(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 December 2018 (27.12.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/236554 A1

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US2018/034981

(22) International Filing Date:
30 May 2018 (30.05.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/628,847 21 June 2017 (21.06.2017) US

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **BANSAL, Deepak**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SHARMA, Parag**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **AGGARWAL, Nimish**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **FU, Longzhang**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **CHANDRAPPA, Harish Kumar**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **FIRESTONE, Daniel**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **AGARWAL, Shekhar**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ADUSUMILLI, Anitha**; Microsoft Technology Licens-

(54) **Title:** SYSTEM AND METHOD FOR LIMITING ACCESS TO CLOUD-BASED RESOURCES INCLUDING TRANSMISSION BETWEEN L3 AND L7 LAYERS USING IPV6 PACKET WITH EMBEDDED IPV4 ADDRESSES AND METADATA

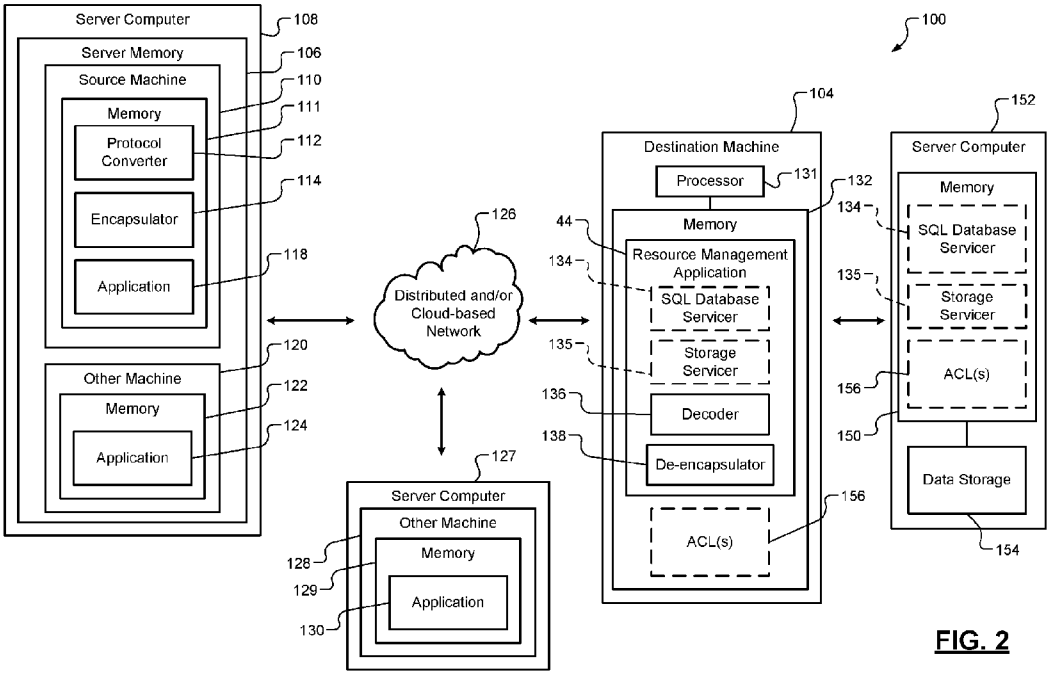

FIG. 2

(57) **Abstract:** A system is provided and includes a processor and a non-transitory computer-readable medium configured to store instructions for execution by the processor. The instructions include: accessing a resource via a first machine in a cloud-based network, where the first machine is a virtual machine; converting at the first machine an IPv4 packet to a IPv6 packet; while converting the IPv4 packet, embedding metadata in the IPv6 packet, where the metadata includes information identifying the first machine or a virtual network of the first machine; and transmitting the IPv6 packet to a second machine to limit access to the resource based on the information identifying the first machine or the virtual network of the first machine. The second machine limits access to the resource based on the information identifying the at least one of the first machine or the virtual network of the first machine.

WO 2018/236554 A1

[Continued on next page]

ing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) **Agent: MINHAS, Sandip S.** et al.; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

SYSTEM AND METHOD FOR LIMITING ACCESS TO CLOUD-BASED RESOURCES INCLUDING TRANSMISSION BETWEEN L3 AND L7 LAYERS USING IPV6 PACKET WITH EMBEDDED IPV4 ADDRESSES AND METADATA

FIELD

[0001] The present disclosure relates to controlling access to cloud-based resources.

BACKGROUND

[0002] The background description provided here is for the purpose of generally presenting the context of the disclosure. Work of the presently named inventors, to the extent it is described in this background section, as well as aspects of the description that may not otherwise qualify as prior art at the time of filing, are neither expressly nor impliedly admitted as prior art against the present disclosure.

[0003] Cloud-based networks allow computer processing and storing needs to be moved from on premises networks to hybrid cloud or fully cloud-based networks while satisfying data security access requirements. A cloud-based network may include physical machines (PMs) and virtual machines (VMs). Cloud applications may be implemented via the PMs and/or the VMs. The cloud applications can be accessed from client stations of organizations at remote locations.

SUMMARY

[0004] A system is provided and includes a processor and a non-transitory computer-readable medium configured to store instructions for execution by the processor. The instructions include: accessing a resource via a first machine in a cloud-based network, where the first machine is a virtual machine; converting at the first machine an IPv4 packet to a IPv6 packet; while converting the IPv4 packet, embedding metadata in the IPv6 packet, where the metadata includes information identifying at least one of the first machine or a virtual network of the first machine; and transmitting the IPv6 packet to a second machine to limit access to the resource based on the information identifying the at least one of the first machine or the virtual network of the first machine. The second machine limits access to the resource based on the information identifying the at least one of the first machine or the virtual network of the first machine.

[0005] In other features, a system and includes a processor and a non-transitory computer-readable medium configured to store instructions for execution by the processor. The instructions include: receiving at a first machine an IPv6 packet from a second machine, where the IPv6 packet includes an IPv4 source address, an IPv4 destination address and

metadata, where the metadata includes information identifying at least one of the second machine or a virtual network of the second machine, and where the second machine is a virtual machine; removing the metadata from the IPv6 packet; and applying an access control list at a traffic controller providing access to a shared resource of a cloud-based network. The access control list includes one or more rules limiting access to a resource based on the information identifying at least one of the second machine or the virtual network of the second machine.

[0006] In other features, a non-transitory computer-readable medium storing processor-executable instructions. The instructions include: accessing a resource via a first machine in a cloud-based network, where the first machine is a virtual machine; converting at the first machine an IPv4 packet to an IPv6 packet; while converting the IPv4 packet, embedding information in the IPv6 packet, where the information identifies at least one of a virtual network, a subnet or an Internet protocol address of the first machine; and transmitting the IPv6 packet to a second machine to limit access to the resource based on the information identifying the at least one of the virtual network, the subnet or the Internet protocol address of the first machine. The second machine limits access to the resource based on the information identifying the at least one of the virtual network, the subnet or the Internet protocol address of the first machine.

[0007] Further areas of applicability of the present disclosure will become apparent from the detailed description, the claims, and the drawings. The detailed description and specific examples are intended for purposes of illustration only and are not intended to limit the scope of the disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The present disclosure will become more fully understood from the detailed description and the accompanying drawings.

[0009] FIG. 1 is a functional block diagram of an example of a resource management system including a resource management application in accordance with an embodiment of the present disclosure.

[0010] FIG. 2 is a functional block diagram of an example of a portion of the resource management system of FIG. 1 illustrating examples of a source machine and a destination machine operating according to an embodiment of the present disclosure.

[0011] FIG. 3 is a functional block diagram of an example of a cloud-computing system illustrating a resource manager, one or more fabric controllers, and one or more load balancers operating according to an embodiment of the present disclosure.

[0012] FIG. 4 is a simplified example of a cluster shown in FIG. 3.

[0013] FIG. 5 is a functional block diagram of an example of a client computer in accordance with an embodiment of the present disclosure.

[0014] FIG. 6 is a functional block diagram of an example of a server computer incorporating applications in accordance with an embodiment of the present disclosure.

[0015] FIG. 7 illustrates an example of a method of operating a source machine in accordance with an embodiment of the present disclosure.

[0016] FIG. 8 is a packet conversion and metadata extraction diagram in accordance with an embodiment of the present disclosure.

[0017] FIG. 9 illustrates an example of a method of operating a destination machine in accordance with an embodiment of the present disclosure.

[0018] In the drawings, reference numbers may be reused to identify similar and/or identical elements.

DETAILED DESCRIPTION

[0019] Below are simplistic examples of a distributed computing environment in which the systems and methods of the present disclosure can be implemented. Throughout the description, references to terms such as servers, client devices, applications and so on are for illustrative purposes only. The terms server and client device are to be understood broadly as representing computing devices with one or more processors and memory configured to execute machine readable instructions. The terms application and computer program are to be understood broadly as representing machine readable instructions executable by the computing devices.

[0020] A cloud-based network includes resources. A portion of the resources may be allocated to a customer and accessed by machines of the customer using a public Internet protocol (IP) address and a corresponding one or more keys. Any machine that has the one or more keys to access the public IP address can gain access to the allocated resources.

[0021] Examples set forth herein provide systems and methods of limiting access of customer allocated resources to one or more applications executed on one or more machines of the customer. The applications may be implemented in a cloud-based network. Access is prevented to other applications and machines. This limited access is provided for resources, such as databases, data storage, files, virtual machines, applications, etc. This limited access provides a layer of protection for customer data stored in publicly accessible databases and data storage.

[0022] The examples include converting Internet protocol version 4 (IPv4) packets to

Internet protocol version 6 (IPv6) packets and embedding metadata in headers of the IPv6 packets. The metadata includes information indicating where the IPv4 packets originated, which is then used to apply a policy of an access control list (ACL) on, for example, a host node of a service provider providing the services. The IPv6 packets are transmitted between a L3 layer of a source machine and a L7 layer of a destination machine. The L3 and L7 layers refer to network and application layers of the Open Systems Interconnection (OSI) model. The metadata provides additional information, not included in a traditional IPv4 packet. After the IPv6 packets are transmitted to the destination machine, the additional information allows the IPv6 packets to be forwarded between applications and/or used to limit access to resource based on ACL(s). Metadata transmission examples disclosed herein are applicable to implementations other than the implementations of limiting access to resources based on ACL(s). The other implementations are described below.

[0023] FIG. 1 shows a resource management system 10 that includes client computers 12, a distributed communication system 14, proxies 16, gateways 18, one or more access server computers 20, and other server computers 22, 24. In some examples, the gateways 18, one or more access server computers 20, and one or more server computers 22 are implemented in a cloud-based network 25. The server computers 22, 24 may be implemented in data centers; example data centers 26 are shown. Each of the data centers may include data storage and any number of server computers. Example data storage is shown in FIGs. 2 and 6. The data centers may be located in corresponding geographical regions, where each geographical region refers to: geographical areas of one or more businesses; one or more towns; one or more cities; one or more states; one or more countries; a continent; and/or other geographical area. In some examples, the server computer 24 is implemented in the cloud-based network 25 or in another service provider network. In another example, the access server computers 20 are not included in the resource management system 10 and the gateways 18 communicate directly with the server computers 22, 24. In the example shown, the client computers are owned by corresponding organizations 27 (also referred to as customers). The organizations 27 may include local area networks (LANs) 28 and firewalls 30. The client computers 12 may access the distributed communication system 14 directly and/or via the LANs 28 and firewalls 30. The distributed communication system 14 may include routers 29. One or more of the organizations 27 may include multiple LANs and corresponding client computers, where each LAN and corresponding client computers are located in a corresponding one of the geographical regions of the data centers 26.

[0024] The cloud-based network 25 may be implemented by a cloud service provider and,

in an embodiment, includes client virtual machines, network appliances and application server computers. Examples of network appliances include routers, switches, firewalls, proxy server computers, World Wide Web (or Web) server computers, wide area network (WAN) accelerators, intrusion detection system (IDS) devices, and intrusion prevention system (IPS) devices. The network appliances provide intermediary services between the application server computers and client computers. The client computers 12 can be implemented in the cloud-based network 25 as VMs and/or PMs 32 or can be located on premises. The network appliances and application server computers may be implemented as one or more VMs of the cloud-based network 25.

[0025] The cloud-based network 25 may include one or more of the server computers 22, 24. The cloud-based network 25 further includes resources that may be shared by the client computers 12. The cloud service provider provisions the resources, such as software applications having corresponding executable code, server computer processing time, server computer processor speed, data storage, VMs, PMs, and/or other resources to tenants (e.g., customers and/or businesses) via the cloud-based network 25. The resources may include user management tools, security application resources, or other resources. A tenant may have one or more subscriptions. A subscription may refer to, for example, rental of one or more resources, a container, a set of machines, a logic subset of machines of a business, and/or a business unit. A business unit includes a set of PMs and/or VMs of a tenant. Cloud service providers implement infrastructure as a service (IaaS) and platform as a service (PaaS) using VMs or containers. A container includes processing, storage and application resources. Data centers may include server computers that host the VMs or containers. Each server can host many VMs and/or containers. The VMs run on a guest operating system and interface with a hypervisor, which shares and manages server hardware and isolates the VMs. Unlike VMs, containers do not need a full OS to be installed or a virtual copy of the host server's hardware. Containers may include a few software modules and libraries and require the use of some portions of an operating system. As a result of the reduced footprint, many more containers can be deployed on a server as compared to virtual machines. The server computers 22, 24 may include VMs (e.g., VMs_{1-Y} are shown), which may be implemented as PMs (e.g., PMs_{1-Y}). The software applications may be implemented on the server computers 22, 24, which may be referred to as physical machines.

[0026] The client computers 12 may be privately owned by different individuals and/or entities. In the example shown, the client computers 12 are owned by organizations 27. Each of the client computers 12 may access one or more cloud applications 34 stored in the server

computers 22 and/or VMs 32. The organizations 27 may own and/or have paid access to corresponding ones of the VMs 32 and/or cloud applications 34. The client computers 12 may include desk/laptop computers, tablets, mobile phones, wearable devices, access terminals, and/or other network devices for accessing the cloud applications 34. Accessing of the cloud applications 34 may include: communicating with the cloud applications 34; transferring data, packets, information, etc. between the client computers 12 and the server computers 22 in association with the cloud applications 34; modifying and/or updating the cloud applications; and/or uploading and/or downloading the cloud applications 34 and/or files associated with the cloud applications 34.

[0027] The distributed communication system 14 may include routers, switches and/or other network devices for directing data and/or packets between (i) the client computers 12 and/or organizations 27 and (ii) the gateways 18. The distributed communication system 14 may include a network, such as a local area network (LAN), a wireless local area network (WLAN), and/or a wide area network (WAN) (e.g., the Internet). The proxies 16 may transfer data, packets, information, etc. between the firewalls 30 and the gateways 18. In an embodiment, the proxies 16 are implemented as part of the distributed communication system 14. The gateways 18 may transfer data, packets, information, etc. between the distributed communication system 14 and the access server computers 20. The access server computers 20 may provide access to direct communication with the server computers 22 and/or the server computer 24. In one embodiment, the access server computers 20 are implemented as client access server computers and facilitate providing services, such as services associated with software as a service (SaaS) applications, from the server computer 24 to the client computers 12. The gateways 18, the access server computers 20, the server computers 22 and/or the server computer 24 may be implemented in an internal network of a service provider that provides software applications.

[0028] The server computer 24 includes a processor 40 that executes software applications, such as a security application and/or a resource management application 44, as shown. The resource management application 44 is stored in memory 42. One or more of the VMs 32 and/or other resources to be allocated to customers may be implemented in the server computer 24. The resource management application 44 may be provided as a service application by the service provider and used to monitor and control traffic flow to cloud applications implemented in the cloud-based network 25. The resource management application 44 monitors behavior of the client computers 12, users of the client computers 12 and/or VMs and PMs of customers directed to access of the cloud applications 34 and

VMs/PMs 32. The VMs and PMs of the customers may include the VMs and PMs 34 and/or on premises VMs and PMs. The client computers 12 may access the resource management application 44 via a web browser and corresponding website. The resource management application 44 may monitor, for example, traffic between client computers of organizations, accessed cloud applications, VMs and PMs.

[0029] The resource management application 44 may receive IPv6 packets having embedded metadata, which the resource management application 44 may then use when applying ACLs at host nodes, server computers, and/or traffic controllers to limit access to resources. Examples of traffic controllers are fabric controllers, allocators, load balancers, storage servers, SQL database servicers, etc. This is further described below. Examples of host nodes, server computers, fabric controllers, allocators, load balancers, a storage and SQL database servicer are shown in FIGs. 1-4. The resource management application 44 may be implemented in the cloud-based network 25 and/or a network of a service provider and is used to monitor activity between (i) client computers and (ii) VMs and/or PMs of the cloud applications. The resource management application 44 may also monitor traffic between VMs and/or PMs in the cloud-based network 25 and/or located on premises. Operations of the resource management application 44 are further described below with respect to at least FIGs. 2 and 6-9.

[0030] FIG. 2 shows a portion 100 of the resource management system 10 of FIG. 1 and includes a source machine 102 and a destination machine 104. The machines 102, 104 may each be a VM or a PM. As an example, the source machine may be a VM implemented in a server computer 108 of the cloud-based network 25 and the destination machine 104 is implemented as a server computer having the resource management application 44. In one embodiment, the destination machine 104 may create and provide ACLs to traffic controllers. As another example, the destination machine 104 may be a traffic controller, such as that in a host node or a server computer and enforce the created ACL(s). In yet another embodiment, a server computer of a service provider other than the server computer of the destination machine 104 may create and provide ACL(s) to traffic controllers. As a few examples, the traffic controllers may be a fabric controller, an allocator, a load balancer, or a storage and/or SQL database servicer.

[0031] As shown, the source machine 102 may be implemented in a server memory 106 of the server computer 108. The server computer 108 may be one of the server computers 22 of FIG. 1. The source machine 102 may include memory 111. In one embodiment, the source machine 102 is implemented as a VM and the memory 111 is VM memory. The

memory 111 includes a protocol converter 112, an encapsulator 114, and may include one or more applications (one application 118 is shown). The protocol converter 112 and the encapsulator 114 may be implemented in a L3 layer of the server computer 108. Operations of the protocol converter 112 and encapsulator 114 are described below with respect to the method of FIG. 7.

[0032] The server memory 106 may include one or more other machines (one other machine 120 is shown). The other machine 120 may include a memory 122 and one or more applications (e.g., application 124). The source machine 110 may be in communication with the destination machine 104 via a distributed network and/or a cloud-based network (shown as "distributed and/or cloud-based network 126). The other machine 120 may be located separate from the server memory 106 and/or server computer 108. As an example, a server computer 127 including machine 128, which has memory 129 is shown. The memory 129 may store one or more applications (e.g., application 130).

[0033] As shown, the destination machine 104 may be a physical machine and replace the server computer 24 of FIG. 1. The destination machine 104 includes a processor 131 and a memory 132. The memory 132 may include the resource management application 44. In one embodiment, the resource management application 44 includes a structured query language (SQL) database servicer 134, a storage servicer 135, a decoder 136, and a de-encapsulator 138. Operations of the SQL database servicer 134, the storage servicer 135, the decoder 136, and the de-encapsulator 138 are described below with respect to the method of FIG. 9.

[0034] Although the SQL database servicer 134 and the storage servicer 135 are shown as being part of the resource management application 44, stored in the memory 132 of the destination machine 104, the SQL database servicer 134 and the storage servicer 135 may be located in a different memory, a different machine and/or a different server computer than the resource management application 44. As an example, the SQL database servicer 134 and the storage servicer 135 may be stored in a memory 150 of a server computer 152. The server computer 152 may include a data storage 154 and/or have access to a data storage. The server computer 152 may be one of the service computers 22 of FIG. 1.

[0035] The resource management application 44, the SQL database servicer 134, and/or the storage servicer 135 may limit access to the data storage 154 and/or other resources of the cloud-based network 25 based on one or more ACL(s) 156 placed in one or more of the memory 132, the memory 150 and/or other memory. The terms "placed", "place", "placing", "put" and "putting" may refer to: creating and storing an ACL; modifying a

previously created ACL; and/or updating of one or more rules of a previously created ACL. The ACL(s) may be initially created and placed by a server computer of a service provider based on a policy generated at and/or a policy request received from a customer machine via a web browser. The web browser may be implemented at a client computer, a customer machine, and/or source machine (e.g., one of the client computers 12, the machines 32 and/or the source machine 110 of FIGs. 1-2).

[0036] After the ACL(s) are created, the ACL(s) may be applied based on information embedded in IPv6 packets. The source machine 110 may generate the IPv6 packets and transmit the IPv6 packets to the destination machine 104. The information may include header information in the IPv6 packets including IPv4 destination addresses and metadata. The metadata may include: a geographical location of the source machine 110; network capabilities (e.g., transmission speed, communication protocols, packet types, types of encoding, etc.) of the source machine 110; a virtual network (VNET) identifier (ID) of a VNET of a customer, a subnetwork (subnet) ID of a portion of the VNET, an IP address of a customer VM, etc. The IP address (or medium access control (MAC) address) of the customer VM is specific to the customer VM and is different than an IPv4 destination address (or MAC address) of a physical machine. A VNET may include any number of VMs, containers, ports, etc. allocated to a customer. A subnet may include any number of the VMs, containers, ports, etc. of a VNET. A VNET may include any number of subnets. The geographical location of the source machine 110 may be used for service provider billing purposes. The network capabilities may be used to maximize connectivity performance with destination machine 104.

[0037] In limiting access, the resource management application 44, the SQL database servicer 134, and/or the storage servicer 135 may permit, for example, the application 118 to access the data storage 154 and/or other resources of the cloud-based network 25 associated with a customer based on the placed ACL. The resource management application 44, the SQL database servicer 134, and/or the storage servicer 135 may permit one or more other applications associated with the same customer to access the data storage 154 and/or other resources of the cloud-based network 25. The resource management application 44, the SQL database servicer 134, and/or the storage servicer 135 may prevent the application 124 and/or the application 130 from accessing the data storage 154 and/or other resources of the cloud-based network 25. The data storage 154 and/or other resources of the cloud-based network 25 may further limit access to a particular VNET, a particular subnet, and/or a particular machine based on the metadata.

[0038] FIG. 3 shows an example of a cloud-computing system (CCS) 200 that includes a cloud controller 202 and at least one data center 204, which may be one of the data centers 26 of FIG. 1. While only one data center is shown for simplicity, the cloud controller 202 can interface with multiple data centers, where each data center may be configured similarly. Further, while the data center 204 is shown as being local to the cloud controller 202, one or more data centers may be geographically remote from the cloud controller 202, may be located in different geographic locations (e.g., in different time zones, different countries or continents, etc.), and may communicate with the cloud controller 202 via various networks. The cloud controller 202 controls the one or more data centers.

[0039] The data center 204 includes fabric controllers 206-1, 206-2, . . . , and 206-n (collectively fabric controllers 206) and corresponding clusters 208-1, 208-2, . . . , and 208-n (collectively clusters 208). Each fabric controller 206 controls a respective cluster 208. Each cluster 208 includes racks (shown in FIG. 4), and each rack includes nodes (shown in FIG. 4). Each node may include one or more server and/or host computers, where each server and/or host computer includes one or more machines. Each fabric controller 206 may include or be associated with an allocator 210 that allocates resources within the cluster 208 for instances of customer services hosted on the cluster 208.

[0040] The cloud controller 202 includes a portal 220 and a software development kit (SDK) 222 that the customers can use to select resources and request service deployment. The cloud controller 202 further includes a resource manager 224, a resource provider 226, and a front-end 228. The resource manager 224 may execute the resource management application 44 of FIGs. 1-2. The front-end 228 interfaces with the fabric controllers 206 of one or more data centers 204. The resource manager 224 receives the customer selections and forwards the customer selections to the resource provider 226. The resource provider 226 generates a tenant model based on the customer selections. The resource provider 226 provisions resources to the customer services according to the tenant model generated based on the customer selections. The resource provider 226 provisions storage, networking, and computing resources by interfacing with a cloud storage (Xstore) 230, a network resource provider 231, and the fabric controllers 206. One or more virtual machines (VMs) may be deployed in one of the clusters 208 based on the tenant model.

[0041] The resource manager 224 and the allocators 210 may be implemented as or include load balancers 240, 242 for controlling flow of traffic to the fabric controllers 206, the clusters 208, the nodes, and/or the server and/or host computers. One or more of the resource manager 224, fabric controllers 206, allocators 210, and load balancers 240, 242

may limit access to resources based on ACLs. The ACLs may include the ACLs described herein as being placed by the resource management application 44.

[0042] FIG. 4 shows a cluster 250, which may be one of the clusters 208 shown in FIG. 4. The cluster 250 includes racks 252. Each of the racks 252 includes a rack controller 254 and nodes 256.

[0043] FIG. 5 shows a client computer 300. The client computers 12 of FIG. 1 may be implemented as the client computer 300. The client computer 300 includes a central processing unit (CPU) or processor 304 and an input device 308 such as a keypad, touchpad, mouse, etc. The client computer 300 further includes memory 312 such as volatile or nonvolatile memory, cache or other type of memory. The client computer 300 further includes bulk storage device 313 such as flash memory, a hard disk drive (HDD) or other bulk storage device.

[0044] The processor 304 of the client computer 300 executes an operating system 314 and one or more applications 318. For example, the applications 318 may include a web browser as described above. The client computer 300 further includes a wired interface (such as an Ethernet interface) and/or wireless interface (such as a Wi-Fi, Bluetooth, near field communication (NFC) or other wireless interface (collectively identified at 320)) that establishes a communication channel over the distributed communication system 14. The client computer 300 further includes a display subsystem 324 including a display 326.

[0045] FIG. 6 shows a server computer 400. The server computers 20, 22, 24 of FIG. 1 may be implemented as and/or have similar architecture as the server computer 400. The server computer 400 includes one or more processors 402 and an input device 404 such as a keypad, touchpad, mouse, etc. The server computer 400 further includes a server memory 406, such as volatile or nonvolatile memory, cache or other type of memory. The processor 402 executes an operating system (OS) 408 and one or more server applications 410 and/or VM applications. Examples of server applications include the resource management application 44 of FIGs. 1-2 and other applications, such as the applications 124, 130 of FIG. 2 and a virtual server service application 412, which is implemented in a virtualization layer and is executed along with the OS 408. The virtual server service application 412 creates a virtual environment in which VM (or guest) OSs (e.g., VM1 OS and VM2 OS) run. Example VM applications App 1, App 2, App 3, and App 4 are shown as being implemented in VM memories 422, 424 of VMs 426, 428, respectively. The VM applications may include instances of auto-generated websites, network appliances, storage applications, productivity applications, and/or other VM applications. VM applications App 1-4 are shown as

examples. Each of the VM memories may include one or more VM applications.

[0046] The server computer 400 further includes a wired or wireless interface 430 that establishes a communication channel over the distributed communication system 14. The server computer 400 further includes a display subsystem 432 that includes a display 434. The server computer 400 may further include a bulk storage device 436 such as flash memory, a hard disk drive (HDD) or other local or remote storage device. The processor 402 may also access a data storage 438 of a remote data store 440 via the interface 430 and the distributed communication system 14, or a data storage 442, such as a database, via the interface 430.

[0047] Operations of the source machine 110, the destination machine 104 and the server computer 152 of FIG. 2 are further described below with respect to the methods of FIGs. 7 and 9 and the packet conversion and metadata extraction diagram of FIG. 8. Although the following operations of FIGs. 7 and 9 are primarily described with respect to the implementations of FIGs. 1-6, the operations may be modified to apply to other implementations of the present disclosure. The operations may be iteratively performed. FIG. 7 shows a method of operating the source machine 110. Although the method of FIG. 7 is described with respect to the source machine 110, the method may be implemented by other machines of FIGs. 1-6.

[0048] The method begins at 500. At 501, a policy and/or policy request may be generated at a customer machine as described above. The customer machine may receive inputs from a user indicating parameters of the policy. The policy may indicate one or more of each of a geographical region, an IP address of a physical machine, a VNET, a subnet, an IP address of a virtual machine, an ID of an application, etc. for which access is permitted to a resource. Access to the resource may be prevented for all other physical machines, virtual machines, applications, etc. At 502, the policy and/or a corresponding policy request may be provided and/or transmitted to a server computer of a service provider. In one embodiment, the policy and/or corresponding policy request is provided to the resource management application 44. The resource management application 44 may create, update and/or modify an ACL. See task 602 of FIG. 9. At 503, the customer machine may receive a confirmation signal and/or indication that the ACL has been created and that the policy is to be enforced.

[0049] At 504, the source machine 110 receives or generates an IPv4 packet, which is to be sent to the destination machine 104. The IPv4 packet may be generated to (i) communicate with an end machine (physical and/or virtual machine) and/or application, (ii) transfer data between the source machine 110 and the end machine (physical and/or virtual

machine) and/or application, and/or (iii) request and gain access to the resource. An example of the IPv4 packet is shown as 550 in FIG. 8 and includes an IPv4 header and payload. The IPv4 header includes prefix fields, an IPv4 source address field and an IPv4 destination address field. The prefix fields may include, for example, version, length, identification, flags, protocol, and checksum fields. The address fields include IPv4 addresses of the source machine 110 and the destination machine 104. Each of the IPv4 address fields may be a public address and includes 32 bits. The IPv4 destination address may identify a node and/or node address. In one example embodiment, the IPv4 packet is associated with and/or generated by the application 118. In another embodiment, the IPv4 packet is associated with and/or requests access to a resource.

[0050] At 505, the protocol converter 112 converts the IPv4 packet to an IPv6 packet. This may occur in the L3 layer of the server computer 108. An example of the IPv6 packet is shown as 552 in FIG. 8 and includes a header and a payload. The header includes prefix fields, an IPv6 source address field and an IPv6 destination address field. The address fields include respectively IPv6 addresses of the source machine 110 and the destination machine 104. Each of the IPv6 addresses may be a private address and includes 128 bits. An example of the IPv6 source address field is shown as 554 and includes an IPv6 source address prefix fields, metadata and the IPv4 source address field. The IPv6 source address prefix fields may include, for example, version, traffic class, flow label, payload length, next header and hop limit fields. The metadata may include: a geographical location of the source machine 110; network capabilities (e.g., transmission speed, communication protocols, packet types, types of encoding, etc.) of the source machine 110; a VNET ID, a subnet ID, an IP address of a customer VM, an ID of an end application, etc. An example of the IPv6 destination address field is shown as 556 and includes an IPv6 destination address prefix field and the IPv4 destination address field with the IPv4 destination address. The IPv6 destination address prefix field may include 96 bits. The payload of the IPv6 packet includes the payload of the IPv4 packet.

[0051] At 505A, the protocol converter 112 may determine parameters of the metadata to be included in the IPv6 source address field 554. This may include determining the geographical location of the source machine 110; the network capabilities of the source machine 110; the VNET ID, the subnet ID, the IP address of a customer VM, the ID of the end application, and/or other parameters. Some or all of these parameters may be stored in memory 111. In one embodiment, 96 bits of the IPv6 source address field 554 are reserved for the metadata. In one embodiment, 16 bits of the 96 bits are reserved for the geographical

location and/or region of the source machine 110. As an example, the 16 bits may indicate whether the source machine 110 is in the United States or is in another country. Of the 96 bits, 32 bits may be reserved for a virtual location identified by the VNET ID. Of the 96 bits, 16 bits may be reserved for the subnet ID. Of the 96 bits, 32 bits may be reserved for the IP address of the VM. The metadata may include geographical location of the source machine 110; the network capabilities of the source machine 110; the VNET ID, the subnet ID, the IP address of a customer VM, and/or the other parameters.

[0052] At 505B, the protocol converter 112 may generate the IPv6 source address field 554 including the determined metadata. At 505C, the protocol converter 112 may generate the IPv6 destination address field 556 to include the IPv4 destination address. The IPv4 destination address may be appended to an end of the IPv6 destination address prefix. Operation 505 may include encoding a portion or the entire IPv6 packet using a predetermined protocol.

[0053] At 506, the encapsulator 114 encapsulates the IPv6 packet. An example of the encapsulated IPv6 packet is shown as 558 in FIG. 8. The encapsulated IPv6 packet includes the IPv4 source address field, the IPv4 destination address field, an encapsulation header appended on the beginning of the IPv6 packet. The encapsulation header may include additional routing information. The IPv6 packet may be encapsulated via network visualization using generic routing encapsulation (NVGRE) or virtual extensible local area network (VXLAN) technologies. This allows the encapsulated packet to be routed and tunneled between network layers.

[0054] At 510, the source machine 110 transmits the encapsulated packet from the source machine 110 to the destination machine 104 based on the IPv4 destination address and the encapsulation header. The method may end at 512. The method of FIG. 9 may be performed subsequent to the method of FIG. 7.

[0055] FIG. 9 shows a method of operating a destination machine 104. Although the method of FIG. 9 is described with respect to the destination machine 104, the method may be implemented by other machines of FIGs. 1-6.

[0056] The method begins at 600 and may include execution of the resource management application 44 by the processor 40. At 601, the resource management application 44 may receive the policy/policy request from the customer (or source) machine 110. At 602, the resource management application 44 may create, update and/or modify the ACL based on the information in the policy/policy request. This may include creating, updating and/or modifying one or more rules of the ACL. The resource management application 44 may

place the ACL at a traffic controller as described above. As an example, the ACL may be placed at the server computer 152 for use by the SQL database servicer 134 and/or the storage servicer 135.

[0057] The ACLs described herein may be endpoint ACLs created for security reasons. The ACLs provide the ability to selectively permit or deny traffic for a resource endpoint. This packet filtering capability provides a layer of security. Network ACLs may be created for endpoints (e.g., ports of resources). The ACLs include lists of rules. When an ACL is created and applied to a resource endpoint, packet filtering takes place on a host node of the resource. This means the traffic from remote IP addresses is filtered by the host node based on ACL rules. When a resource is created, a default ACL may be created to block all incoming traffic to the resource. However, if an endpoint is created for a port of the host node, then the default ACL is modified to allow all inbound traffic for that endpoint. Inbound traffic from any remote subnet is then allowed to that endpoint and no firewall provisioning is required. All other ports are blocked for inbound traffic unless endpoints are created for those ports. Outbound traffic is allowed by default. During operation 602, a pre-existing ACL may be modified and/or a new ACL may be created with one or more rules set up for access to a resource.

[0058] The ACLs allow for: selectively permitting or denying incoming traffic to a resource endpoint; blacklisting IP addresses and/or other forms of identification; creating multiple rules per resource endpoint; and/or assigning a specific ACL and/or set of ACL rules to a particular one of the parameters of the metadata. Each ACL may have a corresponding ACL table relating parameters to permit/deny access values.

[0059] At 604, the resource management application 44 may subsequently receive the encapsulated IPv6 packet from the source machine 110. The encapsulated IPv6 packet may be associated with accessing the resource. In one embodiment, this resource is a same resource as accessed by the application 118 and referred to above for operations 501-503. The encapsulated IPv6 packet may be received from, for example, one of the applications 118, 124, 130 or from another application. At 606, the de-encapsulator 138 de-encapsulates the received packet. This includes removing the IPv4 source address field, the IPv4 destination address field, and the encapsulation field.

[0060] At 608, the resource management application 44 removes the metadata from the IPv6 packet. An example of the metadata is shown as 562 in FIG. 8. The following operations 614-618 may be performed by the resource management application 44 or by one of the above-described traffic controllers to enforce the policy of the ACL.

[0061] At 614, the resource management application 44 and/or the traffic controller may determine whether the machine executing the application that sent the encapsulated IPv6 packet is permitted to access the resource based on the ACL generated, updated, and/or modified during operation 602. This may include determining if parameters of the IPv6 packet satisfy one or more rules of the ACL. This may also include determining whether one or more of the parameters associated with the request and/or the machine executing the application that sent the request match one or more of the parameters in the metadata. The comparison/evaluation of the parameters assures that the application and/or machine executing the application that generated the request are owned and/or subscribed to by a particular customer and/or are part of a particular VNET and/or subnet. This verification may also include determining whether the machine executing the application that generated the request is a particular machine and/or VM. In one embodiment, a particular application (e.g., the application 118) is permitted access and all other applications are prevented from accessing the resource.

[0062] If the requested access is permitted, operation 616 is performed to provide access to the resource. At 617, the de-encapsulated IPv6 packet may be forwarded from the application that initially received the IPv6 packet to another (or end) application, an end resource and/or machine of an end resource. This transfer may be based on the IPv4 source and/or destination addresses and/or the IPv6 source and/or destination addresses in the de-encapsulated IPv6 packet. In one embodiment, this transfer is isolated to the L7 layer. If the requested access is denied, operation 618 is performed and the access to the resource is prevented. The method may end at 620.

[0063] The above-described operations of FIGs. 7 and 9 are meant to be illustrative examples; the operations may be performed sequentially, synchronously, simultaneously, continuously, during overlapping time periods or in a different order depending upon the application. Also, any of the operations may not be performed or skipped depending on the implementation and/or sequence of events.

[0064] Although FIG. 9 is primarily described with respect to the placement of ACLs based on metadata embedded in an IPv6 packet, the metadata may be used for audit logging, domain name server translations, and/or other purposes. For audit logging, the metadata may be used, for example, when during generation of logs of when events occurred and what the events involved. For example, a log of when each VM accessed each resource. The metadata may be used to limit access to a docket key database, a data storage, a SQL database, and/or other resources.

[0065] The foregoing description is merely illustrative in nature and is in no way intended to limit the disclosure, its application, or uses. The broad teachings of the disclosure can be implemented in a variety of forms. Therefore, while this disclosure includes particular examples, the true scope of the disclosure should not be so limited since other modifications will become apparent upon a study of the drawings, the specification, and the following claims. It should be understood that one or more steps within a method may be executed in different order (or concurrently) without altering the principles of the present disclosure. Further, although each of the embodiments is described above as having certain features, any one or more of those features described with respect to any embodiment of the disclosure can be implemented in and/or combined with features of any of the other embodiments, even if that combination is not explicitly described. In other words, the described embodiments are not mutually exclusive, and permutations of one or more embodiments with one another remain within the scope of this disclosure.

[0066] Spatial and functional relationships between elements (for example, between modules) are described using various terms, including "connected," "engaged," "interfaced," and "coupled." Unless explicitly described as being "direct," when a relationship between first and second elements is described in the above disclosure, that relationship encompasses a direct relationship where no other intervening elements are present between the first and second elements, and also an indirect relationship where one or more intervening elements are present (either spatially or functionally) between the first and second elements. As used herein, the phrase at least one of A, B, and C should be construed to mean a logical (A OR B OR C), using a non-exclusive logical OR, and should not be construed to mean "at least one of A, at least one of B, and at least one of C."

[0067] In the figures, the direction of an arrow, as indicated by the arrowhead, generally demonstrates the flow of information (such as data or instructions) that is of interest to the illustration. For example, when element A and element B exchange a variety of information but information transmitted from element A to element B is relevant to the illustration, the arrow may point from element A to element B. This unidirectional arrow does not imply that no other information is transmitted from element B to element A. Further, for information sent from element A to element B, element B may send requests for, or receipt acknowledgements of, the information to element A.

[0068] In this application, including the definitions below, the term "module" or the term "controller" may be replaced with the term "circuit." The term "module" may refer to, be part of, or include processor hardware (shared, dedicated, or group) that executes code and

memory hardware (shared, dedicated, or group) that stores code executed by the processor hardware.

[0069] The module may include one or more interface circuits. In some examples, the interface circuits may include wired or wireless interfaces that are connected to a local area network (LAN), the Internet, a wide area network (WAN), or combinations thereof. The functionality of any given module of the present disclosure may be distributed among multiple modules that are connected via interface circuits. For example, multiple modules may allow load balancing. In a further example, a server (also known as remote, or cloud) module may accomplish some functionality on behalf of a client module.

[0070] The term code, as used above, may include software, firmware, and/or microcode, and may refer to programs, routines, functions, classes, data structures, and/or objects. Shared processor hardware encompasses a single microprocessor that executes some or all code from multiple modules. Group processor hardware encompasses a microprocessor that, in combination with additional microprocessors, executes some or all code from one or more modules. References to multiple microprocessors encompass multiple microprocessors on discrete dies, multiple microprocessors on a single die, multiple cores of a single microprocessor, multiple threads of a single microprocessor, or a combination of the above.

[0071] Shared memory hardware encompasses a single memory device that stores some or all code from multiple modules. Group memory hardware encompasses a memory device that, in combination with other memory devices, stores some or all code from one or more modules.

[0072] The term memory hardware is a subset of the term computer-readable medium. The term computer-readable medium, as used herein, does not encompass transitory electrical or electromagnetic signals propagating through a medium (such as on a carrier wave); the term computer-readable medium is therefore considered tangible and non-transitory. Non-limiting examples of a non-transitory computer-readable medium are nonvolatile memory devices (such as a flash memory device, an erasable programmable read-only memory device, or a mask read-only memory device), volatile memory devices (such as a static random access memory device or a dynamic random access memory device), magnetic storage media (such as an analog or digital magnetic tape or a hard disk drive), and optical storage media (such as a CD, a DVD, or a Blu-ray Disc).

[0073] The apparatuses and methods described in this application may be partially or fully implemented by a special purpose computer created by configuring a general purpose computer to execute one or more particular functions embodied in computer programs. The

functional blocks and flowchart elements described above serve as software specifications, which can be translated into the computer programs by the routine work of a skilled technician or programmer.

[0074] The computer programs include processor-executable instructions that are stored on at least one non-transitory computer-readable medium. The computer programs may also include or rely on stored data. The computer programs may encompass a basic input/output system (BIOS) that interacts with hardware of the special purpose computer, device drivers that interact with particular devices of the special purpose computer, one or more operating systems, user applications, background services, background applications, etc.

[0075] The computer programs may include: (i) descriptive text to be parsed, such as HTML (hypertext markup language), XML (extensible markup language), or JSON (JavaScript Object Notation) (ii) assembly code, (iii) object code generated from source code by a compiler, (iv) source code for execution by an interpreter, (v) source code for compilation and execution by a just-in-time compiler, etc. As examples only, source code may be written using syntax from languages including C, C++, C#, Objective-C, Swift, Haskell, Go, SQL, R, Lisp, Java®, Fortran, Perl, Pascal, Curl, OCaml, Javascript®, HTML5 (Hypertext Markup Language 5th revision), Ada, ASP (Active Server Pages), PHP (PHP: Hypertext Preprocessor), Scala, Eiffel, Smalltalk, Erlang, Ruby, Flash®, Visual Basic®, Lua, MATLAB, SIMULINK, and Python®.

[0076] None of the elements recited in the claims are intended to be a means-plus-function element within the meaning of 35 U.S.C. §112(f) unless an element is expressly recited using the phrase "means for" or, in the case of a method claim, using the phrases "operation for" or "step for."

CLAIMS

1. A system comprising:

a processor; and

a non-transitory computer-readable medium configured to store instructions for execution by the processor, wherein the instructions include

accessing a resource via a first machine in a cloud-based network, wherein the first machine is a virtual machine,

converting at the first machine an IPv4 packet to a IPv6 packet,

while converting the IPv4 packet, embedding metadata in the IPv6 packet, wherein the metadata includes information identifying at least one of the first machine or a virtual network of the first machine, and

transmitting the IPv6 packet to a second machine to limit access to the resource based on the information identifying the at least one of the first machine or the virtual network of the first machine, wherein the second machine limits access to the resource based on the information identifying the at least one of the first machine or the virtual network of the first machine.

2. The system of claim 1, wherein:

the IPv6 packet includes a header and a payload;

the header includes an IPv6 source address and an IPv6 destination address; and

the IPv6 source address includes the metadata.

3. The system of claim 1, wherein the instructions further include:

encapsulating the IPv6 packet to provide an encapsulated IPv6 packet, wherein the encapsulated IPv6 packet includes an IPv4 source address of the IPv4 packet, an IPv4 destination address of the IPv4 packet, and an encapsulation header; and

transmitting the encapsulated IPv6 packet to the second machine.

4. The system of claim 1, wherein the instructions further include:

determining parameters to include in the metadata, wherein the parameters include a virtual network identifier, a subnet identifier and an address of the first machine;

generating an IPv6 source address field to include the parameters; and

generating the IPv6 packet to include the IPv6 source address field,

wherein the IPv6 packet is transmitted to the second machine to limit access to the resource based on the parameters.

5. The system of claim 1, wherein the instructions further include:

determining a parameter to include in the metadata, wherein the parameter indicates

a geographical location of the first machine;

generating an IPv6 source address field to include the parameter; and

generating the IPv6 packet to include the IPv6 source address field,

wherein the IPv6 packet is transmitted to the second machine to limit access to the resource based on the parameter.

6. The system of claim 1, wherein the instructions further include:

determining parameters to include in the metadata, wherein the parameters include network capabilities of the first machine;

generating an IPv6 source address field to include the parameters; and

generating the IPv6 packet to include the IPv6 source address field,

wherein the IPv6 packet is transmitted to the second machine to limit access to the resource based on the parameters.

7. The system of claim 1, further comprising:

a second processor; and

a second non-transitory computer-readable medium configured to store second instructions for execution by the second processor, wherein the instructions include

receiving at the second machine the IPv6 packet from the first machine, wherein the IPv6 packet includes an IPv4 source address, an IPv4 destination address and the metadata,

removing the metadata from the IPv6 packet, and

applying an access control list at a traffic controller providing access to a shared resource of the cloud-based network, wherein the access control list includes one or more rules limiting access to the resource based on the information identifying at least one of the first machine or the virtual network of the first machine.

8. The system of claim 7, wherein the second instructions include:

comparing the metadata to corresponding data in the access control list; and

permitting access to the resource if the metadata matches the data in the access control list.

9. The system of claim 7, wherein:

the second machine includes the traffic controller;

the second instructions further include limiting access to the resource based on the access control list; and

the limiting access to the resource includes permitting an application of the second machine to access the resource and preventing other applications from accessing the

resource.

10. The system of claim 7, wherein:

the metadata includes parameters;

the parameters include a virtual network identifier, a subnet identifier and an address of the first machine; and

the second instructions further include applying the access control list based on the parameters.

11. The system of claim 7, wherein:

the metadata includes a parameter;

the parameter indicates a geographical location of the second machine; and

the second instructions further include applying the access control list based on the parameter.

12. The system of claim 7, wherein:

the metadata includes parameters;

the parameters include network capabilities of the first machine; and

the second instructions further include applying the access control list based on the parameters.

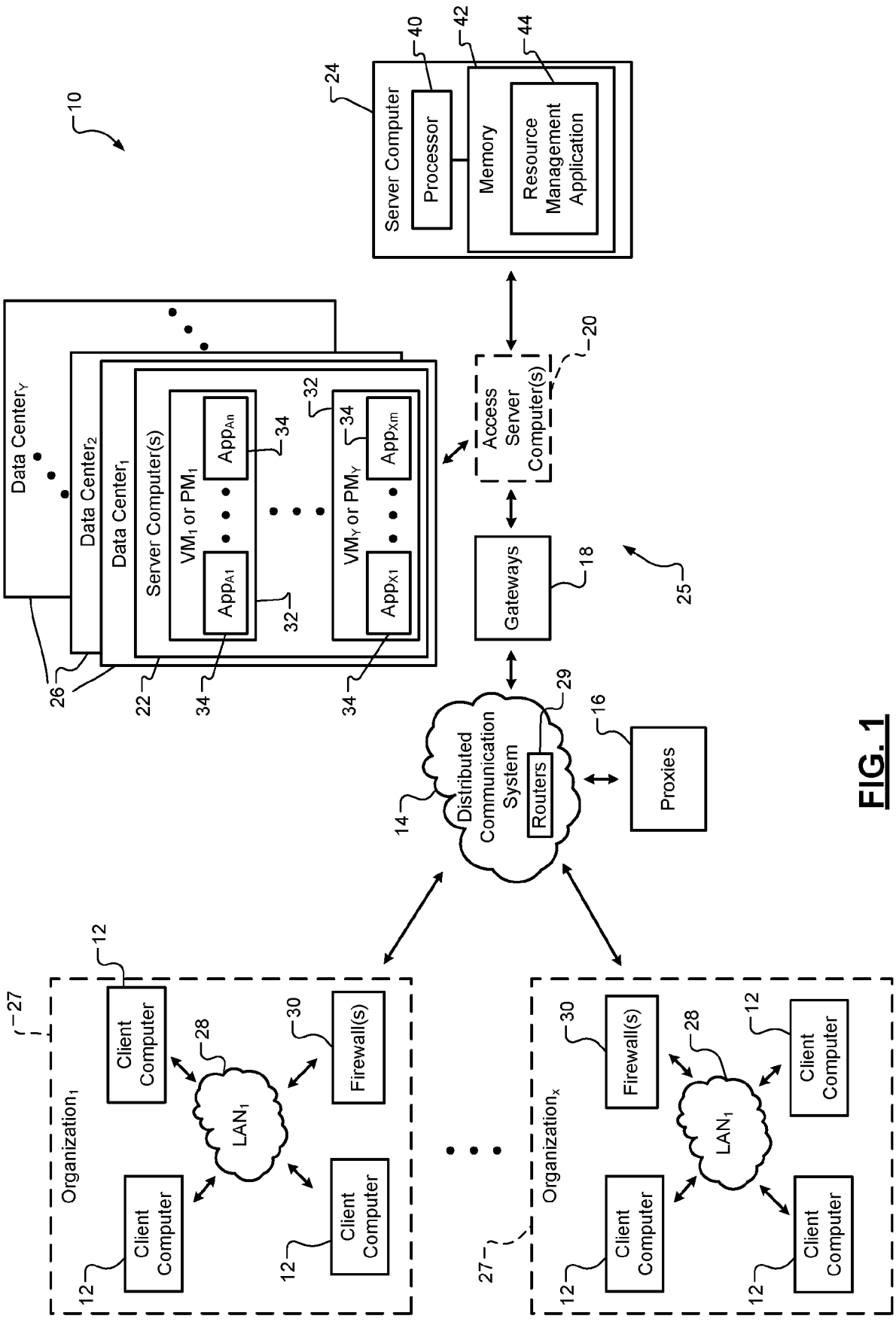
10
24
Server Computer
40
Processor
42
Memory
44
Resource Management Application
Data Center$_Y$
Data Center$_2$
Data Center$_1$
Server Computer(s)
VM$_1$ or PM$_1$
App$_{A1}$
App$_{An}$
VM$_Y$ or PM$_Y$
App$_{X1}$
App$_{Xm}$
32
34
26
22
Access Server Computer(s)
20
Gateways
18
25
Distributed Communication System
Routers
29
14
16
Proxies
27
Organization$_1$
Client Computer
12
28
LAN$_1$
30
Firewall(s)
Organization$_x$

FIG. 1

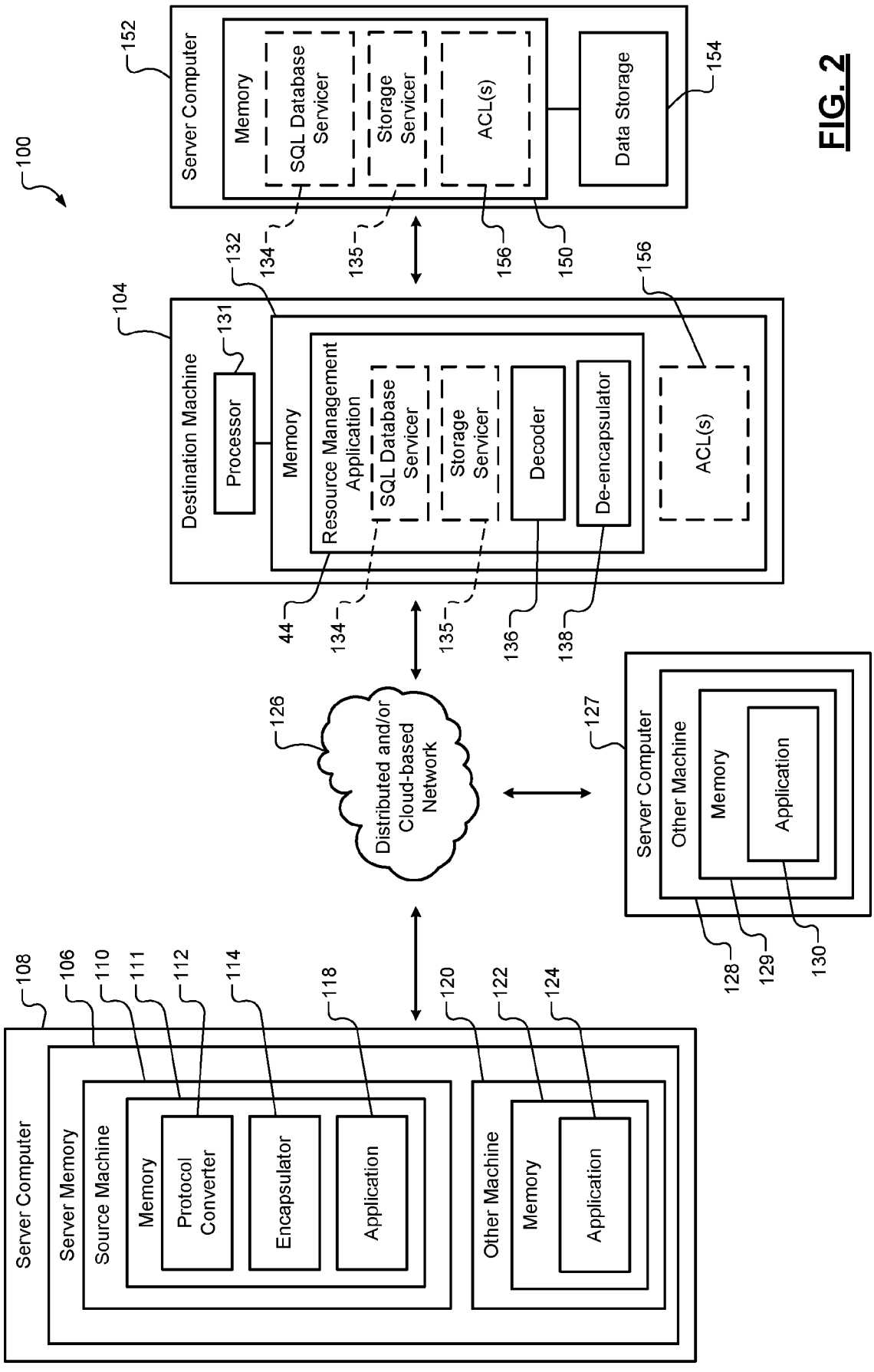
100
Server Computer
152
Memory
SQL Database Servicer
Storage Servicer
ACL(s)
Data Storage
154
132
134
135
156
150
104
Destination Machine
131
Processor
Memory
Resource Management Application
SQL Database Servicer
Storage Servicer
Decoder
De-encapsulator
ACL(s)
156
44
134
135
136
138
126
Distributed and/or Cloud-based Network
127
Server Computer
Other Machine
Memory
Application
128
129
130
108
106
110
111
112
114
118
120
122
124
Server Computer
Server Memory
Source Machine
Memory
Protocol Converter
Encapsulator
Application
Other Machine
Memory
Application

FIG. 2

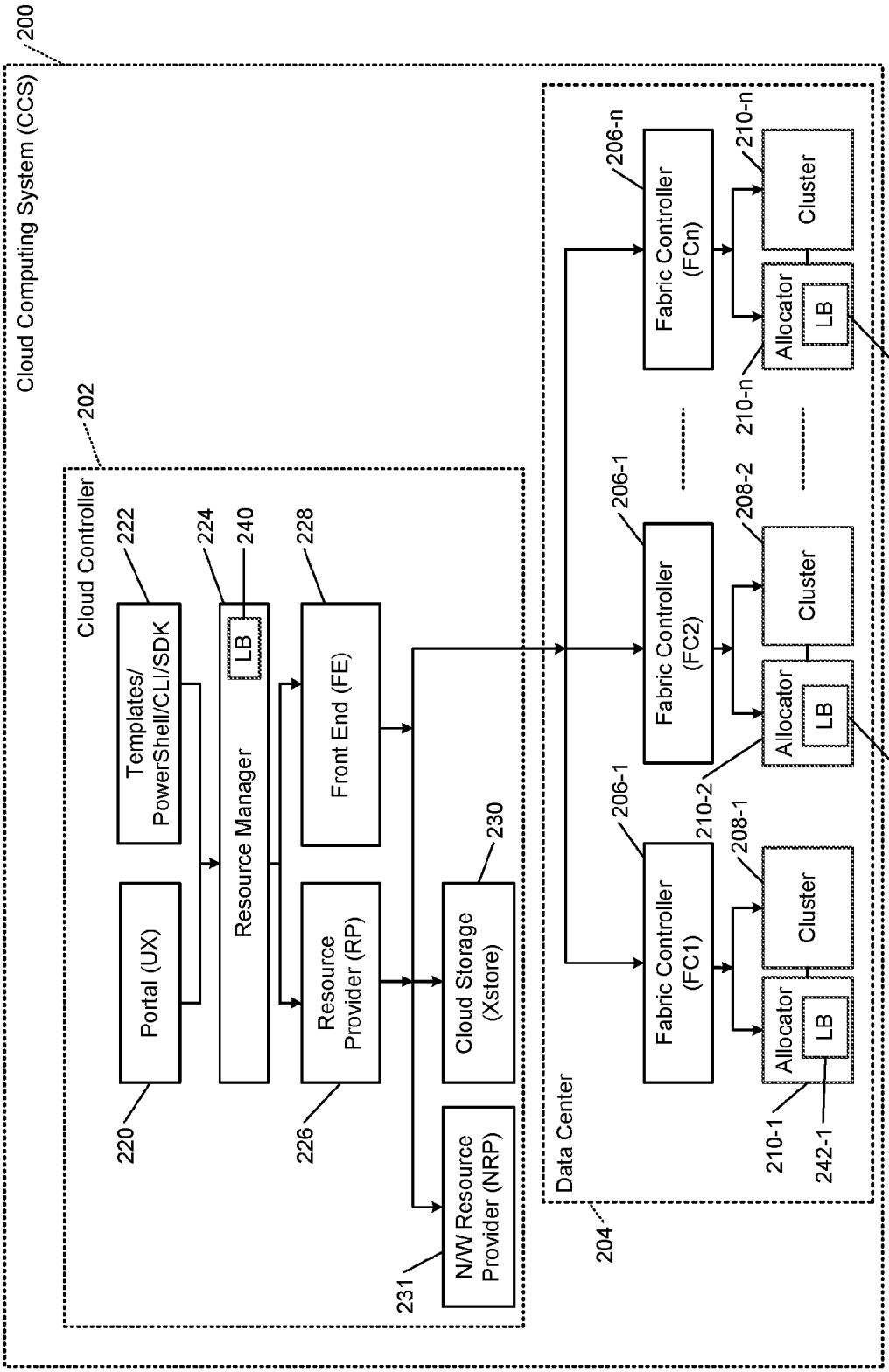
Cloud Computing System (CCS)
200
Cloud Controller
202
Templates/ PowerShell/CLI/SDK
222
Portal (UX)
220
Resource Manager
224
LB
240
Resource Provider (RP)
226
Front End (FE)
228
Cloud Storage (Xstore)
230
N/W Resource Provider (NRP)
231
Data Center
204
Fabric Controller (FC1)
206-1
Fabric Controller (FC2)
206-1
Fabric Controller (FCn)
206-n
Allocator
LB
210-1
210-2
210-n
Cluster
208-1
208-2
242-1
242-2
242-n

FIG. 3

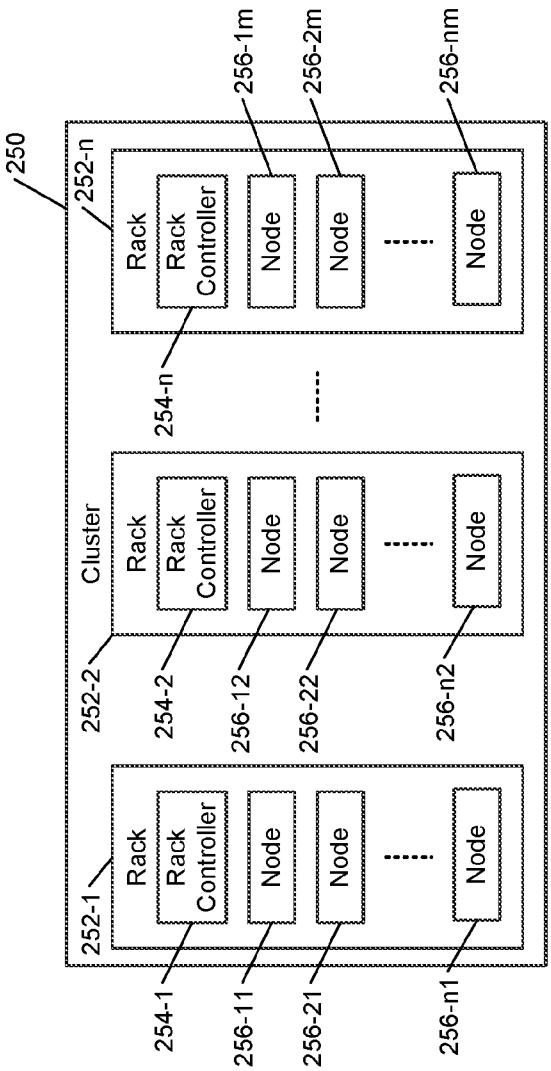
250
Cluster
252-n
252-2
252-1
Rack
Rack Controller
Node
254-n
254-2
254-1
256-1m
256-2m
256-nm
256-12
256-22
256-n2
256-11
256-21
256-n1

FIG. 4

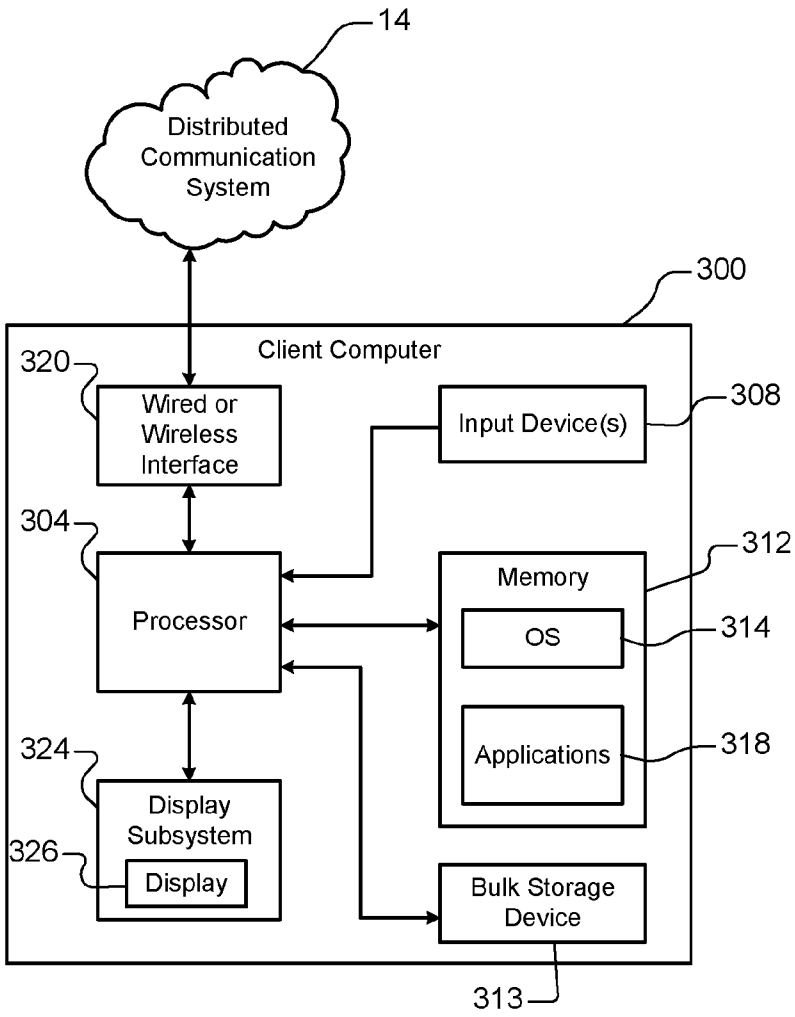
14
Distributed Communication System
300
Client Computer
320
Wired or Wireless Interface
308
Input Device(s)
304
Processor
312
Memory
OS
314
Applications
318
324
Display Subsystem
326
Display
Bulk Storage Device
313

FIG. 5

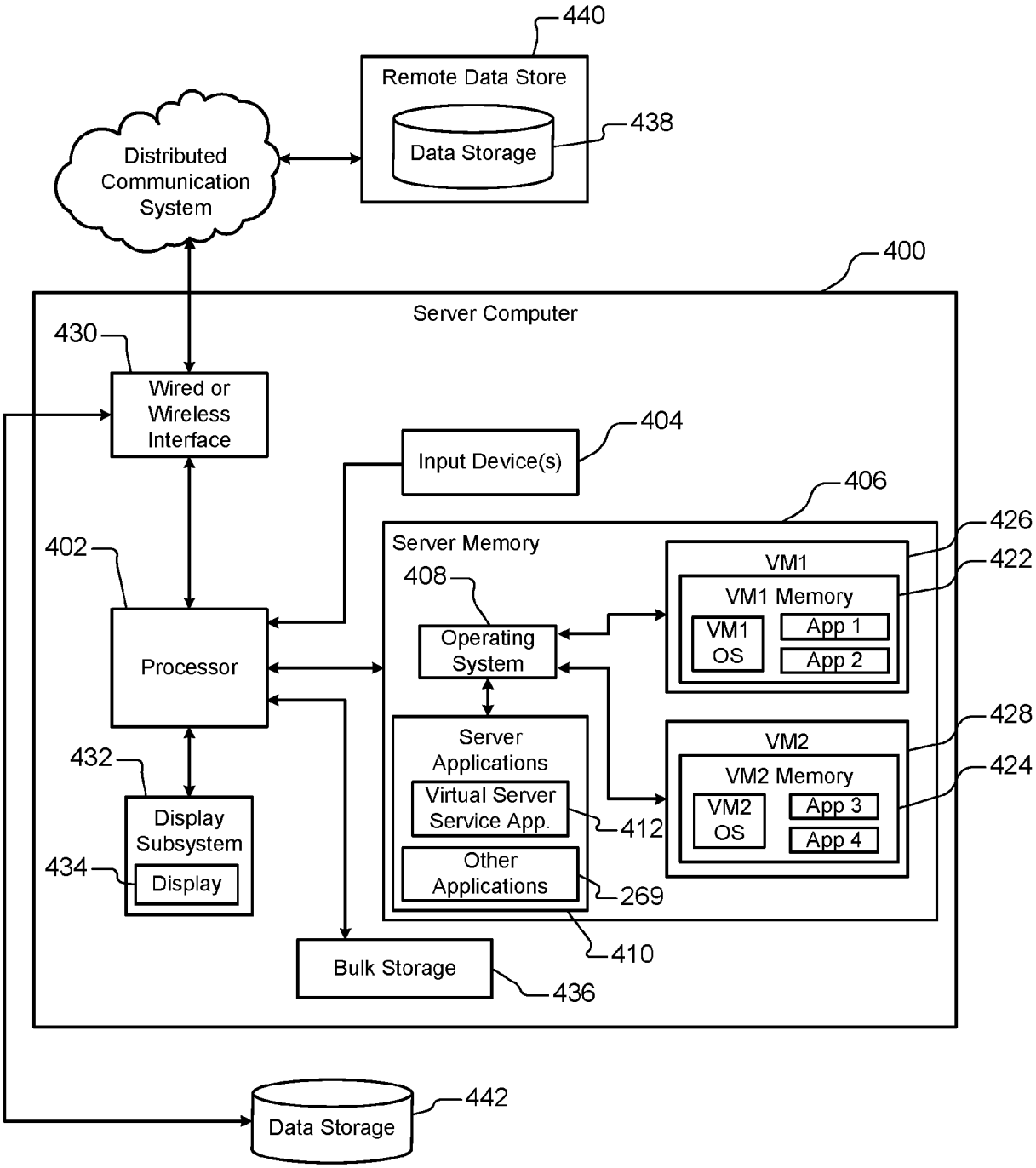
440
Remote Data Store
438
Data Storage
Distributed Communication System
400
Server Computer
430
Wired or Wireless Interface
404
Input Device(s)
406
426
Server Memory
422
402
408
VM1
VM1 Memory
VM1 OS
App 1
App 2
Operating System
Processor
428
Server Applications
VM2
424
Virtual Server Service App.
412
VM2 Memory
VM2 OS
App 3
App 4
432
Display Subsystem
Other Applications
269
434
Display
410
Bulk Storage
436
442
Data Storage

FIG. 6

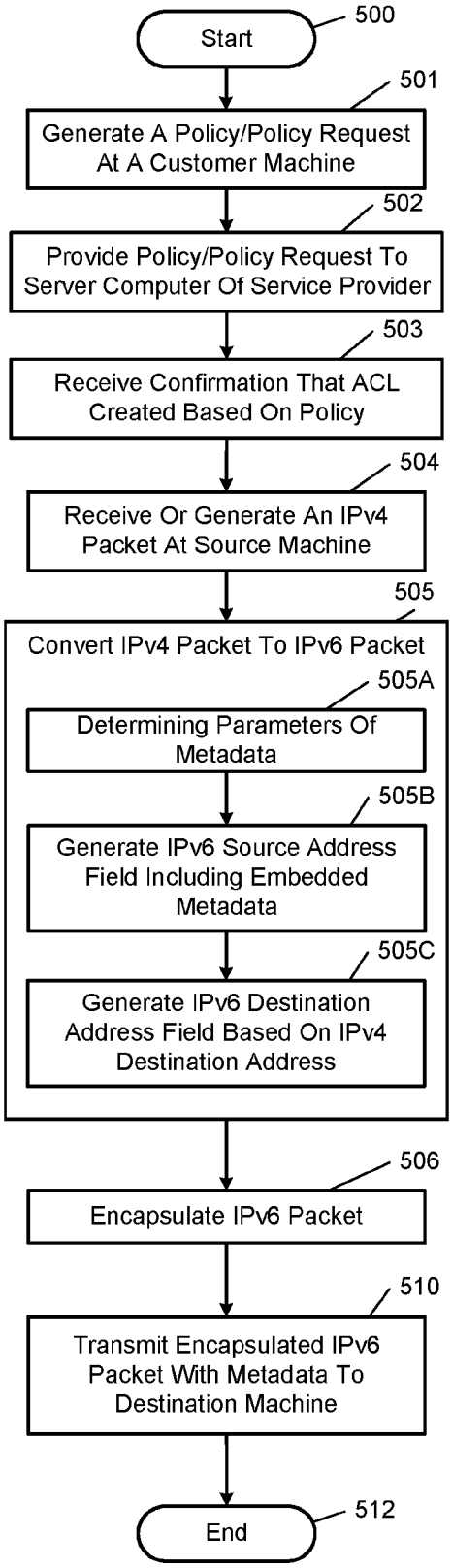
Start
500
Generate A Policy/Policy Request At A Customer Machine
501
Provide Policy/Policy Request To Server Computer Of Service Provider
502
Receive Confirmation That ACL Created Based On Policy
503
Receive Or Generate An IPv4 Packet At Source Machine
504
Convert IPv4 Packet To IPv6 Packet
505
Determining Parameters Of Metadata
505A
Generate IPv6 Source Address Field Including Embedded Metadata
505B
Generate IPv6 Destination Address Field Based On IPv4 Destination Address
505C
Encapsulate IPv6 Packet
506
Transmit Encapsulated IPv6 Packet With Metadata To Destination Machine
510
End
512

FIG. 7

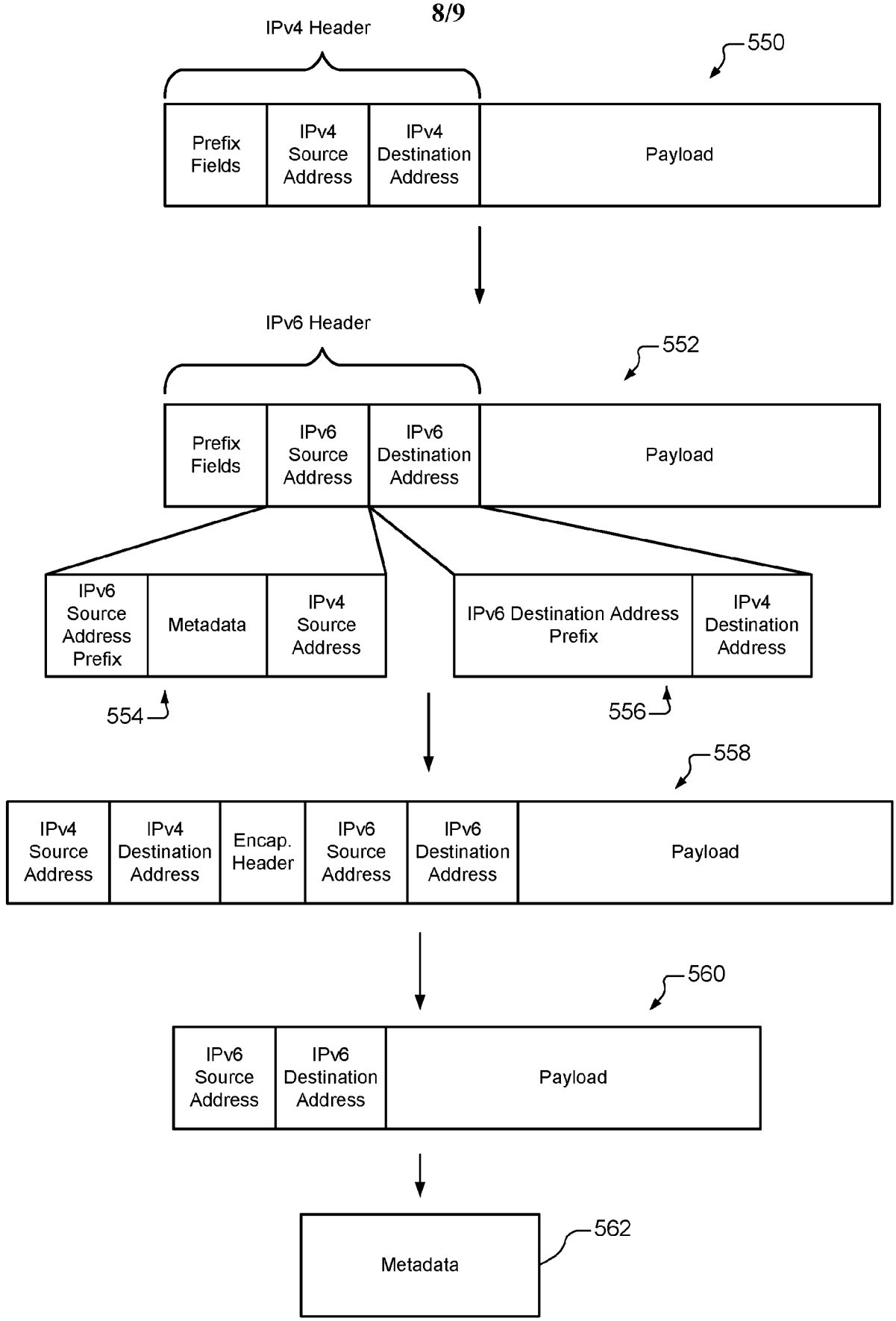
IPv4 Header
550
Prefix Fields
IPv4 Source Address
IPv4 Destination Address
Payload
IPv6 Header
552
Prefix Fields
IPv6 Source Address
IPv6 Destination Address
Payload
IPv6 Source Address Prefix
Metadata
IPv4 Source Address
IPv6 Destination Address Prefix
IPv4 Destination Address
554
556
558
IPv4 Source Address
IPv4 Destination Address
Encap. Header
IPv6 Source Address
IPv6 Destination Address
Payload
560
IPv6 Source Address
IPv6 Destination Address
Payload
Metadata
562

FIG. 8

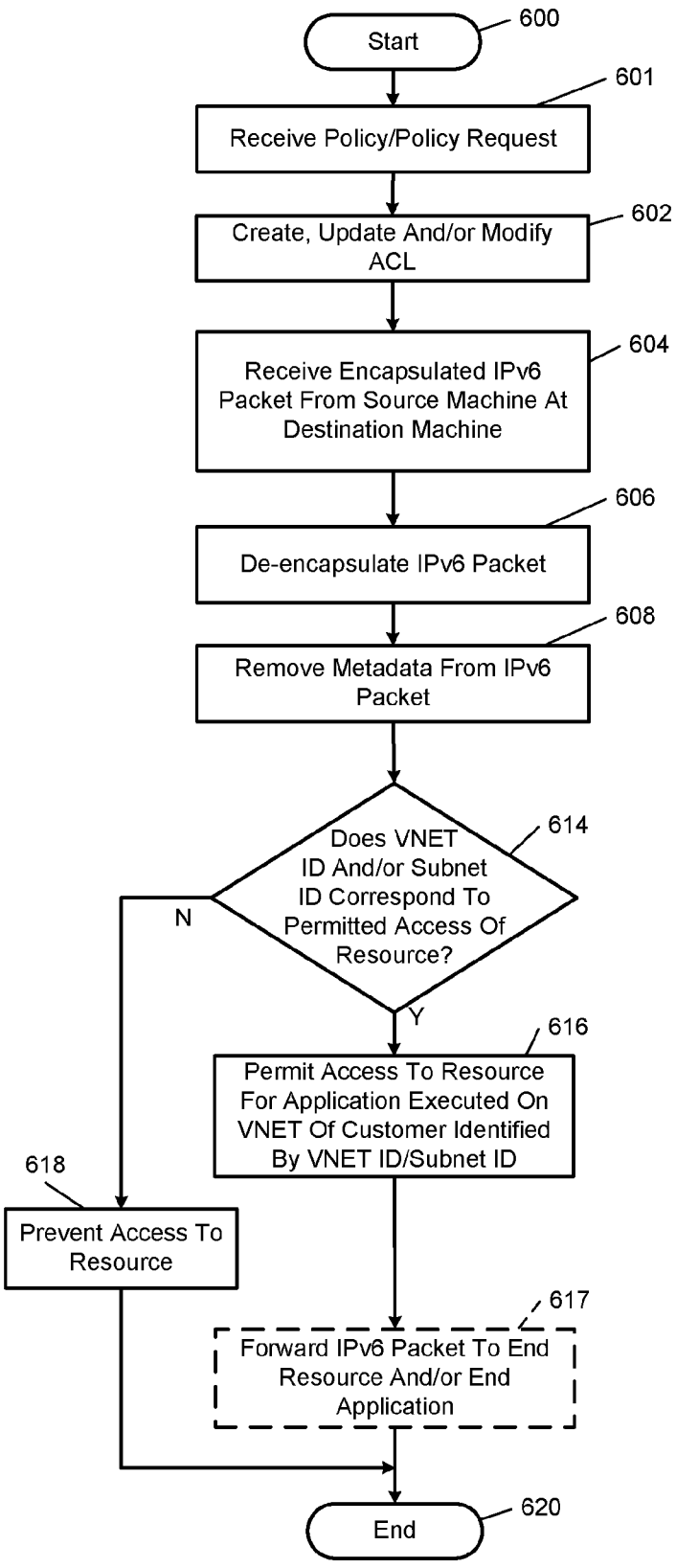
Start
600
Receive Policy/Policy Request
601
Create, Update And/or Modify ACL
602
Receive Encapsulated IPv6 Packet From Source Machine At Destination Machine
604
De-encapsulate IPv6 Packet
606
Remove Metadata From IPv6 Packet
608
Does VNET ID And/or Subnet ID Correspond To Permitted Access Of Resource?
614
N
Y
Permit Access To Resource For Application Executed On VNET Of Customer Identified By VNET ID/Subnet ID
616
Prevent Access To Resource
618
Forward IPv6 Packet To End Resource And/or End Application
617
End
620

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/034981

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06
ADD.

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

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 482 678 A1 (AT & T CORP [US]) 1 December 2004 (2004-12-01) paragraphs [0007], [0012] paragraph [0017] - paragraph [0020] paragraph [0024] - paragraph [0027] paragraph [0035] - paragraph [0039] figures 1, 2B	1-12
X A	US 2017/118173 A1 (ARRAMREDDY SUJITH [US] ET AL) 27 April 2017 (2017-04-27) paragraph [0011] paragraph [0066] - paragraph [0067] paragraph [0071] - paragraph [0073] paragraph [0094] - paragraph [0096]	1,3,4, 7-10 2,5,6, 11,12

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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	Date of mailing of the international search report
4 September 2018	10/09/2018

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Oechsner, Simon

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GROSS J ET AL: "Geneve: Generic Network Virtualization Encapsulation; draft-ietf-nvo3-geneve-04.txt", GENEVE: GENERIC NETWORK VIRTUALIZATION ENCAPSULATION; DRAFT-IETF-NVO3-GENEVE-04.TXT, INTERNET ENGINEERING TASK FORCE, IETF; STANDARDWORKINGDRAFT, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH- 1205 GENEVA, SWITZERLAND, 13 March 2017 (2017-03-13), pages 1-26, XP015118717, [retrieved on 2017-03-13] Section 3; page 9 -----	1-12

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2018/034981

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1482678 A1	01-12-2004	CA 2468480 A1	26-11-2004
		EP 1482678 A1	01-12-2004
		FR 2855697 A1	03-12-2004
		JP 2004357292 A	16-12-2004
		KR 20040101933 A	03-12-2004
		US 2005025157 A1	03-02-2005
		US 2008192771 A1	14-08-2008
US 2017118173 A1	27-04-2017	US 2017118173 A1	27-04-2017
		WO 2017070712 A1	27-04-2017