

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.

Petitioner,

v.

ORCKIT CORPORATION

Patent Owner.

Case IPR2024-00895

U.S. Patent No. 10,652,111

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 10,652,111**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	STANDING AND IDENTIFICATION OF CHALLENGE.....	2
A.	Grounds for Standing (37 C.F.R. §42.104(a))	2
B.	Identification of Challenge and Relief Requested (37 C.F.R. §42.104(b))	2
III.	BACKGROUND	3
A.	Deep Packet Inspection	3
B.	Software-Defined Networking	5
IV.	THE CHALLENGED PATENT	6
A.	'111 Patent Overview	6
B.	Prosecution History	8
C.	Level of Ordinary Skill in the Art	9
D.	Claim Construction	9
1.	“controller”	9
2.	“criterion”	10
V.	THE PRIOR ART	11
A.	Lefebvre Overview	12
B.	Chua Overview	13
C.	Rash Overview	15
VI.	GROUND 1: CLAIMS 32-36 AND 39-54 ARE UNPATENTABLE OVER LEFEBVRE IN VIEW OF CHUA.....	16

A.	A POSITA Would've Been Motivated to Combine Lefebvre and Chua.....	16
B.	Independent Claim 32	25
1.	[32.preamble] “A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method by the network node comprising:”	25
2.	[32.1] “receiving, from the controller, the instruction and the criterion;”	28
3.	[32.2] “receiving, from the first entity over the packet network, a packet addressed to the second entity;”	32
4.	[32.3] “checking if the packet satisfies the criterion;”	33
5.	[32.4] “responsive to the packet not satisfying the criterion, sending over the packet network, the packet to the second entity; and”	34
6.	[32.5] “responsive to the packet satisfying the criterion, sending the packet over the packet network, to an entity that is included in the instruction and is other than the second entity.”	35
C.	Dependent Claims 33-36, 39-54.....	36
1.	[33] The method according to claim 32, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving the ‘terminate’ instruction, the method further comprising blocking, the packet from being sent to the second entity and to the controller.	36
2.	[34] The method according to claim 32, wherein the instruction is a ‘probe’, a ‘mirror’, or a ‘terminate’ instruction, and upon receiving the ‘mirror’ instruction and responsive to the packet satisfying the criterion, the method further comprising sending the packet to the second entity and to the controller.	40

3. [35] The method according to claim 32, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving the ‘probe’ instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet to the controller; receiving the packet, from the controller; and responsive to receiving the packet, sending the packet, to the second entity.....42
4. [36] The method according to claim 32, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof to the controller. 45
5. [39] The method according to claim 32, wherein the packet comprises distinct header and payload fields, the header comprises one or more flag bits, and wherein the packet-applicable criterion is that one or more of the flag bits is set. ..46
6. [40] The method according to claim 39, wherein the packet is an Transmission Control Protocol (TCP) packet, and wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof. 50
7. [41] The method according to claim 32, wherein the packet comprises distinct header and payload fields, the header comprises at least the first and second entities addresses in the packet network, and wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses. 51
8. [42] The method according to claim 41, wherein the addresses are Internet Protocol (IP) addresses. 52
9. [43] The method according to claim 32, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and a TCP sequence mask fields, and wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any

	combination thereof, matches a predetermined value or values.	53
10.	[44] The method according to claim 32, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet, Metropolitan Area Network (MAN), Internet Service Provider (ISP) backbone, datacenter network, or inter-datacenter network.	57
11.	[45] The method according to claim 32, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.....	58
12.	[46] The method according to claim 45, wherein the server device comprises a web server, and wherein the client device comprises a smartphone, a tablet computer, a personal computer, a laptop computer, or a wearable computing device.	60
13.	[47] The method according to claim 45, wherein the communication with the controller is based on, or uses, a standard protocol.....	61
14.	[48] The method according to claim 47, wherein the standard protocol is according to, based on, or compatible with, an OpenFlow protocol version 1.3.3 or 1.4.0.	62
15.	[49] The method according to claim 48, wherein the instruction comprises a Type-Length-Value (TLV) structure.	63
16.	[50] The method according to claim 32, wherein the network node comprises a router, a switch, or a bridge.	63
17.	[51] The method according to claim 32, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.....	65

18.	[52] The method according to claim 51, wherein the packet network is a Transmission Control Protocol (TCP) network, and the packet is an TCP packet.	65
19.	[53] The method according to claim 32, further comprising: receiving, from the first entity over the packet network, one or more additional packets; checking, if any one of the one or more additional packets satisfies the criterion; responsive to an additional packet not satisfying the criterion, sending over the packet network, the additional packet to the second entity; and responsive to the additional packet satisfying the criterion, sending the additional packet over the packet network, in response to the instruction.	66
20.	[54] The method according to claim 32, wherein the packet network is a Software Defined Network (SDN), the packet is routed as part of a data plane and the communication with the controller serves as a control plane.	67
VII.	GROUND 2: CLAIMS 37 AND 38 ARE UNPATENTABLE OVER LEFEBVRE IN VIEW OF CHUA AND RASH	68
A.	A POSITA Would've Been Motivated to Combine Lefebvre and Rash.	69
B.	Dependent Claim 37-38	75
1.	[37] The method according to claim 36, further comprising responsive to the packet satisfying the criterion and to the instruction, sending a portion of the packet to the controller. ..	75
2.	[38] The method according to claim 37, wherein the portion of the packet consists of multiple consecutive bytes, and wherein the instruction comprises identification of the consecutive bytes in the packet.	77
VIII.	OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....	77
IX.	NO BASIS EXISTS FOR DISCRETIONARY DENIAL	78

X.	CONCLUSION.....	82
----	-----------------	----

EXHIBIT LIST¹

Exhibit	Description
1001	U.S. Patent No. 10,652,111
1002	File History for U.S. Patent No. 10,652,111
1003	Declaration of Dr. Nader Mir
1004	Curriculum Vitae of Dr. Nader Mir
1005	U.S. Patent No. 10,097,452 (“Lefebvre”)
1006	U.S. Patent No. 9,264,301 (“Chua”)
1007	U.S. Patent No. 9,813,447 (“Rash”)
1008	M. Liao, <i>et al.</i> , “Design and Evaluation of Deep Packet Inspection System: A Case Study,” IET Networks, pp. 1-8 (2012)
1009	A. W. Moore and K. Papagiannaki, “Toward the Accurate Identification of Network Applications,” In: PAM, Springer, vol. 5, pp. 41-54 (2005)
1010	RFC 3654 (available at https://www.rfc-editor.org/rfc/rfc3654 , last accessed at April 4, 2024)
1011	W. Xia, <i>et al.</i> , “A Survey on Software-Defined Networking,” IEEE Communication Surveys & Tutorials, Vol. 17, No. 1, pp. 27-51 (June 13, 2014)
1012	RFC 3746 (available at https://www.rfc-editor.org/in-notes/rfc3746.txt , last accessed April 4, 2024)
1013	N. McKeown, <i>et al.</i> , “OpenFlow: Enabling Innovation in Campus Networks,” ACM SIGCOMM Computer Communication Review, vol. 38, No. 2, pp. 69-74 (April 2008)

¹ Citations to page numbers reflect the number marked in the bottom of exhibits.

Exhibit	Description
1014	Steven Levy, “Going with the Flow: Google’s Secret Switch to the Next Wave of Networking” (April 17, 2012) (available at https://www.wired.com/2012/04/going-with-the-flow-google/ , last accessed April 4, 2024)
1015	RFC 793 (available at https://www.rfc-editor.org/rfc/pdf/rfc793.txt.pdf , last accessed April 20, 2024)
1016	“About the Open Networking Foundation,” (available at https://opennetworking.org/mission/ , last accessed April 13, 2024)
1017	“Open Networking Foundation Specifications,” (available at https://opennetworking.org/software-defined-standards/specifications/ , last accessed April 13, 2024)
1018	“OpenFlow Switch Specification,” (available at https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.4.0.pdf , last accessed April 13, 2024)
1019	RFC 5810 (available at https://www.rfc-editor.org/rfc/rfc5810.html , last accessed April 13, 2024)

CLAIM LISTING

Independent Claim 32	
32.preamble	A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method by the network node comprising:
32.1	receiving, from the controller, the instruction and the criterion;
32.2	receiving, from the first entity over the packet network, a packet addressed to the second entity;
32.3	checking if the packet satisfies the criterion;
32.4	responsive to the packet not satisfying the criterion, sending over the packet network, the packet to the second entity; and
32.5	responsive to the packet satisfying the criterion, sending the packet over the packet network, to an entity that is included in the instruction and is other than the second entity.

Dependent Claims 33-54	
33	The method according to claim 32, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving the ‘terminate’ instruction, the method further comprising blocking, the packet from being sent to the second entity and to the controller.
34	The method according to claim 32, wherein the instruction is a ‘probe’, a ‘mirror’, or a ‘terminate’ instruction, and upon receiving the ‘mirror’ instruction and responsive to the packet satisfying the criterion, the method further comprising sending the packet to the second entity and to the controller.
35	The method according to claim 32, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving the ‘probe’ instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet to the controller; receiving the packet, from the controller; and responsive to receiving the packet, sending the packet, to the second entity.
36	The method according to claim 32, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof to the controller.

Dependent Claims 33-54	
37	The method according to claim 36, further comprising responsive to the packet satisfying the criterion and to the instruction, sending a portion of the packet to the controller.
38	The method according to claim 37, wherein the portion of the packet consists of multiple consecutive bytes, and wherein the instruction comprises identification of the consecutive bytes in the packet.
39	The method according to claim 32, wherein the packet comprises distinct header and payload fields, the header comprises one or more flag bits, and wherein the packet-applicable criterion is that one or more of the flag bits is set.
40	The method according to claim 39, wherein the packet is an Transmission Control Protocol (TCP) packet, and wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof.
41	The method according to claim 32, wherein the packet comprises distinct header and payload fields, the header comprises at least the first and second entities addresses in the packet network, and wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses.
42	The method according to claim 41, wherein the addresses are Internet Protocol (IP) addresses.
43	The method according to claim 32, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and a TCP sequence mask fields, and wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any combination thereof, matches a predetermined value or values.
44	The method according to claim 32, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet, Metropolitan Area Network (MAN), Internet Service Provider (ISP) backbone, datacenter network, or inter-datacenter network.
45	The method according to claim 32, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.
46	The method according to claim 45, wherein the server device comprises a web server, and wherein the client device comprises a smartphone, a

Dependent Claims 33-54	
	tablet computer, a personal computer, a laptop computer, or a wearable computing device.
47	The method according to claim 45, wherein the communication with the controller is based on, or uses, a standard protocol.
48	The method according to claim 47, wherein the standard protocol is according to, based on, or compatible with, an OpenFlow protocol version 1.3.3 or 1.4.0.
49	The method according to claim 48, wherein the instruction comprises a Type-Length-Value (TLV) structure.
50	The method according to claim 32, wherein the network node comprises a router, a switch, or a bridge.
51	The method according to claim 32, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.
52	The method according to claim 51, wherein the packet network is a Transmission Control Protocol (TCP) network, and the packet is a TCP packet.
53	The method according to claim 32, further comprising: receiving, from the first entity over the packet network, one or more additional packets; checking, if any one of the one or more additional packets satisfies the criterion; responsive to an additional packet not satisfying the criterion, sending over the packet network, the additional packet to the second entity; and responsive to the additional packet satisfying the criterion, sending the additional packet over the packet network, in response to the instruction.
54	The method according to claim 32, wherein the packet network is a Software Defined Network (SDN), the packet is routed as part of a data plane and the communication with the controller serves as a control plane.

MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1)

- **§ 42.8(b)(1):** Juniper Networks, Inc. is the real party-in-interest.
- **§ 42.8(b)(2):** The '111 Patent is at issue in the following cases that may affect, or be affected by, a decision in this proceeding:

Orckit Corporation v. Cisco Systems, Inc., No. 2:22-cv-00276 (E.D. Tex.);

Cisco Systems, Inc. v. Orckit Corporation, IPR2023-00554 (P.T.A.B.);

Reexamination Control No. 90/015,261;

Orckit Corporation v. Arista Networks, Inc., No. 1:23-cv-00821 (D. Del.);

Orckit Corporation v. Juniper Networks, Inc., No. 1:23-cv-00822 (D. Del.); and

Juniper Networks, Inc. v. Orckit Corporation, IPR2024-00037 (P.T.A.B.).

- **§§ 42.8(b)(3), (4):** A Power of Attorney accompanies this Petition. Juniper consents to email service at ipr@fischllp.com.

Lead Counsel	Back-up Counsel
Kyle K. Tsui (Reg. No. 62,602) kyle.tsui@fischllp.com FISCH SIGLER LLP 5301 Wisconsin Avenue NW Suite 400 Washington, DC 20015 Phone: 202.362.3527	Ken K. Fung (<i>pro hac</i> vice forthcoming) ken.fung@fischllp.com FISCH SIGLER LLP 400 Concar Drive San Mateo, CA 94402 Phone: 650.362.8200

I. INTRODUCTION

U.S. Patent No. 10,652,111 (“the ’111 Patent”) relates to deep packet inspection (DPI) in a software-defined network (SDN). Both SDNs and DPI were known well before the 2014 filing date of the ’111 Patent. The patent issued after the applicant amended the claims to require that the controller be external to the network nodes, and that packets satisfying “the criterion” were sent to another entity that was included in the instructions provided to the network node. But nothing claimed was new.

The prior art shows that as early as 2004, early attempts at standardizing the physical separation of network controllers from network nodes began. This concept carried through the introduction of OpenFlow in 2008. So, an external controller isn’t novel. And it was known to program network nodes to transmit packets, satisfying certain conditions, to another entity. Here too, the ’111 Patent didn’t contribute these ideas to the state of the art. And the prior art included controllers that executed DPI. Even the ’111 Patent’s controller that executes DPI wasn’t new.

As the declaration of Dr. Nader Mir, a professor with over thirty years of experience in the field confirms, the claims don’t recite anything new or non-obvious to a person of ordinary skill in the art. Juniper therefore requests trial and cancelation of claims 32-54 of the ’111 Patent.

II. STANDING AND IDENTIFICATION OF CHALLENGE

A. GROUNDS FOR STANDING (37 C.F.R. §42.104(a))

Juniper certifies the '111 Patent is available for IPR. Juniper isn't barred or estopped from requesting IPR on the asserted grounds.

B. IDENTIFICATION OF CHALLENGE AND RELIEF REQUESTED (37 C.F.R. §42.104(b))

Juniper asks the Patent Trial and Appeal Board ("Board") to find claims 32-54 ("Challenged Claims") unpatentable under 35 U.S.C. § 103² based on the following patents and printed publications in view of the general knowledge of a person having ordinary skill in the art (POSITA) prior to the purported invention:

Ground	Proposed Challenges
1	Lefebvre ³ and Chua ⁴ render obvious claims 32-36 and 39-54.
2	Lefebvre, Chua, and Rash ⁵ render obvious claims 37-38.

For the purposes of this Petition, Petitioner assumes the '111 Patent has the priority date of April 22, 2014, the filing date of Provisional Application No. 61/982,358 to which the '111 Patent claims priority.⁶

² 35 U.S.C. §§102, 103 (AIA) applies to the '111 Patent.

³ U.S. Patent No. 10,097,452 (Ex. 1005).

⁴ U.S. Patent No. 9,264,301 (Ex. 1006).

⁵ U.S. Patent No. 9,813,447 (Ex. 1007).

⁶ Mir at ¶¶24, 34.

Lefebvre was filed on July 24, 2012, published on October 17, 2013, and issued on October 9, 2018.⁷ Lefebvre qualifies as prior art under 35 U.S.C.

§ 102(a)(1)-(2). Chua was filed on March 15, 2013 and issued on February 16, 2016.⁸ Chua qualifies as prior art under 35 U.S.C. § 102(a)(2). Rash was filed on March 15, 2013, published on September 18, 2014, and issued on November 7, 2017.⁹ Rash qualifies as prior art under 35 U.S.C. § 102(a)(2). Nothing suggests the U.S. Patent and Trademark Office (“Patent Office”) knew of Lefebvre, Chua, or Rash prior to the filing of this proceeding.

III. BACKGROUND

Both DPI and SDN were known prior to the April 22, 2014 priority date of the ’111 Patent.¹⁰

A. DEEP PACKET INSPECTION

In the early 2000’s, bandwidth misuse and security concerns are becoming increasingly prevalent and problematic.¹¹ To help mitigate this misuse, administrators tried to identify the applications operating on the network. One

⁷ Lefebvre at Cover; Mir at ¶46.

⁸ Chua at Cover; Mir at ¶52.

⁹ Rash at Cover; Mir at ¶57.

¹⁰ Mir at ¶¶24-34.

¹¹ Ex. 1008 (“Liao”) at 1-8; Mir at ¶25.

technique for identifying network applications utilizes port numbers.¹² But this presumes network applications consistently use the same port numbers instead of random network ports, or didn't use designated ports to mask themselves.¹³ Adapting, administrators implemented content-based classification schemes with different levels of complexity.¹⁴

DPI classifies network traffic by inspecting the packet's header and payload.¹⁵ DPI relies on patterns or signatures in the packet to classify traffic.¹⁶ An early DPI system utilized the Netfilter framework.¹⁷ The implementers created a system with a set of four types of matching policies to search for signatures in the packet.¹⁸ With an identified signature, the rule table identifies the service type.¹⁹

¹² Mir at ¶¶25.

¹³ Liao at 1; *see also* Ex. 1009 ("Moore") at 2, 10-11 (disclosing malware or attack traffic using well-known ports of known services); Mir at ¶26.

¹⁴ Moore at 4-8; Mir at ¶26.

¹⁵ '111 Patent at 1:21-29 (describing DPI as one type of dedicated service); Liao at 1; Mir at ¶27.

¹⁶ Liao at 1; Mir at ¶27.

¹⁷ *See generally*, Liao; Mir at ¶28.

¹⁸ Liao at 3-4; Mir at ¶29.

¹⁹ Liao at 3-4; Mir at ¶29.

B. SOFTWARE-DEFINED NETWORKING

In networking, the control plane determines how the forwarding plane should handle forwarding packets from one place to another.²⁰ The control plane dictates the forwarding behavior of the data plane by, for example, manipulating the forwarding tables, the state of the interfaces, or by adding or removing Network Address Translation (NAT) bindings.²¹ And in networking, data, or forwarding, planes process the packets arriving on inbound interfaces in accordance with the control plane's decisions.²² The data plane handles per-packet processing and performs functions including layer 3 forwarding, metering, shaping, firewalling, NAT, encapsulation, decapsulation, and encryption.²³

In April 2008, professors from several universities published a whitepaper describing OpenFlow.²⁴ OpenFlow facilitates programming of the flow tables from central controllers using common sets of functions shared by many switches and

²⁰ See generally, Ex. 1010 ("RFC3654"); Exs. 1011-1012; Mir at ¶¶30-31.

²¹ RFC3654 at 4-5; Mir at ¶30.

²² RFC3654 at 4-5; Mir at ¶30.

²³ RFC3654 at 4-5; Mir at ¶30.

²⁴ Ex. 1013 ("McKeown"); Mir at ¶32.

routers.²⁵ Using OpenFlow, implementers including some of the world’s largest companies have created SDNs to support their operations.²⁶

IV. THE CHALLENGED PATENT

A. ’111 PATENT OVERVIEW

The ’111 Patent describes a method for DPI in an SDN “performed by a central controller of the SDN.”²⁷ The **controller 111** instructs **network nodes 112** to determine if incoming packets from **client devices 130** communicating with **destination servers 140** require inspection.²⁸ The **controller 111** and the **network nodes 112** communicate using standard protocols such as OpenFlow.²⁹ The **network nodes 112** redirect or mirror packets for inspection by the **controller 111**.³⁰ The **controller 111** then provides inspected data to **application servers 120** for further processing.³¹

²⁵ McKeown at 2; Mir at ¶32.

²⁶ *See generally*, Ex. 1014; Mir at ¶33.

²⁷ *E.g.*, ’111 Patent at 2:27-30, 2:50-51, 4:5-7 (“[T]he central controller 111 is configured to perform deep packet inspection on designated packets from designated flows or TCP sessions.”); Mir at ¶¶34-35, 38-39.

²⁸ ’111 Patent at 4:8-11, FIG. 6 (step S610); Mir at ¶35.

²⁹ ’111 Patent at 3:44-48; Mir at ¶35.

³⁰ ’111 Patent at 4:15-18; Mir at ¶35.

³¹ ’111 Patent at 3:56-61; Mir at ¶35.

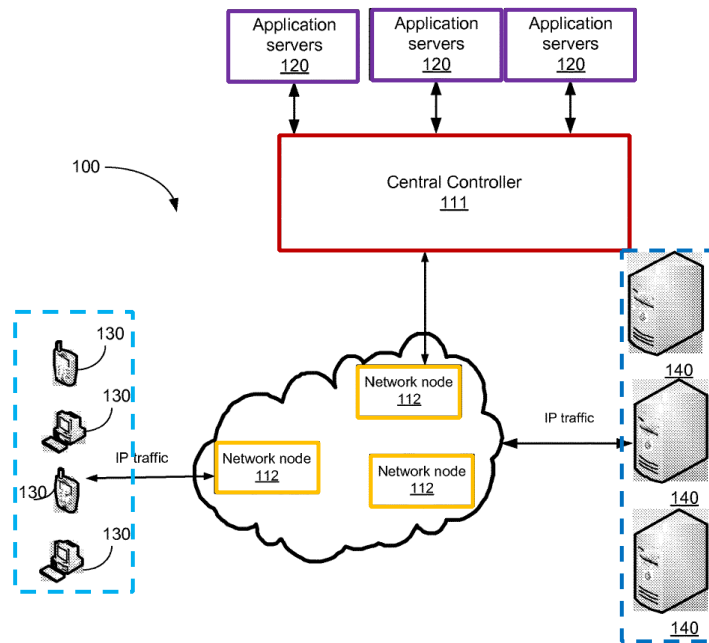


FIG. 1

Annotated '111 Patent FIG. 1

The **controller 111** includes memory 313 and processing unit 314.³² The '111 Patent discloses the **controller 111** provides instructions to the **network nodes 112**.³³ When the instructions apply to a packet, the **network nodes 112** redirect the packet to the **controller 111**.³⁴ And after inspection, the **controller 111** provides the data to the **application servers 120**.³⁵

³² '111 Patent at 5:33-36; Mir at ¶36.

³³ '111 Patent at 6:10-12, 9:40-46, FIG. 6 (step S610); Mir at ¶37.

³⁴ '111 Patent at 5:50-59, 6:12-19, 6:27-32, 6:41-52, 9:47-61, FIG. 6; Mir at ¶37.

³⁵ '111 Patent at 5:36-39; Mir at ¶37.

B. PROSECUTION HISTORY

On April 22, 2014, the applicant filed Provisional Application No. 61/982,358.³⁶ The application entered the U.S. National Stage on September 15, 2016 with 54 claims.³⁷ The Examiner rejected all claims as being unpatentable.³⁸ Applicant eventually amended the independent claims to recite that the controller was “external to the network node” and that the packets satisfying the criterion were sent “to an entity that is included in the instruction and is other than the second entity.”³⁹ Again, the Examiner rejected all claims as being unpatentable, relying on a new secondary reference.⁴⁰ After the applicant’s response, a Notice of Allowance issued on January 7, 2020, stating that the claims were allowed for the reasons the applicant argued.⁴¹ The ’111 Patent issued on May 12, 2020.⁴²

³⁶ ’111 Patent at Cover; Ex. 1002 (“File History”) at 130-146; Mir at ¶40.

³⁷ ’111 Patent at Cover; File History at 54-63 (claims of the 09/15/2016 Preliminary Amendment); Mir at ¶40.

³⁸ File History at 271-296 (09/12/2018 Office Action); Mir at ¶41.

³⁹ File History at 397-417 (05/07/2019 Amendment); Mir at ¶41.

⁴⁰ File History at 425-451 (07/08/2019 Office Action); Mir at ¶42.

⁴¹ File History at 492-510 (10/02/2019 Response; 01/07/2020 Notice of Allowance); Mir at ¶42.

⁴² ’111 Patent at Cover; Mir at ¶43.

C. LEVEL OF ORDINARY SKILL IN THE ART

As of April 22, 2014, a POSITA would've had a bachelor's degree in computer science, computer engineering, or an equivalent, and two years of professional experience, and a POSITA would've had a working knowledge of hardware and software for packet-switched networking.⁴³ Additional education would remedy lack of work experience and vice versa.⁴⁴

D. CLAIM CONSTRUCTION

Aside from the following terms, the terms of the claims have their plain and ordinary meaning as understood by a POSITA.

1. “controller”

The term “controller” should mean “an entity configured to perform deep packet inspection on packets.”⁴⁵ The '111 Patent emphasizes how DPI “is performed by a central controller of the SDN.”⁴⁶ In some embodiments, the

⁴³ Mir at ¶¶60-63.

⁴⁴ *Id.* at ¶61.

⁴⁵ '111 Patent, 10:52-62; Mir at ¶66.

⁴⁶ '111 Patent at 2:27-30, 2:50-51, 3:56-59 (describing the central controller providing “inspected data” to application servers), 4:5-7 (“[T]he central controller 111 is configured to perform deep packet inspection on designated packets from designated flows or TCP sessions.”), 4:8-11, 4:15-18 (describing “[a] packet that requires inspect is either redirected to the controller 111 or mirrored and a copy

controller includes components for performing DPI functions.⁴⁷

In a prior district court case involving the '111 Patent, but that didn't involve Juniper, the district court preliminarily construed the "controller" term to mean "an entity that is capable of controlling deep packet inspection."⁴⁸ Prior to a claim construction decision, the parties jointly stipulated to dismissal of this district court case.⁴⁹ But the patent never uses the term "capable" when describing the controller's functions. Instead, the patent states that the central controller is "**configured** to perform deep packet inspection."⁵⁰ Under either construction of "controller," the Challenged Claims are unpatentable.⁵¹

2. "criterion"

The term "criterion" doesn't appear in the '111 Patent's specification.⁵² But the patent discloses how each network node determines if an incoming packet

thereof is sent to the controller."), 4:49-50 ("The mirrored bytes are sent to the controller 111 to perform the DPI analysis."); Mir at ¶65.

⁴⁷ '111 Patent at 5:30-64; Mir at ¶65.

⁴⁸ *See Cisco Systems, Inc. v. Orckit Corp.*, IPR2023-00554, Paper 22 at 16-17 (PTAB Jan. 15, 2024).

⁴⁹ *Id.*

⁵⁰ '111 Patent at 4:5-6 (emphasis added); Mir at ¶66.

⁵¹ Mir at ¶67.

⁵² *Id.* at ¶68.

requires inspection or not, based on a set of instructions provided by the controller.⁵³ Then, the packet is either redirected to the controller, or mirrored and a copy thereof is sent to the controller.⁵⁴ Further, the patent includes examples of probe instructions that consider various TCP flags such as the SYNC, ACK, FIN, and RST flags.⁵⁵ The patent states that each node receives an incoming packet either from a client or from a server, analyzes the packet's header, and performs the action respective of the value of the TCP flag.⁵⁶ So, a POSITA would've interpreted "criterion" to mean "the conditions, or basis, used to identify packets for actions, such as inspection."⁵⁷

V. THE PRIOR ART

The prior art confirms that each and every element of the Challenged Claims was known before April 22, 2014.⁵⁸

⁵³ '111 Patent at 4:12-15; Mir at ¶68.

⁵⁴ '111 Patent at 4:15-19, 4:61-5:16 (describing type-length-value structures for parsing and identifying packet fields); Mir at ¶68.

⁵⁵ '111 Patent at 4:26-41; Mir at ¶68.

⁵⁶ '111 Patent at 4:39-46; Mir at ¶68.

⁵⁷ Mir at ¶68.

⁵⁸ *Id.* at ¶¶69-71.

A. LEFEBVRE OVERVIEW

Lefebvre discloses that network operators deploy inline services to “inspect and alter network traffic transiting through their network.”⁵⁹ These services typically existed as dedicated appliances with their own costs.⁶⁰ As network traffic demands increase, network operators seek to selectively direct traffic to these inline services and minimize these costs.⁶¹

Lefebvre discloses that traffic steering is “the action of classifying traffic and directing the different classes of traffic through specific service chains.”⁶² Service chains are ordered sets of services.⁶³ Lefebvre discloses network operators defining service policies specifying traffic classes and the service chains each traffic class traverses.⁶⁴ A centralized controller translates these policies into rules for the programmable switches that steer the network traffic through the service chains.⁶⁵ Services can exist in multiple service chains and operators can locate and

⁵⁹ Lefebvre at 1:19-27; Mir at ¶47.

⁶⁰ Lefebvre at 1:25-27; Mir at ¶47.

⁶¹ Lefebvre at 1:28-43; Mir at ¶47.

⁶² Lefebvre at 1:44-46; Mir at ¶¶48-50.

⁶³ Lefebvre at 1:44; Mir at ¶48.

⁶⁴ Lefebvre at 4:41-47; Mir at ¶49.

⁶⁵ Lefebvre at 4:43-47; Mir at ¶50.

chain these service instances arbitrarily.⁶⁶ And Lefebvre discloses implementing an SDN using OpenFlow.⁶⁷

Lefebvre implements traffic steering using a two-step process. First, Lefebvre classifies incoming packets and assigns a service path based on predefined policies.⁶⁸ Then the packets are forwarded to the services in the assigned service path.⁶⁹ Lefebvre's traffic steering process supports three types of policies: subscriber-based, application-based, and flow-based.⁷⁰

B. CHUA OVERVIEW

Chua discloses that OpenFlow facilitates the ability for SDNs use a standalone controller that “determines the routes and program other devices in the network.”⁷¹

Chua discloses that the SDN controller controls the network devices and

⁶⁶ Lefebvre at 4:48-53; Mir at ¶51.

⁶⁷ Lefebvre at 4:30-40; Mir at ¶49.

⁶⁸ Lefebvre at 6:17-19, Abstract; Mir at ¶50.

⁶⁹ Lefebvre at 6:20-21; Mir at ¶50.

⁷⁰ Lefebvre at 6:26-28; Mir at ¶50.

⁷¹ Chua at 1:25-35; *see also id.* at 1:45-55 (describing the control plane being “physically separate from the data plane (or forwarding plane) for an SDN), 1:35-41, 23:62-24:4 (describing OpenFlow as “an example of an SDN protocol”); Mir at ¶54.

service devices.⁷² Using data from the service devices, Chua's SDN controller makes enforcement decisions including allowing, blocking, mirroring, redirecting, transforming, and encapsulating the packets of the packet flow.⁷³ Chua discloses that in SDNs, service devices may provide security related functionalities including intrusion detection, intrusion prevention, denial of service (DoS) protection, malware detection, anti-virus, packet capture and analysis, and honeypots.⁷⁴

In addition to controlling the SDN, Chua discloses the controller may execute other programs.⁷⁵ Chua discloses controllers hosting Layer 3-Layer 7 services instead of external service devices.⁷⁶ As an example, Chua discloses the controller performing DPI on packets from client devices.⁷⁷

⁷² Chua at 6:25-28; Mir at ¶55.

⁷³ Chua at 6:33-54; Mir at ¶55.

⁷⁴ Chua at 5:50-6:24; Mir at ¶55.

⁷⁵ Chua at 6:36-52; Mir at ¶56.

⁷⁶ Chua at 6:42-52; Mir at ¶56.

⁷⁷ Chua at 10:48-52, 31:48-59; Chua at 6:45-52 (describing a Layer 3 routing stack executing on the controller, and the controller commanding network devices to update forwarding tables in response); Mir at ¶56.

C. RASH OVERVIEW

Rash discloses a network that adjusts policies associated with the operation of devices on the network.⁷⁸ Rash discloses “snooping” of the network traffic flows to identify applications operating on the network.⁷⁹ Rash discloses using signatures, heuristics, and other techniques including “port values, protocol values, statistics analysis, customized programs dependent on characteristics unique to a program and OSI layer 2 information.”⁸⁰

To process this information, Rash discloses a targeted approach where packets, or portions of packets, are mirrored to a destination other than the original intended destination.⁸¹ Rash discloses this helps minimize the data mirrored to identification appliances or other infrastructure devices configured to carry out the application identification function, helping reduce costs.⁸²

⁷⁸ Rash at Abstract; Mir at ¶59.

⁷⁹ *E.g.*, Rash at 6:22-40; Mir at ¶59.

⁸⁰ Rash at 6:50-57; Mir at ¶59.

⁸¹ *E.g.*, Rash at 7:51-59, 9:3-12, 25:30-55, 26:11-35, 27:31-43, 29:58-62, 35:66-36:2; Mir at ¶59.

⁸² Rash at 12:59-13:8, 34:43-51; Mir at ¶59.

**VI. GROUND 1: CLAIMS 32-36 AND 39-54 ARE UNPATENTABLE
OVER LEFEBVRE IN VIEW OF CHUA.**

Both DPI and SDN were well-known and ubiquitous. Both Lefebvre and Chua discuss DPI and SDN.⁸³ Lefebvre and Chua both disclose how DPI often executes in dedicated service devices.⁸⁴ And Chua specifically discloses the SDN controller performing DPI as well.⁸⁵

A POSITA would've found that Lefebvre in view of Chua renders obvious, either expressly or inherently, each and every element of claims 32-36 and 38-54.⁸⁶

**A. A POSITA WOULD'VE BEEN MOTIVATED TO COMBINE LEFEBVRE
AND CHUA.**

A POSITA would've been motivated and found it obvious to modify Lefebvre with Chua's teachings.⁸⁷ Lefebvre discloses techniques for an SDN to steer traffic through inline services.⁸⁸ And Lefebvre discloses several types of services that could form part of the service chain, including DPI.⁸⁹ Lefebvre

⁸³ Lefebvre at 1:21-25, 1:34-38; Chua at 10:48-52, 31:48-59; Mir at ¶70.

⁸⁴ Lefebvre at 1:21-25, 1:34-38; Chua at 31:54-59 (describing DPI service devices); Mir at ¶70.

⁸⁵ Chua at 31:48-54; Mir at ¶70.

⁸⁶ Mir at ¶71.

⁸⁷ *Id.* at ¶72.

⁸⁸ Lefebvre at 1:13-21, 2:41-47; Mir at ¶72.

⁸⁹ Lefebvre at 1:28-38, 4:48-54; Mir at ¶¶72-73.

discloses that traffic steering requires classification of incoming packets and assignment of a service path based on predefined policies.⁹⁰ Then the packets are forwarded to the next service in the assigned service path.⁹¹ Lefebvre discloses how operators define the traffic classes and the service chains for processing network traffic.⁹² Lefebvre discloses programming operator instructions into the switches as rules, routing only certain network traffic through the service chains.⁹³ And Lefebvre discloses switches in the SDN using OpenFlow.⁹⁴

Like Lefebvre, Chua discloses an SDN where a controller determines the behavior of devices in the network and how traffic flows through the network.⁹⁵ And like Lefebvre, Chua discloses OpenFlow devices in the network.⁹⁶ These network devices provide data to Chua's controller, which then makes network enforcement decisions.⁹⁷ Chua's controller considers the traffic on a per-packet or

⁹⁰ Lefebvre at 6:17-19; Mir at ¶73.

⁹¹ Lefebvre at 6:20-21; Mir at ¶¶72-73.

⁹² Lefebvre at 4:41-43; Lefebvre at 6:17-19; Mir at ¶¶72-73.

⁹³ Lefebvre at 4:41-47; Mir at ¶72.

⁹⁴ See Lefebvre at 5:41-44; Mir at ¶¶72-73.

⁹⁵ Chua at 1:25-35; Mir at ¶74.

⁹⁶ Chua at 1:35-41; Mir at ¶74.

⁹⁷ Chua at 6:25-28; Mir at ¶74.

per-flow basis.⁹⁸ Chua's controller then programs network devices of the SDN to implement these decisions.⁹⁹ Chua's service devices include those executing DPI.¹⁰⁰ Chua also discloses the SDN controller executes Layer 4-Layer 7 services, instead of external service devices.¹⁰¹ And as an example, Chua discloses the controller performing DPI.¹⁰²

A POSITA would've been motivated to modify Lefebvre with Chua to help reduce the cost and complexity of the separate dedicated service appliances that Lefebvre discloses existed in the prior art. A POSITA would've been motivated and found it obvious to modify Lefebvre with Chua's teachings because both describe SDNs, and both describe DPI as one type of service in an SDN.¹⁰³ And

⁹⁸ Chua at 7:33-54, 10:48-52, 15:9-17; Mir at ¶74.

⁹⁹ Chua at 7:33-54 (describing different programmed actions taken "on packets of a packet flow"), 31:30-34; Mir at ¶74.

¹⁰⁰ Chua at 10:48-52, 15:4-9, 16:30-33, 31:48-50, Fig. 9 (step 402); Mir at ¶74.

¹⁰¹ Chua at 6:36-52, 6:42-45 ("For example, SDN controller 112 itself may host one or more L4-L7 services, **rather than** (or in addition to) services, provided by service devices 116.") (emphasis added); Mir at ¶74.

¹⁰² Chua at 10:48-52, 15:4-9, 16:30-33, 31:48-50; Mir at ¶75.

¹⁰³ Lefebvre at 1:28-38; Chua at 10:48-52, 15:4-9, 16:30-33, 31:48-50, Fig. 9 (step 402); Mir at ¶75.

both Lefebvre and Chua disclose use of OpenFlow devices.¹⁰⁴ Lefebvre and Chua are therefore in the same field of endeavor and contain overlapping disclosures with similar purposes.¹⁰⁵

Modifying Lefebvre in view of Chua's teachings would've been obvious to a POSITA because of similarities in the ways they classify traffic traversing their networks.¹⁰⁶ Lefebvre has three different levels of policies which have different levels of granularity.¹⁰⁷ Lefebvre's subscriber-based policies rely on the subscriber's IP address.¹⁰⁸ Lefebvre's application-based policies rely on an IP address block and/or a UDP/TCP port on a per application basis, and apply to all subscribers.¹⁰⁹ Both Lefebvre's subscriber-based and application-based policies operate on the packets being received by the SDN.¹¹⁰ And Lefebvre's flow-based policies specify the flow or IP 5-tuple (*i.e.*, source IP address, destination IP

¹⁰⁴ Lefebvre at 5:41-44; Mir at ¶75.

¹⁰⁵ Mir at ¶75.

¹⁰⁶ *Id.* at ¶76.

¹⁰⁷ *Id.*

¹⁰⁸ Lefebvre at 6:32-35; Mir at ¶76.

¹⁰⁹ Lefebvre at 6:38-41; Mir at ¶76.

¹¹⁰ Mir at ¶76.

address, protocol, source port, and destination port).¹¹¹ Similarly, Chua discloses that the SDN uses IP ranges to monitor traffic and route traffic to appropriate flowchains.¹¹² And Chua discloses that the SDN controller performs DPI “on packets from client device[s].”¹¹³ Lefebvre and Chua therefore share similar techniques for classifying the traffic traversing their networks.¹¹⁴

A POSITA would’ve been motivated to look to Chua’s more detailed teachings regarding actions an SDN controller takes in response to information from security-related services like DPI and apply them to Lefebvre.¹¹⁵ Lefebvre mentions DPI as one known service,¹¹⁶ but focuses its disclosure on actions relevant to traffic steering and the passing of the packet through assigned service chains including redirecting or dropping the packet.¹¹⁷ Lefebvre doesn’t provide specifics as to actions resulting from the services processing the traffic along the

¹¹¹ Lefebvre at 6:46-48; Mir at ¶76.

¹¹² Chua at 18:9-16; Mir at ¶76.

¹¹³ Chua at 31:48-50; *see also* Chua at 7:33-54 (describing enforcement actions “on packets of a packet flow”), Fig. 9 (step 402); Mir at ¶76.

¹¹⁴ Mir at ¶76.

¹¹⁵ *Id.* at ¶77.

¹¹⁶ Lefebvre at 6:17-31, 1:28-38 (discussing DPI); Mir at ¶77.

¹¹⁷ Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Mir at ¶77.

service path.¹¹⁸ When implementing DPI in an SDN like Lefebvre's, a POSITA would've sought teachings regarding actions taken after DPI identifies malicious traffic. Chua discloses these actions include allowing, blocking, mirroring, redirecting, transforming, or encapsulating the network traffic.¹¹⁹ A POSITA would've looked to Chua's teachings regarding actions an SDN controller takes in response to information from security-related services like DPI and apply them to Lefebvre.¹²⁰

Chua also discloses that the controller executes other programs instead of external service devices.¹²¹ Lefebvre discloses that known services including DPI are typically dedicated, expensive appliances.¹²² Chua, however, discloses programming an SDN controller to perform DPI.¹²³ A POSITA would recognize Chua's teaching means performing DPI isn't an expensive task requiring dedicated appliances.¹²⁴ Similarly, a POSITA would recognize that modifying Lefebvre's

¹¹⁸ See Lefebvre at 6:17-19; Mir at ¶77.

¹¹⁹ Chua at 7:33-54; Mir at ¶77.

¹²⁰ Mir at ¶77.

¹²¹ Chua at 6:42-45; Mir at ¶78.

¹²² Lefebvre at 1:25-27, 1:34-38; Mir at ¶78.

¹²³ Chua at 31:48-50; Mir at ¶78.

¹²⁴ Mir at ¶78.

controller to incorporate DPI, as Chua teaches, removes the need for the dedicated appliance executing DPI that Lefebvre discloses existed in the prior art.¹²⁵ And doing so would remove the possibility of network failures between the controller and service device, which Chua calls a “complex problem.”¹²⁶ Chua’s controller teachings therefore help simplify the SDN and reduce certain types of failures in the SDN, giving the POSITA further motivation to consider Chua.¹²⁷

Implementing Chua’s teachings in Lefebvre’s controller would’ve been well within the ordinary skill of a POSITA, and the POSITA would’ve had a reasonable expectation of success.¹²⁸ Lefebvre discloses redirecting packets to the appropriate service.¹²⁹ Chua discloses additional actions an SDN controller takes in response to the information provided by services including DPI such as allowing, blocking mirroring, redirecting, transforming, or encapsulating the network traffic.¹³⁰

¹²⁵ *Id.*

¹²⁶ *See* Chua at 19:1-53 (describing redundancy possibilities for SDN appliances); Mir at ¶78.

¹²⁷ Mir at ¶78.

¹²⁸ *Id.* at ¶79.

¹²⁹ Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Mir at ¶79.

¹³⁰ Chua at 7:28-54; Mir at ¶79.

Implementing the actions Chua teaches in Lefebvre would've been straightforward and simple for a POSITA because Chua's actions build on standard actions many networking devices already implement, and Lefebvre already discloses redirecting of packets.¹³¹ For example, OpenFlow switches must support some basic actions including forwarding, encapsulating, and dropping packets.¹³²

Implementing a DPI service in Lefebvre's controller, as Chua teaches, would've been straightforward and simple for a POSITA.¹³³ Lefebvre discloses embodiments that exist as a software product executing programs.¹³⁴ Chua confirms how controllers include processors and memory.¹³⁵ So, a POSITA would've recognized that Lefebvre's controller includes programmable components including processors and memory.¹³⁶ And Chua discloses a software development kit (SDK) for creating SDN controller programs.¹³⁷ As an example,

¹³¹ See Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Mir at ¶79.

¹³² McKeown at 3; Mir at ¶79.

¹³³ Mir at ¶80.

¹³⁴ Lefebvre at 17:17-37; Mir at ¶80.

¹³⁵ Chua at 10:67-11:6, 31:60-32:6; Mir at ¶80.

¹³⁶ Mir at ¶80.

¹³⁷ Chua at 6:38-40; Mir at ¶80.

Chua discloses how an SDN controller performs DPI on packets from a client device.¹³⁸ Thus implementing Chua’s teachings in Lefebvre would require only the routine skill of a POSITA.¹³⁹

A POSITA would’ve had a reasonable expectation of success in modifying Lefebvre with Chua’s controller teachings.¹⁴⁰ Lefebvre confirms the programmability of its components when disclosing embodiments existing as a software product executing programs.¹⁴¹ Chua discloses using the SDK to create “one or more L4-L7 services,” or an L3 application, that execute on the controller “rather than” on service devices.¹⁴² Chua also discloses how the programs may use features of the SDN controller, including commanding network equipment to update forwarding tables.¹⁴³ And Chua discloses the SDN controller executing DPI

¹³⁸ Chua at 31:49-54; Mir at ¶80.

¹³⁹ Mir at ¶80.

¹⁴⁰ *Id.* at ¶81.

¹⁴¹ Lefebvre at 17:17-37; Mir at ¶81.

¹⁴² Chua at 6:37-47; Mir at ¶81.

¹⁴³ Chua at 6:47-52; Mir at ¶81.

on packets.¹⁴⁴ Thus, a POSITA would've had a reasonable expectation of success when implementing a DPI service in Lefebvre's controller, as Chua teaches.¹⁴⁵

B. INDEPENDENT CLAIM 32

Lefebvre in view of Chua renders obvious independent claim 32.¹⁴⁶

1. **[32.preamble] "A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method by the network node comprising:"**

To the extent it limits the scope of claim 32, the combination of Lefebvre and Chua renders obvious claim element [32.preamble].¹⁴⁷

Lefebvre discloses operators defining service policies that specify traffic classes that each class must traverse, and the controller translating the policies into rules that are programmed on the switches in the service network.¹⁴⁸ Lefebvre's traffic steering process supports three types of service policies: subscriber-based policies, application-based policies, and flow-based policies.¹⁴⁹

¹⁴⁴ Chua at 31:49-54; Mir at ¶81.

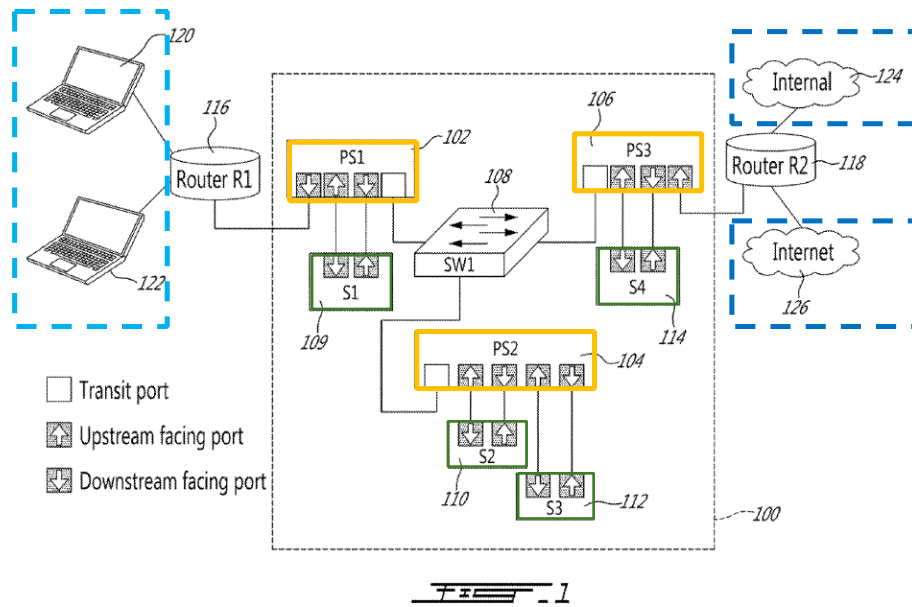
¹⁴⁵ Mir at ¶81.

¹⁴⁶ *Id.* at ¶82.

¹⁴⁷ *Id.* at ¶¶83-89.

¹⁴⁸ Lefebvre at 2:41-47; *see also id.* at 1:13-21; Mir at ¶84.

¹⁴⁹ Lefebvre at 6:26-28; Mir at ¶84.



Annotated Lefebvre FIG. 1

Lefebvre discloses incoming traffic always enters the service network via a perimeter switch and through a node port.¹⁵⁰ Lefebvre discloses processing and steering arriving packets to the next node in the assigned service paths.¹⁵¹ Lefebvre discloses **user equipment 120, 122** connected to router R1 and the **internal network 124** and the **Internet 126** connected to router R2.¹⁵² Lefebvre’s user equipment, internal network, and Internet therefore disclose the “packet network” and “first and second entities.”¹⁵³

¹⁵⁰ Lefebvre at 6:5-8; Mir at ¶85.

¹⁵¹ Lefebvre at 6:8-11; Mir at ¶85.

¹⁵² Lefebvre at 5:38-54, 6:13-16; Mir at ¶85.

¹⁵³ Mir at ¶85.

Lefebvre's service network includes **perimeter switches 102, 104, 106** and an inner switch SW1.¹⁵⁴ The network also includes **services 109, 110, 112, 114**.

Lefebvre discloses that the **perimeter switches** may be OpenFlow switches whereas the inner switch 108 may be either an OpenFlow switch or a plain Ethernet switch.¹⁵⁵ Each of Lefebvre's **perimeter switches** therefore disclose the "network node" of the claims because they, like the '111 Patent's network nodes, communicate with the central controller using OpenFlow.¹⁵⁶

And even though FIG. 1 doesn't depict the controller, Lefebvre discloses the centralized controller pushes these policies to the relevant switches.¹⁵⁷ A POSITA would therefore understand that Lefebvre's controller is external to the **perimeter switches**.¹⁵⁸

And as discussed in Section VII.A, a POSITA would've been motivated and found it obvious to apply Chua's teachings to Lefebvre's SDN controller. Lefebvre's modified controller would've had the programming to execute DPI, as

¹⁵⁴ Lefebvre at 5:38-44; Mir at ¶86.

¹⁵⁵ Lefebvre at 5:41-44; Mir at ¶86.

¹⁵⁶ See '111 Patent at 3:47-48; Mir at ¶86.

¹⁵⁷ Lefebvre at 6:28-31; Mir at ¶87.

¹⁵⁸ Mir at ¶87.

Chua teaches.¹⁵⁹ Because Lefebvre discloses embodiments that already redirect packets to appropriate services and embodiments existing as software programs, a POSITA would've had a reasonable expectation in success in implementing Chua's teachings in Lefebvre's controller.¹⁶⁰ So, a POSITA would recognize Lefebvre's modified controller would disclose the "controller" whether construed as "an entity that is capable of controlling deep packet inspection" or "an entity configured to perform deep packet inspection on packets," as discussed in Section IV.D.2.¹⁶¹

2. [32.1] "receiving, from the controller, the instruction and the criterion;"

The combination of Lefebvre and Chua renders obvious claim element [32.1].¹⁶²

The '111 Patent uses "instruction" for certain commands including "probe instructions," "mirroring instructions," and "termination instructions."¹⁶³ As

¹⁵⁹ *Id.* at ¶88.

¹⁶⁰ *Id.*

¹⁶¹ *Id.*

¹⁶² *Id.* at ¶¶90-96.

¹⁶³ *E.g.*, '111 Patent at 4:23-26; *see also id.* at 4:26-46 (describing redirecting, forwarding, and terminating instructions), 4:47-56 (describing mirroring instructions); *Mir* at ¶91.

discussed in Section IV.D.2, a POSITA would interpret the term “criterion” to mean “the conditions, or basis, used to identify packets for action, like inspection.”¹⁶⁴

Lefebvre discloses the operator defining service policies specifying “traffic classes and the chain of services that each class must traverse.”¹⁶⁵ Lefebvre discloses that the controller programs the rules into the switches of the service network so that the switches steer the network traffic through the ordered chain of services.”¹⁶⁶ And Lefebvre discloses that traffic steering is a two-step process.¹⁶⁷ The first step “classifies incoming packets and assigns them a service path based on predefined policies.”¹⁶⁸ And the second step “forwards packets to a ‘next’ service based on its current position along its assigned service path.”¹⁶⁹

¹⁶⁴ Mir at ¶91.

¹⁶⁵ Lefebvre at 4:43-45; Mir at ¶92.

¹⁶⁶ Lefebvre at 4:43-47, Tables 1, 2 (depicting OpenFlow switch table actions), FIG. 5; Mir at ¶92.

¹⁶⁷ Lefebvre at 6:17-18; Mir at ¶92.

¹⁶⁸ Lefebvre at 6:18-19, 7:35-8:48 (disclosing an example methodology), FIGS. 2, 5 (step 510); Mir at ¶92.

¹⁶⁹ Lefebvre at 6:20-21, FIG. 5 (step 530); Mir at ¶92.

A POSITA would've understood that Lefebvre's traffic steering involving classification of incoming packets to assign them a service path discloses the claimed criterion.¹⁷⁰ Lefebvre's traffic steering policies include subscriber-based policies, application-based policies, and flow-based policies.¹⁷¹ Lefebvre's traffic steering mechanism presumes that all traffic traversing the service network is subscriber traffic.¹⁷² When the traffic isn't subscriber traffic, Lefebvre discloses the SDN functions as a simple router that "does conventional destination-IP-based forwarding" where the traffic does not traverse inline services.¹⁷³ Lefebvre also classifies traffic based on the incoming port, the packet's destination MAC address, the source/destination IP address, the IP address and/or protocol and/or port number, and the position of the packet of the packet on the service path.¹⁷⁴ Lefebvre's classification techniques disclose the claimed criterion because are the conditions, or basis, used to identify packets for actions.¹⁷⁵ Lefebvre discloses the

¹⁷⁰ Mir at ¶93.

¹⁷¹ Lefebvre at 6:26-31; Mir at ¶93.

¹⁷² Lefebvre at 5:19-37; Mir at ¶93.

¹⁷³ Lefebvre at 14:23-33; Mir at ¶93.

¹⁷⁴ Lefebvre at 7:35-8:48 (describing an example methodology), Tables 1, 2 (depicting OpenFlow switch table actions); Mir at ¶93.

¹⁷⁵ Mir at ¶93.

classification decides which service path the packet will traverse. Lefebvre's next destination table routes the packet to the appropriate service.¹⁷⁶ Like the '111 Patent's claimed criterion, Lefebvre's classification techniques depend on instructions provided by the controller and determine which service chain the packet will traverse.¹⁷⁷

A POSITA would've also understood that Lefebvre's traffic steering involving classification of incoming packets to assign them a service path discloses the claimed instruction.¹⁷⁸ For example, Lefebvre's final table, the "next destination table," matches the packet to the appropriate service.¹⁷⁹

And Lefebvre's controller, as modified by Chua's teachings, discloses the claimed controller providing the instruction and criterion.¹⁸⁰ Chua discloses the controller programs network devices "to perform programmed actions on packets" including allowing, blocking, mirroring, redirecting, transforming, and

¹⁷⁶ Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Mir at ¶93.

¹⁷⁷ See '111 Patent at 4:12-46; Mir at ¶93.

¹⁷⁸ Mir at ¶94.

¹⁷⁹ Lefebvre at 8:49-65, Tables 1, 2; Mir at ¶94.

¹⁸⁰ Mir at ¶95.

encapsulating packets.¹⁸¹ Lefebvre’s controller already redirects.¹⁸² As modified by Chua’s teachings, Lefebvre’s controller’s instructions would include allowing, blocking, and mirroring packets.¹⁸³

3. [32.2] “receiving, from the first entity over the packet network, a packet addressed to the second entity;”

Lefebvre discloses claim element [32.2].¹⁸⁴

As discussed for claim element [32.preamble], Lefebvre discloses the “first and second entities.”¹⁸⁵ Specifically, Lefebvre’s **user equipment 120, 122** communicates with the **internal network 124** using the service network 100, so the **user equipment** discloses the first entity.¹⁸⁶ And a POSITA would’ve understood that in doing so, the SDN receives from the **user equipment** a packet addressed to the **internal network**.¹⁸⁷ And Lefebvre confirms this when disclosing the source and destination address fields of the received packet.¹⁸⁸ Lefebvre also

¹⁸¹ Chua at 7:28-54; Mir at ¶95.

¹⁸² See Lefebvre at 8:49-65, Tables 1, 2; Mir at ¶95.

¹⁸³ Mir at ¶95.

¹⁸⁴ *Id.* at ¶¶97-99.

¹⁸⁵ *Id.* at ¶98.

¹⁸⁶ Lefebvre at 5:38-54, 6:8-16; Mir at ¶98.

¹⁸⁷ Mir at ¶98.

¹⁸⁸ Lefebvre at 2:53-64; Mir at ¶98.

discloses that within the network, at transit ports, packets are forwarded to their destination MAC addresses.¹⁸⁹

4. [32.3] “checking if the packet satisfies the criterion;”

Lefebvre discloses claim element [32.3].¹⁹⁰

As discussed for claim element [32.1], Lefebvre discloses that the operator defines service policies specifying traffic classes and the chain of services the traffic classes must traverse,¹⁹¹ and that the first step of traffic steering involve the classification of incoming packets.¹⁹² And as discussed for claim element [32.1], Lefebvre’s classification techniques disclose the claimed criterion because they’re the conditions, or basis, used to identify packets for actions.¹⁹³

A POSITA would’ve understood Lefebvre’s classification of incoming packets discloses the “checking if the packet satisfies the criterion” as claimed.¹⁹⁴

¹⁸⁹ Lefebvre at 5:65-6:12, 7:55-58; Mir at ¶98.

¹⁹⁰ Mir at ¶¶100-103.

¹⁹¹ Lefebvre at 4:43-45; Mir at ¶101.

¹⁹² Lefebvre at 6:18-19, 7:35-8:48 (describing an example methodology), FIGS. 2, 5 (step 510); Mir at ¶101.

¹⁹³ See Lefebvre at 8:49-65, Tables 1, 2; Mir at ¶101.

¹⁹⁴ Mir at ¶102.

Lefebvre's classification checks the incoming packet against the claimed criterion and determines the service path the packet will traverse.¹⁹⁵

5. [32.4] “responsive to the packet not satisfying the criterion, sending over the packet network, the packet to the second entity; and”

Lefebvre discloses claim element [32.4].¹⁹⁶

Lefebvre assumes that the forwarded traffic is subscriber traffic.¹⁹⁷ As discussed for claim element [32.1], one criterion is Lefebvre's distinction between subscriber and non-subscriber traffic.¹⁹⁸ Lefebvre discloses simply forwarding non-subscriber traffic to its destination.¹⁹⁹ Because Lefebvre's non-subscriber traffic doesn't satisfy the criterion of directing subscriber traffic, the network routes the packets from their source, the claimed first entity, to their destination, the claimed second entity.²⁰⁰

¹⁹⁵ *Id.*

¹⁹⁶ Mir at ¶¶104-106.

¹⁹⁷ Lefebvre at 14:23-25; Mir at ¶105.

¹⁹⁸ Mir at ¶105.

¹⁹⁹ Lefebvre at 14:25-29; Mir at ¶105.

²⁰⁰ Mir at ¶105.

6. **[32.5] “responsive to the packet satisfying the criterion, sending the packet over the packet network, to an entity that is included in the instruction and is other than the second entity.”**

The combination of Lefebvre and Chua discloses claim element [32.5].²⁰¹

As discussed for claim element [32.1], Lefebvre’s classification techniques determine which service chain the traffic traverses.²⁰² A POSITA would’ve understood that Lefebvre’s services are “other than” the intended destination of the packet.²⁰³ As discussed in Section VI.A, as modified by Chua’s teachings, Lefebvre’s modified controller would include additional actions including mirroring or redirecting the packets to the service chain.²⁰⁴ As explained in Section VI.A, Lefebvre’s controller, modified by Chua’s teachings, would include additional actions including mirroring the packets to the service chain.²⁰⁵ Chua’s mirroring action allows the traffic and copies the traffic for deeper inspection or recording.²⁰⁶ A POSITA would’ve understood that both actions, mirroring and redirecting of the packets to the service chain, sends the packet to an entity “other

²⁰¹ *Id.* at ¶¶107-110.

²⁰² Lefebvre at 8:49-65, Tables 1, 2; Mir at ¶108.

²⁰³ Mir at ¶108.

²⁰⁴ *Id.*

²⁰⁵ *Id.*

²⁰⁶ *Id.*

than the second entity.”²⁰⁷ A POSITA would’ve been motivated to modify Lefebvre to include these additional actions taught by Chua.²⁰⁸ And a POSITA would understand that Lefebvre’s controller, modified by Chua’s teachings, would execute DPI and therefore be the “entity that is included in the instruction,” which is a destination “other than the second entity.”²⁰⁹

C. DEPENDENT CLAIMS 33-36, 39-54

The combination of Lefebvre and Chua renders obvious dependent claims 33-36 and 39-54.²¹⁰

1. **[33] The method according to claim 32, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving the ‘terminate’ instruction, the method further comprising blocking, the packet from being sent to the second entity and to the controller.**

The combination of Lefebvre and Chua renders obvious dependent claim 33.²¹¹

²⁰⁷ See Lefebvre at 15:27-46 (disclosing forwarding the packet “to the next service on the assigned service chain” and “assigning a new destination address to the received packet in accordance with the next service to be performed”); Mir at ¶108.

²⁰⁸ Mir at ¶109.

²⁰⁹ *Id.*

²¹⁰ *Id.* at ¶¶111-190.

²¹¹ *Id.* at ¶¶112-116.

As discussed for claim element [32.1], Lefebvre's traffic steering policies include subscriber-based policies, application-based policies, and flow-based policies.²¹² Lefebvre's FIG. 2 depicts a methodology for processing and classifying a packet.²¹³

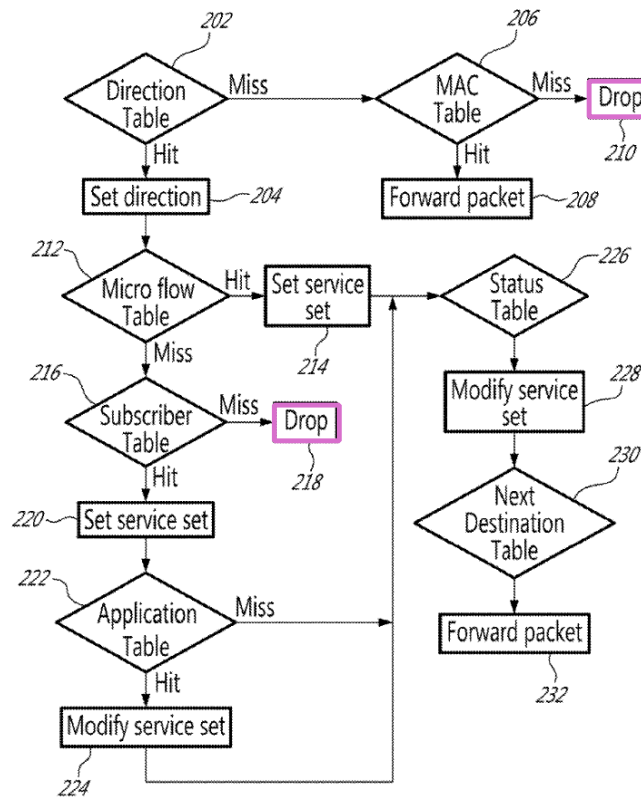


FIG. 2

Annotated FIG. 2 of Lefebvre

²¹² Lefebvre at 6:26-31; Mir at ¶113.

²¹³ Lefebvre at 7:37-39; Mir at ¶113.

In example embodiments, Lefebvre discloses that when a packet arrives on a transit port, it'll "miss" the direction table and proceed to the MAC table 206.²¹⁴ If the packet's destination MAC address doesn't match the contents of the MAC table, Lefebvre drops the packet.²¹⁵ Similarly, Lefebvre discloses the default action drops the packet when there is a miss in the subscriber table.²¹⁶ But Lefebvre doesn't explicitly disclose a terminate instruction that results in blocking.²¹⁷

As discussed for claim element [32.1], Chua discloses the SDN controller programs network devices to perform actions including allowing, blocking, mirroring, redirecting, transforming, or encapsulating traffic.²¹⁸ Chua's blocking action explicitly blocks the packets in the packet flow from traversing the SDN.²¹⁹ Chua's SDN controller 112 performs these actions, including blocking, in response

²¹⁴ Lefebvre at 7:52-54; Mir at ¶114.

²¹⁵ Lefebvre at 7:55-58; Mir at ¶114.

²¹⁶ Lefebvre at 8:11-13, FIG. 2 (step 218); Mir at ¶114.

²¹⁷ Mir at ¶114.

²¹⁸ Chua at 7:28-54; Mir at ¶115.

²¹⁹ Chua at 7:35-38, 7:41-42; Mir at ¶115.

to information from service devices,²²⁰ external triggers or events,²²¹ or business-level security rules.²²²

The Lefebvre and Chua combination would therefore include instructions resulting in blocking, which a POSITA would've recognized as the terminate instruction required by claim 33.²²³ As discussed in Section VI.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's actions including blocking.²²⁴ As modified, the network devices in Lefebvre's SDN would block traffic identified by Lefebvre's controller, which executes DPI as taught by Chua.²²⁵

²²⁰ Chua at 7:28-38, 16:23-44, 28:7-32; Mir at ¶115.

²²¹ Chua at 8:43-47, 17:26-35; Mir at ¶115.

²²² Chua at 14:39-42; Mir at ¶115.

²²³ Mir at ¶115.

²²⁴ See Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller); Mir at ¶115.

²²⁵ Mir at ¶115.

2. **[34] The method according to claim 32, wherein the instruction is a ‘probe’, a ‘mirror’, or a ‘terminate’ instruction, and upon receiving the ‘mirror’ instruction and responsive to the packet satisfying the criterion, the method further comprising sending the packet to the second entity and to the controller.**

The combination of Lefebvre and Chua renders obvious dependent claim 34.²²⁶ As discussed for claim element [32.1], Chua discloses the SDN controller programs network devices to perform actions including allowing, blocking, mirroring, redirecting, transforming, or encapsulating traffic.²²⁷ The combination of Lefebvre and Chua would therefore include the claimed mirror instruction.²²⁸ And as discussed for claim element [32.5], the combination of Lefebvre and Chua renders obvious the claimed “responsive to the packet satisfying the criterion” aspect of claim 34.²²⁹

Lefebvre doesn't disclose a mirroring instruction.²³⁰ But as discussed for claim element [32.1], Chua discloses the SDN controller programming network devices to perform actions including allowing, blocking, mirroring, redirecting,

²²⁶ *Id.* at ¶¶117-120.

²²⁷ Chua at 7:28-54; Mir at ¶118.

²²⁸ Mir at ¶118.

²²⁹ *Id.*

²³⁰ *Id.*

transforming, or encapsulating traffic.²³¹ Chua's mirroring action allows the traffic and copies the traffic for deeper inspection or recording.²³² Chua's SDN controller 112 performs these actions, including mirroring, in response to information from service devices²³³ or business-level security rules.²³⁴

As discussed in Section VI.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's actions including mirroring.²³⁵ A POSITA would've understood that the network devices in Lefebvre's SDN would therefore send the packet to Lefebvre's modified controller as well as the destination.²³⁶ And Lefebvre's controller would execute DPI, as taught by Chua.²³⁷

²³¹ *Id.*; Chua at 7:28-54.

²³² Chua at 7:43-45, 28:7-32; Mir at ¶118.

²³³ Chua at 7:28-38, 16:23-30, 16:36-44, 28:7-32; Mir at ¶118.

²³⁴ Chua at 14:39-42; Mir at ¶118.

²³⁵ *See* Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller); Mir at ¶119.

²³⁶ Mir at ¶119.

²³⁷ *See* Chua at 31:49-54; Mir at ¶119.

3. [35] The method according to claim 32, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving the ‘probe’ instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet to the controller; receiving the packet, from the controller; and responsive to receiving the packet, sending the packet, to the second entity.

The combination of Lefebvre and Chua renders obvious dependent claim

35.²³⁸

The ’111 Patent discloses that the probe instruction includes matching certain aspects of a packet’s header and then redirecting the packet to the controller, and matching certain other aspects of a packet’s header, and then redirecting the packet to the destination.²³⁹

Lefebvre selectively steers traffic away from expensive services, such as DPI,²⁴⁰ but doesn’t explicitly disclose a probe instruction.²⁴¹ But as discussed for claim element [32.1], Chua discloses the SDN controller programs network devices to perform actions including allowing, blocking, mirroring, redirecting, transforming, or encapsulating traffic.²⁴² Because the claimed probe instruction

²³⁸ Mir at ¶¶121-127.

²³⁹ ’111 Patent at 4:23-33; *see also id.* at 6:30-32, FIG. 3 (step S305); Mir at ¶122.

²⁴⁰ *E.g.*, Lefebvre at 1:34-38; Mir at ¶123.

²⁴¹ Mir at ¶123.

²⁴² Chua at 7:28-54; Mir at ¶123.

involves redirects, the combination of Lefebvre and Chua renders obvious the claimed probe instruction.²⁴³ And as discussed for claim element [32.5], the combination of Lefebvre and Chua renders obvious the “responsive to the packet satisfying the criterion” aspect of claim 35.²⁴⁴

Chua’s redirect action sends the traffic to another network device for either inspection or occupying a potential hacker while determining if there is a real security threat.”²⁴⁵ Because Lefebvre’s controller, modified by Chua’s teachings, executes DPI instead of a service device, the combination of Lefebvre and Chua renders obvious the claimed “sending the packet to the controller” of claim 35.²⁴⁶

And Chua discloses that if the service device deems the packets malicious, the controller instructs the service device to send data back to switch such as data of the packet flow after inspection.²⁴⁷ Chua’s SDN controller performs these

²⁴³ Mir at ¶123.

²⁴⁴ *Id.*

²⁴⁵ Chua at 7:46-50; Mir at ¶124.

²⁴⁶ Mir at ¶124.

²⁴⁷ Chua at 21:35-42, 28:18-32 (stating the controller “may program the network devices to, in response to an indication that packets of a packet flow include malicious data, forward packets of the packet flow to a destination of the packet flow.”); Mir at ¶125.

actions, including redirecting, in response to information from service devices²⁴⁸ or business-level security rules.²⁴⁹ Although Chua discloses this in relation to an intrusion detection system (IDS), a POSITA would've found it obvious that service devices executing DPI would've received similar instructions.²⁵⁰ IDS systems analyze network traffic for malicious activity but typically do not employ the more rigorous inspection of packets that DPI systems use.²⁵¹ Because Chua discloses sending data back to the network node, such as “data of the packet flow after the data has been inspected,”²⁵² the combination of Lefebvre and Chua renders obvious the “receiving the packet, from the controller.”²⁵³

As discussed in Section VI.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's teachings regarding SDN actions, including redirecting.²⁵⁴ And Lefebvre's controller would execute DPI, as Chua teaches.²⁵⁵

²⁴⁸ Chua at 7:28-38, 16:23-30, 16:36-44, 28:7-32; Mir at ¶125.

²⁴⁹ Chua at 14:39-42; Mir at ¶125

²⁵⁰ Mir at ¶125.

²⁵¹ *Id.*

²⁵² *See* Chua at 21:35-42, 28:18-32; Mir at ¶125.

²⁵³ Mir at ¶125.

²⁵⁴ *See* Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller); Mir at ¶126.

²⁵⁵ Mir at ¶126.

As modified by Chua’s teachings, Lefebvre’s controller would instruct network devices to redirect certain traffic to the controller which would then execute DPI.²⁵⁶ Once the modified Lefebvre controller completes DPI, the controller sends the inspected packet flow back to the network device for transmission onward to the second entity.²⁵⁷

- 4. [36] The method according to claim 32, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof to the controller.**

The combination of Lefebvre and Chua renders obvious dependent claim 36.²⁵⁸

As discussed for claim element [32.5], the combination of Lefebvre and Chua renders obvious the “responsive to the packet satisfying the criterion” aspect of claim 36.²⁵⁹

As discussed in Section VI.A, Lefebvre’s controller, as modified by Chua’s teachings, would implement Chua’s actions including redirecting.²⁶⁰ And

²⁵⁶ *Id.*

²⁵⁷ *Id.*

²⁵⁸ *Id.* at ¶¶128-131.

²⁵⁹ *Id.* at ¶129.

²⁶⁰ *See* Lefebvre at 8:49-65, Tables 1, 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller); Mir at ¶130.

Lefebvre's controller would execute DPI, as taught by Chua.²⁶¹ Lefebvre's controller, as modified by Chua's teachings, would instruct network devices to redirect certain traffic to the controller which would then execute DPI.²⁶² And because Lefebvre contemplates steering entire packets,²⁶³ the modified Lefebvre controller would continue to redirect the entire packet.²⁶⁴

5. [39] The method according to claim 32, wherein the packet comprises distinct header and payload fields, the header comprises one or more flag bits, and wherein the packet-applicable criterion is that one or more of the flag bits is set.

The combination of Lefebvre and Chua renders obvious dependent claim 39.²⁶⁵

A POSITA would've understood that Transmission Control Protocol (TCP) packets include separate and distinct header and payload fields.²⁶⁶ And a POSITA would've further understood that standard TCP packets include flag bits providing additional information related to the state of the connection.²⁶⁷ RFC 793, published

²⁶¹ See Chua at 31:49-54; Mir at ¶130.

²⁶² Mir at ¶130.

²⁶³ Lefebvre at 6:17-31; *see also id.* at 1:28-38 (discussing DPI); Mir at ¶130.

²⁶⁴ Mir at ¶130.

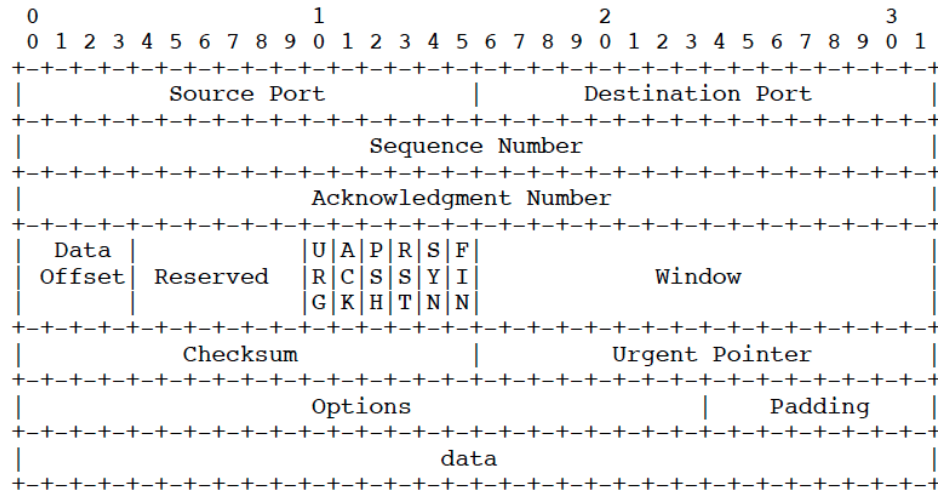
²⁶⁵ *Id.* at ¶¶132-137.

²⁶⁶ *Id.* at ¶133.

²⁶⁷ *Id.*

September 1981, defines TCP.²⁶⁸ Figure 3 of RFC 793 depicts the TCP header format.²⁶⁹

TCP Header Format



TCP Header Format

Note that one tick mark represents one bit position.

Figure 3.

Each TCP header includes source and destination port fields, a sequence number field, along with acknowledgement number, control bits, window, checksum, urgent pointer, and options fields.²⁷⁰ And RFC 793 defines optional features that could exist in the options field including no-operation and maximum segment size.²⁷¹ Since RFC 793, other options including selective

²⁶⁸ See Ex. 1015 (“RFC793”); Mir at ¶133.

²⁶⁹ RFC793 at 21; Mir at ¶133.

²⁷⁰ RFC793 at 21; Mir at ¶134.

²⁷¹ RFC793 at 24-25; Mir at ¶134.

acknowledgements (SACK) and timestamps have utilized the TCP header options fields.²⁷²

Lefebvre discloses using policies considering other standard TCP header fields when classifying traffic, including source/destination addresses, source/destination ports, and protocol fields.²⁷³ Each TCP packet header includes the standard fields defined by RFC 793 such as source/destination ports and addresses, which Lefebvre already considers, and flag bits, which Lefebvre doesn't indicate it considers when classifying traffic.²⁷⁴ Chua similarly confirms the

²⁷² Mir at ¶134.

²⁷³ Lefebvre at 15:55-60 (describing “assigning an ordered list of services to be applied to the packet in accordance with at least one header field of the packet” where the field may include “a source address, a destination address, a source port, a destination port or a protocol”); *see also id.* at 2:44-64, 3:33-45, 16:1-3 (“[t]he default service set can be optionally modified in accordance with at least one header field of the received packet”), 16:67-17:2 (same), 16:58-61 (“[t]he processor 706 can associate the received packet with the service set based on the determined direction and a header field of the packet”), claims 4, 5, 8, 9, 18, 21; Mir at ¶135.

²⁷⁴ RFC793 at 21; Mir at ¶135.

various header fields included with TCP packets including source/destination ports.²⁷⁵

Both Lefebvre and Chua discuss several standard header fields but do not specifically identify the flag bits of TCP packets.²⁷⁶ Each TCP packet has flag bits.²⁷⁷ It therefore would've been obvious to a POSITA to modify Lefebvre to consider the flag bits.²⁷⁸ A POSITA would've been motivated to modify Lefebvre to consider the flag bits' status which each packet includes and provides additional information regarding the connection's state.²⁷⁹ Modifying Lefebvre to consider these standard fields of each packets' header would allow Lefebvre's traffic steering to consider connection state when determining which service chain the packets should traverse.²⁸⁰ Modifying Lefebvre to consider the flag bits' status would've been well within the skill of a POSITA.²⁸¹ Lefebvre already considers other standard aspects of TCP packet headers, so modifying Lefebvre to consider

²⁷⁵ Chua at 26:30-36; Mir at ¶135.

²⁷⁶ Mir at ¶136.

²⁷⁷ *Id.*; RFC793 at 21.

²⁷⁸ Mir at ¶136.

²⁷⁹ *Id.*

²⁸⁰ *Id.*; *see* RFC793 at 21.

²⁸¹ Mir at ¶136.

the flag bits' status in addition to other aspects of the header field would've been well within the ordinary skill of a POSITA.²⁸² Indeed, Chua discloses how the controller programs network devices to perform actions including forwarding based on header data such as source and destination ports.²⁸³

6. **[40] The method according to claim 39, wherein the packet is an Transmission Control Protocol (TCP) packet, and wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof.**

The combination of Lefebvre and Chua renders obvious dependent claim 40.²⁸⁴

As discussed for claim 39, a POSITA would've found it obvious to modify Lefebvre to consider each TCP packet's flag bits.²⁸⁵ A POSITA would've also understood that TCP packet flag bits include the SYN, ACK, FIN, and RST flag

²⁸² *Id.*

²⁸³ *Id.*; Chua at 26:30-36.

²⁸⁴ Mir at ¶¶138-140.

²⁸⁵ *Id.* at ¶139.

bits, among others.²⁸⁶ Each packet includes these standard flag bits defined as part of the TCP standard.²⁸⁷

7. **[41] The method according to claim 32, wherein the packet comprises distinct header and payload fields, the header comprises at least the first and second entities addresses in the packet network, and wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses.**

The combination of Lefebvre and Chua renders obvious dependent claim 41.²⁸⁸

A POSITA would've understood that IP packets include distinct header and payload fields.²⁸⁹ And a POSITA would've understood that IP packets include source and destination addresses.²⁹⁰ For example, Lefebvre discloses associating a service set based on the direction of the packet and the header fields of packets including the "source address, destination address, source port, destination port,

²⁸⁶ *Id.*

²⁸⁷ RFC793 at 22 (describing the defined control bits including SYN, ACK, FIN, and RST); Mir at ¶139.

²⁸⁸ Mir at ¶¶141-144.

²⁸⁹ *Id.* at ¶142.

²⁹⁰ *Id.*

and a protocol.”²⁹¹ The source and destination addresses Lefebvre discloses matching against in the header fields are the claimed first and second entities addresses in the packet network, and the addresses Lefebvre uses to identify the matching packets are the claimed predetermined addresses.²⁹²

8. [42] The method according to claim 41, wherein the addresses are Internet Protocol (IP) addresses.

The combination of Lefebvre and Chua renders obvious dependent claim 42.²⁹³

As discussed for claim 41, IP packets include source and destination addresses.²⁹⁴ A POSITA would’ve understood these addresses are IP addresses.²⁹⁵

²⁹¹ Lefebvre at 2:47-52; Mir at ¶143.

²⁹² Mir at ¶143.

²⁹³ *Id.* at ¶¶145-147.

²⁹⁴ *Id.* at ¶146.

²⁹⁵ *Id.*

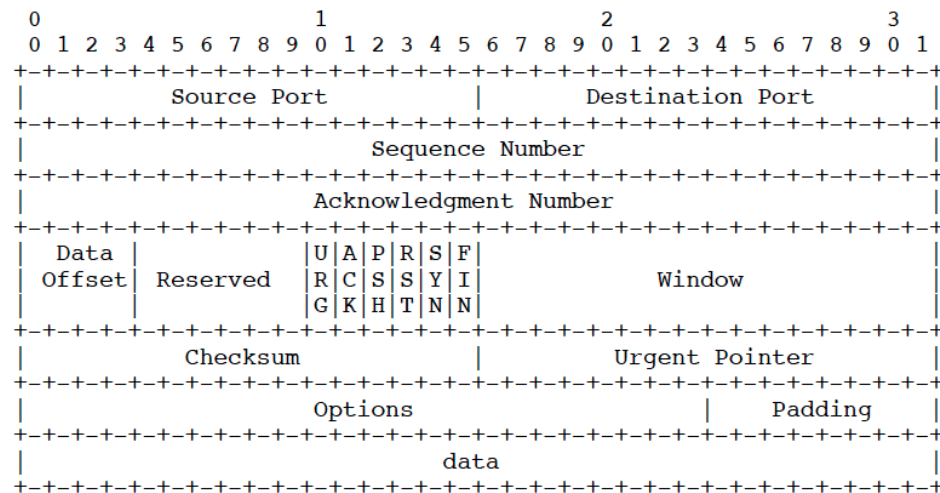
9. [43] The method according to claim 32, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and a TCP sequence mask fields, and wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any combination thereof, matches a predetermined value or values.

The combination of Lefebvre and Chua renders obvious dependent claim

43.²⁹⁶

As discussed, RFC 793 defines TCP and Figure 3 depicts the TCP header format.²⁹⁷

TCP Header Format



TCP Header Format

Note that one tick mark represents one bit position.

Figure 3.

²⁹⁶ *Id.* at ¶¶148-153.

²⁹⁷ RFC793 at 21; Mir at ¶149.

Each TCP header includes source and destination port fields, a sequence number field, along with acknowledgement number, control bits, window, checksum, urgent pointer, and options fields.²⁹⁸ And RFC 793 defines optional features that could exist in the options field including no-operation and maximum segment size.²⁹⁹

RFC 793 doesn't disclose any "TCP sequence mask" field, as claim 43 requires.³⁰⁰ And nothing in the '111 Patent explains what constitutes this "TCP sequence mask" field.³⁰¹ Rather, the patent explains how to compute the TCP sequence mask.³⁰² Regardless of what TCP header field corresponds to "TCP sequence mask," the combination of Lefebvre and Chua render claim 43 obvious.³⁰³

This is because a POSITA would've understood that TCP packets (more typically referred to as TCP segments) include each of the fields required by

²⁹⁸ RFC793 at 21; Mir at ¶150.

²⁹⁹ RFC793 at 24-25; Mir at ¶150.

³⁰⁰ Mir at ¶¶150-151.

³⁰¹ *Id.*

³⁰² '111 Patent at 4:58-60, 6:41-7:40, 9:56-61; Mir at ¶150.

³⁰³ Mir at ¶151.

RFC 793, including source and destination ports, and TCP sequence numbers.³⁰⁴

Lefebvre discloses using the source and destination port and address fields along with the protocol fields, but doesn't specifically identify the TCP sequence number.³⁰⁵ But because the sequence number is part of the TCP standard, defined by RFC 793, as are the fields Lefebvre already considers, it would've been obvious to a POSITA to modify Lefebvre to consider the source and destination addresses, the TCP sequence number, and any other fields necessary to constitute the claimed "TCP sequence mask" field.³⁰⁶

A POSITA would've been motivated to modify Lefebvre to consider these additional standard fields which provide additional information regarding the session including what portion of the data stream the TCP segment constitutes.³⁰⁷ Lefebvre's traffic steering already considers other standard TCP header fields

³⁰⁴ *Id.* at ¶¶151-152.

³⁰⁵ *Id.*; Lefebvre at 15:55-60 (describing "assigning an ordered list of services to be applied to the packet in accordance with at least one header field of the packet" where the field may include "a source address, a destination address, a source port, a destination port or a protocol").

³⁰⁶ *Mir* at ¶¶151-152.

³⁰⁷ *Id.* at ¶152.

when classifying traffic.³⁰⁸ Each TCP packet header includes the fields defined by RFC 793 such as source and destination ports and addresses, which Lefebvre already considers, and sequence numbers, which Lefebvre doesn't.³⁰⁹ Modifying Lefebvre to consider additional standard fields would allow Lefebvre's traffic steering to consider the details provided by these additional standard fields when classifying traffic and determining which service chain the packets should traverse.³¹⁰ Modifying Lefebvre to consider these additional standard fields would've been well within the skill of a POSITA.³¹¹ Lefebvre already considers other standard aspects of TCP headers, so parsing and considering additional standard fields would've been well within the ordinary skill of a POSITA.³¹² Chua confirms this when disclosing how the controller programs network devices to perform actions including forwarding based on standard TCP header data such as source and destination ports.³¹³

³⁰⁸ *Id.*; Lefebvre at 15:55-60; *see also id.* at 2:44-64, 3:33-45, 16:1-3, 16:67-17:2, 16:58-61, claims 4, 5, 8, 9, 18, 21.

³⁰⁹ Mir at ¶152.

³¹⁰ *Id.*

³¹¹ *Id.*

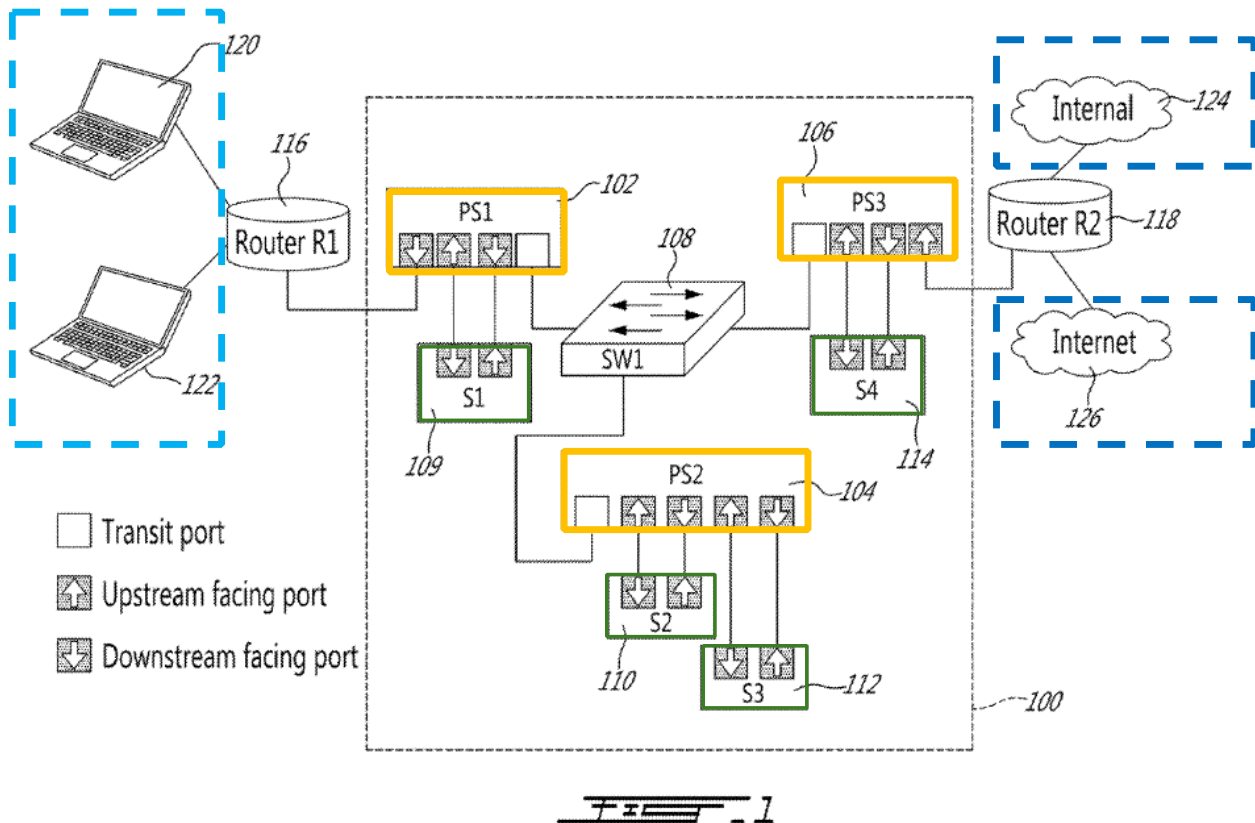
³¹² *Id.*

³¹³ *E.g.*, Chua at 16:27-30; Mir at ¶152.

10. [44] The method according to claim 32, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet, Metropolitan Area Network (MAN), Internet Service Provider (ISP) backbone, datacenter network, or inter-datacenter network.

The combination of Lefebvre and Chua renders obvious dependent claim

44.³¹⁴



Annotated Lefebvre FIG. 1

³¹⁴ Mir at ¶¶154-156.

Lefebvre discloses that the network includes **user equipment** connected to router R1 and the **internal network** and the **Internet** connected to router R2.³¹⁵ A POSITA would've understood that Lefebvre's service network is at least a LAN if not a WAN.³¹⁶ And Lefebvre's service network includes **Internet**.³¹⁷

- 11. [45] The method according to claim 32, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.**

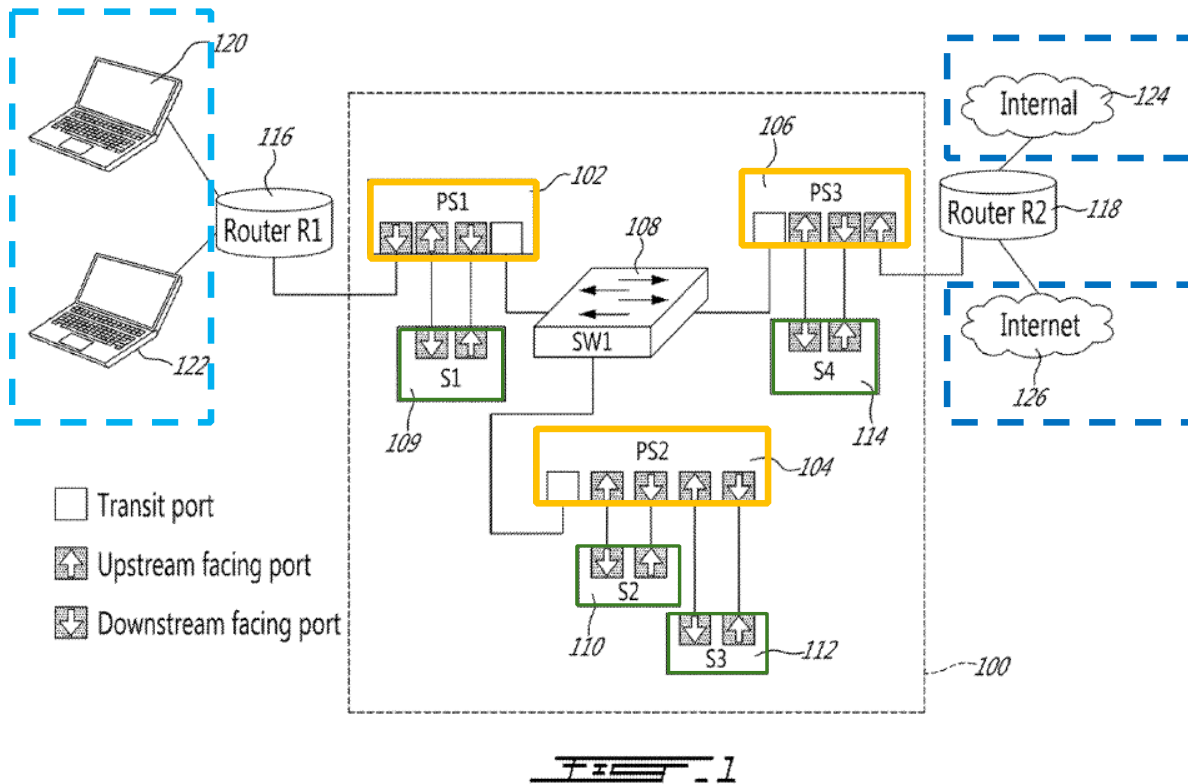
The combination of Lefebvre and Chua renders obvious dependent claim 45.³¹⁸

³¹⁵ Lefebvre at 5:38-54, 6:13-16; Mir at ¶155.

³¹⁶ Mir at ¶155.

³¹⁷ *Id.*

³¹⁸ *Id.* at ¶¶157-159.



Annotated Lefebvre FIG. 1

Lefebvre discloses that the network includes **user equipment** connected to router R1 and the **internal network** and the **Internet** connected to router R2.³¹⁹ A POSITA would've understood that Lefebvre's **user equipment** are client devices.³²⁰ And a POSITA would've understood that the **internal network** and the **Internet** include servers.³²¹

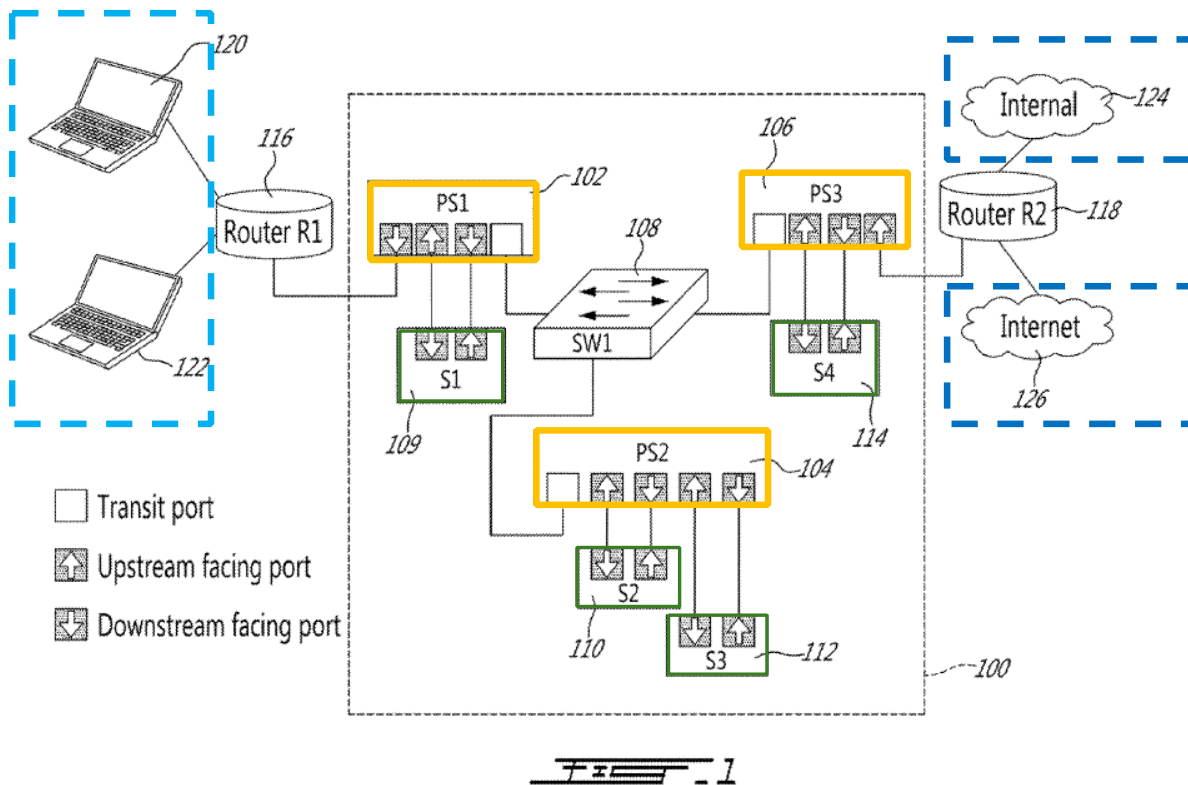
³¹⁹ Lefebvre at 5:38-54, 6:13-16; Mir at ¶158.

³²⁰ Mir at ¶158.

³²¹ *Id.*

- 12. [46] The method according to claim 45, wherein the server device comprises a web server, and wherein the client device comprises a smartphone, a tablet computer, a personal computer, a laptop computer, or a wearable computing device.**

The combination of Lefebvre and Chua renders obvious dependent claim

46.³²²

Annotated Lefebvre FIG. 1

As discussed for claim 45, Lefebvre discloses a service network including **user equipment** connected to router R1 and the **internal network** and the

³²² *Id.* at ¶¶160-162.

Internet connected to router R2.³²³ And Lefebvre depicts **user equipment** as laptops.³²⁴ A POSITA would've understood that Lefebvre's **user equipment** are client devices which could include a smartphone, tablet computer, personal computer, laptop computer, or wearable computer device.³²⁵ And a POSITA would've understood that the **internal network** and the **Internet** include servers, like web servers.³²⁶

13. [47] The method according to claim 45, wherein the communication with the controller is based on, or uses, a standard protocol.

The combination of Lefebvre and Chua renders obvious dependent claim 47.³²⁷

Lefebvre discloses that its SDN use the OpenFlow protocol.³²⁸ Chua confirms one way to implement an SDN is with OpenFlow.³²⁹ The Open Networking Foundation (ONF) manages the OpenFlow protocol and related

³²³ Lefebvre at 5:38-54, 6:13-16; Mir at ¶161.

³²⁴ Lefebvre at FIG. 1; Mir at ¶161.

³²⁵ Mir at ¶161.

³²⁶ See Moore at 11 (describing the detection of web servers operating on non-standard ports on an internal network); Mir at ¶161.

³²⁷ Mir at ¶¶163-166.

³²⁸ Lefebvre at 4:30-40; Lefebvre at 4:33-35; Mir at ¶164.

³²⁹ Chua at 1:35-41; *see also* McKeown; Mir at ¶164.

technologies.³³⁰ So, a POSITA would've understood the OpenFlow protocol is a standard protocol.³³¹

14. [48] The method according to claim 47, wherein the standard protocol is according to, based on, or compatible with, an OpenFlow protocol version 1.3.3 or 1.4.0.

The combination of Lefebvre and Chua renders obvious dependent claim 48.³³²

As discussed for claim 47, the ONF manages the OpenFlow protocol and related technologies.³³³ Lefebvre discloses that its SDN uses certain features from version 1.1 of the OpenFlow protocol.³³⁴

The ONF published version 1.3.3 of the OpenFlow protocol on September 2013.³³⁵ And the ONF published version 1.4.0 of the OpenFlow protocol on October 14, 2013.³³⁶ To the extent that OpenFlow 1.1 isn't compatible with

³³⁰ *E.g.*, Ex. 1016 at 2 (“ONF community standardized the OpenFlow protocol to enable separation of the network control and dataplanes, thus enabling multi-vendor SDN implementations.”); Mir at ¶165.

³³¹ Mir at ¶165.

³³² *Id.* at ¶¶167-170.

³³³ *E.g.*, Ex. 1016 at 2; Mir at ¶168.

³³⁴ Lefebvre at 4:30-40; Mir at ¶168.

³³⁵ Ex. 1017 at 3; Mir at ¶169.

³³⁶ Ex. 1017 at 3; *see also* Ex. 1018; Mir at ¶169.

OpenFlow protocol version 1.3.3 and 1.4.0, a POSITA would've found it obvious to modify Lefebvre's SDN for compatibility, or to be based on, OpenFlow protocol version 1.3.3 or 1.4.0.³³⁷

15. [49] The method according to claim 48, wherein the instruction comprises a Type-Length-Value (TLV) structure.

The combination of Lefebvre and Chua renders obvious dependent claim 49.³³⁸

Type-value-length (TLV) structures are well known data structures often used in protocols.³³⁹ A POSITA would've therefore found it obvious to modify Lefebvre's SDN to utilize instructions with a TLV structure.³⁴⁰

16. [50] The method according to claim 32, wherein the network node comprises a router, a switch, or a bridge.

The combination of Lefebvre and Chua renders obvious dependent claim 50.³⁴¹

Lefebvre discloses a service network in FIG. 1 as annotated below.³⁴²

³³⁷ Mir at ¶169.

³³⁸ *Id.* at ¶¶171-173.

³³⁹ *E.g.*, Ex. 1018 ("RFC5810") at 52; Mir at ¶172.

³⁴⁰ Mir at ¶172.

³⁴¹ *Id.* at ¶¶174-177.

³⁴² *Id.* at ¶175.

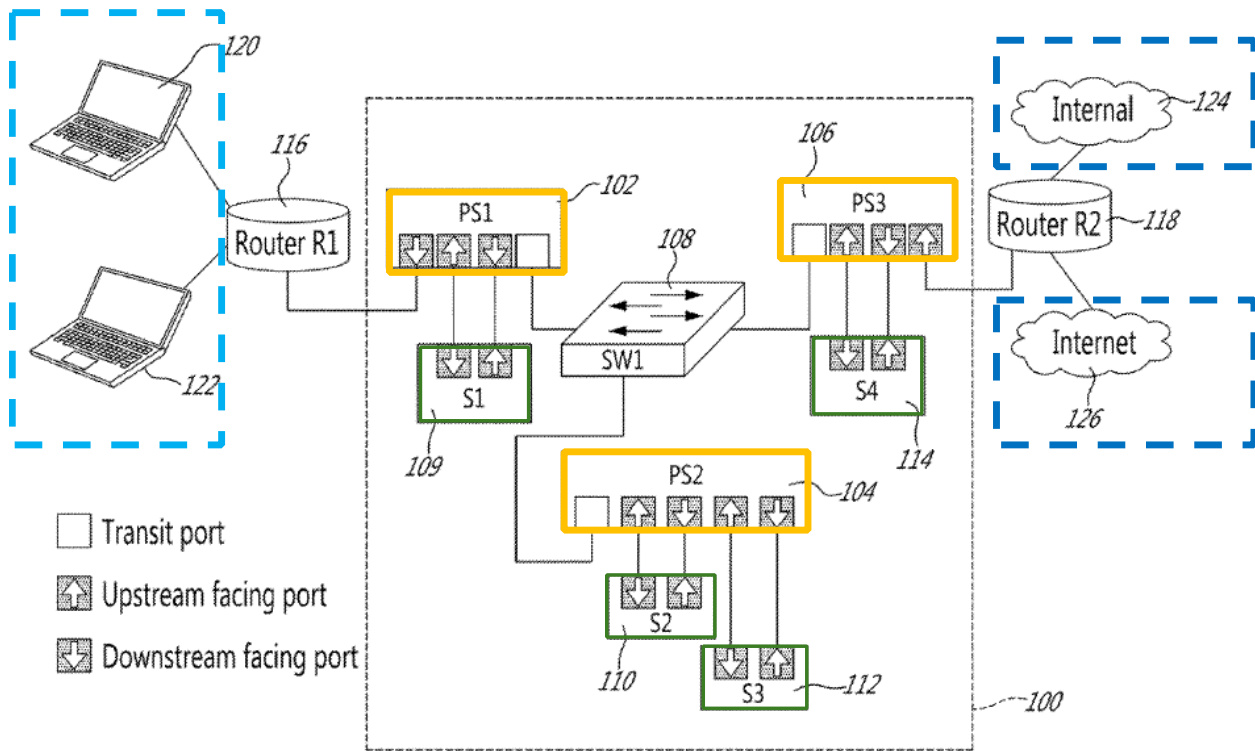


FIG. 1

Annotated Lefebvre FIG. 1

Lefebvre discloses that **perimeter switches** are OpenFlow switches.³⁴³

Lefebvre's controller programs **perimeter switches** to route traffic through the service chain.³⁴⁴

³⁴³ Lefebvre at 5:38-54; Mir at ¶176.

³⁴⁴ E.g., Lefebvre at 5:43-47; Mir at ¶176.

- 17. [51] The method according to claim 32, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.**

The combination of Lefebvre and Chua renders obvious dependent claim 51.³⁴⁵

Lefebvre's traffic steering process contemplates service policies defined by IP address blocks and IP addresses,³⁴⁶ for example. A POSITA would've understood Lefebvre's packet network is an IP network.³⁴⁷

- 18. [52] The method according to claim 51, wherein the packet network is a Transmission Control Protocol (TCP) network, and the packet is an TCP packet.**

The combination of Lefebvre and Chua renders obvious dependent claim 52.³⁴⁸

Lefebvre's traffic steering process contemplates service policies defined by TCP source and destination ports and TCP/UDP flows, for example.³⁴⁹ A POSITA would've understood Lefebvre's packet network is a TCP network with TCP packets.³⁵⁰

³⁴⁵ Mir at ¶¶178-180.

³⁴⁶ Lefebvre at 6:38-41, 6:46-48, 7:60-66; Mir at ¶179.

³⁴⁷ Mir at ¶179.

³⁴⁸ *Id.* at ¶¶181-183.

³⁴⁹ Lefebvre at 6:38-41, 7:60-66; Mir at ¶182.

³⁵⁰ Mir at ¶182.

19. [53] The method according to claim 32, further comprising: receiving, from the first entity over the packet network, one or more additional packets; checking, if any one of the one or more additional packets satisfies the criterion; responsive to an additional packet not satisfying the criterion, sending over the packet network, the additional packet to the second entity; and responsive to the additional packet satisfying the criterion, sending the additional packet over the packet network, in response to the instruction.

The combination of Lefebvre and Chua renders obvious dependent claim

53.³⁵¹

Lefebvre discloses steering packet traffic, which a POSITA would've understood discloses the claimed "one or more additional packets."³⁵² As discussed for claim element [32.1], Lefebvre discloses flow-based policies.³⁵³ A POSITA would've understood this discloses "checking, if any one of the one or more additional packets satisfies the criterion" because they apply to flows, which are one or more additional packets.³⁵⁴ As discussed for claim element [32.4], Lefebvre's non-subscriber traffic doesn't satisfy the criterion of directing subscriber traffic so the packets are routed to the source.³⁵⁵ A POSITA would've

³⁵¹ *Id.* at ¶¶184-186.

³⁵² Lefebvre at 2:30-38; Mir at ¶185.

³⁵³ Mir at ¶185.

³⁵⁴ Lefebvre at 6:46-53; Mir at ¶185.

³⁵⁵ *See* Lefebvre at 14:25-29; Mir at ¶185.

understood this discloses “responsive to an additional packet not satisfying the criterion, sending over the packet network, the additional packet to the second entity.”³⁵⁶ And Lefebvre discloses how multiple packets are “processed and steered” to the next node in their assigned service paths, which a POSITA would’ve understood discloses sending the additional packet over the packet network in response to the instruction.³⁵⁷

20. [54] The method according to claim 32, wherein the packet network is a Software Defined Network (SDN), the packet is routed as part of a data plane and the communication with the controller serves as a control plane.

The combination of Lefebvre and Chua renders obvious dependent claim 54.³⁵⁸

Lefebvre discloses that its service network is a software defined network (SDN).³⁵⁹ A POSITA would’ve understood that in an SDN, the packets are routed as part of the data plane and that communication with the controller serves as the data plane.³⁶⁰

³⁵⁶ Mir at ¶185.

³⁵⁷ Lefebvre at 6:5-12, 6:17-25; Mir at ¶185.

³⁵⁸ Mir at ¶¶187-190.

³⁵⁹ See Lefebvre at 4:30-40; Mir at ¶188.

³⁶⁰ See Ex. 1011 at 2; Mir at ¶188.

VII. GROUND 2: CLAIMS 37 AND 38 ARE UNPATENTABLE OVER LEFEBVRE IN VIEW OF CHUA AND RASH

The Challenged Claims include dependent claims 37 and 38. A POSITA would've recognized that each and every limitation of claims 37-38 are rendered obvious by Lefebvre in view of Chua and Rash.³⁶¹

Well before 2014, both DPI and SDN were well-known and ubiquitous. For example, both Lefebvre and Chua discuss DPI and SDN.³⁶² Lefebvre and Chua both disclose how DPI often executes in dedicated service devices.³⁶³ Chua specifically discloses the SDN controller performing DPI as well.³⁶⁴

Rash discloses how network devices transmit portions of the packet to destinations other than the original intended destination.³⁶⁵ Rash discloses its dynamic mirroring functionality helps minimize the data mirrored to identification appliances or other infrastructure devices configured to carry out the application

³⁶¹ Mir at ¶191.

³⁶² Lefebvre at 1:21-25, 1:34-38; Chua at 10:48-52, 31:48-59; Mir at ¶192.

³⁶³ Lefebvre at 1:21-25, 1:34-38; Chua at 31:54-59 (describing DPI service devices); Mir at ¶192.

³⁶⁴ Chua at 31:48-54; Mir at ¶192.

³⁶⁵ Rash at 7:51-59, 9:3-12, 25:30-55, 26:11-35, 27:31-43, 29:58-62, 35:66-36:2; Mir at ¶193.

identification function.³⁶⁶ And Rash discloses the applicability of its dynamic traffic mirroring function for intrusion detection systems and network access control, among other applications.³⁶⁷

A POSITA would've found that Lefebvre in view of Chua and Rash renders obvious, either expressly or inherently, each and every element of dependent claims 37-38.³⁶⁸

A. A POSITA WOULD'VE BEEN MOTIVATED TO COMBINE LEFEBVRE AND RASH.

A POSITA would've been motivated and found it obvious to modify Lefebvre with Rash's teachings. Lefebvre discloses techniques for an SDN to steer traffic through inline services.³⁶⁹ Lefebvre discloses how operators can define the traffic classes and service chains for those classes for processing traffic flows on a per packet basis.³⁷⁰ Lefebvre discloses these operator instructions are programmed into the switches as rules so that the network traffic passes through the ordered chain of services as desired.³⁷¹

³⁶⁶ Rash at 12:59-13:8, 34:43-51; Mir at ¶193.

³⁶⁷ Rash at 10:29-56; Mir at ¶193.

³⁶⁸ Mir at ¶194.

³⁶⁹ Lefebvre at 1:13-21, 2:41-47; Mir at ¶195.

³⁷⁰ Lefebvre at 4:41-43; Mir at ¶195.

³⁷¹ Lefebvre at 4:41-47; Mir at ¶195.

Lefebvre identifies several types of services that could form part of the service chain, including DPI.³⁷² Lefebvre controls the usage of DPI services by routing network traffic through the service only as appropriate, where the services are separate from the controller.³⁷³

Like Lefebvre, Rash discloses packet inspection systems.³⁷⁴ Rash discloses a packet mirroring function that “provides the capability to select options based on a plurality of criteria to mirror any portion or all of the frames of packets received on the network.”³⁷⁵ Rash discloses how the mirrored traffic may have a different destination based on certain criteria.³⁷⁶ And Rash discloses its dynamic mirroring functionality helps minimize the data mirrored.³⁷⁷

A POSITA would’ve been motivated to modify Lefebvre with Rash to minimize the amount of network capacity used to examine packets, helping reduce

³⁷² Lefebvre at 1:28-38, 4:48-54; Mir at ¶196.

³⁷³ Mir at ¶196.

³⁷⁴ Rash at 7:31-34; Mir at ¶197.

³⁷⁵ Rash at 9:1-12, 27:31-43; *see id.* at 9:12-15, 25:30-55, 26:11-35, 29:46-30:17; Mir at ¶197.

³⁷⁶ Rash at 35:66-36:2; Mir at ¶197.

³⁷⁷ Rash at 12:59-13:8, 34:43-51; Mir at ¶197.

network costs and improving overall efficiency.³⁷⁸ A POSITA would've been motivated and found it obvious to modify Lefebvre with Rash's teachings because both describe configurable networks, and both describe packet inspection as one type of service available in these networks.³⁷⁹ Lefebvre and Rash are therefore in the same field of endeavor and contain overlapping disclosures with similar purposes.³⁸⁰

Modifying Lefebvre in view of Rash's teachings would've been obvious to a POSITA.³⁸¹ Lefebvre has three different levels of policies with different levels of granularity.³⁸² Similarly, Rash explains that the network control engine uses packet fields including address, protocol, length or byte count, and other fields when making its mirroring determinations.³⁸³

³⁷⁸ Mir at ¶198.

³⁷⁹ Lefebvre at 1:28-38; Rash at 9:12-15, 25:30-55, 26:11-35, 29:46-30:17; Mir at ¶198.

³⁸⁰ Mir at ¶198.

³⁸¹ *Id.* at ¶199.

³⁸² Lefebvre at 6:32-48; Mir at ¶199.

³⁸³ Rash at 29:53-58; Mir at ¶199.

Although Lefebvre mentions DPI as one type of known service,³⁸⁴ it considers packets in their entirety and similarly forwards packets in their entirety.³⁸⁵ Rash provides specific details regarding mirroring packets or just portions of packets.³⁸⁶ Rash discloses this approach to mirroring minimizes the data the application identification function must consider.³⁸⁷ A POSITA would therefore look to Rash's detailed teachings regarding mirroring of packets and portions of packets and apply them to Lefebvre.³⁸⁸

A POSITA would've been motivated to modify Lefebvre with Rash's teachings for multiple reasons.³⁸⁹ Lefebvre discloses that services including DPI are typically expensive, dedicated appliances.³⁹⁰ By implementing Rash's mirroring of a portion of a packet instead of the entirety of a packet, a POSITA would recognize the reduction in the amount of network capacity utilized to

³⁸⁴ Lefebvre at 6:17-31; *see also* Lefebvre at 1:28-38 (discussing DPI); Mir at ¶200.

³⁸⁵ *E.g.*, Lefebvre at 6:5-25, 7:42-8:65; Mir at ¶200.

³⁸⁶ Rash at 9:1-12, 27:31-43; Mir at ¶200.

³⁸⁷ Rash at 12:59-13:8, 34:43-51; Mir at ¶200.

³⁸⁸ Mir at ¶200.

³⁸⁹ *Id.* at ¶201.

³⁹⁰ Lefebvre at 1:25-27, 1:34-38; Mir at ¶201.

examine the packets, thereby reducing network costs and improving overall efficiency.³⁹¹ Rash’s dynamic mirroring function leverages existing capabilities in network devices, minimizing additional costs.³⁹² And Rash discloses its mirroring approach minimizes the data the application identification function must consider.³⁹³

Implementing Rash’s teachings in Lefebvre’s controller would’ve been well within the skill of a POSITA.³⁹⁴ Lefebvre already redirects entire packets to the appropriate service.³⁹⁵ Similarly, Rash discloses mirroring packets to the network infrastructure that analyzes the mirrored traffic, and implements mirroring portions of packets.³⁹⁶ Modifying Lefebvre to forward only a portion of the packet as Rash teaches would’ve been well within the skill of a POSITA, as Rash confirms.³⁹⁷

³⁹¹ Rash at 30:18-27; Mir at ¶201.

³⁹² Rash at 30:29-44, 34:37-47; Mir at ¶201.

³⁹³ Rash at 12:59-13:8, 34:43-51; Mir at ¶201.

³⁹⁴ Mir at ¶202.

³⁹⁵ Lefebvre at 8:49-65, Tables 1, 2; Mir at ¶202.

³⁹⁶ *E.g.*, Rash at 7:51-57, 27:15-30 (disclosing the “mirrored traffic destination” includes any device “arranged to receive and either transfer or **analyze** the mirrored frames”) (emphasis added); Mir at ¶202.

³⁹⁷ Mir at ¶203.

A POSITA would've had a reasonable expectation of success in applying Rash's teachings to Lefebvre. Lefebvre already redirects the entirety of packets, so a POSITA would need to modify Lefebvre to consider a portion of a packet.³⁹⁸ Rash's technique allow for "any granularity of the frame of the network traffic" and Rash explains that "[o]ne ordinarily skilled in the art will recognize and appreciate that all or any portion of network traffic, such as all or any available portion of a frame, a packet, or any other protocol data unit, at one or more network protocol layers may be mirrored in practicing the operations of the present invention."³⁹⁹ Because Rash discloses its teachings of mirroring portions of a protocol data unit are within a POSITA's knowledge and ordinary skill, a POSITA would've had a reasonable expectation of success in applying Rash's teachings to Lefebvre. And similarly, implementing this modification would've been simple and straightforward for a POSITA, requiring nothing more than a POSITA's routine skill.⁴⁰⁰

³⁹⁸ See Lefebvre at 8:49-65, Tables 1, 2; Mir at ¶203.

³⁹⁹ Rash at 27:39-43; *see also id.* at 27:44-53; Mir at ¶203.

⁴⁰⁰ Mir at ¶203.

B. DEPENDENT CLAIM 37-38

Lefebvre in view of Chua and Rash renders obvious dependent claims 37 and 38.⁴⁰¹

1. **[37] The method according to claim 36, further comprising responsive to the packet satisfying the criterion and to the instruction, sending a portion of the packet to the controller.**

The combination of Lefebvre, Chua, and Rash renders obvious dependent claim 37.⁴⁰²

As discussed for claim [32.5], the combination of Lefebvre and Chua renders obvious the “responsive to the packet satisfying the criterion” aspect of claim 37.⁴⁰³ And Lefebvre and Chua acknowledge the different aspects of a packet, including the header field.⁴⁰⁴ But Lefebvre and Chua don’t disclose sending a portion of the packet to the controller.⁴⁰⁵

⁴⁰¹ *Id.* at ¶¶204-213.

⁴⁰² *Id.* at ¶¶205-209.

⁴⁰³ *Id.* at ¶206.

⁴⁰⁴ Lefebvre at 15:54-63 (discussing considering the header fields of a packet), 16:67-17:2 (same); Mir at ¶206.

⁴⁰⁵ Mir at ¶206.

Rash discloses a packet inspection system transmitting portions of the packet to a destination other than the intended destination.⁴⁰⁶ Rash discloses a network control manager, controlling the mirroring function, which provides the ability to select options “to mirror any portion or all of the frames of packets”⁴⁰⁷

As discussed in Section VII.A, a POSITA would’ve been motivated and found it obvious to apply Rash’s teachings to Lefebvre.⁴⁰⁸ And Rash discloses that this mirroring (the claimed instruction) sends the portion of packets to “the application identification engine” of the network.⁴⁰⁹ So, Lefebvre’s network, as modified by Rash, would mirror portions of packets to the controller which, as modified by Chua, would execute DPI.⁴¹⁰

⁴⁰⁶ Rash at 7:51-59, 9:3-12, 25:30-55, 26:11-35, 27:31-43, 29:58-62, 35:66-36:2; Mir at ¶207.

⁴⁰⁷ Rash at 9:1-15, 27:31-43, 29:7-12, 29:46-30:17; Mir at ¶207.

⁴⁰⁸ Mir at ¶208.

⁴⁰⁹ *E.g.*, Rash at 7:51-57, 27:15-30 (disclosing the “mirrored traffic destination” includes any device “arranged to receive and either transfer or **analyze** the mirrored frames”) (emphasis added); Mir at ¶208.

⁴¹⁰ Mir at ¶208.

2. **[38] The method according to claim 37, wherein the portion of the packet consists of multiple consecutive bytes, and wherein the instruction comprises identification of the consecutive bytes in the packet.**

The combination of Lefebvre, Chua, and Rash renders obvious dependent claim 38.⁴¹¹

As discussed for claim 37, Lefebvre, Chua, and Rash disclose sending the packet to the controller. Further, Rash discloses that, “based on a plurality of criteria,” its mirroring function can “mirror ... the first L bytes.”⁴¹² A POSITA would’ve understood this means that Rash teaches mirroring multiple consecutive bytes, and that the mirroring (the claimed instruction) identifies the consecutive bytes for mirroring.⁴¹³

VIII. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

Petitioner isn’t aware of any evidence of secondary considerations relevant to the Challenged Claims as of the date of this filing.⁴¹⁴ And Patent Owner hasn’t

⁴¹¹ *Id.* at ¶¶210-212.

⁴¹² Rash at 9:6-12; *see also id.* at 29:7-12, 29:46-62; Mir at ¶211.

⁴¹³ Mir at ¶211.

⁴¹⁴ *Id.* at ¶214.

presented any such evidence either in the district court case or in prior filings before the Office involving the '111 Patent.⁴¹⁵

IX. NO BASIS EXISTS FOR DISCRETIONARY DENIAL

As of the date of this filing, the '111 Patent has been challenged twice. In the first challenge to the '111 Patent, Cisco filed IPR2023-00554 on February 21, 2023 seeking review of different claims (claims 1-31) using different prior art.⁴¹⁶ In the second challenge, Cisco filed Reexamination Control No. 90/015,261 on July 20, 2023 seeking review of all claims of the '111 Patent under a different claim construction standard, using different references.⁴¹⁷ After Patent Owner filed a complaint against Petitioner on July 31, 2023, Petitioner requested, and was granted, joinder with Cisco's IPR.⁴¹⁸ Neither of Cisco's filings challenged claims 32-54 of the '111 Patent with the prior art presented here, in combination with Dr. Mir's expert analysis.

⁴¹⁵ *E.g., Cisco Systems, Inc. v. Orckit Corp.*, IPR2023-00554, Paper 22 (PTAB Jan. 15, 2024).

⁴¹⁶ Petitioner joined this proceeding in a passive, understudy role.

⁴¹⁷ The Board stayed reexamination pending resolution of IPR2023-00554. *See* IPR2023-00554, Paper 27 (Feb. 27, 2024).

⁴¹⁸ *See* IPR2024-00037, Paper 8 (Mar. 26, 2024).

Advanced Bionics, which considers whether the same or substantially the same prior art or arguments have been considered by the Patent Office, offers no reason to exercise discretionary denial.⁴¹⁹ The patentability of claims 32-54 over Lefebvre, Chua, and Rash, with Dr. Mir's expert analysis hasn't been considered by the Patent Office during examination or any other proceeding.⁴²⁰ Lefebvre, Chua, and Rash also materially differ and improve on prior references. For example, Lefebvre's traffic steering technique supports subscriber-based, application-based, and flow-based policies for routing network traffic through service chains.⁴²¹ Chua discloses an SDN controller executing DPI.⁴²² And Rash discloses forwarding portions of packets to the controller.⁴²³ And this Petition's arguments aren't substantially similar to anything the Patent Office previously considered because the prior art combination here differs.⁴²⁴

⁴¹⁹ *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 (Feb. 13, 2020) (precedential).

⁴²⁰ *See generally*, Ex. 1002; IPR2023-00554; Reexamination Control No. 90/015,261.

⁴²¹ *See* Lefebvre at 6:20-28.

⁴²² Chua at 10:48-52, 31:48-59.

⁴²³ Rash at 9:1-12, 27:31-43.

⁴²⁴ *See generally*, Ex. 1002; IPR2023-00554; Reexamination Control No. 90/015,261.

Similarly, all the *General Plastics* factors weigh against discretionary denial.⁴²⁵ **Factors 1, 5:** Petitioner has no relationship with Cisco, who chose to challenge claims 1-31 of the '111 Patent, as well as the art and grounds in IPR2023-00554. The petition in IPR2024-00037 mirrored Cisco's for purposes of joinder and couldn't challenge claims 32-54 in that proceeding. **Factors 2, 4:** Patent Owner filed its complaint on July 31, 2023 and the first petition and motion for joinder were filed October 11, 2023. Petitioner wasn't aware of Lefebvre, Chua, and Rash at the first petition's filing. After discovering Lefebvre, Chua, and Rash, Petitioner diligently worked to prepare and file this Petition. **Factor 3:** Petitioner's decision to file a motion for joinder in IPR2024-00037 occurred within a month of the institution decision in Cisco's IPR. But because IPR2023-00554 didn't challenge claims 32-54, that proceeding's institution decision didn't address the Challenged Claims.⁴²⁶ **Factors 6, 7:** Because this Petition challenges claims 32-54 of the '111 Patent, which don't relate to claims 1-31, the finite resources of the Board aren't implicated and nothing prevents issuance of the final written

⁴²⁵ *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (Sep. 6, 2017).

⁴²⁶ IPR2023-00554, Paper 8 (Sept. 20, 2023).

decision within a year of institution. Given this, *General Plastics* doesn't support discretionary denial.

And likewise, the *Fintiv* factors weigh against discretionary denial.⁴²⁷

Factors 1-3: The district court ordered a stay on November 6, 2023 and no trial date has been set.⁴²⁸ That being less than four months since the complaint's filing, the parties and district court have only minimally invested in the concurrent district court proceeding. This weighs heavily against discretionary denial. **Factor 4:** Petitioner's stipulation in district court confirms it won't assert grounds in the district court that were raised or reasonably could've been raised in this Petition.⁴²⁹

As recognized in *Sotera* and in Patent Office Guidance, "the PTAB will not discretionarily deny institution of an IPR or PGR in view of parallel district court litigation where a petitioner stipulates not to pursue in a parallel district court proceeding the same grounds as in the petition or any grounds that could have

⁴²⁷ See *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

⁴²⁸ *Orckit Corporation v. Juniper Networks, Inc.*, No. 1:23-cv-00822, Dkt. 12 (D. Del. Nov. 6, 2023).

⁴²⁹ *Id.*, Dkt. 11 (D. Del. Nov. 6, 2023).

reasonably been raised in the petition.”⁴³⁰ This too weighs heavily against discretionary denial. **Factor 5:** Petitioner is a defendant in the parallel litigation. But as the Board previously noted, this is “far from an unusual circumstance.”⁴³¹ And, as discussed above, Petitioner only has a passive understudy role in IPR2024-00037. So, this weighs against discretionary denial. **Factor 6:** As the above demonstrates, this Petition presents strong grounds with compelling merits. *Fintiv* doesn’t support discretionary denial either.

X. CONCLUSION

For at least the above reasons, Juniper requests cancelation of claims 32-54 of the ’111 Patent.

Date: May 6, 2024

Respectfully submitted,

/s/ Kyle K. Tsui
Kyle K. Tsui

⁴³⁰ Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (June 21, 2022); *CSC Serviceworks, Inc. v. PayRange Inc.*, IPR2023-01449, Paper 14 at 11 (PTAB April 12, 2024).

⁴³¹ *Sand Revolution II, LLC v. Continental Intermodal Grp.-Trucking LLC*, IPR2019-01393, Paper 24 (June 16, 2020) (instituting where parties were the same) (informative); *Peloton Interactive, Inc. v. NEC Corporation*, IPR2023-01241, Paper 17 at 15 (PTAB Feb. 7, 2024) (declining discretionary denial when there were shared parties between concurrent proceedings).

CERTIFICATE OF COMPLIANCE

Per 37 C.F.R. §42.24(a) and (d), the undersigned hereby certifies that the Petition complies with the type-volume limitation of 37 C.F.R. §42.24(a)(i) because, exclusive of exempted portions, it contains 13,251 words as counted by the word-processing program used to prepare it.

Dated: May 6, 2024

Respectfully submitted,

/s/ Kyle K. Tsui
Kyle K. Tsui (Reg. No. 62,602)
kyle.tsui@fischllp.com
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Suite 400
Washington, DC 20015
Phone: 202.362.3527

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§42.6(e) and 42.105(b) on the Patent Owner by Federal Express of a copy of this Petition for *Inter Partes* Review and supporting materials to the correspondence address of record for U.S. Patent No. 10,652,111:

May Patents Ltd. c/o Dorit Shem-Tov
P.O.B 7230
Ramat-Gan, - 5217102
Israel

An electronic courtesy copy is also being served to litigation counsel at:

Jacob R. Kirkham
Kobre & Kim LLP
600 North King Street, Suite 501
Wilmington, DE 19801
jacob.kirkham@kobrekim.com

Dated: May 6, 2024

Respectfully submitted,

/s/ Kyle K. Tsui
Kyle K. Tsui (Reg. No. 62,602)
kyle.tsui@fischllp.com
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Suite 400
Washington, DC 20015
Phone: 202.362.3527