UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

CISCO SYSTEMS, INC.,

Petitioner,

- vs. -

CENTRIPETAL NETWORKS, INC.,

Patent Owner

---

IPR2018-01512
U.S. Patent 9,565,213

---

**SECOND PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 9,565,213**

## TABLE OF CONTENTS

## PETITIONER'S EXHIBIT LIST

| | |
|---|---|
| EX1001 | U.S. Patent No. 9,565,213 (the '213 Patent) |
| EX1002 | Prosecution History of the '213 Patent |
| EX1003 | CV of Dr. Kevin Jeffay |
| EX1004 | Declaration of Dr. Kevin Jeffay dated August 15, 2018 |
| EX1005 | Patent Owner's Opening *Markman* Brief in Related Litigation, *Centripetal Networks, Inc. v. Keysight Tech. Inc.*, Case 2:17-cv-00383-HCM-LRL ("*Keysight* lawsuit"), Doc. 106 (filed April 20, 2018) |
| EX1006 | Patent Owner's Rebuttal *Markman* Brief in *Keysight* lawsuit, Doc. 117 (filed May 3, 2018) |
| EX1007 | Parties' Joint Claim Construction Chart in *Keysight* lawsuit, Doc. 124, Ex. 1 (Filed May 11, 2018) |
| EX1008 | Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5 ("ACNS Guide") |
| EX1009 | U.S. Patent No. 9,172,627 ("Kjendal") |
| EX1010 | Hypertext Transfer Protocol (HTTP) 1.1 Specification, as defined in RFC 2616 (the "HTTP 1.1 Specification") |
| EX1011 | "An Architecture for Differentiated Services," also known as the Diffserv architecture, as defined in RFC 2475 ("Diffserv") |
| EX1012 | Declaration of Eli Fuchs, Dated July 2, 2018 |
| EX1013 | INTENTIONALLY NOT USED |
| EX1014 | INTENTIONALLY NOT USED |
| EX1015 | Dictionary.com definition of "monitor" found at http://www.dictionary.com/browse/monitor (visited June 7, 2018) |
| EX1016 | Excerpts from File History of European Pat. App. No. 15722292.8 |
| EX1017 | "The Internet Standards Process – Revision 3," as defined in RFC 2026, (the "RFC Specification") |

| | |
|---|---|
| EX1018 | U.S. Patent Pub. No. 2004/0114518 (“MacFaden”) |
| EX1019 | U.S. Patent No. 8,156,206 (“Kiley”) |

# I. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

## A. Real Party-in-Interest

The Petitioner and real party in interest is Cisco Systems, Inc. ("Cisco").

## B. Related Matters

To the best knowledge of the Petitioner, U.S. Patent No. 9,565,213 ("the '213 Patent") is or has been involved in the following matters:

| Name | Number | Court | Filed |
|---|---|---|---|
| *Centripetal Networks, Inc. v. Cisco Sys., Inc.* | 2:18-cv-00094-MSD-LRL | E.D. VA. | Feb. 13, 2018 |
| *Centripetal Networks, Inc. v. Keysight Tech. Inc.* | 2:17-cv-00383-HCM-LRL | E.D. VA | Jul. 20, 2017 |
| *Cisco Sys., Inc. v. Centripetal Networks, Inc.* | IPR2018-01386 | PTAB | Jul. 12, 2018 |

## C. Lead and Back-up Counsel and Service Information

Lead Counsel

Daniel W. McDonald
MERCHANT & GOULD P.C.
3200 IDS Center
80 South Eighth Street
Minneapolis, MN 55402

Phone: 612-336-4637
Fax: 612-332-9081
dmcdonald@merchantgould.com
USPTO Reg. No. 32,044

Back-up Counsel

Jeffrey D. Blake
MERCHANT & GOULD P.C.
191 Peachtree Street NE
Suite 3800
Atlanta, GA 30303

Phone: 404-954-5040
Fax: 612-332-9081
jblake@merchantgould.com
USPTO Reg. No. 58,884

| | |
|---|---|
| Kathleen E. Ott<br>MERCHANT & GOULD P.C.<br>1801 California Street<br>Suite 3300<br>Denver, CO 80202 | Phone: 303-357-1656<br>Fax: 612-332-9081<br>kott@merchantgould.com<br>USPTO Reg. No. 64,038 |
| Christopher C. Davis<br>MERCHANT & GOULD P.C.<br>3200 IDS Center<br>80 South Eighth Street<br>Minneapolis, MN 55402 | Phone: 612-371-5275<br>Fax: 612-332-9081<br>cdavis@merchantgould.com<br>USPTO Reg. No. 76,880 |

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


## II. GROUNDS FOR STANDING

Petitioner certifies that the ’213 Patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting *inter partes* review challenging the patent claims on the grounds identified in this Petition.

## III. REQUESTED RELIEF

Petitioner asks that the Board review the accompanying prior art and analysis, institute a trial for an *inter partes* review of claims 1-16 of the ’213 Patent (“the challenged claims”), and cancel them as unpatentable.

## IV. REASONS FOR THE REQUESTED RELIEF

As explained below and in the declaration of Cisco Systems' expert, Dr. Kevin Jeffay, the challenged claims of the '213 Patent are unpatentable because they would have been obvious to a person of ordinary skill in the art (POSA). Accordingly, the Board should institute trial and cancel claims 1-16.

### A. Networking Concepts Including Firewalls and Packet Filtering Were Well-Known Long Before the '213 Patent

A network interconnects computing devices to communicate with one another. EX1004, ¶27. The devices within a network process or forward transmissions to other devices in the network. Each device may be a computer, server, router, switch, or a combination thereof. *Id.*, ¶¶27-46. Different networks may be interconnected through routers or other interconnection devices, located at the juncture between two networks. *Id.*, ¶37.

When a device of one network wants to send data to a device in another network, the data is sent in the form of a datagram consisting of a header and a payload. EX1004, ¶38. The header specifies a network layer address, used to transmit the datagram between computers on different networks through a series of routers. *Id.*, ¶¶38-42. Accordingly, each router receives the data and determines the next network to which the datagram should be forwarded. *Id.*, ¶¶39-40. If such datagrams are not monitored as they enter different networks, bad actors are able to disrupt networks or gain improper access to devices within networks. *Id.*, ¶¶54-57.

To protect against such bad actors, network security techniques, including firewalls and packet filters, have been commonly implemented since the early 1990s. EX1004, ¶¶25-26, 54-57. Indeed, packet filtering in routers is one of the oldest and most widely available means to control access to networks. *Id.*, ¶55. In general, the concept is simple: a device is implemented to determine whether a packet is allowed to enter or exit a network by comparing some identifying pieces of information located in the packet's header. *Id*., ¶¶55-57. To make such a determination, rules are stored that identify criteria for how to filter the packets. *Id.*, ¶¶53-56.

The analysis of packets has long been based on information commonly referred to as the "5-Tuple," which includes the (1) source IP address, (2) source port, (3) destination IP address, (4) destination port, and (5) the transport protocol of each packet. EX1004, ¶46. If the analyzed packet data satisfied a rule's criteria, the packet was either dropped, forwarded to its destination, or subjected to other action such as routing the packet to a different destination. *Id.*, ¶¶35, 55-57. Effectively, a router or other interconnection device acts as a controlled gateway between two or more networks through which all traffic must pass. *Id.*, ¶¶38-42, 62-65.

The implementation of rules for analyzing packets, including changes or updates to the rules, is often controlled by a central management server. EX1004, ¶¶27, 72. Further, a log of the filtered packets is kept by sending filtered packets to a remote location via widely-used methods such as syslog. *Id.*, ¶56.

In networking terminology, it has long been common to refer to the structure of data exchanged between nodes (sometimes generically referred to as "packets") as comprising five "layers," including: the physical or interface layer (layer 1), the link layer (layer 2), the network layer (layer 3), the transport layer (layer 4), and the application layer (layer 5). EX1004, ¶¶27-53. The physical layer relates to information about the network's physical hardware/firmware. *Id.*, ¶30. The link layer allows for hardware addressing and often includes unique Media Access Control ("MAC") addresses assigned to each computer connected to the network. *Id.*, ¶¶31-37. The network layer facilitates the routing and delivery of data over the network. Network layer information often includes Internet Protocol ("IP") data. *Id.*, ¶¶38-42. The transport layer provides information for interfacing the application with the network infrastructure. *Id.*, ¶¶43-46. Common transport layer information includes Transport Control Protocol ("TCP") data. *Id*. The application layer provides information about the particular application program being used. *Id.*, ¶¶47-51. Together, these layers operate to facilitate the transmission of information across a network.

**B. Summary of the '213 Patent**

The '213 Patent purports to describe methods and systems for "protecting a secure network" that include a "packet security gateway (PSG)" located at boundaries between networks. EX1004, ¶¶79-85. The PSG provides an

"informational service" to "store and/or forward packet logs using a logging system based on a standard," such as "syslog, or the like," as shown in Figure 1 of the '213 Patent. EX1001, Col. 1:39-40, 11:34-57; EX1004, ¶80.

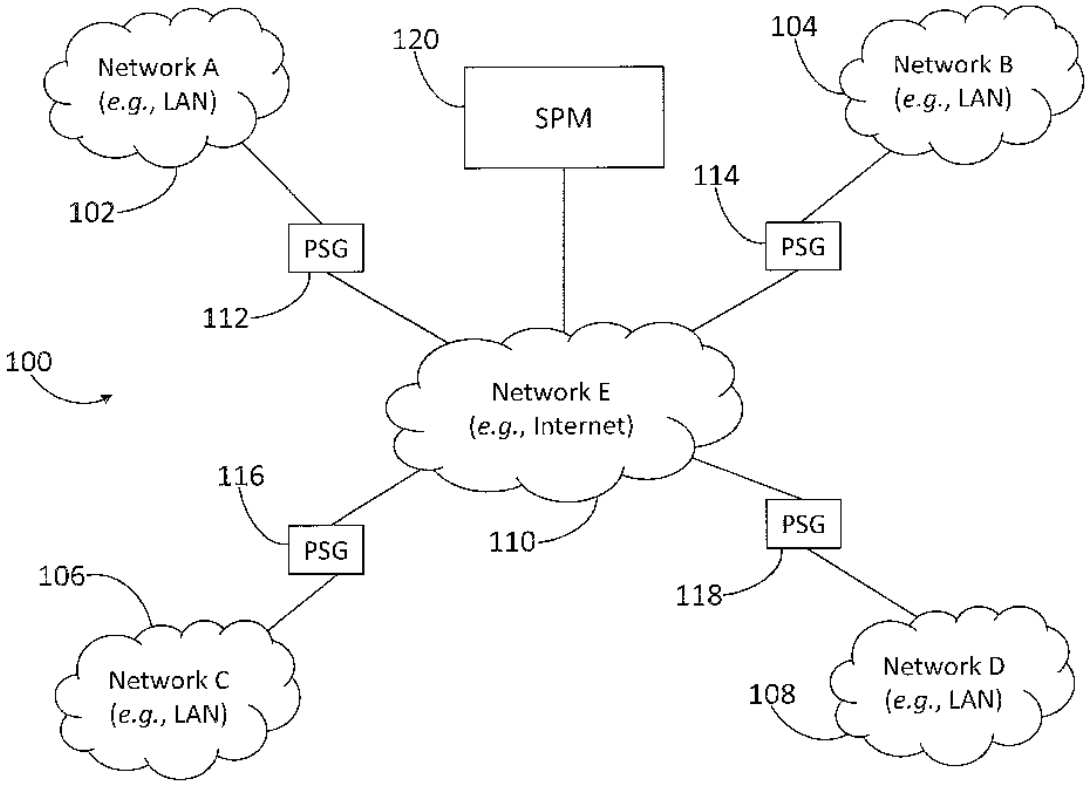



FIG. 1

The ’213 Patent includes a method figure describing the packet logging and digest functionality provided by the PSG (FIG. 12)(reproduced to the right). In the method depicted in Figure 12, each PSG receives a “dynamic security policy” from a “security policy management server” (SPM Server) that includes “one or more rules, the combination of which may effectuate an implementation of an informational service,” which may perform “a network communications awareness service, a network security awareness service, and/or a network threat awareness service (e.g., for a particular network environment).” EX1001, Col. 1:41-43, Col. 11:34-40; EX1004, ¶82. The rules identify “packet communications that are of interest to an organization that operates the secured network.” The rules of the ’213 Patent “further specify a packet transformation function that, when applied to a packet that matches such a rule, produces a digest version, or log, of the packet.” EX1001, Col. 11:46-48. The specification admits that such logging can be implemented using a known or “standard” logging format called “SysLog:” the PSG “store[s] and/or forward[s]

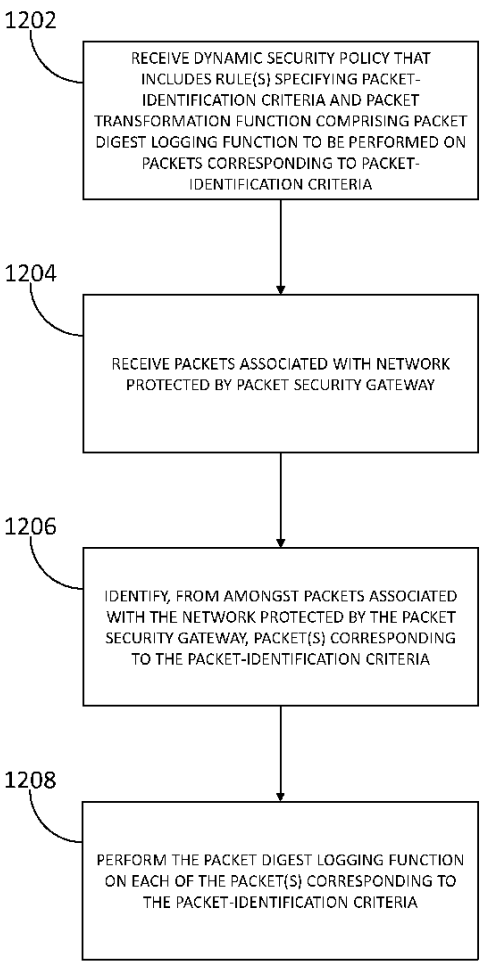


FIG. 12

packet logs using a logging system based on a standard (e.g., syslog, or the like)." *Id*., Col. 11:54-57. Syslog is a standard logging utility and log format originally developed for the UNIX operating system as early as the 1980s. EX1004, ¶¶81-85.

The '213 Patent contains 16 claims that embody the purported invention, of which claims 1 and 10 are the independent claims. *Id.*, ¶¶86-88.

**C. Prosecution History**

The application that resulted in the '213 Patent (U.S. Patent Application No. 14/253,992) was filed on April 16, 2014 and issued on February 7, 2017. EX1001; EX1004, ¶89.

The Office issued a restriction requirement on October 15, 2015. In response, applicant elected Group 1 (claims 1-18 "drawn to managing network security proactively by applying dynamic network security policies"). EX1002, pp. 366-373; EX1004, ¶90. The Office issued a first Office Action on February 26, 2016, including an obviousness-type double patenting rejection of pending claims 1-18 over U.S. Patent No. 9,137,205 (the "'205 Patent"), and a further rejection of those claims based on obviousness over the combination of Narayanaswamy (U.S. Publication No. 2009/0328219) in light of Kapoor (U.S. Publication No. 2008/0229415), and, for some dependent claims, two other prior art references. EX1002, pp. 427-459; EX1004, ¶91.

Patent Owner filed an amendment in response to that Office Action, and a terminal disclaimer over the '205 Patent. EX1002, pp. 1198-1207; EX1004, ¶¶92-94. In its response, Patent Owner amended independent claim 1 to recite an additional limitation of "routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to performing the packet transformation function." EX1002, pp. 1205-1206. Patent Owner also amended independent claim 10 to specifically recite that "the packet-identification criteria comprises a Differentiated Service Code Point (DSCP) selector." *Id.*, p. 1206. Patent Owner argued that neither Narayanaswamy nor Kapoor teach or suggest the features of independent claims 1 or 10, and that the claims are therefore allowable over the cited references. *Id.*, pp. 1205-1206. Further, on November 8, 2016, Patent Owner filed a terminal disclaimer to overcome the obviousness-type double patenting rejection in view of the '205 Patent. *Id.*, pp. 1229-1230.

The above-identified amendments were insufficient to procure allowance of the claims. EX1004, ¶95. The Office issued a Notice of Allowance of further-amended claims on November 16, 2016, in which the Examiner indicated that the prior claim rejections have been withdrawn in view of an "Examiner's Amendment." EX1002, pp. 1243-1248; EX1004, ¶95. Specifically, the Examiner amended the independent claims to incorporate new limitations directed to "a packet digest

logging function" based on language from dependent pending claims 11 and 15. EX1002, pp. 1243-1246; EX1004, ¶96. Claim 1 is representative, and was amended to recite:

> 1. A method comprising:
>
> receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying application-layer packet-header information and a packet transformation function ***comprising a packet digest logging function*** to be performed on packets comprising the application-layer packet-header information;
>
> receiving, by a packet security gateway of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway;
>
> identifying, by the packet security gateway, from amongst the packets associated with the network protected by the packet security gateway, and on a packet-by-packet basis, one or more packets comprising the application-layer packet-header information;
>
> performing, by the packet security gateway and on a packet-by-packet basis, the packet transformation function on each of the one or more packets comprising the application-layer packet-header ~~information; and~~ ***information, wherein the performing the packet transformation function comprises***
>
> ***identifying a subset of information specified by the packet digest logging function for each of the one or more***

***packets comprising the application-layer packet-header information;***

***generating, for each of the one or more packets comprising the application-layer packet-header information, a record comprising the subset of information specified by the packet digest logging function; and***

***reformatting, for each of the one or more packets comprising the application-layer packet-header information, the subset of information specified by the packet digest logging function in accordance with a logging system standard; and***

routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function.

*Id*., pp. 1243-1245 (emphasis in original); EX1004, ¶96. Claim 10 was amended to recite similar limitations relating to the "packet digest logging function," and was further amended to incorporate a "routing" limitation similar to that of claim 1. EX1002, pp. 1245-1246; EX1004, ¶96. The Reasons for Allowance set forth by the Examiner indicate claims 1 and 10 were allowable over the prior art specifically because the prior art fails to teach the "identifying," "generating," and "reformatting" aspects of the "packet digest logging function." EX1002, pp. 1248-1250; EX1004, ¶¶97-101.

**D. Priority Date**

All of the references cited in the instant Petition qualify as prior art based on the priority date of April 16, 2014, the filing date of the non-provisional application for the ’213 Patent.

**E. Challenged Claims**

Claims 1-16 of the ’213 Patent are challenged.

**F. Discretionary Denial Under § 325(d) or § 314 is Not Warranted**

Although Petitioner has filed another petition for *inter partes* review of the ’213 Patent, IPR2018-01386, the instant Petition presents different grounds for review by the Office than were presented in the other petition.

In addition, the combination of references forming the grounds for this petition has not been previously considered by the Office, nor is it redundant to any combinations of references considered during prosecution. Thus, the analysis in the instant Petition has never been presented or considered by the Office. Additionally, the instant Petition provides the analysis of Dr. Kevin Jeffay regarding this prior art, which has likewise not been presented to the Office. As the same or similar arguments have not been previously presented to the Office, discretionary denial under §325(d) is not warranted here.

Likewise, denial under §314(a) is not warranted as the application of the factors in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha* to the present circumstances demonstrates that the filing of the instant Petition is not

abusive, that neither the Patent Owner, the Office, nor the Board has addressed the merits of the grounds in this Petition, and instituting the present proceeding is an efficient use of the Board's resources. *See* IPR2016-01357 (PTAB Sept. 6, 2017) (precedential). As such, the Board should proceed to institute trial on the instant Petition.

### G. Secondary Considerations

Neither Petitioner nor Dr. Jeffay is aware of any secondary considerations of non-obviousness that would support the patentability of the challenged claims. EX1004, ¶¶304-305.

## V. CLAIM CONSTRUCTION

This Petition analyzes the claims consistent with the broadest reasonable interpretation (BRI) in light of the specification. *See* 37 C.F.R. § 42.100(b); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2142 (2016); EX1004, ¶104. Petitioner also analyzed the claims and compared them to the prior art should the claims be 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); EX1004, ¶104. Petitioner believes the claims are unpatentable regardless of whether BRI constructions or litigation constructions are invoked. EX1004, ¶104. Indeed, in some cases, such as where the specification provides guidance as to a term's scope or Petitioner uses the Patent Owner's agreed construction from litigation, there may

be no difference between the constructions. *See id*. Further, terms not construed below have their plain and ordinary meaning. *See id*.

**A. "Dynamic Security Policy"**

A POSA would understand that "dynamic security policy" includes what the '213 Patent explicitly says the phrase includes: "any rule, message, instruction, file, data structure, or the like that specifies criteria corresponding to one or more packets and identifies a packet transformation function to be performed on packets corresponding to the specified criteria." EX1001, Col. 4:58-63; EX1004 ¶¶105-107. The '213 Patent further describes "dynamic security policy" as being created either "manually" or "automatically." EX1001, Col. 14:41-47 and 14:56-15:5; EX1004, ¶105.

Patent Owner proposed in litigation that a "dynamic security policy" means "a non-static set of one or more rules, messages, instructions, files, or data structures associated with one or more packets." EX1005, p. 6; EX1006, p. 3; EX1007, Attachment 2, p. 2. This construction differs from the express definition set forth in the specification. EX1004, ¶106. Nevertheless, the claims of the '213 Patent are also invalid under Patent Owner's construction. *Id.*

**B. "Rule/Rules"**

A POSA would understand that a "rule" is a part of a dynamic security policy. Further, a "rule" includes what the '213 Patent states it may include: "one or more

rules" that "may specify criteria and one or more packet transformation functions that should be performed for packets associated with the specified criteria." EX1001, Col. 6:11-12; 7:10-13; *see also* EX1004, ¶¶108-110.

**C. "Security Policy Management Server (SPM Server)"**

The '213 Patent states that an SPM Server "includes any computing device configured to communicate a dynamic security policy to a packet security gateway." EX1001, Col. 4:38-40. In copending litigation regarding this and similar patents (the "*Keysight* Litigation"), Patent Owner agreed to construe the term to mean "a server configured to manage policies and communicate a dynamic security policy to a packet security gateway." EX1004, ¶¶111-112; EX1007. Without agreeing that this definition is correct for all purposes, Petitioner will use it in this petition.

**D. "Packet Security Gateway" (PSG)**

The '213 Patent states that a "packet security gateway" "includes any computing device configured to receive packets and perform a packet transformation function on the packets." EX1001, Col. 4:48-50. In the *Keysight* Litigation, Patent Owner agreed to construe the term to mean "a gateway computer configured to receive packets and perform a packet transformation function on the packets." EX1004, ¶¶113-114; EX1007. Without agreeing that this definition is correct for all purposes, Petitioner will use it in this petition.

### E. "Packet Transformation Function"

The '213 Patent does not explicitly define "packet transformation function." It does, however, describe a packet transformation function as an action taken upon a packet, including forwarding, dropping, accepting, routing, and queueing such packets. EX1001, Col. 2:37-57, 8:8-14, 10:6-10, and FIG 3; *see also* EX1004, ¶¶115-120.

Despite this description, Patent Owner attempted to exclude forwarding and dropping packets from the definition of "packet transformation function" in the *Keysight* Litigation. EX1007. Patent Owner's proposed construction in the *Keysight* Litigation is nonsensical in light of the '213 Patent's claims and specification. For example, claim 2 recites "performing the packet transformation function comprises **forwarding**, by the packet security gateway, the each of the one or more packets…" EX1001, Col. 25:56-63 (emphasis added). Claim 3 recites "performing the packet transformation function on comprises **dropping**, by the packet security gateway, the each of the one or more packets…" *Id.*, Col. 25:64-26:3 (emphasis added). Further, the '213 Patent specification states, "[D]ynamic security policy 300 may include one or more rules that specify a packet transformation function **other than forwarding (accepting or allowing) or dropping (denying) a packet**." *Id.*, Col. 8:4-14 (emphasis added). A POSA would understand this statement to mean that forwarding and dropping are ordinarily part of the packet transformation function,

but that the example of "dynamic security policy 300" excludes those two packet transformation functions. EX1004, ¶¶116-119.

Thus, a POSA would have understood the correct interpretation of "packet transformation function" to be "an action taken upon a packet." *Id*., ¶120.

**F. "Monitoring Device"**

The '213 Patent does not explicitly define "monitoring device." It refers, however, to a monitoring device as "stor[ing] the packets or data contained within them for subsequent review or analysis . . . . In such embodiments, it may not be necessary for monitoring device 608 to strip the encapsulating header from the packets or route them based on their destination address . . . ." EX1001, Col. 17:9-21; EX1004, ¶¶121-123; EX1014, Dictionary.com definition of "Monitor" ("15. to observe, record, or detect (an operation or condition) with instruments that have no effect upon the operation or condition."); *Compare* EX1001, Col. 11:26-33 (monitoring device may strip encapsulating header and forward packets to destination address).

Thus, a POSA would have understood the correct interpretation of "monitoring device" to be a "device configured to copy (or store) packets or data contained within them." EX1004, ¶123.

## VI. STATUTORY GROUNDS FOR CHALLENGES

**Ground #1**: Claims 1-9 of the '213 Patent are obvious under 35 U.S.C. § 103 over "Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5" ("ACNS") in view of U.S. Patent No. 9,172,627 ("Kjendal").

ACNS is a printed publication that bears a 2006 copyright notice. EX1008, p. 2. It was published and made available for public download by at least March, 2008. EX1012 ¶¶8-13, 15; EX1004, ¶127. Accordingly, ACNS qualifies as prior art to the '213 Patent under at least AIA 35 U.S.C. §§ 102(a)(1).

Kjendal was filed March 15, 2013, and published on September 18, 2014. Kjendal is prior art under at least AIA 35 U.S.C. §§ 102(a)(2); EX1004 ¶128.

**Ground #2**: Claims 10-16 of the '213 Patent are obvious under 35 U.S.C. § 103 over ACNS in view of Internet Engineering Task Force (IETF) Request for Comment (RFC) 2475 titled "An Architecture for Differentiated Services" ("Diffserv") and further in view of Kjendal.

Diffserv is a printed publication available from the IETF website (ietf.org) and was publicly available at least as early as December 1998. EX1004, ¶¶129-136. Accordingly, Diffserv qualifies as prior art under at least AIA 35 U.S.C. §§ 102(a)(1).

## VII. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art ("POSA") in the field of the '213 Patent, as of April 16, 2014, would have had a working knowledge of packet-switched networking, firewalls, security policies, communication protocols and layers, and the use of customized rules to address cyber-attacks. EX1004, ¶¶22-24. A POSA would have had a bachelor's degree in computer science, computer engineering, or an equivalent, and four years of professional experience. *Id.* Lack of work experience can be remedied by additional education, and vice versa. *Id.; see also id.* at ¶¶1-11; EX1003.

## VIII. NOTE REGARDING PAGE CITATIONS AND EMPHASIS

Petitioner's citations to the exhibits used throughout use the page, paragraph, or column numbers in their original publication. Page numbers to the ACNS prior art are those printed in the document (generally chapter-page except for prologue pages with individual page numbers). Unless otherwise specified, all bold and italics emphasis has been added.

## IX. CLAIMS 1-16 ARE UNPATENTABLE

This petition's showing that the cited art renders the challenged claims unpatentable is fully supported by the Declaration of Kevin Jeffay (EX1004), a Computer Science Professor at the University of North Carolina for 29 years. EX1004, ¶4. Professor Jeffay is familiar with the state of the computer networking

art before the ’213 Patent was filed (*see id*., ¶¶1-11, 20, 25-78; *see also* EX1003), and fully agrees with and supports the showing herein that the claims at issue merely recite aspects of computer network security that have been long known. EX1004, ¶¶145, 235, 236, 303; *see also* EX1004, *generally*.

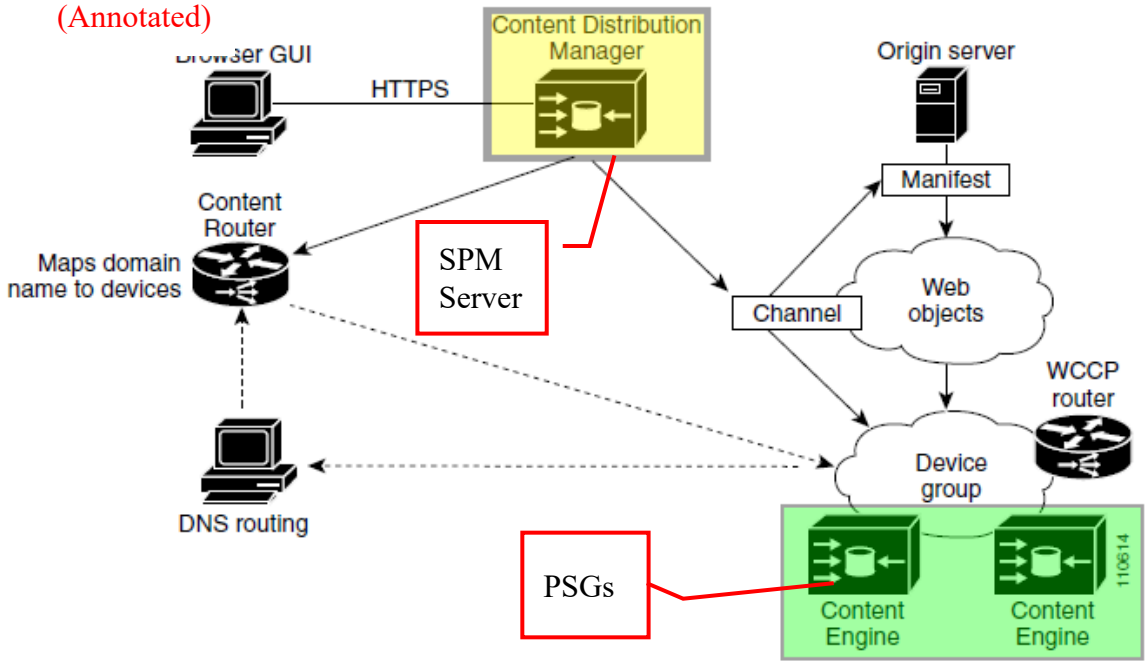
### A. Ground 1 – ACNS in View of Kjendal (Claims 1-9)

#### 1. Overview of ACNS

ACNS is a configuration guide for the Cisco Application and Content Networking System, version 5.5, released well before the priority date of the ’213 patent. EX1008, pp. 2, 1-2; EX1004, ¶¶146-147. ACNS teaches software operating in four modes: Content Engine, Content Distribution Manager (“CDM”), Content Router, and Cisco IP/TV Program Manager. *Id*. The CDM meets the requirements for the claimed security policy management server (“SPM Server”) in each of the independent claims. *Id*. Moreover, the ACNS system’s Content Engine meets the requirements for the claimed packet security gateways (“PSGs”). *Id*. Figure 1-2 (annotated) of ACNS shows a simplified layout for use of the ACNS software and matches its elements to the ’213 Patent claim elements:

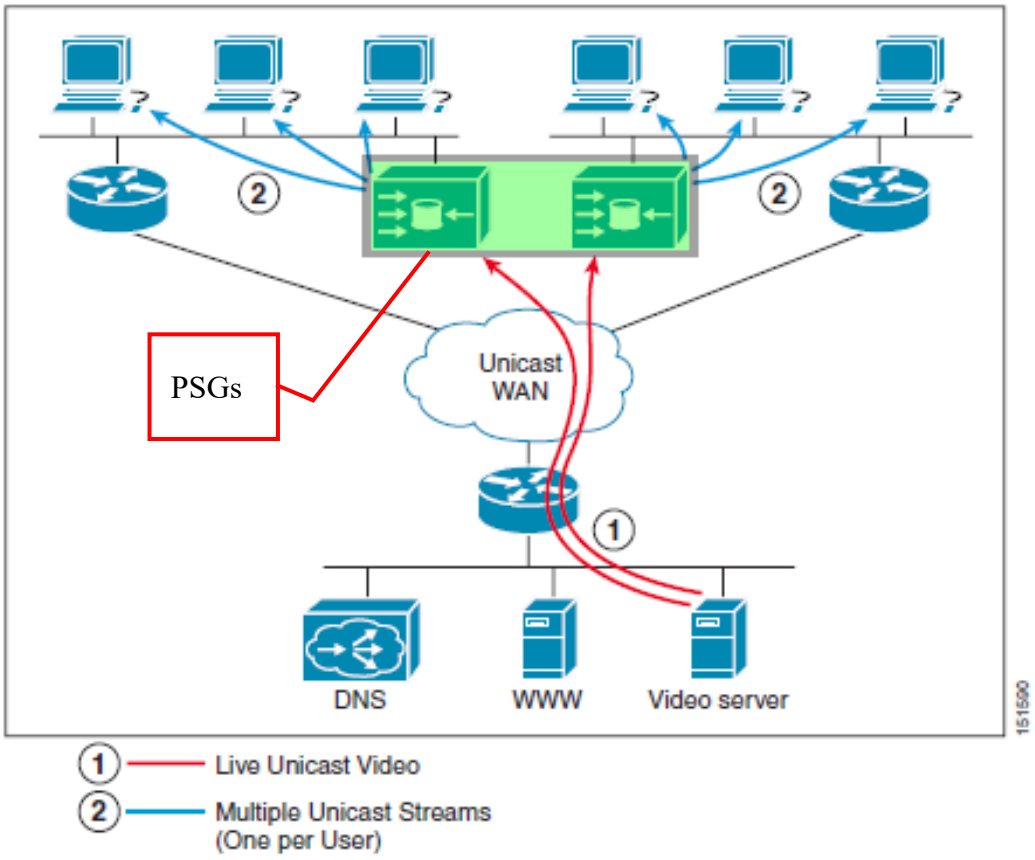
*Figure 1-2 One-to-One Channel Mapping to Devices*

(Annotated)




EX1004. ¶147. An administrator may connect to the CDM and cause it to issue various policies to Content Engines. *Id.* Those policies may include rules specifying application-layer packet-header information and a packet transformation function that comprises a packet digest logging function. *Id.* The Content Engines execute these policies while acting as the claimed PSGs for network elements, as shown in Figure 9-1 of ACNS (annotated) showing a unicast streaming setup for the ACNS system:

***Figure 9-1 Unicast Live Stream Splitting***
(Annotated)




EX1004, ¶147. The above diagram is merely one iteration of how Content Engines operate within the ACNS system. Other layouts include: multicast, hybrid-unicast-multicast, demand cached, and pre-positioned video setups. *See id.*; EX1008, 9-1 – 9-6.

ACNS describes the software's purpose as including "content caching and access control" to "**authenticate, monitor, log**, and control web access from end users to implement corporate policies…and system security" and describes objectives such as preventing viruses from entering the system and preventing

employees from accessing objectionable content. EX1008, 1-9. ACNS states that all these objectives may be met through the implementation of Content Engines at the network edge such that "the Content Engines can then intercept content requests made by end users and enforce the above policies." *Id*. These statements demonstrate that the Content Engines protect a network.

ACNS further describes the Content Engines receiving packets and policies, using application-packet header information in packets as criteria in the received policies, performing packet transformation functions defined within the received policies that identify and then reformat packet-header information into a logging standard, and finally, routing the packets to a monitoring device. EX1004, ¶¶147.

An example operation is described in chapters 8 and 19 of ACNS. Chapter 8 is titled "Configuring Caching Services" and describes how the ACNS system may store data from external websites for other users without having to retrieve the same data multiple times – thus conserving external bandwidth. EX1008, 8-1. To do so, the ACNS system investigates the user's HTTP request packet for information, determines whether the request is allowed, determines whether the system already has the information cached, and generates a log entry for the request. EX1008, Chapters 8 and 19, generally. As part of this process, the ACNS determines whether the system is configured to allow the specific HTTP request method within the request packet. *Id*., 8-23 – 8-24. This series of events teaches the majority of

elements claimed in the '213 Patent, and all remaining elements are elsewhere taught or suggested by ACNS or by Kjendal, discussed below. The Examiner was not aware of ACNS during prosecution.

### 2. Overview of Kjendal

Kjendal is titled "Device and Related Method for Dynamic Traffic Mirroring," and relates to monitoring traffic for analysis and security by sending copies of certain, identified packets to a monitoring or logging device, based on policies provided to the device. EX1009, *Abstract*, Col. 6:15-19; EX1004, ¶¶148, 192-196. Specifically, Kjendal discloses "general policy-based mirroring [that] allows network administrators to set network mirror-policies for various network monitors including, for example…***network loggers***, analyzers, etc." EX1009, 10:46-49 (emphasis added). Kjendal further provides for identifying packets to mirror by analyzing the contents of their application-layer packet-header information. EX1004, ¶¶148, 165, 195; EX1009, Col. 30:14-16; 10:23-33 (providing example dynamic rules such as "mirror[ing] the first N packets of any new application flow from a source…"). The Examiner was not aware of Kjendal during prosecution.

### 3. Motivations to Combine ACNS and Kjendal

A POSA would understand that port-mirroring for monitoring was well-known at the time of alleged invention, and that such systems were routinely deployed on Cisco router systems. EX1004, ¶¶148, 197-198. A POSA would expect

to include monitoring features to improve another, similar Cisco product such as ACNS and such a POSA would expect such a combination to yield predictable benefits. *Id*.

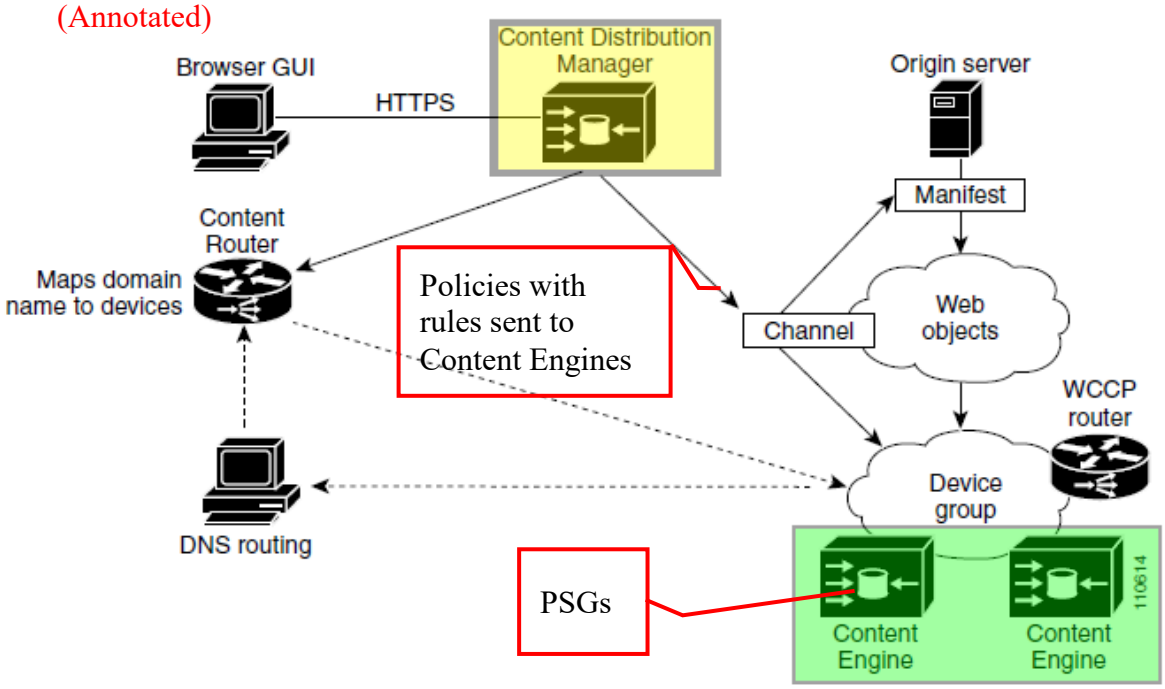
A POSA would find it obvious and straightforward to incorporate the specific packet-mirroring teachings of Kjendal with the teachings of ACNS, including ACNS's teaching of selecting, on a packet-by-packet basis, those packets comprising selected application-layer packet header information. *Id*. A POSA would understand that the ACNS provides logging for a wide range of purposes, including logging packets routed to a monitoring device. *Id*. ACNS explains, for example, that "[s]ome enterprises have a requirement to monitor, manage, and restrict employee access to nonbusiness and objectionable content on the Internet." *Id.* ¶166; EX1008, 16-25. This teaching of monitoring by ACNS would prompt a POSA to combine its teachings with related art, such as Kjendal, which specifically teaches how to monitor packets through mirroring.

**4. Claim 1 – Element [1.1]**

***[1.1] A method comprising: receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying application-layer packet-header information***

ACNS teaches element [1.1], which requires that packet security gateways (PSGs) receive a dynamic security policy from a security policy management server (SPM Server), and that the dynamic security policy comprises a rule specifying application-layer packet-header information. EX1004, ¶¶149-159. In ACNS, the Content Engines (marked in green, below) represent the claimed PSGs and are associated with a Content Distribution Manager (CDM, marked in yellow) that represents the claimed SPM Server. EX1004, ¶150.

*Figure 1-2 One-to-One Channel Mapping to Devices*
(Annotated)




EX1008, Fig. 1-2 at page 1-11 (annotated).

With respect to the dynamic security policy, as discussed above regarding claim construction, the '213 Patent broadly describes a "dynamic security policy" as including "any rule, message, instruction, file, data structure, or the like that

specifies criteria corresponding to one or more packets and identifies a packet transformation function to be performed on packets corresponding to the specified criteria." EX1001, Col. 4:58-63. The '213 Patent further explains that such policies may be "dynamic" in that they can be modified based on new information, either manually or automatically. *Id.*, Col 14:36-47.

In ACNS, the dynamic security policy includes (1) the set of rules used to configure the operation of the Content Engines (addressed in this claim element) and (2) packet transformation functions to be performed on packets that pass through the Content Engines (addressed below in element [1.2]). EX1004, ¶¶149-167. The dynamic security policy is generated at the CDM based on input by a user. *Id.*, ¶153; *see also* EX1008, 8-24 (showing step-by-step instructions for generating and updating a policy, which include rules, via the "Content Distribution Manager GUI"); EX1008, C-1 – C-27 (showing command-line interface commands for creating and modifying parts of a policy, such as "http add-method" to update a rule identifying a list of allowed HTTP request methods with an additional method). The policy is then provisioned to one or more Content Engines by the CDM, as shown above in Figure 1-2 of ACNS. *See* EX1008, 1-2 ("…the Content Distribution Manager…also manages policy settings and configurations on individual Content Engines"); EX1004, ¶¶150-153. The Content Engines then act as the claimed PSGs by "intercept[ing] content requests made by end users and enforc[ing] the above

policies" on the packets to eliminate objectionable content or provide virus protection. EX1008, 1-9; EX1004, ¶152.

ACNS discloses at least two examples of "*rule[s] specifying application-layer packet-header information*" that are part of the dynamic security policy provisioned to the Content Engines. The first example rule specifies application-layer packet-header information concerning HTTP request methods. EX1004, ¶¶153-157. ACNS teaches that the Content Engines may identify HTTP (i.e. application-layer) packets and the HTTP request method associated with those packets, which is located within the application-layer packet-header information. EX1004, ¶¶153-154; EX1008, 8-23 – 8-24 (describing filtering of HTTP packets based on request method). The Content Engines apply this rule to determine whether the ACNS system supports the HTTP request method contained within a particular packet. *Id*. at ¶¶153-157; EX1008, 8-23 – 8-24. ACNS teaches that the Content Engine may accept or deny a packet based on whether the request method matches an entry on a list of acceptable request methods (i.e. whether the application-layer packet-header information satisfies the criteria set out by the rule). EX1004, ¶¶153-157; EX1008, 8-23 – 8-24 (describing filtering of HTTP packets based on request method). This rule is "dynamic" at least because the list of acceptable request methods can be modified to add or subtract different request methods. EX1008, 8-23; EX1004, ¶¶155-157.

The second example rule "*specifying application-layer packet-header information*" that is part of the dynamic security policy provisioned to the ACNS Content Engines relates to the use of Internet Content Adaptation Protocol ("ICAP"). *Id.*, ¶158. ACNS describes ICAP as follows:

> ICAP is an open standards protocol that can be used for content adaptation, typically at the network edge. Content adaptation includes virus scanning, content translation, content filtering, content insertion, and other ways of improving the value of content to end users.

EX1008, 16-1; *see also* EX1004, ¶158. ACNS teaches that its Content Engines can use ICAP-based servers to examine, filter, and modify HTTP (application-layer) packets. EX1004, ¶158. To examine, filter and modify content, the ACNS system implements "rules" to identify specific application-layer packets, based on their header information, and "rule actions" to transform identified packets in specific ways. EX1008, 16-13 – 16-23 (titled "Configuring Service Rules"); EX1004, ¶158.

ACNS specifically teaches implementing ICAP to match the "header-field" pattern within an HTTP header. *Id*. pp. 16-17; EX1005, ¶158. ACNS refers to matching patterns of "request-line," "referer [sic]," and "user-agent." *Id*. Each of these fields is an element of an HTTP packet header. *See* EX1010, pp. 26, (listing "Request Header Fields," including "referrer[sic]," and "user-agent") and 24, 86, 89 (defining each of the fields in the HTTP protocol). ACNS also teaches modifying each of these fields with a substitution string, to re-route or otherwise modify the HTTP packet. EX1008, 16-17 (describing the "header-field-sub" functionality);

EX1004, ¶158. In this way, ACNS teaches using the ICAP system within the ACNS system to identify (and transform) packets based on application-layer packet-header information. EX1004, ¶158. Further, ACNS discloses that the "Content Distribution Manager GUI" (i.e. the claimed SPM Server) can be used to set up ICAP rules for packet identification and field substitution, and the rules are then provisioned to the ACNS Content Engines. *See* EX1008, 16-15.

Thus, ACNS teaches element [1.1] as shown above at least because it discloses two types of rules specifying application-layer packet-header information (i.e. a HTTP request method rule and an ICAP-based rule) that are part of a dynamic security policy that can be provisioned from the CDM to the Content Engines.

**5. Element [1.2]**

***[1.2] and a packet transformation function comprising a packet digest logging function to be performed on packets comprising the application-layer packet-header information;***

The combination of ACNS and Kjendal teaches element [1.2], requiring a "packet transformation function comprising a packet digest logging function to be performed on packets comprising the application-layer packet-header information." EX1004, ¶¶160-167. ACNS teaches several packet transformation functions, including a packet digest logging function. ACNS does not explicitly state that its packet digest logging functions are performed on packets based on their meeting

criteria in the two application-layer packet-header rules discussed above in element [1.1]. EX1004, ¶165. However, given the teachings of Kjendal and ACNS, it would have been obvious to a POSA to implement the ACNS system such that the logging function logs packets based on application-layer packet-header rules. EX1004, ¶¶164-167. Kjendal discloses a system where application-layer packet-header information is used to determine what should be logged. *Id.* Moreover, a POSA would have been motivated to implement these teachings of Kjendal into ACNS in order to only log packets that are of particular interest and thus avoid logging every packet that comes through the ACNS Content Engines, which could provide unnecessary data that may overwhelm the system. EX1004, ¶166. For example, as explained by ACNS, logging every entry to the Syslog server (called "real-time transaction logging") may result in that server becoming overloaded and dropping log packets. EX1008, 19-22 ("However, if you log all transactions, there may be a significant UDP drop rate if your syslog host cannot handle the rate of incoming traffic.").

ACNS generally teaches packet transformation functions that can be performed only on packets that are identified by application-layer packet-header information. For example, with the above-discussed rule relating to HTTP request methods (i.e. application-layer packet-header information), the Content Engine may accept or deny a packet (i.e. perform a packet transformation function) based on

whether the request method is found within a list of functions supported by the ACNS system. EX1004, ¶¶153-157; EX1008, 8-23 – 8-24 (describing filtering of HTTP packets based on request method). Further, with respect to rules based on ICAP, the Content Engine can modify selected packets to, for example, re-route them. EX1004, ¶158; EX1008, 16-17 (describing the "header-field-sub" functionality).

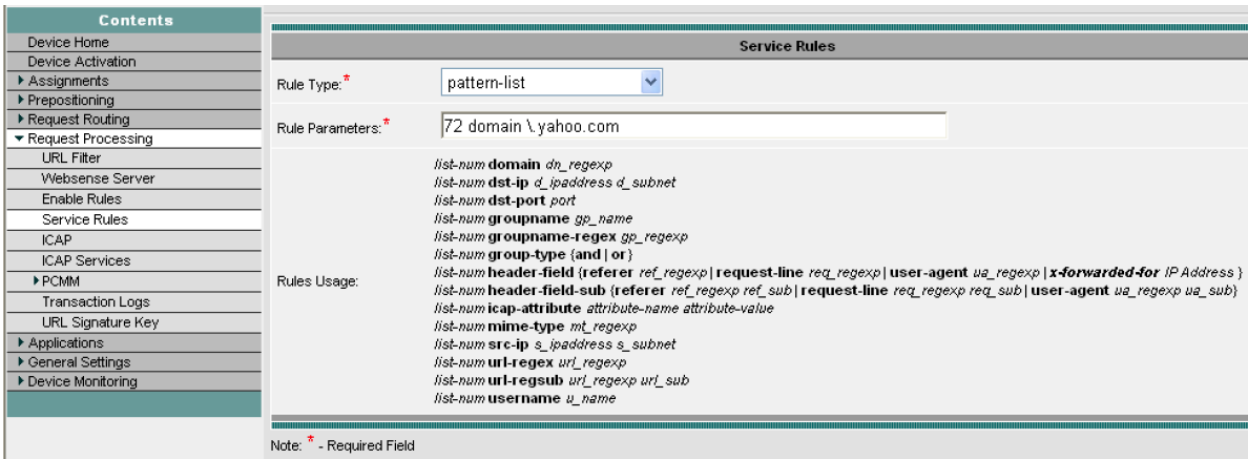
ACNS also discloses "*a packet digest logging function*" as a "*packet transformation function*" that can be performed on packets passing through the ACNS Content Engines. EX1004, ¶¶161-163; EX1008, 19-1 – 19-11 (describing details of logging functionality of the ACNS system). ACNS teaches that a user may choose whether to implement the logging functionality such that all packets are logged or only certain packets are logged. EX1008, 19-21 – 19-22; EX1004, ¶163. Further, ACNS discloses logging the application-layer packet-header information of certain packets. EX1004, ¶¶161-163. For example, a "squid" format sample log entry shown in ACNS contains information found within the application-layer packet-header such as the URL and HTTP method:

> 1012429341.115 100 172.16.100.152 TCP_REFRESH_MISS/304 1100 ***GET*** ***http://www.cisco.com/images/homepage/news.gif*** - DIRECT/www.cisco.com -

EX1008, 19-1; EX1004, ¶162. Thus, the ACNS logging function extracts and digests application-layer packet-header information. As a second example, ACNS

can take the packets that are originally logged in squid format (or any other format shown in Chapter 19 of ACNS) and further filter them to send only packets that are the subject of HTTP request authentication failures to a separate syslog server, in a process called "real-time transaction logging". EX1008, 19-19 – 19-22; EX1004, ¶163. In order to determine whether a packet contains an HTTP authentication failure notice, the ACNS system again relies on application-layer packet-header information. EX1004, ¶163.

A POSA would consider the logging functionality of the ACNS system to be part of the dynamic security policy provisioned from the CDM to the Content Engines at least because ACNS discloses that the "Content Distribution Manager GUI" can be used for modifying the logging functionality. EX1004, ¶¶161-163; EX1008, 19-10 – 19-11 (explaining how to enable transaction logging using the CDM GUI), 19-20 – 19-21 (teaching user selection of "the type of transactions you want to log . . . ."). An example of the "Content Distribution Manager GUI" may be found on page 16-15 of ACNS at Figure 16-3:




On the left of Figure 16-3, one may see drop-down menus for various aspects of the ACNS system, including the one selected in the above screenshot, "Request Processing" for service rules. EX1008, 16-15. Additional elements under "Request Processing" includes "Transaction Logs," which includes the UI for configuring the ACNS logging. *See* EX1008, 19-20 – 19-21. Below "Request Processing" on the left drop-down is "Applications," wherein a user may configure the HTTP request method filtering described in Chapter 8 of ACNS. *See* EX1008, 8-24 (describing functionality under "Applications > Web > HTTP > HTTP Method."). In this way, all of the ACNS functionality described herein may be configured within a single policy and sent to one or more Content Engines from the Content Distribution Manager. EX1004, ¶¶161-163; see also EX1008, Appendix C (pp. C-1 – C-26) (disclosing that an administrator may also configure Content Engines through a command line interface (CLI)).

The packet digest logging function disclosed in ACNS is a flexible tool, but ACNS does not explicitly disclose that the logging function can be applied to log packets based on the application-layer packet-header information identified by the HTTP request method or ICAP rules discussed above for element [1.1]. Nonetheless, this would have been an obvious implementation of the ACNS logging function to a POSA. A POSA would have understood that the logging function of ACNS could have been performed on packets other than those containing an HTTP authentication failure notice, i.e. on packets having application-layer packet-header information as specified in the above rule. EX1004, ¶¶163-166.

Indeed, Kjendal demonstrates that it was known in the art to use a "policy-based mirroring" system that, based on certain rules, sends copies of packets to monitoring or logging devices. EX1009, Col. 10:46-58 ("This general policy-based mirroring allows network administrators to set network mirror-policies for various network monitors including, for example…***network loggers***, analyzers, etc.") (emphasis added); EX1004, ¶165. Specifically, Kjendal teaches that packets may be mirrored (i.e. sent to a network logger) based on dynamic rules specifying application-layer packet header information, such as identification of specific application types. EX1009, Col. 10:23-33 (providing example dynamic rules such as "mirror[ing] the first N packets of any new application flow from a source…"); EX1004, ¶165. Thus, Kjendal discloses a packet transformation function including

a "*packet digest logging function*" which is performed based on application-layer packet-header information. EX1004, ¶¶164-167.

It would have been obvious to a POSA that the rule-based mirroring and logging system of Kjendal could be implemented in ACNS to log packets based on application-layer packet-header information specified in HTTP request method or ICAP-based rules – the ACNS rules that are discussed above regarding element [1.1]. *Id.* A POSA would have understood that the selective logging of packets, based on application-layer packet-header information disclosed in Kjendal, addressed a specific problem noted in ACNS with respect to the logging of HTTP authentication failures. ACNS explains that the "real-time transaction logging" of authentication failures is especially desirable for the security of the system, but that transmitting every log entry of every packet to a syslog server for analysis may overwhelm said server. EX1008, 19-21 – 19-22. For this reason, an administrator using ACNS may choose whether to selectively send *only* logs of HTTP authentication failures to the syslog server, as opposed to sending every log entry to the syslog server. *Id*.; EX1004, ¶163. A POSA would understand that an administrator of the ACNS system could also choose to selectively log other types of application-layer information. EX1004, ¶¶163-165. Indeed, Kjendal teaches how to implement such selective logging for other types of application-layer information and further suggests the benefits of filtering packets before mirroring to another

device (i.e. for logging) in order to reduce the chances of overloading the target of the mirrored packets. EX1009, Col. 34:51-54; EX1004, ¶165.

Kjendal further teaches that a POSA would understand that its port mirroring and other operations may be practiced with "any bridgeable or routable protocol," encouraging their use in a wide variety of contexts. *Id.* Col. 28:3-18. ACNS uses a number of routable protocols, such as TCP/IP, and would easily integrate the teachings of Kjendal. EX1004, ¶166. Given the above, the value of selective, real-time transaction logging of packets based on application-layer packet-header information would have been well-known to a POSA at the time of the alleged invention, as evidenced by Kjendal, and a POSA would have been motivated to apply that selective logging in the ACNS system. EX1004, ¶¶165-166.

The combination of ACNS and Kjendal thus teaches the "*packet transformation function comprising a packet digest logging function*" recited in element [1.2].

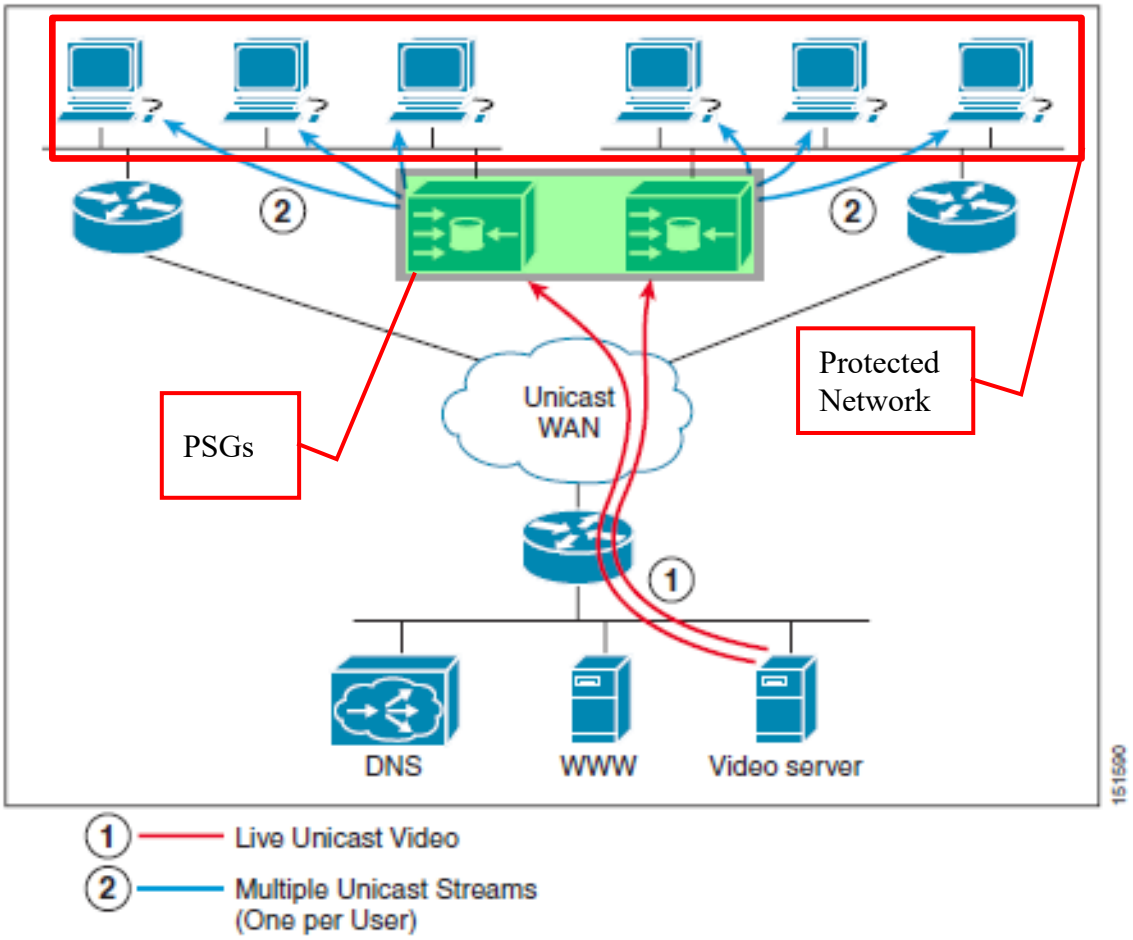
**6. Element [1.3]**

***[1.3] receiving, by a packet security gateway of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway;***

ACNS teaches element [1.3], requiring the gateways to receive "packets associated with a network protected by the packet security gateway." EX1004,

¶¶168-171. Content Engines (PSGs, shown in green, below) receive packets associated with a network behind and protected by said Content Engines.

***Figure 9-1 Unicast Live Stream Splitting***
(Annotated)



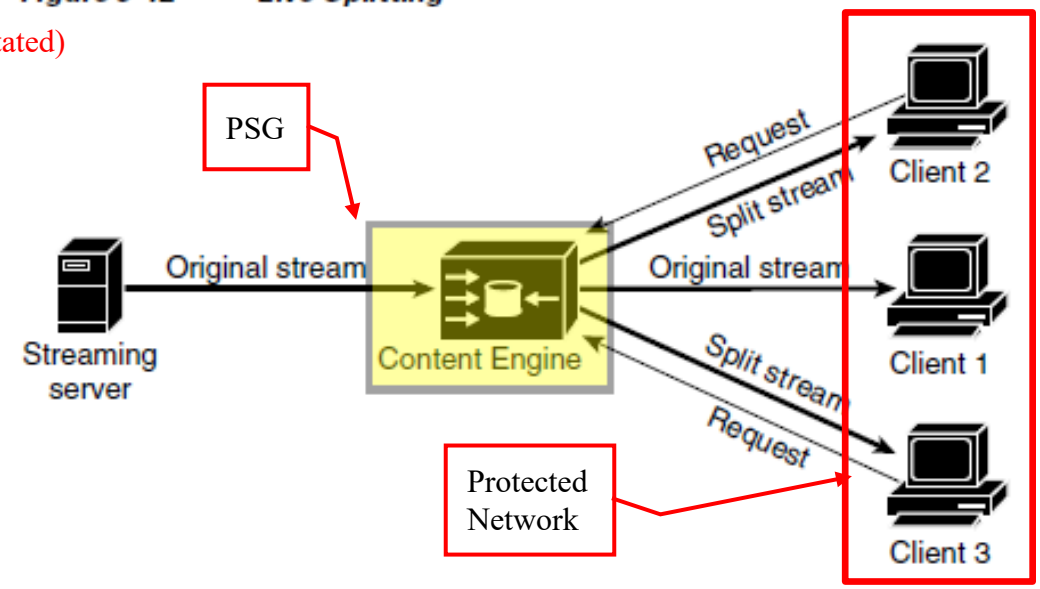

EX1008, 9-3, Figure 9-1; *see also* 9-5, Figure 9-3; EX1004, ¶¶169; EX1008, 1-1 ("Content Engines in an ACNS network save bandwidth and protect networks…"); 1-9 (describing network protections offered by content engines).

Additionally, ACNS teaches that Content Engines may be used as a proxy for a streaming server or broadcast server to replicate content for multiple recipients.

EX1008, 9-27. Figure 9-12 (annotated) shows the Content Engine (PSG) receiving packets associated with the network protected by the Content Engine:

*Figure 9-12 Live Splitting*

(Annotated)




EX1004, ¶170. As further evidence that the Content Engines receive packets associated with the networks protected by those Content Engines, ACNS describes how a user may view statistics for each Content Engine, including the number of packets sent and received. EX1004, ¶171. EX1008, 21-14 – 21-18.

### 7. Element [1.4]

***[1.4] identifying, by the packet security gateway, from amongst the packets associated with the network protected by the packet security gateway, and on a packet-by-packet basis, one or more packets comprising the application-layer packet-header information;***

ACNS teaches element [1.4], directed to the PSG identifying, on a packet-by-packet basis, packets that comprise application-layer packet-header information.

EX1004, ¶¶172-175. Content Engines examine the application-layer packet-headers of HTTP packets to determine whether the HTTP request method is supported by the Content Engine. EX1008, 8-23; EX1004, ¶173. The Content Engine identifies the packet as an HTTP (application-layer) packet and packet-header information meeting the criteria of having a supported HTTP request method. *Id*. A POSA would understand that such identification occurs on a packet-by-packet basis so that all packets are properly examined. *Id*.

Alternatively, ACNS teaches that a Content Engine examines this packet-header information through configuration of its ICAP server functionality, described above in relation to elements [1.1] and [1.2]. EX1008, Chap. 16, *generally*; EX1004, ¶158. ICAP is a protocol for communicating between Web proxy servers and is implemented on the ACNS Content Engines. *Id*. ACNS states:

> ICAP is an open standards protocol that can be used for content adaptation, typically at the network edge. Content adaptation includes virus scanning, content translation, content filtering, content insertion, and other ways of improving the value of content to end users.

EX1008, 16-1. ACNS teaches using ICAP rules to identify packets and the "rule actions" to apply. *Id*., EX1008, 16-15 – 16-25. Such rules include identifying packets using, for example, application-layer packet-header information such as URL and Request Method. EX1004, ¶160; EX1008, 16-16 – 16-18.

### 8. Element [1.5]

***[1.5] performing, by the packet security gateway and on a packet-by-packet basis, the packet transformation function on each of the one or more packets comprising the application-layer packet-header information, wherein the performing the packet transformation function comprises***

ACNS teaches element [1.5], relating to performing a packet transformation function on each packet containing application-layer packet-header information. EX1004, ¶¶176-178. ACNS teaches packet transformations that include (i) allowing or denying packets based on the included HTTP request method (EX1008, 8-23) and (ii) a packet digest logging function to notify the Content Engine to extract and log defined information from the packet-header (EX1008, 19-1 – 19-5). EX1004, ¶177. A POSA would understand that the ACNS system performs packet transformations on a packet-by-packet basis to avoid skipping packets and thereby avoid allowing invalid packets or refusing valid ones. EX1004, ¶177. *Id*.

The ACNS system implementation of ICAP, as described in Chapter 16 of ACNS, and described above for elements [1.1], [1.2] and [1.4], also demonstrates performing a packet transformation as claimed. EX1004, ¶178.

### 9. Element [1.6]

***[1.6] identifying a subset of information specified by the packet digest logging function for each of the one or more packets comprising the application-layer packet-header information;***

ACNS teaches element [1.6], directed to "identifying a subset of information specified by the packet digest logging function" to be logged. EX1004, ¶¶179-181. ACNS teaches logging in a variety of formats, including "apache," "extended-squid," "squid," and "custom" logging formats. EX1004, ¶¶180-181; EX1008, 19-1 – 19-5. Each format may contain a specified subset of information identified from the application packet, including, for example, information from the application-layer packet-header, such as "remotehost," "URL," "Request Method," and other information. EX1004, ¶181; *see also* EX1008, 19-1 – 19-5. ACNS provides the following example of a "squid" log entry (identified subset of application packet information, including application-layer packet-header information, emphasized):

> 1012429341.115 100 **172.16.100.152** TCP_REFRESH_MISS/304 1100 **GET** **http://www.cisco.com/images/homepage/news.gif** - DIRECT/www.cisco.com -

EX1004, ¶181; EX1008, 19-1. A POSA would have known that information such as the URL and HTTP method ("GET") are found in the header of an HTTP application-layer packet, and would further understand that any subset of information from the packet could be logged. EX1004, ¶181. Other, similar

teachings include the custom logging option, which includes identifying and logging a variety of subsets of information from the packet. *Id.*; EX1008, 19-2 – 19-5.

**10. Element [1.7]**

***[1.7] generating, for each of the one or more packets comprising the application-layer packet-header information, a record comprising the subset of information specified by the packet digest logging function;***

ACNS teaches element [1.7], directed to generating a record of the identified information. EX1004, ¶¶182-187. Logging, described above for element [1.6], involves generating, for each packet, a record comprising the specified subset of information. *See also* EX1004, ¶¶183-185. ACNS generates a record in the form of a "squid" log entry, containing information identified by the rule containing a packet digest logging function (identified subset of application packet information emphasized):

> 1012429341.115 100 **172.16.100.152** TCP_REFRESH_MISS/304 1100 **GET** **http://www.cisco.com/images/homepage/news.gif** - DIRECT/www.cisco.com -

EX1004, ¶¶181, 184-185; EX1008, 19-1. A POSA would understand that the ACNS software generates a record of the information (for example, as highlighted in the above example "squid" log entry) for use in writing a log entry in one of the specified or custom formats. EX1004, ¶186.

The ACNS system identifies and extracts application packet information and combines information not found therein (such as current timestamp, size of the packet, source and destination network-layer addresses, etc.) to generate an internal record before writing it in the specified format. *Id*.; EX1008 19-1 – 19-5. ACNS also teaches that the subset of information, already formatted in one log format, may be reformatted into syslog format. EX1004, ¶¶186-187, EX1008, 19-19 – 19-21. In that instance, the initial logging format constitutes a further record of the subset of information generated before reformatting. In at least all of these respects, ACNS teaches a POSA that a record is generated for each logged packet. EX1004, ¶¶186-187.

### 11. Element [1.8]

***[1.8] and reformatting, for each of the one or more packets comprising the application-layer packet-header information, the subset of information specified by the packet digest logging function in accordance with a logging system standard;***

ACNS teaches element [1.8], directed to reformatting the information into a logging system standard. EX1004, ¶¶188-191. As demonstrated above for elements [1.6] and [1.7], ACNS teaches that the identified application packet information from HTTP packets is reformatted to generate log files in logging formats such as "squid." *See also* EX1004, ¶189; EX1008, 19-1. ACNS also teaches that log files

may be generated in "extended-squid" and "apache" logging system standards. *Id*. In each case, the provided samples include a subset of information from HTTP packets. *Id*.

A POSA would understand that, to generate customized log files in these various standards, the ACNS system would reformat the packet information from the internal record described above for element [1.7] into the chosen logging system standard. EX1004, ¶190.

ACNS further teaches reformatting the information again for "Real-Time Transaction Logging," wherein the Content Engine reformats log files into the syslog standard format. EX1004, ¶191; EX1008, 19-19 – 19-22. This is the same reformatting process described in the '213 Patent. *See* EX1001, Col. 23:14-18. At least in all of these respects, ACNS teaches element [1.8].

### 12. Element [1.9]

***[1.9] and routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function.***

As explained in more detail above regarding element [1.2], Kjendal teaches routing identified packets to a monitoring device based on rules. EX1004, ¶¶192-198; EX1009, Col. 7:43-51 (describing "mirroring of the one or more packets or

portions of packets to a destination other than the original intended destination"), 27:48-56. Kjendal further teaches that selected packets may be forwarded to monitoring devices (mirrored) based on application-layer packet-header information. EX1004, ¶194; EX1009, Col. 8:59-9:3, 36:34-61, Fig. 12. Kjendal further teaches the use of dynamic mirroring policies in devices such as "application identification appliances, IDSs, network loggers, analyzers, etc." EX1004, ¶195, EX1009, Col. 46-58. Kjendal teaches a wide variety of devices to receive mirrored traffic for various reasons, including analysis, application identification, decryption, etc. EX1004, ¶196, EX1009, Col. 27:32-47. Kjendal teaches routing specific, policy-identified packets to a monitoring device ("logger"). EX1009, Col. 10:46-58. Kjendal combined with ACNS thus further renders obvious routing selected packets to monitoring devices.

As shown in more detail in the section discussing element [1.2], a POSA would have found it obvious to combine these teachings of Kjendal with ACNS. Kjendal teaches, for example, that its port mirroring and other operations may be practiced with "any bridgeable or routable protocol." EX1009, Col. 28:3-18. ACNS uses a number of routable protocols, such as TCP/IP, and would easily integrate the teachings of Kjendal. EX1004, ¶¶166; 197-198. Kjendal renders obvious element [1.9]. EX1004, ¶196.

A POSA would further understand that port-mirroring for monitoring of packets meeting selected rule-based criteria was well-known at the time of the alleged invention, and that such systems were routinely deployed on Cisco router systems. EX1004, ¶¶197-198. Therefore, a POSA would reasonably expect to include monitoring features to improve Cisco products such as ACNS, and would expect such a combination to yield predictable and beneficial results. *Id*.

For all the above reasons, ACNS and Kjendal render claim 1 obvious to a POSA.

### 13. Claim 2

***2. The method of claim 1, wherein the at least one rule indicates performing an accept packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet transformation function comprises forwarding, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information toward its respective destination.***

ACNS teaches all elements of claim 2, directed to forwarding packets that meet a rule's criteria to their destination address. EX1004 ¶¶199-201. ACNS teaches that certain HTTP Request Methods, found within the application-layer packet-header, may be accepted and forwarded. EX1004, ¶200; EX1008, 8-23 ("Content Engines **accept** or reject an HTTP request depending on whether the

HTTP request method is supported.")(emphasis added). A POSA would understand that "accept[ing]…an HTTP request" indicates that the HTTP packet will be accepted based on HTTP (application-layer) Request Method (packet-header) information and forwarded toward its destination. EX1004, ¶200. If an acceptable HTTP Request Method is found, the ACNS system forwards the packet to its destination. *Id*.; EX1008, 8-23; *see also* EX1008, 1-3 (explaining that the Content Engine acts as a "forward proxy" which receives requests from users and may either respond to the request with cached content, or forward the request to an outside server to obtain the content). A POSA would understand that the logging rules of Chapter 19 of ACNS may create log entries whenever the Content Engine performs a task, such as above. EX1004, ¶200. Therefore, logged packets may also be forwarded as part of packet transformation.

ACNS further teaches that, when acting as a caching server, the ACNS software, by rule, may forward the packet to its original destination based on the application-layer packet-header information. EX1004, ¶201; EX1008, 8-48 ("The Content Engine forwards the request to the proxy to which the client had originally addressed the request.").

**14. Claim 3**

***3. The method of claim 1, wherein the at least one rule indicates performing a deny packet transformation function on the packets comprising the application-***

***layer packet-header information, and wherein the performing the packet transformation function on comprises dropping, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information.***

ACNS teaches all elements of claim 3, directed to dropping packets that meet a rule's criteria. EX1004, ¶202-204. As described above for claim 2, a Content Engine may also reject an HTTP packet based on the Request Method contained therein. EX1004, ¶¶203-204; EX1008, 8-23. A POSA would understand that "reject[ing] an HTTP request" indicates that the packet will be rejected and dropped (i.e., that forwarding/routing will be "denied"). *Id*. Thus, based on the list-based rule in ACNS, if an HTTP request method is not included in the list of allowed methods the ACNS system will deny and drop said packet. *Id*.

**15. Claim 4**

***4. The method of claim 1, wherein the application-layer packet-header information identifies one or more hypertext transfer protocol (HTTP) packets, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets.***

ACNS teaches all elements of claim 4, directed to identifying certain HTTP packets. EX1004, ¶¶205-208. As previously detailed for claims 1-3, ACNS teaches a rule for identifying HTTP application packets and performing transformations based on data (the Request Method) contained within the application-layer packet-header information of the HTTP packets. EX1004, ¶¶206-207; EX1008, 8-23. "Content Engines **accept or reject an HTTP request depending on whether the HTTP request method is supported**. HTTP 1.1 request methods (for example, GET, HEAD, or POST) are supported by default." EX1008, 8-23 (emphasis added); EX1004, ¶¶206-207.

ACNS also teaches implementation of a URL / URI filter system (part of the ICAP rules discussed above in element [1.1]), which identifies and allows packets specifying HTTP methods such as "GET" and "PUT," based on the URI contained within the application-layer packet-header information of the HTTP packet. EX1004, ¶208; EX1008, 16-25 – 16-33. ACNS further explains, "[s]ome enterprises have a requirement to monitor, manage, and restrict employee access to nonbusiness and objectionable content on the Internet." EX1008, 16-25. It also teaches implementing URL filtering for a variety of protocols, including HTTP. EX1008, 16-25 – 16-33; *see* e.g., 16-28 (listing "Protocol URL Filter Settings for HTTP, RTSP, and WMT"); EX1004, ¶208.

### 16. Claim 5

***5. The method of claim 4, wherein the one or more HTTP packets comprise an HTTP GET method call, and wherein determining that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets comprises determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP GET method call.***

ACNS teaches every element of claim 5, which merely specifies the long-known HTTP GET method call among the rule criteria. EX1004, ¶¶209-212. HTTP GET requests a web server to retrieve and send content stored on that web server. EX1004 ¶209; EX1010, p. 35 §9.3. As shown above, for claim 4, ACNS teaches transforming HTTP packets based on the Request Method contained therein. *See* EX1008, 8-23 – 8-24. The provided example specifically mentions determining whether the application-layer packet-header information includes the HTTP GET call. *Id*. Furthermore, as shown above and via the examples in ACNS, the logging system identifies the HTTP packet and the Request Method (including "GET") in order to create a log entry for said packet containing the "GET" method, as shown in the provided example log entries. EX1004, ¶¶211-212; EX1008, 19-1 – 19-5.

### 17. Claim 6

***6. The method of claim 5, wherein the HTTP GET method call specifies a uniform resource identifier (URI), and wherein determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP GET method call comprises determining that the one or more packets comprising the application-layer packet-header information originated from or are destined for a network address corresponding to the URI.***

ACNS teaches each element of claim 6, which further specifies the common technique of using URIs in HTTP GET packets. EX1004, ¶¶213-219. A POSA would understand that a URI is an element of HTTP packets. EX1004, ¶¶48, 155-158, 184, 213; EX1010, p. 10 §1.4 ("The HTTP protocol is a request/response protocol. A client sends a **request to the server in the form of** a request method, **URI**, and protocol version, followed by…") (emphasis added). As detailed above for claims 4-5, ACNS teaches identifying HTTP packets associated with the HTTP GET method call. EX1004, ¶¶2015-212.

As further detailed above, ACNS teaches implementing URL filtering for HTTP packets containing a Request Method, such as GET, based on the URI contained within the HTTP (application-layer) packet-header. EX1004, ¶215; EX1008, 16-25 – 16-33. ACNS provides specific URL Filter list files, named in the

fields "Bad Site Filename" and "Good Site Filename" that include lists of URLs to which access is forbidden or allowed, respectively. EX1004, ¶216; EX1008, 16-28.

As also detailed above, the ACNS logging system (*see* EX1008, 19-1 – 19-5) provides examples showing the GET method, the associated URL of the HTTP packet, and the network source and destination of the packet. EX1004, ¶218. Specifically, in the example of "squid" log format, the log comprises the "peerhost" field to demonstrate whether the packet is being forwarded to the address of the URI or to another address. *Id*. In the example, a POSA would understand that the packet is destined for the address ("www.cisco.com") corresponding to the URI because the "peerhost" field is the same address as the URI. *Id*.

### 18. Claim 7

***7. The method of claim 4, wherein the one or more HTTP packets comprise an HTTP PUT method call, and wherein determining that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets comprises determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP PUT method call.***

ACNS teaches every element of claim 7, which specifies another long-known HTTP method call: PUT. EX1004, ¶¶220-223. HTTP PUT requests a server to create or replace content stored on said server. EX1004 ¶220; EX1010, p. 36 §9.6.

As shown above for claim 4, ACNS teaches identifying HTTP application packets and performing transformations based on the Request Method found therein. EX1004, ¶221; EX1008, 8-23 – 8-24. "HTTP 1.1 request methods (for example, GET, HEAD, or POST) are supported by default." EX1004, ¶222; EX1008, 8-23 – 8-24. A POSA would know that the specification for HTTP 1.1 is provided in RFC 2616 (EX1010). EX1004, ¶223. The HTTP 1.1 Specification includes eight Request Methods, including PUT. EX1004, ¶¶222-223; EX1010, 33-37. Therefore, a POSA would understand ACNS as teaching the identification of packets including the PUT Request Method, since all HTTP 1.1 methods are supported by default and PUT is one of a limited number of options. EX1004, ¶¶222-223. *See also* analysis, above, for claim 5.

### 19. Claim 8

***8. The method of claim 7, wherein the HTTP PUT method call specifies a uniform resource identifier (URI), and wherein determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP PUT method call comprises determining that the one or more packets comprising the application-layer packet-header information originated from or are destined for a network address corresponding to the URI.***

ACNS teaches every element of claim 8, which further specifies using URIs in HTTP PUT calls. EX1004, ¶¶224-230. For the same reasons listed, above, for

claim 6, relating to the use of addresses corresponding to URIs within HTTP GET packets, and claim 7, regarding the ACNS teaching of the HTTP PUT method, ACNS also teaches all elements of claim 8. A POSA would understand that addresses in PUT calls would be useful criteria for packet rules. *Id.*

### 20. Claim 9

***9. The method of claim 1, wherein the at least one rule comprises a five-tuple specifying one or more transport-layer protocols, a range of source addresses, a range of source ports, a range of destination addresses, and a range of destination ports, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that each of the one or more packets comprising the application-layer packet-header information corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range of source ports, a destination address within the range of destination addresses, and a destination port within the range of destination ports.***

ACNS teaches every element of claim 9, directed to using the "five-tuple" of protocol, source address, destination address, source port, and destination port as identification criteria. EX1004, ¶¶231-234. The "five-tuple" was well-known to a

POSA as a source of packet identification for implementing rules at the time of alleged invention. EX1004, ¶¶231-232.

Chapter 17 of ACNS teaches filtering packets using Access Control Lists based on a number of criteria, including a five-tuple as claimed. Table 17-4 of ACNS, reproduced below, identifies each of the claimed elements as follows (highlighted in yellow): a transport-layer protocol (listed as the "Extended Type" field and identifying TCP as the default), "Source IP" address, "Source Port 1," "Destination IP" address and "Destination Port 1." EX1004, ¶¶233-234; EX1008, Chapter 17, *generally*; Table 17-4 (showing control list filtering based on all five-tuple elements). Further, the "operator" and "wildcard" elements in Table 17-4 (highlighted in green) allow the user to specify a "range" of source and destination addresses and ports to use as filtering criteria. EX1004, ¶¶233-234.

*Table 17-4 Extended IP ACL TCP Condition*

| Field | Default Value | Description |
|---|---|---|
| Purpose*[1] | Permit | Specifies whether a packet is to be passed (**Permit**) or dropped (**Deny**). |
| Extended Type* | TCP | Matches the TCP Internet protocol. |
| Established | Unchecked (false) | When checked, a match with the ACL condition occurs if the TCP datagram has the ACK or RST bits set, indicating an established connection. Initial TCP datagrams used to form a connection are not matched. |
| Source IP* | 0.0.0.0 | Number of the network or host from which the packet is being sent, specified as a 32-bit quantity in 4-part dotted decimal format. |
| Source IP Wildcard* | 255.255.255.255 | Wildcard bits to be applied to the source, specified as a 32-bit quantity in 4-part dotted decimal format. Place a 1 in the bit positions that you want to ignore and identify bits of interest with a 0. |
| Source Port 1 | 0 | Decimal number or name of a TCP port. Valid port numbers are 0 to 65535. Valid TCP port names are as follows:<br>• ftp<br>• ftp-data<br>• https<br>• netbios-dgm<br>• netbios-ns<br>• netbios-ss<br>• nfs<br>• rtsp<br>• ssh<br>• telnet<br>• www |
| Source Operator | range | Specifies how to compare the source ports against incoming packets. Choices are <, >, ==, !=, or range. |
| Source Port 2 | 65535 | Decimal number or name of a TCP port. See Source Port 1. |
| Destination IP | 0.0.0.0 | Number of the network or host to which the packet is being sent, specified as a 32-bit quantity in 4-part dotted decimal format. |
| Destination IP Wildcard | 255.255.255.255 | Wildcard bits to be applied to the source, specified as a 32-bit quantity in 4-part dotted decimal format. Place a 1 in the bit positions that you want to ignore and identify bits of interest with a 0. |

*Table 17-4 Extended IP ACL TCP Condition (continued)*

| Field | Default Value | Description |
|---|---|---|
| Destination Port 1 | 0 | Decimal number or name of a TCP port. Valid port numbers are 0 to 65535. Valid TCP port names are as follows:<br>• ftp<br>• ftp-data<br>• https<br>• netbios-dgm<br>• netbios-ns<br>• netbios-ss<br>• nfs<br>• rtsp<br>• ssh<br>• telnet<br>• www |
| Destination Operator | range | Specifies how to compare the destination ports against incoming packets. Choices are <, >, ==, !=, or range. |
| Destination Port 2 | 65535 | Decimal number or name of a TCP port. See Destination Port 1. |

A POSA would understand that the Access Control Lists, as defined in Chapter 17, would be used in conjunction with the previously described application-layer packet-header identification methods, thus meeting the elements of claim 9 (which as a "comprises" claim includes but is not limited to using the five-tuple criteria). EX1004, ¶234.

**B. Ground 2 – ACNS in View of Kjendal and Diffserv (Claims 10-16)**

A POSA would have found the subject matter of claims 10-16 of the '213 patent obvious over ACNS in light of Kjendal and in further light of Diffserv. EX1004, ¶236.

### 1. Overview of Diffserv

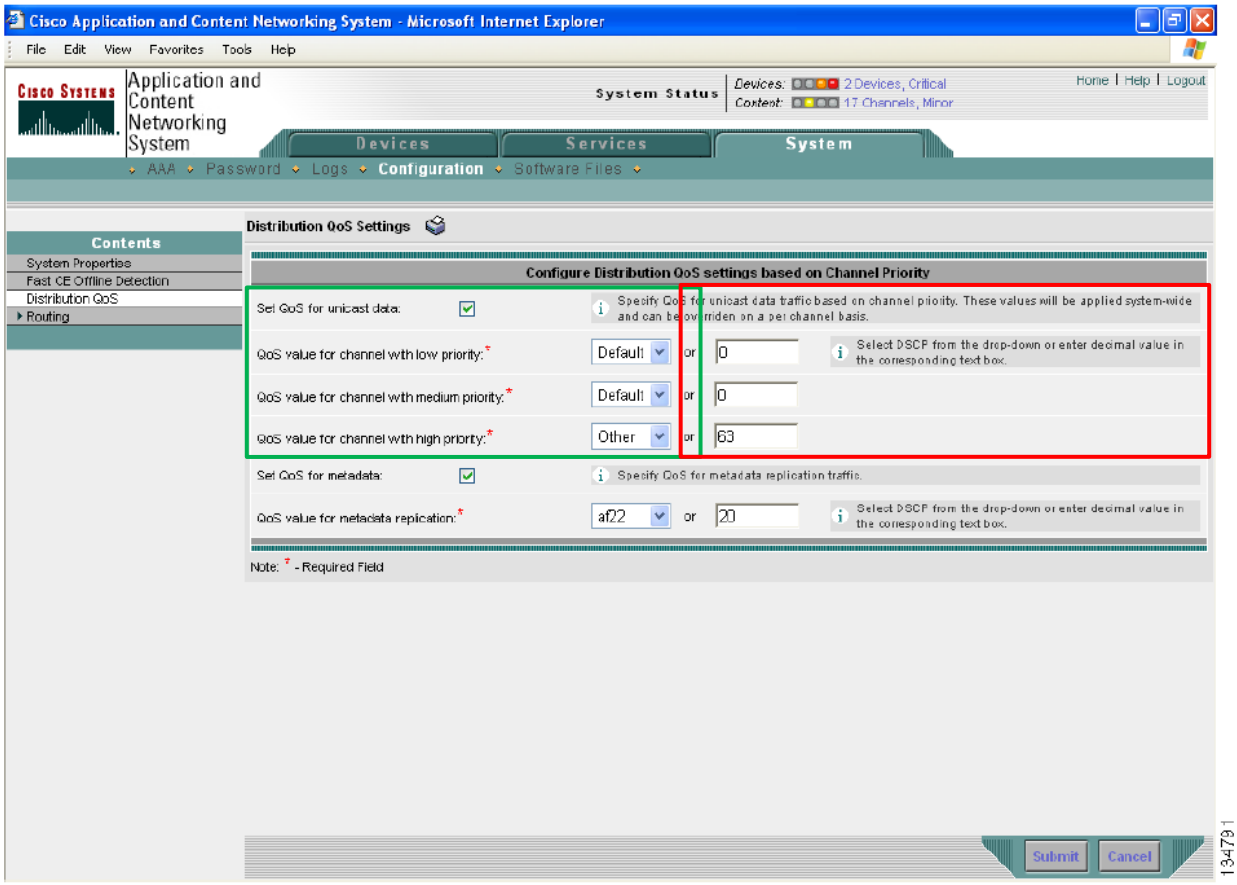
Diffserv lays out the general architecture for differentiated services, including a detailed specification for the Differentiated Services Code Point (DSCP) field. EX1011; EX1004, ¶¶68-71, 238-240. Differentiated services, generally, allow a system to give priority to certain packets over others. EX1004, ¶¶68-71. For example, a system implementing differentiated services, through a mechanism like Diffserv, may give higher priority to voice-over-IP packets than to HTTP packets, because the former are more susceptible to lag or delay. *Id*. A POSA would readily understand that the Diffserv architecture is designed for software implementation and easily adaptable to implement differentiated service in systems such as the ACNS system. EX1004, ¶¶239-240.

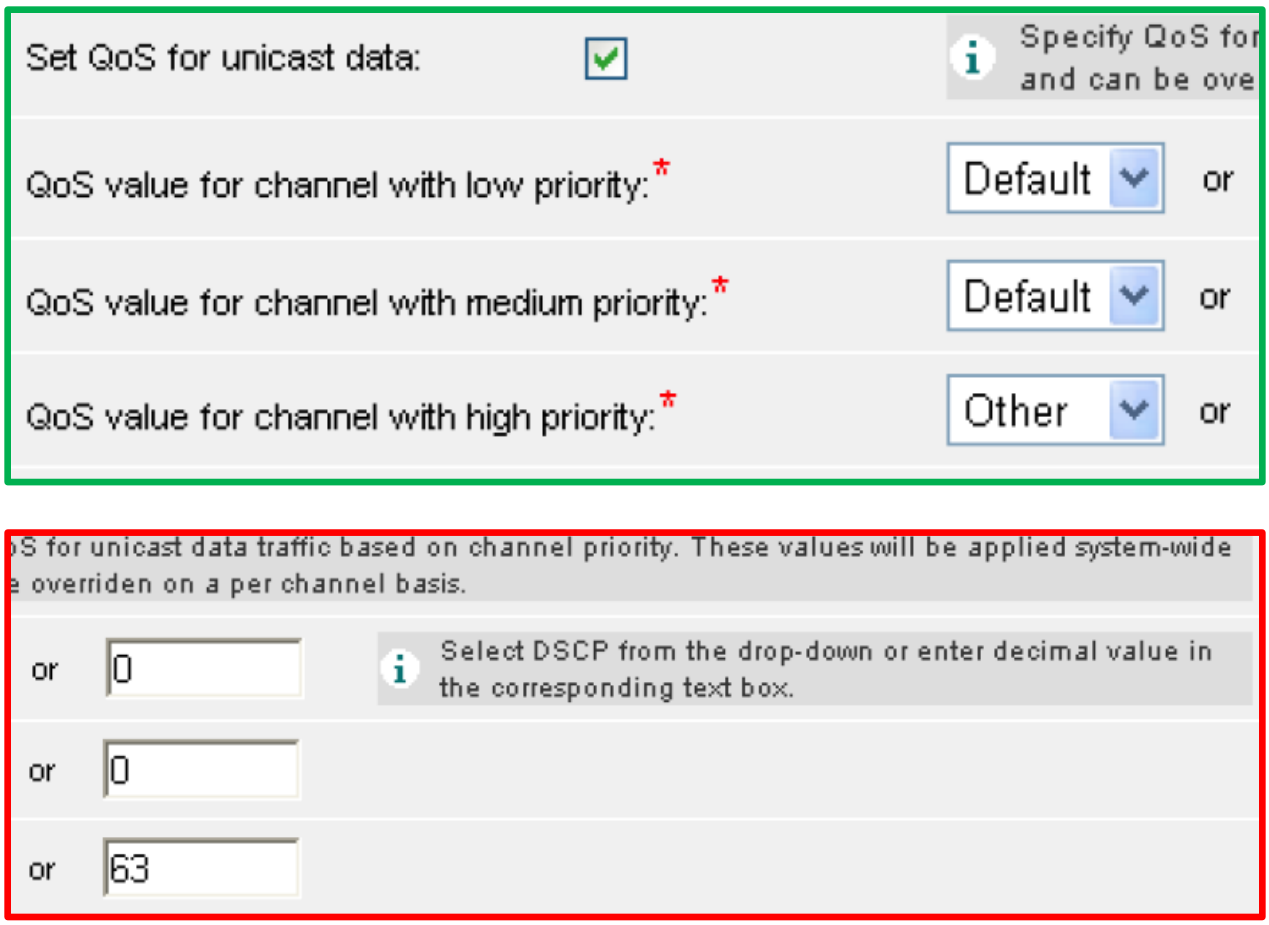
Diffserv lists its copyright date as December 1998, and Dr. Jeffay has personal knowledge that it has been freely available to the public since approximately that time. EX1011; EX1004, ¶129-133. Diffserv is a flexible standard for marking each network packet with a priority value within the DSCP field. EX1004, ¶¶68-71; EX1011, §§2, 2.3.1 (describing general architecture of Diffserv and describing "classifiers" for prioritizing packets based on DSCP, according to a rule defined within the system).

### 2. Motivations to Combine Diffserv with ACNS and Kjendal

Upon reading ACNS, a POSA would be naturally motivated to consider its teachings in light of Diffserv. EX1004, ¶¶237-240. ACNS teaches that the ACNS System may implement differentiated services that utilize the DSCP selector. EX1004, ¶238; *see* EX1008, 5-10 – 5-14; 10-9; 16-19 – 16-23; and 20-16 – 20-18 The following ACNS screenshot is of its differentiated services management screen (referring to "QoS" or quality of service), with magnified portions provided for clarity:

***Figure 20-5 Distribution QoS Settings Window***






EX1004, ¶238; EX1008, 20-17 (elements of GUI magnified for clarity, green box is left side; red box is right side, showing use of DSCP within the ACNS QoS [Quality of Service] configuration). In accordance with Diffserv, the ACNS system uses DSCP values carried in (network-layer) packet headers to enable Content Engines to classify packets and provide better-than-best effort processing to the classified packets—i.e. quality of service differentiation. EX1011, §§2.3.1-2.3.2. Filtering packets based on the already-included DSCP field would be an obvious variant to a POSA, viewing ACNS, given that filtering may occur based on any network packet

field. EX1004, ¶¶238-240; 247-249. Diffserv, moreover, explicitly teaches such filtering as a purpose for the DSCP field. EX1004, ¶248; EX1011, §2.3.1.

Kjendal teaches that a POSA would recognize and appreciate that its port mirroring and other operations may be practiced with "any bridgeable or routable protocol," such as TCP/IP, which underlies ACNS and all of the HTTP protocol. EX1009, Col. 28:3-18. Kjendal further teaches use of mirroring with network services including QoS. EX1009, Col. 2:14-16; 3:1-7. Based on the teachings of ACNS and Kjendal, a POSA would expect to combine these references with the teachings of Diffserv to obtain predictable results without undue experimentation. EX1004, ¶240.

### 3. Claim 10 – Element [10.1]

***[10.1] A method comprising: receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying packet-identification criteria***

The combination of ACNS and Diffserv teaches element [10.1]. EX1004, ¶¶2420249. Claim element [10.1] is similar to element [1.1] except that element [10.1] more broadly recites that the rule includes packet-identification criteria, rather than application-layer packet-header information. EX1004, ¶243. Therefore, any

teaching that satisfies [1.1] also satisfies [10.1] and all discussion of ACNS for [1.1] is incorporated here. *Id*.

Diffserv also teaches that a system using Diffserv may "select packets in a traffic stream based on the content of some portion of the packet header" and may specifically select packets either "based on the DS codepoint [i.e. DSCP] only" or based on a combination of the DSCP and other information found in the packet header. EX1004, ¶¶247-248; EX1011, § 2.3.1. These "classifier" rules include packet identification criteria for the purposes of claim 10 and its dependent claims. *Id*.

Based on the teachings of ACNS and Diffserv, as discussed above, a POSA would combine these specific teachings. EX1004, ¶249.

#### 4. Element [10.2]

***[10.2] and a packet transformation function comprising a packet digest logging function to be performed on packets corresponding to the packet-identification criteria***

The combination of Kjendal and ACNS teaches element [10.2]. EX1004, ¶¶250-252. This element is identical to element [1.2], regarding a packet digest logging function, except for the packet criteria used. EX1004, ¶250. Element [1.2] refers to packets "comprising the application-layer packet-header information" rather than "corresponding to the packet-identification criteria." Elements [10.5],

[10.6], [10.7], [10.8], [10.9], and [10.10] are correspondingly similar to their Claim 1 counterparts, except for the same difference in packet identification criteria. The broader selection of packet criteria does not change the functional aspects of the claim element, so the discussion of element [1.2] is equally applicable, here, and is incorporated by reference. *Id*.

**5. Element [10.3]**

***[10.3] wherein the packet-identification criteria comprises a Differentiated Service Code Point (DSCP) selector;***

The combination of ACNS and Diffserv teaches element [10.3], using DSCP as a packet identifier criteria. EX1004, ¶¶253-256. As detailed above, ACNS teaches the use of Diffserv mechanisms, especially the DSCP field, and it would have been obvious to a POSA, based solely on ACNS, to use the included DSCP field to identify and filter packets, given that ACNS teaches a variety of methods for filtering packets at the network level. *Id*.; *see also* EX1008, Chapter 17, generally (describing "access control lists" (ACL) for filtering packets based on a variety of network-level fields)

It would be further obvious to a POSA to combine the teachings of Diffserv with those of ACNS, as Diffserv explicitly teaches identifying packets based on the DSCP field. EX1004, ¶247. Diffserv teaches that a system using Diffserv may "select packets in a traffic stream based on the content of some portion of the packet

header" and may specifically select packets either "based on the DS codepoint [i.e. DSCP] only" or based on a combination of the DSCP and other information found in the packet header. EX1011 § 2.3.1. These classifier rules are also nearly identical to the various criteria for selecting and filtering packets described in ACNS and, because of ACNS's explicit implementation of the Diffserv architecture, it would be obvious to a POSA to combine these references to select or filter packets based, at least in part, on the DSCP field. EX1004, ¶¶254-256.

As such, element [10.3] would have been obvious to a POSA in view of ACNS alone, or in combination with Diffserv.

**6. Element [10.4]**

***[10.4] receiving, by a packet security gateway of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway;***

The combination of ACNS and Diffserv teaches element [10.4]. EX1004, ¶¶257-260. This claim element is identical to element [1.3] regarding gateways receiving packets, and analysis of that element is incorporated here.

**7. Element [10.5]**

***[10.5] identifying, by the packet security gateway, from amongst the packets associated with the network protected by the packet security gateway, and on a***

***packet-by-packet basis, one or more packets corresponding to the packet-identification criteria;***

The combination of ACNS and Diffserv teaches element [10.5], directed to the PSG identifying packets that comprise packet-identification criteria. EX1004, ¶¶261-265. Element [10.5] is similar to element [1.4] except that the packet-identification criteria of claim 10 is not limited to application-layer packet-header information but requires the inclusion of the DSCP field. EX1004, ¶261. The analysis of element [1.4] is incorporated by reference.

ACNS teaches that a Content Engine identifies packets comprising a DSCP field in the network packet header because ACNS already teaches access control lists for identifying and filtering packets based on a variety of other network-layer header fields, such as source address, destination address. EX1004, ¶263; EX1008, 17-1 – 17-8. It would be obvious to a POSA to combine the specific teachings of Diffserv with the system disclosed by ACNS because the ACNS system specifically contemplates such a combination. EX1004, ¶263. ACNS teaches the presence of the DSCP field in packets and the implementation of QoS based on the DSCP fields. *Id*. Diffserv would supplement this by teaching specifically identifying packets based on the DSCP field.

As detailed above, Diffserv teaches identifying packets based in whole or in part on the DSCP field on a packet-by-packet basis. EX1004, ¶264; EX1011 §2.3.1.

### 8. Element [10.6]

***[10.6] performing, by the packet security gateway and on a packet-by-packet basis, the packet digest logging function on each of the one or more packets corresponding to the packet-identification criteria, wherein the performing the packet digest logging function comprises:***

The combination of ACNS and Diffserv teaches element [10.6], directed to performing a packet digest logging function. EX1004, ¶¶266-271. Element [10.6] is similar to element [1.5]. Thus, the analysis of element [1.5] is incorporated by reference. *See* EX1004, ¶¶267-271; EX1008, Chapter 19, *generally*.

### 9. Element [10.7]

***[10.7] identifying a subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria;***

The combination of ACNS and Diffserv teaches element [10.7] for the same reasons articulated above, for element [1.6]. EX1004, ¶¶272-277. This claim is identical to element [1.6] except [10.7] recites packets "corresponding to the packet-identification criteria" whereas [1.6] recites "comprising the application-layer packet-header information." As this distinction is fully addressed in the above discussion of claim 10 including element [10.5], the analysis of elements [1.6] and [10.5] is incorporated by reference and shows this element is obvious.

### 10. Element [10.8]

***[10.8] generating, for each of the one or more packets corresponding to the packet-identification criteria, a record comprising the subset of information specified by the packet digest logging function;***

ACNS teaches the record generating element [10.8]. EX1004, ¶¶278-279. This claim is functionally the same as element [1.7] except as to how the packets are identified, which is addressed fully as to element [10.5] and has no effect on the functional aspect of the element. The analysis of [1.7] and [10.5] is applicable and incorporated by reference, and shows this element is obvious. *Id*.

### 11. Element [10.9]

***[10.9] and reformatting, for each of the one or more packets corresponding to the packet-identification criteria, the subset of information specified by the packet digest logging function in accordance with a logging system standard;***

ACNS teaches reformatting element [10.9]. EX1004, ¶280. This element is functionally identical to claim element [1.8], except for how the packets are identified, which is fully addressed as to element [10.5] and makes no functional difference here. *Id*. The analysis of [1.8] and [10.5] is applicable and incorporated by reference to show that this element is obvious.

### 12. Element [10.10]

***[10.10] and routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding***

***to the packet-identification criteria in response to the performing the packet digest logging function.***

The combination of ACNS and Kjendal teaches element [10.10], regarding routing packets to a monitoring device. EX1004, ¶¶281-282. This element is very similar to element [1.9], except for how the packets are identified as addressed for element [10.5], and the analysis of elements [1.9] and [10.5] is incorporated by reference to show this element is obvious.

### 13. Claim 11

***11. The method of claim 10, wherein the subset of information comprises at least one of a portion of data from the packet, a source address of the packet, a source port of the packet, a destination address of the packet, a destination port of the packet, a transport-protocol type of the packet, a uniform resource identifier (URI) from the packet, an arrival time of the packet, a size of the packet, a flow direction of the packet, an identifier of an interface of the packet security gateway that received the packet, or one or more media access control (MAC) addresses associated with the packet.***

ACNS and Diffserv each teaches every element of claim 11, which merely specifies that the subset of information comprises one of a set of listed criteria. EX1004, ¶¶283-287. ACNS teaches that a Content Engine may identify several of the claimed criteria from various identified packets as information to log: "Typical

fields in the transaction log are **the date and time when a request was made, the URL that was requested**, . . . , **the number of bytes transferred, and the source IP address**." EX1004, ¶¶284-286; EX1008, 19-1 (emphasis added); *see also* 19-3 – 19-7 (custom logging includes additional claimed fields for logging).

Diffserv also teaches specific information that should be gathered for such a log record, including several of the criteria recited in claim 11: "… the generated audit log entry should include **the date/time the packet was received,** [and] **the source and destination IP addresses**." EX1004, ¶287; EX1011 § 6.1 (emphasis added). A POSA would be motivated to combine the teachings of ACNS and Diffserv based on the specific teachings of each. EX1004, ¶287. This analysis, together with the analysis of claim 10 on which claim 11 depends, renders claim 11 obvious.

### 14. Claim 12

***12. The method of claim 10, comprising: temporarily storing data in a memory of the packet security gateway, the data comprising the subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria; and utilizing, by the packet security gateway, the data to generate a message comprising the subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria.***

ACNS teaches all elements of claim 12, directed to temporarily storing the specified subset as data in memory of the PSG to generate a message containing that information. EX1004, ¶288-290. As discussed above, in element [1.8], ACNS teaches "real time transaction logging" to generate log messages and send them to a remote syslog server. EX1004, ¶290; EX1008, 19-19 – 19-22. ACNS describes "an example of the format of the real-time syslog message sent from the transaction logging module (Content Engine) to the remote syslog host." EX1008, 19-19. The generated syslog message comprises the subset of information (contained in the local logging record). EX1004, ¶290; EX1008, 19-19. ACNS teaches that this message is generated (and thus temporarily stored in a memory) at the Content Engine (i.e. the claimed PSG) and then sent to a "remote syslog server." *Id*. Temporarily storing such a message in memory is an obvious step because a computer commonly stores messages in some form of memory, temporarily, before sending them. EX1004, ¶290. This analysis, together with the analysis of claim 10 on which claim 12 depends, renders claim 12 obvious.

### 15. Claim 13

***13. The method of claim 12, comprising communicating, by the packet security gateway and to the security policy management server, the message comprising the subset of information specified by the packet digest logging***

***function for each of the one or more packets corresponding to the packet-identification criteria.***

ACNS teaches the element of claim 13, sending the message of claim 12 to an SPM Server. EX1004, ¶¶291-294. As discussed for claim 12, the ACNS Guide teaches the common-sense step of sending the message comprising the logged subset of information to a remote syslog server. *Id*.; EX1008 19-19 – 19-21.

ACNS provides instructions for how to configure the destination of the syslog messages. EX1004, ¶294; EX1008, 19-20 – 19-21. A POSA would understand that among the limited number of options available, a likely location for the syslog server that receives log messages would be the same machine that houses the CDM (acting as the SPM Server). EX1004, ¶294. In fact, such a location would be preferable, as it would allow an administrator to easily and efficiently search said log file entries. *Id*. These teachings, together with the analysis of claims 10 and 12 on which claim 13 depends, render claim 13 obvious.

### 16. Claim 14

***14. The method of claim 10, wherein reformatting the subset of information specified by the packet digest logging function in accordance with the logging system standard comprises reformatting the subset of information specified by the packet digest logging function in accordance with a syslog logging system standard.***

ACNS teaches all elements of claim 14, wherein reformatting comprises reformatting in the syslog standard. EX1004, ¶¶295-297. The '213 Patent specification admits that logging can be implemented using a known or "standard" logging format called "SysLog;" the PSG "store[s] and/or forward[s] packet logs using a logging system based on a standard (e.g., syslog, or the like)." *Id*., Col. 11:54-57. As discussed above regarding claims 12 and 13, ACNS teaches that information may be reformatted into the syslog system standard. Those arguments are incorporated by reference and, together with the analysis of claim 10 on which claim 14 depends, render claim 14 obvious. EX1004, ¶295-297; EX1008, 19-19 – 19-22.

### 17. Claim 15

***15. The method of claim 10, wherein the packet-identification criteria comprises a five-tuple specifying one or more transport-layer protocols, a range of source addresses, a range of source ports, a range of destination addresses, and a range of destination ports, and wherein identifying the one or more packets corresponding to the packet-identification criteria comprises determining by the packet security gateway that each of the one or more packets corresponding to the packet-identification criteria corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range of source ports, a destination address***

***within the range of destination addresses, and a destination port within the range of destination ports.***

ACNS teaches every element of claim 15. EX1004, ¶298. Claim 15 is functionally identical to claim 9, except for referring specifically to the "packet identification criteria" of the rule. *Id*. As discussed for claim 9, ACNS teaches rules comprising the five-tuple as packet identification criteria. That analysis is incorporated here and, together with the analysis of claim 10 on which claim 15 depends, renders claim 15 obvious. *Id*.

**18. Claim 16**

***16. The method of claim 10, wherein the packet-identification criteria comprises application-layer packet-header information, and wherein identifying the one or more packets corresponding to the packet-identification criteria comprises determining by the packet security gateway that each of the one or more packets corresponding to the packet-identification criteria comprises at least a portion of the application-layer packet-header information.***

ACNS teaches all elements of claim 16, which specifies application-layer packet-header information as criteria for the rule of claim 10. EX1004, ¶299-302. As described for claim 1, elements [1.1] ("comprises at least one rule specifying application-layer packet-header information") and [1.4] ("identifying… one or more packets comprising the application-layer packet-header information"), ACNS

teaches rules specifying application-layer packet-header information, and identifying packets comprising application-layer packet-header information. The analysis found in claim elements [1.1] and [1.4] is incorporated by reference and, together with the analysis of claim 10 on which claim 16 depends, shows this claim is obvious. *Id*.

## X. CONCLUSION

For the reasons above, Petitioner asks that the Patent Office order an *inter partes* review trial for claims 1-16 and cancel these claims as unpatentable.

Respectfully submitted,

Date: August 20, 2018

/Daniel W. McDonald/
Daniel W. McDonald
Counsel for Petitioner
Registration No. 32,044

## XI. CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned attorney for the Petitioner, Cisco Systems, Inc., declares that the argument section of this Petition has 13,685 words, according to the word count tool in Microsoft Word™.

/Daniel W. McDonald/
Daniel W. McDonald
Counsel for Petitioner
Registration No. 32,044

## Appendix A – Claim Listing

| Claim or Element # | Claim Language |
|---|---|
| [1.0] | A method comprising: receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying application-layer packet-header information |
| [1.1] | and a packet transformation function comprising a packet digest logging function to be performed on packets comprising the application-layer packet-header information; |
| [1.2] | receiving, by a packet security gateway of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway; |
| [1.3] | identifying, by the packet security gateway, from amongst the packets associated with the network protected by the packet security gateway, and on a packet-by-packet basis, one or more packets comprising the application-layer packet-header information; |
| [1.4] | performing, by the packet security gateway and on a packet-by-packet basis, the packet transformation function on each of the one or more packets comprising the application-layer packet-header information, wherein the performing the packet transformation function comprises |
| [1.5] | identifying a subset of information specified by the packet digest logging function for each of the one or more packets comprising the application-layer packet-header information; |
| [1.6] | generating, for each of the one or more packets comprising the application-layer packet-header information, a record comprising the subset of information specified by the packet digest logging function; |
| [1.7] | and reformatting, for each of the one or more packets comprising the application-layer packet-header information, the subset of |

| Claim or Element # | Claim Language |
|---|---|
| | information specified by the packet digest logging function in accordance with a logging system standard; |
| [1.8] | and routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function. |
| 2 | The method of claim 1, wherein the at least one rule indicates performing an accept packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet transformation function comprises forwarding, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information toward its respective destination. |
| 3 | The method of claim 1, wherein the at least one rule indicates performing a deny packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet transformation function on comprises dropping, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information. |
| 4 | The method of claim 1, wherein the application-layer packet-header information identifies one or more hypertext transfer protocol (HTTP) packets, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets. |
| 5 | The method of claim 4, wherein the one or more HTTP packets comprise an HTTP GET method call, and wherein determining that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets comprises determining that the one or more packets comprising the |

| Claim or Element # | Claim Language |
|---|---|
| | application-layer packet-header information are associated with the HTTP GET method call. |
| 6 | The method of claim 5, wherein the HTTP GET method call specifies a uniform resource identifier (URI), and wherein determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP GET method call comprises determining that the one or more packets comprising the application-layer packet-header information originated from or are destined for a network address corresponding to the URI. |
| 7 | The method of claim 4, wherein the one or more HTTP packets comprise an HTTP PUT method call, and wherein determining that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets comprises determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP PUT method call. |
| 8 | The method of claim 7, wherein the HTTP PUT method call specifies a uniform resource identifier (URI), and wherein determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP PUT method call comprises determining that the one or more packets comprising the application-layer packet-header information originated from or are destined for a network address corresponding to the URI. |
| 9 | The method of claim 1, wherein the at least one rule comprises a five-tuple specifying one or more transport-layer protocols, a range of source addresses, a range of source ports, a range of destination addresses, and a range of destination ports, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that each of the one or more packets comprising the application-layer packet-header information corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range |

| Claim or Element # | Claim Language |
|---|---|
| | of source ports, a destination address within the range of destination addresses, and a destination port within the range of destination ports. |
| [10.0] | A method comprising: receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying packet-identification criteria |
| [10.1] | and a packet transformation function comprising a packet digest logging function to be performed on packets corresponding to the packet-identification criteria |
| [10.2] | wherein the packet-identification criteria comprises a Differentiated Service Code Point (DSCP) selector; |
| [10.3] | receiving, by a packet security gateway of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway; |
| [10.4] | identifying, by the packet security gateway, from amongst the packets associated with the network protected by the packet security gateway, and on a packet-by-packet basis, one or more packets corresponding to the packet-identification criteria; |
| [10.5] | performing, by the packet security gateway and on a packet-by-packet basis, the packet digest logging function on each of the one or more packets corresponding to the packet-identification criteria, wherein the performing the packet digest logging function comprises: |
| [10.6] | identifying a subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria; |
| [10.7] | generating, for each of the one or more packets corresponding to the packet-identification criteria, a record comprising the subset of information specified by the packet digest logging function; |

| Claim or Element # | Claim Language |
|---|---|
| [10.8] | and reformatting, for each of the one or more packets corresponding to the packet-identification criteria, the subset of information specified by the packet digest logging function in accordance with a logging system standard; |
| [10.9] | and routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the packet-identification criteria in response to the performing the packet digest logging function. |
| 11 | The method of claim 10, wherein the subset of information comprises at least one of a portion of data from the packet, a source address of the packet, a source port of the packet, a destination address of the packet, a destination port of the packet, a transport-protocol type of the packet, a uniform resource identifier (URI) from the packet, an arrival time of the packet, a size of the packet, a flow direction of the packet, an identifier of an interface of the packet security gateway that received the packet, or one or more media access control (MAC) addresses associated with the packet. |
| 12 | The method of claim 10, comprising: temporarily storing data in a memory of the packet security gateway, the data comprising the subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria; and utilizing, by the packet security gateway, the data to generate a message comprising the subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria. |
| 13 | The method of claim 12, comprising communicating, by the packet security gateway and to the security policy management server, the message comprising the subset of information specified by the packet digest logging function for each of the one or more packets corresponding to the packet-identification criteria. |
| 14 | The method of claim 10, wherein reformatting the subset of information specified by the packet digest logging function in accordance with the logging system standard comprises reformatting |

| Claim or Element # | Claim Language |
|---|---|
| | the subset of information specified by the packet digest logging function in accordance with a syslog logging system standard. |
| 15 | The method of claim 10, wherein the packet-identification criteria comprises a five-tuple specifying one or more transport-layer protocols, a range of source addresses, a range of source ports, a range of destination addresses, and a range of destination ports, and wherein identifying the one or more packets corresponding to the packet-identification criteria comprises determining by the packet security gateway that each of the one or more packets corresponding to the packet-identification criteria corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range of source ports, a destination address within the range of destination addresses, and a destination port within the range of destination ports. |
| 16 | The method of claim 10, wherein the packet-identification criteria comprises application-layer packet-header information, and wherein identifying the one or more packets corresponding to the packet-identification criteria comprises determining by the packet security gateway that each of the one or more packets corresponding to the packet-identification criteria comprises at least a portion of the application-layer packet-header information. |

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | |
|---|---|
| Cisco Systems, Inc. | Second Petition for Inter Partes Review |
| Petitioner | U.S. Patent No. 9,565,213 |
| | Case No.: IPR2018-01512 |

## CERTIFICATE OF SERVICE

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

| | |
|---|---|
| *Date of service* | August 20, 2018 |
| *Manner of service* | EMAIL and UPS OVERNIGHT DELIVERY |
| *Documents served* | Second Petition for *Inter Partes* Review, Power of Attorney and Exhibits shown in table below |

| | | |
|---|---|---|
| EX1001 | U.S. Patent No. 9,565,213 (the ’213 Patent) | Served in Paper and electronic form on flash drive |
| EX1002 | Prosecution History of the ’213 Patent | Served in electronic form on flash drive |
| EX1003 | CV of Dr. Kevin Jeffay | Served in electronic form on flash drive |
| EX1004 | Declaration of Dr. Kevin Jeffay, dated August 15, 2018 | Served in electronic form on flash drive |

| | | |
|---|---|---|
| EX1005 | Patent Owner's Opening *Markman* Brief in Related Litigation, *Centripetal Networks, Inc. v. Keysight Tech. Inc.*, Case 2:17-cv-00383-HCM-LRL ("*Keysight* lawsuit"), Doc. 106 (filed April 20, 2018) | Served in electronic form on flash drive |
| EX1006 | Patent Owner's Rebuttal *Markman* Brief in *Keysight* lawsuit, Doc. 117 (filed May 3, 2018) | Served in electronic form on flash drive |
| EX1007 | Parties' Joint Claim Construction Chart in *Keysight* lawsuit, Doc. 124, Ex. 1 (Filed May 11, 2018) | Served in electronic form on flash drive |
| EX1008 | Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5 ("ACNS Guide") | Served in electronic form on flash drive |
| EX1009 | U.S. Patent No. 9,172,627 ("Kjendal") | Served in electronic form on flash drive |
| EX1010 | Hypertext Transfer Protocol (HTTP) 1.1 Specification, as defined in RFC 2616 (the "HTTP 1.1 Specification") | Served in electronic form on flash drive |
| EX1011 | "An Architecture for Differentiated Services," also known as the Diffserv architecture, as defined in RFC 2475 ("Diffserv") | Served in electronic form on flash drive |
| EX1012 | Declaration of Eli Fuchs, Dated July 2, 2018 | Served in electronic form on flash drive |
| EX1013 | INTENTIONALLY NOT USED | N/A |
| EX1014 | INTENTIONALLY NOT USED | N/A |
| EX1015 | Dictionary.com definition of "monitor" found at http://www.dictionary.com/browse/monitor (visited June 7, 2018) | Served in electronic form on flash drive |
| EX1016 | Excerpts from File History of European Pat. App. No. 15722292.8 | Served in electronic form on flash drive |

| | | |
|---|---|---|
| EX1017 | "The Internet Standards Process – Revision 3," as defined in RFC 2026, (the "RFC Specification") | Served in electronic form on flash drive |
| EX1018 | U.S. Patent Pub. No. 2004/0114518 ("MacFaden") | Served in electronic form on flash drive |
| EX1019 | U.S. Patent No. 8,156,206 ("Kiley") | Served in electronic form on flash drive |

*Persons served*

Centripetal Networks, Inc.
2251 Corporate Park Drive, Suite 150
Herndon, Virginia 20171
(Via UPS Overnight)

Bradley C. Wright
BANNER & WITCOFF, LTD.
1100 13th Street, N.W.
Suite 1200
Washington, DC 20005-4051
bwright@bannerwitcoff.com
(Via email and UPS Overnight)

Hannah Yunkyung Lee
James Russell Hannah
Lisa Kobialka
Paul Joseph Andre
Kramer Levin Naftalis & Frankel LLP
990 Marsh Rd
Menlo Park, CA 94025

jhannah@kramerlevin.com
pandre@kramerlevin.com
lkobialka@kramerlevin.com
hlee@kramerlevin.com
(Via email and UPS overnight)

/Daniel W. McDonald/
Daniel W. McDonald
Counsel for Petitioner
Registration No. 32,044