

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.,
Petitioner,

v.

MONARCH NETWORKING SOLUTIONS LLC,
Patent Owner.

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

“Method of Receiving a Data Packet Coming from an IPv4 Domain in an IPv6
Domain, an Associated Device, and Associated Access Equipment”

IPR2024-00485

**SUPPLEMENTAL DECLARATION OF DR. SAMRAT
BHATTACHARJEE IN SUPPORT OF PETITIONER’S REPLY
TO PATENT OWNER’S RESPONSE**

I.	INTRODUCTION	1
II.	ALL GROUNDS: DR. MANGIONE-SMITH’S CONSTRUCTION OF “DOMAIN” IS NOT SUPPORTED BY THE PATENT OR EXTRINSIC EVIDENCE	1
III.	ALL GROUNDS: FIGURE 7 DEPICTS AN INTER-DOMAIN NETWORK	4
IV.	ALL GROUNDS: FIGURES 8, 10, AND 11 ALSO MAP TO THE ’844 CLAIMS	8
V.	GROUND 2: THE SIMULTANEOUS INVENTIONS DISCLOSE ALL OF THE ’844 CLAIM’S LIMITATIONS	12
A.	The IVI References Discloses All Limitations of the ’844 Claims	12
1.	Claim 1	12
2.	Claim 3	27
B.	Bush Discloses All Limitations of the ’844 Claims	37
1.	Claim 1	39
2.	Claim 3	47
VI.	GROUND 3: WU’S ADDRESS CONSTRUCTION IS STATELESS	52
VII.	GROUND 3: WU TEACHES CONSTRUCTING AN IPV6 ADDRESS COMPRISING A DESTINATION PORT NUMBER	55
VIII.	CONCLUSION	62

PETITIONER'S EXHIBIT LIST

EX1001	U.S. Patent No. 8,451,844 to Mohamed Boucadair et al. ("844 patent")
EX1002	Prosecution History of U.S. Patent No. 8,451,844 ("844 Prosecution History")
EX1003	Declaration of Dr. Samrat Bhattacharjee under 37 C.F.R. §1.68
EX1004	<i>Curriculum Vitae</i> of Dr. Bhattacharjee
EX1005	Declaration of Dr. Sylvia Hall-Ellis under 37 C.F.R. §1.68
EX1006	<i>Curriculum Vitae</i> of Dr. Hall-Ellis
EX1007	Declaration of Alexa Morris under 37 C.F.R. §1.68
EX1008	Declaration of Ron Bonica under 37 C.F.R. §1.68
EX1009	Declaration of Alexa Morris under 37 C.F.R. §1.68
EX1010	Partial <i>Ex Parte</i> Reexamination History of the '844 patent as U.S. Application No. 90/014,717
EX1011	RFC 791, "Internet Protocol Specification" (Sept. 1981), available at https://datatracker.ietf.org/doc/html/rfc791
EX1012	RFC 793, "Transmission Control Protocol" (Sept. 1981), available at https://datatracker.ietf.org/doc/html/rfc793
EX1013	P. Richter et al., "A Primer on IPv4 Scarcity" (Feb. 27, 2015), available at https://arxiv.org/pdf/1411.2649.pdf
EX1014	U.S. Patent Application Publication No. 2006/0146870 to Harvey et al. ("Harvey")
EX1015	RFC 2460, "Internet Protocol, Version 6 Specification" (Dec. 1998), available at https://datatracker.ietf.org/doc/html/rfc2460
EX1016	RFC 1884, "IP Version 6 Addressing Architecture" (Dec. 1995), available at https://datatracker.ietf.org/doc/html/rfc1884

EX1017	RFC 3587, "IPv6 Global Unicast Address Format" (Aug. 2003), available at https://www.rfc-editor.org/rfc/pdf/rfc3587.txt.pdf
EX1018	Pete Loshin, "IPv6: Theory, Protocol, and Practice" (2d Ed. 2004)
EX1019	U.S. Patent No. 7,391,768 to Samprathi et al. ("Samprathi")
EX1020	T. Rooney, "IPv4-to-IPv6 Transition Strategies," INS (Feb. 2007), available at https://www.hpc.mil/images/hpcdocs/ipv6/migration_white_paper_1e5eb697-e7d7-4f62-9452-0121aa0ff0f1.pdf
EX1021	H. Yeom et al., "IP Multiplexing by Transparent Port-Address Translator," 10th USENIX System Administration Conference (LISA X) (Sept. 29, 1996), available at https://www.usenix.org/conference/lisa-96/ip-multiplexing-transparent-port-address-translator
EX1022	A. Reitz, "The Dynamic Port Reservation Protocol," Univ. Ill. at Urbana-Champaign (2001), available at http://redefine.dyndns.org/~andyr/documents_and_writings/thesis/docs/areitz_msthesis.pdf
EX1023	Y. Zhu et al., "Stateless Mapping and Multiplexing of IPv4 Addresses in Migration to IPv6 Internet," Institute of Electrical and Electronics Engineers ("IEEE") (Nov. 2008), available at https://ieeexplore.ieee.org/document/4698208
EX1024	U.S. Patent No. 8,032,639 to King et al. ("King")
EX1025	U.S. Patent No. 8,019,889 to Biswas et al. ("Biswas")
EX1026	X. Li et al., "Bi-Directional Address Borrowing (IVI)" (June 23, 2008)
EX1027	X. Li et al., "Prefix-Specific and Stateless Address Mapping (IVI) for IPv4/IPv6 Coexistence and Transition," IETF (July 6, 2008), available at https://datatracker.ietf.org/doc/html/draft-xli-behave-ivi-00
EX1028	X. Li et al., "Prefix-Specific and Stateless Address Mapping (IVI) for IPv4/IPv6 Coexistence and Transition," IETF-72, Dublin (July 29,

	2008), available at https://www.ietf.org/proceedings/72/slides/behave-5.pdf
EX1029	R. Bush et al., "A Better Approach than Carrier-Grade-NAT," Columbia Univ. (Sept. 2008), available at https://academiccommons.columbia.edu/doi/10.7916/D89S1ZWC
EX1030	U.S. Patent Application Publication No. 2004/0190549 to Huitema et al. ("Huitema")
EX1031	U.S. Patent No. 7,149,225 to Thubert et al. ("Thubert")
EX1032	C. Huitema, "Teredo: Tunneling IPv6 over UDP Through NATs," Internet Engineering Task Force ("IETF") (Sept. 17, 2002), available at https://www.ietf.org/archive/id/draft-ietf-ngtrans-shipworm-08.txt
EX1033	Petition for <i>Inter Partes</i> Review, <i>Cisco Sys., Inc. v. Monarch Networking Sol'ns LLC</i> , IPR2020-01678, Paper 2 (PTAB Oct. 1, 2020)
EX1034	Patent Owner's Preliminary Response, <i>Cisco Sys., Inc. v. Monarch Networking Sol'ns LLC</i> , IPR2020-01678, Paper 6 (PTAB Feb. 16, 2021)
EX1035	Institution Decision, <i>Cisco Sys., Inc. v. Monarch Networking Sol'ns LLC</i> , IPR2020-01678, Paper 9 (PTAB Mar. 31, 2021)
EX1036	Claim Construction Memorandum and Order, <i>Monarch Networking Sol'ns LLC v. Cisco Sys., Inc.</i> , No. 2:20-cv-00015-JRG (E.D. Tex. Jan. 7, 2021)
EX1037	X. Li et al., "Forwarding IPv4 Traffics in Pure IPv6 Backbone with Stateless Address Mapping," IEEE (Apr. 2006), available at https://ieeexplore.ieee.org/document/1687557 ("Li")
EX1038	X. Li, "Bi-directional Address Borrowing (IVI)," IETF (June 21, 2008), available at https://www.ietf.org/staging/draft-xli-ivi-00-7430-cancelled.txt ("Li-2")
EX1039	Certified Translation of CN Patent No. 1525699 to Wu et al. ("Wu")
EX1040	CN Patent No. 1525699 to Wu et al.

EX1041	R. Bush, "IPv4 / IPv6 Coexistence Mechanisms" (Oct. 26, 2008)
EX1042	Monarch's Opposition to Juniper's Motion to Transfer, <i>Monarch Networking Solutions LLC v. Juniper Networks, Inc.</i> , No. 1:23-cv-670-TSE-LRV (E.D. Va. Oct. 3, 2023), ECF No. 52
EX1043	Complaint, <i>Monarch Networking Sol'ns LLC v. Juniper Networks, Inc.</i> , No. 1:23-cv-670-TSE-LRV (E.D. Va. May 23, 2023), ECF No. 1
EX1044	Screenshot IETF Staging Folder, available at https://www.ietf.org/staging/
EX1045	R. Bonica, "draft-kumar-mpls-fec-to-nhlfe-mib-00," IETF (Jan. 7, 2010), available at https://www.ietf.org/staging/draft-kumar-mpls-fec-to-nhlfe-mib-00-16143-cancelled.txt
EX1046	Correspondence from badra@isima.fr (Oct. 9, 2007), available at https://mailarchive.ietf.org/arch/msg/ngo/6JM4rSBIZM8Ajw02luUG4OqSrmQ/
EX1047	Correspondence from dirk.hugo@t-systems.com (Oct. 12, 2007), available at https://mailarchive.ietf.org/arch/msg/multimob/H8V3v2MvxhYRhjqOKSN7mhM39Q8/
EX1048	Correspondence from dirk.hugo@t-systems.com (Oct. 25, 2007), available at https://mailarchive.ietf.org/arch/msg/multimob/FvyjLiCNdh88V2I3Fm8B21vMtkA/
EX1049	Correspondence from asaeda@sfc.wide.ad.jp (Nov. 12, 2007), available at https://mailarchive.ietf.org/arch/msg/multimob/6Zj-nouoPaLdo46ART2ZGWMJqkE/
EX1050	Correspondence from fluffy@cisco.com (Nov. 13, 2007), available at https://mailarchive.ietf.org/arch/msg/tools-development/CPLnZyjcureXq6j6mII5eTZF1Zo/
EX1051	Correspondence from lowekamp@sipeerior.com (Nov. 14, 2007), available at

	https://mailarchive.ietf.org/arch/msg/sipping/70F4vroEAckioD214LTt2FJpInk/
EX1052	Correspondence from lowekamp@sipeerior.com (Nov. 17, 2007), available at https://mailarchive.ietf.org/arch/msg/behave/yUKKzuhlhnKkRRC0V_zGXEKRP4/#
EX1053	Correspondence from schierl@hhi.fhg.de (Nov. 19, 2007), available at https://mailarchive.ietf.org/arch/msg/mmusic/4FxACR25hKzg59oSR-uSoqoSjc0/
EX1054	Correspondence from andrew.dolganow@alcatel-lucent.com (Feb. 11, 2008), available at https://mailarchive.ietf.org/arch/msg/pwe3/ceWfUIVFI-tMV-1_ueo7HcNCums/
EX1055	Correspondence from victor.pascual.avila@gmail.com (Feb. 23, 2008), available at https://mailarchive.ietf.org/arch/msg/p2psip/P71OF43ons3EYw7UKg0_u3M_tww/
EX1056	October 8, 2006 Capture of IETF Datatracker, available at https://web.archive.org/web/20061008141652/https://datatracker.ietf.org/public/pidtracker.cgi
EX1057	Supplemental Declaration of Dr. Samrat Bhattacharjee under 37 C.F.R. §1.68
EX1058	R. Kuhn, K. Sriram, & D. Montgomery, "Border Gateway Protocol Security: Recommendations of the National Institute of Standards and Technology," NIST (July 2007), available at https://csrc.nist.gov/library/NIST%20SP%20800-054%20Border%20Gateway%20Protocol%20Security,%202007-07-17%20(Final).pdf
EX1059	S. Kent et al., "Secure Border Gateway Protocol (S-BGP)—Real World Performance and Deployment Issues," BBN Techs. (Feb. 2000), available at

	http://dpm.postech.ac.kr/research/08/KT_BGP/BGP_Security/sbgp.pdf
EX1060	I. Kalaydjieva, "The Border Gateway Protocol and Its Convergence Properties," Tech. Univ. Munich (June 2003), available at https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4eb2026d1c6e9babb414ddbc1359c06c92a340a0
EX1061	RFC 1771, "A Border Gateway Protocol 4 (BGP-4)," IETF (Mar. 1995), available at https://datatracker.ietf.org/doc/html/rfc1771
EX1062	J. Verdu et al., "The Impact of Traffic Aggregation on the Memory Performance of Networking Applications," J. Embedded Comput. (2006), available at https://futur.upc.edu/654889
EX1063	U.S. Patent No. 8,451,845 to Mohamed Boucadair et al. ("845 patent")
EX1064	Final Rejection, EPR Control No. 90/019,246 (June 3, 2024)
EX1065	Reexamination Certificate, EPR Control No. 90/019,246 (Oct. 24, 2024)

I. INTRODUCTION

1. I have been retained by Quinn Emanuel Urquhart & Sullivan, LLP on behalf of Juniper Networks, Inc. ("Juniper" or "Petitioner") as an independent expert in this *inter partes* review before the Patent Trial and Appeal Board of the United States Patent and Trademark Office to review claims 1-10 ("Challenged Claims") of U.S. Patent No. 8,451,844 ("the '844 patent").

2. In forming my opinions, I have considered and relied on my education, knowledge of the relevant field, knowledge of scientific and engineering principles, and my general experience in the field of art. I have reviewed the Patent Owner Response ("POR"), all exhibits, and the declaration of Dr. Mangione-Smith, as well as other materials and references cited in this declaration.

II. ALL GROUNDS: DR. MANGIONE-SMITH'S CONSTRUCTION OF "DOMAIN" IS NOT SUPPORTED BY THE PATENT OR EXTRINSIC EVIDENCE

3. I disagree with Dr. Mangione-Smith's assertion that "domain" means "a network or subnetwork and its address space that are under common control" and further implies that each domain must be controlled by a "different entit[y]." EX2015 ¶¶46-49. The patent defines domain to mean "[a] telecommunications network administered by an operator." EX1001, 1:30-31; EX2015 ¶47. This definition says no such thing about control, much less common control by different entities. It merely states that an operator must administer the domain. A POSITA

thus would not further confine this definition to include common control by different entities.

4. Dr. Mangione-Smith also cites various portions of the specification to argue that domains must be controlled by different entities so that the controlling entity can take “precautions” to “ensur[e] private addresses are not re-used within that domain.” EX2015 ¶¶47-48. First, I disagree that such precautions require common control by different entities. Dr. Mangione-Smith provides no evidence supporting this assertion. An administrator can take these precautions in the multiple domains that it controls. In other words, taking these so-called precautions does not require different entities to control the different domains. Second, I have reviewed Dr. Mangione-Smith's citations. The disclosures in Dr. Mangione-Smith's citations are unrelated to this argument and do not support his point. For example, Dr. Mangione-Smith cites 2:11-22 of the '844 patent as support. EX2015 ¶48. But this section merely says that “Double NAT or Operator NAT” was created as a solution “[t]o limit the number of IPv4 public addresses necessary to provide an IP connectivity service to an installed base of clients” and then describes how Operator NAT works. EX1001, 2:11-22. It says nothing about requiring “common administration” of a network in order to take “precautions” to ensure that private addresses are not re-used.

5. I also disagree with Dr. Mangione-Smith's opinions about the extrinsic evidence. EX2015 ¶49. I note that the dictionary definition that Dr. Mangione-Smith provides actually de-emphasizes the definition that Dr. Mangione-Smith advocates for. The Newton's Telecom Dictionary definition says, for example, that the definition relied on by Dr. Mangione-Smith is "the formal technical definition," but that it is not what the term means in everyday language. EX2003, 3. "[W]hen the Internet came along, the definition got a little looser." *Id.* "Domain" can therefore refer to "a group of computers masquerading as a single computer" because, "[o]n the Internet, the domain is thus *whatever the address is that gets you there.*" *Id.* (emphasis added). "Some people" even "think of a domain as a group of computers on a network that share something in common." *Id.* 4. Rather than support its construction, this extrinsic evidence actually confirms that "domain" has a broad meaning in the art. And neither the Newton's Telecom Dictionary definition nor the IBM Dictionary of Computing definition requires common control *by different entities*. One of the ways "domain" *is* referred to in the art is as an "autonomous system." *E.g.*, EX1058, 13; EX1059, 1. I note that there is no *a priori* requirement that different autonomous systems be operated by different entities. And, as I will explain in further detail below, Li's Figure 7 teaches autonomous-system domains.

6. I also note that I preempted Dr. Mangione-Smith's construction in my original declaration and identified several reasons why it is incorrect. EX1003 ¶¶90. I have reviewed Dr. Mangione-Smith's declaration, and I see no response to my original arguments. In conclusion, I disagree with Dr. Mangione-Smith's construction of "domain." There is simply no support for the narrow definition that he provides in either the intrinsic or extrinsic evidence.

III. ALL GROUNDS: FIGURE 7 DEPICTS AN INTER-DOMAIN NETWORK

7. I disagree with Dr. Mangione-Smith's assertion that Figure 7 of Li depicts an *intra*-domain environment rather than an *inter*-domain environment. See EX2015 ¶¶64-76. Li's Figure 7 teaches the correct construction of "domain," as used in the patent and as it was commonly understood in the art.

8. For one, Figure 7 of Li is in a section that explicitly covers inter-domain environments. EX1037, 6-9. Li explains "for an inter-domain routing environment, [Border Gateway Protocol] provides useful information on the origin of any network prefix." EX1037, 6 (citation omitted). Border Gateway Protocol is explicitly designed for inter-domain routing. EX1061, 1 ("This document, together with its companion document, 'Application of the Border Gateway Protocol in the Internet', define an inter-autonomous system routing protocol for the Internet."); EX1061, 2 ("BGP-4 provides a new set of mechanisms for supporting classless interdomain routing."). This allows Li to be extended to cover two new models, including one

containing “stub autonomous systems [] connected to an IPv6 backbone.” EX1037,

6. That stub-autonomous model is depicted in Figure 7, complete with using BGP to discover network information. EX1037, 6-7.

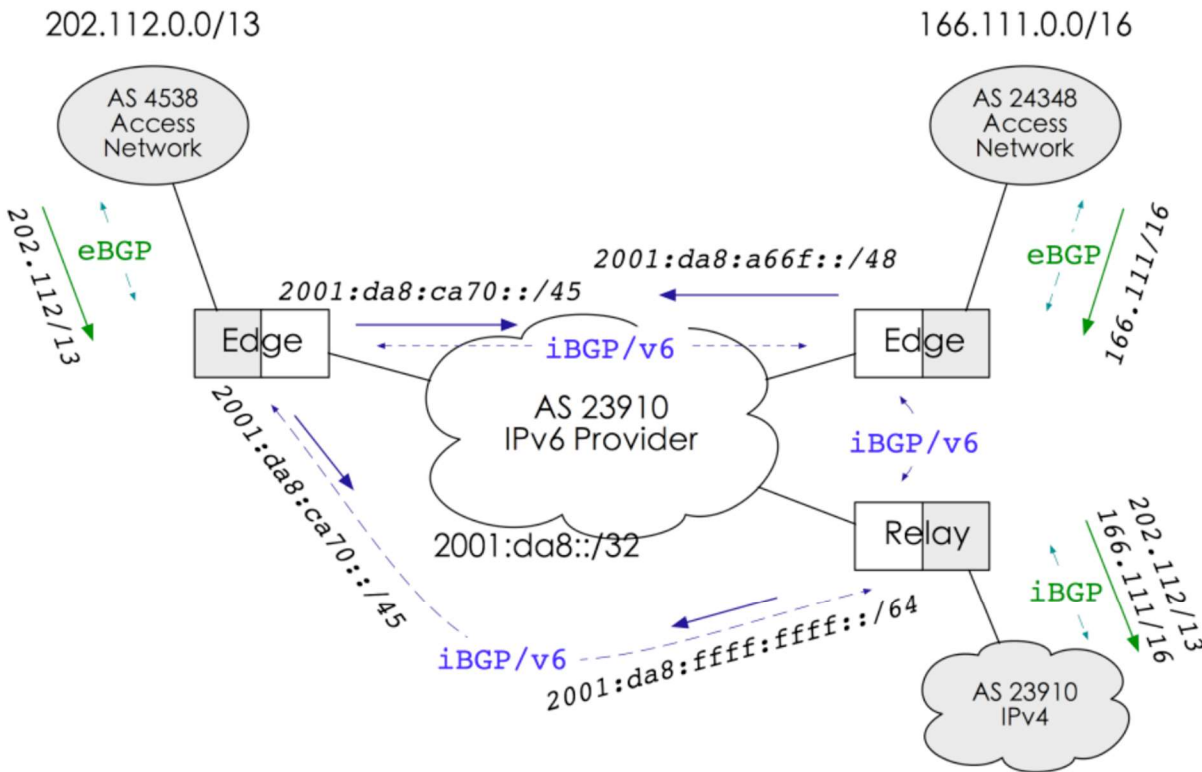


Figure 7. IPv4-over-IPv6 transition for access networks

EX1037, 7.

9. Further, Figure 7 is not identical to Figure 1, as Dr. Mangione-Smith suggests. EX2015 ¶74. The IPv4 networks in Figure 1 are subnets, part of the larger intra-domain system.

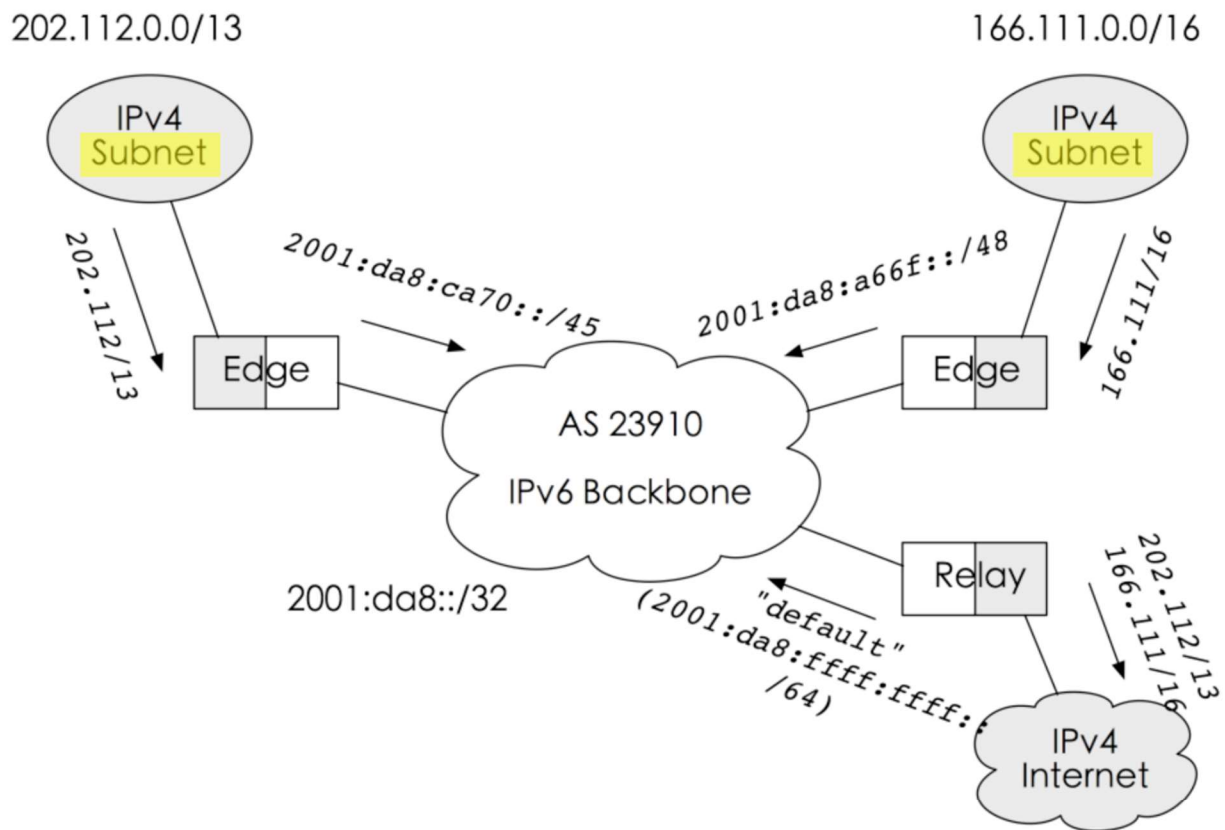


Figure 1. Intra-domain IPv4-over-IPv6 transition

EX1037, 3 (highlighted). Thus, “[t]his rule of addressing is deployed without difficulty because the address assignment is fully controlled by the autonomous system”—*i.e.*, the IPv6 backbone—“itself, ***which is aware of all the IPv4 prefixes under its administration.***” *Id.* (emphasis added). Because “the IPv6 aggregation of the autonomous domain may contain ***all the IPv4 blocks belonging to this domain***” it can communicate routing information across the entire network using the Interior Gateway Protocol (“IGP”)—a protocol used to exchange information within a

single, self-contained domain. *Id.* (emphasis added); *see* EX1060, 1 (explaining that IGP is an intra-domain protocol). The IPv4 networks in Figure 7, on the other hand, are *autonomous* systems that must communicate routing information via the Border Gateway Protocol (“BGP”)—a protocol used to exchange routing information between independent domains. EX1037, 6.

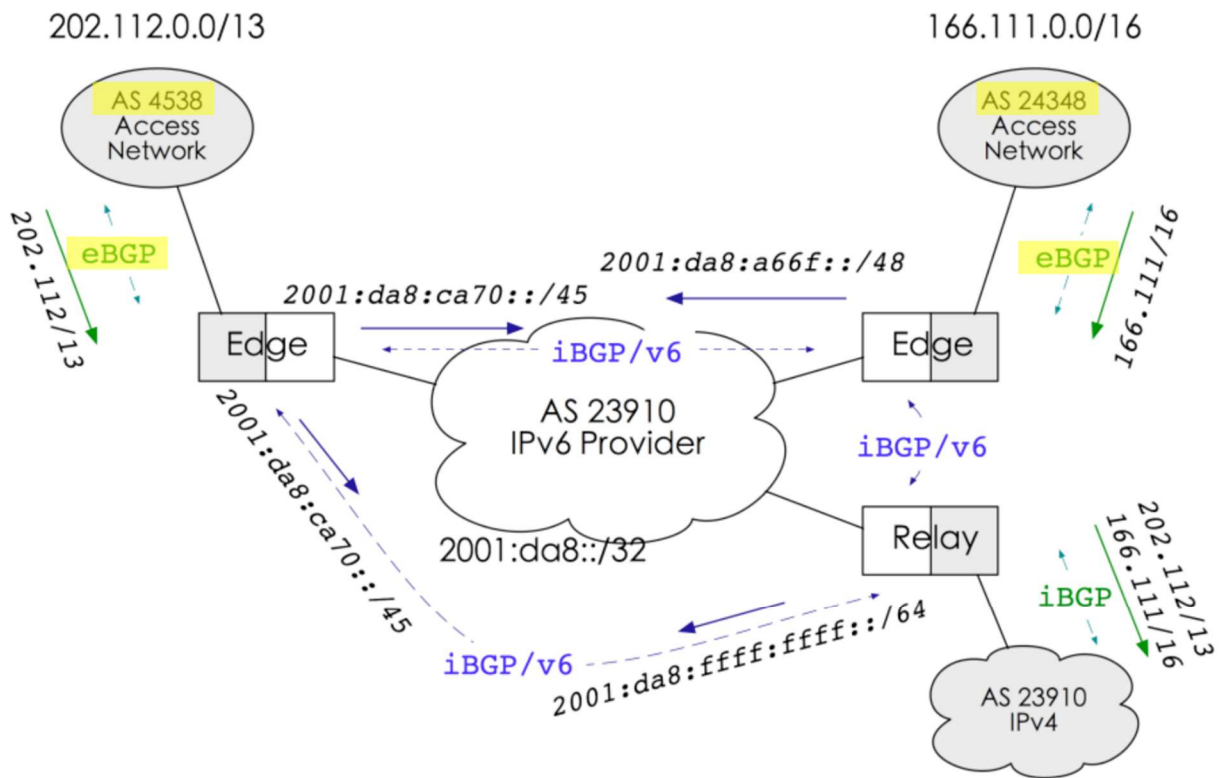


Figure 7. IPv4-over-IPv6 transition for access networks

Id. at 7 (highlighted).

10. A POSITA would understand that autonomous systems are “domains.” “The collection of routers, computers, and other components within a single administrative domain is known as an autonomous system (AS).” EX1058, 13.

“Internet routing is implemented using a distributed system composed of many routers, grouped into administrative domains called Autonomous Systems (ASes).” EX1059, 1. “There are two types of routing protocols”: (1) “IGPs (Internal Gateway Protocols)” that are “also called *intra*-domain protocols” and (2) “EGPs (External Gateway Protocols)” that are “*inter*-domain protocols...responsible for the exchange of reachability information between different ASes.” EX1060, 1. “The only inter-domain protocol currently employed on the Internet is the Border Gateway Protocol (BGP).” EX1060, 1. “BGP is in fact the primary interdomain routing protocol.” EX1058, 13. Thus, “[r]outing information is exchanged between ASes using Border Gateway Protocol.” EX1059, 1. Simply, “BGP-4 provides a new set of mechanisms for supporting classless interdomain routing.” EX1061, 2. In sum, I disagree with Dr. Mangione-Smith; Figure 7 of Li is an inter-domain environment because Li identifies it as an inter-domain environment and Figure 7 uses protocols (BGP) and network types (autonomous systems) that are specific to inter-domain environments.

IV. ALL GROUNDS: FIGURES 8, 10, AND 11 ALSO MAP TO THE '844 CLAIMS

11. I understand that Patent Owner and Dr. Mangione-Smith concede that other figures in Li—Figures 8, 10, and 11 (“Admitted Inter-Domain Figures”)—depict inter-domain environments. EX2015 ¶¶71-72. I agree with Dr. Mangione-Smith. Although Figure 7 teaches all claim elements, as I discussed in my previous

declaration, the Admitted Inter-Domain Figures also teach the '844 claims. There is no material difference between Figure 7 and the Admitted Inter-Domain Figures that prevents mapping the '844 claims to the Admitted Inter-Domain Figures in the same manner as disclosed in my prior declaration. *E.g.*, EX1003 ¶¶86-162 (discussing the combination of Li and Li-2 in ground 1). In my prior declaration, I explained that Figure 7 teaches the '844 claims in the following example, where a host in an IPv4 domain sends an IPv4 data packet through an IPv6 domain to a host in another IPv4 domain:

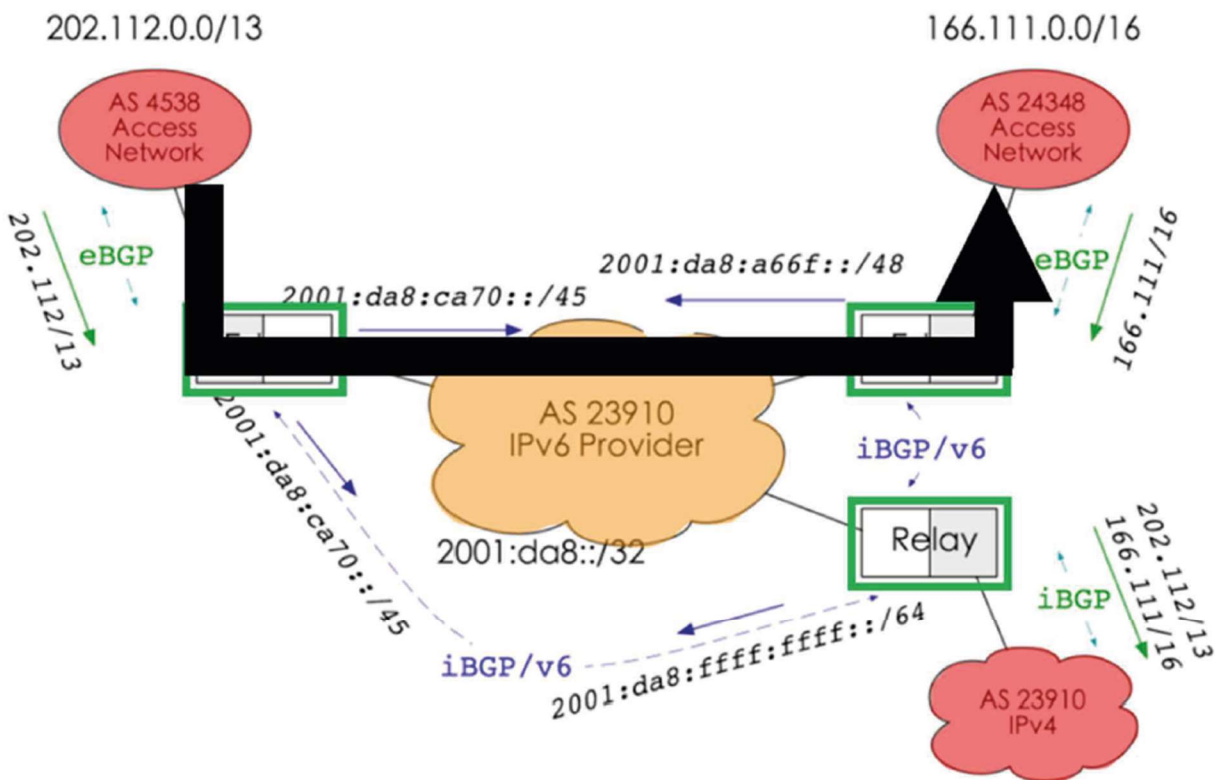


Figure 7. IPv4-over-IPv6 transition for access networks

Id. at ¶62 (citing EX1037, 7 (highlighting and annotations added)). The Admitted Inter-Domain Figures do the same thing. For example, Figure 8 also discloses a host in an IPv4 domain sending an IPv4 data packet through an IPv6 domain to a host in another IPv4 domain:

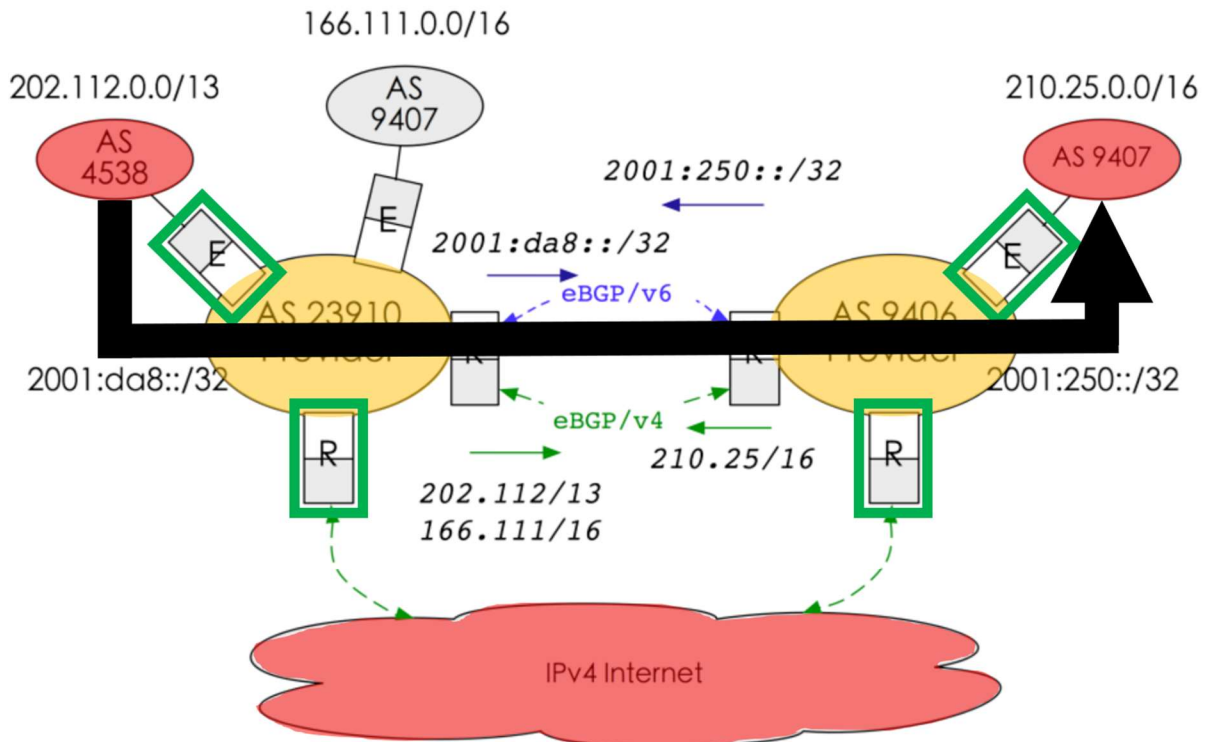
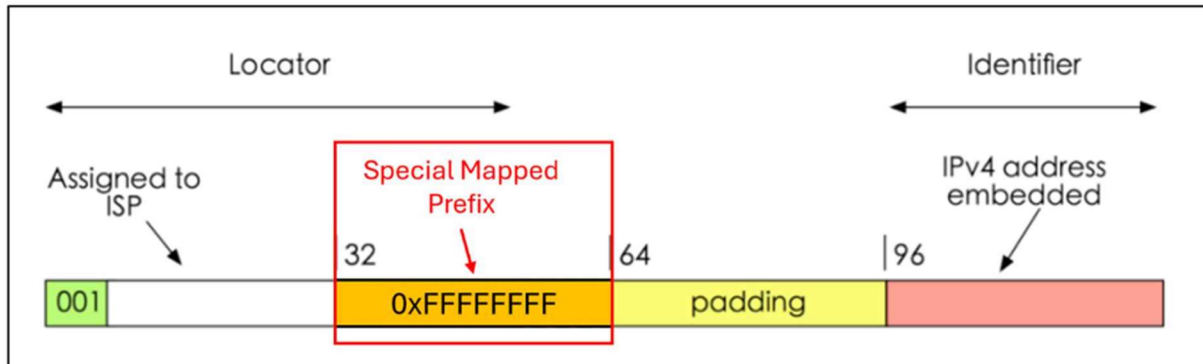


Figure 8. Transition in inter-domain environment

EX1037, 8 (highlighting and annotations added). Alternatively, Li discloses routing the data packet in the Admitted Inter-Domain Figures through the IPv4 Internet or to a host in the IPv4 Internet. EX1037, 8 (“Note that, in this new model, ends in IPv4 internet would treat the IPv6 backbone as an alterative option of routing their packets.”).

12. The actual operation of the Li system does not change between Figure 7 and the Admitted Inter-Domain Figures. In fact, Dr. Mangione-Smith and Patent Owner concede that the constructed IPv6 address in these examples still contains (1) an IPv6 prefix, (2) the IPv4 destination address, and (3) zero padding.



EX2015 ¶54 (citing EX1037, 3 (annotations and highlighting added)); POR 13-14.

13. Neither Dr. Mangione-Smith nor Patent Owner suggest that the Admitted Inter-Domain Figures somehow operate differently than Figure 7 in Li. That is because they do not; all figures are inter-domain environments and operate—as relevant to the '844 claims—in the same manner. Had either Dr. Mangione-Smith or Patent Owner identified a difference between Figure 7 and the Admitted Inter-Domain Figures, I would have addressed it in this declaration, and I reserve the right to address any purported differences that Dr. Mangione-Smith or Patent Owner might attempt to identify in sur-reply.

V. GROUND 2: THE SIMULTANEOUS INVENTIONS DISCLOSE ALL OF THE '844 CLAIM'S LIMITATIONS

14. In my original declaration, I explained that multiple groups disclosed the idea of forming an IPv6 address using an operator prefix, destination IPv4 address, and destination port number. EX1003 ¶¶44-48, 170. I understand that Dr. Mangione-Smith asserts that certain limitations in claims 1 and 3 are not taught in the simultaneous-invention references. EX2015 ¶¶91-96. I have been instructed by counsel to perform a limitation-by-limitation analysis of whether the simultaneous-invention references teach the entire claims, in order to respond to Patent Owner's assertions that the references do not teach every aspect of the claims. Based on my analysis, the simultaneous-invention references disclose all of the '844 patent's claim limitations.

A. The IVI References Discloses All Limitations of the '844 Claims

15. Li-2 (EX1038), the IVI Memo (EX1027)—which is substantively identical to Li-2—and the IVI Presentation (EX1028) (collectively, the “IVI References”) describe an independent invention by Xing Li and his co-authors that, considered either individually or altogether, discloses all claim limitations.

1. Claim 1

- (a) 1[pre]: A method of receiving a data packet from an IPv4 domain in an IPv6 domain, said data packet comprising

an IPv4 destination address and a destination port number, said method comprising the following steps:

16. To the extent the preamble is limiting, the IVI References disclose limitation 1[pre]: “A method of receiving a data packet from an IPv4 domain [*e.g.*, **a first ISP's IPv4 network**] in an IPv6 domain [*e.g.*, **a second ISP's IPv6 network**], said data packet comprising an IPv4 destination address and a destination port number, said method comprising the following steps.”

17. As the IVI References explain, “[t]he experiences for the IPv6 deployment in the past 10 years strongly indicate that for a successful transition, the IPv6 hosts need to communicate with the global IPv4 networks.” EX1027, 4; EX1028, 3; EX1038, 4. To address this issue, and others, the IVI References disclose “special IPv6 address blocks based on *bi-directional* borrowing scheme (IVI).” EX1038, 4 (emphasis added); *see also* EX1027, 7 (“The process in both directions are symmetric.”); EX1028, 4 (“The IVI is a prefix-specific and explicit bidirectional address mapping scheme.”). The term “IVI” can be broken down as follows: “IV means 4 and VI means 6 in Roman representation, so IVI means bi-directional borrowing, mapping and translation between IPv4 and IPv6.” EX1038, 5; *see also* EX1027, 6 (“IVI: IV means 4 and VI means 6 in Roman representation, so IVI means mapping and translation between IPv4 and IPv6.”). Li-2 therefore teaches an “IPv4 to IPv6 transition scheme” that “can communicate with the global IPv6 networks directly and with the global IPv4 networks.” EX1038, 2, 4; *see also*

EX1027, 7 (“In addition, the special IPv6 addresses can communicate with the global IPv6 networks.”). The IVI References provide a generic portrayal of an exemplary network in Figure 2 of Li-2, Figure 3 of the IVI Memo, and in the IVI Presentation:

```
(Global )  ---- IPv4-2          ----- IPv6-2 ---- (Global )
(IPv4    )--|R1|-----|IVI gateway|-----|R2|--- (IPv6    )
(network)  ---- IPv4-1          ----- IPv6-1 ---- (network)
```

Figure 2

EX1038, 7.

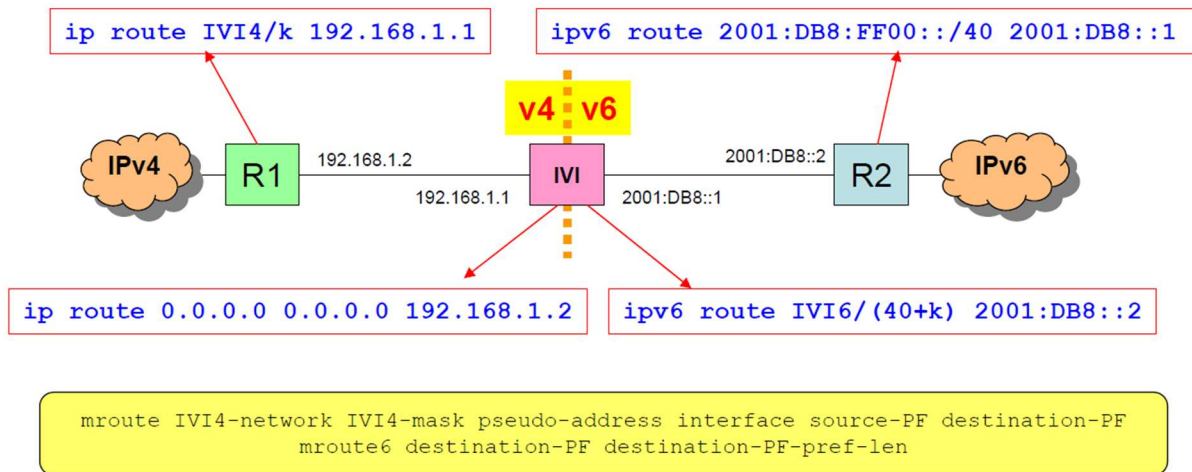
IVI Routing

```
/-----\
(Global )  ----192.168.1.2  ----- 2001:DB8::2---- (Global )
(IPv4    )--|R1|-----|IVI gateway|-----|R2|--- (IPv6    )
(network)  ---- 192.168.1.1-----2001:DB8::1 ---- (network)
\-----/
\-----/
```

Figure 3

EX1027, 9.

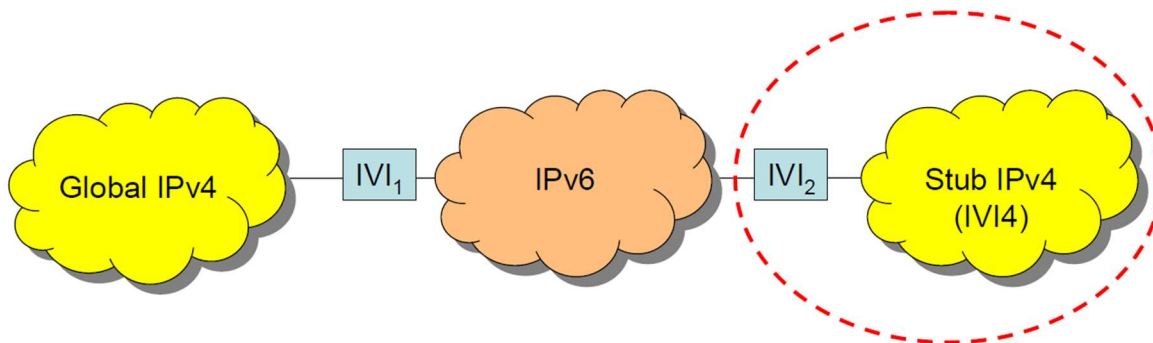
Routing and mapping configuration example



EX1028, 9. In these examples, the IVI gateway is situated between an IPv4 domain and an IPv6 domain and “connects to both IPv4 and IPv6 networks.” EX1027, 7; *see also* EX1028, 9. “IVI can be implemented independently in each ISP’s network”; meaning, these domains can belong to different ISPs. EX1027, 10; EX1038, 8. For example, the IVI References describe a scenario in which a host within a first ISP, ISP(1), communicates with a host within a second ISP, ISP(2). EX1027, 11-12 (Scenario 2); EX1028, 12; EX1038, 8-9 (Scenario 2). The IVI References also explain that “[t]he IVI scheme can support the IPv4 over IPv6 service (NAT64), i.e. a stub IPv4 network can be connected to an IVI gateway to reach the IPv6 network and via another IVI gateway to reach the global IPv4 network.” EX1027, 16; EX1028, 18; EX1038, 12. A data packet in this network

starts in a first IPv4 domain, traverses an intermediary IPv6 domain, and ends in a second IPv4 domain:

IVI Deployment Scenarios (2)



EX1028, 25.

18. I break down each element of the preamble below within the context of the IVI References' teachings.

19. *A method of receiving a data packet from an IPv4 domain in an IPv6 domain:* The IVI References disclose receiving a data packet from an IPv4 domain. The IVI References disclose a “reachability matrix” that shows IVI can receive data packets from IPv4 domains.

The address reachability matrix of the IPv4, IVI and IPv6 is shown in the following figure.

IVI reachability Matrix

		IPG4	IVI	IPG6

IPG4		OK	OK	NO
IVI		OK	OK	OK
IPG6		NO	OK	OK

Figure 4

EX1027, 10; EX1028, 10; EX1038, 8. Similarly, Figure 2 of Li-2 and Figure 3 of the IVI Memo show that the IVI gateway is capable of receiving data packets from a global IPv4 network.

IVI Routing

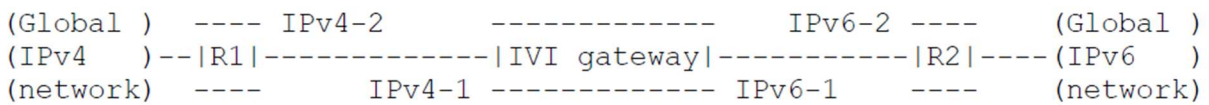


Figure 2

EX1038, 7.

IVI Routing

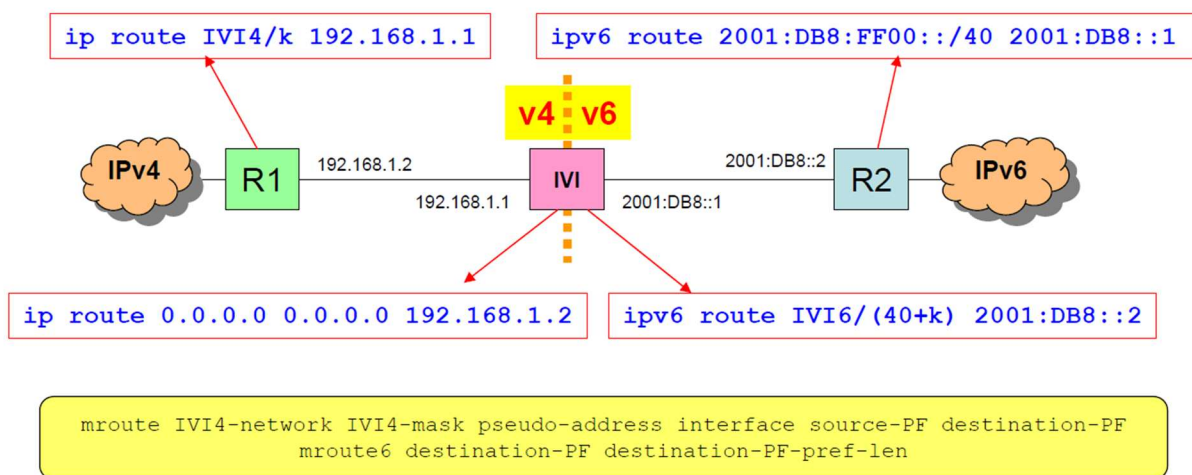
```

/-----\
(Global )  ----192.168.1.2  ----- 2001:DB8::2---- (Global )
(IPv4   )--|R1|-----|IVI gateway|-----|R2|---(IPv6   )
(network)  ---- 192.168.1.1-----2001:DB8::1 ---- (network)
\-----/
  
```

Figure 3

EX1027, 9.

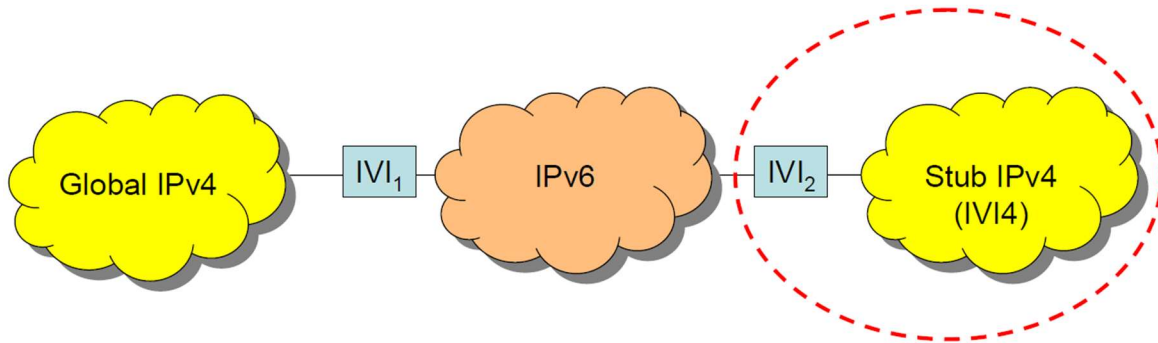
Routing and mapping configuration example



EX1028, 9. The IVI References' "stub" example also involves IVI gateways that receive and transmit data packets between IPv4 domains across an IPv6 domain.

EX1027, 16; EX1028, 18; EX1038, 12.

IVI Deployment Scenarios (2)



EX1028, 25. A POSITA would know that IVI gateways sit in an IPv6 domain; otherwise, they would not be able to receive or send IPv6 data packets. *See* EX1027, 10 (“IVI gateway has IPv6 route of IVI6(i)/(40+k) with next hop equals to 2001:DB8::2.”); EX1027, 10 (confirming in Figure 4 that IVI can communicate with IPv6); EX1038, 7 (“IVI gateway has IPv6 route of IVI6(i)/(40+k) with next hop equals to IPv6-2.”); EX1038, 8 (confirming in Figure 3 that IVI can communicate with IPv6).

20. *said data packet comprising an IPv4 destination address and a destination port number, said method comprising the following steps:* The received data packet contains an IPv4 address. The IVI References explain that because IPv4 and IPv6 “are different protocols with different network layer header format,” the system must translate the IPv4 information into IPv6 information.

EX1027, 14; EX1038, 10. This includes translating the destination IPv4 address into an IPv6 address based on the IVI mapping (which I will discuss later):

IPv4 to IPv6 Header translation based on IVI scheme

IPv4 Field	Translated to IPv6
Version (0x4)	Version (0x6)
IHL	discarded
Type of Service	discarded
Total Length	Payload Length = Total Length - IHL * 4
Identification	discarded
Flags	discarded
Offset	discarded
Time to Live	Hop Limit
Protocol	Next Header
Header Checksum	discarded
Source Address	IVI address mapping
Destination Address	IVI address mapping
Options	discarded

Figure 8

EX1027, 14; EX1038, 10.

21. The received data packet also contains a TCP or UDP header, which a POSITA would know includes a port number. EX1027, 15; EX1038, 11. The IVI References explain that IVI “MUST” support TCP and UDP. EX1027, 23; EX1028, 32; EX1038, 18. As I will discuss later, the IVI References utilize this port number in the constructed IPv6 address to prevent IPv4 address exhaustion.

- (b) 1[a]: constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number;

22. The IVI References disclose “constructing an IPv6 destination address [e.g., **an IVI address**] by concatenating an operator prefix [e.g., **an IPv6 prefix**], said IPv4 destination address, and the destination port number.”

23. The IVI References teach “borrow[ing] a subset of ISP’s global routable IPv4 address[es]” and “map[ping] it into IPv6.” EX1038, 6; *see also* EX1027, 7 (“Each provider can borrow a portion of its IPv4 addresses and maps them into IPv6 based on the above mapping rule.”). The IVI References’ default mapping scheme comprises an IPv6 prefix, IPv4 address, and then zero padding—just like in Li. EX1038, 6.

IVI Address Mapping

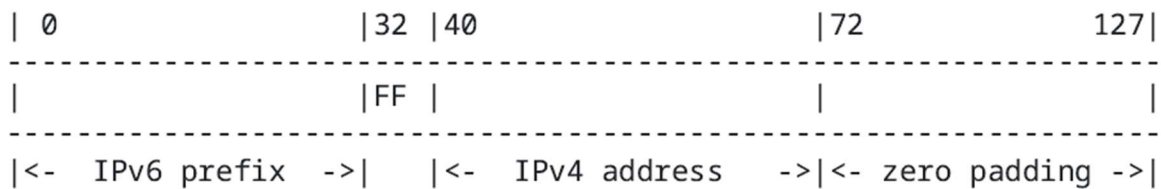
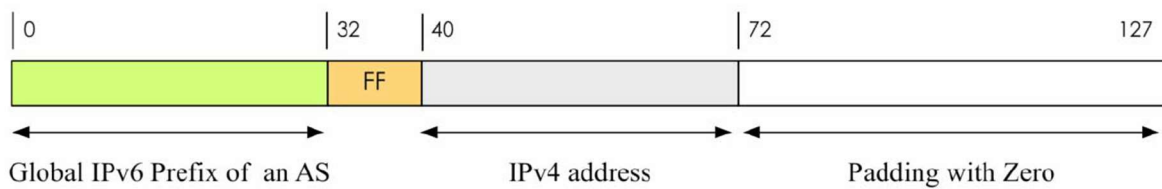
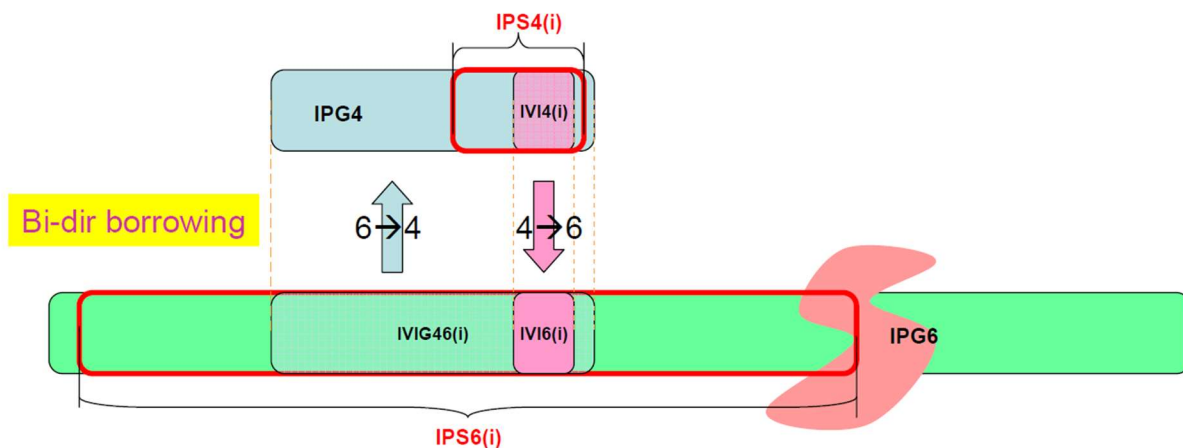


Figure 1

EX1027, 8; EX1038, 6.



EX1028, 6. Bit numbers 0 to 31 “are the prefix of ISP(i)’s /32.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. This IPv6 prefix corresponds to the ’844 patent’s “operator prefix.” Bit numbers 32 to 39 “are all one’s as the identifier of IVI.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. Bit numbers 40 to 71 “are the global IPv4 space (IPG4) presented in hexadecimal format.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. This IPv4 address corresponds to the ’844 patent’s “said IPv4 destination address.” And the remaining bits are placeholder binary 0s, known as “zero padding.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. This mapping allows IPv4 addresses to be converted into IPv6 addresses and vice versa:



EX1028, 7.

24. The IVI References then extend this default mapping by replacing the unused zero padding with a port number. EX1027, 18-19; EX1038, 14-15. The IVI References call this “port multiplexing.” EX1027, 18; EX1028, 20-22; EX1038, 14. The IVI References teach port multiplexing “[s]ince public-IPv4 address is a scarce resource” and “the effective use of the IPv4 address is important for the IVI scheme.” EX1027, 18; EX1038, 14. In other words, simply allowing IPv4 hosts to communicate with IPv6 hosts does not solve the IPv4 address exhaustion problem. Thus, the IVI References teach port multiplexing to allow multiple IPv4 hosts to share a single IPv4 global address.

25. Port multiplexing works by replacing “the least significant bits in the IVI6(i)” address—*i.e.*, the unused zero padding—with the destination port number. EX1027, 18; EX1038, 14. “This is to say that a single IPv4 address (IVI4(i)) can be used for multiple IVI6(i) addresses.” EX1027, 18; EX1038, 14. For example, the IVI References teach “defin[ing] ‘pseudo-well-known port number’, which is unique for IVI4(i)” —*i.e.*, each IPv4 host—“and known to the peers.” EX1027, 19; 1038, 15. “By combining address and pseudo-well-known port number” into an IPv6 address, “the feature of end-to-end address transparency can still be maintained.” EX1027, 19; EX1038, 15. The IVI References provide examples of such port multiplexing:

```
202.38.108.2#81 --> IVI61=2001:250:ffca:266c:0200::81#81
202.38.108.2#82 --> IVI61=2001:250:ffca:266c:0200::82#82
202.38.108.2#83 --> IVI61=2001:250:ffca:266c:0200::83#83
202.38.108.2#84 --> IVI61=2001:250:ffca:266c:0200::84#84
```

EX1028, 22 (highlighting added). As I explained in my prior declaration, the IPv6 prefix is the first box, highlighted in blue. EX1003 ¶74. The second box, highlighted in red, is a flag that specifies that the IPv6 address has been constructed according to IVI. *Id.* The third box, which highlights the next 32 bits in purple, is the IPv4 address from the left side, translated from decimal to hexadecimal. *Id.* The fourth box, highlighted in yellow, is zero padding. *Id.* And the fifth box, highlighted in gray, is the port number that has been taken from the left side of the translation and concatenated to the end of the constructed IPv6 address, replacing unused zero padding as appropriate. *Id.* As I noted in my prior declaration, the authors did not translate, perhaps for the sake of convenience, the port number from decimal (*i.e.*, “81”) to hexadecimal (*i.e.*, “51”). *Id.* ¶75. However, a POSITA would have understood that the port number extracted from the IPv4 data packet on the left-hand side of the equation is the exact same 16-bit number that is concatenated to form the IPv6 address on the right-hand side of the equation. *Id.*

26. Thus, “[b]ased on multiplexing techniques, the global IPv4 addresses can be used effectively.” EX1027, 25; EX1038, 20. With proper port multiplexing,

IVI can support “2.6 billion [IPv6] addresses, possibly enough for the unwired population in the world.” EX1027, 25; EX1028, 35; EX1038, 20.

- (c) 1[b]: generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet; and

27. The IVI References disclose limitation 1[b]: “generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet.”

28. The IVI References explain that “[t]he packets generated from the global IPv4 addresses and sent to the special IPv6 addresses are routed to the IPv4 interface of the IVI gateway via the IPv4 routing protocol and the packets generated from the special IPv6 addresses and sent to the global IPv4 addresses are routed to the IPv6 interface of the IVI gateway via the IPv6 routing protocol.” EX1027, 7. In other words, the received IPv4 packet must be “translated by the IVI gateway” into an IPv6 packet. EX1027, 15; EX1038, 11. Because IPv4 and IPv6 “are different protocols with different network layer header format,” the system must translate the IPv4 information into IPv6 information. EX1027, 14; EX1038, 10. This includes translating the destination IPv4 address into an IPv6 address based on the IVI mapping:

IPv4 to IPv6 Header translation based on IVI scheme

IPv4 Field	Translated to IPv6
Version (0x4)	Version (0x6)
IHL	discarded
Type of Service	discarded
Total Length	Payload Length = Total Length - IHL * 4
Identification	discarded
Flags	discarded
Offset	discarded
Time to Live	Hop Limit
Protocol	Next Header
Header Checksum	discarded
Source Address	IVI address mapping
Destination Address	IVI address mapping
Options	discarded

Figure 8

EX1027, 14; EX1038, 10.

- (d) 1[c]: routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address, said constructed address belonging to a range of IPv6 addresses routable to an interconnection equipment of the IPv6 domain with the IPv4 destination address

29. The IVI References disclose limitation 1[c]: “routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address [e.g., **IVI address**], said constructed address belonging to a range of IPv6 addresses [e.g., **range of IVI address corresponding to devices in the IPv4 domain**] routable to an interconnection equipment [e.g., **Router R2 in the IPv6 domain**] of the IPv6 domain [e.g., **the second ISP's IPv6 domain**] with the IPv4 destination address.”

30. The IVI References explain that “[t]he packets generated from the global IPv4 addresses and sent to the special IPv6 addresses are routed to the IPv4 interface of the IVI gateway via the IPv4 routing protocol and the packets generated from the special IPv6 addresses and sent to the global IPv4 addresses are routed to the IPv6 interface of the IVI gateway via the IPv6 routing protocol.” EX1027, 7. The IVI References further teach that its gateway can send the modified data packet to a second ISP’s IPv6 domain. EX1027, 10-12; EX1038, 7-9. In this way, “a pure IPv6 host A’ can communicate with B, where B is a host with arbitrary IPv4 address.” EX1038, 8. Indeed, “the routing is straightforward” to a POSITA, so little explanation is needed on how to route data packets. EX1027, 9; EX1038, 7.

2. Claim 3

- (a) 3[pre]: A method of sending a data packet from an IPv6 domain to a destination address of an IPv4 domain, 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, said method executing the following steps on reception of the data packet:

31. To the extent the preamble is limiting, the IVI References disclose limitation 3[pre]: “A method of sending a data packet from an IPv6 domain [*e.g.*, **the second ISP’s IPv6 domain**] to a destination address of an IPv4 domain [*e.g.*, **a host in the first ISP’s IPv4 domain**], said packet comprising an IPv6 destination address [*e.g.*, **an IVI address**] of the packet constructed by concatenating an

operator prefix [*e.g.*, **an IPv6 prefix**], the IPv4 destination address, and a destination port number, said method executing the following steps on reception of the data packet.”

32. The IVI References disclose that the IVI gateway, sitting in the IPv6 domain, can send a data packet to an IPv4 domain. This exact configuration is shown in Figure 2 of Li-2, Figure 3 of the IVI Memo, and in the IVI Presentation:

```
(Global )  ---- IPv4-2          ----- IPv6-2 ---- (Global )
(IPv4   ) --|R1|-----|IVI gateway|-----|R2|---- (IPv6   )
(network) ----          IPv4-1 ----- IPv6-1 ---- (network)
```

Figure 2

EX1038, 7.

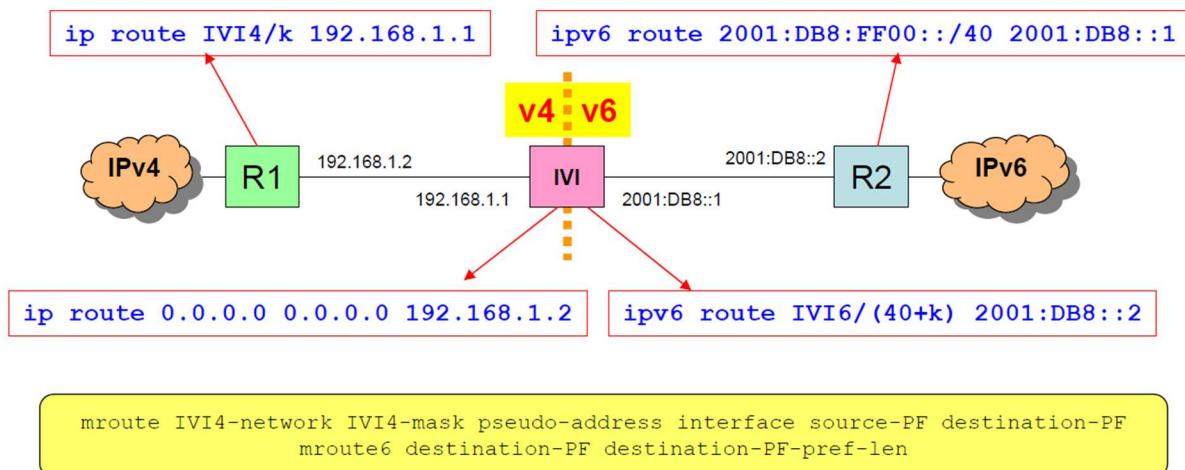
IVI Routing

```
/-----\
(Global )  ----192.168.1.2  ----- 2001:DB8::2---- (Global )
(IPv4   ) --|R1|-----|IVI gateway|-----|R2|---- (IPv6   )
(network) ---- 192.168.1.1-----2001:DB8::1 ---- (network)
\-----/
\-----/
```

Figure 3

EX1027, 9.

Routing and mapping configuration example



EX1028, 9. The IVI References also disclose a “reachability matrix” that shows IVI can communicate (*i.e.*, send and receive data packets) with IPv6 domains:

The address reachability matrix of the IPv4, IVI and IPv6 is shown in the following figure.

IVI reachability Matrix

	IPG4	IVI	IPG6
IPG4	OK	OK	NO
IVI	OK	OK	OK
IPG6	NO	OK	OK

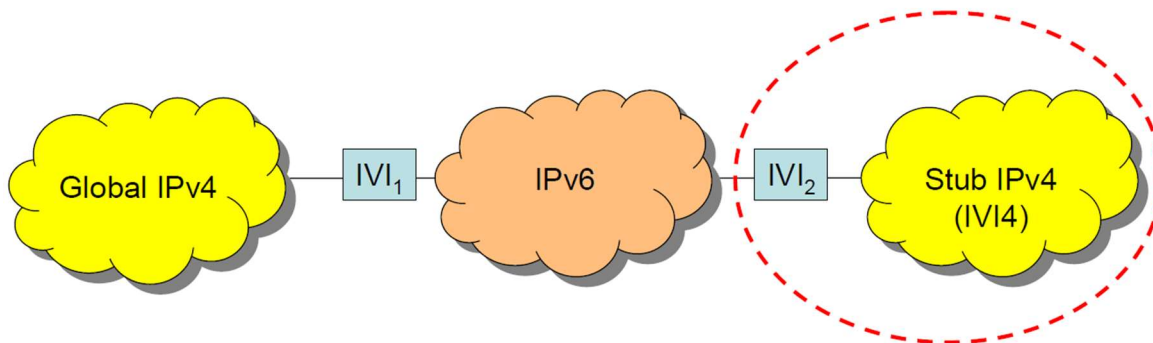
Figure 4

EX1027, 10; EX1028, 10; EX1038, 8. The IVI References’ “stub” example also involves IVI gateways that receive and transmit data packets in an intermediary IPv6

domain before then transmitting them to their destination in an IPv4 domain.

EX1027, 16; EX1028, 18; EX1038, 12.

IVI Deployment Scenarios (2)



EX1028, 25.

33. The IVI References disclose that packets received from the IPv6 domain contain an IVI address that comprises an operator prefix (*e.g.*, an IPv6 prefix), an IPv4 destination address, and optionally a destination port number. EX1027, 8, 18; EX1038, 6, 14. In this way “a pure IPv6 host A’ can communicate with B, where B is a host with arbitrary IPv4 address.” EX1038, 8; *see also* EX1027, 11-12 (Scenario 2); EX1028, 12.

34. Specifically, the IVI References teach “borrow[ing] a subset of ISP’s global routable IPv4 address[es]” and “map[ping] it into IPv6.” EX1038, 6; *see also* EX1027, 7 (“Each provider can borrow a portion of its IPv4 addresses and maps them into IPv6 based on the above mapping rule.”). The IVI References’ default

mapping scheme comprises an IPv6 prefix, IPv4 address, and then zero padding—just like in Li. EX1038, 6.

IVI Address Mapping

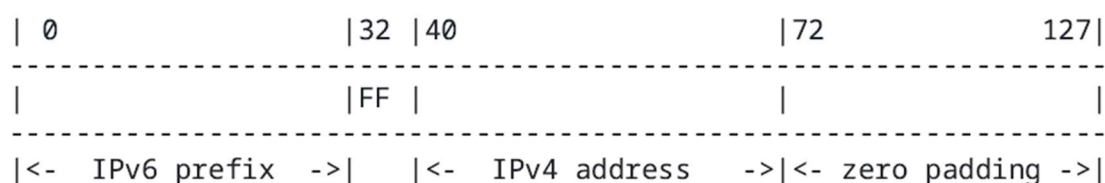
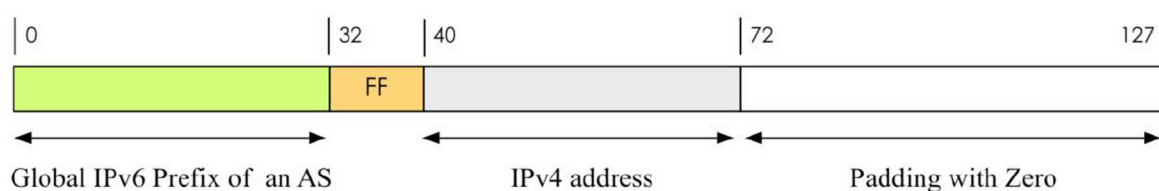


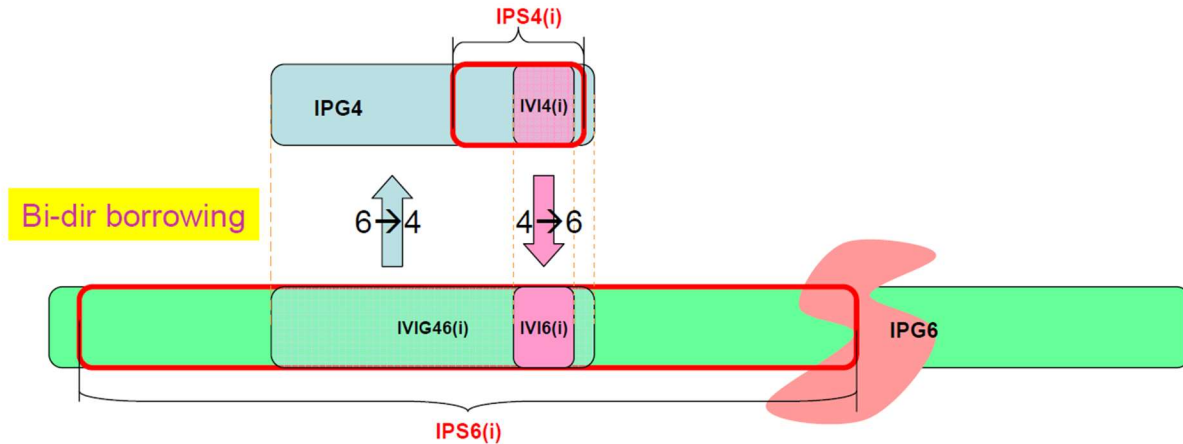
Figure 1

EX1027, 8; EX1038, 6.



EX1028, 6. Bit numbers 0 to 31 “are the prefix of ISP(i)’s /32.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. This IPv6 prefix corresponds to the ’844 patent’s “operator prefix.” Bit numbers 32 to 39 “are all one’s as the identifier of IVI.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. Bit numbers 40 to 71 “are the global IPv4 space (IPG4) presented in hexadecimal format.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. This IPv4 address corresponds to the ’844 patent’s “said IPv4 destination address.” And the remaining bits are placeholder binary 0s, known as

“zero padding.” EX1027, 8; EX1038, 6; *see also* EX1028, 6. This mapping allows IPv4 addresses to be converted into IPv6 addresses and vice versa:



EX1028, 7.

35. The IVI References then extend this default mapping by replacing the unused zero padding with a port number, called “port multiplexing.” EX1027, 18-19; EX1028, 20-22; EX1038, 14-15. The IVI References disclose port multiplexing “[s]ince public-IPv4 address is a scarce resource” and “the effective use of the IPv4 address is important for the IVI scheme.” EX1027, 18; EX1038, 14. In other words, simply allowing IPv4 hosts to communicate with IPv6 hosts does not solve the IPv4 address exhaustion problem. Thus, Li-2 teaches port multiplexing to allow multiple IPv4 hosts to share a single IPv4 global address.

36. Port multiplexing works by replacing “the least significant bits in the **IVI6(i)**” address—*i.e.*, the unused zero padding—with the destination port number. EX1027, 18; EX1028, 20-22; EX1038, 14. “This is to say that a single IPv4 address

(IVI4(i)) can be used for multiple IVI6(i) addresses.” EX1027, 18; EX1038, 14. Thus, “[b]ased on multiplexing techniques, the global IPv4 addresses can be used effectively.” EX1027, 25; EX1038, 20. With proper port multiplexing, IVI can support “2.6 billion [IPv6] addresses, possibly enough for the unwired population in the world.” EX1027, 25; EX1028, 35; EX1038, 20.

(b) 3[a]: extracting the IPv4 destination address and the destination port number from the IPv6 destination address;

37. The IVI References disclose limitation 3[a]: “extracting the IPv4 destination address and the destination port number from the IPv6 destination address.” As the IVI References explain, IVI works by “borrow[ing] a subset of [an] ISP’s global routable IPv6 addresses” and “map[ping] it into IPv4.” EX1038, 6; *see also* EX1027, 7 (“Each provider can borrow a portion of its IPv4 addresses and maps them into IPv6 based on the above mapping rule.”). This mapping includes fitting an IPv4 address into the IPv6 address:

IVI Address Mapping

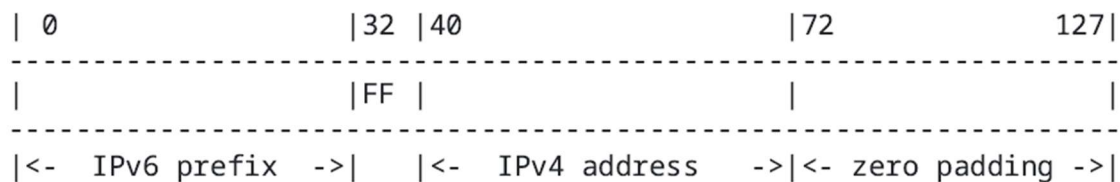


Figure 1

EX1027, 8; EX1038, 6. In other words, IVI maps a global IPv4 address—which is “virtually used by global IPv4 hosts” and can no longer “be used by IPv6 hosts”—into a global IPv6 address. EX1027, 8; EX1038, 6. Accordingly, when an IVI gateway receives a data packet containing an IPv6 address, it can extract the IPv4 address based on the IVI address mapping:

IPv6 to IPv4 Header translation based on IVI scheme

IPv6 Field	Translated to IPv4 Header
Version (0x6)	Version (0x4)
Traffic Class	discarded
Flow Label	discarded
Payload Length	Total Length = Payload Length + 20
Next Header	Protocol
Hop Limit	TTL
Source Address	IVI address mapping
Destination Address	IVI address mapping
-	IHL = 5
-	Header Checksum recalculated

Figure 9

EX1027, 15; EX1038, 11. The IVI References also teach extracting the destination port number; as the references recognize, “[s]ince the TCP and UDP headers consist of check sums which include the IP header, the recalculation and updating of the transport-layer headers must be performed.” EX1027, 15 (citations omitted); EX1038, 11.

- (c) 3[b]: generating an IPv4 data packet from the IPv6 data packet, said packet including said IPv4 destination address and said destination port number; and

38. The IVI References disclose limitation 3[b]: “generating an IPv4 data packet from the IPv6 data packet, said packet including said IPv4 destination address and said destination port number.”

39. The IVI References explain that “[t]he packets generated from the global IPv4 addresses and sent to the special IPv6 addresses are routed to the IPv4 interface of the IVI gateway via the IPv4 routing protocol and the packets generated from the special IPv6 addresses and sent to the global IPv4 addresses are routed to the IPv6 interface of the IVI gateway via the IPv6 routing protocol.” EX1027, 7. In other words, the received packet is “translated by the IVI gateway” into an IPv4 packet. EX1027, 15; EX1038, 11. Because IPv4 and IPv6 “are different protocols with different network layer header format,” the system must translate the IPv6 information into IPv4 information. EX1027, 14; EX1038, 10. This includes translating the destination IPv6 address into an IPv4 address based on the IVI mapping:

IPv6 to IPv4 Header translation based on IVI scheme

IPv6 Field	Translated to IPv4 Header
Version (0x6)	Version (0x4)
Traffic Class	discarded
Flow Label	discarded
Payload Length	Total Length = Payload Length + 20
Next Header	Protocol
Hop Limit	TTL
Source Address	IVI address mapping
Destination Address	IVI address mapping
-	IHL = 5
-	Header Checksum recalculated

Figure 9

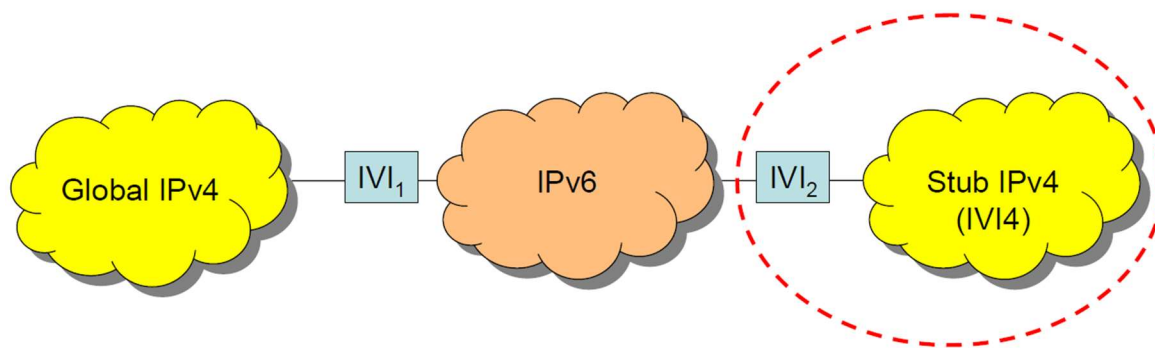
EX1027, 15; EX1038, 11.

(d) 3[c]: routing the IPv4 data packet to its IPv4 destination.

40. The IVI References disclose limitation 3[c]: “routing the IPv4 data packet to its IPv4 destination.” The IVI References explain that “[t]he packets generated from the global IPv4 addresses and sent to the special IPv6 addresses are routed to the IPv4 interface of the IVI gateway via the IPv4 routing protocol and the packets generated from the special IPv6 addresses and sent to the global IPv4 addresses are routed to the IPv6 interface of the IVI gateway via the IPv6 routing protocol.” EX1027, 7. The IVI References further teach that its gateway can send the modified data packet to an IPv4 domain. EX1027, 10-13; EX1038, 7-9. In this way, “a pure IPv6 host A' can communicate with B, where B is a host with arbitrary

IPv4 address.” EX1038, 8; *see also* EX1027, 11-12 (Scenario 2); EX1028, 12. The IVI References also disclose a “stub” example in which the data packet passes through an IPv6 domain to its ultimate destination in an IPv4 domain. EX1027, 16; EX1028, 18; EX1038, 12.

IVI Deployment Scenarios (2)



EX1028, 25. Indeed, “the routing is straightforward” to a POSITA, so little explanation is needed on how to route data packets. EX1027, 9; EX1038, 7.

B. Bush Discloses All Limitations of the '844 Claims

41. The Bush Paper (EX1029) and the Bush Presentation (EX1041) (collectively, “Bush”) describe an independent invention by Randy Bush and his team. These references also, considered either individually and altogether, disclose all claim elements, exemplified in the slide below and discussed in detail in this section:

IPv6 Encap Toward CPE

- BR receives IPv4 packet w/ src/dest

PREAMBLE

- Encapsulates in IPv6 packet

LIMITATION [1B]

- src = WKP+src

- dest = WKP+dest

LIMITATION [1A]

- But note that dest is A+P

- It routes normally within ISP core

LIMITATION [1C]

2008.10.26 A+P

21

EX1041, 21 (annotations added).

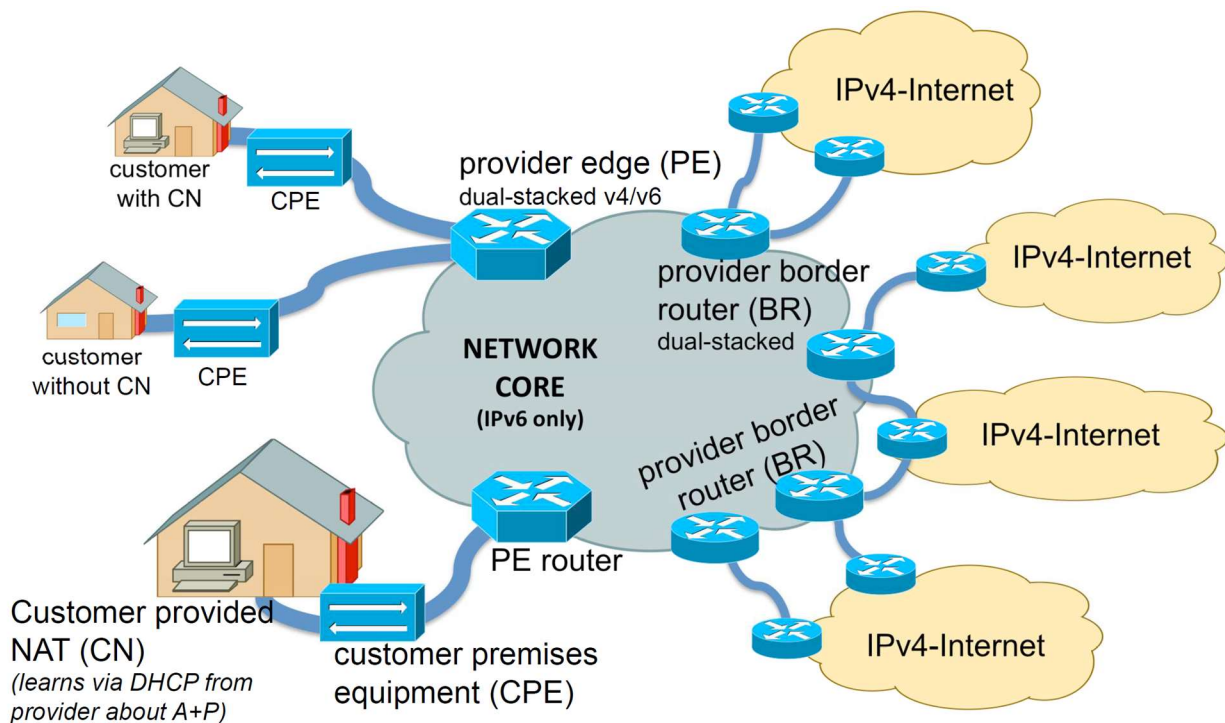
42. In fact, Bush identified the same problem as the '844 patent, (EX1001, 1:61-2:10): IPv4 addresses were running out. EX1029, 1; EX1041, 6. IPv6 promised to solve the problem of IPv4 address exhaustion, but “there [wa]s only marginal IPv6 deployment” at the time. EX1029, 1. Thus “[i]t appear[ed] inevitable that the Internet w[ould] become a polymorphic architecture” such that “many different technologies w[ould] have to co-exist in the future.” EX1029, 1. Just like the '844 patent, Bush thus proposed extending the IPv4 address space by constructing IPv6 addresses comprising the IPv4 address and a port number. EX1029, 2. Thus, Bush is also evidence of simultaneous invention.

1. Claim 1

- (a) 1[pre]: A method of receiving a data packet from an IPv4 domain in an IPv6 domain, said data packet comprising an IPv4 destination address and a destination port number, said method comprising the following steps:

43. To the extent the preamble is limiting, Bush discloses limitation 1[pre]:
“A method of receiving a data packet from an IPv4 domain [*e.g.*, **a customer network or the IPv4 Internet**] in an IPv6 domain [*e.g.*, **the provider core**], said data packet comprising an IPv4 destination address and a destination port number, said method comprising the following steps.”

44. Bush depicts the following exemplary configuration:



EX1029, 2. At the center is an IPv6 network core managed by a provider. EX1029, 2 (Figure 1); EX1029, 3 (“[T]he provider’s network could be IPv6 only, or any other

layer 3/2.5 protocol.”). The IPv6 network core interconnects different IPv4 domains, like the IPv4 Internet or customer IPv4 networks. EX1029, 2 (Figure 1); *see also* EX1029, 2 (“As the specific problem is insufficient IPv4 address space to number the **IPv4-speaking customer**, we propose to extend the IPv4 address space by assigning to each customer a single IPv4 address which is extended by ‘stealing’ bits from the port number in the TCP/UDP header....” (emphasis added)). Intermediary hardware, like customer premises equipment, provider edge routers, and border routers interface the IPv6 and IPv4 domains. EX1029, 2 (Figure 1); *see also* EX1029, 2 (“PE is dual-stack IPv4/IPv6 and connect customer via a direct interface; BR’s are also dual-stack and terminate the IPv4 tunnels.”); *see also* EX1029, 3 (“The customer premises equipment would be configured, hopefully automatically, with IPv4 and/or IPv6 addressing for the customer’s LAN” and “[a]n IPv6 address for the WAN side to connect to the provider.”); EX1041, 12-14. Bush also explains that the intermediary hardware can be a “home gateway,” (EX1041, 12-14), which is the exact term the ’844 patent uses, (*e.g.*, EX1001, 4:59-5:19, 6:54-7:15, 7:32-44, 12:17-19). For consistency, I’ll refer to this hardware generally as “intermediary hardware.”

45. Bush explains that, at the time of the presentation, network providers implemented NAT in the core of their network to mitigate IPv4 exhaustion and translate between IPv4 and IPv6. EX1029, 1-2; EX1041, 7.

Carrier Grade NAT

- NAT in the core of the provider's network to allow 4/6/4 or similar translations
- Customer has 4to6 NAT and the core re-NATs 6to4 for v4 destinations

2008.10.26 A+P

7

EX1041, 7. Bush cautioned against this approach, (EX1029, 1-2; EX1041, 8-11), and recommended moving the translation process to intermediary hardware that interconnects the customer's network with the provider's network.

46. Specifically, Bush teaches an encapsulation process that occurs “in IPv6 in the ISP core.” EX1041, 15. Bush explains that intermediary hardware in an IPv6 domain encapsulates and decapsulates data packets to facilitate IPv4-to-IPv6 communications. EX1041, 15.

47. Bush further teaches that the IPv4 data packet contains an IPv4 destination address and port number. Bush explains that its translation process “steal[s]” bits from the port number to extend the received IPv4 address. EX1041,

15; *see also* EX1041, 21. In other words, the addressing scheme described by Bush involves “assigning to each customer a single IPv4 address which is extended by ‘stealing’ bits from the port number in the TCP/UDP header” of the received data packet. EX1029, 2. “[N]o packets outside of the assigned address and port number range should leave the [provider edge router] for the network.” EX1029, 3.

48. Thus, Bush teaches receiving a data packet from an IPv4 domain in an IPv6 domain, as succinctly summarized in the slide below:

IPv6 Encap Toward CPE

- BR receives IPv4 packet w/ src/dest

PREAMBLE

- Encapsulates in IPv6 packet

 - src = WKP+src

 - dest = WKP+dest

- But note that dest is A+P

- It routes normally within ISP core

2008.10.26 A+P

21

EX1041, 21 (annotations added).

- (b) 1[a]: constructing an IPv6 destination address by concatenating an operator prefix, said IPv4 destination address, and the destination port number;

49. Bush discloses “constructing an IPv6 destination address by concatenating an operator prefix [*e.g.*, **well-known prefix (“WKP”) or IPv6 prefix**], said IPv4 destination address, and the destination port number.”

50. Bush teaches ““steal[ing] bits from the port number to extend the IPv4 Address.” EX1041, 15; *see also* EX1029, 2 (“As the specific problem is insufficient IPv4 address space to number the IPv4-speaking customers, we propose to extend the IPv4 address space by assigning to each customer a single IPv4 address which is extended by ‘stealing’ bits from the port number in the TCP/UDP header.”). Bush explains that this approach of using port numbers to extend IPv4 addresses is “like” past approaches and is “Nothing New.” EX1041, 16-17. “There is a long history of treating port numbers as part of the network address,” including four prior art references identified by Bush that focus on this very idea, some of which pre-date the priority date by decades. EX1029, 4.

51. With this “long history” as background, Bush describes the following process: After the intermediary hardware receives an IPv4 data packet containing an IPv4 destination address and a destination port number, it constructs an IPv6 data packet. EX1041, 21. The IPv6 data packet contains an IPv6 destination address comprising a well-known prefix, EX1041, 19 (giving an example well-known

prefix)), plus a variable called “dest,” (EX1041, 21). Bush “note[s] that dest is A+P,” which is short for “address + port.” EX1041, 21. Thus, the intermediary hardware constructs an IPv6 address by “encapsulat[ing] the A+P pseudo address within an IPv6 address using a well-known IPv6 prefix.” EX1029, 3. Accordingly, Bush teaches constructing an IPv6 address comprising an IPv6 prefix (*i.e.*, an operator prefix), an IPv4 destination address, and a destination port number:

IPv6 Encap Toward CPE

- BR receives IPv4 packet w/ src/dest
- Encapsulates in IPv6 packet
 - src = WKP+src
 - dest = WKP+dest
- But note that dest is A+P
- It routes normally within ISP core

LIMITATION [1A]

2008.10.26 A+P

21

EX1041, 21 (annotations added).

- (c) 1[b]: generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet; and

52. Bush discloses limitation 1[b]: “generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 packet.”

53. Bush explains that the intermediary hardware encapsulates the received IPv4 data packet in an IPv6 data packet, thus generating the IPv6 data packet from the received IPv4 packet. EX1041, 21. As Bush explains, the intermediary hardware “could use IPv6 as a tunneling mechanism.” EX1029, 3. In this case, the intermediary hardware “would encapsulate the A+P pseudo address within an IPv6 address using a well-known IPv6 prefix. Then the core would route on the IPv6 address.” EX1029, 3. As explained above, the generated IPv6 data packet would contain the IPv6 destination address constructed above. *See* EX1041, 21. In this manner, Bush teaches encapsulating the IPv4 data packet in an IPv6 data packet. EX1041, 15, 22. Thus, Bush teaches generating an IPv6 data packet from the IPv6 constructed destination address and the received IPv4 data packet:

IPv6 Encap Toward CPE

- BR receives IPv4 packet w/ src/dest
- Encapsulates in IPv6 packet
 - src = WKP+src
 - dest = WKP+dest
- But note that dest is A+P
- It routes normally within ISP core

LIMITATION [1B]

EX1041, 21 (annotations added).

- (d) 1[c]: routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address, said constructed address belonging to a range of IPv6 addresses routable to an interconnection equipment of the IPv6 domain with the IPv4 destination address

54. Bush discloses limitation 1[c]: “routing the generated IPv6 data packet in the IPv6 domain using the IPv6 constructed destination address, said constructed address belonging to a range of IPv6 addresses [*e.g.*, **range of IPv6 addresses corresponding to devices in the IPv4 domain**] routable to an interconnection equipment [*e.g.*, **the intermediary hardware**] of the IPv6 domain with the IPv4 destination address.”

55. Bush teaches that after the intermediary hardware constructs an IPv6 data packet, it “route[s] on the IPv6 address” “normally within [the] ISP core.” EX1029, 3; EX1041, 21. The first intermediary hardware routes the IPv6 data packet to a second intermediary hardware that decapsulates the IPv6 data packet back into an IPv4 data packet. EX1041, 22-23. The second intermediary hardware services a range of IPv6 destination addresses; it is “[c]onfigured to use a restricted range of ports” and “[c]onfiguration can be as simple or complex as [the POSITA] want[s] it to be.” EX1041, 18. For example, the intermediary hardware “would be configured, hopefully automatically, with...[t]he range of port number to use on the WAN side.” EX1029, 3. The intermediary hardware uses this range of port numbers

to construct IPv6 addresses by “assigning to each customer a single IPv4 address which is extended by ‘stealing’ bits from the port number in the TCP/UDP header.” EX1029, 2. Bush provides one example where “1024 (or 512) customers could be multiplexed on the same IPv4 address, while allowing them a fixed range of 64 (or 128) ports.” EX1029, 2.

56. Thus, not only does Bush teach routing the generated IPv6 data packet in the IPv6 domain using the constructed destination address, but it also teaches that the constructed IPv6 address belongs to a range of IPv6 addresses routable to an interconnection equipment of the IPv6 domain with the IPv4 destination address:

IPv6 Encap Toward CPE

- BR receives IPv4 packet w/ src/dest
- Encapsulates in IPv6 packet
 - src = WKP+src
 - dest = WKP+dest
- But note that dest is A+P
- It routes normally within ISP core

LIMITATION [1C]

2008.10.26 A+P

21

EX1041, 21 (annotations added).

2. Claim 3

- (a) 3[pre]: A method of sending a data packet from an IPv6 domain to a destination address of an IPv4 domain, 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, said method executing the following steps on reception of the data packet:

57. To the extent the preamble is limiting, Bush discloses limitation 3[pre]: “A method of sending a data packet from an IPv6 domain [*e.g.*, **the provider core**] to a destination address of an IPv4 domain [*e.g.*, **a host in the customer's network or a host in the IPv4 Internet**], said packet comprising an IPv6 destination address of the packet constructed by concatenating an operator prefix [*e.g.*, **a well-known prefix or an IPv6 prefix**], the IPv4 destination address, and a destination port number, said method executing the following steps on reception of the data packet.”

58. Bush largely focuses on the process of a first intermediary hardware receiving an IPv4 data packet from an IPv4 domain and encapsulating the IPv4 data packet in an IPv6 data packet with an IPv6 address that comprises an IPv6 prefix, IPv4 destination address, and destination port number. *E.g.*, EX1029, 2; EX1041, 21. But Bush also teaches that—after the first intermediary hardware sends the IPv6 data packet to a second intermediary hardware—the second intermediary hardware decapsulates the IPv6 data packet back into an IPv4 data packet. *See* EX1029, 2 (depicting in Figure 1 that a host in a customer IPv4 network can send a data packet through an IPv6 core to another host in a different customer IPv4 network); EX1029,

3 (Provider Border Routers “must have the ability to forward packets to the PE based on IPv4 address and port. For example, the provider network could use IPv6 as a tunneling mechanism. The CPE or PE routers would encapsulate the A+P pseudo address within an IPv6 address using a well-known IPv6 prefix. Then the core would route on the IPv6 address. The border routers would recognize the well-known IPv6 prefix and decapsulate and forward the IPv4 address.”); EX1041, 15 (“Border Routers also en/decapsulate.”); EX1041, 19 (describing the decapsulation process); EX1041, 22 (“CPE – NATs and handles IPv4 A+P de/encapsulation in IPv6.”); EX1041, 22 (“Border Router – de/encapsulates.”). Bush does not describe the decapsulation process in great detail, but it does not need to. A POSITA would have understood that the decapsulation process is the reverse of the encapsulation process and that it is therefore redundant to explain it in depth.

59. Thus, Bush teaches that the received data packet contains an IPv6 destination address constructed by concatenating an operator prefix, the IPv4 destination address, and a destination port number.

- (b) 3[a]: extracting the IPv4 destination address and the destination port number from the IPv6 destination address;

60. Bush discloses limitation 3[a]: “extracting the IPv4 destination address and the destination port number from the IPv6 destination address.” As Bush teaches, the intermediary hardware “makes [a] global v4 packet” from the received

IPv6 data packet. EX1041, 19. The destination address is the IPv4 destination address found in the IPv6 destination address. EX1041, 19. Bush explains that a “dest” includes both the address and the port number. EX1041, 21. Further, a POSITA would have known that if the intermediary hardware is sending the data packet to another “A + P” intermediary hardware, then the destination port number should be set to the port number contained in the IPv6 destination address, as described throughout Bush. *E.g.*, EX1041, 15, 18-19, 21-22. Doing so enables Bush’s teaching of receiving an IPv6 data packet, “recogniz[ing] the well-known IPv6 prefix” in the IPv6 data packet, and “decapsulat[ing] and forward[ing] the IPv4 address” in the IPv4 domain. EX1029, 3.

(c) 3[b]: generating an IPv4 data packet from the IPv6 data packet, said packet including said IPv4 destination address and said destination port number; and

61. Bush discloses limitation 3[b]: “generating an IPv4 data packet from the IPv6 data packet, said packet including said IPv4 destination address and said destination port number.” Bush explains that the intermediary hardware constructs an IPv4 packet from the received IPv6 packet. EX1041, 19. Bush teaches including the IPv4 destination address as the destination address in the IPv4 data packet. EX1041, 19. Bush explains that a “dest” includes both the address and the port number. EX1041, 21. Further, a POSITA would have known that if the intermediary hardware is sending the data packet to another “A + P” intermediary

hardware, then the destination port number should be set to the port number contained in the IPv6 destination address, as described throughout the Bush Presentation. *E.g.*, EX1041, 15, 18-19, 21-22. Doing so enables Bush's teaching of receiving an IPv6 data packet, "recogniz[ing] the well-known IPv6 prefix" in the IPv6 data packet, and "decapsulat[ing] and forward[ing] the IPv4 address" in the IPv4 domain. EX1029, 3.

(d) 3[c]: routing the IPv4 data packet to its IPv4 destination.

62. Bush discloses limitation 3[c]: "routing the IPv4 data packet to its IPv4 destination." Bush is directed to routing data packets between IPv4 and IPv6 domains. *E.g.*, EX1029, 2 (depicting that a host in a customer IPv4 network can send a data packet through an IPv6 core to another host in a different customer IPv4 network) EX1041, 15, 22 ("Encapsulate in IPv6 in the ISP core and use normal routing to the edge" to "Border Routers" or "CPE[s]" that "also en/decapsualte."). Bush explains that the intermediary hardware "use normal routing to the edge" of each domain, (EX1041, 15), for example, "[r]outing out from CPE" as described on slide 19, (EX1041, 19-20). And Bush explains that the recipient border router "would recognize the well-known IPv6 prefix and decapsulate and forward the IPv4 address." EX1029, 3. Thus, Bush teaches that the intermediary hardware routes the constructed IPv4 data packet to its IPv4 destination.

VI. GROUND 3: WU'S ADDRESS CONSTRUCTION IS STATELESS

63. As an initial matter, I note that Dr. Mangione-Smith analyzes Wu in isolation rather than in the combination proposed in the petition. As I explained in my original declaration, a POSITA would have been motivated to combine Wu's teaching of including a port number in an IPv6 address to extend the IPv4 address space into the system described in Li. EX1003 ¶¶207-214, 217-218. Dr. Mangione-Smith does not address this discrete combination. Instead, Dr. Mangione-Smith attacks Wu's system as a whole, assuming that a POSITA would bodily incorporate Wu into Li. That misunderstands the combination. As explained in my original report, a POSITA would have been motivated to combine a discrete teaching from Wu (*i.e.*, inclusion of a port number into an IPv6 address) into Li's system. EX1003 ¶¶207-214, 217-218.

64. To the extent the Board allows Patent Owner and Dr. Mangione-Smith to argue that a POSITA would have bodily incorporated Wu into Li, I disagree that this combination would eliminate the benefits of Li. For one, I disagree with Dr. Mangione-Smith regarding whether Wu teaches a stateless method of constructing IPv6 addresses and extracting IPv4 information. In my opinion, Wu teaches a stateless process, not a stateful one. In a stateful process, intermediary servers maintain state, for example, a database table containing mappings of an internal address and port number to an external address and port number. *See*

EX1062, 3 (“Finally, the third category is the *stateful* applications, that keep track of the state of packet processing, usually by setting fields of state related to the flows or connections.”). The servers consult that table to route data packets by performing a lookup based on address and port information in a received data packet and then routing the packet to the internal host according to the result of the lookup. In a stateless process, the server forgoes database tables and lookups and instead dynamically constructs and deconstructs addresses based on information found in the received data packet. *See* EX1062, 2-3 (“There is another category called *stateless* applications. Stateless means that there is no record of previous packet processing and each packet processing has to be handled based entirely on its own information.”).

65. Wu does the latter. “After receiving [data] packet[s],” the external servers “extract[] the external address and external port from the destination IPv6 address.” EX1039, 6; *see also* EX1039, 7, 14-16 (same). And the client dynamically “constructs a SilkRoad address” based on information it receives from the server. EX1039, 5; *see also* EX1039, 13 (same). Wu consistently teaches that constructing the IPv6 address and extracting information from the IPv6 address are dynamic processes, not static ones that involve database lookups. And Wu never says or suggests that devices consult a database or any other persistent state to obtain the

appropriate IPv4 and IPv6 addresses. The address construction and deconstruction described in Wu is therefore not a stateful process, but a stateless one.

66. I disagree with Patent Owner and Dr. Mangione-Smith's reliance on a single quote in Wu to argue that Wu is stateful. Dr. Mangione-Smith says that "Wu is a stateful system" because the "Silkroad address [is] assigned to the client end by the server." EX2015 ¶102 (quoting EX1039, 11). In my opinion, this singular quote is taken out of context; what Wu actually teaches in its description of how the system works is that the server assigns component information to the host—like an IPv4 address and a port number—and then the host dynamically constructs an IPv6 address from that information. Wu explains that the server sends a "router advertisement message" to the host that contains "information for *constructing* the SilkRoad address," and "the client end [then] *constructs* a SilkRoad address" per the received information. EX1039, 13 (emphasis added). When the server later receives a data packet from the host, it "*extracts* the external address and external port of the Client end 1 from the source IPv6 address." EX1039, 14 (emphasis added). These actions—constructing an IPv6 address and extracting information from it—are typical of stateless processes.

67. I note that is exactly what the '844 patent does as well. The system assigns information to the host. EX1001, 8:58-61 ("An example of an IPv6 address prefix...such as the prefix IPv6_prefix_ports_range *assigned* to the home gateway.")

(emphasis added)), 13:57-59 (“This step proposes *assigning* the same IPv4 address to a plurality of gateways.” (emphasis added)), 13:61-63 (“a range of authorized source ports *is assigned* each gateway.”). Then, the system dynamically constructs and deconstructs IPv6 addresses based on this assigned information in order to transmit data packets. *E.g.*, EX1001, 6:13-15 (“The general principle of the invention relies on constructing an IPv6 address from an IPv4 address, a predetermined operator prefix, and a port number.”). Thus, I disagree with Dr. Mangione-Smith’s assertion that Wu teaches a stateful process for constructing and deconstructing IP addresses. Wu teaches the opposite—dynamically creating IP addresses based on received information, which does not involve state. Thus, Wu teaches a stateless process.

VII. GROUND 3: WU TEACHES CONSTRUCTING AN IPV6 ADDRESS COMPRISING A DESTINATION PORT NUMBER

68. Finally, I disagree with Dr. Mangione-Smith regarding whether Wu teaches constructing an IPv6 address comprising a destination port number. Again, Dr. Mangione-Smith is analyzing Wu in isolation rather than its role in the proposed combination. In the combination, a POSITA would have been motivated to incorporate Wu’s teaching that a port number can be included in an IPv6 address in order to expand an IPv4 address space.

69. To the extent the Board allows Patent Owner and Dr. Mangione-Smith to analyze Wu’s broader disclosures that are irrelevant to this discrete teaching that

a POSITA would have been motivated to incorporate in the proposed combination, I disagree with their arguments. Specifically, Wu describes constructing an IPv6 address that contains a 16-bit pseudo external port number:

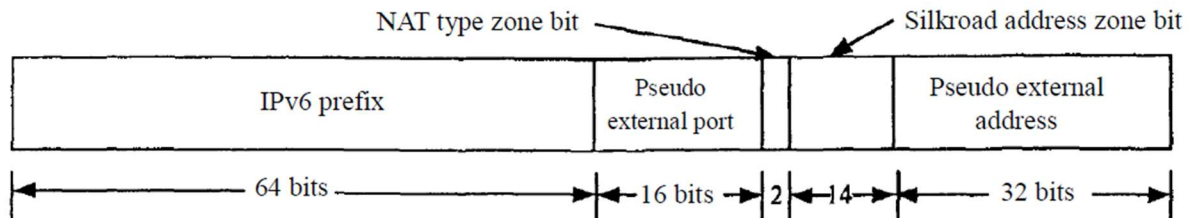


Fig. 1

EX1039, 17.

70. To construct a pseudo external port number, Wu takes an internal, or private port that belongs to the internal host and maps it to an external port via traditional NAT. EX1039, 9 (“When receiving the data packet from the internal network, NAT . . . translates the TCP or UDP source port number¹ into another port

¹ Note that this sentence discusses sending a data packet away from the internal network, which is why it uses the term “*source* port number.” When a responsive data packet is routed back into the internal network—which is the scenario that I mapped to the ’844 claims in the analysis I did in my original declaration (See EX1003 ¶¶86-87 (describing the data packet’s path in the base reference, Li); EX1003 ¶¶205-243 (describing the combination of Li and Wu))—then this port number serves as a *destination* port number.

number, we call this port number as the external port, then sends the translated data packet to the external network.”); EX1039, 3 (“External port is the port mapped by NAT”); EX1039, 3 (“[I]ts UDP port is mapped to a different port when the packet passes NAT, we call it as an external port.”); EX1039, 11 (discussing the “external port (NAT-mapped port)”); EX1039, 11 (“[T]he UDP port of this packet is mapped to a different port when it passes NAT, we call it as the external port.”).

71. Then, Wu performs binary operations on the external port number to create a pseudo external port number. For example, Wu describes ANDing the external port with 0xFFFF. EX1039, 3 (“External port is the port mapped by NAT and 0xFFFF”); EX1039, 11 (describing the “Pseudo external port” as “[t]he external port (NAT-mapped port) and 0xFFFF”). I disagree with Dr. Mangione-Smith that “a POSITA would never implement” a pseudo external port because it allegedly “would [not] make sense to a POSITA” to perform “an ‘and’ bitwise operation between 0xFFFF and any 16-bit number” since that would “simply result[] in the same value as the original number.” EX2015 ¶109. A POSITA would understand that ANDing a number with 0xFFFF ensures that the result is 16 bits. Thus, this operation ensures that any number received that is larger than 16 bits is truncated to 16 bits so that it fits within the constructed IPv6 address:

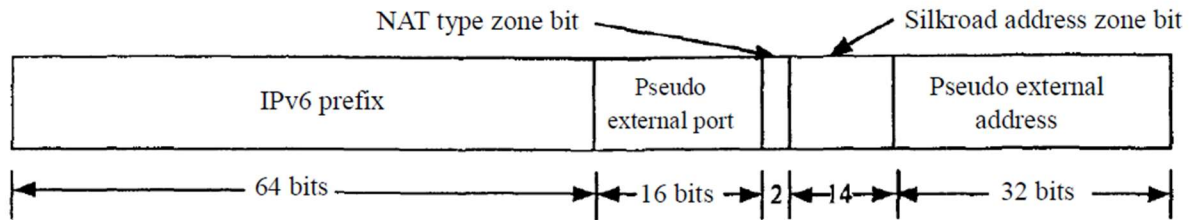


Fig. 1

EX1039, 17. Further, Wu itself explains that it performs these operations “to prevent some NAT from translating the external ports in UDP data packet load.” EX1039, 3; *see also* EX1039, 11 (“The aim of the operation conducted on the external port is to prevent some ‘smarty-pants’ NAT to translate the external port of UDP data packet load, i.e. IPv6 header + IPv6 load.”).

72. The pseudo external port number teaches the '844 patent's destination port number for two primary reasons. First, Wu repeatedly explains that the pseudo external port is itself a destination port. For example, when “Server 2 extracts the external address and external port from the destination IPv6 address” and then sends it to “Client end 2 after encapsulating this packet in UDP, *the destination port of UDP is the extracted external port.*” EX1039, 6 (emphasis added). And when “Client end 2 extracts the external address and external port” from the IPv6 address received from Server 2, “*the destination port is the extracted external port.*” EX1039, 6 (emphasis added). There are many more examples throughout Wu teaching that the pseudo port number used in the IPv6 address is a destination port. EX1039, 6 (“Before sending, it extracts the external address and external port from

the destination IPv6 address which is also the Silkroad address of Client end 2, then encapsulate the data packet in UDP, the destination IPv4 address is the extracted external address, . . . *the destination port is the extracted external port . . .*” (emphasis added)); EX1039, 7 (“Server 2, i.e. the server registered by Client end 2 extracts the external address and external port from the destination IPv6 address after receiving this packet, encapsulates this packet in UDP, and then sends to Client end 2, i.e. the destination host, *the destination port of UDP is the extracted external port . . .*” (emphasis added)); EX1039, 7 (“The server extracts the external address and external port from the destination IPv6 address after receiving this packet, and encapsulates this data packet in the UDP . . . *the destination port is the extracted external port . . .*” (emphasis added)); EX1039, 7 (“This data packet will be routed to the server finally, the server extracts the external address and external port from the destination IPv6 address, and encapsulates this data packet in UDP, . . . *the destination port is the extracted external [port]*,”² then sends the encapsulated data packets to the client end.” (emphasis added)); EX1039, 14 (“After receiving this

² The translation at this line says that “the destination port is the extracted external *address*” instead of *port*, but a POSITA would have recognized this to be a clerical error since an address cannot serve as a port and because the rest of Wu consistently explains that the destination port is the extracted external *port*.

packet, Server 2 extracts the external address and external port from the destination IPv6 address, and sends to Client end 2 after encapsulating this packet in UDP. ***The destination port of UDP is the extracted external port . . .***” (emphasis added)); EX1039, 14 (“After receiving the data packet from Server 2, Client end 2 extracts the external address and external port of the Client end 1 from the source IPv6 address, and learns from the address of Client end 1 that it isn’t located in ‘Symmetric’ NAT field. Then send a maintaining packet to Client end 1, wherein the . . . ***the destination port is the extracted external port.***” (emphasis added)); EX1039, 14-15 (“Next, client end 1 can directly send data packets to client end 2 without forwarding from an intermediate server. Before sending, extract the external address and external port from the destination IPv6 address which is also the SilkRoad address of Client end 2, then encapsulate the data packet in UDP, . . . ***the destination port is the extracted external port . . .***” (emphasis added)); EX1039, 15 (“Server 2 (i.e. the server registered by Client end 2) extracts the external address and external port from the destination IPv6 address after receiving this packet, encapsulates this packet in UDP, and then sends to the destination host of Client end 2. ***The destination port of UDP is the extracted external port . . .***” (emphasis added)); EX1039, 15 (“The server extracts the external address and external port from the destination IPv6 address after receiving this packet, and encapsulates this data packet in the UDP, . . . ***the destination port is the extracted external port.***”

(emphasis added)); EX1039, 15-16 (“This data packet will be routed to the server finally, the server extracts the external address and external port from the destination IPv6 address, and encapsulates this data packet in UDP, . . . ***the destination port is the extracted external address.***” (emphasis added)). Thus, Wu teaches that the pseudo external port the comprises the IPv6 destination address is a destination port.

73. Second, a POSITA would have recognized that there is no need to perform these additional operations on the original port number when combining Wu's teachings with Li. Wu explains that the purpose of these operations (*i.e.*, NATing the original port number into an external port number and then performing binary operations on the external port number to transform it into a pseudo external port number) is to prevent intermediary NATs from interfering with or changing the data packet. EX1039, 3 (“[T]he aim of conducting the operation on the external port is to prevent some NAT from translating the external ports in UDP data packet load, *i.e.* IPv6 header + IPv6 load.”); EX1039, 11 (“The aim of the operation conducted on the external port is to prevent some ‘smarty-pants’ NAT to translate the external port of UDP data packet load, *i.e.* IPv6 header + IPv6 load.”). As Dr. Mangione-Smith explains, Li teaches a stateless system that does not use NATs. *E.g.*, EX2015 ¶97 (“Second, Li is stateless, but Wu is stateful.”); EX2015 ¶103 (“On the other hand, Li is a stateless solution.”). Thus, a POSITA would have recognized that it would be wasteful and inefficient to perform these unneeded operations on the

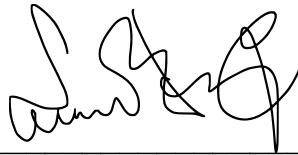
original port number when implementing Wu's teachings into Li, and would have simply concatenated the host's original port number when constructing the IPv6 address.

VIII. CONCLUSION

74. For the reasons stated above, it is my opinion that the Challenged Claims would have been obvious at the time of the invention.

75. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Respectfully submitted,



Date: January 22, 2025

Samrat Bhattacharjee, Ph.D