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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 March 2019 (28.03.2019)

W 1P O | PCT

(10) International Publication Number
WO 2019/060001 Al

(51) International Patent Classification:
H04L 12/911 (2013.01) *H04L 12/813* (2013.01)
H04L 12/927 (2013.01) *H04L 12/24* (2006.01)
H04L 12/869 (2013.01) *H04L 12/46* (2006.01)

(21) International Application Number:
PCT/US2018/039204

(22) International Filing Date:
25 June 2018 (25.06.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/71 1,774 2 1 September 2017 (21.09.2017) US

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

(72) Inventors: **TO, Khoa, Anh;** Microsoft Technology Li¬censing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **CARDONA, Omar;** Microsoft Technol¬ogy Licensing, LLC, One Microsoft Way, Redmond, Wash¬ington 98052-6399 (US). **FIRESTONE, Daniel;** Microsoft Technology Licensing, LLC, One Microsoft Way, Red¬mond, Washington 98052-6399 (US). **DABAGH, Alireza;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: **MINHAS, Sandip, S.** et al; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washing¬ton 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: VIRTUALIZING DCB SETTINGS FOR VIRTUAL NETWORK ADAPTERS

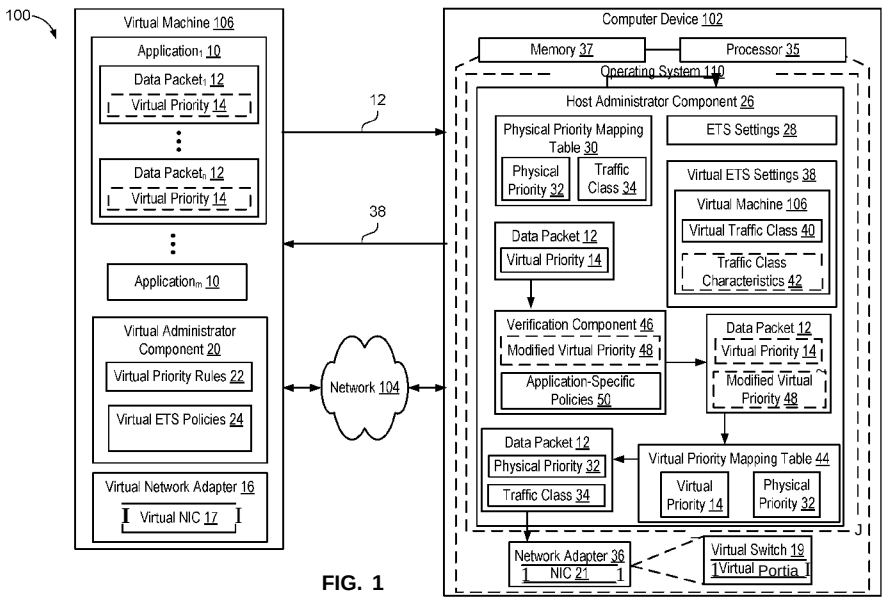


FIG. 1

(57) Abstract: Methods and devices for restricting data traffic received from a virtual machine to a subset of traffic classes may include receiving a data packet with a virtual priority from at least one virtual port. The methods and devices may include converting the virtual priority to a physical priority based on one or more priority rules. The methods and devices may include determining a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

[Continued on nextpage]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, ML, 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(H))*

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

Published:

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

VIRTUALIZING DCB SETTINGS FOR VIRTUAL NETWORK ADAPTERS

BACKGROUND

[0001] The present disclosure relates to virtual networks.

[0002] Network and host administrators configure Datacenter Bridging (DCB) settings for each physical hop in the network to segment the data pipe into multiple traffic classes (TCs) throughout the network. Applications then tag traffic properly with 802.1p priority tags, so that the traffic can be classified into one of the traffic classes. This enables end-to-end quality of service for different traffic types, based on the settings for the traffic class with which the traffic is associated. Most end-users, however, are not familiar with DCB technology, and/or are not aware of how DCB is configured in the network. The issue is further magnified for applications inside a virtual machine, since the virtual machine administrator might not be allowed to see the DCB settings of the underlying physical network.

[0003] Applications on top of virtual network interface controllers (NIC)s, such as applications inside virtual machines, currently do not have a standard mechanism to programmatically discover the datacenter bridging (DCB) policy that the host administrator has configured for a virtual NIC. Therefore DCB-aware applications cannot move seamlessly between native and virtualization environments.

[0004] Generally, a virtual machine administrator would learn about the DCB policy configured for that virtual NIC through some out-of-band mechanism, and hardcode the information to the DCB applications running on top of the virtual NICs. This prevents the ability of the applications to auto-discover DCB capabilities, as they would do on top of physical NICs, and therefore requires manual intervention from the virtual machine administrator if the DCB policy changes, or if the applications move to a different environment.

[0005] Thus, there is a need in the art for improvements in virtual network communications.

SUMMARY

[0006] The following presents a simplified summary of one or more implementations of the present disclosure in order to provide a basic understanding of such implementations. This summary is not an extensive overview of all contemplated implementations, and is intended to neither identify key or critical elements of all implementations nor delineate the scope of any or all implementations. Its sole purpose is to present some concepts of one or more implementations of the present disclosure in a simplified form as a prelude to the more

detailed description that is presented later.

[0007] One example implementation relates to a computer device. The computer device may include a memory to store data and instructions, a processor in communication with the memory, and an operating system in communication with the processor and the memory, wherein the operating system is operable to instruct a network interface controller of the computer device to: receive a data packet with a virtual priority from at least one virtual port, convert the virtual priority to a physical priority based on one or more priority rules, and determine a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

[0008] Another example implementation relates to a method for restricting data traffic received from a virtual machine to a subset of traffic classes. The method may include receiving, at a network interface controller on a computer device, a data packet with a virtual priority from at least one virtual port. The method may include converting, at the network interface controller, the virtual priority to a physical priority based on one or more priority rules. The method may include determining a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

[0009] Another example implementation relates to computer-readable medium storing instructions executable by a computer device. The computer-readable medium may include at least one instruction for causing the computer device to receive a data packet with a virtual priority from at least one virtual port. The computer-readable medium may include at least one instruction for causing the computer device to convert the virtual priority to a physical priority based on one or more priority rules. The computer-readable medium may include at least one instruction for causing the computer device to determine a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

[0010] Additional advantages and novel features relating to implementations of the present disclosure will be set forth in part in the description that follows, and in part will become more apparent to those skilled in the art upon examination of the following or upon learning by practice thereof.

DESCRIPTION OF THE FIGURES

[0011] In the drawings:

[0012] Fig. 1 is a schematic block diagram of example computer devices in accordance with

an implementation of the present disclosure;

[0013] Fig. 2 is an example physical priority mapping table in accordance with an implementation of the present disclosure;

[0014] Fig. 3 is an example virtual priority mapping table in accordance with an implementation of the present disclosure;

[0015] Fig. 4 is a schematic diagram of an example system with two virtual machines in communication with two host computer devices in accordance with an implementation of the present disclosure;

[0016] Fig. 5 is a flowchart of an example method for determining virtual ETS settings that a virtual machine may use in accordance with an implementation of the present disclosure;

[0017] Fig. 6 is a flowchart of an example method for restricting data traffic received from a virtual port to a subset of traffic classes in accordance with an implementation of the present disclosure;

[0018] Fig. 7 is a flowchart of an example method for tagging data packets with a virtual priority value in accordance with an implementation of the present disclosure; and

[0019] Fig. 8 is a schematic block diagram of an example computer device in accordance with an implementation of the present disclosure.

DETAILED DESCRIPTION

[0020] This disclosure relates to devices and methods for providing virtualized ETS information to virtual machines so that applications operating in the virtual environment may work in a similar manner as applications operating in a native environment. In addition, the devices and methods may restrict data traffic received from virtual machines to a subset of traffic classes and may further specify restrictions for different traffic types.

[0021] In a native environment, the physical NIC advertises its DCB capabilities and current settings to the operating system (OS) stack. Applications and administrators on top of those physical NICs can programmatically query the DCB settings and determine how to properly tag the traffic for the applications. As such, applications can tag traffic with different priority values in order to steer outgoing traffic to a desired traffic class. Moreover, for NIC-generated traffic, such as remote direct memory access (RDMA), administrators may use network quality of service (NetQoS) policies to tag data packets with desired priority values.

[0022] Virtual NICs (or vNICs) inside virtual machines are generally not DCB capable. For example, virtual NICs may not advertise to a virtual operating system as being DCB capable. As such, virtual machines may not be able to query for the DCB settings and/or discover the DCB settings. Moreover, even if a virtual operating system sends DCB commands, virtual

NICs may not respond to DCB commands received. As such, DCB-aware applications cannot move seamlessly between native and virtualization environments. Moreover, applications inside virtual machines generally do not have visibility into how priority tagging may affect the traffic from the applications. For example, applications within a virtual machine may not understand which priority value to tag traffic with to ensure the data packets are transmitted on a lossless traffic class. Furthermore, a host administrator may have limited options on restricting priorities on data packets coming out of a virtual machine. Typically, the host administrator may either allow virtual machines to send traffic with no priorities or send traffic with all priorities. In addition, there may not be an option for specifying RDMA traffic associated with the same network direct port (NDPort) numbers (e.g., server message block (SMB) traffic) for different virtual ports (vPorts) with different priority values.

[0023] The devices and methods may allow virtual machine administrators and applications to behave as they would in native environment, by virtualizing the ETS information that the host administrator may configure for the virtual NIC, and expose the ETS information to the virtual NIC. The devices and methods may provide the host administrator the ability to restrict outgoing traffic from each vPort to a subset of physical priorities. Each vPort may be associated with a virtual NIC (e.g., a virtual NIC is connected to a vPort). As such, any policy configured for a specific vPort may only affect traffic to and/or from that virtual NIC. The host administrator may specify which priorities each vPort may be allowed to use. The host administrator may further specify restrictions for different traffic types. For example, the host administrator may specify that a vPort can use priorities 2, 3, and 4, and furthermore, RDMA traffic is restricted to only priority 3 and 4.

[0024] The devices and methods may present applications operating in the virtual machines with virtual ETS settings information and not the ETS settings of the physical network. For example, the virtual NICs may expose the information to the virtual operating system stack, as if they are DCB-capable NICs, by using existing DCB mechanism already supported by the operating system in the virtual machine. The virtual ETS information may specify how (virtual) priorities are mapped to (virtual) traffic classes, just as in native environment. The virtual traffic classes (TCs) may correspond to the physical traffic classes that the host administrator allows the virtual machine to send traffic on.

[0025] As such, applications in the virtual machines may tag packets with any virtual priority values. When packets are sent to the host, the virtual priorities may be re-written by the host-side entity to one of the allowed physical priorities. The ability of the host operating

system and hardware to re-write packet priorities provides the mechanism for the host administrator to ensure that packets sent from the virtual machine may only go out on the channel with specified priority values, regardless of how they are tagged (or not tagged) inside the virtual machine.

[0026] In addition to priority re-writes based on a mapping table (that the host operating system provides for each vPort), the methods and devices may also further restrict traffic of certain types to a subset of the host-side priorities, if specified by the host administrator. For example, with priority re-writes, all traffic coming out of a vPort may have the virtual priorities re-mapped to physical priorities 2, 3, or 4. The host administrator may further specify, for example, that "RDMA traffic can only use priorities 3 and 4, and RDMA packets not mapped to 3 or 4 will be mapped to priority 3." As such, RDMA traffic with virtual priorities mapped to physical priorities 3 or 4 will not need any further mapping. However, RDMA traffic with virtual priorities mapped to physical priority 2 will be mapped to priority 3 instead.

[0027] By providing traffic type restrictions, backward compatibility may be provided to virtual administrator components 20 and/or applications 10 that are not aware of or know to query for DCB information. For example, if a virtual machine sends untagged Ethernet and RDMA traffics, which maps to the same priority/TC on the host side, the host administrator may steer RDMA traffic to a different (maybe lossless) TC while keeping Ethernet traffic on the original TC. Additionally, traffic type restrictions may allow NICs to offer a limited tenant DCB feature if they do not support mapping of virtual to physical priorities. In that case, DCB setting for the virtual NIC will not be virtualized. Priorities of packets out of the vPort will not be re-mapped to a list of allowed priorities. Instead, packets with priorities outside of the allowed priority set will be dropped.

[0028] By virtualizing ETS information to virtual machines, the virtual machines may tag traffic with any priority value the virtual machines may want, in a similar manner to native environments (where there is no restriction on which priorities applications can tag traffic with). In addition, virtualizing ETS information to virtual machines may allow DCB-aware virtual machine applications to work the same way they would in native environment.

[0029] As such, the devices and methods ensure that regardless of whether the applications properly classify traffic, the host administrator may ensure that uninformed and/or malicious applications transmit traffic onto the physical network with proper DCB priority tags, so that the traffic may be on the assigned class of service (e.g., the correct DCB traffic class) for the applications.

[0030] Referring now to Fig. 1, an example system 100 for providing virtualized ETS information to virtual machines so that applications operating in the virtual environment may work in a similar manner as applications operating in a native environment. System 100 may include one or more virtual machines 106 in communication with one or more computer devices 102 over a network 104. A virtual machine 106 may include, for example, software emulations that have the same characteristics as physical computers. In addition, virtual machines 106 may include a complete operating system capable of running applications 10 independently of other virtual machines 106. Computer devices 102 may host one or more virtual machines 106 and may intercept all packets coming in and out of virtual machines 106. For example, computer device 102 may include a network adapter 36 that includes a network interface controller (NIC) 21 in communication with a virtual switch 19 and a virtual port 18 that computer device 102 may use to intercept packets coming in and out of virtual machines 106. Virtual machines 106 may also include a virtual administrator component 20 operable to manage external communications via computer device 102 using virtual network adapter 16 and virtual NIC 17. Each virtual port 18 may be associated with a virtual NIC 17 (e.g., a virtual NIC 17 is connected to a virtual port 18). As such, any policy configured for a specific virtual port 18 may only affect traffic to and/or from the virtual NIC 17 associated with the virtual port 18.

[0031] Computer device 102 may include an operating system 110 executed by processor 35 and/or memory 37 of computer device 102. Memory 37 of computer device 102 may be configured for storing data and/or computer-executable instructions defining and/or associated with operating system 110, and processor 35 may execute operating system 110. An example of memory 37 can include, but is not limited to, a type of memory usable by a computer, such as random access memory (RAM), read only memory (ROM), tapes, magnetic discs, optical discs, volatile memory, non-volatile memory, and any combination thereof. An example of processor 35 can include, but is not limited to, any processor specially programmed as described herein, including a controller, microcontroller, application-specific integrated circuit (ASIC), field programmable gate array (FPGA), system on chip (SoC), or other programmable logic or state machine.

[0032] Computer device 102 may include any mobile or fixed computer device, which may be connectable to a network. Computer device 102 may be, for example, a computer device such as a desktop or laptop or tablet computer, a cellular telephone, a gaming device, a mixed reality or virtual reality device, a music device, a television, a navigation system, a camera, a personal digital assistant (PDA), or a handheld device, or any other computer

device having wired and/or wireless connection capability with one or more other devices and/or communication networks.

[0033] NIC 21 may communicate with a host administrator component 26 in order to manage data traffic for each physical hop in the physical network. For example, host administrator component 26 may partition the network adapter 36 transmit pipe into multiple Enhanced Transmission Selection (ETS) channels and assign traffic classes 34 to each ETS channel. Different ETS channels may have different quality of service for traffic transmitted on the ETS channel. For example, one channel may provide lossless traffic, while another channel may periodically drop data packets transmitted on the channel and/or have a delay in transmission. In addition, each traffic class may be allocated a percentage of available bandwidth to use for transmitting traffic. Network adapters 36 operating on the same physical network may be partitioned the same way so the entire network has consistent channel partitioning.

[0034] Host administrator component 26 may also configure ETS settings 28 for the physical network and virtual ETS settings 38 for virtual machines 106 and may communicate ETS settings 28 and the virtual ETS settings 38 to NIC 21. For example, host administrator component 26 may assign physical priorities 32 for data packets to a specific traffic class 34. Physical priorities 32 may correspond to 802.1p priorities (e.g., 0 to 7) and may be used for tagging data packets that are transmitted on the physical network. Applications may tag data traffic with 802.1p priority tags, so that the data traffic may be classified into one of the traffic classes. This enables end-to-end quality of service for different data traffic types, based on the settings for the traffic class. As such, the physical priorities 32 may determine an ETS traffic channel the data packet may be transmitted on in the physical network. A traffic class 34 may have one or more physical priorities 32 assigned to the traffic class 34. Thus, data packets with different physical priorities 32 may be transmitted on the same traffic channel. Host administrator component 26 may store the associations between the physical priority 32 and the traffic class 34, for example, in a physical priority mapping table 30, although other mechanisms of storing the associations may be utilized.

[0035] For instance, referring now to Fig. 2, an example physical priority mapping table 30 for traffic classes 34 may be configured by host administrator component 26 for the physical NIC. For example, physical priority mapping table 30 may identify one or more traffic classes 34 on the physical network. In addition, host administrator component 26 may associate one or more physical priorities 32 to a respective traffic class 34 and may identify

whether the traffic channel is Priority-based Flow Control (PFC) enabled 202, e.g., by listing the associated information in a same row of the table 30. For example, traffic class 0 may be mapped to priority values 0, 1, 2, and 3 and may not be PFC enabled, traffic class 1 may be mapped to priority values 4 and 5 and may not be PFC enabled, and traffic class 2 may be mapped to priority value 6 and may be PFC enabled, and traffic class 3 may be mapped to priority 7 and may be PFC enabled.

[0036] Referring back to Fig. 1, host administrator component 26 may be further operable to restrict data traffic received from virtual machines 106 by specifying which traffic classes 34 each virtual machine 106 may be allowed to use. Host administrator component 26 may determine virtual ETS settings 38 for each virtual machine 106. The virtual ETS settings 38 may include, but are not limited to, one or more virtual traffic classes 40 assigned to virtual machine 106 and traffic class characteristics 42 (e.g., lossless traffic channel). The virtual traffic classes 40 may correspond to the physical traffic classes 34 that host administrator component 26 allows virtual machine 106 to send traffic on. For example, host administrator component 26 may provide access to four traffic channels (e.g., Ethernet, RDMA, Small Computer System Interface *(SCSI),* and TCP), where each traffic channel corresponds to a different physical traffic class 34. One virtual machine may pay less relative to other virtual machines and may only receive access to the Ethernet channel. Another virtual machine may pay more relative to other virtual machines and may be able to access all four channels. As such, the virtual ETS settings 38 may be different for each of the virtual machines 106.

[0037] The virtual ETS settings 38 information may be generated by using operating system 110 internal heuristics, based on, for example, a tenant DCB policy the host administrator component 26 has configured for a specific virtual port 18. For example, the host administrator component 26 may allow the virtual machine 106 to send traffic on physical priorities 2, 3 and 4, which maps to physical traffic classes on the physical NIC. Furthermore, priority 4 may be PFC enabled on the physical network. Host administrator component 26 may expose the virtual ETS settings 38 to virtual machine 106, where the virtual ETS settings 38 indicate that virtual machine 106 has three TCs, one of which is PFC enabled. For example, priority 0-5 maps to TC0, priority 6 maps to TCI, priority 7 maps to TC2, and is PFC enabled. This enables one or more (e.g., up to *m*, where *m* is an integer) DCB-aware applications 10 inside the virtual machine 106 to know that application 10 may send traffic out of three traffic channels, and if application 10 wants to send lossless traffic, application 10 may tag traffic with priority 7.

[0038] Host administrator component 26, may transmit a message to virtual machine 106

with the virtual ETS settings 38 for virtual machine 106. A virtual administrator component 20 on virtual machine 106 may receive the virtual ETS settings 38 and may create virtual ETS policies 24 that map from virtual priorities 14 assigned to data packets 12 to the virtual traffic classes 40 received in the virtual ETS settings 38. Virtual administrator component 20 may also create one or more virtual priority rules 22 that determine how applications 10 may tag data packets 12. For example, the virtual priority rules 22 may indicate that SMB storage traffic may be tagged with a virtual priority value 5 and lossless traffic may be tagged with a virtual priority value 7. As such, applications 10 operating on virtual machine 106 may have the ability to tag data packets 12 with any priority value in a similar manner as applications operating in a native environment.

[0039] Virtual administrator component 20 and applications 10 may be agnostic of how the virtual priorities 14 are mapped to the physical network. As such, applications 10 may easily move between different host environments. When applications 10 move between different host environments, applications 10 may continue to use the virtual priority rules 22 to tag data packets 12 and may be unaware that a change in host environment occurred. Moreover, by restricting the information provided to virtual administrator component 20 and applications 10 about the physical network, virtual administrator component 20 and/or applications 10 may be less likely to learn information about the physical network. As such, the physical network topology may be protected from virtual machines that may want to harm the physical network and/or perform malicious actions on the physical network.

[0040] Virtual NIC 17 may transmit one or more data packets 12 to host administrator component 26 and/or NIC 21 via virtual port 18. Each of the data packets 12 may include a virtual priority 14 assigned to the data packet 12 by application 10 based on one or more virtual priority rules 22.

[0041] Host administrator component 26 may include a verification component 46 operable to verify whether the virtual priority 14 of data packet 12 corresponds to application-specific policies 50 that may be defined for application 10. Verification component 46 may check whether the virtual priority 14 of data packets 12 received from application 10 correspond to the correct priority value for the application-specific policy 50. If the virtual priority 14 does correspond to the correct priority value, verification component 46 may not need to perform additional processing on data packet 12. However, if the virtual priority 14 does not correspond to the correct priority value, verification component 46 may need to replace the virtual priority 14 of data packet 12 with a modified virtual priority 48 based on the application-specific policy 50. For example, an application-specific policy 50 may include

mapping RDMA traffic to priority value 5. If a RDMA data packet was tagged with a virtual priority value of 2, verification component 46 may replace the virtual priority value 2 with a modified virtual priority value 5. Moreover, if virtual machine 106 sends untagged Ethernet and RDMA traffics, which maps to the same physical priority 32, verification component 46 may use the application-specific policy 50 to tag the RDMA traffic with a physical priority 32 of a different (maybe lossless) traffic class 34, while keeping the Ethernet traffic tagged with the original physical priority 32 that maps to the original traffic class 34. When an application-specific policy 50 does not exist, host administrator component 26 may apply a default virtual priority value for untagged data packets 12 received from virtual port 18. In addition, host administration component 26 may instruct NIC 21 to apply a default virtual priority value for untagged data packets 12 received from virtual port 18.

[0042] In addition, host administrator component 26 may use a virtual priority mapping table 44 to replace the virtual priority 14 of the received data packet 12 with a corresponding physical priority 32. Host administrator component 26 may have a virtual priority mapping table 44 for each virtual machine 106. By using the virtual priority mapping table 44, host administrator component 26 may ensure that data packets 12 received from virtual machine 106 are transmitted using the correct physical priorities 32 assigned to the virtual machine 106. Thus, regardless of how virtual machine 106 tagged data packet 12, host administrator component 26 may restrict traffic from virtual machine 106 to the allowed traffic classes 34. If virtual machine 106 attempts to use different traffic channels than the traffic channels assigned to virtual machine 106, the virtual priority mapping table 44 may be used to restrict the data traffic received from virtual machine 106 to the correct traffic channels.

[0043] For instance, referring now to Fig. 3, an example virtual priority mapping table 44 associates virtual priority 14 values with a corresponding physical priority 32 value. The physical priority 32 value may also be referred to as a host-side priority, e.g., a priority value established by a host computer device that hosts the virtual machine and intercepts all packets coming in and out of virtual machine 106 in order to re-classify packets with the physical priority 32 values. For example, virtual priority value 0 may correspond to physical priority value 5, virtual priority value 1 may correspond to physical priority value 6, virtual priority value 2 may correspond to physical priority value 5, virtual priority value 3 may correspond to physical priority value 5, virtual priority value 4 may correspond to physical priority value 5, virtual priority value 5 may correspond to physical priority value 5, virtual priority value 6 may correspond to physical priority value 5, and virtual priority value 7 may

correspond to physical priority value 5. Virtual priority mapping table 44 may further include priority mappings 302 between the traffic classes 34 and the physical priorities 32 and whether the traffic class 34 is PFC enabled 304. For example, physical priority values 0 and 2-7 may correspond to traffic class 0 and may not be PFC enabled, while physical priority value 1 may correspond to traffic class 1 and may be PFC enabled.

[0044] Referring back to Fig. 1, host administrator component 26 may replace the virtual priority 14 of data packet 12 with the corresponding physical priority 32 based on the virtual priority mapping table 44. For example, host administrator component 26 may instruct NIC 21 to replace the virtual priority 14 of data packet 12 with the corresponding physical priority 32 based on the virtual priority mapping table 44. Host administrator component 26 may also use the physical priority mapping table 30 to determine the corresponding traffic class 34 for the physical priority 32. Host administrator component 26 may send data packet 12 with physical priority 32 to network adapter 36 so that data packet 12 may be transmitted on the traffic channel for the corresponding traffic class 34.

[0045] Referring now to Fig. 4, an example system 400 where two virtual machines 106, 107 may communicate with two host computer devices 102, 103 via a virtual network. Virtual machine 107 may be the same as or similar to virtual machine 106 of Fig. 1. In addition, computer device 103 may be the same as or similar to computer device 102 of Fig. 1. Computer devices 102, 103 may execute host administrator component 26 to partition the transmit pipe of the network adapters 36, 33 into multiple ETS channels 412. Network adapters 36, 33 on the same physical network may be partitioned the same way so that the network may have consistent channel partitioning. In addition, applications operating in a native environment on computer devices 102, 103 may tag network data packets with different 802.1p priorities values so that data packets may be classified for transmission over one of the ETS channels 412. For example, applications may tag data packets for SMB storage with a priority 3 value, internet browser data packets with a priority 1 value, video streaming data packets with a priority 6 value, and cluster management data packets with a priority 7 value.

[0046] Computer devices 102, 103 may create ETS policies 402, 404 to map the 802.1p priority values to ETS channels 412 (also known as ETS traffic classes) so that data packets may be transmitted on specific ETS channels 412 based on a priority value of the data packet. For example, ETS policy 402 may map priority values 1 and 2 to a first traffic channel, priority values 0, 4, and 6 to a second traffic channel, and priority values 3, 5, and 7 to a third traffic channel. In addition, ETS policy 404 may map priority values 0-2 to a

first traffic channel, priority value 3 to a second traffic channel, and priority values 4-7 to a third traffic channel.

[0047] Computer devices 102, 103 may also determine a number of virtual ETS channels 412 that virtual machines 106, 107 may use. Operating systems 110 on computer devices 102, 103 may inform the virtual network adapters 410, 408 in virtual machines 106, 107 the number of virtual ETS channels virtual machines 106, 107 may access, so virtual network adapters 410, 408 may inform the virtual network stack.

[0048] For example, virtual machine administrators 20 (Fig. 1) may create virtual ETS policies 404, 405 inside virtual machines 106, 107 to map 802.1p priorities to the virtual ETS channels assigned to virtual machines 106, 107. For example, virtual machine 106 may have two virtual ETS channels assigned to virtual machine 106. As such, virtual ETS policy 404 for virtual machine 106 may map priority values 0 and 2 to traffic channel 1 and may map priority values 1 and 3-7 to traffic channel 2. Virtual machine 107 may have three virtual ETS channels assigned to virtual machine 107.

[0049] In addition, applications 10 (Fig. 1) operating in virtual machines 106, 107 may create policies to tag application traffic with 802.1p priorities. For example, applications 10 in virtual machine 106 may tag SMB storage traffic with a priority value 5. However, virtual machine 107 may not have a policy to tag application traffic with 802.1p priority values. As such, all traffic coming from virtual machine 107 may by default go to traffic channel 1.

[0050] Host administrator component 26 (Fig. 1) may create a virtual to physical priority mapping table 411 for each virtual machine 106, 107 that maps the virtual priority 14 values to physical priority 32 values. For example, host administrator component 26 may determine that virtual machine 106 may be allowed to use physical priority values 1 and 3 and virtual machine 107 may be allowed to use physical priority values 1, 4, and 5. Thus, regardless of how virtual machines 106, 107 tag traffic, the virtual to physical priority mapping table 411 ensures that traffic coming out of virtual machines 106, 107 may only be tagged with the physical priorities assigned to virtual machines 106, 107. For example, traffic coming out of virtual machine 106 may have the virtual priority values changed to only priority values 1 and 3.

[0051] In an implementation, host administrator component 26 may further create application-specific policies for tagging traffic received from applications 10 operating in virtual machines 106, 107. For example, an application-specific policy 414 for virtual machine 107 may indicate that RDMA traffic is tagged with a priority value 5. As data packets are received by host administrator component 26, if RDMA traffic is not already

mapped to a priority 5 value, the priority value of the data packet may be changed to a priority value 5. Moreover, application-specific policy 414 may indicate that any non-RDMA traffic may not use physical priority 5. As such, the priority values of non-RDMA traffic may change to a different priority value if the non-RDMA traffic has a priority value 5. Application-specific policy 414 may further indicate that the other traffic may be dropped if the traffic is not mapped to priority values 1 or 4. Regardless of how the applications on the virtual machine and/or the virtual administrator component configure the priority value mappings, the application-specific policy may automatically configure the traffic classification for the data packets based on the application-specific policy. Therefore, if applications misclassify data traffic and/or are unable to classify data traffic, the application-specific policy may ensure traffic is transmitted on the correct traffic channels.

[0052] Referring now to Fig. 5, an example method 500 may be used by host administrator component 26 (Fig. 1) and/or NIC 21 (Fig. 1) of computer device 102 (Fig. 1) to determine virtual ETS settings 38 (Fig. 1) that a virtual machine (Fig. 1) may use.

[0053] At 502, method 500 may include partitioning a physical network into a plurality of traffic classes. For example, host administrator component 26 may partition the network adapter 36 transmit pipe into multiple ETS channels and assign traffic classes 34 to each ETS channel. Different ETS channels may have different quality of service for traffic transmitted on the ETS channel. For example, one channel may provide lossless traffic, while another channel may periodically drop data packets transmitted on the channel and/or have a delay in transmission. Network adapters 36 operating on the same physical network may be partitioned the same way so the entire network has consistent channel partitioning. Host administrator component 26 may instruct NIC 21 to partition the network adapter 36 transmit pipe into multiple ETS channels and assign traffic classes 34 to each ETS channel.

[0054] At 504, method 500 may include assigning each of the plurality of traffic classes at least one physical priority value from a plurality of physical priority values. Host administrator component 26 may assign physical priorities 32 for data packets to a specific traffic class 34. For example, physical priorities 32 may correspond to 802.1p priorities (e.g., 0 to 7) and may be used for tagging data packets that are transmitted on the physical network. Applications may tag data traffic with 802.1p priority tags, so that the data traffic may be classified into one of the traffic classes. This enables end-to-end quality of service for different data traffic types, based on the settings for the traffic class. As such, the physical priorities 32 may determine an ETS traffic channel the data packet may be transmitted on in the physical network. A traffic class 34 may have one or more physical priorities 32

assigned to the traffic class 34. Thus, data packets with different physical priorities 32 may be transmitted on the same traffic channel. Host administrator component 26 may store the associations between the physical priority 32 and the traffic class 34 in a physical priority mapping table 30.

[0055] At 506, method 500 may include determining a virtual ETS setting for a virtual machine that includes at least one virtual traffic class that corresponds to one of the plurality of traffic classes. Host administrator component 26 may determine virtual ETS settings 38 for each virtual machine 106. The virtual ETS settings 38 may include, but are not limited to, one or more virtual traffic classes 40 assigned to virtual machine 106 and traffic class characteristics 42 (e.g., loss less traffic channel). The virtual traffic classes 40 may correspond to the physical traffic classes 34 that host administrator component 26 allows virtual machine 106 to send traffic on. For example, host administrator component 26 may provide access to two channels (e.g., Ethernet and RDMA) to virtual machines 106. One virtual machine may pay less relative to other virtual machines and may only receive access to the Ethernet channel. Another virtual machine may pay more relative to other virtual machines and may be able to access both channels. As such, the virtual ETS settings 38 may be different for each of the virtual machines 106. Host administrator component 26 may use the virtual ETS settings 38 information to restrict data traffic received from virtual machines 106 by specifying which traffic classes 34 virtual machine 106 may use to transmit traffic. In addition, host administrator component 26 may instruct NIC 21 to restrict data traffic received from virtual machine 106 by specifying which traffic classes 34 virtual machine 106 may use.

[0056] The virtual ETS settings 38 information may be generated by using operating system 110 internal heuristics, based on, for example, a tenant ETS policy the host administrator component 26 has configured for a specific virtual port 18. For example, the host administrator component 26 may allow the virtual machine 106 to send traffic on physical priorities 2, 3 and 4, which maps to physical TC4, TC5, and TC6 on the physical NIC. Furthermore, priority 4 may be PFC enabled on the physical network. Host administrator component 26 may expose the virtual ETS settings 38 to virtual machine 106, where the virtual ETS settings 38 indicate that virtual machine 106 has three TCs, one of which is PFC enabled. For example, priority 0-5 maps to TC0, priority 6 maps to TCI, priority 7 maps to TC2, and is PFC enabled. This enables DCB-aware applications 10 inside the virtual machine 106 to know that application 10 may send traffic out of three traffic channels, and if application 10 wants to send lossless traffic, application 10 may tag traffic

with priority 7.

[0057] At 508, method 500 may include transmitting a notification to the virtual machine identifying the virtual ETS setting. Host administrator component 26 may transmit a message to virtual machine 106 with the virtual ETS settings 38 for virtual machine 106. A virtual administrator component 20 on virtual machine 106 may receive the virtual ETS settings 38 and may create virtual ETS policies 24 that map virtual priorities 14 assigned to data packets 12 to the virtual traffic classes 40 received in the virtual ETS settings 38. Virtual administrator component 20 may also create one or more virtual priority rules 22 that determine how applications 10 may tag data packets 12. For example, the virtual priority rules 22 may indicate that SMB storage traffic may be tagged with a virtual priority value 5 and lossless traffic may be tagged with a virtual priority value 7. As such, applications 10 operating on virtual machine 106 may have the ability to tag data packets 12 with any priority value in a similar manner as applications operating in a native environment.

[0058] Virtual administrator component 20 and applications 10 may be unaware of how the virtual priorities 14 are mapped to the physical network. As such, applications 10 may easily move between different host environments. When applications 10 move between different host environments, applications 10 may continue to use the virtual priority rules 22 to tag data packets 12 and may be unaware that a change in host environment occurred. Moreover, by restricting the information provided to virtual administrator component 20 and applications 10 about the physical network, virtual administrator component 20 and/or applications 10 may be less likely to learn information about the physical datacenter. As such, the physical network topology may be protected from virtual machines that may want to harm the physical network and/or perform malicious actions on the physical network.

[0059] Referring now to Fig. 6, an example method 600 may be used by NIC 21 (Fig. 1) and/or host administrator component 26 (Fig. 1) of computer device 102 (Fig. 1) to restrict data traffic received from a virtual machine 106 (Fig. 1) to a subset of traffic classes 34 (Fig. 1).

[0060] At 602, method 600 may include receiving a data packet with a virtual priority from at least one virtual port. NIC 21 and/or host administrator component 26 may receive a data packet 12 from virtual machine 106. For example, data packet 12 may be transmitted via virtual NIC 17 in communication with virtual port 18. Each of the data packets 12 may include a virtual priority 14 assigned to the data packet 12 by an application 10 operating on the virtual machine 106. In an implementation, operating system 110 may also have virtual NICs outside of virtual machine 106 connected to virtual port 18. As such,

applications running on top of the virtual NICs outside of virtual machine 106 may be subjected to the same policies as applications 10 running on top of virtual NICs 17 inside a virtual machine 106.

[0061] At 604, method 600 may include determining whether there is an application-specific policy for the virtual machine. Host administrator component 26 may include a verification component 46 operable to verify whether the virtual priority 14 of data packet 12 corresponds to application-specific policies 50 that may be defined for application 10. For example, an application-specific policy 50 may include mapping RDMA traffic received from application 10 to priority value 5. For example, host administrator component 26 may communicate the application-specific policy 50 to NIC 21.

[0062] At 606, method 600 may include determining whether the virtual priority matches the application-specific policy. Verification component 46 may check whether the virtual priority 14 of data packets 12 received from application 10 correspond to the correct priority value for the application-specific policy 50. If the virtual priority 14 does correspond to the correct priority value, verification component 46 may not need to perform additional processing on data packet 12.

[0063] At 608, method 600 may include changing the virtual priority based on the application-specific policy when the virtual priority does not match the application-specific policy. However, if the virtual priority 14 does not correspond to the correct priority value, verification component 46 may need to replace the virtual priority 14 of data packet 12 with a modified virtual priority 48 based on the application-specific policy 50. For example, an application-specific policy 50 may include mapping RDMA traffic to priority value 5. If a RDMA data packet was tagged with a virtual priority value of 2, verification component 46 may replace the virtual priority value 2 with a modified virtual priority value 5. Moreover, if virtual machine 106 sends untagged Ethernet and RDMA traffics, which maps to the same physical priority 32, host administrator component 26 may use verification component 46 to move the RDMA traffic to a different (maybe lossless) traffic class 34, while keeping the Ethernet traffic on the original traffic class 34. In addition, host administrator component 26 may instruct NIC 21 to change the virtual priority 14 of the received data packet 12 to a modified virtual priority 48.

[0064] At 610, method 600 may include converting the virtual priority to a physical priority based on one or more priority rules. For example, host administrator component 26 may use a virtual priority mapping table 44 to convert the virtual priority 14 of the received data packet 12 to a corresponding physical priority 32. The one or more priority rules may define

the mappings between the virtual priority 14 value and the physical priority 32 value. For example, virtual priority mapping table 44 may have the following priority rules: virtual priority values 0 - 5 map to physical priority value 2, virtual priority value 6 maps to physical priority value 3, and virtual priority value 7 maps to physical priority value 4, which is PFC enabled on the physical network.

[0065] Host administrator component 26 may have a virtual priority mapping table 44 for each virtual machine 106. By using the virtual priority mapping table 44, host administrator component 26 may ensure that data packets 12 received from virtual machine 106 are transmitted using the correct physical priorities 32 assigned to the virtual machine 106. Thus, regardless of how virtual machine 106 tagged data packet 12, host administrator component 26 may restrict traffic from virtual machine 106 to the allowed traffic classes 34. If virtual machine 106 attempts to use different traffic channels than the traffic channels assigned to virtual machine 106, the virtual priority mapping table 44 may be used to restrict the data traffic received from virtual machine 106 to the correct traffic channels. Host administrator component 26 may replace the virtual priority 14 of data packet 12 with the corresponding physical priority 32 based on the virtual priority mapping table 44. For example, host administrator component 26 may instruct NIC 21 to convert the virtual priority 14 of data packet 12 with a corresponding physical priority 32 based on the virtual priority mapping table 44.

[0066] At 612, method 600 may include determining a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority. Host administrator component 26 may also use a physical priority mapping table 30 to determine the corresponding traffic class 34 for the physical priority 32. Host administrator component 26 may notify NIC 21 of the corresponding traffic class 34 of physical priority 32 so that data packet 12 may be transmitted on the traffic channel for the corresponding traffic class 34.

[0067] Referring now to Fig. 7, an example method 700 may be used by virtual administrator component 20 (Fig. 1) and/or one or more applications 10 (Fig. 1) in a virtual machine 106 (Fig. 1) for tagging data packets 12 (Fig. 1) with a virtual priority value 14 (Fig. 1) in accordance with an implementation of the present disclosure.

[0068] At 702, method 700 may include receiving virtual ETS settings information from a host computer device. A virtual administrator component 20 on virtual machine 106 may receive the virtual ETS settings 38 from computer device 102. The virtual ETS settings 38 may identify a number of virtual ETS channels that virtual machine 106 may use. For example, host administrator component 26 on computer device 102 may inform virtual

network adapter 16 on virtual machine 106 the number of virtual ETS channels virtual machine 106 may access, so virtual network adapter 16 may inform the virtual network stack.

[0069] At 704, method 700 may optionally include creating virtual ETS policies based on the ETS settings information. For example, virtual administrator component 20 may create virtual ETS policies 24 that map virtual priorities 14 assigned to data packets 12 to the virtual traffic classes 40 received in the virtual ETS settings 38. Virtual ETS policies 24 may map 802.1p priorities to the virtual ETS channels assigned to virtual machine 106. For example, virtual machine 106 may have two virtual ETS channels assigned to virtual machine 106. As such, virtual traffic class policy 24 may map priority values 0 and 2 to traffic channel 1 and may map priority values 1 and 3-7 to traffic channel 2.

[0070] At 706, method 700 may include creating priority rules for tagging data packets with a virtual priority value. Virtual administrator component 20 may also create one or more virtual priority rules 22 that determine how applications 10 may tag data packets 12 with a virtual priority value 14. For example, the virtual priority rules 22 may indicate that SMB storage traffic may be tagged with a virtual priority value 5 and lossless traffic may be tagged with a virtual priority value 7.

[0071] At 708, method 700 may include tagging one or more data packets with a virtual priority value. Applications 10 may tag data packets 12 with one or more virtual priority values 14 based on the priority rules 22. As such, applications 10 operating on virtual machine 106 may have the ability to tag data packets 12 with any priority value in a similar manner as applications operating in a native environment.

[0072] At 710, method 700 may include transmitting one or more data packets 12 to a host computer device. Virtual network adapter 16 may transmit one or more data packets 12 to host administrator component 26 via virtual port 18.

[0073] Referring now to Fig. 8, an example computer device 102 in accordance with an implementation may include additional component details as compared to Fig. 1. In one example, computer device 102 may include processor 35 for carrying out processing functions associated with one or more of components and functions described herein. Processor 35 can include a single or multiple set of processors or multi-core processors. Moreover, processor 35 can be implemented as an integrated processing system and/or a distributed processing system.

[0074] Computer device 102 may further include memory 37, such as for storing local versions of applications being executed by processor 35. Memory 37 can include a type of

memory usable by a computer, such as random access memory (RAM), read only memory (ROM), tapes, magnetic discs, optical discs, volatile memory, non-volatile memory, and any combination thereof. Additionally, processor 35 and memory 37 may include and execute host administrator component 26 (Fig. 1), verification component (Fig. 1), and/or network adapter 36 (Fig. 1).

[0075] Further, computer device 102 may include a communications component 41 that provides for establishing and maintaining communications with one or more parties utilizing hardware, software, and services as described herein. Communications component 41 may carry communications between components on computer device 102, as well as between computer device 102 and external devices, such as devices located across a communications network and/or devices serially or locally connected to computer device 102. For example, communications component 41 may include one or more buses, and may further include transmit chain components and receive chain components associated with a transmitter and receiver, respectively, operable for interfacing with external devices.

[0076] Additionally, computer device 102 may include a data store 43, which can be any suitable combination of hardware and/or software, that provides for mass storage of information, databases, and programs employed in connection with implementations described herein. For example, data store 43 may be a data repository for host administrator component 26, verification component, network adapter 36, physical priority mapping table 30 (Fig. 1), ETS Settings 28 (Fig. 1), Virtual ETS settings 38 (Fig. 1), and/or virtual priority mapping table 44 (Fig. 1).

[0077] Computer device 102 may also include a user interface component 45 operable to receive inputs from a user of computer device 102 and further operable to generate outputs for presentation to the user. User interface component 45 may include one or more input devices, including but not limited to a keyboard, a number pad, a mouse, a touch-sensitive display, a navigation key, a function key, a microphone, a voice recognition component, any other mechanism capable of receiving an input from a user, or any combination thereof. Further, user interface component 45 may include one or more output devices, including but not limited to a display, a speaker, a haptic feedback mechanism, a printer, any other mechanism capable of presenting an output to a user, or any combination thereof.

[0078] In an implementation, user interface component 45 may transmit and/or receive messages corresponding to the operation of host administrator component 26, verification component, and/or network adapter 36. In addition, processor 35 executes host administrator component 26, verification component, and/or network adapter 36 or data

store 43 may store them.

[0079] Thus, in some implementations, the described device and methods enable virtualization of DCB information to each vNIC so that applications on top of each vNIC can auto-discover DCB policy configured for that vNIC, which allows the applications to work in virtualization environment as they would in native environment.

[0080] Alternatively or in addition, in some implementations, the described device and methods enable virtualization of DCB information to each vNIC, so that DCB settings on each host vNIC is abstracted from the underlying physical NIC DCB settings.

[0081] Alternatively or in addition, in some implementations, the described device and methods enable virtualization auto-correct classification of application-specific traffic coming out of each vNIC, so that application owners do not need to be DCB-aware in order for the application traffic to be sent on the correct DCB traffic class.

[0082] As used in this application, the terms "component," "system" and the like are intended to include a computer-related entity, such as but not limited to hardware, firmware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a computer device and the computer device can be a component. One or more components can reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers. In addition, these components can execute from various computer readable media having various data structures stored thereon. The components may communicate by way of local and/or remote processes such as in accordance with a signal having one or more data packets, such as data from one component interacting with another component in a local system, distributed system, and/or across a network such as the Internet with other systems by way of the signal.

[0083] Moreover, the term "or" is intended to mean an inclusive "or" rather than an exclusive "or." That is, unless specified otherwise, or clear from the context, the phrase "X employs A or B" is intended to mean any of the natural inclusive permutations. That is, the phrase "X employs A or B" is satisfied by any of the following instances: X employs A; X employs B; or X employs both A and B. In addition, the articles "a" and "an" as used in this application and the appended claims should generally be construed to mean "one or more" unless specified otherwise or clear from the context to be directed to a singular form.

[0084] Various implementations or features may have been presented in terms of systems

that may include a number of devices, components, modules, and the like. It is to be understood and appreciated that the various systems may include additional devices, components, modules, etc. and/or may not include all of the devices, components, modules etc. discussed in connection with the figures. A combination of these approaches may also be used.

[0085] The various illustrative logics, logical blocks, and actions of methods described in connection with the embodiments disclosed herein may be implemented or performed with a specially-programmed one of a general purpose processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but, in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computer devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. Additionally, at least one processor may comprise one or more components operable to perform one or more of the steps and/or actions described above.

[0086] Further, the steps and/or actions of a method or algorithm described in connection with the implementations disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, a hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium may be coupled to the processor, such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. Further, in some implementations, the processor and the storage medium may reside in an ASIC. Additionally, the ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal. Additionally, in some implementations, the steps and/or actions of a method or algorithm may reside as one or any combination or set of codes and/or instructions on a machine readable medium and/or computer readable medium, which may be incorporated into a computer program product.

[0087] In one or more implementations, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the

functions may be stored or transmitted as one or more instructions or code on a computer-readable medium. Computer-readable media includes both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage medium may be any available media that can be accessed by a computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc where disks usually reproduce data magnetically, while discs usually reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0088] While implementations of the present disclosure have been described in connection with examples thereof, it will be understood by those skilled in the art that variations and modifications of the implementations described above may be made without departing from the scope hereof. Other implementations will be apparent to those skilled in the art from a consideration of the specification or from a practice in accordance with examples disclosed herein.

CLAIMS

1. A computer device, comprising:

a memory to store data and instructions;

a processor in communication with the memory;

an operating system in communication with the processor and the memory, wherein the operating system is operable to instruct a network interface controller of the computer device to:

receive a data packet with a virtual priority from at least one virtual port;

convert the virtual priority to a physical priority based on one or more priority rules; and

determine a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

2. The computer device of claim 1, wherein the network interface controller is further operable to:

determine whether an application-specific policy applies for the virtual machine; and

determine whether the virtual priority matches the application-specific policy when the application policy applies to the virtual machine.

3. The computer device of claim 2, wherein the network interface controller is further operable to change the virtual priority to a modified virtual priority when the virtual priority does not match the application-specific policy and maintain the virtual priority when the virtual priority does match the application-specific policy.

4. The computer device of claim 1, wherein the network interface controller is further operable to assign at least one physical priority value from a plurality of physical priority values to each of the plurality of traffic channels.

5. The computer device of claim 1, wherein the network interface controller is further operable to:

determine a virtual enhanced transmission selection (ETS) setting for the virtual machine, wherein the virtual ETS setting comprises at least one virtual traffic class that corresponds to one of the plurality of traffic classes; and

transmit a notification to the virtual machine identifying the virtual ETS setting.

6. The computer device of claim 1, wherein the one or more priority rules identify a mapping between a virtual priority value and a physical priority value.

7. The computer device of claim 1, wherein the virtual priority value is added

to the data packet from one or more of an application operating on the virtual machine and a virtual administrator component operating on the virtual machine.

8. A method for restricting data traffic received from a virtual machine to a subset of traffic classes, comprising:

receiving, at a network interface controller on a computer device, a data packet with a virtual priority from at least one virtual port;

converting, at the network interface controller, the virtual priority to a physical priority based on one or more priority rules; and

determining a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

9. The method of claim 8, further comprising:

determining whether an application-specific policy applies for the virtual machine; and

determining whether the virtual priority matches the application-specific policy when the application policy applies to the virtual machine.

10. The method of claim 9, further comprising:

changing the virtual priority to a modified virtual priority when the virtual priority does not match the application-specific policy; and

maintaining the virtual priority when the virtual priority does match the application-specific policy.

11. The method of claim 8, further comprising:

assigning at least one physical priority value from a plurality of physical priority values to each of the plurality of traffic channels.

12. The method of claim 8, further comprising:

determining a virtual enhanced transmission selection (ETS) setting for the virtual machine, wherein the virtual ETS setting comprises at least one virtual traffic class that corresponds to one of the plurality of traffic classes; and

transmitting a notification to the virtual machine identifying the virtual ETS setting.

13. The method of claim 8, wherein the one or more priority rules identify a mapping between a virtual priority value and a physical priority value.

14. The method of claim 8, wherein the virtual priority value is added to the data packet from one or more of an application operating on the virtual machine and a virtual administrator component operating on the virtual machine.

15. A computer-readable medium storing instructions executable by a computer device, comprising:

at least one instruction for causing the computer device to receive a data packet with a virtual priority from at least one virtual port;

at least one instruction for causing the computer device to convert the virtual priority to a physical priority based on one or more priority rules; and

at least one instruction for causing the computer device to determine a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority, wherein the physical priority is associated with the traffic class.

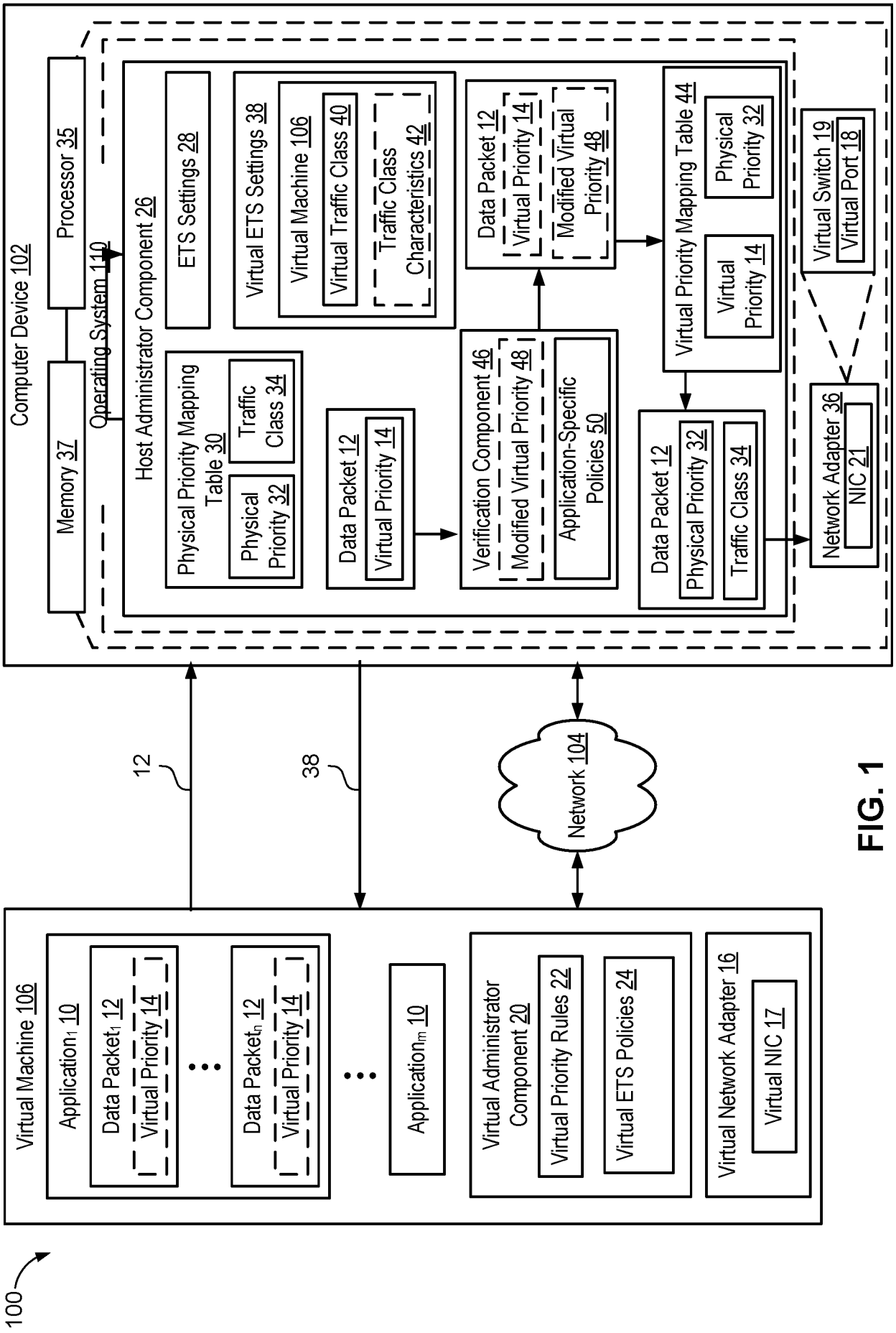
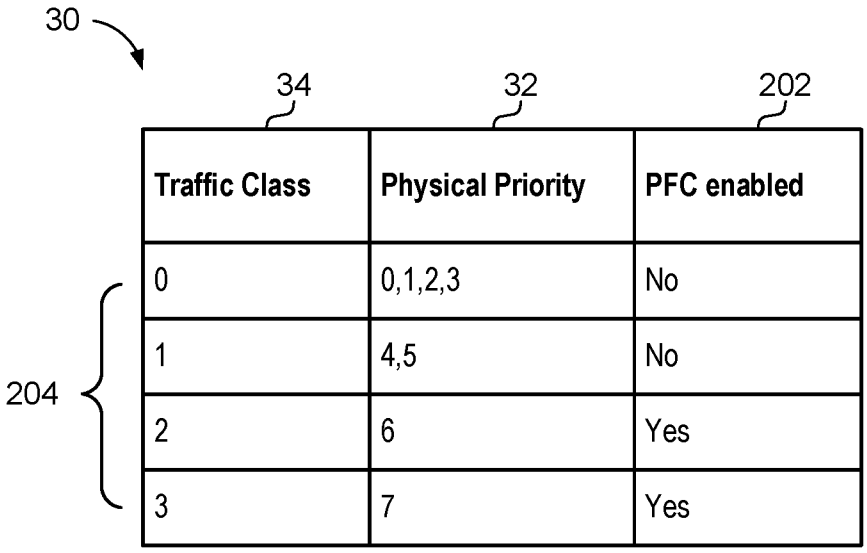

100
Computer Device 102
Processor 35
Memory 37
Operating System 110
Host Administrator Component 26
ETS Settings 28
Physical Priority Mapping Table 30
Traffic Class 34
Physical Priority 32
Virtual ETS Settings 38
Virtual Machine 106
Virtual Traffic Class 40
Traffic Class Characteristics 42
Data Packet 12
Virtual Priority 14
Verification Component 46
Modified Virtual Priority 48
Application-Specific Policies 50
Data Packet 12
Virtual Priority 14
Modified Virtual Priority 48
Virtual Priority Mapping Table 44
Physical Priority 32
Virtual Priority 14
Data Packet 12
Physical Priority 32
Traffic Class 34
Virtual Switch 19
Virtual Port 18
Network Adapter 36
NIC 21
12
38
Network 104
Virtual Machine 106
Application1 10
Data Packet1 12
Virtual Priority 14
Data Packetn 12
Virtual Priority 14
Applicationm 10
Virtual Administrator Component 20
Virtual Priority Rules 22
Virtual ETS Policies 24
Virtual Network Adapter 16
Virtual NIC 17

FIG. 1



Traffic Class	Physical Priority	PFC enabled
0	0,1,2,3	No
1	4,5	No
2	6	Yes
3	7	Yes

FIG. 2

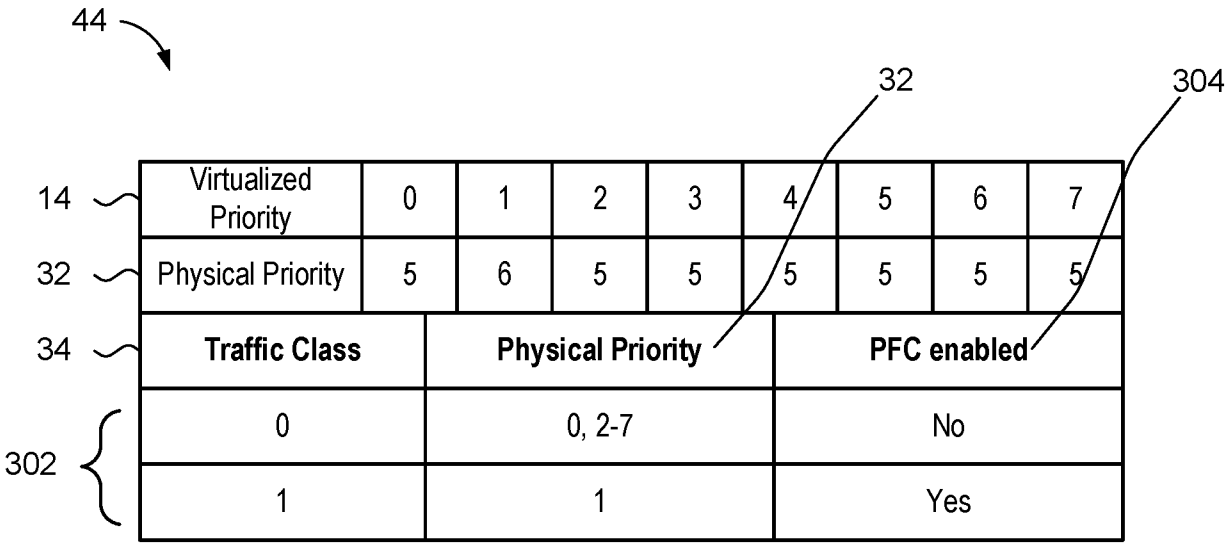

44
32
304
14
32
34
302
Virtualized Priority
0
1
2
3
4
5
6
7
Physical Priority
5
6
5
5
5
5
5
5
Traffic Class
Physical Priority
PFC enabled
0
0, 2-7
No
1
1
Yes

FIG. 3

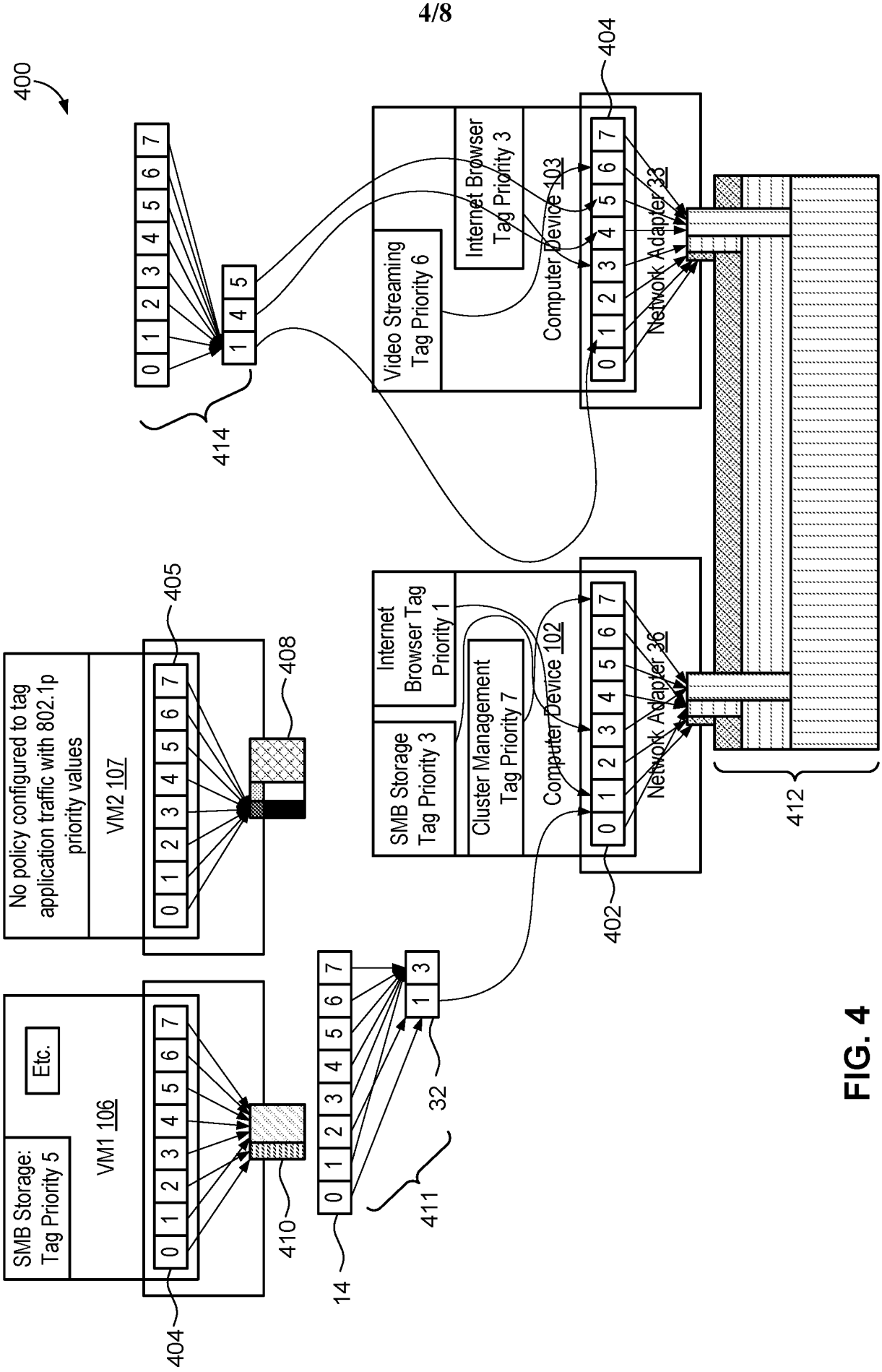

400
404
414
Video Streaming Tag Priority 6
Internet Browser Tag Priority 3
Computer Device 103
Network Adapter 33
405
408
No policy configured to tag application traffic with 802.1p priority values
VM2 107
Internet Browser Tag Priority 1
SMB Storage Tag Priority 3
Cluster Management Tag Priority 7
Computer Device 102
Network Adapter 36
412
402
SMB Storage: Tag Priority 5
Etc.
VM1 106
410
14
411
32
404

FIG. 4

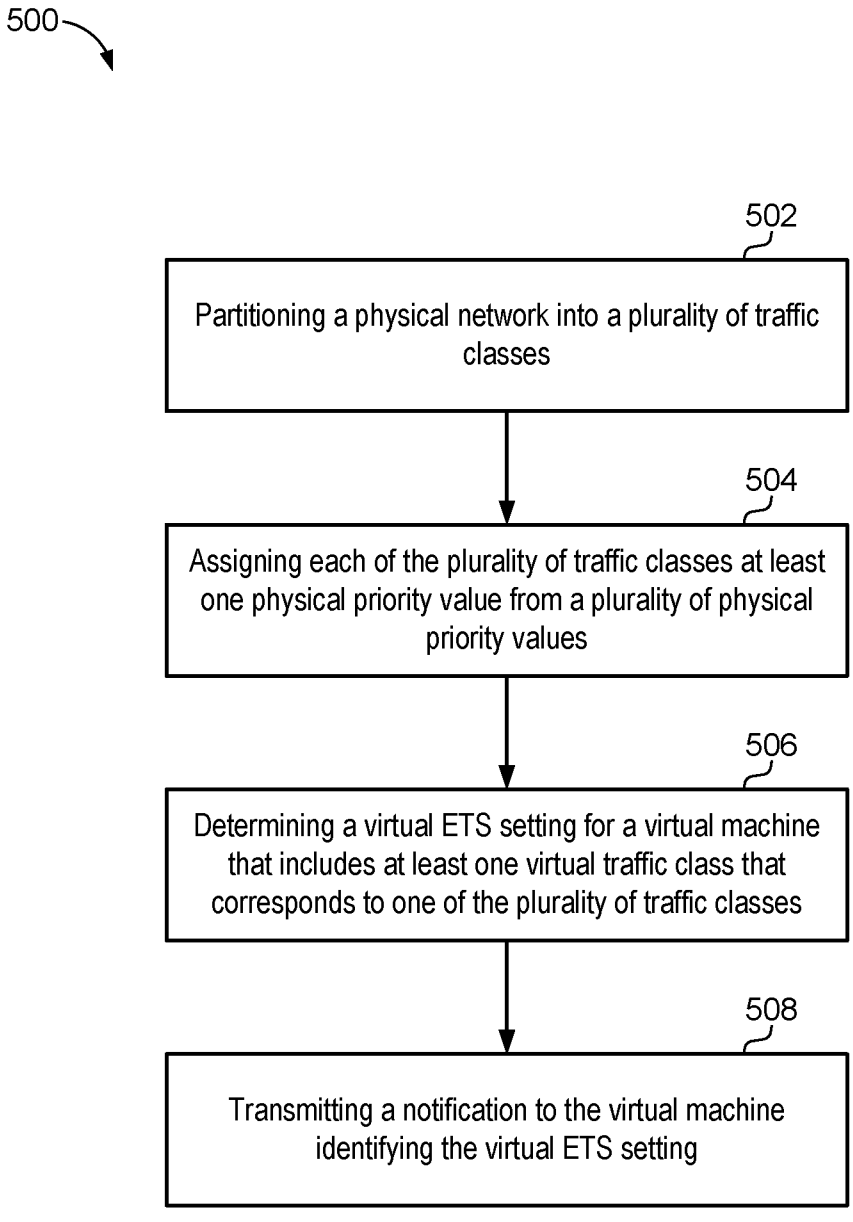

500
502
Partitioning a physical network into a plurality of traffic classes
504
Assigning each of the plurality of traffic classes at least one physical priority value from a plurality of physical priority values
506
Determining a virtual ETS setting for a virtual machine that includes at least one virtual traffic class that corresponds to one of the plurality of traffic classes
508
Transmitting a notification to the virtual machine identifying the virtual ETS setting

FIG. 5

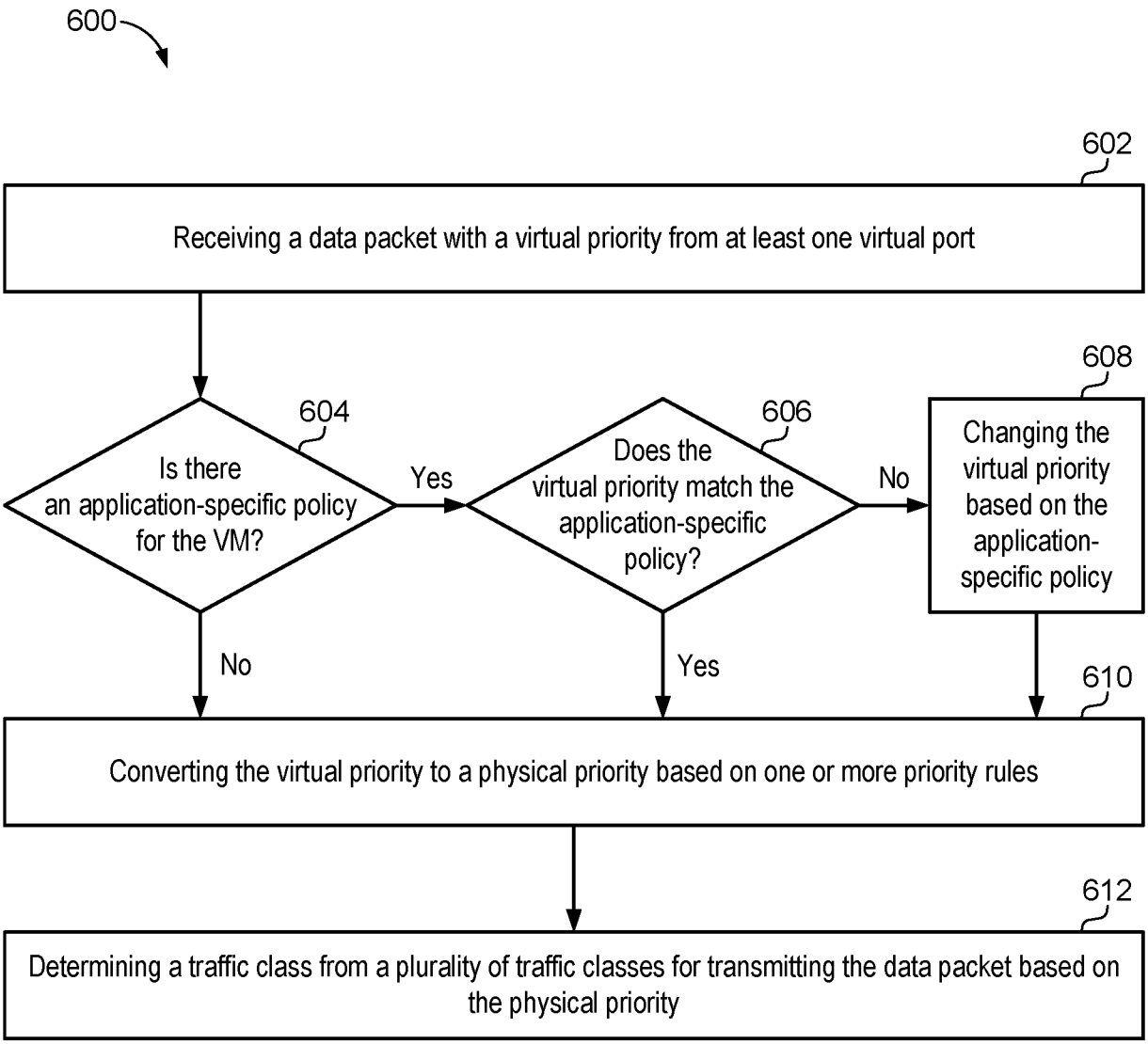

600
602
Receiving a data packet with a virtual priority from at least one virtual port
604
Is there an application-specific policy for the VM?
Yes
606
Does the virtual priority match the application-specific policy?
No
608
Changing the virtual priority based on the application-specific policy
No
Yes
610
Converting the virtual priority to a physical priority based on one or more priority rules
612
Determining a traffic class from a plurality of traffic classes for transmitting the data packet based on the physical priority

FIG. 6

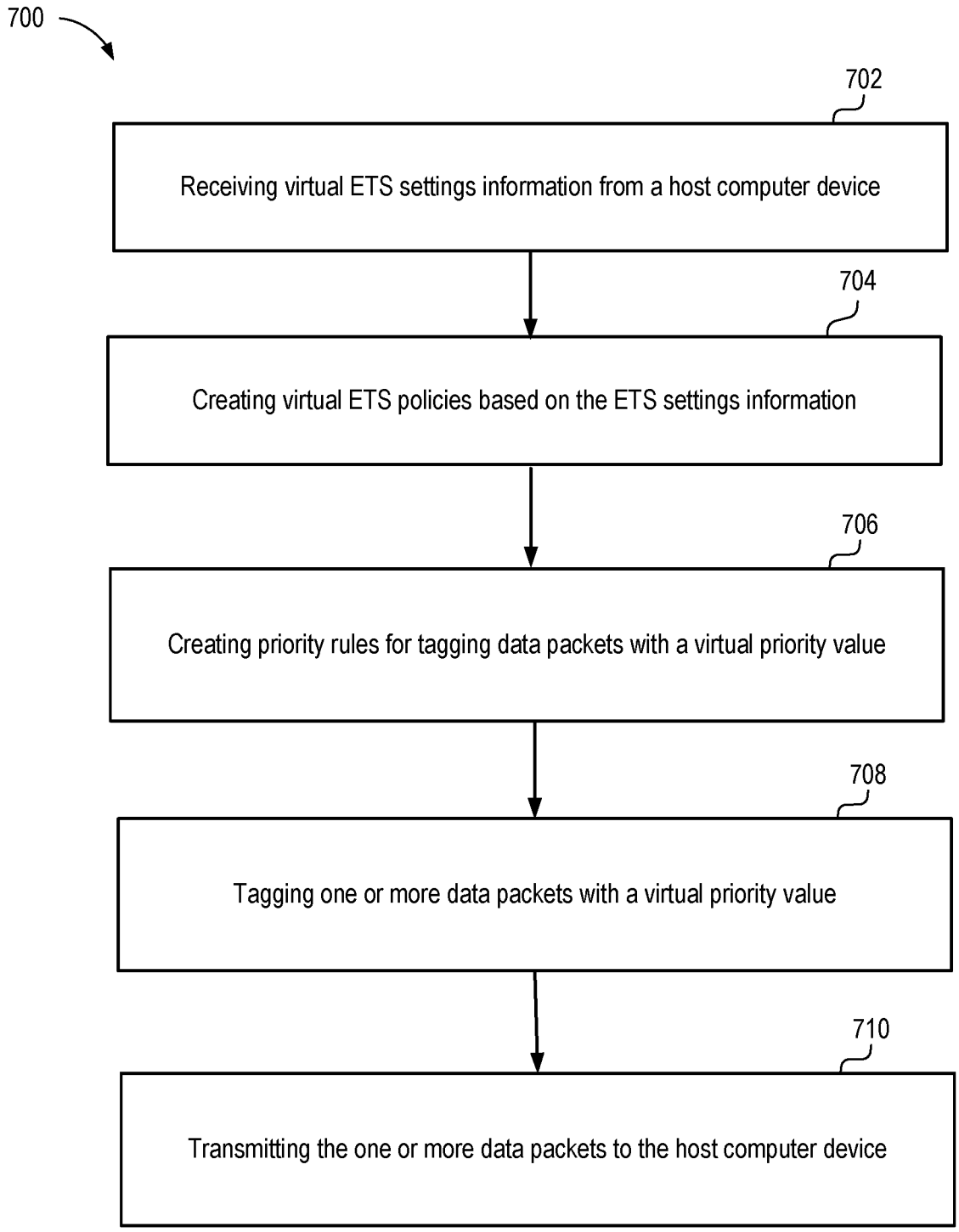

700
702
Receiving virtual ETS settings information from a host computer device
704
Creating virtual ETS policies based on the ETS settings information
706
Creating priority rules for tagging data packets with a virtual priority value
708
Tagging one or more data packets with a virtual priority value
710
Transmitting the one or more data packets to the host computer device

FIG. 7

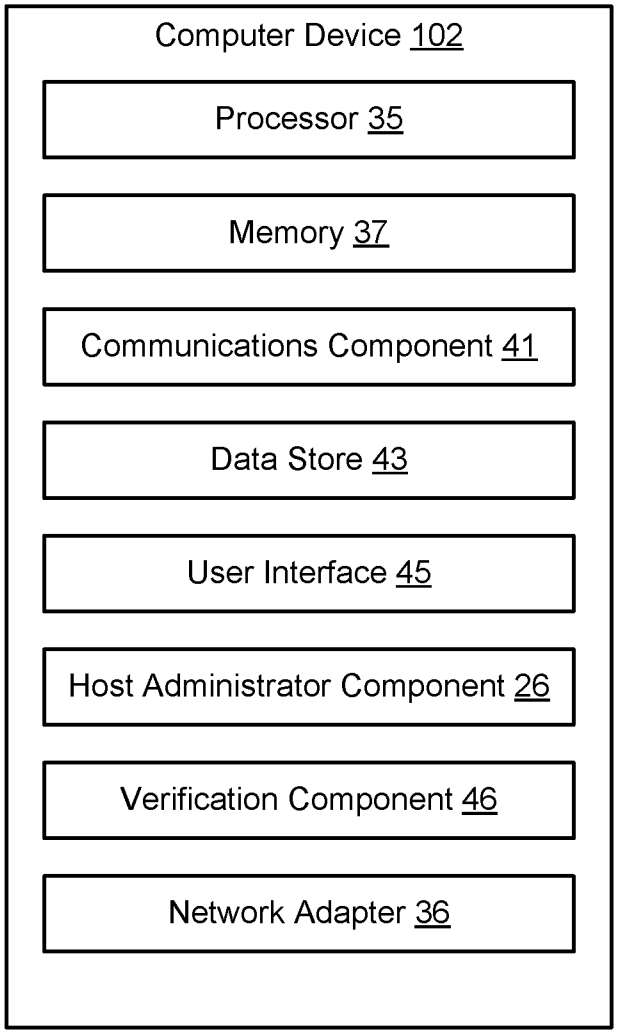

Computer Device 102
Processor 35
Memory 37
Communications Component 41
Data Store 43
User Interface 45
Host Administrator Component 26
Verification Component 46
Network Adapter 36

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/039204

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/911 H04L12/927 H04L12/869 H04L12/813 H04L12/24 H04L12/46
ADD.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 8 027 354 B1 (PORTOLANI MAURIZIO [CH] ET AL) 27 September 2011 (2011-09-27) figures 1,2,5 column 1, line 60 - column 2, line 26 column 3, line 18 - line 64 column 4, line 43 - line 67 column 5, line 18 - line 37 column 6, line 31 - column 8, line 8 column 9, line 6 - line 27 ----- -/-	1,2,4-9 , 11-15 3, 10

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

[X] See patent family annex.

* Special categories of cited documents :

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

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

"L" documentwhich may throw doubts on priority claim(s) orwhich 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
10 September 2018	17/09/2018

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

2

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 2013/166753 Al (ARMSTRONG WI LLIAM J [US] ET AL) 27 June 2013 (2013-06-27) f i gures 1,6,7A,8 paragraph [0019] paragraph [0021] paragraph [0033] - paragraph [0035] paragraph [0054] paragraph [0056] - paragraph [0057] paragraph [0060] paragraph [0062] - paragraph [0067] paragraph [0070] - paragraph [0077] -----	1, 2 ,4,8, 9 , 11 , 15 3 ,5-7 , 10, 12-14
X A	US 2011/035498 Al (SHAH HEMAL [US] ET AL) 10 February 2011 (2011-02-10) f i gure 3 paragraph [0030] paragraph [0033] - paragraph [0035] paragraph [0041] -----	1-3 , 8-10, 15 4-7 , 11-14
A	Renato Reci o ET AL: "Automated Ethernet Vi rtual Bri dgi ng" , 21st Internati onal Tel etraffi c Congress (ITC 21) , 15 September 2009 (2009-09-15) , XP055505367 , Retri eved from the Internet: URL: http://www. i tc21 .net/fi l eadmi n/ITC21 _f i 1es/DC-CAVES/DC-CAVES_-_AutomatedEthernet Vi rtual Bri dgi ng.pdf [retri eved on 2018-09-07] the whol e document -----	1-15
A	US 2017/155599 Al (V0BBI LISETTY SURESH [US] ET AL) 1 June 2017 (2017-06-01) the whol e document -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2018/039204

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8027354 B1	27-09-2011	NONE	
US 2013166753 A1	27-06-2013	CN 104012057 A	27-08-2014
		DE 112012004957 T5	14-08-2014
		GB 2515643 A	31-12-2014
		JP 5967633 B2	10-08-2016
		JP 2015502724 A	22-01-2015
		TW 201338445 A	16-09-2013
		US 2013163611 A1	27-06-2013
		US 2013166753 A1	27-06-2013
		US 2018145926 A1	24-05-2018
		WO 2013093734 A1	27-06-2013
US 2011035498 A1	10-02-2011	US 2011035498 A1	10-02-2011
		US 2013297787 A1	07-11-2013
US 2017155599 A1	01-06-2017	US 2011299413 A1	08-12-2011
		US 2017155599 A1	01-06-2017