

(51) International Patent Classification:
H04L 12/813 (2013.01)(21) International Application Number:
PCT/US2013/073130(22) International Filing Date:
4 December 2013 (04.12.2013)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: HEWLETT PACKARD DEVELOPMENT
COMPANY, L.P. [US/US]; 11445 Compaq Center Drive
West, Houston, TX 77070 (US).(72) Inventors: STIEKES, Bryan; 31754 Day Lily, Brown-
stown Township, MI 48173 (US). TANDEL, Sebastien;
Av. Ipiranga, 6.681-predio 91a, 90619900 Porto Alegre,
Rio Grande Do Sul (BR). CHIU, Jechun; 14231 Tandem
Blvd, Austin, TX 78728 (US).(74) Agents: GARDINER, Austin, W. et al.; Hewlett-packard
Company, Intellectual Property Administration, 3404 East
Harmony Road, Mail Stop 35, 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, LT, 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, 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: POLICY RULE BASED ON A REQUESTED BEHAVIOR

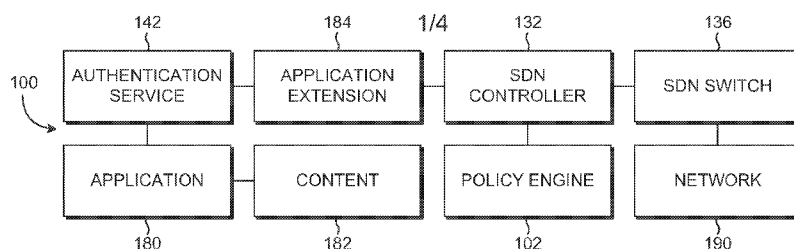


FIG. 1

(57) Abstract: In one implementation, an example system may include a policy engine. The policy engine may receive a context of an application to request a set of network traffic and provide a policy rule to a network device of a network path. In another implementation, an example system may identify a party and a requested behavior of the service and maintain a context to determine a policy rule to regulate a set of network traffic associated with the service based on the party and the requested behavior. In another implementation, an example method may comprise receiving a service request and an authentication, identifying a party, identifying a behavior, and deploying a policy to a network device of a network based on the party and the behavior.



Policy Rule Based on a Requested Behavior

BACKGROUND

[0001] A network may commonly be accessed through a network perimeter, or network edge, to request access to a service. A common approach to secure the network edge may include deployment of statically configured firewall devices. The firewall devices may be configured to permit communication between the maximal set of potential services to a source of request. For example, an internet protocol ("IP") address may be designated as an allowable source and the network may use a protocol port associated with the IP address to determine if traffic is allowable. Commonly, the traffic may be inspected according to a predetermined pattern to identify potentially safe traffic. Packet inspection and specially-configured network devices may create bottlenecks or other impedances in network traffic. Such specially-configured network devices and packet inspection devices may be relatively expensive additions to a network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figures 1-3 are block diagrams depicting example systems for providing a service.

[0003] Figure 4 depicts an example environment in which various examples for providing a service may be implemented.

[0004] Figure 5 depicts example modules used to implement example systems for providing a service.

[0005] Figures 6 and 7 are flow diagram depicting example methods for providing a service.

DETAILED DESCRIPTION

[0006] In the following description and figures, some example implementations of systems and/or methods for providing a service are described. Some examples of the systems and methods are described specifically for use in a software-defined networking ("SDN") environment. However it should be noted that examples of providing a service described herein may be utilized in a variety of appropriate systems and applications. In particular, a system for providing a service may regulate a service that is able to map to a network flow to a network path based on the service. Therefore, SDN environments are a potential use of the network element identification system. Thus any reference to SDN-specific elements and/or methods is included to provide context for specific examples described herein.

[0007] SDN-compatible networks may provide a service or multiple services to devices or other networks. As used herein, a service is any appropriate supplying of communication, transmissions, software, storage, or any other product, resource, or activity that may be capable of executing on a network of electronic devices. SDN-compatible networks may abstract the hardware of the system from the services being provided. For example, an SDN network may decouple the traffic control decisions from the physical systems that forward network traffic. An SDN network may allow the service to be provided without regard to the underlying physical hardware. For example, a first network device may become latent from receiving too many requests and the SDN network may initiate the service to follow a different traffic flow through a second network device. As another example, the network device or a port of the network device may be malfunctioning and traffic may be rerouted. In both examples, the customer may not notice a change in service because the SDN controller may make the network routing decisions autonomously. For security purposes, a network perimeter that depends largely on the deployment of statically configured firewall devices may congest the network or may otherwise be inconsistent with the abstracted control plane objective of an SDN network.

[0008] Various examples described below relate to providing a service based on a requested behavior of the service. By determining a network policy based on a

requested behavior, a network may determine and forward a policy to a set of network devices in a customized way and/or controlled in a manner based on a desired degree of precision. For example, the network perimeter may be secured based on service behavior for which the traffic is being requested. As used herein, the term “based on” 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, as used herein, a policy is any appropriate manner of management, government, or administration of a network, and a behavior is any appropriate function or action that is part of or related to a service. For example, a request for service may include multiple behaviors, commonly expressed via a protocol port. Behavior-level policy decision making may allow for minimizing access to the network perimeter rather than allowing maximum access based on IP address or protocol authentication.

[0009] Figure 1 is a block diagram depicting an example system 100 for providing a service. In general, an example system 100 for providing a service may generally comprise an application extension 184 and a policy engine 102. The application extension is any combination of circuitry and executable instructions configured to perform as an intermediary between the application 180 and the network 190. More particularly, the application extension 184 may extend from an application 180 to with the policy engine 102 of a controller that manages the network 190. The example policy engine 102 may manage a network policy and provide the network policy to the network 190. For example, in the SDN environment depicted in figure 1, the policy engine 102 on the SDN controller 132 may provide the network policy, such as forwarding policy, to an SDN switch 136 connected to the network 190. The policy engine 102 may provide a default policy to the SDN switch 136 and other devices of the network 190. For example, to secure the network 190, the default policy may be to deny traffic. That is, unless authentication has been provided to create an exception to the default policy, network traffic may be dropped, routed to the network perimeter, or otherwise not allowed to pass over the network 190.

[0010] The example system 100 may also include an example application 180 having content 182 and an authentication service 142. For example, access requests

may only be allowed to the authentication service 142 of the application 180 until a user is authenticated. Once authenticated, the application extension 184 may identify the party requesting authentication based on supplied credentials. The application extension 184 may also identify the service and the behaviors of the service that are being requested by the identified party. The application extension 184 may provide the party's identity and the requested behavior may be provided to the SDN controller 132 having, or otherwise in communication with the policy engine 102. The application extension 184 may also pass on other information, such as metadata regarding the request, to the policy engine 102.

[0011] The information may be used to determine a network policy. For example, the application extension 184 may provide a context of the service request, which may include the identity of the party, the service and behavior(s) being requested, and the protocol that will be used to perform the requested behavior. In another example, the application 180 may be exposed by an application programming interface ("API") and the context may be retrieved by the SDN controller 132 via the API. As used herein, the term "context" is used to describe the information used to determine the network policy. The policy engine 102 may determine the minimal amount of authorization required to perform the requested behavior and provide that policy to the SDN network, and the rest of the device on the network 190.

[0012] The service may be performed and the SDN controller 132 may be aware of when the service has completed or may terminate. For example, the context may include the size or number of packets used to transmit the content 182. The SDN controller 132 may then retrieve the default policy rule and forward that rule to the device 136 of the network 190. For example, the default policy may be to deny all traffic across the network 190 and thereby, secure the network 190 from unauthorized traffic. By maintaining a default denial state and limiting user access to the authentication service 142 of the application, the network 190 may be minimally open to access when the request is known to be authenticated and prevent other traffic over the network 190.

[0013] Figures 2 and 3 are block diagrams depicting example systems for providing a service. Referring to figure 2, an example system 200 for providing a service may generally comprise a policy engine 204. The example system 200 may

also include a context engine 202 and a data store 210. The terms “include,” “have,” and variations thereof, as used herein, have the same meaning as the term “comprise” or appropriate variation thereof. The dotted lines of figures 2 and 3 refer to connections, engines, and/or modules that may be possible variations of the system. For example in figure 2, the system 200 may include a policy engine 204 and/or a context engine 202.

[0014] The context engine 202 represents any combination of circuitry and executable instructions configured to maintain a context of an application. An application, as used hereinafter, is a set of executable instructions to provide a service, such as a computer program. For example, a communications application may include instructions that when executed provide a communications service. A context may be a set of data that represents the state of the application, the state of the request, and/or the state of a system. The context may be maintained by adding, removing, comparing, modifying, retrieving, or other action to prepare and/or make available the set of data associated with the context of the application, the request, and/or the system. For example, the context may be maintained by receiving the context as part of the service request, requesting the context from the system, or monitoring the application, the request, and/or the system to update the context. The context may be related to a service request of an application. For example, the application may request a set of network traffic to perform a service and the context of the request for service may include a requested behavior of the service enabled by the application. The context engine 202 may receive data from an application and identify the context from the data. For example, the context engine 202 may identify a party requesting service and identify a requested behavior of the service. The data from the application may be received directly, by a set of network traffic, or by retrieval from the data store 210. A set of network traffic, as used herein, is any appropriate form of data transmission, such as a number of network packets or a bit stream. A set of network traffic may be the same as a network flow.

[0015] The context is any appropriate set of data containing information related to the service request. As mentioned, that may include a party and a requested behavior. The party may be the source of the request, such as a user. Other data and metadata may be part of the context. For example, the context may include an IP address and

protocol, object data of the application, and/or a system event. The context may be maintained and provided as a set of parameters related to the service request. The application may track or otherwise perform an action to maintain data to determine a context and the context may be provided to the system 200 using an application extension. As discussed further herein, the context may be maintained by the context engine 202 to determine a policy rule to regulate a set of network traffic associated with the service. For example, the context may be maintained and provided to the policy engine to regulate a set of network traffic based on a party and a requested behavior of the context.

[0016] The context engine 202 may be part of an application extension that may integrate with the application's authentication mechanism. For example, the application extension may communicate the context to a policy engine 204. The context may be communicated in a traditional network-centric syntax, including source address, destination address, and protocol. Alternatively, the context may be communicated in a natural language syntax interpretable by a policy engine 204 and/or a controller. The policy, and the policy rule discussed below, may also be communicated in a traditional network-centric syntax or a natural language syntax.

[0017] The policy engine 204 represents any combination of circuitry and executable instructions configured to provide a policy rule to a network device of a network path. The policy rule is any appropriate procedure, control, principle, or other directive expression associated with network traffic and/or communication related to the service to determine conduct based on a condition. For example, the policy rule may be a forwarding rule and the condition is any appropriate conditional expression, such as matching of parameters. The condition parameters may be similar to the parameters of the context. The condition parameters may include a duration adverb or otherwise relate to duration, such as "always forward a set of network traffic matching context A to security device X."

[0018] The policy engine 204 may be located on a device connected to the network, such as a client device, server device, or network device. For example, the policy engine 204 may be located on a controller of a network device to provide a policy rule to forward a set of traffic through a particular route across the network. The

network device is a device of a network of electronic devices for providing an exchange of information, or network flow, among the electronic devices, such as a network router or network switch, and may be located on a network path that may be a set of network devices of the network. A controller may be able to create network flows and communicate with other controllers and/or devices of the network. For example, the controller may be an SDN controller and a SDN-enabled network device may receive the policy rule from the SDN controller to forward the set of network traffic based on the policy rule through a network path that includes a set of network devices authorized to provide the service.

[0019] The policy rule may regulate a set of network traffic based on the context. For example, a party may request a particular behavior that satisfies the condition of a to result in forwarding the set of network traffic, sending a packet, such as an acknowledgement packet ("ACK") or a reset packet ("RST") to the application providing the request, or awaiting a second set of network traffic to make a forwarding determination. A policy rule may be inclusive or exclusive. For example, the policy rule may list a set of impermissible behaviors, and request associated with an impermissible behavior may be denied. The policy engine 204 may receive a context of an application to request a set of network traffic. For example, the policy engine 204 may receive from the context engine 202 an identity of a party and a requested behavior of a service enabled by an application. The policy engine 204 may receive the data directly from the context engine 202, from the set of network traffic, or from the data store 210.

[0020] A policy rule determination based on context may allow for precise control of traffic over the network in comparison to a policy rule based on only IP address or identity of a party. For example, a party may be authorized to communicate over an instant message feature of an application and traffic associated with the video conference feature of the application may be forwarded to the edge of the network because the requested behavior is not authorized, even if requested by an authenticated party.

[0021] The policy engine 204 may maintain a set of permissible behaviors to occur on a network based on the party and the service. For example, the application may allow identity A to communicate via instant messaging, and identity B to

communicate via instant messaging or computer-to-computer video conference. In that example, both identity A and identity B may be authenticated by the application and may communicate over the instant messaging feature, but video conferencing traffic may be allowed only if it is associated with identity B.

[0022] The policy engine 204 may also maintain a set of impermissible behaviors. For example, the network may not allow traffic associated with voice over IP ("VOIP") communication to lines outside the company. More particularly, any packets associated with VOIP to an outside line may be dropped.

[0023] The policy engine 204 may perform at least one of forward the set of network traffic to a destination based on the policy rule, send a packet to an application, and deny the set of network traffic. The policy rule may be a forwarding rule. For example, the forwarding rule may forward a set of network traffic to a destination when the requested behavior is in the set of permissible behaviors. Denial of a set of network traffic may result in forwarding the set of network traffic to an edge of the network, dropping the packets of the set of network traffic, or other action resulting in refusal of the set of network traffic. An example destination may be a client computer, a security engine, or other resource for the service request. The forwarding rule may forward a set of network traffic over a particular route over the network based on the context and the policy rule. For example, the forwarding rule may forward a set of network traffic through a route that includes a set of network devices authorized to provide the service. The security engine and traffic forwarding is discussed in more detail in the description associated with figure 4.

[0024] The policy rule may provide non-forwarding communication or activity. For example, the policy rule may determine to wait for further packets or send a packet to an application when the condition of the policy rule is met. Examples of packets that may be sent are ACKs, RSTs, or a packet understandable by the application, such as a packet to provide a message regarding the requested behavior.

[0025] The policy rule may also deny the set of network traffic based on the context. For example, the set of network traffic may be dropped or forwarded to a network edge (perimeter of the network) if the set of network traffic is associated with an impermissible behavior and identity combination.

[0026] The policy engine 204 may use context data associated with any appropriate layer of the program and/or communication. For example, the context may include the data from the application layer and/or the data from the transport layer. The policy engine 204 may map lower layer data to higher level data. For example, the policy engine 204 may compare a set of transport layer data with the context from the application to determine the policy rule.

[0027] The policy engine 204 may establish a network path. The policy engine 204 may rely on a SDN-enabled device or controller or other module to determine a network path. For example, the policy engine 204 may work in conjunction with a path engine, such as path engine 406 of figure 4, to determine a network path based on the context. The path engine is discussed in more detail in the description associated with figure 4.

[0028] The policy engine 204 may provide a policy rule to secure the network. For example, the network may have a default policy of network routing and routing may be modified based on the context. The network may allow minimal access by maintaining a default rule to deny network traffic and providing a dynamic policy rule to restrict the set of network traffic to the requested behavior when the requested behavior is included in a set of permissible behaviors. For example, the policy engine 204 may maintain a default rule to deny network traffic, maintain a set of permissible behaviors of a service to occur on the network based on the party and the service, provide a policy rule to a controller for a duration of a requested behavior when the party has been authorized to request the requested behavior, and provide the default rule to the controller after the duration, such as completion of the requested behavior.

[0029] The policy engine 204 may be reactive to a change in context. For example, authorization of a party and/or a behavior may change as the application executes and the system operates. The policy engine 204 may maintain a policy rule associated with the party and the requested behavior based on an authentication of the party and update the policy based on a system event. For example, a party may no longer be authorized to perform a behavior after a violation occurs over the network and the policy rule may update to drop all current network traffic related to that party or combination of party and behavior. For another example, the application may be

missing a software update to a fix bug or security threat, and the policy rule may determine to drop packets associated with a behavior affected by the bug or security threat.

[0030] The context engine 202 and the policy engine 204 may be integrated on a single device or distributed across multiple devices. For example, the context engine 202 and the policy engine 204 may be located on a compute device executing the application or an SDN controller distributed across network devices. A distributed context engine 202 and/or policy engine 204 may allow for any appropriate number of devices of the network to provide the policy rule and removes potential bottlenecks, such as always routing traffic through security appliances to inspect traffic before forwarding. For example, in a SDN environment where the SDN controller is distributed across devices, each SDN-compatible device may deploy the policy rule, allowing for the network and the security enforcement to scale at the same rate.

[0031] The data store 210 may store data used by or otherwise associated with the system 200. Specifically, the data store 210 may store data used by the context engine 202 and the policy engine 204. For example, the data store 210 may include data associated with the context of the service request, a policy rule, a list of identities, a set of permissible behaviors, a mapping between the context to a policy rule, etc.

[0032] Figure 3 depicts that the example system 300 for providing a service may be implemented on a memory resource 320 operatively coupled to a processor resource 322. The processor resource 322 may be operatively coupled to a data store 310. The data store 310 may be similar to the data store 210 of figure 2.

[0033] Referring to figure 3, the memory resource 320 may contain a set of instructions that may be executable by the processor resource 322. The set of instructions may implement the system 300 when executed by the processor resource 322. The set of instructions stored on the memory resource 320 may be represented as a context module 302. The set of instructions stored on the memory resource 320 may be represented as multiple modules, such as a policy module 304 or other module discussed herein, such as the path module 406 or analysis module 408 of figure 4. The processor resource 322 may carry out the set of instructions to execute the context module 302, the policy module 304, and/or any appropriate operations between or

associated with the modules of the system 300. For example, the processor resource 322 may carry out a set of instructions to identify a party requesting a service, identify a requested behavior of the service, and maintain a context to determine a policy rule to regulate a set of network traffic associated with the service based on the party and the requested behavior. The context module 302 and the policy module 304 represent program instructions that when executed function as the context engine 202 and the policy engine 204 of figure 2, respectively.

[0034] The processor resource 322 may be one or multiple CPUs capable of retrieving instructions from the memory resource 320 and executing those instructions. The processor resource 322 may process the instructions serially, concurrently, or in partial concurrence, unless described otherwise herein.

[0035] The memory resource 320 and the data store 310 represent a medium to store data utilized by the system 300. The medium is any non-transitory medium or combination of non-transitory mediums able to electronically store data and/or capable of storing the modules of the system 300 and/or data used by the system 300. A storage medium, such as memory resource 320, is distinct from a transmission medium, such as a signal. The medium may be machine-readable, such as computer-readable. The data of the data store 310 may include representations of data and/or information mentioned herein, such as identities, behaviors, policy rules, IP protocol information, and a state of the application or system.

[0036] In the discussion herein, engines 202 and 204 of figure 2 and the modules 302 and 304 of figure 3 have been described as a combination of circuitry and executable instructions. Such components may be implemented in a number of fashions. Looking at figure 3, the executable instructions may be processor executable instructions stored on the memory resource 320, which is a tangible, non-transitory computer readable storage medium, and the circuitry may be electronic circuitry, such as processor resource 322, for executing those instructions. The processor resource 322, for example, may include one or multiple processors. Such multiple processors may be integrated in a single device or distributed across devices. The memory resource 320 may be said to store program instructions that when executed by the processor resource 322 implements the system 300 in figure 3. The memory resource

320 may be integrated in the same device as the processor resource 322 or it may be separate but accessible to that device and the processor resource 322. The memory resource 320 may be distributed across devices, such as network devices 436 of figure 4. The memory resource 320 and the data store 310 may represent the same physical medium unless otherwise described herein.

[0037] In one example, the program instructions may be part of an installation package that when installed may be executed by processor resource 322 to implement the system 300. In that example, the memory resource 320 may be a portable medium such as a CD, a DVD, a flash drive, or memory maintained by a network device, such as network device 436 of figure 4, from which the installation package may be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, the memory resource 320 may include integrated memory such as a hard drive, solid state drive, or the like.

[0038] Figure 4 depicts an example environment in which various examples for providing a service may be implemented. The example environment 400 is shown to include an example system 4 for providing a service. The system 4 (described herein with respect to figures 2 and 3) represents generally any combination of circuitry and executable instructions configured to provide a service. The environment 400 may include a compute device 430, an SDN controller 432, an analytics device 434, a network device 436, a security device 438, and a client device 440. The system may include a context module 402 and a policy module 404 that may be similar to the context module 302 and the policy module 304 of figure 3, respectively, and, for brevity, the associated descriptions are not repeated. The example system of figure 4 depicts the context module 402 and the policy module 404 on separate devices; however, other example systems may integrate the modules 402 and 404 on the same device or distribute the modules 402 and 404 in a different manner. For example, the context module 402 and the policy module 404 may be distributed across the compute device 430 and network devices 436. The system 4 may also include a path module 406, an analysis module 408, an authentication module 442, an event module 444, a stats module 446, a SDN driver 448, a forward module 450, and an inspection module 454. Referring to the solid lines and dotted lines between blocks of figure 4, the blocks may

be connected and/or communicate in the example arrangements depicted. The blocks of figure 4 may be connected and/or communicate in other arrangements.

[0039] The example environments may comprise a network that includes a compute device 430, a set of network devices 436, a security device 438, and a client device 440. The devices shown provide an example number of devices and redundant devices may exist in a network. For example, the example system of figure 4 shows multiple network devices 436; however, an example environment may include any number of network devices 436 and SDN controllers 432.

[0040] The compute device 430 may include a context module 402 and an authentication module 442. The compute device 430 may execute the application extension 452 having a context module 402 to maintain the context of the service request. The application extension 452 executed on the compute device 430 may use the authentication module 442 to authenticate the party identified in the context. Alternatively, the application extension 452 may receive the context and the authentication from an application executing on the compute device 430. The context and/or the authentication may be sent to the SDN controller 432.

[0041] The SDN controller 432 is a controller of the network that is SDN-compatible. The SDN controller 432 may be integrated in a single device or distributed across multiple devices, such as network devices 436. The SDN controller 432 may include a policy module 404, a path module 406, a stats module 446, and a SDN driver 448. The policy module 404 may receive the context and/or the authentication from the compute device 430. The policy module 404 may determine a policy rule based on the context and the authentication. The policy module 404 may forward the context to the event module 444 to determine if an analysis should be made on the set of network traffic.

[0042] The path module 406 represents program instructions that when executed by a processor resource function as a combination of circuitry and executable instructions configured to determine a network path of the network. For example, the path module 406 may determine a network path with a set of network devices that are authorized to provide a service. The policy rule may be used by the path module 406 to determine the network path. The path module 406 may also use information from the

stats module 446 and/or analysis module 408 to determine the network path. The path module 406 may use SDN techniques in determining a network path. For example, if a network device is relatively latent, the path module 406 may determine a path through the network that avoids the latent network device.

[0043] The stats module 446 represents program instructions that when executed by a processor resource function as a combination of circuitry and executable instructions configured to maintain statistics regarding the network. For example, the stats module 446 may monitor a set of network traffic and the network path to determine if an alternative network path should be established. The stats module 446 may be distributed across devices or integrated in multiple devices and provide statistics to an analytics device 434 and/or an analysis module 408.

[0044] The analytics device 434 receives context, statistics, system events and other information associated with the set of network traffic to analyze the set of network traffic. The analytics device 434 may aggregate or other action to calculate an analysis of the set of network traffic and make network path determinations based on the analysis. For example, if the analysis module 408 receives information from the stats modules 446 that indicates the traffic is suspicious, the analytics device 434 may send a system event to update the policy rule and/or change the network path.

[0045] The analysis module 408 represents program instructions that when executed by a processor resource function as a combination of circuitry and executable instructions configured to analyze the set of network traffic. The analysis module 408 may be integrated into the same device as the SDN controller 432, integrated into a separate device, or distributed across devices. For example, the SDN controller may be logically connected to a network device having an analytics engine, such as analytics device 434. The analytics module 408 may analyze a set of network traffic of the network when the set of network traffic is forwarded by the SDN controller. For example, the analytics module 408 may determine a suspicion level of network traffic. The SDN controller 432 may make policy determination based on the analysis of the analysis module 408. For example, the SDN controller 432 may update the policy to route traffic associated with a particular party and behavior combination to a security device 438.

[0046] The security device 438 represents a security engine, which is a combination of circuitry and executable instructions configured to provide a security function to the network. For example, the security device 438 may contain a security module (program instructions that when executed by a processor resource function as a security engine), which may include an inspection module 454. For example, the security device 438 may include a combination of circuitry and executable instructions configured to inspect a set of network traffic. The security device 438 may be configured to handle traffic of a suspicion level, such as a high suspicion level. For example, the security device 438 may include an inspection module 454 to inspect the set of network traffic, such as deep packet inspection, and make a determination of the security threat level of the set of network traffic. The analysis module 408 may determine a suspicion level of traffic and notify the SDN controller of the analysis. The path engine 406 may identify a security device, such as security device 438, having the security engine based on the suspicion level of traffic, maintain the network path to include the security engine based on an analysis of the set of network traffic, and provide the policy rule to the set of network devices of the network path based on a system event, such as a network traffic analysis event. The network may contain multiple security devices 438 and the policy engine 404 may determine a forwarding rule to forward the set of network traffic to a number of the security devices 438. The SDN controller 432 and/or the analytics device 434 may receive a set of data from the security device 438 and the policy rule may be updated based on the set of data received from the security device 438. For example, the policy rule may update to forward traffic associated with a particular party and behavior combination to a particular security device 438 or update the policy rule to forward the network traffic around the security device 438 because the network traffic was determined to be safe.

[0047] The SDN driver 448 may manage the network traffic and pass along the policy rule to the network devices 436. For example, the network path determined by the path engine 406 may be sent to the forward module 450 on each network device 436. Each network device 436 may forward the set of network traffic based on the forwarding rule received from the SDN controller 432. For example, the policy rule may determine the set of network traffic is not a security threat and the set of network traffic

may be routed to the client device 440 to transact the requested behavior. The policy rule may be forwarded to implement security at each of the ports of a set of network devices of the network path, such as the network edge ports. Once the requested behavior is complete, the policy rule and the forward modules 450 may then be updated to a default forwarding rule, such as forwarding the set of network traffic to the edge of the network.

[0048] The modules 402, 404, 406, and 408 of the example system of figure 4 may be integrated into an electronic device, such as compute device 430, SDN-enabled switch 432, analytics device 434, network device 436, security device 438, or client device 440. The example system may be distributed across a combination of devices 430, 432, 434, 436, 438, and 440. The environment may include a traditional client-server environment and/or a cloud computing environment. For example, any appropriate combination of the example system of figure 4 and devices 430, 432, 434, 436, 438, and 440 may be a virtual instance and/or may reside and/or execute on a virtual shared pool of resources described as a "cloud."

[0049] In the example of figure 4, a compute device 430 may request to communicate with a client device 440. The compute device 430 represents generally any computing devices configured to respond to a network request received from another device such as client device 440. For example, a compute device 430 may provide the service and the client device 440 may be a computing device configured to receive and/or communicate with the service. A compute device 430 may include a web server, an application server, or a data server, for example. The client devices 440 represents generally any computing devices configured with a browser or other application to communicate such requests and receive and/or process the corresponding responses.

[0050] Referring to figure 4, the lines between blocks depict example interconnections. For example, the compute device 430 may be in communication with the SDN controller 432. Figure 4 provides example interconnections among devices and example interconnections among modules. The interconnections may be physical and/or logical. A physical interconnect is generally one or any combination of a cable, wireless, fiber optic, or remote connections via a telecommunications link, an infrared

link, a radio frequency link, or any other connectors of systems that provide electronic communication. A physical interconnect may include, at least in part, intranet, the Internet, or a combination of both. A physical interconnect may also include intermediate proxies, routers, switches, load balancers, and the like. A logical interconnect is generally a transmission method that may provide communication among blocks. For example, the SDN driver 448 may be communicate with the forward module 450 of the network device 436 in a SDN environment where the control plane is separated from the transport layer of the network. In that example, the forward modules 450 may not be physically connected, but may be able to communicate with each other.

[0051] Referring to figures 2-4, the engines 202 and 204 and/or the modules 302, 304, 402, 404, 406, 408, 442, 444, 446, 448, 450, and 454 may be distributed across devices 430, 432, 434, 436, 438, and 440, storage mediums, or a combination thereof. The engines and/or modules may complete or assist completion of operations performed in describing another engine and/or module. For example, the policy module 404 may request and/or complete the operations and/or perform the methods of the path module 406. The engines and/or modules may perform the example methods described in connection with figures 5-7.

[0052] Figure 5 depicts example modules used to implement example systems for providing a service. The compute device 530 and the SDN network device 532 may represent the same as the compute device 430 and the SDN controller 432 of figure 4, and, for brevity, the descriptions are not repeated.

[0053] The compute device 530 may execute the application extension which may receive a service request 560. The compute device 530 may determine and/or receive the context 564 of the service request 560. For example, the context module 502 may determine the party making the request and the requested behavior of the service based on the service request 560. The party may be identified and authenticated by the authentication module 542. For example, the application extension may request and/or receive a set of credentials 562 from the party making the request to determine the identity of the party is authentic.

[0054] The SDN network device 532 may receive a context 564 and a set of policy data 566. The policy module 504 may make a policy rule determination based on

the context 564 and the policy data 566. For example, the policy data 566 may be a set of policy rules having a set of conditions and the context 564 may determine which set of conditions are satisfied, and thereby, which policy rule(s) may be implemented and/or provided by the SDN network device 532. The stat module 546 may be used by the SDN network device 532 to update the policy rule based on statistics related to the network. For example, the SDN network device 532 may utilize health statistics of the set of network devices of the network, such as the network devices 436 of figure 4, to determine a policy rule and/or a network path. The path module 506 may receive a set of network devices 568 of the network and determine a network path. For example, the path module 506 may receive a dynamic set of network devices 568 based on health monitoring and authorization of the devices to provide the service. The forward module 550 may translate the selected network path into a forwarding rule 570. The forwarding rule 570 may be forwarded to the set of network devices of the network path. The forwarding rule 570 may include a time duration, such as a maximum time for completion of the requested behavior. For example, the policy rule may update the forwarding rule 570 to a default rule once the time duration is achieved.

[0055] Figures 6 and 7 are flow diagram depicting example methods for providing a service. Referring to figure 6, example methods for providing a service may generally comprise receiving a service request, identifying a party, identifying a behavior requested by the party, and deploying a policy to a network device of a network based on the party and the behavior.

[0056] At block 602, a service request may be received. The service request may be provided from the application executing on a device of the network. The service request may include a set of data related to a context of the service request. For example, the service request may include a destination for the request in the form of an IP address. The set of data may be gathered by the context module 502 of figure 5, for example.

[0057] At block 604, an authentication may be received. The authentication may be performed by an application extension executing on a compute device, such as compute device 530 of figure 5. By including an authentication as part of the context to be received by the policy engine, the SDN controller may route the set of network traffic

knowing the source of the request for network traffic is authentic, rather than guessing the set of network traffic originates from the authentic source, such as by pattern matching or analysis of the network packets.

[0058] At block 606, a party may be identified. The party may be identified as authorized to communicate with a destination associated with the service. For example, the destination may be a client device, such as client device 340 of figure 3, and the party may be authorized if the destination is expecting a communication from the party. The destination may be a client device, a server device, or a network device that may contain a resource to be used by the application or otherwise be desirable to communicate with to perform the behavior of the service.

[0059] At block 608, a behavior requested by the party may be identified. The behavior may be provided as a list of possible behaviors associated with a service. The behavior may be identified based on the context. For example, the behavior may be included in the service request and the context module, such as context module 502 of figure 5, may parse the behavior from the parameters received with the service request.

[0060] At block 610, a policy may be deployed to a network device of a network based on the party and the behavior. For example, the policy engine, such as policy engine 504 of figure 5, may determine a set of network devices, such as network devices 436 of figure 4, and may deploy the policy by sending the policy rule to each of the set of network devices. The policy may be a forwarding policy or other action taken in response to the set of network traffic associated with the combination of party and behavior. The policy may be deployed to all or a subset of the network devices of the network. For example, if the policy is to route the set of network traffic through a particular set of network devices authorized to provide the service, then the policy may be deployed on each of the set of network devices. The set of network devices may receive a forwarding rule or otherwise forward the set of network traffic based on the policy rule. For example, the set of network traffic may be permissible the policy rule may configure the set of network devices to forward the set of network traffic to the destination or the set of network traffic may be impermissible and the policy rule may configure the set of network devices to forward the set of network traffic to an edge of the network.

[0061] Figure 7 includes similar blocks as figure 6 and provides additional blocks and additional details. In particular, figure 7 depicts additional blocks and details generally regarding forwarding a set of network traffic to a security device and reverting a policy state to a denial state. The blocks 702, 704, and 706 are similar to blocks 602, 604, 606, 608, and 610 of figure 6, and for brevity, the associated descriptions are not repeated.

[0062] At block 708, a set of network traffic may be forwarded to a security device for inspection. For example, an analysis module, such as analysis module 308 of figure 3, may analyze a set of network traffic and determine the network traffic achieves a threat threshold and should be further inspected. The threat threshold may be a degree of suspicion that the traffic may offer a threat to the network. The controller, such as SDN controller 432 of figure 4, may deploy a policy rule to route the traffic to a security device, such as security device 438 of figure 4, based on the analysis by the analysis module. Once received by the security device, an inspection module, such as inspection module 454 of figure 4, may perform an inspection of the set of network traffic based on a security protocol. For example, the security protocol may execute deep packet inspection to determine if the set of network traffic contains a security threat. The security protocol may be a procedure based on the level of suspicion.

[0063] At least one of the policy and the network path may be updated based on the inspection of the set of network traffic. For example, the results of the inspection may determine that the policy is to be updated to forward the traffic associated with the behavior to a particular network device, such as a honey-pot, or to the edge of the network.

[0064] At block 710, a current policy state may be reverted to a previous policy state after a communication regarding the behavior terminates. For example, the network's policy state may be a default policy to send network traffic to a security device at the time of initiating the set of network traffic, the policy may be updated to allow network traffic related to the service request during the transmission of the set of network traffic (e.g. once the traffic has been inspected by the security engine, such as security device 438 of figure 4), and may revert to the previous policy once the behavior completes execution (e.g. send traffic to the security device again). The previous policy

state may be a state of denial of network traffic. For example, the default policy may be to restrict a set of network traffic to an edge of the network. Restricting communications of a network until a set of network traffic has been authorized by an application extension based on the context of the service request may allow for the minimal amount of access to the network at any given time. Thereby, the network may maximize security by allowing network traffic based on context, such as a requested behavior, rather than on identity or transport layer data alone.

[0065] Although the flow diagrams of figures 5-7 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 invention.

[0066] 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 invention that is defined in the following claims.

CLAIMS

What is claimed is:

- 1 1. A system for providing a service comprising:
2 a policy engine to:
3 receive a context of an application to request a set of network traffic, the
4 context to include a requested behavior of a service enabled by the application; and
5 provide a policy rule to a network device of a network path, the policy rule to
6 regulate a set of network traffic based on the context.
- 1 2. The system of claim 1, wherein the policy engine is to:
2 maintain a set of permissible behaviors to occur on a network based on the
3 context and the service; and
4 perform at least one of:
5 forward the set of network traffic to a destination based on the policy rule
6 when the requested behavior is in the set of permissible behaviors;
7 send a packet to the application; and
8 deny the set of network traffic.
- 1 3. The system of claim 1, wherein the policy engine is to:
2 compare a set of transport layer data with the context to determine the policy
3 rule; and
4 provide the policy rule to establish the network path based on the context and a
5 system event, the set of network devices authorized to provide the service.
- 1 4. The system of claim 3, comprising an analysis engine to analyze the set of network
2 traffic, a security engine to handle traffic of a suspicion level, and a path engine to
3 determine a network path of the network,
4 the analysis engine to:
5 determine a suspicion level of traffic; and
6 the path engine to:

7 identify a security device having the security engine based on the
8 suspicion level of traffic;
9 maintain the network path to include the security engine; and
10 provide the policy rule to the set of network devices of the network path
11 based on the system event, the policy rule to be a forwarding rule.

1 5. The system of claim 1, comprising:

2 a software-defined networking ("SDN") controller to contain the policy engine;
3 and

4 a SDN-enabled network device to receive the policy rule from the SDN controller
5 to forward the set of network traffic based on the policy rule over the network path,
6 wherein the set of network devices of the network path are authorized to provide the
7 service.

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

3 identify a party requesting a service;

4 identify a requested behavior of the service; and

5 maintain a context to determine a policy rule to regulate a set of network traffic
6 associated with the service based on the party and the requested behavior, the
7 context to include the party and the requested behavior.

1 7. The medium of claim 6, wherein the set of instructions is to:

2 maintain a default rule to deny the set of network traffic;

3 maintain a set of permissible behaviors of a service to occur on the network
4 based on the party and the service; and

5 provide the policy rule to a controller for a duration of the requested behavior
6 when the party has been authorized to request the requested behavior, the policy
7 rule to restrict the set of network traffic to the requested behavior when the
8 requested behavior is included in the set of permissible behaviors; and

9 provide the default rule to the controller after completion of the requested
10 behavior.

1 8. The medium of claim 6, wherein the set of instructions is to:
2 maintain the policy rule associated with the party and requested behavior based
3 on an authentication of the party; and
4 update the policy rule based on a system event.

1 9. The medium of claim 6, wherein the set of instructions is to:
2 maintain the policy rule to establish a network path to include a security module
3 based on an analysis of the set of network traffic.

1 10. The medium of claim 9, wherein the set of instructions is to:
2 receive a set of data from the security module; and
3 update the policy rule based on the set of data.

1 11. A method for providing a service comprising:
2 receiving a service request and an authentication from an application;
3 identifying a party authorized to communicate with a destination associated with
4 the service;
5 identifying a behavior requested by the party; and
6 deploying a policy to a network device of a network based on the party and the
7 behavior to restrict a set of network traffic of the service to the behavior.

1 12. The method of claim 11, comprising:
2 reverting a current policy state to a previous policy state after a communication
3 regarding the behavior terminates.

1 13. The method of claim 12, wherein the previous policy state is a denial state to restrict
2 the set of network traffic to an edge of the network.

1 14. The method of claim 11, comprising:
2 analyzing the set of network traffic; and
3 forwarding the set of network traffic to a security device based on an analysis;
4 and
5 performing an inspection of the set of network traffic based on a security
6 protocol.

1 15. The method of claim 14, comprising:
2 updating at least one of the policy and a network path based on the inspection of
3 the set of network traffic.

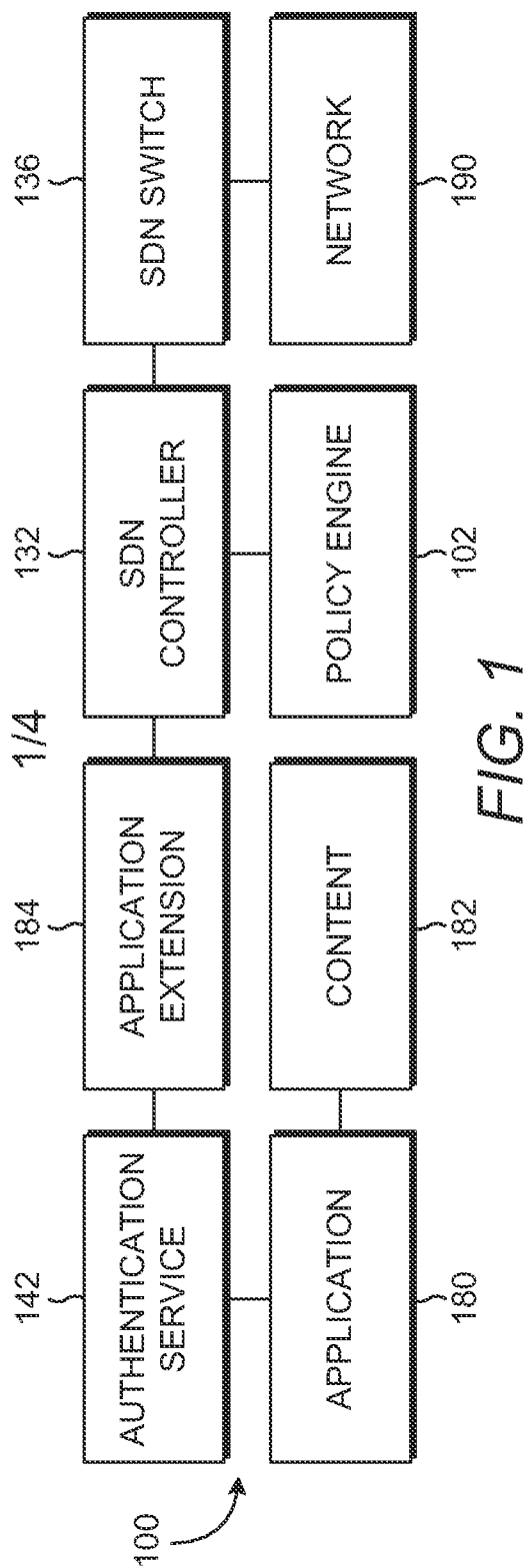


FIG. 1

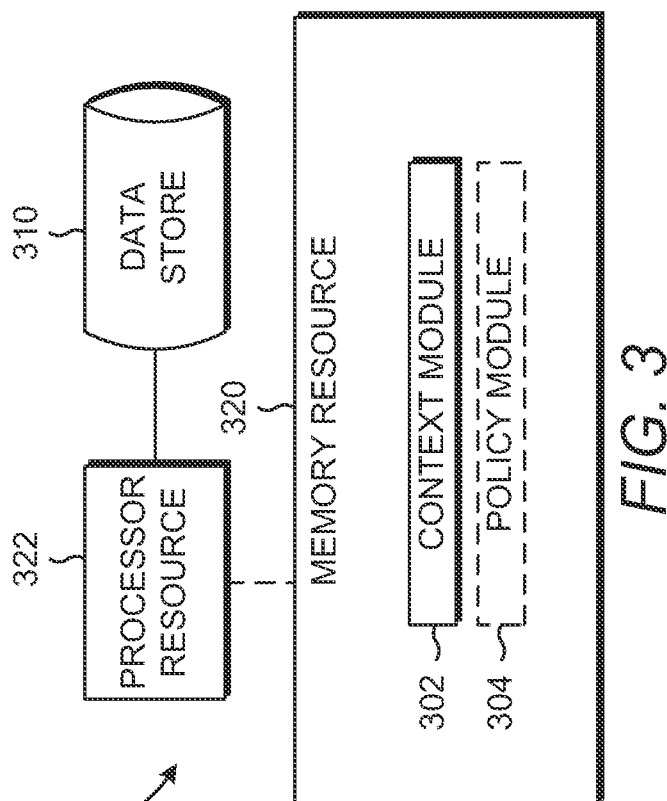


FIG. 3

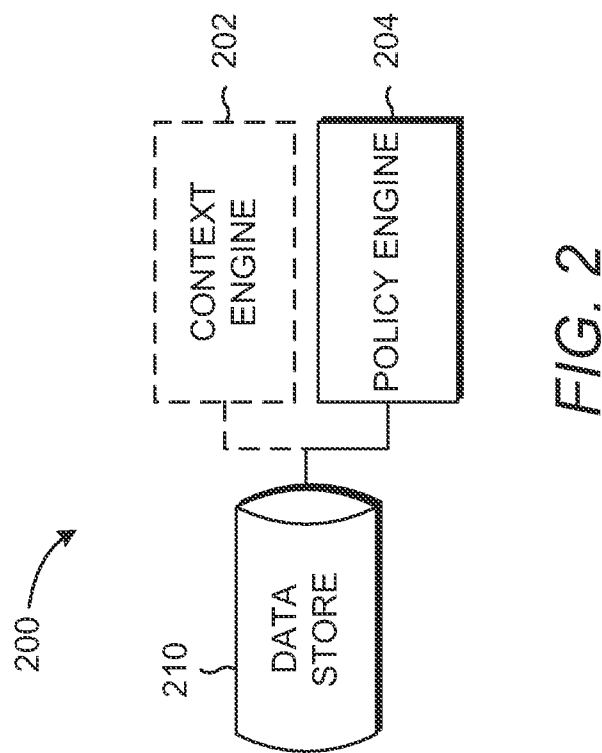
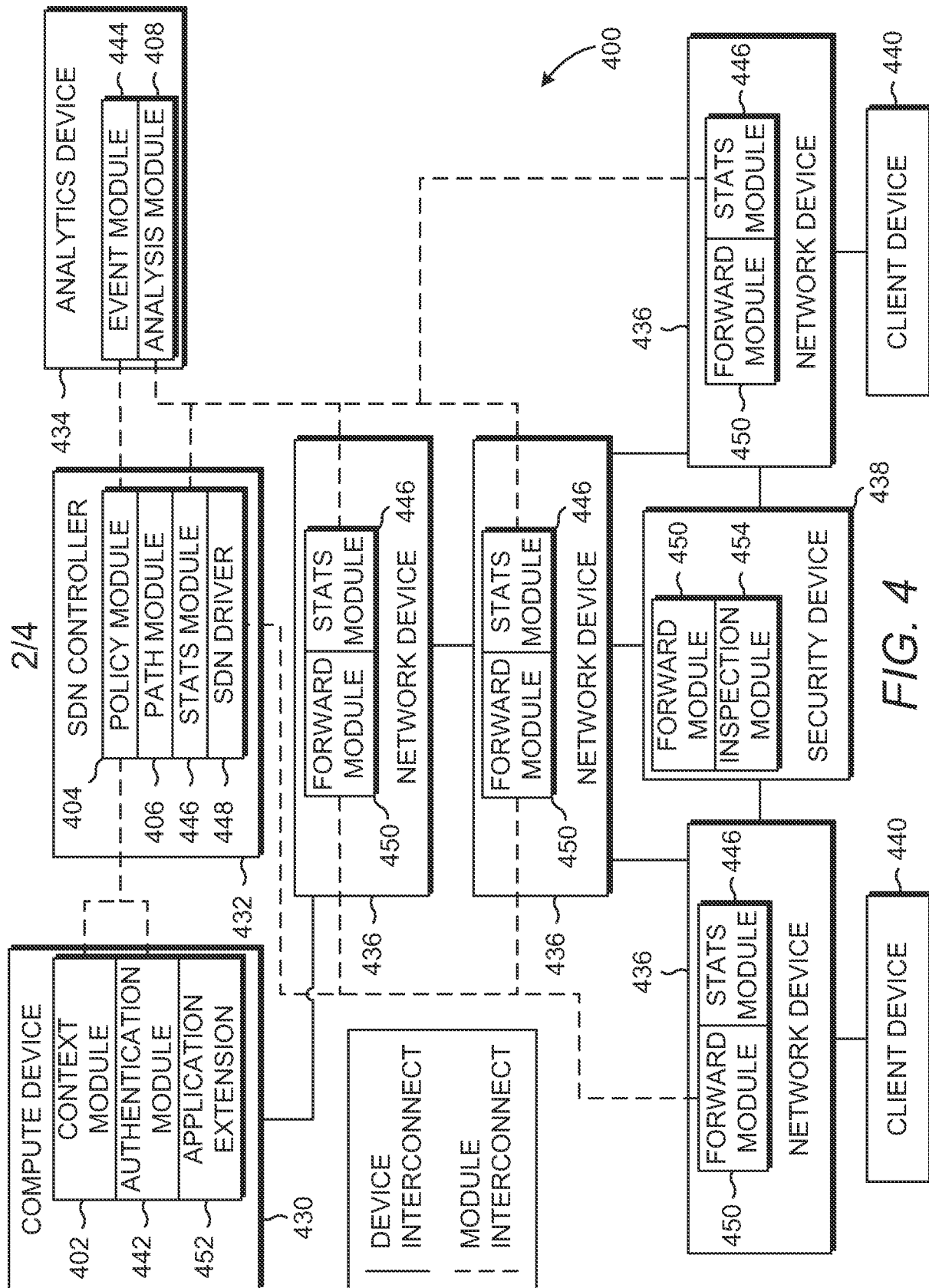


FIG. 2



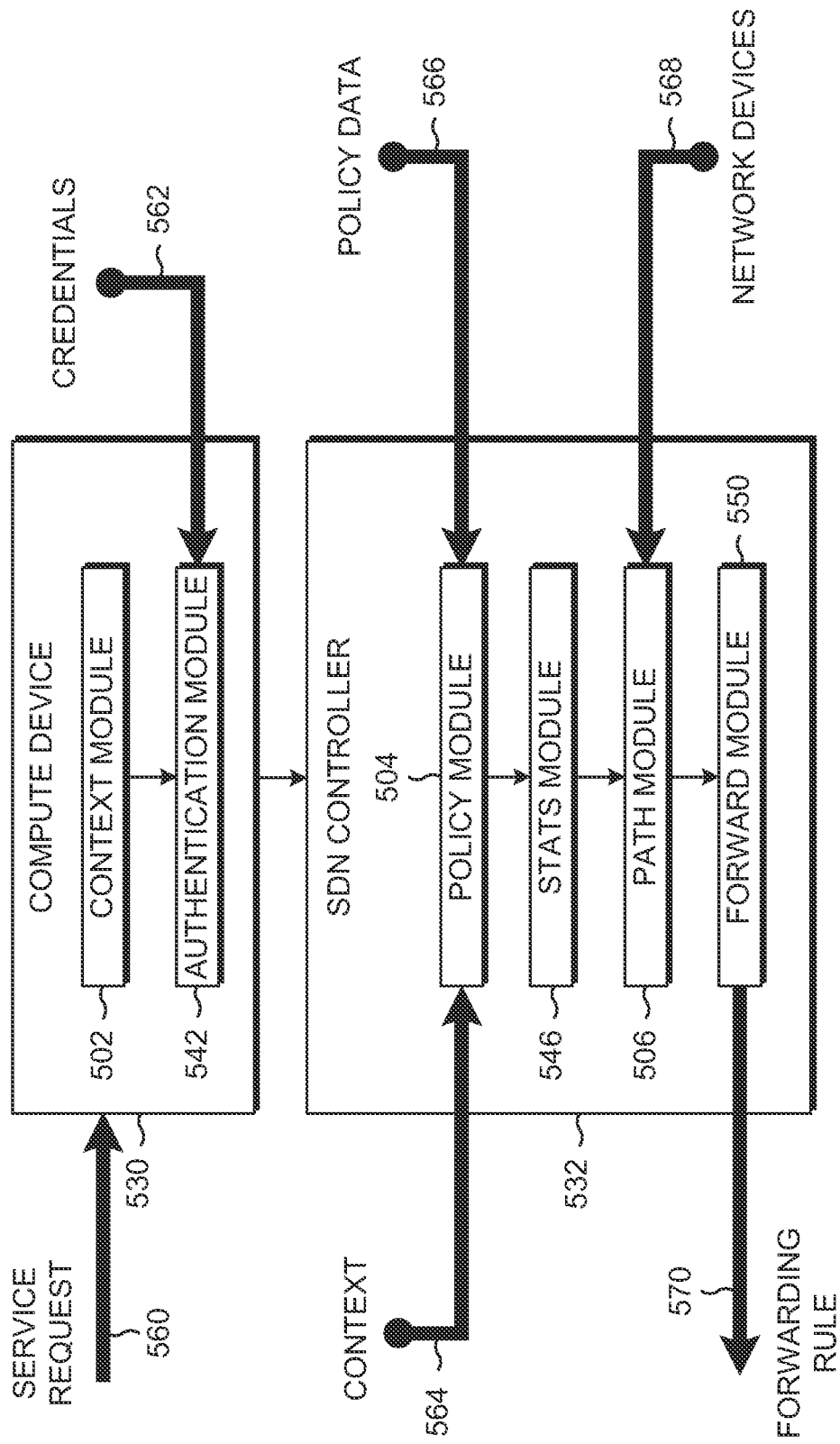
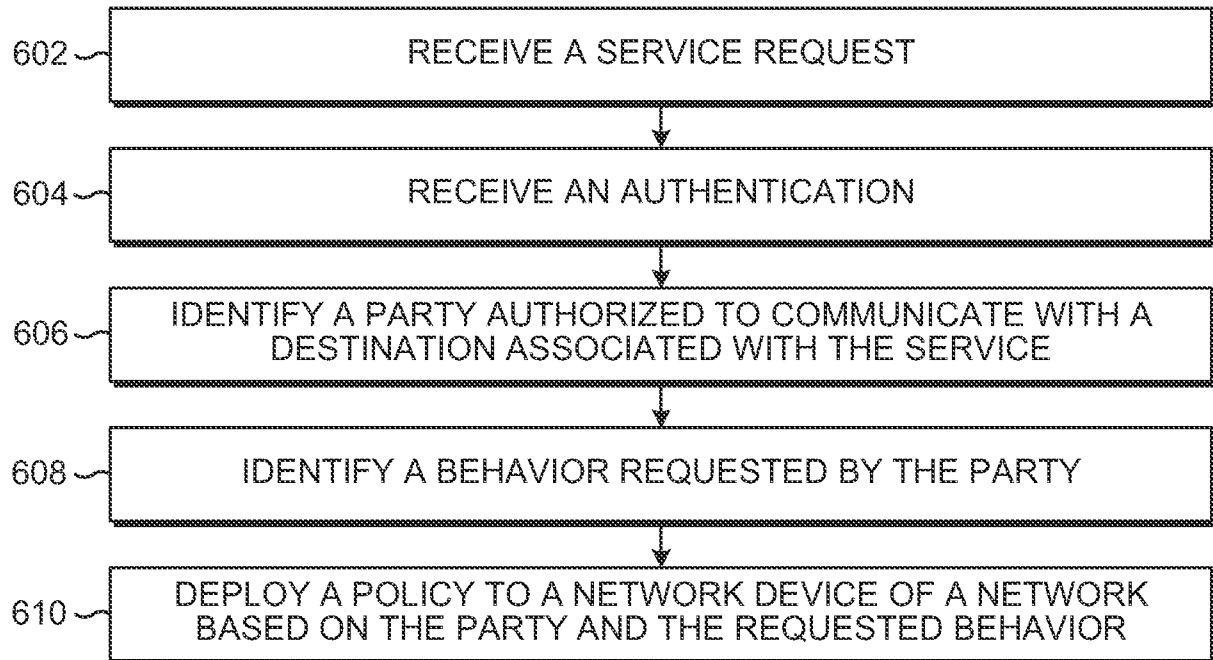
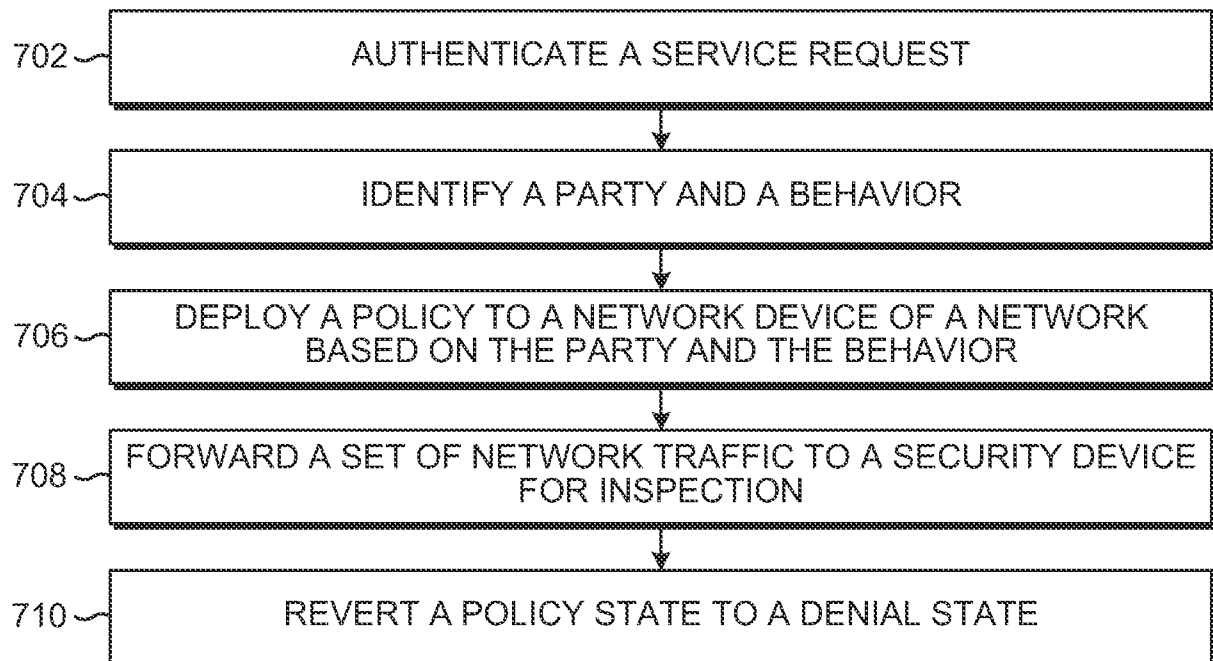


FIG. 5

*FIG. 6**FIG. 7*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/073130**A. CLASSIFICATION OF SUBJECT MATTER****H04L 12/813(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 12/813; H04J 3/18; H04L 12/26; H04L 9/00; G06F 21/00; H04L 12/56; H04L 12/22

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: network, policy, regulate, traffic, application, context, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2011-0219426 A1 (YUSUN KIM et al.) 08 September 2011	1-3, 6-15
Y	See paragraphs 18-31; claims 1, 14; and figures 1-3.	4-5
Y	US 2007-0136783 A1 (ART SHELEST et al.) 14 June 2007	4
	See paragraphs 16, 21; and figures 2-4.	
Y	US 2013-0223442 A1 (RAJESH NARAYANAN et al.) 29 August 2013	5
	See paragraphs 36-37; and figure 6.	
A	US 2010-0290485 A1 (LUCA MARTINI et al.) 18 November 2010	1-15
	See paragraphs 14-43; and figures 1-6.	
A	KR 10-2013-0124692 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE)	1-15
	15 November 2013	
	See paragraphs 46-59; and figures 4-5.	



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

01 September 2014 (01.09.2014)

Date of mailing of the international search report

02 September 2014 (02.09.2014)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
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/US2013/073130

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0219426 A1	08/09/2011	CN 102823197 A EP 2543163 A2 US 8458767 B2 WO 2011-109821 A2 WO 2011-109821 A3	12/12/2012 09/01/2013 04/06/2013 09/09/2011 12/01/2012
US 2007-0136783 A1	14/06/2007	US 7698548 B2	13/04/2010
US 2013-0223442 A1	29/08/2013	None	
US 2010-0290485 A1	18/11/2010	None	
KR 10-2013-0124692 A	15/11/2013	US 2013-0298220 A1	07/11/2013