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

**DECLARATION OF KEVIN JEFFAY, PH.D.
IN SUPPORT OF SECOND PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 9,565,213**

TABLE OF CONTENTS

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

I, Kevin Jeffay, Ph.D., hereby declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I have been retained by Cisco Systems, Inc. ("Cisco") to provide my technical review, analysis, insights, and opinions concerning the validity of the claims of U.S. Patent No. 9,565,213 (EX1001; "the '213 Patent") entitled "Methods and Systems for Protecting a Secured Network." The patent is assigned to Centripetal Networks, Inc. ("Centripetal").

2. I am a tenured professor in the Department of Computer Science at the University of North Carolina at Chapel Hill where I currently hold the position of Gillian T. Cell Distinguished Professor of Computer Science. I also currently serve as the Chairman of the Department. I have been a faculty member at UNC since 1989.

3. I received a Ph.D. in computer science from the University of Washington in 1989. Before that I received a M.Sc. degree in computer science from the University of Toronto in 1984, and a B.S. degree with Highest Distinction in mathematics from the University of Illinois at Urbana-Champaign in 1982.

4. I have been involved in the research and development of networked computing systems for over 35 years. I have been a faculty member at the University of North Carolina since 1989 where I perform research and I teach in the areas of computer networks, distributed systems, real-time systems, operating systems,

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

multimedia networking, traffic patterns in computer networks, and network performance evaluation, among others. A major theme of my research has been the development of technology to improve the performance of data transfers on the Internet. My research has examined problems ranging from network support for real-time multimedia applications such as audio and video streaming, voice-over-Internet protocol (VoIP) and Internet videoconferencing, to the design of congestion control mechanisms in network routers, to measurements and analysis of network traffic to passively assess the performance of servers on the Internet. In addition, I have also explored problems in the design and implementation of operating systems. Much of my research has been performed jointly with industry. For example, starting in 1991, in collaboration with IBM and Intel my research group constructed and operated one of the first Internet videoconferencing systems. We also developed a data conferencing, "shared window system" that was functionally and visually equivalent to today's Cisco's WebEx and Citrix's GoToMeeting screen sharing products and services.

5. In much of my research I regularly build and use clusters of computers interconnected by network switches, bridges, and routers to form and evaluate experimental and production networks. For example, in the late 1990s and early 2000s, my research evolved to consider router-based mechanisms for controlling the performance of network traffic. In much of this research, my students and I built

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

and instrumented network routers and performed large scale experiments with this equipment. Based on these experiments, in 2003, my group at UNC won the most prestigious research award for original research in computer networking.

6. This project, and others, took place in a networking lab my students and have I constructed at UNC over a number of years. The lab consists of several hundred computers and networking devices. Managing this lab required establishing and configuring firewalls and other security appliances to isolate the lab from the campus network (and vice versa). In late 2003 my research group began a collaboration with Cloudshield Technologies to use a deep packet inspection ("DPI") and processing device made by Cloudshield to investigate ways of detecting anomalous and/or malicious traffic in a high-speed network. Relevant to this matter is the fact that the Cloudshield device enabled us to inspect packets and identify, modify, monitor, and log the contents of those packets in real-time.

7. I have authored or co-authored over 100 articles in peer-reviewed journals, conference proceedings, texts, and monographs in the aforementioned areas of computer science and others. I have previously served as Editor-in-Chief for the journal *Multimedia Systems* and Associate Editor for the journal *Real-Time Systems*. In addition, I have edited and co-edited numerous published proceedings of technical conferences and have edited a book of readings in multimedia computing and networking (with Hong-Jiang Zhang) published by Morgan

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

Kaufman. I am a co-author (with Long Le and F. Donelson Smith) of a monograph related to computer network protocols, and a co-author (with Jay Aikat and F. Donelson Smith) of a second monograph related to experimental computer networking.

8. I have served on numerous proposal review panels for the National Science Foundation and other international funding agencies in the aforementioned areas of computer science. I have served as a program chair or member of the technical program committee for over 100 professional, international, and technical conferences, workshops, and symposia.

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

10. I have served as an expert witness and technical consultant in litigation and inter-partes review matters concerning computer networks, distributed systems, operating systems, multimedia networking, cellular and wireline telephony, voice over IP (VoIP) telephony, datacenter networking, embedded systems and embedded software, and real-time systems, among others. I have testified in several trials, arbitrations, and claim construction hearings as an expert witness.

11. I attach as Exhibit EX1003 my curriculum vitae, which includes a more detailed list of my qualifications. My C.V. also contains a list of all other cases in which, during the previous 4 years, I testified as an expert at trial or by deposition.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

My work on this case is being billed at a rate of $800.00 per hour, with reimbursement for actual expenses. My compensation is not contingent upon the outcome of this *inter partes* review.

II. UNDERSTANDING OF THE GOVERNING LAW

A. Invalidity by Anticipation

12. I understand an anticipation analysis involves comparing a claim to the prior art to determine whether a hypothetical person of ordinary skill in the art ("POSA") would understand the claimed invention is anticipated in view of the prior art, and in light of the general knowledge in the art. I understand that to anticipate a claim, a prior art reference must disclose each and every claim limitation, either expressly or inherently. I also understand that to explain the meaning of a prior art reference, a POSA can refer to a secondary reference. For instance, I understand that information incorporated by reference by the prior art may be considered when understanding the meaning of that prior art.

B. Invalidity by Obviousness

13. I understand that obviousness is analyzed from the perspective of a POSA at the time of the alleged invention. I also understand that a POSA is presumed to have been aware of all pertinent prior art at the time of the alleged invention.

14. I understand that an obviousness analysis involves comparing a claim to the prior art to determine whether the claimed invention as a whole would have

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

been obvious to a POSA in view of the prior art, and in light of the general knowledge in the art at the time the invention was made. I also understand that the invention may be deemed obvious when a POSA would have reached the claimed invention through routine experimentation.

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

16. I understand that when there is some recognized reason to solve a problem, and there are a finite number of identified, predictable solutions, a POSA has good reason to pursue the known options within his or her technical grasp. If such an approach leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense. In such a circumstance, when a patent simply arranges old elements with each performing the same function it had

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

been known to perform and yields no more than one would expect from such an arrangement, the combination is obvious.

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

18. Finally, I understand that Patent Examiners at the USPTO rely upon certain exemplary rationales in reviewing patent applications to understand whether the subject matter of the claims in those patent applications is obvious. I understand that the following is the list of exemplary rationales relied upon by Patent Examiners at the USPTO:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

(A) Combining prior art elements according to known methods to yield predictable results;

(B) Simple substitution of one known element for another to obtain predictable results;

(C) Use of a known technique to improve similar devices (methods, or products) in the same way;

(D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;

(E) "Obvious to try" – Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;

(F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; and

(G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

C. Interpreting Claims Before the Patent Office

19. I understand that *Inter Partes* Review is a proceeding before the United States Patent & Trademark Office ("USPTO") for evaluating the validity of issued patent claims. I understand that, in an *Inter Partes* Review, a claim term is given the broadest reasonable interpretation that is consistent with the patent's specification and prosecution history. I understand that a patent's "specification" includes all the figures, discussion, and claims within the patent. I understand that the USPTO will look to the specification and prosecution history to see if there is a definition for a given claim term, and if not, will apply the broadest reasonable interpretation from the perspective of a POSA at the time in which the alleged invention was made. I present a more detailed explanation of certain claim terms in the '213 Patent in Section VII ("CLAIM CONSTRUCTIONS") below.

D. Materials Relied on in Forming My Opinions

20. In forming my opinions, I have relied on the '213 Patent's claims, specification, and the prosecution history, on the prior art exhibits to the IPR Petition for the '213 Patent, any other materials cited in this declaration, and my own knowledge, experience, and expertise, and the knowledge of a POSA in the relevant timeframe.

21. I have also reviewed and relied upon the following list of materials in preparation of this declaration, any other cited reference in this declaration:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

- U.S. Patent No. 9,565,213 (the ’213 Patent) (EX1001);
- Prosecution History of the ’213 Patent (EX1002);
- CV (EX1003);
- Note: This document is EX1004;
- 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) (EX1005);
- Patent Owner’s Rebuttal *Markman* Brief in *Keysight* lawsuit, Doc. 117 (filed May 3, 2018) (EX1006);
- Parties’ Joint Claim Construction Chart in *Keysight* lawsuit, Doc. 124, Ex. 1 (Filed May 11, 2018) (EX1007);
- Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5 (“ACNS Guide”) (EX1008);
- U.S. Patent No. 9,172,627 (“Kjendal”) (EX1009);
- Hypertext Transfer Protocol (HTTP) 1.1 Specification, as defined in RFC 2616 (the “HTTP 1.1 Specification”) (EX1010);
- “An Architecture for Differentiated Services,” also known as the Diffserv architecture, as defined in RFC 2475 (the “Diffserv Specification”) (EX1011);

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

- Note: EX1012-14 are intentionally left blank;
- Dictionary.com definition of "monitor" found at http://www.dictionary.com/browse/monitor (visited June 7, 2018) (EX1015);
- Excerpts from File History of European Pat. App. No. 15722292.8 (EX1016);
- "The Internet Standards Process – Revision 3," as defined in RFC 2026, (the "RFC Specification") (EX1017);
- U.S. Patent Pub. No. 2004/0114518 ("MacFaden") (EX1018);
- U.S. Patent No. 8,156,206 ("Kiley") (EX1019); and
- Declaration of Eli Fuchs, Dated July 2, 2018 (EX1020)

III. THE PERSON OF ORDINARY SKILL IN THE ART OF THE '213 PATENT

22. The claims of the '213 Patent are directed to methods and systems for securing a computer network via one or more "packet security gateways" that receive "dynamic security policies" from a "security policy management server." EX1001, Col. 25:14-28:44. The gateways then use policies received to identify packets of interest, perform one or more packet transformation functions on the packets, and route the packets to a monitoring device. EX1001, Abstract, Col. 11:4-9.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

23. Based on this, in my opinion, a POSA 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. A POSA would have had a bachelor's degree in computer science, computer engineering, or an equivalent, and four years of professional experience. Lack of work experience can be remedied by additional education, and vice versa.

24. I believe that I would qualify as a POSA as of April 16, 2014, and that I have a sufficient level of knowledge, experience, and expertise to provide an expert opinion in the field of the '213 Patent.

IV. BACKGROUND ON COMPUTER NETWORKING CONCEPTS

25. To better understand my opinions on the validity of the claims of the '213 patent, in this section I provide some background on computer networking with a particular focus on the concepts of network protocols, protocol layers, and data forwarding and routing.

26. All of the concepts discussed in this section were well known and widely used well prior to the April 2014 priority date for the '213 patent. For example, I frequently taught these concepts to undergraduate computer science majors for over twenty years prior to 2014.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

A. Network Protocols and Protocol Layering

27. When one computer wishes to communicate with another computer over a computer network, hardware and software on that computer will create a message and transmit the message as a series of one or more data units to the destination computer. These data units will be formatted and processed by devices in the network (including the source and destination computers) according to a number of rules that exist to facilitate communications over the network. Certain of these rules are grouped together to define a *protocol* for an aspect of the communication. To communicate messages between a source and a destination computer, multiple protocols, operating in concert, are required.

28. These protocols are organized hierarchically as a series of hardware and software "layers" and are colloquially referred to as a "protocol stack." Layers in the stack are numbered and commonly referred to by their layer number. The layers are numbered from the lowest, most basic or primitive protocol layer, to the highest, most functional protocol layer. Each layer is responsible for providing a discrete communication service that builds upon the service(s) provided by the lower layer(s) to provide a more functional, full-featured communication service to upper layers. Figure 1 provides a graphical illustration of the protocol layering concept using the protocol stack described below.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

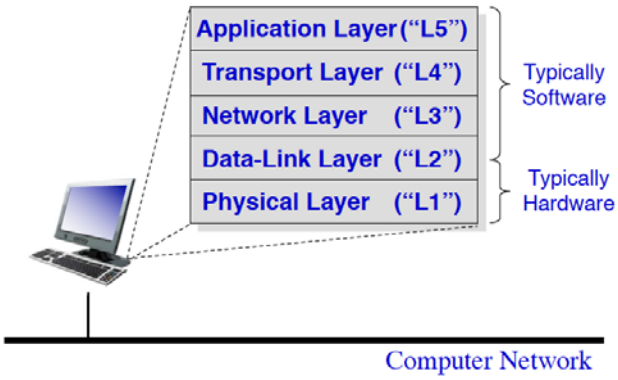


Figure 1: A graphical illustration of the Internet model protocol stack.

29. Historically, the two most dominant models of protocol layers are the OSI (Open Systems Interconnect) model and the "Internet" model. The OSI model defines a seven-layer protocol stack whereas the Internet model defines a simpler five-layer protocol stack. For simplicity, the following concentrates on the five-layer Internet model. (As of the earliest possible priority date of the '213 patent, April 2014, the five-layer Internet model was by far the dominant protocol stack in the networking and distributed systems communities.)

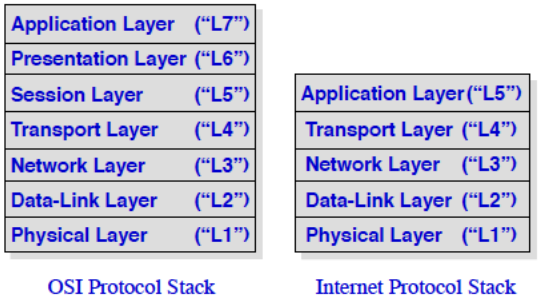


Figure 2: A comparison of the OSI protocol stack and the Internet protocol stack.

30. The lowest layer, layer-1, or "L1," is the *physical layer*. The physical layer protocol is typically implemented exclusively in hardware and is concerned

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

with the low-level details of transmitting binary data ("1s" and "0s") over some physical medium. The physical layer refers to the physical media used to construct the network. Examples of different physical layers would be fiber optic cable, copper twisted pair wiring, or radio frequency spectrum. As these are each different physical media, each would require a different physical layer protocol because the process of transmitting binary data on the medium is different for each type of media.

31. The next layer in the protocol stack, layer-2, or "L2," is the *data link layer* (or simply the *link layer*). The link layer defines a transmission structure, typically called a *frame*, and is responsible for transmitting frames between computers on the same network. The most common example of a link layer protocol is the Ethernet protocol. The link layer uses the services of a layer-1 protocol to transmit the binary data of a frame on the physical medium of the network to a destination on the network.

32. A frame consists of a *header* and a *payload*. The payload is the actual data being communicated across the network and may have additional, internal structure as described below. The header contains, among other things, a data link layer address of the computer generating the frame (a "source address") and a link layer address of the destination computer of the frame (a "destination address"). The payload is analogous to a letter ("data") that is to be mailed to someone in an envelope (a "frame"), and the header is analogous to the address written on the

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

outside of the envelope. The link layer addresses in the header of a frame are often referred as "hardware addresses," or "MAC" (Media Access Control) addresses.

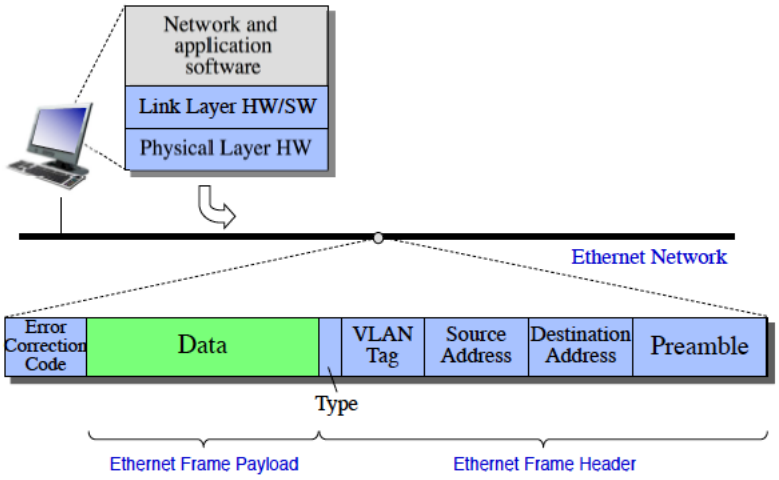


Figure 3: Illustration of the structure and major components of an Ethernet frame. (Not to scale)

33. Frames are variable length structures with the size of a given frame primarily determined by the size of the payload being transmitted. Frames have a maximum size and payloads that are too large to be carried in a maximal size frame must be segmented by the sender into multiple smaller payloads and transmitted in a series of frames. This segmentation process is most commonly performed by higher layer network protocols as described below.

34. Layer-2 protocols can only transmit frames between two computers on the same network. More precisely, layer-2 protocols can only transmit frames between two computers on the same local-area network (LAN). In fact, the technical definition of a LAN is the set of computers that are "reachable" from one another

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

(can communicate with one another) via a layer-2 protocol (and only a layer-2 protocol).

35. LANs are created by a type of interconnection device called a *switch*, *bridge*, or *access point*. Individual computers connect to the LAN switch/bridge/access point via some physical medium such as, in the case of Ethernet, the ubiquitous blue Ethernet cable. The typical interconnection device is capable of interconnecting a large number of computers. When a computer generates and transmits a layer-2 frame, the frame is received by the interconnection device. The device examines the destination layer-2 address in the frame to determine destination of the frame. Based on the layer-2 destination address carried in the frame, the frame is *forwarded* to the appropriate destination. "Forwarding" is a specific term in the networking arts and refers to the process that is used by a layer-2 network interconnection device to transmit a frame to (or towards) its ultimate destination.

36. LANs are limited in terms of both the number of computers that can be on a LAN as well as the geographic area served by the LAN. The physical media used to build the LAN and layer-1 protocol used by the LAN constrain the size of the LAN. The type of interconnection device used to realize the LAN may also constrain the size of the LAN. However, larger networks can be formed by interconnecting LANs together to form an *internetwork* (or more simply, a

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

"network"). As a technical matter, the Internet is simply a (massive!) internetwork of interconnected LANs.

37. LANs can be interconnected via a new type of interconnection device called a *router*. Normally, two computers on two local-area networks interconnected via a router cannot exchange messages with each other using only a layer-2 protocol. The two end-systems cannot exchange messages with each other because the addresses in the layer-2 frame header are only known on the LAN on which the frame was generated. That is, a computer on one LAN cannot transmit data to a computer on another LAN using just a layer-2 protocol because the sending computer cannot address a frame to the destination computer. To transmit data between two computers on different LANs in an internetwork, a higher-layer protocol is required.

38. The next layer in the protocol stack, layer-3, or "L3," is the *network layer*. Network layer protocols solve the problem of communication between LANs by defining a new transmission structure, called a *datagram*, and by defining a new addressing scheme. Like a frame, a datagram also consists of a header and a payload. The datagram header will contain a new type of address, a network layer address that can be used to transmit datagrams between computers in an internetwork.

39. On users' computers, the network layer (and all higher layers) are most commonly implemented in software. When a computer needs to send a message to

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

another computer, the sender's network layer software will construct one or more datagrams addressed to the destination computer's network layer address. The network layer at the sender will then use the services of a link layer protocol to embed the network layer datagram into the payload field of a link layer frame and transmit the frame to a router that connects the sender's LAN to a set of other LANs. The router will receive and process the frame using a link layer protocol to extract the embedded network layer datagram and process the datagram using a layer-3 protocol to determine on which LAN the datagram should be transmitted so as to arrive at its destination (or to get closer to its destination). The router will then use a layer-2 protocol (that may be the same or different layer-2 protocol than that used to receive the original frame) to embed the extracted datagram into the payload field of a new layer-2 frame and transmit the layer-2 frame containing the original sender's datagram on the determined egress network to a new layer-2 destination. The new layer-2 destination may be another router (where the process just described repeats) or it may be the destination computer. Figure 4 illustrates this process.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

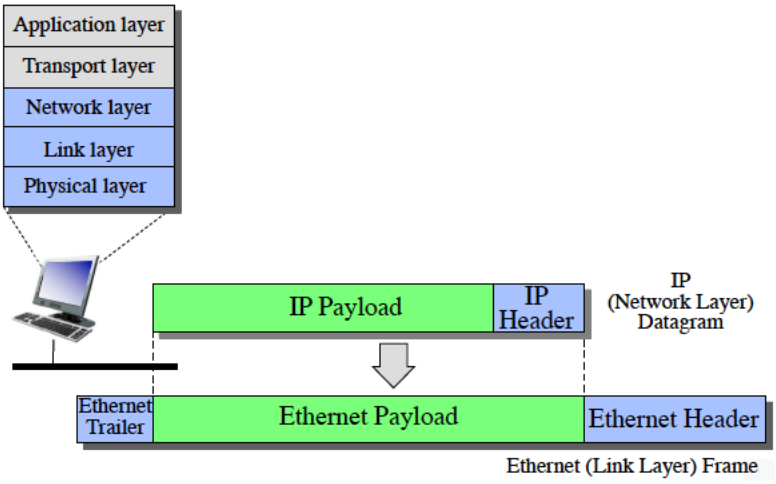


Figure 4: Illustration of datagram embedding.

40. The process of using the layer-3 destination address at an interconnection device to determine the network onto which a layer-3 datagram should be transmitted and transmitting the datagram on that network, is called *routing*. "Routing" is a specific term in the networking arts and refers to the process that is used by a layer-3 network interconnection device (a router) to transmit a datagram to (or towards) its destination.

41. The most common network-layer protocol in use today is the IP protocol (Internet Protocol) of the Internet. The network layer is responsible for providing *network layer addresses* (most commonly IP addresses) and a means of *routing* datagrams between local-area networks. Layer-3 protocols solve the *internetworking* problem of interconnecting local-area networks together. They provide addresses that span "the network" (all interconnected local-area networks) and enable computers to address other computers anywhere in the network of

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

interconnected local-area networks. Figure 5 provides an illustration of the IP datagram structure and components.

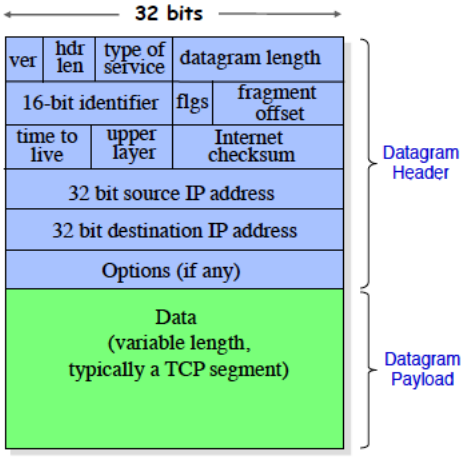


Figure 5: Structure and major components of an IP datagram. The figure shows a datagram as a series of 32-bit words.

42. Network layer protocols can deliver datagrams between two computers on different LANs. However, as datagrams transit the networks (the LANs) between the sender and the receiver, they are subject to a number of pathologies that can hinder communication or even make communication impossible. These pathologies include the "loss" (deletion) of datagrams, the reordering of datagrams (so that, for example, in a sequence of datagrams, datagram 2 arrives before datagram 1), and the creation of duplicate copies of datagrams (so that a receiver may receive more than one copy of a datagram).

43. Protocols in the next layer in the protocol stack, layer-4 or *transport layer* protocols, build upon network layer protocols to provide a communication service that ameliorates some or all of these pathologies. For example, in networks

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

such as the Internet that use the IP network layer protocol, the most common transport layer protocol is the Transmission Control Protocol ("TCP"). The TCP transport layer protocol provides, among other things, a reliable, in-order data delivery service by monitoring transmissions to detect when datagrams have been lost and retransmit copies of these lost datagrams until they have been successfully received.

44. As with other layers, the transport layer defines a new transmission structure called a *segment* that is comprised of a new header and payload. In the case of TCP, the header will include control information such as a sequence number for the segment. The sequence number is used by the destination computer to detect and reorder segments arriving out-of-order, and to detect lost segments. The TCP header will also contain an indication of which application on the destination computer should receive the segment. This indicator, called a *port number*, is a form of address and identifies which of the possibly several applications on the destination computer that may be using the network at the current time, the arriving datagram is destined for.

45. The header of a TCP segment contains two port numbers: a source and destination port number. The source port number identifies the application on the sending computer that generated the datagram (the computer corresponding to the source IP address in the header of the datagram carrying the TCP segment) and the

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

destination port number identifies the application on the destination computer (the computer corresponding to the destination IP address in the datagram header) that should receive the datagram. As an analogy, the IP addresses can be thought of as street addresses — addresses that identify a structure such as a home or office building — and port numbers can be thought of as identifiers (such as names) that identify individuals residing in the destination structure. (For completeness, layer-2 addresses, *i.e.*, the MAC address, can also be thought of as the street address of either the recipient (if the recipient is on the same LAN as the sender), or the street address of the first Post Office between the sender the recipient (that will further process the "letter").) Figure 6 illustrates the structure and components of a TCP segment.

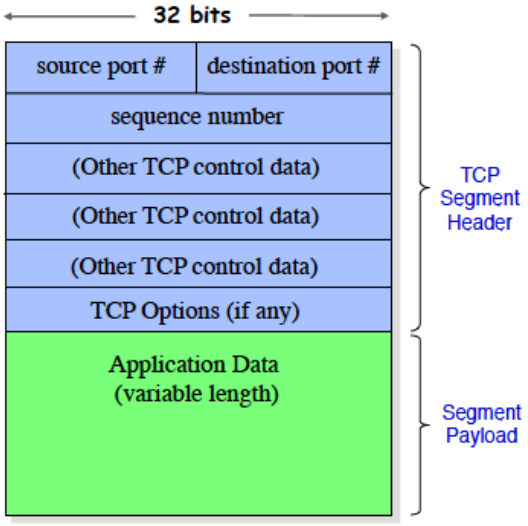


Figure 6: Structure and major components of an TCP segment.
The figure shows a segment as a series of 32-bit words.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

46. Combined, the IP source address, IP destination address, source port number, destination port number, along with an identifier of the transport protocol in use (*e.g.*, TCP or some other transport protocol), uniquely identify all of the datagrams being communicated between an application running on one computer (*e.g.*, a client) and an application running on a second computer (*e.g.*, a server). The 5-tuple consisting of the packet's source and destination IP address, source and destination port number, and (transport) protocol type is called a *flow id* as this tuple can be used to identify datagrams within a single end-to-end (original sender to ultimate destination) communication session. All five components of the flow id are required to determine the communication session to which a datagram belongs as the elimination of any component would introduce ambiguity as to the identity of the communication session. The 5-tuple flow id has been used to associate datagrams with communication sessions since the creation of the Internet.

47. The next layer in the protocol stack, and the highest layer in the Internet model, is the *application layer*. Application layer protocols also define a new transmission structure, simply called a *message*. The message is typically comprised of a new header and payload. The contents of the header vary by application layer protocol but will typically include addresses and control information that are relevant to the application (and hence application protocol specific).

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

48. The most common application layer protocol is the Hypertext Transfer Protocol ("HTTP") that is used to request and receive data from web servers. In the case of HTTP, the header contains an indication of the content that is being requested (the "address" of the content on the server) as well as other control information. The body of an HTTP message contains data such as the data for a requested web page. Figure 7 illustrates the structure and components of two basic HTTP messages: the HTTP request message (sent from a client such as a web browser to a server to request content from that server) and the HTTP response message (sent from a web server back to a client in response to previous request).

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

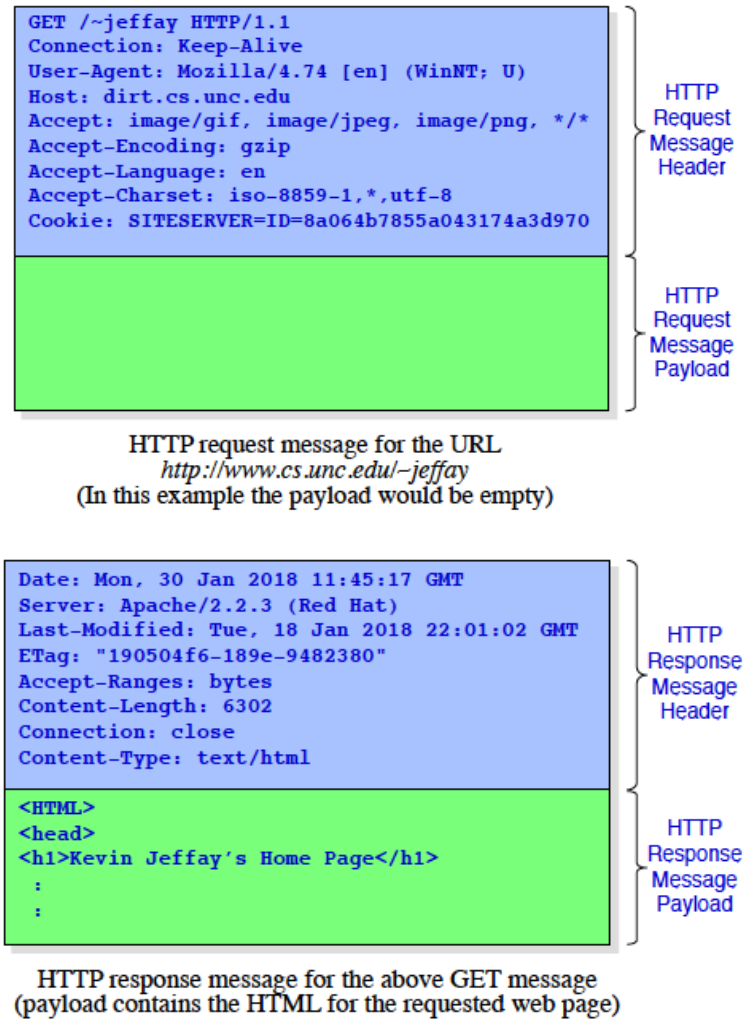


Figure 7: Structure and major components of HTTP request and response messages.

49. Physical, link layer, network, transport, and application protocols work in an orchestrated manner to enable communications between any two computers in an internetwork (*e.g.*, the Internet). When an application on one computer wishes to communicate with an application on another computer (*e.g.*, a web browser on one computer sends a request for content to a web server on a second computer), the application layer constructs an application layer message (according to the rules of

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

its application layer protocol) and "passes" the message to transport layer software. The transport layer may perform processing on the application layer message (*e.g.*, segment the message into smaller sub-messages) and then embed the application layer message (or sub-message) in the payload of a transport layer message structure. The transport layer will add the appropriate header information (*e.g.*, port numbers) and "pass" the resulting transport layer segment to network layer software.

50. The network layer may perform processing on the transport layer segment and then embed the transport layer segment in the payload of a network layer datagram. The network layer will add the appropriate header information (*e.g.*, IP addresses) and "pass" the datagram to data link layer software. The data-link layer may perform processing on the network layer datagram and then embed the network layer datagram in the payload of a data link layer frame. The data link layer will add the appropriate header information (*e.g.*, MAC addresses) and "pass" the frame to physical layer hardware for transmission on the physical network medium. The embedding of protocol data units in one another, also called "encapsulation," is illustrated in Figure 8.

51. The frame will be forwarded across the originating LAN to the first router. A new frame carrying the original datagram in its payload will be created and routed through the internetwork by data-link and network layer hardware and software on routers until a corresponding frame arrives at the destination computer

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

where the steps analogous to those occurring at the sender's computer outlined above, are performed in the reverse order (*i.e.*, data link layer processing to extract an embedded datagram followed by network layer processing to extract an embedded segment followed by transport layer processing to extract an embedded message (or message fragment) followed by application layer processing).

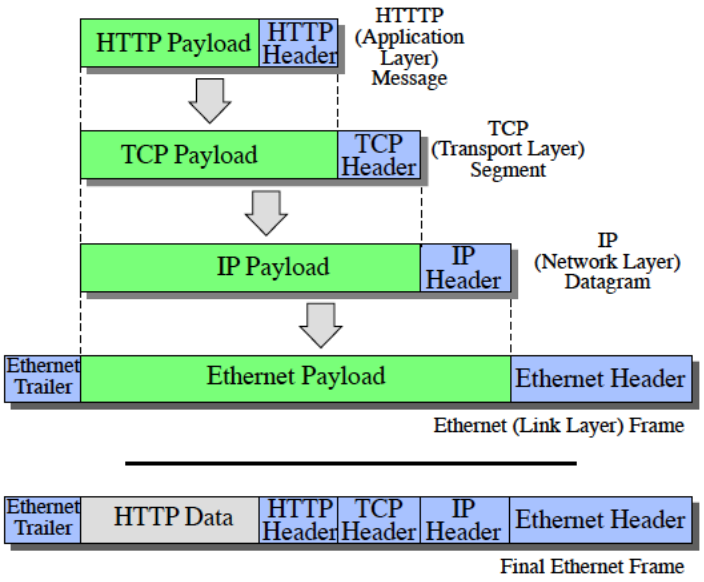


Figure 8: Embedding of protocol data units during the transmission process.

B. Network Packets

52. Layers two and higher within the protocol stack each define a unit of data transmission and a structure for the transmission unit. The foregoing has purposefully used different terms for the transmission units in the protocol stack in the Internet model: layer-2 transmission units are frames, layer-3 units are datagrams, layer-4 units are segments, and layer-5 units are messages. These terms

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

are all terms of art used by networking professionals when trying to be precise in their descriptions of network operations.

53. Nonetheless, networking professionals are often less than precise and frequently use the generic term "packet" to refer to one of more transmission units within the protocol stack. Thus, while it can be confusing, it is not uncommon for persons of ordinary skill in the networking arts to refer to link layer transmission units as "Ethernet packets" instead of the more precise "Ethernet frame," network layer transmission units as "IP packets" instead of the more precise "IP datagram," transport layer transmission units as "TCP packets" instead of the more precise "TCP segment," and application layer transmission units as "HTTP packets" (in the case of the web) instead of the more precise "HTTP message." Given this informal use of the word "packet," one must always use the context in which the word appears to understand exactly which type of transmission unit is being referred to.

C. Firewalls and Network Security

54. It is well known that computers can generate and send messages to other computers that contain "viruses" and other malware (executable code that performs some malicious action upon receipt). For example, computers infected with malware can be used by a bad actor to generate a "denial of service" attack (a "DoS" attack). In a DoS attack, a computer or collection of computers conspire to send a sufficiently large volume of message traffic to another computer (e.g., a

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

server) or network device such that the "victim" computer/device cannot process the traffic and either crashes or otherwise is unable to respond to or process legitimate message traffic. In either case, the "service" provided by the victim ceases to be available to its clients. (In the case where a collection of computers is used to launch a DoS attack, the attack is termed a "distributed denial of service attack," of a "DDoS" attack.)

55. To protect computers from such attacks, devices are placed in networks, typically at network interconnection points, that examine packets transiting the network to identify packets containing potentially malicious content as well as packets that may be part of a denial of service attack. Packets so identified are typically deleted (dropped) so that they do not reach their intended destination. The best-known example of such a security device is a *firewall*. A firewall is either a computer or a program executing on a network device such as a router, that examines packets to identify packets matching one or more patterns ("rules") and to act on these matching packets. Network operators configure firewalls to identify packets of interest. Simple firewalls can be programmed to identify packets based on certain values carried in one or more headers within the packet. For example, if a certain external network is known to be the source of denial of service attacks, packets carrying a source IP address from that network can be blocked from entering the network protected by the firewall. The process of matching packet contents against

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

patterns or rules and acting on packets that match is commonly referred to as "packet filtering" (or simply "filtering").

56. Firewalls can also block traffic leaving a network as part of the business practices of the organization operating the firewall. For example, an organization may forbid its users from accessing certain websites from within the organization's network. In this case the firewall may be configured to examine out-going traffic to look for certain IP addresses (or other information found in one or more packet headers) and block those packets from leaving the organization's network. In any event, organizations frequently record copies of certain packets entering and leaving their networks. Commonly, this is done by logging some or all of the packets examined by the firewall. This logging can be done at the firewall device, by copying and sending ("mirroring") interesting packets to another device for analysis or logging, or by generating and sending messages to a logging server, such as byusing a utility such as syslog, a logging system developed for UNIX in the 1980's.

57. Firewalls can typically be configured (programmed) in many different ways. A common means of configuring a firewall is to simply maintain a list of network addresses that are either allowed or not allowed to be the source or destination of packet traffic. In the case of a list of disallowed source or destination addresses, the list is referred to as a "black list." In the case of a list of allowed source or destination addresses, the list is referred to as a "white list." In the case of

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

black lists, packets carrying header fields with values that match corresponding values on the black list are typically dropped and all other packets are typically allowed to transit the firewall. Conversely, in the case of white lists, packets carrying header fields with values that match corresponding values on the white list are typically allowed to transit the firewall and be forwarded toward their destination, and all other packets are typically dropped. Thus, the use of white lists and black lists are complementary approaches to securing a network. Both require that packets be examined to determine if header fields carry values that appear on a list and if so, both perform one action on the matching packets and perform a different action on non-matching packets. Thus, the mechanisms required to support white or black lists are the same: both require a value or pattern matching facility and both require a means of at least deleting (dropping) packets and a means of allowing packets to transit the firewall.

D. An Example Application-Layer Protocol: SIP and Uniform Resource Identifiers

58. An example application layer protocol is the protocol used to realize so-called voice-over-IP telephony ("VoIP"). VoIP is simply an application that allows computers to make telephone calls over the Internet. A common protocol used to implement VoIP telephony is SIP ("Session Initiation Protocol"). SIP is a protocol used to establish (setup) and terminate calls between parties (users) on the Internet.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

59. Calls in SIP are established by specifying a *uniform resource identifier* ("URI") that identifies the callee. A URI is a text string that identifies an entity (*e.g.*, a file, a person, a mailbox) on the Internet and a means of locating or finding that entity. In the case of SIP, examples URIs include:

sip:1-999-123-4567@voip-provider.example.net, and

sip:jeffay@voip-provider.example.net

These URIs consist of a "scheme" ("sip" in these examples) that define a namespace followed by "path" that provides an indication of the means by which the callee may be contacted (located). In these examples, the callee may be located by contacting the computer (server) "voip-provider.example.net" with the identifier (name) "1-999-123-4567" or "jeffay."

60. URIs can either be a "name," called a Uniform Resource Name ("URN") or a "location," called a Uniform Resource Locator ("URL") (or both). The latter case includes the familiar URLs used to specify web "addresses" via the URI syntax

http://www.cs.unc.edu/~jeffay/papers/

61. Thus, because many applications have a need to specify a resource, URIs are used by a variety of applications, and in a variety of application layer protocols.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

E. Queues in Network Interconnection Devices

62. As discussed above, LANs are created by layer-2 interconnection devices such as bridges or switches. (In the latter case, these are commonly referred to as "LAN switches.") LANs are interconnected by routers, which are layer-3 interconnection devices. Generally, the architecture of these devices is similar. Computers connect to an interconnection device via some transmission means such as an Ethernet cable (in the case of a wired Ethernet network). This connection point serves as both an input interface to the interconnection device (to receive transmission units sent by the connected computer) and an output interface (to transmit transmission units from the interconnection device to the connected computer). Transmission units enter the interconnection device on an input interface and are processed by the device to determine the appropriate output interface on which to transmit the unit. The interconnection device uses addresses in one or more protocol headers of the transmission unit to make this determination (LAN switches use layer-2 addresses and routers use layer-3 addresses for this purpose).

63. Modern interconnection devices can be thought of as comprised of a set of input/output interfaces and a switching "fabric" or "matrix" that interconnects the interfaces. Figure 9 below provides a graphical illustration of the structure of a network interconnection device. Once the determination of the outbound interface is made for a transmission unit, the unit is sent across the internal fabric to this

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

interface. When the transmission unit arrives at the outbound interface, the interface may not be able to transmit the unit immediately. This may be the case because, for example, the outbound interface is in the process of transmitting a previous transmission unit and this transmission must complete before transmission of the newly arrived unit may commence. In this case, the newly arrived transmission unit is queued at the outbound interface until such time as the interface is able to transmit the unit.

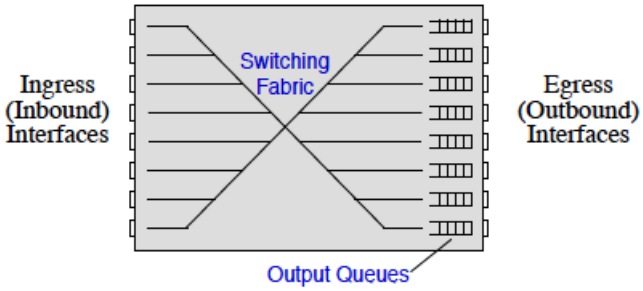


Figure 9: Illustration of the major components of a network interconnection device. (For simplicity, the figure shows only one half (direction) of the data path.)

64. Because network interconnection devices typically provide a large number of input/output interfaces, it is possible that the queue of waiting transmission units at an output interface may be quite large. This can happen, for example, if transmission units arrive on multiple input interfaces at the same, or roughly the same time, and are all destined for the same output interface. In this case, a substantial queue may build at the output interface as "work" (transmission

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

units to be transmitted) is arriving faster than the interface can process the work (transmit units).

65. Queues are simply a memory that stores transmission units as they wait to be transmitted. Importantly, the memory for a queue is of finite size and hence the queue has a finite storage capacity. A queue can only store so many transmission units. Should a "burst" of transmission units arrive at an interconnection device, all destined for the same output interface, it is possible that the storage capacity of the queue at that output interface will be exhausted. In this, case any additional transmission units that arrive at this output interface will be discarded (not queued). This means that the discarded transmission unit is never transmitted out of the interconnection device and is said to be "lost" in the network.

66. Output interfaces in interconnection devices, particularly in routers, may have multiple queues associated with them. With a single queue, transmission units are typically queued in a first-come-first-served manner. Such a queuing scheme is said to be *fair* because all transmission units have an equal chance of being transmitted. In this case the resulting transmission service provided by the interconnection device is called *best-effort.*

67. As the Internet has evolved to support more diverse applications, there has arisen a need for a "better than best effort" transmission service. This need is often expressed as a requirement for "quality-of-service," or "QoS," meaning that a

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

specific level or type of transmission service, such as a bounded delay service or a minimum bit-rate service is required. For example, the usefulness of certain network applications, such as voice-over-IP telephony ("VoIP") depend on the network transmitting messages with minimal end-to-end delay. For these applications, if the end-to-end delay exceeds a certain bound, interactivity is compromised and the application may not function as intended. To support such "real-time" applications, network interconnection devices may provide multiple queues for an output interface and use these queues to implement a priority transmission scheme. If the interconnection device is able to recognize arriving transmission units as those corresponding to a real-time application (or from an "important" or "high-priority" application), it can place those transmission units into a high (or higher) priority queue at the output interface than transmission units that are not so recognized. Other applications that are more delay-tolerant, such as HTTP web traffic, may be placed in a lower priority queue. The transmission logic at the output interface then selects transmission units from the high priority queue when it is able to transmit a unit and only selects transmission units from lower or non-priority queues when the high priority queue is empty.

F. Differentiated Service ("Diffserv")

68. Most modern routers provide multiple queues per each output interface and enable network engineers to provide a variety of better than best effort

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

transmission services. However, in order to do this, a router must have some means of identifying which arriving datagrams are subject to the priority treatment (queuing). To solve this problem, in the late 1990s the Internet research community developed the Differentiated Services Architecture for the Internet ("Diffserv") to standardize a means of including priority or importance information in IP datagrams. (In Diffserv parlance, priority or important is also referred to as "quality-of-service.")

69. As part of the Diffserv architecture, a field was reserved in the header of IP datagrams that can be used by a router to determine how important an arriving datagram is. This field, called the *differentiated services* field, or DS field, carries a value, called a *differentiated services code point* (DSCP), that routers can use to classify arriving datagrams into service classes and process (queue) datagrams differently according to their class (thereby providing different "qualities-of-service").

70. The Internet community standardized the placement of the DS field within the IP datagram header and the range of DSCP values that can appear in a DS field. The community purposefully, however, did not standardize the performance semantics of services associated with DSCP values. While there are recommended types of service classes associated with DSCP values, the exact interpretation of a DSCP value (*i.e.*, the precise manner in which a router will process a datagram with

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

a specific DSCP value) is up to the organization that manages the network. However, forwarding services provided by a device or network of devices implementing the Diffserv architecture are typically described in abstract but suggestive terms, rather than in quantitative terms. For example, the different classes of services possible in a Diffserv implementation are commonly referred to as "gold" services, "silver" services, and "bronze" services (and references to "gold," "silver," and "bronze" service in the literature are typically a reference to a Diffserv implementation).

71. In practice, network operators will typically base decisions on how DSCP values are interpreted and supported based on business arrangements their organization have in place with other network organizations. However, virtually all networks recognize a common default DSCP that indicates the datagram carrying this value is subject to best-effort treatment (*i.e.*, no priority, first-come-first-served queuing).

G. Packet Processing

72. Network interconnection devices use information contained in one or more headers of an arriving packet to determine how the packet should be processed. Indeed, this is precisely the purpose for much of the information carried in protocol headers. "Processing" in this context simply refers to determining what should be done with an arriving packet. By far, the most dominant processing options are to

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

transmit the packet to another device (either via a layer-2 forwarding function, or a layer-3 routing function) or to drop/discard the packet (not transmit the packet to another device). Network managers can control how interconnection devices will process packets by configuring the devices (either manually of via some automatic means) via a set of rules or policies. By April of 2014, most LAN switches and network routers provided a rich vocabulary of options and parameters that could be used to control how the device processed packets.

73. Central to the notion of packet processing in an interconnection device is the imperative of ensuring that the time required to process a packet not exceed the minimum possible interarrival time of any two packets. Processing packets as fast as they arrive is often called "wire speed" processing (or "processing at wire speed").

74. Should the processing time for a packet exceed the minimum interarrival time then the network device is no longer processing packets in real-time (as they arrive) and a queue of packets awaiting processing is certain to be created somewhere internal to the device. That a device cannot process packets in real-time under all circumstances inherently means that the device is not capable of fully utilizing the transmission capacity of outbound links to which it is attached. Potentially not being able to fully utilize the transmission capacity of attached links is an anathema to network operators and equipment vendors. For this reason, great

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

attention is paid to the time required in the worst case to process packets in interconnection devices.

75. Should a particular packet processing function take too long to perform (or should there be concern over the time required to perform a function), standard techniques exist to "speed up" the processing. Importantly, these well-known techniques are long standing and generic in the sense that they are applicable to any computing device. In particular, these techniques were developed and have been routinely employed in computing devices since the 1960s.

76. The most common technique for speeding up a function is to *pipeline* the implementation of the function. Pipelining consists of breaking a single function into multiple sub-functions, ideally of equal duration, and executing the sub-functions in series on separate processing units or separate devices. When processing completes at one processing unit or device, the partially processed unit of work is "passed" to the next processing unit or device for further processing.

77. Pipelining, or serial execution of a function, improves performance because it enables parallelism. For example, in a packet processing pipeline with three stages, in steady state and in the ideal case, the execution of the third stage of processing on packet 1 occurs in parallel with the execution of the second stage of processing on packet 2 which occurs in parallel with the first stage of processing on

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

packet 3. These stages can execute in parallel because they are implemented on separate processors.

78. Pipelining does not reduce the total time required to process any one packet but it increases the throughput of the processing system on average by a factor proportional to the number of stages in the pipeline. For example, consider a packet processing system where packets can arrive, in the worst case, every *t* time units and where packet processing, in the worst case, can take 3*t* time units. In this system, if the worst case interarrival time is ever realized (packets arrive at the highest possible rate) at the same time as the worst-case per-packet processing time is required, an interconnection device will not be able to keep up with arriving packets and a queue of packets awaiting processing will build up (and packets may be dropped if this queue ever exceeds its capacity). If the interconnection device is provisioned with three processing units, we can pipeline packet processing into three stages (assuming we can roughly evenly split packet processing into three processing steps each requiring *t* time units to complete). In the resulting pipeline, each packet still requires time 3*t* to be fully processed (to make it through the three pipeline stages). However, importantly, packets now enter and leave the pipeline at the rate of one packet every *t* time units — precisely the worst-case arrival rate of packets. That is, the delay in processing any one packet remains the same as in the non-pipelined case (3*t* time units) but the rate at which packets are processed has sped up by a factor of

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

three (the number of stages in the pipeline). In this manner, arranging for the processing or examination of a packet to be split up into multiple discrete steps, and pipelining these steps through the serial execution of the steps on multiple processors or multiple devices, can dramatically speed up (increase the throughput of) the computation. Figure 10 provides an illustration of the pipelining process.

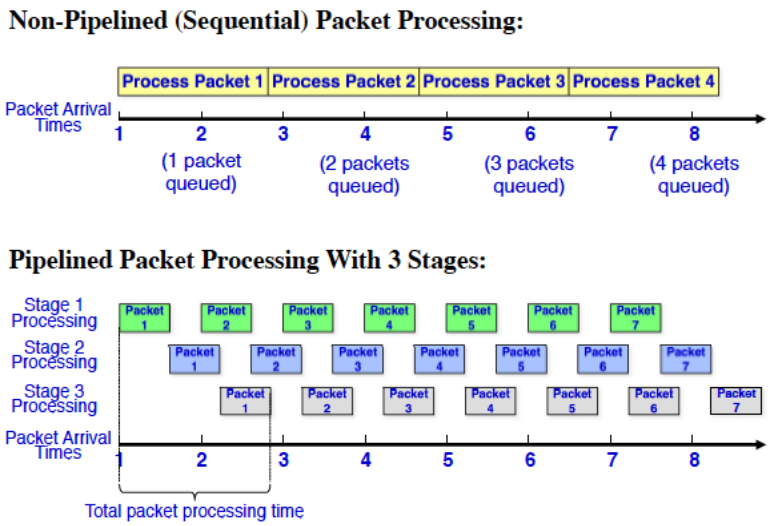


Figure 10: Illustration of the pipelining process for the case where sequential packet processing is nearly twice as long as the packet interarrival time. Note that the time required to process an individual packet in both processing schemes is the same.

V. OVERVIEW OF THE '213 PATENT

A. The Specification of the '213 Patent

79. The '213 Patent describes methods and systems for "protecting a secure network." EX1001, Col. 1:38-39. The '213 Patent sets forth what it considers to be a problem with prior network protection approaches in its "Background" section when it asserts that "proactive" solutions to network protection "have often been

deemed untenable due to an inability to scale to larger networks." EX1001, Col. 1:20-21. The '213 Patent further contends that:

> A significant challenge associated with building a scalable proactive solution is the need to filter substantially all network traffic at a high resolution. In a large network, where traffic volumes may be enormous, the time required to provide high resolution filtering has traditionally been thought to render a proactive solution infeasible.

EX1001, Col. 1:21-27.

80. To protect the different networks, the '213 Patent employs what it refers to as a "packet security gateway" (PSG), which 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." EX1001, Col. 1:39-40 and Col. 11:34-57. A PSG is an interconnection device located at the boundaries between networks, as shown in Figure 1 of the '213 Patent:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

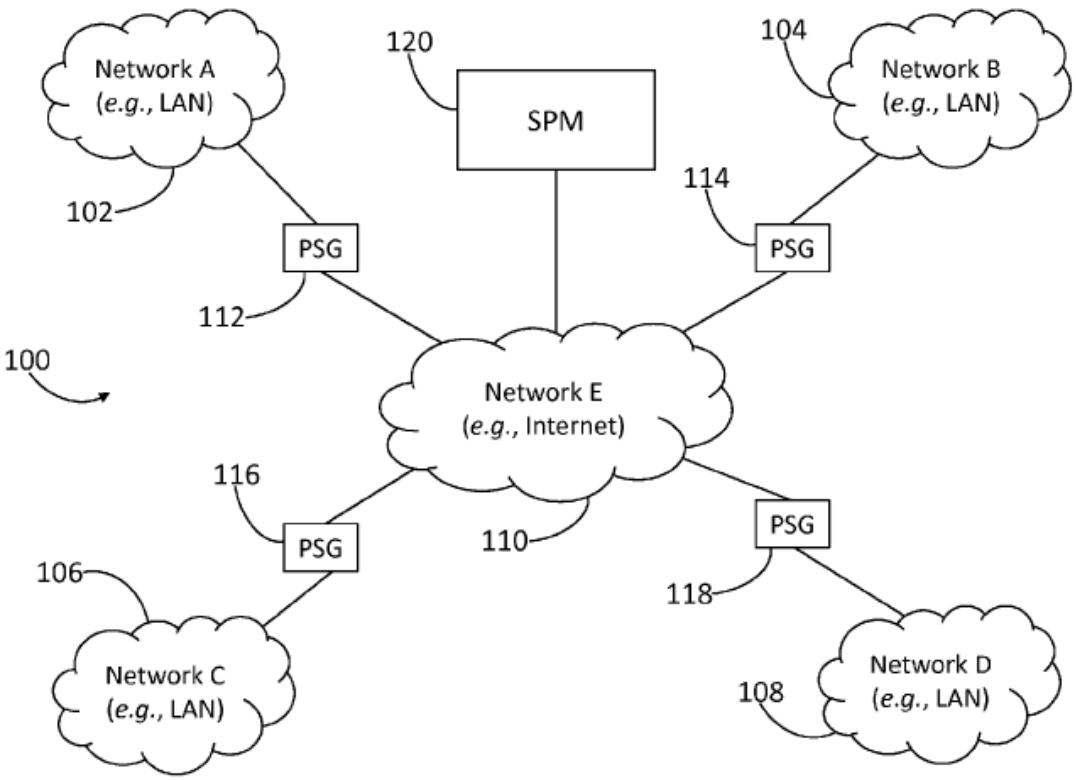


FIG. 1

81. The PSG receives a "dynamic security policy" from a "security policy management server" (SPM). EX1001, Col. 1:41-43. Specifically, the dynamic security policy 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. 11:34-40. The '213 Patent indicates that:

> Such rules may 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. This packet log (or digest) may contain

> selected packet information and/or system information (e.g., associated network addresses, ports, protocol types, URIs, arrival times, packet sizes, directions, interface names, media access control (MAC) addresses, matching rule IDs, metadata associated with matching rules, enforced policy names, or the like)."

EX1001, Col. 11:46-54. The specification admits that such logging can be implemented using a known or "standard" logging format called "syslog."[1] The rules identify "packet communications that are of interest to an organization that operates the secured network," the packet logs or digests of which are stored and/or forwarded based "on a standard (e.g., syslog, or the like)." EX1001, Col. 11:44-57.

82. The '213 Patent shows the process steps pertinent to logging in one summary figure, Figure 12 (annotated), reproduced below.

[1] Syslog is a standard logging utility and log format originally developed for the UNIX operating system. Syslog dates back to the 1980s.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

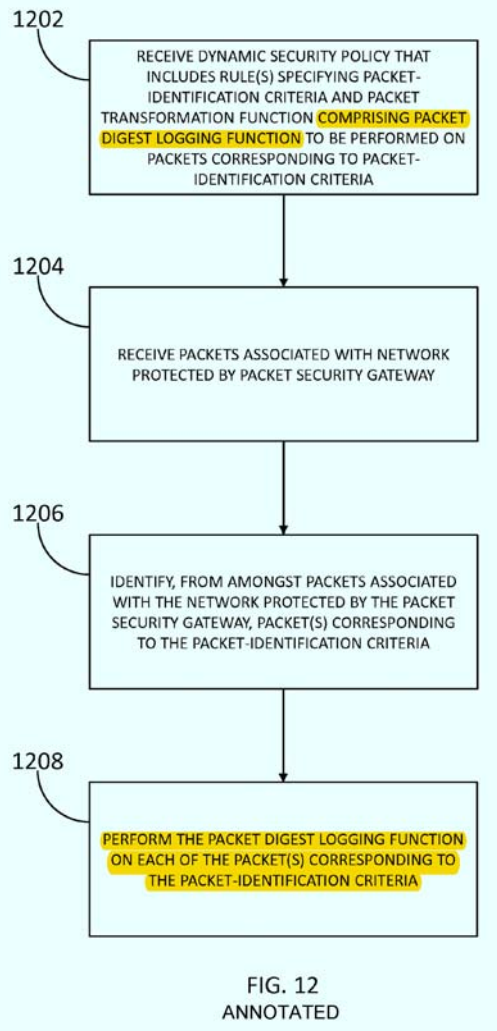


FIG. 12
ANNOTATED

83. The ’213 Patent explains that “packet security gateways 112, 114, 116, and 118 may be associated with security policy management server 120” and “may receive a dynamic security policy” as described above. EX1001, Col. 22:18-21. At the first step (1202) in Figure 12, above, a PSG receives a dynamic security policy specifying packet-identification criteria and comprising a packet digest logging function. At the next step (1204), packets are received at the PSG. The subsequent step (1206) comprises identifying packets that correspond to the packet-identification criteria of the received dynamic security policy.

84. The packet-identification criteria comprises "application-layer packet-header information and/or 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." EX1001, Col. 22:23-26. The '213 Patent provides examples of "application-layer packet-header information," such as "information that identifies one or more HTTP packets." EX1001, Col. 21:29-31. Accordingly, the application-layer packet-header information is used by a PSG to determine:

> . . . that the packets are among the HTTP packets identified by the rule(s) (e.g., the HTTP GET method call and/or the HTTP PUT method call may specify one or more URIs, and packet security gateway 112 may determine that the packet(s) originated from and/or are destined for a network address corresponding to the URI(s)).

EX1001, Col. 21:59-64.

85. At the final step (1208), the packet digest logging function is performed "on each of the packets corresponding to the packet-identification criteria." EX1001, Col. 22:53-57. The resulting record is reformatted according to a "logging system standard", such as "syslog." EX1001, Col. 23:18. The '213 provides the following description which acknowledges the existence of a "logging system standard" such as syslog:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

> In some embodiments, performing the packet digest logging function may include identifying a subset of information specified by the packet digest logging function . . . for each of the identified packets, and generating a record comprising the subset of information for each of the identified packets. In some embodiments, packet security gateway 112 may be configured to temporarily store data comprising the subset of information for each of the identified packets, and to utilize the stored data to generate a message comprising the subset of information (or a portion thereof) for each of the identified packets. In such embodiments, packet security gateway 112 may communicate the message to a different computing device (e.g., security policy management server 120). In some embodiments, the subset of information (or a portion thereof) may be reformatted (e.g., by packet security gateway 112 and/or security policy management server 120) in accordance with a logging system standard (e.g., syslog).

EX1001, Col. 22:60-23:18. Further description of any "logging system standard" is not provided.

B. The Claims of the '213 Patent

86. The '213 Patent includes 16 claims. Claims 1 and 10 are independent claims and are directed to the monitoring and logging embodiments of the '213 Patent. Independent claims 1 and 10 include limitations that are quite similar to one another. Claim 1 recites as follows:

1. A method comprising:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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, 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

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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.

EX1001, Col. 25:14-55.

87. Independent claim 10 is similarly directed to a method and many of the limitations of claim 10 are substantially similar to the limitations of claim 1. EX1001, Col. 26:60-27:35. However, claim 10 more generally recites "packet-identification criteria" instead of the "application-layer packet-header information" of claim 1, and clarifies that the packet-identification criteria "comprises a Differentiated Service Code Point (DSCP) selector" (i.e., network layer packet-header information). Claim 10 also explicitly claims "performing . . . the packet digest logging function," whereas claim 1 instead claims "performing . . . the packet transformation function," wherein the packet transformation function is defined as "comprising a packet digest logging function" in Claim 10.

88. Claims depending from independent claim 1 are directed toward various rules and types of application-layer packet-header information (*see* EX1001, Col. 25:56-26:59), while claims depending from independent claim 10 focus on

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

digest and logging functionality, as well as example packet-identification criteria (*see* EX1001, Col. 27:36-28:44). Based on my review of the prior art, each of the elements of the claims would have been obvious to one having ordinary skill in the art as of the priority date of the '213 Patent for at least the reasons detailed throughout this declaration.

C. The Prosecution History of the '213 Patent

89. The '213 Patent was filed as U.S. Patent Application No. 14/253,992 (the "'992 application") on April 16, 2014, as a continuation-in-part of U.S. Patent Application No. 13/657,010 (the "'010 application"). EX1002, pp. 1, 64. When the '992 application was filed, it included 20 claims. EX1002, pp. 38-43. On April 22, 2014, a preliminary amendment was filed to remove the priority claim to the '010 application. EX1002, pp. 107-109.

90. On October 15, 2015, the Examiner issued a restriction requirement, requiring that Patent Owner elect either Group 1 (claims 1-18 "drawn to managing network security proactively by applying dynamic network security policies") or Group 2 (claims 19-20 "drawn to updating network security rules by identifying and determining different set of network addresses"). EX1002, p. 369. Accordingly, on December 15, 2015, Patent Owner canceled claims 19-20 and elected to prosecute Group 1 (claims 1-18) with no additional argument or discussion. EX1002, pp. 414-422.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

91. During prosecution of the ’213 Patent, claims 1-18 were rejected by the Examiner in an Office Action dated February 26, 2016. EX1002, pp. 427-459. Specifically, the Examiner issued an obviousness-type double patenting rejection over U.S. Patent No. 9,137,205 (which was filed as the ’010 application, hereinafter the “’205 Patent”) and rejected the independent claims and certain dependent claims over the combination of Narayanaswamy (U.S. Publication No. 2009/0328219) in light of Kapoor (U.S. Publication No. 2008/0229415). EX1002, pp. 430-447. Certain dependent claims were rejected based on Kapoor and Narayanaswamy, combined with Erb (U.S. Publication No. 2014/0215574) and Nappier (U.S. Publication No. 2011/0185055). EX1002, pp. 447-458.

92. Patent Owner filed a response to the Office Action on August 23, 2016. EX1002, pp. 1198-1207. In that response, Patent Owner amended claims 1-3, 10, and 14; and added claims 21 and 22. EX1002, pp. 1199-1204. Independent claim 1 was amended to recite the additional element 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 (emphasis in original). Independent claim 10 was amended to specifically recite that “the packet-identification criteria comprises a Differentiated Service Code Point (DSCP) selector.” EX1002, p. 1206. Patent Owner subsequently only

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

generally asserted 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. EX1002, pp. 1205-1206.

93. Further, dependent claims 2 and 3 were amended to specifically recite "accept" and "deny" transformations, respectively, while newly added claim 21 defined the "application-layer packet-header information" of claim 1 to be "a Differentiated Service Code Point (DSCP) selector," and newly added claim 22 incorporated the above "routing" element that was added to claim 1 as an additional element of claim 10. EX1002, pp. 1200, 1204.

94. On November 8, 2016, Patent Owner filed a terminal disclaimer to overcome the obviousness-type double patenting rejection in view of the '205 Patent, as set forth in the Office Action dated February 26, 2016. EX1002, pp. 1229-1230.

95. The above-identified amendments were insufficient to procure allowance of the claims. Instead, the Examiner subsequently 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. 1236-1253. The Examiner also provided "Reasons for Allowance." EX1002, pp. 1248-1252.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

96. With respect to the Examiner's Amendment, independent claims 1 and 10 were amended to incorporate specific new elements beyond the previously identified elements added by Patent Owner. In particular, the Examiner only allowed the claims after an amendment specifically incorporating "a packet digest logging function" into independent claims 1 and 10, based on language in dependent claims 11 and 15. 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 <u>comprising a packet digest logging function</u> 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

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

~~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.

EX1002, pp. 1243-1245 (emphasis in original). Claim 10 was amended to recite similar elements relating to the "packet digest logging function," and was further amended to incorporate a "routing" element similar to that of claim 1. EX1002, pp. 1245-1246. Claims 11 and 15 were cancelled, as were claims 21 and 22. EX1002, pp. 1247-1248.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

97. 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," as incorporated from claims 11 and 15 in the claims by the Examiner's Amendment. EX1002, pp. 1248-1250. Specifically, the Examiner highlighted the allowable aspects of the claims over the cited art as follows:

> In claim **1:**
> "***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;***" in combination with other elements recited as specified in the independent claim(s).

EX1002, pp. 1244-1245

> In claim **10**:
> "***wherein the performing the packet digest logging function comprises:***

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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;

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; 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;" in combination with other elements recited as specified in the independent claim(s).

EX1002, pp. 1249-1250 (emphasis in original).

98. The Examiner provided a summary of the "closest prior art of record." EX1002, pp. 1250-1252. Specifically, the Examiner summarized Narayanaswamy as follows:

> Krishna Narayanaswamy (US 2009/0328219 A1, cited by the applicant in the 03/13/2015 IDS) teaches dynamic policy provisioning. A network security device may comprise a memory that stores a first policy that identifies a first set of patterns that correspond to a first set of network attacks and a second policy, and a control unit that applies the first policy to the network traffic to detect the first set of network attacks. The control unit, while applying the first policy, monitors parameters corresponding to one or more resources and dynamically determines whether to apply a second policy to the network traffic based on the parameters. The control unit, based on the dynamic

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

> determination, applies the second policy to the network traffic to detect a second set of network attacks and forwards the network traffic based on the application of the second policy. In this manner, the network security device may implement the dynamic policy provisioning techniques.

EX1002, pp. 1250-1251.

99. With respect to Kapoor, the Examiner provided the following summary:

> Kapoor et al. (US 2008/0229415 A1) teaches A flow processing for switching, security, and other network applications, including a facility that processes a data flow to address patterns relevant to a variety of conditions are directed at internal network security, virtualization, and web connection security. A flow processing facility for inspecting payloads of network traffic packets detects security threats and intrusions across accessible layers of the IP-stack by applying content matching and behavioral anomaly detection techniques based on regular expression matching and self-organizing maps. Exposing threats and intrusions within packet payload at or near real-time rates enhances network security from both external and internal sources while ensuring security policy is rigorously applied to data and system resources.

EX1002, p. 1251.

100. The Examiner also summarized two other references that were used to support rejections of various dependent claims as follows:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

> Erb et al. (US 2014/0215574 A1) teaches A hosted storage service stores a virtual data object that corresponds to data. The virtual data object includes metadata that enables access to the data in a delegated storage service but does not include the data. A delegate storage service stores the data. The hosted storage service receives a request for access to the virtual object and sends a response that includes metadata to access the data in a delegated storage service. The delegate storage service receives a request for access to the data based on the metadata. In response to receiving the request for access to the data object, the delegate storage service sends the data to the client application.
>
> Nappier et al. (US 2011/0185055 A1, cited by the applicant in the 07/21/2015 IDS) teaches a log correlation engine which analyzes various event logs that describe activity in a network to learn relationships between network identities and network addresses and generates alerts in response to discovering changes in the learned relationships. For example, the log correlation engine may identify authentication events described in the logs to map network identities to IP addresses, and may further analyze the logs to map the IP addresses to Ethernet addresses. Thus, the log correlation engine may discover new and changed relationships between the network identities, the IP addresses, and the Ethernet addresses.

EX1002, pp. 1251-1252.

101. Patent Owner subsequently filed Comments on Statement of Reasons for Allowance, in which Patent Owner generally stated "that specific excerpts quoted or paraphrased in the Statement . . . are not necessarily the only reasons for

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

distinguishing the claims from the various references, as there are many other features in the independent and dependent claims." EX1002, p. 1294. The '213 Patent later issued on February 7, 2017. EX1002, p. 1305.

D. The Priority Date of the '213 Patent

102. I understand that the '213 Patent was filed on April 16, 2014. I have therefore analyzed anticipation and obviousness with reference to a priority date of April 16, 2014.

VI. STATE OF THE ART PRIOR TO THE '213 PATENT

103. As discussed above, in Section IV, the prior art was mature well before April, 2014. Devices that could inspect packets and identify, modify, monitor, and log the contents of those packets were commonplace by that era. As I describe in further detail, below, a variety of systems and patents described systems that performed all of these actions.

VII. CLAIM CONSTRUCTIONS

A. Legal Standard and Constructions Used

104. As discussed above, I applied the broadest reasonable interpretation ("BRI") standard to my review of the claims of the '213 Patent and comparison to the prior art. I 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). In some cases, such as where the specification defines a term, the construction

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

may be the same under both standards. Where applicable, I also used Centripetal's proposed constructions and the agreed-upon constructions in the litigation between Centripetal and Keysight Technologies, Inc. in my analysis as well. *See* Section II.C. My conclusions and opinions are the same regardless of which claim construction standard is used, as shown in detail herein.

1. Dynamic security policy

105. The '213 Patent specification states, and thus a POSA would understand, that the term "dynamic security policy" 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. The '213 Patent explains that a "dynamic security policy" may be created automatically or manually. For example, a "dynamic security policy" may be "received from the security policy management server" (*id.*, Col. 1:41-43), which may "automatically create or alter one or more . . . rules," e.g., based on new malicious network traffic aggregated by a service (*id.*, Col. 14:56-15:5). Additionally or alternatively, "an administrator associated with security policy management server 120 may manually construct one or more dynamic security policies offline and then utilize security policy management server 120's management interface 510 to load such dynamic

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

security policies into security policy management server 120's memory 502." *Id.*, Col. 14:41-47.

106. I understand that Centripetal, in another proceeding concerning the the '213 patent, has offered a claim construction for a "dynamic security policy" of "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. 4; EX1007, p. 9. This contradicts the broader disclosure of the term in the '213 Patent specification. In my opinion, in the context of the '213 Patent, a dynamic security policy is not limited to a "non-static" set of rules and may be updated dynamically (e.g., automatically) or statically (e.g., manually). *See* EX1001, Col. 4:58-63, 14:41-47; 14:56-15:5. However, as discussed in the Analysis section below, my conclusions would be the same even if Centripetal's narrower construction of "dynamic security policy" were adopted.

107. Accordingly, in my opinion, the proper construction of the term "dynamic security policy" includes at least "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."

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

2. Rule/Rules

108. The ’213 Patent specification states, and thus a POSA would understand, that a “dynamic security policy” may include “one or more rules” (EX1001, Col. 6:11-12), and “[e]ach rule may specify criteria and one or more packet transformation functions that should be performed for packets associated with the specified criteria” (EX1001, Col. 7:10-13). Examples of criteria include “a set of network addresses” (EX1001, Col. 8:42-43, Col. 8:59-60) and “a five-tuple of values selected from packet header information, specifying a protocol type of the data section of the IP packet (e.g., TCP, UDP, ICMP, or any other protocol), one or more source IP addresses, one or more source port values, one or more destination IP addresses, and one or more destination ports” (EX1001, Col. 7:14-20).

109. I understand that Centripetal, in the *Keysight* lawsuit, has offered a claim construction for a “rule/rules” of “condition or set of conditions that when satisfied cause a specific action to occur.” EX1005, p. 7; EX1006, p. 6; EX1007, p. 10. This definition differs from the disclosure in the ’213 Patent specification. For example, the ’213 Patent specification does not use the term “condition” with respect to a rule, nor does the specification indicate that a satisfied condition “cause[s] a specific action to occur.” Rather, rules specify functions (or actions) that “should be performed” on packets associated with the specified criteria. *See* EX1001, Col. 7:10-12. However, as discussed in the Analysis section below, my conclusions

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

would be the same even if Centripetal's unsupported, different construction of "rule" were adopted.

110. Accordingly, in my opinion, in the context of the '213 Patent, the term "rule" refers to at least part of a "dynamic security policy" that "specifies criteria and one or more packet transformation functions that should be performed for packets associated with the specified criteria."

3. Security policy management server

111. The '213 Patent specification states, and thus a POSA would understand, that the term "security policy management server" includes at least "any computing device configured to communicate a dynamic security policy to a packet security gateway." EX1001, Col. 4:53-55. In the *Keysight* lawsuit, the parties agreed that this term means "a server configured to manage policies and communicate a dynamic security policy to a packet security gateway." EX1007, p. 6.

112. For my purposes, without necessarily agreeing that the scope of the term is this narrow, I used the agreed construction from the *Keysight* litigation: "a server configured to manage policies and communicate a dynamic security policy to a packet security gateway."

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

4. Packet security gateway

113. The ’213 Patent specification states, and thus a POSA would understand, that the term “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, the agreed construction is “a gateway computer configured to receive packets and perform a packet transformation function on the packets.” EX1007, p. 6.

114. For my purposes, without necessarily agreeing that the scope of the term is this narrow, I used the agreed construction from the *Keysight* litigation: “a gateway computer configured to receive packets and perform a packet transformation function on the packets.”

5. Packet transformation function

115. The ’213 Patent specification describes the term “packet transformation function” as an action taken on a packet, including: “forward[ing] . . . packets into [a] network” EX1001 (Col. 2:50-51); “forward[ing] . . . packets out of [a] network” (Col. 2:52-54); “forward[ing] . . . packets to an IPsec stack” (Col. 2:54-57); “drop[ping] . . . packets” (Col. 2:57-59); “routing packets . . . to a network address different from a destination network address specified by the packets” (Col. 2:38-42); “rout[ing] the packets to [a] monitoring device” (Col. 16:37-44); “not only . .

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

. accept[ing] . . . but . . . rout[ing] [packets] to a monitoring device" (Col. 8:8-14, FIG. 3); and "queu[ing] packets in one or more queues" (Col. 10:6-10).

116. In the *Keysight* lawsuit, Centripetal took the position that this term excludes forwarding and dropping packets. EX1005, pp. 4-5; EX1006, p. 2. The specification of the '213 Patent makes clear that the term "packet transformation function" includes both "forwarding" and "dropping" packets. For example, the '213 Patent describes a packet transformation function "other than" forwarding or dropping as follows:

> [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**. For example, rule 3 306 may specify that IP packets containing TCP packets, originating from a source IP address that begins with 150, having any source port, destined for any IP address that begins with 120, and destined for port 90 **should not only have an accept packet transformation function performed on them, but should also be routed to a monitoring device**.

EX1001, Col. 8:3-14 (emphasis added). In my opinion, this example does not exclude forwarding or dropping from the definition of packet transformation function. The quoted language expressly states that the dynamic security policy *may include* packet transformation functions other than forwarding or dropping. A POSA would not understand this statement to be excluding forwarding or dropping as

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

acceptable packet transformation functions. Instead, a POSA would understand that the example identifies functions in addition to forwarding or dropping that are included in the definition of packet transformation function. A packet transformation function "other than" forwarding or dropping includes, at least, "accepting a packet and also routing the packet to a monitoring device." Additionally, in my opinion, a POSA would also understand packet transformation functions "other than" forwarding and dropping to include other actions, including at least "routing a packet to a monitoring device," "routing a packet to a network address different from a destination address specified by the packet" and/or "queuing packets in one or more queues."

117. It would be illogical for the specification to explicitly exclude forwarding and dropping from the list of possible transformation functions used in certain situations if forwarding and dropping were not packet transformation functions in the first place. By way of analogy, the phrase "fruit other than apples and oranges" makes sense, as both apples and oranges are types of fruit. By contrast, the phrase "vegetables other than apples and oranges" is illogical, as apples and oranges are not types of vegetables, and, as such, there is no need to exclude them. Indeed, as noted above, the '213 Patent specification explicitly includes forwarding and dropping as packet transformation functions. EX1001, Col. 2:50-59. The conclusion that forwarding and dropping are packet transformation functions is

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

further supported by claims of the ‘213 patent, including claims 2 and 3. Claim 2 recites, in relevant part, “wherein the 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.

118. Performing a packet transformation function involves performing the action on the packet and may be achieved in different ways depending on the particular action. For instance, “forwarding” may simply involve allowing a packet to pass to its destination (Col. 6:22-26, 10:58-59, 11:51-57); whereas “dropping” a packet may involve blocking a packet from passing to its destination (Col. 15:1-5) and, in some cases, may involve sending the packet to a null device file or “infinite sink” (Col. 6:28-31). In contrast, “routing” a packet to a different network address may involve “alter[ing] or modify[ing] the destination address of the packets” (Col. 10:59-63), “assign[ing] such packets to a particular Layer-2 VLAN . . . which may or may not be on the IP-layer path that the packet would have taken if it were routed according to the packet’s destination IP address” (Col. 10:63-11:3), or “encapsulate the packets (or a copy of the packets) with an IP header having an address corresponding to [a] monitoring device” (Col. 17:5-8).

119. I understand that Centripetal, in the *Keysight* lawsuit concerning the ’205 patent, has offered a claim construction for a “packet transformation function” of a “function that transforms one or more packets from one state to another.”

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

EX1005, p. 4; EX1006, p. 2; EX1007, p. 8. Centripetal further states that "this transformation or change in the state of a packet can happen in a number of different ways, including replacing headers in the packets, removing information from the headers of packets, or encapsulating them." EX1006, p. 2. In my opinion, in the context of the '213 Patent, a packet transformation function is not limited to "transform[ing] one or more packets from one state to another," which (despite being unclear as to what "state" means) would, according to Centripetal, exclude explicit examples of packet transformation functions (e.g., forwarding or dropping) identified in the specification of the '213 Patent. Further, in my opinion, any construction should include, at least, packet transformation functions of "accepting a packet and also routing the packet to a monitoring device," "routing a packet to a monitoring device," and/or "routing a packet to a network address different from a destination address specified by the packet." Again, regardless of which of these claim constructions may be adopted by the Board, my opinions with respect to the validity of the '213 Patent would not change.

120. Accordingly, in my opinion, the proper construction of the term "packet transformation function" refers to "an action taken on a packet," including actions such as "forwarding a packet," "dropping a packet," "accepting a packet and also routing the packet to a monitoring device," "routing a packet to a monitoring

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

device," "routing a packet to a network address different from a destination address specified by the packet," and/or "queuing packets in one or more queues."

6. Monitoring device

121. A POSA would understand that the term "monitoring device" has a meaning of a "device configured to copy (or store) packets or data contained within them." The '213 Patent specification does not define a monitoring device but provides that a monitoring device "may store the packets or data contained within them for subsequent review or analysis (e.g., by a law enforcement or national security authority). 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. In some examples, a "network address corresponding to the monitoring device may be different from the packets' destination network address." EX1001, Col. 11:19-22. These packets are encapsulated into a new packet with a destination address of the monitoring device, routed to the monitoring device where the device may "strip the IP header from [packets], and then forward the packets to their destination address" (Col. 11:26-33).

122. The definition located within the '213 Patent also comports with standard dictionary definitions of "monitor" and "monitoring. For example, Dictionary.com defines the verb "monitor" to be "to observe, record, or detect (an operation or condition) with instruments that have no effect upon the operation or

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

condition." EX1015. This definition requires observing, recording, or detecting a condition, while the '213 Patent requires that its monitoring devices copy or store information from within packets. Thus, the '213 Patent reflects what a POSA would expect from a monitoring device.

123. Accordingly, in my opinion, the proper construction of the term "monitoring device" refers to a "device configured to copy (or store) packets or data contained within them."

VIII. GROUNDS OF INVALIDITY

124. I am of the opinion that a POSA would understand that claims 1-9 of the '213 Patent are rendered obvious by the Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5 ("ACNS Guide") (released in 2006) (EX1008) in combination with U.S. Patent No. 9,172,627 ("Kjendal") (EX1009).

125. I am also of the opinion that a POSA would understand claims 10-16 of the '213 Patent are rendered obvious by the ACNS Guide in light of the specification titled "An Architecture for Differentiated Services," also known as the Diffserv standard, as defined in RFC 2475 (the "Diffserv Specification") (released in 1998) (EX1011). The foregoing references may be further combined with the Kjendal to demonstrate additional points of obviousness for claims 10-16.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

126. The following two grounds of invalidity are discussed in this declaration:

Ground	Claims	Combined Art
1	1-9	ACNS Guide, in light of Kjendal
2	10-16	ACNS Guide + Diffserv Specification + Kjendal

I understand that two other grounds of invalidity are included in a separate proceeding.

127. The ACNS Guide is a written publication, titled "Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5." The ACNS Guide is copyrighted as of 2006 and was published and made available to members of the public by at least March, 2008. EX1020 (Fuchs Decl.), ¶¶8-12, 13, and 15. Accordingly, the ACNS Guide qualifies as prior art to the '213 Patent under at least AIA 35 U.S.C. §§ 102(a)(1) and (2). Because the information in the ACNS Guide was published more than a year before the earliest possible priority date for the '213 Patent, I understand that the ACNS Guide cannot be excluded under any of the provisions of AIA 35 U.S.C. § 102(b).

128. Kjendal is U.S. Patent No. 9,172,627, titled "Device and Related Method for Dynamic Traffic Mirroring." Kjendal was filed on March 15, 2013 and published on September 18, 2014. EX1009. Kjendal claims priority to no previous applications. Accordingly, Kjendal qualifies as prior art to the '213 Patent under at

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

least AIA 35 U.S.C. §§ 102(a)(1) and (2). Because Kjendal names other inventors than the ’213 Patent, was never owned by the same assignee, and is effective as prior art more than a year before the earliest possible priority date for the ’213 Patent, under 35 U.S.C. § 102(d), I understand that Kjendal cannot be excluded under any of the provisions of AIA 35 U.S.C. § 102(b).

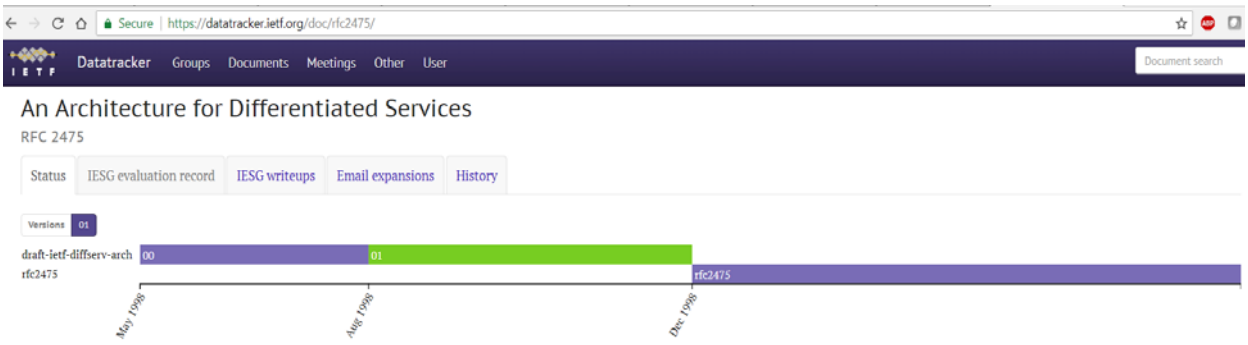
129. The Diffserv Specification is a written publication, titled “An Architecture for Differentiated Services,” and lists its copyright date as December, 1998 and publication to the public in December, 1998. EX1011, p. 1. I have personal knowledge that it has been freely available to the public (e.g., via the web) since approximately that time. It is currently available on the website of the Internet Engineering Task Force (“IETF”), a working-group subsidiary of The Internet Society, who owns the copyright to each of the various internet-related specifications to which I refer in this declaration. The Diffserv Specification may be found at: https://tools.ietf.org/html/rfc2475 (Last visited June 13, 2018).

130. I first encountered the Diffserv Specification prior to December, 1998, when the specification was in draft from. I am in possession of an electronic copy of the final Diffserv Specification (RFC 2475) that I downloaded from a publicly accessible website in 1999, well before the earliest filing date of the ’213 Patent. Therefore, I can attest that the Diffserv Specification was publicly available at least

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

one year before the effective filing date of the ’213 Patent and thus represents authentic prior art to the ’213 Patent.

131. According to the IETF, RFC 2475 has not changed since it published in December 1998. *See* https://datatracker.ietf.org/doc/rfc2475/, as shown in the below diagram:



According to the standard records kept by IETF, draft versions of the Diffserv Specification were published in May 1998 and August 1998, with the final version published in December, 1998 with no additional versions published after that time.

132. According to the IETF website, while published RFCs do not change, errata may be issued for technical and syntax errors that are undetected in the editing process. *See* https://www.ietf.org/standards/rfcs/ (Last viewed June 13, 2018). The Diffserv Specification has no reported errata and has not been edited since at least one year before the effective filing date of the ’213 patent. *See* https://www.rfc-editor.org/errata_search.php?rfc=2475 &rec_status=15&presentation=table (Last viewed, June 13, 2018).

133. I am aware that other professionals in the field (POSAs) were in possession of the Diffserv Specification before the alleged invention of the ’213 Patent as well. For example, U.S. Publication No. 2004/0114518, published June 17, 2004, lists the Diffserv Specification as background to its invention and discusses RFC 2475 a number of times within its disclosure, related to known methods of differentiated services. EX1018, ¶¶[0004], [0013], [0025], [0033], and FIG 2 (depicting prior art, Diffserv architecture).

134. I am also aware of the process by which IETF promulgates specifications and requests for comment. The process is, itself, documented and defined in RFC 2026 (the “RFC Specification”), issued October, 1996. EX1017, p. 1. I have been personally aware of the RFC process by the IETF for many years and was in possession of the RFC Specification before April, 2014. I also understand that the RFC Specification has previously been authenticated by the Board as an “ancient document” and that the Board has recognized it and other RFC documents “as authoritative, well-known Internet publications.” *Apple Inc. v. VirnetX, Inc.*, IPR2016-01585, Paper 32 at 56-58 (P.T.A.B. Feb. 20, 2018).

135. According to the RFC Specification, any interested person may obtain RFC documents from several different internet hosts that use document-retrieval systems available on the World Wide Web, as well as other internet-based document-retrieval systems. EX1017, p. 6. Therefore, an interested person would

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

be able to access any RFC reference published by the IETF, including the Diffserv Specification (EX1011, RFC 2475) and the later-discussed HTTP 1.1 Specification (RFC 2616). Moreover, networking textbooks frequently encourage students to access and consider IETF RFCs. For example, the texts that I used in my class prior to 2014 specifically reference RFC 2475 and 2616, and encourage students to view these documents.

136. Accordingly, the Diffserv Specification qualifies as prior art to the '213 Patent under at least AIA 35 U.S.C. §§ 102(a)(1) and (2). Because the information in the Diffserv Specification was published more than a year before the earliest possible priority date for the '213 Patent, I understand that the Diffserv Specification cannot be excluded under any of the provisions of AIA 35 U.S.C. § 102(b).

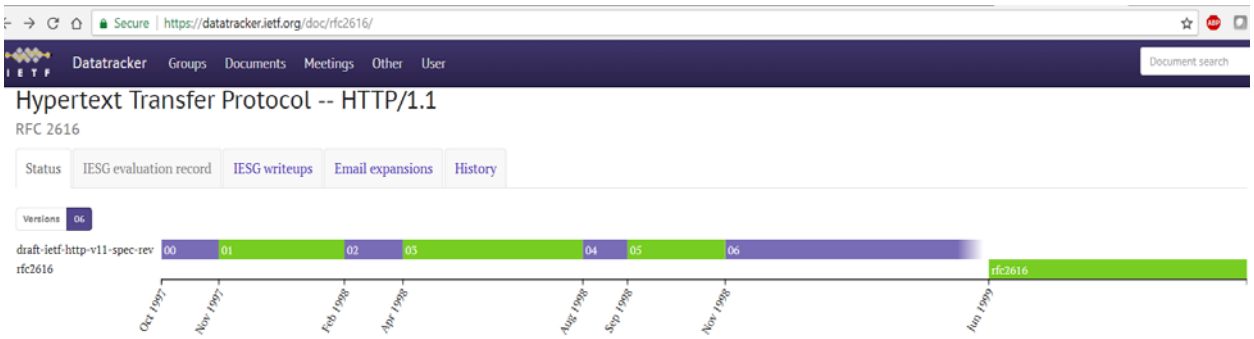
137. The HTTP 1.1 Specification is a written publication, titled "Hypertext Transfer Protocol – HTTP/1.1," and lists its copyright date as June, 1999 and publication to the public in June, 1999. EX1010, p. 1. I have personal knowledge that it has been freely available to the public since approximately that time. It is currently available on the IETF website. The HTTP 1.1 Specification may be found at: https://tools.ietf.org/html/rfc2616 (Last visited June 13, 2018).

138. I first encountered the HTTP 1.1 Specification shortly after the time of its publication and have downloaded several copies of the final HTTP 1.1 Specification prior to April, 2014. Therefore, I can attest that the HTTP 1.1

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

Specification was publicly available at least one year before the effective filing date of the ’213 Patent and thus represents authentic prior art to the ’213 Patent.

139. According to the IETF, RFC 2616 has not changed since it published in June 1999. *See* https://datatracker.ietf.org/doc/rfc2616/, as shown in the below diagram:



According to the standard records kept by IETF, seven draft versions of the HTTP 1.1 Specification were published between October, 1997 and November, 1998, and the final version published in June, 1999, with no additional versions published after that time.

140. I understand that the HTTP 1.1 Specification has several reported errata regarding minor typographical and technical errors. *See* https://www.rfc-editor.org/errata_search.php?rfc=2616&rec_status=15&presentation=records. The HTTP 1.1 Specification has not, however, been substantively edited since at least one year before the effective filing date of the ’213 patent.

141. Based on both my own experience and the RFC Specification, any interested person would have had access to the HTTP 1.1 Specification since June,

1999. *See* EX1017, p. 6 (describing how each version of an RFC is a distinct publication).

142. I am aware that other professionals in the field (POSAs) were in possession of the HTTP 1.1 Specification before the alleged invention of the '213 Patent as well. For example, U.S. Patent No. 8,156,206, issued April 10, 2012, provides detailed discussion of RFC 2616 (the HTTP 1.1 Specification) and its teachings with relation to the specification of that patent. EX1019, Col. 7:61-8:10 ("Development of HTTP was coordinated by the W3C and the Internet Engineering Task Force (IETF), culminating in the publication of a series of RFCs, most notably RFC 2616 (1999), which defines HTTP/1.1, the version of HTTP in common use today…").

143. Accordingly, the HTTP 1.1 Specification qualifies as prior art to the '213 Patent under at least AIA 35 U.S.C. §§ 102(a)(1) and (2). Because the information in the HTTP 1.1 Specification was published more than a year before the earliest possible priority date for the '213 Patent, I understand that the HTTP 1.1 Specification cannot be excluded under any of the provisions of AIA 35 U.S.C. § 102(b).

144. The sections that follow provide facts and reasons to support my opinions concerning the anticipation or obviousness of claims 1-16 in light of the prior art considered herein.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

A. Ground 1: Claims 1-9 of the ’213 Patent are invalid under 35 U.S.C. § 103(a) on the ground that they are rendered obvious by ACNS in combination with Kjendal.

145. I am of the opinion that a POSA would have found that the subject matter of the ACNS Guide combined with Kjendal renders obvious claims 1-9 of the ’213 Patent.

1. Invalidity Analysis

146. The ACNS Guide is a configuration guide for the Cisco Application and Content Networking System, version 5.5, released well over one year before the priority date of the ‘213 Patent. EX1008, p. 1-2; Ex 1020 (Fuchs Decl). ACNS is a sophisticated platform for managing and delivering content (digital media) for an enterprise. As such, it includes facilities for inspecting requests for content entering the platform and transforming and logging certain requests according to dynamically configurable sets of policies.

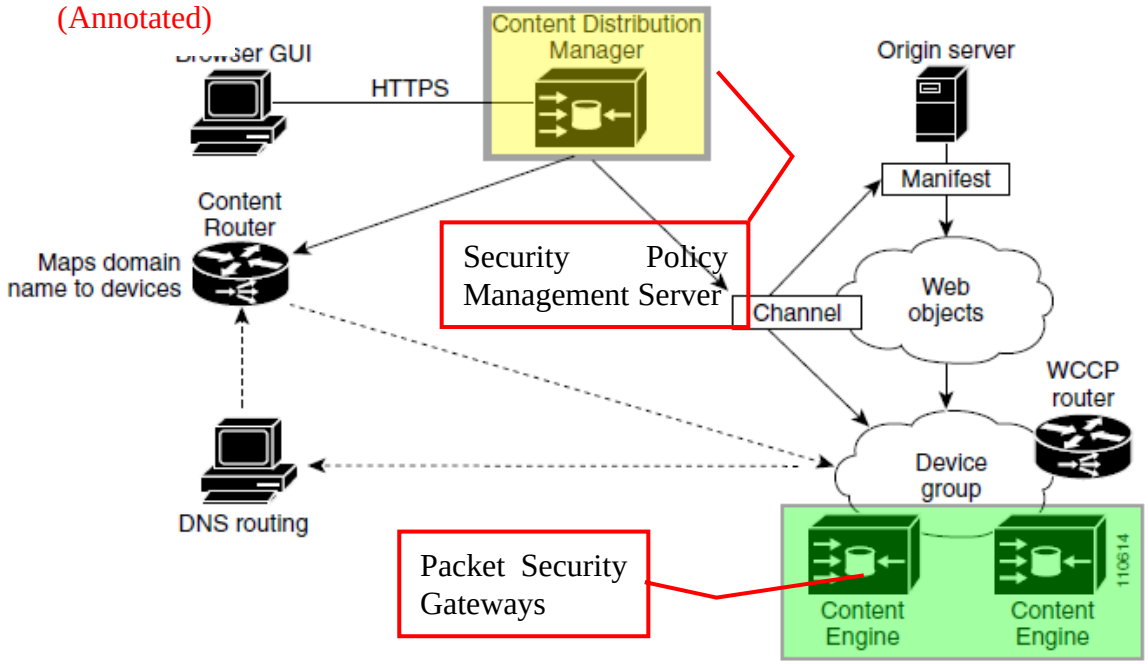
147. The ACNS Guide teaches that the ACNS software may operate in one of four modes: Content Engine, Content Distribution Manager, Content Router, and Cisco IP/TV Program Manager. EX1008, p. 1-2. The ACNS Guide describes the Content Distribution Manager (“CDM”) as a device that meets the requirements for the claimed “security policy management server” in each of the independent claims (claims 1 and 10). Moreover, the ACNS Guide describes using the ACNS system’s Content Router as a device that meets the requirements for the claimed “packet

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

security gateways." Figure 1-2 (annotated) of the ACNS Guide shows a simplified layout for use of the ACNS software and its elements corresponding to the claim elements:

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

(Annotated)

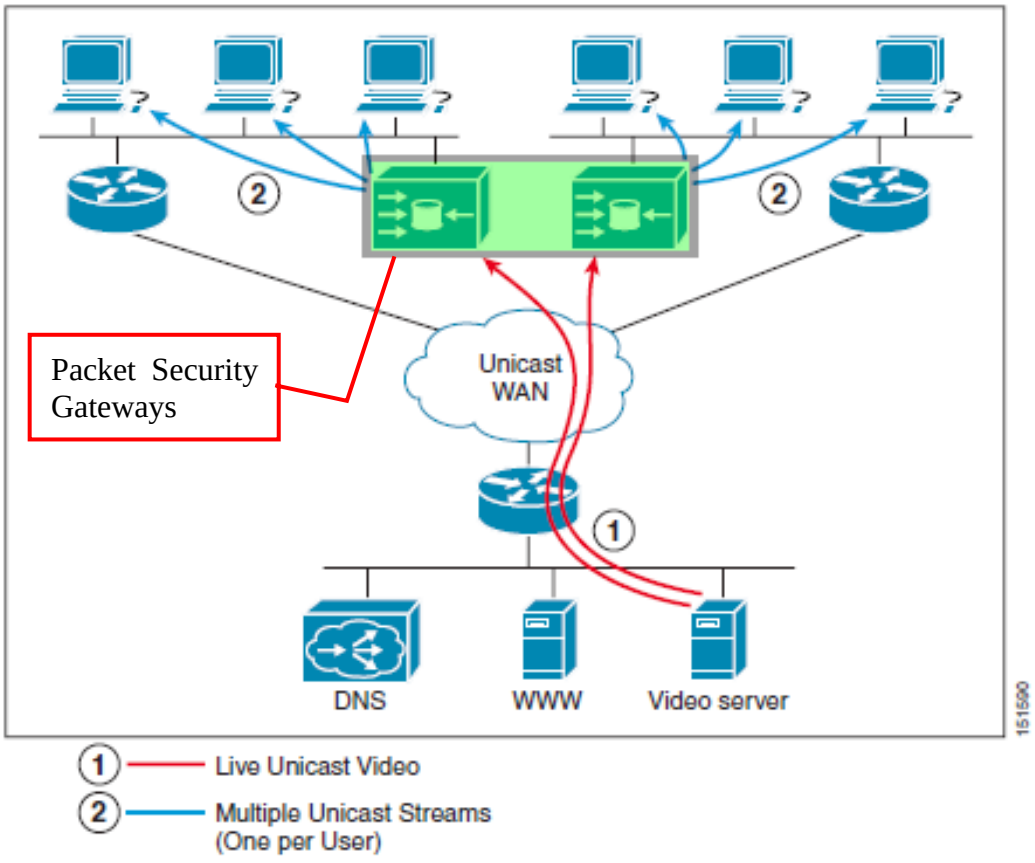


The Content Engine and Content Distribution Manager (CDM) elements have been highlighted for clarity. In this setup, an administrator may use a web browser to connect to the CDM and cause it to issue various policies to Content Routers and Content Engines. The ACNS Guide teaches that those policies may include rules that comprise a rule specifying application-layer packet-header information and a packet transformation function that comprises a packet digest logging function. The Content Engines execute these policies while acting as security gateways for

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

network elements, as shown in Figure 9-1 of the ACNS guide (Content Engines highlighted for clarity) showing a unicast streaming setup for the ACNS system:

Figure 9-1 Unicast Live Stream Splitting
(Annotated)



The above diagram is merely one iteration of how Content Engines may operate within the ACNS system. Other layouts and broadcast methods exist, such as multicast, hybrid-unicast-multicast, demand cached, and pre-positioned video setups. *See* EX1008, pp. 9-1 – 9-6. The ACNS Guide further describes the Content Engines of the ACNS system receiving packets, identifying packets based on received policies, performing packet transformation functions defined within the

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

received policies that comprise identifying application-packet header information from within the packets, generating records from the identified information, and then reformatting the information into a logging standard, and finally, routing the packet to a monitoring device. These activities will be described in more detail, below.

148. 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., 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, Col. 10:46-49 (emphasis added). Kjendal also teaches identifying packets to mirror by analyzing the contents of their application-layer packet-header information. 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…"). 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. I personally routinely employed port mirroring as part of the networking research that transpired in my UNC lab in the late 1990s and early 2000s. Therefore, a POSA would very reasonably expect to include monitoring and logging features from Kjendal to

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

improve ACNS by allowing the system to send packets to other devices for analysis and logging, and such a POSA would expect such a combination to yield predictable results (packets would be mirrored to the monitoring device without problems).

a) 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

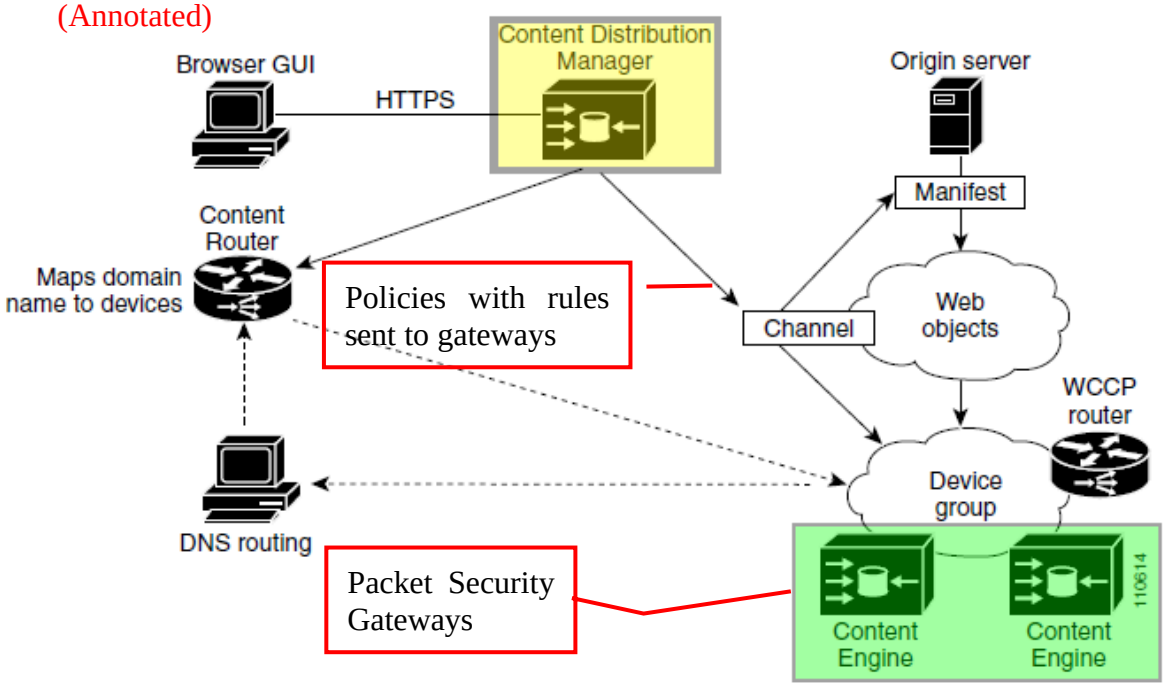
149. The ACNS Guide teaches limitation [1.1] of Claim 1.

150. Within the ACNS system, described in the ACNS Guide, the Content Engines (marked in green, below) represent the claimed security gateways and are associated with a Content Distribution Manager (marked in yellow). These "gateways" receive dynamic security policies with at least one rule specifying application-layer packet-header information.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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

(Annotated)



EX1008, Fig. 1-2 (annotated), p. 1-11.

151. The ACNS Guide explains that the software may act in four different modes and that the Content Distribution Manager remotely controls multiple devices via policies. *Id.*, p. 1-2 ("Cisco ACNS software allows content services to be configured, reconfigured, and monitored centrally from the Content Distribution Manager graphical user interface (GUI), Cisco's web-based application management tool"). The role of the policy management server is performed by the Content Distribution Manager. *Id.* ("…the Content Distribution Manager…also manages policy settings and configurations on individual Content Engines.").

152. As previously noted, the role of packet security gateway is performed by the Content Engine. Such security functions include the Content Engines

intercepting traffic and enforcing policies within the network, as a gateway. *Id*., p. 45. The ACNS Guide offers an example of the Content Engines acting as a security gateway and enforcing policies:

> For example, BigCorp Enterprise wants to make sure that the following policy, security, and budgetary objectives are met:
>
> •Only full-time employees can access the web.
>
> •No employee can access objectionable content through the corporate network.
>
> •Viruses such as "Code Red" are prevented from being propagated inside the corporate network.
>
> •Monthly payments to the company's ISP are reduced.
>
> BigCorp Enterprise can achieve all of these objectives by placing Content Engines at the network edge, where the corporate network interfaces with the Internet, and by placing Content Engines at remote branch offices. **The Content Engines can then intercept content requests made by end users and enforce the above policies.**

EX1008, p. 1-9 (emphasis added). The ACNS Guide emphasizes that the Content Engine may perform such security measures "because of the Content Engine's special position as an 'in-line' device between end users and the Internet." *Id*. It is my opinion that such actions are those of the claimed packet security gateway.

153. The dynamic policies, generated at the Content Distribution Manager (CDM), and sent to (and hence received by) a plurality of Content Engines, include at least one rule specifying application-layer packet-header information. For example, the CDM may propagate a rule allowing or refusing packets with certain HTTP Request Methods. EX1008, pp. 8-23 – 8-24. These policies are dynamic

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

because they are generated by the CDM and sent to the Content Engines based on UI input from the administrator, but without a user needing to write the rules, themselves. HTTP is an application layer protocol and hence HTTP packets are application packets. Moreover, according to the HTTP 1.1 Specification, the Request Method for such a packet may be found in the application-layer packet-header information. EX1010, p. 24 ("A [HTTP] request message from a client to a server includes, within the first line of that message, **the method to be applied to the resource**, the identifier of the resource, and the protocol version in use.") (emphasis added).

154. As discussed in detail, above, "packet" is a generic term for the transmission structure of data sent across some form of network. Within the specification of the '213 Patent, the Patent Owner uses the term "packet" to refer to a network-layer structure (EX1001, Col. 6:36-41 (referring to "packets" specifically operating on the network layer), a transport-layer structure (EX1001, Col. 8:8-14 (referring to "IP packets" (network layer) and "TCP packets" (transport layer))), and an application-layer structure (EX1001, Col. 7:49-56 (referring to "application-layer hyper-text transfer protocol (HTTP) packets"). Because the '213 Patent uses the term "packet" so broadly, I will analyze the patent and prior art using the same terms. As explained above, packets may be viewed as pertaining to various "layers" of network. For example, an HTTP (application layer) packet is likely to be enclosed

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

in one or more TCP (transport layer) packets, each of which is in turn likely enclosed in an IP (network layer) packet. *See* Section IV.A, Figure 8.

155. An application may recognize a packet as being an HTTP packet by the contents of the first line of the application-layer header. Per the HTTP 1.1 Specification, the first line necessarily has the format: "Method URL HTTP-Version." *See* EX1010, p. 10. At the time of alleged invention, April 2014, the HTTP 1.1 standard had been in place for nearly 15 years and hence a POSA would be well aware of the standard and understand that its teachings may be combined with that of the ACNS Guide, because the ACNS Guide explicitly states that, by default, the Content Engine allows Request Methods of HTTP 1.1. EX1008, p. 8-23 ("**HTTP 1.1 request methods (for example, GET, HEAD, or POST)** are supported by default.")) (emphasis added). Therefore, to allow or refuse packets based on the included request method, the ACNS system would necessarily identify such packets to be excluded or allowed based on application-layer packet-header information.

156. The ACNS Guide teaches analyzing an HTTP packet to determine the request method located within the application-layer packet-header information. EX1008, pp. 8-23 – 8-24. If the request method, such as "GET" or "POST" or some other method name, is contained in the "allowed" policy list, the ACNS system will allow the packet and forward or process it as required. *Id*. Administrators may

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

modify this list of allowed methods by adding or subtracting request method names. *Id*. This implementation is an example of a "whitelist" or "allow list" filter, as discussed in the "Firewalls and Network Security" portion of the Background on Computer Networking, section IV.C, above, and in the '213 Patent specification. *See* EX1001, Col. 8:55-63.

157. The ACNS Guide also teaches that if a request method in an HTTP request is *not* on the allow list, then the ACNS system will notify the sender of the error and log the name of the disallowed request method to a list of unsupported methods. EX1008, p. 8-23. This function represents rule-based logging of application-layer packet-header information, because the ACNS guide specifies that the request method name, extracted from the HTTP packet header, is logged in response to the system identifying a packet, based on the application-layer packet-header information and determining that the packet should be disallowed. Immediately after explaining how the system keeps track of the requested-but-unsupported methods, the ACNS Guide teaches system administrators how to add new method names to the list of allowed methods. EX1008, pp. 8-23 – 8-24. A POSA, reading this section, would understand that a system administrator would likely monitor this list to help determine which methods the system should support in the future.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

158. Another example of the Content Engine specifying application-layer packet-header information as part of a rule within a policy is found in Chapter 16 of the ACNS Guide, which teaches configuration of ICAP servers running on Content Engines. EX1008, Chapter 16, pp. 16-1 – 16-66, *generally*. ICAP, the Internet Content Adaptation Protocol, is an HTTP-like protocol specified in RFC 3507 that is used to communicate between Web proxy servers. The ACNS Guide explains ICAP as:

> 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, p. 16-1. A POSA would understand that the ACNS Guide's use of the terms "adaptation" or "rule action" for the actions of the ICAP server is functionally identical to the '213 Patent's use of the term "transformation." In both cases, the word describes how the server will identify a packet based on specified information and interact with the packet, all based on a set of rules. The ACNS Guide goes on to describe rules by which an ICAP server can identify packets and the "rule actions" to apply, based on those rules. EX1008, pp. 16-15 – 16-25. Such rules can include specifying packets based on the IP source address, IP destination address, destination port, and application-layer packet-header information such as URL, and other HTTP header fields (including the request method). EX1008, 16-16 – 16-17. The ICAP server may perform actions (transformations), based on such rules, that include

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

allowing, dropping, appending a username, modifying the DSCP field, redirecting the packet to a specified URL, and other transformations. *Id*., 16-19 – 16-23.

159. Therefore, the ACNS Guide teaches "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**."**

b) 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;

160. The combination of the ACNS Guide and Kjendal teaches limitation [1.2] of claim 1.

161. The ACNS Guide teaches packet transformation functions, as that term would be understood in the '213 Patent. For example, the ACNS Guide teaches packet transformation functions in the form of digest logging of packets. EX1008, Chapter 19, pp. 19-1 – 19-22, *generally*. In ACNS, the Content Distribution Manager may instruct a plurality of Content Engines, via a single policy, whether to log packet transactions and how to log those packets. *Id.* The Content Distribution Manager GUI provides functionality for managing that security policy that dictates whether

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

and how the ACNS system logs information. The security policy may be sent to one or more Content Engines. *Id*., 19-10 – 19-11. The security policy for the ACNS System may also be managed through a command line interface. *See* EX1008, Appendix C, pp. C-1 – C-27, *generally*.

162. The information used by the ACNS system to log packets may include application-layer packet-header information, such as HTTP packet header information. *Id*. The ACNS Guide teaches logging in a variety of formats, including "apache," "extended-squid," "squid," and "custom" logging formats. Each format contains (or may be configured to contain) detailed HTTP application-layer packet header information such as "remotehost," "URL," "timestamp," and other information. *See* EX1008, pp. 19-1 – 19-5. For example, a "squid" format log contains the following information: time, remotehost, code/status, bytes, HTTP method, URL, rfc931 (authentication server), peerstatus/peerhost, and type, each separated by spaces and filled in with a "-" if the field is not applicable to a particular log entry. The ACNS Guide provides the following example of a "squid" log entry:

> 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, p. 19-1. A POSA would understand that, per the HTTP 1.1 Specification, elements appearing in these log entries, such as the URL, HTTP method, and code/status, may only be found in the packet-header information of an HTTP

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

application-layer packet. Further examples of packet digest logging functionality may be found in the ACNS Guide's section on custom logging, which identifies how a user may set up their own logging format, including numerous fields from the HTTP application-layer packet-header information. EX1008, pp. 19-2– 19-5.

163. The ACNS Guide further teaches that logs may be formatted in syslog format and sent to a remote logging server as part of the ACNS System's "real-time transaction logging." EX1008, pp. 19-19 – 19-22. In the "real-time transaction logging," packets that are already logged, such as packets logged in squid format, may be re-formatted in a syslog format and forwarded to a syslog server for further monitoring and analysis. An administrator may configure the ACNS system to send *all* logs to the remote syslog server. *Id*. at 717-718. The UI interface and command-line commands for doing so are described in ACNS. *Id*. However, the ACNS Guide recognizes that sending too many log entries may overload the syslog server. *Id*. Thus, by default, the ACNS system will only send log entries related to authentication failures. *Id*. In order to determine which packets are related to authentication errors, however, the system would need to identify those packets based on the HTTP application-layer packet-header information, as that is where the code for authentication errors may be found in HTTP packets.

164. Further, the '213 Patent itself acknowledges that logging in standard formats (e.g., syslog) is known. *See* EX1001, Col. 11:54-57; 23:14-18. The patent

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

essentially admits that a POSA would have known how to format logs of digested information about packets, because such formatting was standard. In fact, the syslog standard has existed since before 1990, was documented in 2001 within RFC 3164, and the current iteration was further standardized in 2009 by RFC 5424. Therefore, the ’213 Patent is correct in recognizing that the use of standard formats existed before the filing of the patent application. I note that formatting standards, as considered by the ’213 Patent, could not be “standard” if they were not well known to a POSA.

165. While the ACNS Guide does not explicitly teach logging only those packets identified by a rule within a dynamic policy, Kjendal does so. Specifically, Kjendal teaches a policy-based system for mirroring packets to monitoring devices 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). Mirroring is a term of art that refers to a device making a copy of packets and sending those copies to another location. Kjendal further teaches a system that selects packets for mirroring based on application-layer packet-header information. EX1009, Cols. 29:60-30:16 (describing first and second criteria for selecting packets based on fields within the header of packets and stating that “[t]he application layer of the packet or series of packets may be used as either or both of the first and second

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

criteria."). Thus. by sending packets to a "logger," as described in Kjendal, the policies determine which packets will be logged and which will not, based on application-layer packet-header information and thus comprising the type of "packet-digest logging function" described in the '213 patent. This sort of rule-based logging was well known to POSAs at the time of invention.

166. A POSA would understand that rule-based logging, as described in Kjendal, would be easily and beneficially added to systems such as ACNS. Such a combination would yield predictable results, in that, rather than logging every HTTP packet, the ACNS could log only packets that met certain criteria. Such a system would also be advantageous because it would give an administrator the flexibility to choose all packets of interest, but save system resources by not logging all packets. Thus, it would be obvious and straightforward to a POSA to combine the rule-based mirroring to a logging system from Kjendal with the general teachings of the ACNS Guide. In my experience, port mirroring was well known before 2014, and such functionality was routinely included in Cisco routing devices, such as the type that would run ACNS or similar software. A POSA would understand that the same criteria that identifies packets that should be monitored may also indicate that those same packets should be logged—for example, because the packet has some indicator that it is a security threat or is objectionable. The ACNS Guide explains, for example, that "[s]ome enterprises have a requirement to monitor, manage, and

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

restrict employee access to nonbusiness and objectionable content on the Internet." EX1008, p. 16-25. In fact, the ACNS Guide already teaches similar functionality and desirability of rule-based logging by providing the option to only log authentication failures as part of its "real-time transaction logging" to a syslog server. EX1008, pp. 19-19 – 19-22. Kjendal also teaches that its port mirroring and other operations may be performed 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.

167. As such, the combination of the ACNS Guide and Kjendal teaches a system that distributes policies that include a packet digest logging function to be performed on packets comprising the application-layer packet-header information.

c) 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;

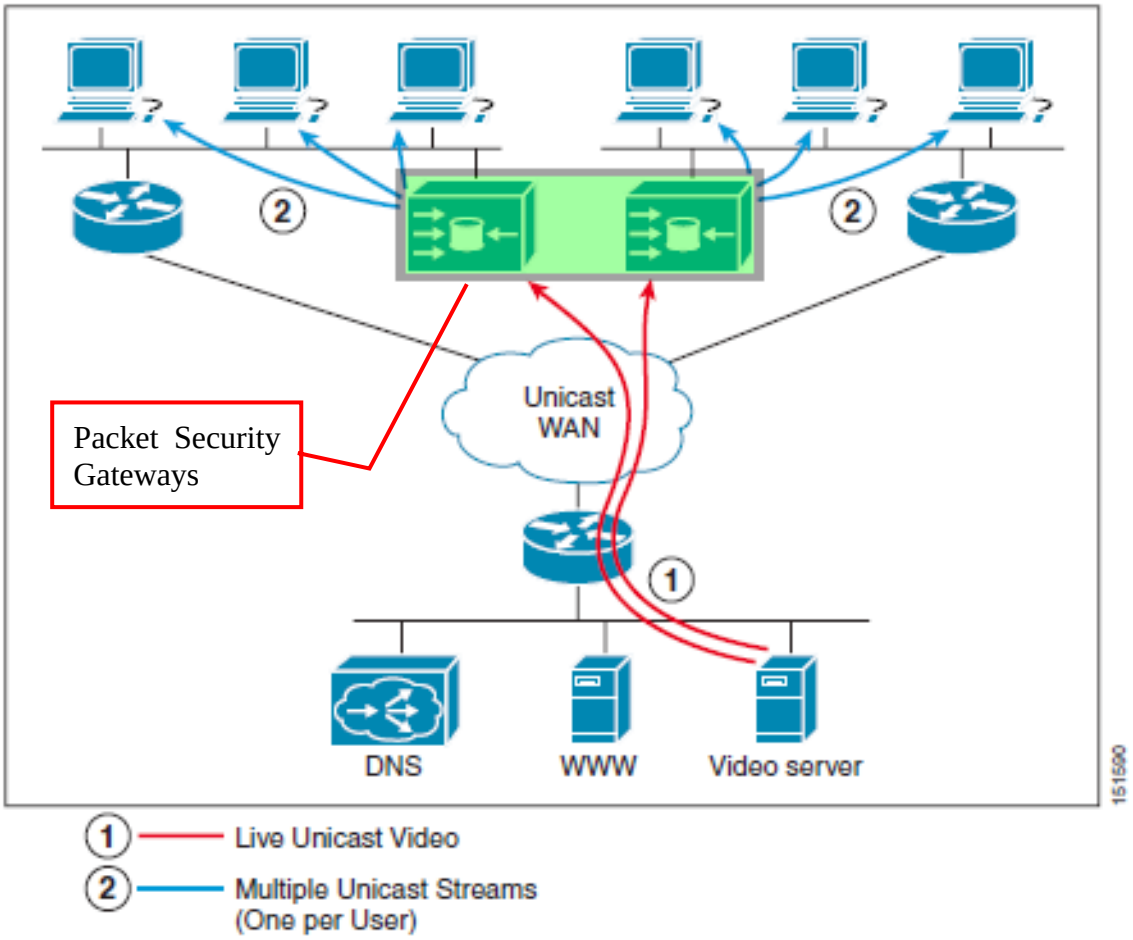
168. The ACNS Guide teaches limitation [1.3] of claim 1.

169. The ACNS Guide teaches that a packet security gateway of a plurality of packet security gateways may receive packets associates with the network protected by the packet security gateway. Specifically, the ACNS Guide shows

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

multiple Content Engines (security gateways, shown in green, below) receiving packets associated with a network behind said Content Engines.

Figure 9-1 Unicast Live Stream Splitting
(Annotated)



EX1008, p. 9-3, Figure 9-1; *see also* p. 9-5, Figure 9-3.

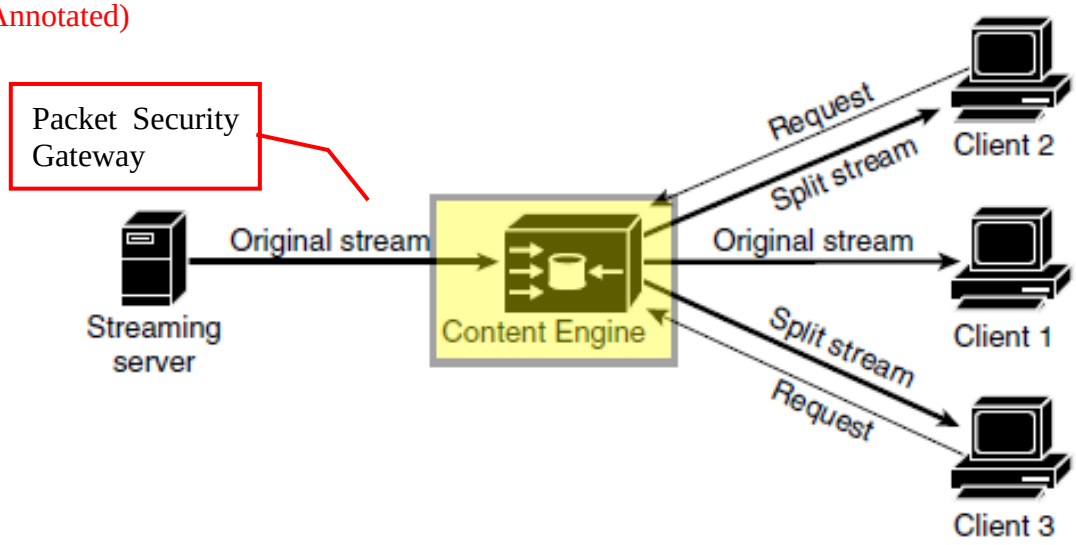
170. The ACNS Guide also describes how the Content Engines may be used as a proxy for a streaming server or broadcast server to replicate content for multiple recipients without each recipient making a request to the internet, thus saving

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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

Figure 9-12 Live Splitting

(Annotated)



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

d) 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;

172. The ACNS Guide teaches limitation [1.4] of claim 1.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

173. The ACNS Guide, for example, teaches that Content Engines may examine the application-layer packet-header of HTTP packets to determine whether the HTTP Request Method is supported by the Content Engine. EX1008, p. 8-23. To do so, the Content Engine must first identify the packet as an HTTP (application-layer) packet and specifically identify the one or more packets which contain the application-layer packet-header information that meets the criteria of having a supported HTTP request method. A POSA would understand that a rule requiring such packet-header information in a packet is needed because, in order to first correctly accept or reject packets based on the included HTTP Request Method (as taught by the ACNS Guide) the system must first identify, on a packet-by-packet basis, each packet comprising such application-layer packet-header information.

174. Another example of the ACNS Guide teaching that the ACNS system identifies packets based on application-layer packet-header information is the Application-specific Access Control Lists (ACLs). EX1008, pp. 17-12 – 17-13 (filtering access based on application-specific parameters related to applications such as SNMP, TFTP, FTP, and WCCP).

175. Therefore, based on the above, the ACNS Guide teaches limitation [1.4].

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

e) 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

176. The ACNS Guide teaches limitation [1.5] of claim 1.

177. The ACNS Guide teaches that the ACNS software may perform, on a packet-by-packet basis, a packet transformation function on each of the one or more packets comprising the application-layer packet-header information. Such packet transformations may include allowing or dropping packets based on the included HTTP Request Method (EX1008, p. 8-23) but may also include a packet digest logging function to notify the Content Engine what information to extract and log from the application-layer packet-header (*id.*, pp. 19-1 – 19-5). The ACNS system must perform packet transformations (e.g., accept/deny and log) on a packet-by-packet basis in order to actually implement its logic. To skip packets would defeat the purpose of this functionality because it would miss packets and potentially allow invalid or unacceptable packets to be forwarded. A POSA, therefore, would understand that all such packet analysis and transformation must be done packet-by-packet.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

178. The ACNS system implantation of ICAP, as described in Chapter 16 (EX1008, pp. 16-1 – 16-66) of the ACNS Guide and described above, for element [1.4] also demonstrates performing a packet transformation. That analysis is incorporated here by reference.

f) 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;

179. The ACNS Guide teaches limitation [1.6] of claim 1.

180. The ACNS Guide teaches that, during the packet transformation, the system must identify a subset of information from the application-layer packet-header information that will be extracted (or digested) from the packet and included in a log file. The ACNS Guide teaches a wide variety of potential information that may be identified, digested from the packets, and logged.

181. The ACNS Guide teaches logging in a variety of formats, including "apache," "extended-squid," "squid," and "custom" logging formats. Each format contains (or may be configured to contain) information that requires identifying a subset of the information in the HTTP application-packet header information, such as "remotehost," "URL," "timestamp," and other information. *See* EX1008, pp. 19-1 – 19-5. For example, a "squid" format log identifies the following information:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

time, remotehost, code/status, bytes, HTTP method, URL, rfc931 (authentication server), peerstatus/peerhost, and type, each separated by spaces and filled in with a "-" if the field is not applicable to a particular log entry. The ACNS Guide provides the following example of a "squid" log entry:

> 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, p. 19-1. A POSA would have known that elements such as the URL and HTTP method ("GET") may only be found in the application-layer packet-header information of an HTTP packet, and that identifying such information is necessary to create a record of it for logging purposes. Moreover, the ACNS Guide's section on custom logging identifies how a user may set up their own logging format, including specifying numerous fields from the HTTP application packet-header information that the ACNS system will use to identify and log packets. EX1008, pp. 19-2 – 19-5. As such, the ACNS Guide teaches a system that distributes policies that include a packet digest logging function that specifies which information needs to be identified from the application-layer packet-header information of the packets, for use with packet digest logging.

g) Element [1.7]

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

[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;

182. The ACNS Guide teaches limitation [1.7] of claim 1.

183. The ACNS Guide teaches generating a record for each of the one or more packets comprising the application-layer packet-header information, the record comprising the subset of information specified by the packet digest logging function described above. Specifically, the ACNS software, operating in Content Engine mode, generates and stores log records containing the identified information. As detailed below, the ACNS Guide offers examples of generated log records. EX1008, pp. 19-1 – 19-3.

184. The ACNS Guide teaches logging in a variety of formats, including "apache," "extended-squid," "squid," and "custom" logging formats. Each format contains (or may be configured to contain) information that requires identifying a subset of the information in the HTTP application-layer packet header information, such as "remotehost," "URL," "timestamp," and other information. *See* EX1008, pp. 19-1 – 19-5. For example, a "squid" format log contains the following information generated based on the information identified from the application-layer packet-header: remotehost, code/status, bytes, HTTP method, URL, rfc931 (authentication server), peerstatus/peerhost, and type. The following diagram of an

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

HTTP packet's structure, including IP, TCP, and Application layers shows (in red) where certain log elements are extracted from the header information of the packets:

<table>
<tr><th colspan="6">IP Layer</th></tr>
<tr><td>Ver. (4 bits)</td><td>IFL (4 bits)</td><td>DSCP (6 bits)</td><td colspan="2">ECN (2 bits)</td><td>Length (16 bits)</td></tr>
<tr><td colspan="6">Other Header Information (64 bits)</td></tr>
<tr><td colspan="6">Source Address (32 bits)</td></tr>
<tr><td colspan="6">Destination Address (32 bits)</td></tr>
<tr><td colspan="6">Options (32 bits)</td></tr>
<tr><th colspan="6">TCP Layer</th></tr>
<tr><td colspan="3">Source Port (16 bits)</td><td colspan="3">Destination Port (16 bits)</td></tr>
<tr><td colspan="6">Other Header Information (160 bits)</td></tr>
<tr><th colspan="6">Application Layer</th></tr>
<tr><td>Request Method (e.g., "GET")</td><td>SP</td><td>URL/URI (e.g., /doc/test.html)</td><td>SP</td><td>HTTP Version (e.g., HTTP/1.1)</td><td>CRLF</td></tr>
<tr><td colspan="5">Field:Value (e.g., remotehost: www.test.com)</td><td>CRLF</td></tr>
<tr><td colspan="5">Field:Value (e.g., Accept: image/gif, image/jpeg, */*)</td><td>CRLF</td></tr>
<tr><td colspan="5">Additional lines of Field:Value pairs…</td><td>CRLF</td></tr>
<tr><td colspan="6">CRLF</td></tr>
<tr><td colspan="6">Entity Body</td></tr>
</table>

185. The ACNS Guide provides the following example of a "squid" log entry:

> 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, p. 19-1. Elements in the log entries such as the URL, HTTP method, and code/status are only logged if they are found in the application-layer packet-header information of an HTTP packet. Further examples of packet digest logging

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

functionality specifying information to be identified by the system may be found in the ACNS Guide's section on custom logging, which teaches how a user may set up their own logging format, including generating records from any of numerous fields from the HTTP application packet-header information. EX1008, pp. 19-2 – 19-5.

186. In my opinion, a POSA would understand that as part of the logging process disclosed in ACNS, the record recited in limitation [1.7] is necessarily generated by the system as an internal form in memory, after the information has been identified and extracted from the application-layer packet-header information, but before that information is saved out to disk or sent to a logging server in the format selected by the user.

187. Alternatively, a POSA may "generate" the record in the ACNS system during the step of saving a subset of packet information as part of the log record in a log format. The ACNS Guide teaches this "generating" element (and the "reformatting" element, below) under either approach.

h) 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;

188. The ACNS Guide teaches limitation [1.8] of claim 1.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

189. The ACNS Guide teaches that the subset of information identified from the application-layer packet-header information may be reformatted into a logging system standard, as defined by the packet digest logging function sent from the Content Distribution Manager. The ACNS Guide teaches the use of such logging system standards as "squid," "extended-squid," and "Apache." EX1008, p. 19-1. Such logging standards were well known in the art before the priority date of the '213 Patent.

190. One of ordinary skill in the art would have known that, under the HTTP 1.1 Specification (see RFC 2616 as of June 1999), HTTP packet headers may include one or more name/value pairs (fields and values) in any order. *See* EX1010, § 4.2, pp 21-22. Thus, reformatting is used to present information from the HTTP header in a consistent manner in a log entry, as the log entries are compiled from packets that may present the header information in different order.

191. Alternatively, a POSA would have seen that the ACNS Guide further teaches reformatting the information specified from within the application-layer packet-header information via its "Real-Time Transaction Logging." EX1008, pp. 19-19 – 19-22. In the ACNS Real-Time Transaction Logging, the Content Engine reformats information from an existing log entry (in any format) into a syslog formatted message to be sent to a syslog server for interpretation and/or storage. Specifically, the ACNS Guide teaches that the previously-generated and formatted

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

log entry files are reformatted as they are appended to a rigidly formatted syslog message, in the following format:

> fac-pri date sending-host sending-process Cisco-standard-syslog-header-with-priority **formatted-log-message**
>
> **Example**: Apr 22 20:10:46 ce-host cache: %CE-TRNSLG-6-460012: **formatted-log-message** (as above)

EX1008, pp. 19-19 – 19-20. Thus, whether a record as disclosed in the ACNS Guide is generated before it is included in a formatted log file, or, as discussed above, a record is not generated until it is saved as part of a log file, the ACNS Guide teaches reformatting the subset of information specified by the packet digest logging function in a log file entry in accordance with a standard format such as the syslog format. This is the same reformatting process described in the '213 Patent. *See* EX1001, Col. 23:14-18.

i) 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.

192. Kjendal teaches limitation [1.9] of claim 1. Kjendal relates to monitoring traffic for analysis by sending copies of certain, identified packets to a monitoring device, based on policies provided to the device that identify certain

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

criteria to be applied to the packets. EX1009, *Abstract*. "The criteria include **what frames of traffic to mirror**, **what portions of the selected frames to mirror**, one or more portals through which to mirror the selected frames, **a destination for the mirroring**…" *Id*. (emphasis added). Specifically, Kjendal supplements ACNS by explicitly teaching mirroring traffic (also known as "port mirroring") based on the results on an analysis of the contents of application-layer packet-header information. *Id*., Col. 30:14-16 ("The application layer of the packet or series of packets may also be used as either or both of the first and second criteria."). To "mirror" port traffic means that a device forwards one copy of each packet toward its listed destination, but also sends an additional copy (the "mirror") to another location – frequently for monitoring of some kind.

193. Kjendal explicitly teaches mirroring traffic from a router to a traffic analyzer or other, specific monitoring device:

> An aspect of the present invention is the **forwarding of one or more frames contained in packets** or one or more portions of frames in packets of a flow or at the initiation of a session to the application identification engine of one or more devices of the network infrastructure **for the purpose of determining the application associated** with the flow/session being established. **That is, the mirroring of the one or more packets or portions of packets to a destination other than the original intended destination**.

EX1009, Col. 7:43-51 (emphasis added). Kjendal also refers to mirroring various portions of packets, including the (application) payload elements of packets:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

> As noted, **the mirror source point 414 may be configured to copy some or all of the traffic associated with the traffic source** 404. A portion of any frame may be configured to be any granularity of the frame of the network traffic. For example, one or more fields of one or more protocol layers of the frame may be configured to be mirrored. In another example, **only the data payload of one of the protocol layers of the frame may be mirrored**.

EX1009, Col 27:48-56 (emphasis added). Kjendal describes the packets forwarded to such a monitoring device as "mirrored." A POSA would understand that mirroring (copying and forwarding) as disclosed in Kjendal includes the same routing function (as that term is described in the '213 Patent), as the Kjendal device must send the mirrored packets to the monitoring device.

194. Kjendal further discloses that packets may be mirrored (routed) to a monitoring device for a variety of reasons and by a variety of methods:

> The dynamic traffic mirroring function of the present invention is not limited to only facilitating the identification of computer applications running on the network. It provides the capability to select options based on a plurality of criteria to mirror any portion or all of the frames of packets received on the network. The dynamic traffic mirroring function can do at least one or more of: 1) mirror all or any portion of the flow, such as the first N packets, the first M fields, the first L Bytes, selected fields and one-way or bidirectional flows; 2) define flows by source address-destination address, Tuple, 5-tuple, **application**, etc.; and 3) **transport the packets** by portals **to generalized or specific destinations**.

EX1009, Col. 8:59-9:3 (emphasis added).

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

195. Kjendal goes on to provide further examples of reasons for mirroring, such IDSs, etc., and makes clear that such mirroring is best accomplished using dynamic policy distribution, rather than manual, static policies:

> This general policy-based mirroring allows network administrators to **set network mirror policies for various network monitors including**, for example, **application identification appliances**, **IDSs, network loggers, analyzers, etc.** These policies may be set based on device, user, topology, time-of-day, network events (e.g., triggers) mirrored to device availability, location and load. Given the breath of dynamic mirroring capabilities, the capabilities exceed administrator manual set-up capabilities. **Rapid dynamic mirrors can better optimize monitor and IDS devices and possibly other network devices** and remove their physical location as a primary factor in their physical placement in the network topology.

EX1009, Col. 10:46-58 (emphasis added).

196. Kjendal also describes that the monitoring device for receiving the packets may be any device or function set up to receive such packets, and offers a number of potential options:

> **The mirrored traffic destination 408 may be any function or device of the network infrastructure 101, including the device 400, arranged to receive and either transfer or analyze the mirrored frames**, including a device having the application identification function 200 described herein. The destination 408 may include a function configurable to receive mirrored and acknowledge network traffic. The mirror destination point 408 may also optionally be part of the network system 100 but as part of a network infrastructure different from the network infrastructure 101. The mirrored traffic destination 408 may include a mechanism to decrypt, de-encapsulate, or un-scramble a received mirrored frame or frames. Further, the destination also contains the necessary protocol elements to provide flow control, frame acknowledgement and such matters as required by packet based transport as selected or used by the portals 412.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

EX1009, Col. 27:32-47. Thus, Kjendal teaches routing packets (via encapsulation) to a monitoring device, exactly as claimed in the claims of the ’213 Patent.

197. A POSA would understand that port-mirroring for monitoring was well-known at the time of the alleged invention, and that such functions were routinely provided on Cisco switches and routers. I personally routinely employed port mirroring as part of the networking research that transpired in my UNC lab in the late 1990s and early 2000s. Therefore, a POSA would very reasonably expect to include monitoring features to improve ACNS by allowing the system to send packets to other devices for analysis, and such a POSA would expect such a combination to yield predictable results (packets would be mirrored to the monitoring device without problems).

198. A POSA would understand that it would be obvious and straightforward to incorporate the specific packet-duplication teachings of Kjendal with the general teachings of the ACNS Guide, including all the ACNS Guide’s teachings of selecting, on a packet-by-packet basis those packets comprising the application-layer packet-header information, such as HTTP packets. A POSA would understand that the logging functions provided by the ACNS provide logging for a wide range of purposes, including logging packets that are also routed to a monitoring device. A POSA would understand that the same criteria that identifies packets that should be monitored may also indicate that those same packets should

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

be logged—for example, because the packet has some indicator that it is a security threat or objectionable. The ACNS Guide 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." EX1008, p. 16-25.

j) Claim 2

[2.0] 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.

199. The ACNS Guide teaches all elements of claim 2. This claim merely recites the extremely common networking functions of accepting selected packets and forwarding the selected packets towards their destination.

200. The ACNS guide teaches a rule for performing an accept-packet transformation and forwarding accepted packets toward their destination. For example, the ACNS Guide teaches that certain HTTP Request Methods, found within the application-layer packet-header, may be accepted and forwarded or denied and dropped. EX1008, p. 8-23. "Content Engines **accept** or reject an HTTP request depending on whether the HTTP request method is supported." (emphasis

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

added). A POSA would understand that "accept[ing]…an HTTP request" indicates that the HTTP packet will be accepted based on application-layer packet-header information and forwarded toward its destination. Thus, if an acceptable HTTP Request Method is found, the ACNS system will (if the content is not locally cached) forward the packet to its destination. EX1008, p. 8-23. Users may, through the Content Distribution Manager, create additional rules related to whether to accept or reject given HTTP Request Methods. *Id*. A POSA would understand that the logging rules of Chapter 19 (pp. 697-718) of the ACNS Guide create log entries whenever the Content Engine performs a task, such as those described above. Therefore, a POSA would understand that the digest-logging and the packet transformation would be part of the same logical function.

201. The ACNS Guide further teaches that, when acting as a caching server, the ACNS software, by rule and depending on the data in the application layer of a request packet, either forwards the packet to its original destination for fulfillment, or to the Content Engine. EX1008, p. 8-48. Specifically, the ACNS Guide states:

> When in transparent caching mode, the Content Engine can intercept requests sent to another proxy and send these requests to one of the following two destinations:
>
> •Default server—This is the default option. The Content Engine retrieves the objects from the web server itself, or if it is configured to use an outgoing proxy for this protocol, it forwards the request to its outgoing proxy. In this scenario, the client browser configuration is ignored, and the Content Engine configuration is used to retrieve the object from the server.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

> •Original proxy—**The Content Engine forwards the request to the proxy to which the client had originally addressed the request**. This proxy may be different from the Content Engine's own outgoing proxy for the specified protocol.

Id. (emphasis added). Note that, in the above, as previously discussed, the term "request" refers specifically to an HTTP application-layer packet. In this way, a POSA would understand that the packet transformation function of the ACNS Guide teaches performing an accept packet function and forwarding the packet toward its destination.

k) Claim 3

[3.0] 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.

202. The ACNS Guide teaches all elements of claim 3. This claim merely recites the extremely common networking function of dropping selected packets.

203. The ACNS guide teaches a rule for performing a deny packet transformation. For example, the ACNS Guide teaches that certain HTTP Request Methods, found within the application-layer packet-header, may be accepted and forwarded or denied and dropped. EX1008, p. 8-23 "Content Engines accept **or**

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

reject an HTTP request depending on whether the HTTP request method is supported." (emphasis added). 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"). Thus, if an unacceptable HTTP Request Method is found, the ACNS system will drop the packet. *Id*. Users may, through the Content Distribution Manager, create additional rules related to whether to accept or reject given HTTP Request Methods. *Id*. As noted above, a POSA would understand that the digest-logging and transformation would be seen as part of the same logical function and applied to the same packets.

204. The ACNS Guide further states that: "[w]hen the Content Engine receives an HTTP request from a client using an unsupported method, ACNS software adds the method to the list of unsupported methods and sends an error to the client." EX1008, p. 8-23. A POSA would understand that, when receiving an unsupported request method, in addition to adding the method to the list of unsupported methods and sending an error to the client, the Content Engine would also deny / drop the packet (because the method was unsupported and hence the request could not be satisfied).

l) Claim 4

[4.0] The method of claim 1, wherein the application-layer packet-header information identifies one or more hypertext transfer protocol (HTTP) packets,

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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.

205. The ACNS Guide teaches all elements of claim 4. This claim merely recites the processing of HTTP packets, which long before 2014 comprised a large portion of network traffic processed at gateway devices.

206. The ACNS Guide teaches a rule for identifying HTTP application-layer packets and performing transformations based on data contained within the application-layer packet-header information of the HTTP packets. *See* EX1008, p. 8-23. Specifically, the ACNS Guide teaches that information found within the application-layer header specifying certain HTTP Request Methods, may be used to accepted and forward the packet or deny and drop the packet. *Id*. ("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.") (emphasis added). If an acceptable HTTP Request Method is found within the application-layer packet-header information, the ACNS system will (if the content is not locally cached) forward the packet to its destination. *Id*. Otherwise, the packet will be dropped. *Id*.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

207. The ACNS Guide states that, by default, the Content Engine allows Request Methods of HTTP 1.1. EX1008, p. 8-23 ("**HTTP 1.1 request methods (for example, GET, HEAD, or POST)** are supported by default.") (emphasis added). Thus, the ACNS Guide teaches identification of HTTP packets and identification of information within the application-layer packet-header information of such packets in order to determine whether to allow such packets based on the HTTP Request Methods contained therein.

208. In yet another example, the ACNS Guide also teaches implementation of a URL / URI filter system, which allows (accepts and forwards) 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. *See* EX1008, pp. 16-25 – 16-33. The ACNS Guide explains that "[s]ome enterprises have a requirement to monitor, manage, and restrict employee access to nonbusiness and objectionable content on the Internet." EX1008, p. 16-25. The ACNS Guide goes on to explain how to implement URL filtering for a variety of protocols, including HTTP. *See* EX1008, pp. 16-25 – 16-33; *see* e.g., 16-28 (listing "Protocol URL Filter Settings for HTTP, RTSP, and WMT"). In this way, the ACNS Guide further teaches identifying HTTP packets and identifying information within the application-layer packet-header of such HTTP packets.

m) Claim 5

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

[5.0] 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.

209. The ACNS Guide teaches every element of claim 5. This claim merely recites processing HTTP packets corresponding to requests for Web content via the HTTP method GET. HTTP GET is one of a limited number of long-standard HTTP functions. *See* EX1010. An HTTP GET request is simply an HTTP packet asking a web server to send the content associated with a given URL. An HTTP GET request packet (and sometimes multiple such GET packets) is sent every time a user, browsing the web, clicks on a link or types in an address into a web browser. A POSA would understand that an HTTP request specifying a GET method is the most common form of HTTP packet.

210. As previously discussed, the ACNS Guide teaches a rule for identifying HTTP application-layer packets and performing rules based on data contained within the packet headers of such HTTP packets, such as the HTTP Request Method. *See* EX1008, p. 8-23.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

211. The ACNS Guide explicitly states that, by default, the Content Engine allows Request Methods of HTTP 1.1. EX1008, p. 8-23 ("**HTTP 1.1 request methods (for example, GET, HEAD, or POST)** are supported by default.")) (emphasis added). Thus, the ACNS Guide teaches identification of HTTP packets and identification of information within the application-layer packet-header information of such packets in order to determine whether to allow such packets based on the HTTP Request Methods contained therein, wherein the packets are associated with the HTTP GET method call. Furthermore, as shown above and via the examples in the ACNS Guide, the logging system necessarily identifies the HTTP packet and the Request Method (including "GET") in order to create a log entry for the packet. EX1008, pp. 19-1 – 19-5. Thus, given that the behavior of the Content Engine changes based on the GET method call, the logging-digest functionality of the ACNS system would depend on the GET method call.

212. A POSA would, based on the ACNS Guide's explicit teachings of default support for HTTP 1.1 requests, understand that the specification for HTTP 1.1 is provided in RFC 2616. *See* EX1010. The HTTP 1.1 Specification includes a total of eight Request Methods, including OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, and CONNECT. *See* EX1010, pp. 33-37. Therefore, a POSA, reading the ACNS Guide would understand it as teaching the identification of packets including the GET Request Method, in addition to the specific mention of

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

the "GET" Request Method in the ACNS Guide. EX1008, p. 8-23 ("HTTP 1.1 request methods (for example, GET, HEAD, or POST) are supported by default.").

n) Claim 6

[6.0] 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.

213. The ACNS Guide teaches each element of claim 6. This claim merely recites a required feature of an HTTP GET packet—the presence of a URI (Uniform Resource Identifier, see Section IV.D)—as criteria for a rule.

214. As described above, the ACNS Guide teaches identifying HTTP packets associated with the HTTP GET method call.

215. In another example, as explained above, the ACNS Guide also teaches implementation of a URL / URI filter system, which allows HTTP methods such as "GET" and "PUT," based on the URI also contained within the header of the HTTP (application layer) packet. *See* EX1008, pp. 16-25 – 16-33. The ACNS Guide explains that "[s]ome enterprises have a requirement to monitor, manage, and restrict employee access to nonbusiness and objectionable content on the Internet." EX1008,

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

p. 16-25. The ACNS Guide explains how to implement URL filtering for a variety of protocols, including HTTP. *See* EX1008, pp. 16-25 – 16-33; *see* e.g., 16-28 (listing "Protocol URL Filter Settings for HTTP, RTSP, and WMT"). In this way, the ACNS Guide further teaches identifying HTTP packets and identifying information, such as a URI/URL within the application-layer packet-header of such HTTP packets.

216. The ACNS Guide 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. EX1008, p. 16-28.

217. A POSA would note that a URL (Uniform Resource Locater) is type of URI. This is spelled out in detail in the specification for URIs, RFC 3986, published in 2005 (see also, Section IV.D). Thus, a POSA would understand that any form of "URL filter" necessarily searches for (and filters on) a URI.

218. The ACNS Guide also provides specific examples of the system logging the Content Engine's interaction with HTTP packets. *See* EX1008, p. 19-1. Specifically, it shows a log entry for a GET request method and contains the URI related to the GET, and confirms that the packet is destined for the network address corresponding to the URI:

> 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** -

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

EX1008, p. 19-1 (emphasis added for GET, URI, and packet target destination (peerhost). A POSA would understand that if the packet was being sent to a destination that did not correspond to the URI, the "peerhost" field in the log entry would be different. Thus, ACNS teaches using application-layer packet-header information associated with the HTTP GET method to determine that the one or more packets originated from, or are destined for, a network address corresponding to the URI.

219. A POSA would understand that in order to filter all HTTP packets carrying specific URLs, as taught by the ACNS Guide, one would filter HTTP packets using the GET Request Method, because the GET Request Method requires such a URL. In any event, a POSA would understand that such filtering would be a trivial combination of filters taught by the ACNS Guide.

o) Claim 7

[7.0] 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.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

220. The ACNS Guide teaches every element of claim 7. This claim merely recites processing HTTP packets corresponding to a long-standard HTTP function. HTTP PUT is one of a limited number of long-standard HTTP functions. *See* EX1010, pp. 33-37. An HTTP PUT request is simply an HTTP packet requesting a web server to create or replace content on the server at the URL specified in the HTTP header with content found within the payload of the packet. A POSA would understand that an HTTP PUT is a commonly used type of HTTP packet.

221. As previously discussed, the ACNS Guide teaches a rule for identifying HTTP application packets and performing rules based on data contained within the packet headers of the HTTP packets, such as the HTTP Request Method. *See* EX1008, p. 8-23.

222. The ACNS Guide explicitly states that, by default, the Content Engine allows Request Methods of HTTP 1.1. EX1008, p. 8-23 ("**HTTP 1.1 request methods (for example, GET, HEAD, or POST)** are supported by default.") (emphasis added). Thus, the ACNS Guide teaches identification of HTTP packets and identification of information within the application-layer packet-header information of such packets in order to determine whether to allow such packets based on the HTTP Request Methods contained therein, based on all the Request Methods disclosed in HTTP 1.1, which includes the PUT method. As discussed above, for claim 5, the ACNS system may also log each interaction between a

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

Content Engine and a packet, therefore, a POSA would understand that the transformation and digest-logging would be viewed as part of the same logical function.

223. As stated above, a POSA would, based on the ACNS Guide's explicit teachings of default support for HTTP 1.1 requests, understand that the specification for HTTP 1.1 is provided in RFC 2616. *See* EX1010. The HTTP 1.1 Specification includes a total of eight Request Methods, including OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, and CONNECT. *See* EX1010, pp. 33-37. Therefore, a POSA, reading the ACNS Guide would understand it as teaching the identification of packets including the PUT Request Method. EX1008, p. 8-23 ("HTTP 1.1 request methods (for example, GET, HEAD, or POST) are supported by default.").

p) Claim 8

[8.0] 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.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

224. The ACNS Guide teaches every element of claim 8. This claim merely recites invoking a common feature of an HTTP PUT packet—source or destination URI addresses—as criteria for a rule. A POSA has long known that source or destination addresses may be indicative of malicious traffic or otherwise indicative of how a packet should be processed, so invoking the address as a rule criterion has long been known to the POSA.

225. As described above, the ACNS Guide teaches identifying HTTP packets associated with the HTTP PUT method call.

226. In another example, as explained above, , the ACNS Guide also teaches implementation of a URL / URI filter system, which allows HTTP methods such as "GET" and "PUT," based on the URI also contained within the header of the HTTP (application layer) packet. *See* EX1008, pp. 16-25 – 16-33. The ACNS Guide explains that "[s]ome enterprises have a requirement to monitor, manage, and restrict employee access to nonbusiness and objectionable content on the Internet." EX1008, p. 16-25. The ACNS Guide explains how to implement URL filtering for a variety of protocols, including HTTP. *See* EX1008, pp. 16-25 – 16-33; *see* e.g., 16-28 (listing "Protocol URL Filter Settings for HTTP, RTSP, and WMT"). In this way, the ACNS Guide further teaches identifying HTTP packets and identifying information, such as a URI/URL within the application-layer packet-header of such HTTP packets.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

227. The ACNS Guide 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. EX1008, p. 16-28.

228. My discussion from Claim 7 is applicable here, too, regarding logging and how the examples of Chapter 19 of the ACNS Guide demonstrate the Request Method, URL, and other data from the HTTP (application-layer) packet are identified and included in a packet digest for logging during the packet transformation process.

229. Also, as discussed above, a POSA would note that a URL is a type of URI.

230. A POSA would understand that, to filter all HTTP packets destined for specific URLs, as taught by the ACNS Guide, one would necessarily filter HTTP packets using the PUT Request Method, because the PUT Request Method requires such a URL to operate. Whether the ACNS Guide filtering for Request Method and URL occurs within a single filter or separate filters, the ACNS Guide teaches a system where two such filters occur in series. A POSA would understand that filtering on both the PUT Request Method and the URL/URI within the HTTP packet, simultaneously would be a trivial variation on the combination of filters already taught by the ACNS Guide and offer predictable results readily implemented in view of the filtering explicitly taught by the ACNS Guide.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

q) Claim 9

[9.0] 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.

231. The ACNS Guide teaches every element of claim 9. This claim merely recites the use of a five-tuple to identify a packet, which has been long known to the POSA.

232. A POSA would recognize the recited network "five-tuple" as the familiar flow id (see Section IV.A) that can be used to filter based on transport-layer protocol, source addresses, source ports, destination addresses, and destination ports. While the ACNS Guide does not use the specific term, as shown below, it uses the five defined elements of the five-tuple.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

233. The ACNS Guide teaches filtering packets based on a number of parameters, including 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 through an Access Control List (ACL). EX1008, Chapter 17, pp. 17-1 – 17-16, *generally*. For example, Table 17-4, shown below, identifies control list filtering based on all of the five-tuple elements, which are highlighted in yellow. *See* EX1008, pp. 17-7 – 17-8. I note that the list includes additional filters, but the claim similarly states that the rule ***"comprises"*** a five-tuple, meaning the rule includes the five tuple but may also include other criteria. Further, as shown by the green highlight, the "operator" and "wildcard" elements in in the table allow a user to specify a "range" of source and destination addresses and ports to use as filtering criteria.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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: • ftp • ftp-data • https • netbios-dgm • netbios-ns • netbios-ss • nfs • rtsp • ssh • telnet • 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.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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: • ftp • ftp-data • https • netbios-dgm • netbios-ns • netbios-ss • nfs • rtsp • ssh • telnet • 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.

234. One of ordinary skill, viewing the ACNS Guide would understand that the Access Control Lists, as defined in Chapter 17, EX1008, pp. 17-1 – 17-16, would be used in conjunction with the previously described application-layer packet-header identification methods, thus meeting the elements of claim 9. *Id.*

r) Conclusion of Ground 1

235. I am of the opinion, therefore, that a POSA would have found that the ACNS Guide, alone or in combination with Kjendal and HTTP specification, renders obvious the subject matter of claims 1-9 of the ’213 Patent.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

B. Ground 2: Claims 10-16 of the ’213 Patent are invalid under 35 U.S.C. § 103(a) on the ground that they are rendered obvious by ACNS in combination with Kjendal and the Diffserv Specification.

236. I am of the opinion that a POSA would have found that the subject matter of the ACNS Guide in combination with Kjendal and the Diffserv Specification renders obvious claims 10-16 of the ’213 Patent.

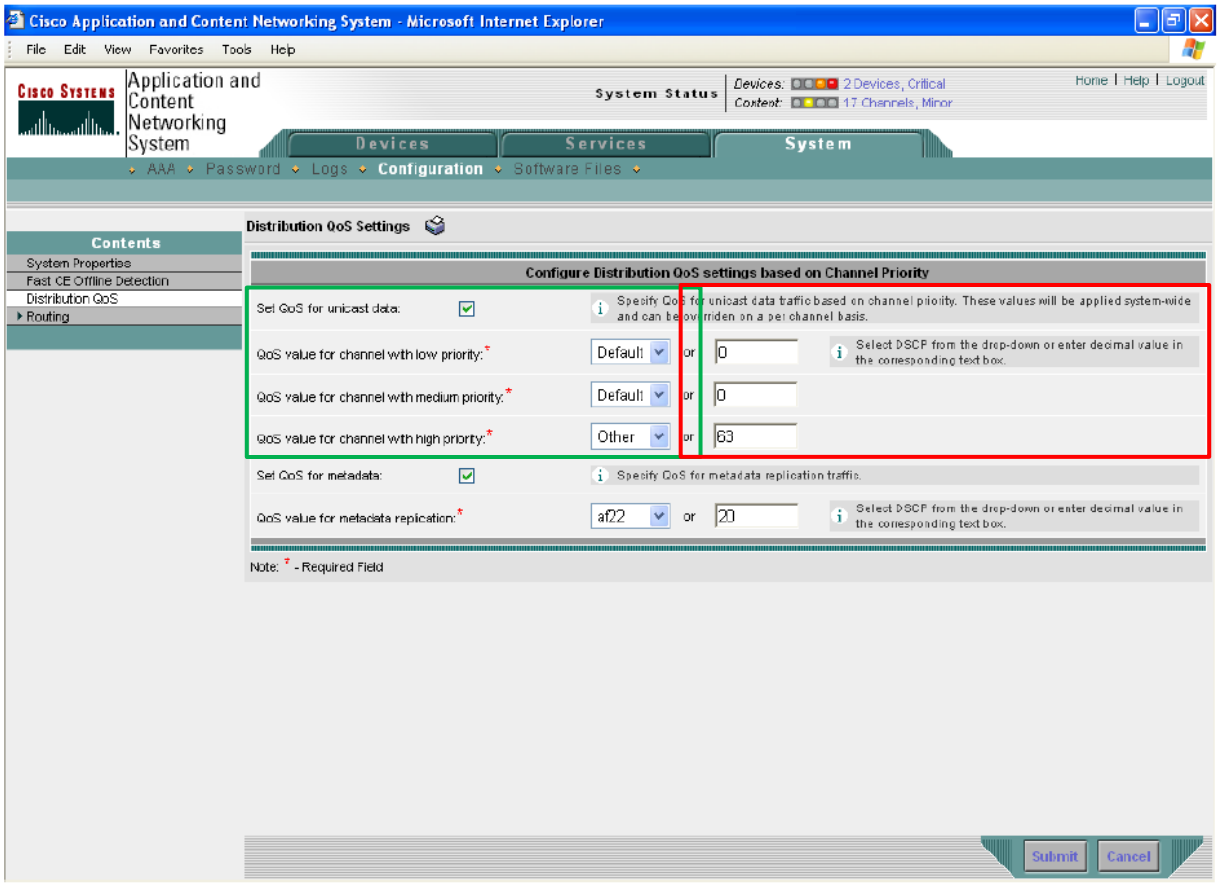
1. Motivation to Combine

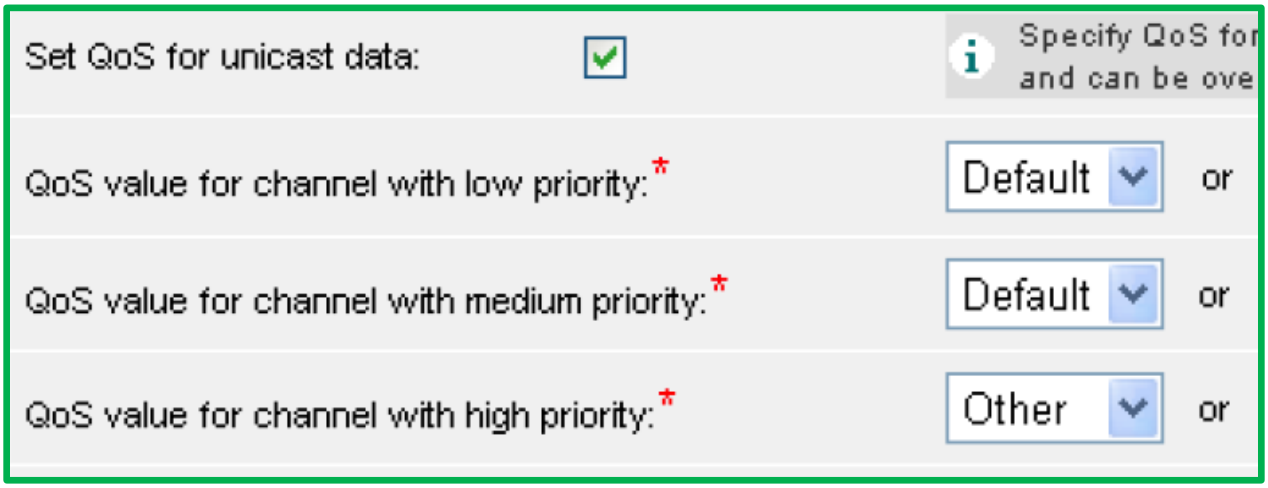
237. Upon reading the ACNS Guide, a POSA would be naturally motivated to consider its teachings in light of the Diffserv Specification.

238. The ACNS Guide teaches that the ACNS System may implement differentiated services as defined in the Diffserv Specification, and especially differentiated services that utilize the Differentiated Services Code Point (DSCP) selector. *See* EX1008, pp. 5-10 – 5-14 (creating a channel, including with DSCP-based quality of service); p. 461 (configuring IP differentiated services using DSCP); pp. 627-631 (table listing rule actions to apply to packets meeting certain criteria, including DSCP modification); and pp. 20-16 – 20-18 (instructions for implementing DSCP-based quality of service rules for the entire system). The following is a screenshot, found in the ACNS Guide, of the differentiated services management screen, with magnified portions provided for clarity:

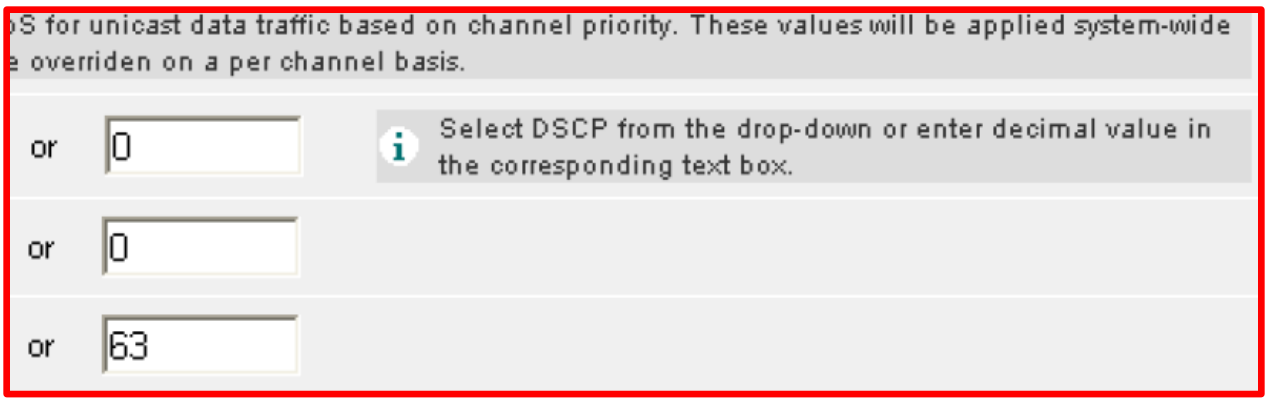
CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

Figure 20-5 Distribution QoS Settings Window





CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512



EX1008, p. 20-17 (elements of GUI magnified for clarity, where green box is left side of configuration screen and red box is right side of configuration screen, showing the use of DSCP within the ACNS Quality of Service configuration). As can be seen, the user may set the QoS (Quality of Service) value (a form of priority) for channels, and, as stated, that value must correspond to the expected DSCP field value in incoming packets. That is, per the Diffserv Specification, 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.

239. The use of the DSCP field was introduced in, and remains tied to, the use of the Differentiated Services Architecture (or "Diffserv") as defined in RFC 2475, published in December, 1998. *See* EX1011. Additional RFCs relating to specific implementations of differentiated services with the DSCP field have been published since December, 1998, but the fundamental definition and architecture have not been revised, since. Thus, a POSA, seeing reference to the DSCP field,

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

would have naturally considered the teachings of the Diffserv Specification at all times relevant to these proceedings.

240. The Diffserv Specification lays out the general architecture for differentiated services, including a detailed specification for the DSCP field (also referred to as the "DS Codepoint" within the Specification). It is designed to be used in network management systems, and thus a POSA would readily understand that it is designed and easily adaptable for systems such as the ACNS system without any undue effort and that its use would yield predictable results.

2. Invalidity Analysis

241. As previously described, the ACNS Guide is a configuration guide for the Cisco Application and Content Networking System, version 5.5, released in 2006. EX1008, p. 1-2; *see also* the above description for claim 1 for a description of ACNS, *generally*.

a) 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

242. The combination of the ACNS Guide and Diffserv Specification teaches limitation [10.1] of claim 10.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

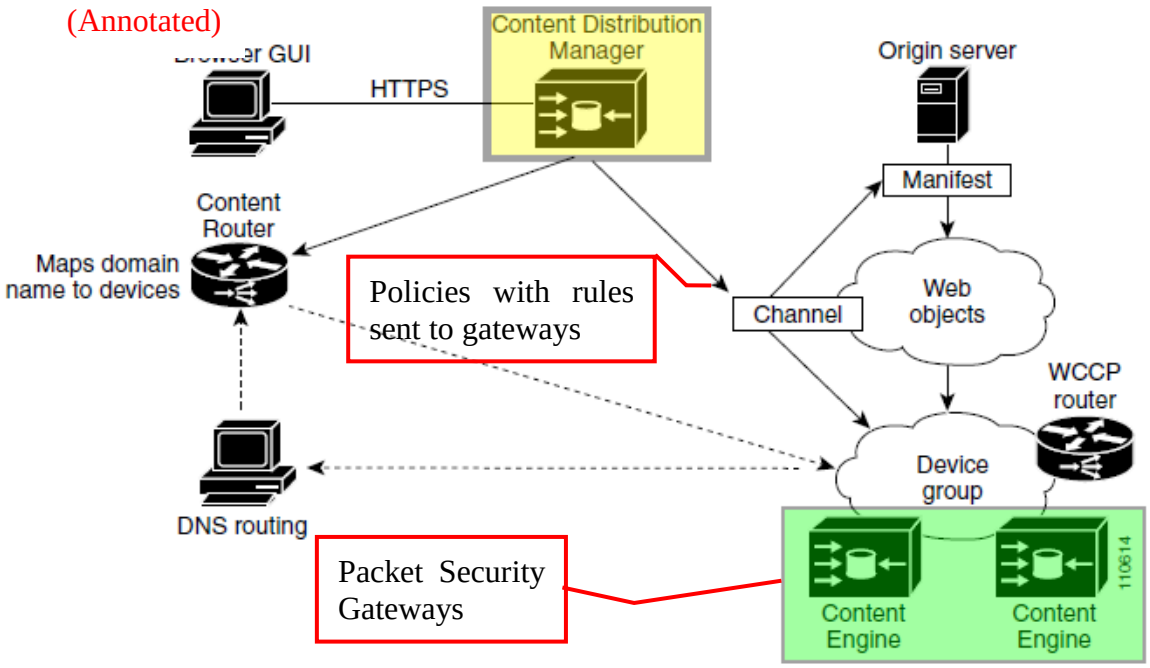
243. This limitation is similar to the initial limitation of claim 1, and my analysis and conclusions are correspondingly similar. The difference between this element and element [1.1] is that claim [10.1] requires a rule specifying a packet-identification criteria rather than application-layer packet-header information. I incorporate my analysis of claim element [1.1] here, and will discuss those parts of claim 10 that are different. However, as an initial matter, I note that the application-layer packet-header information disclosed in ACNS and discussed in the analysis of claim element [1.1] are packet-identification criteria.

244. Within the ACNS system, described in the ACNS Guide, the Content Engines (marked in green, below) represent the claimed security gateways and are associated with a Content Distribution Manager (marked in yellow) and receive dynamic security policies with at least one rule specifying packet identification criteria.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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

(Annotated)



EX1008, Fig. 1-2 (annotated), p 1-11.

245. The dynamic policies, generated at the Content Distribution Manager (CDM), and sent to a plurality of Content Engines, include at least one rule specifying packet-identification criteria. For example, the CDM may propagate a rule allowing or refusing packets with certain HTTP Request Methods. EX1008, pp. 8-23 – 8-24. HTTP packets are packets that must be first identified as such in order to filter the packets based on their contents.

246. The ACNS Guide also teaches filtering packets based on 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 through an Access Control List (ACL), as discussed in connection with claim

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

9. EX1008, Chapter 17, pp. 17-1 – 17-16, *generally*; Table 17-4, pp. 17-7 – 17-8. In this way, the ACNS Guide teaches identifying packets based on packet-identification criteria found in the headers of transport and network layer packets.

247. The Diffserv Specification teaches a means for systems of all kinds to identify packets based, at least in part, on the Differentiated Service Code Point (DSCP) selector field within the network-layer header of the packet. DSCP is an integral part of the Diffserv specification and was created for use in Diffserv implementations. *See* EX1011, § 1.2, pp. 4-5.

248. The Diffserv Specification 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 only" or based on a combination of the DS codepoint and other information found in the packet header. EX1011, § 2.3.1, p. 14. These "classifier" rules qualify as packet identification criteria for the purposes of claim 10 and its dependent claims.

249. The ACNS Guide teaches the use of the Diffserv mechanisms and specifically the use of the DSCP field. *See* EX1008, pp. 5-10 – 5-14 (creating a channel, including with DSCP-based quality-of-service); pp. 10-9 (configuring IP differentiated services using DSCP); pp. 16-19 – 16-23 (table listing rule actions to apply to packets meeting certain criteria, including DSCP modification); and pp. 20-16 – 20-18 (instructions for implementing DSCP-based quality-of-service rules for

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

the entire system). Given that all DSCP-based quality-of-service implementations stem from the Diffserv Specification, it would be obvious to one of ordinary skill in the art to combine the specific teachings of the Diffserv Specification with the system disclosed by the ACNS Guide because the ACNS system specifically contemplates such a combination.

b) 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,

250. The combination of the ACNS Guide and Kjendal teaches limitation [10.2] of claim 10. This limitation is similar to language in element [1.2] of claim 1, and my analysis and conclusions are correspondingly similar. The only difference is applying the packet transformation to packets identified by the packet-identification criteria, rather than to packets identified based on specific application-layer packet header information. However, I note that the application-layer packet-header information disclosed in ACNS and discussed in the analysis of claim element [1.2] are packet-identification criteria.

251. The Content Distribution Manager may instruct a plurality of Content Engines whether to log transactions and how to log those transactions via policies. EX1008, Chapter 19, pp. 19-1 – 19-22, *generally*. The Content Distribution

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

Manager provides a function for logging information and such logging information comprises information from packets identified via packet-identification criteria, such as HTTP packet header information. *Id*., pp. 19-10 – 19-11; *see also* section above relating to claim element [1.2] for specific details regarding digest-logging of identified packet information found in both the ACNS Guide and Kjendal.

252. The combination of the ACNS Guide and Kjendal teaches a system that distributes policies that include a packet digest logging function to be performed on packets identified via criteria.

c) Element [10.3]

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

253. The combination of the ACNS Guide and the Diffserv Specification teaches limitation [10.3] of claim 10. This limitation is not found in claim 1 or its dependent claims and is addressed fully below.

254. The ACNS Guide teaches the use of Diffserv mechanisms and specifically the use of the DSCP field. *See* EX1008, pp. 5-10 – 5-14 (creating a channel, including with DSCP-based quality-of-service); pp. 10-9 (configuring IP differentiated services using DSCP); pp. 16-19 – 16-23 (table listing rule actions to apply to packets meeting certain criteria, including DSCP modification); and pp. 20-16 – 20-18 (instructions for implementing DSCP-based quality-of-service rules for

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

the entire system). Given that all DSCP quality-of-service implementations stem from the Diffserv RFC, it would be obvious to one of ordinary skill in the art to combine the specific teachings of the Diffserv RFC with the system disclosed by the ACNS Guide at least because the ACNS system specifically teaches and contemplates such a combination.

255. The Diffserv Specification teaches a means for systems to identify packets based on the Differentiated Service Code Point (DSCP) selector field within a given packet. The DSCP field is an integral part of the Diffserv specification and was created for use in implementations of the Diffserv architecture. *See* EX1011, § 1.2, pp. 4-5. In particular, the DSCP field exists to enable network devices to identify and classify packets.

256. The Diffserv RFC 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 only" or based on a combination of the DS codepoint and other information found in the packet header. EX1011, § 2.3.1, pp. 14-15. These "classifier" rules use the DSCP field and qualify as packet identification criteria for the purposes of claim 10 and its dependent claims. These classifier rules are also nearly identical to the various criteria for selecting and filtering packets described in the ACNS Guide and, because of the ACNS Guide's explicit implementation of the Diffserv architecture, it would be

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

obvious to a POSA to combine these references to select or filter packets based, at least in part, on the DSCP field.

d) 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;

257. The combination of the ACNS Guide and Diffserv Specification teaches limitation [10.4] of claim 10. This claim element is identical to element [1.3] of claim 1, and my analysis of that element is incorporated here.

258. The Diffserv Specification also teaches receiving packets at a security gateway called a "Differentiated Service Node" or "DS Node" that include a "packet classifier" for identifying packets with DSCP fields. EX1011, §§ 1.2; 2, pp. 4-5, 12-16, *generally*. Specifically, the Diffserv Specification teaches that a classifier (within a DS Node) may "select packets in a traffic stream based on the content of some portion of the packet header." *Id*., § 2.3.1. The Diffserv Specification also teaches that nodes receive traffic (packets) on either the inbound or outbound boundary of a network: "Traffic streams may be classified, marked, and otherwise conditioned on either end of a boundary link (the DS egress node of the upstream domain or the DS ingress node of the downstream domain)." *Id*., § 2.3.4.2, p. 18;

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

see also § 6.1, pp. 28-29 (theft and denial of service explanation referring to DS ingress and interior nodes receiving and classifying packets).

259. A POSA would be motivated to combine the teachings of the Diffserv Specification with ACNS because of ACNS's specific reliance on the Diffserv architecture as disclosed in the Difffserv Specification.

260. For the same reasons listed above and in claim element [1.3] the combination of the ACNS Guide and Diffserv Specification teaches receiving by a packet security gateway, of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway.

e) 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;

261. The combination of the ACNS Guide and Diffserv Specification teaches limitation [10.5] of claim 10. The claim element, and my analysis, is similar to that found for element [1.4] of claim 1. The only difference is that, rather than the specific requirement of identifying packets based on application-layer packet-header information, claim 10 requires only identifying packets that correspond to the packet-identification criteria (that includes the DSCP field). My analysis as it relates

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

to the ACNS is the same as for element [1.4] and is incorporated here, by reference, since identifying packets based on application-layer packet-header information is a form of identifying packets based on packet identification criteria.

262. The ACNS Guide teaches a system that identifies, on a packet-by-packet basis from amongst the packets associated with the network protected by the packet security gateway, one or more packets comprising the identified packet information.

263. Specifically, a Content Engine may identify packets comprising a DSCP field in the packet header information. To do so, the ACNS Guide teaches the implementation of the Diffserv protocol, and specifically the use of the DSCP field. *See* EX1008, pp. 5-10 – 5-14 (creating a channel, including with DSCP-based quality-of-service); p. 10-9 (configuring IP differentiated services using DSCP); pp. 16-19 – 16-23 (table listing rule actions to apply to packets meeting certain criteria, including DSCP modification); and pp. 20-16 – 20-18 (instructions for implementing DSCP-based quality-of-service rules for the entire system). Given that the use of the DSCP field to specify or indicate quality-of-service levels or classes stems from the Diffserv Specification, it would be obvious to one of ordinary skill in the art to combine the specific teachings of the Diffserv Specification with the system disclosed by the ACNS Guide because the ACNS system specifically contemplates such a combination.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

264. The Diffserv Specification teaches identifying packets based on the packet-identification criteria, using, at least in part, DSCP (or "codepoint") values. Such identification is taught based on the use of "packet classifiers." EX1011, § 2.3.1, pp. 14-15. The Diffserv Specification states: "Packet classifiers **select packets** in a traffic stream **based on the content of some portion of the packet header**. We define two types of classifiers. The **BA (Behavior Aggregate) Classifier classifies packets based on the DS codepoint only**. **The MF (Multi-Field) classifier selects packets based on the value of a combination of one or more header fields**, such as source address, destination address, **DS field**, protocol ID, source port and destination port numbers, **and other information** such as incoming interface." *Id*., § 2.3.1, p. 14 (emphasis added).

265. Given the teachings of the ACNS Guide and the Diffserv Specification, a POSA would find it obvious to identify packets based, in part, on the DSCP field.

f) 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:

266. The combination of the ACNS Guide and Diffserv Specification teaches limitation [10.6] of claim 10.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

267. The ACNS Guide teaches that the ACNS software may perform, on a packet-by-packet basis, a packet digest logging function to notify the Content Engine what information to extract and log from the identified packet. EX1008, pp. 19-1 – 19-5. The ACNS system must necessarily perform packet transformations (accept/deny and log) on a packet-by-packet basis in order to actually implement its logic. To skip packets would defeat the purpose of this functionality because it would miss packets and potentially allow invalid or unacceptable packets to be forwarded. A POSA, therefore, would understand that all such packet analysis, transformation, and packet-digest logging must be done packet-by-packet. Further details about the ACNS Guide's logging teachings are provided in connection with the above analysis regarding Claim 1.

268. The ACNS Guide teaches that each packet interaction with the Content Engine may be logged. *See* EX1008, Chapter 19, pp. 19-1 – 19-22, *generally*. Thus, ACNS teaches performing a patent digest logging function on the one or more identified packets.

269. The Diffserv Specification teaches that a system should necessarily log certain types of events ("auditable" events) during the packet transformation process and identifies the specific packet-header information that should be gathered for such a log record: "If such a [redundant traffic] check fails because the upstream domain is not fulfilling its responsibilities, **that failure is an auditable event; the**

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

generated audit log entry should include the date/time the packet was received, the source and destination IP addresses, and the DS codepoint that caused the failure." EX1011, § 6.1 p. 29 (emphasis added); *see also* "Any detected failure of such a check [of an interior node for traffic conditioning] is an auditable event and the generated audit log entry should include the date/time the packet was received, the source and destination IP addresses, and the DS codepoint that caused the failure." *Id.*

270. The Diffserv Specification goes on to describe audit logging in general terms such that one of ordinary skill in the art would understand to seek out additional resources for a specific implementation of such a logging system: "For the most part, the granularity of auditing is a local matter. However, several auditable events are identified in this document and **for each of these events a minimum set of information that should be included in an audit log is defined**. Additional information (e.g., packets related to the one that triggered the auditable event) may also be included in the audit log for each of these events…" *Id.*, § 6.3 p. 32 (emphasis added).

271. Such additional information for logging, as contemplated by the Diffserv Specification, may be found in resources such as the ACNS Guide or other art, and would be readily available and obvious to a POSA at the time of alleged invention.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

g) 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;

272. The ACNS Guide teaches limitation [10.7] of claim 10.

273. The ACNS Guide teaches that, during the packet digest logging function, the system must identify a subset of information from the identified packet that will be extracted from the packet and included in a log file. The ACNS Guide teaches a wide variety of potential information that may be specified and subsequently logged. *See* EX1008, pp. 19-1 – 19-5.

274. As discussed in connection with Claim 1, the ACNS Guide teaches logging in a variety of formats, including "apache," "extended-squid," "squid," and "custom" logging formats. Each format requires (or may be configured to require) identifying a subset of information from the HTTP application-packet header information, such as "remotehost," "URL," "timestamp," and other information. *See* EX1008, pp. 19-1 – 19-5. For example, a "squid" format log identifies the following information: time, remotehost, code/status, bytes, HTTP method, URL, rfc931 (authentication server), peerstatus/peerhost, and type, each separated by spaces and filled in with a "-" if the field is not applicable to a particular log entry. The ACNS Guide provides the following example of a "squid" log entry:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

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, p. 19-1. A POSA would have known that elements such as the URL and HTTP method may only be found in the packet-header information of an identified HTTP packet, and that identifying such information is necessary to creating a record of it for logging purposes. Moreover, the ACNS Guide's section on custom logging, which describes how a user may set up their own logging format, includes examples of numerous fields from an identified packet that may be logged. EX1008, pp. 19-2 – 19-5. As shown above, regarding element [1.7], the combined TCP/IP/HTTP packet includes the fields necessary to create the Squid format log, above, and it also contains the DSCP field. Thus, a POSA would be motivated to include that field in logging if it were also used in identifying packets for transformation. As such, the ACNS Guide teaches a system that distributes policies that include a packet digest logging function that specifies which information needs to be identified from the packets identified by a packet-identification.

275. As discussed above, a POSA would know that additional elements for packet identification, as discussed for the ACNS Guide, would include the DSCP field, as taught by the Diffserv Specification. Such a POSA would also know that, in a system where packets were identified for transformation by the DSCP field, a user would likely wish to include the value of the DSCP field within the packet

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

digest log. Such a method is explicitly taught by the Diffserv Specification. For example, the Diffserv Specification teaches that a system should necessarily log certain types of events and states that "**the generated audit log entry should include the date/time the packet was received, the source and destination IP addresses, and the DS codepoint that caused the failure**." EX1011, § 6.1, p. 29 (emphasis added). This reference, combined with the general packet-digest logging functionality taught by the ACNS Guide teaches all elements of element [10.7].

276. Beyond the specific teaching and suggestion to combine found within the ACNS Guide, combining the teachings of the ACNS Guide and the Diffserv Specification would represent use of a known technique (DSCP codepoint filtering/classification) to improve similar methods (filtering/logging in ACNS). It would also represent a known method the use of which would yield predictable results.

h) 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; and

277. The ACNS Guide teaches limitation [10.8] of claim 10. This claim is functionally the same as limitation [1.7] of claim 1, except that the identified packet of claim 10 need not have application-layer packet information, as required in claim

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

1, but instead requires identification using the DSCP field, as required in element [10.3]. Regardless of the difference in how the packet was identified, the generated record of information need not be different. For that reason, my explanation for limitation [1.7] is incorporated here by reference and for the reasons therein, the ACNS Guide teaches all elements of claim element [10.8].

278. As detailed above, the Diffserv Specification also teaches that records should be generated for logging certain information, based on certain identified packets. A POSA would understand that such logging necessarily involves generating a record. At the time of alleged invention, a POSA would also combine the ACNS Guide and the Diffserv Specification with respect to generating a record for the reasons listed above. Such a combination would yield predictable results and would simply combine the references as both the ACNS Guide and the Diffserv Specification indicate they should be.

i) Element [10.9]

[10.9] 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; and

279. The ACNS Guide teaches limitation [10.9] of claim 10. This element is functionally identical to claim element [1.8] of claim 1, except for how the logged packet was identified. That difference has no effect on how the information should

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

be reformatted. Thus, I incorporate my explanation from element [1.8] here by reference. For the reasons stated therein, the ACNS Guide teaches all elements of limitation [10.9].

j) Element [10.10]

[10.10] 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.

280. Kjendal teaches element [10.10] of claim 10. This claim element is very similar to element [1.9] of claim 1, and my analysis therein is incorporated by reference. The only differences are the identification of the packet for routing (but that should not affect the routing, itself) and that the routing comes in response to performing a packet digest logging function, rather than a more generic packet transformation function (that comprises a packet digest logging function) as claimed in claim 1.

281. In the way described above, Kjendal teaches limitation [10.10] of claim 10. I incorporate by reference my discussion of Kjendal, found in element [1.9] of claim 1, as it applies equally to this element.

k) Claim 11

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

[11.0] 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.

282. Either one of the ACNS Guide or the Diffserv Specification teaches every additional element of claim 11. Claim 11 simply lists well-known pieces of information commonly found in packet headers since well before 2014 as possible logged information. Prior art that teaches logging any one of these pieces of information would support the conclusion that the claim is obvious.

283. The ACNS Guide teaches a system wherein performing a packet digest logging function comprises identifying a subset of information specified by the packet digest logging that 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

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

packet security gateway that received the packet, or one or more media access control (MAC) addresses associated with the packet.

284. Specifically, the ACNS Guide teaches that the ACNS software, running in Content Engine mode, has a robust logging system that may be enabled to capture information about all transactions. Specifically, the Content Engine may identify specific information from various identified packets: "Transaction logs allow administrators to view the traffic that has passed through the Content Engine. Typical fields in the transaction log are **the date and time when a request was made, the URL that was requested**, whether it was a cache hit or a cache miss, the type of request, **the number of bytes transferred, and the source IP address**." EX1008, p. 19-1 (emphasis added to show claimed information referred to by logging).

285. When using Custom Logging, the Content Distribution Manager (security policy management server) may specify which information to identify from within each packet. For example, the Content Engine may identify:

- Requesting IP Address
- Transport protocol
- URL (URI) path requested
- Size, in bytes, of the request
- Arrival time of the packet, when it was logged

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

EX1008, pp. 19-3 – 19-7. Use of any one of these features teaches this limitation of claim 11. I note that claim 11 indicates the subset "comprises" the listed information, and thus the subset of information may include more information than recited in the claim. Therefore, the ACNS Guide teaches the limitations of claim 11.

286. The Diffserv Specification also discloses this limitation. It teaches that a system should generate log records for certain types of events ("auditable" events) during the packet transformation process and identifies the specific packet-header information that should be gathered for such a log record: "… the generated audit log entry should include **the date/time the packet was received, the source and destination IP addresses,** and the DS codepoint that caused the failure." EX1011, § 6.1, p. 29 (emphasis added). A POSA would be motivated to combine the ACNS Guide with the Diffserv Specification because the Diffserv Specification refers only generally to logging and does not provide details for how a POSA would perform the task. Therefore, a POSA would necessarily look to a specific implementation of logging to identify the information and generate the record. The ACNS Guide is such an implementation. Combining its teachings with the Diffserv Specification would readily improve the teachings of related systems and would yield predictable results.

l) Claim 12

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

[12.0] 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.

287. The ACNS Guide teaches all elements of claim 12. Claim 12 is directed to one of the limited number of options available to a POSA to generate and store a record of information to be logged and would have been an obvious approach to logging.

288. The ACNS Guide teaches temporarily storing the subset of logged information at the Content Engine and generating a message with said information identified from packets corresponding to the packet-identification criteria.

289. Specifically, the ACNS Guide teaches using "real time transaction logging" to generate logs of packet transactions and generate messages to send to a remote syslog server. EX1008, pp. 19-19 – 19-22. To do so, the Content Engine generates a message in the syslog format. Specifically: "**The following is an example of the format of the real-time syslog message sent from the transaction logging module (Content Engine) to the remote syslog host**:

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

fac-pri Apr 22 20:10:46 ce-host cache: %CE-TRNSLG-6-460012: *translog formatted msg*"

EX1008, p. 19-19 (bold emphasis added). In the preceding example, the "*translog formatted msg*" contains the logfile generated as described above for claim 10, in one of several standard or user-definable formats. *Id*. The ACNS Guide specifies that this message is generated (and thus stored in a memory) at the Content Engine and then sent to a "remote syslog server." *Id*. Preparing such a message would be an obvious step (as explained in the ACNS Guide) because remote logging allows an administrator to monitor multiple packet security gateways for errors in real time. EX1008, p. 19-19 ("By sending HTTP transaction log messages to a remote syslog server, you can monitor the remote syslog server for HTTP request authentication failures in real-time."). Storing such a message in memory is an obvious step because a computer must store any message in some form of memory before either sending or permanently saving it.

m) Claim 13

[13.0] 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.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

290. The ACNS Guide teaches all additional elements of claim 13. This claim recites one of a limited number of options available and known to a POSA at the time of invention for using information to be logged.

291. The ACNS Guide teaches sending a message from the Content Engine to the security policy management server with said information identified from packets corresponding to the packet-identification criteria.

292. Specifically, the ACNS Guide teaches using "real time transaction logging" to send logs of transactions to a remote syslog server. EX1008, pp. 19-19 – 19-22. To do so, the Content Engine sends a message to a syslog server in syslog format. Specifically, as described in the ACNS Guide: "The following is an example of the format of the real-time syslog message **sent from the transaction logging module (Content Engine) to the remote syslog host**:

> *fac-pri* Apr 22 20:10:46 ce-host cache: %CE-TRNSLG-6-460012: *translog formatted msg*"

EX1008, p. 19-19 (bold emphasis added).

293. The ACNS Guide provides detailed instructions for how to configure the location to which the syslog messages should be sent. EX1008, pp. 19-20 – 19-20. One of ordinary skill, viewing those instructions, would understand that among the limited number of options available, a likely location for the target syslog server would be the same machine that houses the Content Distribution Manager (CDM)

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

(acting as the security policy management server). In fact, such a location would be preferable, as it would allow an administrator to easily and efficiently search said logfile entries. The CDM is one of a limited number of viable options for locating the syslog server for a POSA in the ACNS system. Thus, the ACNS Guide teaches or suggests the limitations of claim 13.

n) Claim 14

[14.0] 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.

294. The ACNS Guide teaches all additional limitations of claim 14. This claim merely recites that the logging should be done in the syslog standard, which as the '213 specification acknowledges, was a known standard in 2012 for precisely the purpose of logging information about packets that meet certain criteria. *See* EX1001, Col. 11:54-60.

295. The ACNS Guide teaches formatting the subset of information in the syslog standard format.

296. Specifically, the ACNS Guide teaches using "real time transaction logging" to reformat logs of packet transactions in the syslog format and send them

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

to a remote server. EX1008, pp. 19-19 – 19-22. Specifically, as described in the ACNS Guide: "The following is an example of the format of the real-time **syslog** message sent from the transaction logging module (Content Engine) to the **remote syslog host**:

> *fac-pri* Apr 22 20:10:46 ce-host cache: %CE-TRNSLG-6-460012: *translog formatted msg*"

EX1008, p. 19-19 (bold emphasis added). In the preceding example, the "*translog formatted msg*" contains the logfile generated as described above in claim 10, in one of several standard or user-definable formats. *Id*. In this way, and as shown in the provided example, the ACNS Guide teaches reformatting packet information in the syslog logging system format. As noted above, syslog is a standard from the 1980s and was well known at the time of alleged invention. Within the specification of the '213 Patent, it refers to storing or forwarding packet logs "using a logging system based on a standard (e.g., syslog, or the like). EX1001, Col. 11:54-57. Syslog could not be a "standard" as required by the '213 Patent unless it was previously known. Thus, Patent Owner even admits that reformatting records into syslog format was well known.

o) Claim 15

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

[15.0] 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.

297. The ACNS Guide teaches every element of claim 15. Claim 15 is functionally identical to claim 9, except for the identified packet in claim 9 necessarily needing to comprise application-layer packet-header information and the packet of claim 15 not needing such information, but necessarily including DSCP information. For that reason, the ACNS Guide suggests identifying packets based on a five-tuple as required by claim 15 for all the reasons listed above for claim 9. That logic and analysis is incorporated here by reference.

p) Claim 16

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

[16.0] 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.

298. The ACNS Guide teaches all elements of claim 16. This claim simply extends the identification criteria for a packet to the application layer. Such identification of packets based on application-layer information in addition to other criteria was well known by 2014.

299. The ACNS Guide teaches packet-identification criteria that comprises application-layer packet-header information and wherein identifying each of the one or more packets comprises determining by the Content Engine (packet security gateway) that each of the one or more packets comprise at least a portion of the application-layer packet-header information. Such a method would include identifying an HTTP packet based on application-layer packet-header information such as Request Method or URL and also based on the DSCP field from the network layer packet header information.

300. For example, the ACNS guide teaches identifying packets based on application-level packet header information such as HTTP header information by

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

the Content Engines to insure that such application packets conform to policies and rules. HTTP Request Methods correspond to information found within the application-layer packet-header and the rules defined to the Content Engine by the Content Distribution Manager comprise such information: "**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. Nonstandard request methods, such as Web-Based Distributed Authoring and Versioning (WebDAV) are not." EX1008, p. 8-23 (emphasis added).

301. Based on the above, it would be obvious to a POSA to include application-layer packet-header information in the packet identification criteria and to include application-layer packet-header information when identifying packets, as claimed in claim 10.

q) Conclusion of Ground 2

302. I am of the opinion, therefore, that a POSA would have found that the ACNS Guide, combined with the teachings of the Diffserv Specification, and potentially in light of Kjendal, renders obvious the subject matter of claims 10-16 of the '213 Patent.

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

IX. SECONDARY CONSIDERATIONS

303. I am not aware of any secondary considerations that would make claims 1-16 of the ’213 Patent non-obvious over the prior art considered herein. In my opinion, any possible secondary considerations would not overcome the prior art that clearly demonstrates that the subject matter of claims 1-16 of the ’213 Patent would have been obvious to a POSA as of April 16, 2014.

304. I reserve the right to supplement my opinions as appropriate if Centripetal Networks, Inc. alleges or offers any purported evidence of any such secondary considerations of non-obviousness.

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

Executed this 15th day of August, 2018.

Respectfully submitted,

Dr. Kevin Jeffay

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512

CISCO EXHIBIT 1004
Jeffay Declaration
IPR2018-01512