

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,

Petitioner,

- vs. -

ORCKIT IP, LLC,

Patent Owner.

Case IPR2023-00554

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

**DECLARATION OF SAMRAT BHATTACHARJEE, Ph.D.
IN SUPPORT OF PETITION FOR *INTER PARTES*
REVIEW OF U.S. PATENT NO. 10,652,111**

TABLE OF CONTENTS

I.	INTRODUCTION AND QUALIFICATIONS.....	1
II.	UNDERSTANDING OF THE GOVERNING LAW	4
	A. Invalidity by Obviousness.....	4
	B. Interpreting Claims Before the Patent Office	8
	C. Materials Relied on in Forming My Opinions.....	8
III.	BACKGROUND DISCUSSION COMPUTER NETWORKING TECHNOLOGY	9
IV.	OVERVIEW OF THE '111 Patent.....	14
	A. Specification of the '111 Patent.....	14
	B. The Claims of the '111 Patent.....	18
	C. The Prosecution History of the '111 Patent.....	19
	D. The Priority Date of the '111 Patent	25
V.	STATE OF THE ART PRIOR TO THE '111 Patent	25
	A. The Person of Ordinary Skill in the Art.....	25
	B. Lin	26
	C. Shieh.....	30
	A. Swenson	35
VI.	CLAIM CONSTRUCTION	40
VII.	GROUND OF INVALIDITY	44
	A. Ground 1: Claims 1-9, 12-24 and 27-31 of the '111 Patent are obvious over Lin in view of Swenson.	44

a. Claim 1	44
b. Claim 2	74
c. Claim 3	75
d. Claim 4	78
e. Claim 5	81
f. Claim 6	82
g. Claim 7	84
h. Claim 8	85
i. Claim 9	85
j. Claim 12	87
k. Claim 13	88
l. Claim 14	88
m. Claim 15	89
n. Claim 16	91
o. Claim 17	92
p. Claim 18	93
q. Claim 19	95
r. Claim 20	95
s. Claim 21	97
t. Claim 22	98
u. Claim 23	98
v. Claim 24	99
w. Claim 27	100

x. Claim 28.....	101
y. Claim 29.....	101
z. Claim 30.....	102
aa. Claim 31.....	103
B. Ground 2: Claims 1, 5-9, 12-24 and 27-30 of the '111 Patent are obvious over Shieh in view of Swenson.....	104
a. Claim 1.....	104
b. Claim 5.....	127
c. Claim 6.....	128
d. Claim 7.....	129
e. Claim 8.....	130
f. Claim 9.....	131
g. Claim 12.....	132
h. Claim 13.....	133
i. Claim 14.....	134
j. Claim 15.....	135
k. Claim 16.....	136
l. Claim 17.....	138
m. Claim 18.....	139
n. Claim 19.....	140
o. Claim 20.....	141
p. Claim 21.....	141
q. Claim 22.....	142

r. Claim 23.....	143
s. Claim 24.....	144
t. Claim 27.....	145
u. Claim 28.....	145
v. Claim 29.....	146
w. Claim 30.....	146
VIII. OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....	148

I, Samrat Bhattacharjee, Ph.D., hereby declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I have been retained on behalf of Cisco Systems, Inc. (“Cisco”) to provide my technical review, analysis, insights, and opinions concerning the validity of the claims of U.S. Patent No. 10,652,111 (EX1001; “the ’111 Patent”) entitled “Method and System for Deep Packet Inspection in Software Defined Networks.” The patent is assigned to Orckit IP, LLC (“Orckit”).

2. I am a tenured professor in the Computer Science Department and the Institute for Advanced Computer Studies at the University of Maryland. I am also an Affiliate Professor with the Department of Electrical and Computer Engineering at the University of Maryland. I have been an appointed faculty member at the University of Maryland since 1999.

3. I received a Ph.D. in computer science from the Georgia Institute of Technology in 1999. Before that I received a B.S. degree in Mathematics and Computer Science with Highest Distinction (Summa Cum Laude) and was recognized as the Outstanding Department Major at Georgia College and State University, Milledgeville, Georgia in 1994.

4. I have been involved in the research and development of computing systems for over two decades. I perform research and I teach in the areas of computer networks, distributed systems, computer security and operating systems, among

others. A major theme of my research has been the development of technology to improve the performance and security of data transfers on the Internet. My research has examined problems ranging from network and operating system support audio and video streaming, large scale Internet video distribution, the design of novel network architectures for better and secure content distribution, efficiency of content distribution over wireless networks, and so on.

5. For example, starting in the late 1990s, the focus of my research was the development of network and operating system technology to design a new network architecture that allowed for more efficient data transfer. Part of my Ph.D. research was to develop new and more efficient architectures for video delivery on the Internet, and I have published papers on this architecture during my graduate studies. I continued to work on video delivery as a faculty member, and have published various papers on video streaming, content delivery architectures, and on resilient large scale content delivery. During 2007, I was a visiting researcher at AT&T Labs, and one of the projects I focused on was a video content delivery platform. This work resulted in both publications and a granted US patent (US8,752,100 B2).

6. Along with working on research problems related to video distribution, I routinely taught this material in both graduate and undergraduate courses. A

number of my papers in this domain are with students at the University of Maryland, many of whom started in this area after taking my classes.

7. I have authored or co-authored over 100 articles in peer-reviewed journals, conference proceedings, and workshops in the aforementioned areas of computer science and others. Over the years I have received several fellowships, prizes, and awards for teaching excellence and technical papers, including an Alfred P. Sloan Jr. Fellowship in 2004 and the National Science Foundation CAREER award in 2001. I have also received an award from the SIGCOMM foundation for a “Test of Time” paper (that is given to an influential paper published ten years ago). My research work has been supported by multiple grants from the US National Science Foundation, and the Department of Defense. I have also started a Joint Ph.D. program with the University of Maryland and the Max Planck Society in Germany, and co-founded the annual Cornell, Maryland, Max Planck Research School that provides research exposure to about 80 students from across the world during a week-long school.

8. I have served on numerous paper and proposal review panels and participated on program committees for ACM/IEEE groups, technical journals, and conferences in Networking, Security and Systems in the aforementioned areas of computer science. I have created research software for all of my projects, much of which is available online for others to build on.

9. I am a named inventor on three U.S. Patents. These patents are generally related to computer networking and the delivery of services over networks and video delivery systems.

10. I have served as an expert witness and technical consultant in litigation and *Inter Partes* Review matters concerning videoconferencing, data conferencing and screen sharing, voice over IP (VoIP) telephony, multimedia networking, distributed systems, operating systems, computer networks, and datacenter networking, among others. I have testified in several trials and depositions as an expert witness.

11. I attach as Exhibit 1003 my curriculum vitae, which includes a more detailed list of my qualifications. My curriculum vitae also contains a list of all other cases in which, during the previous 4 years, I testified as an expert at trial or by deposition. My work on this case is being billed at a rate of \$700 per hour, with reimbursement for actual expenses. I have no direct financial interest in the dispute between the Cisco and Orckit, and my compensation is not contingent upon the outcome of this *Inter Partes* review.

II. UNDERSTANDING OF THE GOVERNING LAW

A. Invalidity by Obviousness

12. I understand that obviousness is analyzed from the perspective of a person of ordinary skill in the art (“POSA”) at the time of the alleged invention. I

also understand that a POSA is presumed to have been aware of all pertinent prior art at the time of the alleged invention.

13. I understand that an obviousness analysis involves comparing a claim to the prior art to determine whether the claimed invention as a whole would have been obvious to a POSA in view of the prior art, and in light of the general knowledge in the art at the time the invention was made. I also understand that the invention may be deemed obvious when a POSA would have reached the claimed invention through routine experimentation.

14. I understand that obviousness can be established by combining or modifying the disclosures of the prior art to achieve the claimed invention. It is also my understanding that where there is a reason to modify or combine the prior art to achieve the claimed invention, there must also be a reasonable expectation of success in so doing to render the claimed invention obvious. I understand that the reason to combine prior art references can come from a variety of sources, not just the prior art itself or the specific problem the patentee was trying to solve. I also understand that the references themselves need not provide a specific hint or suggestion of the alteration needed to arrive at the claimed invention; the analysis may include recourse to logic, judgment, and common sense available to a POSA.

15. I understand that when there is some recognized reason to solve a problem, and there are a finite number of identified, predictable solutions, a POSA

has good reason to pursue the known options within his or her technical grasp. If such an approach leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense. In such a circumstance, when a patent simply arranges old elements with each performing the same function it had been known to perform and yields no more than one would expect from such an arrangement, the combination is obvious.

16. I understand that when considering the obviousness of an invention, one should also consider whether there are any objective indicia that support the non-obviousness of the invention. I further understand that objective indicia of non-obviousness include failure of others, copying, unexpected results, information that “teaches away” from the claimed subject matter, perception in the industry, commercial success, and long-felt but unmet need. I also understand that in order for objective indicia of non-obviousness to be applicable, the indicia must have some sort of nexus to the subject matter in the claim that was not known in the art. I understand that this nexus includes a factual connection between the patentable subject matter of the claim and the objective indicia alleged. I also understand that an independently made invention that is made within a comparatively short period of time is evidence that the claimed invention was the product of ordinary skill.

17. Finally, I understand that patent examiners at the U.S. Patent and Trademark Office (“USPTO”) rely upon certain exemplary rationales in reviewing

patent applications to understand whether the subject matter of the claims is obvious.

I understand that the following is the list of exemplary rationales relied upon by patent examiners at the USPTO:

- (A) Combining prior art elements according to known methods to yield predictable results;
- (B) Simple substitution of one known element for another to obtain predictable results;
- (C) Use of a known technique to improve similar devices (methods, or products) in the same way;
- (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- (E) “Obvious to try” – Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- (F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; and
- (G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art

reference or to combine prior art reference teachings to arrive at the claimed invention.

B. Interpreting Claims Before the Patent Office

18. I understand that *Inter Partes* Review is a proceeding before the United States Patent & Trademark Office (“USPTO”) for evaluating the validity of issued patent claims. I understand that, in an *Inter Partes* Review, a claim term is interpreted in a manner consistent with the standard used in patent litigation, as set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). That standard generally construes the claims according to their “ordinary and customary” meaning in view of the claim language, specification, and file history, and where applicable, relevant other evidence.

19. I understand that a patent’s “specification” includes all the figures, discussion, and claims within the patent. I understand that the USPTO will look to the specification and prosecution history to see if there is a definition for a given claim term, and if not, will apply the ordinary and customary meaning from the perspective of a POSA at the time in which the alleged invention was made.

C. Materials Relied on in Forming My Opinions

20. In forming the opinions expressed in this declaration, I have relied on my own knowledge, experience, and expertise, as well as the knowledge of a POSA

in the relevant timeframe. In addition, I have reviewed and relied upon all documents referenced in this declaration and the following list of materials:

- U.S. Patent No. 10,652,111 (“the ’111 Patent”) (EX1001);
- Prosecution History of the ’111 Patent (EX1002);
- U.S. Patent No. 9,264,400 (“Lin”) (EX1005);
- U.S. Patent Application Publication No. 2013/0291088 (“Shieh”) (EX1006);
- U.S. Patent Application Publication No. 2013/0322242 (“Swenson”) (EX1007);
- RFC 2460, Internet Protocol, Version 6 (IPv6) Specification (EX1008); and
- Nunes, A., et al., A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks (EX1009).

III. BACKGROUND DISCUSSION COMPUTER NETWORKING TECHNOLOGY

21. To better understand my opinions on the validity of the claims of the ’111 Patent, I provide below some background on computer networking with a particular focus on software defined networking (“SDN”). All of the concepts discussed in this section were well known and used prior to the April 22, 2014 priority date for the ’111 Patent.

A. Software Defined Networks (“SDN”)

22. “Computer networks are typically built from a large number of network devices such as routers, switches and numerous types of middleboxes (i.e., devices

that manipulate traffic for purposes other than packet forwarding, such as a firewall) with many complex protocols implemented on them.” EX1009 at 1. “Network operators are responsible for configuring policies to respond to a wide range of network events and applications.” EX1009 at 1. As computer networks have evolved, “programmable networks” have been developed to help keep up with the pace of that evolution. EX1009 at 1.

23. One programmable networking concept that became popular before the priority date for the ’111 Patent was filed is SDN. EX1005, Col. 1:11-12 (“Software defined networking (SDN) is an emerging architecture for computer networking.”); see EX1009 at 1. SDN is a “networking paradigm in which the forwarding hardware [in a network] is decoupled from control decisions.” EX1009 at 1. “In SDN, the network intelligence is logically centralized in software-based controllers (the control plane), and network devices become simple packet forwarding devices (the data plane) that can be programmed via an open interface (e.g., ForCES, OpenFlow, etc.).” EX1009 at 1; see EX1005, Col. 1:12-14 (“Unlike traditional computer network architectures, SDN separates the control plane from the data plane.”).

24. SDN “was developed to facilitate innovation and enable simple programmatic control of the network data-path.” EX1009 at 3. As shown in the figure below, “the separation of the forwarding hardware from the control logic allows easier deployment of new protocols and applications, straightforward

network visualization and management, and consolidation of various middleboxes into software control.” EX1009 at 3; see EX1005, Col. 1:14-17 (stating that the separation of the control plane from the data plane “provides many advantages, including relatively fast experimentation and optimization of switching and routing policies.”). “Instead of enforcing policies and running protocols on a convolution of scattered devices, the network is reduced to ‘simple’ forwarding hardware and the decision-making network controllers.” EX1009 at 3.

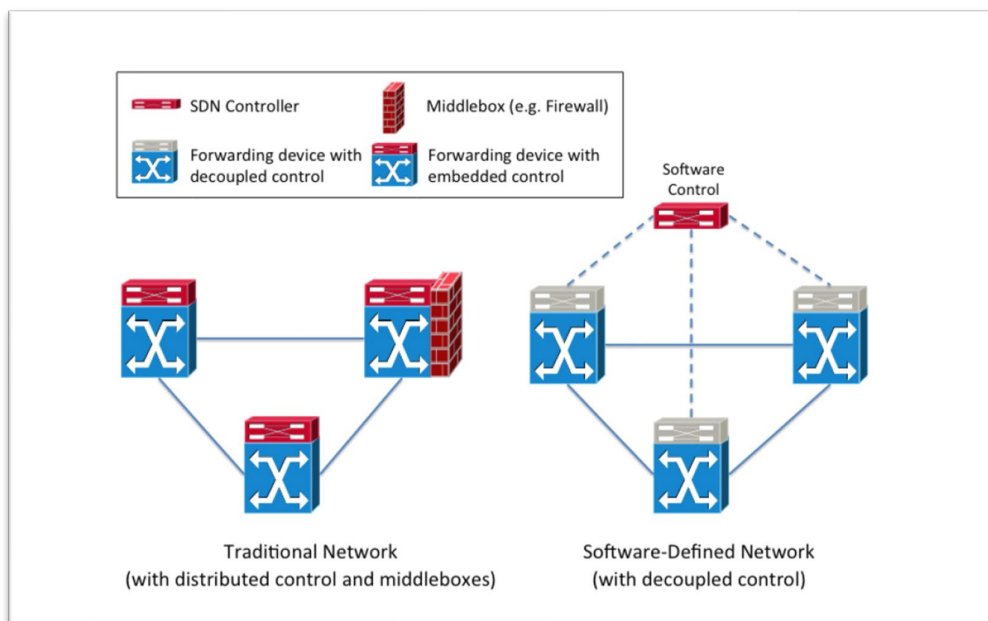


Figure 1 of EX1009

25. “The underlying network infrastructure may involve a number of different physical network equipment, or forwarding devices such as routers, switches, virtual switches, wireless access points, to name a few.” EX1009 at 5. “In a [SDN], such devices are often represented as basic forwarding hardware accessible

via an open interface at an abstraction layer, as the control logic and algorithms are off-loaded to a controller.” EX1009 at 5. “Such forwarding devices are commonly referred to, in SDN terminology, simply as ‘switches, as illustrated in [the figure reproduced below].” EX1009 at 5.

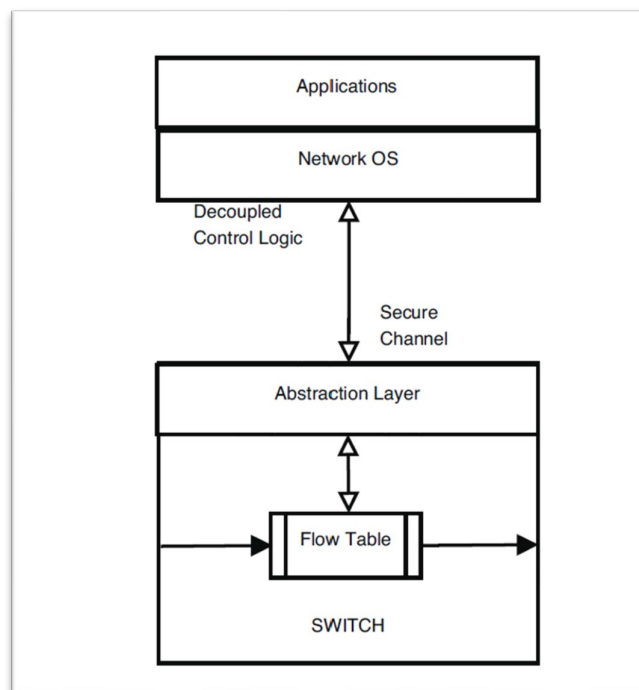


Figure 3 of EX1009

26. “Traditionally, the basic unit of networking has been the packet.” EX1009 at 7. “Each packet contains address information necessary for a network switch to make routing decisions.” EX1009 at 7. “However, most applications send data as a flow of many individual packets.” EX1009 at 7. “A network that wishes to provide [Quality of Service] or service guarantees to certain applications may benefit from individual flow-based control.” EX1009 at 7.

27. However, “[i]n a [SDN] network where network elements are controlled remotely, overhead is caused by traffic between the data-plane and the control-plane.” EX1009 at 7. “As such, using packet level granularity would incur additional delay as the controller would have to make a decision for each arriving packet.” EX1009 at 7-8. “When controlling individual flows, the decision made for the first packet of the flow can be applied to all subsequent packets of that flow.” EX1009 at 8. “The overhead may be further reduced by grouping flows together, such as all traffic between two hosts, and performing control decisions on the aggregated flows.” EX1009 at 8.

28. Among the functionalities that can be implemented when using SDN are “network security services, such as firewall or deep packet inspection (DPI).” EX1005, Col. 3:11-12; see EX1006, ¶[0021].

B. The OpenFlow™ Protocol

29. As discussed above, the OpenFlow™ protocol can be used to implement SDN. EX1009 at 1; see EX1005, Col. 1:18-32. “The OpenFlow™ protocol is an open protocol for remotely controlling forwarding tables of network switches that are enabled for SDN.” EX1005, Col. 1:18-20. “Generally speaking, the OpenFlow protocol allows direct access to and manipulation of the forwarding plane of network devices, such as switches and routers.” EX1005, Col. 1:20-22. “A control plane of an OpenFlow™ protocol-compliant computer network (also referred to as

an ‘OpenFlow™ controller’) may communicate with OpenFlow™ switches (i.e., network switches that are compliant with the OpenFlow™ protocol) to set flow policies that specify how the switches should manipulate packets of network traffic.” EX1005, Col. 1:22-28. “Example packet manipulation actions include forwarding a packet to a specific port, modifying one or more fields of the packet, asking the controller for action to perform on the packet, or dropping the packet.” EX1005, Col. 1:28-32.

IV. OVERVIEW OF THE ’111 PATENT

A. Specification of the ’111 Patent

30. The methods and systems described in the ’111 Patent relate to “deep packet inspection (DPI) in a software defined network (SDN).” EX1001, Abstract; see EX1001, Col. 1:14-16. The method disclosed in the ’111 Patent “is performed by a central controller of the SDN.” EX1001, Col. 2:27-30. The method comprises “configuring a plurality of network nodes operable in the SDN with at least one probe instruction.” EX1001, Col. 2:3-32. The method further comprises “receiving from a network node a first packet of a flow, wherein the first packet matches the at least one probe instruction” and “wherein the first packet includes a first sequence number.” EX1001, Col. 2:32-35. The method also comprises “receiving from a network node a second packet of the flow, wherein the second packet matches the at least one probe instruction” and “the second packet includes a second sequence

number. EX1001, Col. 2:35-38. “[T]he second packet is a response of the first packet.” EX1001, Col. 2:38-39. The method further includes “computing a mask value respective of at least the first and second sequence numbers, wherein the mask value indicates which bytes to be mirrored from subsequent packets belonging to the same flow.” EX1001, Col. 2:39-44. The method also includes “generating at least one mirror instruction based on at least the mask value.” EX1001, Col. 2:45-46. The method further includes “configuring the plurality of network nodes with at least one mirror instruction.” EX1001, Col. 2:46-47. Finally, “the mirrored bytes are inspected.” EX1001, Col. 2:44.

31. “[T]he central controller 111 [shown below in Figure 1 of the ’111 Patent] is configured to perform deep packet inspection on designated packets from designated flows or TCP sessions.” EX1001, Col. 4:5-7. “To this end, the central controller 111 is further configured to instruct each of the network nodes 112 which of the packets and/or sessions should be directed to the controller 111 for packet inspections.” EX1001, Col. 4:8-11. “The determination [of whether a packet requires inspection] is performed based on a set of instructions provided by the controller 111.” EX1001, Col. 4:14-15. “A packet that requires inspection is either redirected to the controller 111 or mirrored and a copy thereof is sent to the controller 111.” EX1001, Col. 4:15-18. “It should be noted that traffic flows that are

inspected are not affected by the operation of the network node 112.” EX1001, Col. 4:18-19.

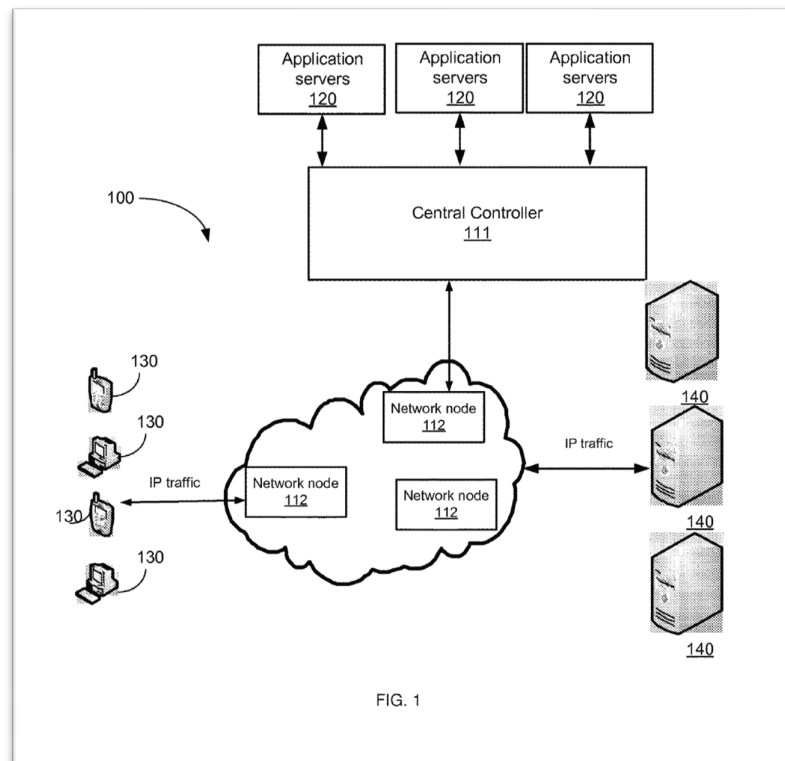


Figure 1 of the '111 Patent

32. F[igure] 6 [of the '111 Patent, shown below] shows an exemplary and non-limiting flowchart 600 illustrating the operation of the central controller 111 according to one embodiment.” EX1001, Col. 9:40-42. “At S[tep] 610, all network nodes 112 are configured with a set of probe instructions utilized to instruct each node 112 to redirect a TCP packet having at least a flag value as designated in each probe instruction.” EX1001, Col. 9:42-45. “At S[tep] 620, a first TCP packet with at least one TCP FLAG SYN value equal to 1 is received.” EX1001,

Col. 9:47-48. “This packet may have a sequence number M and may be sent from a client device 130.” EX1001, Col. 9:48-50. “At S630, a second TCP packet with at least one TCP FLAG ACK value equal to 1 is received.” EX1001, Col. 9:50-51. “This packet may have a sequence number N and may be sent from a destination server 140 in response to the first TCP packet.” EX1001, Col. 9:51-53. “In an embodiment, the flow table is updated with the respective flow ID and the state of the first and second packets.” EX1001, Col. 9:53-55.

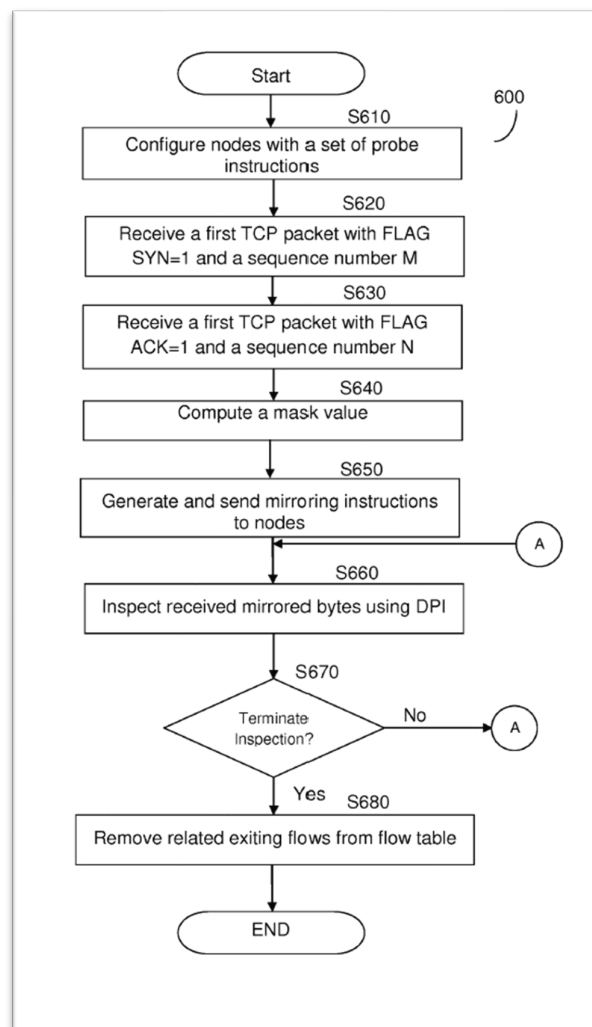


Figure 6 of the '111 Patent

33. “At S[tep] 640, using at least the sequence numbers of the first and second packets a mask value is computed.” EX1001, Col. 9:56-57. “The mask value is utilized to determine which bytes from the flow respective of the sequence numbers N and M should be mirrored by the nodes.” EX1001, Col. 9:40-42. “At S[tep] 650, a set of mirroring instructions are generated using the mirror value and sent to the network nodes.” EX1001, Col. 9:62-63. “Each such instruction defines the packets (designed at least by a specific source/destination IP addresses, and TCP sequences), the number of bytes, and the bytes that should be mirrored.” EX1001, Col. 9:63-67. At S[tep] 660, the received mirror bytes are inspected using a DPI engine in the controller 111.” EX1001, Cols. 9:67-10:1. “In addition, the flow table is updated with the number of the received mirror bytes.” EX1001, Col. 10:1-3.

B. The Claims of the '111 Patent

34. The '111 Patent includes 54 claims. EX1001, Cols. 10:52-14:60. Claims 1 and 32 are independent claims. EX1001, Cols. 10:52-11:4, 13:8-23.

Representative Claim 1 recites:

A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method comprising:

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

receiving, by the network node from the controller, the instruction and the criterion; receiving, by the network node from the first entity over the packet network, a packet addressed to the second entity;

checking, by the network node, if the packet satisfies the criterion;

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

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.

EX1001, Cols. 10:52-11:4.

C. The Prosecution History of the '111 Patent

35. The application that resulted in the '111 Patent was U.S. Patent Application No. 15/126,288 (“the '288 Application”). EX1001. The '288 Application was filed, originally as an application under the Patent Cooperation Treaty, on April 21, 2015. EX1001; EX1002 at 1. The '288 Application claims priority to U.S. provisional application No. 61/982,358, which was filed on April 22, 2014. EX1001. I understand that this is the date to which the '288 Application claims its earliest priority.

36. On September 15, 2016, the Applicant filed a Preliminary Amendment cancelling original Claims 1-19 of the '288 Application and adding new Claims 20-73. EX1002 at 53-64.

37. The USPTO issued a Non-Final Rejection of the '288 Application on September 12, 2018, which rejected Claims 20-73. EX1002 at 272-295. The Non-

Final Rejection rejected Claims 20-73 as obvious under 35 U.S.C. § 103 over U.S. Patent Application Publication No. 2013/0291088 (“Dolganow”) in view of U.S. Patent Application Publication No. 2014/0052836 (“Nguyen”). EX1002 at 275-293.

38. The Applicant filed an Amendment in response on December 10, 2018. EX1002 at 322-330. The Applicant did not amend the pending claims. Instead, with respect to the obviousness rejections, the Applicant argued “that the Dolganow and Nguyen references are directed toward respectively different purposes and are based on respectively different structures, and thus are not analogous to one another and cannot logically be combined.” EX1002 at 323. For example, the Applicant stated, “The Dolganow reference involves DPI, where packets are handled based on their content, while the Nguyen reference involves SDN, where packets are handled based on externally received instructions.” EX1002 at 323. The Applicant further noted that Dolganow and Nguyen were in different UPSTO classifications. EX1002 at 324-325. The Applicant also argued that the Examiner provided no reason why Dolganow would have been modified to incorporate the functionality of Nguyen. EX1002 at 325-328.

39. With respect to the claim language, the Applicant argued that Dolganow fails to disclose “... 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.”

EX1002 at 328. The Applicant stated that Dolganow “fails to disclose any controller in general or any ‘an instruction and a packet-applicable criterion.’” EX1002 at 328.

40. The USPTO issued a Final Rejection of the ’288 Application on April 15, 2019, which again rejected pending Claims 20-73. EX 1002 at 331-357. The Final Rejection maintained the rejection of Claims 20-73 as obvious under 35 U.S.C. 103 over Dolganow in view of Nguyen. EX 1002 at 333-355. The Examiner addressed the Applicant’s arguments in the December 10, 2018 Amendment and disagreed with them. EX 1002 at 333-336. For example, the Examiner stated that “the combination of Dolganow and Nguyen allows for efficient deep packet inspection of traffic in cloud-based networks utilizing software-defined networks (SDN).” EX 1002 at 334. Further, the Examiner found a motivation to combine the references “in Nguyen’s reference, such that to build a SDN application that provides all of the desired SDN functionality in a system in order to implement a new SDN functionality. Such an advantage would include a flow-based handler receiving a packet, and determining that the packet is associated with a flow session and associating the rules with the packet.” EX 1002 at 334-335.

41. The Applicant filed an Amendment in response to the Final Rejection on May 7, 2019. EX1002 at 397-417. The Applicant amended independent Claims 20 and 51 to state that the claimed controller “is external to the network node” and that a packet satisfying the claimed criterion is sent “to an entity that is included in

the instruction and is other than the second entity.” EX1002 at 398, 404. The Applicant relied on these new limitations to distinguish the claims from the Dolganow and Nguyen references. EX1002 at 408-416. The Applicant also repeated its earlier arguments that Dolganow and Nguyen do not disclose “... 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.” EX1002 at 414. The Applicant stated that these limitations had not been properly addressed by the Examiner. EX1002 at 414-415. Further, the Applicant also argued that “Claim 20 clearly recites a single packet network” but “the Dolganow reference teaches two different networks, namely 120 and 140, connected via the bode 130a.” EX1002 at 414 (emphasis in original).

42. In addition, the Applicant again argued that there was not proper motivation to combine Dolganow and Nguyen. EX1002 at 408-413. The Applicant stated both that Dolganow and Nguyen were not analogous art and were not pertinent to the same problem. EX1002 at 408-410. The bases for the Applicant’s arguments were similar to those stated in the previous amendment. EX1002 at 408-413.

43. The USPTO issued another Non-Final Rejection of the ’288 Application on July 8, 2019, which rejected pending Claims 20-73. EX 1002 at 426-450. The Non-Final Rejection rejected Claims 20-73 as obvious under 35 U.S.C. §

103 over Dolganow in view of U.S. Patent Application Publication No. 2016/0219080 (“Huang”). EX1002 at 429-449.

44. The Applicant filed an Amendment in response to the Non-Final Rejection on October 2, 2019. EX1002 at 492-501. The Applicant did not amend the claims. Instead, the Applicant argued that the Examiner had not provided a motivation to combine the Dolganow and Huang references. EX1002 at 493-498. The Applicant noted that the Examiner stated that the motivation was “... to realize data collection and transmission of a control policy for one-end-to-multiple ends.” EX1002 at 493. The Applicant argued that this motivation was not proper, at least because Dolganow was for peer-to-peer systems, not one-end-to-multiple-ends systems. EX1002 at 493-494. The Applicant also argued “that the Dolganow and Huang references are directed toward respectively different purposes and are based on respectively different structures, and thus are not analogous to one another and cannot logically be combined.” EX1002 at 496. For example, the Applicant stated, “The Dolganow reference involves DPI, where packets are handled based on their content, while the Huang reference involves SDN, where packets are handled based on externally received instructions.” EX1002 at 496. The Applicant further noted that Dolganow and Huang were in different UPSTO classifications. EX1002 at 497-498.

45. With respect to the claim language, the Applicant argued that Dolganow fails to disclose “... 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.” EX1002 at 499. The Applicant stated that Dolganow “fails to disclose any controller in general or any ‘an instruction and a packet-applicable criterion.’” EX1002 at 499. The Applicant also stated that Dolganow fails to disclosed that the claimed controller “is external to the network node.” EX1002 at 500. Further, the Applicant also argued that “Claim 20 clearly recites a single packet network” but “the Dolganow reference teaches two different networks, namely 120 and 140, connected via the bode 130a.” (emphasis in original). EX1002 at 498.

46. On January 7, 2020, the USPTO issued a Notice of Allowance for the '288 Application. EX1002 at 504-510. The Notice of Allowance stated that Claims 20-73 of the '288 Application were allowed. EX1002 at 508. The Notice of Allowance stated the reasons for allowance as follows: “Claims 20-73 are allowed for the reasons as set forth in Applicant’s amendment and remarks/argument filed 10/02/2019.” EX1002 at 509. The '288 Application issued as the '111 Patent on May 12, 2020. EX 1001.

D. The Priority Date of the '111 Patent

47. I understand that the earliest-filed patent application to which the '111 Patent claims priority (U.S. Provisional Patent Application No. 61/982,358) was filed on April 22, 2014. EX1001. I have therefore been asked for purposes of this declaration to analyze obviousness as of April 22, 2014.

V. STATE OF THE ART PRIOR TO THE '111 PATENT

A. The Person of Ordinary Skill in the Art

48. As discussed above, the claims of the '111 Patent are directed to a method and system for SDN. EX1001, Abstract, Col 1:14-16. Based on this, it is my opinion that, as of April 22, 2014, a POSA would have had a working knowledge of hardware and software for packet-switched networking. A POSA also would have had a bachelor's degree in computer science, computer engineering, or an equivalent, and two years of professional experience. Further, it is my opinion that a person with additional education but less professional experience may still qualify as a POSA, and a person with additional professional experience but less education may still qualify as a POSA.

49. I believe that I would qualify as understanding the knowledge and skills of a POSA as of April 22, 2014, and that I have a sufficient level of knowledge, experience, and expertise to provide an expert opinion in the field of the '111 Patent.

B. Lin

50. Lin “relates generally to computer security, and more particularly but not exclusively to software defined networking.” EX1005, Col. 1:7-9; see EX1005, Abstract. “In one embodiment, a software defined networking (SDN) computer network includes an SDN controller and an SDN switch.” EX1005, Col. 1:58-60. “The SDN controller inserts flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.” EX1005, Col. 1:60-62; see EX1005, Cols. 4:8-31, 6:1-12. “A broadcast function of the SDN switch to the ports that form the SDN pipe may be disabled.” EX1005, Col. 1:62-64; see EX1005, Col. 4:53-65. “The SDN pipe allows outgoing packets sent by the sender component to be received by the security component.” EX1005, Col. 1:64-65; see EX1005, Cols. 3:25-31, 6:40-48. “The security component inspects the outgoing packets for compliance with security policies and allows the outgoing packets to be forwarded to their destination when the outgoing packets pass inspection.” EX1005, Cols. 1:66-2:2; see EX1005, Cols. 3:31-33, 6:48-63. “The SDN controller may also insert a flow rule in the flow table of the SDN switch to bypass inspection of specified packets.” EX1005, Col. 2:2-4; see EX1005, Cols. 7:24-8:18.

51. Figure 6 of Lin, reproduced below, shows “a schematic diagram of an SDN computer network 600 in accordance with an embodiment of the present invention.” EX1005, Col. 3:40-42. “In one embodiment, the SDN computer

network 600 is compliant with the OpenFlow™ protocol.” EX1005, Col. 3:42-44.

“The SDN controller 610 provides a logically centralized framework for controlling the behavior of the SDN computer network 600.” EX1005, Col. 4:7-9. “This is in marked contrast to traditional computer networks where the behavior of the computer network is controlled by low-level device configurations of switches and other network devices.” EX1005, Col. 4:9-12.

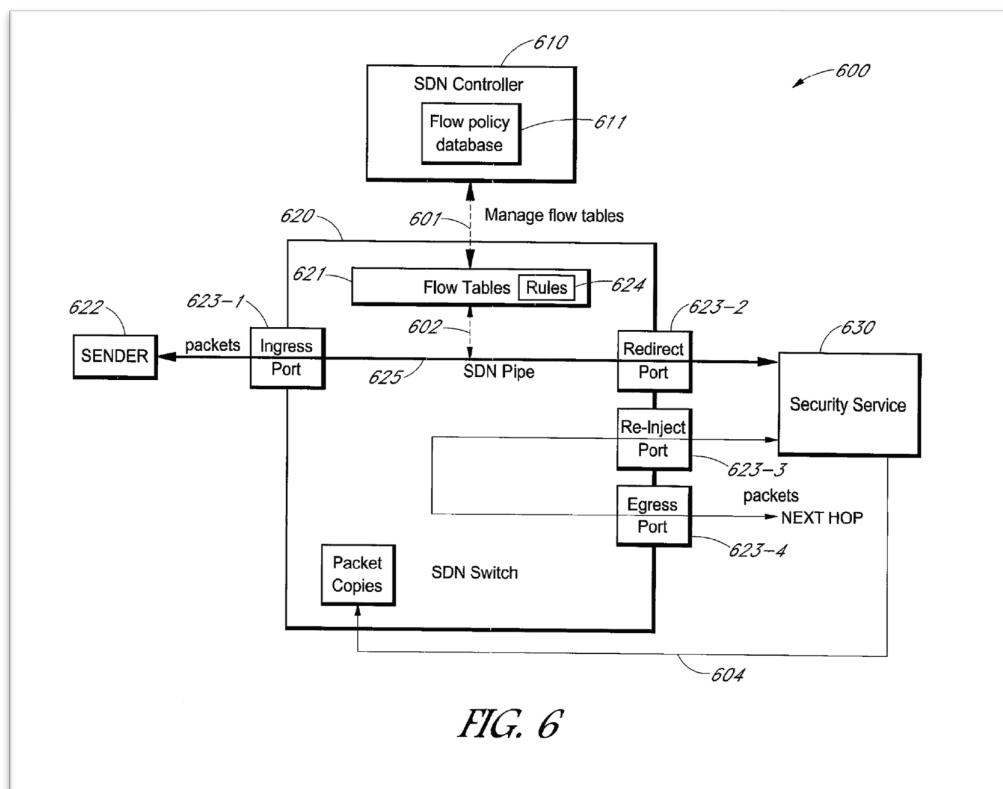


Figure 6 of Lin

52. “The SDN controller 610 may include a flow policy database 611.” EX1005, Col. 4:12-13. “The flow policy database 611 may comprise flow policies

that are enforced by the controller 610 on network traffic transmitted over the SDN computer network 600.” EX1005, Col. 4:13-16. “The flow policies may specify security policies that govern transmission of packets over the SDN computer network 600.” EX1005, Col. 4:16-18. “The flow policies may be enforced in terms of flow rules (labeled as 624) that are stored in the flow tables 621 of the SDN switch 620.” EX1005, Col. 4:18-20.

53. “The 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.” EX1005, Col. 6:1-4. “The SDN pipe allows outgoing packets sent by the sender component 622 or incoming packets going to the sender component 622 to be redirected to the security service 630 for inspection before the packets are sent out of the SDN switch 620.” EX1005, Col. 6:4-8; see EX1005, Col. 4:20-26. “In one embodiment, the SDN pipe is created by creating a first flow rule that forwards packets received in the ingress port 623-1 to the redirect port 623-2, and a second flow rule that forwards packets received in the redirect port 623-2 to the ingress port 623-1.” EX1005, Col. 6:8-12.

54. “After the redirect flow rules for creating the SDN pipe are inserted in the flow tables 621, any packet received by the SDN switch 620 in the ingress port 623-1 will be identified as to be forwarded to the redirect port 623-2, and any packet received by the SDN switch 620 in the redirect port 623-2 will be identified

as to be forwarded to the ingress port 623-1 (see arrow 602).” EX1005, Col. 6:40-46; see EX006, Col. 4:53-65. “This allows the security service 630 to receive from the redirect port 623-2 all outgoing packets sent by the sender component 622 to the ingress port 623-1.” EX1005, Col. 6:46-48. “The security service 630 may inspect the outgoing packets for compliance with security policies.” EX1005, Col. 6:48-50. “The security service 630 may drop, or perform other security response, to packets that do not pass inspection (e.g., packets that do not meet firewall policies, packets containing prohibited payload, packets with malicious content, etc.).” EX1005, Col. 6:50-54. “The security service 630 may forward those packets that pass inspection toward their destination by re-injecting the packets back into the SDN switch 620 by way of the re-inject port 623-3.” EX1005, Col. 6:54-57. “Once back in the SDN switch 620 by way of the re-inject port 623-3, the flow rules that govern packets received in the ingress port 623-1 and the redirect port 623-2 no longer apply.” EX1005, Col. 6:57-60. “Accordingly, the re-injected packets are forwarded to the egress port 623-4 (or some other port) toward the next hop in accordance with the L2 switching logic of the SDN computer network 600.” EX1005, Col. 6:60-63.

55. “Re-injecting packets that pass inspection consume bandwidth, as the packets will have to be transmitted by the security service 630 to the re-inject port 623-3.” EX1005, Col. 7:9-11. “For optimization, the SDN switch 620 may be configured to copy packets that are redirected to the security service 630 for

inspection.” EX1005, Col. 7:11-13. “This way, the security service 630 simply has to inform the SDN switch 620 an action to take on packets based on the result of the inspection (see arrow 604).” EX1005, Col. 7:13-16. “For example, the security service 630 may send an index identifying the packets and an action on how to manipulate the packets.” EX1005, Col. 7:16-18. “The action may instruct the SDN switch 620 to drop the copied packets, forward the copied packets to their destinations, quarantine the copied packets, etc.” EX1005, Col. 7:18-21.

56. Lin also discloses that its system may have “bypass flow rules may instruct the SDN switch to forward particular packets directly from one port to another port of the SDN switch without being redirected to a security component, such as a security service, for inspection.” EX1005, Col. 9:12-16; see EX1005, Cols. 7:24-8:18.

C. Shieh

57. Shieh relates “generally to network security” and discloses a “network system [that] includes a security device and a network access device.” EX1006, ¶[0002]. “The network access device is to receive a packet from a source node destined to a destination node, and to examine a data structure maintained by the network access device to determine whether the data structure stores a data member having a predetermined value, the data member indicating whether the packet should undergo security processing.” EX1006, Abstract; see EX1006, ¶[0002]. “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.” EX1006, Abstract. “[I]n response to a response received from the security device, the packet is routed to the destination node dependent upon the response.” EX1006, Abstract.

58. “In one embodiment, the mechanism includes two functions: 1) an input/output (IO) function that performs the distribution of network traffic; and 2) a security-processing function that performs security processing, including security inspection and policy enforcement.” EX1006, ¶[0017]. “The IO function receives the packets and uses a session table to forward the packets to the security-processing function.” EX1006, ¶[0017]. “A session table is a data structure that stores connection states, including the destination of the security-processing function.” EX1006, ¶[0017]. “In one embodiment, the IO function determines, based on an internal data structure such as a session or flow table, whether the packet should be forwarded to the security processing function for security inspection.” EX1006, ¶[0017]. “The configuration of the IO function to control whether to forward the packets to the security processing function can be set based on a command received from an administrator or alternatively, based on a signal received from the security processing function.” EX1006, ¶[0017].

59. “[A]n administrator can configure, for example, via a controller or a management entity, a network access device to set up a set of filtering rules specifying whether and/or what types of packets should be forwarded to a security device and which of the security devices for security inspection.” EX1006, ¶[0018]; see EX1006, ¶[0023], ¶[0029]. “[T]he controller is configured to manage multiple network access devices and/or multiple security devices.” EX1006, ¶[0018]. “Alternatively, a security device may inform a network access device that subsequent packets of a particular session should be forwarded from the network access device for security inspection.” EX1006, ¶[0018]; see EX1006, ¶[0023]. “[A] security device performs the security inspection at the beginning of the flow or session, and at a certain point, the security device decides that it no longer needs to inspect further packets of the same session.” EX1006, ¶[0018]; see EX1006, ¶[0023].

60. “The I/O function does a couple of operations when it receives the packets.” EX1006, ¶[0033]. “The first is to look up the connection to which the packets belong to.” EX1006, ¶[0033]. “It performs a lookup into the session table (e.g., table 251B) to determine if it has previously processed any packets of the connection.” EX1006, ¶[0033]. “If the connection state has been in the session table, it can determine where the security-processing function is then forward the packet.” EX1006, ¶[0033]. “If it cannot find the connection in the session table, it can either

forward to any known security-processing function, or another network devices to determine to which security-processing function to forward the packets.” EX1006, ¶[0033].

61. “For example, device 272 may look up in its session table such as the one as shown in F[igure] 5 [shown below] to determine whether a bypass flag has been set to a predetermined value.” EX1006, ¶[0037]. “If the bypass flag matches the predetermined value, the packet is forwarded to security device 273 via path 282; otherwise, the packet is routed to destination node 274.” EX1006, ¶[0037].

Session/Flow ID	Address(es) of Security Processing Module(s)	Bypass Flag 501	Notification Event(s) 502	Filtering Rule(s) 503
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

FIG. 5

Figure 5 of Shieh

“Alternatively, if there is no entry in the session table corresponding to the current session, the packet will also be transmitted to security device 273.” EX1006,

¶[0037]. “After network device 272 receives a response from security device 273 via path 283, dependent upon the response, the packet may then be routed to destination node 274 via path 284.” EX1006, ¶[0037]. “These processes may continue until a notification is received from security device 273 via path 285 indicating that it no longer wishes to receive further packets of the same session for inspection, such that subsequent packets will be directly routed to destination node 274 via path 286 without routing to security device 273.” EX1006, ¶[0037]. “If there are certain events that have been registered from security device 273, network device 272 may notify security device 274 via path 287 upon detecting the registered events.” EX1006, ¶[0037].

62. “Referring to F[igure] 7 [shown below], at block 701, a network access device receives a packet from a source node destined to a destination node.” EX1006, ¶[0049]. “At block 702, the network access device determines whether the packet should be forwarded to a security device for security inspection.” EX1006, ¶[0049]. “For example, processing logic may check whether there is an entry exists in a session table for the current session.” EX1006, ¶[0049]. “If not, it may forward the packet to the security device for security processing at block 704.” EX1006, ¶[0049]. “Alternatively, the processing logic may check whether there is a bypass flag set to a predetermined value for the current session.” EX1006, ¶[0049]. “If there

is, the packet will not be forwarded to the security device; instead, the packet will be directly routed to the destination node at block 703.” EX1006, ¶[0049].

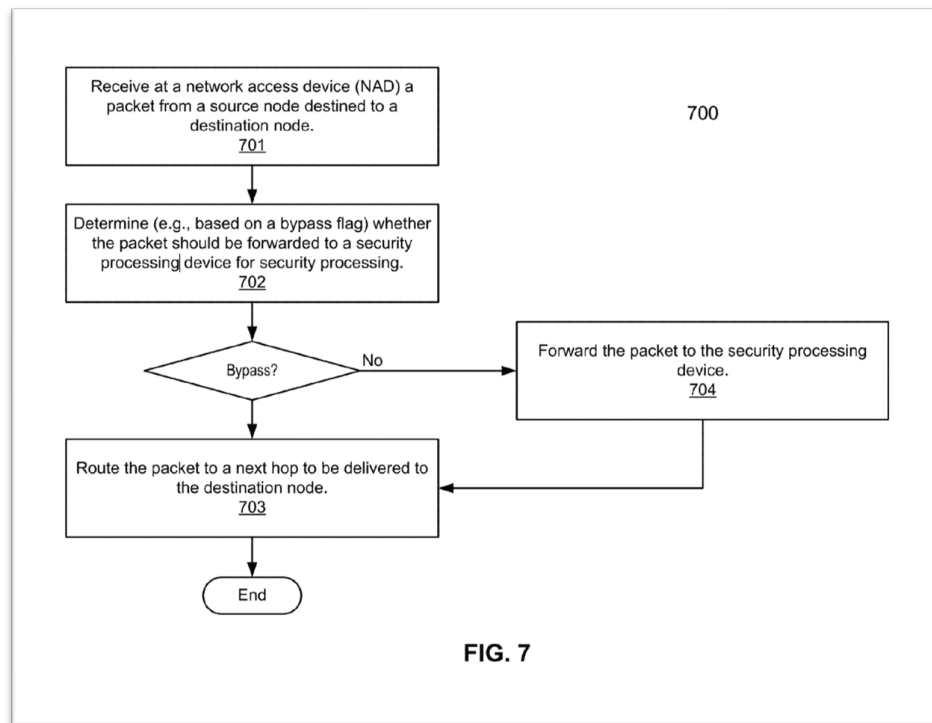


FIG. 7

Figure 7 of Shieh

A. Swenson

63. Swenson discloses a system and method for “selectively monitoring traffic in a service provider network.” EX1007, Abstract. “The system receives a notice for a beginning of a network data flow which responds to a request from a user device for content at an origin server.” EX1007, Abstract. “The system then determines whether to monitor the data flow from the origin server to the user device.” EX1007, Abstract. “If so determined, the system collects statistic

information of the data flow and stores the statistic information to a flow record in a database.” EX1007, Abstract. Swenson discloses that its system and method can be used for “real-time data gathering on the state of existing network traffic flows and mapping flow data to respective users in the network to predict available bandwidth and level of congestion.” EX1007, ¶[0018]; see EX1007, ¶¶[0019]-[0022].

64. Figure 1 of Swenson (shown below) “illustrates a high level block diagram of an example communication environment 100 for selective on-demand real-time network monitoring and subscriber identification.” EX1007, ¶[0023]. “The environment 100 comprises user devices 110, an origin server 160, a steering device 130, a network controller 140, a video optimizer 150, and a network 120.” EX1007, ¶[0023]. “The network 120 is a communication network that transmits data between the user devices 110, the steering devices 130 and the origin server 160 and/or the video optimizer 150.” EX1007, ¶[0023]. “In one embodiment the network 120 includes wireless network and the Internet.” EX1007, ¶[0023].

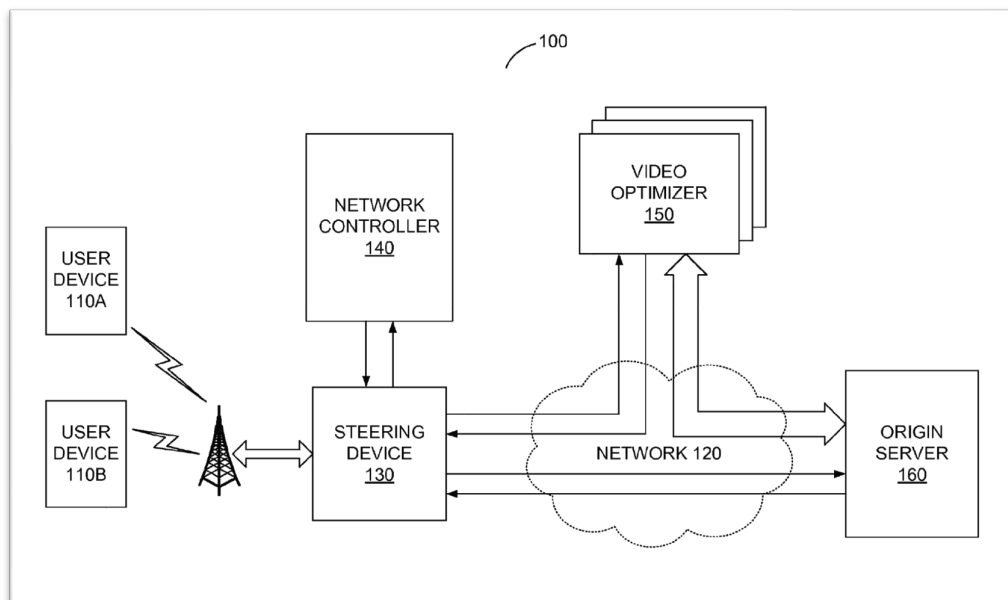


Figure 1 of Swenson

65. The steering device 130 may be a load balancer or a router located between the user device 110 and the network 120.” EX1007, ¶[0026]. “The steering device 130 provides the user device 110 with access to the network and thus, provides the gateway through which the user device traffic flows onto the network and vice versa.” EX1007, ¶[0026]. “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.” EX1007, ¶[0026]. “Alternatively, the network controller 140 interfaces with the steering device 130 to coordinate the monitoring and characterization of network traffic, such as identifying large and small objects in HTTP traffic flows.” EX1007, ¶[0026]. “In this case, the steering device 130 receives instructions from the network controller 140 based on the

desired criteria for characterizing flows of interest for further inspection.” EX1007, ¶[0026].

66. Figure 4A of Swenson (shown below) “illustrates one embodiment of an example working mode of the network controller for providing selective on-demand real-time network monitoring and subscriber identification.” EX1007, ¶[0055]. In this embodiment, “network traffic flows from the user device 110 through the steering device 130 and arrive at the origin server 160 over the network request path.” EX1007, ¶[0058]. “For example, a browser on the user device 110 may request web content from the origin server 160.” EX1007, ¶[0058]. “A HTTP request message initiated at the user device 110 is forwarded to the steering device 130 over the network link 411.” EX1007, ¶[0058]. “A data switch 402 inside the steering device 130 then relays the request message to the origin server 160 over the network link 412.” EX1007, ¶[0058]. “On the opposite direction, network traffic originated from the origin server 160 flows through the steering device 130 back to the user device 110 over the network response path.” EX1007, ¶[0058]. “For example, the origin server 160 responds to the user request by sending web content over the network link 413 to the steering device 130, which forwards the web content to the user device 110 over the network link 416.” EX1007, ¶[0058].

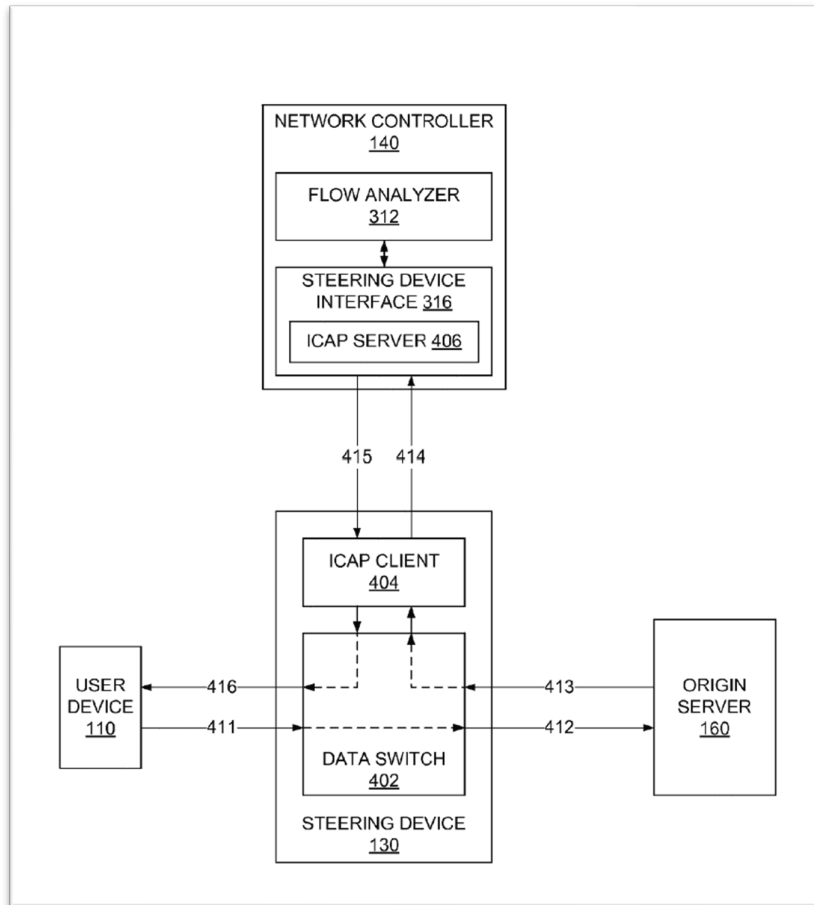


Figure 4A of Swenson

67. “In one embodiment, as the steering device 130 monitors network responses, it is looking for flows that match one or more signatures for video and images.” EX1007, ¶[0059]. “When a matching flow is detected, 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.” EX1007, ¶[0059]. “After receiving the request and the portion of response at the ICAP server interface 406, the flow analyzer 312 of the network

controller 140 performs a deep flow inspection to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059].

68. “If the flow is deemed of interest, the steering device 130 is notified to steer the flow through the network controller 140.” EX1007, ¶[0060]. “This is known as the ‘continue’ working mode for bandwidth monitoring.” EX1007, ¶[0060]. “In the ‘continue’ mode, the network controller 140 interfaces with the steering device 130 to function, on-demand, as a traditional inline network element for flows deemed of interest.” EX1007, ¶[0060]. “Thus, the network controller 140 ingests the network flow for inspection and subsequently forwards the network flow on the network response path.” EX1007, ¶[0060]. “For example, for this particular flow, the origin server 160 responds to the user request by sending video or images over the network link 413 to the steering device 130, which forwards the video or images to the network controller 140 over a network link 414.” EX1007, ¶[0060]. “After the network controller 140 updates the flow statistics, the video or images are returned to the steering device 130 over a network link 415, which transmits the video or images to the user device 110 over the network link 416.” EX1007, ¶[0060].

VI. CLAIM CONSTRUCTION

69. It is my understanding that claim terms in an IPR must be given their ordinary and customary meaning as understood by a POSA at the time of the

invention in light of the specification and the prosecution history pertaining to the patent. *See* 37 C.F.R. § 42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2015) (en banc). It is also my understanding that terms not specifically construed have their plain and ordinary meaning as understood by a POSA.

70. It is my opinion that the claim term “controller” should be construed to mean “an entity configured to perform deep packet inspection on packets.” EX1001, Col. 10:52-62. The ’111 Patent describes “a method for deep packet inspection (DPI) in a software defined network (SDN), wherein **the method is performed by a central controller** of the SDN.” EX1001, Col. 2:27-30 (emphasis added); see EX1001, Cols. 2:49-51, 3:56-59. Further, the patent explains that “the central controller 111 is **configured to perform deep packet inspection on designated packets** from designated flows or TCP sessions.” EX1001, Col. 4:5-7 (emphasis added). “To this end, the central controller 111 is further configured to instruct each of the network nodes 112 which of the packets and/or sessions should be directed to the controller 111 for packet inspections.” EX001, Col. 4:8-11. “A packet that requires inspection is either redirected to the controller 111 or mirrored and a copy thereof is sent to the controller 111.” EX1001, Col. 4:15-18. The ’111 Patent explains that “the mirrored bytes [of the packet] are sent to the controller 111 to perform the DPI analysis.” EX1001, Col. 4:49-50.

71. In addition, the '111 Patent states that “the central controller 111 includes a DPI flow detection module 311, a DPI engine 312, and a memory 313, and a processing unit 314,” as shown below in Figure 3. EX1001, Col. 5:33-36; see EX1001, Col. 5:40-59 (discussing DPI flow detection module 311 located within central controller 111 in Figure 3). “The DPI engine 312 [is] configured to inspect a packet or a number of bytes to provide application metadata as required by an application executed by an application server 120.” EX1001, Col. 5:36-39. A POSA would have understood that the central controller was configured in this manner in order to provide deep packet inspection (“DPI”) on redirected packets, as all of the embodiments in the '111 Patent disclose that redirected packets are sent to the central controller for DPI. See, e.g., EX1001, Cols. 4:15-18, 4:49-50, 8:1-5. Further, Figure 6 of the '111 Patent shows a flow chart for the method disclosed in that patent, and the specification explains that, at step 660, “the received mirror bytes are inspected using a DPI engine in the controller 111.” EX1001, Cols. 9:67-10:1.

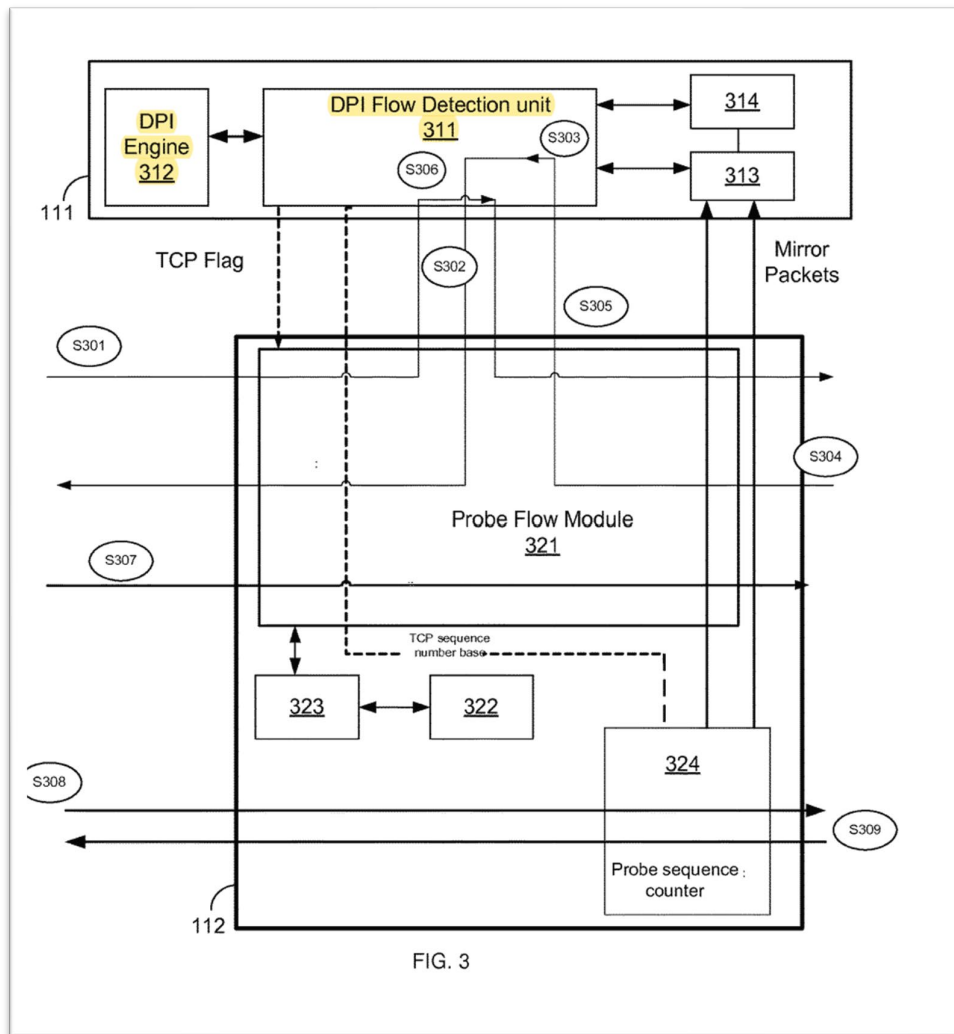


Figure 3 of the '111 Patent (Annotated)

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

72. Further, it is my opinion that the claim term “instruction” should be construed to mean “a command to determine if a packet requires inspection or not.” EX1001, Col. 10:56-62. The '111 Patent explains that “each network node 112 is configured to determine if an incoming packet requires inspection or not.” EX1001,

Col. 4:12-14. “The determination is performed based on a set of instructions provided by the controller 111.” EX1001, Col. 4:14-15. The patent explains that “the central controller 111 is further configured to instruct each of the network nodes 112 which of the packets and/or sessions should be directed to the controller 111 for packet inspections.” In addition, the exemplary instructions identified in the ’111 Patent provide various commands used to determine whether or not a packet requires inspection. EX1001, Col. 4:23-56; see EX1001, Cols. 8:23-32, 8:40-53, 9:26-28.

73. I have applied the plain and ordinary meaning as understood by a POSA of the remaining claim terms discussed herein.

VII. GROUNDS OF INVALIDITY

A. Ground 1: Claims 1-9, 12-24 and 27-31 of the ’111 Patent are obvious over Lin in view of Swenson.

74. It is further my opinion that a POSA would have found Claims 1-9, 12-24 and 27-31 of the ’111 Patent obvious over Lin in view of Swenson, and the knowledge of a POSA.

a. Claim 1

[1.0] A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method comprising:

75. To the extent that Element [1.0] is limiting, it is my opinion that Element [1.0] is disclosed or rendered obvious by the combination of Lin and Swenson. EX1001, Col. 10:51-55.

76. Lin discloses a method for use with a packet network. For example, Lin's Abstract states that it relates to a "computer network." EX1005, Abstract. The specification further states goes on to say that "[i]n one embodiment, a software defined networking (SDN) **computer network** includes an SDN controller and an SDN switch." EX1005, Col. 1:58-60 (emphasis added); see EX1005, Cols. 2:47-65, 3:25-33, 3:40-64. It is my opinion that a POSA would have readily understood that the computer network discussed in Lin is a packet network. Indeed, Lin refers to the "**transmission of packets** over the SDN **computer network** 600." EX1005, Col. 4:19-21 (emphasis added). Moreover, Figure 6 of Lin shows that the system disclosed in Lin is for a packet network, as can be seen below:

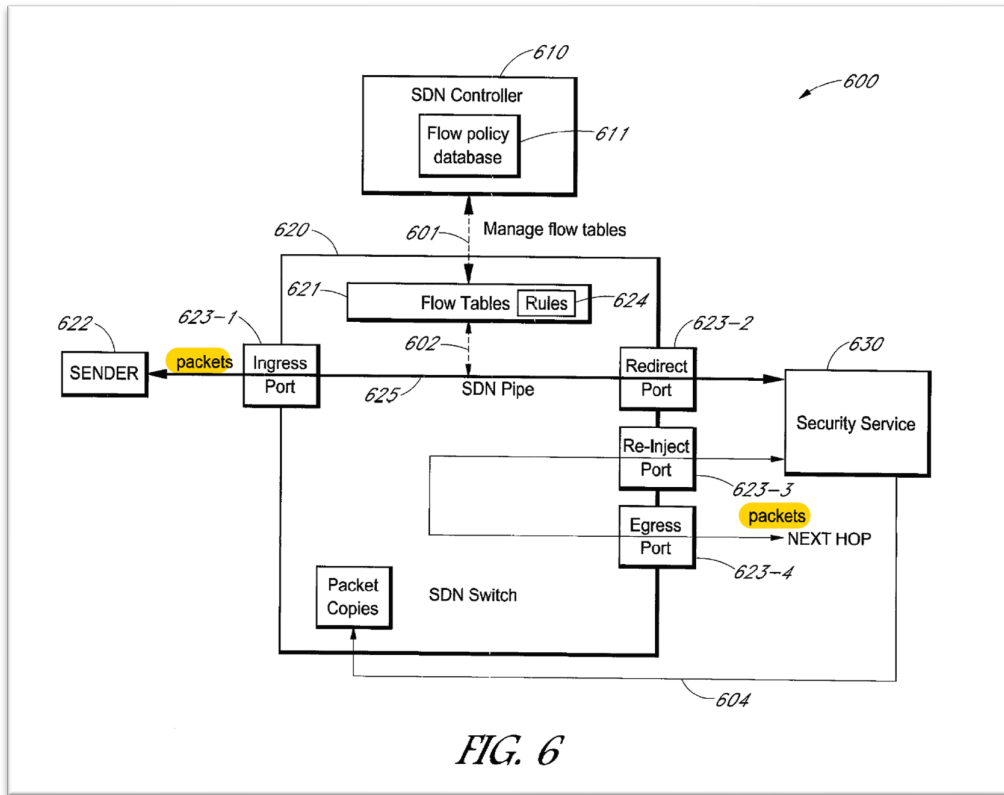


Figure 6 of Lin (Annotated)

EX1005, Figure 6; see EX1005, Figures 1-5, 7-8. In addition, Figure 9 of Lin shows that the method disclosed in Lin is for a packet network, as can be seen below:

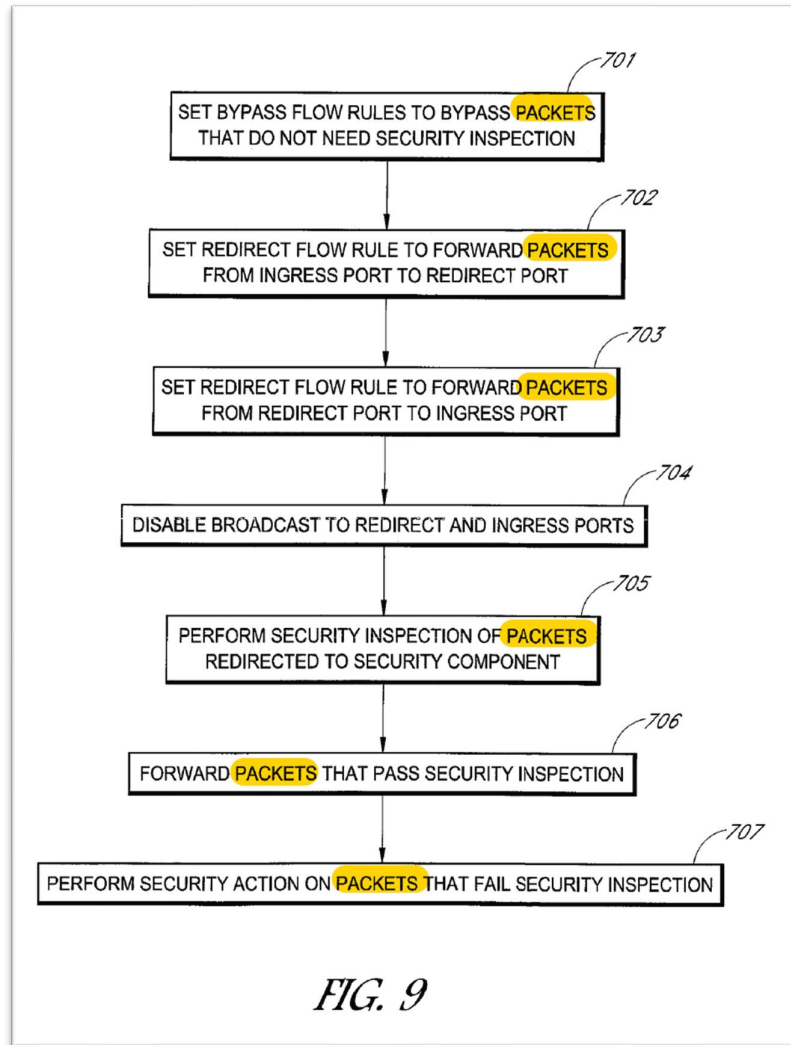


Figure 9 of Lin (Annotated)

EX1005, Figure 9.

77. Lin also teaches a network node for transporting packets between first and second entities. Lin discloses that an “SDN switch” that corresponds to the claimed network node. EX1005, Cols. 1:58-2:4, 4:33-67, 6:13-23, 6:57-63. Lin explains that the SDN switch (i.e., the network node) 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). EX1005, Cols. 1:58-2:4, 4:33-67, 6:13-23, 6:57-63, 7:10-23; 7:39-8:18, 9:63-10:22. Figure 6 of Lin (shown below) demonstrates that the SDN switch can be used to transport packets from a first entity to a second entity:

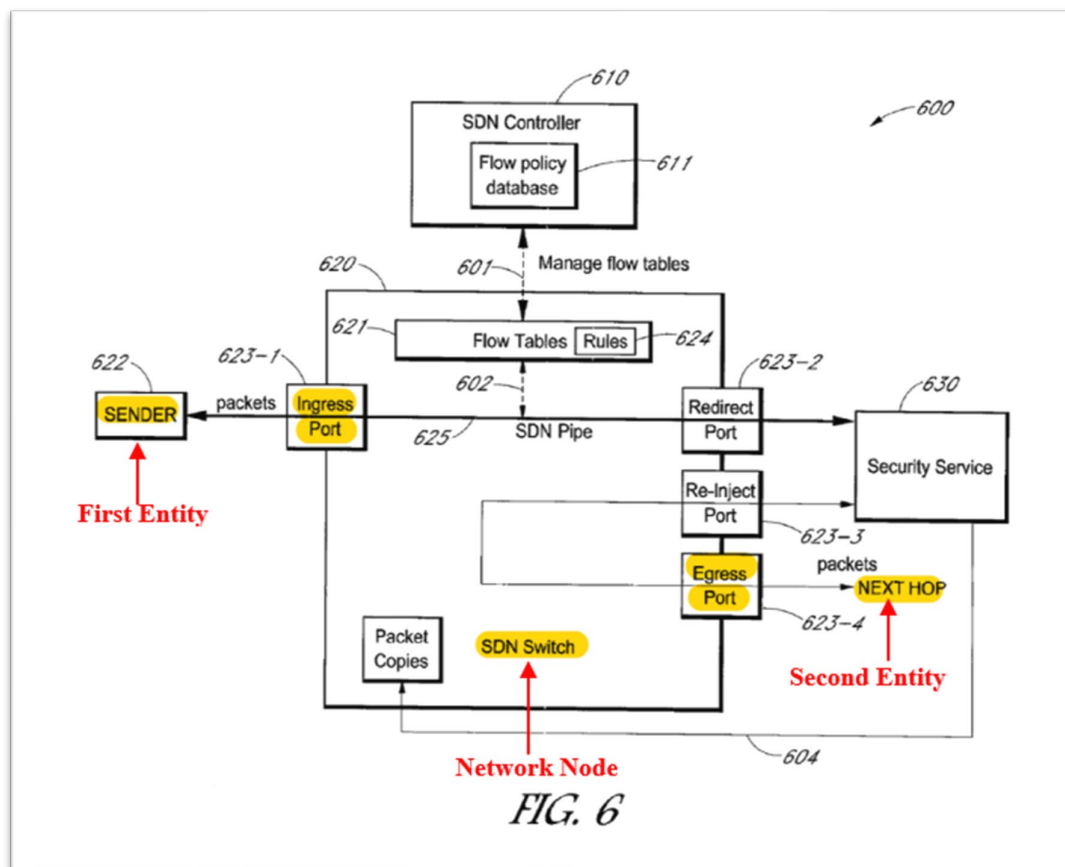


Figure 6 of Lin (Annotated)

EX1005, Figure 6.

78. Lin also discloses that the SDN switch (i.e., the network node) is under the control of a “SDN controller” that is external to the SDN switch. Lin explains that its controller “provides a logically centralized framework for controlling the behavior of the SDN computer network 600,” including one or more SDN switches:

The SDN **controller 610** provides a logically centralized framework for controlling the behavior of the SDN computer network 600. This is in marked contrast to traditional computer networks where the behavior of the computer network is controlled by low-level device configurations of switches and other network devices. The SDN **controller 610** may include a flow policy database 611. 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. The flow policies may specify security policies that govern transmission of packets over the SDN computer network 600. 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. As a particular example, a flow policy in the flow policy database 611 may indicate inspection of particular packets (e.g., those that meet one or more conditions) by a security service 630. That flow policy may be implemented as a flow rule that forwards the particular packets received in an ingress port 623-1 to the redirect port 623-2 for inspection, for example.

EX1005, Col. 4:8-31 (emphasis added). Lin explains that the controller includes a “flow policy database” that contains flow policies to control the transmission of packets through the SDN switch. EX1005, Col. 4:8-31; see EX1005, Cols. 1:58-2:4, 6:1-12. Further, Lin states that the controller is external to and separate from the SDN switch (i.e., the network node): “The SDN controller 610 and the SDN switch 620 are logically separate components.” EX1005, Col. 3:51-52. This would have made sense to a POSA because the controller is meant to be a centralized device for

controlling multiple SDN switches across a computer network. EX1005, Col. 4:8-

10. Figure 6 of Lin shows the SDN controller as external to the SDN switch:

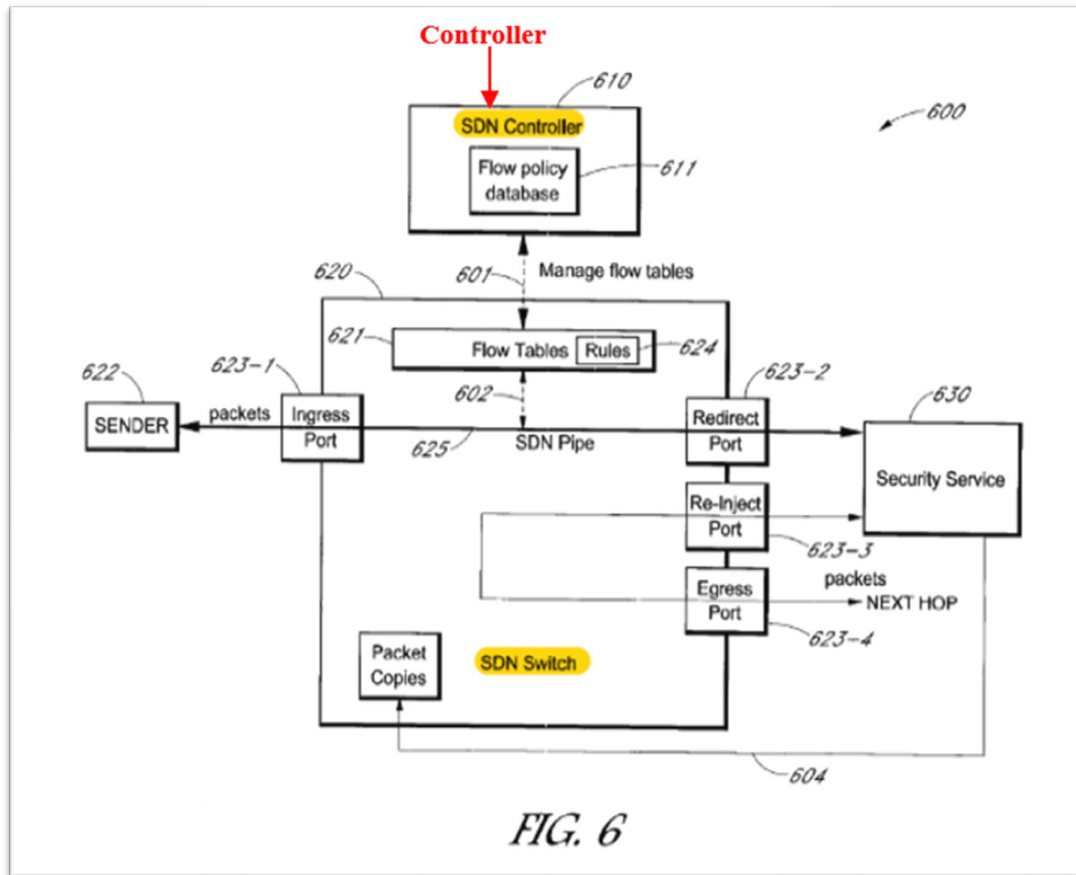


Figure 6 of Lin (Annotated)

EX1005, Figure 6; see EX1005, Figures 1, 7, 8.

79. Further, as discussed above with respect to claim construction, it is my opinion that the controller is an entity configured to perform DPI on packets.¹ Here, the combination of Lin and Swenson discloses a controller configured to perform DPI. Lin discloses that the analysis performed by security service 630 includes DPI: “Network security vendors provide network security services, such as firewall or deep packet inspection (DPI).” EX1005, Col. 3:11-12. Further, Lin teaches that security service 630 “may also comprise a physical machine, e.g., a server computer, an appliance, or a gateway computer, etc.” EX1005, Col. 5:51-55. Lin also discloses that “[t]he security service 630 may be connected to the SDN switch 620 by a physical link (i.e., using a wire), a virtual link (i.e., in a virtualized environment), or by a software tunnel.” A POSA would have understood these disclosures in Lin to teach that the security service 630 can be implemented using 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. As such, a POSA would have understood that one of the limited number of design options would have been to implement the security service as part of a controller configured to perform DPI

¹ To the extent that the PTAB does not agree with this construction, Lin still discloses Element [1.0] for the reasons discussed above.

analysis on packets, and a POSA would have had a reasonable expectation that the controller would have been successful in performing DPI analysis.

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

81. Further, Paragraph [0065] of Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the

video.” EX1007, ¶[0065], Figure 5. A POSA would have understood the analysis of the “response payload” to be DPI by a DPI-capable controller, as DPI refers to monitoring the payload of a packet. The above-discussed disclosures in Swenson demonstrate that it would have been well-known to a POSA as of the priority date for the ’111 Patent to implement a controller configured to perform DPI analysis in a system such as Lin. EX1007, ¶[0046], ¶[0059]-¶[0060], ¶[0073], ¶[0076]-[0077], ¶[0084]-¶[0086], Figures 1 and 4A-4B.

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

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

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

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

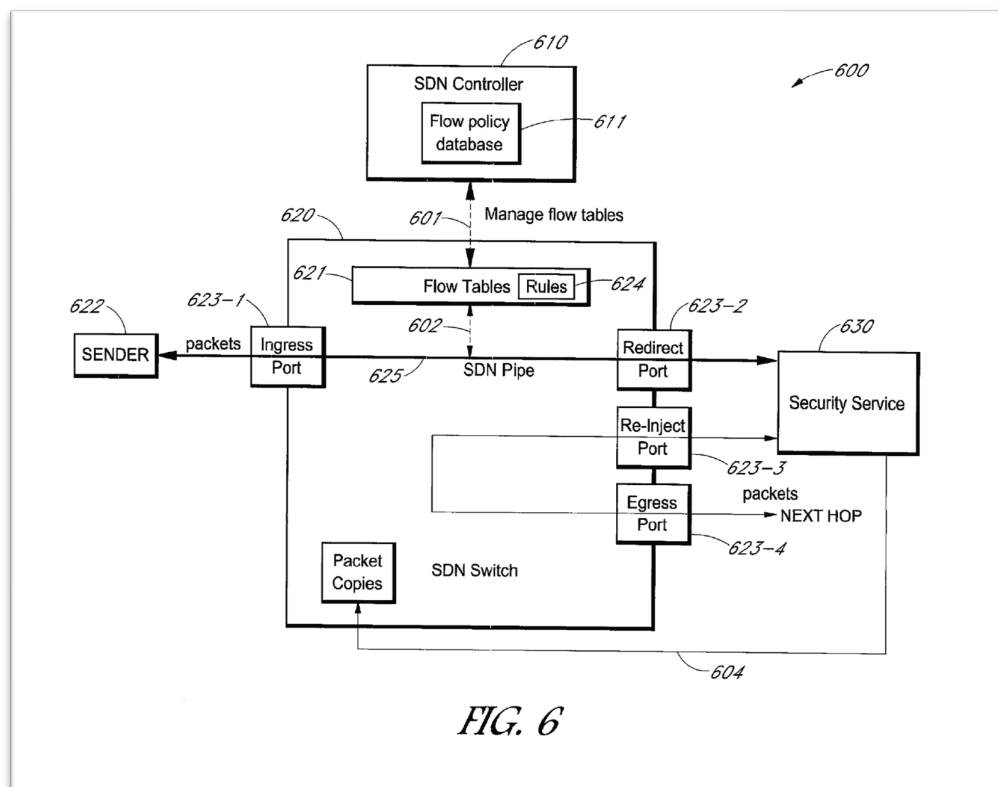


Figure 6 of Lin

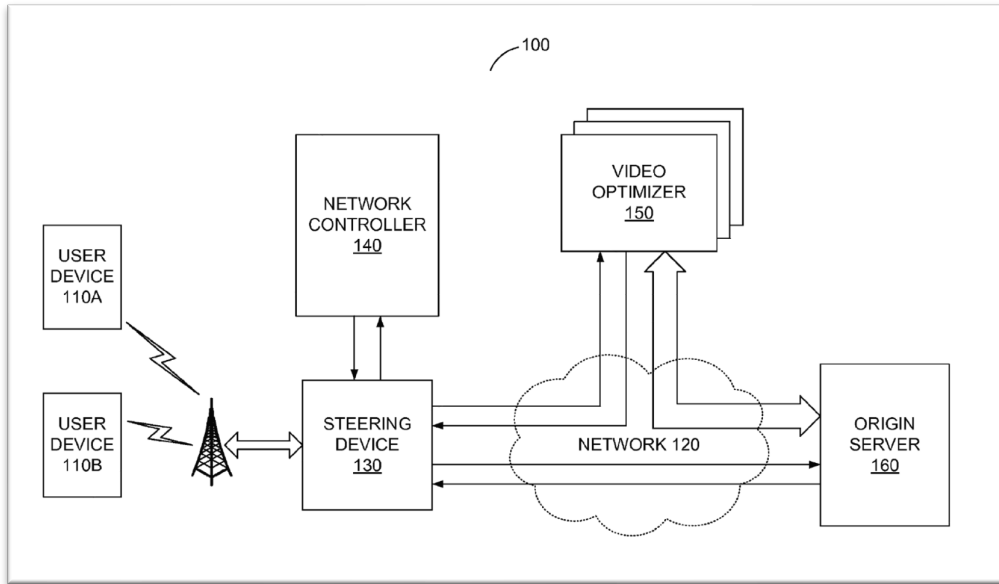


Figure 1 of Swenson

84. The determination of which packets to redirect (as opposed to redirecting all packets) increases the efficient operation of a computer network. Lin and Swenson each teach that only certain packets may need to be redirected for further processing before being sent to the destination node. EX1005, Cols. 1:58-2:4, 4:8-31, 5:8-55, 6:1-12, 6:40-63, 7:24-8:18, Figures 6-9; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A. This allows the methods in Lin and Swenson to improve the operation of transporting packets across a network while maintaining a proper speed for processing those packets by allowing

the system to stop redirecting certain packets in a flow if it is no longer necessary to do so.

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

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

1-4A. If Swenson could be constructed to have the network nodes send packets to the central controller for security inspection, a POSA would have understood that the same arrangement could be implemented in Lin. Indeed, it would have been common sense for a POSA to modify Lin in this way after reading Swenson.

87. In view of the above, it is my opinion that the combination of Lin and Swenson discloses or renders obvious Element [1.0].

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

88. It is my opinion that Element [1.1] is disclosed by Lin. EX1001, Col. 10:56-58.

89. As discussed above for Element [1.0], Lin discloses an external controller (i.e., the SDN controller) that controls network nodes (i.e., the SDN switches) in a packet network. EX1005, Cols. 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.

90. Further, Lin identifies an instruction sent by the controller to the network node over the packet network. As discussed above in the Claim Construction section, the instruction includes a command to determine whether or

not a packet requires inspection.² In Lin, the SDN controller inserts “flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.” These flow rules correspond to the claimed instruction, and they are sent to the SDN switches in order to provide instruction to the SDN switches on what packets should be sent to the security component for inspection:

In one embodiment, a software defined networking (SDN) computer network includes an SDN controller and an SDN switch. **The SDN controller inserts flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.** A broadcast function of the SDN switch to the ports that form the SDN pipe may be disabled. The SDN pipe allows outgoing packets sent by the sender component to be received by the security component. 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. The SDN controller may also insert a flow rule in the flow table of the SDN switch to bypass inspection of specified packets.

EX1005, Col. 1:58-2:4 (emphasis added); see EX1005, Cols. 4:14-31, 4:53-67, 6:1-

12. Though the quote above refers to “one embodiment,” the insertion of flows rules in a flow table of the SDN switch by the SND controller is common to the embodiments disclosed in Lin. For instance, Lin provides examples of flow rules

² To the extent that the PTAB does not agree with this construction, Lin still discloses Element [1.1] for the reasons discussed herein.

that are inserted into the SDN switch to determine how to handle HTTP packets received at the SDN switch. EX1005, Col. 7:39-8:18.

91. Lin also discloses a separate packet-applicable criterion sent by the controller to the network node over the packet network. Lin states that the flow rules “may indicate inspection of particular packets (e.g., those that meet one or more conditions) by a security service 630.” EX1005, Col. 4:23-31; see EX1005, Col. 5:8-12. It is my opinion that a POSA would have understood that the “one or more conditions” used to identify “particular packets” for inspection would be packet-applicable criterion. In other words, Lin teaches that criterion specific to a particular packet are used to determine what action to take on a packet, such as determining which packets are forwarded to security service 630. EX1005, Col. 4:23-31; see EX1005, Col. 5:8-12. Lin provides examples of the conditions (or criterion) applicable to a particular packet, including “‘IN_PORT’, ‘MAC src’ (media access control (MAC) address of the source of the packet), ‘MAC dst’ (MAC address of the destination of the packet), ‘IP src’ (Internet Protocol (IP) address of the source of the packet), ‘IP dst’ (IP address of the destination of the packet).” EX1005, Col. 5:16-21. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, Col. 5:22-24; see EX1005, Col. 5:26-36, 6:1-12, 6:40-54, Table 1. The “Action” corresponds to the claimed instruction.

92. A POSA would have understood that the conditions used to identify a packet (i.e., the claimed packet-applicable criterion) and the instruction of what action to take on a packet (i.e., the claimed instruction) are separate things even though they may both be part of the same flow rule in Lin. For example, Table 3 of Lin (shown below) demonstrates an implementation of Lin’s flow rules in an SDN switch. The first two rows in Table 3 contain two flow rules that instruct the SDN switch on what to do with HTTP packets. EX1005, Col. 8:10-18, Table 3. Table 3 identifies conditions (or criterion) applicable to a particular packet, including “IP src” (the IP address of the source of the packet), “TCP src port” (the TCP port of the previous router or switch that forwarded a packet) and “TCP dst port” (the TCP port of a router or switch through which the packet will be forwarded). EX1005, Col. 8:10-18, Table 3. The Action column instructs the SDN switch what to do with a packet the meets the packet-applicable criterion. The flow rule on the first row states that, if the packet has a “TCP dst port” of 80, which indicates an HTTP packet, the corresponding action is to send the packet to the “redirect port” that will send the packet to security service 630. This action corresponds to the claimed instruction.

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

Table 3 of Lin (Annotated)

EX1005, Col. 8:10-18 (emphasis added), Table 3.

93. Lin also teaches that the flow rules containing the packet-applicable criterion are sent by the SDN controller to the flow table in the SDN switch (which corresponds to the claim network node). Specifically, Lin states “[t]he SDN controller 610 may insert flow rules in the flow tables 621 (see arrow 601) to create an SDN pipe (labeled as 625) between the sender component 622 and the security service 630,” as shown below in Figure 6. EX1005, Col. 6:1-4; see EX1005, Col. 6:40-41 (“redirect flow rules for creating the SDN pipe are inserted in the flow tables 621”). 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.” EX1005, Col. 7:24-26; see EX1005, Cols. 7:27-8:18, Tables 2-3. It is my opinion that a POSA would have understood from these disclosures that Lin discloses a packet-applicable

criterion sent by the controller to the network node (i.e., the SDN switch) over the packet network.

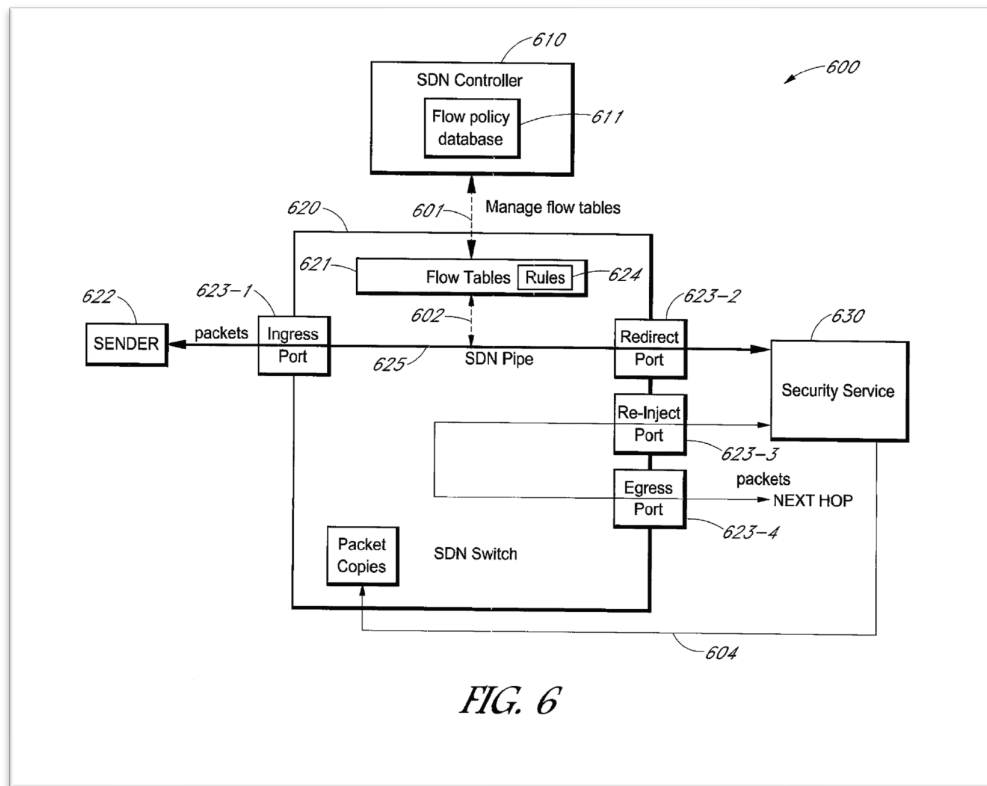


FIG. 6

Figure 6 of Lin

94. Accordingly, it is my opinion that Lin discloses Element [1.1].

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

95. It is my opinion that Element [1.2] is disclosed by Lin. EX1001, 10:59-60.

96. Lin describes the SDN switch (i.e., the network node) receiving an instruction sent by the controller over the packet network. Specifically, the SDN

switch receives “flow rules [i.e., the claimed instruction] in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.” These flow rules are sent to the SDN switches to in order to provide instruction to the SDN switches on what packets should be sent to the security component for analysis:

In one embodiment, a software defined networking (SDN) computer network includes an SDN controller and an SDN switch. **The SDN controller inserts flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.** A broadcast function of the SDN switch to the ports that form the SDN pipe may be disabled. The SDN pipe allows outgoing packets sent by the sender component to be received by the security component. 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. The SDN controller may also insert a flow rule in the flow table of the SDN switch to bypass inspection of specified packets.

EX1005, Col. 1:58-2:4 (emphasis added); see EX1005, Cols. 4:14-31, 4:53-67, 6:1-12. Lin also provides examples of flow rules being received by the SDN switch that determine how to handle HTTP packets received at the SDN switch. EX1005, Col. 7:39-8:18.

97. Lin also discloses a packet-applicable criterion received by the network node from the controller. As discussed above in Element [1.]1, Lin states that the flow rules “may indicate inspection of particular packets (e.g., those that meet one or more conditions) by a security service 630.” EX1005, Col. 4:23-31; see EX1005, Col. 5:8-12. It is my opinion that a POSA would have understood that the “one or

more conditions” used to identify “particular packets” for inspection would be packet-applicable criterion. In other words, Lin teaches that criterion specific to a particular packet are used to determine which actions to take on a packet, such as forwarding a packet to security service 630. As discussed above for Element [1.1], Lin provides examples of the conditions (or criterion) applicable to a particular packet.” EX1005, Col. 5:16-21. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, Col. 5:22-24 see EX1005, Col. 5:26-36, Table 1.

98. Further, as discussed above for Element [1.1], Lin provides examples of flow rules in Table 3 that identify conditions (or criterion) applicable to a particular packet, including “IP src,” “TCP src port,” and “TCP dst port.” EX1005, Col. 8:10-18, Table 3. The Action column in Table 3 instructs the SDN switch what to do with a packet the meets the packet-applicable criterion. The flow rule on the first row states that, if the packet has a “TCP dst port” of 80, which indicates an HTTP packet, the corresponding action is to send the packet to the “redirect port” that will send the packet to security service 630. This action corresponds to the claimed instruction. EX1005, Cols. 2:2-4, 7:24-8:18.

99. Further, Lin discloses that the flow rules containing the packet-applicable criterion are received in the flow table in the SDN switch (which

corresponds to the claim network node) after being sent by the SDN controller. Specifically, Lin states “[t]he SDN controller 610 may insert flow rules in the flow tables 621 (see arrow 601) to create an SDN pipe (labeled as 625) between the sender component 622 and the security service 630,” as shown below in Figure 6. EX1005, Col. 6:1-4; see EX1005, Col. 6:40-41 (“redirect flow rules for creating the SDN pipe are inserted in the flow tables 621”). 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.” EX1005, Col. 7:24-26; see EX1005, Cols. 7:27-8:18, Tables 2-3. It is my opinion that a POSA would have understood from these disclosures that Lin discloses a packet-applicable criterion received by the network node (i.e., the SDN switch) from the controller.

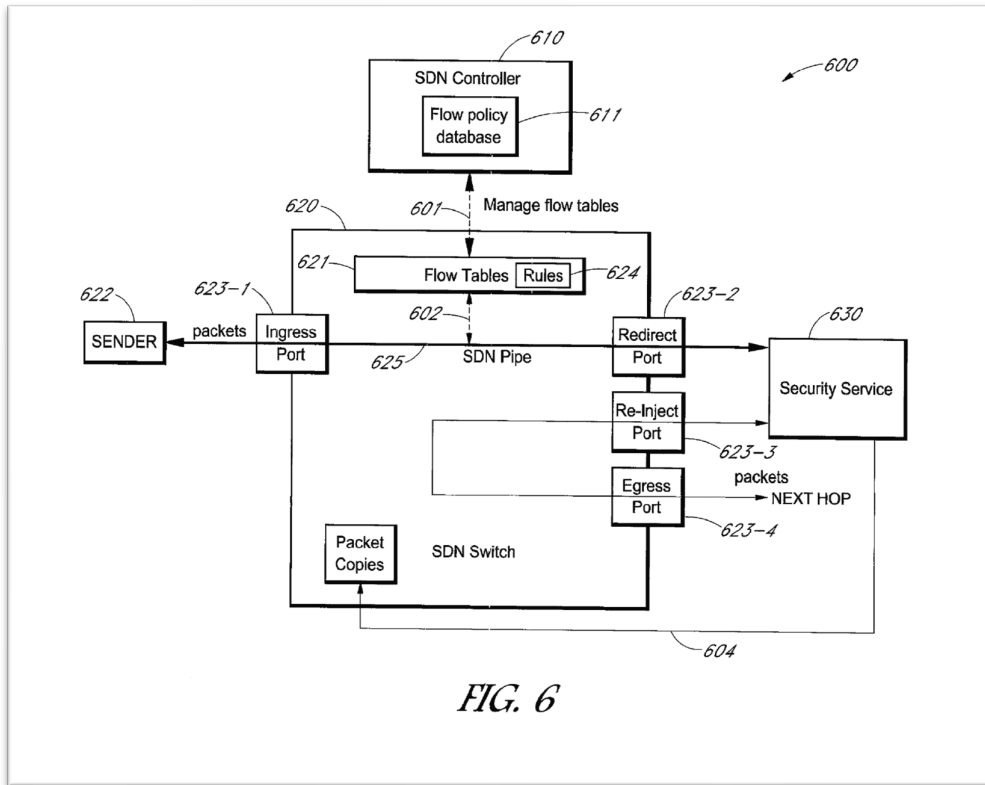


FIG. 6

Figure 6 of Lin

100. Accordingly, it is my opinion that Lin discloses Element [1.2].

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

101. It is my opinion that Element [1.3] is disclosed by Lin. EX1001, Col. 10:60-62.

102. Lin explains that the SDN switch (i.e., the network node) 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). EX1005, Cols. 1:58-2:4, 4:33-67,

6:13-23, 6:57-63, 7:10-23; 7:39-8:18, 9:63-10:22. A POSA would have understood that the packet is addressed to the next hop or second entity. Indeed, the IP address of the destination of the packet is one of the criterion that is specifically discussed in Lin. EX1005, Col. 5:16-31. Further, Figure 6 of Lin (shown below) demonstrates that the SDN switch can be used to transport packets from a first entity to a second entity:

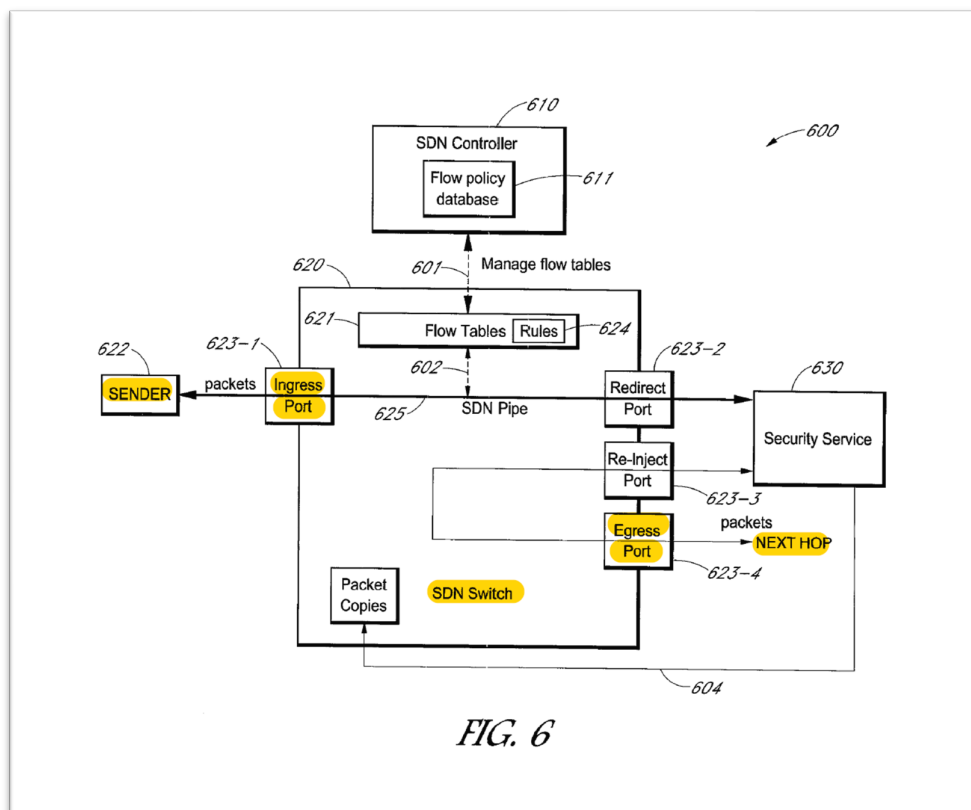


Figure 6 of Lin (Annotated)

EX1005, Figure 6.

103. Accordingly, it is my opinion that Lin discloses obvious Element [1.3].

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

104. It is my opinion that Element [1.4] is disclosed by Lin. EX1001, Col. 10:63-64.

105. Lin teaches that packet-applicable criterion are sent by the controller (the SDN controller) and received by the network node (the SDN switch). EX1005, Cols. 6:1-4, 6:40-41, 7:24-8:18. Lin states that its flow rules “may indicate inspection of particular packets (e.g., those that meet one or more conditions) by a security service 630.” EX1005, Col. 4:23-31; see EX1005, Col. 5:8-12. It is my opinion that a POSA would have understood this disclosure to mean that the “one or more conditions” used to identify “particular packets” for inspection would be packet-applicable criterion, and that the SDN switch (i.e., the network node) checks to determine whether incoming packets satisfy the packet-applicable criterion. EX1005, Cols. 4:23-31, 5:8-12. Indeed, Lin provides examples of the conditions (or criterion) applicable to a particular packet that are checked by the SDN switch, including “IN_PORT,” “MAC src,” “MAC dst,” “IP src,” and “IP dst.” EX1005, Col. 5:16-21. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, Col. 5:22-24; see EX1005, Col. 5:26-36, 6:1-12, 6:40-54, Table 1. Further, in discussing the examples shown in Table 2 and Table 3, Lin discloses that the SDN switch implements “bypass flow rules” that check whether

the packet meets certain packet-applicable criterion, such as identification of HTTP packets, which indicate that the packet should be routed to the destination node instead of the security device. EX1005, Cols. 7:24-8:18.

106. Accordingly, it is my opinion that Lin discloses Element [1.4].

[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

107. It is my opinion that Element [1.4] is disclosed by Lin. EX1001, 10:65-67.

108. Lin teaches that packet-applicable criterion are sent by the controller (the SDN controller) and received by the network node (the SDN switch). EX1005, Cols. 5:16-24, 6:1-4, 6:40-41, 7:24-8:18. Indeed, Lin provides examples of the conditions (or criterion) applicable to a particular packet that are checked by the SDN switch, including “IN_PORT,” “MAC src,” “MAC dst,” “IP src,” and “IP dst.” EX1005, Col. 5:16-21. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, Col. 5:22-24.

109. Further, Lin discloses that the SDN switch implements “bypass flow rules” that check whether the packet meets certain packet-applicable criterion, such as identification of HTTP packets, which indicate that the packet should be routed to the destination node instead of the security device. EX1005, Cols. 7:24-8:18. Lin’s discussion of Table 3 teaches that the devices 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. EX1005, Cols. 7:64-8:18. Lin states that, if a packet does not satisfy this criterion (i.e., the packet does not indicate that port 80 is the source or destination port), then the SDN switch (which corresponds to the claimed network node) sends the packet over the packet network to its destination node (which corresponds to the claimed second entity). EX1005, Col. 8:10-18. Specifically, Lin provides the following discussion for the operations shown in Table 3 (which is reproduced below in annotated form):

In the example of Table 3, the top two rows are redirect flow rules for redirecting HTTP packets to the security service 630 for inspection, **while the bottom two rows [highlighted below] are bypass flow rules for all packets.** Because the redirect flow rules are at higher priority than the bypass flow rules, HTTP packets are sent through the SDN pipe formed in the SDN switch 620 between the sender component 622 and the security service 630. **All other packets bypass the SDN pipe, and are accordingly not inspected by the security service 630.**

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

Table 3 of Lin (Annotated)

EX1005, Col. 8:10-18 (emphasis added), Table 3.

110. Accordingly, it is my opinion that that Lin discloses Element [1.5].

[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.

111. It is my opinion that Element [1.6] is disclosed by Lin. EX1001, Col. 11:1-4.

112. Lin teaches that packet-applicable criterion are sent by the controller (the SDN controller) and received by the network node (the SDN switch). EX1005, Cols. 5:16-24, 6:1-4, 6:40-41, 7:24-8:18. Indeed, Lin provides examples of the conditions (or criterion) applicable to a particular packet that are checked by the SDN switch, including “IN_PORT,” “MAC src,” “MAC dst,” “IP src,” and “IP dst.” EX1005, Col. 5:16-21; see EX1005, Col. 5:26-36, 6:1-12, 6:40-54, Table 1. Lin states, “When the conditions are met, i.e., the particular packet is identified, the action indicated in the corresponding ‘Action’ column is performed on the packet.” EX1005, Col. 5:22-24.

113. Further, Lin discloses that the SDN switch implements “bypass flow rules” that check whether the packet meets certain packet-applicable criterion, such as identification of HTTP packets, which indicate that the packet should be routed to the security device. EX1005, Cols. 7:24-8:18. Lin’s discussion of Table 3 teaches that the devices check 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. EX1005, Cols. 7:64-8:18. Lin states that, if a packet satisfies this criterion (i.e., the packet indicates that port 80 is the source or destination port), then the SDN switch (which corresponds to the claimed network node) sends the packet over the packet network to the security device (which corresponds to the claimed entity that is included in the instruction and is other than the second entity). EX1005, Col. 8:10-18. Specifically, Lin provides the following discussion for the operations shown in Table 3 (which is reproduced below in annotated form):

In the example of Table 3, the top two rows are redirect flow rules for redirecting HTTP packets to the security service 630 for inspection, while the bottom two rows [highlighted below] are bypass flow rules for all packets. Because the redirect flow rules are at higher priority than the bypass flow rules, HTTP packets are sent through the SDN pipe formed in the SDN switch 620 between the sender component 622 and the security service 630. All other packets bypass the SDN pipe, and are accordingly not inspected by the security service 630.

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

Table 3 of Lin (Annotated)

EX1005, Col. 8:10-18 (emphasis added), Table 3.

114. Accordingly, it is my opinion that Lin discloses Element [1.6].

b. Claim 2

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

115. It is my opinion that Claim 2 is obvious in view of Lin and the knowledge of a POSA. EX1001, Col. 11:5-10.

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

117. Further, it is my opinion that, where the SDN switch (i.e., the network node) in Lin receives a terminate instruction, which 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. Lin states

that the SDN switch can be instructed to drop (i.e., block) packets before the packets are sent to their destination node. EX1005, Cols. 1:28-32, 7:19-22. Lin does not expressly state that its packets are blocked from being sent to the controller, but a POSA would have understood that such a terminate instruction easily could have been implemented. It would have been straightforward for the flow rules sent to the SDN switch of Lin to instruct that certain packets are dropped by the SDN switch anytime they are received and are not forwarded to the controller (or the destination node). For example, a POSA would have known to implement a terminate instruction to drop (i.e., block) packets received at the SDN switch from an IP address that is known to originate malicious code, and thereby prevent those packets from being sent to the controller (or destination node) where the malicious code could do damage.

118. Accordingly, it is my opinion that Lin in view of the knowledge of a POSA discloses Claim 2.

c. Claim 3

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

119. It is my opinion that Claim 3 is obvious in view of Lin. EX1001, Col. 11:11-16.

120. It is my opinion that Lin teaches that the instruction can be a “mirror” instruction. A POSA would have understood that a mirror instruction was a common instruction that can operate in one of two ways: (1) redirecting a packet from its intended destination to a new destination (such as a security service) or (2) making a copy of a packet and sending the copy of the packet to a new destination (such as a security service). Lin discloses both methods of mirroring a packet. For example, Lin states that incoming packets can be “redirected or mirrored to the security service”:

Another way of intercepting network traffic is to **mirror the packets** to be inspected on a switch that provides vendor specific mirroring application programming interface (API) as shown in [Figure] 4. A user may make an API call such that particular packets that enter the ingress port of the Switch are **redirected or mirrored to the security service** by way of a connection tunnel or a mirror port. The security service may forward the redirected or mirrored packets back to an egress port of the switch after inspection.

EX1005, Col. 3:25-33 (emphasis added). Lin also discloses that a copy of a packet can be made and the copy mirrored to the security service:

Re-injecting packets that pass inspection consume[s] bandwidth, as the packets will have to be transmitted by the security service 630 to the re-inject port 623-3. For optimization, the SDN switch 620 may be configured to **copy packets that are redirected to the security service 630 for inspection**. This way, the security service 630 simply has to inform the SDN switch 620 an action to take on the packets based on the result of the inspection (see arrow 604). For example, the security service 630 may send an index identifying the packets and an action how to manipulate the packets. The action may instruct the SDN switch 620 to drop the **copied packets**, forward the **copied packets** to their destinations, quarantine the **copied packets**, etc.

EX1005, Col. 7:10-22 (emphasis added). A POSA would have understood these passages from Lin to teach that the flow rules in Lin could include a mirror instruction.

121. Where the SDN switch (i.e., the network node) 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 the SDN switch to send the packet to the destination node (i.e., the second entity) and to the controller, as discussed below:

122. Re-injecting packets that pass inspection consume[s] bandwidth, as the packets will have to be transmitted by the security service 630 to the re-inject port 623-3. For optimization, the SDN switch 620 may be configured to **copy packets that are redirected to the security service 630 for inspection**. This way, the security service 630 simply has to inform the SDN switch 620 an action to take on the packets based on the result of the inspection (see arrow 604). For example, the security service 630 may send an index identifying the packets and an action how to manipulate the packets. The action may instruct the SDN switch 620 to drop the copied packets, **forward the copied packets to their destinations**, quarantine the copied packets, etc.

EX1005, Col. 7:10-22 (emphasis added). This passage from Lin states that the mirror instruction instructs the SDN switch to send a packet to the security service 630 for inspection. EX1005, Col. 7:10-22. As discussed above for Element [1.6], it would have been obvious to implement the security service 630 in the controller; thus, the original packet is sent to the controller by the SDN switch. Further, the above-quoted passage from Lin states that, if the packet passes inspection, the mirror instruction instructs the SDN switch to forward a copy of the packet to the destination node. EX1005, Col. 7:10-22.

123. Accordingly, it is my opinion that Lin discloses Claim 3.

d. Claim 4

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

124. It is my opinion that Claim 4 is obvious over Lin in view of the knowledge of a POSA. EX1001, Col. 11:17-26.

125. It is my opinion that Lin teaches that the instruction can be a “probe” instruction. A POSA would have understood that a probe instruction was a well-known instruction to send an incoming packet to a security service so that the packet can be probed or inspected for malicious code. 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:

In one embodiment, a software defined networking (SDN) computer network includes an SDN controller and an SDN switch. **The SDN controller inserts flow rules in a flow table of the SDN switch to create an SDN pipe between a sender component and a security component.** A broadcast function of the SDN switch to the ports that form the SDN pipe may be disabled. The SDN pipe allows outgoing packets sent by the sender component to be received by the security component. 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. The SDN controller may also insert a flow rule in the flow table of the SDN switch to bypass inspection of specified packets.

EX1005, Col. 1:58-2:4 (emphasis added); see EX1005, Cols. 4:14-31, 4:53-67, 6:1-12; 9:23-40, Figures 6-9. Lin teaches an example wherein the flow rules received by the SDN switch instruct the SDN switch to redirect non-HTTP packets to the security service 630 for inspection. EX1005, Col. 7:39-63. Lin also provides an example of flow rules being received by the SDN switch that instruct the SDN switch to redirect “HTTP packets to the security service 630 for inspection.” EX1005, Col. 7:64-8:18. A POSA would have understood from these disclosures in Lin that the flow rules sent to the SDN switch include a probe instruction for sending the packets to the security component.

126. Where the SDN switch (i.e., the network node) in Lin receives a probe instruction, and the packet satisfies the criterion (as discussed above for Element

[1.6]), Lin teaches that the probe instruction instructs the SDN switch to send the packet to the security service 630 for inspection. EX1005, Abstract, Cols. 1:58-2:4, 4:53-67, 5:26-36, 6:1-12, 6:40-63, 7:10-8:18, Figures 6-9. As discussed above for Element [1.6], it would have been obvious to implement the security service 630 in the controller; thus, the packet is sent to the controller for inspection.

127. A POSA would have understood from Lin that, when the packet is received by the controller, the packet is analyzed by the security service 630 in the controller. EX1005, Cols. 6:48-50 (“The security service 630 may inspect the outgoing packets for compliance with security policies.”), 5:45-55, 7:10-22, 8:33-35, Figure 9 (“perform security inspection of packets redirected to security component”). When the analysis is complete, a POSA would have understood from Lin that the packet is sent from the controller back to the SDN switch. EX1005, Cols. 6:54-57 (“The security service 630 may forward those packets that pass inspection toward their destination by re-injecting the packets back into the SDN switch 620 by way of the re-inject port 623-3.”), 7:10-22, 8:35-38. Further, a POSA would have understood from Lin that, when the packet is received by the SDN switch, the packet is sent from the SDN switch to the destination node (i.e., the second entity). EX1005, Cols. 6:60-63 (“Accordingly, the re-injected packets are forwarded to the egress port 623-4 (or some other port) toward the next hop in accordance with the L2 switching

logic of the SDN computer network 600.”), 7:10-22, 8:38-45, Figure 9 (“forward packets that pass security inspection”).

128. Accordingly, it is my opinion that Lin in view of the knowledge of a POSA discloses Claim 4.

e. **Claim 5**

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

129. It is my opinion that Claim 5 is obvious in view of the combination of Lin and Swenson. EX1001, Col. 11:27-30.

130. As discussed above for Elements [1.1] and [1.2], Lin discloses that a network node has an instruction and packet-applicable criterion. Further, as discussed above for Elements [1.4]-[1.6], Lin discloses that an action is taken responsive to the instruction and to the packet satisfying the packet-applicable criterion.

131. Further, Swenson teaches that the network node sends a packet or a portion thereof to the controller in response to the packet satisfying a criterion. As discussed above for Element [1.0], Swenson discloses that, when its system detects a HTTP packet flow matching a particular signature, “the steering device 130 forwards the HTTP request and a portion of the HTTP response to the network controller 140 over the ICAP client interface 404.” EX1007, ¶[0059]

(emphasis added]; see EX1007, ¶[0060], ¶[0065], Figures 1, 4A-4B. A POSA would have understood this to mean that the steering device 130 corresponds to at least part of the claimed network node, and the signature that is compared to the HTTP packet flow corresponds to the claimed criterion. Responsive to the HTTP packet flow satisfying that criterion, the steering device applies an instruction pursuant to which the HTTP request and a portion of the HTTP response is sent to the network controller 140.

132. Accordingly, it is my opinion that the combination of Lin and Swenson discloses and renders obvious Claim 5.

f. Claim 6

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

133. It is my opinion that Claim 6 is obvious over Swenson in view of the knowledge of a POSA. EX1001, Col. 11:31-33.

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

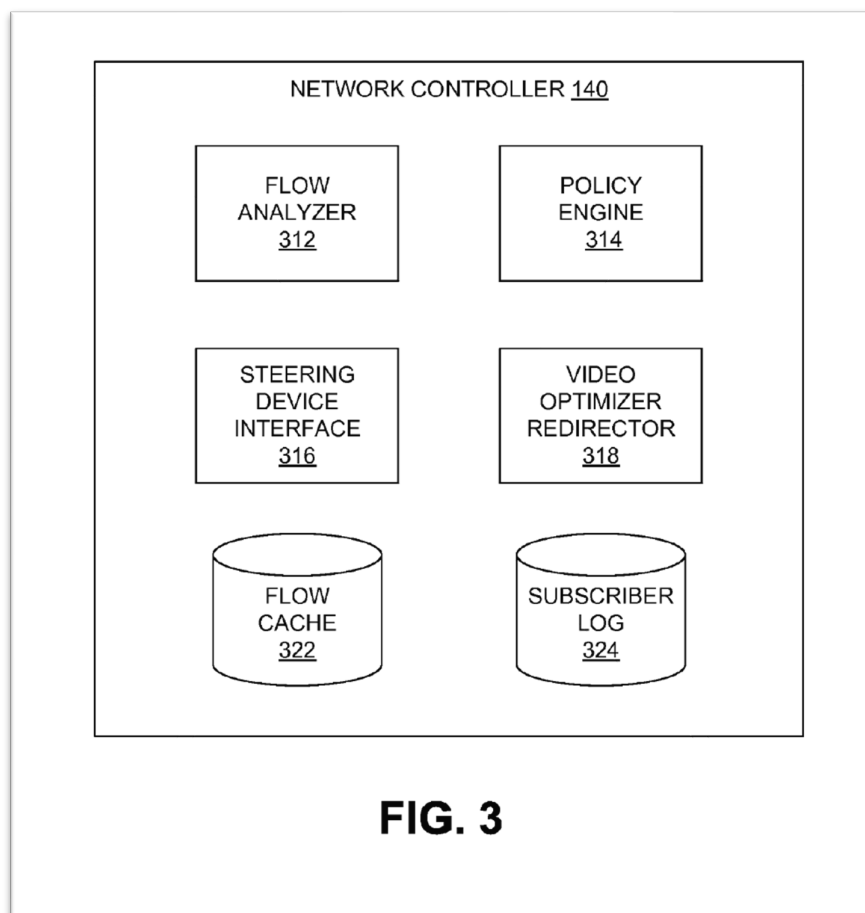


FIG. 3

Figure 3 of Swenson

Swenson also explains that “[t]he flow cache entry keeps track of the flow and its associated bandwidth.” EX1007, ¶[0061]; see E1007, ¶[0040], ¶[0046], ¶[0062]. In discussing Figure 7, Swenson states that the flow cache entry stores at least a portion of the received packet, including at least the IP address of the received packet. EX1007, ¶[0073] (discussing Figure 7); see EX1007, ¶[0074]-¶[0085]. Further, Swenson states that a flow cache entry can also store the MAC address or TCP source port associated with a received packet. EX1007, ¶[0084]. A POSA would

have understood from these disclosures that Swenson teaches that the controller stores at least a portion of a packet in a flow cache memory.

135. Accordingly, it is my opinion that Swenson in view of the knowledge of a POSA discloses and renders obvious Claim 6.

g. Claim 7

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

136. It is my opinion that Claim 7 is obvious over Swenson. EX1001, Col. 11:34-37.

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

applies an instruction pursuant to which a portion of a HTTP request or a portion of a HTTP response is sent to the network controller.

138. Accordingly, it is my opinion that Swenson discloses and renders obvious Claim 7.

h. Claim 8

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

139. It is my opinion that Claim 8 is obvious over Swenson in view of the knowledge of a POSA. EX1001, Col. 11:38-41.

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

141. Accordingly, it is my opinion that Swenson in view of the knowledge of a POSA discloses and renders obvious Claim 8.

i. Claim 9

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

142. It is my opinion that Claim 9 is obvious in view of Swenson. EX1001, Col. 11:42-44.

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

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

EX1007, ¶[0059] (emphasis added); see EX1007, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B. Further, Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶[0065], Figure 5. A POSA would have understood the inspection of the “HTTP GET request header and a portion of the [HTTP] response payload” to be analyzing the packet upon the controller receiving the packet. This is consistent with Lin, which explains that an analysis is performed on its packets. EX1005, Cols. 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.

144. Accordingly, it is my opinion that Swenson discloses and renders obvious Claim 9.

j. **Claim 12**

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

145. It is my opinion that Claim 12 is obvious over both Lin and Swenson, along with the knowledge of a POSA. EX1001, Col. 11:55-57.

146. Lin states that the analysis performed on the packets includes a security processing function. EX1005, Cols. 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.

147. Further, Swenson discloses that packets are sent to the controller for DPI “to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059]-¶[0060]; see EX1007, ¶[0065] (disclosing that the controller inspects packets to determine whether to perform bandwidth monitoring on the flow). As discussed above for Element [1.0], a POSA would have understood that bandwidth monitoring can be used for applying a security or data analytic application that monitors for DOS attacks. EX1007, ¶[0059]-¶[0060]; see EX1007, ¶[0039] (explaining that the controller can incorporate “security functions”).

148. Accordingly, it is my opinion that both Lin and Swenson disclose and render obvious Claim 12.

k. **Claim 13**

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

149. It is my opinion that Claim 13 is obvious over Lin. EX1001, Col. 11:58-60.

150. Lin discloses that its system analyzes packets by applying a security application that comprises either a firewall or intrusion detection functionality, or both: “The security service 630 may inspect packets for compliance/non-compliance with security policies, such as for the presence of malicious code, **compliance with firewall rules** and access control lists, **network intrusion detection**, and other computer network security services.” EX1005, Col. 5:45-50 (emphasis added), see EX1005, Cols. 3:11-12, 6:50-54.

151. Accordingly, it is my opinion that Lin discloses and renders obvious Claim 13.

l. **Claim 14**

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

152. It is my opinion that Claim 14 is obvious over Lin and Swenson. EX1001, Col. 11:61-63.

153. Lin discloses that its analysis includes DPI: “Network security vendors provide network security services, such as firewall or deep packet inspection (DPI).” EX1005, Col. 3:11-12.

154. Swenson also explains that “the flow analyzer 312 of the network controller 140 performs a deep flow inspection to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059]. Further, as explained above for Element [1.0], Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶[0065], Figure 5. A POSA would have understood the analysis of the “response payload” to be DPI.

155. Accordingly, it is my opinion that both Lin and Swenson disclose and render obvious Claim 14.

m. Claim 15

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

156. It is my opinion that Element of Claim 15 is obvious over both Swenson and Lin, in view of the knowledge of a POSA. EX1001, Col. 11:64-67.

157. It would have been well-known to any POSA as of the priority date of the '111 Patent that a packet includes distinct header and payload fields. EX1008, Section 2.

158. Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶¶[0065], Figure 5. A POSA would have understood the analysis of the “response payload” by the Swenson system to be checking part of the payload field of a packet by the network controller.

159. Further, a POSA would have understood from the disclosures of Lin that the packet analysis includes checking part or the whole of the payload field. As discussed above for Claim 14, Lin discloses analysis of packets through DPI. EX1005, Col. 3:11-12; EX1006, ¶¶[0021], ¶¶[0028], ¶¶[0040]-¶¶[0041]. A POSA would have understood that DPI refers to the inspection of at least part of the payload field. Further, Lin specifically discusses that its packet analysis includes inspection of packets “for compliance/non-compliance with security policies, such as for presence of malicious code, compliance with firewall rules and access control lists, network intrusion detection, and other computer network security services.” EX1005, Col.

5:45-50. A POSA would have understood from this passage of Lin that, as of the priority date for the '111 Patent, analysis of packets for malicious code or network intrusion detection would have included analysis of at least part of the payload field of a packet.

160. In view of the above, it is my opinion that both Swenson and Lin, in view of the knowledge of a POSA, disclose and render obvious Claim 15.

n. **Claim 16**

[16] The method according to claim 1, wherein the packet comprises distinct header and payload fields, the header comprises one or more flag bits, and wherein the packet-applicable criterion is that one or more of the flag bits is set.

161. It is my opinion that the element of Claim 16 is obvious over Lin in view of the knowledge of a POSA. EX1001, Col. 12:1-4.

162. It would have been well-known to any POSA as of the priority date of the '111 Patent that a packet includes distinct header and payload fields. EX1008, Section 2. Further, it would have been well-known to a POSA that the header field includes one or more flag bits. EX1008, Section 2. In addition, it would have been obvious to a POSA that the flow rules of Lin could have used a determination of whether or not a flag bit was set as one of Lin's "conditions" that serve as packet-applicable criterion. EX1005, Col. 4:23-31; see EX1005, Cols. 5:8-12, 7:24-8:18.

163. Accordingly, it is my opinion that Lin in view of the knowledge of a POSA discloses Claims 16.

o. Claim 17

[17] The method according to claim 16, wherein the packet is an Transmission Control Protocol (TCP) packet, and wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof.³

164. It is my opinion that Claim 17 is obvious over Lin in view of the knowledge of a POSA. EX1001, Col. 12:5-9.

165. Lin discloses embodiments in which the packet is a TCP packet. EX1005, Cols. 7:24-8:18, Tables 2-3. Further, it would have been well-known to a POSA that flag bits such as a SYN flag bit, an ACK flag bit, a FIN flag bit, and a RST flag bit were 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). As such, it would have been obvious to a POSA that the flow rules of Lin could have used a determination of whether or not one or more of these flag bits was set as one of Lin’s “conditions” that serve as packet-applicable criterion. EX1005, Col. 4:23-31; see EX1005, Cols. 5:8-12, 7:24-8:18.

³ Claim 17 uses the language “comprises comprise,” which I assume is an error. For the purposes of my analysis, I have interpreted that claim language to mean “comprises.”

166. Accordingly, it is my opinion that Lin in view of the knowledge of a POSA discloses Claims 17.

p. **Claim 18**

[18] The method according to claim 1, wherein the packet comprises distinct header and payload fields, the header comprises at least the first and second entities addresses in the packet network, and wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses.

167. It is my opinion that Claim 18 is obvious over Lin in view of the knowledge of a POSA. EX1001, Col. 12:10-16.

168. It would have been well-known to any POSA as of the priority date of the '111 Patent that a packet includes distinct header and payload fields. See EX1008, Section 2. Moreover, any POSA would have known as of the priority date for the '111 Patent that the header field could include at least the addresses of the first entity that was the source node for a packet and the second entity that is the destination node for the packet. It would have been well-known to a POSA that this could include the IP address of the source of the packet and the IP address of the destination of the packet.

169. Further, as disclosed above for Element [1.1], Lin provides an example where the predetermined criterion is that the address of the first entity, the address of the second entity, or both match a predetermined address or addresses. In discussing Table 1, Lin states as follows:

A flow table may include columns that indicate one or more conditions, a column that indicates an action to take when the conditions are met, and a column for statistics. A row on the flow table may comprise a flow rule. In the example of Table 1, the “Action” column indicates statistics, such as byte count. The rest of the columns of Table 1 indicate conditions. For example, “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), “IPsrc” (Internet Protocol (IP) address of the source of the packet), “IP dst” (IP address of the destination of the packet), etc. are conditions that identify a particular packet. 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.

TABLE 1							
IN_PORT	MAC src	MAC dst	IP src	IP dst	...	Action	Count
Ingress_port_ID	*	*	*	*	*	Redirect port	10
Redirect_port_ID	*	*	*	*	*	Ingress port	10

Table 1 of Lin

EX1005, Col. 5:8-25 (emphasis added); see EX1005, Col. 5:26-36, 6:1-12, 6:40-54, Table 1. A POSA would have understood this passage of Lin to explain that when the IP or MAC addresses are packet-applicable criterion used to determine whether conditions are met, such as whether they match a predetermined address or addresses.

170. Accordingly, it is my opinion that Lin in view of the knowledge of a POSA discloses Claim 18.

q. **Claim 19**

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

171. It is my opinion that Claim 19 is obvious over Lin. EX1001, Col. 12:17-18. Lin discloses Claim 19 for the same reasons as those discussed above for Claim 18. EX1005, Cols. 5:8-36, 6:1-12, 6:40-54, Table 1.

r. **Claim 20**

[20] The method according to claim 1, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and a TCP sequence mask fields, and wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any combination thereof, matches a predetermined value or values.

172. It is my opinion that Claim 20 is obvious over Lin in view of the knowledge of a POSA. EX1001, Col. 12:20-27.

173. Lin provides specific examples of TCP packets that are transported over a TCP network when discussing Tables 2 and 3 of that patent. EX1005, Cols. 7:24-8:18; see EX1005, Col. 10:48-50. Further, Lin discloses tracking HTTP packets whose source and destination TCP ports are port 80. EX1005, Cols. 7:24-8:18. Lin does not specifically disclose that a TCP packet has a TCP sequence number or a TCP sequence mask field, but this would have been obvious to a POSA because the TCP sequence number and TCP sequence mask field are always present in any TCP segment/packet.

174. Further, Lin discloses 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. For example, Lin's discussion of Table 3 teaches that the devices check for a specific packet-applicable criterion – that port 80 is the source or destination port of the TCP packets, which indicates that they are HTTP packets. EX1005, Cols. 7:64-8:18. Lin states that, if a packet satisfies this criterion (i.e., the packet indicates that port 80 is the source or destination port), then the SDN switch (which corresponds to the claimed network node) sends the packet over the packet network to the security device (which corresponds to the claimed entity that is included in the instruction and is other than the second entity). EX1005, Col. 8:10-18. Specifically, Lin provides the following discussion for the operations shown in Table 3 (which is reproduced below in annotated form):

In the example of Table 3, the top two rows are redirect flow rules for redirecting HTTP packets to the security service 630 for inspection, while the bottom two rows [highlighted below] are bypass flow rules for all packets. Because the redirect flow rules are at higher priority than the bypass flow rules, HTTP packets are sent through the SDN pipe formed in the SDN switch 620 between the sender component 622 and the security service 630. All other packets bypass the SDN pipe, and are accordingly not inspected by the security service 630.

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

Table 3 of Lin (Annotated)

EX1005, Col. 8:10-18 (emphasis added), Table 3.

175. Accordingly, it is my opinion that Lin in view of the knowledge of a POSA, discloses Claim 20.

s. Claim 21

[21] The method according to claim 1, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet, Metropolitan Area Network (MAN), Internet Service Provider (ISP) backbone, datacenter network, or inter-datacenter network.

176. It is my opinion that Claim 21 is obvious over Lin in view of the knowledge of a POSA. EX1001, Col. 12:28-32.

177. As discussed above for Claims 18-19, Lin discloses the use of IP and MAC addresses to identify particular packets. EX1005, Cols. 5:8-36, 6:1-12, 6:40-54, Table 1. A POSA would have understood that, if IP and MAC addresses were being used in Lin, then the packet network would have included one of the Internet, a Wide Area Network (“WAN”), a Local Area Network (“LAN”), a Metropolitan

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

178. Accordingly, it is my opinion that Lin discloses Claim 21.

t. **Claim 22**

[22] The method according to claim 1, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.

179. It is my opinion that Claim 22 is obvious over Swenson. EX1001, Col. 12:33-36.

180. Swenson teaches “user devices 110” on one end of its system and an “origin server 160” on the other end of its system. EX1007, ¶[0023]. Swenson discloses that the user devices “are computing devices with network capabilities,” such as “laptops, notebooks, tablets, smart telephones, or personal digital assistants (PDAs).” EX1007, ¶[0025]. A POSA would have understood these to be client devices. Given that Paragraph 58 of Swenson discloses two way communication between the user devices and the origin server, Swenson identifies user devices 110 (i.e., client devices) as the claimed first entity, and origin server 160 (i.e., a server) as the claimed second entity, or vice versa. EX1007, ¶[0058].

181. Accordingly, it is my opinion that Swenson discloses Claim 22.

u. **Claim 23**

[23] The method according to claim 22, wherein the server device comprises a web server, and wherein the client device comprises a smartphone, a tablet

computer, a personal computer, a laptop computer, or a wearable computing device.

182. It is my opinion that Claim 23 is obvious over Swenson. EX1001, Col. 12:37-41.

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

184. Accordingly, it is my opinion that Swenson discloses Claim 23.

v. Claim 24

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

185. It is my opinion that Claim 24 is obvious in view of Lin. EX1001, Col. 12:42-44.

186. Lin discloses the use of the OpenFlow protocol for the communication between the SDN switch (i.e., the network node) and the SDN controller. EX1005, Cols. 1:18-43, 2:12-13, 3:42-46, 3:65-4:6, 6:35-39. Figure 1 of Lin (shown below) describes the use of OpenFlow protocol as “prior art” to Lin, which is consistent

with my opinion that OpenFlow protocol was a standard protocol known long before the priority date for the '111 Patent.

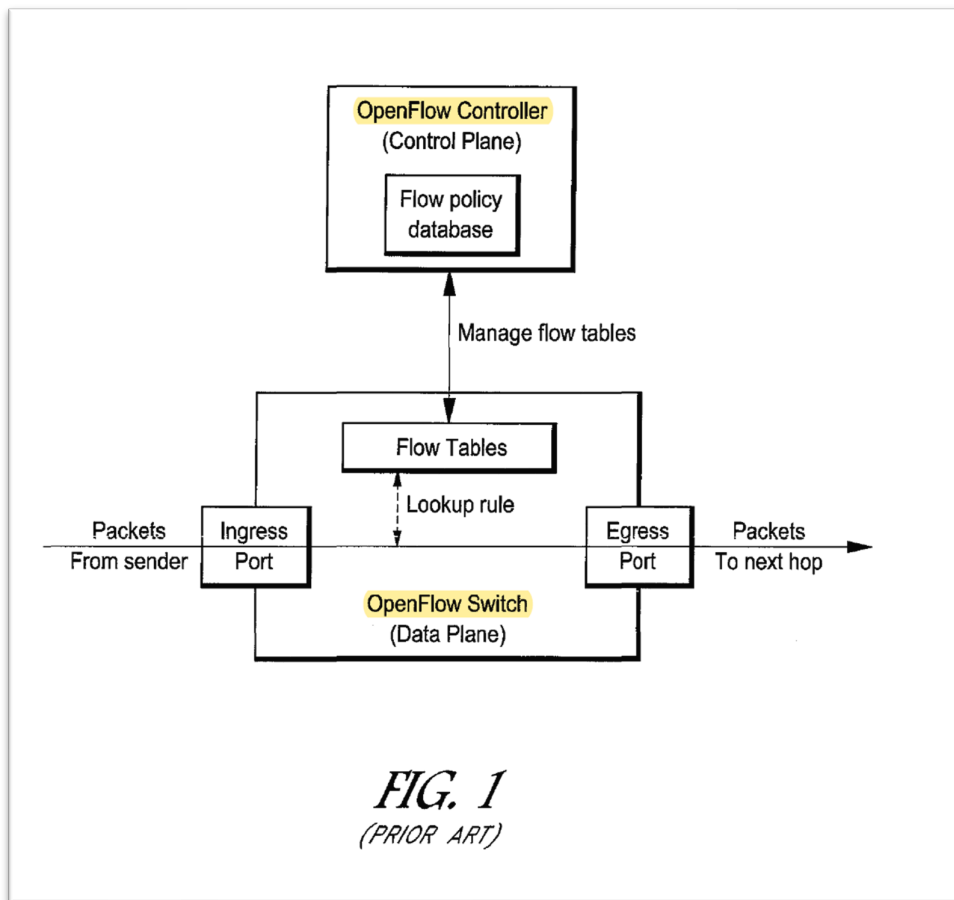


Figure 1 of Lin (Annotated)

EX1005, Figure 1.

187. Accordingly, it is my opinion that Lin discloses Claim 24.

w. **Claim 27**

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

188. It is my opinion that Claim 27 is obvious in view of Lin. EX1001, Col. 12:51-52.

189. As discussed above for Element [1.0], Lin discloses a SDN switch that operates as the claimed network node. EX1005, Cols. 1:58-2:4, 4:33-67, 6:13-23, 6:57-63.

190. Accordingly, it is my opinion that Lin discloses Claim 27.

x. **Claim 28**

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

191. It is my opinion that Claim 28 is obvious in view of Lin. EX1001, Col. 12:53-55.

192. As discussed above for Claims 18-19, Lin discloses the use of IP addresses to identify packets being transmitted across the packet network. EX1005, Cols. 5:8-36, 6:1-12, 6:40-54, Table 1. A POSA would have understood from the use of the IP addresses that packet network in Lin is an Internet Protocol (IP) network and the packet is an IP packet.

193. Accordingly, it is my opinion that Lin discloses Claim 28.

y. **Claim 29**

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

194. It is my opinion that Claim 29 is obvious in view of Lin. EX1001, Col. 12:56-58.

195. Lin provides specific examples of TCP packets that are transported over a TCP network when discussing Tables 2 and 3 of that patent. EX1005, Cols. 7:24-8:18; see EX1005, Col. 10:48-50.

196. Accordingly, it is my opinion that Lin discloses Claim 29.

z. Claim 30

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

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

197. It is my opinion that Element [30.0] is obvious in view of Lin. EX1001, Col. 12:59-63.

198. Lin teaches that its method can be applied to “any packet received by the SDN switch.” EX1005, Col. 6:40-63. As such, Lin would check if one or more additional packets after the first packet satisfies the packet-applicable criterion for the reasons discussed above for Elements [1.4]-[1.6].

199. Accordingly, it is my opinion that Lin discloses Element [30.0].

[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.

200. It is my opinion that the Element [30.1] is obvious in view of Lin. EX1001, Col. 12:64-13:3.

201. Lin discloses that the additional packets are sent by the SDN switch (i.e., the claimed network node) over the packet network to the destination node (i.e., the second entity) in response to the packets not satisfying the packet-applicable criterion for the reasons discussed above for Element [1.5].

202. Further, Lin discloses that the additional packets are sent by the SDN switch over the packet network in response to the packets not satisfying the packet-applicable criterion and in response to the instruction for the reasons discussed above for Element [1.6].

203. Accordingly, it is my opinion that Lin discloses Element [30.1].

aa. Claim 31

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

204. It is my opinion that Claim 31 is obvious in view of Lin. EX1001, Col. 13:4-7.

205. Lin explains that its packet network is a SDN network that uses an SDN controller and an SDN switch. EX1005, Abstract, Cols. 1:7-9, 1:58-2:4, 3:53-4:67, Figures 2, 6-8. 2, 6-8. Lin also discloses that it uses an Openflow protocol in which a packet is routed as part of the data plane through its SDN network. EX1005, Cols.

1:11-54, 2:49-50, 3:42-52, Figure 1. Further, Lin discloses that, when using the Openflow protocol, the communication between the SDN switch (i.e., the network node) and the SDN controller is through a control plane. EX1005, Cols. 1:11-54, 2:49-50, 3:42-52, 3:65-4:5, Figure 1.

206. Accordingly, it is my opinion that Lin discloses and renders obvious Claim 31.

B. Ground 2: Claims 1, 5-9, 12-24 and 27-30 of the '111 Patent are obvious over Shieh in view of Swenson.

207. It is my opinion that a POSA would have found Claims 1, 5-9, 12-24 and 27-30 of the '111 Patent to be obvious over Shieh in view of Swenson.

a. Claim 1

[1.0] A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method comprising:

208. To the extent that Element [1.0] is limiting, it is my opinion that Element [1.0] is disclosed or rendered obvious by the combination of Shieh and Swenson. EX1001, Col. 10:52-55.

209. Shieh discloses a method for use with a packet network. 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." EX1006, Abstract. The specification of Shieh further explains that its system relates to "network security" (EX1006, ¶[0002]) and performs operations on packets passing through the network system:

In one embodiment, the mechanism includes two functions: 1) an input/output (IO) function that performs the **distribution of network traffic**; and 2) a security-processing function that performs security processing, including security inspection and policy enforcement. The IO function receives the **packets** and uses a session table to forward the **packets** to the security-processing function. A session table is a data structure that stores connection states, including the destination of the security-processing function. In one embodiment, the IO function determines, based on an internal data structure such as a session or flow table, whether the **packet** should be forwarded to the security processing function for security inspection. The configuration of the IO function to control whether to forward the **packets** to the security processing function can be set based on a command received from an administrator or alternatively, based on a signal received from the security processing function.

EX1006, ¶[0017] (emphasis added); see EX1006, ¶[0018], ¶[0021], ¶[0023].

Further, Figure 7 of Shieh, shown below, shows the method disclosed in Shieh, which involves a “network access device” performing operations on packets passing through that network access device. EX1006, ¶[0049], Figure 7.

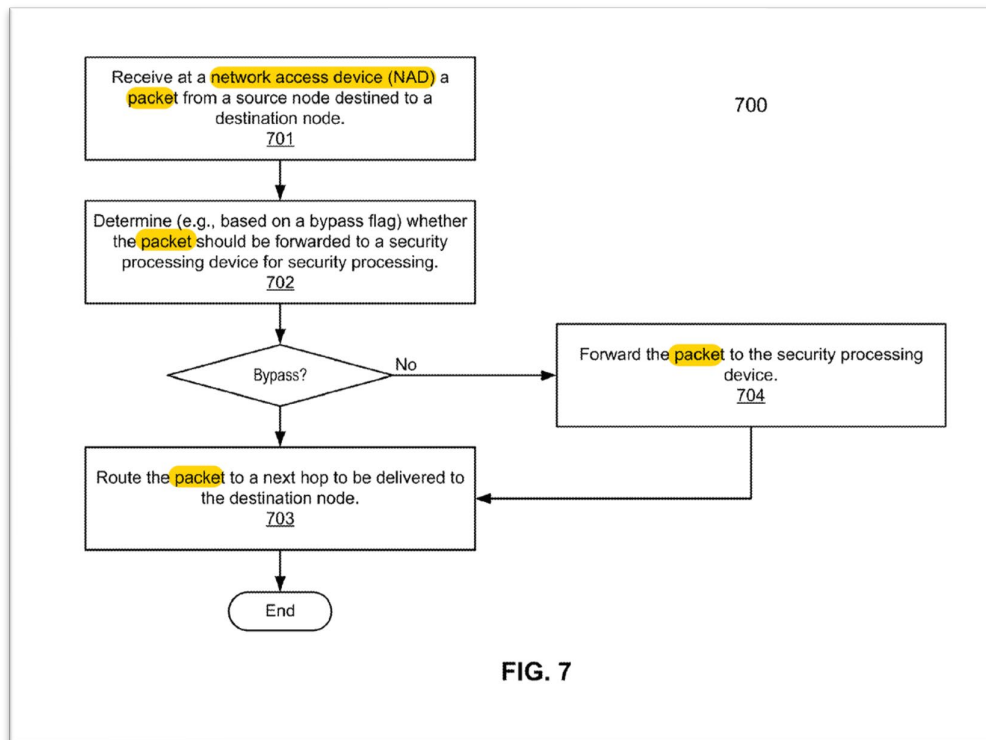


Figure 7 of Shieh (Annotated)

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

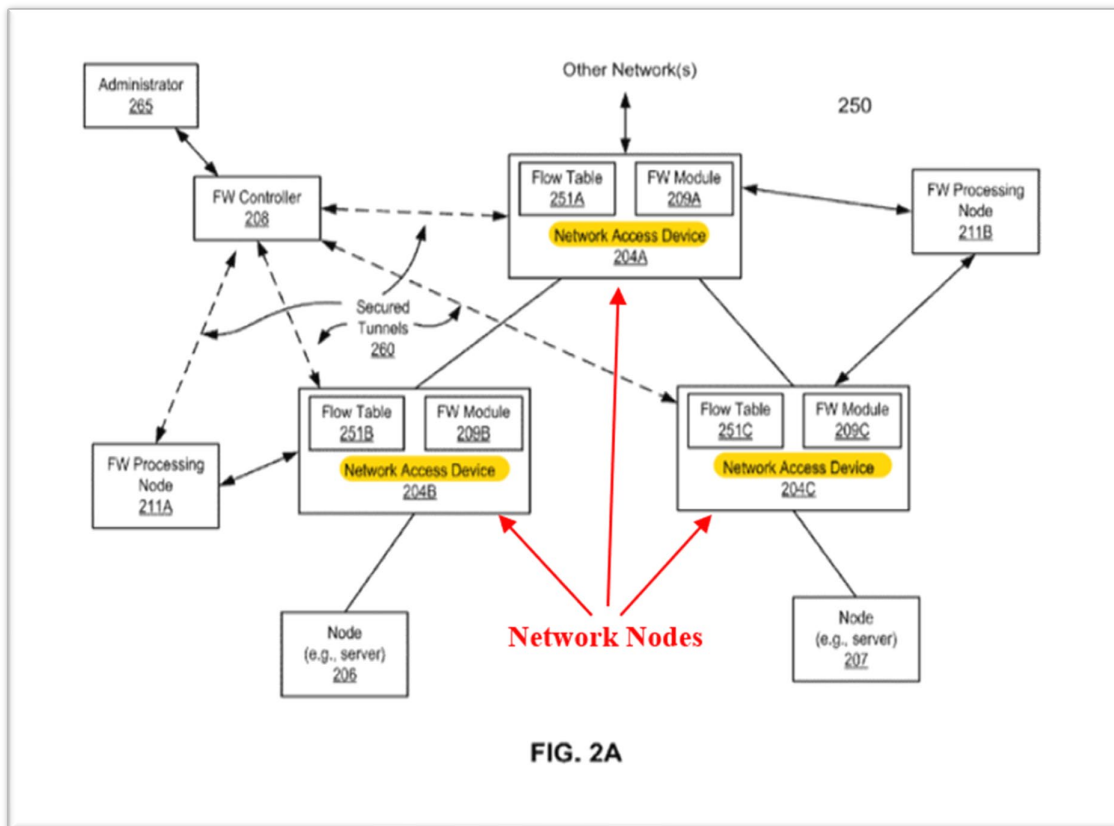


Figure 2A of Shieh (Annotated)

Further, Paragraph [0037] of Shieh states that “network access devices 204A-204C” (each of which corresponds to the claimed network node) are used to transport a packet from a “source node 271” (which corresponds to the claimed first entity) to a “destination node 274” (which corresponds to the claimed second entity). EX1006, ¶[0027], Figure 2B. The method shown in Figure 7 of Shieh also states that “a network access device receives a packet from a source node destined to a destination node.” EX1006, ¶[0049], Figure 7.

211. Shieh also discloses that the network nodes (e.g., network access devices 204A-204C) are under the control of a controller that is external to the network node. Shieh explains that network access devices 204A-204C are “configured” by (i.e., under the control of) “a controller or a management entity.” EX1006, ¶[0018]; see EX1006, ¶[0021] (noting that firewall controller 208 may be “external to network access device 204”), ¶[0025], ¶[0029]. In addition, Shieh discloses that the controller 208 is external to the network node, as shown below in Figure 2A:

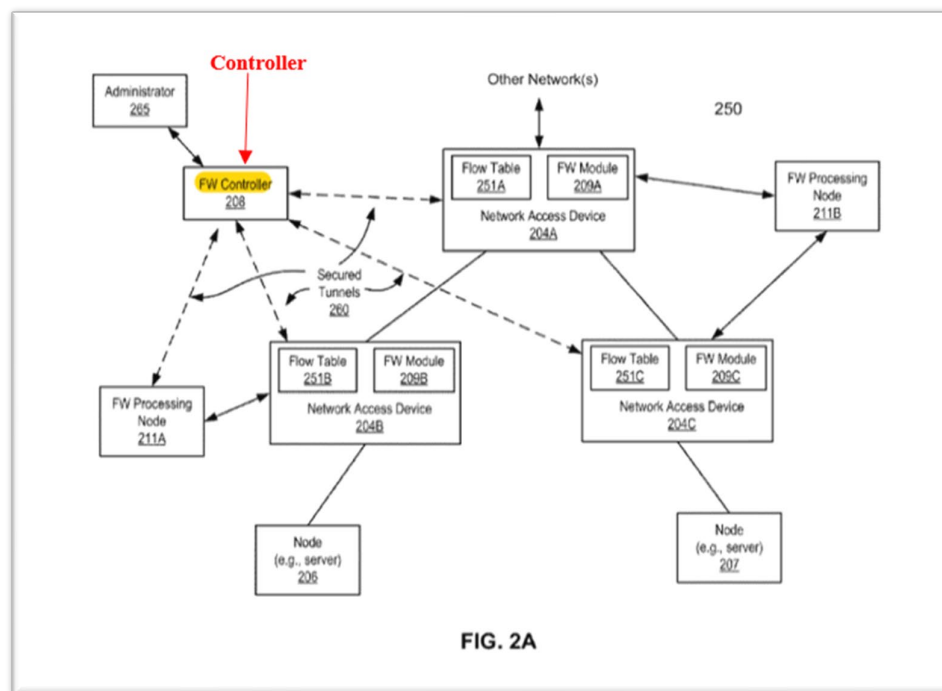


Figure 2A of Shieh (Annotated)

EX2005, Figure 2A, ¶[0029]; see EX2005, ¶[0025]-¶[0026].

212. Further, as discussed above with respect to claim construction, it is my opinion that the claimed controller is an entity configured to perform DPI on packets.⁴ Here, the combination of Shieh and Swenson discloses a controller configured to perform DPI. Shieh discloses that its security processing function includes DPI. EX1006, ¶[0021]. Shieh also discloses an embodiment in which packets are sent to a central controller in order to determine whether to forward the packets to a security processing function implementing the DPI. EX1006, ¶[0028]. If the packets are sent to the controller, it would have been obvious to a POSA that an efficient system would be to have the security processing function implemented within the controller in order to perform the DPI on the packets while they are within the controller. EX1006, ¶[0021], ¶[0028], ¶[0035]-¶[0037], ¶[0039]-¶[0041].

213. Moreover, Swenson teaches the use of a controller configured to perform DPI. Swenson discloses that, when its systems detect a HTTP packet flow matching a particular signature, “the steering device 130 forwards the HTTP request and a portion of the HTTP response to the network controller 140 over the ICAP client interface 404.” EX1007, ¶[0059]; see EX1007, ¶[0060]. “After receiving the request and the portion of response at the ICAP server interface 406, **the flow**

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

analyzer 312 of the network controller 140 performs a deep flow inspection to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059] (emphasis added); see EX1007, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B. As discussed above in Ground 1, a POSA would have understood that a “flow” is a series of packets having a specific signature, and thus Swenson’s reference to “deep flow inspection” refers to performing DPI on more than one or all of the packets in a flow.

214. Further, as discussed in Ground 1, Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶[0065], Figure 5. A POSA would have understood the analysis of the “response payload” to be DPI by a DPI-capable controller, as DPI refers to monitoring the payload of a packet. The above-discussed disclosures in Swenson demonstrate that it would have been well-known to a POSA as of the priority date for the ’111 Patent to implement a controller configured to perform DPI analysis in a system such as Lin. EX1007, ¶[0046], ¶[0059]-¶[0060], ¶[0073], ¶[0076]-[0077], ¶[0084]-¶[0086], Figures 1 and 4A-4B.

215. A POSA would have been motivated to combine the teachings of Shieh and Swenson for similar reasons as those disclosed above in Ground 1 for the combination of Lin and Swenson. For example, Shieh and Swenson address the same technology and attempt to resolve the same issues relating to routing network traffic in an efficient manner so as to conserve network resources. EX1006, ¶¶[0017]-¶¶[0018], ¶¶[0024]-¶¶[0025], ¶¶[0029]-¶¶[0030], Figures 1-3, 7; EX1007, ¶¶[0023]-¶¶[0032], ¶¶[0038]-¶¶[0043], ¶¶[0057]-¶¶[0061], Figures 1-4A. Both Shieh and Swenson 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. EX1006, ¶¶[0017]-¶¶[0018], ¶¶[0024]-¶¶[0025], ¶¶[0029]-¶¶[0030], Figures 1-3, 7; EX1007, ¶¶[0023]-¶¶[0032], ¶¶[0038]-¶¶[0043], ¶¶[0057]-¶¶[0061], Figures 1-4A. For example, Shieh discloses a method in which a controller sends instructions to network access devices through which packet streams pass, and those instructions can be used to redirect certain packets to a security processing device. EX1006, ¶¶[0017]-¶¶[0018], ¶¶[0024]-¶¶[0025], ¶¶[0029]-¶¶[0030], Figures 1-3. Further, Shieh teaches that packets can be routed or communicated to a destination node after a security processing device completes its analysis. EX1006, ¶¶[0037], Claim 1, Figure 7. In addition, Shieh discloses the use of bypass rules that determine if a packet should be delivered to a destination node

without having been sent to the security processing device. EX1006, ¶¶0035]-¶¶0037], ¶¶0049], Figure 5.

216. Similarly, Swenson discloses a method in which a network controller sends instructions to steering devices through which packet flows pass. EX1007, EX1007, ¶¶0023]-¶¶0032], ¶¶0038]-¶¶0043], ¶¶0057]-¶¶0061], Figures 1-4A. Swenson states that, when its systems detect a HTTP packet flow matching a particular signature, “the steering device 130 forwards the HTTP request and a portion of the HTTP response to the network controller 140 over the ICAP client interface 404.” EX1007, ¶¶0059]; see EX1007, ¶¶0060]. Swenson explains that only certain (or all) packets are redirected to the controller in order to be efficient in analyzing packets of interest, which is similar to the operation of Shieh’s bypass rules. EX1007, ¶¶0059]-¶¶0060]. Further, a POSA would have understood that Swenson could involve analyzing packets for a security function, like Shieh. A POSA would have understood that the bandwidth monitoring in Swenson can be used as a security application that monitors for DOS attacks that occupy significant bandwidth in a network. EX1007, ¶¶0059]-¶¶0060]; see EX1007, ¶¶0039] (explaining that the controller can incorporate “security functions”). Like Shieh, Swenson also teaches that packets can be routed or communicated to a destination node after the processing of those packets is completed. EX1007, ¶¶0060]. Further, the architecture of Swenson is substantially similar to the architecture of Shieh, as

can be seen in the below two figures showing a central controller directly connected to a network node (the Network Access Devices in Shieh and the steering device in Swenson) through which packet flows pass:

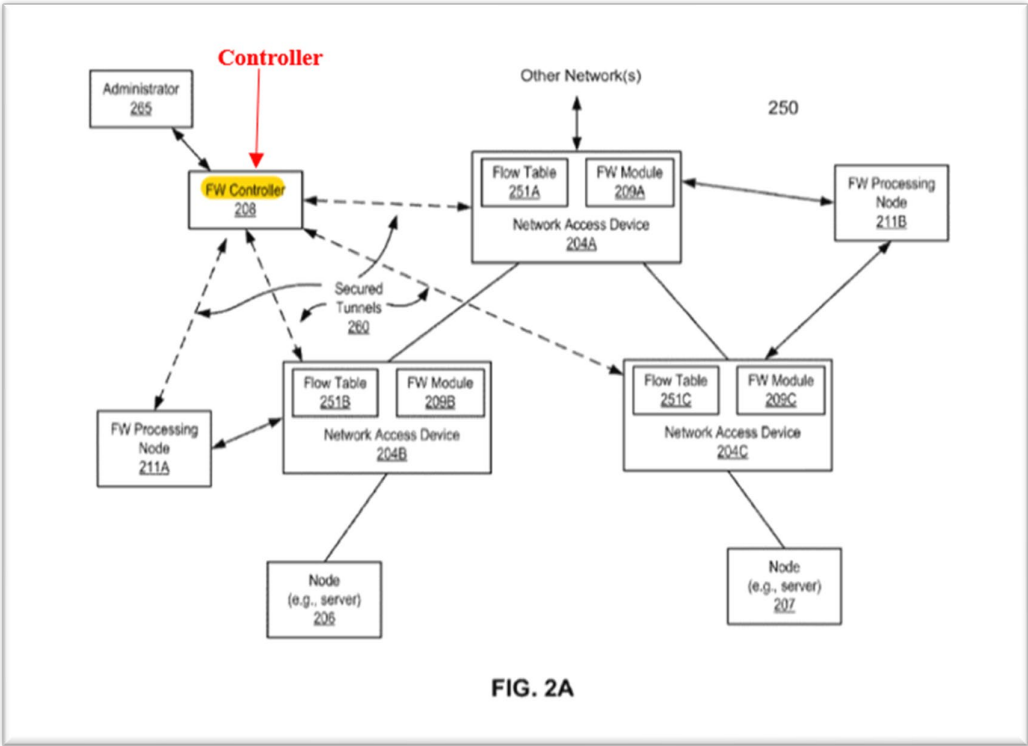


Figure 2A of Shieh (Annotated)

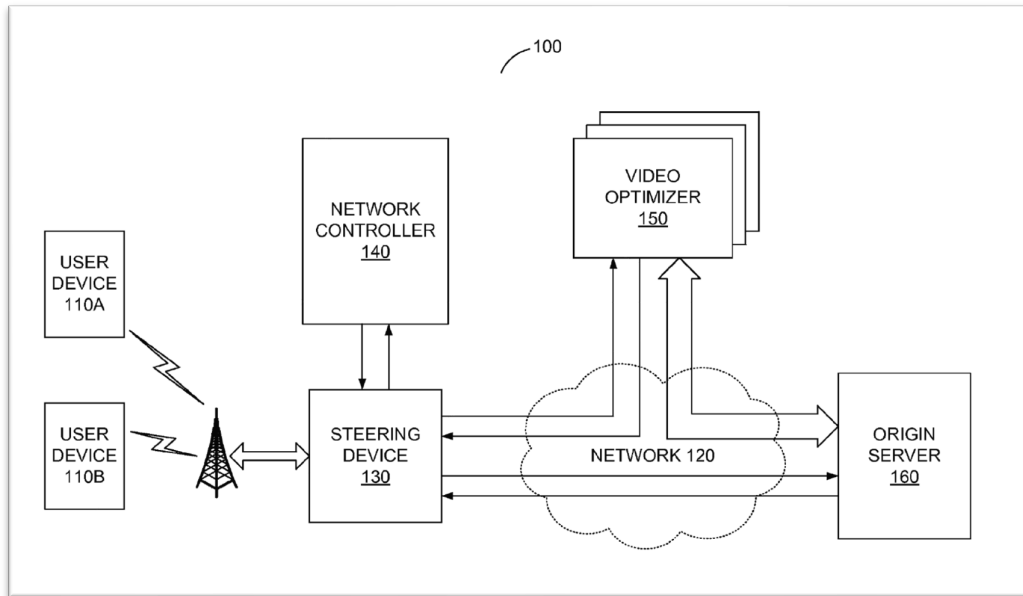


Figure 1 of Swenson

217. The determination of which packets to redirect (as opposed to redirecting all packets) increases the efficient operation of a computer network. Shieh and Swenson each teach that only certain packets may need to be redirected for further processing before being sent to the destination node. EX1006, ¶[0017]-¶[0018], ¶[0024]-¶[0025], ¶[0029]-¶[0030], Figures 1-3, 7; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A. This allows the methods in Shieh 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.

218. Further, a POSA would have found it obvious to implement the security processing module in Shieh as part of the controller in light of the disclosures in Swenson to send packets to the controller for DPI. EX1007, ¶¶[0059]-¶[0060], Figures 1, 4A-4B. A POSA would have found it efficient to implement Shieh's security processing module as part of the controller. Packets, or portions of packets, sent from various network nodes would go to a central controller, and that controller determines that the packets need to be processed by a security component as part of the analysis of the packets. The controller would then forward the packets to a security module within the controller for inspection by the security component. This 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. Further, a POSA would have known that security algorithms in a security component are often updated via software updates as new threats are identified. A POSA would have understood 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.

219. Moreover, a POSA would have had a reasonable expectation of success in modifying Shieh to implement the disclosure of Swenson to route packets to the central controller for inspection. As discussed above, Shieh and Swenson have a similar architecture in which network nodes are controlled by a separate external

controller. EX1006, ¶[0017]-¶[0018], ¶[0024]-¶[0025], ¶[0029]-¶[0030], Figures 1-3, 7; EX1007, ¶[0023]-¶[0032], ¶[0038]-¶[0043], ¶[0057]-¶[0061], Figures 1-4A. If Swenson could be constructed to have the network nodes send packets to the central controller for security inspection, a POSA would have understood that the same arrangement could be implemented in Shieh. Indeed, it would have been common sense for a POSA to modify Shieh in this way after reading Swenson.

220. In view of the above, it is my opinion that the combination of Shieh and Swenson discloses and renders obvious Element [1.0].

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

221. It is my opinion that Element [1.1] is obvious in view of Shieh. EX1001, Col. 10:56-58.

222. As discussed above for Element [1.0], Shieh discloses an external controller (i.e., Controller 208) that controls network nodes (i.e., network access devices 204A-204C) in a packet network. EX1006, ¶[0018], ¶[0025]-¶[0026], ¶[0029].

223. As discussed above in the Claim Construction section, the claimed instruction includes a command to determine whether or not a packet requires

inspection.⁵ 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. Shieh states that “a persistent connection” exists over the packet network to allow for communication between the controller and the network access devices 204A-204C. EX1006, ¶[0025]; see EX1006, ¶[0018], ¶[0028]-¶[0029]. This connection is used to send a “command” (i.e., an instruction) from an administrator via the controller over the packet network 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”:

The configuration of the IO function to control whether to forward the packets to the security processing function can be set based on a **command received from an administrator** or alternatively, based on a signal received from the security processing function.

* * *

According to one embodiment, an administrator can configures for example, **via a controller** or a management entity, **a network access device to set up a set of filtering rules concerning whether and/or what types of packets should be forwarded to a security device** and which of the security devices for security inspection. In this embodiment, **the controller is configured** to manage multiple network access devices and/or multiple security devices.

⁵ To the extent that the PTAB does not agree with this construction, Shieh still discloses Element [1.1] for the reasons discussed herein.

EX1006, ¶[0017]-¶[0018] (emphasis added); see EX1006, ¶[0023] (“The configuration of IO function 209 to control whether to forward the packets to security processing function 210 can be set based on a command received from an administrator...”), ¶[0028]-¶[0029].

224. While Shieh does not specifically disclose that the controller sends a packet-applicable criterion to the network access devices 204A-204C over the packet network, it is my opinion that this would have been obvious in light of the disclosure of Shieh. As discussed above, Shieh sends a command from 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.” EX1006, ¶[0017]-¶[0018], ¶[0023], ¶[0028]-¶[0029]. A POSA would have understood that the network access device would use a packet-applicable criterion to determine which packets should be forwarded to the security device pursuant to the filtering rules. For example, Shieh discloses a bypass rule that “functions to enable packets to bypass security-processing function if there is no more need to inspect the packets of the connection.” EX1006, ¶[0035]; see ¶[0049]. Shieh also discloses criterion specific to a particular packet being checked for the bypass rule, such as the identification of “TCP FIN or TCP RST packets.” EX1006, ¶[0036]. Shieh does not state whether these criterion are sent from a controller to the network access device. However, it is my opinion that a POSA would have known that, in

SDN, the packet-applicable criterion would be sent from a controller to one or more network access devices. This allows the same packet-applicable criterion to be applied at multiple network nodes (i.e., network access devices).

225. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [1.1].

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

226. It is my opinion that Element [1.2] is obvious in view of Shieh. EX1001, Col. 10:59-60.

227. As discussed above for Elements [1.0] and [1.1], Shieh discloses an external controller (i.e., Controller 208) that controls network nodes (i.e., network access devices 204A-204C) in a packet network. EX1006, ¶[0018], ¶[0025]-¶[0026], ¶[0029].

228. Further, Shieh discloses an instruction sent by the controller and received by the network node over the packet network. Shieh states that “a persistent connection” exists over the packet network to allow the network access devices 204A-24C to receive communications from the controller:

229. According to one embodiment, each of network access devices 204A-204C maintains a **persistent connection** such as secure connections or tunnels 260 with a controller or management entity 208 for exchanging management messages

and configurations, or distributing routing information to network access devices 204A-204C, etc.

230.

EX1006, ¶[0025]; see EX1006, ¶[0018], ¶[0028]-¶[0029]. In addition, Shieh discloses that a “command” (i.e, an instruction) is received by the network access devices 204A-204C from the controller “to set up a set of filtering rules concerning whether and/or what types of packets should be forwarded to a security device”:

The configuration of the IO function to control whether to forward the packets to the security processing function can be set based on a **command received from an administrator** or alternatively, based on a signal received from the security processing function.

* * *

According to one embodiment, an administrator can configures for example, **via a controller** or a management entity, **a network access device to set up a set of filtering rules concerning whether and/or what types of packets should be forwarded to a security device** and which of the security devices for security inspection. In this embodiment, **the controller is configured** to manage multiple network access devices and/or multiple security devices.

EX1006, ¶[0017]-¶[0018] (emphasis added); see EX1006, ¶[0023] (“The configuration of IO function 209 to control whether to forward the packets to security processing function 210 can be set based on a command **received from an administrator...**”) (emphasis added), ¶[0028]-¶[0029].

231. While Shieh does not specifically disclose that the network access devices 204A-204C receive a packet-applicable criterion from the controller, it is

my opinion that this would have been obvious from the disclosure of Shieh for the same reasons as those stated above for Element [1.1]. As discussed above, Shieh teaches that network access devices 204A-204C receive a command from a controller “to set up a set of filtering rules concerning whether and/or what types of packets should be forwarded to a security device.” EX1006, ¶[0017]-¶[0018], ¶[0023], ¶[0028]-¶[0029]. A POSA would have understood that the network access devices would use a packet-applicable criterion to determine which packets should be forwarded to the security device pursuant to the filtering rules. A POSA also would have understood that, in SDN, the packet-applicable criterion would be received by one or more network access devices from the controller of Shieh. This allows the same packet-applicable criterion to be applied at multiple network nodes (i.e., network access devices).

232. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [1.2].

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

233. It is my opinion that Element [1.3] is obvious in view of Shieh. EX1001, Col. 10:60-62.

234. 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): “A computer-

implemented method, comprising: receiving at a network access device a packet from a source node destined to a destination node.” EX1005, Claim 1. The Abstract of Shieh also states that “[t]he network access device is to receive a packet from a source node destined to a destination node...” EX1006, Abstract. Further, Paragraph 27 of Shieh discloses that the method of Shieh is used to process a “network flow,” which is “a sequence of packets from a source computer to a destination, which may be another host, a multicast group, or a broadcast domain.” EX1006, ¶[0027]; see EX1006, ¶[0020], ¶[0037], ¶[0049]. Shieh explains that the packets may be addressed with, among other things, “Source and Destination IP address,” which are used to address the source computer (i.e., the claimed first entity) and the destination (i.e., the claimed second entity). EX1006, ¶[0027]. In addition, Figure 1 of Shieh, reproduced below, shows a system in which “Network Access Device(s) 204” (i.e., the claimed network node) receive packets from client 201 or client 202 (which correspond to the claimed first entity) that are addressed to server 206 or server 207 (which correspond to the claimed second entity):

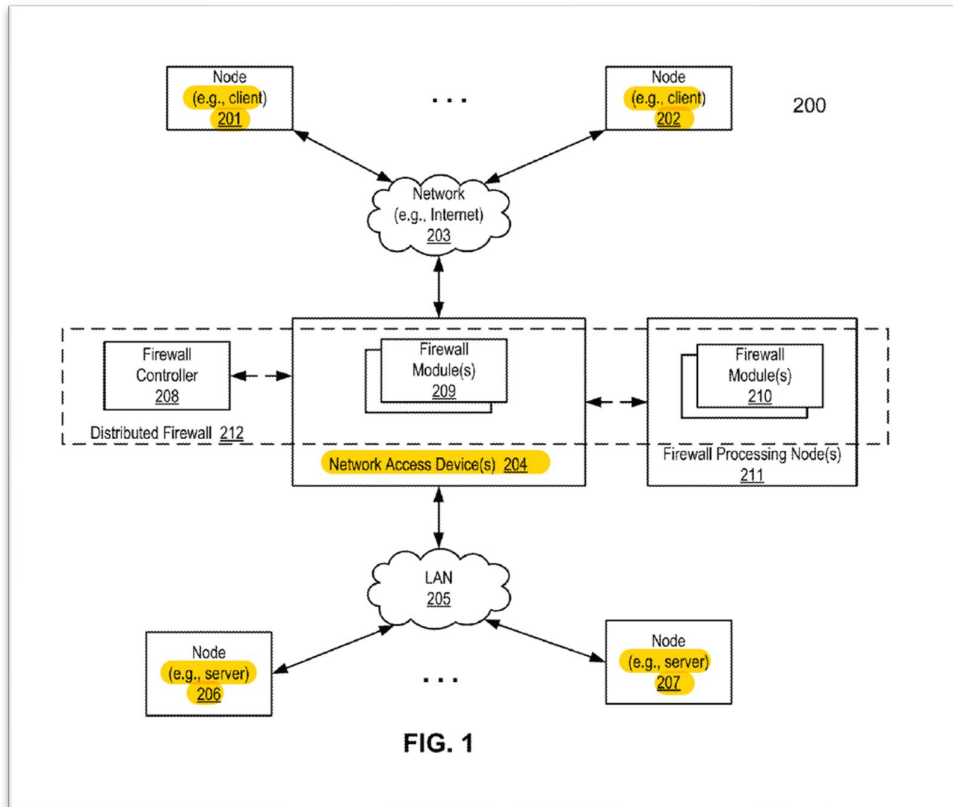


Figure 1 of Shieh (Annotated)

EX1006, Figure 1; see EX1006, ¶[0049] (discussing Figure 7 and stating that “a network access device receives a packet from a source node destined to a destination node”), Figures 2A-2B, 7.

235. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [1.3].

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

236. It is my opinion that Element [1.4] is obvious in view of Shieh. EX1001, Col. 10:63-64.

237. As discussed above, Shieh discloses a network access device (i.e., the claimed network node) that uses “filtering rules concerning whether and/or what types of packets should be forwarded to a security device”:

According to one embodiment, an administrator can configure, for example, via a controller or a management entity, a network access device to set up a set of **filtering rules concerning whether and/or what types of packets should be forwarded to a security device** and which of the security devices for security inspection. In this embodiment, the controller is configured to manage multiple network access devices and/or multiple security devices.

EX1006, ¶[0018] (emphasis added); see EX1006, ¶[0023], ¶[0028]-¶[0029]. A POSA would have understood that the network access device is checking to see if an incoming packet satisfies the filtering rules and using that determination to decide whether to forward the packet to a security device. For example, and as discussed above, Shieh discloses a bypass rule that “functions to enable packets to bypass security-processing function if there is no more need to inspect the packets of the connection.” EX1006, ¶[0035]; see ¶[0049]. Shieh also discloses criterion specific to a particular packet being checked by the network access device for the bypass rule, such as the identification of “TCP FIN or TCP RST packets.” EX1006, ¶[0036]; see EX1006, ¶[0037] (“When device 272 receives a packet from a source node 271 via transaction 281, device 272 may determine whether the packet should be forwarded to security device 273.”), Figure 7 (method determines or checks whether

a packet should be forwarded to a security processing device “based on a bypass flag”).

238. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [1.4].

[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

239. It is my opinion that Element [1.5] is obvious in view of Shieh. EX1001, Col. 10:65-67.

240. As discussed above for Element [1.4], it is my opinion that Shieh teaches that its network access device is checking to see if an incoming packet satisfies its filtering rules and using that determination to decide whether to forward the packet to the destination node (which corresponds to the claimed second entity) or to a security device. EX1006, ¶[0018], ¶[0023], ¶[0028]-¶[0029]. In particular, Shieh discloses a bypass rule that “functions to enable packets to bypass security-processing function if there is no more need to inspect the packets of the connection.” EX1006, ¶[0035]; see EX1006, ¶[0049]. Shieh also discloses criterion specific to a particular packet being checked by the network access device applying the bypass rule. EX1006, ¶[0035], Figure 7. Shieh may check whether the packets are “TCP FIN or TCP RST packets to determine whether to bypass the security function. EX1006, ¶[0036].

241. Further, the method disclosed in Shieh checks whether a packet should be forwarded to a security processing device “based on a bypass flag.” See EX1006, Figure 7. Paragraph 37 of Shieh explains that the network access device “determine[s] whether a bypass flag has been set to a predetermined value.” EX1006, ¶[0037]. If the packet does not satisfy this “bypass flag” criterion, then the packet is sent by the network access device over the packet network to the “destination node” (which corresponds to the claimed second entity):

For example, device 272 may look up in its session table such as the one as shown in F[igure] 5 to determine whether a **bypass flag has been set to a predetermined value**. If the bypass flag matches the predetermined value, the packet is forwarded to security device 273 via path 282; **otherwise, the packet is routed to destination node 274**.

EX1006, ¶[0037] (emphasis added), Figures 5, 7.

242. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [1.5].

[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.

243. It is my opinion that Element [1.6] is obvious in view of Shieh. EX1001, Col. 11:1-4.

244. Shieh teaches that its network access device is checking to see if an incoming packet satisfies its filtering rules and using that determination to decide to forward the packet to a security device. EX1006, ¶[0018], ¶[0023], ¶[0028]-¶[0029].

Shieh also teaches that its bypass rule utilizes packet-applicable criterion, such as a bypass flag, to make this determination. EX1006, ¶[0035]-¶[0037], ¶[0049], Figure 7.

245. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [1.6].

b. Claim 5

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

246. It is my opinion that Claim 5 is obvious in view of the combination of Shieh and Swenson. EX1001, Col. 11:27-30.

247. As discussed above for Elements [1.1] and [1.2], Shieh discloses that a network node has an instruction and packet-applicable criterion. Further, as discussed above for Elements [1.4]-[1.6], Shieh discloses that an action is taken responsive to the instruction and to the packet satisfying the packet-applicable criterion.

248. Moreover, as discussed above in Ground 1, Swenson teaches that the network node sends a packet or a portion thereof to the controller in response to the packet satisfying a criterion. As discussed above for Element [1.0], Swenson discloses that, when its system detects a HTTP packet flow matching a particular signature, “the steering device 130 **forwards the HTTP request and a portion of**

the HTTP response to the network controller 140 over the ICAP client interface 404.” EX1007, ¶[0059] (emphasis added); see EX1007, ¶[0060], Figures 1, 4A-4B. A POSA would have understood that the steering device 130 corresponds to at least part of the claimed network node, and the signature that is compared to the HTTP packet flow corresponds to the claimed criterion. Responsive to the HTTP packet flow satisfying that criterion, the steering device applies an instruction pursuant to which the HTTP request and a portion of the HTTP response is sent to the network controller 140.

249. Accordingly, it is my opinion that the combination of Shieh and Swenson discloses and renders obvious Claim 5.

c. Claim 6

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

250. It is my opinion that Claim 6 is obvious over Swenson in view of the knowledge of a POSA. EX1001, Col. 11:31-33.

251. As discussed above in Ground 1, Swenson discloses a flow cache memory 322 that is part of the network controller 140. EX1007, Figure 3. Further, Swenson states that “[o]nce a flow is reported to the network controller 140, a flow cache entry is created for the flow in the flow cache 322.” EX1007, ¶[0061]. Further, Swenson states that “[t]he flow cache entry keeps track of the flow and its associated bandwidth.” EX1007, ¶[0061]; see EX1007, ¶[0040], ¶[0046], ¶[0062]. In discussing

Figure 7, Swenson explains that the flow cache entry stores at least a portion of the received packet, including at least the IP address of the received packet. EX1007, ¶[0073] (discussing Figure 7); see EX1007, ¶[0074]-¶[0085]. Further, Swenson states that a flow cache entry can also store the MAC address or TCP source port associated with a received packet. EX1007, ¶[0084]. A POSA would have understood from these disclosures that Swenson teaches that the controller stores at least a portion of a packet in a flow cache memory.

252. Accordingly, it is my opinion that Swenson in view of the knowledge of a POSA discloses and renders obvious Claim 6.

d. Claim 7

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

253. It is my opinion that Claim 7 is obvious over Swenson. EX1001, Col. 11:34-37.

254. As discussed above for Ground 1, Swenson explains that the network node sends a portion of a packet to the controller in response to the packet satisfying a criterion. Swenson discloses that, when its system detects a HTTP packet flow matching a particular signature, “the steering device 130 **forwards the HTTP request and a portion of the HTTP response to the network controller 140** over the ICAP client interface 404.” EX1007, ¶[0059] (emphasis added); see EX1007,

¶[0060], Figures 1, 4A-4B. A POSA would have understood 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. Further, a POSA would have understood that, responsive to the HTTP packet flow satisfying that criterion, the steering device applies an instruction pursuant to which a portion of a HTTP request or a portion of a HTTP response is sent to the network controller.

255. Accordingly, it is my opinion that Swenson discloses and renders obvious Claim 7.

e. **Claim 8**

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

256. It is my opinion that Claim 8 is obvious over the combination of Shieh and Swenson, in view of the knowledge of a POSA. EX1001, Col. 11:38-41.

257. As discussed above for Claim 7, Swenson discloses sending a portion of a HTTP request or a HTTP response to a controller responsive to the packet satisfying a criterion and responsive to an instruction. EX1007, ¶[0059]-¶[0060], Figures 1, 4A-4B.

258. Further, it is my opinion that a POSA would have found it obvious from Shieh that the portion of the packet sent to the controller consists of multiple consecutive bytes and the instruction comprises identification of the consecutive

bytes in the packet. A POSA would have understood from Shieh that only certain portions of a packet need to be sent to the controller for analysis. EX1006, ¶[0035]-¶[0037], Figure 5. For example, Paragraph [0036] of Shieh states that its system may look for “TCP FIN or TCP RST packets” when applying Shieh’s bypass rule, and these packets would have been identified by examining whether a FIN flag bit was set or a RST flag was set. EX1006. Further, Paragraph [0037] of Shieh states that its system looks to “whether a bypass flag has been set to a predetermined value.” EX1006, ¶[0037], Figure 5. It would have been obvious to a POSA that the necessary portions of the packet may have been contained within consecutive bytes of the packet, such as the portion of a packet containing the header information where the FIN or RST bits would be located. Further, it would have been obvious to a POSA that the instruction would identify the consecutive bytes in the packet that need to be sent to the controller for analysis.

259. As such, it is my opinion that the combination of Shieh and Swenson, in view of the knowledge of a POSA, discloses and renders obvious Claim 8.

f. Claim 9

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

260. It is my opinion that Claim 9 is obvious in view of Swenson. EX1001, Col. 11:42-44.

261. As discussed above for Ground 1, 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 140 performs a deep flow inspection to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059]; see EX1007, ¶[0060] (stating that the “controller 140 ingests the network flow for inspection”), Figures 1, 4A-4B. Further, as explained above for Ground 1, Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶[0065], Figure 5. A POSA would have 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.

262. Accordingly, it is my opinion that Swenson discloses and renders obvious Claim 9.

g. Claim 12

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

263. It is my opinion that Claim 12 is obvious over the combination of Shieh and Swenson, along with the knowledge of a POSA. EX1001, Col. 11:55-57.

264. Shieh discloses that the analysis performed on a packet included a security processing function. EX1006, ¶[0002], ¶[0017]- ¶[0019], ¶[0021], ¶[0023], ¶[0029]- ¶[0031], ¶[0035]-¶[0037], ¶[0042], ¶[0049], Figures 1, 2B, 3, 7.

265. Further, Swenson discloses that packets are sent to the controller for DPI “to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶[0059]-¶[0060]. As discussed above for Element [1.0], a POSA would have understood that bandwidth monitoring can be used for applying a security or data analytic application the monitors for DOS attacks. EX1007, ¶[0059]-¶[0060]; see EX1007, ¶[0039] (explaining that the controller can incorporate “security functions.

266. As such, it is my opinion that both Shieh and Swenson disclose and render obvious Claim 12.

h. Claim 13

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

267. It is my opinion that Claim 13 is obvious over Shieh. EX1001, Col. 11:58-60.

268. Shieh discloses that its system analyzes packets by applying a security application that comprises a firewall functionality:

According to one embodiment, network access device 204 is associated with a **distributed firewall 212** that includes **various firewall**

processing modules, for example, each being executed within a virtual machine (VM). In one embodiment, each **firewall module** is responsible for performing one or more firewall functions, but it does not include all of the **firewall functions of a firewall**. Examples of the **firewall functions** include, but are not limited to, network address translation (NAT), virtual private network (VPN), deep packet inspection (DPI), and/or anti-virus, etc. In one embodiment, some of the **firewall processing modules** are located within network access device 204 (e.g., **firewall modules 209**) and some are located external to network access device 204 (e.g., **firewall modules 210** maintained by **firewall processing node(s) 211**, which may be a dedicated firewall processing machine. All of the **firewall modules 209-210** are managed by **firewall controller 208**, which may be located within network access device 204, or external to network access device 204, such as, for example, in a public cloud associated with network 203, or in a private cloud associated with network 205. **Controller 208 and firewall processing modules 209-210 collectively are referred to herein as distributed firewall 212.**

EX1006, ¶[0021] (emphasis added); see EX1006, ¶[0023], ¶[0027]-¶[0028], ¶[0038], ¶[0043]-¶[0049].

269. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 13.

i. **Claim 14**

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

270. It is my opinion that Claim 14 is obvious over Shieh and Swenson. EX1001, Col. 11:61-63.

271. Shieh discloses the use of DPI as part of its analysis functionality:

Examples of the firewall functions include, but are not limited to, network address translation (NAT), virtual private network (VPN), **deep packet inspection (DPI)**, and/or anti-virus, etc.

EX1006, ¶¶[0021] (emphasis added); see EX1006, ¶¶[0028], ¶¶[0040]-¶¶[0041].

272. As discussed above for Ground 1, Swenson also explains that “the flow analyzer 312 of the network controller 140 performs a deep flow inspection to determine if the flow is worth bandwidth monitoring and/or user detection.” EX1007, ¶¶[0059]. Further, as explained above for Ground 1, Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶¶[0065], Figure 5. A POSA would have understood the analysis of the “response payload” to be DPI.

273. Accordingly, it is my opinion that both Shieh and Swenson disclose and render obvious Claim 14.

j. **Claim 15**

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

274. It is my opinion that Element of Claim 15 is obvious in view of both Swenson and Shieh, in view of the knowledge of a POSA. EX1001, Col. 11:64-67.

275. It would have been well-known to any POSA as of the priority date of the '111 Patent that a packet includes distinct header and payload fields. EX1008, Section 2.

276. Swenson discusses “an example event trace of [Swenson’s] ‘continue’ working mode” in which the steering device 130 “sends an ICAP request message 516 comprising [a] HTTP GET request header and a portion of the [HTTP] response payload to the network controller 140, which inspects the message to determine whether to monitor the flow or optimize the video.” EX1007, ¶[0065], Figure 5. A POSA would have understood the analysis of the “response payload” by the Swenson system to be checking part of the payload field of a packet by the network controller.

277. Further, a POSA would have understood from the disclosures of Shieh that the packet analysis includes checking part or the whole of the payload field. As discussed above for Claim 14, Shieh discloses analysis of packets through DPI. EX1006, ¶[0021], ¶[0028], ¶[0040]-¶[0041]. A POSA would have understood that DPI refers to the inspection of at least part of the payload field.

278. In view of the above, it is my opinion that both Swenson and Shieh, in view of the knowledge of a POSA, disclose and render obvious Claim 15.

k. Claim 16

[16] The method according to claim 1, wherein the packet comprises distinct header and payload fields, the header comprises one or more flag bits, and

wherein the packet-applicable criterion is that one or more of the flag bits is set.

279. It is my opinion that the element of Claim 16 is obvious over Shieh in view of the knowledge of a POSA. EX1001, Col. 12:1-4.

280. As discussed above for Claim 16 in Ground 1, it would have been well-known to any POSA as of the priority date of the '111 Patent that a packet includes distinct header and payload fields. EX1008, Section 2. Shieh discusses the header fields in a packet and how they are used to route packets. EX1006, ¶[0044]. Further, Shieh states that the packet streams passing through its system include “TCP FIN or TCP RST packets,” both of which were well-known in the art before the priority date for the '111 Patent. EX1006, ¶[0036]. A POSA would have understood that a TCP FIN packet includes a FIN flag bit in its header that is set. Further, a POSA would have understood that a TCP RST packet includes a RST flag bit in its header that is set. It is my opinion that a POSA would have found it obvious that the TCP FIN flag bit being set or the TCP RST flag bit being set could function as the packet-applicable criterion recited in Claim 1. Either of these bits easily could have been used by a POSA to identify certain types of packets that do or do not require inspection by a security component. Indeed, Shieh states that TCP FIN or TCP RST packets “are special events in the packet stream” that serve the purpose of determining whether or not to send packets to a security component.

281. Accordingly, it is my opinion that Shieh in view of the knowledge of a POSA discloses Claims 16.

I. Claim 17

[17] The method according to claim 16, wherein the packet is an Transmission Control Protocol (TCP) packet, and wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof.

282. It is my opinion that the element of Claim 17 is obvious over Shieh in view of the knowledge of a POSA. EX1001, Col. 12:5-9.

283. Shieh states that the packet streams passing through its system include TCP packets. EX1006, ¶[0027], ¶[0036]. Further, Shieh discloses that some of the packets can be “TCP FIN or TCP RST packets,” both of which were well-known in the art before the priority date for the ’111 Patent. EX1006, ¶[0036]. A POSA would have understood that a TCP FIN packet includes a FIN flag bit in its header that is set. Further, a POSA would have understood that a TCP RST packet includes a RST flag bit in its header that is set. It is my understanding that the disclosure of the FIN flag bit and the RST flag bit is sufficient to render Claim 17 obvious, where the claim language is in the format of “A, B, C, D, or any combination thereof.” However, it is further my opinion that a POSA would have known that SYN and ACK were common flags in a TCP packet header that could have been used to identify packets in a packet stream.

284. Accordingly, it is my opinion that Shieh in view of the knowledge of a POSA discloses Claims 17.

m. Claim 18

[18] The method according to claim 1, wherein the packet comprises distinct header and payload fields, the header comprises at least the first and second entities addresses in the packet network, and wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses.

285. It is my opinion that the element of Claim 18 is obvious in view of Shieh. EX1001, Col. 12:10-16.

286. As discussed above for Claim 18 in Ground 1, it would have been well-known to any POSA as of the priority date of the '111 Patent that a packet includes distinct header and payload fields. EX1008, Section 2. Indeed, Shieh discusses the header fields in a packet and how they are used to route packets. EX1005, [0044]. Moreover, any POSA would have known as of the priority date for the '111 Patent that the header field could include at least the addresses of the first entity that was the source node for a packet and the second entity that is the destination node for the packet. It would have been well-known to a POSA that this could include the IP address of the source of the packet and the IP address of the destination of the packet.

287. 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. Shieh states the following:

According to one embodiment, referring back to FIG. 2A, when a security-processing function (e.g., processing node 211A) receives the packets, it does the security inspection and security policy enforcement. The packets then are forwarded to the next I/O function (e.g., modules 209A-209C). The choices of the next I/O function could be from the decision from layer 2 such as **Ethernet MAC address lookup, or IP address routing**, or other methods.

EX1006, ¶[0031] (emphasis added). Shieh also states that a “TCP/IP flow can be identified by the following parameters within a certain time period: 1) Source and Destination IP addresses; 2) Source and Destination Port [i.e., MAC address], and 3) Layer 4 Protocol (TCP/UDP/ICMP).” EX1006, ¶[0027]; see EX1006, ¶[0026]. A POSA would have understood from these passages in Shieh that the IP or MAC addresses are packet-applicable criterion compared to a predetermined address or addresses.

288. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 18.

n. **Claim 19**

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

289. It is my opinion that Claim 19 is obvious over Shieh. EX1001, Col. 12:17-18. Shieh discloses and renders obvious Claim 19 for the same reasons as those discussed above for Claim 18. EX1006, ¶[0027]; ¶[0031].

o. **Claim 20**

[20] The method according to claim 1, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and a TCP sequence mask fields, and wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any combination thereof, matches a predetermined value or values.

290. It is my opinion that the element of Claim 20 is obvious over Shieh in view of the knowledge of a POSA. EX1001, Col. 12:20-27.

291. Shieh describes tracking TCP packets and states that “a TCP/IP flow [i.e, a TCP packet] can be uniquely identified by the following parameters within a certain time period: 1) source and destination IP address; 2) **source and destination port**; and 3) Layer 4 Protocol (TCP/UDP/ICMP).” EX1006, ¶[0027] (emphasis added). Shieh does not specifically disclose that a TCP packet has a TCP sequence number or a TCP sequence mask field, but this would have been obvious to a POSA because the TCP sequence number and TCP sequence mask field are always present in any TCP segment/packet. In addition, Shieh discloses the use of a source TCP port or a destination TCP port as the packet applicable criterion used to determine whether to take an action on a packet. EX1006, ¶[0027], ¶[0036].

292. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 20.

p. **Claim 21**

[21] The method according to claim 1, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet,

Metropolitan Area Network (MAN), Internet Service Provider (ISP) backbone, datacenter network, or inter-datacenter network.

293. It is my opinion that the element of Claim 21 is obvious over Shieh. EX1001, Col. 12:28-32.

294. As discussed above for Claim 21 in Ground 1, Shieh states that its packet network includes the Internet, a WAN or a LAN: “Network 203 may be an external network such as a wide area network (WAN) (e.g., Internet) while network 205 represents a local area network (LAN).” EX1006, ¶[0020]. Shieh also discloses that its system can be in communication with an ISP. EX1006, ¶[0053].

295. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 21.

q. Claim 22

[22] The method according to claim 1, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.

296. It is my opinion that the element of Claim 22 is obvious over Shieh. EX1001, Col. 12:33-36.

297. As discussed above for Claim 22 in Ground 1, Figure 1 of Shieh (reproduced below) shows that the source of the network packets (i.e., the claimed first entity) can be a server device (Nodes 206 or 207), and the destination of the claimed packets (i.e., the claimed second entity) can be a client device (Nodes 201 or 202), or vice versa.

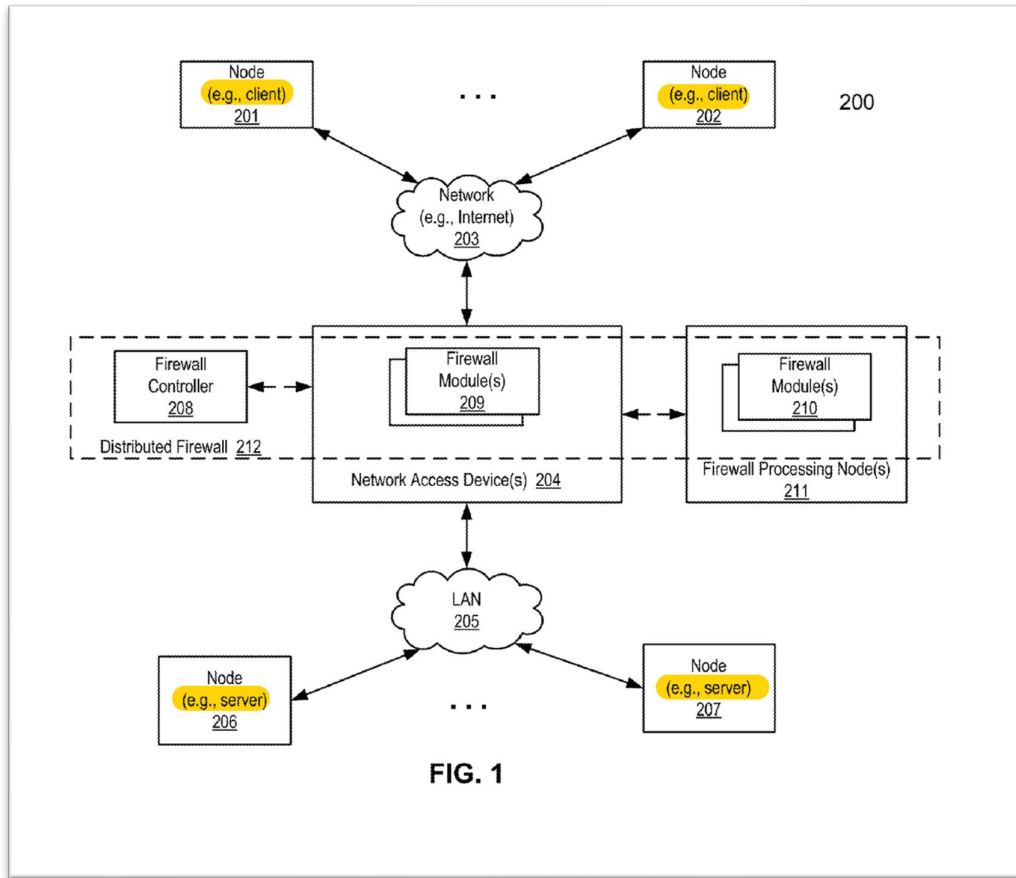


Figure 1 of Shieh (Annotated)

EX1006, Figure 1; see EX1006, ¶[0020], ¶[0022], ¶[0039], ¶[0053], ¶[0057], Figures 2A, 3.

298. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 22.

r. Claim 23

[23] The method according to claim 22, wherein the server device comprises a web server, and wherein the client device comprises a smartphone, a tablet computer, a personal computer, a laptop computer, or a wearable computing device.

299. It is my opinion that the element of Claim 23 is obvious over Shieh. EX1001, Col. 12:37-41.

300. As discussed above for Claim 23 in Ground 1, Shieh discloses that the source or destination of the packets can be a server device, such as those shown as Nodes 206 or 207 in Figure 1 of Shieh. EX1006, Figure 1; see EX1006, ¶[0020], ¶[0022], ¶[0039], ¶[0053], ¶[0057], Figures 2A, 3. Further, a POSA would have understood that the server can be a web server. For example, Paragraph [0053] of Shieh states that “[n]etwork interface 448 may provide a direct connection to a remote server via a direct link to the Internet via a POP (point of presence).” EX1006, ¶[0053], A POSA would have understood from this statement that the remote server is likely a web server used to transmit information over the Internet.

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

302. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 23.

s. **Claim 24**

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

303. It is my opinion that the element of Claim 24 is obvious in view of Shieh. EX1001, Col. 12:42-44. As discussed above for Claim 24 in Ground 1, Shieh

discloses the use of the standard OpenFlow protocol for communication between the network access devices (i.e., the network node) and the controller. EX1006, ¶¶[0025]-¶[0026]. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 24.

t. Claim 27

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

304. It is my opinion that the element of Claim 27 is obvious in view of Shieh. EX1001, Col. 12:51-52.

305. As discussed above for Claim 27 in Ground 1, Shieh states that its network access devices (which correspond to the claimed network nodes) “may be a router or a gateway, a switch or an access point...” EX1006, ¶[0020]; see EX1006, ¶[0019], ¶[0025]- ¶[0026], ¶[0032].

306. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 27.

u. Claim 28

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

307. It is my opinion that the element of Claim 28 is obvious in view of Shieh. EX1001, Col. 12:53-55.

308. As discussed above for Claim 28 in Ground 1, Shieh teaches the use of IP addresses to identify packets being transmitted across the packet network.

EX1006, ¶[0027], ¶[0031]. A POSA would have understood from the use of the IP addresses that packet network in Shieh is an Internet Protocol (IP) network and the packet is an IP packet. Shieh also discloses the communication of packets through IP networks. EX1006, ¶[0038].

309. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 28.

v. **Claim 29**

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

310. It is my opinion that the element of Claim 29 is obvious in view of Shieh. EX1001, Col. 12:56-58. As discussed above for Claim 29 in Ground 1, Shieh discloses the transportation of TCP packets across a TCP network using TCP protocol. EX1006, ¶[0027], ¶[0036].

311. Accordingly, it is my opinion that Shieh discloses and renders obvious Claim 29.

w. **Claim 30**

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

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

312. It is my opinion that the element [30.0] is obvious in view of Shieh. EX1001, Cols. 12:59-63.

313. Shieh discloses that its method can be applied to “subsequent packets of a particular session.” EX1006, ¶[0018]; see EX1006, ¶[0037] (discussing applying the method of Shieh to “further packets of the same session”). As such, Shieh 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].

314. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [30.0].

[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.

315. It is my opinion that the element [30.1] is obvious in view of Shieh. EX1001, Col. 12:64-13:3.

316. Shieh discloses that the additional packets are sent by the SDN switch (i.e., the claimed network node) over the packet network to the destination node (i.e., the second entity) in response to the packets not satisfying the packet-applicable criterion for the reasons discussed above for Element [1.5].

317. Further, Shieh discloses that the additional packets are sent by the SDN switch over the packet network in response to the packets not satisfying the packet-

applicable criterion and in response to the instruction for the reasons discussed above for Element [1.6].

318. Accordingly, it is my opinion that Shieh discloses and renders obvious Element [30.1].

VIII. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

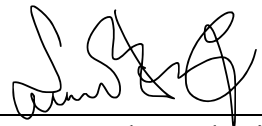
319. I am not aware of any objective indicia of non-obviousness that would make Claim 1 of the '111 Patent non-obvious over the prior art considered herein. In my opinion, any possible objective indicia of non-obviousness would not overcome the prior art that clearly demonstrates that the subject matter of Claim 1 of the '111 Patent would have been obvious to a POSA as of April 22, 2014.

320. I reserve the right to supplement my opinions as appropriate if Orckit alleges or offers any purported evidence of any such objective indicia of non-obviousness.

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

Executed this 21st day of February, 2023.

Respectfully submitted,



Samrat Bhattacharjee, Ph.D.