SOFTWARE DEFINED NETWORKING FOR HYBRID NETWORKS

An SDN controller for a hybrid network including legacy switches and SDN switches, the SDN controller including a topology viewer to receive from the SDN switches discovery messages sent by other SDN switches as directed by the SDN controller, to receive from the SDN switches routing information messages including internal routing topologies sent by the legacy switches, to receive from the SDN switches intercepted neighbor relationship messages sent by the legacy switches, and to determine a global topology of the hybrid network by determining direct links between SDN switches based on the discovery messages, by determining links between legacy switches based on the interior gateway protocol message, and by determining links between SDN switches and legacy switches based on the neighbor relationship messages.

Declarations under Rule 4.17:
— as to the identity of the inventor (Rule 4.17(ii))
— 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))
SOFTWARE DEFINED NETWORKING FOR HYBRID NETWORKS

Background

[0001] Software Defined Networking (SDN) provides advantages over networks consisting of legacy networking devices (i.e., non-programmable packet forwarding devices, such as switches and routers). For example, by enabling dynamic programming of network-wide forwarding states, SDN provides flexibility for achieving centralized, fine-grained network traffic control, reduces link congestion, and enables fast failure recovery. Simultaneously upgrading all legacy networking devices of a network to SDN devices (i.e. programmable packet forwarding devices) can be cost prohibitive and operationally burdensome (e.g. if the network must remain operational during such a conversion). As such, SDN devices are typically incrementally introduced into a network, resulting in a hybrid network including both SDN and legacy networking devices until the network is completely transitioned to a fully SDN network.

Brief Description of the Drawings

[0002] Figure 1 is a block and schematic diagram generally illustrating an SDN deployment planner according to one example.
[0003] Figure 2 is a table illustrating a representative example of a link states for an example network.
[0004] Figure 3 is a block and schematic diagram generally illustrating an SDN controller for a hybrid network according to one example.

[0005] Figure 4 is a flow diagram illustrating a method of transitioning a legacy network to an SDN network according to one example.

[0006] Figure 5 is a block and schematic diagram generally illustrating a computing system for implementing an SDN deployment planner and an SDN controller according to one example.

[0007] Figure 6 is a block and schematic diagram generally illustrating a non-transitory computer-readable medium including computer executable instructions for implementing an SDN deployment planner, according to one example.

Detailed Description

[0008] In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific examples in which the disclosure may be practiced. It is to be understood that other examples may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present disclosure is defined by the appended claims. It is to be understood that features of the various examples described herein may be combined, in part or whole, with each other, unless specifically noted otherwise.

[0009] Software-Defined Networking (SDN) provides advantages over networks consisting of traditional legacy-type networking devices (i.e., non-programmable packet forwarding devices, such as switches and routers). By enabling dynamic programming of network-wide forwarding states, SDN provides flexibility to achieve centralized, fine-grained network traffic engineering (TE), provides reduced link congestion, and enables fast failure recovery.

[0010] While the full potential of many SDN application may be realized only when SDN is fully deployed, simultaneously upgrading all legacy networking
devices of a network to SDN devices (i.e. programmable packet forwarding devices) can be cost prohibitive and operationally burdensome (e.g. the network must remain operational during such a conversion). As a result, SDN devices are typically incrementally introduced into a network, thereby creating a hybrid network including both SDN and legacy networking devices until the network is completely transitioned to a fully SDN network.

[0011] Presently, when incrementally upgrading non-programmable legacy packet forwarding devices with programmable SDN packet forwarding devices (referred to herein simply as “SDN switches”), network operators, including ISP (Internet Service Provider) and Enterprise network operators, often upgrade only network edge devices for quality of service (QoS) and security related applications. However, such an upgrade strategy does not take advantage of improved TE (e.g., load balancing) and failure recovery applications enabled by SDN.

[0012] As will be described in greater detail below, given a budget (i.e., a number) of legacy forwarding devices to be replaced, rather than replacing only edge devices, the present disclosure provides a system and techniques for identifying a number of legacy forwarding devices in a network (e.g., ISP and Enterprise networks) to be replaced with SDN forwarding devices to best leverage TE and load balancing benefits afforded by SDN forwarding devices in the resulting hybrid network, such as by minimizing maximum link usage (e.g., ratio of link load to link bandwidth).

[0013] Figure 1 is block and schematic diagram generally illustrating an SDN deployment planner 50 (also referred to simply as deployment planner 50), according to one example, for identifying a number of legacy packet forwarding devices from a plurality of legacy packet forwarding devices, L1 to L8 (also referred to simply as “legacy switches”) of a network 60 to be replaced with SDN packet forwarding devices (also referred to simply as “SDN switches”).

[0014] Each of the legacy switches L1 to L8 of network 60 are interconnected to one or more of the other legacy switches via a plurality of links, as indicated by the link 62 between legacy switches L1 and L6. Network 60 may be one of any number of different network types, such as an ISP network or an enterprise
network for example. In one example, deployment planner 50 determines legacy switches for replacement based on optimizing traffic engineering (TE) goals, such as minimizing maximum link loads, for example, in view of one or more constraints, such as a number of legacy switches budgeted for replacement by the network administrator and link capacities, for example. Any number of other constraints could also be considered, such as available SDN versions and other hardware constraints, for instance.

[0015] In one example, as indicated at 52, deployment planner 50 receives information regarding network 60 such as topology information and traffic history information, for example. In one instance, such information is received from an administrator of network 60. In one example, traffic history includes information describing packet flow rates on links and legacy switches. In one example, topology information includes link-state information for each legacy switch describing direct links to other legacy switches of the network, and a “cost” or “weight” associated with each link, where such weight is typically set by a network administrator and is based on factors such as link-type, link-bandwidth, link load, link latency, and link length, for example.

[0016] Figure 2 is a table 70 representing an example of link-state information for legacy switches L3 and L4 of example network 60, which may be included as topology information at 52. As indicated, legacy switch L3 has direct links to legacy switches L4, L5, L6, and L8, and legacy switch L4 has directly links with legacy switches L2, L3, an L8, with each of the links having an assigned weight.

[0017] With reference to Figure 1 according to one example, based on the topology of network 60, the traffic history (e.g., end-to-end traffic demands), and an upgrade budget (e.g., a maximum percentage of a total number of legacy switches to be upgraded), deployment planner 50 formulates the deployment of SDN switches as a path-constrained, multi-commodity flow problem with a goal of minimizing maximum link usage (i.e., a ratio of link load to link bandwidth). In one example, such commodity flow problem includes solving for two unknowns, one for selecting the legacy switches to upgrade with SDN switches, and the other selecting paths to balance traffic. While such an approach is capable of determining which legacy switches to replace and for selecting paths for
balancing traffic, such commodity flow problem is an integer-linear programming problem which is NP-complete (meaning that while solutions are possible, there is not an efficient way to find such solutions).

[0018] In one example, rather than employing a path-constrained multi-commodity flow problem approach to determining which legacy switches to replace, deployment planner 50 applies a heuristic-based approach. According to one heuristic, legacy switches having the highest degrees are selected for replacement, as switches having the highest degrees are likely to be traversed by more end-to-end routing paths. According to such heuristic, deployment planner 50 constructs a topology graph of network 60 (e.g., similar to that illustrated in Figure 1) from topology information 52 (including link-state information as illustrated by Figure 2, for example), where each connection or link a legacy switch has to another legacy switch is defined as a “degree.” In one example, if a legacy switch has more than one link with a given legacy switch, each link is considered a degree.

[0019] For instance, referring to the topology graph of network 60 in Figure 1, legacy switch L1 is illustrated as having a degree of 2 while legacy switch L6 has a degree of 4. According to one example, deployment planner 50 selects the legacy switches having the highest degrees for replacement with SDN switches based on an upgrade budget (e.g., a number of switches to be upgraded). For instance, if the upgrade budget is for three legacy switches, deployment planner 50 selects the three legacy switches having the highest degrees for upgrading to SDN switches.

[0020] According to another example, deployment planner 50 employs a different heuristic which is based on link weights (see Figure 2, for example). According to such heuristic, deployment planner 50, using the network topology and link weights provided as topology information at 52, employs a K-shortest path algorithm to determine for each pair of source and destination legacy switches a primary forwarding path and a selected number of alternate or backup forwarding paths. A K-shortest path algorithm is an algorithm which finds a primary forwarding path and a selected number of backup forwarding paths in ascending order of cost or weight between nodes, such as between
packet forwarding elements of a network, including legacy switches L1 to L8. For example, with reference to the topology graph of network 60 of Figure 1 and the example link-state table of Figure 2, a primary forwarding path between legacy switches L3 and L4 is a forwarding path L3-L8-L4 having a weight of 15, and a backup forwarding path is the direct forwarding path L3-L4 having a weight of 20.

[0021] According to one implementation, after determining the primary forwarding path and the selected number of backup forwarding paths for each pair of legacy switches via the K-shortest path algorithm, deployment planner 50 determines the frequency at which each of the legacy switches L1 to L8 appears in all of the primary and backup forwarding paths for each pair of source and destination legacy switches. The legacy switches are then arranged in order of decreasing frequency, with deployment planner 50 selecting the legacy switches having the highest frequencies for replacement with SDN switches based on an upgrade budget (e.g., a number of switches to be upgraded). For instance, if the upgrade budget is for three legacy switches, deployment planner 50 selects the three legacy switches having the highest frequency for upgrading to SDN switches.

[0022] Figure 3 is a block and schematic diagram generally illustrating an SDN controller 80, according to one example of the present disclosure, for operating a hybrid network having both legacy packet forwarding devices and SDN packet forwarding devices, such as hybrid network 90 formed from legacy network 60 of Figure 1 after replacing a number of legacy switches with SDN switches as selected by SDN deployment planner 50. In the illustrated example, legacy switches L6, L7, and L8 of legacy network 60 of Figure 1 have been respectively replaced with SDN switches S1, S2, and S3 to form hybrid network 90, with SDN switches S1 to S3 being in communication with SDN controller 80 as indicated at 82.

[0023] In one example, SDN controller 80 includes a global topology viewer 84, a TE (traffic engineering) module 86, and a failover module 88. According to one example, as will be described in greater detail below, global topology viewer 84 maintains a real-time topology of the hybrid network (including link-states
and link loads, for example) by monitoring interactions between legacy and SDN switches, TE module 86 controls traffic forwarding paths to achieve TE goals (e.g., loading balancing to minimize maximum link load and minimize delay for end-to-end traffic), and failover module 88 alleviates link congestion when link failures occur and enables fast failure recovery.

[0024] A network-wide view of the network topology enables SDN controller 80 to dynamically distribute traffic to meet desired TE goals, and to detect link/switch failures in real time to enable fast failure recovery. While topology information is often available, such as from a network management system, for instance, such topology information does not reflect dynamic changes in real time, such as a link or switch being up or down, for example.

[0025] As described in greater detail below, according to one example, global topology viewer 84 maintains a dynamic and real-time topology of a hybrid network by tracking interactions between legacy switches and SDN switches, such as between legacy switches L1 to L5, and SDN switches S1 to S3 of hybrid network 90.

[0026] According to one example, to determine links between SDN switches, such as SDN switches S1 to S3, SDN controller 80 periodically instructs (e.g., every 5 seconds) the SDN switches to flood discovery messages onto the network (e.g., broadcasting discovery messages on every port). In one example, where SDN switches S1 to S3 employ OpenFlow protocol, SDN controller 80 instructs SDN switches S1 to S3 to flood the network with discovery messages including Link Layer Discovery Protocol (LLDP) messages and Broadcast Domain Discovery Protocol (BDDP) messages. SDN switches receiving LLDP messages forward the received LLDP message to SDN controller 80, while LLDP messages are dropped by legacy switches. In such fashion, global topology viewer 84 determines direct interconnections between SDN switches from LLDP messages forwarded by SDN switches, such as the direction connection between SDN switches S1 and S2 in Figure 3.

[0027] BDDP messages are received and forwarded by legacy switches and, upon receipt by an SDN switch, are forwarded to global topology viewer 84. In such fashion, while unable to determine the exact path traversed by a BDDP
message, global topology viewer 84 determines indirect connections between SDN switches (i.e. connections that traverse legacy switches) based on BDDP messages forwarded from SDN switches, such as the indirect connection between SDN switches S1 and S3.

[0028] According to one example, global topology viewer 84 determines connections between legacy switches based on what is referred herein generally as “routing information messages.” Such routing information messages are sent by legacy switches and include topology information of the source switch which is indicative of links with other switches. In one example where legacy switches employ an Interior Gateway Protocol (IGP), such as Open Shortest Path First (OSPF) protocol, legacy switches periodically flood the network with OSPF link-state advertisements (LSAs), including network and router LSAs, for example, where such router LSAs announce the presence of the router/switch and lists links to other routers/switches of the network, and network LSAs list routers/switches that are joined together by a network segment. According to one example, intermediate SDN switches in the hybrid network, such as SDN switches S1-S3 of hybrid network 90, intercept the LSAs and forward them as Packet-In messages to SDN controller 80 (where Packet-In messages are employed by OpenFlow protocol for forwarding “captured” messages). In one example, global topology viewer 84 parses the LSAs received by SDN controller 80 to determine links between legacy devices.

[0029] In one example, such routing information messages may include Border Gateway Protocol (BGP) messages forwarded to global topology viewer 84 by SDN switches receiving such messages from legacy switches. In one instance, SDN controller 80 may direct SDN switches to carry on BGP sessions with routers, where received BGP route updates from legacy switches are forwarded by the SDN switch to global topology viewer 84. Other techniques may also be employed, such SNMP4SDN ODL, for example, where legacy switches are configured by CLI to send SNMP trap to a plug-in in the controller when the switch boots up, and the plug-in also queries LLDP data on legacy switches for topology discovery.
[0030] In one example, to determine links between legacy switches and SDN switches, SDN switches forward to global topology viewer 84 what are referred to herein as “neighbor relationship messages”, such as OSPF and IS-IS protocol Hello messages, and Border Gateway Protocol (BGP) “KeepAlive” messages. In one implementation, where legacy switches employ an IGP, legacy switches (such as legacy switches L1 to L5) periodically (e.g., every 5 seconds) send “Hello” messages (such as OSPF Hello messages, for example) to establish and confirm network relationships with adjacent devices. Similar to that described above, upon receiving a Hello message, SDN switches (such as SDN switches S1-S3) forward the received Hello messages as Packet-In messages to an SDN controller, such as SDN controller 80. From the Hello messages received by SDN controller 80, global topology viewer 84 determines links between legacy switches and SDN switches, such as between SDN switch S1 and legacy switch L2 in Figure 3.

[0031] By receiving discovery, routing information, and neighbor relationship messages, as indicated at 84a, global topology viewer 84 determines connections between SDN switches (e.g., based on LLDP and BDDP messages), determines connections between legacy switches (e.g., based on LSA messages), and determines connections between legacy switches and SDN switches based on neighbor relationship messages (e.g., based on Hello and KeepAlive messages), and thereby determines and maintains a centralized global network topology of the hybrid network which reflects topology changes in real time, as indicated at 84b. For example, global topology viewer 84 detects whether there are changes in links or whether links are up/down between legacy switches by detecting differences between previous and current LSAs. Additionally, global topology viewer 84 determines if links between legacy switches and SDN switches are down based on whether a neighbor relationship message (e.g., a Hello message) experiences a TIMEOUT, where a TIMEOUT indicates a link is down.

[0032] It is noted that while specific routing protocols are described herein (e.g., OSPF, BGP, IS-IS, IGP, BGP, etc.), the teachings of the present disclosure may be extended to any suitable routing protocol.
In one example, SDN controller 80 further determines real-time link loads on the network. In one instance, where OpenFlow protocol is employed, SDN controller 80 determines real-time link loads based on the “meter table” feature of OpenFlow protocol (such as OpenFlow 1.3) for measuring per-flow packet rates. Because SDN controller 80, via global topology viewer 84, has global knowledge of forwarding paths for each pair of source and destination nodes (legacy and SDN switches), if a data flow traverses at least one SDN switch, SDN controller 80 can determine packet the packet flow rate for associated with the particular flow on links in the forwarding path by accessing the meter table entries attached to the flow in any SDN switch the flow traverses. For example, with reference to Figure 3, the link load on links between SDN switch S1 and legacy switches L2 and L3 associated with a packet flow between legacy switches L2 and L3 which traverses SDN switch S1 can be determined by accessing the associated entry in the meter table of SDN switch S1.

In one example, for a packet flow between source and destination legacy switches that does not traverse an SDN switch along the forwarding path, such as a packet flow between legacy switches L2-L4-L3 in Figure 3, SDN controller 80 employs an SNMP-based (Simple Network Management Protocol) estimate of bandwidth utilization. In one example, SDN controller 80 periodically polls SDN switches for meter table entries and SNMP states of legacy switches and aggregates the packet flows to determine a combined link load for each link of the hybrid network.

TE module 86, according to one example, is configured to meet desired traffic engineering goals by controlling traffic forwarding paths. In one example, one desired traffic engineering goal is to provide link load balancing to minimize the maximum link utilization. Certain SDN-based traffic engineering techniques are ill-suited for hybrid networks because the forwarding procedures of non-SDN switches cannot be dynamically controlled. In one example, TE module 86 accommodates conventional default routing of non-SDN legacy switches while applying SDN-based forwarding principles to SDN switches in view of the global topology to optimize benefits afforded by SDN in overall operation of the hybrid network.
[0036] In one example, where legacy switches are assumed to employ IGP, for each new packet flow (where a flow is defined), the flow forwarded by legacy switches along the shortest path in compliance with IGP Shortest Path First (SPF) algorithms for forwarding path calculation. With regard to SDN switches, according to one example, TE module installs rules on SDN switches to control forwarding paths based on routing policies and determined real-time link loads (as described above).

[0037] In one example, when a new packet flow reaches an SDN switch, TE module 86 implements several balancing heuristics to forward the flow. One such heuristic, in accordance with the present disclosure, is referred to as “Hybrid-LLF”, where LLF stands for “least loaded first”. According to this heuristic, the SDN switch forwards the packet flow to an output along the least loaded path. As an example, with reference to Figure 3, a flow from legacy switch L1 (source node) destined for legacy switch L4 (destination node) is first forwarded along its shortest path to reach SDN switch S1. SDN switch S1 has two paths to reach legacy switch L4, through legacy switch L2 or through legacy switch L3. According to Hybrid-LLF, SDN switch S1 chooses the path having the smaller maximum link usage based on real-time link usage (as described above). Assuming the path through legacy switch L3 has a smaller link load at the moment, TE module 86 installs forwarding rules on SDN switch S1 to forward the packet flow to the output port associated with legacy switch L3.

[0038] Another heuristic, in accordance with the present disclosure, is referred to as “Hybrid-Weighted”, which splits flows to go through multiple paths with different possibilities by using the “select group table” feature of OpenFlow 1.1. Using the same example as used to illustrate Hybrid-LLF, according Hybrid-Weighted, SDN switch S1 splits flows to legacy switch L4 along two forwarding paths, S1-L2-L4 or S1-L3-L4. In one example, Hybrid-Weighted assigns weights to each path which are inversely proportional to the maximum link usage (e.g. with weight 0.4, S1 forwards to L2; and with weight 0.6, S1 forwards to L3).

[0039] It is noted that if a packet flow does not reach an SDN switch, conventional IGP-based forwarding is applied. It is also noted that linear
programming can be used to determine the optimal set of forwarding paths and their splitting ratio across SDN nodes, but such approach is not well-suited for real-time dynamic load balancing.

[0040] Failover module 88, according to one example, is configured to alleviate congestion when failures occur and to provide fast failure recovery. In one example, failover module 88 pre-computes and configures backup routing paths in the case of single-link (non-partition) failure for each pair of source-destination switches of network 90. Upon detection of link and/or switch failures by global topology viewer 84, TE module 86 directs SDN switches to direct affected flows to different output ports to the predetermined routes to avoid failed links, reroutes high-priority flows to avoid congested links and, in one example, adjusts weights of group table entries of SDN switches to rebalance traffic, reduce congestion, and reduce packet loss during failure recovery.

[0041] By employing an SDN planner to determine strategic replacement of selected legacy switches with SDN switches in a hybrid network environment, and by maintaining a real-time global topology view of the hybrid network (i.e. both SDN and legacy switches), monitoring real-time traffic of the hybrid network, and controlling SDN switches to take advantage of the fine-grained and flexible packet forwarding afforded by SDN in view of the hybrid network global topology to provide hybrid traffic management, an SDN controller in accordance with the present disclosure optimizes operation of the hybrid network for the particular combination of SDN and legacy switches. In one case, based on actual ISP and enterprise network topologies, employing an SDN planner and SDN controller, in accordance with the present disclosure, where 20% of legacy devices were upgraded to SDN devices, maximum link usage was reduced by an average of 32% compared with pure-legacy networks (using shortest path routing), while requiring an average of only 41% of flow table capacity compared with pure-SDN networks.

[0042] Figure 4 is a flow diagram illustrating a method 100 of incrementally converting a legacy network to a hybrid network, according to one example. At 102, topology information is received, such as topology data representative of interconnections in the form of links between legacy switches (including
characteristics of the links such as the type of links, weights associated with each link, etc.), and historical traffic patterns/demands on links, for example. At 104, the interconnection characteristics of links between the legacy switches are evaluated. According to one example, the interconnection characteristics are evaluated by determining a number of links to each legacy switch of the legacy network. According to one example, the interconnection characteristics are evaluated by determining a primary routing path and a selected number of alternate routing paths between each pair of legacy switches based on a weight of each link of the routing path.

[0043] At 106, a selected number of legacy switches are replaced with SDN switches based on the interconnection evaluation. In one example, legacy switches having the greatest number of links thereto are selected for replacement with SDN switches. In one example, legacy switches appearing the greatest number of times in the primary and alternate routing paths determined at 104 are selected for replacement with SDN switches.

[0044] At 108, discovery messages sent by SDN devices, routing information messages (e.g., OSPF LSA, and BGP/IS-IS routing protocol messages), and neighbor relationship messages periodically sent by remaining legacy switches are received. In one example, SDN switches provide discovery messages received from other SDN switches. In one example the discovery messages comprise LLDP messages. In one example, the discovery messages comprises BDP messages. According to one example, SDN switches provide intercepted IGP messages in the form of Packet-In messages, the IGP messages sent periodically by legacy switches and including internal routing topologies of the associated legacy switch. In one instance, the IGP messages comprise OSPF link-state advertisements (LSAs). In another example, BGP/IS-IS routing protocol packets are received and forwarded by SDN switches.

[0045] At 110, a global topology of the hybrid network defining links between legacy and SDN switches is determined from the received messages at 108. In one example, direct links between SDN switches is determined from the discovery messages. In one instance, direct links between legacy switches are determined from the IGP messages (e.g., OSPF LSAs). In one case, direct
links between legacy switches and SDN switches are determined from the neighbor relationship messages (e.g., Hello messages). According to one example, determining a network topology at 110 further includes determining real-time link loads of the hybrid network. In one example, link-loads of packet flows traversing SDN switches are determined based on OpenFlow meter tables maintained by each of the SDN switches. In one example, link loads associated with packet flows traversing only legacy switches are determined using SNMP-based bandwidth utilization estimates.

[0046] In one example, SDN planner 50 and SDN controller 80, including global topology viewer 84, TE module 86, and failover module 88 may be implemented by a computing system. In such examples, each of SDN planner 50 and SDN controller 80 of the computing system may include any combination of hardware and programming to implement the functionalities of SDN planner 50 and SDN controller 80, including global topology viewer 84, TE module 86, and failover module 88, as described herein in relation to any of FIGS. 1-4. For example, programming for SDN planner 50 and SDN controller 80, including global topology viewer 84, TE module 86, and failover module 88, may be implemented as processor executable instructions stored on at least one non-transitory machine-readable storage medium and hardware may include at least one processing resource to execute the instructions. According to such examples, the at least one non-transitory machine-readable storage medium stores instructions that, when executed by the at least one processing resource, implement SDN planner 50 and SDN controller 80, including global topology viewer 84, TE module 86, and failover module 88. In one example, as indicated by Figures 1 and 3, SDN planner deployment planner 50 may be implemented and stored as processor executable instructions separately from those of SDN controller 80.

[0047] Figure 5 is a block and schematic diagram generally illustrating a computing system 200 for implementing secure software system 100 according to one example. In the illustrated example, computing system or computing device 200 includes processing units 202 and system memory 204, where system memory 204 may be volatile (e.g. RAM), non-volatile (e.g. ROM, flash
memory, etc.), or some combination thereof. Computing device 200 may also have additional features/functionality and additional or different hardware. For example, computing device 200 may include input devices 210 (e.g. keyboard, mouse, etc.), output devices 212 (e.g. display), and communication connections 214 that allow computing device 10 to communicate with other computers/applications 216, wherein the various elements of computing device 200 are communicatively coupled together via communication links 218.

[0048] In one example, computing device 200 may include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape. Such additional storage is illustrated in Figure 5 as removable storage 206 and non-removable storage 208. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any suitable method or technology for non-transitory storage of information such as computer readable instructions, data structures, program modules, or other data, and does not include transitory storage media.

Computer storage media includes RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, and magnetic disc storage or other magnetic storage devices, for example.

[0049] System memory 204, removable storage 206, and non-removable storage 208 represent examples of computer storage media, including non-transitory computer readable storage media, storing computer executable instructions that when executed by one or more processors units of processing units 202 causes the one or more processors to perform the functionality of a system, such as secure software system 100. For example, as illustrated by Figure 5, system memory 204 stores computer executable instructions 250 for SDN deployment planner 50, and computer executable instructions 280 for SDN controller 80, including topology viewer instructions 284, TE module instructions 286, and failover module instructions 288, that when executed by one or more processing units of processing units 202 implement the functionalities of SDN deployment planner 50 and SDN controller 80 as described herein. In one example, one or more of the at least one machine-readable medium storing
instructions for at least one of SDN deployment planner 50, SDN controller 80, global topology viewer 84, TE module 86, and failover module 88 may be separate from but accessible to computing device 200. In other examples, hardware and programming may be divided among multiple computing devices.

[0050] In some examples, the computer executable instructions can be part of an installation package that, when installed, can be executed by at least one processing unit to implement the functionality of at least one of SDN deployment planner 50, SDN controller 80, global topology viewer 84, TE module 86, and failover module 88. In such examples, the machine-readable storage medium may be a portable medium, such as a CD, DVD, or flash drive, for example, or a memory maintained by a server from which the installation package can be downloaded and installed. In other examples, the computer executable instructions may be part of an application, applications, or component already installed on computing device 200, including the processing resource. In such examples, the machine readable storage medium may include memory such as a hard drive, solid state drive, or the like. In other examples, the functionalities of at least one of SDN deployment planner 50, SDN controller 80, global topology viewer 84, TE module 86, and failover module 88 may be implemented in the form of electronic circuitry.

[0051] In Figure 6, as described above, in one example, the functionalities of SDN deployment planner 50 may be implemented as processor executable instructions stored on a non-transitory computer-readable medium, such as computer-readable medium 300. In one example, computer executable instructions 350 for SDN deployment planner 50 are stored on computer-readable medium 300 including instructions to receive network topology data 352, to evaluate interconnection characteristics between switches 354, and to select legacy switches for replacement with SDN switches based on the evaluated interconnection characteristics 356.

[0052] Although specific examples have been illustrated and described herein, a variety of alternate and/or equivalent implementations may be substituted for the specific examples shown and described without departing from the scope of the present disclosure. This application is intended to cover any adaptations or
variations of the specific examples discussed herein. Therefore, it is intended that this disclosure be limited only by the claims and the equivalents thereof.
CLAIMS

1. An SDN controller for a hybrid network including legacy switches and SDN switches, the SDN controller comprising:
   a topology viewer to:
   receive from the SDN switches discovery messages sent by other SDN switches as directed by the SDN controller;
   receive from the SDN switches routing information messages including internal routing topologies sent by the legacy switches;
   receive from the SDN switches intercepted neighbor relationship messages sent by the legacy switches; and
   determine a global topology of the hybrid network by determining direct links between SDN switches based on the discovery messages, by determining links between legacy switches based on the interior gateway protocol message, and by determining links between SDN switches and legacy switches based on the neighbor relationship messages.

2. The SDN controller of claim 1, the discovery messages comprising Link Layer Discovery Protocol (LLDP) messages and Broadcast Domain Discovery Protocol (BDDP) messages.

3. The SDN controller of claim 1, the routing information messages comprising interior gateway protocol (IGP) messages and border gateway protocol (BGP) messages, the IGP messages comprising Open Shortest Path First (OSPF) link state advertisements (LSAs).

4. The SDN controller of claim 1, the neighbor relationship messages comprising IGP Hello messages and Border Gateway Protocol (BGP) KeepAlive messages.
5. The SDN controller of claim 1, further including a traffic engineering module, the traffic engineering module to:
   determine real-time link loads for each link of the hybrid network; and
   provide rules to each SDN switch to control packet forwarding based on routing policies and real-time link loads so as to balance loads between links.

6. The SDN controller of claim 5, the determination of real-time link loads comprising determining link-loads of packet flows traversing SDN switches based on OpenFlow meter tables, and determining link loads of packet flows traversing only legacy switches using a Simple Network Management Protocol (SNMP)-based bandwidth utilization estimate.

7. The SDN controller of claim 5, the traffic engineering module to instruct SDN switches to route a packet flow via a least loaded path, based on the real-time link loads, when more than one path exists between the SDN switch and a destination to which the packet flow is being routed.

8. A non-transitory computer-readable storage medium comprising computer-executable instructions, executable by at least one processor to:
   implement an SDN deployment planner to:
   receive data representative of a topology of a network including legacy switches, the legacy switches interconnected by links, the data including interconnection characteristics;
   evaluate the interconnection characteristics between the legacy switches; and
   determine a selected number of legacy switches to replace with SDN switches based on the evaluated interconnection characteristics, the selected number being less than a total number of legacy switches in the network.
9. The non-transitory computer-readable storage medium of claim 8, further including instructions executable by the at least one processor to implement the SDN deployment planner to:
   evaluate interconnection characteristics by determining a number of links to each legacy switch of the network; and
   select legacy switches having the greatest number of links connected thereto as legacy switches to replace with SDN switches.

10. The non-transitory computer-readable storage medium of claim 8, further including instructions executable by the at least one processor to implement the SDN deployment planner:
    evaluate interconnection characteristics by determining a primary routing path and a selected number of alternate routing paths between each pair of legacy switches; and
    select legacy switches appearing the greatest number of times in the primary and alternate routing paths as legacy switches to replace with SDN switches.

11. The non-transitory computer-readable storage medium of claim 10, the received data representative of the network including a weight assigned to each link between the legacy switcher, the instructions executable by the at least one processor to implement the SDN deployment planner to:
    determine the primary routing path and alternate routing paths between each pair of legacy switches based on lowest total link weights of routing paths between each pair of legacy switches.

12. A method of transitioning a legacy switch network to an SDN network, the method including:
    receiving data representative of a topology of the legacy switch network including data representative of links and interconnection characteristics between legacy switches;
evaluating the interconnection characteristics between the legacy switches; and
replacing a selected number of legacy switches with SDN switches based on the evaluated interconnection characteristics.

13. The method of claim 12, including:
receiving from the SDN switches:
discovery messages sent by other SDN switches;
routing information messages comprising intercepted interior gateway protocol (IGP) discovery messages and border gateway protocol (BGP) messages including internal routing topologies sent by legacy switches; and
neighbor relationship messages sent by the legacy switches; and
determining a global topology of a hybrid network comprising the SDN switches and the remaining legacy switches by determining:
links between SDN switches based on the discovery messages;
links between legacy switches based on the internal routing topologies from the routing information messages; and
links between SDN switches and legacy switches based on the neighbor relationship messages.

14. The method of claim 12, including:
determining real-time link loads for each link of the hybrid network; and balancing loads between links by providing rules to each SDN switch to control packet forwarding based on real-time link loads and routing policies.

15. The method of claim 12, where determining real-time link loads includes;
determining real-time link loads of packet flows traversing only legacy switches using Simple Network Management Protocol (SNMP) bandwidth utilization estimates;
determining real-time link loads of packet flows traversing SDN switches using Simple Network Management Protocol (SNMP) based bandwidth utilization estimates; and routing packet flows traversing SDN switches to a least loaded path based on the real-time link loads.
<table>
<thead>
<tr>
<th>LINK - STATE</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3 – L4</td>
<td>20</td>
</tr>
<tr>
<td>L3 – L5</td>
<td>15</td>
</tr>
<tr>
<td>L3 – L6</td>
<td>10</td>
</tr>
<tr>
<td>L3 – L8</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LINK - STATE</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4 – L2</td>
<td>15</td>
</tr>
<tr>
<td>L4 – L3</td>
<td>20</td>
</tr>
<tr>
<td>L4 – L8</td>
<td>10</td>
</tr>
</tbody>
</table>
Fig. 3
RECEIVE DATA REPRESENTATIVE OF TOPOLOGY OF LEGACY NETWORK INCLUDING INTERCONNECTION CHARACTERISTICS

EVALUATE INTERCONNECTION CHARACTERISTICS BETWEEN LEGACY SWITCHES

REPLACE SELECTED LEGACY SWITCHES BASED ON EVALUATED INTERCONNECTION CHARACTERISTICS
Fig. 6
A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/931(2013.01)i, H04L 12/751(2013.01)i, H04L 12/24(2006.01)i, H04L 29/06(2006.01)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 12/931; H04L 12/703; H04L 12/46; H04L 12/24; H04L 12/26; H04L 12/715; H04L 12/721; H04L 12/73; H04L 12/56; H04L 12/751; H04L 29/06

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: SDN, legacy, hybrid, switch, topology, discovery, routing, link

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 2014-0146664 A1 (LEVEL 3 COMMUNICATIONS, LLC) 29 May 2014 See paragraphs [0015]-[0034]; and figures 1, 2A.</td>
<td>1-15</td>
</tr>
<tr>
<td>Y</td>
<td>US 2015-0043382 A1 (NEC LABORATORIES AMERICA, INC.) 12 February 2015 See paragraphs [0017]-[0021]; figure 2; and claim 1.</td>
<td>1-15</td>
</tr>
<tr>
<td>Y</td>
<td>WO 2015-167597 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 05 November 2015 See paragraphs [0013]-[0019]; and figure 1.</td>
<td>5-7, 14, 15</td>
</tr>
<tr>
<td>A</td>
<td>US 2013-0260007 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 10 October 2013 See paragraphs [0071]-[0083]; and figure 5.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>EP 2787698 A2 (DEUTSCHE TELEKOM AG et al.) 08 October 2014 See paragraphs [0066]-[0095].</td>
<td>1-15</td>
</tr>
</tbody>
</table>

☐ 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.

"V" 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.

"X" 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.

"S" document member of the same patent family.

Date of the actual completion of the international search
01 November 2016 (01.11.2016)

Date of mailing of the international search report
08 November 2016 (08.11.2016)

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongna-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer
KIM, Seong Woo

Telephone No. +82-42-481-3348

Form PCT/ISA/210 (second sheet) (January 2015)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2014-0146664 A1</td>
<td>29/05/2014</td>
<td>CA 2892362 A1</td>
<td>30/05/2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2923462 A1</td>
<td>30/09/2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2923462 A4</td>
<td>13/04/2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2014-082056 A1</td>
<td>30/05/2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2015-021248 A1</td>
<td>12/02/2015</td>
</tr>
<tr>
<td>WO 2015-167597 A1</td>
<td>05/11/2015</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 9225635 B2</td>
<td>29/12/2015</td>
</tr>
</tbody>
</table>