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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 July 2020 (30.07.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/152691 A1

(51) International Patent Classification:
H04L 29/12 (2006.01) *H04L 12/781* (2013.01)

(21) International Application Number:
PCT/IN2019/050046

(22) International Filing Date:
21 January 2019 (21.01.2019)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventor; and

(71) Applicant *(for SC only)*: **C H, Vyshakh Krishnan** [IN/IN]; Fern Icon, Survey No 28 and 36/5 Doddanakundi Village, Bangalore 560037 (IN).

(72) Inventor: **K, Faseela**; Flat No D0, Samhita Blossom, Outer Ring Road, Doddenakkundi, Bengaluru, Karnataka 560037, INDIA (IN).

(74) Agent: **SINGH, Manisha**; LEXORBIS, 709-710 Tolstoy House, 15-17 Tolstoy Marg, New Delhi 110001 (IN).

(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.

(54) Title: MULTI-NETWORK INTERNET PROTOCOL VERSION 6 (IPV6) DUPLICATE ADDRESS DETECTION USING ETHERNET VIRTUAL PRIVATE NETWORK (EVPN)

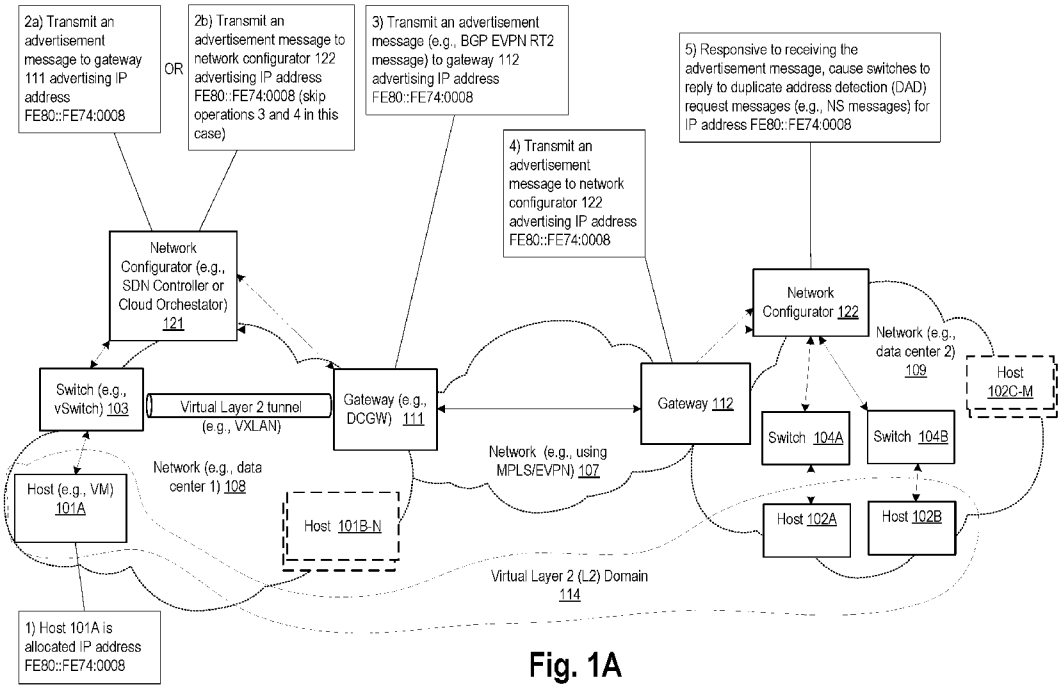


Fig. 1A

(57) Abstract: A method implemented by a first network element for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, where the first network element is in a first network of the plurality of networks. The method includes learning, based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network and causing a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address detection request message for the IP address from a host in the first network, where the host in the first network and the host in the second network are within the same virtual L2 domain.

WO 2020/152691 A1

[Continued on next page]

(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))*

Published:

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

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

MULTI-NETWORK INTERNET PROTOCOL VERSION 6 (IPV6) DUPLICATE ADDRESS DETECTION USING ETHERNET VIRTUAL PRIVATE NETWORK (EVPN)

TECHNICAL FIELD

[0001] Embodiments relate to the field of computer networks, and more specifically, to providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across multiple networks.

BACKGROUND ART

[0002] Software defined networking (SDN) is an approach to computer networking that employs a split architecture network in which the forwarding (data) plane is decoupled from the control plane. The use of a split architecture network simplifies the network devices (e.g., switches) implementing the forwarding plane by shifting the intelligence of the network into one or more controllers that oversee the switches. SDN facilitates rapid and open innovation at the network layer by providing a programmable network infrastructure. An SDN network typically includes one or more controllers that oversee multiple switches. The one or more controllers can program the switches to implement the desired forwarding behavior.

[0003] Some of the main use cases of SDN are datacenters. A datacenter is a physical building or facility that houses computing, storage, and telecommunications resources. A datacenter can include one or more performance optimized datacenters (PODs). A POD is a modular, portable, and self-contained computing environment that can be deployed with high-capacity and low operational cost in a scalable manner. Physical datacenters can be divided into virtual datacenters (e.g. virtual PODs).

[0004] In SDN controlled datacenters, the SDN controller manages and facilitates the connectivity within and between datacenters. Network elements within a datacenter can communicate with external networks and/or remote datacenters via one or more gateways (which may be referred to as datacenter gateways). Thus, gateways can provide connectivity between datacenters.

[0005] Extending Layer 2 (L2) domains across multiple datacenters has become a very common feature in cloud networking solutions. Border Gateway Protocol Ethernet Virtual Private Network (BGP EVPN) is commonly used to provide an L2 Ethernet interconnect between multiple datacenters. In an SDN controlled datacenter, the SDN controller can act as a BGP speaker that runs BGP EVPN in addition to programming the forwarding plane (e.g., using a southbound protocol such as OpenFlow). BGP EVPN network connectivity between

datacenters can also be established via non-centralized (e.g., distributed) approaches. For example, BGP EVPN network connectivity can be established using virtual routers that run BGP EVPN.

[0006] Since virtualization proliferates more machines than physical hardware would otherwise allow, many datacenters have already been forced to adopt Internet Protocol version 6 (IPv6) (due to Internet Protocol version 4 (IPv4) address exhaustion). However, this requires the datacenter to be able to support both versions (IPv4 and IPv6) while older devices transition to IPv6. Datacenters can have two types of IPv6 deployments: 1) dual stack – where both IPv4 and IPv6 interfaces are supported; and 2) IPv6 only – where IPv6 interfaces are supported.

[0007] There are three types of IPv6 addresses: 1) global unicast address; 2) unique local address; and 3) link local address. A global unicast address is an address that is globally unique and is used for routing on the internet. A unique local address is an address that is unique within an internal network or Virtual Private Network (VPN) and is used for routing internally but not on the internet. A link local address is an address that is unique on a network link and is not used for routing internally or externally (e.g., on the internet), but is used for communicating within a Local Area Network (LAN). An IPv6 link local address can be formed by appending an interface identifier to a link local prefix. The link local prefix is a known prefix that identifies the link. For example, an IPv6 link local address may be formed such that the first 64 bits are composed of the link local prefix (e.g., FE80::/64) and the remaining bits are generated based on the Media Access Control (MAC) address of the interface.

[0008] Before a host can use an IP address assigned to an interface (which can be assigned manually or automatically), the node must first perform duplicate address detection to determine whether that IP address is currently being used by another host on the link. In IPv4, this was accomplished using gratuitous Address Resolution Protocol (ARP). In IPv6, a host checks whether an IP address is being used by another host by sending a Neighbor Solicitation (NS) message to the solicited-node multicast address of that IP address. If the host subsequently receives a Neighbor Advertisement (NA) message from another node using that IP address, it determines that the IP address is already being used by another host and does not assign that IP address to its interface.

[0009] In the case of an L2 domain that spans across multiple datacenters (forming a virtual L2 domain), the same IPv6 subnet spans across multiple datacenters. Since the L2 domain is extended across multiple datacenters, any NS messages need to be sent to all hosts in all of the datacenters where the L2 domain reaches. This results in NS messages being proliferated/flooded throughout multiple datacenters. Also, the host that transmitted the NS message may have to wait for a long period of time before it receives an NA message (that is

responsive to the NS message it transmitted) since the NA message may be coming from a remote datacenter.

SUMMARY

[0010] A method is implemented by a first network element for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, where the first network element is in a first network of the plurality of networks. The method includes learning, based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network and causing a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address detection request message for the IP address from a host in the first network, where the host in the first network and the host in the second network are within the same virtual L2 domain.

[0011] A network device for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, where the network device is to implement a first network element in a first network of the plurality of networks. The network device includes a set of one or more processors and a non-transitory computer-readable storage medium having stored therein instructions, which when executed by the set of one or more processors, causes the network device to learn, based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network and cause a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address detection request message for the IP address from a host in the first network, where the host in the first network and the host in the second network are within the same virtual L2 domain.

[0012] A non-transitory computer-readable storage medium storing instructions (e.g., computer code), which when executed by one or more processors of a network device, cause the network device to perform operations for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, where the network device is to implement a first network element in a first network of the plurality of networks. The operations include learning, based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network and causing a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address

detection request message for the IP address from a host in the first network, where the host in the first network and the host in the second network are within the same virtual L2 domain.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The invention may best be understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

[0014] Figure 1A is a block diagram of a system in which efficient duplicate address detection can be provided within a virtual Layer 2 (L2) domain that spans across multiple networks, according to some embodiments.

[0015] Figure 1B is a block diagram of a network in which efficient duplicate address detection can be provided, according to some embodiments.

[0016] Figure 2 is a flow diagram of a process (e.g., by a network configurator) for providing duplicate address detection within a virtual L2 domain that spans across a plurality of networks, according to some embodiments.

[0017] Figure 3 is a flow diagram of a process (e.g., by a switch) for providing duplicate address detection within a virtual L2 domain that spans across a plurality of networks, according to some embodiments.

[0018] Figure 4A illustrates connectivity between network devices (NDs) within an exemplary network, as well as three exemplary implementations of the NDs, according to some embodiments.

[0019] Figure 4B illustrates an exemplary way to implement a special-purpose network device according to some embodiments.

[0020] Figure 4C illustrates various exemplary ways in which virtual network elements (VNEs) may be coupled according to some embodiments.

[0021] Figure 4D illustrates a network with a single network element (NE) on each of the NDs, and within this straight forward approach contrasts a traditional distributed approach (commonly used by traditional routers) with a centralized approach for maintaining reachability and forwarding information (also called network control), according to some embodiments.

[0022] Figure 4E illustrates the simple case of where each of the NDs implements a single NE, but a centralized control plane has abstracted multiple of the NEs in different NDs into (to represent) a single NE in one of the virtual network(s), according to some embodiments.

[0023] Figure 4F illustrates a case where multiple VNEs are implemented on different NDs and are coupled to each other, and where a centralized control plane has abstracted these

multiple VNEs such that they appear as a single VNE within one of the virtual networks, according to some embodiments.

[0024] Figure 5 illustrates a general purpose control plane device with centralized control plane (CCP) software), according to some embodiments.

DETAILED DESCRIPTION

[0025] The following description describes methods and apparatus for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across multiple networks. In the following description, numerous specific details such as logic implementations, opcodes, means to specify operands, resource partitioning/sharing/duplication implementations, types and interrelationships of system components, and logic partitioning/integration choices are set forth in order to provide a more thorough understanding of the present invention. It will be appreciated, however, by one skilled in the art that embodiments may be practiced without such specific details. In other instances, control structures, gate level circuits and full software instruction sequences have not been shown in detail in order not to obscure the invention. Those of ordinary skill in the art, with the included descriptions, will be able to implement appropriate functionality without undue experimentation.

[0026] References in the specification to "one embodiment," "an embodiment," "an example embodiment," etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0027] Bracketed text and blocks with dashed borders (e.g., large dashes, small dashes, dot-dash, and dots) may be used herein to illustrate optional operations that add additional features to embodiments. However, such notation should not be taken to mean that these are the only options or optional operations, and/or that blocks with solid borders are not optional in certain embodiments.

[0028] In the following description and claims, the terms "coupled" and "connected," along with their derivatives, may be used. It should be understood that these terms are not intended as synonyms for each other. "Coupled" is used to indicate that two or more elements, which may or may not be in direct physical or electrical contact with each other, co-operate or interact with

each other. “Connected” is used to indicate the establishment of communication between two or more elements that are coupled with each other.

[0029] An electronic device stores and transmits (internally and/or with other electronic devices over a network) code (which is composed of software instructions and which is sometimes referred to as computer program code or a computer program) and/or data using machine-readable media (also called computer-readable media), such as machine-readable storage media (e.g., magnetic disks, optical disks, solid state drives, read only memory (ROM), flash memory devices, phase change memory) and machine-readable transmission media (also called a carrier) (e.g., electrical, optical, radio, acoustical or other form of propagated signals – such as carrier waves, infrared signals). Thus, an electronic device (e.g., a computer) includes hardware and software, such as a set of one or more processors (e.g., wherein a processor is a microprocessor, controller, microcontroller, central processing unit, digital signal processor, application specific integrated circuit, field programmable gate array, other electronic circuitry, a combination of one or more of the preceding) coupled to one or more machine-readable storage media to store code for execution on the set of processors and/or to store data. For instance, an electronic device may include non-volatile memory containing the code since the non-volatile memory can persist code/data even when the electronic device is turned off (when power is removed), and while the electronic device is turned on that part of the code that is to be executed by the processor(s) of that electronic device is typically copied from the slower non-volatile memory into volatile memory (e.g., dynamic random access memory (DRAM), static random access memory (SRAM)) of that electronic device. Typical electronic devices also include a set or one or more physical network interface(s) (NI(s)) to establish network connections (to transmit and/or receive code and/or data using propagating signals) with other electronic devices. For example, the set of physical NIs (or the set of physical NI(s) in combination with the set of processors executing code) may perform any formatting, coding, or translating to allow the electronic device to send and receive data whether over a wired and/or a wireless connection. In some embodiments, a physical NI may comprise radio circuitry capable of receiving data from other electronic devices over a wireless connection and/or sending data out to other devices via a wireless connection. This radio circuitry may include transmitter(s), receiver(s), and/or transceiver(s) suitable for radiofrequency communication. The radio circuitry may convert digital data into a radio signal having the appropriate parameters (e.g., frequency, timing, channel, bandwidth, etc.). The radio signal may then be transmitted via antennas to the appropriate recipient(s). In some embodiments, the set of physical NI(s) may comprise network interface controller(s) (NICs), also known as a network interface card, network adapter, or local area network (LAN) adapter. The NIC(s) may facilitate in connecting the electronic device to

other electronic devices allowing them to communicate via wire through plugging in a cable to a physical port connected to a NIC. One or more parts of an embodiment may be implemented using different combinations of software, firmware, and/or hardware.

[0030] A network device (ND) is an electronic device that communicatively interconnects other electronic devices on the network (e.g., other network devices, end-user devices). Some network devices are "multiple services network devices" that provide support for multiple networking functions (e.g., routing, bridging, switching, Layer 2 aggregation, session border control, Quality of Service, and/or subscriber management), and/or provide support for multiple application services (e.g., data, voice, and video).

[0031] As mentioned above, in an L2 domain that extends across multiple datacenters (forming a virtual L2 domain), performing duplicate address detection requires sending Neighbor Solicitation (NS) messages to all hosts in all of the datacenters where the virtual L2 domain reaches. This results in NS messages being proliferated/flooded throughout multiple datacenters (incurring excessive bandwidth and processing power utilization). Also, the host that transmitted the NS message may have to wait for a long period of time before it receives a Neighbor Advertisement (NA) message (that is responsive to the NS message it transmitted) since the NA message may be coming from a remote datacenter.

[0032] Embodiments described herein provide an efficient duplicate address detection mechanism that avoids/overcomes some of the disadvantages of the existing duplicate address detection mechanisms mentioned above. An embodiment provides duplicate address detection within a virtual L2 domain that spans across multiple networks (e.g., datacenters). According to some embodiments, a first network element (e.g., a Software Defined Networking (SDN) controller or a cloud orchestrator) in a first network learns, based on receiving an advertisement message (e.g., a Border Gateway Protocol Ethernet Virtual Private Network Route Type 2 (BGP EVPN RT2) message), the Internet Protocol (IP) address allocated to a host in a second network. The first network element then causes a second network element (e.g., a switch) in the first network to reply with a duplicate address detection response message (e.g., an NA message) if the second network element receives a duplicate address detection request message (e.g., an NS message) for that IP address from a host in the first network. That is, the second network element may be configured to respond to duplicate address detection messages for that IP address (on behalf of the host using that IP address and thus acts as a "proxy" for the host using that IP address). This avoids the need to proliferate/flood the duplicate address request message for that IP address to other networks, which helps conserve bandwidth and processing power in the networks. Also, this allows a host that transmits a duplicate address detection request message to receive duplicate address detection response messages sooner (if there is another

host using that IP address) since any response messages come from a network element that is in the same network as the host (e.g., from the second network element in the first network instead of from the host in the second network).

[0033] According to some embodiments, the first network element is a BGP speaker that runs BGP EVPN. In such an embodiment, the first network element may learn the IP address allocated to the host in the second network based on receiving a BGP EVPN RT2 message. Thus, embodiments may make use of existing BGP EVPN messages to learn which IP addresses are currently being used by hosts in other networks without having to introduce additional protocols or messages. This minimizes/reduces the amount of additional overhead needed to implement the duplicate address detection mechanisms described herein.

[0034] Figure 1A is a block diagram of a system in which efficient duplicate address detection can be provided within a virtual L2 domain that spans across multiple networks, according to some embodiments. As shown in the diagram, the system includes network 108 coupled with network 109 through network 107. In one embodiment, network 108 and network 109 are datacenters (e.g., datacenter 1 and datacenter 2, respectively) and network 107 is a network that provides connectivity between the datacenters. In one embodiment, network 107 is a Multiprotocol Label Switching (MPLS) network over which BGP EVPN can be run.

[0035] As shown in the diagram, network 108 includes a network configurator 121, a switch 103, a gateway 111, and hosts 101A-N. Network configurator 121 is a network element that oversees/manages other network elements in network 108. Network configurator 121 is communicatively coupled to switch 103 and gateway 111. Network configurator 121 may configure the forwarding behavior of switch 103 and other switches in network 108 (e.g., using a southbound protocol such as OpenFlow). Network configurator 121 may act as a speaker of a reachability and forwarding protocol. For example, in one embodiment, network configurator 121 is a BGP speaker running BGP EVPN that exchanges BGP EVPN messages with other BGP speakers such as gateway 111. In one embodiment, network configurator 121 is an SDN controller, a cloud orchestrator, or a network element that combines the functionality of both an SDN controller and a cloud orchestrator. Switch 103 provides switching functionality within network 108. Switch 103 can be a physical switch or a virtual switch (vSwitch). In one embodiment, switch 103 is an SDN switch (e.g., an OpenFlow switch). Host 101A is communicatively coupled to switch 103. Each of the other hosts 101B-N may also be communicatively coupled to one or more switches (not shown) in network 108. In one embodiment, one or more of the hosts 101 are virtual machines (VMs).

[0036] In a symmetrical manner, as shown in the diagram, network 109 includes a network configurator 122, switches 104A and 104B, a gateway 112, and hosts 102A-M. Network

configurator 122 is a network element that oversees/manages other network elements in network 109. Network configurator 122 is communicatively coupled to switches 104A and 104B and gateway 112. Network configurator 122 may configure the forwarding behavior of switches 104A and 104B and other switches in network 109 (e.g., using a southbound protocol such as OpenFlow). Network configurator 122 may act as a speaker of a reachability and forwarding protocol. For example, in one embodiment, network configurator 122 is a BGP speaker running BGP EVPN that exchanges BGP EVPN messages with other BGP speakers such as gateway 112. In one embodiment, network configurator 122 is an SDN controller, a cloud orchestrator, or a network element that combines the functionality of both an SDN controller and a cloud orchestrator. Switches 104A and 104B provide switching functionality within network 109. Switches 104A and 104B can be physical switches or virtual switches (vSwitches). In one embodiment, switches 104A and 104B are SDN switches (e.g., OpenFlow switches). Host 102A is communicatively coupled to switch 104A and host 102B is communicatively coupled to switch 104B. Each of the other hosts 102C-M may also be communicatively coupled to one or more switches (not shown) in network 109. In one embodiment, one or more of the hosts 102 are virtual machines (VMs).

[0037] Network 108 and network 109 may be communicatively coupled through network 107 via their respective gateways (e.g., gateway 111 and gateway 112), which may be datacenter gateways (DCGWs). Gateway 111 and gateway 112 may communicate with each other using a routing and reachability protocol. For example, gateway 111 and gateway 112 may be BGP speakers running BGP EVPN that exchange BGP EVPN messages with each other over network 107. As mentioned above, network 107 may be an MPLS network or other suitable underlay network over which BGP EVPN can be run. Thus, gateway 111 may transmit BGP EVPN messages to gateway 112 to advertise routes in network 108. Likewise, gateway 112 may transmit BGP EVPN messages to gateway 111 to advertise routes in network 109.

[0038] Switch 103 is communicatively coupled to gateway 111 via a virtual L2 tunnel (e.g., Virtual Extensible Local Area Network (VXLAN) tunnel) implemented over an underlay network. The underlay network can be implemented using a Layer 3 protocol. Similarly, switches 104A and 104B may be communicatively coupled to gateway 112 via virtual L2 tunnels (e.g., VXLAN tunnels) (not shown) implemented over an underlay network. Again, the underlay network can be implemented using a Layer 3 protocol. The virtual L2 tunnels (e.g., VXLAN tunnels) and the interconnect between the networks (e.g., established over network 107 using BGP EVPN) can be used to extend an L2 domain across networks 108 and 109, thus creating a virtual L2 domain 114. As shown in the diagram, in an exemplary scenario, host 101A in network 108 and hosts 102A and 102B in network 109 are within the same virtual L2

domain 114. Operations for performing duplicate address detection will now be described with reference to Figs. 1A and 1B.

[0039] At operation 1, host 101A is allocated the IP address FE80::FE74:0008 (a link local IP address). This address may have been self-generated (e.g., where FE80::/64 is the link local prefix and FE74:0008 is generated based on the host's Media Access Control (MAC) address), assigned to host 101A by an IP address allocator (e.g., using Dynamic Host Control Protocol (DHCP) or similar mechanism), or assigned to host 101A manually (e.g., by a network administrator).

[0040] Since network configurator 121 (e.g., which may be an SDN controller or cloud orchestrator) oversees and manages network 108, it is aware of the IP addresses allocated to the hosts 101 in network 108, including the IP address allocated to host 101A. In one embodiment, at operation 2a, network configurator 121 transmits an advertisement message to gateway 111 advertising the IP address allocated to host 101A. In another embodiment, at operation 2b, network configurator 121 transmits an advertisement message directly to network configurator 122 (in network 109). In this case, operations 3 and 4 may be skipped. In one embodiment, network configurator 121 is a BGP speaker that runs BGP EVPN and the advertisement message is a BGP EVPN RT2 message. BGP EVPN RT2 messages are used for advertising the MAC address and IP address of hosts. In one embodiment, the BGP EVPN RT2 message may include the following fields: 1) route distinguisher (8 octets); 2) ethernet segment identifier (10 octets); 3) ethernet tag identifier (ID) (4 octets); 4) MAC address length (1 octet); 5) MAC address (6 octets); 6) IP address length (1 octet); 7) IP address (0, 4, or 16 octets); 8) MPLS label-1 (3 octets); and 9) MPLS label-2 (0 or 3 octets). Network configurator 121 may advertise the IP address of host 101A by transmitting a BGP EVPN RT2 message that specifies the IP address of host 101A in the IP address field of the BGP EVPN RT2 message.

[0041] At operation 3, responsive to receiving the advertisement message from network configurator 121, gateway 111 in turn transmits an advertisement message to gateway 112 (over network 107) advertising the IP address allocated to host 101A. This advertisement message may be an advertisement message that has a similar format as the advertisement message transmitted in operation 2a (e.g., it may be a BGP EVPN RT2 message).

[0042] At operation 4, responsive to receiving the advertisement message from gateway 111, gateway 112 in turn transmits an advertisement message to network configurator 122 advertising the IP address allocated to host 101A. This advertisement message may have a similar format as the advertisement messages transmitted in operations 2a and operation 3 (e.g., it may be a BGP EVPN RT2 message). Thus, network configurator 122 is able to learn the IP address allocated

to host 101A based on receiving the advertisement message (e.g., from gateway 112 or from network configurator 121).

[0043] At operation 5, responsive to receiving the advertisement message (e.g., from gateway 112 or from network configurator 121), network configurator 122 causes one or more of the switches that it oversees/manages (e.g., switches 104A and 104B) to reply to duplicate address detection request messages for the IP address allocated to host 101A. For example, if the switches 104 are SDN switches (e.g., OpenFlow switches), network configurator 122 may achieve this by programming suitable flow entries in the switches 104 to implement such behavior. That is, network configurator 122 may program/configure switches 104A and 104B such that if they receive a duplicate address detection request message from a host (e.g., hosts 102A or 102B) for the same IP address allocated to host 101A, they will respond by transmitting a duplicate address detection response message. This is further described with reference to Figure 1B.

[0044] As shown in Figure 1B, at operation 6, host 102B tentatively chooses the same IP address that is currently being used by host 101A (FE80::FE74:0008). The chosen IP address is "tentative" in the sense that host 102B cannot use this IP address until it verifies (using duplicate address detection) that the chosen IP address is not already being used by another host.

[0045] For this purpose, at operation 7, host 102B transmits a duplicate address detection request message for the IP address (e.g., to the solicited-node multicast address) (to determine whether the IP address is already being used by another host). As used herein, a duplicate address detection request message for an IP address is a message that is used for querying whether the IP address is being used by another host. In one embodiment, the duplicate address detection request message is an NS message (e.g., that has its source IP address set to the unspecified address and its destination IP address set to the solicited-node multicast address corresponding to the target address; also its destination MAC address set to the multicast MAC address).

[0046] In one embodiment, responsive to receiving the duplicate address detection request message (transmitted by host 102B), at operation 8a, switch 104B transmits a duplicate address detection response message for the IP address to host 102B (e.g., as programmed/configured by network configurator 122 at operation 5) without further forwarding the request message towards other networks. Alternatively, in another embodiment, at operation 8b, switch 104B forwards the duplicate address detection request message for the IP address to gateway 112. In this case, responsive to receiving the duplicate address detection request message, at operation 9, gateway 112 may transmit a duplicate address detection response message for the IP address to host 102B (e.g., via switch 104B) without further forwarding the request message towards other

networks. As used herein, a duplicate address detection response message for an IP address is a message indicating that the IP address is being used by another host. In one embodiment, the duplicate address detection response message is an NA message (e.g., that has its source IP address set to the address assigned to the interface from which the NA message is sent and its destination IP address set to the all-nodes multicast address).

[0047] Based on receiving the duplicate address detection response message, host 102B may determine that the tentatively chosen IP address is currently being used by another host and thus not assign that IP address to its interface. In this manner, switch 104B (or gateway 112) may act as a "proxy" for host 101A by transmitting a duplicate address detection response message if it sees a duplicate address detection request message for the IP address being used by host 101A.

[0048] For purposes of illustration only, the exemplary scenario shown in the diagrams and described above focuses on duplicate address detection of a single IP address (the IP address of host 101A). It should be understood, however, that network configurator 122 can learn the IP addresses allocated to other hosts in network 108 and/or the IP addresses allocated to hosts in other networks based on receiving advertisement messages and cause one or more switches 104 to respond to duplicate address detection request messages for those IP addresses.

[0049] Embodiments described herein offer several advantages over existing duplicate address detection mechanisms. For example, an advantage of embodiments described herein is that duplicate address detection request/response messages are contained within a network (e.g., network 109) and need not be propagated/flooded to other networks to which the virtual L2 domain reaches. This helps conserve bandwidth and processing power in the networks. Another advantage of embodiments described herein is that a host that transmits a duplicate address detection request message can receive duplicate address detection response messages sooner (compared to existing duplicate address detection mechanisms) since any response messages come from network elements that are within the same network as the host (e.g., by a switch directly connected to the host) instead of coming from some host in a remote network. Also, embodiments described herein can make use of existing BGP EVPN RT2 messages to advertise IP addresses to other networks without introducing additional proprietary protocols or messages, which simplifies implementation and minimizes/reduces the amount of overhead introduced to implement the duplicate address detection mechanisms described herein.

[0050] Figure 2 is a flow diagram of a process for providing duplicate address detection within a virtual L2 domain that spans across a plurality of networks, according to some embodiments. In one embodiment, the process is implemented by a first network element in a first network (which may be referred to as the local network). The operations in the flow diagrams will be described with reference to the exemplary embodiments of the other figures. However, it should

be understood that the operations of the flow diagrams can be performed by embodiments other than those discussed with reference to the other figures, and the embodiments discussed with reference to these other figures can perform operations different than those discussed with reference to the flow diagrams.

[0051] At block 210, the first network element learns, based on receiving an advertisement message, an IP address allocated to a host in a second network (which may be referred to as the remote network). In one embodiment, the first network element is a network configurator. For example, the first network element may be an SDN controller that manages/oversees switches in the local network, a cloud orchestrator, or other network element that combines the functionality of an SDN controller and a cloud orchestrator. In one embodiment, the advertisement message is a BGP EVPN RT2 message (that advertises the IP address allocated to the host in the remote network). In one embodiment, the advertisement message is received from a third network element in the remote network (e.g., from a network configurator, a gateway, and/or other BGP speaker in the remote network). In one embodiment, the IP address allocated to the host in the remote network is an IPv6 address that comprises a first portion and a second portion, where the first portion includes a link local prefix and the second portion is generated based on the MAC address of the host.

[0052] At block 220, the first network element causes a network element (a second network element) in the local network to reply with a duplicate address detection response message if that network element receives a duplicate address detection request message for the IP address from a host in the local network, where the host in the local network and the host in the remote network are within the same virtual L2 domain. In one embodiment, the second network element is a virtual switch. In one embodiment, the first network element is a datacenter gateway, where the first network element and the second network element are the same network element (the datacenter gateway learns the IP address based on receiving the advertisement message and also responds to duplicate address detection request messages for that IP address). In one embodiment, the duplicate address detection request message for the IP address is an NS message and the duplicate address detection response message is an NA message.

[0053] Figure 3 is a flow diagram of a process for providing duplicate address detection within a virtual L2 domain that spans across a plurality of networks, according to some embodiments. In one embodiment, the process is implemented by a programmable switch (e.g., an SDN switch such as an OpenFlow switch) in a first network (which may be referred to as the local network).

[0054] At block 310, the switch receives, from a network configurator in the local network, an instruction to reply with a duplicate address detection response message if a duplicate address detection request message for an IP address allocated to a host in a second network (which may

be referred to as the remote network) is received from a host in the local network, where the IP address was learned by the network configurator based on receiving an advertisement message. In one embodiment, the network configurator is an SDN controller or a cloud orchestrator in the local network (or a network element that combines the functionality of both). In one embodiment, the duplicate address detection request message for the IP address is an NS message and the duplicate address detection response message is an NA message.

[0055] At block 320, the switch receives, from the host in the local network, a duplicate address detection request message for the IP address. The IP address may be a tentative IP address for which the host in the local network is performing duplicate address detection (to determine whether the IP address is being used by another host). In one embodiment, the host is a VM (and the switch is a virtual switch).

[0056] At block 330, the switch transmits, to the host in the local network, a duplicate address detection response message (for the IP address) according to the instruction received from the network configurator, thereby allowing the host in the local network to determine that the IP address is being used by another host.

[0057] Figure 4A illustrates connectivity between network devices (NDs) within an exemplary network, as well as three exemplary implementations of the NDs, according to some embodiments. Figure 4A shows NDs 400A-H, and their connectivity by way of lines between 400A-400B, 400B-400C, 400C-400D, 400D-400E, 400E-400F, 400F-400G, and 400A-400G, as well as between 400H and each of 400A, 400C, 400D, and 400G. These NDs are physical devices, and the connectivity between these NDs can be wireless or wired (often referred to as a link). An additional line extending from NDs 400A, 400E, and 400F illustrates that these NDs act as ingress and egress points for the network (and thus, these NDs are sometimes referred to as edge NDs; while the other NDs may be called core NDs).

[0058] Two of the exemplary ND implementations in Figure 4A are: 1) a special-purpose network device 402 that uses custom application–specific integrated–circuits (ASICs) and a special-purpose operating system (OS); and 2) a general purpose network device 404 that uses common off-the-shelf (COTS) processors and a standard OS.

[0059] The special-purpose network device 402 includes networking hardware 410 comprising a set of one or more processor(s) 412, forwarding resource(s) 414 (which typically include one or more ASICs and/or network processors), and physical network interfaces (NIs) 416 (through which network connections are made, such as those shown by the connectivity between NDs 400A-H), as well as non-transitory machine-readable (e.g., computer-readable) storage media 418 having stored therein networking software 420. During operation, the networking software 420 may be executed by the networking hardware 410 to instantiate a set of one or more

networking software instance(s) 422. Each of the networking software instance(s) 422, and that part of the networking hardware 410 that executes that network software instance (be it hardware dedicated to that networking software instance and/or time slices of hardware temporally shared by that networking software instance with others of the networking software instance(s) 422), form a separate virtual network element 430A-R. Each of the virtual network element(s) (VNEs) 430A-R includes a control communication and configuration module 432A-R (sometimes referred to as a local control module or control communication module) and forwarding table(s) 434A-R, such that a given virtual network element (e.g., 430A) includes the control communication and configuration module (e.g., 432A), a set of one or more forwarding table(s) (e.g., 434A), and that portion of the networking hardware 410 that executes the virtual network element (e.g., 430A).

[0060] Networking software 420 can include code such as duplicate address detection (DAD) component 425, which when executed by networking hardware 410, causes the special-purpose network device 402 to perform operations of one or more embodiments described herein above as part of networking software instances 422 (e.g., to provide efficient duplicate address detection as described herein).

[0061] The special-purpose network device 402 is often physically and/or logically considered to include: 1) a ND control plane 424 (sometimes referred to as a control plane) comprising the processor(s) 412 that execute the control communication and configuration module(s) 432A-R; and 2) a ND forwarding plane 426 (sometimes referred to as a forwarding plane, a data plane, or a media plane) comprising the forwarding resource(s) 414 that utilize the forwarding table(s) 434A-R and the physical NIs 416. By way of example, where the ND is a router (or is implementing routing functionality), the ND control plane 424 (the processor(s) 412 executing the control communication and configuration module(s) 432A-R) is typically responsible for participating in controlling how data (e.g., packets) is to be routed (e.g., the next hop for the data and the outgoing physical NI for that data) and storing that routing information in the forwarding table(s) 434A-R, and the ND forwarding plane 426 is responsible for receiving that data on the physical NIs 416 and forwarding that data out the appropriate ones of the physical NIs 416 based on the forwarding table(s) 434A-R.

[0062] Figure 4B illustrates an exemplary way to implement the special-purpose network device 402 according to some embodiments. Figure 4B shows a special-purpose network device including cards 438 (typically hot pluggable). While in some embodiments the cards 438 are of two types (one or more that operate as the ND forwarding plane 426 (sometimes called line cards), and one or more that operate to implement the ND control plane 424 (sometimes called control cards)), alternative embodiments may combine functionality onto a single card and/or

include additional card types (e.g., one additional type of card is called a service card, resource card, or multi-application card). A service card can provide specialized processing (e.g., Layer 4 to Layer 7 services (e.g., firewall, Internet Protocol Security (IPsec), Secure Sockets Layer (SSL) / Transport Layer Security (TLS), Intrusion Detection System (IDS), peer-to-peer (P2P), Voice over IP (VoIP) Session Border Controller, Mobile Wireless Gateways (Gateway General Packet Radio Service (GPRS) Support Node (GGSN), Evolved Packet Core (EPC) Gateway)). By way of example, a service card may be used to terminate IPsec tunnels and execute the attendant authentication and encryption algorithms. These cards are coupled together through one or more interconnect mechanisms illustrated as backplane 436 (e.g., a first full mesh coupling the line cards and a second full mesh coupling all of the cards).

[0063] Returning to Figure 4A, the general purpose network device 404 includes hardware 440 comprising a set of one or more processor(s) 442 (which are often COTS processors) and physical NIs 446, as well as non-transitory machine-readable (e.g., computer-readable) storage media 448 having stored therein software 450. During operation, the processor(s) 442 execute the software 450 to instantiate one or more sets of one or more applications 464A-R. While one embodiment does not implement virtualization, alternative embodiments may use different forms of virtualization. For example, in one such alternative embodiment the virtualization layer 454 represents the kernel of an operating system (or a shim executing on a base operating system) that allows for the creation of multiple instances 462A-R called software containers that may each be used to execute one (or more) of the sets of applications 464A-R; where the multiple software containers (also called virtualization engines, virtual private servers, or jails) are user spaces (typically a virtual memory space) that are separate from each other and separate from the kernel space in which the operating system is run; and where the set of applications running in a given user space, unless explicitly allowed, cannot access the memory of the other processes. In another such alternative embodiment the virtualization layer 454 represents a hypervisor (sometimes referred to as a virtual machine monitor (VMM)) or a hypervisor executing on top of a host operating system, and each of the sets of applications 464A-R is run on top of a guest operating system within an instance 462A-R called a virtual machine (which may in some cases be considered a tightly isolated form of software container) that is run on top of the hypervisor - the guest operating system and application may not know they are running on a virtual machine as opposed to running on a "bare metal" host electronic device, or through para-virtualization the operating system and/or application may be aware of the presence of virtualization for optimization purposes. In yet other alternative embodiments, one, some or all of the applications are implemented as unikernel(s), which can be generated by compiling directly with an application only a limited set of libraries (e.g., from a library operating system

(LibOS) including drivers/libraries of OS services) that provide the particular OS services needed by the application. As a unikernel can be implemented to run directly on hardware 440, directly on a hypervisor (in which case the unikernel is sometimes described as running within a LibOS virtual machine), or in a software container, embodiments can be implemented fully with unikernels running directly on a hypervisor represented by virtualization layer 454, unikernels running within software containers represented by instances 462A-R, or as a combination of unikernels and the above-described techniques (e.g., unikernels and virtual machines both run directly on a hypervisor, unikernels and sets of applications that are run in different software containers).

[0064] The instantiation of the one or more sets of one or more applications 464A-R, as well as virtualization if implemented, are collectively referred to as software instance(s) 452. Each set of applications 464A-R, corresponding virtualization construct (e.g., instance 462A-R) if implemented, and that part of the hardware 440 that executes them (be it hardware dedicated to that execution and/or time slices of hardware temporally shared), forms a separate virtual network element(s) 460A-R.

[0065] The virtual network element(s) 460A-R perform similar functionality to the virtual network element(s) 430A-R - e.g., similar to the control communication and configuration module(s) 432A and forwarding table(s) 434A (this virtualization of the hardware 440 is sometimes referred to as network function virtualization (NFV)). Thus, NFV may be used to consolidate many network equipment types onto industry standard high volume server hardware, physical switches, and physical storage, which could be located in datacenters, NDs, and customer premise equipment (CPE). While embodiments are illustrated with each instance 462A-R corresponding to one VNE 460A-R, alternative embodiments may implement this correspondence at a finer level granularity (e.g., line card virtual machines virtualize line cards, control card virtual machine virtualize control cards, etc.); it should be understood that the techniques described herein with reference to a correspondence of instances 462A-R to VNEs also apply to embodiments where such a finer level of granularity and/or unikernels are used.

[0066] In certain embodiments, the virtualization layer 454 includes a virtual switch that provides similar forwarding services as a physical Ethernet switch. Specifically, this virtual switch forwards traffic between instances 462A-R and the physical NI(s) 446, as well as optionally between the instances 462A-R; in addition, this virtual switch may enforce network isolation between the VNEs 460A-R that by policy are not permitted to communicate with each other (e.g., by honoring virtual local area networks (VLANs)).

[0067] Software 450 can include code such as DAD component 463, which when executed by processor(s) 442, cause the general purpose network device 404 to perform operations of one or

more embodiments described herein above as part of software instances 462A-R (e.g., to provide efficient duplicate address detection as described herein).

[0068] The third exemplary ND implementation in Figure 4A is a hybrid network device 406, which includes both custom ASICs/special-purpose OS and COTS processors/standard OS in a single ND or a single card within an ND. In certain embodiments of such a hybrid network device, a platform VM (i.e., a VM that that implements the functionality of the special-purpose network device 402) could provide for para-virtualization to the networking hardware present in the hybrid network device 406.

[0069] Regardless of the above exemplary implementations of an ND, when a single one of multiple VNEs implemented by an ND is being considered (e.g., only one of the VNEs is part of a given virtual network) or where only a single VNE is currently being implemented by an ND, the shortened term network element (NE) is sometimes used to refer to that VNE. Also in all of the above exemplary implementations, each of the VNEs (e.g., VNE(s) 430A-R, VNEs 460A-R, and those in the hybrid network device 406) receives data on the physical NIs (e.g., 416, 446) and forwards that data out the appropriate ones of the physical NIs (e.g., 416, 446). For example, a VNE implementing IP router functionality forwards IP packets on the basis of some of the IP header information in the IP packet; where IP header information includes source IP address, destination IP address, source port, destination port (where "source port" and "destination port" refer herein to protocol ports, as opposed to physical ports of a ND), transport protocol (e.g., user datagram protocol (UDP), Transmission Control Protocol (TCP), and differentiated services code point (DSCP) values.

[0070] Figure 4C illustrates various exemplary ways in which VNEs may be coupled according to some embodiments. Figure 4C shows VNEs 470A.1-470A.P (and optionally VNEs 470A.Q-470A.R) implemented in ND 400A and VNE 470H.1 in ND 400H. In Figure 4C, VNEs 470A.1-P are separate from each other in the sense that they can receive packets from outside ND 400A and forward packets outside of ND 400A; VNE 470A.1 is coupled with VNE 470H.1, and thus they communicate packets between their respective NDs; VNE 470A.2-470A.3 may optionally forward packets between themselves without forwarding them outside of the ND 400A; and VNE 470A.P may optionally be the first in a chain of VNEs that includes VNE 470A.Q followed by VNE 470A.R (this is sometimes referred to as dynamic service chaining, where each of the VNEs in the series of VNEs provides a different service – e.g., one or more layer 4-7 network services). While Figure 4C illustrates various exemplary relationships between the VNEs, alternative embodiments may support other relationships (e.g., more/fewer VNEs, more/fewer dynamic service chains, multiple different dynamic service chains with some common VNEs and some different VNEs).

[0071] The NDs of Figure 4A, for example, may form part of the Internet or a private network; and other electronic devices (not shown; such as end user devices including workstations, laptops, netbooks, tablets, palm tops, mobile phones, smartphones, phablets, multimedia phones, Voice Over Internet Protocol (VOIP) phones, terminals, portable media players, GPS units, wearable devices, gaming systems, set-top boxes, Internet enabled household appliances) may be coupled to the network (directly or through other networks such as access networks) to communicate over the network (e.g., the Internet or virtual private networks (VPNs) overlaid on (e.g., tunneled through) the Internet) with each other (directly or through servers) and/or access content and/or services. Such content and/or services are typically provided by one or more servers (not shown) belonging to a service/content provider or one or more end user devices (not shown) participating in a peer-to-peer (P2P) service, and may include, for example, public webpages (e.g., free content, store fronts, search services), private webpages (e.g., username/password accessed webpages providing email services), and/or corporate networks over VPNs. For instance, end user devices may be coupled (e.g., through customer premise equipment coupled to an access network (wired or wirelessly)) to edge NDs, which are coupled (e.g., through one or more core NDs) to other edge NDs, which are coupled to electronic devices acting as servers. However, through compute and storage virtualization, one or more of the electronic devices operating as the NDs in Figure 4A may also host one or more such servers (e.g., in the case of the general purpose network device 404, one or more of the software instances 462A-R may operate as servers; the same would be true for the hybrid network device 406; in the case of the special-purpose network device 402, one or more such servers could also be run on a virtualization layer executed by the processor(s) 412); in which case the servers are said to be co-located with the VNEs of that ND.

[0072] A virtual network is a logical abstraction of a physical network (such as that in Figure 4A) that provides network services (e.g., L2 and/or L3 services). A virtual network can be implemented as an overlay network (sometimes referred to as a network virtualization overlay) that provides network services (e.g., layer 2 (L2, data link layer) and/or layer 3 (L3, network layer) services) over an underlay network (e.g., an L3 network, such as an Internet Protocol (IP) network that uses tunnels (e.g., generic routing encapsulation (GRE), layer 2 tunneling protocol (L2TP), IPSec) to create the overlay network).

[0073] A network virtualization edge (NVE) sits at the edge of the underlay network and participates in implementing the network virtualization; the network-facing side of the NVE uses the underlay network to tunnel frames to and from other NVEs; the outward-facing side of the NVE sends and receives data to and from systems outside the network. A virtual network instance (VNI) is a specific instance of a virtual network on a NVE (e.g., a NE/VNE on an ND,

a part of a NE/VNE on a ND where that NE/VNE is divided into multiple VNEs through emulation); one or more VNIs can be instantiated on an NVE (e.g., as different VNEs on an ND). A virtual access point (VAP) is a logical connection point on the NVE for connecting external systems to a virtual network; a VAP can be physical or virtual ports identified through logical interface identifiers (e.g., a VLAN ID).

[0074] Examples of network services include: 1) an Ethernet LAN emulation service (an Ethernet-based multipoint service similar to an Internet Engineering Task Force (IETF) Multiprotocol Label Switching (MPLS) or Ethernet VPN (EVPN) service) in which external systems are interconnected across the network by a LAN environment over the underlay network (e.g., an NVE provides separate L2 VNIs (virtual switching instances) for different such virtual networks, and L3 (e.g., IP/MPLS) tunneling encapsulation across the underlay network); and 2) a virtualized IP forwarding service (similar to IETF IP VPN (e.g., Border Gateway Protocol (BGP)/MPLS IPVPN) from a service definition perspective) in which external systems are interconnected across the network by an L3 environment over the underlay network (e.g., an NVE provides separate L3 VNIs (forwarding and routing instances) for different such virtual networks, and L3 (e.g., IP/MPLS) tunneling encapsulation across the underlay network)). Network services may also include quality of service capabilities (e.g., traffic classification marking, traffic conditioning and scheduling), security capabilities (e.g., filters to protect customer premises from network – originated attacks, to avoid malformed route announcements), and management capabilities (e.g., full detection and processing).

[0075] Figure 4D illustrates a network with a single network element on each of the NDs of Figure 4A, and within this straight forward approach contrasts a traditional distributed approach (commonly used by traditional routers) with a centralized approach for maintaining reachability and forwarding information (also called network control), according to some embodiments. Specifically, Figure 4D illustrates network elements (NEs) 470A-H with the same connectivity as the NDs 400A-H of Figure 4A.

[0076] Figure 4D illustrates that the distributed approach 472 distributes responsibility for generating the reachability and forwarding information across the NEs 470A-H; in other words, the process of neighbor discovery and topology discovery is distributed.

[0077] For example, where the special-purpose network device 402 is used, the control communication and configuration module(s) 432A-R of the ND control plane 424 typically include a reachability and forwarding information module to implement one or more routing protocols (e.g., an exterior gateway protocol such as Border Gateway Protocol (BGP), Interior Gateway Protocol(s) (IGP) (e.g., Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), Routing Information Protocol (RIP), Label Distribution Protocol

(LDP), Resource Reservation Protocol (RSVP) (including RSVP-Traffic Engineering (TE): Extensions to RSVP for LSP Tunnels and Generalized Multi-Protocol Label Switching (GMPLS) Signaling RSVP-TE)) that communicate with other NEs to exchange routes, and then selects those routes based on one or more routing metrics. Thus, the NEs 470A-H (e.g., the processor(s) 412 executing the control communication and configuration module(s) 432A-R) perform their responsibility for participating in controlling how data (e.g., packets) is to be routed (e.g., the next hop for the data and the outgoing physical NI for that data) by distributively determining the reachability within the network and calculating their respective forwarding information. Routes and adjacencies are stored in one or more routing structures (e.g., Routing Information Base (RIB), Label Information Base (LIB), one or more adjacency structures) on the ND control plane 424. The ND control plane 424 programs the ND forwarding plane 426 with information (e.g., adjacency and route information) based on the routing structure(s). For example, the ND control plane 424 programs the adjacency and route information into one or more forwarding table(s) 434A-R (e.g., Forwarding Information Base (FIB), Label Forwarding Information Base (LFIB), and one or more adjacency structures) on the ND forwarding plane 426. For layer 2 forwarding, the ND can store one or more bridging tables that are used to forward data based on the layer 2 information in that data. While the above example uses the special-purpose network device 402, the same distributed approach 472 can be implemented on the general purpose network device 404 and the hybrid network device 406.

[0078] Figure 4D illustrates that a centralized approach 474 (also known as software defined networking (SDN)) that decouples the system that makes decisions about where traffic is sent from the underlying systems that forwards traffic to the selected destination. The illustrated centralized approach 474 has the responsibility for the generation of reachability and forwarding information in a centralized control plane 476 (sometimes referred to as a SDN control module, controller, network controller, OpenFlow controller, SDN controller, control plane node, network virtualization authority, or management control entity), and thus the process of neighbor discovery and topology discovery is centralized. The centralized control plane 476 has a south bound interface 482 with a data plane 480 (sometime referred to the infrastructure layer, network forwarding plane, or forwarding plane (which should not be confused with a ND forwarding plane)) that includes the NEs 470A-H (sometimes referred to as switches, forwarding elements, data plane elements, or nodes). The centralized control plane 476 includes a network controller 478, which includes a centralized reachability and forwarding information module 479 that determines the reachability within the network and distributes the forwarding information to the NEs 470A-H of the data plane 480 over the south bound interface 482 (which

may use the OpenFlow protocol). Thus, the network intelligence is centralized in the centralized control plane 476 executing on electronic devices that are typically separate from the NDs.

[0079] In one embodiment, the network controller 478 may include a DAD component 481 that when executed by the network controller 478, causes the network controller 478 to perform operations of one or more embodiments described herein above (e.g., to provide efficient duplicate address detection as described herein).

[0080] For example, where the special-purpose network device 402 is used in the data plane 480, each of the control communication and configuration module(s) 432A-R of the ND control plane 424 typically include a control agent that provides the VNE side of the south bound interface 482. In this case, the ND control plane 424 (the processor(s) 412 executing the control communication and configuration module(s) 432A-R) performs its responsibility for participating in controlling how data (e.g., packets) is to be routed (e.g., the next hop for the data and the outgoing physical NI for that data) through the control agent communicating with the centralized control plane 476 to receive the forwarding information (and in some cases, the reachability information) from the centralized reachability and forwarding information module 479 (it should be understood that in some embodiments, the control communication and configuration module(s) 432A-R, in addition to communicating with the centralized control plane 476, may also play some role in determining reachability and/or calculating forwarding information – albeit less so than in the case of a distributed approach; such embodiments are generally considered to fall under the centralized approach 474, but may also be considered a hybrid approach).

[0081] While the above example uses the special-purpose network device 402, the same centralized approach 474 can be implemented with the general purpose network device 404 (e.g., each of the VNE 460A-R performs its responsibility for controlling how data (e.g., packets) is to be routed (e.g., the next hop for the data and the outgoing physical NI for that data) by communicating with the centralized control plane 476 to receive the forwarding information (and in some cases, the reachability information) from the centralized reachability and forwarding information module 479; it should be understood that in some embodiments, the VNEs 460A-R, in addition to communicating with the centralized control plane 476, may also play some role in determining reachability and/or calculating forwarding information – albeit less so than in the case of a distributed approach) and the hybrid network device 406. In fact, the use of SDN techniques can enhance the NFV techniques typically used in the general purpose network device 404 or hybrid network device 406 implementations as NFV is able to support SDN by providing an infrastructure upon which the SDN software can be run, and NFV and SDN both aim to make use of commodity server hardware and physical switches.

[0082] Figure 4D also shows that the centralized control plane 476 has a north bound interface 484 to an application layer 486, in which resides application(s) 488. The centralized control plane 476 has the ability to form virtual networks 492 (sometimes referred to as a logical forwarding plane, network services, or overlay networks (with the NEs 470A-H of the data plane 480 being the underlay network)) for the application(s) 488. Thus, the centralized control plane 476 maintains a global view of all NDs and configured NEs/VNEs, and it maps the virtual networks to the underlying NDs efficiently (including maintaining these mappings as the physical network changes either through hardware (ND, link, or ND component) failure, addition, or removal).

[0083] While Figure 4D shows the distributed approach 472 separate from the centralized approach 474, the effort of network control may be distributed differently or the two combined in certain embodiments. For example: 1) embodiments may generally use the centralized approach (SDN) 474, but have certain functions delegated to the NEs (e.g., the distributed approach may be used to implement one or more of fault monitoring, performance monitoring, protection switching, and primitives for neighbor and/or topology discovery); or 2) embodiments may perform neighbor discovery and topology discovery via both the centralized control plane and the distributed protocols, and the results compared to raise exceptions where they do not agree. Such embodiments are generally considered to fall under the centralized approach 474, but may also be considered a hybrid approach.

[0084] While Figure 4D illustrates the simple case where each of the NDs 400A-H implements a single NE 470A-H, it should be understood that the network control approaches described with reference to Figure 4D also work for networks where one or more of the NDs 400A-H implement multiple VNEs (e.g., VNEs 430A-R, VNEs 460A-R, those in the hybrid network device 406). Alternatively or in addition, the network controller 478 may also emulate the implementation of multiple VNEs in a single ND. Specifically, instead of (or in addition to) implementing multiple VNEs in a single ND, the network controller 478 may present the implementation of a VNE/NE in a single ND as multiple VNEs in the virtual networks 492 (all in the same one of the virtual network(s) 492, each in different ones of the virtual network(s) 492, or some combination). For example, the network controller 478 may cause an ND to implement a single VNE (a NE) in the underlay network, and then logically divide up the resources of that NE within the centralized control plane 476 to present different VNEs in the virtual network(s) 492 (where these different VNEs in the overlay networks are sharing the resources of the single VNE/NE implementation on the ND in the underlay network).

[0085] On the other hand, Figures 4E and 4F respectively illustrate exemplary abstractions of NEs and VNEs that the network controller 478 may present as part of different ones of the

virtual networks 492. Figure 4E illustrates the simple case of where each of the NDs 400A-H implements a single NE 470A-H (see Figure 4D), but the centralized control plane 476 has abstracted multiple of the NEs in different NDs (the NEs 470A-C and G-H) into (to represent) a single NE 470I in one of the virtual network(s) 492 of Figure 4D, according to some embodiments. Figure 4E shows that in this virtual network, the NE 470I is coupled to NE 470D and 470F, which are both still coupled to NE 470E.

[0086] Figure 4F illustrates a case where multiple VNEs (VNE 470A.1 and VNE 470H.1) are implemented on different NDs (ND 400A and ND 400H) and are coupled to each other, and where the centralized control plane 476 has abstracted these multiple VNEs such that they appear as a single VNE 470T within one of the virtual networks 492 of Figure 4D, according to some embodiments. Thus, the abstraction of a NE or VNE can span multiple NDs.

[0087] While some embodiments implement the centralized control plane 476 as a single entity (e.g., a single instance of software running on a single electronic device), alternative embodiments may spread the functionality across multiple entities for redundancy and/or scalability purposes (e.g., multiple instances of software running on different electronic devices).

[0088] Similar to the network device implementations, the electronic device(s) running the centralized control plane 476, and thus the network controller 478 including the centralized reachability and forwarding information module 479, may be implemented a variety of ways (e.g., a special purpose device, a general-purpose (e.g., COTS) device, or hybrid device). These electronic device(s) would similarly include processor(s), a set or one or more physical NIs, and a non-transitory machine-readable (e.g., computer-readable) storage medium having stored thereon the centralized control plane software. For instance, Figure 5 illustrates, a general purpose control plane device 504 including hardware 540 comprising a set of one or more processor(s) 542 (which are often COTS processors) and physical NIs 546, as well as non-transitory machine-readable storage media 548 having stored therein centralized control plane (CCP) software 550 and a DAD component 551.

[0089] In embodiments that use compute virtualization, the processor(s) 542 typically execute software to instantiate a virtualization layer 554 (e.g., in one embodiment the virtualization layer 554 represents the kernel of an operating system (or a shim executing on a base operating system) that allows for the creation of multiple instances 562A-R called software containers (representing separate user spaces and also called virtualization engines, virtual private servers, or jails) that may each be used to execute a set of one or more applications; in another embodiment the virtualization layer 554 represents a hypervisor (sometimes referred to as a virtual machine monitor (VMM)) or a hypervisor executing on top of a host operating system, and an application is run on top of a guest operating system within an instance 562A-R called a

virtual machine (which in some cases may be considered a tightly isolated form of software container) that is run by the hypervisor ; in another embodiment, an application is implemented as a unikernel, which can be generated by compiling directly with an application only a limited set of libraries (e.g., from a library operating system (LibOS) including drivers/libraries of OS services) that provide the particular OS services needed by the application, and the unikernel can run directly on hardware 540, directly on a hypervisor represented by virtualization layer 554 (in which case the unikernel is sometimes described as running within a LibOS virtual machine), or in a software container represented by one of instances 562A-R). Again, in embodiments where compute virtualization is used, during operation an instance of the CCP software 550 (illustrated as CCP instance 576A) is executed (e.g., within the instance 562A) on the virtualization layer 554. In embodiments where compute virtualization is not used, the CCP instance 576A is executed, as a unikernel or on top of a host operating system, on the "bare metal" general purpose control plane device 504. The instantiation of the CCP instance 576A, as well as the virtualization layer 554 and instances 562A-R if implemented, are collectively referred to as software instance(s) 552.

[0090] In some embodiments, the CCP instance 576A includes a network controller instance 578. The network controller instance 578 includes a centralized reachability and forwarding information module instance 579 (which is a middleware layer providing the context of the network controller 478 to the operating system and communicating with the various NEs), and an CCP application layer 580 (sometimes referred to as an application layer) over the middleware layer (providing the intelligence required for various network operations such as protocols, network situational awareness, and user – interfaces). At a more abstract level, this CCP application layer 580 within the centralized control plane 476 works with virtual network view(s) (logical view(s) of the network) and the middleware layer provides the conversion from the virtual networks to the physical view.

[0091] The DAD component 551 can be executed by hardware 540 to perform operations of one or more embodiments described herein above as part of software instances 552 (e.g., to provide efficient duplicate address detection as described herein).

[0092] The centralized control plane 476 transmits relevant messages to the data plane 480 based on CCP application layer 580 calculations and middleware layer mapping for each flow. A flow may be defined as a set of packets whose headers match a given pattern of bits; in this sense, traditional IP forwarding is also flow–based forwarding where the flows are defined by the destination IP address for example; however, in other implementations, the given pattern of bits used for a flow definition may include more fields (e.g., 10 or more) in the packet headers. Different NDs/NEs/VNEs of the data plane 480 may receive different messages, and thus

different forwarding information. The data plane 480 processes these messages and programs the appropriate flow information and corresponding actions in the forwarding tables (sometime referred to as flow tables) of the appropriate NE/VNEs, and then the NEs/VNEs map incoming packets to flows represented in the forwarding tables and forward packets based on the matches in the forwarding tables.

[0093] Standards such as OpenFlow define the protocols used for the messages, as well as a model for processing the packets. The model for processing packets includes header parsing, packet classification, and making forwarding decisions. Header parsing describes how to interpret a packet based upon a well-known set of protocols. Some protocol fields are used to build a match structure (or key) that will be used in packet classification (e.g., a first key field could be a source media access control (MAC) address, and a second key field could be a destination MAC address).

[0094] Packet classification involves executing a lookup in memory to classify the packet by determining which entry (also referred to as a forwarding table entry or flow entry) in the forwarding tables best matches the packet based upon the match structure, or key, of the forwarding table entries. It is possible that many flows represented in the forwarding table entries can correspond/match to a packet; in this case the system is typically configured to determine one forwarding table entry from the many according to a defined scheme (e.g., selecting a first forwarding table entry that is matched). Forwarding table entries include both a specific set of match criteria (a set of values or wildcards, or an indication of what portions of a packet should be compared to a particular value/values/wildcards, as defined by the matching capabilities – for specific fields in the packet header, or for some other packet content), and a set of one or more actions for the data plane to take on receiving a matching packet. For example, an action may be to push a header onto the packet, for the packet using a particular port, flood the packet, or simply drop the packet. Thus, a forwarding table entry for IPv4/IPv6 packets with a particular transmission control protocol (TCP) destination port could contain an action specifying that these packets should be dropped.

[0095] Making forwarding decisions and performing actions occurs, based upon the forwarding table entry identified during packet classification, by executing the set of actions identified in the matched forwarding table entry on the packet.

[0096] However, when an unknown packet (for example, a “missed packet” or a “match-miss” as used in OpenFlow parlance) arrives at the data plane 480, the packet (or a subset of the packet header and content) is typically forwarded to the centralized control plane 476. The centralized control plane 476 will then program forwarding table entries into the data plane 480 to accommodate packets belonging to the flow of the unknown packet. Once a specific forwarding

table entry has been programmed into the data plane 480 by the centralized control plane 476, the next packet with matching credentials will match that forwarding table entry and take the set of actions associated with that matched entry.

[0097] A network interface (NI) may be physical or virtual; and in the context of IP, an interface address is an IP address assigned to a NI, be it a physical NI or virtual NI. A virtual NI may be associated with a physical NI, with another virtual interface, or stand on its own (e.g., a loopback interface, a point-to-point protocol interface). A NI (physical or virtual) may be numbered (a NI with an IP address) or unnumbered (a NI without an IP address). A loopback interface (and its loopback address) is a specific type of virtual NI (and IP address) of a NE/VNE (physical or virtual) often used for management purposes; where such an IP address is referred to as the nodal loopback address. The IP address(es) assigned to the NI(s) of a ND are referred to as IP addresses of that ND; at a more granular level, the IP address(es) assigned to NI(s) assigned to a NE/VNE implemented on a ND can be referred to as IP addresses of that NE/VNE.

[0098] Some portions of the preceding detailed descriptions have been presented in terms of algorithms and symbolic representations of transactions on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of transactions leading to a desired result. The transactions are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0099] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[00100] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method transactions. The required structure for a variety of these systems will appear from the description above. In addition, embodiments are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of embodiments as described herein.

[00101] An embodiment may be an article of manufacture in which a non-transitory machine-readable (e.g., computer-readable) medium (such as microelectronic memory) has stored thereon instructions (e.g., computer code) which program one or more data processing components (generically referred to here as a "processor") to perform the operations described above. In other embodiments, some of these operations might be performed by specific hardware components that contain hardwired logic (e.g., dedicated digital filter blocks and state machines). Those operations might alternatively be performed by any combination of programmed data processing components and fixed hardwired circuit components.

[00102] Throughout the description, embodiments have been presented through flow diagrams. It will be appreciated that the order of transactions and transactions described in these flow diagrams are only intended for illustrative purposes and not intended as being limiting.

[00103] In the foregoing specification, embodiments have been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the invention as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method implemented by a first network element for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, wherein the first network element is in a first network of the plurality of networks, the method comprising:

learning (210), based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network; and

causing (220) a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address detection request message for the IP address from a host in the first network, wherein the host in the first network and the host in the second network are within the same virtual L2 domain.

2. The method of claim 1, wherein the first network element is a Software Defined Networking (SDN) controller that manages switches in the first network.

3. The method of claim 2, wherein the second network element is a virtual switch.

4. The method of claim 1, wherein the first network element is a cloud orchestrator.

5. The method of claim 1, wherein the first network element is a datacenter gateway, and wherein the first network element and the second network element are the same network element.

6. The method of claim 1, wherein the advertisement message is a Border Gateway Protocol (BGP) Ethernet Virtual Private Network (EVPN) Route Type 2 (RT2) message.

7. The method of claim 1, wherein the advertisement message is received from a third network element in the second network.

8. The method of claim 1, wherein the duplicate address detection request message for the IP address is a Neighbor Solicitation (NS) message and the duplicate address detection response message is a Neighbor Advertisement (NA) message.

9. The method of claim 1, wherein the IP address allocated to the host is an IP version 6 (IPv6) address that comprises a first portion and a second portion, wherein the first portion includes a link local prefix and the second portion is generated based on a Media Access Control (MAC) address of the host.

10. The method of claim 1, wherein the second network element is configured not to forward the duplicate address detection request message towards the second network.

11. A network device (404) for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, the network device to implement a first network element in a first network of the plurality of networks, the network device comprising:

a set of one or more processors (442); and

a non-transitory computer-readable storage medium (448) to store instructions (463), which when executed by the set of one or more processors, causes the network device to:

learn, based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network and

cause a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address detection request message for the IP address from a host in the first network, wherein the host in the first network and the host in the second network are within the same virtual L2 domain.

12. The network device of claim 11, wherein the first network element is a Software Defined Networking (SDN) controller that manages switches in the first network.

13. The network device of claim 12, wherein the second network element is a virtual switch.

14. The network device of claim 11, wherein the first network element is a cloud orchestrator.

15. The network device of claim 11, wherein the first network element is a datacenter gateway, and wherein the first network element and the second network element are the same network element.

16. A non-transitory computer-readable storage medium storing instructions, which when executed by one or more processors of a network device, cause the network device to perform operations for providing duplicate address detection within a virtual Layer 2 (L2) domain that spans across a plurality of networks, wherein the network device implements a first network element in a first network of the plurality of networks, the operations comprising:

learning (210), based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a second network of the plurality of networks that is different from the first network; and

causing (220) a second network element in the first network to reply with a duplicate address detection response message if the second network element receives a duplicate address detection request message for the IP address from a host in the first network, wherein the host in the first network and the host in the second network are within the same virtual L2 domain.

17. The non-transitory computer-readable storage medium of claim 16, wherein the advertisement message is a Border Gateway Protocol (BGP) Ethernet Virtual Private Network (EVPN) Route Type 2 (RT2) message.

18. The non-transitory computer-readable storage medium of claim 16, wherein the advertisement message is received from a third network element in the second network.

19. The non-transitory computer-readable storage medium of claim 16, wherein the duplicate address detection request message for the IP address is a Neighbor Solicitation (NS) message and the duplicate address detection response message is a Neighbor Advertisement (NA) message.

20. The non-transitory computer-readable storage medium of claim 16, wherein the second network element is configured not to forward the duplicate address detection request message towards the second network.

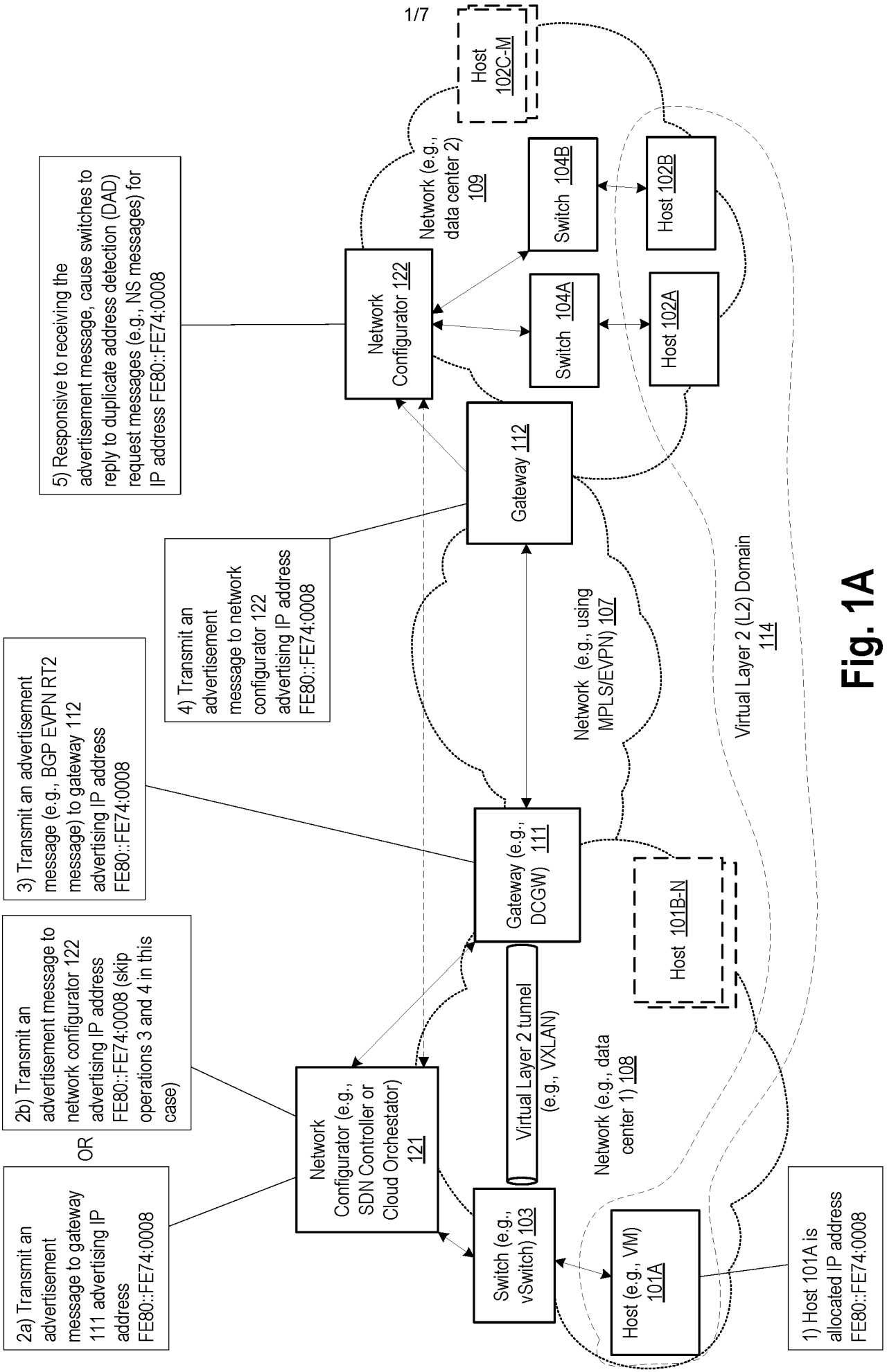

1) Host 101A is allocated IP address FE80::FE74:0008
2a) Transmit an advertisement message to gateway 111 advertising IP address FE80::FE74:0008
OR
2b) Transmit an advertisement message to network configurator 122 advertising IP address FE80::FE74:0008 (skip operations 3 and 4 in this case)
3) Transmit an advertisement message (e.g., BGP EVPN RT2 message) to gateway 112 advertising IP address FE80::FE74:0008
4) Transmit an advertisement message to network configurator 122 advertising IP address FE80::FE74:0008
5) Responsive to receiving the advertisement message, cause switches to reply to duplicate address detection (DAD) request messages (e.g., NS messages) for IP address FE80::FE74:0008
Host (e.g., VM) 101A
Switch (e.g., vSwitch) 103
Network Configurator (e.g., SDN Controller or Cloud Orchestator) 121
Virtual Layer 2 tunnel (e.g., VXLAN)
Gateway (e.g., DCGW) 111
Gateway 112
Network Configurator 122
Network (e.g., data center 1) 108
Network (e.g., using MPLS/EVPN) 107
Network (e.g., data center 2) 109
Host 101B-N
Host 102A
Host 102B
Host 102C-M
Switch 104A
Switch 104B
Virtual Layer 2 (L2) Domain 114

Fig. 1A

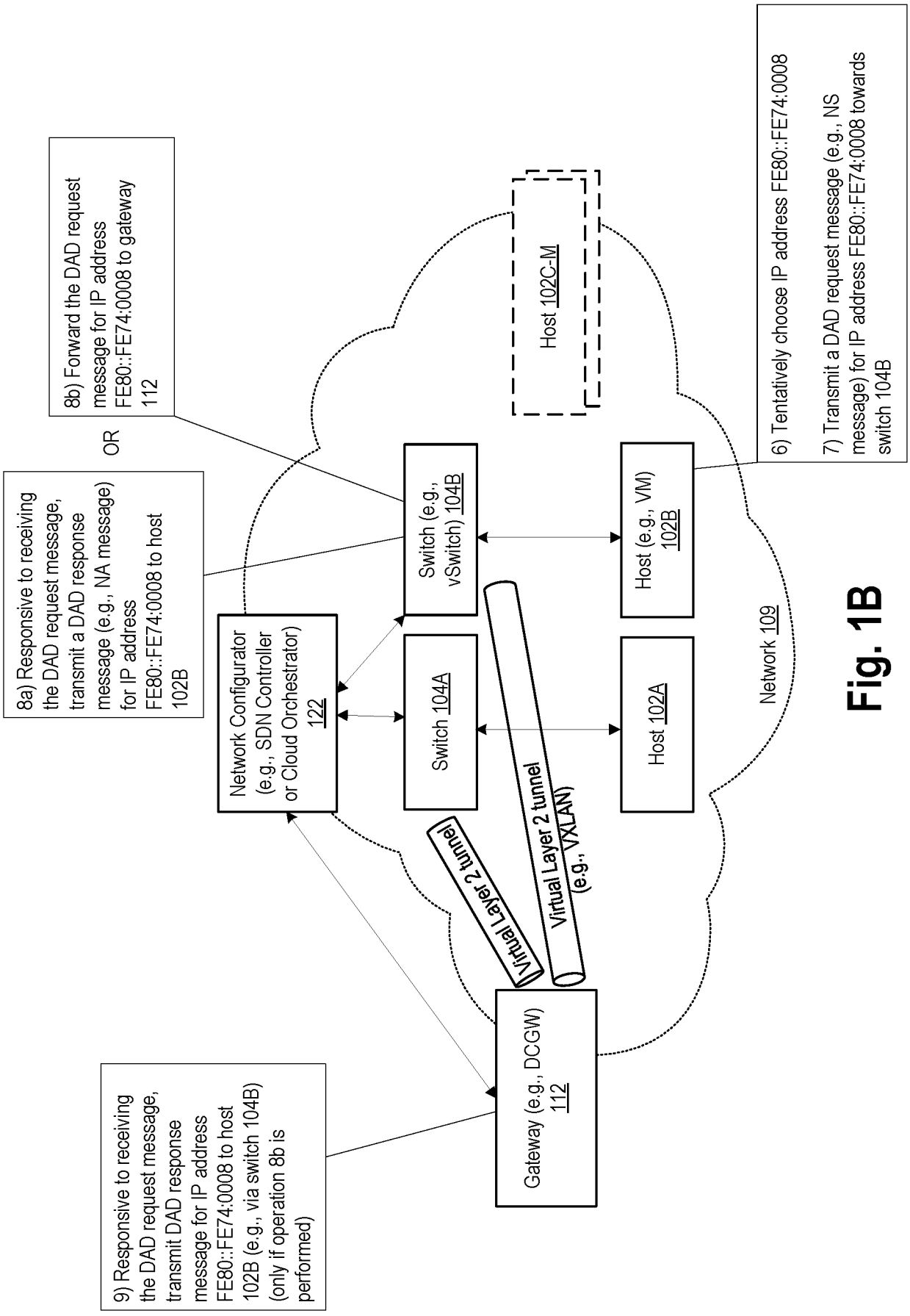

8b) Forward the DAD request message for IP address FE80::FE74:0008 to gateway 112
OR
8a) Responsive to receiving the DAD request message, transmit a DAD response message (e.g., NA message) for IP address FE80::FE74:0008 to host 102B
Network Configurator (e.g., SDN Controller or Cloud Orchestrator) 122
Switch (e.g., vSwitch) 104B
Switch 104A
Host 102C-M
Host (e.g., VM) 102B
Host 102A
Virtual Layer 2 tunnel
Virtual Layer 2 tunnel (e.g., VXLAN)
Gateway (e.g., DCGW) 112
Network 109
6) Tentatively choose IP address FE80::FE74:0008
7) Transmit a DAD request message (e.g., NS message) for IP address FE80::FE74:0008 towards switch 104B
9) Responsive to receiving the DAD request message, transmit DAD response message for IP address FE80::FE74:0008 to host 102B (e.g., via switch 104B) (only if operation 8b is performed)

Fig. 1B

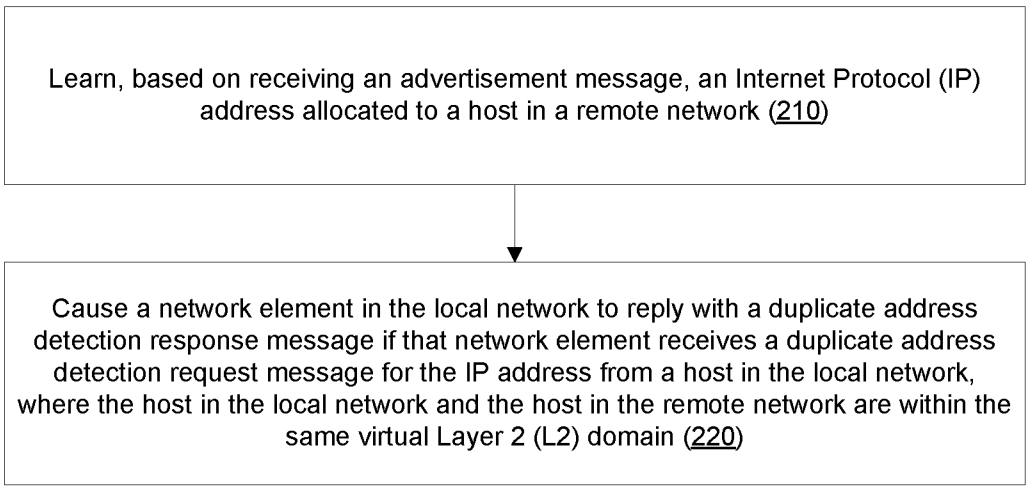

Learn, based on receiving an advertisement message, an Internet Protocol (IP) address allocated to a host in a remote network (210)
Cause a network element in the local network to reply with a duplicate address detection response message if that network element receives a duplicate address detection request message for the IP address from a host in the local network, where the host in the local network and the host in the remote network are within the same virtual Layer 2 (L2) domain (220)

Fig. 2

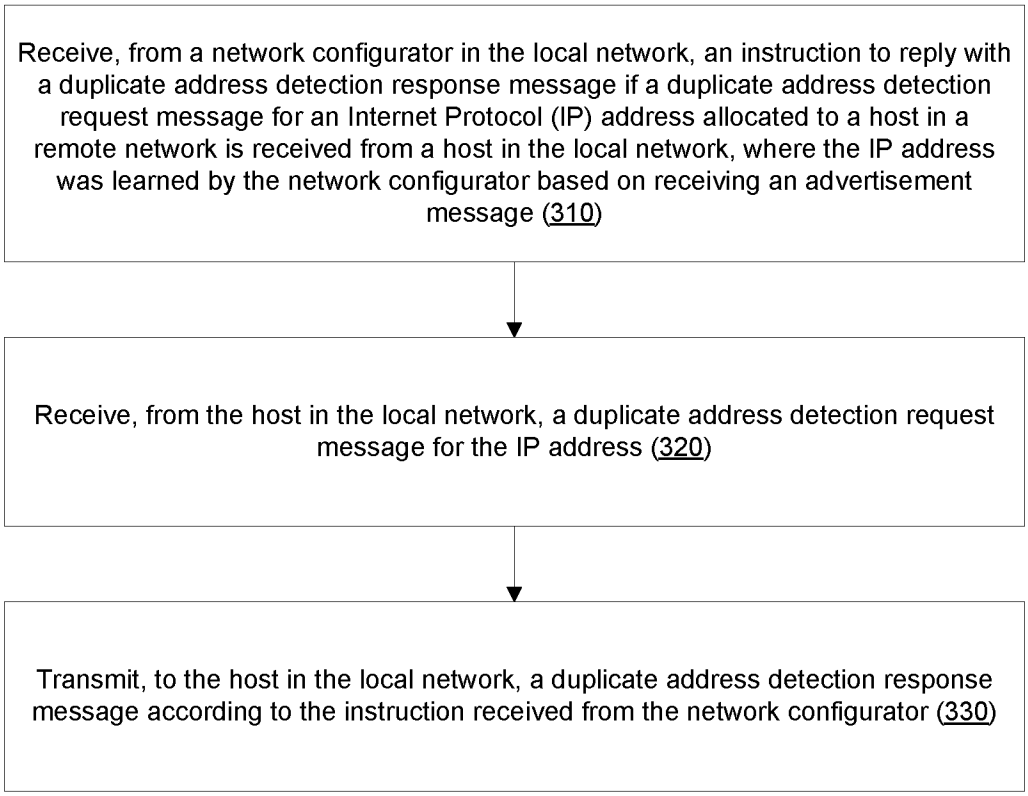

Receive, from a network configurator in the local network, an instruction to reply with a duplicate address detection response message if a duplicate address detection request message for an Internet Protocol (IP) address allocated to a host in a remote network is received from a host in the local network, where the IP address was learned by the network configurator based on receiving an advertisement message (310)
Receive, from the host in the local network, a duplicate address detection request message for the IP address (320)
Transmit, to the host in the local network, a duplicate address detection response message according to the instruction received from the network configurator (330)

Fig. 3

Fig. 4A

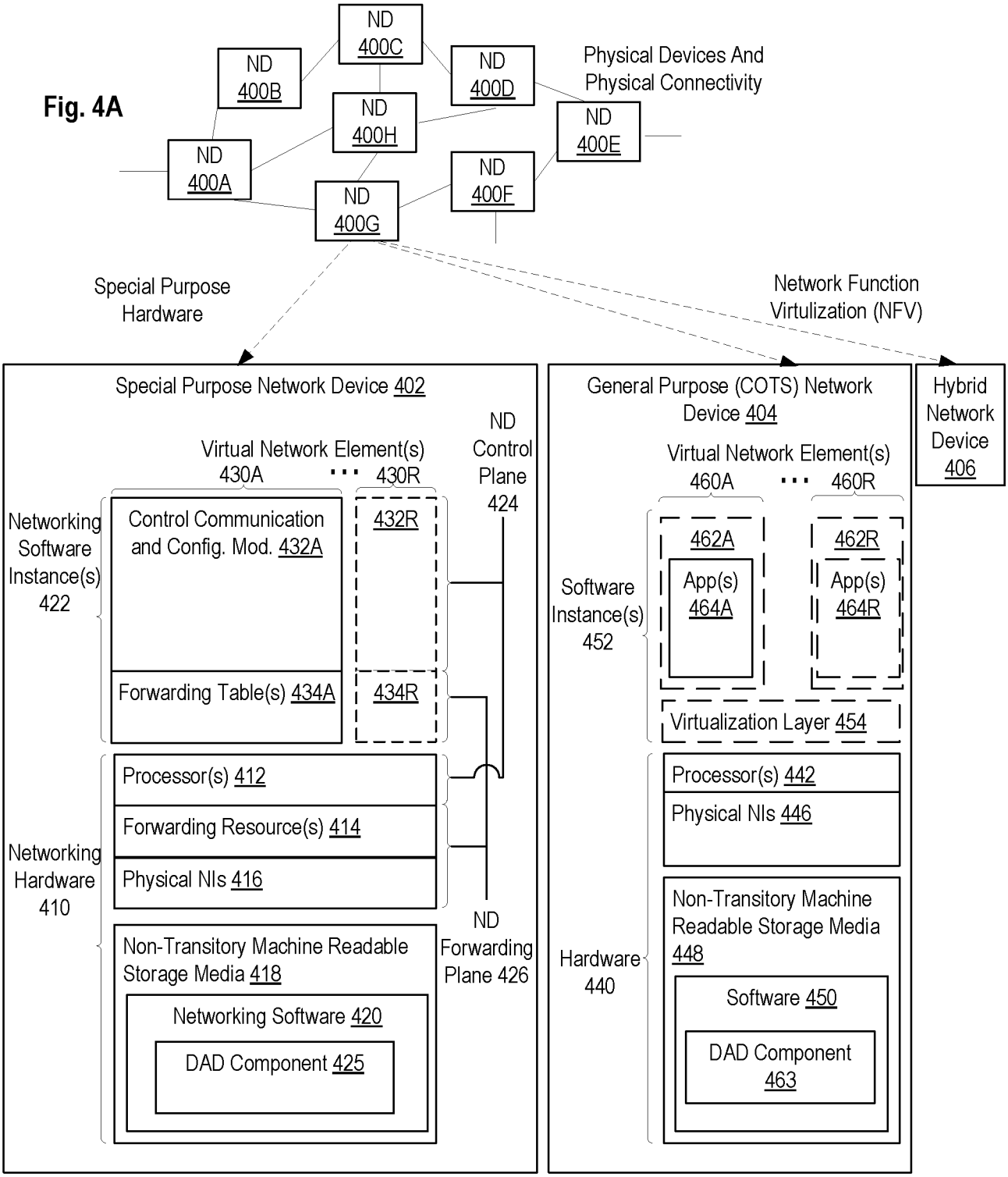

ND 400C
ND 400B
ND 400D
Physical Devices And Physical Connectivity
ND 400H
ND 400E
ND 400A
ND 400F
ND 400G
Special Purpose Hardware
Network Function Virtulization (NFV)
Special Purpose Network Device 402
Virtual Network Element(s) 430A ... 430R
ND Control Plane 424
Networking Software Instance(s) 422
Control Communication and Config. Mod. 432A
432R
Forwarding Table(s) 434A
434R
Networking Hardware 410
Processor(s) 412
Forwarding Resource(s) 414
Physical NIs 416
Non-Transitory Machine Readable Storage Media 418
Networking Software 420
DAD Component 425
ND Forwarding Plane 426
General Purpose (COTS) Network Device 404
Virtual Network Element(s) 460A ... 460R
462A
462R
App(s) 464A
App(s) 464R
Software Instance(s) 452
Virtualization Layer 454
Processor(s) 442
Physical NIs 446
Non-Transitory Machine Readable Storage Media 448
Software 450
DAD Component 463
Hardware 440
Hybrid Network Device 406

Fig. 4B

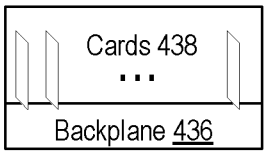

Cards 438
...
Backplane 436

Fig. 4C

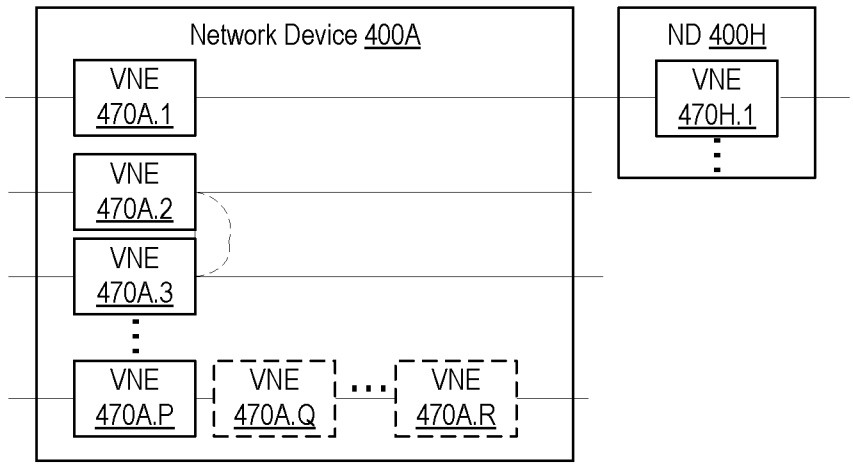


Fig. 4D

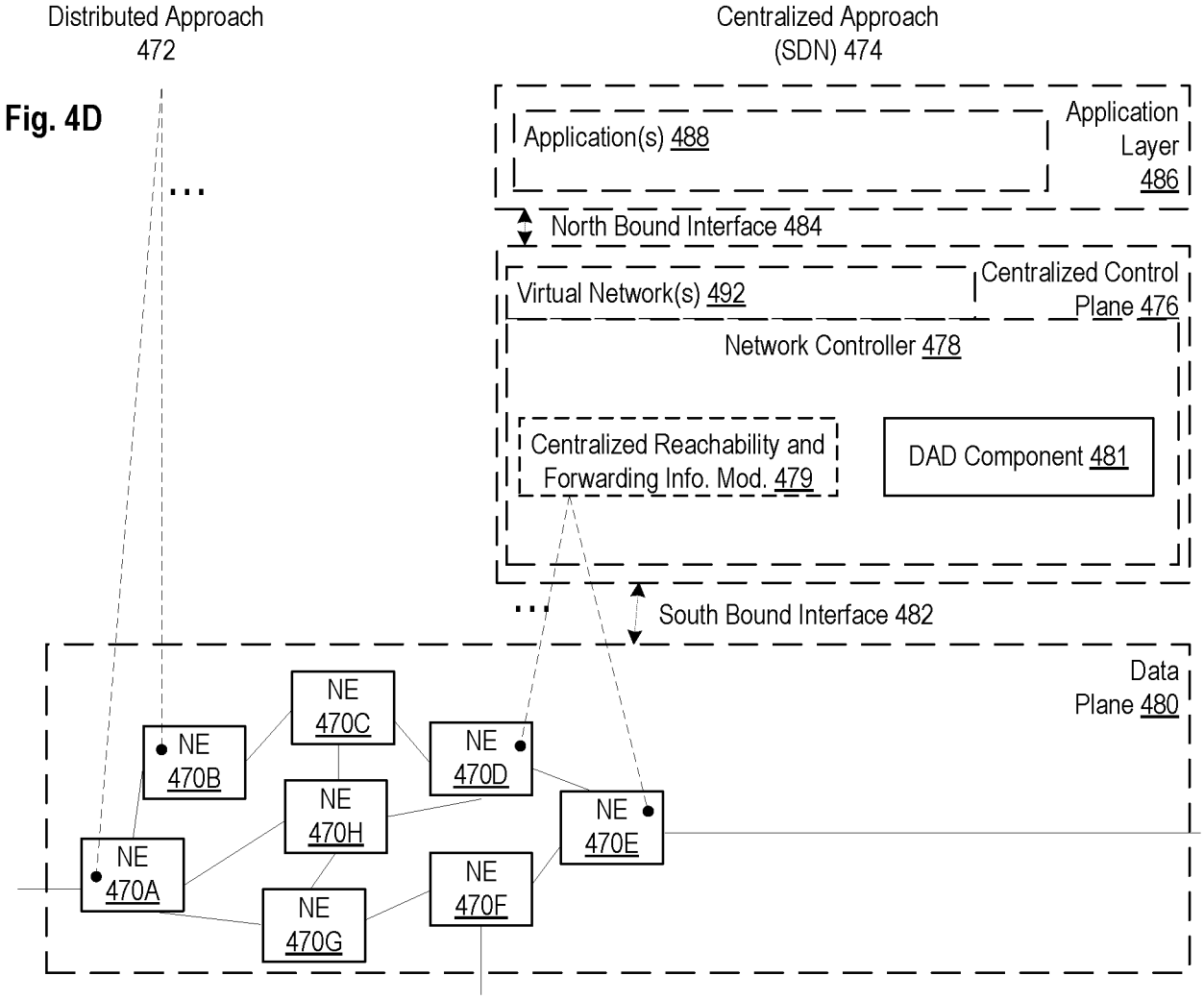


Fig. 4E

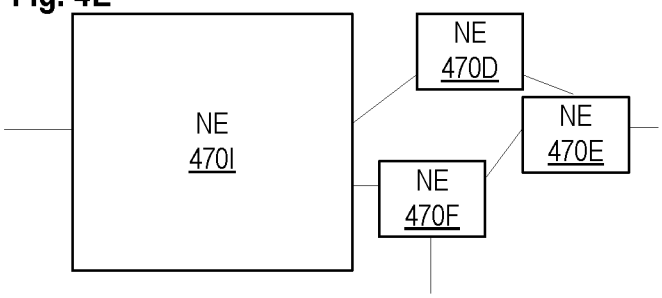


Fig. 4F

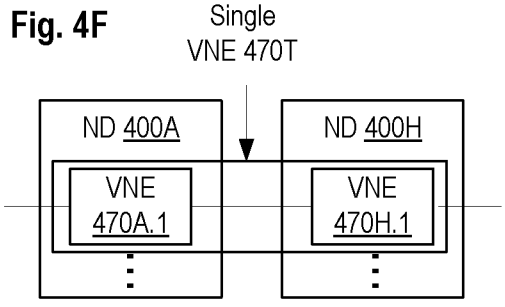
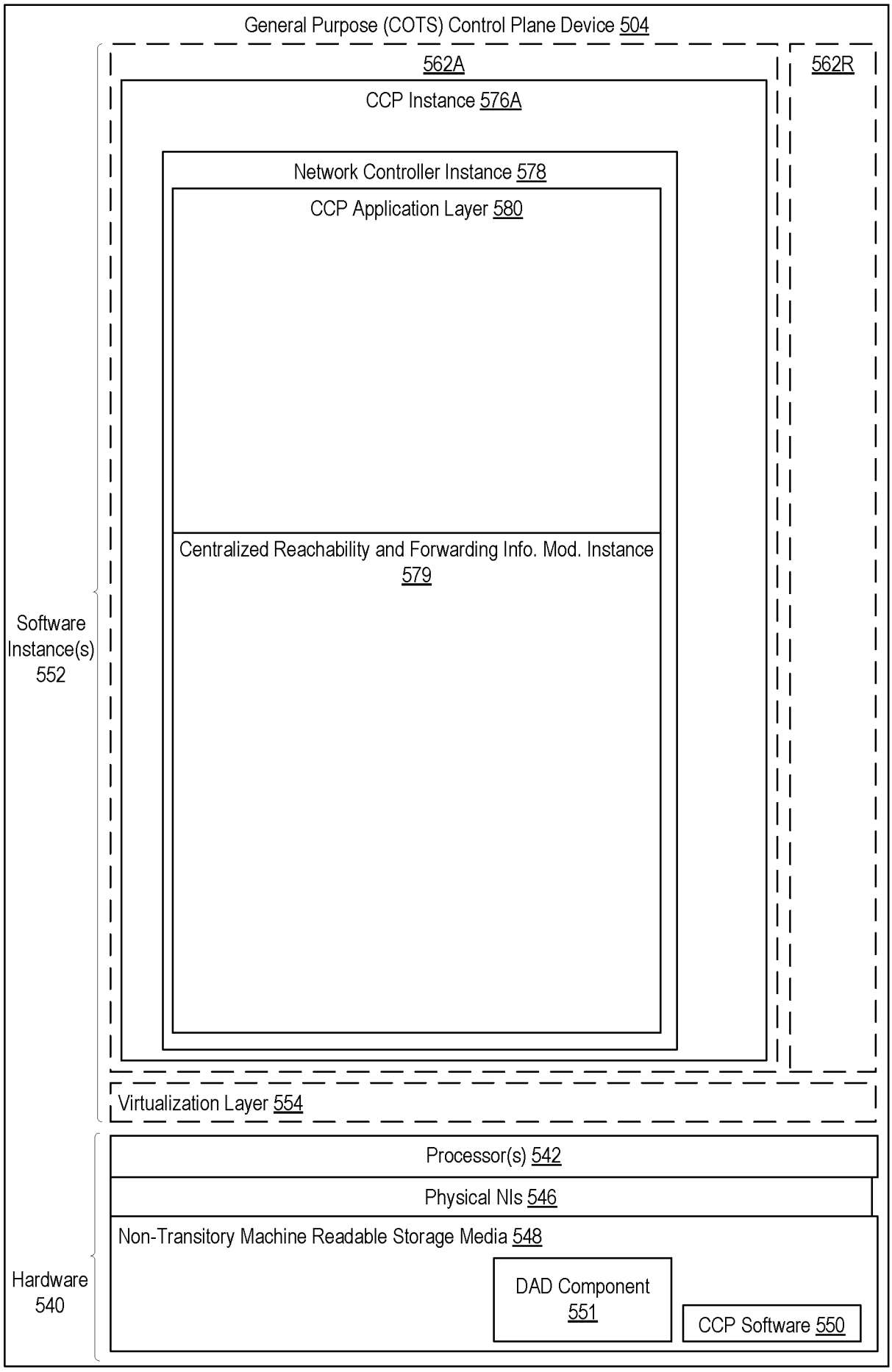


Fig. 5


General Purpose (COTS) Control Plane Device 504
562A
562R
CCP Instance 576A
Network Controller Instance 578
CCP Application Layer 580
Centralized Reachability and Forwarding Info. Mod. Instance 579
Software Instance(s) 552
Virtualization Layer 554
Processor(s) 542
Physical NIs 546
Non-Transitory Machine Readable Storage Media 548
DAD Component 551
CCP Software 550
Hardware 540

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IN2019/050046

A. CLASSIFICATION OF SUBJECT MATTER
H04L29/12,H04L12/781 Version=2019.01

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)

TotalPatent One, IPO Internal Database
Keywords:- Duplicate address detection, advertisement message, switch

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN109120741 A (ZTE CORPORATION) 01 January 2019 (01-01-2019) PARAGRAPHS [0022]-[0046], FIGURES 1a-1e	1-20
Y	CN1901551 A (BELL ALCATE CO LTD SHANGHAI) 24 January 2007 (24-01-2007) ABSTRACT, PAGES 5-7	1-20
Y	EP1450523 B1 (KABUSHIKI KAISHA TOSHIBA) 30 January 2008 (30-01-2008) PARAGRAPHS [0043]-[0044]	2-10, 12-15 & 17-20

☐ 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	Date of mailing of the international search report
29-04-2019	29-04-2019
Name and mailing address of the ISA/ Indian Patent Office Plot No.32, Sector 14,Dwarka,New Delhi-110075 Facsimile No.	Authorized officer Ajay Kumar Yadav Telephone No. +91-1125300200

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IN2019/050046

Citation	Pub.Date	Family	Pub.Date
CN 1901551 A	24-01-2007	WO 2007009367 A1	25-01-2007
EP 1450523 B1	30-01-2008	CN 1723660 A	18-01-2006
		GB 2398704 A	25-08-2004
		JP 2006518961 A	17-08-2006
		US 2004240474 A1	02-12-2004
		WO 2004075493 A2	02-09-2004