



- (51) **International Patent Classification:**
H04L 12/70 (2013.01) *H04L 29/06* (2006.01)
- (21) **International Application Number:**
PCT/US2015/011008
- (22) **International Filing Date:**
12 January 2015 (12.01.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** NGUYEN, Ted T.; Hewlett-Packard Company, 8000 Foothills Blvd., Roseville, California 95747 (US). SAWYER, Michael Anthony; Hewlett-Packard Company, 8000 Foothills Blvd, Roseville, California 95747 (US). VENTURA, Patrick; Hewlett-Packard Company, 8000 Foothills Blvd., Roseville, California 95747 (US).
- (74) **Agents:** POJUNAS, Scott A. et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79 , Fort Collins, CO 80528 (US).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) **Title:** DATA LINK LAYER INFORMATION

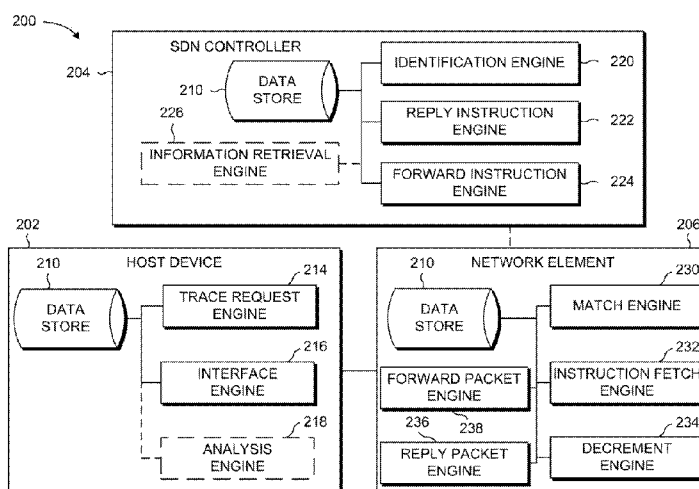


FIG. 2

(57) **Abstract:** In one implementation, a controller coupled to a network includes logic to instruct a first network element to send a reply packet with data link layer information to a source in response to a receipt of a first probe packet having a first trace value at the first network element and instruct the first network element to send a second probe packet to a second network element with a second trace value.

Data Link Layer Information

BACKGROUND

[0001] Devices, such as computers, commonly communicate across network infrastructure, where network elements provide a path of communication from one device to another. For example, the network infrastructure can include switches and routers as connection points in a path from one host device to another host device. Services related to the computing industry can be offered over a network of devices to route the service from one destination to another. Service performance is commonly reviewed for quality of service and service fulfillment purposes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figure 1 depicts an example environment in which various systems for retrieving data link layer information can be implemented.

[0003] Figures 2 and 3 are block diagrams depicting example systems for retrieving data link layer information.

[0004] Figure 4 is an example sequence diagram depicting operations of example systems for retrieving data link layer information.

[0005] Figures 5-8 are flow diagrams depicting example methods for retrieving data link layer information.

DETAILED DESCRIPTION

[0006] In the following description and figures, some example implementations of apparatus, systems, and/or methods for retrieving data link layer information are described. Data link layer information can be pertinent to network performance and service performance. Some examples herein are described specifically for use in a software-defined network ("SDN") environment. SDN-compatible networks may provide

a service or multiple services to devices or other networks. As used herein, a service may be any appropriate supplying of communication, transmissions, software, storage, or any other product or activity that may be capable of running on a network of devices, or network elements. SDN-compatible networks abstract the hardware of the system from the services being provided. For example, an SDN network can decouple the traffic control decisions from the physical network elements that forward the network traffic. In a network using SDN, the information of how the service gets to the customer is abstracted and a system administrator would be unaware as to what port or element may have an error or problem and be in need of repair or replacement.

[0007] Various examples described below relate to identifying data link layer information from each network element in a network. By utilizing instructions from a network controller, a network element can receive a packet identifying a request for data link layer information, decrement the trace value of the packet, send a reply to the source making the request, and send the packet to the next network element in a path to a destination device from the source device.

[0008] The terms "include," "have," and variations thereof, as used herein, mean the same as the term "comprise" or appropriate variation thereof. Furthermore, the term "based on," as used herein, means "based at least in part on." Thus, a feature that is described as based on some stimulus can be based only on the stimulus or a combination of stimuli including the stimulus.

[0009] Figure 1 depicts an example environment 190 in which various systems for retrieving data link layer information can be implemented. The example environment 190 is shown to include an example system 100 for retrieving data link layer information can include compute devices, such as a host device 102, a controller 104, network elements 106, and destination devices 108. The system 100 (described herein with respect to Figures 1-3) can represent generally any circuitry or combination of circuitry and executable instructions to retrieve data link layer information. The system 100 can include a host device 102 with trace request logic 110, a controller 104 with trace instruction logic 120, and network elements 106 with reply packet logic 130. As shown in Figure 3, the logic 110, 120, and 130 can be integrated into a compute device. The logics 110, 120, and 130 can be integrated via circuitry or as installed instructions into a

memory resource of the compute device, as described further with respect to Figures 2 and 3. For example, the logics 110, 120, and 130 can represent engines that perform functionalities of the devices to allow a host device 102 to retrieve data link layer information from each network element 106 on a path to a destination device 108 from the host device 102 using instructions provided by the controller 104.

[0010] The host device 102 can be any compute device capable of executing an operating system and including circuitry to communicate across a network. For example the host device 102 can be a server device connected to one of the plurality of switches (e.g., network elements 106 discussed below) of the network. The controller 104 represents a circuitry or a combination of circuitry and executable instructions to manage the network. For example, the controller 104 can be an SDN controller that manages the control plane of the network via northbound application programming interfaces ("APIs") with the application layer and data plane interfaces to the physical network elements 106. As shown in Figure 1, the network can include compute devices 102-108 connected by links. In Figure 1, the solid lines connecting the host device 102, the network elements 106, and the destination 108 represent physical links, while the dotted lines connecting the controller 104 to the network elements 106 that represent logical links (which may or may not be physical connections). A physical link generally represents one or a combination of a cable, wireless connection, fiber optic connection, or remote connections via a telecommunications link, an infrared link, a radio frequency link, or any other connectors of systems that provide electronic communication. The links can include, at least in part, intranet, the Internet, or a combination of both. The links can also include intermediate proxies, routers, switches, load balancers, and the like.

[0011] A network element 106 can be any device of the network for transferring information at the data link layer (a.k.a., layer 2 of the open system interconnection model of computer networking). Example network elements include switches, routers, hubs, and the like.

[0012] Referring to Figures 2 and 3 are block diagrams depicting example systems for retrieving data link layer information. The example system 200 of Figure 2 generally includes a host device 202, an SDN controller 204, and a network element

206 that represent the same devices as the host device 102, the controller 104, and the network element 106 of Figure 1 where the logics 110, 120, and 130 of Figure 1 are implemented as circuitry or a combination of circuitry and executable instructions.

[0013] The host device 202 generally includes a trace request engine 214 and an interface engine 216. In general, the interface engine 216 can display data link layer information received based on a request to the network element 206 via a packet sent by the trace request engine 214 to request data link layer information associated with a path through the network. The trace request engine 214 represents circuitry or a combination of circuitry and executable instructions to send a probe packet to the network element 206. For example, the trace request engine 214 can represent a combination of circuitry and executable instructions to generate a probe packet with headers representing the packet is requesting data link layer information associated with a path through the network and send the packet to the network element 206 via a transmission medium (such as via a network interface card connected to the network). For another example, the host computer device 202 can have an operating system functionality to send a request comprising a packet (e.g., a probe packet as discussed herein) designated to retrieve data link layer information regarding the plurality of switches of the network. A probe packet is any appropriate network packet having packet information to designate the transmission as being associated with a request for data link layer information. For example in an SDN environment, a probe packet is an ETHERNET frame including a header and a payload, where the header is distinct from other packets received based on the structure and/or contents of the header, such as the header containing a trace value.

[0014] The trace request engine 214 can be located on the host device 202 to send a probe request to a first network element 206 coupled to the host device 202. The probe request can activate a set of instructions on the first network element 206 to instruct the SDN controller to, in response to identification of the data link layer probe packet (e.g., the probe request), provide a first instruction to send a reply packet to a host device 202, and provide a second instruction to forward the data link layer probe packet to a second network element as discussed further herein with regards to engines 222, 224, 232, 236, and 238. The trace request engine 214 can initialize or otherwise

instruct the SDN controller to know how to respond to the probe request (e.g., when a probe packet is received, instruct the switch making the request to send a reply packet to the host device 202 and forward the packet on with a decremented trace value to another switch). The trace request can initiate a data link layer information session (i.e., a set of operations at elements 206 of the network resulting from a trace request from the host device 202) to retrieve layer 2 information (e.g., the trace value) of each network element 206. For example, the trace values of each switch can be associated with a decremented trace value based on the number of switches encountered in a data link layer information request session initiated by the host computer device 202. Each probe packet can contain an identifier to designate which packets belong to which session. A reply packet can contain an identifier of network element 206 and a trace value. A reply packet can also include other data link layer information such as a media access control ("MAC") address or a device name.

[0015] The interface engine 216 represents circuitry or a combination of circuitry and executable instructions to provide an interface, such as a command line interface ("CLI"), to make the request to generate and send the packet to an element of the network, and cause information received in response to the request to display, such as display via the CLI. The interface engine 216 can cause the data link information to display based on the plurality of reply packets. The data link information to include the plurality of source identifiers and the plurality of trace values. For example, the interface engine 216 can generate the packet associated with the probe request and display the trace value and the MAC address of the network element of each reply packet received.

[0016] The interface engine 216 can produce data link layer information based on an analysis of a plurality of reply packets received by the host device 202. For example, a plurality of reply packets can include a plurality of source identifiers and a plurality of trace values and the interface engine 216 can provide a status of each network element 206 based on an analysis of the trace values (e.g., the status of "functional" or "up" when a trace value is received or identify a distance of each of the plurality of switches from the host device 202 based on the plurality of trace values).

[0017] The example system 200 can include an analysis engine 218 that represents circuitry or a combination of circuitry and executable instructions to analyze

replied information from the trace request and/or diagnose a network element status based on the replied information. For example, the analysis engine 218 can build a topology from a plurality of response received from the trace request or identify the network element status of the network element 206 is to be set to "failure" or "down" because a reply was not received from the network element 206 (likely due to a failure at the network element 206). The analysis engine 218 can perform any operations to provide layer 2 information. For example, the analysis engine 218 can identify the location of the first switch in a network topology, analyze the decremented trace value of the reply packet to identify the distance of the first switch from the source device, and diagnose the network based on network element information of the reply packet. For another example, the analysis engine 218 can allow the host device 202 to collect the data link layer information from the plurality of reply packets, identify a topology of the network based on the data link layer information, trace a path of through the network based on the data link layer information, and diagnose an error with one of the plurality of switches based on the plurality of reply packets.

[0018] The SDN controller 204 generally includes an identification engine 220, a reply instruction engine 222, and a forward instruction engine 224. The identification engine 220 represents any circuitry or combination of circuitry and executable instructions to identify a request from a network element 206 is in response to a probe packet being received at the network element 206. For example, a data link layer information retrieval request can include packet with a header structure (or other packet information) to denote the request is for a layer two trace and identification of the header structure (or other packet information) initiates sending probe packet received by the first switch.

[0019] The reply instruction engine 222 represents any circuitry or combination of circuitry and executable instructions to instruct a network element 206 of the network (e.g., a first switch of a plurality of switches of the network) to send a reply packet with data link layer information to a source device (e.g., the host device 202 that sent the probe packet as part of the trace request) in response to a receipt of the probe packet at the network element 206. The reply instructions can request the trace value of the probe packet to be decremented before sending the reply. The instruction to send a reply

packet to the source device can include instructions to send the network element information (e.g., information associated with the network element 206 such as a MAC address, a switch identifier, and/or a port associated with a neighbor switch, etc.).

[0020] The forward instructions engine 224 represents any circuitry or combination of circuitry and executable instructions to instruct the network element 206 to send a probe packet to another network element 206 of the network. For example, sending a probe packet can comprise forwarding a probe packet to a subsequent network element 206 in a path to a destination where the forwarded probe packet has a trace value that is decremented from the probe packet upon receipt. The probe packet contains a trace value used to identify a network element of a network path. The trace value of a probe packet can be decremented after being received and before forwarding on a probe packet to a next network element 206 along a path to a destination device. The forwarded instructions can request the trace value of the probe packet be decremented before forwarding the probe packet to another network element 206.

[0021] The SDN controller 204 can include an information retrieval engine 226. The information retrieval engine 226 represents any circuitry or combination of circuitry and executable instructions to retrieve network element information of the network element 206 based on a path derived from a source address and a destination address. The source address can represent the address of the device that initiated the data link layer information retrieval request (e.g., the host device 202) and the destination address can represent the device that is the final target of the data link layer information retrieval request, such as a destination device. The information retrieval engine 226 can be used to store a list of network elements of the network on a data store 210 of the SDN controller. The SDN controller can instruct which one of the list of switches is to receive a forwarded probe packet via the forward instruction engine 224.

[0022] The network element 206 generally includes a match engine 230, a retrieval engine 232, a decrement engine 234, a reply packet engine 236, and a forward packet engine 238. The match engine 230 represents any circuitry or combination of circuitry and executable instructions to perform a lookup of a received packet to determine if the network element 206 knows what to do with the packet. For example, the match engine 230 can lookup entries in a flow table for a match associated with the

information of the packet (e.g., the source, the destination, and/or other header information). In that example, if a table-miss occurs (e.g., if the received packet does not match an entry in the flow table or matches the default entry in the flow table), the instruction fetch engine 232 can be initiated. The instruction fetch engine 232 represents any circuitry or combination of circuitry and executable instructions to request an instruction from the controller 204 regarding the packet. For example in a table-miss, the switch does not know what to do with the packet so the switch can request particular instructions from the SDN controller 204 regarding how to forward the packet. The network element 206 can receive instructions from the controller 204 to cause the decrement engine 234, the reply packet engine 236, and/or the forward engine 238 to operate.

[0023] The decrement engine 234 represents any circuitry or a combination of circuitry and executable instructions to decrement the trace value of the probe packet. For example, a probe packet header can be parsed to discover data link layer information including a trace value, the trace value can be extracted and decremented, and the decremented trace value can be used for the packet being forwarded to the next network element 206. The trace value is to be decremented at each network element 206 so that each network element 206 is associated with a particular trace value. For example, a first switch closer to the host device 202 can have a higher trace value than a second switch further away from the host device 202.

[0024] The reply packet engine 236 represents any circuitry or a combination of circuitry and executable instructions to generate and/or send a reply packet to the source of the probe packet. For example, the source of the data link layer information request can be retrieved from the header of the received probe packet and the source can be used to address a reply packet generated by the reply packet engine 236 where the reply packet contains a trace value associated with the network element 206. The trace value of the reply packet can be the trace value of the probe packet when received at the network element 206 or a decremented trace value based on the implementation that allows each network element 206 to be associated with a trace value when received at the host device 202.

[0025] The forward packet engine 238 represents any circuitry or a combination of circuitry and executable instructions to generate and/or forward the probe packet to a subsequent network element 206 in a path across the network (e.g., from the source host device 202 to a destination device). The forward packet engine 238 can generate a probe packet similar to the probe packet received by the network element 206, but the trace value of the forwarded probe packet is decremented. Thus, a subsequent network element 206 can receive a probe packet, send a reply packet, and further decrement the trace value for a subsequent network element 206, which can repeat until the destination device is reached (or a network failure occurs).

[0026] The data stores 210 can contain information utilized by the engines of devices 202, 204, and 206. For example, the data store 210 can store transmission control protocol/internet protocol ("TCP/IP") information, address information, flow entry information, a flow table, instructions, etc. As shown in Figure 2, each device can contain a data store 210. Alternatively or in addition, the data stores 210 can be distributed across devices (e.g., data nodes of a database stored across devices).

[0027] Figure 3 depicts the example system 300 can comprise a host device 302, a SDN controller 304, and a network element 306 that are similar to the host device 202, the SDN controller 204, and the network element 206 of Figure 2. Referring to Figure 3, the engines of Figure 2 can be implemented as a combination of circuitry and executable instructions represented as a memory resource 340 operatively coupled to a processor resource 342 on a device of the system 300. The processor resources 342 can be operatively coupled to data stores 310. The data stores 310 can be the same as the data stores 210 of Figure 2.

[0028] Referring to Figure 3, the memory resources 340 can contain a set of instructions that are executable by the processor resources 342. The set of instructions are operable to cause the processor resources 342 to perform operations of the system 300 when the set of instructions are executed by a processor resource 342. The set of instructions stored on the memory resources 340 can be represented as a trace request module 312, an interface module 314, an analysis module 316, an identification module 320, a reply instruction module 322, a forward instruction module 324, an information retrieval module 326, a match module 330, an instruction fetch module 332, a

decrement module 334, a reply packet module 336, and a forward packet module 338 that represent program instructions that when executed function as the trace request engine 214, the interface engine 216, the analysis engine 218, the identification engine 220, the reply instruction engine 222, the forward instruction engine 224, the information retrieval engine 226, the match engine 230, the instruction fetch engine 232, the decrement engine 234, the reply packet engine 236, and the forward packet engine 238 of Figure 2, respectively. The processor resource 342 can carry out a set of instructions to execute the modules 312-338 of the system 300, and/or any other appropriate operations among and/or associated with the modules of the system 300. For example, the processor resource 342 can carry out a set of instructions to send, from a host device, a probe request to a first switch coupled to the host device where the probe request is to activate a first set of instructions on the first switch to send a reply packet to the host device with a trace value, forward a data link layer probe packet of the probe request to second switch, and produce data link layer information based on an analysis of a plurality of reply packets containing a plurality of source identifiers and the plurality of trace values received by the host device. For another example, the processor resource 342 can carry out a set of instructions to instruct a first switch to send a reply packet with a decremented trace value to a source of the first trace packet and instruct the first switch to send a second trace packet to a second switch with a decremented trace value.

[0029] Although these particular modules and various other modules are illustrated and discussed in relation to Figure 3 and other example implementations, other combinations or sub-combinations of modules can be included within other implementations. Said differently, although the modules illustrated in Figure 3 and discussed in other example implementations perform specific functionalities in the examples discussed herein, these and other functionalities can be accomplished, implemented, or realized at different modules or at combinations of modules. For example, two or more modules illustrated and/or discussed as separate can be combined into a module that performs the functionalities discussed in relation to the two modules. As another example, functionalities performed at one module as discussed in relation to these examples can be performed at a different module or different modules.

[0030] The processor resource 342 is any appropriate circuitry capable of processing (e.g. compute) instructions, such as one or multiple processing elements capable of retrieving instructions from the memory resource 340 and executing those instructions. For example, the processor resource 342 can be a central processing unit ("CPU") that enables data link layer information retrieval by fetching, decoding, and executing modules 312-338. Example processor resources 222 include at least one CPU, a semiconductor-based microprocessor, an application specific integrated circuit ("ASIC"), a field-programmable gate array ("FPGA"), and the like. The processor resource 342 can include multiple processing elements that are integrated in a single device or distributed across devices. The processor resource 342 can process the instructions serially, concurrently, or in partial concurrence.

[0031] The memory resource 340 and the data store 310 represent a medium to store data utilized and/or produced by the system 300. The medium is any non-transitory medium or combination of non-transitory mediums able to electronically store data, such as modules of the system 300 and/or data used by the system 300. For example, the medium can be a storage medium, which is distinct from a transitory transmission medium, such as a signal. The medium can be machine-readable, such as computer-readable. The medium can be an electronic, magnetic, optical, or other physical storage device that is capable of containing (i.e. storing) executable instructions. The memory resource 340 can be said to store program instructions that when executed by the processor resource 342 cause the processor resource 342 to implement functionality of the system 300 of Figure 3. The memory resource 340 can be integrated in the same device as the processor resource 342 or it can be separate but accessible to that device and the processor resource 342. The memory resource 340 can be distributed across devices. The memory resource 340 and the data store 310 can represent the same physical medium or separate physical mediums. The data of the data store 310 can include representations of data and/or information mentioned herein.

[0032] In the discussion herein, the engines 212-238 of Figure 2 and the modules 312-338 of Figure 3 have been described as circuitry or a combination of circuitry and executable instructions. Such components can be implemented in a number of fashions.

Looking at Figure 3, the executable instructions can be processor-executable instructions, such as program instructions, stored on the memory resource 340, which is a tangible, non-transitory computer-readable storage medium, and the circuitry can be electronic circuitry, such as processor resource 342, for executing those instructions. The instructions residing on the memory resource 340 can comprise any set of instructions to be executed directly (such as machine code) or indirectly (such as a script) by the processor resource 342.

[0033] In some examples, the system 300 can include the executable instructions can be part of an installation package that when installed can be executed by the processor resource 342 to perform operations of the system 300, such as methods described with regards to Figures 4-8. In that example, the memory resource 340 can be a portable medium such as a compact disc, a digital video disc, a flash drive, or memory maintained by a computer device, such as a host device 302 of Figure 3, from which the installation package can be downloaded and installed. In another example, the executable instructions can be part of an application or applications already installed. The memory resource 340 can be a non-volatile memory resource such as read only memory ("ROM"), a volatile memory resource such as random access memory ("RAM"), a storage device, or a combination thereof. Example forms of a memory resource 340 include static RAM ("SRAM"), dynamic RAM ("DRAM"), electrically erasable programmable ROM ("EEPROM"), flash memory, or the like. The memory resource 340 can include integrated memory such as a hard drive ("HD"), a solid state drive ("SSD"), or an optical drive.

[0034] Referring to Figures 1-3, the engines 212-238 of Figure 2 and the modules 312-338 can be distributed across devices 102, 104, 106, 108 or a combination thereof. The engine and/or modules can complete or assist completion of operations performed in describing another engine and/or module. For example, the reply packet module 326 of Figure 3 can be executed on a processor resource 342 to request, complete, or perform the methods or operations described with regards to the reply packet engine 236 of Figure 2 as well as the match engine 230, the instruction fetch engine 232, the decrement engine 234, and the forward packet engine 238 of Figure 2. Thus, although the various engines and modules are shown as separate engines in Figures 2 and 3, in

other implementations, the functionality of multiple engines and/or modules may be implemented as a single engine and/or module or divided in a variety of engines and/or modules. In some example, the engines of the system 200 and a processor resource executing the modules of the system 300 can perform example methods described in connection with Figures 4-8.

[0035] Figure 4 is an example sequence diagram depicting operations of example systems for retrieving data link layer information. Blocks 402, 404, and 406 represent the devices being interacted with and the arrows 410-430 represent operations.

[0036] A `trace.setup()` request 410 can initialize the controller 404 to allow the network to handle probe packets. The controller 404 can set up the flow table entries to activate a miss using `tablemiss.setup()` operation 412 where a network element 406 is instructed to contact the controller 404 when a probe packet is encountered.

[0037] The host device 402 sends a `trace.switch()` request 414 to a network element 406 (e.g., a switch, such as the first switch of a path of switches to a destination). The network element 406 performs a `table.match()` operation 416 to lookup the packet information in the flow table. A table-miss is identified and a request to contact the controller 404 is made via `table.miss()` operation 418. The controller 404 performs `packet.ID()` operation 420 to identify the packet and instructions associated with the packet (in this case, the instructions are to reply to the host device 402 and forward on the probe packet to the next network element of the trace path). The controller 404 performs `send.instructions()` operation 422 to inform the network element 406 of the operations to perform with the received packet.

[0038] In this example, the trace value of probe packet is decremented via the network element 406 performing a `decrement()` operation 424. The network element performs the `trace.reply()` operation 426 to send a reply packet to the host device 402 where the host device 402 uses `trace.display()` operation 428 to display the trace value of the received reply packet (which is the decremented trace value). The network element also performs the `trace.next()` operation 430 to forward on a trace request (i.e., forward the probe packet) with a decremented trace value for the next network element to perform the same operations. In this manner, the host device 402 can receive a reply

packet from each operational switch in the network that is along a path from the host device 402 to a destination device (not shown). For example, all the switches and routers in a path from a host device to a peer device can provide a trace value (in contrast to only routers identified by a layer 3 trace route operation.) The trace values can then be used, for example, to identify the distance of each network element from the host device 402.

[0039] Figures 5-8 are flow diagrams depicting example methods for retrieving data link layer information. Referring to Figure 5, example methods at a host device for retrieving data link information can generally comprise sending a trace request to a first network element, receiving a reply packet, and causing data link layer information to display.

[0040] At block 502, a host device sends a trace request to a network element. The trace request is represented in packet information (e.g., a particular packet structure) that is identifiable by each network element of the network. For example in an SDN environment, the SDN controller can be configured to identify customized packet structures and forward traffic based on the information in the packet headers. In this manner, an implementation of the example methods herein are not subject to a single protocol and the SDN controller can act as a mediator and/or translator for requests across device types and brands to allow the trace request to be protocol agnostic. At block 504, a reply packet from the network element is received by the host device and the data link layer information of the received reply packet is caused to be displayed at block 506.

[0041] Referring to Figure 6, example methods at a network element for retrieving data link information can generally comprise identifying a probe packet associated with a source, retrieving instructions from a controller regarding the probe packet, decrementing a trace value of the probe packet, sending a first reply to the source of the probe packet, and forwarding the probe packet with the decremented trace value to a network element.

[0042] At block 602, a probe packet associated with a source is identified at a first network element. Instructions are retrieved by the first network element from a controller at block 604. The instructions are associated with how to handle the probe

packet when the probe packet is received at the network element. For example, the probe packet can cause a table miss in the local flow table and initiate a request for further instructions from the SDN controller regarding how to forward the probe packet.

[0043] At block 606, a trace value of the probe packet is decremented at the first network element. For example, the trace value is decremented based on the path from the source to the destination of a data link layer trace request (i.e., at each network element of a network path from a designated source to a designated destination). For another example, each network element of a path through the network can decrement the probe packet trace value before making a lookup in the local storage to identify which network element is next based on a last-known path to a destination.

[0044] At block 608, a reply packet is sent from the first network element to the source of the probe packet with a second trace value. The second trace value can be the received trace value or the decremented trace value based on the desired implementation of the description herein. The trace values of the reply packet are to be different for each network element to allow for the host device to identify which network elements are reached in a data link layer trace request session for retrieving data link layer information associated with a path through the network. At block 610, the probe packet with the decremented trace value is forwarded from the first network element to a second network element. For example, the probe packet can be forwarded from a first switch to a second switch connected to the first switch or from a first switch to a router connected to the first switch.

[0045] Referring to Figure 7, example methods at a controller for retrieving data link information can generally comprise identifying a probe packet and providing instructions regarding the probe packet. At block 702, a controller identifies a probe packet associated with a source. For example, the source can be parsed from the packet header structure. The controller can provide instructions to a network element based on the probe packet header(s) (e.g., the header information of the probe packet) at block 704. The instructions can include a first instruction to send a reply packet to a source (such as a host device) and forward the probe packet to a neighbor network element (e.g., the next network element in a path of network elements from a source to

a destination). A neighbor network element is a second network element directly connected to a first network element at a port on the first network element.

[0046] Figure 8 includes blocks similar to blocks of Figure 6 and provides additional blocks and details. In particular, Figure 8 depicts additional blocks and details generally regarding parsing information from the probe packet, identifying a probe packet, retrieving instructions, identifying a forwarding switch, and maintaining the probe packet. Blocks 806, 812, 820, 822, and 828 are the same as blocks 602, 604, 606, 608, and 610 of Figure 6 and, for brevity, their respective descriptions have not been repeated.

[0047] At blocks 802 and 804, information is parsed from the headers of the probe packet. For example, the trace value and the source of the data link layer information request are parsed from the probe packet header information.

[0048] At block 808, the probe packet is compared to a flow table stored on the network element, such as a switch. At block 810, whether a miss or a match occurs is determined from the comparison of the probe packet to the flow table. The instructions retrieved at block 812 are based on whether the probe packet is matched or not at block 814. If the flow table contains a match, then a flow entry associated with the probe packet is executed at block 816. (e.g., in the case of a match, the flow entry is previously received from the SDN controller and locally stored on the switch with instructions to perform operations associated with blocks 820-828). If the flow table does not contain a match, then the probe packet is sent to the SDN controller for further instructions. (e.g., in the case of a miss, the SDN controller can send the network element instructions to perform operations 820-828).

[0049] At block 824, a second network element is identified as a neighbor of the first network element based on a port identified from path information of a path between a source and a destination (e.g., path information stored on the first network element). For example, a second switch can be identified as a neighbor of a first switch based on a port connection (i.e., a connection of a port of the first switch to a port of the second switch) derived from a previously used path to forward a packet from a source address to a destination address of the data link layer information request.

[0050] At block 826, the probe packet is maintained to send a second reply packet to the source when forwarding the probe packet to a second switch. For example, the trace value to be forwarded should be maintained as one less than the trace value received by the network element. In this manner, a first switch sends a reply packet having a trace value that is one value higher than the trace value of the reply packet sent from a second switch when the first switch is before the second switch in a path from the source device to the destination device. Thus, the data link layer information can be sent from the network element to the host device and avoid storage of the data link layer information on the network controller.

[0051] Although the flow diagrams of Figures 4-8 illustrate specific orders of execution, the order of execution may differ from that which is illustrated. For example, the order of execution of the blocks may be scrambled relative to the order shown. Also, the blocks shown in succession may be executed concurrently or with partial concurrence. All such variations are within the scope of the present description.

[0052] The present description has been shown and described with reference to the foregoing examples. It is understood, however, that other forms, details, and examples may be made without departing from the spirit and scope of the following claims. The use of the words "first," "second," or related terms in the claims are not used to limit the claim elements to an order or location, but are merely used to distinguish separate claim elements.

CLAIMS

What is claimed is:

- 1 1. A system comprising:
2 a software defined-network ("SDN") controller coupled to a network comprising a
3 plurality of network elements, the SDN controller including logic to:
4 instruct a first network element of the plurality of network elements to send a
5 reply packet with data link layer information to a source device of a first probe
6 packet in response to a receipt of the first probe packet at the first network
7 element, the data link layer information including a first trace value; and
8 instruct the first network element to send a second probe packet to a second
9 network element of the plurality of network elements, the second probe packet
10 including a second trace value from the first probe packet, the second value
11 being a decremented trace value.
- 1 2. The system of claim 1, wherein the SDN controller is further to:
2 retrieve network element information of the second network element based on a
3 path derived from a source address and a destination address of data link layer
4 information retrieval request that initiates sending a third probe packet received by
5 the first network element, the third probe packet having an undecrement trace
6 value; and
7 wherein the instruction to send a reply packet to the source device includes
8 instructions to send the network element information, the network element
9 information including a switch identifier, a MAC address, and a port associated with
10 the second network element.
- 1 3. The system of claim 1, wherein the SDN controller includes a data store containing a
2 list of switches of the network, the SDN controller to instruct which one of the list of
3 switches is to receive a forwarded probe packet.

- 1 4. The system of claim 1, wherein the source device is a host computer device
2 connected to one of the plurality of network elements, the host computer device
3 having an operating system functionality to send a request comprising the first probe
4 packet to retrieve data link layer information regarding the plurality of network
5 elements of the network; and the decremented trace value is based on a number of
6 switches encountered in a data link layer information request session initiated by the
7 host computer device.
- 1 5. The system of claim 4, wherein the SDN controller is coupled to the source device,
2 the source device is to:
3 identify a location of the first network element in a network topology;
4 analyze the decremented trace value of the reply packet to identify the distance
5 of the first network element from the source device; and
6 diagnose the network based on network element information of the reply packet,
7 the information of the reply packet to include a MAC address of the first network
8 element and the decremented trace value.
- 1 6. A computer-readable storage medium comprising a set of instructions executable by
2 a processor resource to:
3 send, from a host device, a probe request to a first switch coupled to the host
4 device, the probe request to activate a first set of instructions on the first switch to:
5 send a reply packet to the host device with a trace value; and
6 forward a data link layer probe packet of the probe request to a second
7 switch, the data link layer probe packet to contain a decremented trace value
8 when forwarded to the second switch and the second switch connected to a port
9 on the first switch,
10 wherein the first set of instructions is received from a software-defined
11 network ("SDN") controller;
12 produce data link layer information based on an analysis of a plurality of reply
13 packets received by the host device, the plurality of reply packets to include the
14 plurality of source identifiers and the plurality of trace values.

- 1 7. The medium of claim 6, wherein the set of instructions is executable by the
2 processor resource to:
3 cause the data link information to display based on the plurality of reply packets,
4 the data link information to include the plurality of source identifiers and the plurality
5 of trace values; and
6 identify a distance of each of the plurality of switches from the host device based
7 on the plurality of trace values.
- 1 8. The medium of claim 6, wherein the set of instructions is executable by the
2 processor resource to:
3 collect the data link layer information from the plurality of reply packets;
4 identify a topology of the network based on the data link layer information; and
5 trace a path of through the network based on the data link layer information.
- 1 9. The medium of claim 6, wherein the set of instructions is executable by the
2 processor resource to:
3 diagnose an error with one of the plurality of switches based on the plurality of
4 reply packets.
- 1 10. The medium of claim 6, wherein the set of instructions is executable by the
2 processor resource to:
3 instruct the SDN controller to, in response to identification of the data link layer
4 probe packet, provide a first instruction to send a reply packet to a host device, and
5 provide a second instruction to forward the data link layer probe packet to the
6 second switch.
- 1 11. A method for retrieving data link layer information comprising:
2 identifying, by a first switch, a probe packet associated with a source;
3 retrieving, by the first switch, instructions from a software-defined network
4 ("SDN") controller regarding the probe packet;

5 decrementing, at the first switch, a first trace value of the probe packet;
6 sending, from the first switch, a first reply packet to the source of the probe
7 packet with a second trace value; and
8 forwarding, from the first switch, the probe packet with the decremented trace
9 value to a second switch coupled to the first switch.

1 12. The method of claim 11, wherein identifying comprises:

2 comparing, by the first switch, the probe packet to a flow table, the flow table
3 stored on the first switch.

1 13. The method of claim 12, comprising:

2 determining, by the first switch, whether the probe packet is missed in the flow
3 table.

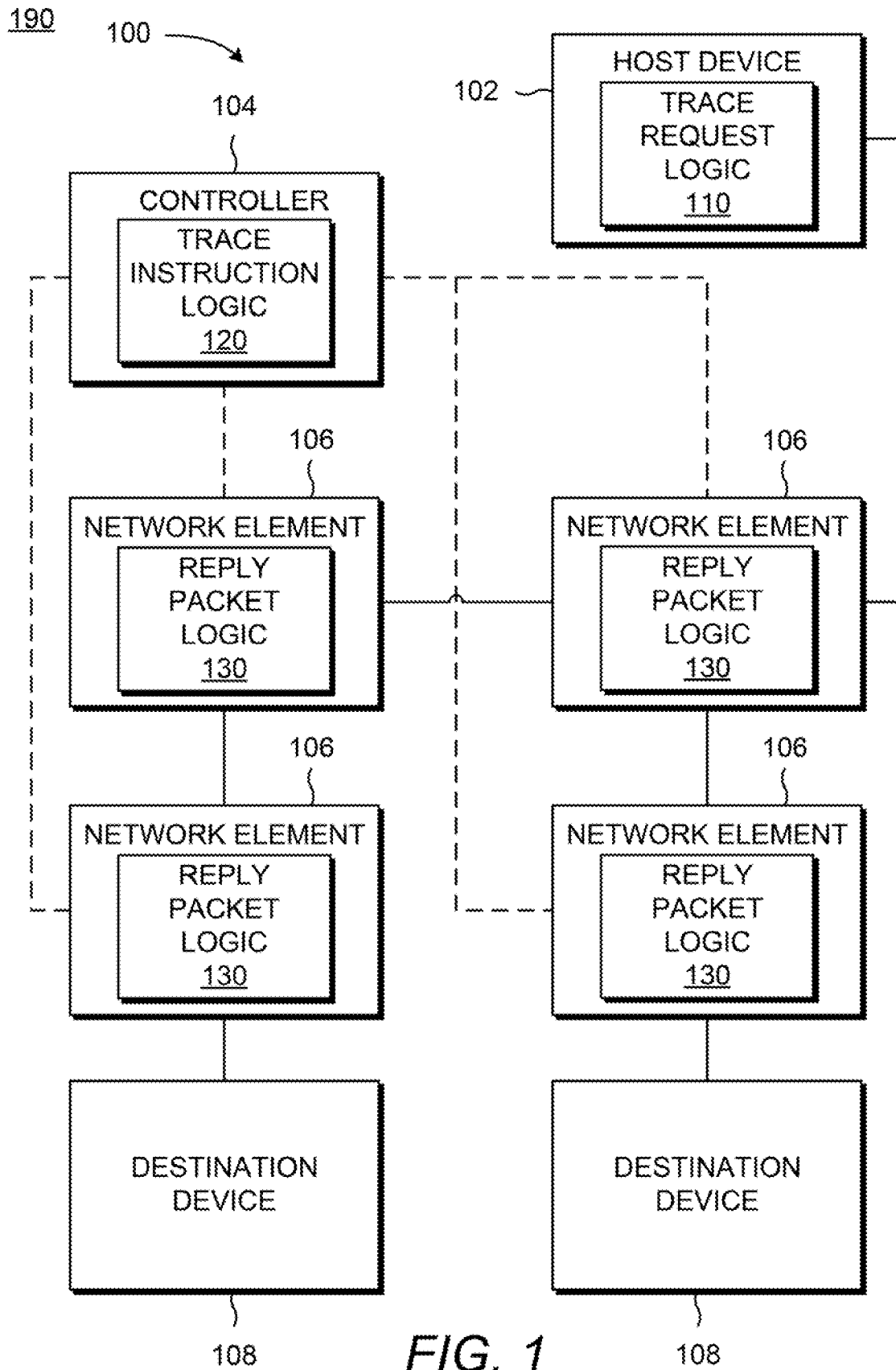
1 14. The method of claim 13, wherein retrieving comprises one of:

2 executing a flow entry associated with the probe packet when the probe packet is
3 matched, the flow entry provided by the SDN controller and stored in flow table of
4 the first switch; and
5 sending the probe packet to the SDN controller when the probe packet is missed.

1 15. The method of claim 11, comprising:

2 parsing the source from a probe packet header;
3 parsing the first trace value from the probe packet header;
4 maintaining the probe packet to send a second reply packet to the source when
5 forwarding the probe packet to the second switch; and
6 identifying the second switch is a neighbor of the first switch based on a port
7 connection of the first switch, the port connection identified based on path
8 information between a source address and a destination address stored on the first
9 switch.

1/6



2/6

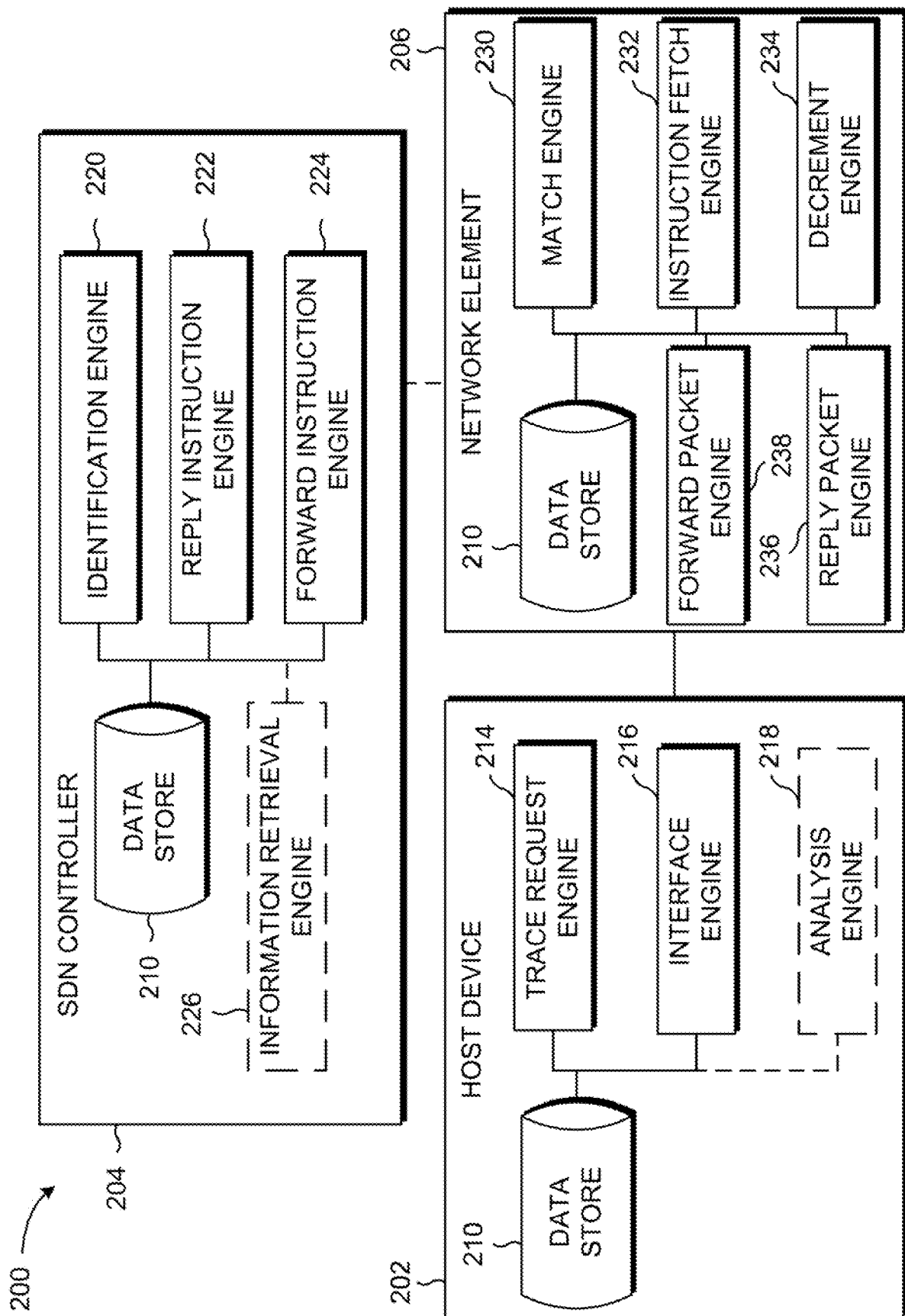


FIG. 2

3/6

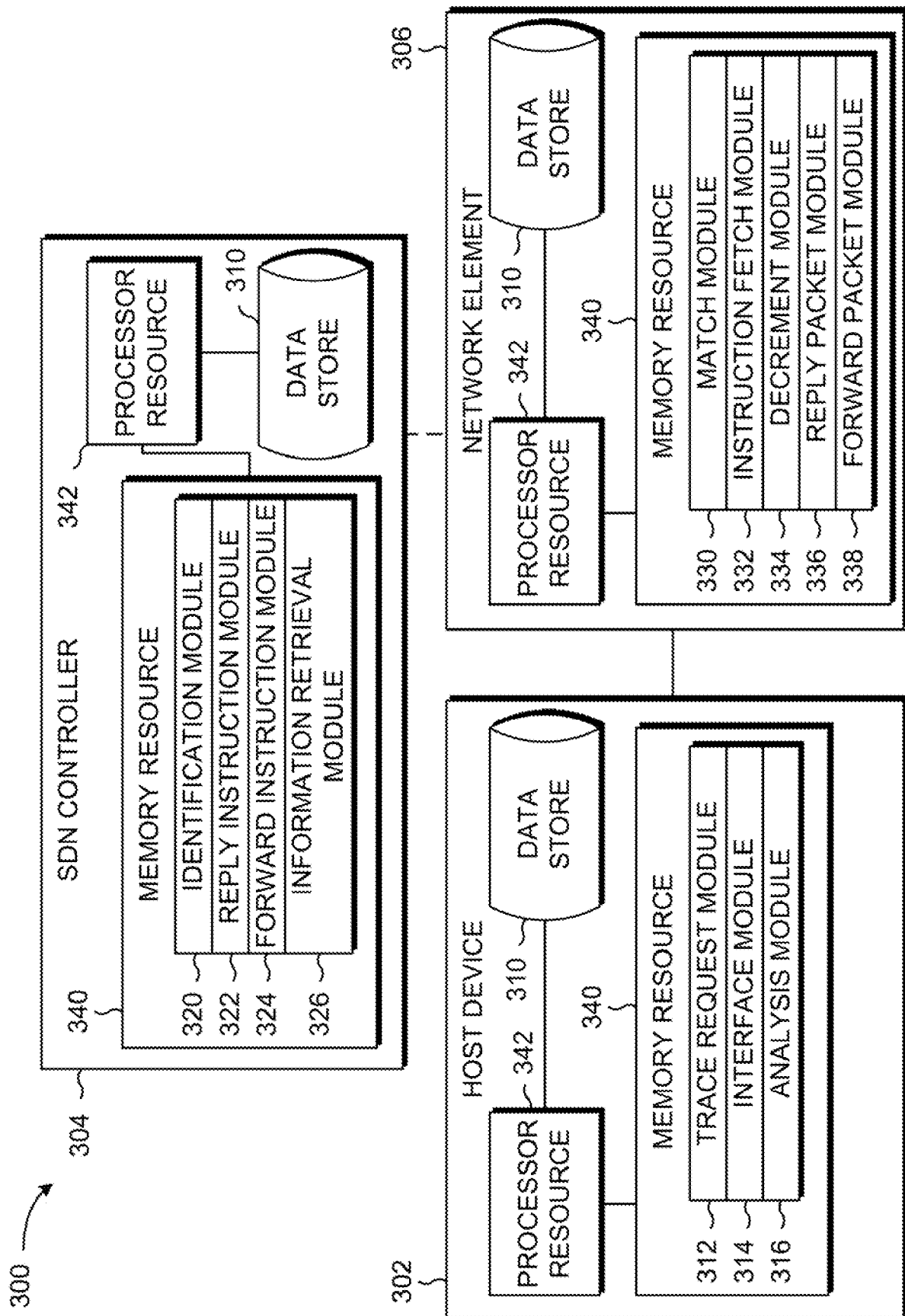


FIG. 3

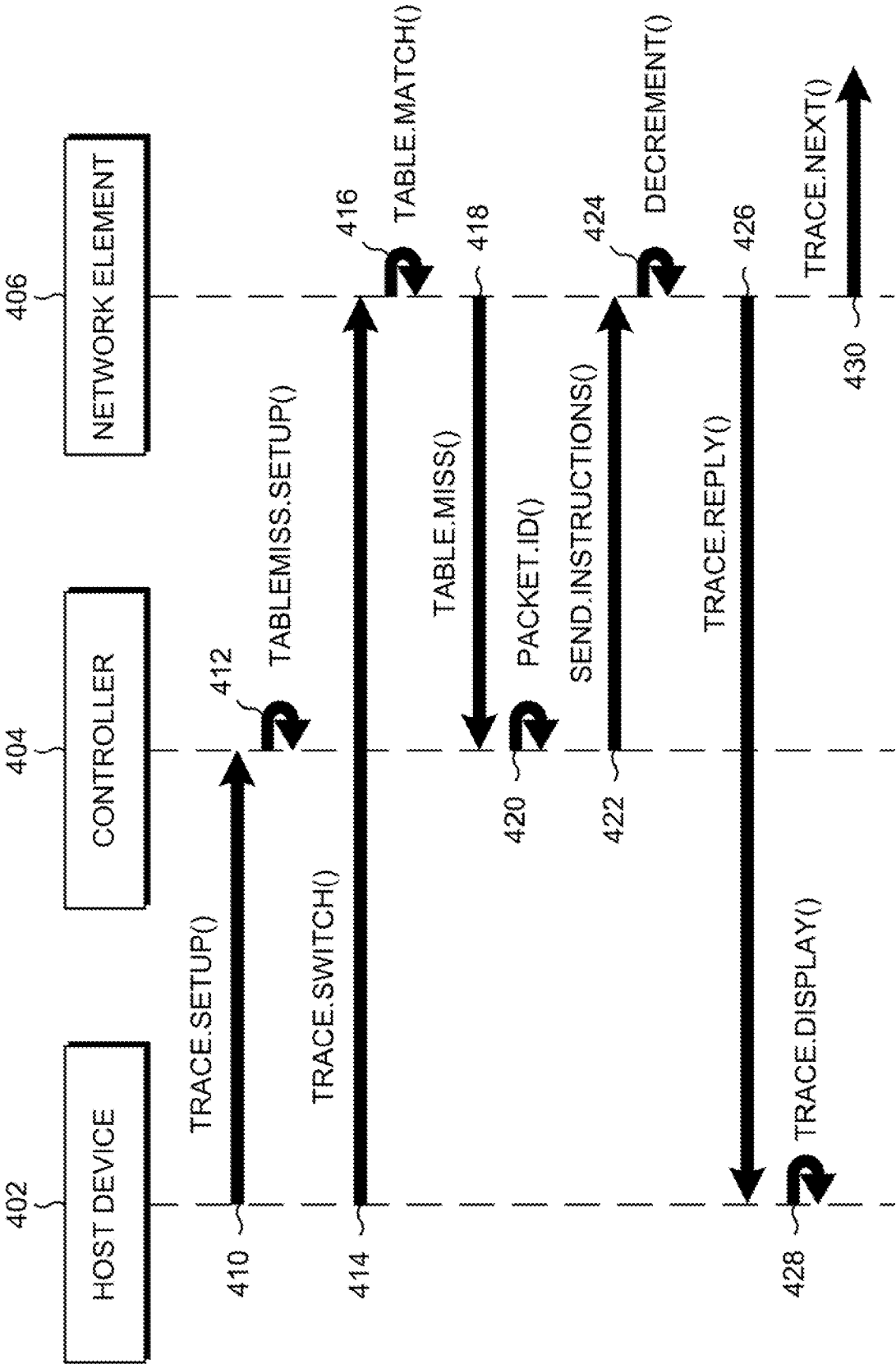
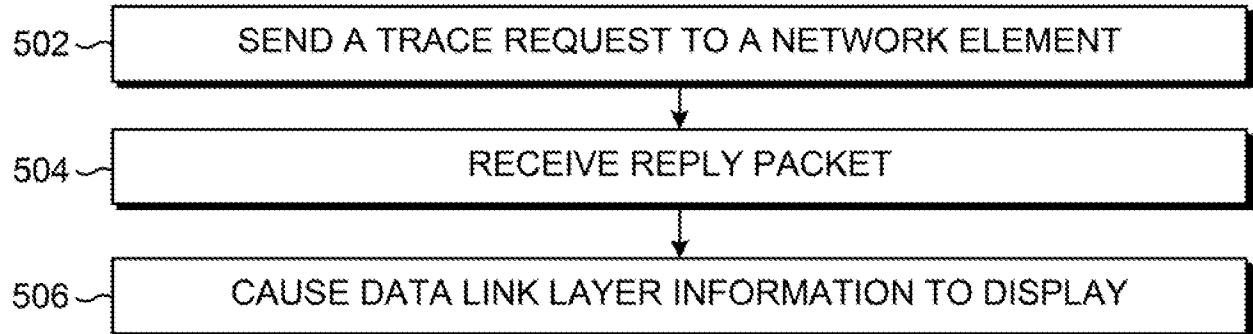
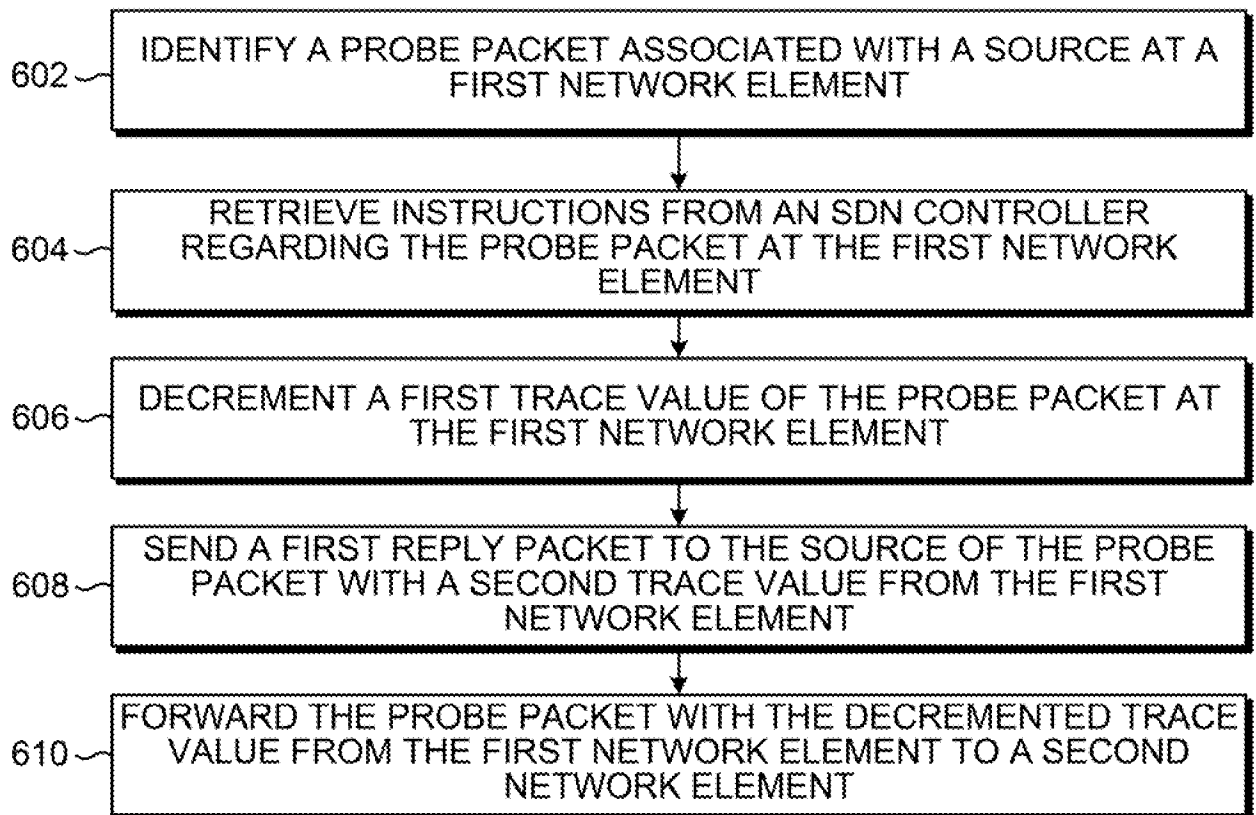
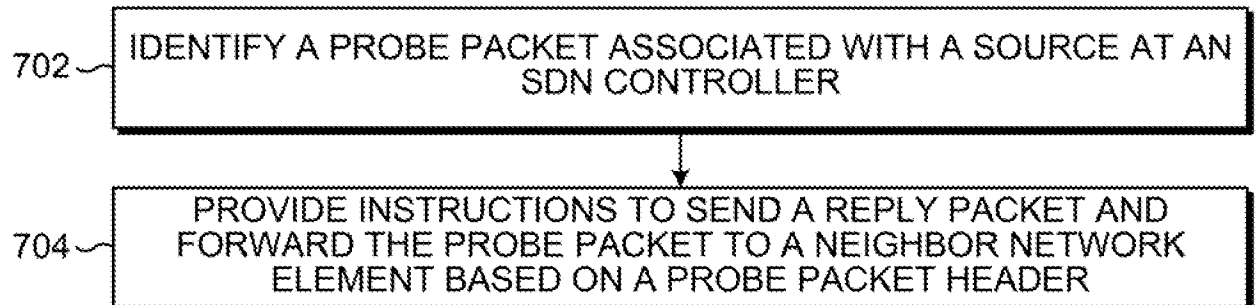


FIG. 4

5/6

*FIG. 5**FIG. 6**FIG. 7*

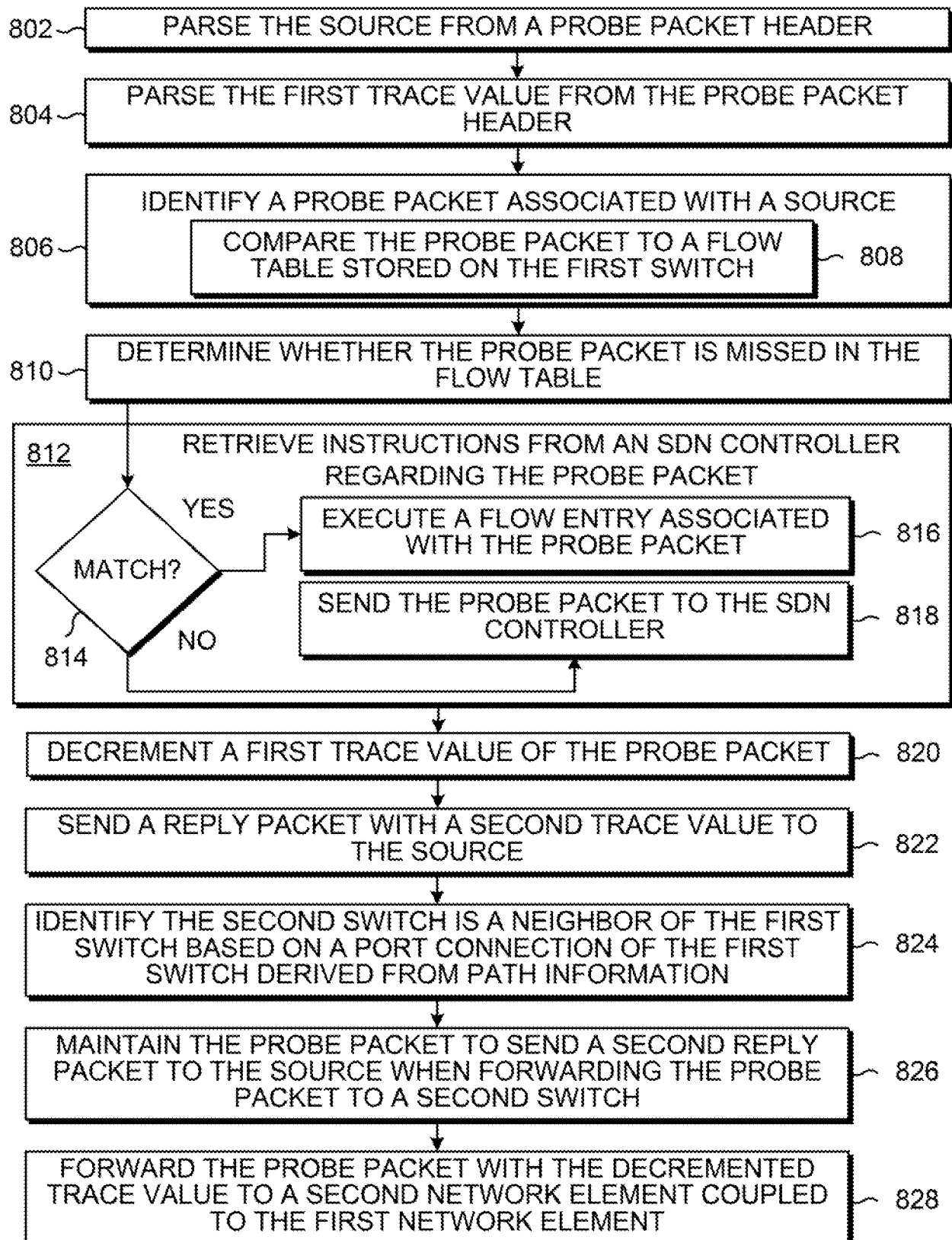


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/011008**A. CLASSIFICATION OF SUBJECT MATTER****H04L 12/70(2013.01)i, H04L 29/06(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 12/70; H04L 12/46; G06F 15/16; H04L 12/24; H04L 12/26; H04L 29/12; H04L 12/56; H04L 29/06

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: trace value, network element, controller, monitoring

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2013-052564 A2 (BIG SWITCH NETWORKS, INC.) 11 April 2013 See page 11, lines 1-4, page 14, lines 14-32, page 18, line 11 - page 22, line 34, page 25, line 27 - page 26, line 11, claim 1 and figure 8.	1-15
Y	WO 2014-074542 A1 (ALCATEL LUCENT) 15 May 2014 See paragraphs [0102], [0113], claims 1-2, 7-8 and figure 12.	1-15
A	US 2008-0022389 A1 (GEORGE CALCEV et al.) 24 January 2008 See paragraphs [0017]-[0022] and figure 3.	1-15
A	US 2014-0098669 A1 (VIPIN GARG et al.) 10 April 2014 See paragraphs [0055]-[0063], claim 1 and figure 1.	1-15
A	US 2013-0128751 A1 (SRIKANTH KEESARA et al.) 23 May 2013 See paragraphs [0018]-[0020] and figures 2-3.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

06 October 2015 (06.10.2015)

Date of mailing of the international search report

06 October 2015 (06.10.2015)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

LEE, Dong Yun

Telephone No. +82-42-481-8734



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/011008

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2013-052564 A2	11/04/2013	EP 2748992 A2 JP 2014-531158 A WO 2013-052564 A3	02/07/2014 20/11/2014 15/08/2013
WO 2014-074542 A1	15/05/2014	KR 10-2015-0067365 A US 2013-0064102 A1	17/06/2015 14/03/2013
US 2008-0022389 A1	24/01/2008	US 2011-0075845 A1 US 7865717 B2 US 8245028 B2 WO 2008-011219 A1	31/03/2011 04/01/2011 14/08/2012 24/01/2008
US 2014-0098669 A1	10/04/2014	CN 104704779 A US 9071529 B2 WO 2014-057375 A2 WO 2014-057375 A3	10/06/2015 30/06/2015 17/04/2014 05/06/2014
US 2013-0128751 A1	23/05/2013	None	