

Filed on Behalf of: Cisco Systems, Inc.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,

Petitioner,

- vs. -

ORCKIT IP, LLC,

Patent Owner

PETITION FOR INTER PARTES REVIEW

OF U.S. PATENT NO. 10,652,111

Case No.: IPR2023-00554

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING.....	1
III.	REQUESTED RELIEF AND REASONS FOR REQUESTED RELIEF	2
IV.	OVERVIEW OF THE '111 PATENT	2
	A. Summary of the '111 Patent.....	2
	B. Priority Date	4
V.	STATUTORY GROUNDS FOR CHALLENGES	4
VI.	LEVEL OF ORDINARY SKILL IN THE ART	5
VII.	CLAIM CONSTRUCTION	5
VIII.	THE ART PRIOR TO THE '111 PATENT	8
	A. Lin	8
	B. Shieh.....	10
	C. Swenson	12
IX.	GROUND 1: CLAIMS 1-9, 12-24 AND 27-31 ARE UNPATENTABLE AS OBVIOUS OVER LIN IN VIEW OF SWENSON.	13
	A. Claim 1	13
	B. Claim 2	33
	C. Claim 3	34
	D. Claim 4	36
	E. Claim 5	37
	F. Claim 6	39

G. Claim 7	40
H. Claim 8	41
I. Claim 9	41
J. Claim 12	43
K. Claim 13	43
L. Claim 14	43
M. Claim 15	44
N. Claim 16	44
O. Claim 17	45
P. Claim 18	46
Q. Claim 19	47
R. Claim 20	47
S. Claim 21	49
T. Claim 22	49
U. Claim 23	50
V. Claim 24	50
W. Claim 27	52
X. Claim 28	52
Y. Claim 29	52
Z. Claim 30	53
AA. Claim 31	54
X. GROUND 2: CLAIMS 1, 5-9, 12-24 and 27-30 ARE UNPATENTABLE AS OBVIOUS OVER SHIEH IN VIEW OF SWENSON.	54

A. Claim 1	54
B. Claims 5-9	66
C. Claim 12	66
D. Claim 13	67
E. Claim 14	67
F. Claim 15	67
G. Claim 16	68
H. Claim 17	68
I. Claim 18	69
J. Claim 19	69
K. Claim 20	69
L. Claim 21	70
M. Claim 22	70
N. Claim 23	71
O. Claim 24	72
P. Claim 27	72
Q. Claims 28.....	72
R. Claims 29.....	73
S. Claim 30	73
XI. OBJECTIVE INDICIA OF NONOBVIOUSNESS	73
XII. DISCRETIONARY DENIAL UNDER § 325(D) OR § 314 IS NOT WARRANTED.....	74
XIII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8.....	77

A. Real Party-in-Interest	77
B. Related Matters	78
C. Lead and Back-up Counsel and Service Information	78
XIV. CONCLUSION.....	78
APPENDIX A – CLAIM LISTING	A-1

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB March 20, 2020)	75, 76
<i>Dish Network LLC v. Broadband iTV, Inc.</i> , IPR2020-01359, Paper 15 (PTAB Feb. 12, 2021).....	75
<i>General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha</i> , PR2016-01357, Paper 19 (PTAB Sept. 6, 2017).....	74
<i>Orckit Corp. v. Cisco Systems, Inc.</i> , Case No. 2:22-cv-00276 (E.D. Tex.).....	78
<i>PEAG LLC v. Varta Microbattery GMBH</i> , IPR2020-01214, Paper 8 (Jan. 6, 2021).....	76
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2015) (en banc)	5
<i>Sand Revolution II LLC v. Continental Intermodal Group-Trucking LLC</i> , IPR2019-01393, Paper 24 (PTAB June 16, 2020)	75, 77
<i>Sega of Am., Inc. v. Uniloc USA, Inc.</i> , IPR2014-01453, Paper 11 (PTAB Mar. 10, 2015).....	73
<i>Verizon v. Huawei</i> , IPR2020-01079, Paper 10 (Jan. 14, 2021).....	77
Statutes	
35 U.S.C. § 102(a)	4, 5
35 U.S.C. § 103	4, 5
35 U.S.C. § 311	1
35 U.S.C. §314(a)	1, 74

35 U.S.C. § 315(e)(2).....77

35 U.S.C. §325(d) 74

Other Authorities

37 C.F.R. § 42.8 77

37 C.F.R. § 42.8(b)(1).....77

37 C.F.R. § 42.100 1, 5

PETITIONERS' EXHIBIT LIST

1001	U.S. Patent No. 10,652,111 (the '111 Patent)
1002	Prosecution History of the '111 Patent
1003	Curriculum Vitae of Samrat Bhattacharjee
1004	Declaration of Dr. Samrat Bhattacharjee, dated February 21, 2023
1005	U.S. Patent No. 9,264,400 ("Lin")
1006	U.S. Patent Application Publication No. 2013/0291088 ("Shieh")
1007	U.S. Patent Application Publication No. 2013/0322242 ("Swenson")
1008	RFC 2460, Internet Protocol, Version 6 (IPv6) Specification
1009	Nunes, A., et al., A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks
1010	Complaint in <i>Orckit Corp. v. Cisco Systems, Inc.</i> , Case No. 2:22-cv-00276 (E.D. Tex.)
1011	Amended Scheduling Order in <i>Orckit Corp. v. Cisco Systems, Inc.</i> , Case No. 2:22-cv-00276 (E.D. Tex.)
1012	Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation, Vidal, K., United States Patent and Trademark Office, June 21, 2022
1013	Federal Case Management Statistics for the Eastern District of Texas, as of June 30, 2022

I. INTRODUCTION

Pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, Cisco Systems, Inc. (“Petitioner”) respectfully requests that the Board review and cancel Claims 1-9, 12-24 and 27-31 (the “Challenged Claims”) of U.S. 10,652,111 (“the ’111 Patent,” EX1001). The claimed methods in the ’111 Patent would have been obvious to a person of ordinary skill in the art (“POSA”) well before the ’111 Patent’s earliest priority date. For example, the combination of U.S. Patent No. 9,264,400 (“Lin,” EX1005) and U.S. Patent Application Publication No. 2013/0333342 (“Swenson,” EX1007) discloses deep packet inspection (“DPI”) of packets in a computer network where a network node is under the control of a central controller. The combination of U.S. Patent Application Publication No. 2013/0291088 (“Shieh,” EX1006) and Swenson teaches the same thing. The disclosures in these three prior art references, along with the knowledge of a POSA, render the Challenged Claims unpatentable as obvious, as explained below in Grounds 1 and 2.

II. GROUNDS FOR STANDING

Petitioner certifies that the ’111 Patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting *inter partes* review challenging the patent claims on the grounds identified in this Petition.

III. REQUESTED RELIEF AND REASONS FOR REQUESTED RELIEF

Petitioner asks that the Board institute a trial for an *inter partes* review of the Challenged Claims, and that the Director cancel them as unpatentable. The analysis demonstrating the obviousness of the Challenged Claims is set forth in the below sections of this Petition and supported by the declaration of Petitioner's expert, Dr. Samrat Bhattacharjee. EX1004, ¶¶74-318; EX1003.

IV. OVERVIEW OF THE '111 PATENT

A. Summary of the '111 Patent

The '111 Patent discloses methods and systems relating to “deep packet inspection (DPI) in a software defined network (SDN).” EX1001, Abstract; *see id.*, 1:14-16, EX1004, ¶¶30-34. The '111 Patent discloses a “central controller of the SDN” that is used to “configure[e] a plurality of network nodes operable in the SDN” with instructions that tell the network nodes what to do with incoming packets. EX1001, 2:27-30, 2:3-3; EX1004, ¶30. For example, the central controller may send a “probe” instruction to a network node such that, when the network node receives a packet that matches a “packet-applicable criterion,” the network node will “mirror” (i.e., send) some or all of the packet to a security component for inspection. EX1001, 2:3-44; EX1004, ¶30.

“[T]he central controller 111 [shown below in Figure 1 of the '111 Patent] is configured to perform deep packet inspection on designated packets from designated

flows or TCP sessions.” EX1001, 4:5-7; EX1004, ¶31. “To this end, the central controller 111 is further configured to instruct each of the network nodes 112 which of the packets and/or sessions should be directed to the controller 111 for packet inspections.” EX1001, 4:8-11. “The determination [of whether a packet requires inspection] is performed based on a set of instructions provided by the controller 111.” EX1001, 4:14-15. “A packet that requires inspection is either redirected to the controller 111 or mirrored and a copy thereof is sent to the controller 111.” EX1001, 4:15-18.

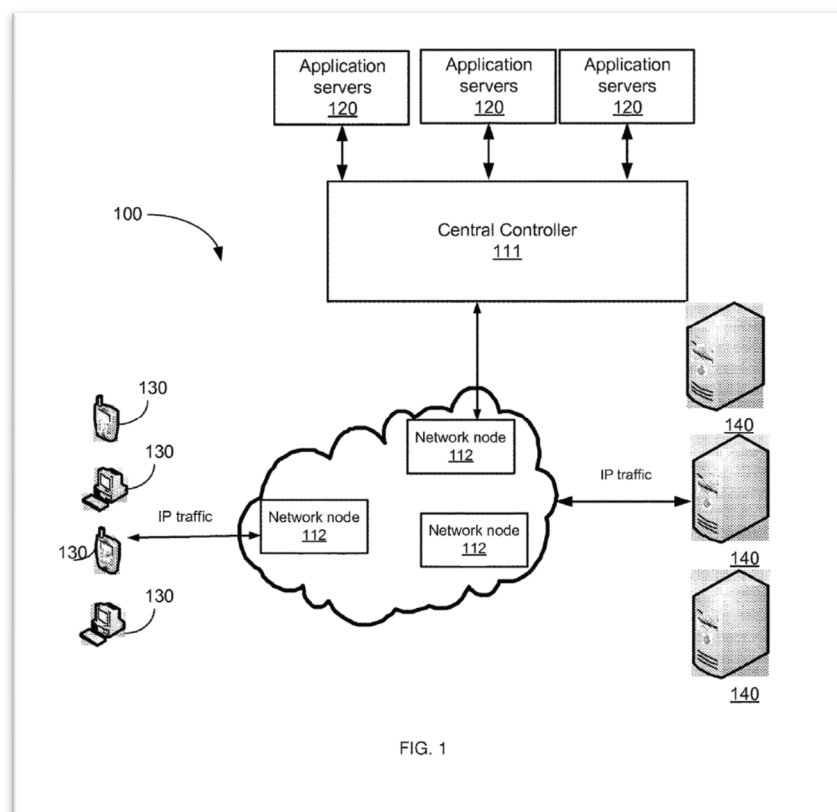


Figure 1 of the '111 Patent

During prosecution, the Applicant relied heavily on claim limitations reciting “... sending, by the controller to the network node over the packet network, an instruction and a packet-applicable criterion” and “... receiving by the network node from the controller, the instruction and the criterion” to distinguish the prior art, along with arguments that there was no motivation to combine the cited art. EX1002 at 322-330, 397-417, 492-501; EX1004, ¶¶35-46.

B. Priority Date

Solely for the purposes of this Petition, Petitioner assumes that the priority date for the ’111 Patent is April 22, 2014, the filing date of U.S. Provisional Patent Application No. 61/982,358 to which the ’111 Patent claims priority. EX1004, ¶47.

V. STATUTORY GROUNDS FOR CHALLENGES

Ground #1: Claims 1-9, 12-24 and 27-31 of the ’111 Patent are obvious under 35 U.S.C. § 103 over Lin in view of Swenson and the knowledge of a POSA. Lin was filed on December 2, 2013, and issued on February 16, 2016. EX1005. Thus, Lin qualifies as prior art under at least post-AIA 35 U.S.C. § 102(a)(2).

Swenson claims priority to a pair of provisional applications filed on June 1, 2012, and January 18, 2013, respectively. Swenson was filed as a non-provisional application on May 31, 2013. Swenson published on December 5, 2013. Thus, Swenson qualifies as prior art under at least post-AIA 35 U.S.C. §§ 102(a)(1)-(2).

Ground #2: Claims 1, 5-9, 12-24 and 27-30 of the '111 Patent are obvious under 35 U.S.C. § 103(a) over Shieh in view of Swenson and the knowledge of a POSA. Shieh was filed as a provisional application on April 11, 2012 and as a non-provisional application on April 10, 2013. Shieh published on October 31, 2013. Thus, Shieh qualifies as prior art under at least post-AIA 35 U.S.C. §§ 102(a)(1)-(2). Swenson qualifies as prior art for the reasons stated above for Ground 1.

VI. LEVEL OF ORDINARY SKILL IN THE ART

As of April 22, 2014, a POSA would have had a bachelor's degree in computer science, computer engineering, or an equivalent, and two years of professional experience, and a POSA would have had a working knowledge of hardware and software for packet-switched networking. EX1004, ¶¶48-49. Lack of work experience can be remedied by additional education and vice versa. *Id.*, ¶48.

VII. CLAIM CONSTRUCTION

In *inter partes* review, claim terms must be given their ordinary and customary meaning as understood by a POSA at the time of the invention in light of the specification and the prosecution history pertaining to the patent. *See* 37 C.F.R. § 42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2015) (en banc).

The claim term “controller” should be construed to mean “an entity configured to perform deep packet inspection on packets.” EX1001, 10:52-62;

EX1004, ¶¶69-71. The '111 Patent discloses “a method for deep packet inspection (DPI) in a software defined network (SDN), wherein **the method is performed by a central controller** of the SDN.” EX1001, 2:27-30 (emphasis added); *see id.*, 3:56-59. Further, the patent states that “the central controller 111 is **configured to perform deep packet inspection on designated packets** from designated flows or TCP sessions.” *Id.*, 4:5-7 (emphasis added); *see id.*, 2:49-51, 4:8-11, 9:67-10:1.

Further, the '111 Patent describes that “the central controller 111 includes a DPI flow detection module 311, a DPI engine 312, and a memory 313, and a processing unit 314,” as shown below in Figure 3. EX1001, 5:33-36. “The DPI engine 312 [is] configured to inspect a packet or a number of bytes to provide application metadata as required by an application executed by an application server 120.” EX1001, 5:36-39; *see id.*, 5:40-59. A POSA would have known from this description that the central controller was configured in this manner to provide DPI on redirected packets, as all of the embodiments in the '111 Patent disclose that redirected packets are sent to the central controller for DPI. *See, e.g.*, EX1004, ¶71; EX1001, 4:8-18, 4:49-50, 8:1-5.

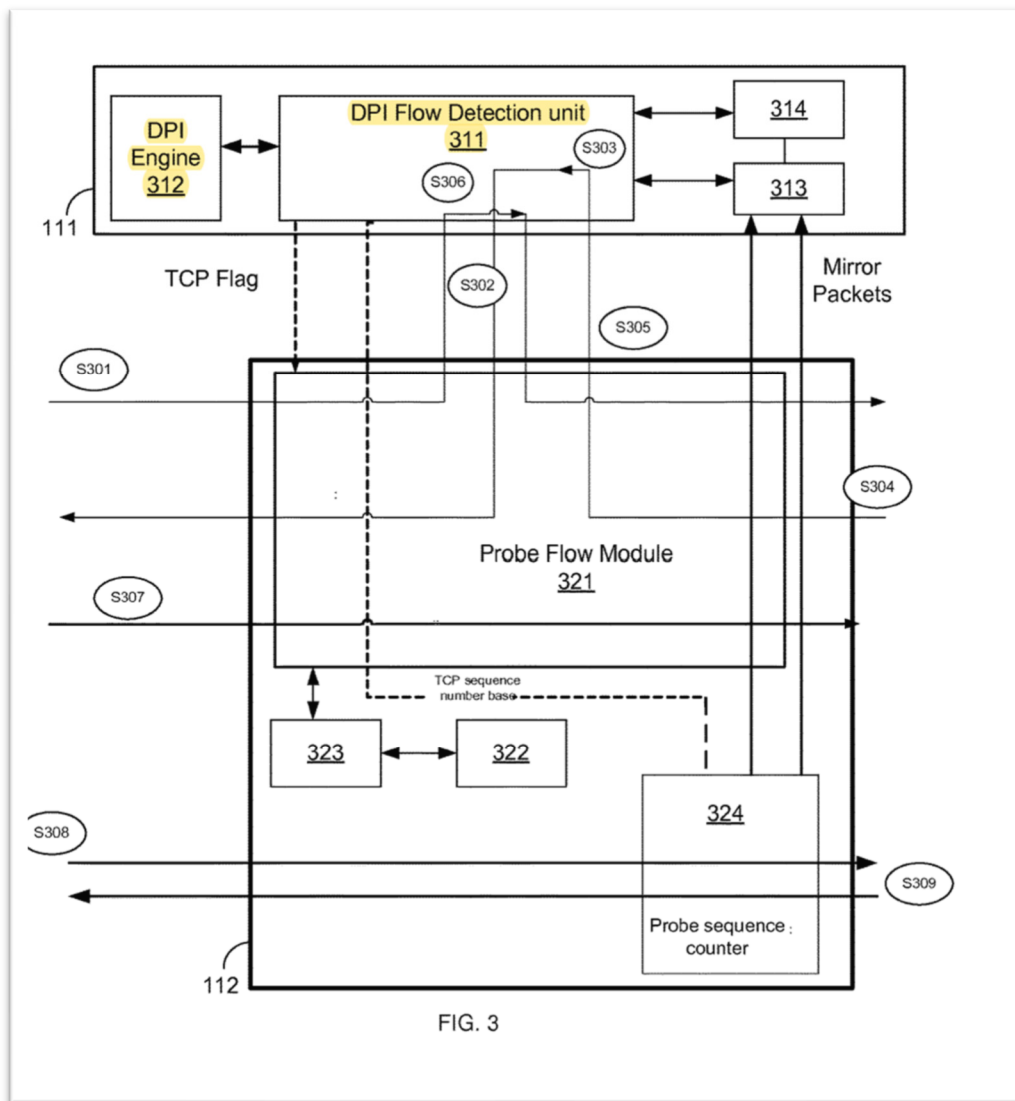


Figure 3 of the '111 Patent (Annotated)

EX1001, Figure 3; *see id.*, Figures 4-6.

Further, the claim term “instruction” should be construed to mean “a command to determine if a packet requires inspection or not.” EX1001, 10:56-62; EX1004, ¶72. The '111 Patent discloses that “each network node 112 is configured to determine if an incoming packet requires inspection or not.” EX1001, 4:12-14.

The patent states that “the central controller 111 is further configured to instruct each of the network nodes 112 which of the packets and/or sessions should be directed to the controller 111 for packet inspections.” Moreover, the exemplary instructions provided in the ’111 Patent are various commands used to determine whether or not a packet requires inspection. EX1001, 4:23-56; *see id.*, 8:23-32, 8:40-53, 9:26-28; EX1004, ¶72.

Terms not specifically construed have their plain and ordinary meaning as understood by a POSA. EX1004, ¶73.

VIII. THE ART PRIOR TO THE ’111 PATENT

A. Lin

Lin “relates generally to computer security, and more particularly but not exclusively to software defined networking.” EX1005, 1:7-9; *see id.*, Abstract; EX1004, ¶¶50-56.¹ “In one embodiment, a software defined networking (SDN) computer network includes an SDN controller and an SDN switch.” EX1005, 1:58-60; Figures 6-8. “The SDN controller inserts flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security

¹ Background discussion of software defined networking can be found in Paragraphs 21-29 of Dr. Bhattacharjee’s declaration and in EX1009.

component.” EX1005, 1:60-62; *see id.*, 1:62-64, 4:8-31, 4:53-65, 6:1-12. “The SDN pipe allows outgoing packets sent by the sender component to be received by the security component.” EX1005, 1:64-65; *see id.*, 3:25-31, 6:40-48. “The security component inspects the outgoing packets for compliance with security policies and allows the outgoing packets to be forwarded to their destination when the outgoing packets pass inspection.” EX1005, 1:66-2:2; *see id.*, 3:31-33, 6:48-63, 7:9-21.

Figure 6 of Lin, reproduced below, shows “a schematic diagram of an SDN computer network 600” in which “[t]he SDN controller 610 provides a logically centralized framework for controlling the behavior of the SDN computer network 600.” EX1005, 3:40-42, 4:7-9; *see id.*, 3:42-44, 4:9-12; EX1004, ¶51. “The SDN controller 610 may include a flow policy database 611.” EX1005, 4:12-13. “The flow policy database 611 may comprise flow policies that are enforced by the controller 610 on network traffic transmitted over the SDN computer network 600.” EX1005, 4:13-16; *see id.*, 4:16-18. “The flow policies may be enforced in terms of flow rules (labeled as 624) that are stored in the flow tables 621 of the SDN switch 620.” EX1005, 4:18-20.

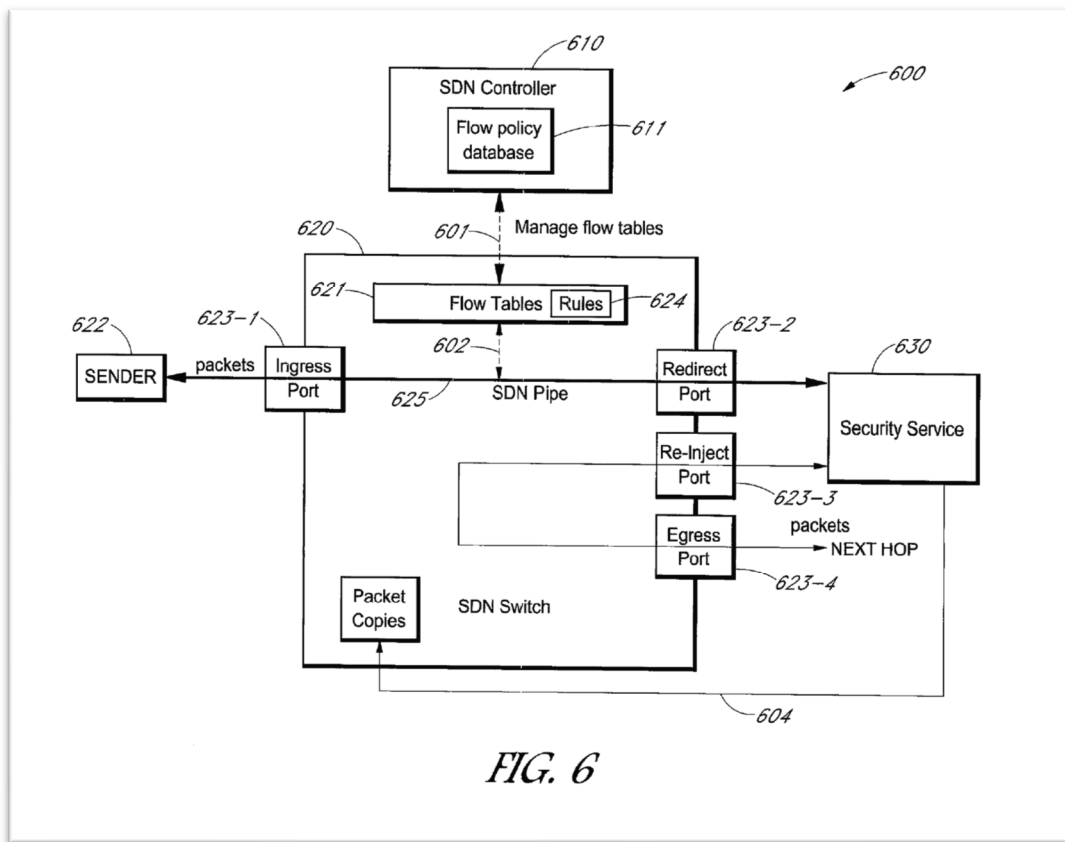


FIG. 6

Figure 6 of Lin

B. Shieh

Shieh relates “generally to network security” and discloses a “network system [that] includes a security device and a network access device.” EX1006, ¶[0002]; EX1004, ¶¶57-62. “The network access device is to receive a packet from a source node destined to a destination node, and to examine a data structure maintained by the network access device to determine whether the data structure stores a data member having a predetermined value, the data member indicating whether the packet should undergo security processing,” as shown below in Figure 1 of Shieh. EX1006, Abstract; *see id.*, ¶[0002], Figures 1, 2A, 3. “If the data member matches

the predetermined value, the packet is transmitted to a security device associated with the network access device to allow the security device to perform content inspection.” *Id.*, Abstract; *see id.*, ¶[0042], ¶[0049]. “[I]n response to a response received from the security device, the packet is routed to the destination node dependent upon the response.” *Id.*, Abstract; *see id.*, ¶[0017], ¶[0018], ¶[0023], ¶[0029], ¶[0037], Claim 1, Figure 2B; EX1004, ¶57.

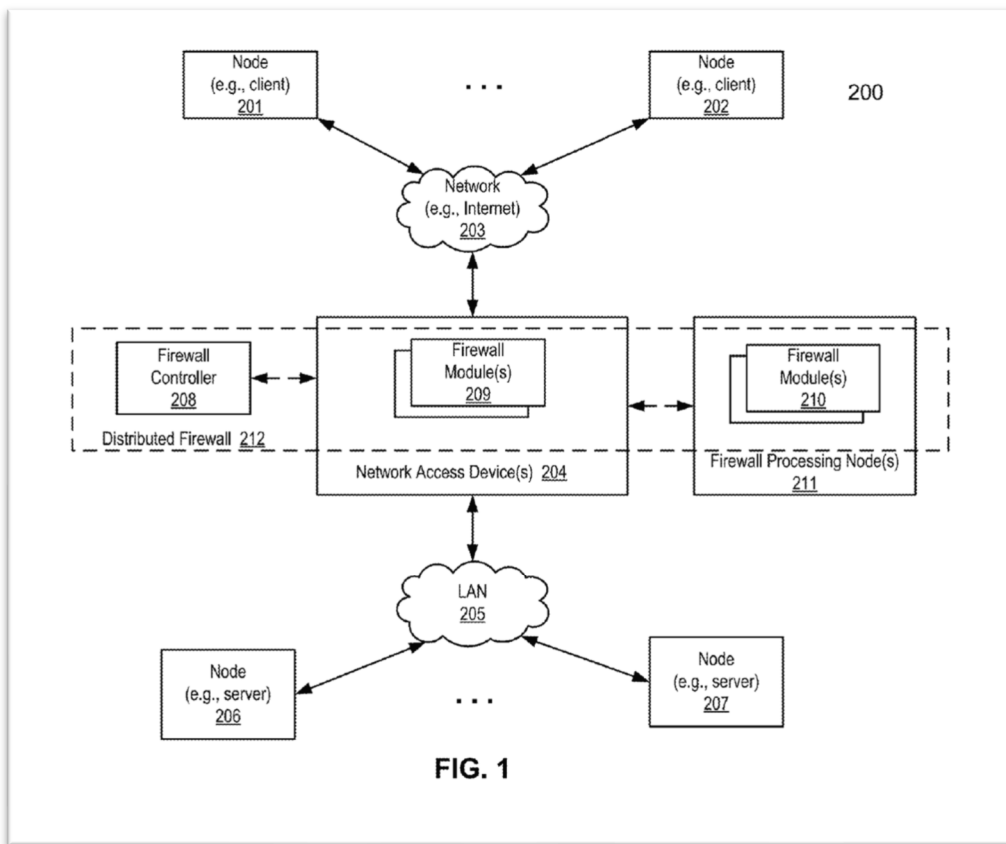


Figure 1 of Shieh

C. Swenson

Swenson discloses a system and method for “selectively monitoring traffic in a service provider network.” EX1007, Abstract; *see id.*, ¶¶[0018]-¶¶[0022]; EX1004, ¶¶63-68. Figure 1 of Swenson (reproduced below) shows that “[t]he network 120 is a communication network that transmits data between the user devices 110, the steering devices 130 and the origin server 160 and/or the video optimizer 150.” EX1007, ¶¶[0023]. “In one embodiment, the steering device 130 characterizes traffic routed through it to identify flows of interest for further inspection at the network controller 140.” *Id.*, ¶¶[0026]; *see id.*, ¶¶[0058]. “Alternatively, the network controller 140 interfaces with the steering device 130 to coordinate the monitoring and characterization of network traffic, such as identifying large and small objects in HTTP traffic flows.” *Id.*, ¶¶[0026]. “In this case, the steering device 130 receives instructions from the network controller 140 based on the desired criteria for characterizing flows of interest for further inspection.” *Id.*, ¶¶[0026]. When a flow matches a particular signature, “the steering device 130 forwards the HTTP request and a portion of the HTTP response to the network controller 140 over the [Internet content adaption protocol] client interface 404.” *Id.*, ¶¶[0059]; *see id.*, ¶¶[0060]. “After receiving the request and the portion of response at the ICAP server interface 406, the flow analyzer 312 of the network controller 140 performs a deep flow inspection

to determine if the flow is worth bandwidth monitoring and/or user detection.” *Id.*, ¶¶0059]; *see id.*, ¶¶0060]; EX1004, ¶¶67-68.

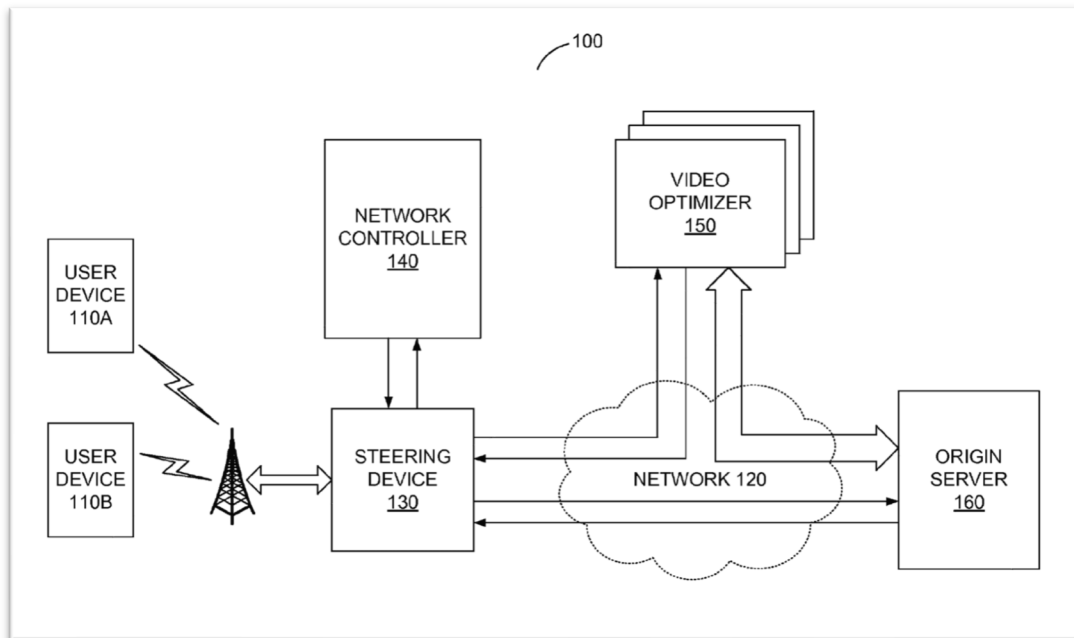


Figure 1 of Swenson

IX. GROUND 1: CLAIMS 1-9, 12-24 AND 27-31 ARE UNPATENTABLE AS OBVIOUS OVER LIN IN VIEW OF SWENSON.

The combination of Lin and Swenson, along with the knowledge of a POSA, renders Claims 1-9, 12-24 and 27-31 obvious. EX1004, ¶¶74-206.

A. Claim 1

[1.0] 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 comprising:

Element [1.0], to the extent it is limiting, is disclosed by Lin. EX1001, 10:51-

55.

Lin discloses a method for use with a packet network. For example, Lin's Abstract states that it relates to "[a] software defined networking (SDN) **computer network**." EX1005, Abstract (emphasis added). The specification further states that "[t]he present invention relates generally to computer security, and more particularly but not exclusively to software defined networking." *Id.*, 1:7-9; *see id.*, 1:58-60, 2:47-65, 3:25-33, 3:40-64. A POSA would have known that this computer network is a packet network. EX1004, ¶¶75-76. Indeed, Lin refers to the "transmission of packets over the SDN computer network 600." EX1005, 4:19-21. Moreover, Figure 6 of Lin shows that the system disclosed in Lin is for a packet network, as can be seen below:

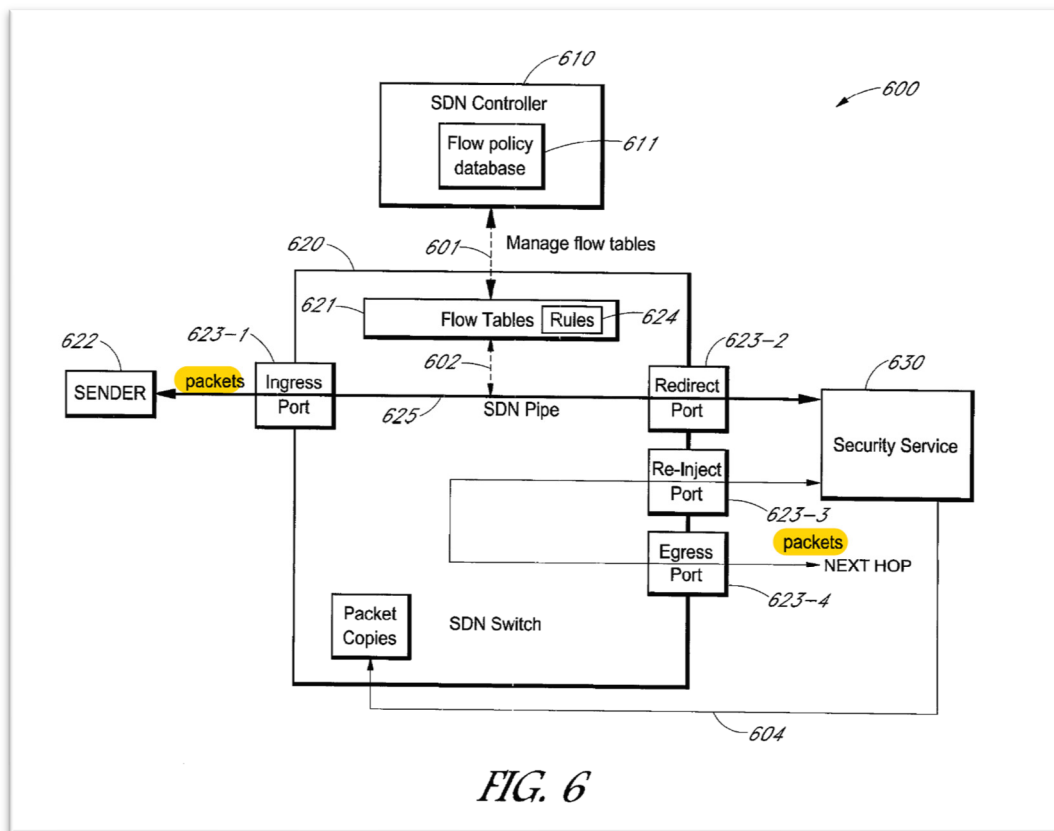


Figure 6 of Lin (Annotated)

EX1005, Figure 6; *see id.*, Figures 1-5, 7-9.

Further, Lin discloses an “SDN switch” that **corresponds to the claimed network node** for transporting packets between first and second entities. EX1005, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63; EX1004, ¶77. Lin explains that the SDN switch transports packets from a “sender” component (**which corresponds to the claimed first entity**), through an ingress port, out an egress port, and to the “next hop” or destination (**which corresponds to the claimed second entity**), as shown below in Figure 6. EX1005, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63, 7:10-23; 7:39-8:18, 9:63-10:22; EX1004, ¶77.

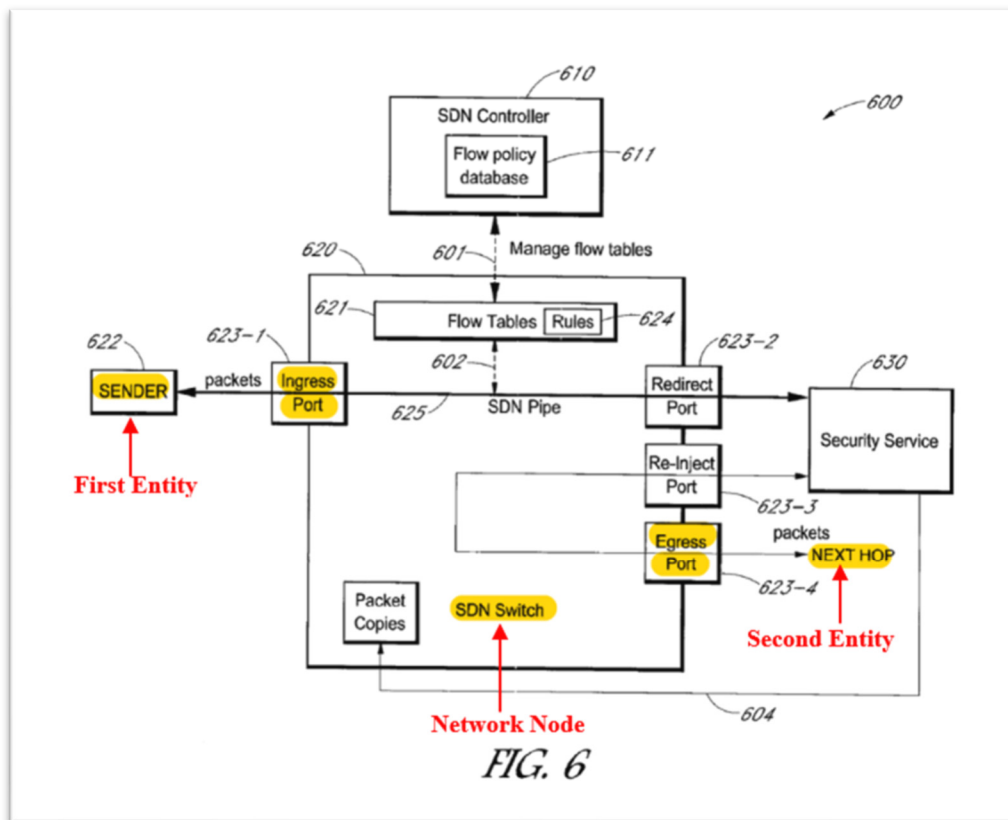


Figure 6 of Lin (Annotated)

EX1005, Figure 6.

Lin explains that the SDN switch is under the control of a “SDN controller” (which corresponds to the claimed controller) that is external to the SDN switch. EX1004, ¶78. Lin states that “the SDN controller 610 provides a logically centralized framework for controlling the behavior of the SDN computer network 600,” including one or more SDN switches. EX1005, 4:8-31. The SDN controller includes a “flow policy database” that contains flow policies to control the transmission of packets through the SDN switch. *Id.*, 4:8-31; *see id.*, 1:58-2:4, 6:1-12. Lin explains that the SDN controller is external to the SDN switch (i.e., the

network node): “The SDN controller 610 and the SDN switch 620 are logically separate components.” *Id.*, 3:51-52, *see id.*, 4:8-10. Further, Figures 1 and 6-8 of Lin show the SDN controller as external to the SDN switch:

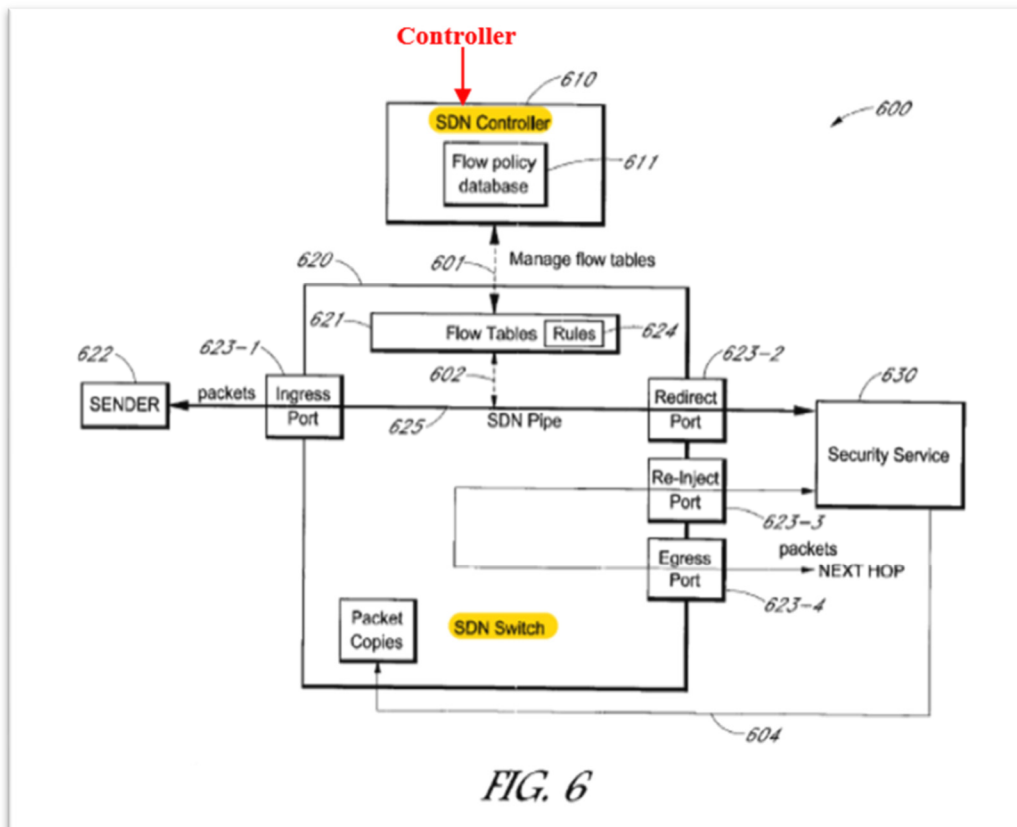


Figure 6 of Lin (Annotated)

Id., Figure 6; *see id.*, Figures 1, 7-8.

In addition, as addressed above in the claim construction section, the controller is an entity configured to perform DPI on packets.² EX1004, ¶79. The combination of Lin and Swenson discloses such a controller. Lin states that the analysis performed by security service 630 includes DPI: “Network security vendors provide network security services, such as firewall or deep packet inspection (DPI).” EX1005, 3:11-12. Moreover, Lin discloses that security service 630 “may also comprise a physical machine, e.g., a server computer, an appliance, or a gateway computer, etc.” EX1005, 5:51-55. Further, Lin states that “[t]he security service 630 may be connected to the SDN switch 620 by a physical link (i.e., using a wire), a virtual link (i.e., in a virtualized environment), or by a software tunnel.” A POSA would have known from these disclosures in Lin that the security service 630 can use the same hardware or software as the controller, and that the security service 630 can be connected to the SDN switch 620 in the same way as the controller. EX1004, ¶79. Thus, a POSA would have understood that one of the limited number of design options would have been to implement the security service as part of a controller configured to perform DPI analysis on packets, and a POSA would have had a

² To the extent that the PTAB does not agree with this construction, Lin still discloses Element [1.0] for the reasons discussed above.

reasonable expectation that the controller would have been successful in performing DPI analysis. EX1004, ¶79.

Further, Swenson teaches the use of a controller configured to perform DPI. EX1004, ¶80. Swenson discloses that, when its system detects a HTTP packet flow matching a particular signature, “the steering device 130 forwards the HTTP request and a portion of the HTTP response to the network controller 140 over the ICAP client interface 404.” EX1007, ¶[0059]; *see id.*, ¶[0060]. “After receiving the request and the portion of response at the ICAP server interface 406, **the flow analyzer 312 of the network controller 140 performs a deep flow inspection** to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059] (emphasis added); *see id.*, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B. A POSA would have known that a “flow” is a series of packets having a specific signature. EX1004, ¶80. As such, it would have been obvious to a POSA that Swenson’s reference to “deep flow inspection” refers to performing DPI on one or more packets in a flow. *Id.*

In addition, Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to

determine whether to monitor the flow or optimize the video.” EX1007, ¶[0065], Figure 5. A POSA would have known that the analysis of the “response payload” to be DPI by a DPI-capable controller, as DPI refers to monitoring the payload of a packet. EX1004, ¶81. Swenson thus demonstrates that it would have been well-known to a POSA as of the priority date for the ’111 Patent to implement a controller configured to perform DPI analysis in a system such as Lin. EX1007, ¶[0046], ¶[0059]-¶[0060], ¶[0073], ¶[0076]-[0077], ¶[0084]-¶[0086], Figures 1 and 4A-4B; EX1004, ¶81.

A POSA would have been motivated to combine the teachings of Lin and Swenson. EX1004, ¶¶82-87. Lin and Swenson are analogous art references that address the same technology and attempt to resolve the same issues relating to routing network traffic in an efficient manner that conserves network resources. EX1005, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; EX1004, ¶82. They each use a central controller to provide instructions and packet-applicable criterion to network nodes to determine which packets should be redirected and which packets can be sent directly to a destination node. EX1005, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; EX1004, ¶82. Lin provides a method in which a “SDN controller inserts flow rules in a flow table of the SDN switch” that

provide instructions and packet-applicable criterion to the SDN switch for use in identifying which packets should be sent to a security service. EX1005, 1:60-2:4, 3:21-24, 4:8-31, 4:53-67, 6:1-12, 6:54-63, Figure 6. Lin also discloses that the security service sends packets to their destination if the packets pass inspection. *Id.* Further, Lin explains that “bypass rules [that] are inserted in the flow tables 621 such that particular packets that do not need to be inspected are not redirected to the security service 630.” EX1005, 7:23-27; *see id.*, 9:12-16, Figure 9.

Similarly, Swenson discloses a method in which a network controller sends instructions to steering devices through which packet flows pass. EX1007, EX1007, ¶¶0023]-¶¶0032], ¶¶0038]-¶¶0043], ¶¶0057]-¶¶0061], Figures 1-4A; EX1004, ¶83. Swenson states that, when its systems detect a HTTP packet flow matching a particular signature, “the steering device 130 forwards the HTTP request and a portion of the HTTP response to the network controller 140 over the ICAP client interface 404.” EX1007, ¶¶0059]; *see id.*, ¶¶0060]. Like Lin, Swenson teaches that only certain packets are redirected to the controller in order to be efficient in analyzing packets of interest. EX1007, ¶¶0059]-¶¶0060]; EX1004, ¶83. Further, a POSA would have understood that Swenson, like Lin, could involve analyzing packets for a security function. EX1004, ¶83. A POSA would have known that the bandwidth monitoring in Swenson can be used as a security application that monitors for Denial of Service (“DOS”) attacks that occupy significant bandwidth in a

network. EX1007, ¶¶0059]-¶¶0060]; *see id.*, ¶¶0039] (explaining that the controller can incorporate “security functions”); EX1004, ¶83. Similar to Lin, Swenson teaches that packets can be routed or communicated to a destination node after the processing of those packets is completed. EX1007, ¶¶0060]. Moreover, the architecture of Swenson is substantially similar to the architecture of Lin, as shown in the below figures showing a central controller directly connected to a network node (the SDN switch in Lin and the steering device in Swenson) through which packet flows pass:

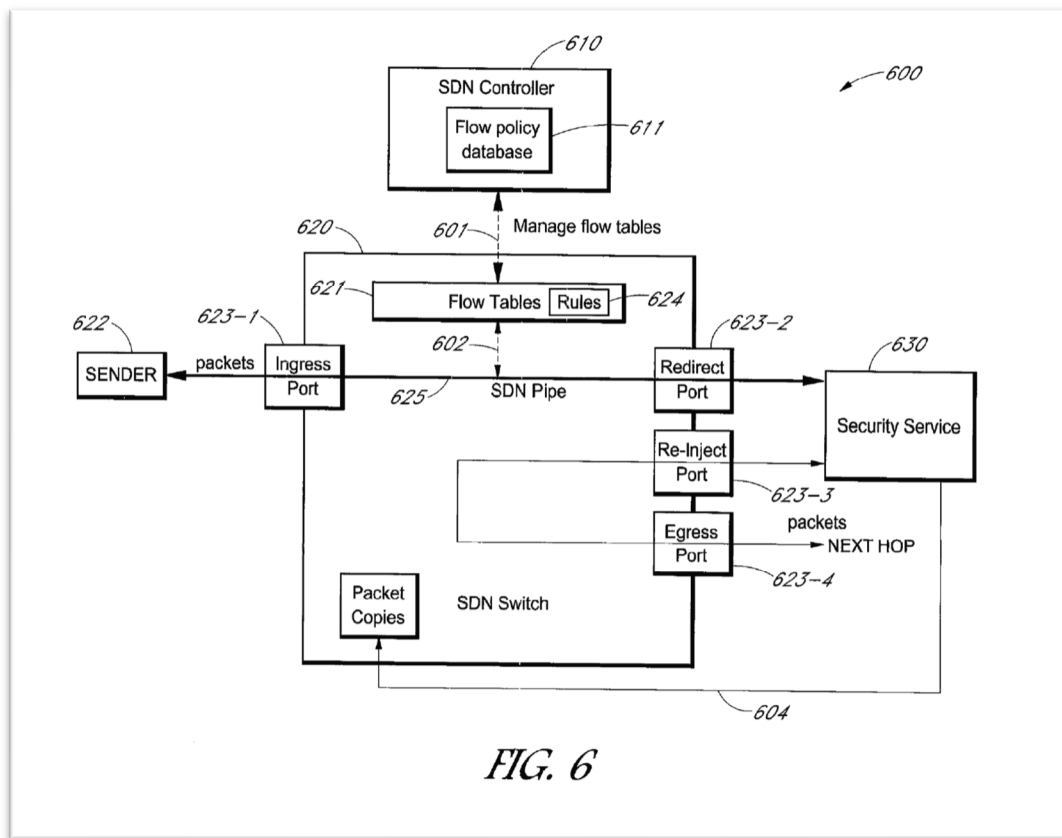


Figure 6 of Lin

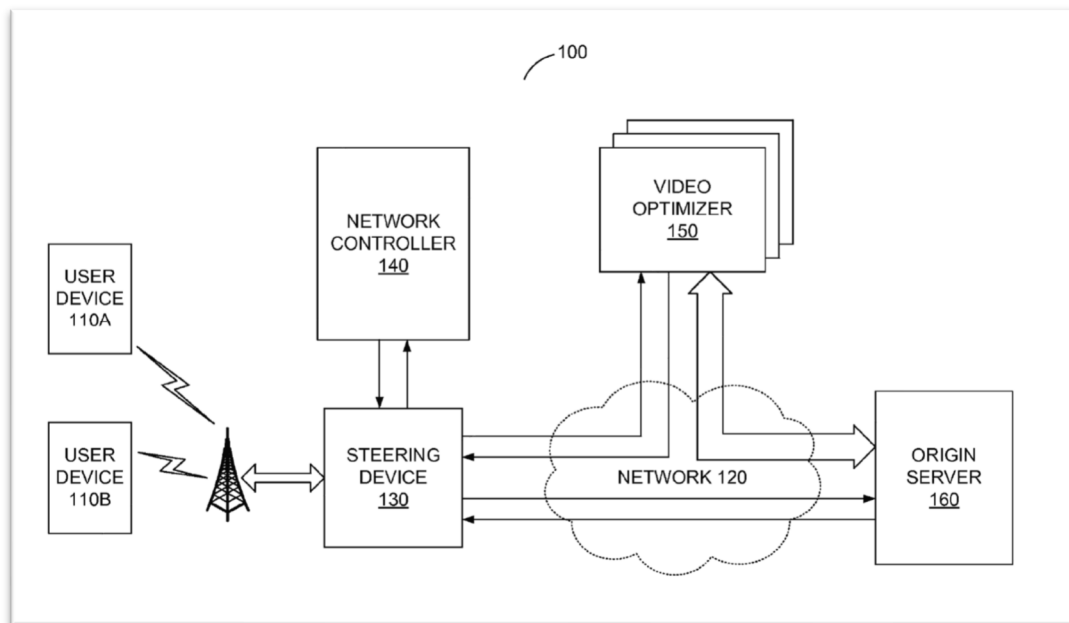


Figure 1 of Swenson

EX1004, ¶83.

The determination of which packets to redirect (as opposed to redirecting all packets) increases the efficient operation of a computer network. EX1004, ¶84. Lin and Swenson each teach that only certain packets may need to be redirected for further processing before being sent to the destination node. EX1005, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A. This allows the methods in Lin and Swenson to improve the operation of transporting packets across a network while maintaining a proper speed for processing those packets by allowing the system to stop redirecting certain packets in a flow if it is no longer necessary to do so. EX1004, ¶84.

Further, a POSA would have found it obvious to implement the security processing module in Lin as part of the controller in light of the disclosures in Swenson to send packets to the controller for DPI. EX1007, ¶¶[0059]-¶[0060], Figures 1, 4A-4B; EX1004, ¶85. A POSA would have understood there were efficiencies to implementing Lin's security processing module as part of the controller. EX1004, ¶85. For example, the use of a central location for packets from different nodes to undergo inspection by a security component allows the same security algorithms to be applied to each analyzed packet. Further, a POSA would have known that security algorithms in a security component are often updated via software updates as new threats are identified. *Id.* A POSA would have understood that an efficient way to keep the security component up-to-date would be to have a central security component that is part of the central processor. *Id.*

A POSA would have had a reasonable expectation of success in modifying Lin to implement the disclosure of Swenson to route packets to the central controller for inspection. EX1004, ¶86. As discussed above, Lin and Swenson have a similar architecture in which network nodes are controlled by a separate external controller. EX1005, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A. If Swenson could be constructed to have the network nodes send packets to the central controller for security inspection, a POSA would have understood that the same arrangement

could be implemented in Lin. EX1004, ¶86. Indeed, it would have been common sense for a POSA to modify Lin in this way after reading Swenson. *Id.*

[1.1] sending, by the controller to the network node over the packet network, an instruction and a packet-applicable criterion;

Element [1.1] is disclosed by Lin. EX1001, 10:56-58.

Lin explains that a controller (i.e., the SDN controller) controls network nodes (i.e., the SDN switches) in a packet network for the reasons stated above for Element [1.0]. EX1005, 1:58-2:4, 3:25-64; 4:8-31, 4:53-67, 6:1-12, 6:40-64, 7:39-8:18; Figures 6-9; EX1004, ¶¶88-89.

Lin also describes an instruction sent by the controller to the network node over the packet network that includes a command to determine whether or not a packet requires inspection. EX1004, ¶90. Specifically, the SDN controller inserts “flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.” EX1005, 1:58-2:4. **These flow rules correspond to the claimed instruction**, and they are sent to the SDN switches to provide instruction to the SDN switches on what packets should be sent to the security component for inspection. *Id.*; *see id.*, 4:14-31, 4:53-67, 6:1-12; 7:39-8:18.

Further, Lin explains that a separate packet-applicable criterion sent by the controller to the network node over the packet network. EX1004, ¶91. Lin describes that the flow rules “may indicate inspection of particular packets (e.g., those that

meet one or more conditions) by a security service 630.” EX1005, 4:21-31; *see id.*, 1:60-62, 5:8-12. A POSA would have known that the “one or more conditions” used to identify “particular packets” for inspection **would be packet-applicable criterion**. EX1004, ¶¶91-92; EX1005, 4:23-31, 5:8-12.

Indeed, Lin discloses examples of the criterion applicable to a particular packet, including “‘IN_PORT’, ‘MAC src’ (media access control (MAC) address of the source of the packet), ‘MAC dst’ (MAC address of the destination of the packet), ‘IP src’ (Internet Protocol (IP) address of the source of the packet), ‘IP dst’ (IP address of the destination of the packet).” EX1005, 5:16-21; EX1004, ¶91. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, 5:22-24; *see id.*, 5:26-36, 6:1-12, 6:40-54, Table 1. Further, in discussing the examples shown in Table 2 and Table 3, Lin discloses “bypass flow rules” where packet-applicable criterion are used to identify HTTP packets. *Id.*, 2:2-4, 7:24-8:18.

Further, Lin explains that the flow rules containing the packet-applicable criterion are sent by the SDN controller to the flow table in the SDN switch. EX1004, ¶93. Lin states “[t]he SDN controller 610 may insert flow rules in the flow tables 621 (see arrow 601) to create an SDN pipe (labeled as 625) between the sender component 622 and the security service 630,” as shown below in Figure 6. EX1005, 6:1-4; *see id.*, 6:40-41. With respect to Figure 6, Lin discloses an embodiment in

which “bypass flow rules are inserted in the flow tables 621 such that particular packets that do not need to be inspected are not redirected to the security service 630.” *Id.*, 7:24-26; *see id.*, 7:27-8:18, Tables 2-3. A POSA would have understood from these disclosures that Lin discloses a packet-applicable criterion sent by the controller to the network node (i.e., the SDN switch) over the packet network. EX1004, ¶¶93-94.

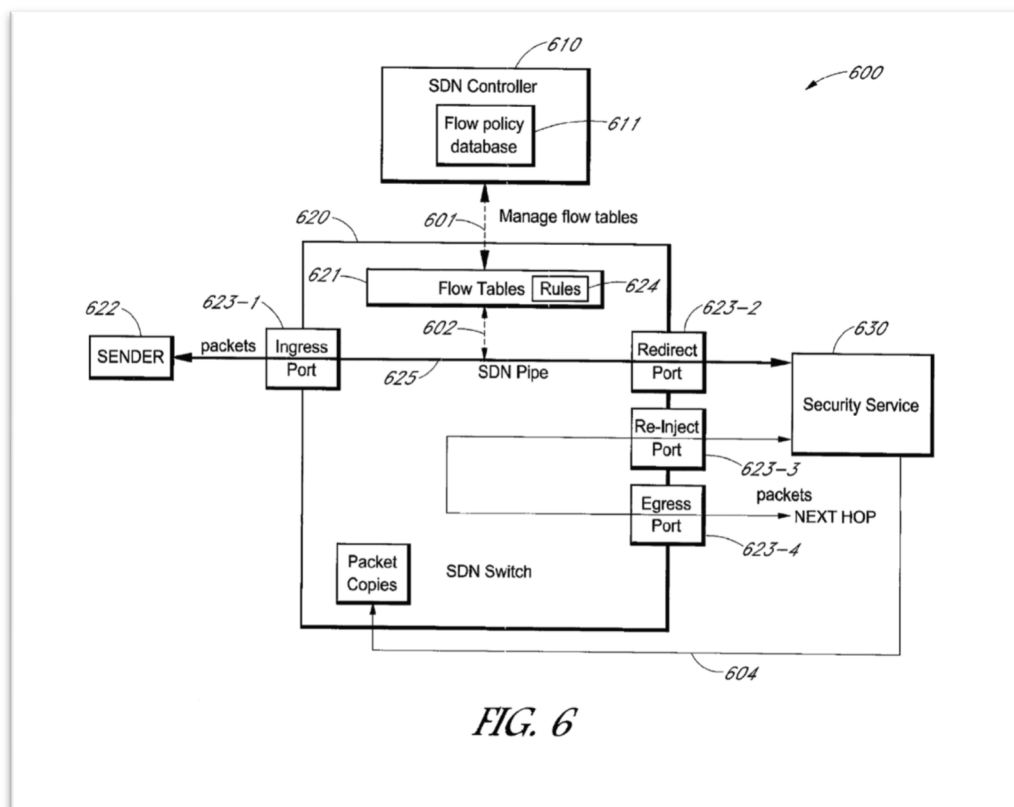


Figure 6 of Lin

[1.2] receiving, by the network node from the controller, the instruction and the criterion;

Element [1.2] recites that the instruction and packet-applicable criterion that are sent by the controller to the network node in Element [1.1] are received by the

network node. EX1001, 10:59-60. As such, Lin discloses Element [1.2] for substantially the same reasons that Lin discloses Element [1.1]. EX1004, ¶¶95-100.

[1.3] receiving, by the network node from the first entity over the packet network, a packet addressed to the second entity;

Element [1.3] is disclosed by Lin. EX1001, 10:60-62.

Lin discloses that the SDN switch (i.e., the network node) receives packets from a “sender” component (which corresponds to the claimed first entity) and ultimately sends at least some packets to the “next hop” or destination (which corresponds to the claimed second entity), as shown below in Figure 6. EX1005, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63, 7:10-23; 7:39-8:18, 9:63-10:22; EX1004, ¶¶101-103. A POSA would have understood that the packet is addressed to the next hop or second entity. EX1004, ¶102. The IP address of the destination of the packet is one of the criterion that is specifically discussed in Lin. EX1005, 5:16-31.

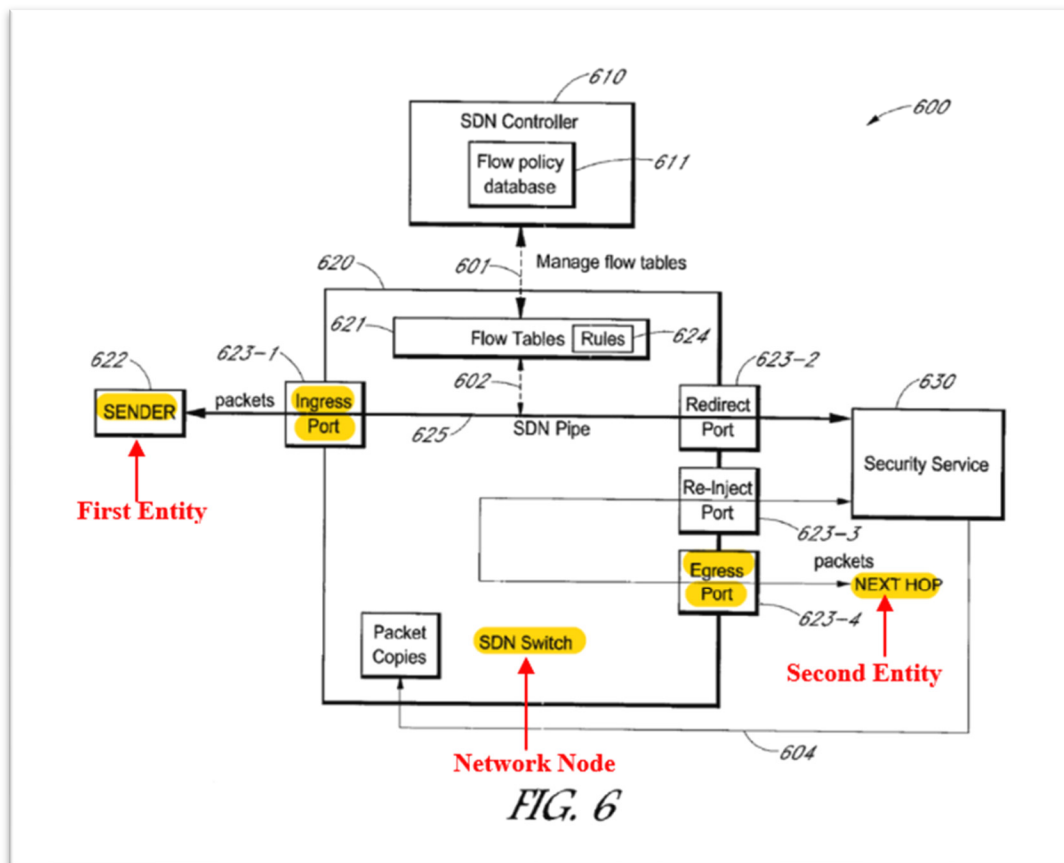


Figure 6 of Lin (Annotated)

Id., Figure 6.

[1.4] checking, by the network node, if the packet satisfies the criterion;

Element [1.4] is disclosed by Lin. EX1001, 10:63-64.

As discussed above for Element [1.1], Lin discloses that its SDN switch implements flow rules that “may indicate inspection of particular packets (e.g., those that meet one or more conditions) by a security service 630.” EX1005, 4:23-31; *see id.*, 5:8-12, 6:1-4, 6:40-41, 7:24-8:18. A POSA would have understood this disclosure to mean that the “one or more conditions” used to identify “particular

packets” for inspection would be packet-applicable criterion, and that the SDN switch checks to determine whether incoming packets satisfy the packet-applicable criterion. EX1004, ¶104-105; EX1005, 4:23-31, 5:8-12. Indeed, Lin provides examples of the conditions (or criterion) applicable to a particular packet that are checked by the SDN switch, including “IN_PORT,” “MAC src,” “MAC dst,” “IP src,” and “IP dst.” EX1005, 5:16-21; EX1004, ¶105. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, 5:22-24; *see id.*, 5:26-36, 6:1-12, 6:40-54, Table 1. Further, in discussing the examples shown in Table 2 and Table 3, Lin discloses that the SDN switch implements “bypass flow rules” that check whether the packet meets certain packet-applicable criterion, such as identification of HTTP packets, which indicate that the packet should be routed to the destination node instead of the security device. *Id.*, 7:24-8:18; EX1004, ¶¶105-106.

[1.5] responsive to the packet not satisfying the criterion, sending, by the network node over the packet network, the packet to the second entity; and

Element [1.4] is disclosed by Lin. EX1001, 10:65-67.

Lin explains that the SDN switch implements “bypass flow rules” that check whether a packet meets certain packet-applicable criterion. EX1005, 7:24-8:18; *see id.*, 5:16-24, 5:26-36, 6:1-12, 6:40-54; EX1004, ¶¶107-109. In certain embodiments

of the bypass flow rules, if the packet does not satisfy the packet-applicable criterion, the packet is routed to the destination node (i.e., the second entity) instead of the security device. EX1005, 7:24-8:18. For example, Lin’s discussion of Table 3 teaches that the SDN switch checks for a specific packet-applicable criterion – that port 80 is the source or destination port of the packets, which indicates that they are HTTP packets. *Id.*, 7:64-8:18. Lin states that, if a packet does not satisfy this criterion (i.e., the packet does not indicate that port 80 is the source or destination port), then the SDN switch (which corresponds to the claimed network node) sends the packet over the packet network to its destination node (which corresponds to the claimed second entity). *Id.*, 8:10-18; EX1004, ¶¶109-110. Specifically, with respect to Table 3 (reproduced below), Lin explains that “the bottom two rows [highlighted below] are bypass flow rules” that cause non-HTTP packets to “bypass the SDN pipe.”

TABLE 3							
IN_PORT	...	IP src	TCP src port	TCP dst port	...	Action	Count
Ingress_port_ID	*	*	*	80	*	Redirect port	10
Redirect_port_ID	*	*	80	*	*	Ingress port	10
Ingress_port_ID	*	*	*	*	*	Egress port	130
Egress_port_ID	*	*	*	*	*	Ingress port	130

Table 3 of Lin (Annotated)

EX1005, 8:10-18, Table 3.

[1.6] responsive to the packet satisfying the criterion, sending the packet, by the network node over the packet network, to an entity that is included in the instruction and is other than the second entity.

Element [1.6] is disclosed by Lin. EX1001, 11:1-4.

As discussed above for Element [1.5], Lin discloses that the SDN switch implements “bypass flow rules” that check whether a packet meets certain packet-applicable criterion. EX1005, 7:24-8:18; *see id.*, 5:16-24, 5:26-36, 6:1-12, 6:40-54; EX1004, ¶¶111-114. Further, in some embodiments, the bypass flow rules provide that packet which meet certain packet-applicable criterion should be routed to the security device. EX1005, 7:24-8:18. For example, Lin’s discussion of Table 3 teaches that the SDN switch checks for a specific packet-applicable criterion – that port 80 is the source or destination port of the packets, which indicates that they are HTTP packets. *Id.*, 7:64-8:18. Lin states that, if a packet satisfies this criterion (i.e., the packet indicates that port 80 is the source or destination port), then the SDN switch (which corresponds to the claimed network node) sends the packet over the packet network to the security device (which corresponds to the claimed entity that is included in the instruction and is other than the second entity). *Id.*, 8:10-18, Table 3 (reproduced below.)

TABLE 3							
IN_PORT	...	IP src	TCP src port	TCP dst port	...	Action	Count
Ingress_port_ID	*	*	*	80	*	Redirect port	10
Redirect_port_ID	*	*	80	*	*	Ingress port	10
Ingress_port_ID	*	*	*	*	*	Egress port	130
Egress_port_ID	*	*	*	*	*	Ingress port	130

Table 3 of Lin (Annotated)

Id., 8:10-18 (emphasis added), Table 3; EX1004, ¶113.

B. Claim 2

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

Claim 2 is obvious in view of Lin and the knowledge of a POSA. EX1001, 11:5-10.

Lin teaches that the instruction can be a “terminate” instruction. EX1004, ¶¶115-116. A POSA would have understood that a terminate instruction was a well-known instruction that drops packets from the network such that they are no longer forwarded to a destination within the network. *Id.* Lin discloses that one of its “packet manipulation actions” is “dropping the packet.” EX1005, 1:28-32. Lin also states that the security service 630 “may instruct the SDN switch 620 to drop the copied packets...” *Id.*, 7:19-22. A POSA would have understood these passages

from Lin to teach that the flow rules in Lin could include a terminate instruction. EX1004, ¶116.

Where the SDN switch in Lin receives a terminate instruction, a POSA would have understood that the terminate instruction easily could be implemented such that the SDN switch blocks a packet from being sent to the destination node (i.e., the claimed second entity) and the controller. EX1004, ¶117. Lin states that the SDN switch can be instructed to drop (i.e., block) packets before the packets are sent to their destination node. EX1005, 1:28-32, 7:19-22. It would have been straightforward for the flow rules in Lin to also instruct that packets are dropped by the SDN switch before they can be sent to controller. EX1004, ¶¶117-118. For example, a POSA would have known to implement a terminate instruction to drop (i.e., block) packets received at the SDN switch from an IP address that is known to originate malicious code, and thereby prevent those packets from being sent to the controller (or destination node) where the malicious code could do damage. *Id.*

C. Claim 3

[3] The method according to claim 1, wherein the instruction is a ‘probe’, a ‘mirror’, or a ‘terminate’ instruction, and upon receiving by the network node the ‘mirror’ instruction and responsive to the packet satisfying the criterion, the method further comprising sending the packet, by the network node, to the second entity and to the controller.

Claim 3 is obvious in view of Lin. EX1001, 11:11-16.

Further, Lin teaches that the instruction can be a “mirror” instruction. EX1004, ¶¶119-120. A POSA would have understood that a mirror instruction was a common instruction that can operate in one of two ways: (1) redirecting a packet from its intended destination to a new destination (such as a security service) or (2) making a copy of a packet and sending the copy of the packet to a new destination (such as a security service). *Id.* Lin discloses both methods of mirroring a packet. For example, Lin states that incoming packets can be “redirected or mirrored to the security service.” EX1005, 3:25-33. Lin also discloses that a copy of a packet can be made, the packet mirrored to the security service, and an action (such as drop, forward or quarantine) taken on the copied packet. *Id.*, 7:10-22. A POSA would have understood these passages from Lin to teach that the flow rules in Lin could include a mirror instruction. EX1004, ¶120.

Where the SDN switch in Lin receives a mirror instruction, and the packet satisfies the criterion (as discussed above for Element [1.6]), Lin teaches that the mirror instruction instructs that “the SDN switch 620 may be configured to copy packets that are redirected to the security service 630 for inspection” and to “forward the copied packets to their destinations.” EX1005, 7:10-22. As discussed above for Element [1.6], it would have been obvious to implement the security service 630 in Lin as part of the controller; thus, the original packet would be sent to the controller

by the SDN switch and a copy of the packet is sent to the destination node. *Id.*; EX1004, ¶¶121-123.

D. Claim 4

[4] The method according to claim 1, wherein the instruction is ‘probe’, ‘mirror’, or ‘terminate’ instruction, and upon receiving by the network node the ‘probe’ instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet, by the network node, to the controller; responsive to receiving the packet, analyzing the packet, by the controller; sending the packet, by the controller, to the network node; and responsive to receiving the packet, sending the packet, by the network node, to the second entity.

Claim 4 is obvious over Lin, in view of the knowledge of a POSA. EX1001, 11:17-26.

Lin teaches that the instruction can be a “probe” instruction. EX1004, ¶¶124-125. A POSA would have understood that a probe instruction was a well-known instruction to send an incoming packet to a security service so that the packet can be probed or inspected for malicious code. *Id.* As discussed above for Element [1.1], Lin describes the SDN switch (i.e., the network node) receiving an instruction in the form of a flow rule that instructs the SDN switches that packets should be sent to the security component for analysis. EX1005, 1:58-2:4; *see id.*, 4:14-31, 4:53-67, 6:1-12; 9:23-40, Figures 6-9. Lin teaches an example wherein the flow rules received by the SDN switch instruct the SDN switch to redirect non-HTTP packets to the security service 630 for inspection. *Id.*, 7:39-63; *see id.*, 7:64-8:18 (instructing the

SDN switch to redirect “HTTP packets to the security service 630 for inspection”). A POSA would have understood from these disclosures in Lin that the flow rules sent to the SDN switch include a probe instruction for sending the packets to the security component. EX1004, ¶125.

Where the SDN switch in Lin receives a probe instruction, and the packet satisfies the criterion (as discussed above for Element [1.6]), Lin teaches that the probe instruction instructs the SDN switch to send the packet to the security service 630 for inspection. EX1005, Abstract, 1:58-2:4, 4:53-67, 5:26-36, 6:1-12, 6:40-63, 7:10-8:18, Figures 6-9. It would have been obvious in view of Swenson to implement the security service 630 in the controller (as discussed above for Element [1.0]); thus, the packet is sent to the controller for inspection. EX1004, ¶126.

Further, a POSA would have known from Lin that, when the packet is received by the controller, the packet is analyzed by the security service in the controller. EX1004, ¶127; EX1005, 6:48-50, 5:45-55, 7:10-22, 8:33-35, Figure 9. A POSA also would have known that, when the analysis is complete, the packet is sent from the controller back to the SDN switch and on to the destination node (i.e., the second entity). EX1004, ¶127; EX1005, 6:54-63, 7:10-22, 8:35-45, Figure 9.

E. Claim 5

[5] The method according to claim 1, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof, by the network node, to the controller.

Claim 5 is obvious in view of the combination of Lin and Swenson. EX1001, 11:27-30.

As discussed above for Claim 1, Lin discloses that a network node has an instruction and packet-applicable criterion and sending the packet to the controller responsive to the instruction and to the packet satisfying the packet-applicable criterion. EX1004, ¶¶129-130.

Moreover, Swenson teaches that the network node sends a packet or a portion thereof to the controller in response to the packet satisfying a criterion. EX1004, ¶131. As discussed above for Element [1.0], Swenson discloses that, when its system detects a HTTP packet flow matching a particular signature, “the steering device 130 **forwards the HTTP request and a portion of the HTTP response to the network controller 140** over the ICAP client interface 404.” EX1007, ¶[0059] (emphasis added); *see id.*, ¶[0060], ¶[0065], Figures 1, 4A-4B. A POSA would have understood this to mean that the steering device 130 corresponds to at least part of the claimed network node, and the signature that is compared to the HTTP packet flow corresponds to the claimed criterion. EX1004, ¶¶131-132. Responsive to the HTTP packet flow satisfying that criterion, the steering device applies an instruction pursuant to which the HTTP request and a portion of the HTTP response is sent to the network controller 140. EX1007, ¶[0059].

F. Claim 6

[6] The method according to claim 5, further comprising storing the received packet or a portion thereof, by the controller, in a memory.

Claim 6 is obvious over Swenson in view of the knowledge of a POSA.
EX1001, 11:31-33.

Swenson discloses a flow cache memory 322 that is part of the network controller 140, as shown below in Figure 3 of Swenson. EX1004, ¶¶133-134. Further, Swenson states that “[o]nce a flow is reported to the network controller 140, a flow cache entry is created for the flow in the flow cache 322.” EX1007, ¶[0061].

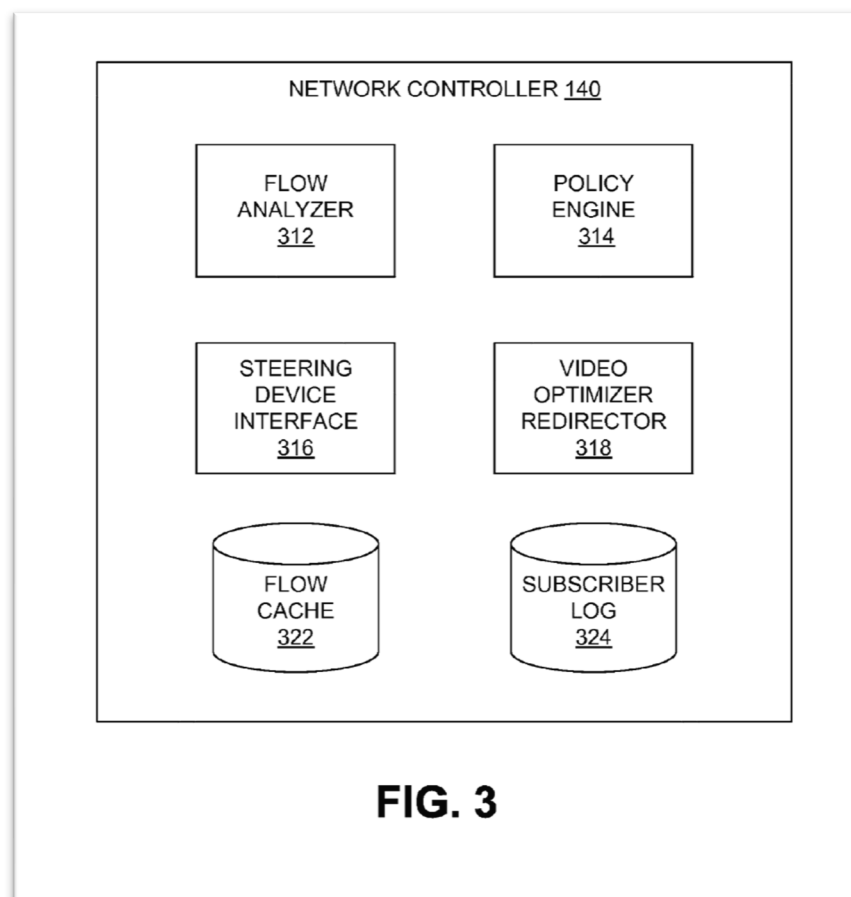


Figure 3 of Swenson

Swenson also explains that “[t]he flow cache entry keeps track of the flow and its associated bandwidth.” EX1007, ¶[0061]; *see id.*, ¶[0040], ¶[0046], ¶[0062]. In discussing Figure 7, Swenson states that the flow cache entry stores at least a portion of the received packet, including at least the IP address of the received packet. *Id.*, ¶[0073] (discussing Figure 7); *see id.*, ¶[0074]-¶[0085]. Further, Swenson states that a flow cache entry can also store the MAC address or TCP source port associated with a received packet. *Id.*, ¶[0084]. A POSA would have understood from these disclosures that Swenson teaches that the controller stores at least a portion of a packet in a flow cache memory. EX1004, ¶¶134-135.

G. Claim 7

[7] The method according to claim 5, further comprising responsive to the packet satisfying the criterion and to the instruction, sending a portion of the packet, by the network node, to the controller.

Claim 7 is obvious over Swenson. EX1001, 11:34-37.

As discussed above for Element [1.0] and Claim 5, Swenson teaches that the network node sends a portion of a packet to the controller in response to the packet satisfying a criterion. Swenson discloses that, when its system detects a HTTP packet flow matching a particular signature, “the steering device 130 **forwards the HTTP request and a portion of the HTTP response to the network controller 140** over the ICAP client interface 404.” EX1007, ¶[0059] (emphasis added); *see id.*, ¶[0060], ¶[0065], Figures 1, 4A-4B. A POSA would have understood that the steering device

130 corresponds to at least part of the claimed network node, and the signature that is compared to the HTTP packet flow corresponds to the claimed criterion. EX1004, ¶¶136-137. Further, a POSA would have understood that, responsive to the HTTP packet flow satisfying that criterion, the steering device applies an instruction pursuant to which a portion of a HTTP request or a portion of a HTTP response is sent to the network controller. EX1004, ¶¶137-138.

H. Claim 8

[8] The method according to claim 7, wherein the portion of the packet consists of multiple consecutive bytes, and wherein the instruction comprises identification of the consecutive bytes in the packet.

Claim 8 is obvious over Swenson in view of the knowledge of a POSA. EX1001, 11:38-41. As discussed above for Claim 7, Swenson discloses that a portion of the HTTP request or the HTTP response is sent to the controller. EX1007, ¶¶0059]-¶[0060], Figures 1, 4A-4B. It would have been obvious to a POSA that the portion of the HTTP request or the HTTP response sent to the controller consists of multiple consecutive bytes in the packet that were identified by an instruction implemented in steering device 130 of Swenson. EX1004, ¶¶139-141.

I. Claim 9

[9] The method according to claim 5, further comprising responsive to receiving the packet, analyzing the packet, by the controller.

Claim 9 is obvious in view of Swenson. EX1001, 11:42-44.

As discussed above for Element [1.0], Claim 5 and Claim 7, Swenson discloses that packets are sent to the network controller. Swenson also explains that, after receiving the packet, the flow analyzer 312 of the network controller analyzes the packet through DPI:

After receiving the request and the portion of response at the ICAP server interface 406, **the flow analyzer 312 of the network controller 140 performs a deep flow inspection** to determine if the flow is worth bandwidth monitoring and/or user detection.

EX1007, ¶[0059] (emphasis added); *see id.*, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B; EX1004, ¶¶142-143. Further, Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” *Id.*, ¶[0065], Figure 5. A POSA would have understood the inspection of the “HTTP GET request header and a portion of the [HTTP] response payload” to be analyzing the packet upon the controller receiving the packet. EX1004, ¶143. This is similar to Lin, which explains that an analysis is performed on its packets. EX1005, 1:58-2:4, 3:25-33, 4:8-31, 4:61-65, 5:26-36, 5:45-51, 6:1-12, 6:50-57, 7:59-61, 8:12-16, Figures 2, 5-9; EX1004, ¶¶143-144.

J. Claim 12

[12] The method according to claim 9, wherein the analyzing comprises applying security or data analytic application.

Claim 12 is obvious over Lin. EX1001, 11:55-57. Lin states that the analysis performed on the packets includes a security processing function. EX1005, 1:58-2:4, 3:25-33, 4:8-31, 4:61-65, 5:26-36, 5:45-51, 6:1-12, 6:50-57, 7:59-61, 8:12-16, Figures 2, 5-9; EX1004, ¶¶145-148.

K. Claim 13

[13] The method according to claim 9, wherein the analyzing comprises applying security application that comprises firewall or intrusion detection functionality.

Claim 13 is obvious over Lin. EX1001, 11:58-60. Lin discloses that its system analyzes packets by applying a security application that comprises either a firewall or intrusion detection functionality, or both. EX1005, 5:45-50 (referring specifically to “compliance with firewall rules” and “network intrusion detection”), 3:11-12, 6:50-54; EX1004, ¶¶149-151.

L. Claim 14

[14] The method according to claim 9, wherein the analyzing comprises performing Deep Packet Inspection (DPI) or using a DPI engine on the packet.

Claim 14 is obvious over both Lin and Swenson. EX1001, 11:61-63. Lin states that its analysis includes “deep packet inspection (DPI).” EX1005, 3:11-12; EX1004, ¶¶152-155. As discussed above for Element [1.0], Swenson also discloses

that the flow analyzer in network controller 140 performs DPI. EX1007, ¶¶0059]-¶¶0060], ¶¶0065].

M. Claim 15

[15] The method according to claim 9, wherein the packet comprises distinct header and payload fields, and wherein the analyzing comprises checking part of, or whole of, the payload field.

Claim 15 is obvious in view of both Lin and Swenson. EX1001, 11:64-67.

A POSA would have known that a packet includes distinct header and payload fields. EX1008, Section 2; EX1004, ¶¶156-157.

Further, Swenson discloses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶¶0065], Figure 5; EX1004, ¶158. A POSA would have understood the analysis of the “response payload” in Swenson to be checking part of the payload field of a packet by the network controller. Further, a POSA would have understood the references to DPI in both Swenson and Lin to involve checking part or the whole of the payload field. EX1004, ¶¶159-160; EX1005, 3:11-12, 5:45-50; EX1007, ¶¶0059]-¶¶0060].

N. Claim 16

[16] The method according to claim 1, 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.

Claim 16 is obvious over Lin in view of the knowledge of a POSA. EX1001, 12:1-4. A POSA would have known that a packet includes distinct header and payload fields. EX1008, Section 2; EX1004, ¶¶161-162. Further, it would have been well-known to a POSA that the header field includes one or more flag bits. EX1004, ¶¶162-163; EX1008, Section 2. In addition, it would have been obvious to a POSA that the flow rules of Lin could have used a determination of whether or not a flag bit was set as one of Lin's "conditions" that serve as packet-applicable criterion. EX1004, ¶¶162-163; EX1005, 4:23-31; *see id.*, 5:8-12, 7:24-8:18.

O. Claim 17

[17] The method according to claim 16, 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.³

Claim 17 is obvious over Lin in view of the knowledge of a POSA. EX1001, 12:5-9. Lin discloses embodiments in which the packet is a TCP packet. EX1005, 7:24-8:18, Tables 2-3. Further, it would have been well-known to a POSA that flag bits such as a SYN flag bit, an ACK flag bit, a FIN flag bit, and a RST flag bit were

³ Claim 17 uses the language "comprises comprise." This may be an error. For this Petition, Petitioner interprets that claim language to mean "comprises."

commonly used as part of the “handshake” process to initiate transmission of TCP packets (in the case of SYN and ACK flag bits) and to reset or terminate a TCP connection (in the case of RST and FIN flags, respectively). EX1004, ¶¶164-165. As such, it would have been obvious to a POSA that the flow rules of Lin could have used a determination of whether or not one or more of these flag bits was set as one of Lin’s “conditions” that serve as packet-applicable criterion. EX1004, ¶¶165-166; EX1005, 4:23-31; *see id.*, 5:8-12, 7:24-8:18.

P. Claim 18

[18] The method according to claim 1, 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.

Claim 18 is obvious over Lin in view of the knowledge of a POSA. EX1001, 12:10-16. A POSA would have known that a packet includes distinct header and payload fields. EX1008, Section 2; EX1004, ¶¶167-168. Moreover, it would have been well-known that the header would include the IP address of the source of the packet and the IP address of the destination of the packet. EX1004, ¶168. Further, as disclosed above for Element [1.1], Lin provides an example where the packet-applicable criterion is that the address of the first entity (“MAC src” or “IPsrc”), the address of the second entity (“MAC dst” or “IP dist”), or both match a predetermined

address or addresses to identify a particular packet. EX1005, 5:8-25; *see id.*, 5:26-36, 6:1-12, 6:40-54, Table 1; EX1004, ¶¶169-170.

Q. Claim 19

[19] The method according to claim 18, wherein the addresses are Internet Protocol (IP) addresses.

Claim 19 is obvious over both Lin for the reasons as those discussed above for Claim 18. EX1001, 12:17-18; EX1005, 5:8-36, 6:1-12, 6:40-54, Table 1; EX1004, ¶¶171.

R. Claim 20

[20] The method according to claim 1, 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.

Claim 20 is obvious over Lin in view of the knowledge of a POSA. EX1001, 12:20-27.

Lin provides examples of TCP packets that are transported over a TCP network when discussing Tables 2 and 3 of that patent. EX1005, 7:24-8:18; *see id.*, 10:48-50; EX1004, ¶¶172-173. Further, Lin discloses tracking HTTP packets whose source and destination TCP ports are port 80. EX1005, 7:24-8:18. Lin does not specifically disclose that a TCP packet has a TCP sequence number or a TCP

sequence mask field, but this would have been obvious to a POSA because those fields are always present in a TCP packet. EX1004, ¶173.

In addition, Lin explains that the packet applicable criterion used to determine whether to take an action on a packet can include either the source TCP port or destination TCP port. EX1004, ¶¶174-175. Lin's discussion of Table 3 (shown below) teaches that the devices check for a specific packet-applicable criterion – that port 80 is the source or destination port of the TCP packets, which indicates that they are HTTP packets. EX1005, 7:64-8:18. Lin states that, if a packet satisfies this criterion (i.e., the packet indicates that port 80 is the source or destination port), then the SDN switch (which corresponds to the claimed network node) sends the packet over the packet network to the security device (which corresponds to the entity that is included in the instruction and is other than the second entity). *Id.*, 8:10-18.

TABLE 3							
IN_PORT	...	IP src	TCP src port	TCP dst port	...	Action	Count
Ingress_port_ID	*	*	*	80	*	Redirect port	10
Redirect_port_ID	*	*	80	*	*	Ingress port	10
Ingress_port_ID	*	*	*	*	*	Egress port	130
Egress_port_ID	*	*	*	*	*	Ingress port	130

Table 3 of Lin (Annotated)

Id., 8:10-18 (emphasis added), Table 3.

S. Claim 21

[21] The method according to claim 1, 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.

Claim 21 is obvious over Lin in view of the knowledge of a POSA. EX1001, 12:28-32. As discussed above for Claims 18-19, Lin discloses the use of IP and MAC addresses to identify particular packets. EX1005, 5:8-36, 6:1-12, 6:40-54, Table 1. A POSA would have understood that, if IP and MAC addresses were being used in Lin, then the packet network would have included one of the Internet, a Wide Area Network (“WAN”), a Local Area Network (“LAN”), a Metropolitan Area Network (“MAN”), Internet Service Provider (“ISP”) backbone, datacenter network, or inter-datacenter network. EX1004, ¶¶176-178.

T. Claim 22

[22] The method according to claim 1, 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.

Claim 22 is obvious over Swenson. EX1001, 12:33-36. Swenson teaches “user devices 110” on one end of its system and an “origin server 160” on the other end of its system. EX1007, ¶[0023]; EX1004, ¶¶179-180. Swenson discloses that the user devices “are computing devices with network capabilities,” such as “laptops, notebooks, tablets, smart telephones, or personal digital assistants (PDAs).” EX1007, ¶[0025]. A POSA would have understood these to be client

devices. EX1004, ¶¶180-181. Given that Paragraph 58 of Swenson discloses two way communication between the user devices and the origin server, Swenson identifies user devices 110 (i.e., client devices) as the claimed first entity, and origin server 160 (i.e., a server) as the claimed second entity, or vice versa. EX1007, ¶[0058]; EX1004, ¶180.

U. Claim 23

[23] The method according to claim 22, 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.

Claim 23 is obvious over Swenson. EX1001, 12:37-41. As discussed above for Claim 22, Swenson teaches “user devices 110” on one end of its system and an “origin server 160” on the other end of its system. EX1007, ¶[0023]. A POSA would have understood that origin server 160 comprises a web server because origin server 160 is able to process HTTP requests, which are requests for web content. EX1007, ¶[0058]; EX1004, ¶¶182-184. Further, Swenson discloses that the user devices (i.e., client devices) can include “laptops, notebooks, tablets, smart telephones, or personal digital assistants (PDAs).” EX1007, ¶[0025].

V. Claim 24

[24] The method according to claim 22, wherein the communication between the network node and the controller is based on, or uses, a standard protocol.

Claim 24 is obvious in view of Lin. EX1001, 12:42-44. Lin discloses the use of the OpenFlow protocol for the communication between the SDN switch (i.e., the network node) and the SDN controller. EX1005, 1:18-43, 2:12-13, 3:42-46, 3:65-4:6, 6:35-39; EX1004, ¶¶185-187. Figure 1 of Lin (shown below) describes the use of OpenFlow protocol as “prior art” to Lin, which indicates that OpenFlow protocol was known as a standard protocol well before the priority date for the ’111 Patent.

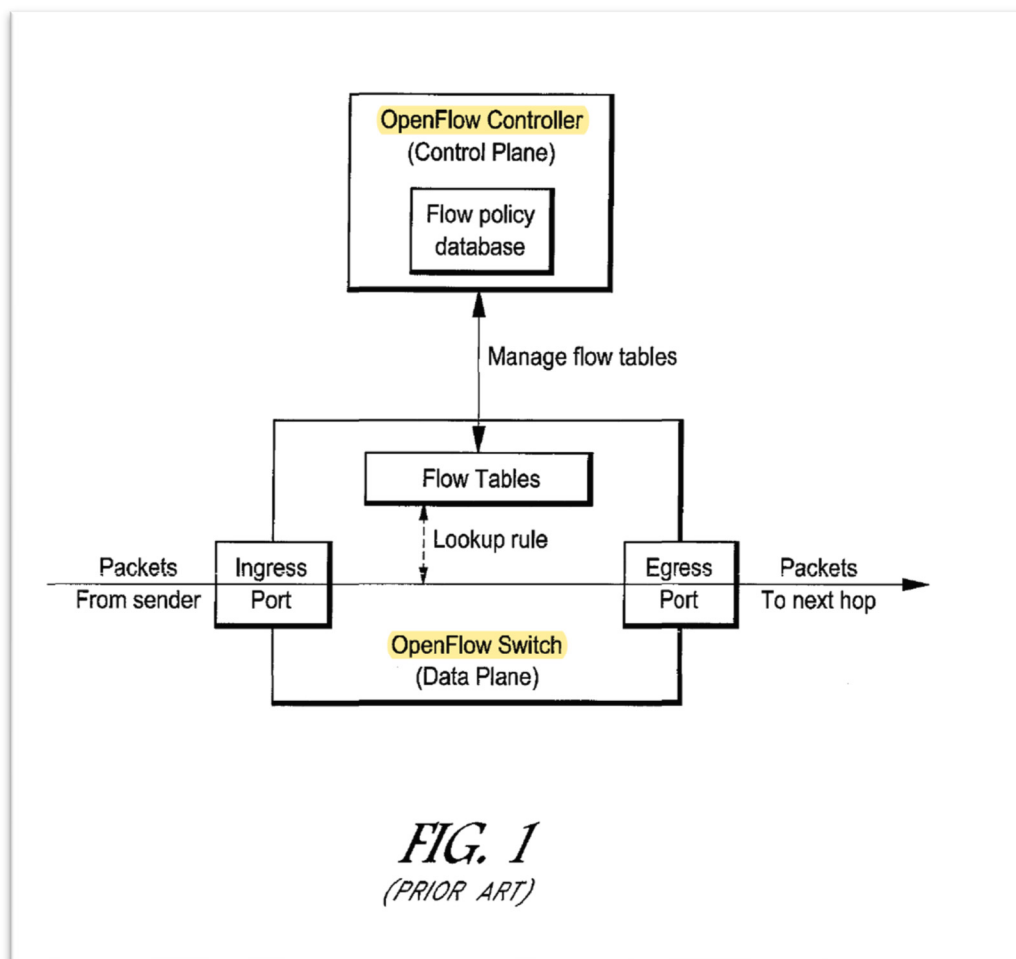


Figure 1 of Lin (Annotated)

EX1005, Figure 1.

W. Claim 27

[27] The method according to claim 1, wherein the network node comprises a router, a switch, or a bridge.

Claim 27 is obvious in view of Lin. EX1001, 12:51-52. As discussed above for Element [1.0], Lin discloses a SDN switch that operates as the claimed network node. EX1005, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63; EX1004, ¶¶188-190.

X. Claim 28

[28] The method according to claim 1, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.

Claim 28 is obvious in view of Lin. EX1001, 12:53-55. As discussed above for Claims 18-19, Lin discusses the use of IP addresses to identify packets being transmitted across the packet network. EX1005, 5:8-36, 6:1-12, 6:40-54, Table 1. A POSA would have understood from the use of the IP addresses that packet network in Lin is an Internet Protocol (IP) network and the packet is an IP packet. EX1004, ¶¶191-193.

Y. Claim 29

[29] The method according to claim 28, wherein the packet network is an Transmission Control Protocol (TCP) network, and the packet is an TCP packet.

Claim 29 is obvious in view of Lin. EX1001, 12:56-58. Lin provides specific examples of TCP packets that are transported over a TCP network when discussing

Tables 2 and 3 of that patent. EX1005, 7:24-8:18; *see id.*, 10:48-50; EX1004, ¶¶194-196.

Z. Claim 30

[30.0] The method according to claim 1, further comprising:

receiving, by the network node from the first entity over the packet network, one or more additional packets; checking, by the network node, if any one of the one or more additional packets satisfies the criterion;

Element [30.0] is obvious in view of Lin. EX1001, 12:59-63. Lin teaches that its method can be applied to “any packet received by the SDN switch.” EX1005, 6:40-63. As such, Lin would check if one or more additional packets after the first packet satisfies the packet-applicable criterion for the reasons discussed above for Elements [1.4]-[1.6]. EX1004, ¶¶197-199.

[30.1] responsive to an additional packet not satisfying the criterion, sending, by the network node over the packet network, the additional packet to the second entity; and responsive to the additional packet satisfying the criterion, sending the additional packet, by the network node over the packet network, in response to the instruction.

Element [30.1] is obvious in view of the combination of Lin and Shieh. EX1001, 12:64-13:3. Lin discloses that the SDN switch forwards additional packets over the packet network in response to whether not the packets satisfy the packet-applicable criterion for the reasons discussed above for Elements [1.5]-[1.6]. EX1004, ¶¶200-203.

AA. Claim 31

[31] The method according to claim 1, wherein the packet network is a Software Defined Network (SDN), the packet is routed as part of a data plane and the network node communication with the controller serves as a control plane.

Claim 31 is obvious in view of Lin. EX1001, 13:4-7. Lin explains that its packet network is a SDN network that uses an SDN controller and an SDN switch. EX1005, Abstract, 1:7-9, 1:58-2:4, 3:53-4:67, Figures 2, 6-8. 2, 6-8. Lin also discloses that it uses an Openflow protocol in which a packet is routed as part of the data plane through its SDN network. *Id.*, 1:11-54, 2:49-50, 3:42-52, Figure 1; EX1004, ¶¶204-205. Further, Lin explains that, when using the Openflow protocol, the communication between the SDN switch and the SDN controller is through a control plane. EX1005, 1:11-54, 2:49-50, 3:42-52, 3:65-4:5, Figure 1; EX1004, ¶¶205-206.

X. GROUND 2: CLAIMS 1, 5-9, 12-24 AND 27-30 ARE UNPATENTABLE AS OBVIOUS OVER SHIEH IN VIEW OF SWENSON.

The combination of Shieh and Swenson, along with the knowledge of a POSA, renders Claims 1, 5-9, 12-24 and 27-30 obvious. EX1004, ¶¶207-318.

A. Claim 1

Element [1.0]

To the extent Element [1.0] is limiting, it is obvious in view of Shieh. EX1001, 10:52-55.

Shieh discloses a method for use with a packet network. EX1004, ¶¶208-209. Shieh’s Abstract states that it relates to “[a] network system” that operates on a “packet from a source node destined to a destination node.” EX1005, Abstract. The specification of Shieh further explains that its system performs operations on packets passing through a network. EX1006, ¶[0002], ¶[0017], ¶[0018], ¶[0021], ¶[0023]. Further, Figure 7 of Shieh, reproduced below, shows the method disclosed in Shieh, which involves a “network access device” performing operations on packets passing through that network access device. *Id.*, ¶[0049], Figure 7.

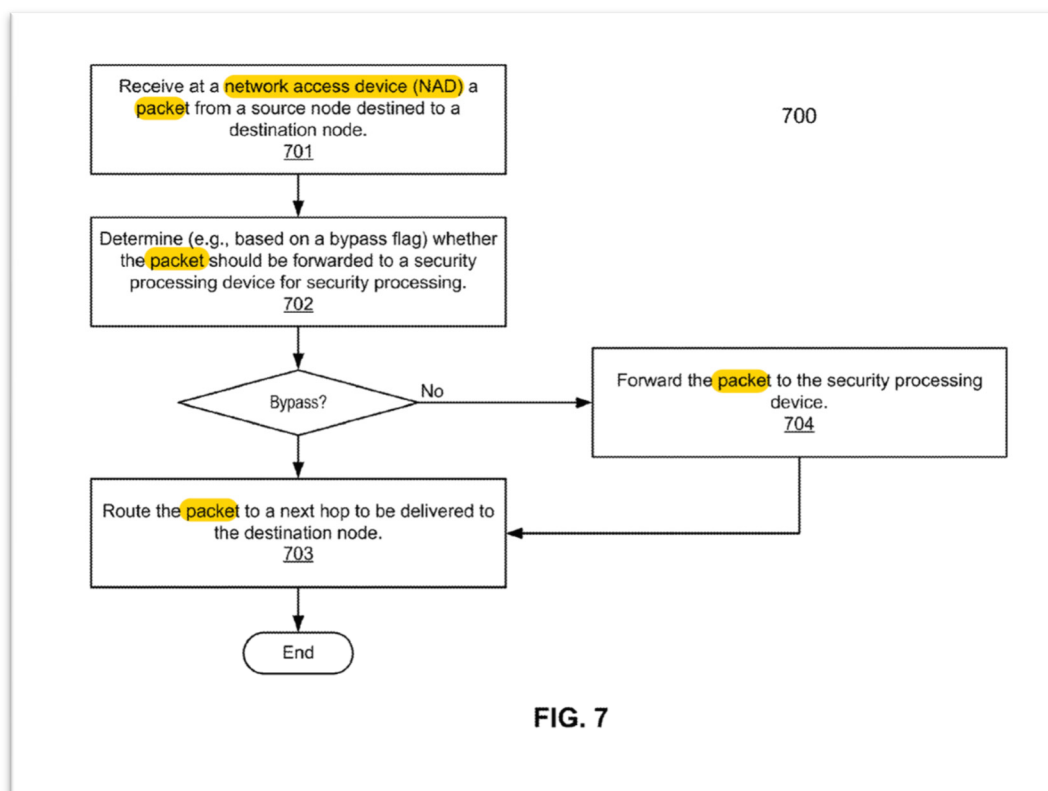


Figure 7 of Shieh (Annotated)

Further, Shieh discusses a network node for transporting packets between first and second entities. EX1004, ¶210. Shieh discloses “network access devices 204A-204C” (**each of which corresponds to the claimed network node**) for transporting a packet from a “source computer” (**which corresponds to the claimed first entity**) to a “destination, which may be another host, a multicast group or a broadcast domain” (**any of which corresponds to the claimed second entity**). EX1006, ¶[0027]; ¶[0037], ¶[0049], Figures 2B, 7. The way in which packets pass from a source node through network access devices 204A-204C to a destination node can be seen below in Figure 2A of Shieh:

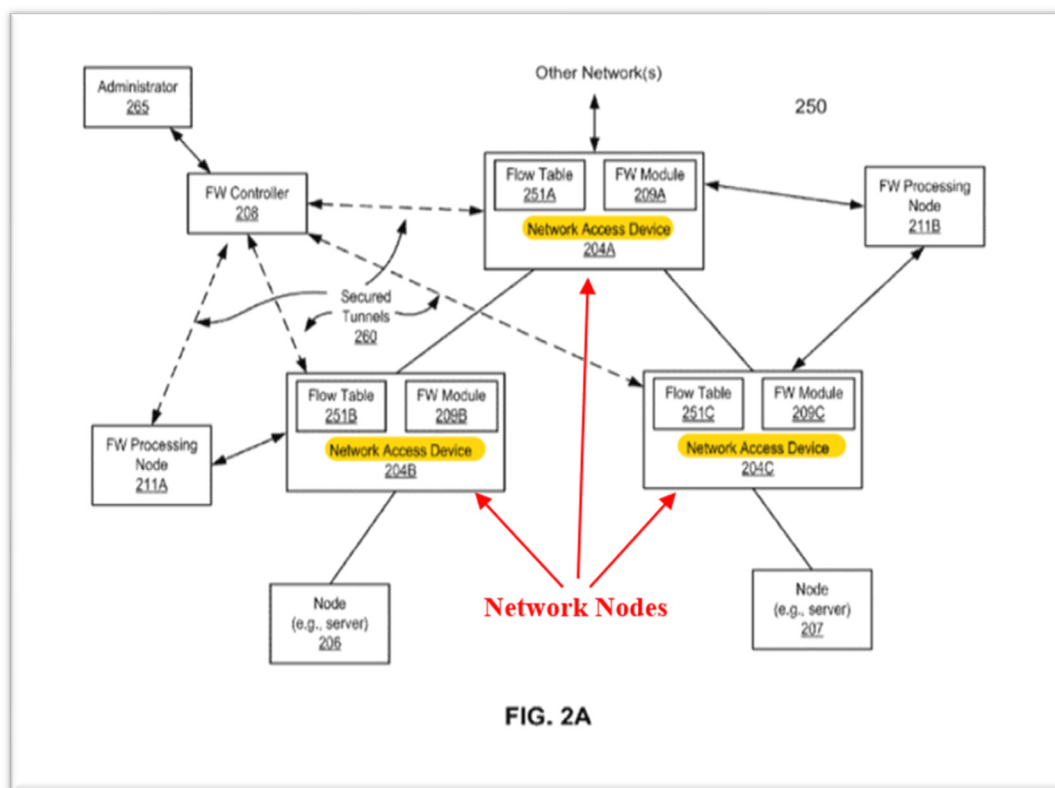


Figure 2A of Shieh (Annotated)

Id., Figure 2A.

Moreover, Shieh explains that the network nodes (e.g., network access devices 204A-204C) are under the control of a controller (**which corresponds to the claimed controller**) that is external to the network node. EX1004, ¶211. Shieh explains that network access devices 204A-204C are “configured” by (i.e., under the control of) “a controller or a management entity.” EX1006, ¶[0018]; *see id.*, ¶[0021] (noting that firewall controller 208 may be “external to network access device 204”), ¶[0025], ¶[0029], ¶[0042]. Figure 2A shows that the controller 208 is external to the network node:

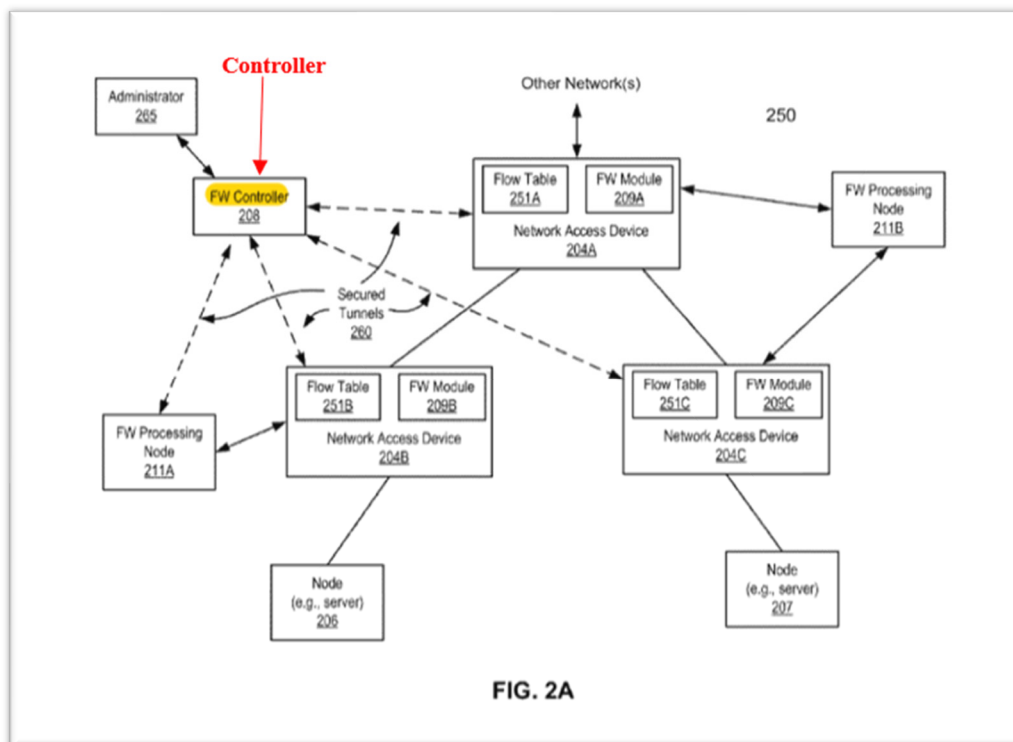


Figure 2A of Shieh (Annotated)

Id., Figure 2A, ¶[0029]; *see id.*, ¶[0025]-¶[0026].

In addition, as addressed in the above claim construction section, the claimed controller is an entity configured to perform DPI on packets.⁴ EX1004, ¶212. The combination of Shieh and Swenson renders it obvious that the controller is configured to perform DPI. EX1004, ¶¶212-214. Shieh discloses that its security processing function includes DPI. EX1006, ¶[0021]. Moreover, as discussed above in Ground 1, Swenson discloses a controller configured to perform DPI. *See* EX1007, ¶[0059] (“the flow analyzer 312 of the network controller 140 performs a deep flow inspection ...”), ¶[0060], (stating that the “controller 140 ingests the network flow for inspection”), ¶[0065] (discussing inspection of HTTP response payload by network controller 140), Figures 1, 4A-4B; EX1004, ¶¶213-214.

Given the similarities between the disclosures in Lin and Shieh, a POSA would have been motivated to combine the teachings of Shieh and Swenson for similar reasons as those disclosed above in Ground 1 for the combination of Lin and Swenson. EX1004, ¶¶82-86, 215-220. For instance, Shieh and Swenson relate to routing traffic through a computer network and use a central controller to provide instructions and packet-applicable criterion to network nodes to determine which

⁴ To the extent that the PTAB does not agree with this construction, Shieh still discloses Element [1.0] for the reasons discussed above.

packets should be redirected and which packets can be sent directly to a destination node. EX1006, ¶¶[0017]-¶[0018], ¶[0024]-¶[0025], ¶[0029]-¶[0030], Figures 1-3, 7; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; EX1004, ¶215.

Further, a POSA would have understood both Shieh and Swenson to disclose redirecting packets for analysis as part of a security function. EX1004, ¶¶216-217. Shieh states that a “packet is transmitted to a security device associated with the network access device to allow the security device to perform content inspection.” EX1006, Abstract; *see id.*, ¶[0042], ¶[0049]. Similarly, a POSA would have understood that the bandwidth monitoring in Swenson can be used as a security application that monitors for DOS attacks that occupy significant bandwidth in a network. EX1007, ¶[0059]-¶[0060]; *see id.*, ¶[0039] (explaining that the controller can incorporate “security functions”; EX1004, ¶216. Further, the architecture of Swenson is functionally similar to the architecture of Shieh, as can be seen in the below two figures showing a central controller directly connected to a network node (the Network Access Devices in Shieh and the steering device in Swenson) through which packet flows pass:

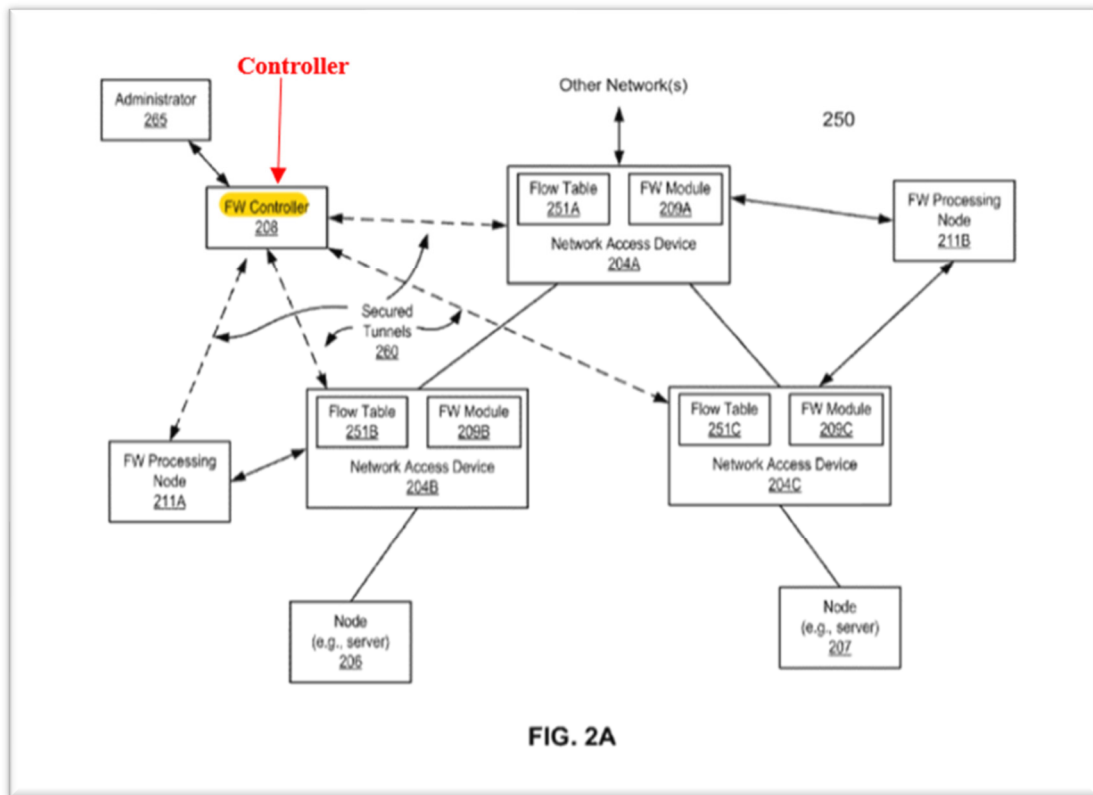
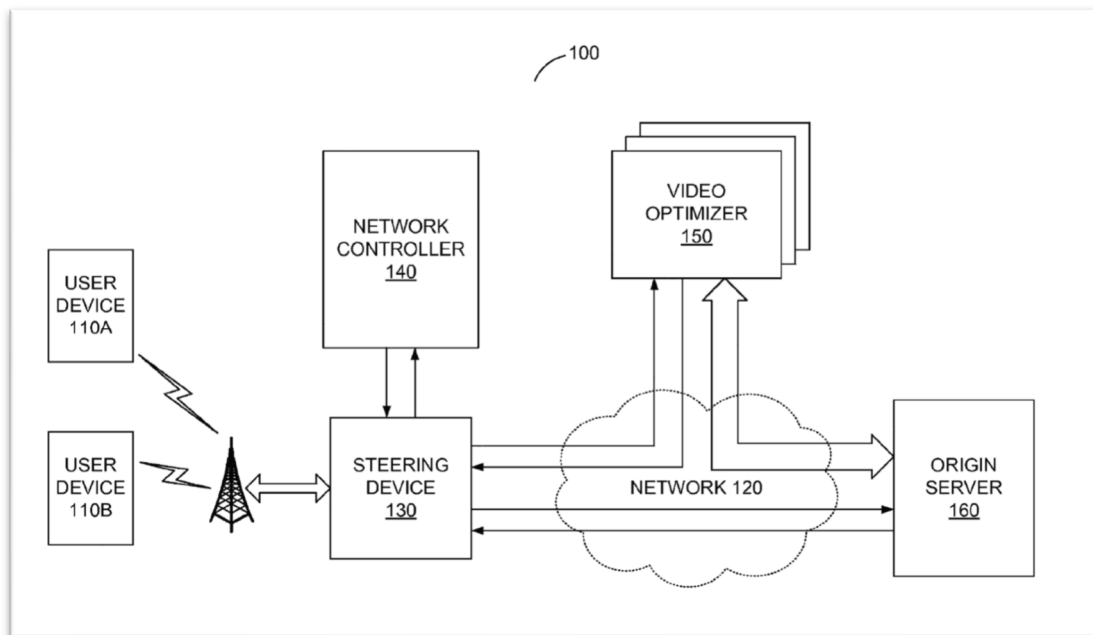


Figure 2A of Shieh (Annotated)



EX1004, ¶216. In addition, Shieh and Swenson each teach that only certain packets may need to be redirected for further processing before being sent to the destination node. EX1006, ¶[0017]-¶[0018], EX1007, ¶[0023]-¶[0032].

Further, a POSA would have found it obvious to implement the security processing module in Shieh as part of the controller in light of the disclosures in Swenson to send packets to the controller for DPI. EX1007, ¶[0059]-¶[0060], Figures 1, 4A-4B; EX1004, ¶218. The controller provides a central location for packets from different nodes to undergo inspection by a security component that applies the same security algorithms to each analyzed packet. EX1004, ¶218. Further, a POSA would have known that security algorithms in a security component are often updated via software updates as new threats are identified. *Id.* A POSA would have known that an efficient way to keep the security component up-to-date was to have a central security component that is part of the central processor. *Id.*

Moreover, a POSA would have had a reasonable expectation of success in modifying Shieh to implement the disclosure of Swenson given that they have a functionally similar architecture. EX1006, ¶[0017]-¶[0018], ¶[0024]-¶[0025], ¶[0029]-¶[0030], Figures 1-3, 7; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; EX1004, ¶219. If Swenson could be constructed to have the network nodes send packets to the central controller for inspection, a POSA

would have understood that the same arrangement could be implemented in Shieh. EX1004, ¶219.

Element [1.1]

Element [1.1] is obvious in view of Shieh. EX1001, 10:56-58.

Shieh discloses that Controller 208 controls network access devices 204A-204C (i.e., the claimed network nodes) in a packet network for the reasons discussed above for Element [1.0]. EX1006, ¶[0018], ¶[0025]-¶[0026], ¶[0029]; EX1004, ¶¶221-222.

Further, Shieh identifies an instruction sent by the controller to the network node over the packet network that determines whether or not a packet requires inspection. EX1004, ¶223. Shieh states that “a persistent connection” exists over the packet network to allow for communication between the controller and the network access devices 204A-204C. EX1006, ¶[0025]; *see id.*, ¶[0018], ¶[0028]-¶[0029]. Further, Shieh discloses that a “command” (**which corresponds to the claimed instruction**) is sent from an administrator via the controller to the network access devices 204A-204C “to set up a set of filtering rules concerning whether and/or what types of packets should be forwarded to a security device.” *Id.*, ¶[0017]-¶[0018]; *see id.*, ¶[0023], ¶[0028]-¶[0029].

Shieh also discloses separate packet-applicable criterion to determine which packets should be forwarded to the security device pursuant to the filtering rules.

EX1004, ¶224. For example, Shieh provides criterion specific to a particular packet being checked, such as the identification of “TCP FIN or TCP RST packets.” *Id.*, ¶[0036]; *see id.*, ¶[0035], ¶[0049]. **These correspond to the claimed packet-applicable criterion.** Shieh does not state whether these criterion are sent from a controller to the network access devices. However, a POSA would have understood that the packet-applicable criterion would be sent from a controller to one or more network access devices. EX1004, ¶¶224-225. This allows the same packet-applicable criterion to be applied at multiple network access devices (i.e., network nodes). *Id.*

Element [1.2]

Element [1.2] recites that the instruction and packet-applicable criterion that are sent by the controller to the network node in Element [1.1] are received by the network node. EX1001, 10:59-60. Shieh therefore discloses Element [1.2] for substantially the same reasons that it discloses Element [1.1]. EX1004, ¶¶226-232.

Element [1.3]

Element [1.3] is obvious in view of Shieh. EX1001, 10:60-62.

Claim 1 of Shieh is a method claim that discloses that a network access device (i.e., a network node) receives a packet from a source node (i.e., a first entity) that is addressed to a destination node (i.e., a second entity). EX1006, Claim 1; EX1004, ¶¶233-235; *see* EX1006, Abstract, ¶[0020], ¶[0027], ¶[0037], ¶[0049]. Shieh

explains that the packets may be addressed with, among other things, “Source and Destination IP address,” which are used to address the source computer (i.e., the claimed first entity) and the destination (i.e., the claimed second entity). *Id.*, ¶[0027]. In addition, Figure 1 of Shieh (reproduced below) shows a system in which “Network Access Device(s) 204” receive packets from client 201 or client 202 that are addressed to server 206 or server 207:

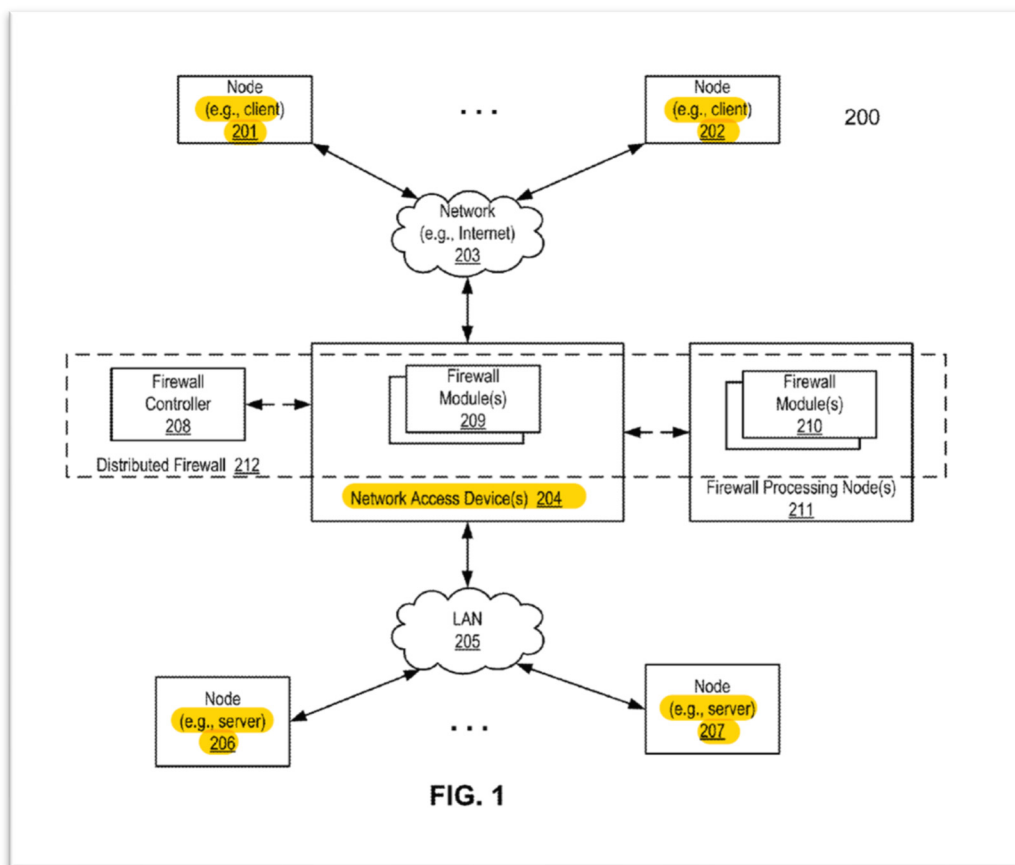


Figure 1 of Shieh (Annotated)

Id., Figure 1; *see id.*, Figures 2A-2B, 7.

Element [1.4]

Element [1.4] is obvious in view of Shieh. EX1001, 10:63-64.

As discussed above for Element [1.1], Shieh discusses the implementation of packet-applicable criterion, such as the identification of “TCP FIN or TCP RST packets,” which are checked by the network access device to determine whether a packet satisfies that criterion. EX1006, ¶[0036]; *see id.*, ¶[0035], ¶[0037], ¶[0049], Figure 7; EX1004, ¶¶236-238.

Element [1.5]

Element [1.5] is obvious in view of Shieh. EX1001, 10:65-67.

As discussed above for Elements [1.1] and [1.4], Shieh discloses criterion specific to a particular packet being checked by the network access device. EX1006, ¶[0018], ¶[0023], ¶[0028]-¶[0029], ¶[0035], ¶[0049], Figure 7. Shieh checks whether a packet is a TCP FIN or TCP RST packet, and, responsive to the packet not satisfying this criterion, sends the packet to the destination node (i.e., the second entity). *Id.*, ¶[0036]; *see id.*, ¶[0037] (discussing whether packet should be sent to security processing device “based on a bypass flag”); EX1004, ¶¶239-242.

Element [1.6]

Element [1.6] is obvious in view of Shieh. EX1001, 11:1-4.

As discussed above for Elements [1.1], [1.4] and [1.5], Shieh teaches that its network access device utilizes packet-applicable criterion to determine whether a

packet should be forwarded to a security device. EX1006, EX1006, ¶[0018], ¶[0023], ¶[0028]-¶[0029], ¶[0035], ¶[0049], Figure 7; EX1004, ¶¶243-244. Shieh checks whether a packet is a TCP FIN or TCP RST packet, and, responsive to the packet satisfying this criterion, sends the packet to the security device (i.e., the entity that is included in the instruction and is other than the second entity). EX1006, ¶[0036]; *see id.*, ¶[0037] (discussing whether packet should be sent to security processing device “based on a bypass flag”); EX1004, ¶¶244-245.

B. Claims 5-9

Claims 5 and 7 are obvious over the combination of Shieh (which discloses the instruction and packet-applicable criterion for the reasons discussed above in Elements [1.1]-[1.2] and [1.4]-[1.6]) and Swenson (which discloses sending the packet or a portion thereof to the controller for the reasons discussed in Ground 1). EX1004, ¶¶246-249, 253-255.

Claims 6 and 8-9 are obvious over Swenson, in view of the knowledge of a POSA, for the reasons discussed above in Ground 1. EX1004, ¶¶250-252, 256-262.

C. Claim 12

Claim 12 is obvious over Shieh. EX1001, 11:55-57. Shieh repeatedly states that the analysis performed on the packets includes a security processing function. EX1006, ¶[0002], ¶[0017]-¶[0019], ¶[0021], ¶[0023], ¶[0029]-¶[0031], ¶[0035]-

¶[0037], ¶[0042], ¶[0049], Figures 1, 2B, 3, 7. A POSA would have understood from this that the analysis includes the security processing function. EX1004, ¶¶263-266.

D. Claim 13

Claim 13 is obvious over Shieh. EX1001, 11:58-60. Shieh discloses that its system analyzes packets by applying a security application that comprises a firewall functionality. EX1006, ¶[0021], ¶[0023], ¶[0027]-¶[0028], ¶[0038], ¶[0043]-¶[0049]; EX1004, ¶¶267-269.

E. Claim 14

Claim 14 is obvious over both Shieh and Swenson. EX1001, 11:61-63. Shieh discloses the use of “deep packet inspection (DPI)” as part of its analysis functionality. EX1006, ¶[0021]; *see id.*, ¶[0028], ¶[0040]-¶[0041]; EX1004, ¶¶270-273. Swenson also discloses that the flow analyzer in network controller 140 performs DPI. EX1007, ¶[0059]-¶[0060], ¶[0065].

F. Claim 15

Claim 15 is obvious in view of both Shieh and Swenson. EX1001, 11:64-67. As discussed above for Claim 14, Shieh discloses analysis of packets through DPI. EX1006, ¶[0021], ¶[0028], ¶[0040]-¶[0041]; EX1005, 3:11-12. A POSA would have understood that DPI refers to the inspection of at least part of the payload field. EX1004, ¶¶274-278. Further, Swenson discloses Claim 15 for the reasons discussed above in Ground 1.

G. Claim 16

Claim 16 is obvious over Lin in view of the knowledge of a POSA. EX1001, 12:1-4. A POSA would have known that a packet includes distinct header and payload fields. EX1008, Section 2; EX1004, ¶¶279-280. Further, Shieh states that the packet streams passing through its system include “TCP FIN or TCP RST packets.” EX1006, ¶[0036]. A POSA would have understood that a TCP FIN packet includes a FIN flag bit in its header that is set. EX1004, ¶281. Further, a POSA would have understood that a TCP RST packet includes a RST flag bit in its header that is set. *Id.* A POSA would have found it obvious that the TCP FIN flag bit or the TCP RST flag bit being set could function as the packet-applicable criterion recited in Claim 1. *Id.*

H. Claim 17

Claim 17 is obvious over Shieh in view of the knowledge of a POSA. EX1001, 12:5-9. Shieh states that the packet streams passing through its system include TCP packets. EX1006, ¶[0027], ¶[0036]. Further, the TCP FIN or TCP RST packets discussed in Shieh render Claim 17 obvious for the reasons discussed above with respect to Claim 16. *Id.*, ¶[0036]; EX1004, ¶¶282-284. In addition, a POSA also would have understood that SYN and ACK were common flags in a TCP packet header that could have been used to identify packets in a packet stream. *Id.*

I. Claim 18

Claim 18 is obvious over Shieh in view of the knowledge of a POSA. EX1001, 12:10-16. Shieh discusses the header fields in a packet and how they are used to route packets. EX1005, [0044]. Similarly, Shieh discloses that IP or MAC addresses are packet-applicable criterion used to determine whether to forward packets to a security processing function based on whether the IP or MAC addresses match a predetermined address or addresses. EX1006, ¶[0031]; ¶[0027]; EX1004, ¶¶285-288.

J. Claim 19

Claim 19 is obvious over Shieh for the reasons as those discussed above for Claim 18. EX1001, 12:17-18; EX1006, ¶[0027]; ¶[0031]; EX1004, ¶289.

K. Claim 20

Claim 20 is obvious over Shieh in view of the knowledge of a POSA. EX1001, 12:20-27. Shieh discloses the transportation of TCP packets across a TCP network using TCP protocol. EX1006, ¶[0027], ¶[0036]; EX1004, ¶¶290-292. Further, Shieh describes tracking TCP packets and states that “a TCP/IP flow [i.e., a TCP packet] can be uniquely identified by ... source and destination port...” EX1006, ¶[0027]. Shieh does not state that a TCP packet has a TCP sequence number or a TCP sequence mask field, but a POSA would have known those fields are always present in a TCP segment/packet. EX1004, ¶291. Further, Shieh teaches the use of a source

TCP port or a destination TCP port as the packet applicable criterion. EX1006, ¶¶0027], ¶¶0036]; EX1004, ¶291.

L. Claim 21

Claim 21 is obvious over Shieh. EX1001, 12:28-32. Shieh states that its packet network includes the Internet, a WAN or a LAN. EX1006, ¶¶0020]. Shieh also discloses that its system can be in communication with an ISP. *Id.*, ¶¶0053]; EX1004, ¶¶293-295.

M. Claim 22

Claim 22 is obvious over Shieh. EX1001, 12:33-36. Figure 1 of Shieh (reproduced below) shows that the source of the network packets (i.e., the claimed first entity) can be a server device (Nodes 206 or 207), and the destination of the claimed packets (i.e., the claimed second entity) can be a client device (Nodes 201 or 202), or vice versa.

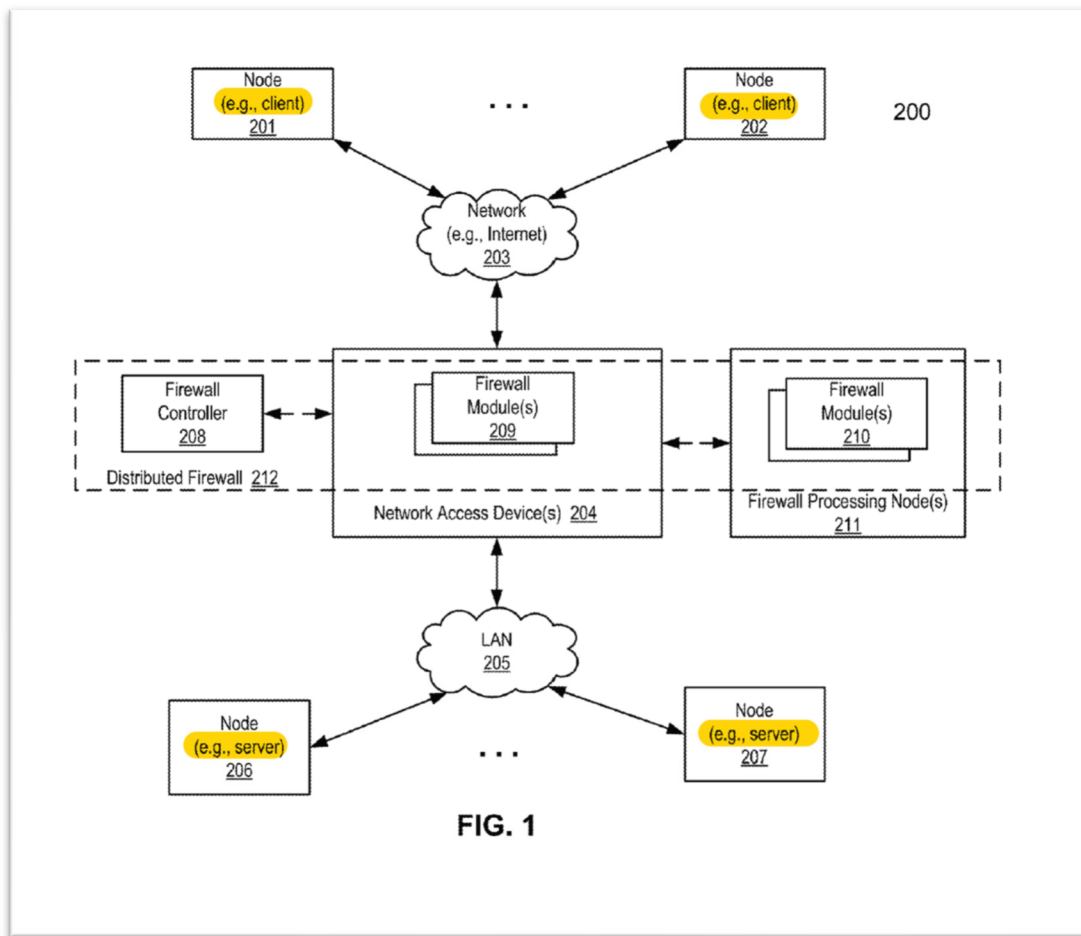


Figure 1 of Shieh (Annotated)

EX1006, Figure 1; *see id.*, ¶[0020], ¶[0022], ¶[0039], ¶[0053], ¶[0057], Figures 2A, 3; EX1004, ¶¶296-298.

N. Claim 23

Claim 23 is obvious over Shieh. EX1001, 12:37-41. As discussed above for Claim 22, Shieh discloses that the source or destination of the packets can be a server device, such as Nodes 206 or 207 in Figure 1. EX1006, Figure 1; *see id.*, ¶[0020], ¶[0022], ¶[0039], ¶[0053], ¶[0057], Figures 2A, 3. A POSA would have known that the server can be a web server. EX1004, ¶¶299-300; EX1006, ¶[0053]. Further,

Shieh discloses that “[a]ny of nodes 201-202 and 206-207 may be a client device (e.g., a desktop, laptop, Smartphone, gaming device) or a server.” EX1006, ¶[0020]; EX1004, ¶¶301-302.

O. Claim 24

Claim 24 is obvious in view of Shieh. EX1001, 12:42-44. Shieh discloses the use of the standard OpenFlow protocol for communication between the network access devices (i.e., the network node) and the controller. EX1006, ¶[0025]-¶[0026]; EX1004, ¶303.

P. Claim 27

Claim 27 is obvious in view of Shieh. EX1001, 12:51-52. Shieh states that its network access devices (which correspond to the claimed network nodes) “may be a router or a gateway, a switch or an access point...” EX1006, ¶[0020]; *see id.*, ¶[0019], ¶[0025]-¶[0026], ¶[0032]; EX1004, ¶¶304-306.

Q. Claims 28

Claim 28 is obvious in view of Shieh. EX1001, 12:53-55. Shieh discloses the communication of IP packets through IP networks. EX1006, ¶[0038]; *see id.*, ¶[0027] (discussing the use of IP addresses to identify IP packets being transmitted across an IP packet network), ¶[0031] (same); EX1004, ¶¶307-309.

R. Claims 29

Claim 29 is obvious in view of Shieh. EX1001, 12:56-58. Shieh discloses the transportation of TCP packets across a TCP network using TCP protocol. EX1006, ¶¶0027, ¶¶0036; EX1004, ¶¶310-311.

S. Claim 30

Claim 30 is obvious in view of Shieh. EX1001, 12:59-63. Shieh states that its method can be applied to “subsequent packets of a particular session.” EX1006, ¶¶0018; *see id.*, ¶¶0037; EX1004, ¶¶312-314. Further, Shieh explains that its network access devices forward additional packets over the packet network in response to whether not the packets satisfy the packet-applicable criterion for the reasons discussed above for Elements [1.5]-[1.6]. EX1004, ¶¶315-318.

XI. OBJECTIVE INDICIA OF NONOBVIOUSNESS

At this stage of these proceedings, Petitioner has no burden to identify and rebut objective indicia of nonobviousness. EX1004, ¶319-320. Patent Owner must first present a *prima facie* case for such consideration, which Petitioner should then have the chance to rebut on reply. *Sega of Am., Inc. v. Uniloc USA, Inc.*, IPR2014-01453, Paper 11, at *20 (PTAB Mar. 10, 2015).

XII. DISCRETIONARY DENIAL UNDER § 325(D) OR § 314 IS NOT WARRANTED

Discretionary denial under 35 U.S.C. §325(d) is not warranted here because the challenges presented in this petition are neither cumulative nor redundant to the prosecution of the '111 Patent. The PTO has not previously considered Lin, Shieh or Swenson in connection with the claims of the '111 Patent. Further, the combination is not redundant to any combination of references considered during prosecution. Additionally, Dr. Bhattacharjee is an expert in the relevant art, and his analysis of the prior art has not been presented to the PTO. EX1004, ¶¶1-11. In view of the new information presented in this Petition, the Board has ample discretion to institute an IPR on the grounds presented.

Likewise, denial under 35 U.S.C. §314(a) is not warranted, as the application of the factors in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha* demonstrates that the filing of the instant Petition is not abusive, that none of the Patent Owner, the PTO, or the Board has addressed the merits of the grounds in this Petition, and instituting the present proceeding is an efficient use of the Board's resources. *See* IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (precedential). Petitioner has not challenged the '111 Patent in a prior petition.

Moreover, the Petition should not be denied under the discretionary factors set out in *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11, at 6 (PTAB March 20, 2020). Petitioner addresses each of those factors below:

1. Whether the court granted a stay – A motion to stay has not yet been filed in the parallel litigation, so the Board should not infer the outcome if such a motion is later filed. *Sand Revolution II LLC v. Continental Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB June 16, 2020) (informative); *Dish Network LLC v. Broadband iTV, Inc.*, IPR2020-01359, Paper 15 at 11 (PTAB Feb. 12, 2021). Thus, this factor is neutral on discretionary denial.

2. Parallel proceeding trial date – This factor weighs against discretionary denial because the projected trial date in the parallel litigation is “around the same time” as the Board’s expected Final Written Decision. The parallel litigation was filed in the U.S. District Court for the Eastern District of Texas on July 22, 2022. EX1010. While trial is currently proposed for March 4, 2024 (EX1011 at 1), the Board recognizes “that scheduled trial dates are unreliable and often change.” EX1012 at 8. For this reason, the Board now uses median time-to-trial statistics in the litigation venue to determine a projected trial date. EX1012 at 8-9. The median time-to-trial in the Eastern District of Texas was 24.5 months as of June 30, 2022. EX1013. Thus, the projected trial date in the litigation for *Fintiv* purposes is August 2024, approximately 24.5 months after July 2022. The Board’s Final Written

Decision is also expected in August 2024, which is 18 months after Petitioner expects a notice of accorded filing for this Petition. Accordingly, this factor weighs against discretionary denial.

3. Investment in the parallel proceeding by the court and the parties – The parallel litigation is in its early stages, and the Court has not issued any substantive ruling relating to the '111 Patent. The Parties have not exchanged preliminary positions on claim construction, and the claim construction hearing is not scheduled until September 7, 2023, a month after the expected institution decision by the Board. EX1011 at 3-4. Further, the Parties only exchanged preliminary invalidity contentions on February 2, 2023, 19 days before this Petition was filed. *Id.* at 5. Moreover, the Parties have not exchanged their first set of fact discovery requests, and expert discovery does not begin until October 19, 2023. *Id.* at 3. The early stage of and minimal investment in the parallel litigation weighs against discretionary denial. *See PEAG LLC v. Varta Microbattery GMBH*, IPR2020-01214, Paper 8 at 17 (Jan. 6, 2021).

4. Overlapping issues with the parallel litigation proceeding – Preliminary invalidity contentions were only served in the parallel litigation on February 2, 2023, and thus it is too early to determine overlapping invalidity issues. EX1011 at 5. Nonetheless, instituting a proceeding will allow the Board to address the art, and a Final Written Decision would narrow the issues in the parallel litigation due to the

estoppel provisions of 35 U.S.C. § 315(e)(2). Moreover, there will be no overlap of prior art issues because, if the Board institutes trial, Petitioner will cease asserting in the parallel litigation the combination of references on which trial is instituted for the claims on which trial is instituted, to the extent Petitioner even asserts the same combination in the parallel litigation. This factor weighs against discretionary denial. *See Verizon v. Huawei*, IPR2020-01079, Paper 10 at 38 (Jan. 14, 2021) (finding a similar stipulation “mitigates the concern about overlapping issues” and weighs “against discretionary denial of the Petition.”).

5. Identity of the Parties – Petitioner is a defendant in the parallel litigation, but that is true of most petitioners in IPR proceedings. Accordingly, this factor is neutral on discretionary denial.

6. Other circumstances, including the merits – As discussed in detail above, the analysis in Grounds 1-2 provides a compelling unpatentability challenge to Claims 1-9, 12-24 and 27-31. The merits of Petitioners’ arguments are strong, and this factor weighs against discretionary denial. *See* EX1012 at 3-5; *Sand Revolution II*, IPR2019-01393, Paper 24 at 13.

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

A. Real Party-in-Interest

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

B. Related Matters

The '111 Patent is asserted in *Orckit Corp. v. Cisco Systems, Inc.*, Case No. 2:22-cv-00276 (E.D. Tex.).

C. Lead and Back-up Counsel and Service Information

<u>Lead Counsel</u>	
Jeffrey D. Blake MERCHANT & GOULD P.C. 191 Peachtree Street NE Suite 3800 Atlanta, GA 30303	Phone: 404-954-5040 Fax: 612-332-9081 jblake@merchantgould.com USPTO Reg. No. 58,884
<u>Back-up Counsel</u>	
Daniel W. McDonald MERCHANT & GOULD P.C. 150 South Fifth Street Suite 200 Minneapolis, MN 55402	Phone: 612-336-4637 Fax: 612-332-9081 dmcdonald@merchantgould.com USPTO Reg. No. 32,044

Please address all correspondence to lead and back-up counsel at OrckitIPR@merchantgould.com. Petitioner consents to electronic service.

XIV. CONCLUSION

For the reasons above, Petitioners ask that the Board order an *inter partes* review trial for Claims 1-9, 12-24 and 27-31, and that the Director cancel these claims as unpatentable.

Respectfully submitted,

Date: February 21, 2023

/Jeffrey D. Blake/
Jeffrey D. Blake
Registration No. 58,884

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned attorney for the Petitioner, Cisco Systems, Inc., declares that the argument section of this Petition has 13,982 words, according to the word count tool in Microsoft Word™.

/Jeffrey D. Blake/
Jeffrey D. Blake
Registration No. 58,884

APPENDIX A – CLAIM LISTING

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

Claim or Element #	Claim Language
Claim 1	
[1.0]	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 comprising:
[1.1]	sending, by the controller to the network node over the packet network, an instruction and a packet-applicable criterion;
[1.2]	receiving, by the network node from the controller, the instruction and the criterion;
[1.3]	receiving, by the network node from the first entity over the packet network, a packet addressed to the second entity;
[1.4]	checking, by the network node, if the packet satisfies the criterion;
[1.5]	responsive to the packet not satisfying the criterion, sending, by the network node over the packet network, the packet to the second entity; and
[1.6]	responsive to the packet satisfying the criterion, sending the packet, by the network node over the packet network, to an entity that is included in the instruction and is other than the second entity.
Claim 2	
	The method according to claim 1, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction, and upon receiving by the network node the 'terminate' instruction, the method further comprising blocking, by the network node, the packet from being sent to the second entity and to the controller.
Claim 3	
	The method according to claim 1, wherein the instruction is a 'probe', a 'mirror', or a 'terminate' instruction, and upon receiving by the network node the 'mirror' instruction and responsive to the packet satisfying the criterion, the method

Claim or Element #	Claim Language
	further comprising sending the packet, by the network node, to the second entity and to the controller.
Claim 4	
	The method according to claim 1, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction, and upon receiving by the network node the 'probe' instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet, by the network node, to the controller; responsive to receiving the packet, analyzing the packet, by the controller; sending the packet, by the controller, to the network node; and responsive to receiving the packet, sending the packet, by the network node, to the second entity.
Claim 5	
	The method according to claim 1, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof, by the network node, to the controller.
Claim 6	
	The method according to claim 5, further comprising storing the received packet or a portion thereof, by the controller, in a memory.
Claim 7	
	The method according to claim 5, further comprising responsive to the packet satisfying the criterion and to the instruction, sending a portion of the packet, by the network node, to the controller.
Claim 8	
	The method according to claim 7, wherein the portion of the packet consists of multiple consecutive bytes, and wherein the instruction comprises identification of the consecutive bytes in the packet.
Claim 9	
	The method according to claim 5, further comprising responsive to receiving the packet, analyzing the packet, by the controller.

Claim or Element #	Claim Language
Claim 12	
	The method according to claim 9, wherein the analyzing comprises applying security or data analytic application.
Claim 13	
	The method according to claim 9, wherein the analyzing comprises applying security application that comprises firewall or intrusion detection functionality.
Claim 14	
	The method according to claim 9, wherein the analyzing comprises performing Deep Packet Inspection (DPI) or using a DPI engine on the packet.
Claim 15	
	The method according to claim 9, wherein the packet comprises distinct header and payload fields, and wherein the analyzing comprises checking part of, or whole of, the payload field.
Claim 16	
	The method according to claim 1, 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.
Claim 17	
	The method according to claim 16, 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.
Claim 18	
	The method according to claim 1, 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.

Claim or Element #	Claim Language
Claim 19	
	The method according to claim 18, wherein the addresses are Internet Protocol (IP) addresses.
Claim 20	
	The method according to claim 1, 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.
Claim 21	
	The method according to claim 1, 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.
Claim 22	
	The method according to claim 1, 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.
Claim 23	
	The method according to claim 22, 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.
Claim 24	
	The method according to claim 22, wherein the communication between the network node and the controller is based on, or uses, a standard protocol.
Claim 27	
	The method according to claim 1, wherein the network node comprises a router, a switch, or a bridge.

Claim or Element #	Claim Language
Claim 28	
	The method according to claim 1, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.
Claim 29	
	The method according to claim 28, wherein the packet network is an Transmission Control Protocol (TCP) network, and the packet is an TCP packet.
Claim 30	
[30.0]	The method according to claim 1, further comprising: receiving, by the network node from the first entity over the packet network, one or more additional packets; checking, by the network node, if any one of the one or more additional packets satisfies the criterion;
[30.1]	responsive to an additional packet not satisfying the criterion, sending, by the network node over the packet network, the additional packet to the second entity; and responsive to the additional packet satisfying the criterion, sending the additional packet, by the network node over the packet network, in response to the instruction.
Claim 31	
	The method according to claim 1, wherein the packet network is a Software Defined Network (SDN), the packet is routed as part of a data plane and the network node communication with the controller serves as a control plane.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Cisco Systems, Inc.	§	
	§	Petition for <i>Inter Partes</i> Review
Petitioner	§	U.S. Patent No. 10,652,111
	§	
	§	

CERTIFICATE OF SERVICE

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

<i>Date of service</i>	February 21, 2023
<i>Manner of service</i>	PRIORITY MAIL EXPRESS
<i>Documents served</i>	Petition for <i>Inter Partes</i> Review, including Exhibit List; Exhibits 1001 through 1013; Power of Attorney
<i>Persons served</i>	May Patents Ltd. c/o Dorit Shem-Tov P.O.B. 7230 Ramat-Gan, 5217102 Israel

/Jeffrey D. Blake/
Jeffrey D. Blake
Registration No. 58,884