

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.

Petitioner,

v.

ORCKIT CORPORATION

Patent Owner.

Case IPR2024-00037

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

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

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	STANDING AND IDENTIFICATION OF CHALLENGE UNDER §§42.104(A)-(B)	1
A.	Grounds for Standing (§42.104(a))	1
B.	Identification of Challenge and Relief Requested (§42.104(b))	1
III.	SUMMARY OF THE '111 PATENT	3
A.	Overview of the '111 Patent.....	3
B.	Priority Date	5
C.	Level of Ordinary Skill in the Art	5
D.	Claim Construction.....	5
IV.	NO BASIS EXISTS FOR DISCRETIONARY DENIAL	8
A.	Discretionary Denial Under <i>Fintiv</i> Isn't Appropriate	8
B.	Discretionary Denial Under <i>General Plastic</i> Isn't Appropriate	9
V.	GROUND 1: CLAIMS 1-9, 12-24, AND 27-31 ARE UNPATENTABLE OVER LIN IN VIEW OF SWENSON.	10
A.	Lin.....	10
B.	Swenson.....	12
C.	Claim 1	14
D.	Claim 2	33
E.	Claim 3	35
F.	Claim 4	37

G.	Claim 5	38
H.	Claim 6	40
I.	Claim 7	41
J.	Claim 8	42
K.	Claim 9	43
L.	Claim 12	44
M.	Claim 13	45
N.	Claim 14	45
O.	Claim 15	45
P.	Claim 16	46
Q.	Claim 17	47
R.	Claim 18	48
S.	Claim 19	48
T.	Claim 20	49
U.	Claim 21	50
V.	Claim 22	51
W.	Claim 23	52
X.	Claim 24	52
Y.	Claim 27	53
Z.	Claim 28	53
AA.	Claim 29	54

BB.	Claim 30	54
CC.	Claim 31	55
VI.	GROUND 2: CLAIMS 1, 5-9, 12-24, AND 27-30 ARE UNPATENTABLE OVER SHIEH IN VIEW OF SWENSON	55
A.	Shieh.....	55
B.	Claim 1	57
C.	Claims 5-9	69
D.	Claim 12	69
E.	Claim 13	70
F.	Claim 14	70
G.	Claim 15	71
H.	Claim 16	71
I.	Claim 17	72
J.	Claim 18	72
K.	Claim 19	73
L.	Claim 20	73
M.	Claim 21	74
N.	Claim 22	74
O.	Claim 23	75
P.	Claim 24	76
Q.	Claim 27	76
R.	Claim 28	77

S.	Claim 29	77
T.	Claim 30	77
VII.	OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....	78
VIII.	CONCLUSION.....	78

EXHIBIT LIST

Exhibit	Description
1001	U.S. Patent No. 10,652,111
1002	Prosecution History of the '111 Patent
1003	Curriculum Vitae of Samrat Bhattacharjee
1004	Declaration of Dr. Samrat Bhattacharjee
1005	U.S. Patent No. 9,264,400 (“Lin”)
1006	U.S. Patent Publication No. 2013/0291088 (“Shieh”)
1007	U.S. Patent 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 Orckit Corp. v. Cisco Systems, Inc., No. 2:22-cv-00276 (E.D. Tex.)
1011	Amended Scheduling Order in Orckit Corp. v. Cisco Systems, Inc., 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

CLAIM LISTING

Claims 1-9, 12-24, and 27-31	
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.
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.
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.
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.

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.
6	The method according to claim 5, further comprising storing the received packet or a portion thereof, by the controller, in a memory.
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.
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.
9	The method according to claim 5, further comprising responsive to receiving the packet, analyzing the packet, by the controller.
12	The method according to claim 9, wherein the analyzing comprises applying security or data analytic application.
13	The method according to claim 9, wherein the analyzing comprises applying security application that comprises firewall or intrusion detection functionality.
14	The method according to claim 9, wherein the analyzing comprises performing Deep Packet Inspection (DPI) or using a DPI engine on the packet.
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.
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.
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.
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.
19	The method according to claim 18, wherein the addresses are Internet Protocol (IP) addresses.

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 packetapplicable 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.
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.
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.
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.
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.
27	The method according to claim 1, wherein the network node comprises a router, a switch, or a bridge.
28	The method according to claim 1, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.
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.
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.

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.
----	---

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

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

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

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

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

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

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

Lead Counsel	Back-up Counsel
Kyle K. Tsui (Reg. No. 62,602) kyle.tsui@fischllp.com FISCH SIGLER LLP 5301 Wisconsin Avenue NW Suite 400 Washington, DC 20015 Phone: 202.362.3527	Andrew L. Ramos (Reg. No. 76,865) andrew.ramos@fischllp.com FISCH SIGLER LLP 5301 Wisconsin Avenue NW Suite 400 Washington, DC 20015 Phone: 202.362.3522

I. INTRODUCTION

U.S. Patent No. 10,652,111 (“the ’111 Patent”) relates to deep packet inspection (“DPI”) in a software defined network (“SDN”). 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.¹ But as shown below, and as Dr. Bhattacharjee’s Declaration (Ex. 1004) confirms, this wasn’t new and would’ve been obvious to a person of ordinary skill in the art.

Included with this Petition is a timely Motion for Joinder. Juniper Networks, Inc. (“Juniper,” “Petitioner”) respectfully requests *inter partes* review (“IPR”) of claims 1-9, 12-24, and 27-31 of the ’111 Patent.

II. STANDING AND IDENTIFICATION OF CHALLENGE UNDER §§42.104(A)-(B)

A. Grounds for Standing (§42.104(a))

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

B. Identification of Challenge and Relief Requested (§42.104(b))

A Motion for Joinder accompanies this Petition, which, in the interest of judicial economy, relies on copies of the exhibits filed in IPR2023-00554. Relying

¹ Ex. 1002 at 322-330, 397-417, 492-501.

on this identical evidence challenging the same claims with substantially identical grounds, Juniper asks the Board to find claims 1-9, 12-24, and 27-31 (“Challenged Claims”) unpatentable under §103² based on the following patents and printed publications in view of the general knowledge of a person having ordinary skill in the art (“POSITA”) prior to the purported invention:

Ground	Proposed Challenges
1	Lin ³ and Swenson ⁴ render obvious claims 1-9, 12-24, and 27-31.
2	Shieh ⁵ and Swenson render obvious claims 1, 5-9, 12-24, and 27-30.

Lin was filed December 2, 2013 and issued February 16, 2016. Lin qualifies as prior art under at least 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 and published on December 5, 2013. Thus, Swenson qualifies as prior art under at least 35 U.S.C. §§102(a)(1)-(2). And Shieh was filed as a provisional application on April 11, 2012 and as a nonprovisional application on April 10, 2013. Shieh

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

³ U.S. Patent No. 9,264,400 (Ex. 1005) (“Lin”).

⁴ U.S. Patent Publication No. 2013/0322242 (Ex. 1007) (“Swenson”).

⁵ U.S. Patent Publication No. 2013/0291088 (Ex. 1006) (“Shieh”).

published on October 31, 2013. Thus, Shieh qualifies as prior art under at least 35 U.S.C. §§102(a)(1)-(2).

III. SUMMARY OF THE '111 PATENT

A. Overview of the '111 Patent

The '111 Patent discloses methods and systems relating to “deep packet inspection (DPI) in a software defined network (SDN).”⁶ 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.⁷ 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.⁸

“[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.”⁹ “To this end, the central controller 111 is further configured to instruct each of the network nodes 112 which of the packets

⁶ '111 Patent, Abstract; *see id.*, 1:14-16, Bhattacharjee, ¶¶30-34.

⁷ '111 Patent, 2:27-30, 2:3-32; Bhattacharjee, ¶30.

⁸ '111 Patent, 2:3-44; Bhattacharjee, ¶30.

⁹ '111 Patent, 4:5-7; Bhattacharjee, ¶31.

and/or sessions should be directed to the controller 111 for packet inspections.”¹⁰

“The determination [of whether a packet requires inspection] is performed based on a set of instructions provided by the controller 111.”¹¹ “A packet that requires inspection is either redirected to the controller 111 or mirrored and a copy thereof is sent to the controller 111.”¹²

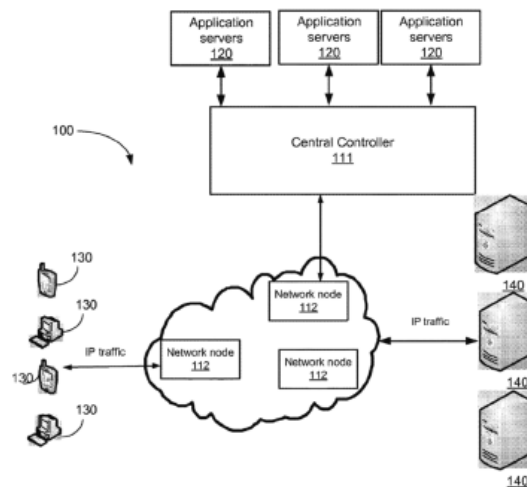


FIG. 1

'111 Patent, Fig. 1.

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

¹⁰ '111 Patent, 4:8-11.

¹¹ *Id.*, 4:14-15.

¹² *Id.*, 4:15-18.

the prior art, along with arguments that there was no motivation to combine the cited art.¹³

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.¹⁴

C. Level of Ordinary Skill in the Art

As of April 22, 2014, a POSITA would've had a bachelor's degree in computer science, computer engineering, or an equivalent, and two years of professional experience, and a POSITA would've had a working knowledge of hardware and software for packet-switched networking.¹⁵ Lack of work experience can be remedied by additional education and vice versa.¹⁶

D. Claim Construction

In *inter partes* review, claim terms must be given their ordinary and customary meaning as understood by a POSITA at the time of the invention in

¹³ EX1002 at 322-330, 397-417, 492-501; Bhattacharjee, ¶¶35-46.

¹⁴ Bhattacharjee, ¶47.

¹⁵ *Id.*, ¶¶48-49.

¹⁶ *Id.*, ¶48.

light of the specification and the prosecution history pertaining to the patent.¹⁷

The claim term “controller” should be construed to mean “an entity configured to perform deep packet inspection on packets.”¹⁸ 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.”¹⁹ 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.”²⁰

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 3.²¹ “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.”²² A POSITA would’ve known from this description that the central controller was configured in

¹⁷ See 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2015) (en banc).

¹⁸ ’111 Patent, 10:52-62; Bhattacharjee, ¶¶69-71.

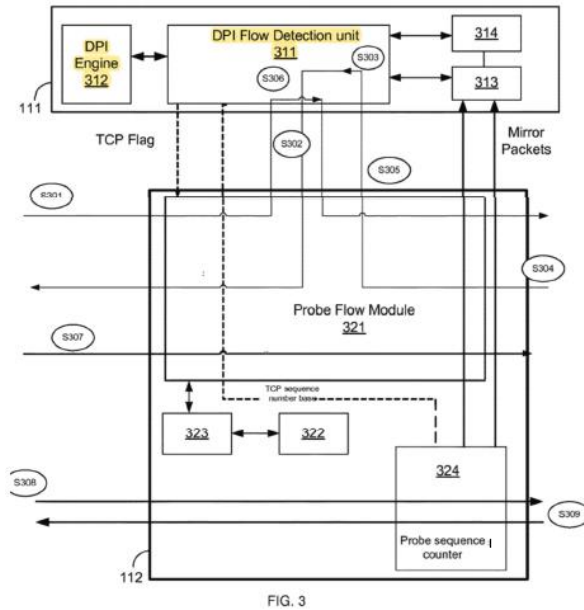
¹⁹ ’111 Patent, 2:27-30 (emphasis added); see *id.*, 3:56-59.

²⁰ *Id.*, 4:5-7 (emphasis added); see *id.*, 2:49-51, 4:8-11, 9:67-10:1.

²¹ ’111 Patent, 5:33-36.

²² *Id.*, 5:36-39; see *id.*, 5:40-59.

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.²³



'111 Patent, Fig. 3 (annotated); see *id.* Figs. 4-6.

Further, the claim term “instruction” should be construed to mean “a command to determine if a packet requires inspection or not.”²⁴ The '111 Patent discloses that “each network node 112 is configured to determine if an incoming packet requires inspection or not.”²⁵ The patent states that “the central controller

²³ See, e.g., Bhattacharjee, ¶71; '111 Patent, 4:8-18, 4:49-50, 8:1-5.

²⁴ '111 Patent, 10:56-62; Bhattacharjee, ¶72.

²⁵ '111 Patent, 4:12-14.

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.²⁶

Terms not specifically construed have their plain and ordinary meaning as understood by a POSITA.²⁷

IV. NO BASIS EXISTS FOR DISCRETIONARY DENIAL

Although institution is discretionary, the grounds for unpatentability presented in this Petition are neither cumulative or redundant to the prosecution of the ’111 Patent. The prior art here wasn’t previously considered and isn’t redundant to any combination of references previously considered. Nor do the *Fintiv* or *General Plastic* factors provide any basis for denial.

A. Discretionary Denial Under *Fintiv* Isn’t Appropriate

The six *Fintiv* factors considered for §314 denial favor institution.²⁸ Factor 1 is neutral. Patent Owner filed its complaint against Juniper asserting infringement of the ’111 Patent on July 31, 2023. Because the district court proceeding is in its

²⁶ *Id.*; *see id.*, 8:23-32, 8:40-53, 9:26-28; Bhattacharjee, ¶72.

²⁷ Bhattacharjee, ¶73.

²⁸ *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

early stages, no motion for stay has been filed. Due to the early stage of the district court proceeding, the parties have only minimally invested in the district court proceeding, no case schedule currently exists, and no overlap in prior art issues exist. Indeed, Juniper has yet to respond to the complaint. A final written decision will issue from this IPR well before the 27-month median time to trial in Delaware elapses. Factors 2-4 therefore weight in favor of institution. The parties are the same so factor 5 doesn't provide a basis for denying institution. And this Petition seeks joinder of already instituted proceedings. The institution of the original proceeding, IPR2023-00554, confirms the compelling evidence of unpatentability included in this petition which relies on the same evidence. Thus, factor 6 also weighs in favor of institution. And *Fintiv* offers no basis for denying institution here.

B. Discretionary Denial Under *General Plastic* Isn't Appropriate

Although the Board applies *General Plastic* to petitions seeking joinder, the factors don't support denying institution here.²⁹ This Petition is Petitioner's first challenge to the '111 Patent. And as its Motion for Joinder details, Petitioner agrees to assume a passive understudy role so that its participation won't affect the

²⁹ Cf. *Apple Inc. v. Uniloc 2017 LLC*, IPR2020-00854, Paper 9 (P.T.A.B. Oct. 28, 2020) (precedential) (applying *General Plastic* factors where Petitioner previously filed unsuccessful IPR petitions).

schedule of IPR2023-00554 or impact the Board’s finite resources. In doing so, Petitioner “effectively neutraliz[es] the *General Plastic* [discretionary denial] factors.”³⁰ *General Plastics* therefore offers no basis for discretionary denial.³¹

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

The combination of Lin and Swenson, along with the knowledge of a POSITA, renders claims 1-9, 12-24 and 27-31 obvious.³²

A. Lin

Lin “relates generally to computer security, and more particularly but not exclusively to software defined networking.”³³ “In one embodiment, a software defined networking (SDN) computer network includes an SDN controller and an SDN switch.”³⁴ “The SDN controller inserts flow rules in a flow table of the SDN

³⁰ *Apple Inc. v. Uniloc 2017 LLC*, IPR2018-00580, Paper 13 at 10-11 (P.T.A.B. Aug. 21, 2018).

³¹ *See General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 9-10 (P.T.A.B. Sept. 6, 2017) (precedential) (listing factors considered when petitioners have previously filed petitions).

³² Bhattacharjee, ¶¶74-206.

³³ Lin, 1:7-9; *see id.*, Abstract; Bhattacharjee, ¶¶50-56. Background discussion of software defined networking can be found in Paragraphs 21-29 of Dr. Bhattacharjee’s declaration and in EX1009.

³⁴ Lin, 1:58-60; Figures 6-8.

switch to create an SDN pipe between a sender component and a security component.”³⁵ “The SDN pipe allows outgoing packets sent by the sender component to be received by the security component.”³⁶ “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.”³⁷ 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.”³⁸ “The SDN controller 610 may include a flow policy database 611.”³⁹ “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.”⁴⁰ “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.”⁴¹

³⁵ *Id.*, 1:60-62; *see id.*, 1:62-64, 4:8-31, 4:53-65, 6:1-12.

³⁶ *Id.*, 1:64-65; *see id.*, 3:25-31, 6:40-48.

³⁷ *Id.*, 1:66-2:2; *see id.*, 3:31-33, 6:48-63, 7:9-21.

³⁸ *Id.*, 3:40-42, 4:7-9; *see id.*, 3:42-44, 4:9-12; Bhattacharjee, ¶51.

³⁹ Lin, 4:12-13.

⁴⁰ *Id.*, 4:13-16; *see id.*, 4:16-18.

⁴¹ *Id.*, 4:18-20.

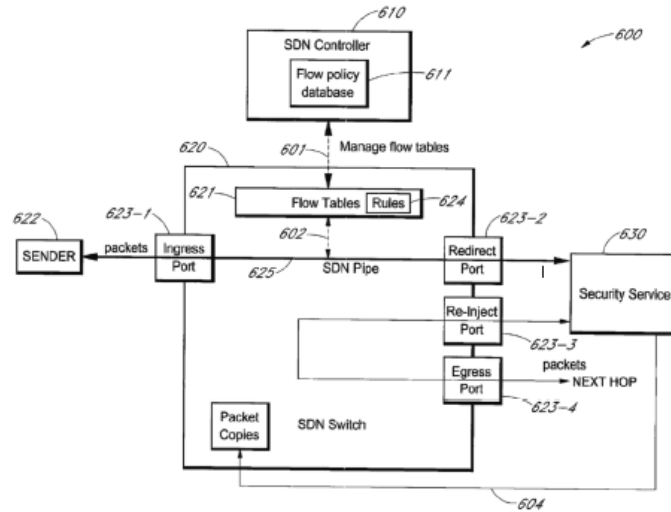


FIG. 6

Lin, Fig. 6.

B. Swenson

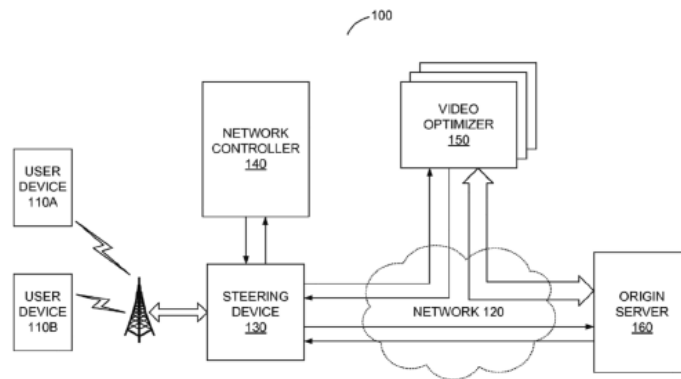
Swenson discloses a system and method for “selectively monitoring traffic in a service provider network.”⁴² Figure 1 of Swenson (reproduced below) shows that “[t]he network 120 is a communication network that transmits data between the user devices 110, the steering devices 130 and the origin server 160 and/or the video optimizer 150.”⁴³ “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.”⁴⁴ “Alternatively, the network controller 140 interfaces

⁴² Swenson, Abstract; *see id.*, ¶¶[0018]-¶¶[0022]; Bhattacharjee, ¶¶[63-68].

⁴³ Swenson, ¶¶[0023].

⁴⁴ *Id.*, ¶¶[0026]; *see id.*, ¶¶[0058].

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.”⁴⁵ “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.”⁴⁶ 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.”⁴⁷ “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.”⁴⁸



⁴⁵ *Id.*, ¶[0026].

⁴⁶ *Id.*, ¶[0026].

⁴⁷ *Id.*, ¶[0059]; *see id.*, ¶[0060].

⁴⁸ *Id.*, ¶[0059]; *see id.*, ¶[0060]; Bhattacharjee, ¶¶67-68.

Swenson, Fig. 1.**C. Claim 1**

Element [1.0], to the extent it is limiting, is disclosed by Lin.⁴⁹

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."⁵⁰ The specification further states that "[t]he present invention relates generally to computer security, and more particularly but not exclusively to software defined networking."⁵¹ A POSITA would've known that this computer network is a packet network.⁵² Indeed, Lin refers to the transmission of packets over the SDN computer network 600."⁵³ Moreover, Figure 6 of Lin shows that the system disclosed in Lin is for a packet network, as can be seen below:

⁴⁹ '111 Patent, 10:51-55.

⁵⁰ Lin, Abstract.

⁵¹ *Id.*, 1:7-9; *see id.*, 1:58-60, 2:47-65, 3:25-33, 3:40-64.

⁵² Bhattacharjee, ¶¶75-76.

⁵³ Lin, 4:19-21.

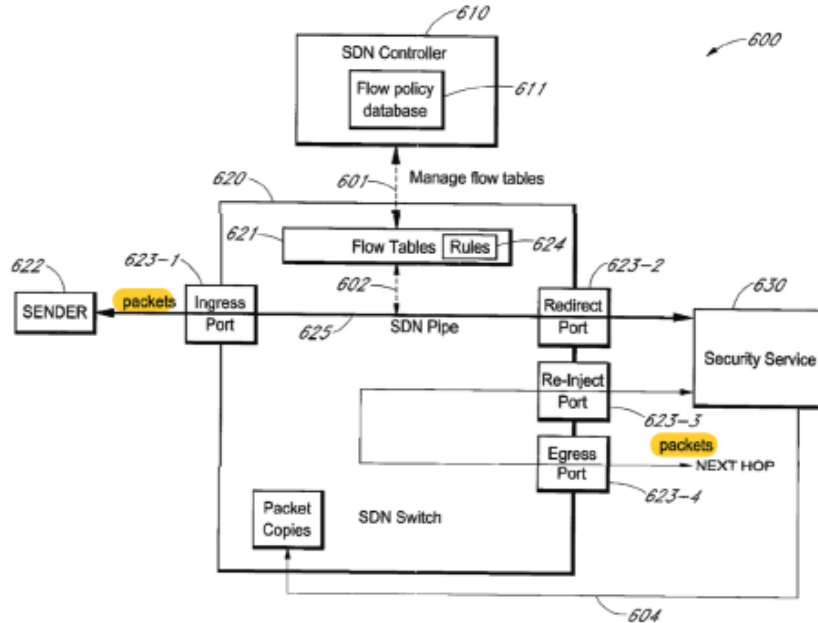


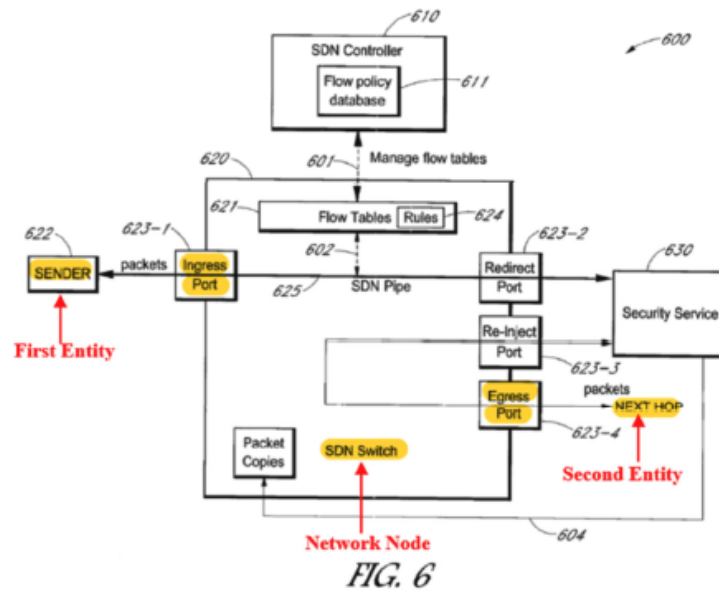
FIG. 6

Lin, Fig. 6 (annotated).

Further, Lin discloses an “SDN switch” that corresponds to the claimed network node for transporting packets between first and second entities.⁵⁴ 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 6.⁵⁵

⁵⁴ *Id.*, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63; Bhattacharjee, ¶77.

⁵⁵ Lin, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63, 7:10-23; 7:39-8:18, 9:63-10:22; Bhattacharjee, ¶77.



Lin, Fig. 6 (annotated).

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.⁵⁶ 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.⁵⁷ The SDN controller includes a “flow policy database” that contains flow policies to control the transmission of packets through the SDN switch.⁵⁸ 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

⁵⁶ Bhattacharjee, ¶78.

⁵⁷ Lin, 4:8-31.

⁵⁸ *Id.*, 4:8-31; *see id.*, 1:58-2:4, 6:1-12.

separate components.”⁵⁹ Further, Figures 1 and 6-8 of Lin show the SDN controller as external to the SDN switch:

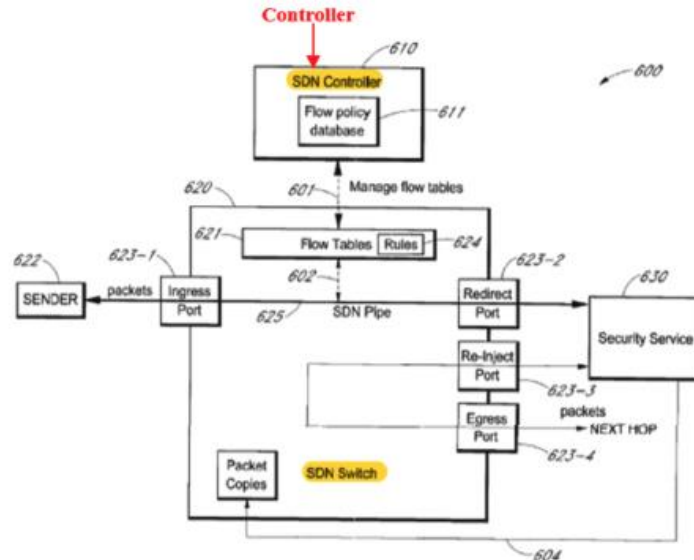


FIG. 6

Lin, Fig. 6 (annotated); see *id.*, Figs. 1, 7-8.

In addition, as addressed above in the claim construction section, the controller is an entity configured to perform DPI on packets.⁶⁰ 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).”⁶¹ Moreover,

⁵⁹ *Id.*, 3:51-52; see *id.*, 4:8-10.

⁶⁰ Bhattacharjee, ¶79. To the extent that the PTAB does not agree with this construction, Lin still discloses Element [1.0] for the reasons discussed above.

⁶¹ Lin, 3:11-12.

Lin discloses that security service 630 “may also comprise a physical machine, e.g., a server computer, an appliance, or a gateway computer, etc.”⁶² 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 POSITA would’ve 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.⁶³ Thus, a POSITA would’ve understood that one of the limited number of design options would’ve been to implement the security service as part of a controller configured to perform DPI analysis on packets, and a POSITA would’ve had a reasonable expectation that the controller would’ve been successful in performing DPI analysis.⁶⁴

Further, Swenson teaches the use of a controller configured to perform DPI.⁶⁵ 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

⁶² *Id.*, 5:51-55.

⁶³ Bhattacharjee, ¶79.

⁶⁴ *Id.*, ¶79.

⁶⁵ *Id.*, ¶80.

ICAP client interface 404.”⁶⁶ “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.”⁶⁷ A POSITA would’ve known that a “flow” is a series of packets having a specific signature.⁶⁸ As such, it would’ve been obvious to a POSITA that Swenson’s reference to “deep flow inspection” refers to performing DPI on one or more packets in a flow.⁶⁹

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.”⁷⁰ A POSITA would’ve 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.⁷¹

⁶⁶ Swenson, ¶[0059]; *see id.*, ¶[0060].

⁶⁷ *Id.*, ¶[0059] (emphasis added); *see id.*, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B.

⁶⁸ Bhattacharjee, ¶80.

⁶⁹ *Id.*

⁷⁰ Swenson, ¶[0065], Figure 5.

⁷¹ Bhattacharjee, ¶81.

Swenson thus demonstrates that it would've been well-known to a POSITA as of the priority date for the '111 Patent to implement a controller configured to perform DPI analysis in a system such as Lin.⁷²

A POSITA would've been motivated to combine the teachings of Lin and Swenson.⁷³ 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.⁷⁴ 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.⁷⁵ 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

⁷² Swenson, ¶¶[0046], ¶¶[0059]-¶¶[0060], ¶¶[0073], ¶¶[0076]-[0077], ¶¶[0084]-¶¶[0086], Figures 1 and 4A-4B; Bhattacharjee, ¶81.

⁷³ Bhattacharjee, ¶¶82-87.

⁷⁴ Lin, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; Swenson, ¶¶[0023]-¶¶[0032], ¶¶[0038]-¶¶[0043], ¶¶[0057]-¶¶[0061], Figures 1-4A; Bhattacharjee, ¶82.

⁷⁵ Lin, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; Swenson, ¶¶[0023]-¶¶[0032], ¶¶[0038]-¶¶[0043], ¶¶[0057]-¶¶[0061], Figures 1-4A; Bhattacharjee, ¶82.

identifying which packets should be sent to a security service.⁷⁶ Lin also discloses that the security service sends packets to their destination if the packets pass inspection.⁷⁷ 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.”⁷⁸

Similarly, Swenson discloses a method in which a network controller sends instructions to steering devices through which packet flows pass.⁷⁹ 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.”⁸⁰ Like Lin, Swenson teaches that only certain packets are redirected to the controller in order to be efficient in analyzing packets of interest.⁸¹ Further, a POSITA would’ve understood that Swenson, like Lin, could involve analyzing packets for a security

⁷⁶ Lin, 1:60-2:4, 3:21-24, 4:8-31, 4:53-67, 6:1-12, 6:54-63, Figure 6.

⁷⁷ *Id.*

⁷⁸ *Id.*, 7:23-27; *see id.*, 9:12-16, Figure 9.

⁷⁹ Swenson, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; Bhattacharjee, ¶83.

⁸⁰ Swenson, ¶[0059]; *see id.*, ¶[0060].

⁸¹ Swenson, ¶[0059]-¶[0060]; Bhattacharjee, ¶83.

function.⁸² A POSITA would've 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.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.⁸⁴ 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.⁸⁵

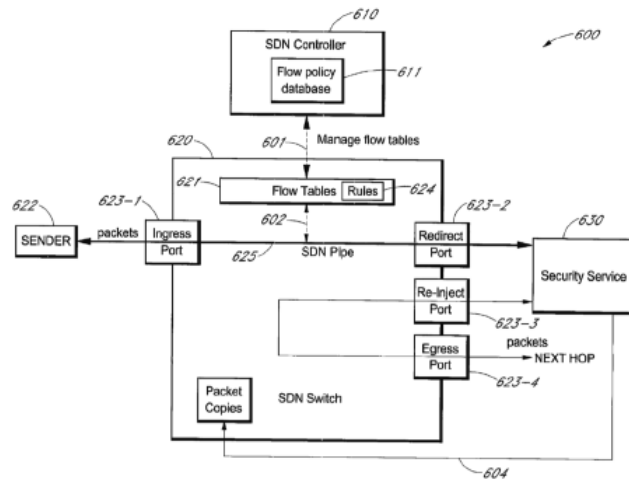


FIG. 6

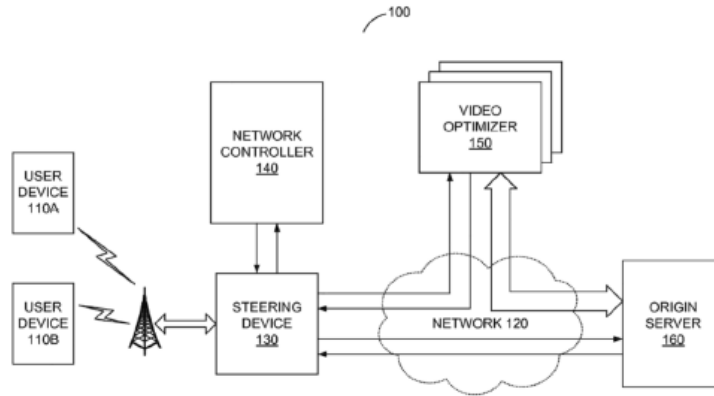
Lin, Fig. 6.

⁸² Bhattacharjee, ¶83.

⁸³ Swenson, ¶[0059]-¶[0060]; *see id.*, ¶[0039] (explaining that the controller can incorporate “security functions”); Bhattacharjee, ¶83.

⁸⁴ Swenson, ¶[0060].

⁸⁵ Bhattacharjee, ¶83.



Swenson, Fig. 1.

The determination of which packets to redirect (as opposed to redirecting all packets) increases the efficient operation of a computer network.⁸⁶ Lin and Swenson each teach that only certain packets may need to be redirected for further processing before being sent to the destination node.⁸⁷ 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.⁸⁸

⁸⁶ *Id.*, ¶84.

⁸⁷ Lin, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; Swenson, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A.

⁸⁸ Bhattacharjee, ¶84.

Further, a POSITA would've 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.⁸⁹ A POSITA would've understood there were efficiencies to implementing Lin's security processing module as part of the controller.⁹⁰ 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 POSITA would've known that security algorithms in a security component are often updated via software updates as new threats are identified. *Id.* A POSITA would've 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.⁹¹

A POSITA would've had a reasonable expectation of success in modifying Lin to implement the disclosure of Swenson to route packets to the central controller for inspection.⁹² As discussed above, Lin and Swenson have a similar architecture in which network nodes are controlled by a separate external

⁸⁹ Swenson, ¶¶[0059]-¶¶[0060], Figures 1, 4A-4B; Bhattacharjee, ¶85.

⁹⁰ Bhattacharjee, ¶85.

⁹¹ *Id.*

⁹² Bhattacharjee, ¶86.

controller.⁹³ If Swenson could be constructed to have the network nodes send packets to the central controller for security inspection, a POSITA would've understood that the same arrangement could be implemented in Lin.⁹⁴ Indeed, it would've been common sense for a POSITA to modify Lin in this way after reading Swenson.⁹⁵

Element [1.1] is disclosed by Lin.⁹⁶

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].⁹⁷

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.⁹⁸ Specifically, the SDN controller inserts “flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component

⁹³ Lin, 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; Swenson, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A.

⁹⁴ Bhattacharjee, ¶86.

⁹⁵ *Id.*

⁹⁶ '111 Patent, 10:56-58.

⁹⁷ Lin, 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; Bhattacharjee, ¶¶88-89.

⁹⁸ Bhattacharjee, ¶90.

and a security component.”⁹⁹ 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.¹⁰⁰

Further, Lin explains that a separate packet-applicable criterion sent by the controller to the network node over the packet network.¹⁰¹ 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.”¹⁰² A POSITA would’ve known that the “one or more conditions” used to identify “particular packets” for inspection would be packet-applicable criterion.¹⁰³

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

⁹⁹ Lin, 1:58-2:4.

¹⁰⁰ *Id.*; *see id.*, 4:14-31, 4:53-67, 6:1-12; 7:39-8:18.

¹⁰¹ Bhattacharjee, ¶91.

¹⁰² Lin, 4:21-31; *see id.*, 1:60-62, 5:8-12.

¹⁰³ Bhattacharjee, ¶¶91-92; Lin, 4:23-31, 5:8-12.

dst' (IP address of the destination of the packet)."¹⁰⁴ 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."¹⁰⁵ 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.¹⁰⁶

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.¹⁰⁷ 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 6.¹⁰⁸ 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."¹⁰⁹ A POSITA would've understood

¹⁰⁴ Lin, 5:16-21; Bhattacharjee, ¶91.

¹⁰⁵ Lin, 5:22-24; *see id.*, 5:26-36, 6:1-12, 6:40-54, Table 1.

¹⁰⁶ *Id.*, 2:2-4, 7:24-8:18.

¹⁰⁷ Bhattacharjee, ¶93.

¹⁰⁸ Lin, 6:1-4; *see id.*, 6:40-41.

¹⁰⁹ *Id.*, 7:24-26; *see id.*, 7:27-8:18, Tables 2-3.

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.¹¹⁰

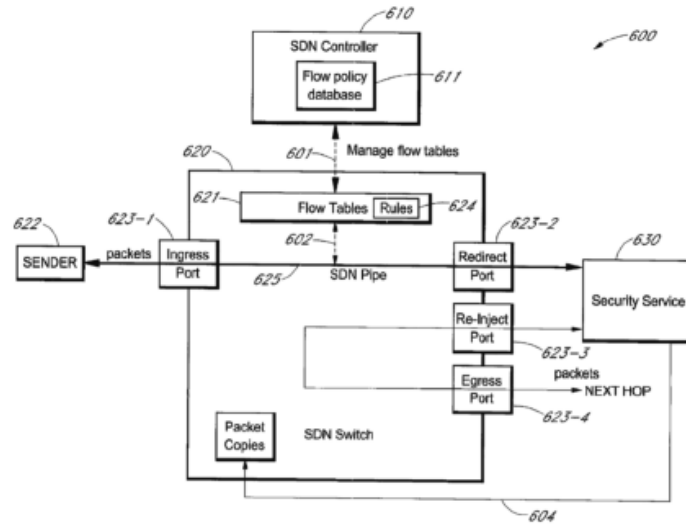


FIG. 6

Lin, Fig. 6.

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.¹¹¹ As such, Lin discloses Element [1.2] for substantially the same reasons that Lin discloses Element [1.1].¹¹²

Element [1.3] is disclosed by Lin.¹¹³

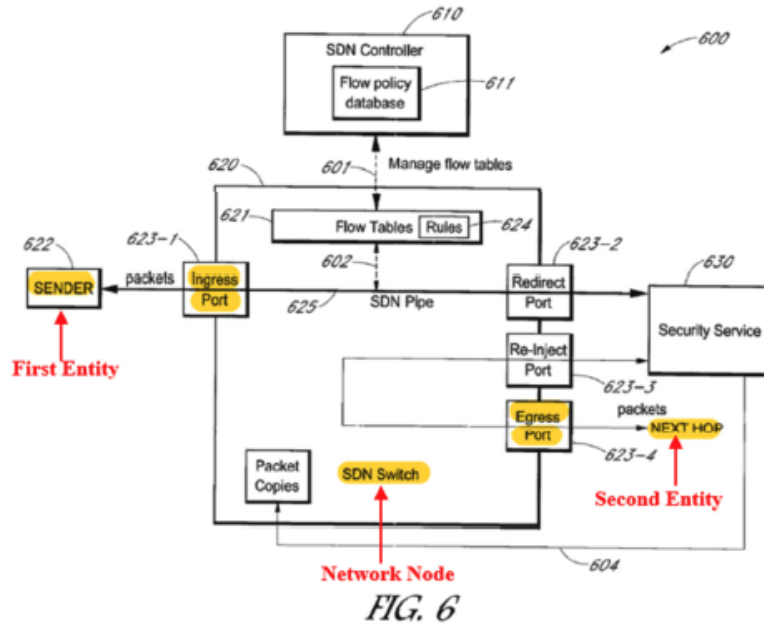
¹¹⁰ Bhattacharjee, ¶¶93-94.

¹¹¹ '111 Patent, 10:59-60.

¹¹² Bhattacharjee, ¶¶95-100.

¹¹³ '111 Patent, 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 6.¹¹⁴ A POSITA would’ve understood that the packet is addressed to the next hop or second entity.¹¹⁵ The IP address of the destination of the packet is one of the criterion that is specifically discussed in Lin.¹¹⁶



¹¹⁴ Lin, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63, 7:10-23; 7:39-8:18, 9:63-10:22; Bhattacharjee, ¶¶101-103.

¹¹⁵ Bhattacharjee, ¶102.

¹¹⁶ Lin, 5:16-31.

Lin, Fig. 6 (annotated).

Element [1.4] is disclosed by Lin.¹¹⁷

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.”¹¹⁸ A POSITA would’ve 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.¹¹⁹ 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.”¹²⁰ 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.”¹²¹ Further, in discussing the examples shown in Table 2 and Table 3, Lin discloses that the SDN switch implements “bypass flow rules” that check

¹¹⁷ ’111 Patent, 10:63-64.

¹¹⁸ Lin, 4:23-31; *see id.*, 5:8-12, 6:1-4, 6:40-41, 7:24-8:18.

¹¹⁹ Bhattacharjee, ¶104-105; Lin, 4:23-31, 5:8-12.

¹²⁰ Lin, 5:16-21; Bhattacharjee, ¶105.

¹²¹ Lin, 5:22-24; *see id.*, 5:26-36, 6:1-12, 6:40-54, Table 1.

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.¹²²

Element [1.5] is disclosed by Lin.¹²³

Lin explains that the SDN switch implements “bypass flow rules” that check whether a packet meets certain packet-applicable criterion.¹²⁴ 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.¹²⁵ 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.¹²⁶ 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

¹²² *Id.*, 7:24-8:18; Bhattacharjee, ¶¶105-106.

¹²³ ’111 Patent, 10:65-67.

¹²⁴ Lin, 7:24-8:18; *see id.*, 5:16-24, 5:26-36, 6:1-12, 6:40-54; Bhattacharjee, ¶¶107-109.

¹²⁵ Lin, 7:24-8:18.

¹²⁶ *Id.*, 7:64-8:18.

over the packet network to its destination node (which corresponds to the claimed second entity).¹²⁷ 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.”¹²⁸

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

Lin, Table 3 (annotated).

Element [1.6] is disclosed by Lin.¹²⁹

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.¹³⁰ Further, in some embodiments, the bypass flow rules provide that packet which meet certain packet-applicable criterion should be routed

¹²⁷ *Id.*, 8:10-18; Bhattacharjee, ¶¶109-110.

¹²⁸ Lin, 8:10-18, Table 3.

¹²⁹ ’111 Patent, 11:1-4.

¹³⁰ Lin, 7:24-8:18; see *id.*, 5:16-24, 5:26-36, 6:1-12, 6:40-54; Bhattacharjee, ¶¶111-114.

to the security device.¹³¹ 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.¹³² 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).¹³³

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

Lin, Table 3 (annotated).

D. Claim 2

Claim 2 is obvious in view of Lin and the knowledge of a POSITA.¹³⁴

¹³¹ Lin, 7:24-8:18.

¹³² *Id.*, 7:64-8:18.

¹³³ *Id.*, 8:10-18, Table 3; Bhattacharjee, ¶113.

¹³⁴ '111 Patent, 11:5-10.

Lin teaches that the instruction can be a “terminate” instruction.¹³⁵ A POSITA would’ve 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.¹³⁶ Lin discloses that one of its “packet manipulation actions” is “dropping the packet.”¹³⁷ Lin also states that the security service 630 “may instruct the SDN switch 620 to drop the copied packets...”¹³⁸ A POSITA would’ve understood these passages from Lin to teach that the flow rules in Lin could include a terminate instruction.¹³⁹

Where the SDN switch in Lin receives a terminate instruction, a POSITA would’ve 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.¹⁴⁰ Lin states that the SDN switch can be instructed to drop (i.e., block) packets before the packets are sent to their destination node.¹⁴¹ It would’ve been straightforward for the flow rules in Lin

¹³⁵ Bhattacharjee, ¶¶115-116.

¹³⁶ *Id.*

¹³⁷ Lin, 1:28-32.

¹³⁸ *Id.*, 7:19-22.

¹³⁹ Bhattacharjee, ¶116.

¹⁴⁰ *Id.*, ¶117.

¹⁴¹ Lin, 1:28-32, 7:19-22.

to also instruct that packets are dropped by the SDN switch before they can be sent to controller.¹⁴² For example, a POSITA would've 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.¹⁴³

E. Claim 3

Claim 3 is obvious in view of Lin.¹⁴⁴

Further, Lin teaches that the instruction can be a “mirror” instruction.¹⁴⁵ A POSITA would've 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).¹⁴⁶ Lin discloses both methods of mirroring a packet. For example, Lin states that incoming packets can be “redirected or mirrored to the security

¹⁴² Bhattacharjee, ¶¶117-118.

¹⁴³ *Id.*

¹⁴⁴ '111 Patent, 11:11-16.

¹⁴⁵ Bhattacharjee, ¶¶119-120.

¹⁴⁶ *Id.*

service.”¹⁴⁷ 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.¹⁴⁸ A POSITA would’ve understood these passages from Lin to teach that the flow rules in Lin could include a mirror instruction.¹⁴⁹

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.”¹⁵⁰ As discussed above for Element [1.6], it would’ve 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.¹⁵¹

¹⁴⁷ Lin, 3:25-33.

¹⁴⁸ *Id.*, 7:10-22.

¹⁴⁹ Bhattacharjee, ¶120.

¹⁵⁰ Lin, 7:10-22.

¹⁵¹ *Id.*; Bhattacharjee, ¶¶121-123.

F. Claim 4

Claim 4 is obvious over Lin, in view of the knowledge of a POSITA.¹⁵²

Lin teaches that the instruction can be a “probe” instruction.¹⁵³ A POSITA would’ve 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.¹⁵⁴ 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.¹⁵⁵ 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.¹⁵⁶ A POSITA would’ve 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.¹⁵⁷

¹⁵² ’111 Patent, 11:17-26.

¹⁵³ Bhattacharjee, ¶¶124-125.

¹⁵⁴ *Id.*

¹⁵⁵ Lin, 1:58-2:4; *see id.*, 4:14-31, 4:53-67, 6:1-12; 9:23-40, Figures 6-9.

¹⁵⁶ *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”).

¹⁵⁷ Bhattacharjee, ¶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.¹⁵⁸ It would've 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.¹⁵⁹

Further, a POSITA would've known from Lin that, when the packet is received by the controller, the packet is analyzed by the security service in the controller.¹⁶⁰ A POSITA also would've 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).¹⁶¹

G. Claim 5

Claim 5 is obvious in view of the combination of Lin and Swenson.¹⁶²

¹⁵⁸ Lin, Abstract, 1:58-2:4, 4:53-67, 5:26-36, 6:1-12, 6:40-63, 7:10-8:18, Figures 6-9.

¹⁵⁹ Bhattacharjee, ¶126.

¹⁶⁰ *Id.*, ¶127; Lin, 6:48-50, 5:45-55, 7:10-22, 8:33-35, Figure 9.

¹⁶¹ Bhattacharjee, ¶127; Lin, 6:54-63, 7:10-22, 8:35-45, Figure 9.

¹⁶² '111 Patent, 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.¹⁶³

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.¹⁶⁴ 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.”¹⁶⁵ A POSITA would’ve 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.¹⁶⁶ Responsive to the HTTP packet flow satisfying that criterion, the steering device applies an instruction pursuant to

¹⁶³ Bhattacharjee, ¶¶129-130.

¹⁶⁴ *Id.*, ¶131.

¹⁶⁵ Swenson, ¶[0059] (emphasis added); *see id.*, ¶[0060], ¶[0065], Figures 1, 4A-4B.

¹⁶⁶ Bhattacharjee, ¶¶131-132.

which the HTTP request and a portion of the HTTP response is sent to the network controller 140.¹⁶⁷

H. Claim 6

Claim 6 is obvious over Swenson in view of the knowledge of a POSITA.¹⁶⁸

Swenson discloses a flow cache memory 322 that is part of the network controller 140, as shown below in Figure 3 of Swenson.¹⁶⁹ 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.”¹⁷⁰

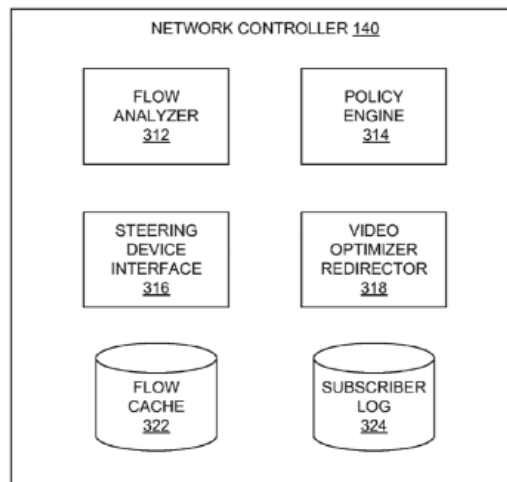


FIG. 3

¹⁶⁷ Swenson, ¶[0059].

¹⁶⁸ '111 Patent, 11:31-33.

¹⁶⁹ Bhattacharjee, ¶¶133-134.

¹⁷⁰ Swenson, ¶[0061].

Swenson, Fig. 3.

Swenson also explains that “[t]he flow cache entry keeps track of the flow and its associated bandwidth.”¹⁷¹ 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.¹⁷² Further, Swenson states that a flow cache entry can also store the MAC address or TCP source port associated with a received packet.¹⁷³ A POSITA would’ve understood from these disclosures that Swenson teaches that the controller stores at least a portion of a packet in a flow cache memory.¹⁷⁴

I. Claim 7

Claim 7 is obvious over Swenson.¹⁷⁵

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

¹⁷¹ *Id.*, ¶[0061]; see *id.*, ¶[0040], ¶[0046], ¶[0062].

¹⁷² *Id.*, ¶[0073] (discussing Figure 7); see *id.*, ¶[0074]-¶[0085].

¹⁷³ *Id.*, ¶[0084].

¹⁷⁴ Bhattacharjee, ¶¶134-135.

¹⁷⁵ ’111 Patent, 11:34-37.

HTTP request and a portion of the HTTP response to the network controller 140 over the ICAP client interface 404.”¹⁷⁶ A POSITA would’ve 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.¹⁷⁷ Further, a POSITA would’ve 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.¹⁷⁸

J. Claim 8

Claim 8 is obvious over Swenson in view of the knowledge of a POSITA.¹⁷⁹

As discussed above for Claim 7, Swenson discloses that a portion of the HTTP request or the HTTP response is sent to the controller.¹⁸⁰ It would’ve been obvious to a POSITA that the portion of the HTTP request or the HTTP response

¹⁷⁶ Swenson, ¶[0059] (emphasis added); *see id.*, ¶[0060], ¶[0065], Figures 1, 4A-4B.

¹⁷⁷ Bhattacharjee, ¶¶136-137.

¹⁷⁸ *Id.*, ¶¶137-138.

¹⁷⁹ ’111 Patent, 11:38-41.

¹⁸⁰ Swenson, ¶[0059]-¶[0060], Figures 1, 4A-4B.

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.¹⁸¹

K. Claim 9

Claim 9 is obvious in view of Swenson.¹⁸²

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.¹⁸³

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

¹⁸¹ Bhattacharjee, ¶¶139-141.

¹⁸² ’111 Patent, 11:42-44.

¹⁸³ Swenson. ¶[0059] (emphasis added); *see id.*, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B; Bhattacharjee, ¶¶142-143.

message to determine whether to monitor the flow or optimize the video.”¹⁸⁴ A POSITA would’ve 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.¹⁸⁵ This is similar to Lin, which explains that an analysis is performed on its packets.¹⁸⁶

L. Claim 12

Claim 12 is obvious over Lin.¹⁸⁷ Lin states that the analysis performed on the packets includes a security processing function.¹⁸⁸

¹⁸⁴ *Id.*, ¶[0065], Figure 5.

¹⁸⁵ Bhattacharjee, ¶143.

¹⁸⁶ Lin, 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; Bhattacharjee, ¶¶143-144.

¹⁸⁷ ’111 Patent, 11:55-57.

¹⁸⁸ Lin, 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; Bhattacharjee, ¶¶145-148.

M. Claim 13

Claim 13 is obvious over Lin.¹⁸⁹ Lin discloses that its system analyzes packets by applying a security application that comprises either a firewall or intrusion detection functionality, or both.¹⁹⁰

N. Claim 14

Claim 14 is obvious over both Lin and Swenson.¹⁹¹ Lin states that its analysis includes “deep packet inspection (DPI).”¹⁹² As discussed above for Element [1.0], Swenson also discloses that the flow analyzer in network controller 140 performs DPI.¹⁹³

O. Claim 15

Claim 15 is obvious in view of both Lin and Swenson.¹⁹⁴

A POSITA would’ve known that a packet includes distinct header and payload fields.¹⁹⁵

¹⁸⁹ ’111 Patent, 11:58-60.

¹⁹⁰ Lin, 5:45-50 (referring specifically to “compliance with firewall rules” and “network intrusion detection”), 3:11-12, 6:50-54; Bhattacharjee, ¶¶149-151.

¹⁹¹ ’111 Patent, 11:61-63.

¹⁹² Lin, 3:11-12; Bhattacharjee, ¶¶152-155.

¹⁹³ Swenson, ¶[0059]-¶[0060], ¶[0065].

¹⁹⁴ ’111 Patent, 11:64-67.

¹⁹⁵ EX1008, Section 2; Bhattacharjee, ¶¶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.”¹⁹⁶ A POSITA would’ve 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 POSITA would’ve understood the references to DPI in both Swenson and Lin to involve checking part or the whole of the payload field.¹⁹⁷

P. Claim 16

Claim 16 is obvious over Lin in view of the knowledge of a POSITA.¹⁹⁸ A POSITA would’ve known that a packet includes distinct header and payload fields.¹⁹⁹ Further, it would’ve been well-known to a POSITA that the header field includes one or more flag bits.²⁰⁰ In addition, it would’ve been obvious to a POSITA that the flow rules of Lin could have used a determination of whether or

¹⁹⁶ Swenson, ¶[0065], Figure 5; Bhattacharjee, ¶158.

¹⁹⁷ Bhattacharjee, ¶¶159-160; Lin, 3:11-12, 5:45-50; Swenson, ¶[0059]-¶[0060].

¹⁹⁸ ’111 Patent, 12:1-4.

¹⁹⁹ EX1008, Section 2; Bhattacharjee, ¶¶161-162.

²⁰⁰ Bhattacharjee, ¶¶162-163; EX1008, Section 2.

not a flag bit was set as one of Lin’s “conditions” that serve as packet-applicable criterion.²⁰¹

Q. Claim 17²⁰²

Claim 17 is obvious over Lin in view of the knowledge of a POSITA.²⁰³ Lin discloses embodiments in which the packet is a TCP packet.²⁰⁴ Further, it would’ve been well-known to a POSITA that flag bits such as a SYN flag bit, an ACK flag bit, a FIN flag bit, and a RST flag bit were 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).²⁰⁵ As such, it would’ve been obvious to a POSITA 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.²⁰⁶

²⁰¹ Bhattacharjee, ¶¶162-163; Lin, 4:23-31; *see id.*, 5:8-12, 7:24-8:18.

²⁰² Claim 17 uses the language “comprises comprise.” This may be an error. For this Petition, Petitioner interprets that claim language to mean “comprises.”

²⁰³ ’111 Patent, 12:5-9.

²⁰⁴ Lin, 7:24-8:18, Tables 2-3.

²⁰⁵ Bhattacharjee, ¶¶164-165.

²⁰⁶ *Id.*, ¶¶165-166; Lin, 4:23-31; *see id.*, 5:8-12, 7:24-8:18.

R. Claim 18

Claim 18 is obvious over Lin in view of the knowledge of a POSITA.²⁰⁷ A POSITA would've known that a packet includes distinct header and payload fields.²⁰⁸ Moreover, it would've 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.²⁰⁹ 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.²¹⁰

S. Claim 19

Claim 19 is obvious over both Lin for the reasons as those discussed above for Claim 18.²¹¹

²⁰⁷ '111 Patent, 12:10-16.

²⁰⁸ EX1008, Section 2; Bhattacharjee, ¶¶167-168.

²⁰⁹ Bhattacharjee, ¶168.

²¹⁰ Lin, 5:8-25; *see id.*, 5:26-36, 6:1-12, 6:40-54, Table 1; Bhattacharjee, ¶¶169-170.

²¹¹ '111 Patent, 12:17-18; Lin, 5:8-36, 6:1-12, 6:40-54, Table 1; Bhattacharjee, ¶¶171.

T. Claim 20

Claim 20 is obvious over Lin in view of the knowledge of a POSITA.²¹²

Lin provides examples of TCP packets that are transported over a TCP network when discussing Tables 2 and 3 of that patent.²¹³ Further, Lin discloses tracking HTTP packets whose source and destination TCP ports are port 80.²¹⁴ Lin does not specifically disclose that a TCP packet has a TCP sequence number or a TCP sequence mask field, but this would've been obvious to a POSITA because those fields are always present in a TCP packet.²¹⁵

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.²¹⁶ 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.²¹⁷ Lin states that, if a packet satisfies this criterion (i.e., the packet

²¹² '111 Patent, 12:20-27.

²¹³ Lin, 7:24-8:18; *see id.*, 10:48-50; Bhattacharjee, ¶¶172-173.

²¹⁴ Lin, 7:24-8:18.

²¹⁵ Bhattacharjee, ¶173.

²¹⁶ *Id.*, ¶¶174-175.

²¹⁷ Lin, 7:64-8:18.

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).²¹⁸

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

Lin, Table 3 (annotated).

U. Claim 21

Claim 21 is obvious over Lin in view of the knowledge of a POSITA.²¹⁹ As discussed above for Claims 18-19, Lin discloses the use of IP and MAC addresses to identify particular packets.²²⁰ A POSITA would've understood that, if IP and MAC addresses were being used in Lin, then the packet network would've included one of the Internet, a Wide Area Network ("WAN"), a Local Area

²¹⁸ *Id.*, 8:10-18.

²¹⁹ '111 Patent, 12:28-32.

²²⁰ Lin, 5:8-36, 6:1-12, 6:40-54, Table 1.

Network (“LAN”), a Metropolitan Area Network (“MAN”), Internet Service Provider (“ISP”) backbone, datacenter network, or inter-datacenter network.²²¹

V. Claim 22

Claim 22 is obvious over Swenson.²²² Swenson teaches “user devices 110” on one end of its system and an “origin server 160” on the other end of its system.²²³ Swenson discloses that the user devices “are computing devices with network capabilities,” such as “laptops, notebooks, tablets, smart telephones, or personal digital assistants (PDAs).”²²⁴ A POSITA would’ve understood these to be client devices.²²⁵ 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.²²⁶

²²¹ Bhattacharjee, ¶¶176-178.

²²² ’111 Patent, 12:33-36.

²²³ Swenson, ¶[0023]; Bhattacharjee, ¶¶179-180.

²²⁴ Swenson, ¶[0025].

²²⁵ Bhattacharjee, ¶¶180-181.

²²⁶ Swenson, ¶[0058]; Bhattacharjee, ¶180.

W. Claim 23

Claim 23 is obvious over Swenson.²²⁷ 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.²²⁸ A POSITA would’ve 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.²²⁹ Further, Swenson discloses that the user devices (i.e., client devices) can include “laptops, notebooks, tablets, smart telephones, or personal digital assistants (PDAs).”²³⁰

X. Claim 24

Claim 24 is obvious in view of Lin.²³¹ Lin discloses the use of the OpenFlow protocol for the communication between the SDN switch (i.e., the network node) and the SDN controller.²³² 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.

²²⁷ ’111 Patent, 12:37-41.

²²⁸ Swenson, ¶[0023].

²²⁹ *Id.*, ¶[0058]; Bhattacharjee, ¶¶182-184.

²³⁰ Swenson, ¶[0025].

²³¹ ’111 Patent, 12:42-44.

²³² Lin, 1:18-43, 2:12-13, 3:42-46, 3:65-4:6, 6:35-39; Bhattacharjee, ¶¶185-187.

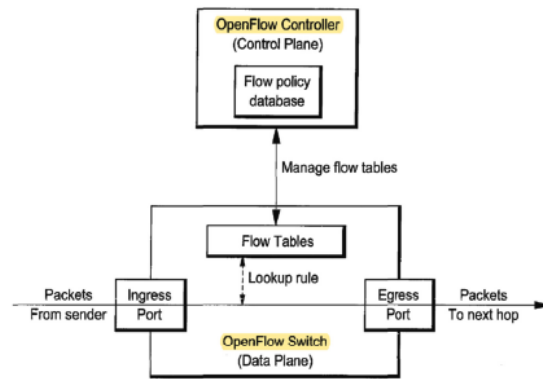


FIG. 1
(PRIOR ART)

Lin, Fig. 1 (annotated).

Y. Claim 27

Claim 27 is obvious in view of Lin.²³³ As discussed above for Element [1.0], Lin discloses a SDN switch that operates as the claimed network node.²³⁴

Z. Claim 28

Claim 28 is obvious in view of Lin.²³⁵ As discussed above for Claims 18-19, Lin discusses the use of IP addresses to identify packets being transmitted across the packet network.²³⁶ A POSITA would've understood from the use of the IP

²³³ '111 Patent, 12:51-52.

²³⁴ Lin, 1:58-2:4, 4:33-67, 6:13-23, 6:57-63; Bhattacharjee, ¶¶188-190.

²³⁵ '111 Patent, 12:53-55.

²³⁶ Lin, 5:8-36, 6:1-12, 6:40-54, Table 1.

addresses that packet network in Lin is an Internet Protocol (IP) network and the packet is an IP packet.²³⁷

AA. Claim 29

Claim 29 is obvious in view of Lin.²³⁸ Lin provides specific examples of TCP packets that are transported over a TCP network when discussing Tables 2 and 3 of that patent.²³⁹

BB. Claim 30

Element [30.0] is obvious in view of Lin.²⁴⁰ Lin teaches that its method can be applied to “any packet received by the SDN switch.”²⁴¹ 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].²⁴²

Element [30.1] is obvious in view of the combination of Lin and Shieh.²⁴³

²³⁷ Bhattacharjee, ¶¶191-193.

²³⁸ ’111 Patent, 12:56-58.

²³⁹ Lin, 7:24-8:18; see id., 10:48-50; Bhattacharjee, ¶¶194-196.

²⁴⁰ ’111 Patent, 12:59-63.

²⁴¹ Lin, 6:40-63.

²⁴² Bhattacharjee, ¶¶197-199.

²⁴³ ’111 Patent, 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].²⁴⁴

CC. Claim 31

Claim 31 is obvious in view of Lin.²⁴⁵ Lin explains that its packet network is a SDN network that uses an SDN controller and an SDN switch.²⁴⁶ 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.²⁴⁷ Further, Lin explains that, when using the Openflow protocol, the communication between the SDN switch and the SDN controller is through a control plane.²⁴⁸

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

A. Shieh

Shieh relates “generally to network security” and discloses a “network system [that] includes a security device and a network access device.”²⁴⁹ “The

²⁴⁴ Bhattacharjee, ¶¶200-203.

²⁴⁵ ’111 Patent, 13:4-7.

²⁴⁶ Lin, Abstract, 1:7-9, 1:58-2:4, 3:53-4:67, Figures 2, 6-8. 2, 6-8.

²⁴⁷ *Id.*, 1:11-54, 2:49-50, 3:42-52, Figure 1; Bhattacharjee, ¶¶204-205.

²⁴⁸ Lin, 1:11-54, 2:49-50, 3:42-52, 3:65-4:5, Figure 1; Bhattacharjee, ¶¶205-206.

²⁴⁹ Shieh, ¶[0002]; Bhattacharjee, ¶¶57-62.

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.²⁵⁰ “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.”²⁵¹ “[I]n response to a response received from the security device, the packet is routed to the destination node dependent upon the response.”²⁵²

²⁵⁰ Shieh, Abstract; *see id.*, ¶[0002], Figures 1, 2A, 3.

²⁵¹ *Id.*, Abstract; *see id.*, ¶[0042], ¶[0049].

²⁵² *Id.*, Abstract; *see id.*, ¶[0017], ¶[0018], ¶[0023], ¶[0029], ¶[0037], Claim 1, Figure 2B; Bhattacharjee, ¶57.

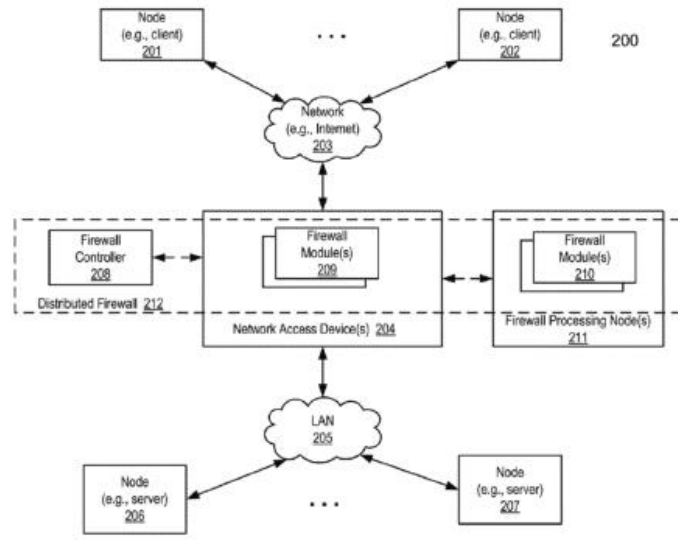


FIG. 1

Shieh, Fig. 1.

B. Claim 1

To the extent Element [1.0] is limiting, it is obvious in view of Shieh.²⁵³

Shieh discloses a method for use with a packet network.²⁵⁴ 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."²⁵⁵ The specification of Shieh further explains that its system performs operations on packets passing through a network.²⁵⁶ Further, Figure 7 of Shieh, reproduced below, shows the method

²⁵³ '111 Patent, 10:52-55.

²⁵⁴ Bhattacharjee, ¶¶208-209.

²⁵⁵ Lin, Abstract.

²⁵⁶ Shieh, ¶[0002], ¶[0017], ¶[0018], ¶[0021], ¶[0023].

disclosed in Shieh, which involves a “network access device” performing operations on packets passing through that network access device.²⁵⁷

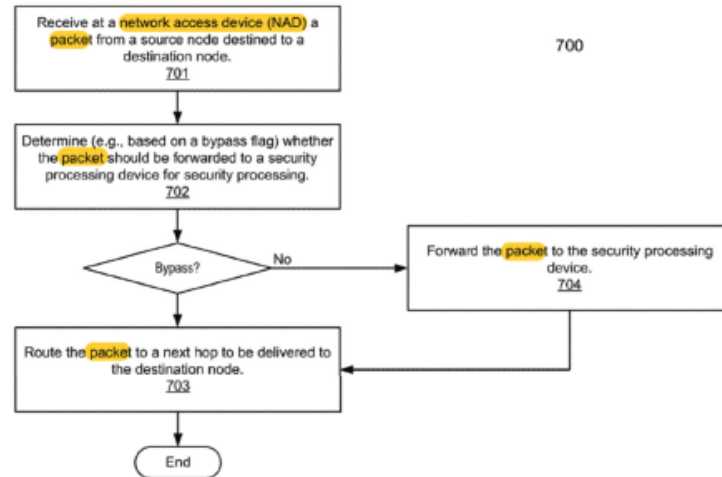


FIG. 7

Lin, Fig. 7 (annotated).

Further, Shieh discusses a network node for transporting packets between first and second entities.²⁵⁸ 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).²⁵⁹ The way in which

²⁵⁷ *Id.*, ¶[0049], Figure 7.

²⁵⁸ Bhattacharjee, ¶210.

²⁵⁹ Shieh, ¶[0027]; ¶[0037], ¶[0049], Figures 2B, 7.

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:

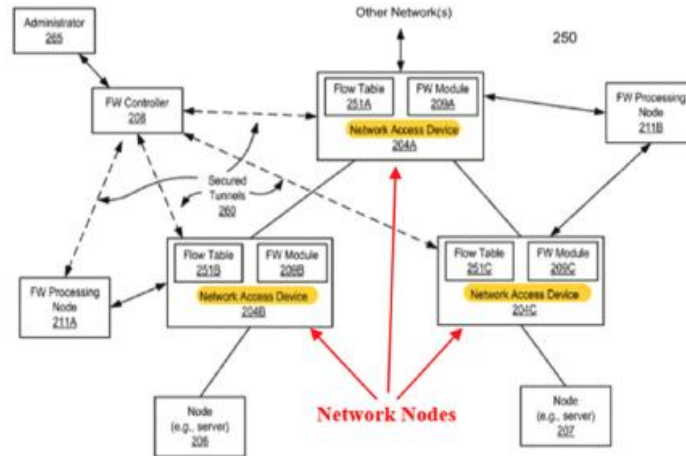


FIG. 2A

Shieh, Fig. 2A (annotated).

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.²⁶⁰ Shieh explains that network access devices 204A-204C are “configured” by (i.e., under the control of) “a controller or a management entity.”²⁶¹ Figure 2A shows that the controller 208 is external to the network node.²⁶²

²⁶⁰ Bhattacharjee, ¶211.

²⁶¹ Shieh, ¶[0018]; *see id.*, ¶[0021] (noting that firewall controller 208 may be “external to network access device 204”), ¶[0025], ¶[0029], ¶[0042].

²⁶² *Id.*, Fig. 2A, ¶[0029]; *see id.*, ¶[0025]-¶[0026].

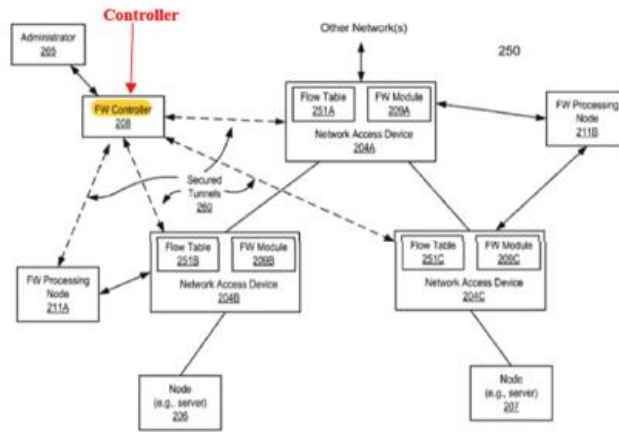


FIG. 2A

Shieh, Fig. 2A (annotated).

In addition, as addressed in the above claim construction section, the claimed controller is an entity configured to perform DPI on packets.²⁶³ The combination of Shieh and Swenson renders it obvious that the controller is configured to perform DPI.²⁶⁴ Shieh discloses that its security processing function includes DPI.²⁶⁵ Moreover, as discussed above in Ground 1, Swenson discloses a controller configured to perform DPI.²⁶⁶

²⁶³ Bhattacharjee, ¶212. To the extent that the PTAB does not agree with this construction, Shieh still discloses Element [1.0] for the reasons discussed above.

²⁶⁴ Bhattacharjee, ¶¶212-214.

²⁶⁵ Shieh, ¶[0021].

²⁶⁶ See Swenson, ¶[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

Given the similarities between the disclosures in Lin and Shieh, a POSITA would've 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.²⁶⁷ 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 packets should be redirected and which packets can be sent directly to a destination node.²⁶⁸

Further, a POSITA would've understood both Shieh and Swenson to disclose redirecting packets for analysis as part of a security function.²⁶⁹ 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.”²⁷⁰ Similarly, a POSITA would've understood that the bandwidth monitoring in Swenson can be used as a security application that monitors for DOS attacks that

response payload by network controller 140), Figures 1, 4A-4B; Bhattacharjee, ¶¶213-214.

²⁶⁷ Bhattacharjee, ¶¶82-86, 215-220.

²⁶⁸ Shieh, ¶[0017]-¶[0018], ¶[0024]-¶[0025], ¶[0029]-¶[0030], Figures 1-3, 7; Swenson, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; Bhattacharjee, ¶215.

²⁶⁹ Bhattacharjee, ¶¶216-217.

²⁷⁰ Shieh, Abstract; *see id.*, ¶[0042], ¶[0049].

occupy significant bandwidth in a network.²⁷¹ 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.²⁷²

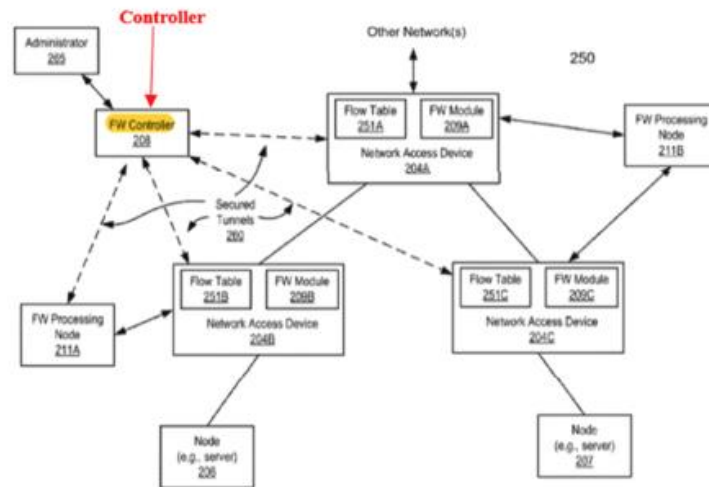
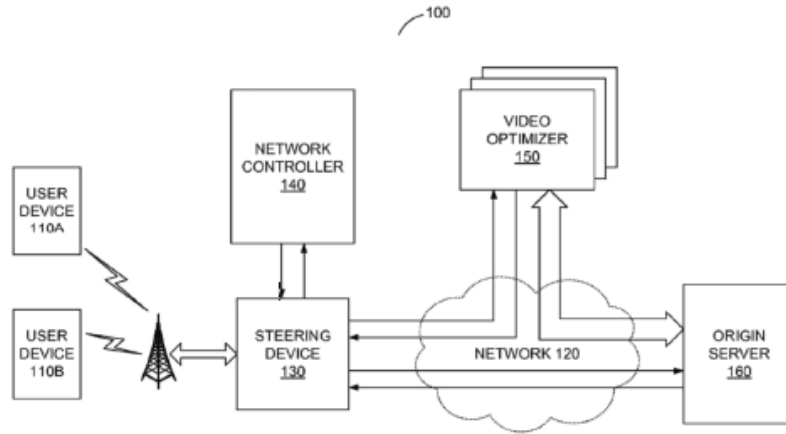


FIG. 2A

Shieh, Fig. 2A (annotated).

²⁷¹ Swenson, ¶[0059]-¶[0060]; *see id.*, ¶[0039] (explaining that the controller can incorporate “security functions”; Bhattacharjee, ¶216.

²⁷² Bhattacharjee, ¶216.



Swenson, Fig. 1.

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.²⁷³

Further, a POSITA would've 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.²⁷⁴ 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.²⁷⁵ Further, a POSITA would've known that security algorithms in a

²⁷³ Shieh, ¶[0017]-¶[0018]; Swenson, ¶[0023]-¶[0032].

²⁷⁴ Swenson, ¶[0059]-¶[0060], Figures 1, 4A-4B; Bhattacharjee, ¶218.

²⁷⁵ Bhattacharjee, ¶218.

security component are often updated via software updates as new threats are identified.²⁷⁶ A POSITA would've 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.²⁷⁷

Moreover, a POSITA would've had a reasonable expectation of success in modifying Shieh to implement the disclosure of Swenson given that they have a functionally similar architecture.²⁷⁸ If Swenson could be constructed to have the network nodes send packets to the central controller for inspection, a POSITA would've understood that the same arrangement could be implemented in Shieh.²⁷⁹

Element [1.1] is obvious in view of Shieh.²⁸⁰

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].²⁸¹

²⁷⁶ *Id.*

²⁷⁷ *Id.*

²⁷⁸ Shieh, ¶[0017]-¶[0018], ¶[0024]-¶[0025], ¶[0029]-¶[0030], Figures 1-3, 7; Swenson, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A; Bhattacharjee, ¶219.

²⁷⁹ Bhattacharjee, ¶219.

²⁸⁰ '111 Patent, 10:56-58.

²⁸¹ Shieh, ¶[0018], ¶[0025]-¶[0026], ¶[0029]; Bhattacharjee, ¶¶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.²⁸² 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.²⁸³ 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.”²⁸⁴

Shieh also discloses separate packet-applicable criterion to determine which packets should be forwarded to the security device pursuant to the filtering rules.²⁸⁵ For example, Shieh provides criterion specific to a particular packet being checked, such as the identification of “TCP FIN or TCP RST packets.”²⁸⁶ 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

²⁸² Bhattacharjee, ¶223.

²⁸³ Shieh, ¶[0025]; *see id.*, ¶[0018], ¶[0028]-¶[0029].

²⁸⁴ *Id.*, ¶[0017]-¶[0018]; *see id.*, ¶[0023], ¶[0028]-¶[0029].

²⁸⁵ Bhattacharjee, ¶224.

²⁸⁶ *Id.*, ¶[0036]; *see id.*, ¶[0035], ¶[0049].

POSITA would've understood that the packet-applicable criterion would be sent from a controller to one or more network access devices.²⁸⁷ This allows the same packet-applicable criterion to be applied at multiple network access devices (i.e., network nodes).²⁸⁸

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.²⁸⁹ Shieh therefore discloses Element [1.2] for substantially the same reasons that it discloses Element [1.1].²⁹⁰

Element [1.3] is obvious in view of Shieh.²⁹¹

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).²⁹² 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

²⁸⁷ Bhattacharjee, ¶¶224-225.

²⁸⁸ *Id.*

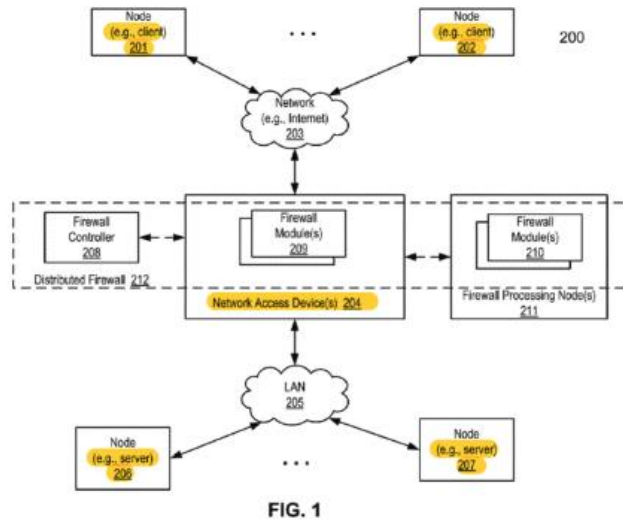
²⁸⁹ '111 Patent, 10:59-60.

²⁹⁰ Bhattacharjee, ¶¶226-232.

²⁹¹ '111 Patent, 10:60-62.

²⁹² Shieh, Claim 1; Bhattacharjee, ¶¶233-235; *see* Shieh, Abstract, ¶[0020], ¶[0027], ¶[0037], ¶[0049].

claimed first entity) and the destination (i.e., the claimed second entity).²⁹³ 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 207.²⁹⁴



Swenson, Fig. 1 (annotated).

Element [1.4] is obvious in view of Shieh.²⁹⁵

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

²⁹³ Shieh, ¶[0027].

²⁹⁴ *Id.*, Fig. 1; *see id.*, Figs. 2A-2B, 7.

²⁹⁵ ’111 Patent, 10:63-64.

packets,” which are checked by the network access device to determine whether a packet satisfies that criterion.²⁹⁶

Element [1.5] is obvious in view of Shieh.²⁹⁷

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.²⁹⁸ 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).²⁹⁹

Element [1.6] is obvious in view of Shieh.³⁰⁰

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.³⁰¹ Shieh checks whether a packet

²⁹⁶ Shieh, ¶[0036]; *see id.*, ¶[0035], ¶[0037], ¶[0049], Figure 7; Bhattacharjee, ¶¶236-238.

²⁹⁷ ’111 Patent, 10:65-67.

²⁹⁸ Shieh, ¶[0018], ¶[0023], ¶[0028]-¶[0029], ¶[0035], ¶[0049], Figure 7.

²⁹⁹ *Id.*, ¶[0036]; *see id.*, ¶[0037] (discussing whether packet should be sent to security processing device “based on a bypass flag”); Bhattacharjee, ¶¶239-242.

³⁰⁰ ’111 Patent, 11:1-4.

³⁰¹ Shieh, ¶[0018], ¶[0023], ¶[0028]-¶[0029], ¶[0035], ¶[0049], Figure 7; Bhattacharjee, ¶¶243-244.

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).³⁰²

C. 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).³⁰³

Claims 6 and 8-9 are obvious over Swenson, in view of the knowledge of a POSITA, for the reasons discussed above in Ground 1.³⁰⁴

D. Claim 12

Claim 12 is obvious over Shieh.³⁰⁵ Shieh repeatedly states that the analysis performed on the packets includes a security processing function.³⁰⁶ A POSITA

³⁰² Shieh, ¶[0036]; *see id.*, ¶[0037] (discussing whether packet should be sent to security processing device “based on a bypass flag”); Bhattacharjee, ¶¶244-245.

³⁰³ Bhattacharjee, ¶¶246-249, 253-255.

³⁰⁴ *Id.*, ¶¶250-252, 256-262.

³⁰⁵ ’111 Patent, 11:55-57.

³⁰⁶ Shieh, ¶[0002], ¶[0017]-¶[0019], ¶[0021], ¶[0023], ¶[0029]-¶[0031], ¶[0035]-¶[0037], ¶[0042], ¶[0049], Figures 1, 2B, 3, 7.

would've understood from this that the analysis includes the security processing function.³⁰⁷

E. Claim 13

Claim 13 is obvious over Shieh.³⁰⁸ Shieh discloses that its system analyzes packets by applying a security application that comprises a firewall functionality.³⁰⁹

F. Claim 14

Claim 14 is obvious over both Shieh and Swenson.³¹⁰ Shieh discloses the use of “deep packet inspection (DPI)” as part of its analysis functionality.³¹¹ Swenson also discloses that the flow analyzer in network controller 140 performs DPI.³¹²

³⁰⁷ Bhattacharjee, ¶¶263-266.

³⁰⁸ '111 Patent, 11:58-60.

³⁰⁹ Shieh, ¶[0021], ¶[0023], ¶[0027]-¶[0028], ¶[0038], ¶[0043]-¶[0049]; Bhattacharjee, ¶¶267-269.

³¹⁰ '111 Patent, 11:61-63.

³¹¹ Shieh, ¶[0021]; *see id.*, ¶[0028], ¶[0040]-¶[0041]; Bhattacharjee, ¶¶270-273.

³¹² Swenson, ¶[0059]-¶[0060], ¶[0065].

G. Claim 15

Claim 15 is obvious in view of both Shieh and Swenson.³¹³ As discussed above for Claim 14, Shieh discloses analysis of packets through DPI.³¹⁴ A POSITA would've understood that DPI refers to the inspection of at least part of the payload field.³¹⁵ Further, Swenson discloses Claim 15 for the reasons discussed above in Ground 1.

H. Claim 16

Claim 16 is obvious over Lin in view of the knowledge of a POSITA.³¹⁶ A POSITA would've known that a packet includes distinct header and payload fields.³¹⁷ Further, Shieh states that the packet streams passing through its system include “TCP FIN or TCP RST packets.”³¹⁸ A POSITA would've understood that a TCP FIN packet includes a FIN flag bit in its header that is set.³¹⁹ Further, a POSITA would've understood that a TCP RST packet includes a RST flag bit in its

³¹³ '111 Patent, 11:64-67.

³¹⁴ Shieh, ¶[0021], ¶[0028], ¶[0040]-¶[0041]; Lin, 3:11-12.

³¹⁵ Bhattacharjee, ¶¶274-278.

³¹⁶ '111 Patent, 12:1-4.

³¹⁷ EX1008, Section 2; Bhattacharjee, ¶¶279-280.

³¹⁸ Shieh, ¶[0036].

³¹⁹ Bhattacharjee, ¶281.

header that is set.³²⁰ A POSITA would've 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 1.³²¹

I. Claim 17

Claim 17 is obvious over Shieh in view of the knowledge of a POSITA.³²² Shieh states that the packet streams passing through its system include TCP packets.³²³ Further, the TCP FIN or TCP RST packets discussed in Shieh render Claim 17 obvious for the reasons discussed above with respect to Claim 16.³²⁴ In addition, a POSITA also would've 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.³²⁵

J. Claim 18

Claim 18 is obvious over Shieh in view of the knowledge of a POSITA.³²⁶ Shieh discusses the header fields in a packet and how they are used to route

³²⁰ *Id.*

³²¹ *Id.*

³²² '111 Patent, 12:5-9.

³²³ Shieh, ¶[0027], ¶[0036].

³²⁴ *Id.*, ¶[0036]; Bhattacharjee, ¶¶282-284.

³²⁵ *Id.*

³²⁶ '111 Patent, 12:10-16.

packets.³²⁷ 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.³²⁸

K. Claim 19

Claim 19 is obvious over Shieh for the reasons as those discussed above for Claim 18.³²⁹

L. Claim 20

Claim 20 is obvious over Shieh in view of the knowledge of a POSITA.³³⁰ Shieh discloses the transportation of TCP packets across a TCP network using TCP protocol.³³¹ 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...”³³² Shieh does not state that a TCP packet has a TCP sequence number or a TCP sequence mask field, but a POSITA would’ve known those fields

³²⁷ Lin, [0044].

³²⁸ Shieh, ¶[0031]; ¶[0027]; Bhattacharjee, ¶¶285-288.

³²⁹ ’111 Patent, 12:17-18; Shieh, ¶[0027]; ¶[0031]; Bhattacharjee, ¶289.

³³⁰ ’111 Patent, 12:20-27.

³³¹ Shieh, ¶[0027], ¶[0036]; Bhattacharjee, ¶¶290-292.

³³² Shieh, ¶[0027].

are always present in a TCP segment/packet.³³³ Further, Shieh teaches the use of a source TCP port or a destination TCP port as the packet applicable criterion.³³⁴

M. Claim 21

Claim 21 is obvious over Shieh.³³⁵ Shieh states that its packet network includes the Internet, a WAN or a LAN.³³⁶ Shieh also discloses that its system can be in communication with an ISP.³³⁷

N. Claim 22

Claim 22 is obvious over Shieh.³³⁸ 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.³³⁹

³³³ Bhattacharjee, ¶291.

³³⁴ Shieh, ¶[0027], ¶[0036]; Bhattacharjee, ¶291.

³³⁵ '111 Patent, 12:28-32.

³³⁶ Shieh, ¶[0020].

³³⁷ *Id.*, ¶[0053]; Bhattacharjee, ¶¶293-295.

³³⁸ '111 Patent, 12:33-36.

³³⁹ Shieh, Figure 1; *see id.*, ¶[0020], ¶[0022], ¶[0039], ¶[0053], ¶[0057], Figures 2A, 3; Bhattacharjee, ¶¶296-298.

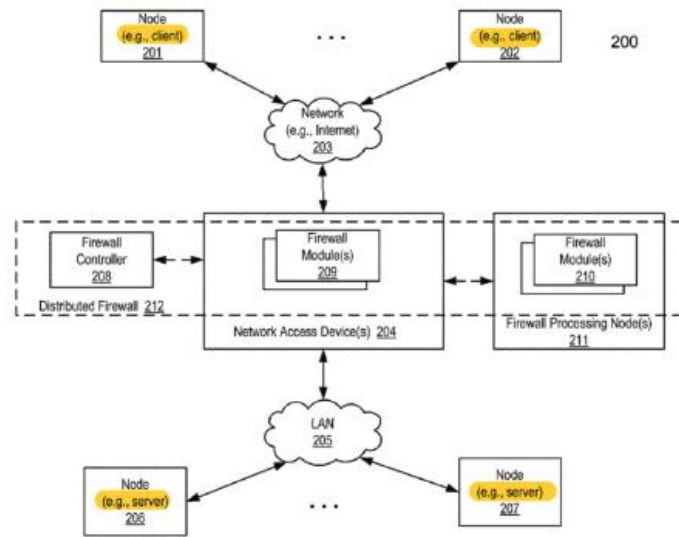


FIG. 1

Shieh, Fig. 1 (annotated).**O. Claim 23**

Claim 23 is obvious over Shieh.³⁴⁰ 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 1.³⁴¹ A POSITA would've known that the server can be a web server.³⁴² Further, Shieh discloses that “[a]ny of nodes 201-202 and 206-

³⁴⁰ '111 Patent, 12:37-41.

³⁴¹ Shieh, Figure 1; *see id.*, ¶[0020], ¶[0022], ¶[0039], ¶[0053], ¶[0057], Figures 2A, 3.

³⁴² Bhattacharjee, ¶¶299-300; Shieh, ¶[0053].

207 may be a client device (e.g., a desktop, laptop, Smartphone, gaming device) or a server.”³⁴³

P. Claim 24

Claim 24 is obvious in view of Shieh.³⁴⁴ Shieh discloses the use of the standard OpenFlow protocol for communication between the network access devices (i.e., the network node) and the controller.³⁴⁵

Q. Claim 27

Claim 27 is obvious in view of Shieh.³⁴⁶ 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...”³⁴⁷

³⁴³ Shieh, ¶[0020]; Bhattacharjee, ¶¶301-302.

³⁴⁴ ’111 Patent, 12:42-44.

³⁴⁵ Shieh, ¶[0025]-¶[0026]; Bhattacharjee, ¶303.

³⁴⁶ ’111 Patent, 12:51-52.

³⁴⁷ Shieh, ¶[0020]; *see id.*, ¶[0019], ¶[0025]-¶[0026], ¶[0032]; Bhattacharjee, ¶¶304-306.

R. Claim 28

Claim 28 is obvious in view of Shieh.³⁴⁸ Shieh discloses the communication of IP packets through IP networks.³⁴⁹

S. Claim 29

Claim 29 is obvious in view of Shieh.³⁵⁰ Shieh discloses the transportation of TCP packets across a TCP network using TCP protocol.³⁵¹

T. Claim 30

Claim 30 is obvious in view of Shieh.³⁵² Shieh states that its method can be applied to “subsequent packets of a particular session.”³⁵³ 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].³⁵⁴

³⁴⁸ '111 Patent, 12:53-55.

³⁴⁹ Shieh, ¶[0038]; *see id.*, ¶[0027] (discussing the use of IP addresses to identify IP packets being transmitted across an IP packet network), ¶[0031] (same); Bhattacharjee, ¶¶307-309.

³⁵⁰ '111 Patent, 12:56-58.

³⁵¹ Shieh, ¶[0027], ¶[0036]; Bhattacharjee, ¶¶310-311.

³⁵² '111 Patent, 12:59-63.

³⁵³ Shieh, ¶[0018]; *see id.*, ¶[0037]; Bhattacharjee, ¶¶312-314.

³⁵⁴ Bhattacharjee, ¶¶315-318.

VII. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

At this stage of these proceedings, Petitioner has no burden to identify and rebut objective indicia of non-obviousness.³⁵⁵ Patent Owner must first present a *prima facie* case for such consideration, which Petitioner should then have the chance to rebut on reply.³⁵⁶

VIII. CONCLUSION

For at least the above reasons, Juniper requests cancelation of claims 1-9, 12-24, and 27-31 of the '111 Patent.

Respectfully submitted,

/s/ Kyle K. Tsui

Kyle K. Tsui

Phone: 202.362.3527

Email: kyle.tsui@fischllp.com

Andrew L. Ramos (Reg. No. 76,865)

Phone: 202.362.3522

Email: andrew.ramos@fischllp.com

FISCH SIGLER LLP

5301 Wisconsin Avenue NW

Suite 400

Washington, DC 20015

³⁵⁵ *Id.*, ¶319-320.

³⁵⁶ *Sega of Am., Inc. v. Uniloc USA, Inc.*, IPR2014-01453, Paper 11 at 20 (P.T.A.B. Mar. 10, 2015).

CERTIFICATE OF COMPLIANCE

Per 37 C.F.R. §42.24(a) and (d), the undersigned hereby certifies that the
Petition complies with the type-volume limitation of 37 C.F.R. §42.24(a)(i)
because, exclusive of exempted portions, it contains 12,203 words as counted by
the word-processing program used to prepare it.

Dated: October 11, 2023

Respectfully submitted,

/s/ Kyle K. Tsui

Kyle K. Tsui (Reg. No. 62,602)

kyle.tsui@fischllp.com

FISCH SIGLER LLP

5301 Wisconsin Avenue NW

Suite 400

Washington, DC 20015

Phone: 202.362.3527

CERTIFICATE OF SERVICE

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

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

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

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

Dated: October 11, 2023

Respectfully submitted,

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