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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 December 2015 (30.12.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/199743 A1

(51) International Patent Classification:
H04L 12/801 (2013.01)

(21) International Application Number:
PCT/US2014/052534

(22) International Filing Date:
25 August 2014 (25.08.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
3087/CHE/2014 25 June 2014 (25.06.2014) IN

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(72) Inventors: **MATHEW, Subin Cyriac**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN). **CHANDRAN, Sugesh**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN). **KANIAMPADY SEBASTIAN, Celestian**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN).

(74) Agents: **FERGUSON, Christopher W.** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (US).

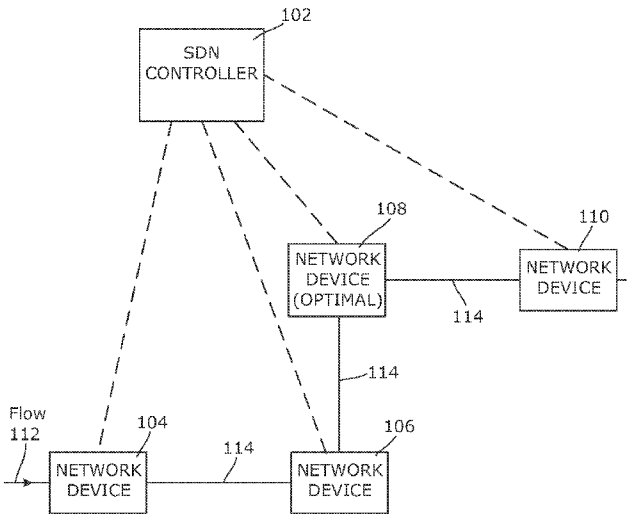
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, 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,

[Continued on next page]

(54) Title: SELECTING AN OPTIMAL NETWORK DEVICE FOR REPORTING FLOW TABLE MISSES UPON EXPIRY OF A FLOW IN A SOFTWARE DEFINED NETWORK

(57) Abstract: Examples disclosed herein relate to selecting an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network, comprising. An SDN enabled device is selected, from a plurality of software defined network (SDN) enabled devices, for reporting a flow table miss upon expiry of a flow to an SDN controller, based on a predefined factor.



100

Fig. 1

WO 2015/199743 A1

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

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

Published:

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

SELECTING AN OPTIMAL NETWORK DEVICE FOR REPORTING FLOW TABLE MISSES UPON EXPIRY OF A FLOW IN A SOFTWARE DEFINED NETWORK

Background

[001] Recent trends in networking include a new approach to routing architecture in which data and control planes have been decoupled. This new split-architecture framework that focuses on splitting of control plane from forwarding and data plane is the basis of software defined networking (SDN). In a software defined network (SDN), the control plane is implemented in an SDN controller and the data plane is implemented in the networking infrastructure (e.g., switches and routers). Data forwarding on a network device is controlled through flow table entries populated by the SDN controller that manages the control plane for that network. A network device that receives packets on its interfaces looks up its flow table to check the actions that need to be taken on a received packet.

Brief Description of the Drawings

[002] For a better understanding of the solution, embodiments will now be described, purely by way of example, with reference to the accompanying drawings, in which:

[003] FIG. 1 is a diagram of an example network system to select an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network;

[004] FIG. 2 is a flow chart of an example method for selecting an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network; and

[005] FIG. 3 is a flow chart of an example method for selecting an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network; and

[006] FIG. 4 is a diagram of an example system to select an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network.

Detailed Description

[007] Software defined networking (SDN) is an approach to networking in which control is decoupled from networking equipment and given to a device called a controller (or SDN controller). The controller may be aware of all the devices and their points of interconnection in a SDN network and performs various functions such as routing, policy implementation, etc. Each new or missed flow through the network is routed via the controller that decides the network path for a flow, and adds an entry for that flow in a flow table in each of the network devices along the path. A SDN enabled device consults a flow table(s) for forwarding packets in the data plane. Each forwarding rule (flow entry) includes an action that dictates how traffic that matches the rule is to be handled. A SDN controller may communicate with network devices via a standardized protocol (such as OpenFlow) or a proprietary API.

[008] Apart from installing new flow rules, an SDN controller may be responsible for removing the flow rules as well. Towards this task, the controller may assign a timeout to each flow entry. When a timeout expires on an SDN enabled device (e.g., switches), the device may evict the rule from the flow table and, optionally, notify the controller regarding the removal. Often, when a new flow arrives at the controller for flow programming, the controller may not know the exact period of time during which the flow may remain active. In an instance, the controller may determine an expected duration of a flow session based on the flow

signature, and heuristics from prior traffic patterns. Based on these, the controller may program a flow entry with a hard timeout. When a hard timeout expires the flow entry is evicted by the switch irrespective of whether or not packets are matching the flow entry. In other words, the flow entry is removed from the network device regardless of the presence of a matching flow.

[009] It is useful to have a flow replacement mechanism to prevent a switch flow table from overflowing as the number of flows increases. However, an issue with having a flow replacement mechanism is that a switch may remove the flow entry when the hard timeout expires even if a session is still ongoing. In many cases, a session may last longer than what was expected by the controller due to network retransmission necessitated by network congestion, delays due to host application bottlenecks, etc. Needless to say such situation is not desirable since any succeeding packets of the same flow may hit a “flow table miss” (i.e., there is no flow table entry for the flow) and may need to be forwarded to the controller to make a forwarding decision. This may cause a number of issues. For example, it may unnecessarily burden the controller which may not only have to receive a flow table miss packet, decide its future network path, but may also need to program a new flow in all the network devices along a future data path so that subsequent packets of the flow may take the new path instead of coming to the controller.

[0010] Further, in a controller with multiple Network Interface Cards (NICs) servicing different portions of a software defined network, high load on some portions may cause some NICs to be overloaded while others may remain underutilized. In other words, the kernel network stack and multi-threaded controller application threads may not be efficiently utilized. Also, it has been observed that new flow misses are almost entirely clustered around access switches whereas flow misses on the upstream devices are due to flow timeouts. But currently a flow miss from upstream devices are processed in

the same way as new flow miss even though the flow may already have been classified in a first-pass. Furthermore, some switches simply may not have the resources to process flow miss packets for reporting to a controller. When a large number of packets (especially in case of a high speed data transfer session, multimedia data streaming, etc.) come to the resource constrained switch control plane due to a flow miss, the switch may not be able to process all of them, resulting in packet drops of multiple flows. This may cause packet re-ordering, retransmission and latency issues. In addition, high flow miss rate may introduce thrashing of a flow table wherein the switch ends up adding and removing the same flow entries continuously. Further, since switches in a flow path are not at equal distance from the controller, it may cause increased latency to report flow miss and the corresponding flow programming.

[0011] To address these issues, the present disclosure describes various examples for selecting an optimal network device for reporting flow table misses on expiry of flow in a software defined network. In an example, an SDN enabled device is selected from a plurality of software defined network (SDN) enabled devices for reporting a flow table miss upon expiry of a flow to an SDN controller based on a pre-defined factor. Upon selection, the selected SDN device may act as a "reporting switch" for reporting a subsequent flow table miss of the flow to the SDN controller.

[0012] As used herein, the term "optimal" is not intended to mean that the selection is objectively the best or optimal, but rather that the selection was the result of the techniques described herein. Such a selection may be deemed subjectively optimal due to any number of criteria being met.

[0013] FIG. 1 is a diagram of an example network system 100 to select an optimal network device for reporting flow table misses on expiry of a flow in a software defined network. Network system 100 may include an SDN controller 102, and a plurality of network devices 104, 106, 108, and 110.

Although only one SDN controller 102 and four network devices are shown in FIG. 1, other examples of this disclosure may include more than one SDN controller and more or less number of network devices. In an example, network system 100 may include one or more computer systems or an end user device(s) (not shown) that may be the source or destination of packet flows in network system 100. In an example, network system 100 may be based on software-defined networking (SDN) architecture.

[0014] SDN controller 102 may be any server, computing device, or the like. In an example, SDN controller 102 may be a computer application (machine-executable instructions). SDN controller 102 may define the data flow that occurs in network system 100. In other words, SDN controller 102 may determine how packets should flow through the network devices 104, 106, 108, and 110 of network system 100. SDN controller 102 may communicate with network devices 104, 106, 108, and 110 via a standardized protocol (example, OpenFlow) or a suitable API.

[0015] SDN controller 102 may maintain all network rules and provide appropriate instructions (such as forwarding instructions) to network devices 104, 106, 108, and 110. SDN controller 102 may centralize the network intelligence, while network maintains a distributed forwarding plane through network devices 104, 106, 108, and 110. In other words, SDN controller 102 may become aware of the network topology prior to computing forwarding paths in network system 100. SDN controller 102 then program rules on each network device (for example, 104, 106, 108, and 110) which may be used by a network device to forward packets to another device in the network 100. In an example, aforesaid programming rules may take the form of a flow entry in one or more flow tables in a network device (for example, 104, 106, 108, and 110). Each flow entry may be associated with an action (example, forward, redirect, drop, etc.) that describes what process may be followed with respect to a flow that corresponds to a flow entry. SDN controller 102 thus controls the way data packets are routed in network 100.

[0016] SDN controller 102 may communicate with network devices 104, 106, 108, and 110 over a computer network 112. Computer network 112 may be a wireless or wired network. Computer network 106 may include, for example, a Local Area Network (LAN), a Wireless Local Area Network (WAN), a Metropolitan Area Network (MAN), a Storage Area Network (SAN), a Campus Area Network (CAN), or the like. Further, computer network 112 may be a public network (for example, the Internet) or a private network (for example, an intranet).

[0017] Network devices 104, 106, 108, and 110 may include, by way of non-limiting examples, a network switch, a network router, a virtual switch, and a virtual router. In an example, at least one of the network devices 104, 106, 108, and 110 may be an SDN enabled device or an Open-Flow enabled device. In an example, network devices 104, 106, 108, and 110 are part of a network path assigned to a flow in network system 100.

[0018] Network devices 104, 106, 108, and 110 may include one or more flow tables (not shown). Each flow table in network devices 104, 106, 108, and 110 may contain a flow entry (or flow entries) 106. SDN controller 102 may add, update, and delete flow entries 106 in flow tables both reactively (in response to packets) and proactively. Network devices 104, 106, 108, and 110 may communicate with SDN controller 102 and the controller 102 may manage the device via a standardized protocol such as OpenFlow. For instance, one of the network devices 104, 106, 108, and 110 may forward the first packet of a flow to SDN controller 102, enabling the controller 102 to decide whether the flow should be added to a flow table of a network device (for example, 104, 106, 108, and 110). Network devices 104, 106, 108, and 110 thus may accept directions from an SDN controller 102 to change values in a flow table.

[0019] A flow table matches an incoming packet to a particular flow and specifies the function that may be performed on the packet. If a flow entry

106 matching with a flow is found in a flow table, instructions associated with the specific flow entry may be executed. A packet matches a flow table entry 106 if the values in the packet match fields used for the lookup match those defined in the flow table entry 106. If no match is found in a flow table (such cases may be termed as "flow table misses"), the outcome may depend on configuration of the table-miss flow entry. For example, the packet may be forwarded to SDN controller 102. This has an advantage that the SDN controller may re-forward the packet after installing the flow entry in the network devices of a future path of the flow. In another implementation, a network device may drop the packet and choose to only send metadata about the packet to the SDN controller. The metadata may include sufficient information about the packet which may enable the SDN controller 102 to associate it with a flow entry which the controller may already have introduced in a network device.

[0020] Each flow entry 106 may consist of match fields, counters, and a set of instructions to apply to matching packets. In an example, each flow table entry 106 may include a number of components, for example, match fields, priority, counters, instructions, timeouts, and cookie.

[0021] A timeout component may define the maximum amount of time before a flow may be expired by a network device. In an example, the timeout period may represent a hard timeout period. The hard timeout period represents maximum amount of time before the flow entry is removed from the network device irrespective of whether or not packets are arriving at the flow entry in the network device. In another example, the time period may represent an inactivity timeout period. If no packets of a flow are observed before the expiry of the inactivity time period, the flow entry is evicted from the network device.

[0022] In an example, an SDN controller (for example, 102) may select an optimal network device from a plurality of network devices (for example, 104,

106, 108, and 110) present in a network system (for example, 100) for reporting a flow table miss for a flow to the SDN controller (for example, 102) based on a pre-defined factor. Upon selection, the selected SDN device may act as a "reporting device" for reporting a subsequent flow table miss for the flow to the SDN controller (for example, 102). In other words, the reporting device may be used to report any subsequent flow miss that may occur due to a timeout for a flow after the initial flow setup.

[0023] An SDN controller (for example, 102) may use various factors to identify an optimal network device in the data path of a flow to act as the reporting device for that flow. These factors may depend on the SDN controller (or controllers) (for example, 102) present in the network system 100, network devices (for example, 104, 106, 108, and 110) present in the network system, and the network system 100.

[0024] Some non-limiting examples of pre-defined factors that may depend on an SDN controller (for example, 102) may include a packet interrupt servicing load at the SDN controller (for instance, in case of a controller with multiple network interfaces (such as NICs), the load may vary across various network interfaces), and an application load at the SDN controller (for instance, in case of multiple controllers in a network system, application load may vary across various controllers or in a multithreaded controller application, the load may vary across the threads).

[0025] A non-limiting example of a pre-defined factor associated with an SDN enabled device (for example, 104, 106, 108, and 110) may include the processing capability of the SDN enabled device to process a flow miss packet for reporting to the SDN controller (for example, 102). In other words, the difference in capability of processing of flow miss packets to be reported to an SDN controller at each of the network devices present in the data path of a flow may be considered a factor for selecting a reporting switch.

[0026] Some non-limiting examples of pre-defined factors that may depend on the network system may include the distance between an SDN controller (for example, 102) and each of the plurality of network devices (for example, 104, 106, 108, and 110). This factor may also include parameters such as management network bandwidth, and network latency between an SDN controller (for example, 102) and each network device in a network system. The utilization rate of a control channel between an SDN controller (for example, 102) and each network device in a network system (for example, 100) may also be considered a factor in selecting a network switch. Another factor may include a network infrastructure induced limitation. For instance, a network system may partially be converted into an SDN based network by using OpenFlow-hybrid switches that may support both OpenFlow as well as legacy protocols. In such case, a fork-lift upgrade of the network system may not be possible, and the existing infrastructure may need to be optimally utilized.

[0027] In an example, a flow policy or a priority assigned to a flow may act as a factor for selecting an optimal network device for reporting flow table misses in a network. In such case, an SDN controller (for example, 102) may assign different reporting switches to each flow while classifying the flow signature based on a pre-configured policy or priority.

[0028] An SDN controller (for example, 102) may select an SDN enabled device (for example, 104, 106, 108, or 110), for reporting a flow table miss for an existing flow to the SDN controller (for example, 102), based on a set or matrix of parameters or factors. In an instance, each factor in a given set or matrix may be assigned a weightage, and for each network device present in the data path under consideration a “cost” of reporting a flow miss may be calculated. The network device that offers the lowest cost for reporting a flow miss entry may be selected as the optimal device (“reporting device”) for reporting a flow table miss to the SDN controller (for example, 102).

[0029] In an example, the cost of reporting a flow miss for each network device 'n' for a flow 'f' and for identifying a reporting device may be determined as follows:

If the controller identified path for a particular flow is N1 →N2→N3.....Nn,

n ∈ {N1, N2, N3, N4, ...Nn},

cost (n,f) = Cost of reporting flow miss via network device Switch 'n' for the flow 'f'.

nodeList= N1,N2, N3, ... Nn

Reporter_node = N1

for each node 'n' in nodeList

if(cost(n, f) < cost(Reporter_node, f)

Reporter_node = n

end if

end for

[0030] Once a reporting device has been identified, in order to route a further flow miss to the controller (example 102) via this device, the controller may explicitly send a delete message to this device when the timeout expires or hard timeouts may be programmed in all network devices in the path such that the reporting device times out first. The timeouts for other network devices may be increased by an amount Δ, which may take into account the time needed to update the flow entries if the controller decides to extend the flow.

[0031] In an example, timeouts may be assigned to network devices in a path as follows:

$HARD_TIMEOUT_f$: The hard timeout for a particular flow 'f' assigned by a controller.

$INACTIVITY_TIMEOUT_f$: The inactivity timeout for a particular flow 'f' assigned by a controller.

$T = \{t_1, t_2, t_3, t_4, t_5 \ldots\ldots\ldots\ldots, t_n\}$, where t_n is the timeout value of switch '*n*' for the flow 'f'.

$\Delta = \{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5, \ldots\ldots\ldots\ldots \Delta_n\}$,where Δ_n is the increment over the $HARD_TIMEOUT_f$ for switch 'n' for the flow 'f'. Note that upper bound of Δ_n is $INACTIVITY_TIMEOUT_f$

for each node 'n' in nodeList
if(n == Reporter_node))
$t_n = HARD_TIMEOUT_f$
else
$t_n = HARD_TIMEOUT_f + \Delta_n$
end if
end for

[0032] By way of an example, FIG. 1 shows selection of an optimal network device ("reporting device") 108 from a plurality of network devices 104, 106, 108, and 110, for reporting a flow table miss upon expiry of a flow to an SDN controller 102, based on a pre-defined factor. In an example, when a new flow 112 reaches the first network device 104, it may result in a flow table miss, and the packet may be sent to the controller. In case the SDN controller decides to continue the flow, it may compute a data path (104→106→108→110) for the flow, and identify an optimal network device 108 based on a pre-defined factor (or factors), as described above. The SDN controller may program flow table entries in each of the network devices (including the identified optimal device) that are present in the data path 114 in such a manner that the timeout period associated with the

identified device 108 is first to timeout. Since the identified device 108 times out first, any subsequent packets may get redirected to the controller from the reporting device. Upon receipt of a packet from the reporting device, if the SDN controller decides to allow a flow to continue, it may reprogram the flow entry in the reporting device and update the flow entries in all other network device switches in the data path prior to their expiry. In an example, each time an SDN controller may reprogram a data path; it may select a reporting device based on one or more parameters as described above. Thus, the controller may control channeling of an unknown flow reporting via an optimal network device.

[0033] FIG. 2 is a flow chart of an example method 200 for selecting an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network. The method 200, which is described below, may be executed on a computing device such as SDN controller 102 of FIG. 1. However, other suitable computing devices may execute method 200 as well. In particular, method 200 may execute, at least partially, in one or more network devices (for example, 104, 106, 108, and 110). At block 202, an SDN controller (for example, 102) may select, from a plurality of software defined network (SDN) enabled devices (for example, 104, 106, 108, and 110) in a software defined network, an SDN enabled device for reporting a flow table miss upon expiry of a flow to the SDN controller, based on a pre-defined factor (or factors). The pre-defined factor(s) may include one or more factors as described above. The pre-defined factor may depend on the SDN controller, the SDN enabled devices in the network, and/or the software defined network itself. In an example, the plurality of SDN enabled devices are present on a common network path of a flow, and the SDN controller selects an SDN enabled device for reporting a flow table miss for the flow from one of the SDN enabled devices.

[0034] FIG. 3 is a flow chart of an example method 300 for selecting an optimal network device for reporting flow table misses upon expiry of a flow in a

software defined network. The method 300, which is described below, may be executed on a computing device such as SDN controller 102 of FIG. 1. However, other suitable computing devices may execute method 300 as well. In particular, method 300 may execute, at least partially, in one or more network devices (for example, 104, 106, 108, and 110). At block 302, an SDN controller 102 may receive a flow miss packet from an SDN enabled device (for example, 104). A flow miss packet may be generated when a new flow reaches the SDN enabled device (for example, 104). Since there is no flow entry in the SDN enabled device (for example, 104) for a new flow, it results in a flow miss, and a packet from the new flow may be forwarded to the SDN controller 102. At block 304, the SDN controller (for example, 102) may determine a network path for said flow and add flow table entries in each SDN enabled device in the network path. At block 306, SDN controller may select an SDN enabled device in the network path for reporting a flow table miss upon a subsequent expiry of the flow to the SDN controller, based on a pre-defined factor (or factors). Method 300 may then proceed to block 308 or 310. At block 308, the SDN controller may associate a timeout period for a flow entry corresponding to the flow in each of the plurality of SDN enabled devices in the network path of the flow, including the selected SDN enabled device, in such a manner that the timeout period associated with the selected SDN enabled device is first to timeout. At block 310, SDN controller may delete a flow entry corresponding to the flow from the selected SDN enabled device when the initially calculated timeout period expires to cause a packet matching the deleted flow entry in the software defined network to reach the SDN controller only from the selected SDN enabled device.

[0035] FIG. 4 is a diagram of an example system 400 to select an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network. System 400 includes a processor 402 and a machine-readable storage medium 404 communicatively coupled through a system bus. In an example, system 400 may be analogous to SDN controller 102 of FIG. 1. Processor 402 may be any type of Central Processing Unit

(CPU), microprocessor, or processing logic that interprets and executes machine-readable instructions stored in machine-readable storage medium 404. Machine-readable storage medium 404 may be a random access memory (RAM) or another type of dynamic storage device that may store information and machine-readable instructions that may be executed by processor 402. For example, machine-readable storage medium 404 may be Synchronous DRAM (SDRAM), Double Data Rate (DDR), Rambus DRAM (RDRAM), Rambus RAM, etc. or storage memory media such as a floppy disk, a hard disk, a CD-ROM, a DVD, a pen drive, and the like. In an example, machine-readable storage medium may be a non-transitory machine-readable medium. Machine-readable storage medium 404 may store instructions 406. In an example, instructions 406 may be executed by processor 402 to identify, from a plurality of network devices, a network device, for reporting a flow table miss for a flow to an SDN controller, based on a pre-defined factor, wherein the plurality of network devices are on a network path assigned to the flow. In an example, machine-readable storage medium 404 may store further instructions to query only the identified network device for a flow parameter to decide regarding continuation or removal of the flow from the software defined network. The storage medium 404 may also store instructions to associate a hard timeout period for a flow entry corresponding to the flow in each of the plurality of network devices, including the identified network device, in such a manner that the hard timeout period associated with the identified network device is first to timeout. The storage medium 404 may further include instructions to report the flow table miss for the flow from the identified network device to the SDN controller.

[0036] For the purpose of simplicity of explanation, the example methods of FIGS. 2 and 3 are shown as executing serially, however it is to be understood and appreciated that the present and other examples are not limited by the illustrated order. The example systems of FIGS. 1 and 4, and methods of FIGS. 2 and 3 may be implemented in the form of a computer

program product including computer-executable instructions, such as program code, which may be run on any suitable computing device in conjunction with a suitable operating system (for example, Microsoft Windows, Linux, UNIX, and the like). Embodiments within the scope of the present solution may also include program products comprising non-transitory computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer. By way of example, such computer-readable media can comprise RAM, ROM, EPROM, EEPROM, CD-ROM, magnetic disk storage or other storage devices, or any other medium which can be used to carry or store desired program code in the form of computer-executable instructions and which can be accessed by a general purpose or special purpose computer. The computer readable instructions can also be accessed from memory and executed by a processor.

[0037] It may be noted that the above-described examples of the present solution is for the purpose of illustration only. Although the solution has been described in conjunction with a specific embodiment thereof, numerous modifications may be possible without materially departing from the teachings and advantages of the subject matter described herein. Other substitutions, modifications and changes may be made without departing from the spirit of the present solution. All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or steps are mutually exclusive.

Claims:

1. A method for selecting an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network, comprising:

selecting, from a plurality of software defined network (SDN) enabled devices, an SDN enabled device, for reporting a flow table miss upon expiry of a flow to an SDN controller, based on a pre-defined factor.

2. The method of claim 1, further comprising associating a timeout period for a flow entry corresponding to the flow in each of the plurality of software defined network (SDN) enabled devices, including the selected SDN enabled device, in such a manner that the timeout period associated with the selected SDN enabled device is first to timeout.

3. The method of claim 1, further comprising deleting a flow entry corresponding to the flow from the selected SDN enabled device upon expiry of a previously determined timeout period to cause a packet matching the deleted flow entry in the software defined network to reach the SDN controller only from the selected SDN enabled device.

4. The method of claim 1, wherein the pre-defined factor is associated with one of the SDN controller, the plurality of SDN enabled devices, and the software defined network.

5. The method of claim 4, wherein the pre-defined factor associated with the SDN controller is one of a packet interrupt servicing load at the SDN controller, and an application load at the SDN controller in the software defined network.

6. The method of claim 4, wherein the pre-defined factor associated with the SDN enabled device includes a processing capability of the SDN enabled device to process a flow miss packet for reporting to the SDN controller.

7. The method of claim 4, wherein the pre-defined factor associated with the software defined network is one of a distance between the SDN controller and each of the plurality of SDN enabled devices, a utilization rate of a respective control channel between the SDN controller and each of the plurality of SDN enabled devices, and a network infrastructure based factor.

8. A system to select an optimal network switch for reporting flow table misses upon expiry of a flow in a software defined network, comprising:

a plurality of network switches; and

an SDN controller to select, from the plurality of network switches, a network switch, for reporting a flow table miss for an existing flow to the SDN controller, based on a set of parameters.

9. The system of claim 8, wherein each parameter in the set of parameters is associated with a weightage.

10. The system of claim 8, wherein a parameter in the set of parameters is a pre-computed parameter or a run-time parameter.

11. A non-transitory machine-readable storage medium comprising instructions for selecting an optimal network device for reporting flow table misses upon expiry of a flow in a software defined network (SDN), the instructions executable by a processor to:

identify, from a plurality of network devices, a network device, for reporting a flow table miss for a flow to an SDN controller, based on a pre-defined factor, wherein the plurality of network devices are on a network path assigned to the flow.

12. The storage medium of claim 11, wherein the pre-defined factor includes a priority associated with the flow in the software defined network.

13. The storage medium of claim 11, further comprising instructions to query only the identified network device for a flow parameter to decide regarding continuation or removal of the flow from the software defined network.

14. The storage medium of claim 11, further comprising instructions to associate a hard timeout period for a flow entry corresponding to the flow in each of the plurality of network devices, including the identified network device, in such a manner that the hard timeout period associated with the identified network device is first to timeout.

15. The storage medium of claim 11, further comprising instructions to report the flow table miss for the flow from the identified network device to the SDN controller.

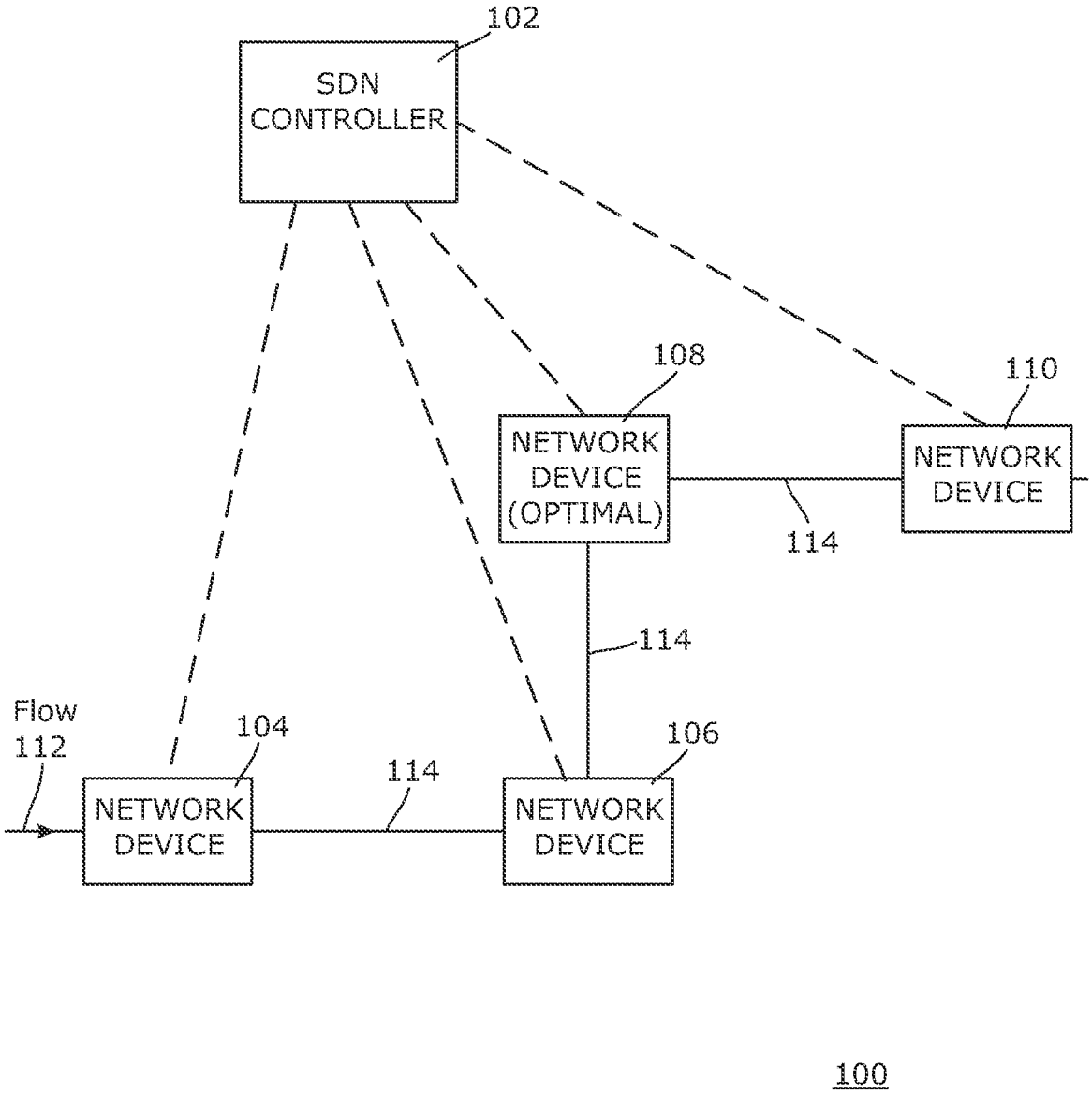

102
SDN CONTROLLER
108
NETWORK DEVICE (OPTIMAL)
110
NETWORK DEVICE
114
114
Flow 112
104
106
114
NETWORK DEVICE
NETWORK DEVICE
100

Fig. 1

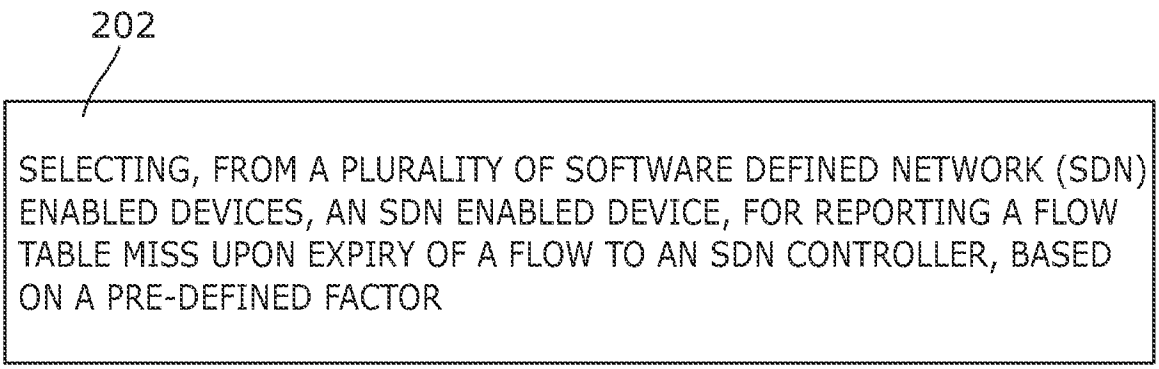

202
SELECTING, FROM A PLURALITY OF SOFTWARE DEFINED NETWORK (SDN) ENABLED DEVICES, AN SDN ENABLED DEVICE, FOR REPORTING A FLOW TABLE MISS UPON EXPIRY OF A FLOW TO AN SDN CONTROLLER, BASED ON A PRE-DEFINED FACTOR
200

Fig. 2

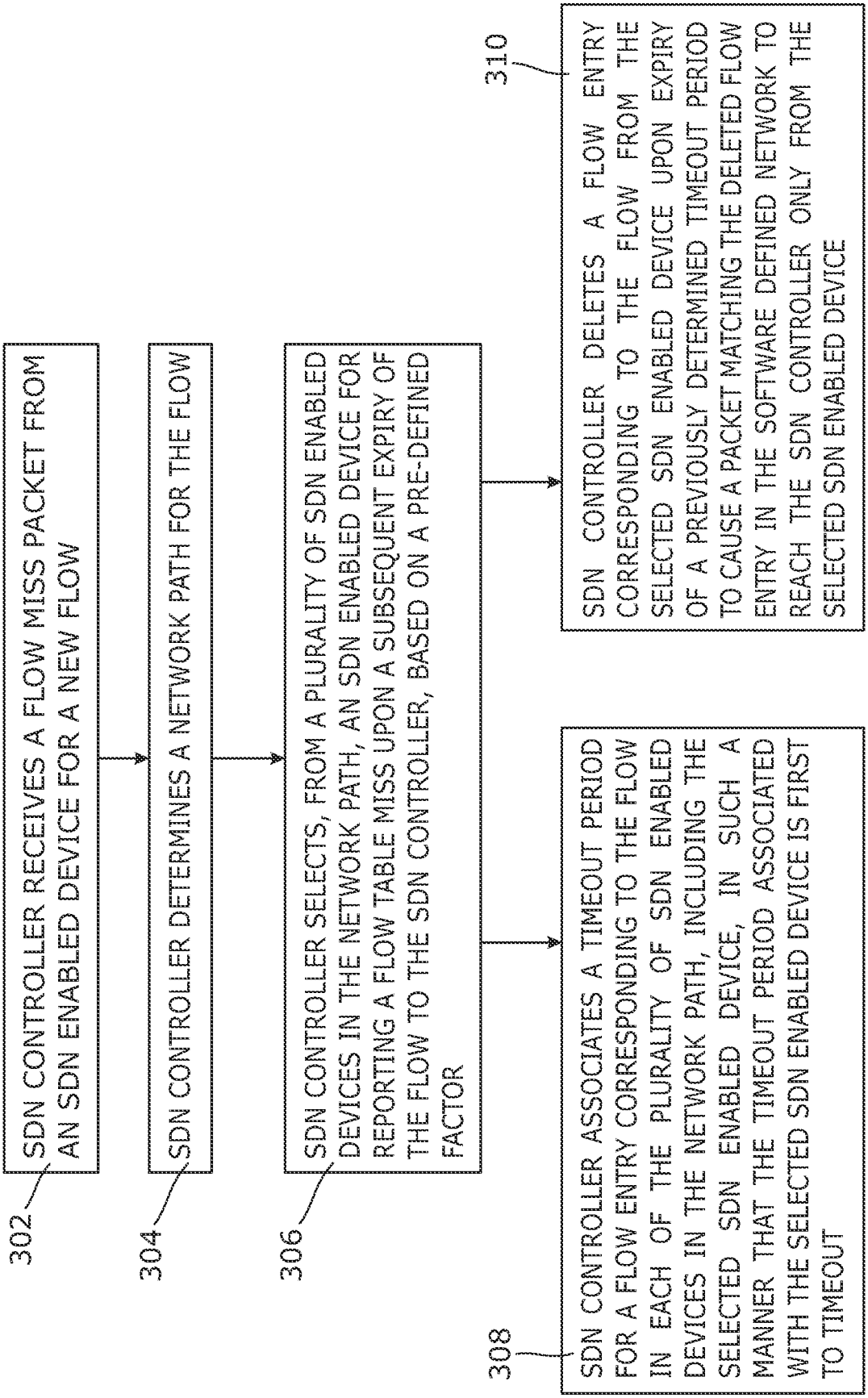

SDN CONTROLLER RECEIVES A FLOW MISS PACKET FROM AN SDN ENABLED DEVICE FOR A NEW FLOW
302
SDN CONTROLLER DETERMINES A NETWORK PATH FOR THE FLOW
304
SDN CONTROLLER SELECTS, FROM A PLURALITY OF SDN ENABLED DEVICES IN THE NETWORK PATH, AN SDN ENABLED DEVICE FOR REPORTING A FLOW TABLE MISS UPON A SUBSEQUENT EXPIRY OF THE FLOW TO THE SDN CONTROLLER, BASED ON A PRE-DEFINED FACTOR
306
SDN CONTROLLER ASSOCIATES A TIMEOUT PERIOD FOR A FLOW ENTRY CORRESPONDING TO THE FLOW IN EACH OF THE PLURALITY OF SDN ENABLED DEVICES IN THE NETWORK PATH, INCLUDING THE SELECTED SDN ENABLED DEVICE, IN SUCH A MANNER THAT THE TIMEOUT PERIOD ASSOCIATED WITH THE SELECTED SDN ENABLED DEVICE IS FIRST TO TIMEOUT
308
SDN CONTROLLER DELETES A FLOW ENTRY CORRESPONDING TO THE FLOW FROM THE SELECTED SDN ENABLED DEVICE UPON EXPIRY OF A PREVIOUSLY DETERMINED TIMEOUT PERIOD TO CAUSE A PACKET MATCHING THE DELETED FLOW ENTRY IN THE SOFTWARE DEFINED NETWORK TO REACH THE SDN CONTROLLER ONLY FROM THE SELECTED SDN ENABLED DEVICE
310
300

Fig. 3

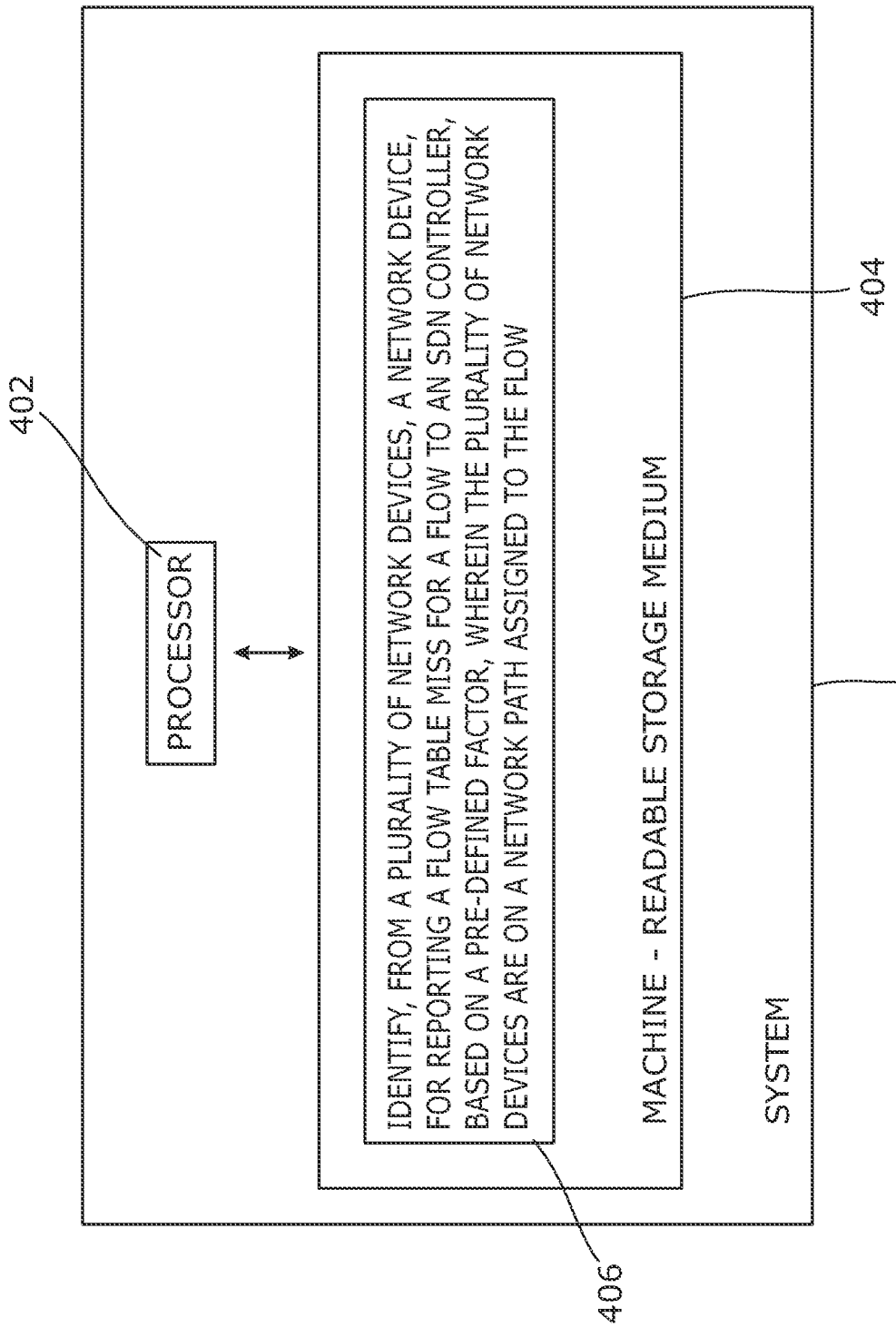

402
PROCESSOR
IDENTIFY, FROM A PLURALITY OF NETWORK DEVICES, A NETWORK DEVICE, FOR REPORTING A FLOW TABLE MISS FOR A FLOW TO AN SDN CONTROLLER, BASED ON A PRE-DEFINED FACTOR, WHEREIN THE PLURALITY OF NETWORK DEVICES ARE ON A NETWORK PATH ASSIGNED TO THE FLOW
406
MACHINE - READABLE STORAGE MEDIUM
404
SYSTEM
400

Fig. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/052534

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/801(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L 12/801; H04L 12/717; H04L 12/26; H04L 12/721; H04L 12/28; H04L 12/56

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: selecting, table miss, flow, expiry, SDN, SDN controller

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2013-0128746 A1 (KIRAN YEDAVALLI) 23 May 2013 See paragraphs [0024], [0044], [0057]-[0058], [0065]; claim 5; and figures 1, 5.	1-15
Y	OPEN NETWORKING FOUNDATION, `OpenFlow Switch Specification` version 1.4.0 (Wire protocol 0x05) 14 October 2013 See page 9, lines 17-20; page 15, lines 25-34; page 17, lines 9-33; page 18, lines 3-4; table 1; and figure 3.	1-15
A	US 2011-0317559 A1 (ANDRAS KERN et al.) 29 December 2011 See paragraphs [0029], [0035]-[0036]; and figure 7.	1-15
A	WO 2013-132395 A1 (TELEFONAKTIEBOLAGET L M ERICSSON (PUBL)) 12 September 2013 See page 5, lines 1-19; page 6, lines 1-10; and figure 4.	1-15
A	US 8391289 B1 (PRAVEEN YALAGANDULA et al.) 05 March 2013 See column 7, lines 19-38; and figure 4.	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 February 2015 (11.02.2015)	**25 February 2015 (25.02.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. ++82 42 472 3473

Authorized officer
KIM, Seong Woo
Telephone No. +82-42-481-3348

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/052534

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0128746 A1	23/05/2013	EP 2597831 A1	29/05/2013
		EP 2597831 B1	03/12/2014
		US 8644149 B2	04/02/2014
US 2011-0317559 A1	29/12/2011	CN 102959910 A	06/03/2013
		EP 2586163 A1	01/05/2013
		US 8897134 B2	25/11/2014
		WO 2011-161575 A1	29/12/2011
WO 2013-132395 A1	12/09/2013	CN 104145457 A	12/11/2014
		US 2013-0230047 A1	05/09/2013
		US 8705536 B2	22/04/2014
US 8391289 B1	05/03/2013	None	