IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

JUNIPER NETWORKS, INC.,

Petitioner,

v.

ORCKIT CORPORATION,

Patent Owner.

_____

In re *Inter Partes* Review of:
U.S. Patent No. 10,652,111

DECLARATION OF DR. NADER F. MIR UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I. INTRODUCTION 1

A. Professional Background 1

B. Compensation 4

C. Documents and Information Considered 4

D. Summary of Opinions 4

II. LEGAL STANDARDS TO BE APPLIED 5

A. Claim Language 5

B. Anticipation 5

C. Obviousness 6

III. TECHNOLOGY OVERVIEW 9

A. Deep Packet Inspection 9

B. Software-Defined Networking 12

C. '111 Patent Overview 15

D. Prosecution History Overview 19

IV. SUMMARY OF THE PRIOR ART 21

A. Lefebvre Overview 21

B. Chua Overview 26

C. Rash Overview 28

V. A PERSON OF ORDINARY SKILL IN THE ART 29

VI. CLAIM CONSTRUCTION 30

A. The "controller" Term 31

B. The "criterion" Term 32

VII. GROUND 1: CLAIMS 32-36 AND 38-54 ARE RENDERED OBVIOUS BY LEFEBVRE IN VIEW OF CHUA. 33

A. A POSITA Would Have Been Motivated to Combine Lefebvre and Chua. 34

B. Independent Claim 32 43

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:" 43

2. [32.1] "receiving, from the controller, the instruction and the criterion;" 46

3. [32.2] "receiving, from the first entity over the packet network, a packet addressed to the second entity;" 50

4. [32.3] "checking if the packet satisfies the criterion;" 51

5. [32.4] "responsive to the packet not satisfying the criterion, sending over the packet network, the packet to the second entity; and" 52

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." 52

C. Dependent Claims 33-36, 39-54 54

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. 54

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. 57

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. 58

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. 61

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. 62

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. 66

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. 66

8. [42] The method according to claim 41, wherein the addresses are Internet Protocol (IP) addresses. 67

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. 68

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. 71

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. 73

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. 74

13. [47] The method according to claim 45, wherein the communication with the controller is based on, or uses, a standard protocol. 75

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. 75

15. [49] The method according to claim 48, wherein the instruction comprises a Type-Length-Value (TLV) structure. 76

16. [50] The method according to claim 32, wherein the network node comprises a router, a switch, or a bridge. 77

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. 79

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. 79

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. 80

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....81

VIII. GROUND 2: CLAIMS 37-38 ARE RENDERED OBVIOUS BY LEFEBVRE IN VIEW OF CHUA AND RASH. 82

A. A POSITA Would Have Been Motivated to Combine Lefebvre and Rash. 83

B. Dependent Claim 37-38 88

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. 88

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....90

IX. OBJECTIVE INDICIA OF NON-OBVIOUSNESS 91

X. CONCLUSION 92

EXHIBIT LIST[1]

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, *et al.*, "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, *et al.*, "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, *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, No. 2, pp. 69-74 (April 2008)
1014	Steven Levy, "Going with the Flow: Google's Secret Switch to the Next Wave of Networking" (April 17, 2012) (available at

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

Exhibit	Description
	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/pdfrfc/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.

Dependent Claims 33-54	
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.
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 an 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.

I. INTRODUCTION

1. My name is Dr. Nader F. Mir. I understand that I am submitting this declaration in connection with *inter partes* review (IPR) proceedings before the United States Patent and Trademark Office for U.S. Patent No. 10,652,111 ("the '111 Patent"). I have been retained on behalf of Juniper Networks, Inc. ("Juniper") to offer technical opinions with respect to the '111 Patent.

2. Regarding the '111 Patent, I have provided my opinions on whether claims 32-54 ("the Challenged Claims") of the '111 Patent would have been obvious to a person of ordinary skill in the art (POSITA) at the time of the alleged invention in view of the prior art. It is my opinion the Challenged Claims would have been obvious to a POSITA.

A. PROFESSIONAL BACKGROUND

3. My complete qualifications and professional experience are described in my *Curriculum Vitae* (CV) (Ex. 1004). I summarize my relevant qualifications and professional experience below.

4. I have over 30 years of industry and academic experience with computer networks. My areas of research include computer networks and protocols, virtualization, cloud data centers, network security, networking devices including switches, routers, gateways, virtual switches, software-defined networks (SDNs), and network function virtualization devices.

5. I am a Full Professor in the Department of Electrical Engineering at San Jose State University. I joined the faculty at San Jose State University in 2001 and became a Full Professor in 2005. I previously held the positions of Associate Chair in the Department of Electrical Engineering from 2006 to 2008, and Director of the university's off-campus graduate programs from 2006 to 2017. I have supervised more than 200 thesis and master's projects during my career and served as a member of more than twenty other master's and Ph.D. thesis (dissertation) defense committees. I have also served in various positions in the university's management and administration including chairing the Department of Electrical Engineering's Computer Networking Curriculum Committee since 2001 and chairing the Department of Electrical Engineering's governance committee from 2006 to 2009. Prior to joining San Jose State University, I was an Associate Professor in the Department of Electrical and Computer Engineering at the University of Kentucky from 1996 to 2001.

6. I have also worked in industry. From 1985 to 1988, I was a telecom research and development engineer at the Telecommunication Research & Development Center in Surrey, England. There, I led engineering projects in telecommunication engineering including the design of a high-speed switch for digital private branch exchange (PBX) and SS7 networks. I received the best "design/idea" award for this work. I worked as a computer/communications chip-design engineer from 1989 to

1995 at the Computer & Communications Research Center in St. Louis, Missouri. There, I designed VLSI chips including a gigabit switch control plane switch and a Huffman encoder transmitter, both using CMOS technology. And I have served as a consultant in the areas of computer networking technology and communications engineering.

7. I received a Ph.D. in electrical engineering in 1995 and an M.S. degree in electrical engineering, both from Washington University, St. Louis. I received a B.S. in electrical engineering from Polytechnic University in 1985.

8. I am technical editor of IEEE Communications Standards Magazine and have served as technical editor for various journals including IEEE Communications Magazine for fifteen years. As technical editor, I accept or reject articles submitted to the journals. I am a senior IEEE member and have served on numerous technical program committees and steering committees for major IEEE communications and networking conferences.

9. I have authored or co-authored over 100 articles in peer-reviewed journals and conference proceedings. I have published two textbooks including "Computer and Communication Networks" which numerous universities worldwide adopted. I am a named inventor on U.S. Patent No. 7,012,895. I have received numerous awards for my work including university, national, and international awards. And I have received numerous research grants from private and governmental funding

agencies.

10. I have served as an expert witness and technical consultant in numerous matters concerning computer networks and telecommunications.

B. COMPENSATION

11. I am being compensated at my standard consulting rate for my work on this declaration. I am also being compensated for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not dependent on, and I have no financial interest in, the outcome of these proceedings or any related litigation.

C. DOCUMENTS AND INFORMATION CONSIDERED

12. I have reviewed the ’111 Patent, including the claims of the patent in view of the specification. In addition, I have reviewed the ’111 Patent’s prosecution history, the prior art discussed herein, and the remaining exhibits listed herein.

D. SUMMARY OF OPINIONS

13. In my opinion, claims 32-54 of the ’111 Patent are rendered obvious by the prior art. In the remainder of this declaration, I demonstrate that:

- U.S. Patent No. 10,097,452 (“Lefebvre”) in view of U.S. Patent No. 9,264,301 (“Chua”) renders obvious claims 32-36 and 39-54 of the ’111 Patent; and

- Lefebvre in view of Chua and U.S. Patent No. 9,813,447 ("Rash") renders obvious claims 37-38 of the '111 Patent.

II. LEGAL STANDARDS TO BE APPLIED

14. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '111 patent, I am relying on certain basic legal principles that Juniper's counsel has explained to me. Specifically, Juniper's counsel has advised me of the legal concepts that are relevant to IPRs, and I have applied these concepts in reaching my opinions in this declaration.

A. CLAIM LANGUAGE

15. I understand that, during IPRs, patent claim terms are given their ordinary and customary meaning to a POSITA in view of the specification and prosecution history, unless those sources show an intent to depart from such meaning. I understand that the specification includes all the figures, description, and the claims of the patent. And I understand that one way to depart from such a meaning includes a definition for a claim term. In addition, I have been asked to assume that the claims are not indefinite. I discuss specific claim terms in the '111 Patent below.

B. ANTICIPATION

16. I understand that prior art references can anticipate a patent claim under 35 U.S.C. § 102 if each and every element and limitation of the claim is found either

expressly or inherently in a single prior art reference. I also understand that for inherency, a POSITA would recognize the missing descriptive matter would necessarily exist. And I understand that the mere fact that something may result from a given set of circumstances isn't sufficient to establish inherency. I further understand that prior art references can anticipate based on inferences that a POSITA would reasonably expect to draw, in addition to the specific teachings of the prior art references.

17. I understand that a prior art document can disclose a claim feature, and anticipate an alleged invention, if that feature is described in another document that has been incorporated by reference. I also understand that, to incorporate by reference, the host document must identify with detailed particularity what specific material it incorporates, and clearly indicate where that material is found in the incorporated document. I understand that, in making the determination of the extent to which material is incorporated into a host document, the standard of a POSITA should be used to determine whether the host document describes the material to be incorporated by reference with sufficient particularity.

C. OBVIOUSNESS

18. I understand that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the claimed subject matter would have been obvious at the time of the claimed

invention. I understand that a POSITA is presumed to have been aware of all pertinent prior art at the time of invention.

19. In analyzing obviousness, I understand that it is important to consider the scope of the claims, the level of skill in the relevant art, the scope and content of the prior art, and the differences between the prior art and the claims.

20. I understand that criteria for determining if prior art is analogous include (1) whether the prior art is from the same field of endeavor, regardless of the problem addressed, and (2) if the prior art is from a different field of endeavor, whether the prior art still is reasonably pertinent to the particular problem at hand. I understand that the field of endeavor is not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus of a field. And I understand that prior art is reasonably pertinent if it a POSITA would have looked at the prior art to solve the particular problem at hand.

21. When considering the prior art, I understand the prior art should be considered as a whole. When considering a combination of prior art, I understand that I may assess whether there were apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of the prior art, the effects of demands known to the community or present in the marketplace, and/or the background knowledge possessed by a POSITA. I understand that other principles for evaluating whether the obviousness of claimed subject matter

include the following:

- Combining familiar elements according to known methods that does no more than yield predictable results;
- Substituting one known element for another element to yield predictable results;
- Applying a known technique used to improve one device to similar devices when a POSITA would recognize the same technique would improve the similar devices in the same way and when applying the same technique is within the POSITA's skill;
- Applying predictable variations of a device or technology available in one field of endeavor in the same or different field of endeavor when a POSITA could implement the predictable variations once prompted by design incentives or other market forces;
- Choosing from a finite number of identified, predictable solutions with a reasonable expectation of success; and
- Some teaching, suggestion, or motivation in the prior art such as a need or known problem in the field of endeavor that would have led a POSITA to modify the prior art or combine prior art teachings.

22. I understand that obviousness considers what the combined teachings of the prior art would have suggested to a POSITA at the time of the claimed invention

and does not require the physical combination or incorporation of the prior art into each other. I understand that one must avoid using hindsight when evaluating the obviousness of a claimed invention.

23. I also understand that when making the determination as to whether the claimed invention would have been obvious to a POSITA, one must consider any objective indicia of non-obviousness, such as commercial success, long-felt but unsolved needs, failure of others, unexpected results, copying, skepticism of experts, industry praise, significant effort or high cost to develop, near simultaneous invention, and industry acceptance via licensing, if present. For these objective indicia to become relevant to the obviousness inquiry, a causal relationship between the claimed invention and the evidence must exist. This causal relationship must arise from the novelty of the claim and not something already known in the prior art. And I also understand that even when these objective indicia exist with the required causal relationship, they may not overcome a sufficiently strong showing of obviousness.

III. TECHNOLOGY OVERVIEW

24. The following summarizes some of the technology already known in the art well before the ’111 Patent’s earliest filing date, April 22, 2014.

A. DEEP PACKET INSPECTION

25. As network utilization continues to grow, bandwidth misuse and security

concerns are becoming increasingly prevalent and problematic.[2] To help mitigate this misuse, network administrators try to identify the network applications operating on the network. In computer network, an endpoint's application is identified by a port number. One technique for identifying network applications utilizes this port number information. The Internet Assigned Numbers Authority (IANA) maintains a commonly used listing of well-known port numbers.[3] Utilizing this or similar listings, network administrators could classify traffic based on the port numbers.

26. But this technique presumes network applications respect this listing instead of using random network ports.[4] And this deficiency was well-known.[5] Indeed, even in 2005, certain types of traffic sought to mask themselves by using port 80 which HTTP would typically utilize.[6] And as network administrators increasingly utilized port numbers to classify network traffic, newer network applications

---

[2] M. Liao, *et al.*, "Design and Evaluation of Deep Packet Inspection System: A Case Study," IET Networks, pp. 1-8 (2012) ("Liao") (Ex. 1008).

[3] *See* https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml (last accessed April 8, 2024).

[4] Liao at 1; *see also* A. W. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," In: PAM, Springer, vol. 5, pp. 41-54 (2005) ("Moore") (Ex. 1009).

[5] Moore at 2, 10-11 ("[M]al-ware or attack traffic may use the well-known ports of a particular service, thus inflating the amount of traffic attributed to that application.").

[6] Moore at 2.

avoided the use of well-known ports entirely and instead used different, randomly assigned ports.[7] To avoid misclassification, network administrators began implementing content-based classification schemes with different levels of complexity.[8]

27. Deep packet inspection (DPI) is one such content-based technique that inspects both the header and the payload of a packet to classify network traffic.[9] DPI relies on patterns or signatures contained in payloads for network applications. For example, HTTP relies on GET or PUT methods sent from the client to the server. DPI systems may therefore rely on this signature to classify this traffic.[10] Some early DPI systems applied regular expressions and string-matching algorithms like Wu-Manber to classify traffic.[11]

28. Another early DPI system utilized the Netfilter framework contained in the Linux kernel's IP layer.[12] First, the implementers used a core switch's port mirroring function to clone all the packets in the LAN for inspection.[13]

---

[7] Moore at 2.
[8] Moore at 4-8.
[9] '111 Patent at 1:21-29 (describing DPI as one type of dedicated service); Liao at 1.
[10] Liao at 1.
[11] Liao at 2.
[12] *See generally*, Liao.
[13] Liao at 2.

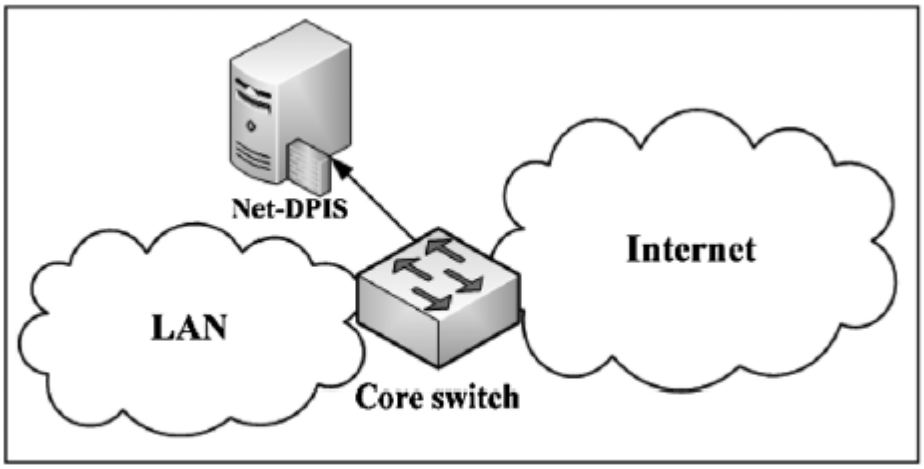


Fig. 3 *Network topology (port mirror)*

29. Then, using the Netfilter framework included as part of the Linux kernel's IP layer, the implementers created a system with a set of four types of matching policies to search for signatures in the packet.[14] With an identified signature, the rule table identifies the service type.[15] The implementers here utilized a personal computer to perform their experiments and found their technique improved on the performance over those using other DPI techniques.[16]

B. SOFTWARE-DEFINED NETWORKING

30. In networking, the concept of a control plane and a data, or forwarding, plane define where certain processes necessary for networking occur. The control plane determines how the forwarding plane should handle forwarding packets from

[14] Liao at 3-4.
[15] Liao at 3-4.
[16] Liao at 5, 7.

one place to another.[17] Data, or forwarding, planes process the packets arriving on inbound interfaces in accordance with the control plane's decisions.[18] The control plane implements management protocols and 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.[19] The data plane handles per-packet processing and performs functions including layer 3 forwarding, metering, shaping, firewalling, NAT, encapsulation, decapsulation, and encryption.[20] Traditionally, both the control and data planes existed in the same hardware device. For example, the Cisco ASR 1000 series routers and Nexus 7000 series switches separated the control and data planes so that active and standby control planes existed to facilitate fault tolerance and transparent software upgrades.[21]

31. But as early as April 2004, there existed a desire to separate these different planes into different physical components for certain applications. The Forwarding and Control Element Separation (ForCES) framework sought to define protocols

---

[17] RFC 3654 at 4-5 (available at https://www.rfc-editor.org/rfc/rfc3654, last accessed at April 4, 2024) ("RFC3654") (Ex. 1010).

[18] RFC3654 at 4-5.

[19] RFC3654 at 4-5.

[20] RFC3654 at 4-5.

[21] *See* W. Xia, *et al.*, "A Survey on Software-Defined Networking," IEEE Communication Surveys & Tutorials, Vol. 17, No. 1, pp. 27-51 (June 13, 2014) (Ex. 1011).

for standardized communication between control and forwarding planes so that the planes could "become physically separated standard components."[22] The authors of ForCES believed this standardization would increase design choices and flexibility, while facilitating more rapid innovation for the control and data plane components.[23]

32. In April 2008, a whitepaper describing OpenFlow was published.[24] The authors recognized a need for programmable networks that would facilitate testing of new network architectures and distributed systems.[25] The authors created OpenFlow to facilitate programming of the flow tables in these devices using common sets of functions shared by many switches and routers.[26] OpenFlow defined a basic set of actions for all OpenFlow devices.[27] OpenFlow controllers remotely control the packet forwarding tables of OpenFlow compliant devices using this defined set of actions.[28]

---

[22] RFC 3746 at 5-6 (available at https://www.rfc-editor.org/in-notes/rfc3746.txt, last accessed April 4, 2024) ("RFC3746") (Ex. 1012).

[23] RFC3746 at 5-6.

[24] N. McKeown, *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, No. 2, pp. 69-74 (April 2008) ("McKeown") (Ex. 1013).

[25] McKeown at 1.

[26] McKeown at 2.

[27] McKeown at 2.

[28] McKeown at 3-4; *see also* U.S. Patent No. 10,097,452 ("Lefebvre") (Ex. 1005) at 4:30-40 (describing OpenFlow SDNs); U.S. Patent No. 9,264,301 ("Chua") (Ex. 1006) at 1:25-41 (same).

33. Using OpenFlow, implementers, including some of the world's largest companies, have created SDNs to support their operations. For example, in 2012, Google publicly disclosed that it used OpenFlow to control its software-defined network (SDN).[29]

C. '111 PATENT OVERVIEW

34. The '111 Patent is titled "Method and System for Deep Packet Inspection in Software Defined Networks." The cover shows the applicants filed a U.S. provisional application on April 22, 2014,[30] and filed the corresponding international application on April 21, 2015.[31]

35. The '111 Patent describes a method for DPI in an SDN where "the method is performed by a central controller of the SDN."[32] Specifically, the **controller 111** instructs **network nodes 112** to determine if incoming packets from **client devices 130** communicating with **destination servers 140** require inspection.[33] The **controller 111** and the **network nodes 112** communicate using standard protocols

[29] S. Levy, "Going with the Flow: Google's Secret Switch to the Next Wave of Networking," (available at https://www.wired.com/2012/04/going-with-the-flow-google/, last accessed April 4, 2024) (Ex. 1014).

[30] '111 Patent at Cover (INID code (60))

[31] '111 Patent at Cover (INID code (22)).

[32] *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.").

[33] '111 Patent at 4:8-11, FIG. 6 (step S610).

such as OpenFlow.[34] The **network nodes 112** redirect or mirror packets for inspection by the **controller 111**.[35] The **controller 111** then provides inspected data to **application servers 120** for further processing associated with security applications, for example.[36]

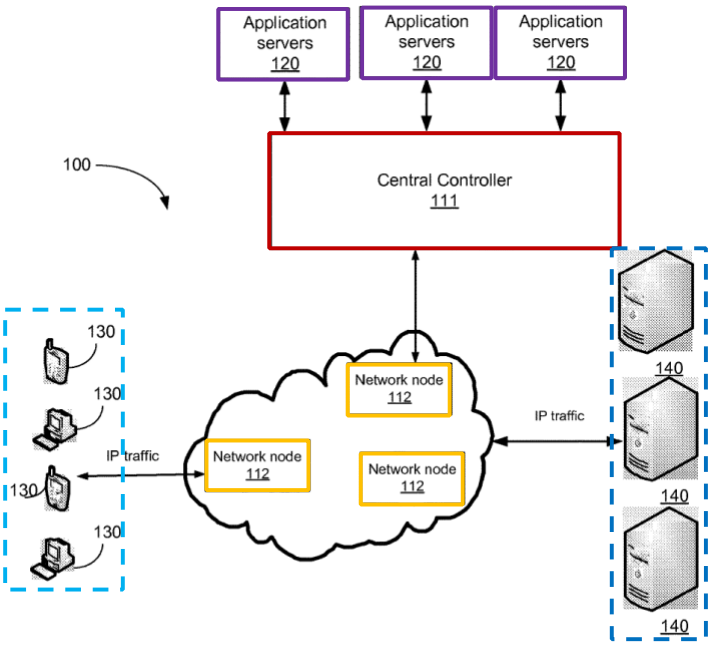


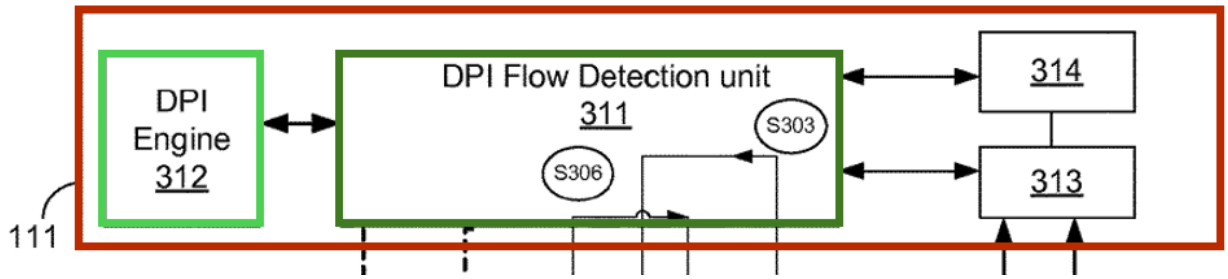
FIG. 1

Annotated '111 Patent FIG. 1

36. The **controller 111** includes memory 313 and processing unit 314.[37] Similarly, **network nodes 112** each include memory 322 and processing unit 323.[38] The processing unit 314 of **controller 111** executes "tasks generally

[34] '111 Patent at 3:44-48.
[35] '111 Patent at 4:15-18.
[36] '111 Patent at 3:56-61.
[37] '111 Patent at 5:33-36.
[38] '111 Patent at 5:50-52.

performed by the central controllers of SDN."[39] The **controller 111 also** includes components such as the **DPI flow detection module 311** and **DPI engine 312**.[40] And similarly, the **network nodes 112** each also include a **probe flow module 321**.[41]



Annotated Excerpt of '111 Patent FIG. 1

37. To perform the '111 Patent's DPI method, the **controller 111** provides instructions to the **network nodes 112**.[42] In an embodiment disclosed by the '111 Patent, when the instructions apply to a packet, the **probe flow module 321** of the **network nodes 112** redirects the packet to the **DPI flow detection module 311** of the **controller 111**.[43] The **DPI flow detection module 311** then executes steps including trapping the packet, creating a new entry in the flow table, and providing the packet for analysis by the **DPI engine 312**.[44] The **DPI flow detection module**

[39] '111 Patent at 5:60-64.
[40] '111 Patent at 5:30-39.
[41] '111 Patent at 5:50-59.
[42] '111 Patent at 6:10-12, 9:40-46, FIG. 6 (step S610).
[43] '111 Patent at 5:50-59, 6:12-19, 6:27-32, 6:41-52, 9:47-61, FIG. 6.
[44] '111 Patent at 6:20-26, 6:33-40, 6:41-52, 8:1-14.

311 may also update the instructions for **network nodes 112**.[45] And after inspection, the **DPI engine 312** of the **controller 111** may provide application metadata to the **application servers 120**.[46] The '111 Patent discloses that implementation of its embodiments as "hardware, firmware, software, or any combination" is possible.[47]

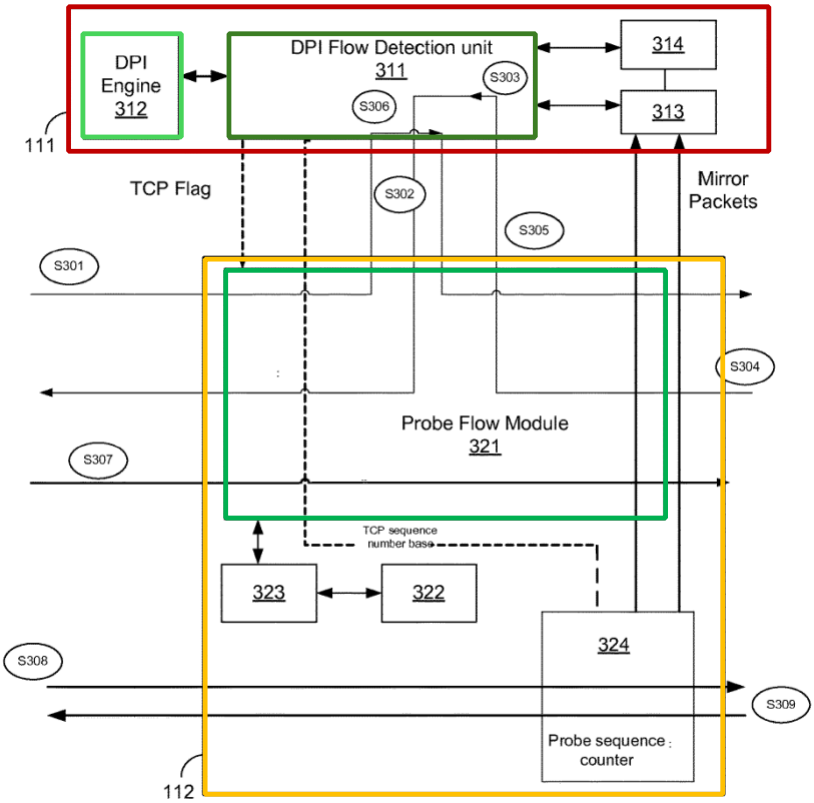


FIG. 3

Annotated Excerpt of '111 Patent FIG. 4

[45] '111 Patent at 7:54-56 ("In an embodiment, a set of mirroring instructions generated respective of the computed mask value are sent to the node 112."), 9:62-10:3, FIG. 6 (step S650).

[46] '111 Patent at 5:36-39.

[47] '111 Patent at 10:15-36.

38. Independent claim 32, which describes the method from the perspective of a network node, recites:

> 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:
>
> receiving, from the controller, the instruction and the criterion;
>
> receiving, from the first entity over the packet network, a packet addressed to the second entity;
>
> checking if the packet satisfies the criterion; responsive to the packet not satisfying the criterion, sending over the packet network, the packet to the second entity; and
>
> 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.

39. Claim 1, the other independent claim, describes the method from the perspective of the packet network. Claims 2-31 depend on independent claim 1 either directly or indirectly. And claims 33-54 depend on independent claim 32 either directly or indirectly.

D. PROSECUTION HISTORY OVERVIEW

40. On April 22, 2014, the applicant filed U.S. Provisional Application

No. 61/982,358 which contained ten pages of figures and descriptions.[48] The applicant filed an International Application on April 21, 2015 that designated the United States, among other countries.[49] And the application entered the U.S. National Stage on September 15, 2016.[50] Upon entry, the applicant presented 54 claims.[51]

41. The Examiner rejected all 54 claims as being unpatentable over U.S. Patent Publication No. 2010/0208590 ("Dolganow") and U.S. Patent Publication No. 2014/0052836 ("Nguyen").[52] The applicant argued against the rejections in its Response,[53] and the Examiner issued a Final Rejection relying on the same unpatentability grounds.[54] The applicant then 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

[48] '111 Patent at Cover (INID code (60)); File History of U.S. Patent No. 10,652,111 ("File History") ("Ex. 1002") at 130-146. As I detail below, all the prior art I consider here predates the April 22, 2014 provisional filing date. I therefore have not considered whether this provisional application provides support for the '111 Patent's claims.

[49] File History at 5 (INID codes (22), (81)).

[50] '111 Patent at Cover (INID code (86)).

[51] File History at 54-63 (claims of the 09/15/2016 Preliminary Amendment).

[52] File History at 271-296 (09/12/2018 Office Action).

[53] File History at 322-330 (12/10/2018 Response).

[54] File History at 331-357 (04/15/2019 Office Action).

is other than the second entity."[55]

42. In the next Office Action, the Examiner rejected all 54 claims as being unpatentable over Dolganow and U.S. Patent Publication No. 2016/0219080 ("Huang").[56] The applicant argued against the rejections in its Response,[57] and the Examiner issued a Notice of Allowance on January 7, 2020, stating that the claims were allowed for the reasons the applicant argued.[58]

43. The '111 Patent issued on May 12, 2020.[59]

IV. SUMMARY OF THE PRIOR ART

44. The following prior art references show that every element of the Challenged Claims was known in the art before April 22, 2014.

A. LEFEBVRE OVERVIEW

45. U.S. Patent No. 10,097,452 ("Lefebvre") is titled "Chaining of Inline Services Using Software Defined Networking."[60] It is my understanding that the U.S. Patent and Trademark Office ("Patent Office") didn't consider Lefebvre during the prosecution of the '111 Patent.

[55] File History at 397-417 (05/07/2019 Amendment).
[56] File History at 425-451 (07/08/2019 Office Action).
[57] File History at 492-501 (10/02/2019 Response).
[58] File History at 504-510 (01/07/2020 Notice of Allowance).
[59] '111 Patent at Cover (INID code (10)).
[60] Lefebvre at Cover (INID code (54)).

46. Lefebvre was filed on July 24, 2012,[61] published on October 17, 2013,[62] and issued on October 9, 2018.[63] I understand that Lefebvre qualifies as prior art under 35 U.S.C. § 102(a)(1)-(2).

47. Lefebvre discloses that network operators deploy various middleboxes or inline services to "inspect and alter network traffic transiting through their network."[64] End users do not notice this inspection and alteration of network traffic, which includes "transparent caching, virus scanning, and deep packet inspection."[65] These services typically exist as dedicated appliances that come with their own costs.[66] As network traffic demands increase, network operators desire the ability to selectively direct traffic to these middleboxes or inline services to minimize these costs.[67] Lefebvre discloses an example where the network operator directs video traffic away from DPI to minimize costs.[68] Lefebvre also discloses that network operators may obtain new revenue sources as users purchase services executed by these middleboxes or inline services.[69]

---

[61] Lefebvre at Cover (INID code (22)).
[62] Lefebvre at Cover (INID code (65)).
[63] Lefebvre at Cover (INID code (45)).
[64] Lefebvre at 1:19-27.
[65] Lefebvre at 1:21-25.
[66] Lefebvre at 1:25-27.
[67] Lefebvre at 1:28-32.
[68] Lefebvre at 1:34-38.
[69] Lefebvre at 1:39-43.

48. Lefebvre discloses that traffic steering is "the action of classifying traffic and directing the different classes of traffic through specific service chains."[70] Service chains are ordered sets of services.[71] Although there exist solutions for implementing traffic steering, each have their own drawbacks. For example, third-party service applications cannot integrate with an extensible router or gateway through additional service cards due to the proprietary nature of these service cards.[72] The scalability of this approach is also limited by the router or gateway's capacity.[73] As another example, routers using policy based routing (PBR) to classify and forward traffic to static service chains do not support defining network policies in a centralized manner and requires each service to configure and classify traffic to the next appropriate service.[74] Static service chains also don't accommodate subscribers each having different sets of subscribed services.[75] And as a final example, services configured to return traffic to routers using PBR after processing forces the router to classify and redirect large amounts of traffic.[76]

49. Instead, Lefebvre discloses how network operators define service policies

[70] Lefebvre at 1:44-46.
[71] Lefebvre at 1:44.
[72] Lefebvre at 1:65-2:5.
[73] Lefebvre at 2:3-5.
[74] Lefebvre at 2:6-11.
[75] Lefebvre at 2:11-17.
[76] Lefebvre at 2:18-21.

specifying traffic classes and the service chains each traffic class traverses.[77] Lefebvre discloses implementing an SDN using a standardized protocol like OpenFlow which includes support for multiple tables and a metadata field for exchanging information between tables.[78] Lefebvre discloses these features help "reduce the number of rules by avoiding cross-products that occur when flattening multi-step classifications."[79]

50. Lefebvre implements traffic steering using a two-step process. First, it classifies incoming packets and assigns a service path based on predefined policies.[80] Then the packets are forwarded to the next service based on the packet's position in the assigned service path.[81] For example, if the assigned service chain includes four different services A, B, C, D, the packet is forwarded to service A, then service B, then service C, and finally service D. Lefebvre's traffic steering process supports three types of policies: subscriber-based, application-based, and flow-based.[82] Lefebvre's subscriber-based policies specify the subscriber's IP address and the set of services each subscriber's traffic should traverse.[83]

---

[77] Lefebvre at 4:41-47.
[78] Lefebvre at 4:30-40.
[79] Lefebvre at 4:37-40.
[80] Lefebvre at 6:17-19, Abstract.
[81] Lefebvre at 6:20-21.
[82] Lefebvre at 6:26-28.
[83] Lefebvre at 6:32-35.

Application-based policies specify either an IP address block and/or a UDP/TCP port on a per application basis, and apply those policies to all subscribers.[84] These refine subscriber-based policies by adding or removing services from the set of services specified in the subscriber-based policies.[85] And flow-based policies are specific to a single flow or IP 5-tuple (i.e., source IP address, destination IP address, protocol, source port, destination port).[86] These dynamically override subscriber-based and application-based policies for specific flows and redirect flows, even mid-flow, to different sets of services.[87]

51. Once an operator defines service policies with traffic classes and service chains, Lefebvre discloses that a centralized controller translates these policies into rules for the programmable switches that steer the network traffic through the service chains.[88] Service instances can exist in multiple service chains and operators can locate and chain these service instances arbitrarily.[89] And steering traffic at finer granularity including subscribers and traffic types is supported.[90]

[84] Lefebvre at 6:38-41.
[85] Lefebvre at 6:41-45.
[86] Lefebvre at 6:46-48.
[87] Lefebvre at 6:48-53.
[88] Lefebvre at 4:43-47.
[89] Lefebvre at 4:48-53.
[90] Lefebvre at 4:53-54.

B. CHUA OVERVIEW

52. U.S. Patent No. 9,264,301 ("Chua") is titled "High Availability for Software Defined Networks."[91] It is my understanding that the Patent Office did not consider Chua during the prosecution of the '111 Patent.

53. Chua was filed on March 15, 2013[92] and issued on February 16, 2016.[93] I understand that Chua qualifies as prior art under 35 U.S.C. § 102(a)(2).

54. Chua discloses that SDNs generally use a standalone controller that "determines the routes and program other devices in the network to behave according to the determinations made by the controller."[94] Physical or virtual instances form the components of the SDN.[95] Protocols including OpenFlow facilitate this ability.[96]

55. Chua discloses that the SDN controller controls the various devices including network devices and service devices.[97] Using data from the service devices, Chua's SDN controller may make network enforcement decisions

---

[91] Chua at Cover (INID code (54)).
[92] Chua at Cover (INID code (22)).
[93] Chua at Cover (INID code (45)).
[94] Chua at 1:25-35; *see also* Chua at 1:45-55 (describing the control plane being "physically separate from the data plane (or forwarding plane) for an SDN).
[95] Chua at 1:52-54.
[96] Chua at 1:35-41, 23:62-24:4 (describing OpenFlow as "an example of an SDN protocol").
[97] Chua at 6:25-28.

including allowing, blocking, mirroring, redirecting, transforming, and encapsulating the packets of the packet flow.[98] Chua also discloses that in these networks, service devices may exist to provide additional functionality, including intrusion detection, intrusion prevention, web proxies, web servers, denial of service (DoS) protection, traffic filtering, malware detection, anti-virus, packet capture and analysis, and honeypots.[99]

56. In addition to controlling the SDN, Chua discloses the controller may execute other programs.[100] Chua discloses that the controller may host additional services, including Layer 4-Layer 7 services, instead of external service devices.[101] Chua also discloses that Layer 3 applications could execute in the SDN.[102] Chua discloses that in such a configuration, the controller may augment these other programs, which are executing on the controller, by continuing to perform its SDN functions as well.[103] In particular, Chua discloses the controller may perform DPI on packets from client devices.[104]

---

[98] Chua at 6:33-54.
[99] Chua at 5:50-6:24.
[100] Chua at 6:36-52.
[101] Chua at 6:42-45.
[102] Chua at 6:45-52.
[103] 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).
[104] Chua at 10:48-52, 31:48-59.

C. RASH OVERVIEW

57. U.S. Patent No. 9,813,447 ("Rash") is titled "Device and Related Method for Establishing Network Policy Based on Applications."[105] It is my understand that the Patent Office did not consider Rash during the prosecution of the '111 Patent.

58. Rash was filed on March 15, 2013,[106] published on September 18, 2014,[107] and issued on November 7, 2017.[108] I understand that Rash qualifies as prior art under 35 U.S.C. § 102(a)(2).

59. Rash discloses a network that adjusts policies associated with the operation of devices on the network.[109] Rash discloses "snooping" of the network traffic flows to identify applications operating on the network.[110] 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."[111] To process this information, Rash discloses a targeted approach where packets, or portions of packets, are mirrored to

[105] Rash at Cover (INID code (54)).
[106] Rash at Cover (INID code (22)).
[107] Rash at Cover (INID code (65)).
[108] Rash at Cover (INID code (45)).
[109] Rash at Abstract.
[110] *E.g.*, Rash at 6:22-40.
[111] Rash at 6:50-57.

a destination other than the original intended destination.[112] Rash discloses that a network control manager controls the mirroring function.[113] Rash discloses the applicability of its dynamic traffic mirroring function for intrusion detection systems and network access control, among other applications.[114] And Rash discloses this dynamic mirroring functionality helps minimize the data mirrored to identification appliances or other infrastructure devices configured to carry out the application identification function.[115]

V. A PERSON OF ORDINARY SKILL IN THE ART

60. It is my understanding that I must analyze and apply the prior art cited above from the perspective of a POSITA as of the time of the invention, which I understand is April 22, 2014 for purposes of this IPR proceeding. When forming my opinions herein, I analyzed and applied the prior art and claim constructions from the perspective of a POSITA as of the time of the invention.

61. The ’111 Patent relates to a method for DPI in an SDN.[116] It is my opinion that a POSITA for purposes of the ’111 Patent would have held at least a bachelor’s degree in computer science, electrical engineering, computer

---

[112] *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.

[113] Rash at 9:12-15, 29:46-30:17.

[114] Rash at 10:29-56.

[115] Rash at 12:59-13:8, 34:43-51.

[116] *E.g.*, ’111 Patent at Abstract.

engineering, or a closely related field, and two or more years of professional experience in packet-switched networking systems. Alternatively, a POSITA would have a master's degree or similar post-graduate work in computer science, electrical engineering, computer engineering, or a closely related field, and less professional experience.

62. I am well aware of the qualifications of such a skilled artisan because I have worked with, supervised, and hired computer scientists and engineers with similar capabilities.

63. Thus, I am familiar with the understanding and knowledge of persons of ordinary skill in the art as of 2014 and was at least as qualified as the POSITA that I have identified above in 2014. Thus, I understand the perspective of a POSITA in 2014, which I have applied in my analysis.

VI. CLAIM CONSTRUCTION

64. For IPR petitions filed after November 13, 2018, I understand the Patent Trial and Appeal Board ("Board") construes claim terms as having their ordinary and customary meaning to a POSITA at the time of invention (i.e., the same standard used in district court). I have applied this standard in rendering my opinions on claim construction. In addition, I have been asked to assume that the claims are not indefinite.

A. THE "CONTROLLER" TERM

65. Independent claim 32 includes the term "controller." It is my opinion that a POSITA would construe the term "controller" to mean "an entity configured to perform deep packet inspection on packets."[117] The '111 Patent repeatedly states how DPI "is performed by a central controller of the SDN."[118] The '111 Patent explains the central controller 111 includes a DPI flow detection module 311 and a DPI engine 312 in addition to the memory 313 and processing unit 314 that performs the tasks generally performed by SDN central controllers.[119] The DPI detection flow module 311 and the DPI engine 312 perform the various DPI functions of the '111 Patent's controller.[120]

66. I understand that in a prior district court case involving the '111 Patent, but that did not involve Juniper, the district court preliminarily construed the

---

[117] I understand that this construction is consistent with that previously offered by Cisco Systems, Inc., an unrelated petitioner, in IPR2023-00554. I also understand that Juniper filed a petition and a motion for joinder in IPR2024-00037 that was granted by the Board.

[118] '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 thereof is sent to the controller 111."), 4:49-50 ("The mirrored bytes are sent to the controller 111 to perform the DPI analysis.").

[119] '111 Patent at 5:30-39, 5:60-64.

[120] '111 Patent at 5:40-59.

"controller" term to mean "an entity that is capable of controlling deep packet inspection." I understand that Patent Owner has advocated for this construction of "controller" in previously filed proceedings. I further understand that the parties jointly stipulated to dismissal of that district court case before the Court issued a decision. I have considered this construction of "controller" but do not believe it conforms to the '111 Patent's disclosure as closely as the construction I offer above. For example, the '111 Patent never uses the term "capable" when describing the controller's functions. Instead, the patent states that "the central controller 111 is **configured** to perform deep packet inspection."[121]

67. Regardless, under either construction of "controller," the Challenged Claims are unpatentable over the prior art as I explain below.

B. THE "CRITERION" TERM

68. The term "criterion" does not appear in the '111 Patent's specification. But the '111 Patent discloses how "each network node 112 is configured to determine if an incoming packet requires inspection or not," and that the "determination is performed based on a set of instructions provided by the controller 111."[122] As a result of this set of instructions, the packet "is either redirected to the controller 111 or mirrored and a copy thereof is sent to the controller 111" without

[121] '111 Patent at 4:5-6 (emphasis added).
[122] '111 Patent at 4:12-15.

alteration.[123] Further, the '111 Patent includes a series of examples of probe instructions that consider various TCP flags such as the SYNC, ACK, FIN, and RST flags.[124] The '111 Patent then states that "each node 112 is configured to receive an incoming packet (either a request from a client device 130 or response from a server 140), analyze the packet's header, and perform the action (redirect the packet to controller 111 or send to destination server 140) respective of the value of the TCP flag."[125] Given this description, it is my opinion that a POSITA would interpret "criterion" to mean "the conditions, or basis, used to identify packets for actions, such as inspection."

VII. GROUND 1: CLAIMS 32-36 AND 38-54 ARE RENDERED OBVIOUS BY LEFEBVRE IN VIEW OF CHUA.

69. The Challenged Claims under this ground include independent claim 32 and claims 33-36 and 38-54, which depend directly or indirectly from claim 32. As I detail below, a POSITA would have recognized that each and every limitation of claims 32-36 and 38-54 are rendered obvious by Lefebvre in view of Chua.

70. As I discussed above in Section III, well before 2014, both SDN and DPI were well-known and ubiquitous. For example, both Lefebvre and Chua discuss

[123] '111 Patent at 4:15-19, 4:61-5:16 (describing type-length-value structures for parsing and identifying packet fields).
[124] '111 Patent at 4:26-41.
[125] '111 Patent at 4:39-46.

SDN and DPI.[126] Lefebvre and Chua both disclose how DPI often executes in dedicated service devices.[127] And Chua specifically discloses the SDN controller performing DPI as well.[128]

71. As I describe below, a POSITA would have found that Lefebvre in view of Chua renders obvious, either expressly or inherently, each and every element of claims 32-36 and 38-54.

A. A POSITA WOULD HAVE BEEN MOTIVATED TO COMBINE LEFEBVRE AND CHUA.

72. It is my opinion that a POSITA would have been motivated and found it obvious to modify Lefebvre with Chua's teachings. Lefebvre discloses techniques for an SDN to steer traffic through inline services.[129] Lefebvre discloses how operators can define the traffic classes and service chains for those classes for processing traffic flows on a per packet basis.[130] 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.[131] Lefebvre discloses that

[126] Lefebvre at 1:21-25, 1:34-38; Chua at 10:48-52, 31:48-59.
[127] Lefebvre at 1:21-25, 1:34-38; Chua at 31:54-59 (describing DPI service devices).
[128] Chua at 31:48-54.
[129] Lefebvre at 1:13-21, 2:41-47.
[130] Lefebvre at 4:41-43.
[131] Lefebvre at 4:41-47.

at least some of the switches in the SDN use OpenFlow.[132] Once defined by the operator and programmed into the switches, Lefebvre discloses that traffic steering requires classification of incoming packets and assignment of a service path based on predefined policies.[133] Then the packets are forwarded to the next service in the assigned service path.[134]

73. Lefebvre identifies several types of services that could form part of the service chain, including DPI.[135] Regarding DPI, Lefebvre discloses "[t]hese services are usually packaged and sold as dedicated appliance (either physical or virtual) and are often expensive."[136] One reason Lefebvre offers for implementing traffic steering is to direct traffic that does not require inspection away from expensive services like DPI, reducing the need to invest in new service appliances.[137] Lefebvre discloses that service nodes, separate from the controller, form the service chain.[138] Lefebvre therefore controls the usage of DPI services by routing network traffic through the service only as appropriate, where the services are separate from the controller.

---

[132] *See* Lefebvre at 5:41-44.
[133] Lefebvre at 6:17-19.
[134] Lefebvre at 6:20-21.
[135] Lefebvre at 1:28-38, 4:48-54.
[136] Lefebvre at 1:25-27, 1:34-38.
[137] Lefebvre at 1:32-38.
[138] *See* Lefebvre at 6:28-31, FIG. 1.

74. Like Lefebvre, Chua discloses using an SDN where a controller determines the behavior of devices in the network and how traffic flows through the network.[139] Like Lefebvre, Chua also discusses OpenFlow. Specifically, Chua discloses protocols including OpenFlow that controls the data-plane devices of the network.[140] These network devices provide data to Chua's controller, which then makes network enforcement decisions.[141] Chua's controller considers the traffic on a per-packet or per-flow basis as necessary.[142] Chua's controller then programs network devices of the SDN to implement these decisions.[143] Chua's service devices include those that execute DPI.[144] Chua also discloses the SDN controller execute other programs, including Layer 4-Layer 7 services, instead of external service devices.[145] And Chua discloses the controller performing DPI.[146]

75. A POSITA would have been motivated and found it obvious to modify Lefebvre with Chua's teachings because both describe SDNs, and both describe

---

[139] Chua at 1:25-35.

[140] Chua at 1:35-41.

[141] Chua at 6:25-28.

[142] Chua at 7:33-54, 10:48-52, 15:9-17.

[143] Chua at 7:33-54 (describing different programmed actions taken "on packets of a packet flow"), 31:30-34.

[144] Chua at 10:48-52, 15:4-9, 16:30-33, 31:48-50, Fig. 9 (step 402).

[145] 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).

[146] Chua at 10:48-52, 15:4-9, 16:30-33, 31:48-50.

DPI as one type of service in an SDN.[147] And both Lefebvre and Chua disclose use of OpenFlow devices.[148] Lefebvre and Chua are therefore in the same field of endeavor and contain overlapping disclosures with similar purposes.

76. Modifying Lefebvre in view of Chua's teachings would have been obvious to a POSITA. Lefebvre has three different levels of policies which have different levels of granularity. Lefebvre's subscriber-based policies rely on the subscriber's IP address.[149] 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.[150] These application-based policies refine subscriber-based policies by adding/removing services from the services specified in subscriber-based policies.[151] Both Lefebvre's subscriber-based and application-based policies operate on the packets being received by the SDN. And Lefebvre's flow-based policies specify the flow or IP 5-tuple (i.e., source IP address, designation IP address, protocol, source port, and destination port).[152] Similarly, Chua discloses that the SDN determines "the appropriate type of flow troubleshooting that

[147] Lefebvre at 1:28-38; Chua at 10:48-52, 15:4-9, 16:30-33, 31:48-50, Fig. 9 (step 402).
[148] Lefebvre at 5:41-44.
[149] Lefebvre at 6:32-35.
[150] Lefebvre at 6:38-41.
[151] Lefebvre at 6:42-45.
[152] Lefebvre at 6:46-48.

includes the appropriate IP ranges to watch on, and the TCP/UDP services that the flowchain is meant to service."[153] And Chua discloses that the SDN controller performs DPI "on packets from client device[s]."[154] Lefebvre and Chua therefore share similar techniques for classifying the traffic traversing their networks.

77. A POSITA would have 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. Lefebvre mentions DPI as one type of known service,[155] but focuses its disclosure on actions relevant to traffic steering and the passing of the packet though assigned service chains including redirecting or dropping the packet.[156] Lefebvre does not expressly disclose specifics regarding actions resulting from the services processing the traffic along the service path.[157] When implementing DPI in an SDN like Lefebvre's, a POSITA would have needed additional disclosures regarding actions taken after DPI identifies malicious traffic. Chua discloses these details regarding actions an SDN controller may take in response to the information provided by

[153] Chua at 18:9-16.

[154] 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).

[155] Lefebvre at 6:17-31, 1:28-38 (discussing DPI).

[156] Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

[157] *See generally,* Lefebvre.

security-related services like DPI, including allowing, blocking, mirroring, redirecting, transforming, or encapsulating the network traffic.[158] A POSITA would therefore have looked to Chua's more detailed teachings regarding actions an SDN controller takes in response to information from services like DPI and apply them to Lefebvre when implementing DPI in the SDN.

78. Chua also discloses that the controller may execute other programs instead of external service devices.[159] A POSITA would have been motivated to modify Lefebvre with this teaching from Chua for multiple reasons. Lefebvre discloses that services including DPI "are usually packaged and sold as dedicated appliances (either physical or virtual) and are often expensive."[160] Chua, however, discloses programming an SDN controller to perform DPI.[161] A POSITA would recognize Chua's teaching to mean that performing DPI is not an expensive task that only dedicated appliances can execute, but a task that an SDN controller could execute. Incorporating DPI in Lefebvre's controller, as Chua teaches, would remove the need for the expensive dedicated appliance that Lefebvre discloses existed in the prior art. A POSITA would have also been motivated to modify Lefebvre with Chua's programmable controller teachings because Chua discloses how programs

[158] Chua at 7:33-54.
[159] Chua at 6:42-45.
[160] Lefebvre at 1:25-27, 1:34-38.
[161] Chua at 31:48-50.

executing on the controller may utilize the SDN controller's existing abilities to implement appropriate updates to the SDN to facilitate the programs' functionality.[162] And including within the controller the service device's functionality, as Chua teaches, would remove the need to address network failures between endpoints that Chua states is a "complex problem," and that would also apply to network failures between network devices, the SDN controller, and the DPI service device.[163] Chua's controller teachings therefore help simplify the SDN and reduce the need to address certain types of failures in the SDN. A POSITA would therefore have motivation to consider Chua's teachings.

79. Implementing Chua's teachings in Lefebvre's controller would have been well within the skill of a POSITA, and the POSITA would have had a reasonable expectation of success. Lefebvre already discloses redirecting packets to the appropriate service.[164] Chua discloses additional actions an SDN controller may take in response to the information provided by services such DPI like allowing, blocking, mirroring, redirecting, transforming, or encapsulating the network traffic.[165] Implementing these additional actions would have been straightforward

[162] Chua at 6:36-52.
[163] *See* Chua at 19:1-53 (describing redundancy possibilities for SDN appliances).
[164] Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).
[165] Chua at 7:28-54.

and simple for a POSITA because these additional actions taught by Chua would simply build on standard actions many networking devices already implement, and Lefebvre already discloses redirecting of packets.[166] For example, in the paper proposing OpenFlow, the authors stated that OpenFlow switches support basic actions including forwarding, encapsulating, and dropping packets.[167]

80. Implementing a DPI service in Lefebvre's controller, as Chua teaches, would also have been straightforward and simple for a POSITA. Although Lefebvre omits detailed discussion of its controller, it is my opinion that a POSITA would recognize that Lefebvre's controller includes components including processors and memory. Chua confirms how controllers include a control unit with "one or more processors and/or computer readable memory for storing software or firmware instructions."[168] And Lefebvre discloses that embodiments of what it describes may exist as a software product executing programs.[169] Similarly, Chua discloses its controller is programmable when describing how "a software development kit (SDK) may be provided for creating programs that can be executed" by the SDN controller.[170] An SDK helps a programmer create

---

[166] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).
[167] McKeown at 3.
[168] Chua at 10:67-11:6, 31:60-32:6.
[169] Lefebvre at 17:17-37.
[170] Chua at 6:38-40.

applications by providing application programming interfaces (APIs) tailored for a specific hardware platform, operating system, or combination of both. SDKs may also include debugging tools or other tools to help programmers when creating applications. As an example, Chua discloses how an SDN controller performs DPI on packets from a client device.[171] It is therefore my opinion that nothing more than the routine skill of a POSITA would be required to implement Chua's teachings in Lefebvre.

81. And a POSITA would have had a reasonable expectation of success in modifying Lefebvre with Chua's controller teachings. Lefebvre discloses that embodiments of what it describes may exist as a software product executing programs.[172] This confirms my prior opinion that a POSITA would recognize that Lefebvre's controller includes components including processors and memory. Chua discloses how, by using the SDK, users may create "one or more L4-L7 services," or an L3 application, that execute on the controller "rather than" on service devices.[173] Chua also discloses how the programs may use features of the SDN controller, including commanding network equipment to update forwarding tables.[174] And as an example, Chua discloses how an SDN controller performs DPI

[171] Chua at 31:49-54.
[172] Lefebvre at 17:17-37.
[173] Chua at 6:37-47.
[174] Chua at 6:47-52.

on packets from a client device.[175] Because Lefebvre discloses how its embodiments exist as software programs, and Chua discloses the inclusion of an SDK and example programs executing on its controller including DPI, it is my opinion that a POSITA would have had a reasonable expectation of success in implementing a DPI service in Lefebvre's controller, as Chua teaches.

B. INDEPENDENT CLAIM 32

82. It is my opinion that Lefebvre in view of Chua renders obvious independent claim 32.

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:"

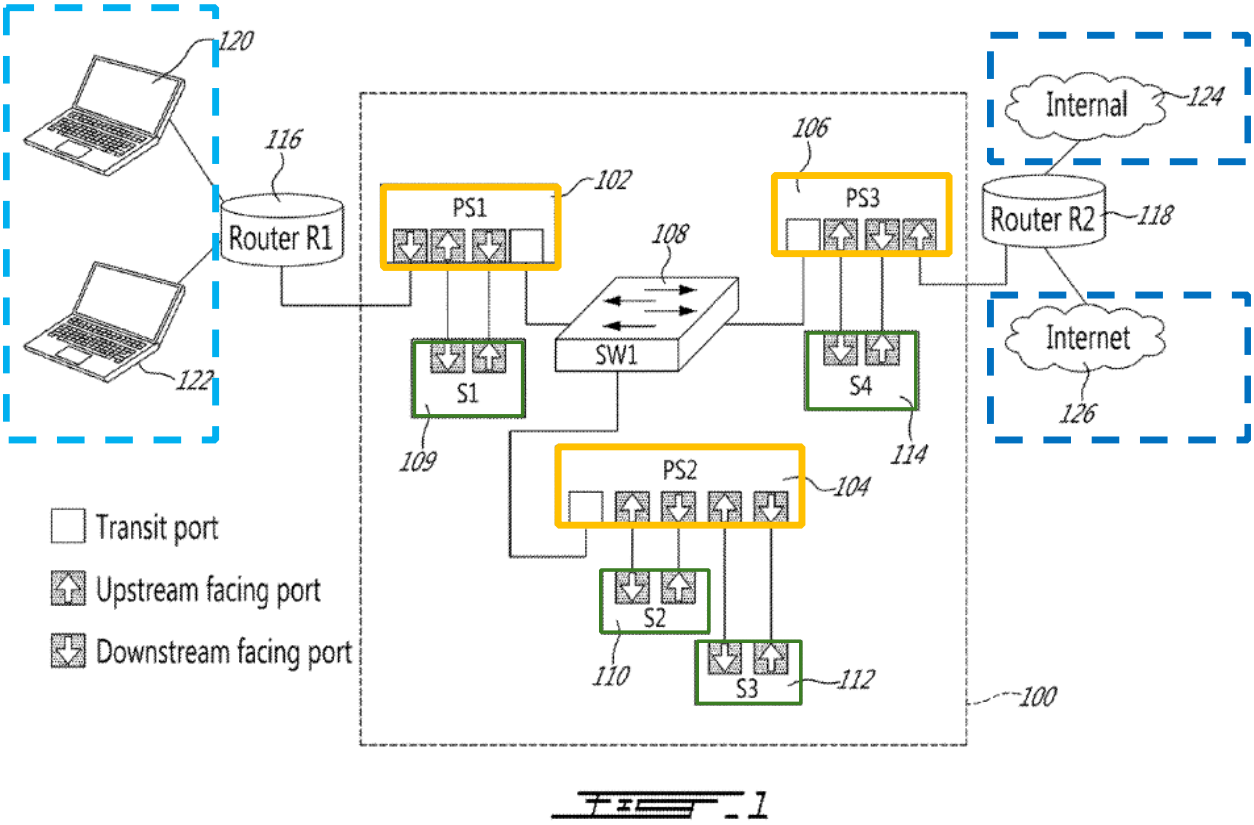
83. The combination of Lefebvre and Chua renders obvious claim element [32.preamble] to the extent it limits the scope of claim 32.

84. Lefebvre discloses how operators "define service policies that specify traffic classes that each class must traverse" and how the "policies are translated by the controller into rules that are programmed on the switches in the service network."[176] Lefebvre discloses that the traffic steering process supports three types of service policies: subscriber-based policies, application-based policies, and

[175] Chua at 31:49-54.

[176] Lefebvre at 2:41-47; *see also* Lefebvre at 1:13-21.

flow-based policies.[177] Below, I annotated FIG. 1 of Lefebvre to highlight certain components.



Annotated Lefebvre FIG. 1

85. Lefebvre discloses that incoming traffic "always enters the service network 100 via a perimeter switch and through a node port,"[178] and that arriving packets are processed and steered to the next node in the assigned service paths.[179] Lefebvre discloses **user equipment 120, 122** connected to router R1 and the

[177] Lefebvre at 6:26-28.
[178] Lefebvre at 6:5-8.
[179] Lefebvre at 6:8-11.

internal network 124 and the **Internet 126** connected to router R2.[180] Lefebvre's user equipment, internal network, and Internet therefore disclose the "packet network" and "first and second entities."

86. Lefebvre's service network includes **perimeter switches 102, 104, 106** and an inner switch SW1.[181] The network also includes **services 109, 110, 112, 114**. Lefebvre discloses that the **perimeter switches 102, 104, 106** may be OpenFlow switches whereas the inner switch 108 may be "either an OpenFlow switch or a plain Ethernet switch."[182] Each of Lefebvre's **perimeter switches 102, 104, 106** therefore disclose the "network node" of the claims because they, like the '111 Patent's network nodes, communicate with the central controller using OpenFlow.[183]

87. And even though Lefebvre states that the centralized controller is "not shown in FIG. 1,"[184] Lefebvre discloses a centralized controller pushes these policies to the relevant switches.[185] A POSITA would therefore understand that Lefebvre's controller is external to the **perimeter switches 102, 104, 106**.

88. And as discussed in Section VII.A, a POSITA would have been motivated

---

[180] Lefebvre at 5:38-54, 6:13-16.
[181] Lefebvre at 5:38-44.
[182] Lefebvre at 5:41-44.
[183] *See* '111 Patent at 3:47-48.
[184] Lefebvre at 6:28-31.
[185] Lefebvre at 6:28-31.

and found it obvious to apply Chua's teachings to Lefebvre's SDN controller. Lefebvre's modified controller would then be programmed to execute DPI, as Chua teaches. As I explained in Section VII.A, because Lefebvre discloses embodiments that already redirect packets to appropriate services and embodiments existing as software programs, a POSITA would have had a reasonable expectation in success in implementing Chua's teachings in Lefebvre's controller. And as a result of this modification, a POSITA would recognize that Lefebvre's modified controller would disclose the claimed "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 VI.A.

89. Thus, the combination of Lefebvre and Chua renders obvious claim element [32.preamble].

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

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

91. The '111 Patent uses the word "instruction" in its specification to describe certain commands including "probe instructions," "mirroring instructions," and

"termination instructions."[186] As discussed in Section VI.B, the term "criterion" does not appear in the '111 Patent's specification, and there is no prior usage of the term in claim 32. And as discussed in Section VI.B, it is my opinion that a POSITA would interpret "criterion" to mean the conditions, or basis, used to identify packets for actions, such as inspection.

92. Lefebvre discloses that the operator defines "service policies that specify traffic classes and the chain of services that each class must traverse."[187] Lefebvre discloses that the controller programs the rules into the switches of the service network, and the "rules steer the network traffic through the ordered chain of services as specified by the policies."[188] And Lefebvre discloses that traffic steering is a two-step process.[189] The first step "classifies incoming packets and assigns them a service path based on predefined policies."[190] And the second step "forwards packets to a 'next' service based on its current position along its assigned service path."[191]

---

[186] *E.g.*, '111 Patent at 4:23-26; *see also* '111 Patent at 4:26-46 (describing redirecting, forwarding, and terminating instructions), 4:47-56 (describing mirroring instructions).

[187] Lefebvre at 4:43-45.

[188] Lefebvre at 4:43-47, Table 1, Table 2 (depicting OpenFlow switch table actions), FIG. 5.

[189] Lefebvre at 6:17-18.

[190] Lefebvre at 6:18-19, 7:35-8:48 (describing "an example methodology for processing and classifying a packet), FIG. 2, FIG. 5 (step 510).

[191] Lefebvre at 6:20-21, FIG. 5 (step 530).

93. A POSITA would have understood that Lefebvre's traffic steering involving classification of incoming packets to assign them a service path discloses the claimed criterion. Lefebvre's traffic steering policies include subscriber-based policies, application-based policies, and flow-based policies.[192] Lefebvre's traffic steering mechanism presumes that all traffic traversing the service network is subscriber traffic.[193] But as Lefebvre also discloses, when the traffic is not subscriber traffic, the SDN functions as a simple router that "does conventional destination-IP-based forwarding" where the traffic does not traverse inline services.[194] Lefebvre also classifies traffic based on the incoming port, the packet's destination MAC address, the source or 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.[195] These classification techniques disclosed by Lefebvre disclose the claimed criterion because they are the conditions, or basis, used to identify packets for actions. Lefebvre discloses the classification decides which service path the packet will traverse. Lefebvre's next destination table routes the packet to

[192] Lefebvre at 6:26-31.
[193] Lefebvre at 5:19-37.
[194] Lefebvre at 14:23-33.
[195] Lefebvre at 7:35-8:48 (describing "an example methodology for processing and classifying a packet), Table 1, Table 2 (depicting OpenFlow switch table actions).

the appropriate service.[196] 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.[197]

94. A POSITA would have also understood that Lefebvre's traffic steering involving classification of incoming packets to assign them a service path discloses the claimed instruction. Lefebvre's final table, the "next destination table," matches the packet to the appropriate service.[198] Like the '111 Patent's instruction, Lefebvre's final table redirects the packet to the appropriate service. Lefebvre's controller therefore discloses the claimed criterion.

95. And Lefebvre's controller, as modified by Chua's teachings, similarly discloses the claimed controller providing the instruction and criterion. Chua discloses the controller programs network devices "to perform programmed actions on packets" including allowing, blocking, mirroring, redirecting, transforming, and encapsulating packets.[199] Lefebvre's controller already redirects.[200] As modified by Chua's teachings, Lefebvre's controller's instructions would include the allowing,

---

[196] Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

[197] *See* '111 Patent at 4:12-46.

[198] Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

[199] Chua at 7:28-54.

[200] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

blocking, and mirroring packets.

96. Thus, the combination of Lefebvre and Chua discloses claim element [32.1].

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

97. Lefebvre discloses claim element [32.2].

98. As discussed for claim element [32.preamble], Lefebvre states that incoming traffic "always enters the service network 100 via a perimeter switch and through a node port."[201] Lefebvre discloses **user equipment 120, 122** and the **internal network 124** along with the Internet 126. Lefebvre's **user equipment**, **internal network**, and Internet therefore disclose the "first and second entities." For example, when Lefebvre's **user equipment 120, 122** communicates with the **internal network 124** using the service network 100, the **user equipment 120, 122** discloses the first entity.[202] And a POSITA would have understood that in doing so, the SDN receives from the **user equipment 120, 122** a packet addressed to the **internal network 124**. Lefebvre confirms this when disclosing the source and destination address fields of the received packet.[203] And similarly, Lefebvre discloses that within the network, at transit ports, packets are forwarded to their

[201] Lefebvre at 6:5-8.
[202] Lefebvre at 5:38-54, 6:8-16.
[203] Lefebvre at 2:53-64.

destination MAC addresses.[204]

99. Thus, Lefebvre discloses claim element [32.2].

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

100. Lefebvre discloses claim element [32.3].

101. 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,[205] and that the first step of traffic steering involve the classification of incoming packets.[206] And as discussed for claim element [32.1], Lefebvre's classification techniques disclose the claimed criterion because they are the conditions, or basis, used to identify packets for actions.[207]

102. A POSITA would have understood Lefebvre's classification of incoming packets discloses the claimed "checking if the packet satisfies the criterion." Lefebvre's classification checks the incoming packet against the claimed criterion and determines the service path the packet will traverse.

103. Thus, Lefebvre discloses claim element [32.3].

---

[204] Lefebvre at 5:65-6:12, 7:55-58.

[205] Lefebvre at 4:43-45.

[206] Lefebvre at 6:18-19, 7:35-8:48 (describing "an example methodology for processing and classifying a packet"), FIG. 2, FIG. 5 (step 510).

[207] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

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

104. Lefebvre discloses claim element [32.4].

105. Lefebvre states that one of its assumptions used when describing its example embodiments is that the forwarded traffic is subscriber traffic.[208] As discussed for claim element [32.1], Lefebvre's distinction between subscriber and non-subscriber traffic is one type of criterion. Lefebvre also discloses that "[f]or non-subscriber traffic, we can treat the steering network as a simple router" and can send packets using "conventional destination-IP-based forwarding."[209] Because Lefebvre's non-subscriber traffic does not 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.

106. Thus, Lefebvre discloses claim element [32.4].

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."

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

108. As discussed for claim element [32.1], Lefebvre's classification techniques determine which service chain the traffic traverses. Lefebvre's next destination

[208] Lefebvre at 14:23-25.
[209] Lefebvre at 14:25-29.

table routes, or redirects, the packet to the appropriate service for the classification of traffic.[210] A POSITA would have understood that Lefebvre's services are a destination different from the intended destination of the packet.[211]

109. Further, as I explained in Section VII.A, Lefebvre's controller modified by Chua's teachings would include additional actions including mirroring or redirecting the packets to the service chain. Chua's mirroring action allows the traffic and also sends a copy of the traffic for deeper inspection or recording.[212] And a POSITA would have understood that both actions, mirroring and redirecting of the packets to the service chain, send the packet to an entity that is other than the second entity. As I explained in Section VII.A, a POSITA would have been motivated to modify Lefebvre to include these additional actions taught by Chua. 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 would also be a destination "other than the second entity."

110. Thus, combination of Lefebvre and Chua renders obvious claim element [32.5].

---

[210] Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

[211] *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").

[212] Chua at 7:43-45, 28:7-32.

C. DEPENDENT CLAIMS 33-36, 39-54

111. It is my opinion that Lefebvre in view of Chua renders obvious dependent claims 33-36 and 39-54.

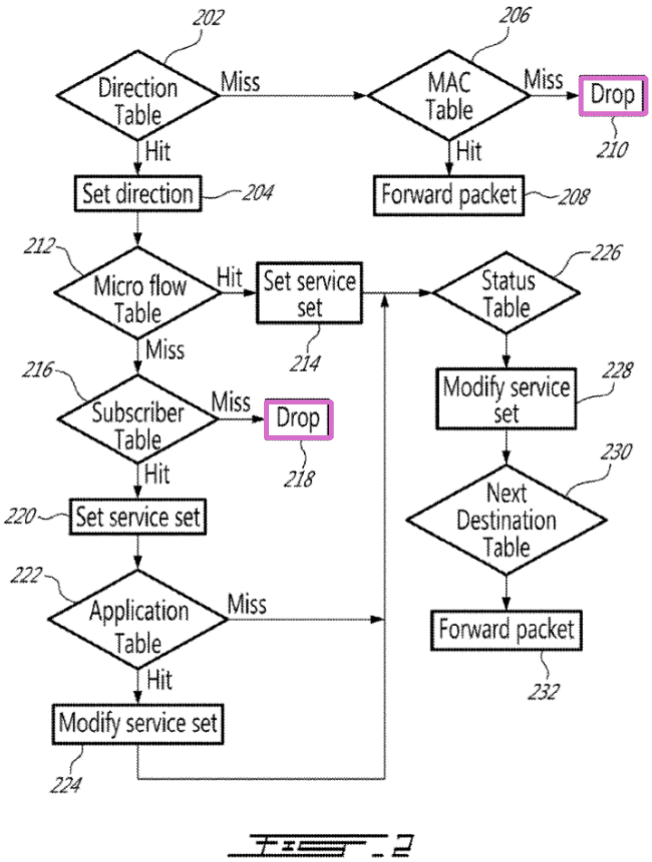
1. [33] The method according to claim 32, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction, and upon receiving the 'terminate' instruction, the method further comprising blocking, the packet from being sent to the second entity and to the controller.

112. The combination of Lefebvre and Chua renders obvious dependent claim 33.

113. As discussed for claim element [32.1], Lefebvre's traffic steering policies include subscriber-based policies, application-based policies, and flow-based policies.[213] FIG. 2 of Lefebvre depicts "an example methodology for processing and classifying a packet."[214]

[213] Lefebvre at 6:26-31.

[214] Lefebvre at 7:37-39.



Annotated FIG. 2 of Lefebvre

114. In the example embodiments, Lefebvre discloses that when a packet arrives on the a transit port, it will "miss" the direction table and proceed to the MAC table 206.[215] If the packet's destination MAC address doesn't match the contents of the MAC table, Lefebvre discloses dropping the packet.[216] Similarly, Lefebvre discloses "the default action is to drop the packet" when there is a miss in the subscriber table.[217] Lefebvre therefore discloses dropping the packet, but Lefebvre

[215] Lefebvre at 7:52-54.
[216] Lefebvre at 7:55-58.
[217] Lefebvre at 8:11-13, FIG. 2 (step 218).

does not explicitly disclose a terminate instruction that results in blocking, as claim 33 requires.

115. 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 may perform these actions, including blocking, in response to information from service devices,[220] external triggers or events,[221] or business-level security rules.[222] The combination of Lefebvre and Chua would therefore include instruction resulting in blocking, which a POSITA would have recognized as the terminate instruction required by claim 33. As discussed in Section VII.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's actions including blocking.[223] And in doing so, the modified Lefebvre controller would provides the instruction, which Chua teaches, to the network devices directing when to apply dropping the packet, which Lefebvre

[218] Chua at 7:28-54.
[219] Chua at 7:35-38, 7:41-42.
[220] Chua at 7:28-38, 16:23-44, 28:7-32.
[221] Chua at 8:43-47, 17:26-35.
[222] Chua at 14:39-42.
[223] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller).

discloses. In doing so, the network devices in Lefebvre's SDN would block traffic identified by Lefebvre's controller, which executes DPI as taught by Chua.[224]

116. The combination of Lefebvre and Chua therefore renders obvious dependent claim 33.

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.

117. The combination of Lefebvre and Chua renders obvious dependent claim 34.

118. Lefebvre does not disclose a mirroring instruction. 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.[225] Chua's mirroring action allows the traffic and also sends a copy of the traffic for deeper inspection or recording.[226] Chua's SDN controller 112 may perform these actions, including mirroring, in response to information from service devices[227] or business-level security rules.[228] And as discussed for claim element [32.5], the combination of Lefebvre and Chua renders

[224] *See* Chua at 31:49-54.
[225] Chua at 7:28-54.
[226] Chua at 7:43-45, 28:7-32.
[227] Chua at 7:28-38, 16:23-30, 16:36-44, 28:7-32.
[228] Chua at 14:39-42.

obvious the claimed "responsive to the packet satisfying the criterion" aspect of claim 34.

119. As discussed in Section VII.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's actions including mirroring.[229] By doing so, a POSITA would have understood the network devices in Lefebvre's SDN would send the packet to Lefebvre's modified controller as well as the destination. And Lefebvre's controller would execute DPI, as taught by Chua.[230]

120. The combination of Lefebvre and Chua therefore renders obvious dependent claim 34.

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.

121. The combination of Lefebvre and Chua renders obvious dependent claim 35.

122. 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

[229] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller).
[230] *See* Chua at 31:49-54.

redirecting the packet to the destination.[231]

123. Lefebvre acknowledges the need to selectively steer traffic away from expensive services, such as DPI,[232] but does not explicitly disclose a probe instruction. And Chua also doesn't explicitly disclose a probe instruction. 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.[233] Because the claimed probe instruction involves redirects,[234] the combination of Lefebvre and Chua renders obvious the claimed probe instruction. 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 35.

124. Chua's redirect action sends the traffic "to another network device (such as a honeypot device or other device of service devices 116) for either inspection or to keep a potential hacker 'busy' to determine if there is a real security threat."[235] 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

[231] '111 Patent at 4:23-33; *see also* '111 Patent at 6:30-32, FIG. 3 (step S305).
[232] *E.g.*, Lefebvre at 1:34-38.
[233] *See* Chua at 7:28-54.
[234] *See* '111 Patent at 4:23-33; *see also* '111 Patent at 6:30-32, FIG. 3 (step S305).
[235] Chua at 7:46-50.

claimed "sending the packet to the controller" of claim 35.

125. And Chua discloses that if the service device deems the packets malicious, the controller may instruct the service device "to send data back to switch 210, such as data indicating whether packets of the packet flow represent a network attack and/or data of the packet flow after the data has been inspected."[236] Chua's SDN controller may perform these actions, including redirecting, in response to information from service devices[237] or business-level security rules.[238] Although Chua discloses this in relation to an intrusion detection system (IDS), a POSITA would have found it obvious that service devices executing DPI would have received similar instructions. This is because IDS systems also analyze network traffic for malicious activity but typically do not employ the more rigorous inspection of packets that DPI systems use. Because Chua discloses sending data back to the network node, such as "data of the packet flow after the data has been inspected,"[239] the combination of Lefebvre and Chua therefore renders obvious the "receiving the packet, from the controller."

---

[236] 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.").

[237] Chua at 7:28-38, 16:23-30, 16:36-44, 28:7-32.

[238] Chua at 14:39-42.

[239] *See* Chua at 21:35-42, 28:18-32.

126. As discussed in Section VII.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's teachings regarding SDN actions, including redirecting.[240] And as discussed in Section VII.A, Lefebvre's controller would execute DPI, as Chua teaches.[241] 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 instead of a service device.[242] 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, as Chua teaches.[243]

127. The combination of Lefebvre and Chua therefore renders obvious dependent claim 35.

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.

128. The combination of Lefebvre and Chua renders obvious dependent claim 36.

---

[240] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller).

[241] *See* Chua at 31:49-54.

[242] *See* Chua at 7:46-50.

[243] *See* 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.").

129. 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.

130. As discussed in Section VII.A, Lefebvre's controller, as modified by Chua's teachings, would implement Chua's actions including redirecting.[244] And Lefebvre's controller would execute DPI, as taught by Chua.[245] So, 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. And because Lefebvre contemplates steering entire packets,[246] the modified Lefebvre controller would continue to redirect the entire packet.

131. The combination of Lefebvre and Chua therefore renders obvious dependent claim 36.

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.

132. The combination of Lefebvre and Chua renders obvious dependent claim 39.

133. A POSITA would have understood that Transmission Control Protocol

[244] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions); Chua at 7:33-54 (describing available actions for an SDN controller).
[245] *See* Chua at 31:49-54.
[246] Lefebvre at 6:17-31; *see also* Lefebvre at 1:28-38 (discussing DPI).

(TCP) packets include separate and distinct header and payload fields. And a POSITA would have further understood that standard TCP packets include flag bits providing additional information related to the state of the connection. RFC 793, published September 1981, defines TCP.[247] I have reproduced below Figure 3 of RFC 793 which depicts the TCP header format.[248]

```
TCP Header Format

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        TCP Header Format

       Note that one tick mark represents one bit position.

                            Figure 3.
```

134. 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.[249] And RFC 793 defines optional

[247] *See* RFC 793 (available at https://www.rfc-editor.org/rfc/pdfrfc/rfc793.txt.pdf, last accessed April 20, 2024) ("RFC793").
[248] RFC793 at 21.
[249] RFC793 at 21.

features that could exist in the options field including no-operation and maximum segment size.[250] Over time, other options including selective acknowledgements (SACK) and timestamps have utilized the TCP header options fields.

135. Lefebvre discloses using policies that consider other standard TCP header fields when classifying traffic, including source and destination addresses, source and destination ports, and protocol fields.[251] Each TCP packet header includes the standard fields defined by RFC 793 such as source and destination ports and addresses, which Lefebvre already considers, and flag bits, which Lefebvre does not indicate it considers when classifying traffic. Chua similarly confirms the various header fields included with TCP packets including source and destination ports.[252]

136. Both Lefebvre and Chua discuss several various header fields but do not specifically identify the flag bits of TCP packets. But, as I explained and RFC793 confirms, flag bits are part of the TCP standard. It is therefore my opinion that

---

[250] RFC793 at 24-25.

[251] 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* Lefebvre 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.

[252] Chua at 26:30-36.

these would have been obvious to a POSITA to modify Lefebvre to consider the flag bits. A POSITA would have been motivated to modify Lefebvre to consider the status of the flag bits which provide additional information regarding the connection's state and are part of the TCP standard. Modifying Lefebvre to consider the status of these standard additional fields of the TCP header would therefore allow Lefebvre's traffic steering to consider connection state when determining which service chain the packets should traverse, and each TCP packet includes these flag bits. Modifying Lefebvre to consider these flag bits would have been well within the skill of a POSITA. Lefebvre already considers other standard aspects of TCP packet headers,[253] so modifying Lefebvre to consider the flag bits in addition to other aspects of the header field would have been well within the ordinary skill of a POSITA. Indeed, Chua discloses how the controller programs network devices to perform actions including forwarding based on header data such as source and destination ports.[254]

137. The combination of Lefebvre and Chua therefore renders obvious dependent

[253] 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* Lefebvre 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.

[254] Chua at 26:30-36.

claim 39.

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.

138. The combination of Lefebvre and Chua renders obvious dependent claim 40.

139. As discussed for claim 39, a POSITA would have found it obvious to modify Lefebvre to consider TCP packet flag bits. As RFC793 confirms, a POSITA would have also understood that TCP packet flag bits include the SYN, ACK, FIN, and RST flag bits, among others. These are some of the standard flag bits defined as part of the TCP standard.[255]

140. The combination of Lefebvre and Chua therefore renders obvious dependent claim 40.

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.

141. The combination of Lefebvre and Chua renders obvious dependent claim 41.

142. A POSITA would have understood that IP packets include separate and

[255] RFC793 at 22 (describing the defined control bits including SYN, ACK, FIN, and RST).

distinct header and payload fields.[256] And a POSITA would have understood that IP packets include source and destination addresses.[257]

143. Confirming this, 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, and a protocol."[258] 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.

144. The combination of Lefebvre and Chua therefore renders obvious dependent claim 41.

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

145. The combination of Lefebvre and Chua renders obvious dependent claim 42.

146. As discussed for claim 41, IP packets include source and destination addresses. A POSITA would have understood these addresses are IP addresses.

147. The combination of Lefebvre and Chua therefore renders obvious dependent claim 42.

---

[256] *See* RFC 793 at 21.
[257] *See* RFC 793 at 21.
[258] Lefebvre at 2:47-52, 7:59-67, 8:4-30.

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.

148. The combination of Lefebvre and Chua renders obvious dependent claim 43.

149. As discussed, RFC 793, published September 1981, defines TCP.[259] I have reproduced below Figure 3 of RFC 793 which depicts the TCP header format.[260]

```
TCP Header Format

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        TCP Header Format

      Note that one tick mark represents one bit position.

                              Figure 3.
```

150. Each TCP header includes source and destination port fields, a sequence

[259] *See* RFC793.

[260] RFC793 at 21.

number field, along with acknowledgement number, control bits, window, checksum, urgent pointer, and options fields.[261] And RFC 793 defines optional features that could exist in the options field including no-operation and maximum segment size.[262] I note that nothing in RFC 793 constitutes a "TCP sequence mask" field, as claim 43 requires. And nothing in the '111 Patent explains what makes up the "TCP sequence mask" field. Rather, the '111 Patent explains how to compute the TCP sequence mask.[263]

151. I have been asked to assume that the claims are not indefinite. Regardless of what TCP header field corresponds to "TCP sequence mask," the combination of Lefebvre and Chua renders claim 43 obvious. This is because a POSITA would have understood that TCP packets (more typically referred to as TCP segments) include each of the fields required by RFC 793, including source and destination ports, and TCP sequence numbers. Lefebvre discloses using the source and destination port and address fields along with the protocol fields, but does not specifically identify the TCP sequence number.[264] But because the sequence

[261] RFC793 at 21.

[262] RFC793 at 24-25.

[263] '111 Patent at 4:58-60, 6:41-7:40, 9:56-61.

[264] 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").

number is part of the TCP standard, defined by RFC 793, as are the fields Lefebvre already considers, it is my opinion that it would have 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.

152. A POSITA would have 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. Lefebvre's traffic steering already considers other standard TCP header fields when classifying traffic, including source and destination addresses, source and destination ports, and protocol fields.[265] Each TCP packet header includes the standard fields defined by RFC 793 such as source and destination ports and addresses, which Lefebvre already considers, and sequence numbers, which Lefebvre does not. Modifying Lefebvre to consider additional standard fields would allow Lefebvre's traffic steering to consider the details provided by these

[265] 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* Lefebvre 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.

additional standard fields when classifying traffic and determining which service chain the packets should traverse. Modifying Lefebvre to consider these additional standard fields would have been well within the skill of a POSITA. Lefebvre already considers other standard aspects of TCP headers, so parsing and considering additional standard fields would have been well within the ordinary skill of a POSITA. Chua provides evidence of 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.[266]

153. The combination of Lefebvre and Chua therefore renders obvious dependent claim 43.

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.

154. The combination of Lefebvre and Chua renders obvious dependent claim 44.

[266] *E.g.*, Chua at 16:27-30.

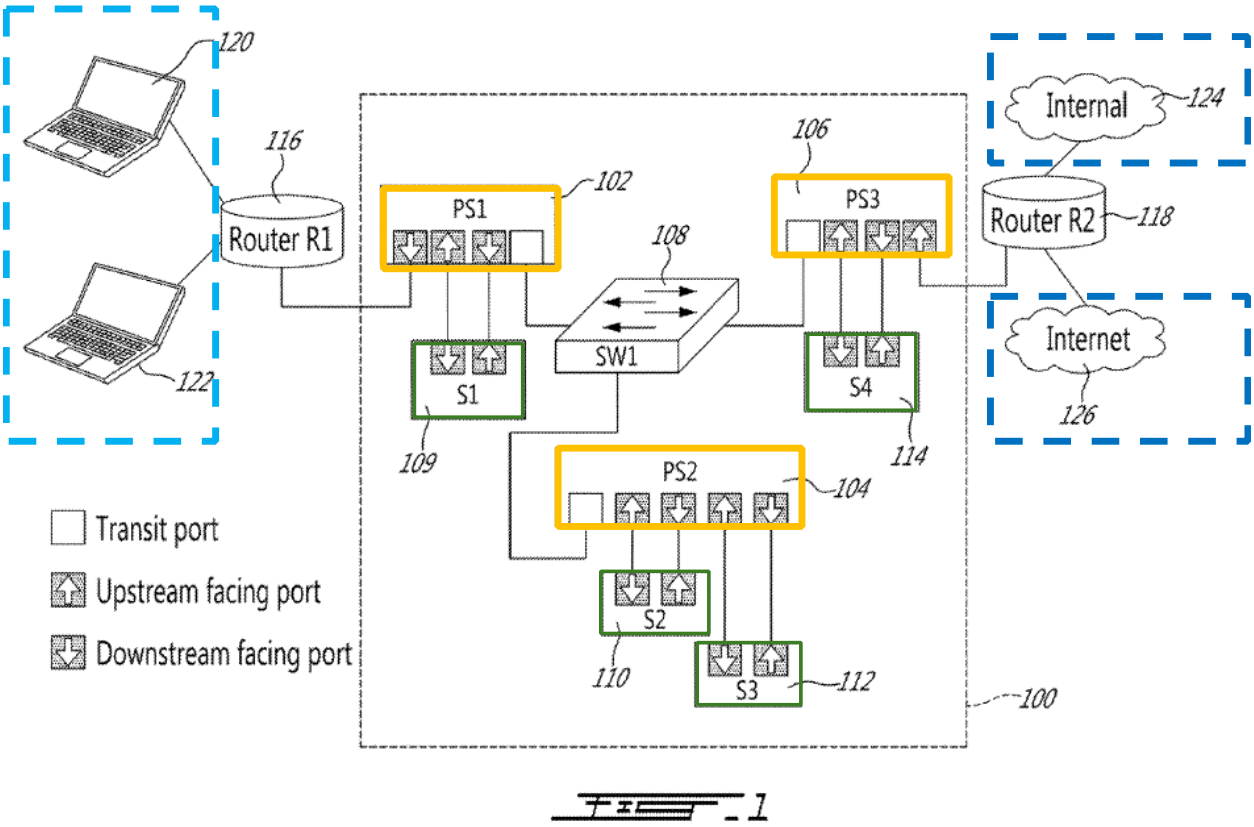


FIG. 1

Annotated Lefebvre FIG. 1

155. Lefebvre discloses that the network includes **user equipment 120, 122** connected to router R1 and the **internal network 124** and the **Internet 126** connected to router R2.[267] A POSITA would therefore understand that Lefebvre's service network is at least a LAN if not a WAN. And Lefebvre's service network includes **Internet 126**.

156. The combination of Lefebvre and Chua therefore renders obvious dependent claim 44.

[267] Lefebvre at 5:38-54, 6:13-16.

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.

157. The combination of Lefebvre and Chua renders obvious dependent claim 45.

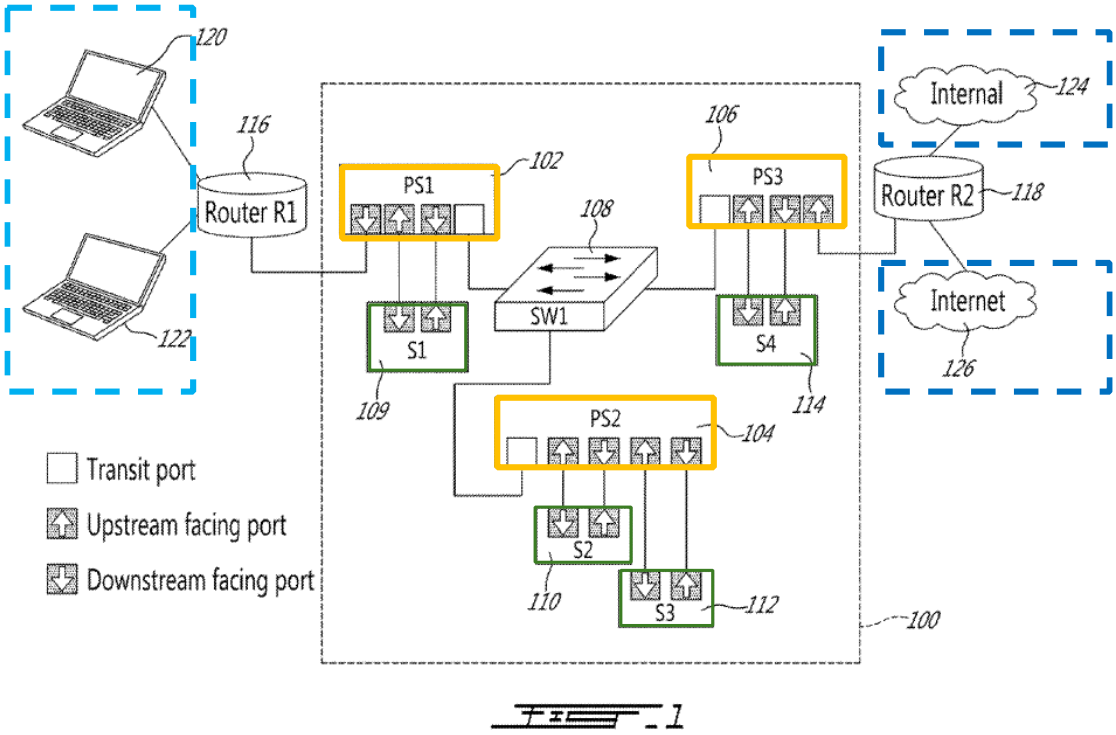


FIG. 1

Annotated Lefebvre FIG. 1

158. Lefebvre discloses that the network includes **user equipment 120, 122** connected to router R1 and the **internal network 124** and the **Internet 126** connected to router R2.[268] A POSITA would therefore understand that Lefebvre's **user equipment 120, 122** are client devices. And a POSITA would understand that the **internal network 124** and the **Internet 126** include servers.

[268] Lefebvre at 5:38-54, 6:13-16.

159. The combination of Lefebvre and Chua therefore renders obvious dependent claim 45.

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.

160. The combination of Lefebvre and Chua renders obvious dependent claim 46.

161. As discussed for claim 45, Lefebvre discloses a service network including **user equipment 120, 122** connected to router R1 and the **internal network 124** and the **Internet 126** connected to router R2.[269] And Lefebvre depicts **user equipment 120, 122** as laptops.[270] A POSITA would therefore understand that Lefebvre's **user equipment 120, 122** are client devices which could include a smartphone, tablet computer, personal computer, laptop computer, or wearable computer device. And a POSITA would understand that the **internal network 124** and the **Internet 126** include servers, like web servers.[271]

162. The combination of Lefebvre and Chua therefore renders obvious dependent claim 46.

[269] Lefebvre at 5:38-54, 6:13-16.

[270] Lefebvre at FIG. 1.

[271] *See* Moore at 11 (describing the detection of web servers operating on non-standard ports on an internal network).

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

163. The combination of Lefebvre and Chua renders obvious dependent claim 47.

164. Lefebvre discloses that its SDN use the OpenFlow protocol.[272] As Lefebvre discloses, "OpenFlow is a communications protocol that gives access to the forwarding plane of a network switch or router over the network."[273] Chua confirms this when disclosing that OpenFlow is an open protocol for implementing software defined networking.[274]

165. The Open Networking Foundation (ONF) manages the OpenFlow protocol and related technologies.[275] A POSITA would have therefore understood the OpenFlow protocol is a standard protocol.

166. The combination of Lefebvre and Chua therefore renders obvious dependent claim 47.

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.

167. The combination of Lefebvre and Chua renders obvious dependent claim 48.

---

[272] Lefebvre at 4:30-40.

[273] Lefebvre at 4:33-35.

[274] Chua at 1:35-41; *see also* McKeown.

[275] *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.").

168. As discussed for claim 47, the ONF manages the OpenFlow protocol and related technologies.[276] Lefebvre discloses that its SDN uses certain features from version 1.1 of the OpenFlow protocol.[277]

169. The ONF published version 1.3.3 of the OpenFlow protocol on September 2013.[278] And the ONF published version 1.4.0 of the OpenFlow protocol on October 14, 2013.[279] To the extent that OpenFlow 1.1 is not compatible with OpenFlow protocol version 1.3.3 and 1.4.0, a POSITA would have 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.

170. The combination of Lefebvre and Chua therefore renders obvious dependent claim 48.

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

171. The combination of Lefebvre and Chua renders obvious dependent claim 49.

172. Type-value-length (TLV) structures are well known data structures often used in protocols. For example, RFC 5810, published March 2010, discloses that

[276] *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.").

[277] Lefebvre at 4:30-40.

[278] Ex. 1017 at 3.

[279] Ex. 1017 at 3; *see also* Ex. 1018.

the ForCES protocol makes extensive use of TLVs.[280] And OpenFlow protocol version 1.4.0, published October 14, 2013, makes use of TLVs.[281] A POSITA would have therefore found it obvious to modify Lefebvre's SDN to utilize instructions with a TLV structure.

173. The combination of Lefebvre and Chua therefore renders obvious dependent claim 49.

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

174. The combination of Lefebvre and Chua renders obvious dependent claim 50.

175. Lefebvre discloses a service network in FIG. 1 which I've annotated below.

[280] RFC 5810 (available at https://www.rfc-editor.org/rfc/rfc5810.html, last accessed April 13, 2024).

[281] *E.g.*, Ex. 1018 at 52.

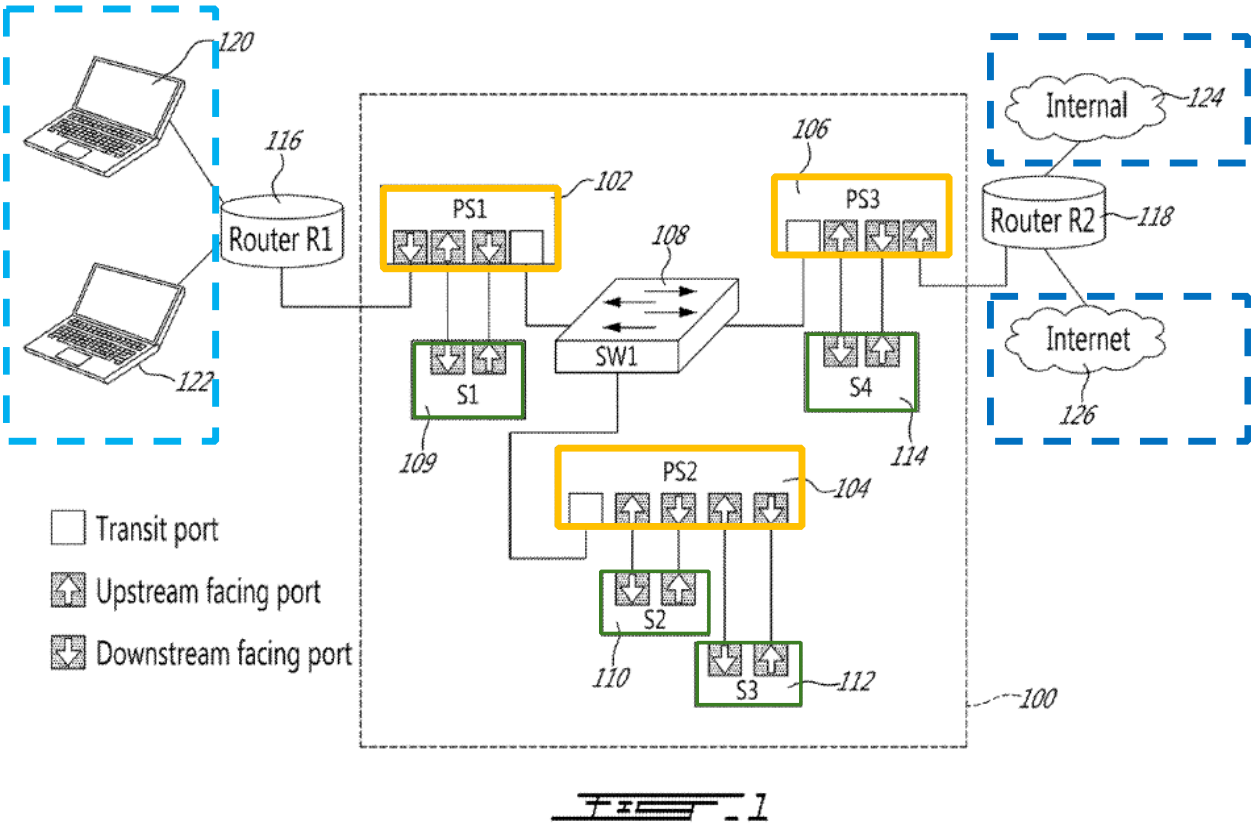


FIG. 1

Annotated Lefebvre FIG. 1

176. Lefebvre discloses that **perimeter switches 102, 104, 106** are OpenFlow switches.[282] Lefebvre's controller programs **perimeter switches 102, 104, 106** to route traffic through the service chain.[283]

177. The combination of Lefebvre and Chua therefore renders obvious dependent claim 50.

[282] Lefebvre at 5:38-54.

[283] *E.g.*, Lefebvre at 5:43-47.

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.

178. The combination of Lefebvre and Chua renders obvious dependent claim 51.

179. Lefebvre's traffic steering process contemplates service policies defined by IP address blocks[284] and IP addresses,[285] for example. A POSITA would have therefore understood Lefebvre's packet network is an IP network.

180. The combination of Lefebvre and Chua therefore renders obvious dependent claim 51.

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.

181. The combination of Lefebvre and Chua renders obvious dependent claim 52.

182. Lefebvre's traffic steering process contemplates service policies defined by TCP source and destination ports and TCP/UDP flows, for example.[286] A POSITA would have therefore understood Lefebvre's packet network is a TCP network with TCP packets.

183. The combination of Lefebvre and Chua therefore renders obvious dependent claim 52.

---

[284] Lefebvre at 6:38-41.

[285] Lefebvre at 6:46-48, 7:60-66.

[286] Lefebvre at 6:38-41, 7:60-66.

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.

184. The combination of Lefebvre and Chua renders obvious dependent claim 53.

185. Lefebvre discloses steering packet traffic, which a POSITA would have understood discloses the required "one or more additional packets."[287] As discussed for claim element [32.1], Lefebvre discloses flow-based policies. A POSITA would have 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.[288] As discussed for claim element [32.4], Lefebvre's non-subscriber traffic does not satisfy the criterion of directing subscriber traffic so the packets are routed to the source.[289] A POSITA would have understood this discloses "responsive to an additional packet not satisfying the criterion, sending over the packet network, the additional packet to the second entity." And Lefebvre discloses how multiple packets are "processed and steered"

[287] Lefebvre at 2:30-38.
[288] Lefebvre at 6:46-53.
[289] *See* Lefebvre at 14:25-29.

to the next node in their assigned service paths.[290] A POSITA would have therefore understood Lefebvre discloses "sending the additional packet over the packet network in response to the instruction."

186. The combination of Lefebvre and Chua therefore renders obvious dependent claim 53.

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.

187. The combination of Lefebvre and Chua renders obvious dependent claim 54.

188. Lefebvre discloses that its service network is a software defined network (SDN).[291] A POSITA would have 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.[292]

189. The combination of Lefebvre and Chua therefore renders obvious dependent claim 54.

190. It is therefore my opinion that claims 32-36 and 39-54 are rendered obvious by Lefebvre in view of Chua.

---

[290] Lefebvre at 6:5-12, 6:17-25.
[291] *See* Lefebvre at 4:30-40.
[292] *See* Ex. 1011 at 2.

VIII. GROUND 2: CLAIMS 37-38 ARE RENDERED OBVIOUS BY LEFEBVRE IN VIEW OF CHUA AND RASH.

191. The Challenged Claims include dependent claims 37 and 38. As I detail below, a POSITA would have recognized that each and every limitation of claims 37-38 are rendered obvious by Lefebvre in view of Chua and Rash.

192. As I discussed above in Section III, well before 2014, both SDN and DPI were well-known and ubiquitous. For example, both Lefebvre and Chua discuss SDN and DPI.[293] Lefebvre and Chua both disclose how DPI often executes in dedicated service devices.[294] Chua specifically discloses the SDN controller performing DPI as well.[295]

193. Rash discloses how network devices transmit portions of the packet to destinations other than the original intended destination.[296] Rash discloses its dynamic mirroring functionality helps minimize the data mirrored to identification appliances or other infrastructure devices configured to carry out the application identification function.[297] And Rash discloses the applicability of its dynamic traffic mirroring function for intrusion detection systems and network access

[293] Lefebvre at 1:21-25, 1:34-38; Chua at 10:48-52, 31:48-59.
[294] Lefebvre at 1:21-25, 1:34-38; Chua at 31:54-59 (describing DPI service devices).
[295] Chua at 31:48-54.
[296] Rash at 7:51-59, 9:3-12, 25:30-55, 26:11-35, 27:31-43, 29:58-62, 35:66-36:2.
[297] Rash at 12:59-13:8, 34:43-51.

control, among other applications.[298]

194. As I describe below, a POSITA would have found that Lefebvre in view of Chua and Rash renders obvious, either expressly or inherently, each and every element of dependent claims 37-38.

A. A POSITA WOULD HAVE BEEN MOTIVATED TO COMBINE LEFEBVRE AND RASH.

195. It is my opinion that a POSITA would have 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.[299] Lefebvre discloses how operators can define the traffic classes and service chains for those classes for processing traffic flows on a per packet basis.[300] 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.[301]

196. Lefebvre identifies several types of services that could form part of the service chain, including DPI.[302] Regarding DPI, Lefebvre discloses "[t]hese services are usually packaged and sold as dedicated appliance (either physical or

[298] Rash at 10:29-56.
[299] Lefebvre at 1:13-21, 2:41-47.
[300] Lefebvre at 4:41-43.
[301] Lefebvre at 4:41-47.
[302] Lefebvre at 1:28-38, 4:48-54.

virtual) and are often expensive."[303] One reason Lefebvre offers for implementing traffic steering is to direct traffic away from expensive services like DPI, reducing the need to invest in new service appliances.[304] Lefebvre therefore controls the usage of DPI services by routing network traffic through the service only as appropriate, where the services are separate from the controller.

197. Like Lefebvre, Rash discloses packet inspection systems.[305] 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."[306] Rash discloses that a network control manager controls the mirroring function.[307] Rash's packet mirroring device may be part of the network infrastructure.[308] Rash discloses the purpose of the mirroring may include application identification, security, or even ease of use, and the destination may exist either logically or physically.[309] Rash discloses how the mirrored traffic may have a different destination based on certain criteria.[310] And Rash discloses its

[303] Lefebvre at 1:25-27, 1:34-38.
[304] Lefebvre at 1:32-38.
[305] Rash at 7:31-34.
[306] Rash at 9:1-12, 27:31-43.
[307] Rash at 9:12-15, 29:46-30:17.
[308] Rash at 25:30-55.
[309] Rash at 26:11-35.
[310] Rash at 35:66-36:2.

dynamic mirroring functionality helps minimize the data mirrored to identification appliances or other infrastructure devices configured to carry out the application identification function.[311]

198. A POSITA would have been motivated to modify Lefebvre with Rash to minimize the amount of network capacity used to examine packets, which would help reduce network costs and improve overall efficiency. A POSITA would have 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.[312] Lefebvre and Rash are therefore in the same field of endeavor and contain overlapping disclosures with similar purposes.

199. Modifying Lefebvre in view of Rash's teachings would have been obvious to a POSITA. Lefebvre has three different levels of policies which have different levels of granularity.[313] Similarly, Rash explains that the network control engine may use packet fields including address, protocol, length or byte count, and other fields when making its mirroring determinations.[314]

---

[311] Rash at 12:59-13:8, 34:43-51.
[312] Lefebvre at 1:28-38; Rash at 9:12-15, 25:30-55, 26:11-35, 29:46-30:17.
[313] Lefebvre at 6:32-48.
[314] Rash at 29:53-58.

200. Although Lefebvre mentions DPI as one type of known service,[315] it considers packets in their entirety and similarly forwards packets in their entirety.[316] Rash provides specific details as to how entire packets or just portions of packets are mirrored.[317] Rash discloses this targeted approach to mirroring minimizes the data the application identification function must consider.[318] A POSITA would therefore look to Rash's detailed teachings regarding mirroring of packets and portions of packets and apply them to Lefebvre.

201. A POSITA would have been motivated to modify Lefebvre with Rash's teachings for multiple reasons. Lefebvre discloses that services including DPI "are usually packaged and sold as dedicated appliances (either physical or virtual) and are often expensive."[319] Rash similarly acknowledges the complexity and expense associated with deploying application identification functions in the network.[320] Rash explains its dynamic mirroring function leverages existing capabilities in network devices, thereby minimizing additional costs.[321] Rash also discloses this targeted approach to mirroring minimizes the data the application identification

[315] Lefebvre at 6:17-31; *see also* Lefebvre at 1:28-38 (discussing DPI).
[316] *E.g.*, Lefebvre at 6:5-25, 7:42-8:65.
[317] Rash at 9:1-12, 27:31-43.
[318] Rash at 12:59-13:8, 34:43-51.
[319] Lefebvre at 1:25-27, 1:34-38.
[320] Rash at 30:18-27.
[321] Rash at 30:29-44, 34:37-47.

function must consider.[322] By implementing Rash's mirroring of a portion of a packet instead of the entirety of a packet, a POSITA would recognize the network would experience a corresponding reduction in the amount of network capacity utilized to examine the packets, thereby reducing network costs and improving overall efficiency.

202. Implementing Rash's teachings in Lefebvre's controller would have been well within the skill of a POSITA. Lefebvre already redirects entire packets to the appropriate service.[323] Similarly, Rash discloses mirroring packets to the network infrastructure that analyzes the mirrored traffic.[324] Modifying Lefebvre to forward only a portion of the packet as Rash teaches would have been well within the skill of a POSITA, as Rash confirms.

203. A POSITA would have 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.[325] Rash's technique allow for "any granularity of the frame of the network traffic"

---

[322] Rash at 12:59-13:8, 34:43-51.

[323] Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

[324] *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).

[325] *See* Lefebvre at 8:49-65, Table 1, Table 2 (depicting OpenFlow switch table actions).

and Rash itself 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 described herein."[326] Rash's technique therefore relies on nothing more than a POSITA's knowledge of how to take any portion of a protocol data unit. Because Rash confirms its teachings of mirroring portions of a protocol data unit are within a POSITA's knowledge and ordinary skill, it is my opinion that a POSITA would have had a reasonable expectation of success in applying Rash's teachings to Lefebvre. And similarly, it is my opinion that implementing this modification would have been simple and straightforward for a POSITA, requiring nothing more than a POSITA's routine skill.

B. DEPENDENT CLAIM 37-38

204. It is my opinion that Lefebvre in view of Chua and Rash renders obvious dependent claims 37 and 38.

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.

205. The combination of Lefebvre, Chua, and Rash renders obvious dependent

[326] Rash at 27:39-43; *see also* Rash at 27:44-53.

claim 37.

206. As discussed for claim [32.5], the combination of Lefebvre and Chua renders obvious the claimed "responsive to the packet satisfying the criterion" aspect of claim 37. And Lefebvre and Chua acknowledge the different aspects of a packet, including the header field.[327] But Lefebvre and Chua do not disclose sending a portion of the packet to the controller.

207. Rash discloses a packet inspection system that transmits portions of the packet to a destination other than the intended destination.[328] 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"[329]

208. As discussed in Section VIII.A, a POSITA would have been motivated and found it obvious to apply Rash's teachings to Lefebvre's network. And Rash discloses that this mirroring (the claimed instruction) sends the portion of packets to "the application identification engine" of the network.[330] So, Lefebvre's network, as modified by Rash, would mirror portions of packets to the controller

[327] Lefebvre at 15:54-63 (discussing considering the header fields of a packet), 16:67-17:2 (same); Chua at 25:57-61.

[328] Rash at 7:51-59, 9:3-12, 25:30-55, 26:11-35, 27:31-43, 29:58-62, 35:66-36:2.

[329] Rash at 9:1-15, 27:31-43, 29:7-12, 29:46-30:17.

[330] *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).

which, as modified by Chua, would execute DPI.

209. The combination of Lefebvre, Chua, and Rash therefore renders obvious dependent claim 37.

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.

210. The combination of Lefebvre, Chua, and Rash renders obvious dependent claim 38.

211. 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."[331] A POSITA would have understood this to mean that Rash teaches mirroring multiple consecutive bytes, and that the mirroring (the claimed instruction) identifies the consecutive bytes for mirroring.

212. The combination of Lefebvre, Chua, and Rash therefore renders obvious dependent claim 38.

213. It is therefore my opinion that claims 37-38 are rendered obvious by Lefebvre in view of Chua and Rash.

[331] Rash at 9:6-12; *see also* Rash at 29:7-12, 29:46-62.

IX. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

214. I am not aware of any objective indicia of non-obviousness that would make the Challenged Claims non-obvious over the prior art discussed. And in my opinion, any such indicia would not overcome the strong showing of obviousness based on the prior art. I reserve the right to supplement my opinions to the extent Patent Owner offers any purported evidence of such objective indicia of non-obviousness.

X. CONCLUSION

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

Dated: 5/5/2024 By: Nader F. Mir
Nader F. Mir