



- (51) **International Patent Classification:**
H04L 29/08 (2006.01) *H04L 12/70* (2013.01)
- (21) **International Application Number:**
PCT/US2015/024102
- (22) **International Filing Date:**
2 April 2015 (02.04.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
426/CHE/2015 29 January 2015 (29.01.2015) IN
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** SHANBAG, Pramod; Hewlett-Packard Company, Sy. No. 192 Whitefield Road, Mahadevapura Post, Bangalore 560048 (IN). PADINHAKARA, Mohammed Javed; #266, 5th Cross, 7th Main Minhajnagar, JP Nagar Post Banashankari, Bangalore 560078 (IN). SINGH, Santosh Kumar; Hewlett-Packard Company, Sy. No. 192 Whitefield Road, Mahadevapura Post, Bangalore 560048 (IN).
- (74) **Agents:** GARDINER, Austin W. 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).

[Continued on next page]

(54) **Title:** INTERMEDIARY NETWORK ELEMENT FOR TAP TRAFFIC

(57) **Abstract:** In one implementation, a method for tap traffic redirection can include identification of a request protocol and a source network element, selection of a redirector protocol based on a list of device capabilities, instruction of the source network element to send a first copy of network traffic to an intermediary network element using the redirector protocol, and instruction of the intermediary network element to forward a second copy of the network traffic using the request protocol.

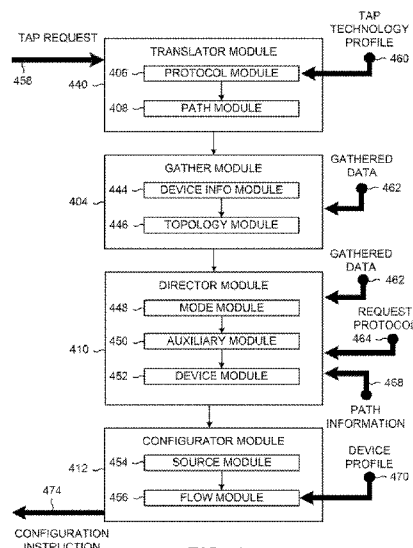


FIG. 4

**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))*

Intermediary Network Element for Tap Traffic

BACKGROUND

[0001] Computer networks commonly include switches, routers, and endpoints (such as servers or client devices) that are communicatively connected to transfer information between endpoints. Networks have evolved to include mobile computing devices, personal devices connected to business networks, virtual machines, and even virtual switches. Networks are commonly distributed across vendors of network devices and host multiple solutions, such as orchestration software and middle boxes, to meet daily customer requirements. As networks evolve, the importance of “network visibility” becomes very pertinent from the perspective of auditing, troubleshooting, and network data analytics. Network monitoring solutions can enable a network administrator to “tap” (i.e., copy and/or redirect) traffic at various tapping points in the network and use the tapped traffic for various analysis such as probing, auditing, and anomaly detection.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figures 1 and 2 are block diagrams depicting example network tap systems.

[0003] Figure 3 depicts an example environment in which various network tap systems can be implemented.

[0004] Figure 4 depicts example modules used to implement example network tap systems.

[0005] Figures 5 and 6 are flow diagrams depicting example methods for tap traffic redirection.

DETAILED DESCRIPTION

[0006] In the following description and figures, some example implementations of network tap apparatus, network tap systems, and/or methods for tap traffic redirection

are described. Network monitoring solutions come in a variety of offerings. Traditionally techniques like sample-based monitoring (e.g., SFLOW), statistics-based monitoring (e.g., NETFLOW), port mirroring (e.g., SPAN), etc. have been widely used for achieving network visibility objectives (e.g., traffic tapping objectives). Software-defined network ("SDN") environments provide opportunities for more precise tapping and traffic analytics because of the centralized visibility and capability to steer and/or copy traffic on a per-flow basis.

[0007] Networks can include network elements from multiple vendors. Network monitoring software can depend on a uniform tapping mechanism across the network. To accomplish this, the network administrator may configure additional firewall policies or other mechanism-dependent configuration for such network monitoring software.

[0008] Various examples described below relate to providing uniform tapping across a network comprising heterogeneous network elements via a federation of devices. By identifying the tap technology capabilities of the network elements of the federation, traffic can be steered and transferred using multiple tap technologies (e.g., multiple tap protocols). A network element can indirectly support a tap technology (that is not directly supported on the network element) by leveraging a federated network element that supports the requested tap technology. In this manner, the use of multiple protocols can create a uniformity among tap technologies.

[0009] A network element, as used herein, represents any appropriate device of a network, such as a switch, a router, a middle box, or an endpoint compute device. 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. Furthermore, the term "maintain" (and variations thereof) as used herein means "to create, delete, add, remove, access, update, and/or modify."

[0010] Figures 1 and 2 are block diagrams depicting example network tap systems 100 and 200. Referring to Figure 1, the example network tap system 100 of Figure 1 generally includes a data store 102, a gather engine 104, a protocol engine 106, a path engine 108, a director engine 110, and a configurator engine 112. In

general, the configurator engine 112 configures a network element to route tap traffic through an intermediary network element identified from the director engine 110 based on information gathered by the gather engine 104, the protocol selected by the protocol engine 106, and the source (e.g., a network element designated as a target of a tap request to copy and/or steer network traffic) identified by the path engine 108. The example network tap system 100 can include a format handler engine 114 to translate tap traffic in one protocol to another, such as in a centralized embodiment as discussed herein.

[0011] The gather engine 104 represents any circuitry or combination of circuitry and executable instructions to obtain a list of device capabilities and topology information. For example, the gather engine 104 can represent a combination of circuitry and executable instructions to identify network elements that are members of a federation and the capabilities of those members. The gather engine 104 can retrieve information about a network element by making a request to the network element directly (e.g., via port connections of the network element to a neighbor network element where a neighbor network element is a directly-connected network element) or by accessing the information from a store location, such as a database stored on the data store 102. For example, each network element of the network can support an exchange of the set of supported protocols at startup of the network element with connected switches and/or the network controller of the network. The retrieved network information (also discussed herein as gathered data) can include a list of devices, a list of device capabilities, and topology information. Information related to device capabilities include supported tap capabilities (e.g., tap protocols and/or tap technologies) and may include device specifications and/or configurations.

[0012] The protocol engine 106 represents any circuitry or combination of circuitry and executable instructions to select a first tap protocol based on a tap request. For example, the protocol engine 106 can represent a combination of circuitry and executable instructions to identify a tap technology to complete the tap request and select a protocol used by the tap technology to copy traffic. The tap protocol of the request can be identified directly from the request or by translating the request into a set of parameters to tapping (such as using a mapping of keywords).

[0013] The path engine 108 represents any circuitry or combination of circuitry and executable instructions to identify a source network element to tap based on the tap request. For example, the path engine 108 can represent a combination of circuitry and executable instructions to parse the tap request for a source to tap or translate the tap request from a keyword to a particular network element targeted for tapping. The path engine 108 can further represent circuitry or a combination of circuitry and executable instructions to identify a path between a destination network element and the source network element based on the first tap protocol, the list of device capabilities, and the topology information. The path can be identified based on a minimum number of element hops between a first network element supporting the first tap protocol and a second network element supporting a second tap protocol. The path is to include an intermediary network element that supports a request protocol. In particular, the path is to include an intermediary network element.

[0014] The path engine 108 can identify a plurality of paths that include an intermediary network element that supports the first tap protocol by looking up route information in a local routing table. The route table information, in conjunction with the data gathered by the gather engine 104, can be used to identify paths from the source to the destination that include a network element that supports the requested tap protocol. The plurality of paths can be compared for the best route, such as based on the number of hops (i.e., network segments) between elements. For example, the path engine 108 can identify a first path that includes a switch A that supports the requested tap protocol via six hops and a second path that includes a switch B that supports the requested tap protocol via four hops. In that example, the director engine 110 can select switch B as the intermediary network element to redirect traffic because the second path is the one of the plurality of paths which supports the requested protocol with the least number of element hops.

[0015] The director engine 110 represents any circuitry or combination of circuitry and executable instructions to identify a second tap protocol supported by the source network element and an intermediary network element that supports the first tap protocol. For example, the director engine 110 can represent a combination of circuitry and executable instructions to identify a second tap protocol usable with the source

network element based on a list of device capabilities and identify an intermediary network element having the first tap protocol based on the topology information. In a distributed mode, the director engine 110 can identify the intermediary network element that has the first tap protocol by using the topology information to identify which path is shortest and includes a network element that supports the requested tap protocol. For another example in distributed mode, the director engine 110 can identify the nearest network element that supports the requested tap protocol to the source network element and then the path engine 108 can identify a path from the nearest network element to a destination. The director engine 110 can identify multiple protocols to transfer tap traffic to the requested tap protocol. For example, the shortest path to a destination from a source may include converting the tap traffic at a first switch, converting the tap traffic again at a second switch, and then converting the tap traffic at a third switch to a protocol that is acceptable by the network element that can reproduce the tap traffic according to the requested tap protocol. In this manner, the director engine 110 and the path engine 108 can work together to identify a path that balances the path length and the number of conversions (i.e., changes of tap protocols) to an acceptable level of bandwidth utilization in order to permit tapping traffic of the network.

[0016] In a centralized embodiment, the director engine 110 can further represent circuitry or a combination of circuitry and executable instructions to cause a plurality of network elements of a tap domain to send a copy of network packets to the intermediary network element where the intermediary network element is to include a central database of tapped traffic of the tap domain (e.g., a network element discussed as a collector herein). In this manner, the director engine 110 can direct tap traffic to a central location via any appropriate protocol supported by the network element and convert the copy of the traffic to a format acceptable by the requested tap protocol. For example, a controller that includes the director engine 110 can configure a tap rule to send the received traffic to a collector (i.e., a network element designated to collect tap traffic) where the tap rule is a network flow rule.

[0017] In a centralized mode, the path identified is a path to a central collector and the tap traffic is steered to the central collector using a secondary protocol available to the source network element. Once at the intermediary network element (which may

be the central collector), the traffic may be converted into the tap protocol requested by the tap request or otherwise sent to the destination via the requested tap protocol. More detail regarding centralized mode and the distributed mode is further illustrated in the description regarding Figure 3.

[0018] The configurator engine 112 represents any circuitry or combination of circuitry and executable instructions to configure the source network element to direct tap traffic to an intermediary network element using a second tap protocol. For example, the flow table can be updated with a flow to redirect traffic using a particular protocol. The configurator engine 112 can adjust settings of a network element in accordance with the tap technology and can obtain the syntax of the protocol used by the tap technology being setup. For example, the manual changes that a network administrator would perform on a network element to adapt the network element to copy traffic in accordance with the tap request can be performed by the configurator engine 112. The configurator engine 112 can represent a combination of circuitry and executable instructions to identify a plurality of configuration settings of a network element, modify the plurality of configuration settings to utilize a tap protocol, and instruct the network element to direct tap traffic to an identified intermediary network element using a tap protocol that is supported by the identified intermediary network element. The configurator engine 112 can also configure the intermediary network element to convert received tap traffic from a first protocol to a second protocol and send the tap traffic using the second protocol on a path to a destination where the destination and the second protocol are specified by the tap request.

[0019] The format handler engine 114 represents any circuitry or combination of circuitry and executable instructions to generate tap traffic in a format used by a particular tap protocol. For example, the format handler engine 114 represents any circuitry or combination of circuitry and executable instructions to identify, based on a tap request, a log entry of a database of traffic associated with the source network element and convert the log entry to a format compatible with the protocol requested by the tap request. A format can be converted between protocols based on parsing a message in a first protocol (e.g., an initial protocol) to identify the relevant information to be transferred by a message in a second protocol (e.g., a final protocol). It is natural for

some information to be lost or unavailable due to the conversion between protocols. The format handler engine 114 is useful in a centralized environment where tap traffic is being collect to a centralized device and requests for the tap traffic are being made to the centralized device, where it is likely that the tap traffic is collected in a protocol format that may not be the protocol format being requested from the centralized device. In a centralized mode, the centralized device can include a mapping (e.g., a data structure, a rule, or a combination thereof) that associates protocols and identifies the rules to convert from one format to another. In that example, the mapping may be updated by a network administrator or autonomously as new tapping protocols are made available to the network or unused protocols are removed from the network tap configuration.

[0020] A mapping of keywords to primitives can include a knowledge base and/or a machine-learned store of terms that can represent network information, such as a network element primitive. For example, the mapping can contain a data structure associating the string "conference room A" with strings "switch A" and "switch B" (or the media access control ("MAC") addresses associated with the switches A and B, respectively), so the map of keywords can generate the primitives of switch A and switch B when a tap configuration associated with the conference room A is requested. A mapping can represent any data structure or combination of data structures to provide association between identifiers (e.g., numbers, characters, strings, etc.). Example data structures usable as a mapping include an associative array, a tree, a map, a dictionary, and the like. In general, the mappings are used for lookups of associative terms and can be adapted to specific use cases by a network administrator.

[0021] The data store 102 can contain information utilized by the engines 104, 106, 108, 110, 112, and 114. For example, the data store 102 can store a tap request, a copy of traffic of the network, a list of devices, a list of device capabilities, topology information, a plurality of tap protocols, a route table, a mapping, and the like.

[0022] Figure 2 depicts the example system 200 can comprise a memory resource 220 operatively coupled to a processor resource 222. The processor resource 222 can be operatively coupled to a data store 202. The data store 202 can be the same as the data store 102 of Figure 1.

[0023] Referring to Figure 2, the memory resource 220 can contain a set of instructions that are executable by the processor resource 222. The set of instructions are operable to cause the processor resource 222 to perform operations of the system 200 when the set of instructions are executed by the processor resource 222. The set of instructions stored on the memory resource 220 can be represented as a gather module 204, a protocol module 206, a path module 208, a director module 210, a configurator module 212, and a format handler engine 214. The gather module 204, the protocol module 206, the path module 208, the director module 210, the configurator module 212, and the format handler engine 114 represent program instructions that when executed function as the gather engine 104, the protocol engine 106, the path engine 108, the director engine 110, the configurator engine 112, and a format handler engine 114 of Figure 1, respectively. The processor resource 222 can carry out a set of instructions to execute the modules 204, 206, 208, 210, 212, 214 and/or any other appropriate operations among and/or associated with the modules of the system 200. For example, the processor resource 222 can carry out a set of instructions to identify a redirector protocol available to a source network element based on a list of device capabilities, identify a path including an intermediary network element that supports a request protocol based on topology information and the tap request, configure a plurality of network elements of the path to form a flow of traffic via the path using a subset of tap protocols, and instruct the intermediary network element to convert the tapped traffic received in a first protocol to a second protocol. For another example, the processor resource 222 can carry out a set of instructions to identify a subset of tap protocol through which the request protocol could be achieved based on the tap request and network information (e.g., topology information and a list of device capabilities) and steer tapped traffic to an intermediary network element that supports the request protocol. For yet another example, the processor resource 222 can carry out a set of instructions to configure a network flow rule to send the received traffic to a collector (i.e., an intermediary network element that stores copies of traffic), distribute the flow rule to a network element, and instruct a destination network element to request traffic associated with the tap request from the collector in the requested tap protocol format.

[0024] Although these particular modules and various other modules are illustrated and discussed in relation to Figure 2 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 2 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. Figure 4 depicts yet another example of how functionality can be organized into modules.

[0025] The processor resource 222 is any appropriate circuitry capable of processing (e.g., computing) instructions, such as one or multiple processing elements capable of retrieving instructions from the memory resource 220 and executing those instructions. For example, the processor resource 222 can be a central processing unit ("CPU") that enables tap traffic redirection by fetching, decoding, and executing modules 204, 206, 208, 210, 212, and 214. 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 222 can include multiple processing elements that are integrated in a single device or distributed across devices. The processor resource 222 can process the instructions serially, concurrently, or in partial concurrence.

[0026] The memory resource 220 and the data store 202 represent a medium to store data utilized and/or produced by the system 200. The medium is any non-transitory medium or combination of non-transitory mediums able to electronically store data, such as modules of the system 200 and/or data used by the system 200. 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 220 can be said to store program instructions that when executed by the processor resource 222 cause the processor resource 222 to implement functionality of the system 200 of Figure 2. The memory resource 220 can be integrated in the same device as the processor resource 222 or it can be separate but accessible to that device and the processor resource 222. The memory resource 220 and/or the data store 202 can be distributed across devices. The memory resource 220 and the data store 202 can represent the same physical medium or separate physical mediums. The data of the data store 202 can include representations of data and/or information mentioned herein.

[0027] In the discussion herein, the engines 104, 106, 108, 110, 112, and 114 of Figure 1 and the modules 204, 206, 208, 210, 212, and 214 of Figure 2 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 2, the executable instructions can be processor-executable instructions, such as program instructions, stored on the memory resource 220, which is a tangible, non-transitory computer-readable storage medium, and the circuitry can be electronic circuitry, such as processor resource 222, for executing those instructions. The instructions residing on the memory resource 220 can comprise any set of instructions to be executed directly (such as machine code) or indirectly (such as a script) by the processor resource 222.

[0028] In some examples, the system 200 can include the executable instructions can be part of an installation package that when installed can be executed by the processor resource 222 to perform operations of the system 200, such as methods described with regards to Figures 4-7. In that example, the memory resource 220 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 service device 334 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 220 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 220 include static RAM ("SRAM"), dynamic RAM ("DRAM"), electrically erasable programmable ROM ("EEPROM"), flash memory, or the like. The memory resource 220 can include integrated memory such as a hard drive ("HD"), a solid state drive ("SSD"), or an optical drive.

[0029] Figure 3 depicts example environments in which various example network tap systems can be implemented. The example environment 390 is shown to include an example system 300 for tap technology selection. The system 300 (described herein with respect to Figures 1 and 2) can represent generally any circuitry or combination of circuitry and executable instructions to select a tap technology based on selection criteria. The system 300 can include a gather engine 304, a protocol engine 306, a path engine 308, a director engine 310, and a configurator engine 312 that are the same as the gather engine 104, the protocol engine 106, the path engine 108, the director engine 110, and the configurator engine 112 of Figure 1, respectively, and the associated descriptions are not repeated for brevity. The system 300 can also include a format handler engine 314 that is the same as the format handler engine 114 of Figure 1. As shown in Figure 3, the data store 302 (which is the same as data stores 102 of Figure 1 and 202 of Figure 2) and the engines 304, 306, 308, 310, 312, and 314 can be integrated into a compute device, such as an intermediary device 342 or a switch 338, or distributed across devices. The engines 304, 306, 308, 310, 312, and 314 can be integrated via circuitry or as installed instructions into a memory resource of the compute device.

[0030] The example environment 390 can include compute devices that are elements of the network, such as a controller 332, service devices 334, user devices 336, switches 338, wireless access point device 340, and an intermediary device 342. The controller 332 represents generally any compute device capable of managing the network. For example, the controller 332 can be a SDN controller that includes circuitry (or a combination of circuitry and executable instructions) to manage flows of traffic through the network 330 and execute a tap application, such as tap application to be a destination of the tap traffic (e.g., a network monitoring application). The service devices 334 represent generally any compute devices to respond to a network request received from a user device 336, whether virtual or real. For example, the service device 334 can

operate a combination of circuitry and executable instructions to provide a network packet in response to a request for a page or functionality of an application. The user devices 336 represent generally any compute devices to communicate a network request and receive and/or process the corresponding responses. For example, a browser application may be installed on the user device 336 to receive the network packet from the service device 334 and utilize the payload of the packet to display an element of a page via the browser application. The switches 338 represent any network device capable of connecting devices of the network together. For example, a switch 338 can retain local routing information of paths describing how to transfer packets from a source to a destination.

[0031] In Figure 3, the dotted lines represent logical connections between devices (such as between the controller 332, the intermediary device 342, and the switches 338) which may or may not be physically connected directly. For example, the switches 338 connected via dotted lines to the controller 332 can represent switches 338 selected as part of the tap domain to copy traffic to the controller 332, where the tap traffic is routed on a path through an intermediary device 342 rather than directly to the controller 332. In another example, the dotted lines may be between multiple switches where a switch can convert or otherwise redirect tapped traffic using the request protocol rather than redirecting the tapped traffic through a dedicated middle box, such as a tap sink device. The controller 332 can include circuitry to execute a tap application, such as a tap application for network monitoring that receives tap traffic (i.e., is a destination of the tap traffic). As mentioned, the intermediary device 342 can be a tap sink device, such as an intrusion detection system or an intrusion prevention system. For example, the intermediary device 342 can be a tap sink device including circuitry to execute a collector application to receive the tap traffic and perform analysis on the tap traffic.

[0032] In a distributed mode, the path engine 308 and the director engine 310 can work in conjunction to identify a path with a minimum number of hops (i.e., network segments that represent connections between devices) between a source and a destination that include a device that supports the tap request protocol. In a centralized mode, the path engine 308 identified the source and the path to the central collection

device, represented as device 342 of Figure 3. In this manner, all tapped traffic can be sent to a common collector of tap traffic. If the intermediary network element 342 is a collector device, then the intermediary network element 342 can include a format handler engine 314 that can format any data collected to a requested format known to the collector, such as the format of the protocol requested by the tap request. In an example of either mode, a first segment can provide tapped traffic in a first format (e.g., a format of a first protocol of a subset of tap protocols used along a path) that is different from a second format (e.g., a format of a second protocol of the subset of tap protocols used along the path) associated with a second segment.

[0033] The compute devices can be located on separate networks 330 or part of the same network 330. The example environment 390 can include any appropriate number of networks 330 and any number of the networks 330 can include a cloud compute environment, such as a virtual shared pool of compute resources. For example, networks 330 can be distributed networks comprising virtual computing resources. Any appropriate combination of the system 300 and compute devices can be a virtual instance of a resource of a virtual shared pool of resources. The engines and/or modules of the system 300 herein can reside and/or execute “on the cloud” (e.g., reside and/or execute on a virtual shared pool of resources).

[0034] A link 328 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 link 328 can include, at least in part, intranet, the Internet, or a combination of both. The link 328 can also include intermediate proxies, routers, switches, load balancers, and the like.

[0035] Referring to Figures 1-3, the engines 104, 106, 108, 110, 112, and 114 of Figure 1 and/or the modules 204, 206, 208, 210, 212, 214 of Figure 2 can be distributed across devices 332, 334, 336, 338, 340, 342, 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 director engine 310 of Figure 3 can request, complete, or perform the methods or operations described with the director engine 110 of Figure 1 as well as the gather engine 104, the protocol

engine 106, the path engine 108, and the configurator engine 112 of Figure 1. Thus, although the various engines and modules are shown as separate engines in Figures 1 and 2, 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 300 can perform example methods described in connection with Figures 4-7.

[0036] Figure 4 depicts example modules used to implement example network tap systems. Referring to Figure 4, the example modules of Figure 4 generally include a translator module 440, a gather module 404, a director module 410, and a configurator module 412. The gather module 404, the director module 410, and the configurator module 412 function similar to the gather module 204, the director module 210, and the configurator module 212 of Figure 2. The example modules of Figure 4 can be implemented on any appropriate compute device, such as a network switch, an SDN controller, or a tap sink.

[0037] The tap request 458 can activate the translator module 440 to cause a processor resource to translate the tap request 458, identify a source network element, and identify a request protocol. The translator module 440 can include program instructions, such as a protocol module 406 and a path module 408, to facilitate identification of the parameters of the tap request. The protocol module 406 and the path module 408 represent the same modules as the protocol module 206 and the path module 208 of Figure 2, respectively, and, for brevity, the associated descriptions are not repeated. For example, the processor resource that executes the protocol module 406 can utilize a tap technology profile 460 to identify a request protocol.

[0038] The gather module 404 can include program instructions, such as a device information module 444 and a topology module 446, to gather data 462 from the network. The device information module 444 represents program instructions that when executed cause a processor resource to retrieve and/or generate information associated with network elements of the network. For example, a processor resource that executes the device information module 444 can retrieve a list of devices and their capabilities via individual device retrieval or by querying a database of device profiles. The topology module 446 represents program instructions that when executed cause a

processor resource to retrieve and/or generate topology information associated with the network elements of the network.

[0039] The director module 410 can include program instructions, such as a mode module 448, an auxiliary module 450, and a device module 452, to enable tap traffic to divert to a network element that can complete a tap request. The mode module 448 represents program instructions that when executed cause a processor resource to identify whether the network supports a distributed mode of operation or a centralized mode of operation. For example, if the processor resource executing the gather module 404 discovers a device is designated as a collector, then tap request can be completed based on a centralized mode of operation. The auxiliary module 450 represents program instructions that when executed cause a processor resource to identify a tap protocol available to the source network element that can be used to transfer the tapped traffic to an intermediary network element. The device module 452 represents program instructions that when executed cause a processor resource to select a network element that supports both the redirector protocol and the request protocol. For example, the intermediary network element is to receive traffic using the redirector protocol and send traffic using the request protocol. A processor resource executing the device module 452 can utilize the request protocol 464 and path information 468 to identify the intermediary network element to use for redirecting the tapped traffic.

[0040] The configurator module 412 can include program instructions, such as a source module 454 and a flow module 456, to provide configuration details to a network element. The source module 454 represents program instructions that when executed cause a processor resource to identify a configuration profile for the source network element to provide the tap traffic using the redirector protocol. The flow module 456 represents program instructions that when executed cause a processor resource to generate instructions to provide a forwarding rule to the network element based on a device profile 470 of the network element. A processor resource executing the configurator module 412 can generate a configuration instruction 474 to send to the network element to redirect the tapped traffic to the intermediary network element using the redirector protocol. The configuration instruction 474 can be sent to the source network element to configure the source network element to tap traffic according to the

bounds of the tap request. Another configuration instruction can be sent to the intermediary network element to configure the intermediary network element to forward, using the requested tap protocol, tap traffic from the source network element. In this manner, the tap domain can be configured with a network tap configuration that can produce uniform tapping information by using an intermediary protocol to convert tapped traffic into the request protocol.

[0041] Figures 5-7 are flow diagrams depicting example methods for tap traffic redirection. Referring to Figure 5, example methods for tap traffic redirection can generally comprise identifying a request protocol and a source network element based on a tap request, selecting a redirector protocol based on a list of device capabilities, instructing the source to send a first copy of network traffic to an intermediary network element using the redirector protocol, and instructing the intermediary network element to forward a second copy of the network traffic using the request protocol.

[0042] At block 502, a request protocol and a source network element are identified. The request protocol and the source network element are identified from a tap request. For example, the request protocol and the source network element can be directly found as part of the tap request. For another example, the tap request can be translated (e.g., from phrases to identifiable keywords of a mapping) to discover the request protocol and the source network element.

[0043] At block 504, a redirector protocol is selected. The redirector protocol is one of the supported protocols of the source network element. For example, a list of device capabilities comprising a list of tap capabilities of the source network element can be retrieved from the source network element or a database of device capabilities, such as a database of device profiles of each network element of the network. The redirector protocol can be selected based on the tap request, such as selecting the redirector protocol that can transfer a sufficient amount of information to be with (e.g., converted to) the request protocol.

[0044] At block 506, the source network element is instructed to send a first copy of network traffic to an intermediary network element using the redirector protocol. For example, a controller can distribute a flow entry to the source network element to update the local flow table of the source network element to send duplicate traffic that

matches a particular field or set of fields of a packet header where the duplicated packet is structured according to the redirector protocol with path information to guide the duplicated packet to a particular intermediary network element that supports the request protocol. At block 508, the intermediary network element is instructed to forward a second copy of network traffic using the request protocol. For example, a controller can distribute a flow entry to the intermediary network element to update the local flow table of the intermediary network element to transfer the information of the duplicated packet originating from the source network element into a second packet structured in the requested protocol and forward that second packet to the destination for tapping analysis (i.e., network monitoring purposes).

[0045] Figure 6 includes blocks similar to blocks of Figure 5 and provides additional blocks and details. In particular, Figure 6 depicts additional blocks and details generally regarding gathering network element information and identifying a subset of network elements that supports the request protocol. Blocks 604, 606, 610, and 612 are the same as blocks 502, 504, 506 and 508 of Figure 5 and, for brevity, their respective descriptions are not repeated.

[0046] At block 602, a list of device capabilities and topology information is gathered. Information about network elements can be gathered in a variety of ways, such as through a centralized database, a network management source, application programming interfaces, and network monitoring to name a few. Figure 7 depicts an appropriate example of obtaining network information.

[0047] At block 608, a subset of network elements that supports the request protocol based on the topology information. Using data gathered at block 602, the network elements of the network that are capable of supporting the requested tap protocol are identified and placed in the subset. The subset can be narrowed to network elements that can receive the tapped traffic in a protocol supported by the source network element. The identification and selection of one of the subset to receive redirected traffic can be based on the closest network element to the source (e.g., least number of hops) and/or based on other thresholds or bounds set by a network administrator.

[0048] Figure 7 includes blocks similar to blocks of Figure 6 and provides additional blocks and details. In particular, Figure 7 depicts additional blocks and details generally regarding gathering network element information and directing tap traffic based on an operation mode. Blocks 702, 704, and 706 are similar to blocks 602, 604, and 606 of Figure 6, and, for brevity, their respective descriptions are not repeated in their entirety.

[0049] The collection of devices, device capabilities, and topology information of block 702 can include registration of network elements, identification of tap capabilities, and generation of topology information. At block 710, a plurality of network elements are registered with a centralized registration database. Gathering data can be facilitated by configuring network elements to register and update a database with their device profiles including their configuration information and/or state information. At block 712, a set of tap capabilities is identified. For example, each of the device profiles can contain information about the tap technologies supported by the device and/or the device profiles can be analyzed for attributes that meet the specifications of the tap technology, such as by comparing the attributes of the device profile to parameters of a tap technology profile. At block 714, topology information is generated based on the connections among the plurality of network devices and the set of tap capabilities. For example, the device profiles and/or routing information of each network element can be analyzed (or directly contain) the connections of each device to construct a topological map of the network element connections where such a map is further organized based on tap capabilities of each of the network elements. This can allow for dynamic construction of paths that are compatible for a given tap protocol.

[0050] At block 708, tap traffic is directed based on an operation mode. At block 718, it is decided whether the operation mode is a centralized mode or a distributed mode. For example, the existence of a collector device may support a centralized mode of operation.

[0051] If the operation mode is central, then the source network element is instructed to send a network copy to a collector device at block 720. At block 722, the data to be transferred to the request protocol is identified and converted to the format of

the request protocol. The converted data is sent to a destination based on the tap request at block 724.

[0052] If the operation mode is distributed, then a network element set (e.g., a subset of network elements of the network) is identified comprising a minimum number of network elements to support the request protocol at block 726. The subset of network elements can be part of a federated model where each member of the subset is also a member of the federation. At block 728, the source network element is instructed to send a copy of network traffic via a path comprising the subset identified at block 726 using the redirector protocol supported by the source network element. An intermediary network element is included in the subset identified at block 726. At block 730, the intermediary network element is instructed to forward the copy of network traffic using the request protocol rather than the redirector protocol. In either mode, the tapped traffic can arrive at the destination network element (e.g., a destination device and/or application) in the protocol requested by the tap request even though the source network element that originates the tap traffic does not support the request protocol directly.

[0053] Although the flow diagrams of Figures 4-7 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.

[0054] 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 network tap system comprising:
 - 2 a gather engine to obtain a list of device capabilities and topology information;
 - 3 a protocol engine to select a first tap protocol based on a tap request;
 - 4 a path engine to identify a source network element to tap based on the tap
 - 5 request; and
 - 6 a director engine to:
 - 7 identify a second tap protocol usable with the source network element based
 - 8 on the list of device capabilities; and
 - 9 identify an intermediary network element having the first tap protocol based
 - 10 on the topology information; and
 - 11 a configurator engine to configure the source network element to direct tap traffic
 - 12 to the intermediary network element using the second tap protocol.
- 1 2. The system of claim 1, wherein the path engine is further to:
 - 2 identify a path between a destination network element and the source network
 - 3 element based on the first tap protocol, the list of device capabilities, and the
 - 4 topology information, where the path includes the intermediary network element.
- 1 3. The system of claim 2, wherein:
 - 2 the gather engine is further to:
 - 3 retrieve a list of device capabilities from a neighbor network element via a
 - 4 port connection; and
 - 5 the path engine is further to:
 - 6 look up route information in a local routing table to identify a plurality of
 - 7 paths that include the intermediary network element that supports the first tap
 - 8 protocol; and
 - 9 compare a number of hops of each of the plurality of paths;

10 wherein the identified path is one of the plurality of paths that has a
11 minimum number of element hops between a first network element supporting
12 the first tap protocol and a second network element supporting the second tap
13 protocol.

1 4. The system of claim 1, wherein the director engine is further to:

2 cause a plurality of network elements of a tap domain to send a copy of network
3 packets to the intermediary network element, the intermediary network element to
4 include a central database of tapped traffic of the tap domain.

1 5. The system of claim 4, comprising:

2 a format handler engine to:

3 identify a log entry of the database associated with the source network
4 element based on the tap request; and

5 convert the log entry to a format compatible with the first protocol.

1 6. The system of claim 1, comprising at least one of:

2 a controller including circuitry to execute a tap application, the tap application to
3 be a destination of the tap traffic; and

4 a tap sink device including circuitry to execute a collector application to receive
5 the tap traffic, the tap sink device to be the intermediary network element.

1 7. A computer readable storage medium comprising a set of instructions executable by
2 a processor resource to:

3 identify a redirector protocol available to a source network element based on a
4 list of device capabilities, the source network element designated as a target to tap
5 network traffic;

6 identify a path including an intermediary network element that supports a request
7 protocol based on topology information and the tap request, the path to comprise a
8 plurality of network elements;

9 configure the plurality of network elements of the path to form a flow of traffic via
10 the path using a subset of protocols, a first segment to provide tapped traffic in a first
11 format of a first protocol of the subset of tap protocols that is different from a second
12 protocol of the subset of tap protocols associated with a second segment; and
13 instruct the intermediary network element to convert the tapped traffic received in
14 the first protocol to the second protocol.

1 8. The medium of claim 7, wherein the set of instructions is executable by the
2 processor resource to:

3 configure a tap rule to send the received traffic to a collector, the tap rule to be a
4 network flow rule and the collector to be a network element of the tap domain.

1 9. The medium of claim 7, wherein the set of instructions is executable by the
2 processor resource to:

3 identify the subset of tap protocols through which the request protocol could be
4 achieved, the subset of tap protocols to include a redirector protocol to steer the
5 tapped traffic to the intermediary network element that supports the request protocol.

1 10. The medium of claim 7, wherein the medium and the processor resource are
2 coupled to a software-defined network ("SDN") controller.

1 11. A method for tap traffic redirection comprising:

2 identifying a request protocol and a source network element based on a tap
3 request;

4 selecting a redirector protocol based on a list of device capabilities of a plurality
5 of network elements, the plurality of network elements comprising the source
6 network element;

7 instructing the source network element to send a first copy of network traffic to an
8 intermediary network element using the redirector protocol; and

9 instructing the intermediary network element to forward a second copy of the
10 network traffic using the request protocol.

1 12. The method of claim 11, comprising:

2 identifying a subset of network elements of a tap domain that support the request
3 protocol based on topology information associated with the tap domain, the tap
4 domain comprising heterogeneous network elements.

1 13. The method of claim 12, comprising:

2 identifying a minimum set of network elements of a tap domain through which the
3 requested taping mechanism could be supported based on topology information
4 associated with the tap domain.

1 14. The method of claim 11, comprising:

2 identifying a format of the request protocol;
3 converting data of the first copy of the network traffic into the identified format;
4 and
5 sending the converted data to a destination network element based on the tap
6 request, the second copy of the network traffic comprising the converted data.

1 15. The method of claim 11, comprising:

2 registering a plurality of network elements with a centralized registration
3 database;
4 identifying a set of tap capabilities of the plurality of network elements; and
5 generating topology information based on connections among the plurality of
6 network elements and the set of tap capabilities.

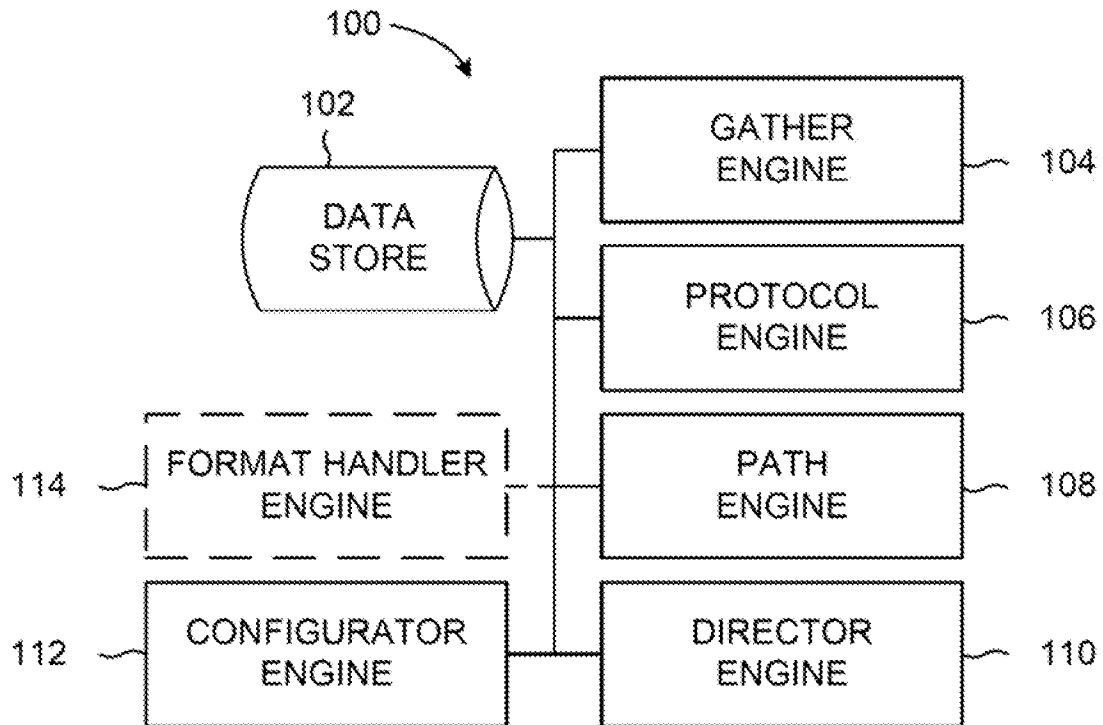


FIG. 1

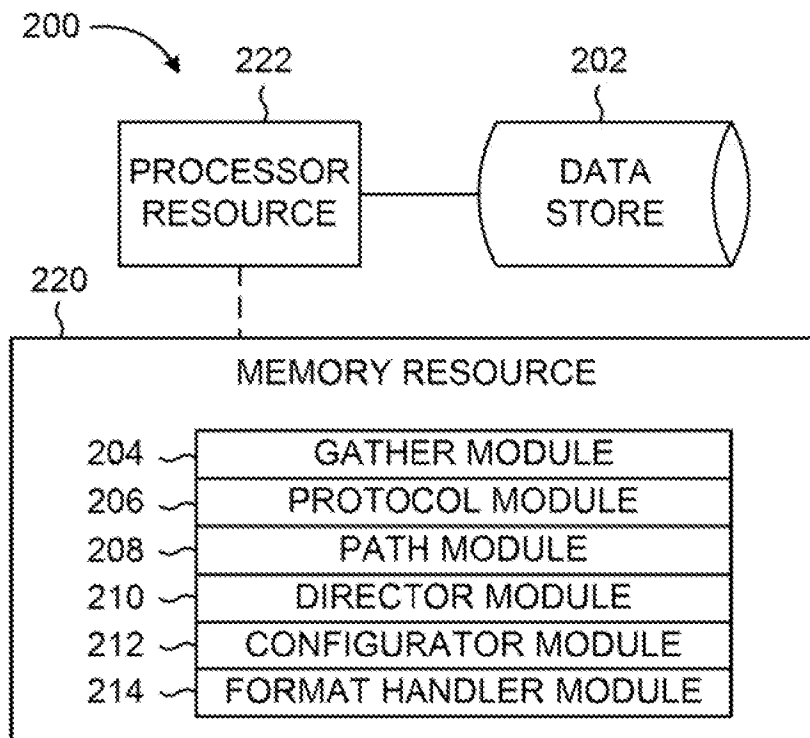


FIG. 2

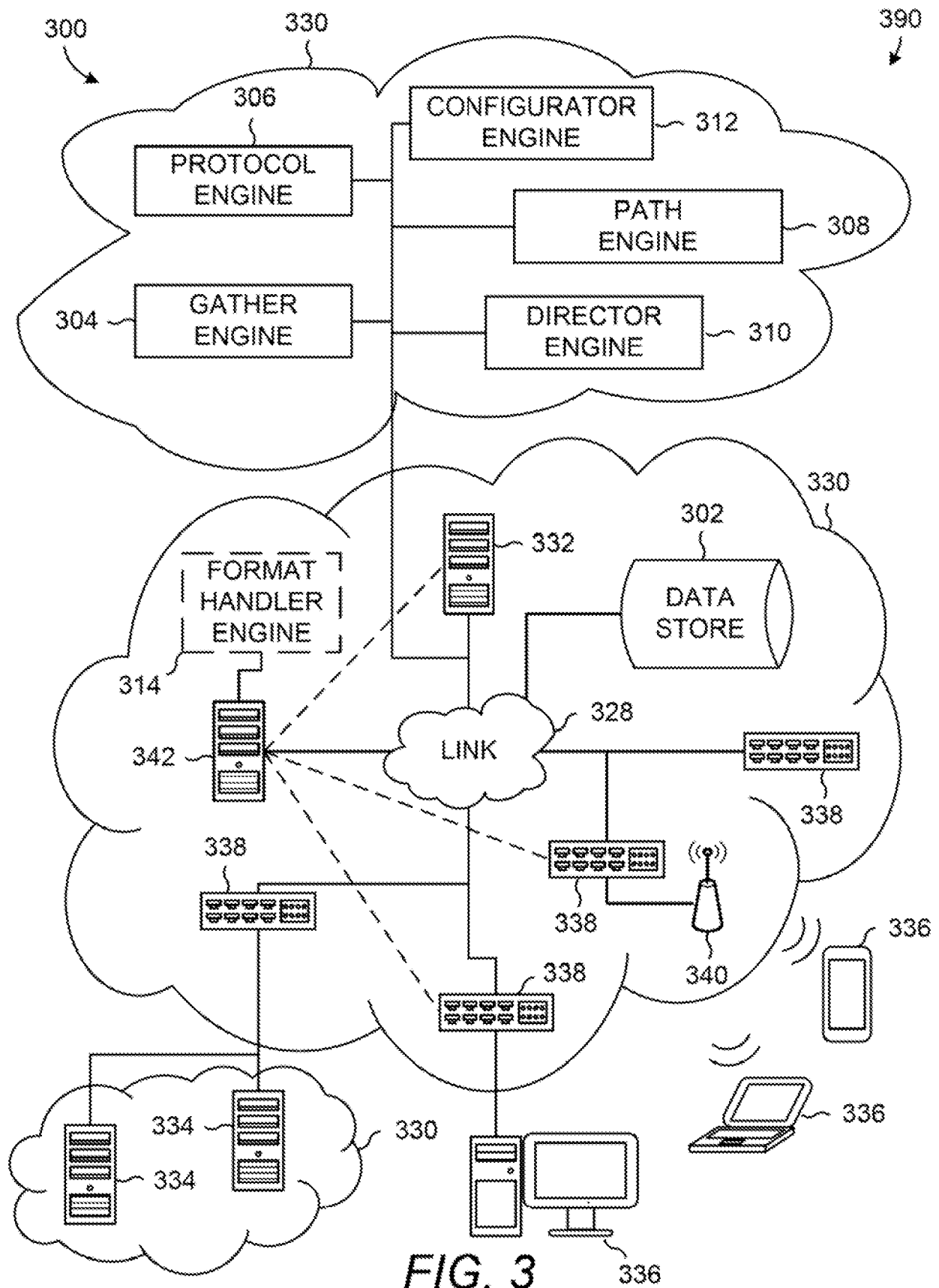


FIG. 3

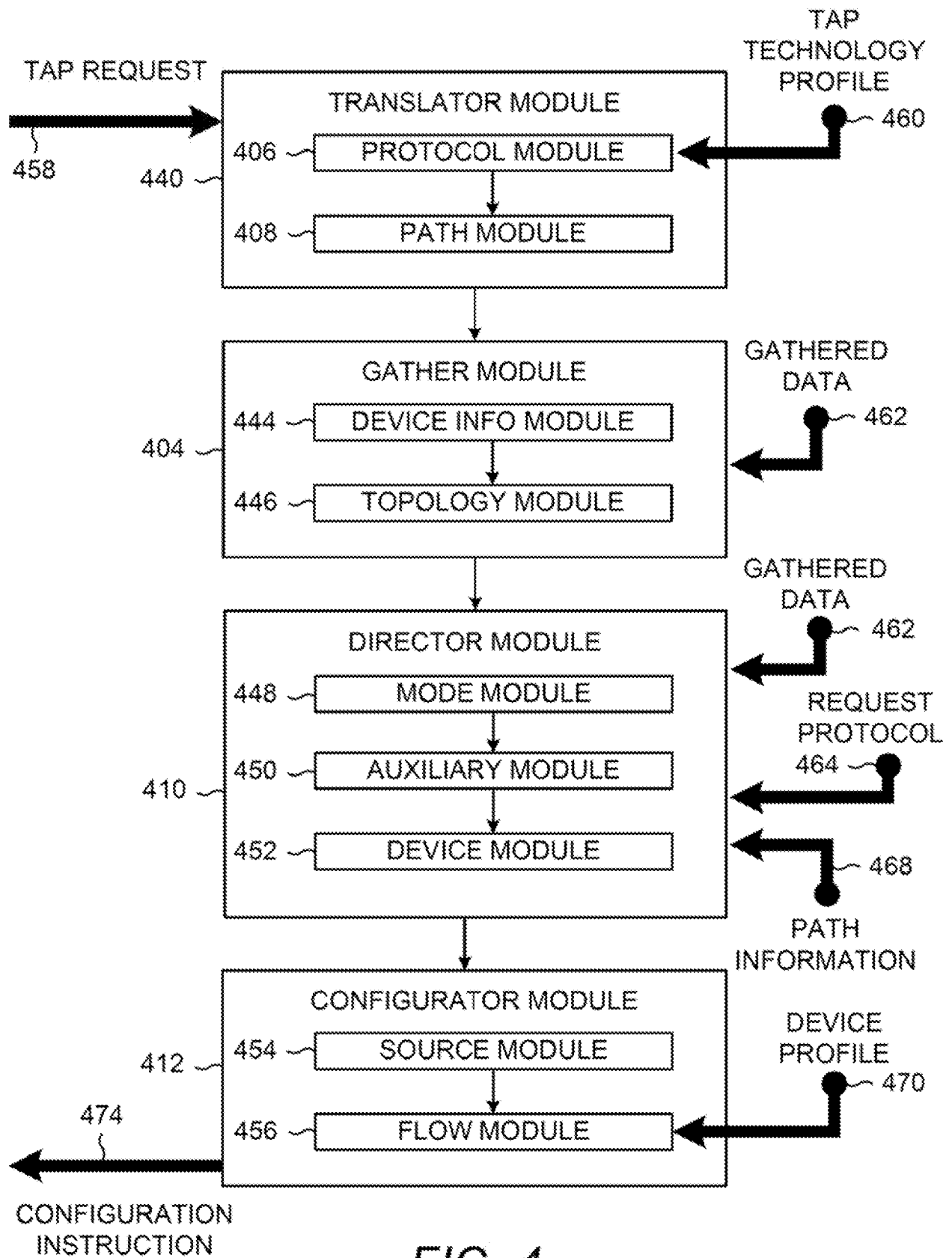
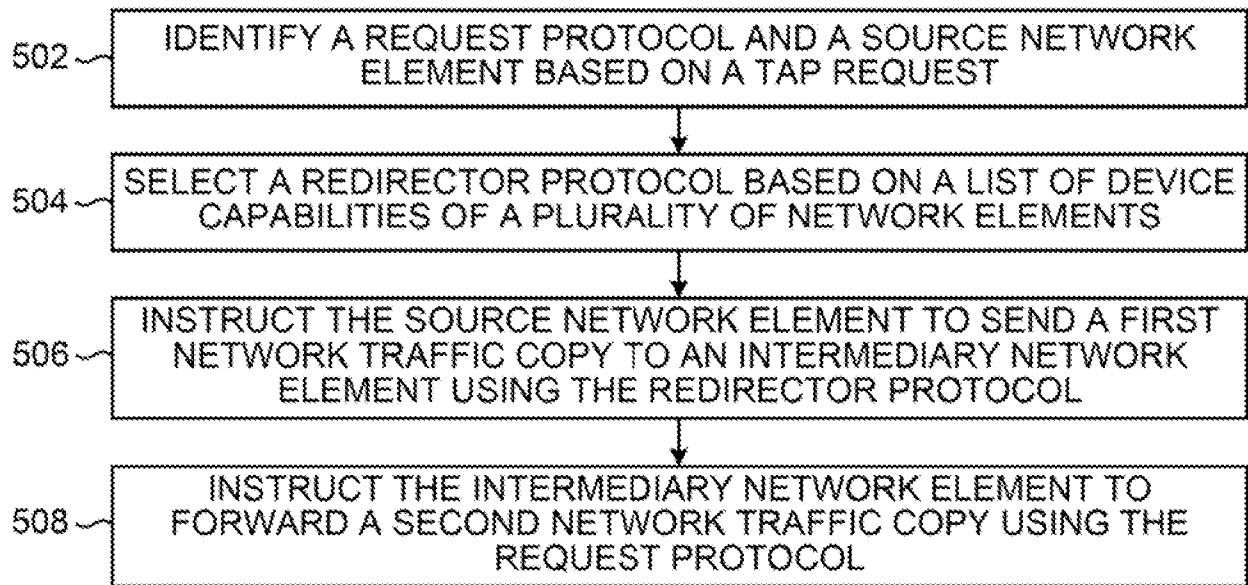
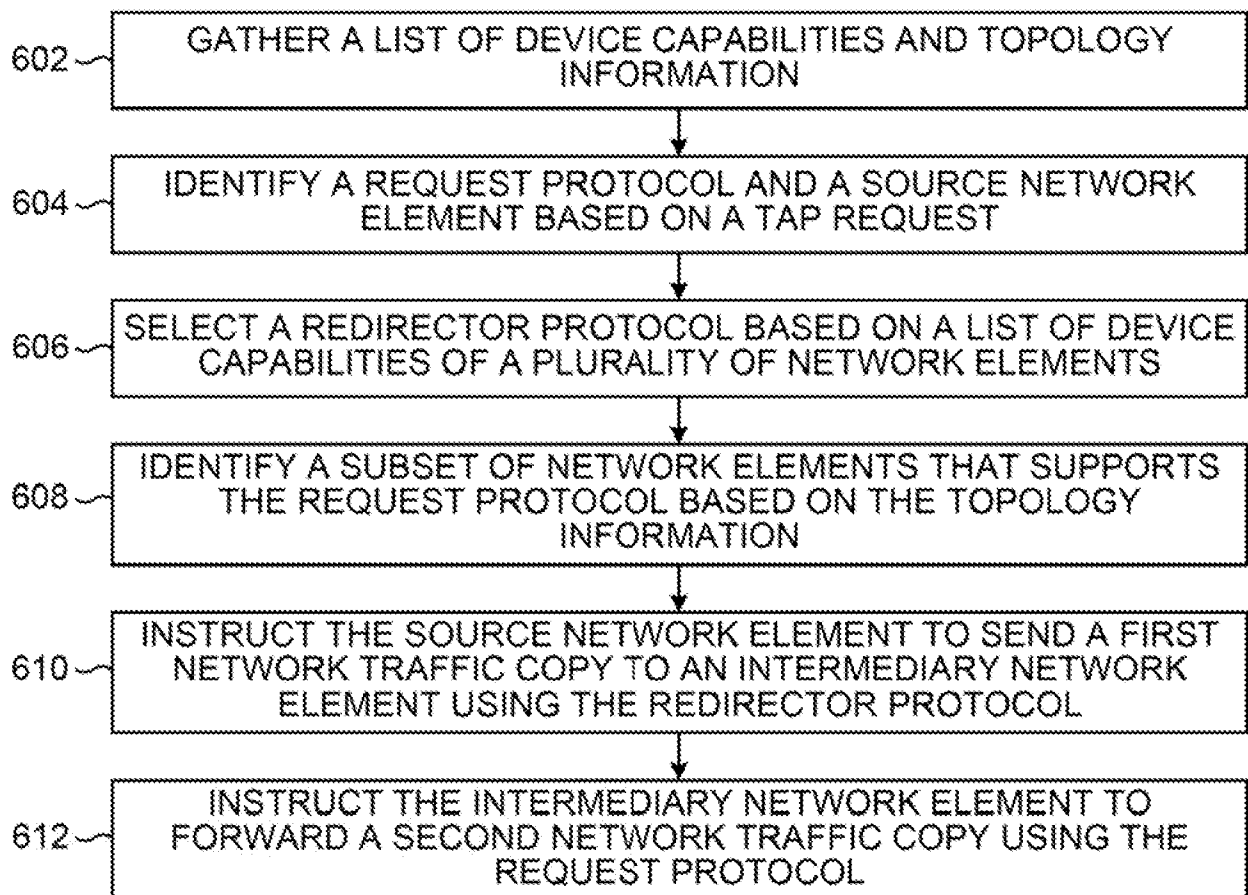


FIG. 4

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

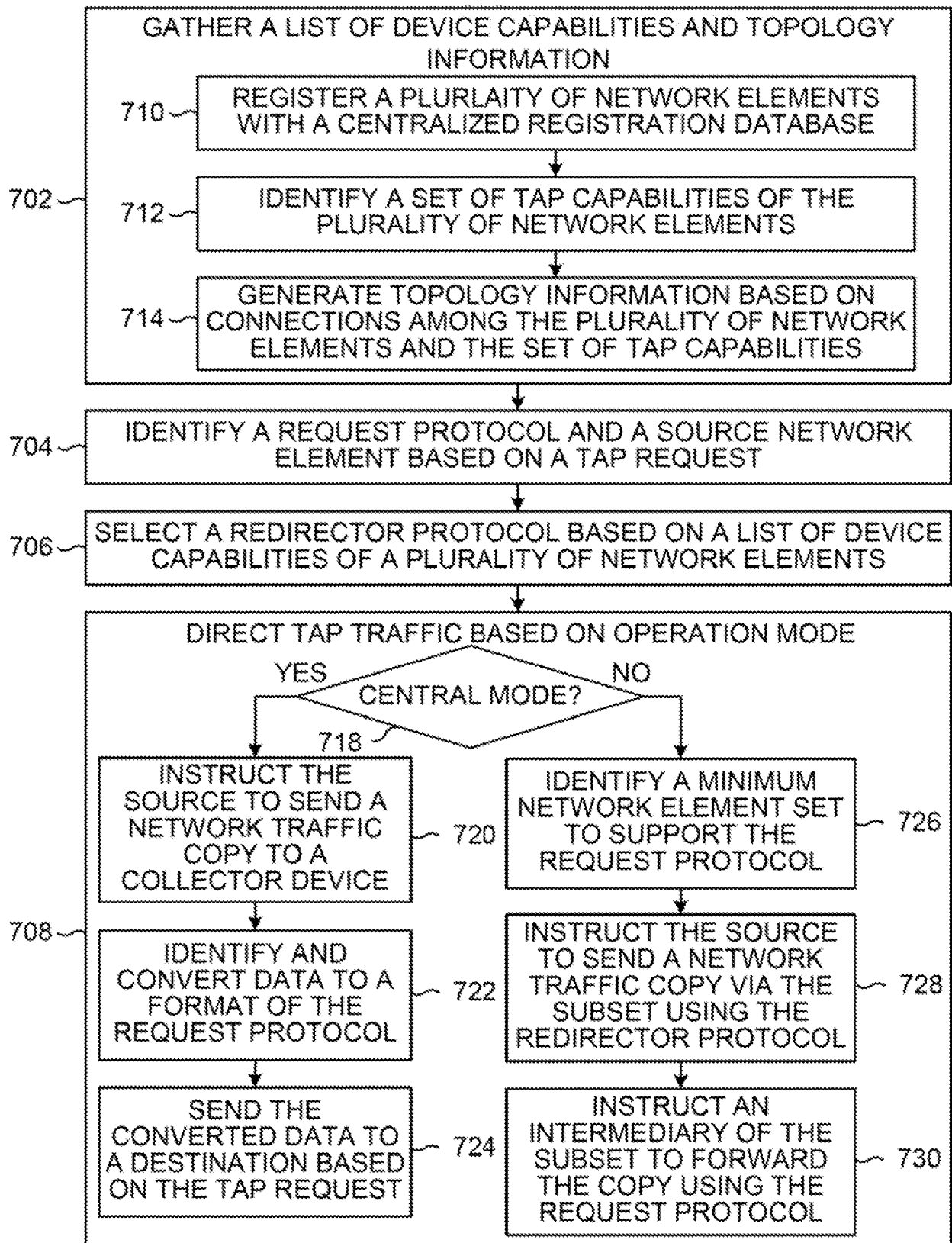


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/024102**A. CLASSIFICATION OF SUBJECT MATTER****H04L 29/08(2006.01)i, H04L 12/70(2013.01)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 29/08; G06F 15/16; G06F 15/173; H04Q 1/24; H04B 10/27; H04L 9/00; H04B 10/07; H04L 12/56; H04L 12/70

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: tap, request, translate, keyword, protocol, path, intermediary, redirect, network

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 8612570 B1 (MANOJ NAIR et al.) 17 December 2013 See column 3, line 34; column 6, lines 37-46; column 11, lines 48-56; column 15, lines 1-21; and figures 1A-1B.	1-15
A	WO 2006-062961 A2 (AKONIX SYSTEMS, INC. et al.) 15 June 2006 See page 9, lines 11-16; page 12, lines 18-22; and figure 3.	1-15
A	US 06141690 A (LYLE A. WEIMAN) 31 October 2000 See column 2, lines 56-63; column 4, lines 29-42; and figure 1.	1-15
A	US 05754532 A (ROGER DEV et al.) 19 May 1998 See column 3, line 59 - column 4, line 64; column 5, lines 17-25; and figure 1.	1-15
A	KR 10-2005-0104106 A (RPA NETWORKS CO., LTD. et al.) 02 November 2005 See paragraphs [0012]-[0019]; and figures 1-2.	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

23 September 2015 (23.09.2015)

Date of mailing of the international search report

27 October 2015 (27.10.2015)

Name and mailing address of the ISA/KR

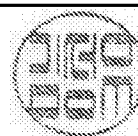
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/024102

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8612570 B1	17/12/2013	None	
WO 2006-062961 A2	15/06/2006	AU 2003-239220 A1 CA 2488731 A1 EP 1552414 A1 EP 1820293 A2 JP 2005-529409 A US 2004-0088423 A1 US 2004-0103318 A1 US 2004-0109518 A1 US 2004-0111623 A1 US 2004-0136386 A1 US 2007-0124577 A1 US 2008-0196099 A1 US 2008-0256257 A1 US 2011-0131653 A1 US 7428590 B2 US 7657616 B1 US 7664822 B2 US 7707401 B2 US 7774832 B2 US 7818565 B2 US 7882265 B2 US 8195833 B2 WO 03-105015 A1 WO 2006-062961 A3	22/12/2003 18/12/2003 13/07/2005 22/08/2007 29/09/2005 06/05/2004 27/05/2004 10/06/2004 10/06/2004 15/07/2004 31/05/2007 14/08/2008 16/10/2008 02/06/2011 23/09/2008 02/02/2010 16/02/2010 27/04/2010 10/08/2010 19/10/2010 01/02/2011 05/06/2012 18/12/2003 08/03/2007
US 06141690 A	31/10/2000	JP 11-112575 A JP 3471622 B2	23/04/1999 02/12/2003
US 05754532 A	19/05/1998	EP 0803157 A2 EP 0803157 B1 JP 10-512413 A US 05627819 A WO 96-21982 A3	11/08/1999 15/03/2000 24/11/1998 06/05/1997 03/10/1996
KR 10-2005-0104106 A	02/11/2005	None	