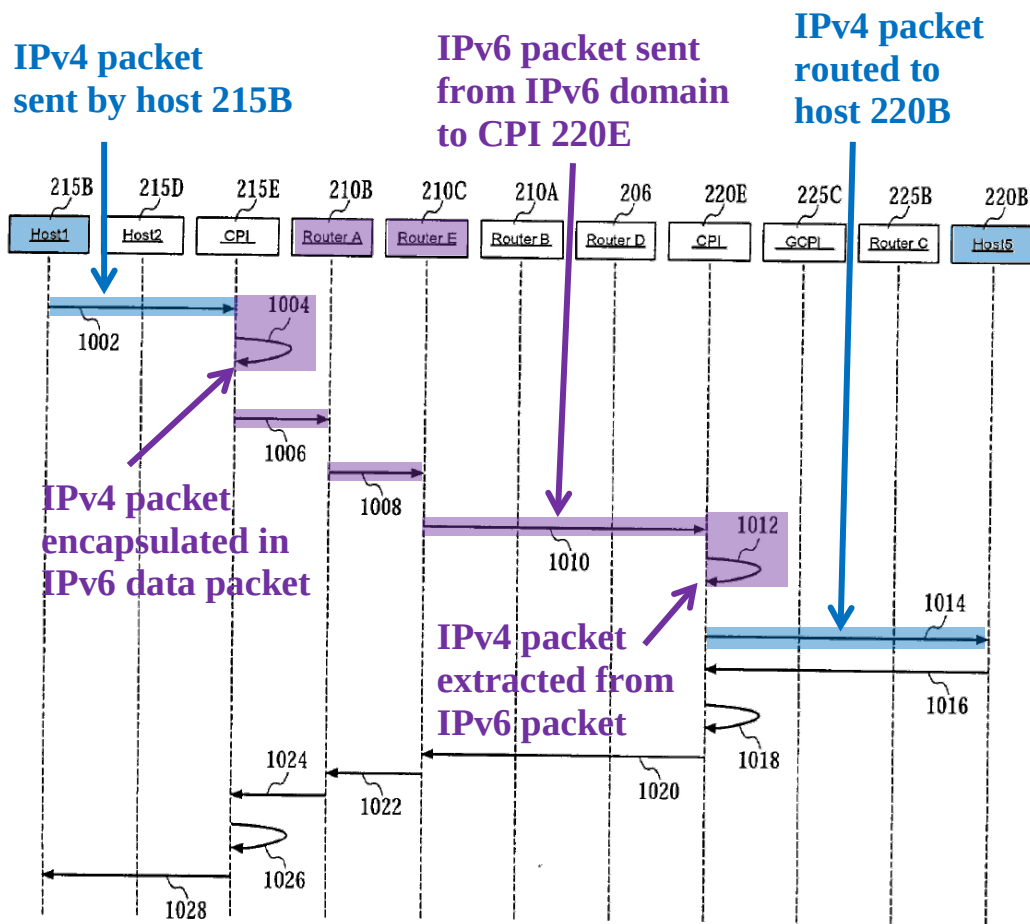
over Ananda's IPv6 backbone to CPI 220E, and therefore, such packets correspond to "said packet" previously identified in [3.0.1].

[3.0.3] said method executing the following steps on reception of the data packet:

Ananda renders obvious limitation [3.0.3]. Ex.1003, ¶ 141-144.

As discussed above at [3.0.1], Ananda teaches that an IPv6 data packet, which encapsulates an IPv4 data packet, is sent from the CPI 215E via routers 210B and 210C of the IPv6 communication backbone 210 to the CPI 220E for delivery to the destination IPv4 host 220B. Ex.1005, ¶¶ 138-143, Fig. 10.

As shown in annotated Figure 10 below, Ananda discloses that upon receiving the IPv6 data packet from router 210C of the IPv6 communication backbone 210 (step 1010), the CPI 220E extracts the IPv4 packet from the IPv6 data packet (step 1012), as will be discussed further at [3.1] and [3.2.1]-[3.2.2] below, and routes the IPv4 data packet to the IPv4 host 220B in subnet 220 (step 1014), as will be discussed further at [3.3] below. Ex.1005, ¶¶ 142-143; Ex.1003, ¶ 143. Accordingly, CPI 220E performs the method steps "*on reception of the data packet.*"



Ex.1003, ¶ 143; Ex.1005, Fig. 10 (annotated).

Thus, Ananda renders obvious limitation [3.0.3].

[3.1] extracting the IPv4 destination address and the destination port number from the IPv6 destination address;

Ananda, Wetterwald, and RFC4380 render obvious limitation [3.1].

First, as discussed in [1.1], it would have been obvious to a POSITA to compare the destination address and port number values within a constructed IPv6 destination address to the corresponding values in the header of an encapsulated IPv4 packet. Ex.1003, ¶ 146. Such a comparison would be useful, for example, in

detecting “spoofed” packets. Ex.1009, 22-23; Ex.1003, ¶ 146. It would have been obvious to a POSITA that to perform such a comparison, the IPv4 destination address and port number should be extracted from the IPv6 destination address. Ex.1003, ¶ 146. For this reason alone, limitation [3.1] is obvious.

Second, in step 1012 of Figure 10, Ananda teaches that when the CPI 220E in subnet 220 receives the IPv6 data packet, the CPI 220E processes the packet using its “decapsulator 354 which **decapsulates the IPv6 header** to produce the original IPv4 data packet.” Ex.1005, ¶ 142. With reference to Figure 5, Ananda teaches a decapsulation process 500 of the decapsulator 354 in which the IPv4 data packet is extracted (step 515), where “[e]xtracting is accomplished by stripping **the IPv6 header** from the IPv6 data packet.” Ex.1005, ¶ 109 & Fig. 5 (step 515).

Further, Ananda teaches that “[t]he decapsulator 354 has an input that is coupled to the IPv6 input handler 332 to receive an IPv6 data packet.” Ex.1005, ¶ 102. As shown in Figure 3, Ananda teaches that the IPv6 input handler 332 “is coupled to input 331 for receiving IPv6 data packets from the data link layer 308.” Ex.1005, ¶ 94. Ananda explains that “[u]pon receiving an IPv6 data packet from the data link layer 308, the IPv6 input handler 332 determines **the destination address of the IPv6 data packet**, to determine how the IPv6 data packet is to be processed.” Ex.1005, ¶ 94.

For example, Ananda teaches that “a data packet processor 336, which is a

part of the IPv6 input handler 332, determines whether the received IPv6 data packet is a resultant IPv6 data packet that encapsulates an IPv4 data packet,” and that “[t]his is accomplished by accessing an IPv6 address proxy 340 to determine whether the destination address of the IPv6 data packet is a virtual IPv6 address of an IPv4 host ... on the multi-protocol subnet [220].” Ex.1005, ¶ 96. As shown in Table 1 below, Ananda teaches that the IPv6 address proxy 340 “stores the virtual IPv6 addresses of the IPv4 host” in association with the IPv4 address of the IPv4 host, e.g., the IPv4 destination address of the destination host 220B, as discussed above at [3.0.1]-[3.0.3]. Ex.1005, ¶¶ 97, 88.

TABLE 1

	IPv4 Address	IPv6 Address	Virtual IPv6 Address
IPv4 host	137.132.80.109/24	NA	2001:208:3:50::6D/64
IPv6 host	NA	2001:208:350:260:8ff:fe39:504f/64	NA
Dual-stack host	137.132.80.110/24	2001:208:3:50.2e0:ff:fe5a:714c/64	2001:208:3:50::6E/64

Ex.1005, ¶ 88 (TABLE 1).

By accessing the stored IPv4 address of the destination IPv4 host 220B based on the destination address of the IPv6 data packet received at Ananda’s CPI 220E, it would have been obvious to a POSITA that the IPv4 destination address is extracted from the IPv6 destination address. Ex.1003, ¶ 147-150.

Third, as discussed above at [3.0.2], Wetterwald teaches that it was known for IPv6 destination addresses to be constructed from the IPv4 destination address and the destination port number specified in the original IPv4 packet, e.g., by including IPv6 extension headers to identify the source and destination port numbers:

If desired, the translator 42 also may **add IPv6 option headers related to the layer 4 TCP/UDP source and destination ports**, as described in detail in RFC 2460.

Ex.1008, 8:24-28, 6:61-7:27 (describing the process of Fig. 4B for translating an IPv4 packet 120 including “the destination address 112 and the destination TCP/UDP port (e.g., ‘1’) in the payload 102” into an IPv6 packet 122 “having a destination address 106”).

As discussed above, RFC4380 teaches that it was known in the art to include a port number in a constructed IPv6 address, for example, to provide connectivity and integration between IPv4 and IPv6 networks. RFC4380 teaches the Teredo protocol, which defines a “Teredo IPv6 address” that includes “the UDP port and IPv4 address at which a client can be reached.” RFC4380 at 5. A POSITA would have been motivated to modify Ananda’s IPv6 address construction technique to incorporate a destination port number in the IPv6 address, as taught by Wetterwald and RFC4380, e.g., for the reasons discussed above at [3.0.2]. Ex.1003, ¶ 152.

Therefore, a POSITA would have found it obvious to modify Ananda's IPv6 address construction technique such that the IPv6 destination address includes both *"the IPv4 destination address and the destination port number,"* as taught by Wetterwald and RFC4380, such that Ananda's decapsulation process includes *"extracting the IPv4 destination address and the destination port number from the IPv6 destination address,"* as recited in limitation [3.1]. Ex.1003, ¶ 145-153.

[3.2.1] generating an IPv4 data packet from the IPv6 data packet,

Ananda renders obvious limitation [3.2.1].

In step 1012 of Figure 10, Ananda teaches that the CPI 220E in subnet 220 processes the IPv6 data packet received in step 1010 by "decapsulat[ing] the IPv6 header to **produce the original IPv4 data packet.**" Ex.1005, ¶¶ 142-143. By decapsulating the received IPv6 data packet to produce the original IPv4 data packet, CPI 220E generates an IPv4 data packet. Ex.1003, ¶¶ 177-179; *see also* Ex.1005, ¶ 109 & Fig. 5 (step 515).

Therefore, Ananda renders obvious limitation [3.2.1]. Ex.1003, ¶ 155-157.

[3.2.2] said packet including said IPv4 destination address and said destination port number; and

As discussed above at [3.0.2], Ananda, Wetterwald, and RFC4380 render obvious that *"said packet compris[es] an IPv6 destination address of the packet constructed by concatenating an operator prefix, the IPv4 destination address, and*

a destination port number.”

Accordingly, Ananda, Wetterwald, and RFC4380 also render obvious that “*said packet includ[es] said IPv4 destination address and said destination port number,*” as recited in limitation [3.2.2]. Ex.1003, ¶ 158-159.

[3.3] routing the IPv4 data packet to its IPv4 destination.

Ananda renders obvious limitation [3.3].

Ananda’s step 1014 in Figure 10 teaches limitation [3.3]. Ex.1005, ¶ 143. Ananda discloses that after CPI 220E produces (extracts) the original IPv4 data packet, the **“IPv4 data packet is then routed to the destination IPv4 host5-220B.”** Ex.1005, ¶¶ 142-143; Ex.1003, ¶¶ 160-162.

8. Claim 4

[4.0] A non-transitory computer program product that is at least one of downloadable from a communications network and/or stored on a computer-readable medium and/or executable by a microprocessor,

See analysis of [2.0], which recites an identical limitation, except that limitation [2.0] recites “and” in place of “and/or.”

[4.1] comprising program code instructions for executing the method according to claim 3 of sending a data packet from an IPv6 domain to a destination address of an IPv4 domain.

As discussed above in [2.1], the prior art renders obvious “*program code instructions for executing the method.*” As discussed above in [3.0.1]-[3.3], the prior art renders obvious the method of claim 3. For the reasons discussed at [2.0]-

[2.1], it would have been obvious for Ananda's CPI (e.g., CPI 220E) to include instructions that would cause its processors to perform the steps of claim 3.

Ex.1003, ¶ 164.

9. Claim 5

[5.0.1] A device for receiving a data packet from an IPv4 domain in an IPv6 domain,

Limitation [5.0.1] is identical to limitation [1.0.1] above, except that it recites a “device” instead of a “method.” As discussed above at [1.0.1], Ananda discloses a *device* in the form of a “communication protocol interface (CPI) for communicating a variety of data packets.” Ex.1005, ¶¶ 35, 18 (“**apparatus** for communicating”).

Ananda's CPI includes “a first communication protocol processor ... for receiving a data packet that complies with a first of the variety of communication protocols” and “a second communication protocol processor ... for receiving another data packet that complies with a second of the variety of communication protocols.” Ex.1005, ¶¶ 19-26; *see also id.*, ¶ 89, Fig. 3.

Therefore, Ananda renders obvious the *device* of limitation [5.0.1]. Ex.1003, ¶¶ 165-167.

[5.0.2] said data packet comprising an IPv4 destination address and an IPv4 destination port number, said device comprising:

See analysis of [1.0.2] above, which is identical except for being directed to

a “method” instead of a “device.” As discussed above at [1.1]-[1.3.2] and [2.0]-[2.1], the prior art renders obvious a computer processor executing instructions to perform the method steps. Ex.1003, ¶¶ 168-169.

[5.1] means for constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number;

Limitation [5.1] is identical to limitation [1.1] except for the addition of “means for.” For the reasons at [1.1] above, the prior art renders obvious the function of the “means for constructing” in [5.1]. Ex.1003, ¶ 170.

The prior art also renders obvious the corresponding structure, i.e., a processor of a device and a NAT table stored in the device’s memory. See Section VIII.B.1, *supra*. Ananda discloses that the CPI 215E includes “an IPv4 processor..., an IPv6 processor..., and a protocol converter... between the IPv4 and IPv6 processors.” Ex.1005, ¶ 63. It would have been obvious to a POSITA that the protocol converter in Ananda’s CPI is a type of communication processor which can process both IPv4 and IPv6 communications. Ex.1003, ¶ 171.

Additionally, Ananda discloses that the CPI comprises a memory for “storing” relevant information. Ex.1005, claims 6, 9 and 13. Ananda’s CPI 215E therefore has the same structure as the device 25 (including the processor 25₁ and the memory 25₅) found within home gateway 20 or access node 40 of the ’844 patent, as discussed above in Section VIII.B.1. Ex.1003, ¶ 171-172.

As discussed for limitation [1.1], a POSITA would have been motivated to modify Ananda's IPv6 address construction technique to incorporate a destination port number in the IPv6 address, as taught by Wetterwald and RFC4380. To assist with maintaining port number information and performing "IPv6-to-IPv4 translations and vice versa" (Ex.1008, 6:38-47), it would have been obvious to a POSITA to modify Ananda's CPI 215E to store within its memory a NAT table, like that taught by Wetterwald, for translating addresses. *See supra*, § [1.0.2]; Ex.1003, ¶¶ 173-175. A POSITA would have recognized the advantages of storing such address translations, including facilitating a simpler address translation process for other packets with the same source or destination address. Ex.1003, ¶ 175.

Thus, the prior art renders obvious limitation [5.1].

[5.2] means for generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet; and

Limitation [5.2] is identical to limitation [1.2], except for the addition of "means for." For the reasons at [1.2] above, the prior art renders obvious the function of the "means for generating" in [5.2]. Ex.1003, ¶ 177.

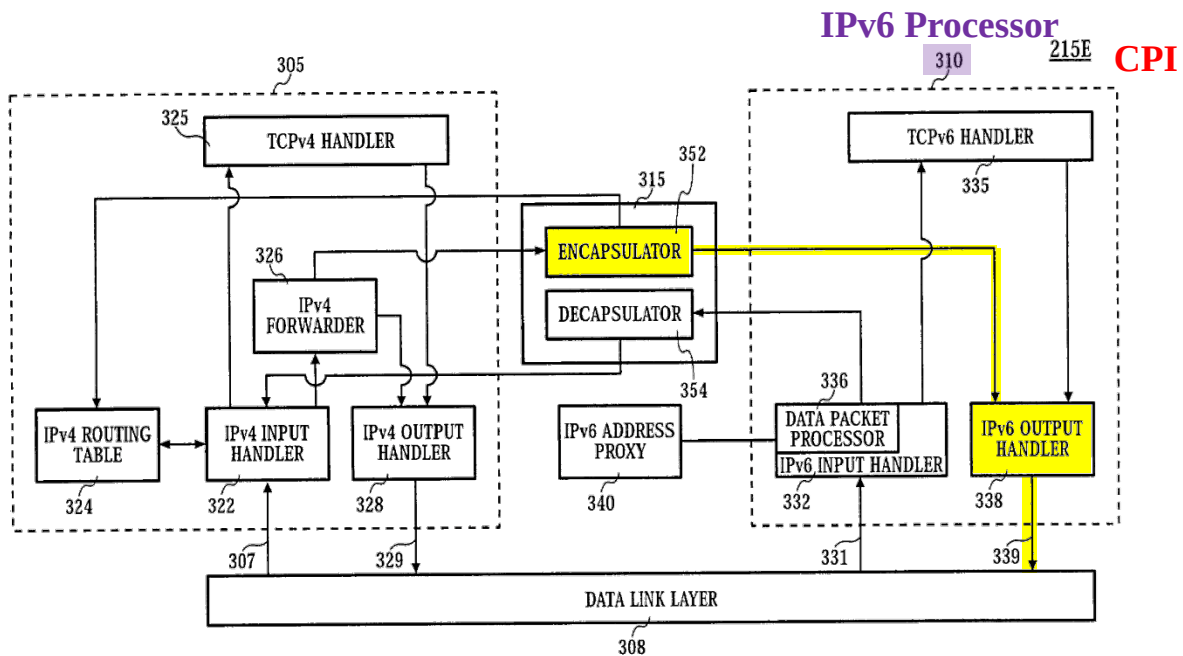
Within Ananda's CPI, protocol converter 315 includes an encapsulator 352 that performs the steps of "converting IPv4 communication... to resultant IPv6 communication." Ex.1005, ¶¶ 89, 103, Fig. 4. Based on Ananda's suggestion that

the entire CPI be “implemented as software modules,” it would have been obvious to a POSITA for Ananda’s encapsulation steps to be implemented as computer program instructions stored in a memory to be executed by a processor within the protocol converter, and it would have been obvious to store those instructions in a memory. Ex.1005, ¶ 118; Ex.1003, ¶ 177-180. Thus, the prior art renders obvious the corresponding structure of [5.2], including a processor executing computer program instructions to perform the recited function in limitation [5.2].

[5.3.1] means for routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address,

Limitation [5.3.1] is identical to limitation [1.3.1], except for the addition of “means for.” For the reasons at [1.3.1] above, the prior art renders obvious the function of the “means for routing” in [5.3.1].

Ananda teaches that an IPv6 output handler 338 of CPI 215E receives a resultant IPv6 data packet from the encapsulator 352 and “provides the resultant IPv6 data packet to the data link layer 308 via the output 339,” as illustrated in annotated Figure 3 below. Ex.1005, ¶ 98. The resultant IPv6 data packet is thus “transmitted ... by the IPv6 output handler 338 to a host beyond the multi-protocol subnet 215.” Ex.1005, ¶ 112. The packet is subsequently routed over an IPv6 backbone including routers “which only route IPv6 data packets.” Ex.1005, ¶¶ 77, 139-141.



Ex.1003, ¶182; Ex.1005, Fig. 3 (annotated).

It would have been obvious to a POSITA that transmitting an IPv6 packet via the IPv6 output handler 339, data link layer 308, and IPv6 backbone 210 would include the use of an IPv6 routing core. Ex.1003, ¶ 183. For example, each device in the transmission path (including CPI 215E) would use the IPv6 packet's destination address to route the packet, e.g., by identifying a physical port for outputting the packet toward its destination. Ex.1003, ¶ 183.

Thus, Ananda, Wetterwald, and RFC 4380 render obvious limitation [5.3.1].

[5.3.2] said constructed address belonging to a range of IPv6 addresses routable to an interconnection equipment of the IPv6 domain with the IPv4 destination address.

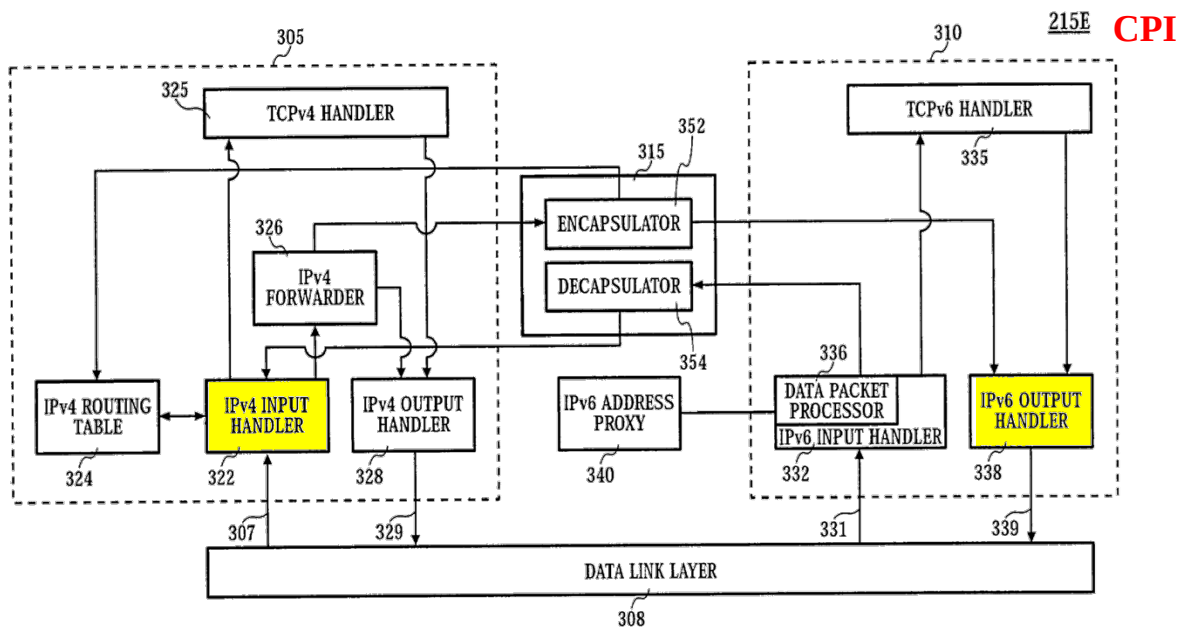
See analysis of [1.3.2] above, which is identical to limitation [5.3.2].

10.Claim 6

[6.0] Access equipment to an IPv6 domain comprising an interface with at least one IPv4 domain, further comprising:

As discussed above at [1.0.1] and [1.3.2], Ananda's CPIs are "interconnection equipment" that provide communication connectivity between IPv4 devices and an IPv6 backbone. See, e.g., Ex.1005, Abstract, ¶¶ 18, 35, 63.

Ananda further describes CPI 215E as including an IPv4 input handler 322 ("an interface with at least one IPv4 domain"). Ex.1005, ¶ 90. CPI 215E further includes an IPv6 output handler providing communications to the IPv6 backbone ("to an IPv6 domain"). Ex.1005, ¶ 98. CPI 220E has substantially the same design as CPI 215E. Ex.1005, ¶ 79. Thus, CPIs 215E, 220E are "access equipment to an IPv6 domain comprising an interface with at least one IPv4 domain." Ex.1003, ¶ 188.



Ex.1003, ¶ 188; Ex.1005, Fig. 3 (annotated).

[6.1] a device according to claim 5 for receiving a data packet coming from an IPv4 domain in the IPv6 domain; and

As discussed above for claim 5, Ananda’s CPI 215E—in combination with the other prior art—renders obvious “a device according to claim 5.” CPI 215E receives a packet from subnet 215 (“an IPv4 domain”). Ex.1005, ¶¶ 82, 90. CPI 215E also provides IPv6 connectivity to the IPv6 backbone and is therefore “in the IPv6 domain.” Ex.1005, ¶¶ 79, 98; Ex.1003, ¶ 191.

[6.2.1] a device for sending a data packet from an IPv6 domain to an IPv4 domain,

As discussed above in claim 3, the prior art renders obvious “sending a data packet from an IPv6 domain to... an IPv4 domain.” Specifically, Ananda’s CPI

220E is a device that performs steps 1012 and 1014 to decapsulate and deliver a received packet to an IPv4 destination. Ex.1005, ¶¶ 137-144; Ex.1003, ¶ 192.

[6.2.2] said packet comprising an IPv6 destination address of the packet constructed by concatenating an operator prefix, the IPv4 destination address, and a destination port number,

See analysis of limitation [3.0.2], which recites an identical limitation.

[6.2.3] said device, on reception of the data packet, making use of means for extracting the IPv4 destination address and the destination port number from the IPv6 destination address;

As discussed above at [3.0.3], [3.1], the prior art renders performing an “extracting” step “on reception of the data packet.” The prior art therefore renders obvious the function of the “means for extracting” recited in limitation [6.2.3].

And as discussed above at [4.0]-[4.1], the prior art renders obvious a processor performing the method of claim 3.

Regarding the use of an address translation table (NAT table), Wetterwald describes **translator 50** also includes an **IPv4 NAT/PAT translation resource 53** that translates between a public IPv4 address and a private IPv4 address, as shown in annotated Figure 5 below. Ex.1008, 7:1-21. Thus, after the translation resource 51 of the **translator 50** translates a received IPv6 packet into an IPv4 packet (Ex.1008, 6:38-56, 7:37-41), the **translation resource 53** uses a NAT table 54 to translate between public and private IPv4 addresses, as described in RFC 1918

(Ex.1007). Ex.1008, 7:1-21, 1:37-40; Ex.1003, ¶ 197.

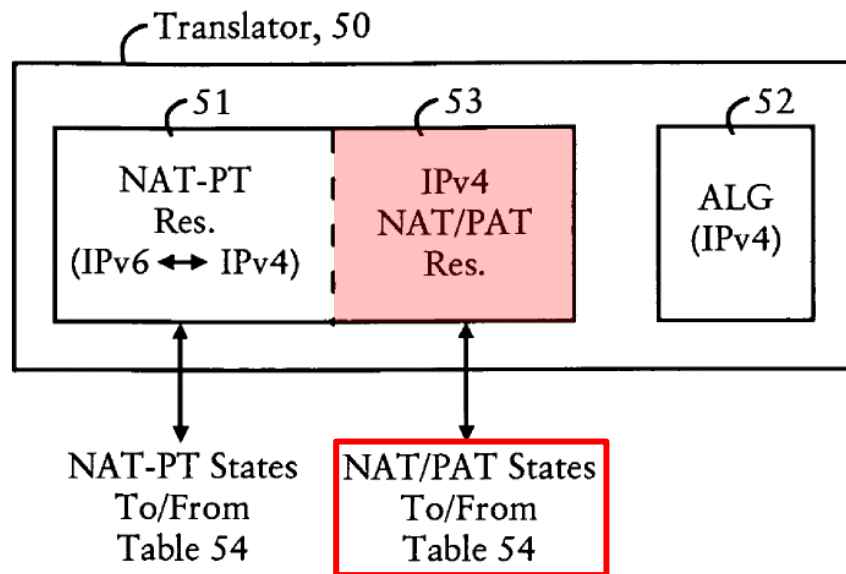


Figure 5

Ex.1003, ¶ 197; Ex.1008, Fig. 5 (annotated).

Using the NAT table 54, Wetterwald’s **translator 50** also performs port address translation (PAT). Ex.1008, 1:7-14. Wetterwald explains that the **translation resource 53** includes PAT functionality that enables the clusterhead 32 to “reuse an assigned public address 112 for multiple connections on the IPv4 network 16,” by translating between the assigned public address 112 and a private IPv4 address. Ex.1008, 7:1-7.

A POSITA would have been motivated to modify Ananda to include a NAT table (like Wetterwald’s NAT 54) to associate public and private IPv4 addresses so

that the CPI 220B would perform public-to-private IPv4 translation, just like the **translation resource 53** of Wetterwald. Ex.1003, ¶ 198. This modification would permit IPv4 host 215B and IPv4 host 220B to use private IPv4 addresses (Ex.1005, ¶ 82), and not reveal their private addresses when transmitting packets outside their respective subnet 215, 220, as recommended by RFC 1918. Ex.1003, ¶ 199; Ex.1007, §§ 3, 5 (packets with private source or destination addresses should not be forwarded over public links). Wetterwald's **translation resource 53** performs public-to-private IPv4 translations as described in RFC 1918. Ex.1008, 7:1-3.

To facilitate similar public-to-private IPv4 address translations by Ananda's similar CPI devices, it would have been obvious to a POSITA for Ananda's CPI 220E to include an address translation table like that taught by Wetterwald. Ex.1003, ¶ 200. A POSITA would have been motivated to follow the guidance in RFC 1918 that packets with private addresses should not be forwarded over public links (e.g., IPv6 domain 14 in Wetterwald, which is similar to Ananda's IPv6 backbone 210). *Id.* A POSITA would have had an expectation of success in the combination because Wetterwald teaches employing public-private IPv4 address translations in the context of IPv4<->IPv6 address translations. *Id.*

Thus, the prior art renders obvious employing a processor and an address translation table to perform the “*extracting*” function.

[6.3.1] means for generating an IPv4 data packet from the IPv6 data packet,

Limitation [6.3.1] is identical to limitation [3.2.1], except for the addition of “means for.” For the reasons at [3.2.1] above, the prior art renders obvious the function of the “means for generating” in [6.3.1].

The prior art also renders obvious the corresponding structure. Within Ananda’s CPI, protocol converter 315 includes an decapsulator 354 that “decapsulates the IPv6 data packet and provides a resultant IPv4 data packet.” Ex.1005, ¶¶ 102, Fig. 4. Based on Ananda’s suggestion that the entire CPI be “implemented as software modules,” it would have been obvious to a POSITA for Ananda’s decapsulation steps to be implemented as computer program instructions stored in a memory to be executed by a processor within the protocol converter, and it would have been obvious to store those instructions in a memory. Ex.1005, ¶ 118; Ex.1003, ¶ 203-206. Thus, the prior art renders obvious the corresponding structure of [6.3.1], including a processor executing computer program instructions to perform the recited “generating” function.

[6.3.2] said packet including said IPv4 destination address and said destination port number; and

See analysis of [3.2.2] above, which is identical to limitation [6.3.2].

[6.4] means for routing the IPv4 data packet to its IPv4 destination.

Limitation [6.4] is identical to limitation [3.3], except for the addition of “means for.” For the reasons at [3.3] above, the prior art renders obvious the

function of the “means for routing” in [6.4].

CPI 220E, which Ananda suggests may be incorporated in a network interface card, includes a data link layer 308 that facilitates communicating with the network. Ex.1005, ¶¶ 118, 90, 92; *see also* Ex.1001, 12:64-65. It would have been obvious to a POSITA that the data link layer would include telecommunication means, which the ’844 patent admits is a conventional component. Ex.1003, ¶ 210; Ex.1001, 12:60-65.

11.Claim 8

[8.0] IPv6 domain access equipment according to claim 6,

See analysis of claim 6.

[8.1] wherein the IPv4 domain is a private network,

As discussed at [6.1] and [5.0.1], Ananda’s subnet 215 is an “IPv4 domain.” And as explained in Section X.E.1 above, Ananda discloses that the subnet 215 corresponds to pre-transition subnet 115, which is a private network, and host 215B is assigned a “private IPv4 address[.]” Ex.1005, ¶¶ 83-84, 67; *see also* Ex.1008, 5:2-4, 6:1-5. Thus, it would have been obvious to a POSITA for subnet 215 to be a “private network.” Ex.1003, ¶ 214.

[8.2] the access equipment is a home gateway adapted to connect user terminals of the private network to the IPv6 domain.

As discussed at [6.0], each CPI 215E, 220E is “access equipment.” The CPI

215E is “a default gateway” that connects IPv4 hosts in subnet 215 (“*user terminals of the private network*”) to the IPv6 backbone 210 (“*IPv6 domain*”).

Ex.1005, ¶¶ 137-139; Ex.1003, ¶ 217. Thus, Ananda’s CPI 215E renders obvious the claimed “*home gateway*.” Ex.1003, ¶ 217.

12.Claim 10

[10.0.1] A device for sending a data packet from an IPv6 domain to an IPv4 domain,

Limitation [10.0.1] is identical to limitation [3.0.1] above, except that limitation [3.0.1] recites a “method.” As discussed at [3.0.1], Ananda discloses a device in the form of CPI 215E. Ex.1005, ¶¶ 35, 18; Ex.1003, ¶ 218.

[10.0.2] said packet comprising an IPv6 destination address of the packet constructed by concatenating an operator prefix, the IPv4 destination address, and a destination port number,

See analysis of [3.0.2].

[10.0.3] said device, on reception of the data packet, making use of:

[10.1] means for extracting the IPv4 destination address and the destination port number from the IPv6 destination address;

[10.2.1] means for generating an IPv4 data packet from the IPv6 data packet,

[10.2.2] said packet including said IPv4 destination address and said destination port number; and

[10.3] means for routing the IPv4 data packet to its IPv4 destination.

See analysis of identical limitations [6.2.3]-[6.4].

F. Ground 2: Claim 7 is obvious over Ananda, Wetterwald, RFC4380, and RFC4271

1. RFC4271

RFC4271 discloses the specification for the Border Gateway Protocol (BGP) for “exchang[ing] network reachability information.” Ex.1046, 1. For example, “a BGP speaker” may advertise to its peers its “most preferred BGP route[s],” that “a previously advertised route is no longer available for use” or that “a replacement route” is available. Ex.1046, 7, 9-10. BGP also supports “advertising a set of destinations as an IP prefix.” Ex.1046, 1.

2. Reasons to combine

An explanation of the multiple reasons to combine the teachings of Ananda, Wetterwald, RFC4380, and RFC4271 are provided in the analysis below.

3. Claim 7

[7.0] IPv6 domain access equipment according to claim 6,

See analysis of claim 6.

[7.1] wherein the IPv4 [domain] is a public network,

Hosts on Ananda’s subnet 215 may be assigned “global IPv4 addresses,” which Ananda distinguishes from “private IPv4 addresses.” Ex.1005, ¶ 84. It would have been obvious to a POSITA that IPv4 hosts having globally routable IP addresses are part of a *public network*. Ex.1003, ¶ 230. Ananda further describes connecting subnet 215 to “the IPv4 portion 105 of the Internet.” Ex.1005, ¶¶ 91,

63. Thus, it would have been obvious to a POSITA for subnet 215 (*“the IPv4 domain”*) to be a *public network*. Ex.1003, ¶ 230-231.

[7.2] said equipment comprising first means for advertising to the IPv4 domain IPv4 addresses of network equipments connected to the IPv6 domain and

Ananda’s CPI 215E (*“said equipment”*) enables the IPv4 hosts in subnet 215 to communicate with other hosts via the IPv6 communication backbone 210 (*“network equipments connected to the IPv6 domain”*). Ex.1005, ¶¶ 63, 78, 82. To facilitate these communications, CPI 215E includes an “IPv4 routing table 324 to determine where the IPv4 data packet should be routed.” Ex.1005, ¶ 90.

RFC4271 teaches a Border Gateway Protocol (BGP) that provides “support for **advertising a set of destinations** as an IP prefix.” Ex.1046, 1. Routers implementing BGP exchange “reachability information” with one another, allowing each router to construct a map (“graph”) of network connectivity. Ex.1046, 7. The information exchange allows each router to build a routing table with which it can determine how to forward packets toward their destinations specified in the IP packet header. Ex.1046, 7. “Incremental updates are sent as the routing tables change.” Ex.1046, 8. RFC4271 also describes using a similar “Interior Gateway Protocol” (IGP) to “exchange routing information among routers.” Ex.1046, 5.

It would have been obvious to a POSITA for CPI 215E to implement BGP

or IGP to advertise destinations reachable through CPI 215E, including the IPv4 hosts on subnet 215 (“*IPv4 addresses of network equipments connected to the IPv6 domain*”). Ex.1003, ¶ 233. As RFC4271 notes, BGP is an “inter-Autonomous System routing protocol,” meaning that it is designed for use in exchanging routing information between routers under different technical administrations, but may also be used within an autonomous system. Ex.1046, 4-5; *See* Ex.1037, 4:39-40; Ex.1003, ¶ 233. Since Ananda describes each subnet as a “physically independent network segment,” it would have been obvious to a POSITA for these independent subnets to represent separate “autonomous systems” for BGP. Ex.1005, ¶ 66; Ex.1003, ¶ 233. The CPIs 215E and 220E would therefore exchange routing information with routers they are connected to—each other—via BGP. Accordingly, it would have been obvious for the processors in CPIs 215E and 220E to execute instructions to advertise the IPv4 addresses of connected network equipment using BGP or IGP. Ex.1003, ¶ 233.

A POSITA would have had an expectation of success because advertising routes was a known technique and BGP was a well-known and commonly used routing protocol. Ex.1003, ¶ 233; *see also* Ex.1038, 1:20-23; Ex.1039, 1:22-25. As discussed above at [1.1]-[1.3.2] and [2.0]-[2.1], the prior art renders obvious a computer processor executing instructions to perform the claimed steps.

[7.3] second means for advertising in the IPv6 domain IPv6 addresses constructed from IPv4 addresses of the IPv4 domain.

As discussed above in [7.2], it would have been obvious for Ananda's CPIs to include a processor executing instructions for BGP or IGP to advertise destination addresses reachable through them. It was well-known in the art to use routing protocols such as BGP or IGP to exchange routing information for IPv6 destinations. Ex.1003, ¶ 235; see, e.g., Ex.1037, 4:19-26. So that IPv6 routers would have the information needed to direct packets addressed to IPv4 hosts accessible via IPv6 addresses (as discussed at [1.1], "*IPv6 addresses constructed from IPv4 addresses of the IPv4 domain*"), it would have been obvious to a POSITA for CPIs 215E, 220E to advertise virtual IPv6 addresses in the IPv6 domain. Ex.1003, ¶ 235. As discussed above at [1.1]-[1.3.2] and [2.0]-[2.1], the prior art renders obvious a computer processor executing instructions to perform the claimed steps.

G. Ground 3: Claim 9 is obvious over Ananda, Wetterwald, RFC4380, and RFC1918

1. RFC1918

RFC1918 discloses many advantages of using private IPv4 network addresses. Ex.1007, 2-5. As discussed in Section X.E.1, Ananda's IPv4 hosts 215B, 220B may be assigned private IPv4 addresses.

2. Reasons to combine Ananda, Wetterwald, RFC4380, and RFC1918

Wetterwald expressly refers to RFC1918 (Ex.1008, 1:36-41), so it would have been obvious to consult RFC1918 when considering the disclosure of Wetterwald. Ex.1003, ¶ 238. Further reasons to combine are provided below.

3. Claim 9

[9.0] IPv6 domain access equipment according to claim 8, comprising means for obtaining an IPv4 address, a range of port numbers, and a range of authorized IPv6 addresses,

Ananda discloses an addressing assignment scheme for the multi-protocol subnet 215, in which “IPv4 hosts and dual-stack hosts can be assigned global IPv4 addresses or private IPv4 addresses.” Ex.1005, ¶ 84. For example, “the IPv4 host 215B is assigned an IPv4 address, and also a virtual IPv6 address [and] the dual-stack host 215C is assigned an IPv4 address, a virtual IPv6 address, and an IPv6 address.” Ex.1005, ¶ 83. As Ananda describes that a DHCP service may be employed such that “[t]he IPv4 address assignment can be static or dynamic,” a POSITA would have understood that the CPI 215E (“*IPv6 domain access equipment*”) would include “*means for obtaining an IPv4 address*” from such a DHCP service (e.g., as provided by DHCP server 215A in subnet 215). Ex.1003, ¶ 240; Ex.1005, ¶¶ 78, 84 & Fig. 2; Ex.1036, 1, 3, 12 (“[a] service provided by DHCP is the allocation of temporary or permanent network (IP) addresses”); *see*

also Ex.1043, ¶¶ 81-84 (citing Exs.1064-1065).

Ananda further describes that “[i]n relation to IPv6 address assignments, **each network interface of an IPv6 host or a dual-stack host can have multiple IPv6 addresses....**” Ex.1005, ¶ 85.

Therefore, a POSITA also would have understood that the CPI 215E includes a means for obtaining a *range of IPv6 addresses*. Ex.1003, ¶ 240-242.

As discussed previously at [1.1], Wetterwald and RFC4380 further teach that it was known, when encapsulating IPv4 packets in IPv6 packets, to include IPv6 extension headers to identify the source and destination port numbers. Accordingly, for the reasons discussed at [1.1] (e.g., address reuse), it would have been obvious to a POSITA that “means for obtaining” in the CPI 215E of Ananda may be used to obtain “*a range of port numbers*” in addition to “*an IPv4 address*” and “*a range of ... IPv6 addresses.*” Ex.1003, ¶ 243.

To the extent that Ananda, Wetterwald, and RFC4380 do not explicitly disclose that the range of IPv6 addresses are “*authorized IPv6 addresses,*” RFC1918 teaches using authority servers to ensure that DNS clients outside of an enterprise do not see addresses in the private address space used by the enterprise, since these addresses would be ambiguous. Ex.1007, 7. For example, RFC1918 describes using two authority servers, where “[o]ne server would be visible from the public address space” and “contain only [a] **subset of the enterprise’s**

addresses which were reachable using public addresses” while the “other server would be reachable only from the private network.” Ex.1007, 7. Although this discussion is in the context of IPv4 addresses, it would have been obvious to a POSITA to use a similar authorization scheme to restrict the available IPv6 addresses to an authorized range of users within Ananda’s communication network. Ex.1003, ¶ 244.

Thus, Ananda, Wetterwald, RFC4380, and RFC1918 render obvious limitation [9.0]. Ex.1003, ¶¶ 239-245.

[9.1] said range of IPv6 addresses being constructed by concatenating an operator prefix, said IPv4 address, and the range of authorized IPv4 port numbers.

As discussed above at [1.1], the prior art renders obvious “*constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number.*” It would have been obvious to a POSITA, when assigning IPv6 addresses used for communications between an IPv4 domain and an IPv6 domain, for the range of assigned IPv6 addresses to correspond to the constructed IPv6 addresses. Ex.1003, ¶ 247.

XI. CONCLUSION

Petitioner asks that the Patent Office order an *inter partes* review trial for claims 1-10 and then cancel these claims as unpatentable.

Respectfully submitted,

October 1, 2020

/ Theodore M. Foster /
Theodore M. Foster
Counsel for Petitioner
Registration No. 57,456

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,996. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: October 1, 2020

/ Theodore M. Foster /
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456

CERTIFICATE OF SERVICE

The undersigned certifies, under 37 C.F.R. §§ 42.105 and 42.6, that service was made on the Patent Owner as detailed below.

Date of service October 1, 2020

Manner of service Federal Express

Documents served Petition for *Inter Partes* Review, including Exhibit List;

Exhibits 1001-1010, 1012-1020, 1022-1033,
1036-1055, and 1058-1071

Persons served Faegre Drinker Biddle & Reath LLP (Chicago)

Attn: Patent Docket Dept.

191 N. Wacker Drive, Suite 3700

Chicago, IL 60606

/Theodore M. Foster/
Theodore M. Foster
Counsel for Petitioner
Registration No. 57,456