



- (51) **International Patent Classification:**
H04L 12/891 (2013.01)
- (21) **International Application Number:**
PCT/US2015/022074
- (22) **International Filing Date:**
23 March 2015 (23.03.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** MENTZE, Duane Edward; 8000 Foothills Blvd., Roseville, California 95747 (US). CLARK, Charles F.; 8000 Foothills Blvd., Roseville, California 95747 (US). WACKERLY, Shaun; 8000 Foothills Blvd., Roseville, California 95747 (US).
- (74) **Agents:** FERGUSON, Christopher W. et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

- *as to the identity of the inventor (Rule 4.17(i))*
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

(54) **Title:** IMPLEMENTING POLICY INSTRUCTIONS IN MULTIPLE TABLES

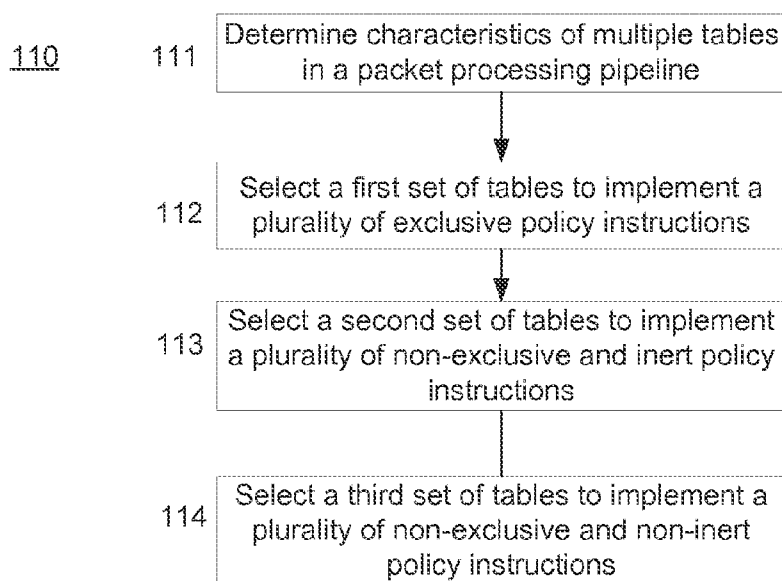


FIG. 1b

(57) **Abstract:** Example implementations relate to implementing policy instructions in multiple tables in a packet processing pipeline. An example method includes determining characteristics of multiple tables in a packet processing pipeline of a network device. The method further includes selecting a first set of tables from the multiple tables to implement a plurality of exclusive policy instructions, selecting a second set of tables from the multiple tables to implement a plurality of non-exclusive and inert policy instructions, and selecting a third set of tables from the multiple tables to implement a plurality of non-exclusive and non-inert policy instructions. The first set of tables, the second set of tables, and the third set of tables do not overlap.

IMPLEMENTING POLICY INSTRUCTIONS IN MULTIPLE TABLES

Background

[0001] Networks can include a plurality of resources connected by communication links, and can be used to connect people, provide services (e.g., internally and/or externally via the Internet and/or intranet), and/or organize information, among other activities associated with an entity. An example network can include a software-defined network (SDN).

Brief Description of the Drawings

[0002] Figure 1a illustrates a flow chart of an example method for implementing policy instructions in multiple tables, according to an example.

[0003] Figure 1b illustrates a flow chart of an example method for implementing policy instructions in multiple tables, according to an example.

[0004] Figure 2 illustrates an example environment with devices for implementing policy instructions in multiple tables, according to an example.

[0005] Figure 3 illustrates an example computer for implementing policy instructions in multiple tables, according to an example.

Detailed Description

[0007] Example implementations relate to implementing policy instructions in multiple tables in a packet processing pipeline. An example method includes determining characteristics of multiple tables in a packet processing pipeline of a network device. The method further includes selecting a first set of tables from the multiple tables to implement a plurality of exclusive policy instructions, selecting a second set of tables from the multiple tables to implement a plurality of non-exclusive and inert policy instructions, and selecting a third set of tables from the multiple tables to implement a plurality of non-exclusive and non-inert policy instructions. The first set of tables, the second set of tables, and the third set of tables do not overlap.

[0008] Networks can include a plurality of resources such as network devices and databases to connect endpoint devices via communication links. Networks can be used to connect people, provide services (e.g., internally and/or externally via the Internet and/or intranet), and organize information, among other activities. Examples of endpoint devices include computers, tablets, phones, printers, cameras, door locks, HVAC controller, among other endpoint devices capable of operating on a network. An example network can include a software-defined network (SDN).

[0009] SDN controllers can direct network devices such as servers, SDN-capable switches and routers, and other computing devices, on how to forward network traffic. SDN applications may execute on or interface with the SDN controller to provide input to the SDN controller and influence how the SDN controller forwards traffic. SDN applications might provide services on the network, including observing network traffic and conditions and taking one or more actions as a result. For instance, one application may look for infected hosts on the network, while another application may attempt to optimize voice over internet protocol (VoIP) calls on the network. Both applications may run on the same SDN controller, and use the SDN controller to communicate down to network devices in a protocol-specific format, such as according to the OpenFlow protocol.

[0010] When applications within a network, such as an SDN, want to tell the same devices in the network what to do, a conflict may arise between the instructions of one application and the instructions of another application with respect to the same endpoint device. In such instances, the SDN controller may be unable to determine which actions from which applications should be executed, and/or if the instructions of both applications should be executed.

[0011] Instructions from applications may be characterized as network policies to be applied to the network. Network policies from different applications may be compiled together to yield a cohesive set of non-overlapping policies to be applied to the network. This set of non-overlapping policies are referred to herein as “orthogonal policies”. An orthogonal policy is a policy generated from one or more original/source policies (e.g., policies that are received from an application) that does not conflict with any other orthogonal policy in a set of orthogonal policies. This means that all policies from the source set of policies to be applied to any single packet in the network would be implemented by a single orthogonal policy. These orthogonal policies may then be transformed into instructions for implementation by network devices. PCT Application No. US2015/015122, entitled “Network Policy Conflict Detection and Resolution” and filed on February 10, 2015, which is hereby incorporated by reference, describes in further detail how policies may be compiled in this manner.

[0012] Brute force compilation of logical terms of network policies includes the evaluation of how those terms overlap. As an example, if policy A requires that traffic from all wireless devices be sent to an intrusion prevention system and policy B requires that devices associated with an employee be given a particular priority level, the terms of policy A and policy B overlap in the case where an employee connects to the network with a wireless device. In general, the processing required to evaluate all overlaps is exponential in nature, and depends on the number of terms and the number of policies.

[0013] Therefore, policy compilation complexity can be exponential in nature for brute force implementations. The magnitude of such processing is a

function of the number of policies and can be represented using big O notation as follows:

$$PCC = \sim O(p^x)$$

where PCC is policy compilation complexity, p is the number of policies, and x is a value that depends on the particular compiler algorithm used by the policy engine compiler. As a result, policy compilation can use a significant number of resources and time, potentially resulting in poor network performance and creating issues if new policies are not able to be implemented quickly enough due to the compilation time. In addition, when policies are compiled together in a brute force manner, a large number of instructions are generated to implement those policies since in general the orthogonal policies increase in complexity as the number of source policies increases.

[0014] By grouping policies into multiple groups and compiling the groups separately, the compilation complexity can be reduced. However, as a result of the grouping and separate compilation of the grouped policies, there are multiple sets of instructions to be implemented in a packet processing pipeline of a network device. These sets of instructions should be implemented in the packet processing pipeline such that the intent of the policies is effected on the network by the network device. By considering characteristics of the available tables in the packet processing pipeline (e.g., the number of available tables, the size of the tables, and other characteristics), tables can be selected for each set of instructions to improve packet processing performance of the network device while effecting the intent of the policies.

[0015] FIGS. 1a and 1b illustrate methods to implement policy instructions in multiple tables of a packet processing pipeline, according to an example. Methods 100 and 110 may be performed by a computing device, computer, server, or the like, such as SDN controller 210 or computer 310. In some examples, network device 220 may be configured to perform these methods. Computer-readable instructions for implementing methods 100 and 110 may be stored on a computer readable storage medium. These instructions as stored on the medium are referred to herein as "modules" and may be executed by a computer.

[0016] Methods 100 and 110 will be described here relative to environment 200 of FIG. 2. Environment 200 may include SDN controller 210 and network device 220. SDN controller 210 may be a computer configured to manage the control plane of a software defined network. SDN controller 210 may include/be implemented by one or multiple computers. Network device 220 may be a network infrastructure device, such as a switch or router, of the software defined network. The network device 220 may thus be part of the data plane of the software defined network, which may include multiple network devices. SDN controller 210 may communicate with network device 220 via an SDN protocol, such as the OpenFlow protocol. SDN controller 210 may program rules in the packet processing pipeline 222 of network device 220. Network device 220 may use these rules to process and forward network traffic. Additionally, a variety of SDN applications may run on or interface with SDN controller 210. These SDN applications may be part of the application plane of the software defined network.

[0017] SDN controller 210 and network device 220 may include one or more controllers and one or more machine-readable storage media. A controller may include a processor and a memory for implementing machine readable instructions. The processor may include at least one central processing unit (CPU), at least one semiconductor-based microprocessor, at least one digital signal processor (DSP) such as a digital image processing unit, other hardware devices or processing elements suitable to retrieve and execute instructions stored in memory, or combinations thereof. The processor can include single or multiple cores on a chip, multiple cores across multiple chips, multiple cores across multiple devices, or combinations thereof. The processor may fetch, decode, and execute instructions from memory to perform various functions. As an alternative or in addition to retrieving and executing instructions, the processor may include at least one integrated circuit (IC), other control logic, other electronic circuits, or combinations thereof that include a number of electronic components for performing various tasks or functions.

[0018] The controller may include memory, such as a machine-readable storage medium. The machine-readable storage medium may be any

electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, the machine-readable storage medium may comprise, for example, various Random Access Memory (RAM), Read Only Memory (ROM), flash memory, and combinations thereof. For example, the machine-readable medium may include a Non-Volatile Random Access Memory (NVRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a NAND flash memory, and the like. Further, the machine-readable storage medium can be computer-readable and non-transitory. Additionally, SDN controller 210 and network device 220 may include one or more machine-readable storage media separate from the one or more controllers.

[0019] Turning to FIG 1a, method 100 may be used to implement policy instructions in multiple tables of a packet processing pipeline, according to an example. At 101, the SDN controller 210 may divide a plurality of network policies into an exclusive policy group and a non-exclusive policy group. For example, the grouping module 211 may perform this task. The plurality of network policies may be received from various sources. For example, the network policies may be received from SDN applications running on or interfacing with SDN controller 210.

[0020] Exclusive policies are policies with associated actions that cannot be combined with the actions of any other policy. For example, a policy may require that all network traffic of a certain type be quarantined and not otherwise be processed. Such a policy is an exclusive policy as the point of the policy is to dictate all processing for that particular type of traffic. In this case, the purpose of the exclusive policy may be to provide network security. Thus, the actions of that policy should not be combined with the actions of any other policy, whether that other policy is an exclusive policy or non-exclusive policy. This is thus a constraint that would be applied during the compilation process of the exclusive policy group, so that two policies with actions intended for the same type of network traffic are not combined. For this reason, exclusive policies can be grouped separately from non-exclusive policies and also compiled separately, since the exclusive policy would always take precedence.

Of course, it may be possible for two exclusive policies to relate to the same type of network traffic. In such a case, the exclusive policy with the higher priority takes precedence, and the other exclusive policy would not be applied. The SDN application responsible for the policy can designate whether the policy is exclusive and can also indicate the priority level of the policy.

[0021] In contrast, non-exclusive policies are policies with associated actions that can be combined with the actions of other policies. For example, policy A may require a first action to be applied to a type of network traffic and policy B may require a second action to be applied to that same type of network traffic, each of which are not inconsistent with each other. Accordingly, because the policy actions are not mutually exclusive and can both be applied to the same network traffic, when compiling non-exclusive policies there is no need to impose the constraint that actions from two different policies cannot be applied to the same network traffic. For this reason, non-exclusive policies can be grouped together for compilation separate from the compilation of exclusive policies.

[0022] In light of this, the grouping module 211 divides the plurality of policies into an exclusive policy group and a non-exclusive policy group. At 102, SDN controller 210 may further divide the non-exclusive policies (from the non-exclusive policy group) into an inert group and a non-inert group. Inert policies are policies that do not change a packet the policy is applied to or alter the packet's delivery to an intended destination. A non-inert policy is a policy that does change a packet the policy is applied to or alter the packet's delivery to an intended destination. For example, a policy that directs the network device 220 to change a value in the header field of a packet would be a non-inert policy because the packet is being changed as a result of the policy. Similarly, a policy that blocks a packet does not change the packet but does prevent the packet from being forwarded to its intended destination, and is thus also a non-inert policy. In contrast, a policy that simply copies the packet or collects statistics related to the packet would be an inert policy because the packet is not being changed and its delivery to the intended destination is not being altered. Because of this difference, inert policies can be separately

compiled from non-inert policies. However, as described below, inert policy actions should be applied to a packet before non-inert policy actions are applied.

[0023] At 103, the SDN controller 210 may compile each group of policies into a first, second, and third plurality of orthogonal policies. For example, the exclusive policies can be compiled into a first plurality of orthogonal policies, the non-exclusive and inert policies can be compiled into a second plurality of orthogonal policies, and the non-exclusive and non-inert policies can be compiled into a third plurality of orthogonal policies. Policy compilation module 212 can perform this task as described in PCT Application No. US2015/015122. The policy groups may be compiled separately by SDN controller 210 in various ways. For example, SDN controller 210 may compile the policy groups at different times, using different processing resources, or both. As a result, assuming that there is at least one policy in each group, the policy compilation complexity is reduced because the number of policies in each group is less than the total number of policies. This reduction in complexity is illustrated by the following equation using big O notation:

$$PCC = O(2^x + 2^y + 2^z) < O(2^{x+y+z})$$

where PCC is policy compilation complexity, x is the number of policies in the exclusive policy group, y is the number of policies in the non-exclusive and inert policy group, and z is the number of policies in the non-exclusive and non-inert policy group.

[0024] At 104 SDN controller 210 may generate policy protocol-specific instructions to implement each of the plurality of orthogonal policies. Policy compilation module 212 can perform this task as described in PCT Application No. US2015/015122. The protocol-specific instructions may be instructions in accordance with a protocol supported by network device 220, such as the OpenFlow protocol. The protocol-specific instructions may thus be instructions suitable for the network device 220 to implement the policies when processing and forwarding traffic. In particular, the protocol-specific instructions may be instructions for creating or modifying flow entries in flow tables in the packet processing pipeline 222, where the flow tables are consulted to determine how

to process and forward a received packet. The protocol-specific instructions are referred to herein as “policy instructions”.

[0025] Thus, a plurality of exclusive policy instructions may be generated for the first plurality of orthogonal policies (corresponding to the exclusive policy group), a plurality of non-exclusive and inert policy instructions may be generated for the second plurality of orthogonal policies (corresponding to the non-exclusive and inert policy group), and a plurality of non-exclusive and non-inert policy instructions may be generated for the third plurality of orthogonal policies (corresponding to the non-exclusive and non-inert policy instructions).

[0026] At 105, a separate set of tables may be selected for implementation of each plurality of policy instructions. This can be performed in accordance with method 110 in FIG. 1b.

[0027] Method 110 begins at 111, where SDN controller 210 determines characteristics of tables in packet processing pipeline 222 of network device 220. Determination module 213 may perform this task. The tables in the packet processing pipeline 222 may be implemented by hardware such as Tertiary Content Addressable Memories (TCAMs). The characteristics of the tables may include the number of tables available in the pipeline 222, the capacity of each table, and the capabilities of each table. The table characteristics may already be stored in a memory of SDN controller 210. The table characteristics may also be determined by SDN controller 210 interrogating network device 220 via a control channel and requesting such information.

[0028] At 112, SDN controller 210 may select a first set of tables to implement the plurality of exclusive policy instructions. Selection module 214 may perform this task. For example, the first set of tables 225 may include one or more tables. At 113, selection module 214 may select a second set of tables to implement the plurality of non-exclusive and inert instructions. For example, the second set of tables 226 may include one or more tables. At 114, selection module 214 may select a third set of tables to implement the plurality of non-exclusive and non-inert instructions. For example, the third set of tables 227 may include one or more tables.

[0029] For blocks 112-114, the selection of the tables may depend on a variety of factors. For example, a set of tables may be selected based on the amount of space needed to implement the given instructions. For instance, if there are many more non-exclusive and inert policy instructions than exclusive policy instructions, more tables may need to be selected for the non-exclusive and inert policy instructions to accommodate all of the instructions. Conversely, fewer tables may be selected for the exclusive policy instructions. Another factor that may be considered is the number of hardware resources (e.g., Application Specific Integrated Circuit (ASIC) resources) required to implement the instructions. The capabilities of certain tables may require more or fewer resources to be applied for a given set of instructions.

[0030] Certain characteristics of the instructions can be considered, as well. To this end, determination module 213 may be used to determine various characteristics of each plurality of instructions, such as the type of actions that are to be applied to a packet in accordance with the instructions. The evaluation module 215 may be used to evaluate which tables would be suitable for each plurality of instructions based on the determined characteristics of the tables and of the instructions. For example, instructions that require that a matching packet be encapsulated for the purpose of Virtual Local Area Network (VLAN) tunneling would require a table with that capability for appropriate implementation of the instructions.

[0031] Additionally, tables may be selected based on the fact that a particular type of policy from which the instructions were generated is a large-scale non-exclusive and inert policy. A large-scale policy is a policy that is expected to have many instances. For example, policy requests received from an SDN application that optimizes network communication for Voice Over Internet Protocol (VOIP) calls on a large network (such as for an large enterprise) may be expected to generate many instances of the policy request since it will generate a new instance for every new call. That knowledge can be used to select an appropriate table or set of tables to accommodate the number of instructions expected to be generated for that policy. To be aware of a large-scale policy, the SDN controller 210 can keep track of policy requests and

identify a threshold at which policy requests of a certain type should be designated as a large-scale policy. The threshold could be a fixed value or could be relative to other policy requests (e.g., a percentage of total requests). Alternatively, the SDN application could inform the SDN controller 210 that a particular policy request should be classified as a large-scale policy. Another approach would be for the SDN controller 210 to be pre-programmed with the knowledge that requests from a certain source or application should be classified as large-scale policies.

[0032] At 106, SDN controller 210 may specify the priority of each set of instructions. For example, SDN controller 210 may specify that flow entries corresponding to the plurality of exclusive policy instructions have priority over flow entries corresponding to the non-exclusive policy instructions. Additionally, SDN controller 210 may specify that flow entries corresponding to the plurality of non-exclusive and inert policy instructions have priority over flow entries corresponding to the non-exclusive and non-inert policy instructions. Examples of the implementation of priorities of this type are described now with reference to block 107.

[0033] At 107, SDN controller 210 may instruct network device 220 to create flow entries in each set of tables 225, 226, 227 for the corresponding instructions that the tables were selected for. Instruction module 216 may perform this task. For example, instruction module 216 may instruct network device 220 to create flow entries corresponding to the plurality of exclusive policy instructions in the first set of tables 225, flow entries corresponding to the plurality of non-exclusive and inert policy instructions in the second set of tables 226, and flow entries corresponding to the plurality of non-exclusive and non-inert policy instructions in the third set of tables 227.

[0034] The packet processing pipeline 222 can be organized in this way to enforce the priority of the different sets of instructions. A received packet will be first attempted for match against flow entries in the first set of tables 225, since the exclusive policy instructions have the highest priority. If the packet matches an entry in one of the tables, the appropriate actions corresponding to the matching entry will be applied to the packet and processing through packet

processing pipeline 222 will cease for that packet (End). If a first table in set 225 does not result in a match, the flow table miss entry will cause network device 220 to attempt to match the packet to the next table in set 225 until either there is a match or there is a final miss. At a final miss, network device will move to the next set of tables in the packet processing pipeline 222, which is the second set of tables 226.

[0035] Network device 220 attempts to match the received packet to an entry in the flow tables of the second set of tables 226. In some implementations, because of the nature of non-exclusive and inert policy instructions, the compilation process may be carried out such that the compiled policies are not orthogonal, in which case multiple potential matching instructions may be implemented through multiple flow entries in set 226. In such a case, the network device 220 can attempt to match the received packet to all entries in set 226 even if there has already been a match. After having traversed the entire set of tables in set 226, network device may move to the third set of tables 227. Of course, if compilation of the non-exclusive and non-inert policy group is carried out to yield orthogonal policies, after a single match the network device 220 would move to the third set of tables 227.

[0036] Network device 220 attempts to match the received packet to an entry in the flow tables of the third set of tables 227. Similar to the exclusive policy instructions, if there is a match to a flow entry in the third set of tables 227, the appropriate actions corresponding to the matching entry will be applied to the packet and processing through pipeline 222 will cease for that packet (End). If a first table in set 227 does not result in a match, the flow table miss entry will cause network device 220 to attempt to match the packet to the next table in set 227 until either there is a match or there is a final miss. At a final miss, processing through pipeline 222 will cease for the packet (End).

[0037] FIG. 3 illustrates a computer to compile and implement policies, according to an example. Computer 310 may be part of SDN controller 210 or network device 220. The computer may include one or more controllers and one or more machine-readable storage media, as described with respect to SDN controller 210 and network device 220, for example.

[0038] Processor 320 may be at least one central processing unit (CPU), at least one semiconductor-based microprocessor, other hardware devices or processing elements suitable to retrieve and execute instructions stored in machine-readable storage medium 330, or combinations thereof. Processor 320 can include single or multiple cores on a chip, multiple cores across multiple chips, multiple cores across multiple devices, or combinations thereof. Processor 320 may fetch, decode, and execute instructions 332-336 among others, to implement various processing. As an alternative or in addition to retrieving and executing instructions, processor 320 may include at least one integrated circuit (IC), other control logic, other electronic circuits, or combinations thereof that include a number of electronic components for performing the functionality of instructions 332-336. Accordingly, processor 320 may be implemented across multiple processing units, and instructions 332-336 may be implemented by different processing units in different areas of computer 310.

[0039] Machine-readable storage medium 330 may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, the machine-readable storage medium may comprise, for example, various Random Access Memory (RAM), Read Only Memory (ROM), flash memory, and combinations thereof. For example, the machine-readable medium may include a Non-Volatile Random Access Memory (NVRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a NAND flash memory, and the like. Further, the machine-readable storage medium 330 can be computer-readable and non-transitory. Machine-readable storage medium 330 may be encoded with a series of executable instructions for managing processing elements.

[0040] The instructions 332-336 when executed by processor 320 (e.g., via one processing element or multiple processing elements of the processor) can cause processor 320 to perform processes, for example, methods 100, 110, and/or variations and portions thereof. Instructions 332-336 will now be briefly described, which description should be read in light of the description of methods 100 and 110, and environment 200 above.

[0041] Computer 310 may implement policy instructions in multiple tables. For example, determination instructions 332 may cause processor 320 to determine characteristics of multiple tables in a packet processing pipeline 222 of a network device 220. Selection instructions 334 may cause processor 320 to select a first set of tables 225 to implement a plurality of exclusive policy instructions, a second set of tables 226 to implement a plurality of non-exclusive and inert policy instructions, and a third set of tables 227 to implement a plurality of non-exclusive and non-inert policy instructions. The first, second, and third sets of tables may be separate sets, such that the tables do not overlap. Evaluation instructions 336 may cause processor 320 to evaluate characteristics of the tables and of the instructions, where the evaluation can be used to select appropriate tables for each plurality of instructions.

[0042] In the present disclosure, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration how a number of examples of the disclosure can be practiced. These examples are described in sufficient detail to enable those of ordinary skill in the art to practice the examples of this disclosure, and it is to be understood that other examples can be used and that process, electrical, and/or structural changes can be made without departing from the scope of the present disclosure.

[0043] As used herein, "logic" is an alternative or additional processing resource to perform a particular action and/or function, etc., described herein, which includes hardware, e.g., various forms of transistor logic, application specific integrated circuits (ASICs), etc., as opposed to computer executable instructions, e.g., software firmware, etc., stored in memory and executable by a processor. Further, as used herein, "a" or "a number of" something can refer to one or more such things. For example, "a number of widgets" can refer to one or more widgets. Also, as used herein, "a plurality of" something can refer to more than one of such things.

[0044] The above specification, examples and data provide a description of the method and applications, and use of the system and method of the present disclosure. Since many examples can be made without departing from the spirit and scope of the systems and methods of the present disclosure, this

specification merely sets forth some of the many possible embodiments, configurations, and implementations.

What is claimed is:

1. A method for implementing policy instructions in multiple tables in a packet processing pipeline, comprising, by a processor:

determining characteristics of multiple tables in a packet processing pipeline of a network device;

selecting a first set of tables from the multiple tables to implement a plurality of exclusive policy instructions;

selecting a second set of tables from the multiple tables to implement a plurality of non-exclusive and inert policy instructions;

selecting a third set of tables from the multiple tables to implement a plurality of non-exclusive and non-inert policy instructions,

the first set of tables, the second set of tables, and the third set of tables not overlapping.

2. The method of claim 1, wherein the network device is a software defined network (SDN)-capable switch and the instructions are to be used to create flow entries in flow tables stored in the tables of the SDN-capable switch.

3. The method of claim 2, further comprising specifying that flow entries corresponding to the plurality of exclusive policy instructions have priority over flow entries corresponding to the non-exclusive policy instructions, and that flow entries corresponding to the plurality of non-exclusive and inert policy instructions have priority over flow entries corresponding to the non-exclusive and non-inert policy instructions.

4. The method of claim 3, wherein:

the network device attempts to match a received packet to flow entries in the first set of tables before attempting to match a received packet to flow entries in the second and third sets of tables in the packet processing pipeline,

the network device attempts to match the received packet to flow entries in the second and third sets of tables only if the received packet does not match any flow entry in the first set of tables, and

the network device attempts to match the received packet to flow entries in the third set of tables after attempting to match the received packet to flow entries in the second set of tables.

5. The method of claim 1, further comprising:
determining characteristics of one or more of the pluralities of instructions,
evaluating which tables would be suitable for the one or more of the pluralities of instructions based on the characteristics of the tables and the characteristics of the one or more of the pluralities of instructions,
wherein the selecting takes the evaluation into account when selecting the tables.

6. The method of claim 1, wherein the selecting selects tables to implement instructions so as to minimize the number of resources in the packet processing pipeline required for implementation of the instructions or otherwise optimize performance of the packet processing pipeline.

7. The method of claim 1, further comprising:
recognizing that instructions in the plurality of non-exclusive and inert policy instructions relate to a large-scale policy, a large-scale policy being a policy that is expected to have additional instances,
wherein selecting the second set of tables includes selecting tables that can accommodate the large-scale policy.

8. A controller in a software defined network (SDN), comprising:
a determination module to determine characteristics of multiple tables in a packet processing pipeline of a network device; and
a selection module to:

select a first set of tables from the multiple tables to implement a plurality of exclusive policy instructions;

select a second set of tables from the multiple tables to implement a plurality of non-exclusive and inert policy instructions;

select a third set of tables from the multiple tables to implement a plurality of non-exclusive and non-inert policy instructions,

the first set of tables, the second set of tables, and the third set of tables not overlapping.

9. The controller of claim 8, wherein the network device is a software defined network (SDN)-capable switch configured to use the instruction to create flow entries in flow tables stored in the tables of the SDN-capable switch.

10. The controller of claim 9, further comprising:

an instruction module to instruct the network device to create flow entries corresponding to the plurality of exclusive policy instructions in the first set of tables, create flow entries corresponding to the plurality of non-exclusive and inert policy instructions in the second set of tables, and create flow entries corresponding to the plurality of non-exclusive and non-inert policy instructions in the third set of tables.

11. The controller of claim 10, wherein the instruction module is to instruct the network device to give priority in the packet processing pipeline to flow entries corresponding to the plurality of exclusive policy instructions over flow entries corresponding to the non-exclusive policy instructions, and to give priority in the packet processing pipeline to flow entries corresponding to the plurality of non-exclusive and inert policy instructions over flow entries corresponding to the non-exclusive and non-inert policy instructions.

12. The controller of claim 11, wherein the instruction module is to instruct the network device (1) to attempt to match a received packet to flow entries in the first set of tables before attempting to match a received packet to

flow entries in the second and third sets of tables in the packet processing pipeline, (2) to attempt to match the received packet to flow entries in the second and third sets of tables only if the received packet does not match any flow entry in the first set of tables, and (3) to attempt to match the received packet to flow entries in the third set of tables after attempting to match the received packet to flow entries in the second set of tables.

13. The controller of claim 8, further comprising:

the determining module to determine characteristics of one or more of the pluralities of instructions;

an evaluation module to evaluate which tables would be suitable for the one or more of the pluralities of instructions based on the characteristics of the tables and the characteristics of the one or more of the pluralities of instructions;

the selection module to take the evaluation of the evaluation module into account when selecting the tables.

14. The controller of claim 8, wherein the selection module is to select tables to implement instructions so as to minimize the number of resources in the packet processing pipeline of the network device required for implementation of the instructions or otherwise optimize performance of the packet processing pipeline.

15. The controller of claim 1,

the evaluation module to recognize that instructions in the plurality of non-exclusive and inert policy instructions relate to a large-scale policy, a large-scale policy being a policy that is expected to have additional instances,

the selection module to select the second set of tables such that the selected tables can accommodate the large-scale policy.

16. A non-transitory computer-readable storage medium storing instructions that, when executed by a processor, cause the processor to:

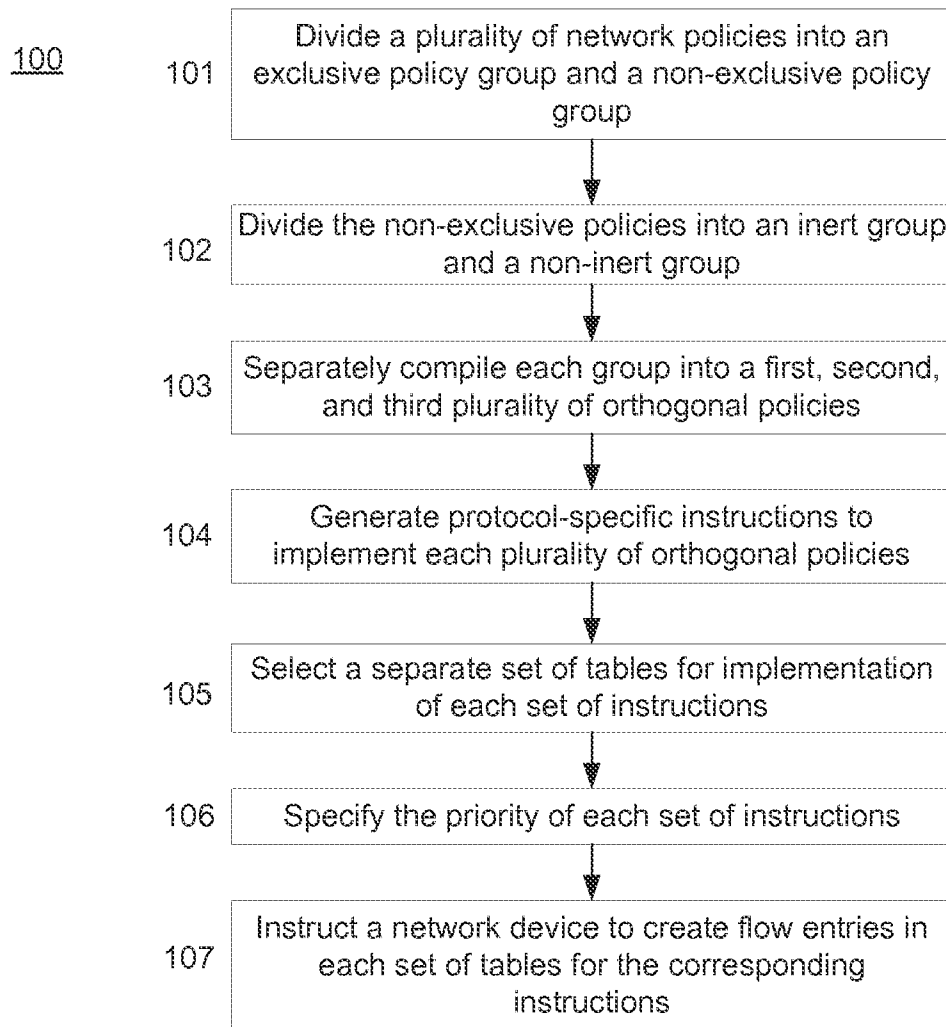
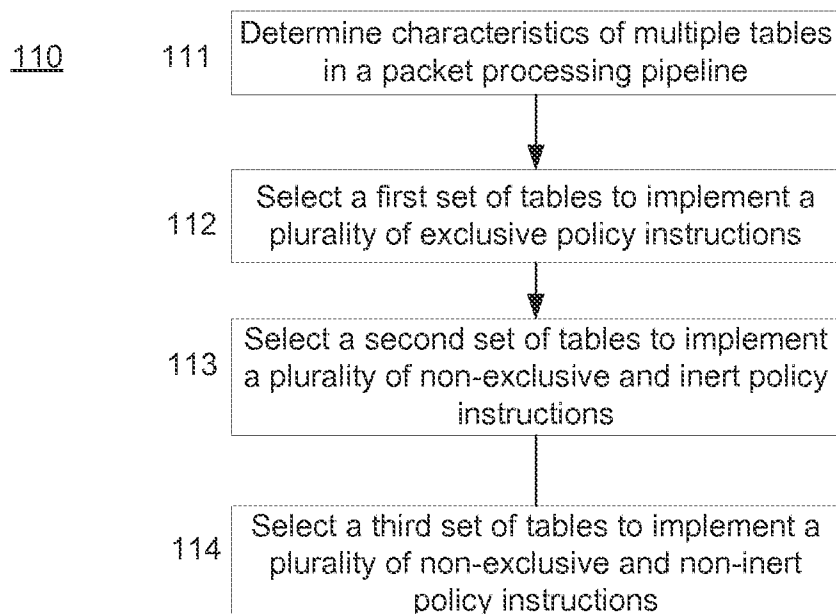
determine characteristics of multiple tables in a packet processing pipeline of a network device;

select a first set of tables from the multiple tables to implement a plurality of exclusive policy instructions;

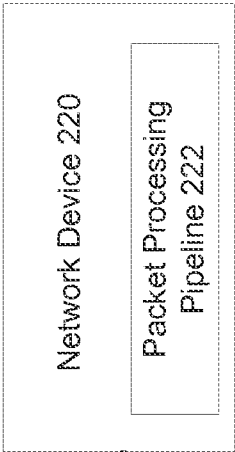
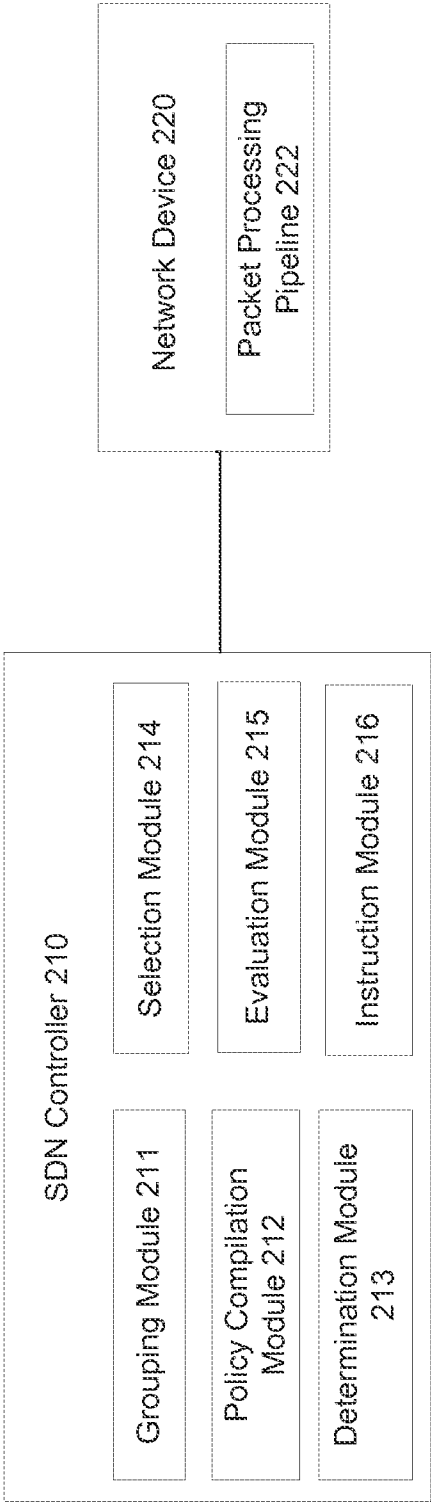
select a second set of tables from the multiple tables to implement a plurality of non-exclusive and inert policy instructions;

select a third set of tables from the multiple tables to implement a plurality of non-exclusive and non-inert policy instructions,

the first set of tables, the second set of tables, and the third set of tables not overlapping.

**FIG. 1a****FIG. 1b**

200



Packet Processing Pipeline 222

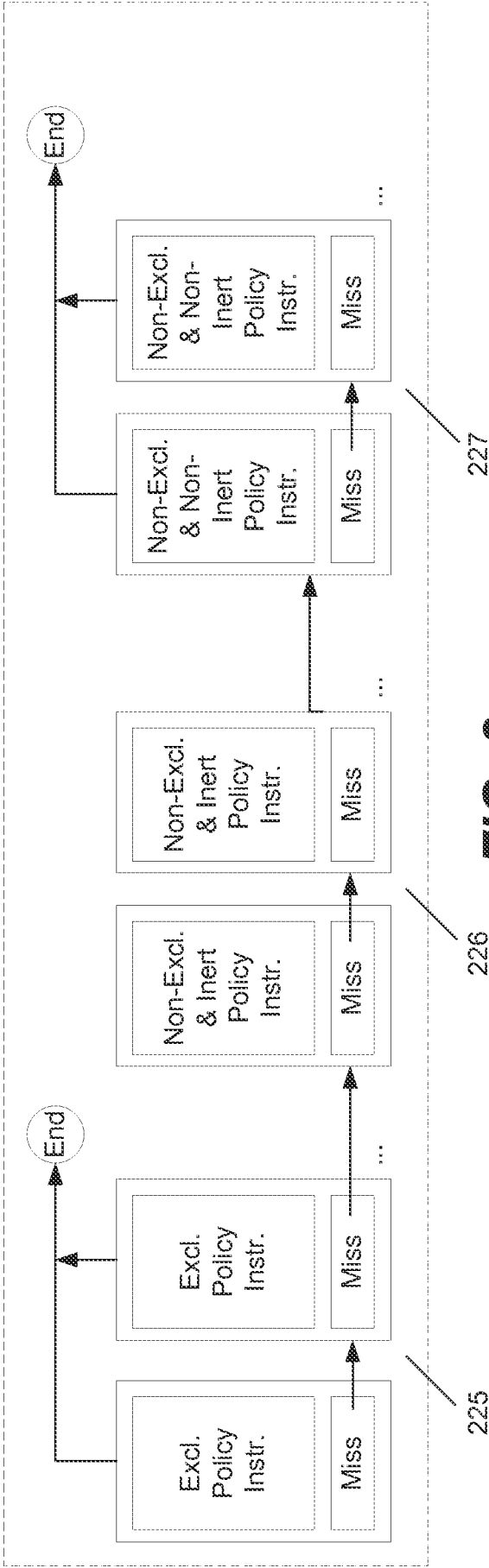
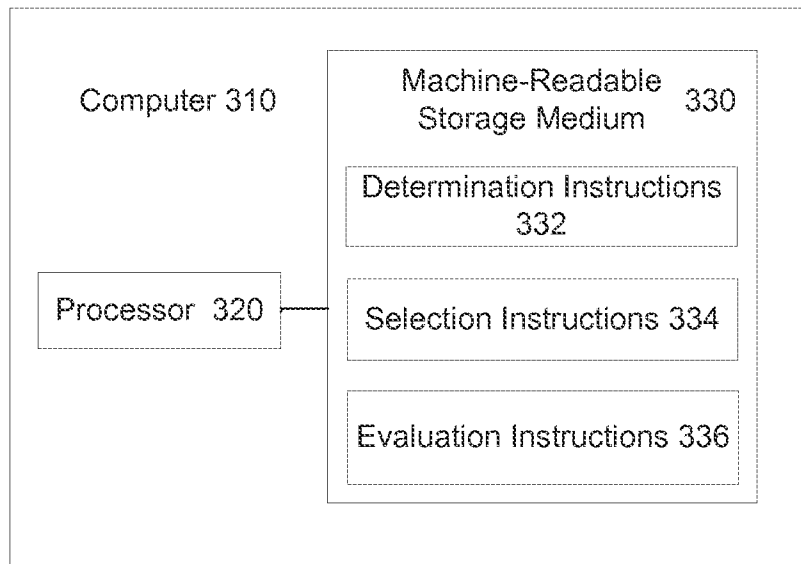


FIG. 2

**FIG. 3**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/022074**A. CLASSIFICATION OF SUBJECT MATTER****H04L 12/891(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 12/891; G06F 15/173; H04L 12/851; H04L 12/741; H04L 9/00; 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: table, select, policy, non-overlapping, SDN.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014-0146674 A1 (FUTUREWEI TECHNOLOGIES, INC.) 29 May 2014 See paragraphs [0005]-[0008]; claims 15-16; and figures 10-14.	1-16
A	US 7554980 B1 (CHIANG YEH et al.) 30 June 2009 See column 6, line 9 - column 7, line 44; claims 21-23; and figures 2-3.	1-16
A	US 2014-0241356 A1 (TELEFONAKTIEBOLAGET L M ERICSSON(PUBL)) 28 August 2014 See paragraphs [0054]-[0060], [0067]-[0071]; claims 1-4; and figures 3-6.	1-16
A	US 2008-0320550 A1 (JOHN C. STRASSNER et al.) 25 December 2008 See paragraphs [0036]-[0047]; claims 1-4; and figures 3-6.	1-16
A	US 6327618 B1 (JOHN K. AHLSTROM et al.) 04 December 2001 See column 5, line 35 - column 6, line 15; claim 1; and figures 2A-2C.	1-16



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

14 December 2015 (14.12.2015)

Date of mailing of the international search report

02 March 2016 (02.03.2016)

Name and mailing address of the ISA/KR

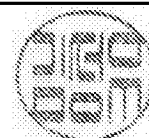
International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/022074

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0146674 A1	29/05/2014	WO 2014-082590 A1	05/06/2014
US 7554980 B1	30/06/2009	None	
US 2014-0241356 A1	28/08/2014	US 8964752 B2 WO 2014-128598 A1	24/02/2015 28/08/2014
US 2008-0320550 A1	25/12/2008	US 8327414 B2 WO 2009-002697 A1	04/12/2012 31/12/2008
US 6327618 B1	04/12/2001	None	