IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 4, 2019, accepted December 22, 2019, date of publication January 1, 2020, date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963475

Centralized Trust-Based In-Band Control for SDN Control Channel

WENTAO FAN [1,2,3] AND FAN YANG [1,2,3,4]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]Peng Cheng Laboratory, Shenzhen 518055, China
[4]Purple Mountain Laboratories, Nanjing 211111, China

Corresponding author: Fan Yang (yfan@bupt.edu.cn)

This work was supported in part by the project of Chinese Academy of Engineering Research on the Development Strategy of Future Network Operating System under Grant 2019-XY-5, in part by the Project of State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China, Research on Network Protocol Universal Forwarding Technology under Grant NST20190301, and in part by the Project of Chinese National Key Research and Development Plan, Research on Next-Generation Network Processor Architecture and Key Technologies under Grant 2018YFB1800602.

ABSTRACT In SDN architecture, the control channel between the control and data planes is extremely significant, which has two connection modes: out-of-band mode and in-band mode. For in-band mode, the key to the establishment of SDN control channel is how to forward in-band control traffic in the data plane (control traffic: traffic between controllers and switches). In almost all the existing implementation of the in-band SDN networks, the forwarding of in-band control traffic in the data plane depends on distributed Layer 2 switching that cannot respond to the network state change immediately. In this article, we propose a Centralized Trust-based In-Band (CTIB) control mechanism for SDN in-band control channel. Then based on CTIB control mechanism, we design CTIB routing algorithm to select the most trusted control path for each data plane switch in order to improve the QoS (Quality of Service) of in-band control traffic. Furthermore, we implement CTIB control mechanism in an in-band SDN network and carry out several experiments to test the QoS of the in-band control traffic. The results indicate that there is a huge improvement of in-band control traffic QoS when using CTIB control mechanism compared to the distributed Layer 2 switching.

INDEX TERMS SDN, centralized in-band controlling, QoS of in-band control traffic, trust-based routing.

I. INTRODUCTION

The concept of SDN (Software Defined Networking) was firstly proposed in 2008 [1]. With continuous research, SDN architecture becomes more and more complete and diversified. Until now, there have been plenty of industrialized implementations of SDN, such as SDN Data Center [2] and SD-WAN [3].

Compared to the traditional network, SDN has a simpler architecture. The logically centralized control plane utilizes the southern protocol to collect the information of the data plane and to manage the behavior of the data plane. In this interaction mechanism, communication between the control and data planes plays a vital role. The connection between control and data planes has two modes: out-of-band and in-band. Out-of-band means that every switch in the data plane connects to a controller through a direct link. This separated control network can keep the control delay at a very low level and guarantee the QoS (Quality of Service) of control traffic. In the field of academic research, the implementation of SDN always mainly focuses on the out-of-band mode. However, in some scenarios, the out-of-band mode has limitations as well. When the scale of the switch cluster keeps growing, the cost of the separated control link will increase rapidly. And the robustness of the out-of-band mode is relatively weaker. If one direct link is failed, the switch on its end will lose contact with the controller immediately. Therefore, in many industrialized scenarios, the in-band mode is a better choice for SDN implementation.

Differing from a separated control network in an out-of-band network, when using in-band mode, control traffic needs to share the data plane physical devices with data traffic

The associate editor coordinating the review of this manuscript and approving it for publication was Nizar Zorba.

(traffic between hosts and hosts). In the in-band SDN network, only one switch connects to a controller directly. Other switches need to build virtual connections with a controller through the data plane. Using in-band mode can save costs and enhance stability. As long as a switch can communicate with the controller through the data plane, the switch will maintain the communication with the controller.

In commonly-used southern protocol [4], [5], there is no precise definition about how to establish an in-band connection between control and data planes. The key to in-band connection is how to forward control traffic in the data plane. In existing implementations, the forwarding of in-band control traffic in the data plane depends on Layer 2 switching and learning operations in every data plane switch [6]. Layer 2 switching is a periodic refreshing mechanism and learns the mapping of MAC (Media Access Control) addresses of received packets and their in-ports. However, Layer 2 switching is a distributed routing strategy and cannot respond to the changes of network state and take full advantage of data plane link resource [7], which may cause two unavoidable issues: 1) The forwarding of control traffic is vulnerable when link congestion occurs in the data plane. 2) The controller is unable to manage the in-band control traffic when using the Layer 2 switching.

By summarizing previous studies and implementations, we deem that the biggest challenge of the SDN in-band network is how to promote the QoS of in-band control traffic. To overcome this challenge, researchers have proposed several solutions, such as, setting high priority queue for control traffic [8], adding the number of in-band connecting points [9] and adopting MPTCP to in-band control traffic [10], [11]. All these researches make great contributions to improving QoS of in-band control traffic. Nonetheless, these solutions still have not mentioned in detail about how to deal with the forwarding of in-band control traffic in the data plane. Existing implementations of in-band mode still use distributed Layer 2 switching to forward control traffic, like the in-band module in OpenVSwitch [12].

In this paper, we propose Centralized Trust-based In-Band (CTIB) control mechanism for SDN control channel and design CTIB routing algorithm to improve the QoS of the in-band control traffic. We define this Centralized Trust-based In-Band control as: Control plane collects the data plane information; Based on the information, control plane makes full use of data plane link resources to select the most trusted control path (control path: the connection path between a controller and a switch) for each data plane switch according to the CTIB routing algorithm. The main objective of our research is to enable the control plane to manage the in-band control traffic and improve the QoS of in-band control traffic.

Therefore, we need to find the most trusted path as the control path. A path consists of a series of links. To measure the path trust level, we define the Link Trust Level (LTL) as the remaining bandwidth of the link. By analyzing the control traffic, we find that the data size of control traffic is very small, and the local large bandwidth in the path has no meaning for in-band control traffic. Based on this feature, we define the Path Trust Level (PTL) as the link trust level of the least trusted link in the path.

Based on the definitions of Link Trust Level and Path Trust Level, we propose the Universal version of Centralized Trust-based In-Band (U-CTIB) routing algorithm to select the most trusted control path. Then aiming two representative network topology, Data Center (DC) network and User Network (UN), we improve U-CTIB to optimize the path selection and increase computational efficiency. DC network topology is commonly layered and well-organized [13]–[15]. According to this, we add a constraint of hops in path catching to increase efficiency and design DC-CTIB routing algorithm. The user network has higher randomness and connectivity. We use the nature of the maximum spanning tree to get the most trusted control path and design UN-CTIB. By simulation, DC-CTIB and UN-CTIB algorithm has much better computational efficiency in the DC and user network topology respectively compared to the Layer 2 switching.

Furthermore, we implement CTIB control mechanism in an in-band SDN network and compare the QoS of in-band control traffic when using CTIB control and distributed Layer 2 switching. We carry out several experiments on DC fat-tree topology and user network topologies. The results indicate that there is a huge improvement of the QoS of in-band control traffic when using CTIB control compared to distributed Layer 2 switching.

The rest of the paper is organized as follows. The following section presents the research backgrounds and states. Section III and IV describe the CTIB control mechanism and CTIB routing algorithm. Section V contains the implementation, experiments and results analysis. The final section is the conclusion of the paper.

II. BACKGROUND AND RELATED WORK

This section is a presentation of research backgrounds and states. The first part is an overview of the SDN architecture and the two connection modes between control and data planes. Then, several previous studies of in-band mode are demonstrated, including some performance improvements for in-band mode. Lastly, we introduce the information collection and rule installation process in an SDN network.

A. SDN OUT-OF-BAND MODE AND IN-BAND MODE

SDN architecture consists of three parts: application, control, and data planes [1]. The application plane is the interface for users to design strategies for data forwarding. The control plane, the direct manager of the data plane, contains many preconfigured and customized functionalities. Application and control planes are connected by REST API and always integrated into a controller device. The data plane refers to a switch network without processing capacity. The control plane implements the forwarding command in the data plane by means of Southern Protocol. The establishment of the communication path between control and data planes has two implementation modes: out-of-band and in-band.

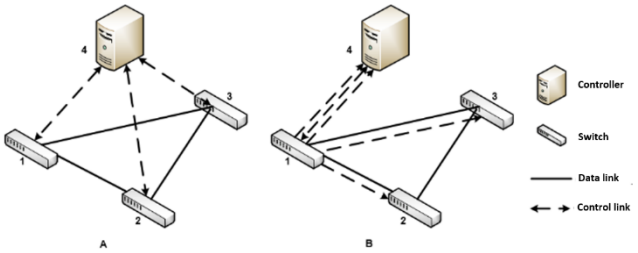

FIGURE 1. There are two connection modes between control and data plane: out-of-band mode (A) and in-band mode (B).

In out-of-band mode, every switch in the data plane connects to a controller through a direct control link (Fig.1 A). All these one-hop control paths constitute a dedicated control network. The advantage of out-of-band is that the direct control link will never suffer from congestions, i.e. control delay is very low. However, the limitation is that the cost of control links will keep growing with the increase of the network scale.

In in-band mode, not every switch directly connects to a controller in a network domain. As shown in Fig.1 B, switch1 is the connecting point, which is connected to the controller directly. Other switches need to communicate with the controller by the relaying of the connection node. Apparently, in-band mode saves a lot of costs, but the consequence is that control traffic needs to share the data links with the data traffic.

In existing researches, switches forward the in-band control traffic using Layer 2 switching. When link congestions or failures occur, control traffic will be influenced as well, which means the control delay is largely determined by the data plane link state.

B. IMPROVEMENTS FOR SDN IN-BAND MODE

1) MULTIPLE CONNECTION NODE FOR IN-BAND MODE

Cheng Suo et al. proposed a new economical and reconfigurable hybrid-band control mechanism [9]. In hybrid-band, there is more than one connection node in the data plane. Comparing to only one connection node, multiple connection nodes will share the control traffic load and lower the control delay. Moreover, in hybrid-band, every switch in the data plane has a backup port for control traffic forwarding in case of link failures. In this paper, we focus on using centralized control for in-band traffic. Therefore we only discuss the traditional in-band scenario with one connection node.

2) MPTCP FOR IN-BAND CONTROL TRAFFIC AND THE SELECTION OF GATEWAY SWITCHES

Ali Raza et al. described an in-band SDN environment using Multipath TCP (MPTCP) to connect the controller and switches [10], [11]. In this research, Multipath TCP (MPTCP) has been introduced for control channels instead of TCP to ensure path availability, better load balancing, and robustness. The MPTCP connection between the controller and switches has multiple subflows. To achieve the availability and reliability of the control channel, the subflows should be disjoint as much as possible and the selection of gate switches is crucial. To select the most suitable gate switch set, they proposed a heuristic algorithm. By means of extensive simulation, they demonstrated that the proposed algorithm performs much better than the Random Search and is comparable to the Exhaustive Search.

3) QUEUING FUNCTIONALITY FOR IN-BAND CONTROL TRAFFIC

Sachin Sharma et al. described a Queuing functionality for in-band control traffic [8]. In this research, they create different queues for control and data traffic. To ensure that the control traffic will be served firstly, the control traffic queue is given the highest priority. In addition, for separating the control and data traffic at the entry of queues, the controller uses the source IP address, destination IP address, and transport layer parameters. This functionality reduces the queuing delay of control traffic when link congestions occur and the experiment result was impressive. Nonetheless, the packet buffer size of a switch might be a restriction to implement queuing functionality. The packet buffer size of frequently-used white box switches is approximately from 12MB to 24MB [16], [17]. When congestions turn up, an extra control queue may cause serious data packet loss.

C. INFORMATION COLLECTION AND RULE INSTALLATION PROCESS IN SDN NETWORK

In this paper, the controller needs to collect the global topology and the port stats from the data plane and install switch rules into the data plane. Here, we introduce the information collection and rule installation process.

1) INFORMATION COLLECTION

The collection of global topology in SDN network is based on Link Layer Discovery Protocol (LLDP) [18]. While a switch *s* builds the TCP connection with a controller, *s* will upload the port information to the controller. Then, the controller sends LLDP packet to *s* through Packet-Out message (packet sent by the controller) and *s* will flood this LLDP packet to every port of itself. Each neighboring switch of *s* receives the LLDP packets and sends back this LLDP packet to the controller through Packet-In message (packet uploaded to the controller). Through this process, the controller can know that every port of *s* connects to which switch. By integrating the connection information of all the switches, the controller can build the network topology.

The collection of port stats depends on the southbound protocol. In the experiment section of our research, we adopt OpenFlow1.3 Protocol as the southbound protocol. The information of switches is uploaded to the controller by the OpenFlow Messages. The controller sends request messages to switches and switches send reply messages containing required parameters back to the controller. We take the port information for instance. The controller sends Ofp_Port_Stats_Request message to switches.

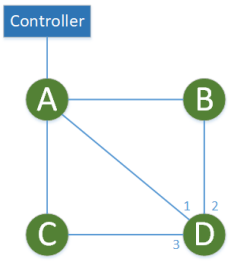

FIGURE 2. This figure shows a simple topology with a controller and four switches and the numbers refer the port numbers of switch D.

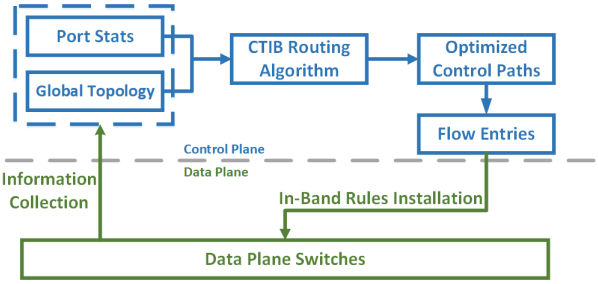

FIGURE 3. Control plane collects global topology and port stats information and uses CTIB routing algorithm to select the optimal control paths for data plane switches.

Switches reply Ofp_Port_Stats messages which contain traffic statistics and timestamps. The traffic statistic comprises the number of received bytes and transmitted bytes. These two collection processes are both periodic and the refresh period can be configured in the controller.

2) RULE INSTALLATION

The controller needs to install the optimized path rule into the data plane. In our experiment, this process depends on the Ofp_Flow_Mod message in OpenFlow1.3 protocol. When a path selection is completed, the controller sends Ofp_Flow_Mod messages to all the switches in this path in order to lead the in-band traffic forwarding through this path. An Ofp_Flow_Mod message contains the match fields and the output port. In our experiment, to identify the control traffic, the match field of the control traffic flow entry is whether the source or destination IP address of the packet is the controller IP. And the output port is the port that connects to the next-hop switch.

III. CENTRALIZED CONTROL FOR IN-BAND CONTROL CHANNELS

A. MOTIVATION

In the SDN network, the control traffic is extremely significant, which is the carrier of communication between the control and data planes. In an in-band SDN network, the control traffic is super vulnerable, since it needs to share data plane link with data traffic. In the case of link congestion caused by data traffic, the delay and the Packet Loss Ratio (PLR) of control traffic will increase apparently.

In existing implementation, Layer 2 switching can not solve this problem. The distributed strategy is unable to take global information of data plane into consideration. Let us suppose a scenario: In the topology shown in Fig.2, switch D receives a control packet from the controller through port 1 and generate a forwarding entry sending its control traffic through port 1 to the controller. When the link from switch D to A is busy, the control traffic of switch D will be delayed. However, as long as switch D can still receive the control packet sent by the controller from port 1, this forwarding entry will not refresh. At the same time, the paths D-B-A and D-C-A with better link states are more suitable for the control traffic from switch D.

What is more, we believe that using distributed routing for in-band traffic violates the basic principles of SDN that separates the control and data planes. The advantage of SDN is that the control plane can use data plane information to optimize the data plane forwarding. However, when using distributed strategies, the control plane is unable to manage the in-band traffic, which is distinctly contradictory to the principle of SDN. Therefore we want to design a centralized control mechanism to fully utilize the resource of data plane to increase the QoS of in-band control traffic.

B. CENTRALIZED TRUST-BASED IN-BAND CONTROL MECHANISM

To improve the QoS of in-band control traffic, we propose this Centralized Trust-based In-Band (CTIB) control mechanism for in-band control traffic. The architecture diagram is shown in Fig.3. This centralized control architecture requires the cooperation of the control plane and the data plane. The starting point of the architecture is the collection of data plane information. The collected information includes the data plane global topology and port stats information of each switch, and the collection process is completed with the help of the southbound protocol. In our research experiments, we use OpenFlow1.3 protocol as the southbound protocol and the specific collection process is described in detail in Section II.C. Furthermore, the information collection process is based on the southbound protocol. Therefore, the traffic required for information collection is a part of control traffic and will not cause additional burdens in the data plane.

By taking the global topology and port stats as input, the control plane uses Centralized Trust-based In-Band (CTIB) traffic routing algorithm to select the optimal control path for every switch. Then the result of the control path selection will be implemented in the data plane and cover Layer 2 switching rules. The Layer 2 switching will not be removed. We keep the self-learning operation operating, so that Layer 2 switching can be a backup routing strategy for in-band control traffic. After the CTIB routing algorithm selects the optimal control path for a target switch, the controller translates the path into a series of flow entries. Each switch on the path except the target switch will install a flow entry, which leads the control traffic sent by the controller

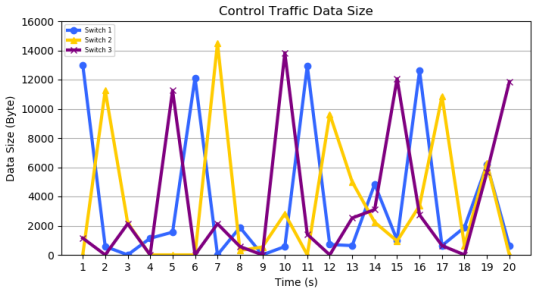

FIGURE 4. This figure shows the statistics of control traffic data size using OpenFlow1.3 Protocol.

to the target switch to the next hop. The target switch will install a flow entry to guide the control traffic from other switches to the next hop of the selected control path. The switch rules installation is completed by the Ofp_Flow_Mod message, as specified in section II.C.

This mechanism requires a periodic refresh. In an SDN network, the information collection process is also periodic. Therefore, we synchronize the cycles of centralized control mechanism, global topology collection and port stats collection. The cycle synchronization ensures that CTIB routing algorithm can get the latest data of a new cycle. In the experiment, we set the refresh cycle to 3 seconds. It should be noted that the calculation of the port rate is based on the port traffic stats in one period. And the global topology is also periodically collected and maintained. Therefore, this centralized mechanism operates based on the information currently acquired by the controller and is not proactive.

IV. CENTRALIZED TRUST-BASED IN-BAND ROUTING ALGORITHM

In this section, we first analyze the feature of control traffic and derive the Universal version of Centralized Trust-based In-Band (U-CTIB) routing algorithm. Then we enhance U-CTIB in two specific scenarios, DC network and User network, to improve computational efficiency. To verify the performance improvement, we conduct simulations in DC and User network topology and test the computation speed.

A. ANALYSIS OF CONTROL TRAFFIC

Our research focuses on optimizing in-band control traffic. Hence, we firstly analyze the feature of control traffic by using Wireshark [19] to collect control traffic in an SDN network. This SDN network uses OpenFlow1.3 protocol as the southbound protocol and the statistics are shown in Fig.4.

We analyze the control traffic between a controller and three switches in 20 seconds. We can find that the data size of control traffic is periodic. According to the line chart, the peak of control traffic comes per 5 seconds and the peak data size is between 10000Byte and 15000Byte, i.e. between 80Kbit and 120Kbit. The peak number 120Kbit is very small compared with common port rates of switches [16], [17].

Through these statistics, we can infer that control traffic does not need high bandwidth. Therefore, the primary

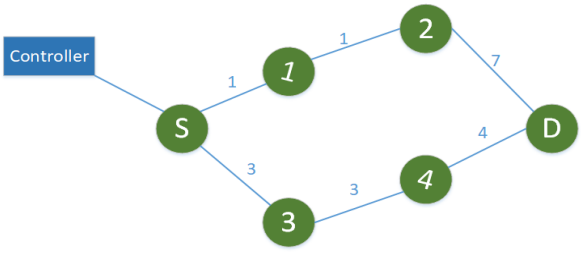

FIGURE 5. This figure shows a simple topology with one controller and six switches and the weights of links refer to the bandwidth occupancies.

demand for QoS of in-band control traffic is to guarantee the connectivity between a controller and a switch. To satisfy this demand, we need the most trusted path as the control path.

B. DEFINITION OF TRUST LEVEL

In order to find the most trusted path between a controller and a switch, we adopt the trust-based routing strategy [20], [21]. In the trust-based routing strategy, we need to measure and define the trust level of a path. A path consists of a series of links, therefore we first define the trust level of a link. For links, the trust level represents the probability of maintaining connected. Here, we use the remaining bandwidth of a link to represent the trust level.

- *Definition (Link Trust Level, LTL(l)):* the remaining bandwidth of the link l.

For control traffic, the required link bandwidth is small. The remaining bandwidth itself is meaningless for in-band control traffic. However, the remaining bandwidth also represents the probability of congestion for a link. The larger the remaining bandwidth, the less the probability that the link will be congested and affect the in-band control traffic transmission in this cycle. Therefore, we use the remaining bandwidth to define the Link Trust Level.

Here, we introduce how the controller calculates the remaining bandwidth of a link. As mentioned above, the control plane collects the port stats using southbound protocol OpenFlow1.3. Then the controller can get the bidirectional port rates by dividing the difference of received/transmitted bytes between two adjacent Ofp_Port_Stats messages by the time interval. Lastly, the controller can calculate the remaining bandwidth of the corresponding link of the port by subtracting the bidirectional port rates from the port maximum rate.

Then, we discuss how to define the trust level of a path. In the research of Sachin Sharma et al., the shortest path has been mentioned for control path establishment [8]. Enlightened by this ideal, we firstly consider to adopt commonly used routing algorithms, like CSP (Constrained Shortest Path), LC (Least Cost) and LD (Least Delay) [22], [23]. These algorithms mostly aim to select an overall optimal path. But we find that an overall optimal path is probably not the most trusted path for in-band control traffic.

We explain it with an instance. In the topology shown in Fig.5, we need to select a path between node S and node D.

The weight of a link refers to the bandwidth occupancy of the link and we assume that the maximum bandwidths of links are the same. When using CSP or LC, the path S-1-2-D will be chosen because the overall cost of path S-1-2-D is less than path S-3-4-D. But from the perspective of in-band control traffic, the link 2-D in path S-1-2-D has the biggest possibility to have congestion and the bandwidth of path S-3-4-D is good enough for in-band control traffic. Therefore path S-3-4-D is more suitable for in-band traffic.

This instance indicates that an overall optimal path is probably not the most suitable path for in-band traffic. Moreover, the trust level of a path could be destroyed by one bad link. Therefore, we deem that the trust level of a path is determined by the worst link of it. We use the LTL of the weakest link in a path to define the trust level of a path:

- *Definition (Path Trust Level, PTL(p)):* the trust level of the least trusted link in the path p.

Based on this definition, the optimal control path between a controller and a switch is the path with the highest PTL between these two nodes.

C. CENTRALIZED TRUST-BASED ROUTING ALGORITHM FOR IN-BAND CONTROL TRAFFIC

Based on the definitions of Link Trust Level and Path Trust Level, we propose the Universal version of Centralized Trust-based In-Band (U-CTIB) routing algorithm, to improve the QoS of in-band control traffic. Before the description of the algorithm, we first define some notations and functions.

- Let $G = (V, E)$ be the graph of vertices, V, and edges, E.
- Let c be the connection node, $c \in V$.
- Let $S\{s_1, s_2, \ldots, s_n\} = V - \{c\}$ be the set of all the switches.
- Let $P(c, s_i)$ be the set of all paths between c and s_i.
- Let $L(p_j)$ be the set of all links in p_j.
- Let $cp(c, s_i)$ be the control path between c and s_i.

Algorithm 1 U-CTIB Routing Algorithm

Input: $G(V, E), c, \{LTL(l), l \in E\}$
Output: control paths,$\{cp(c, s_1), cp(c, s_2), \cdots, cp(c, s_n)\}$

$S \leftarrow V - \{c\}$,the set of normal switches
for $s_i \in S$ **do**
  $P(c, s_i) \leftarrow$ the set of all paths between c, s_i
  for $p_j \in P(c, s_i)$ **do**
    $L(p_j) \leftarrow$ the set of all links in p_j
    $PTL(p_j) \leftarrow \min_k(LTL(l_k)),(l_k \in L(p_j))$
  $cp(c, s_i) \leftarrow \operatorname{argmax}_{p_j}(PTL(p_j)),(p_j \in P(c, s_i))$
return $\{cp(c, s_1), cp(c, s_2), \cdots, cp(c, s_n)\}$

Now we present U-CTIB shown in Algo. 1. We take the topology graph, G, connection node, c and the Link Trust Level of every link in G as input. S is the set of normal switches. For every switch $s_i \in S$, we first get the set of all paths between c and s_i, $P(c, s_i)$. Then we need to calculate the trust level of every path $p_j \in P(c, s_i)$. According to the definition before, we evaluate the trust level of paths by formula 1:

$$PTL(p_j) = \min_k(LTL(l_k)), (l_k \in L(p_j)) \tag{1}$$

After getting the trust level of every path, we choose the path with highest trust level as the control path between c and s_i, $cp(c, s_i)$, using formula 2:

$$cp(c, s_i) = \arg\max_{p_j}(PTL(p_j)), (p_j \in P(c, s_i)) \tag{2}$$

When the traverse of S completes, control plane gets optimal control path for every switch.

This is the Universal version of CTIB. Through simulations, we find that the computation efficiency of U-CTIB is very low. When the scale and the complexity of topology increase, the computation time of U-CTIB grows fast. Specific data of computation time will be presented in the simulation part of this section. To improve the performance of CTIB and make it more practical, we optimize CTIB in two common network scenarios: DC network and random user network.

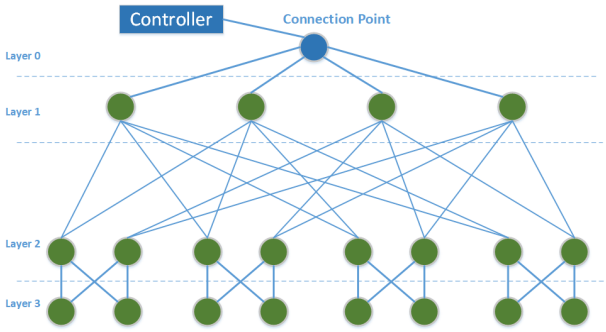

FIGURE 6. This DC fat-tree network connects to a controller using in-band mode. We divide the whole topology into four layers and the topmost Layer 0 is the access layer.

D. DC-CTIB: CTIB FOR DC NETWORK

DC network is commonly well-organized and layered functional. According to this feature, we can simplify CTIB in DC topology by adding constraints. We stipulate that, based on the hierarchical structure of the DC topology, a zeroth layer is added above the original topmost layer, as the access layer which contains connection node and controller. The original topmost layer is now the first layer and the layer number is incremented downward. For example, in a fat-tree topology (k = 4) shown in Fig.6, the network consists of four layers. When selecting control paths for switches in layer n, we add a constraint to the initial set of paths, $P(c, s_i)$. We replace $P(c, s_i)$ with $P(c, s_i, n)$ as follow:

$$P(c, s_i, n) = \{p \mid length(p) \leq n, p \in path(c, s_i)\} \tag{3}$$

We call this new Data Center Centralized Trust-based In-Band routing algorithm as DC-CTIB routing algorithm.

In the data center scenario, the number of layers is small, but the number of switches in each layer is very large. By using $P(c, s_i, n)$, the selection of control paths is more in line with the layered structure of the topology. Control traffic will be transmitted through layers orderly and unidirectionally, rather than back and forth between layers. Moreover, this constraint improves computational efficiency and the time of path searching is greatly reduced.

E. UN-CTIB: CTIB FOR USER NETWORK

Differing from the DC network, user network topology has higher connectivity and randomness. When the complexity of topology increases, getting paths between two nodes will cost more computational resources. Because of the randomness of the user network, it is hard to constrain the accurate length of catching paths. To solve this challenge, we use the nature of Maximum Spanning Tree (MST). Firstly, we describe this theorem:

Theorem 1: The path between c and s_i in MST is the most trusted path between c and s_i in the whole topology graph.

Proof: See Appendix. □

Algorithm 2 UN-CTIB

Input: $G(V, E), c, \{LTL(l), l \in E\}$

Output: control paths,$\{cp(c, s_1), cp(c, s_2), \cdots, cp(c, s_n)\}$

1: $S \leftarrow V - \{c\}$,the set of normal switches
2: $T \leftarrow$ the MST of G // using $\{LTL(l), l \in E\}$ as weight number
3: **for** $s_i \in S$ **do**
4: Traverse T starting with c to find $p_T(c, s_i)$
5: $cp(c, s_i) \leftarrow p_T(c, s_i)$
6: **return** $\{cp(c, s_1), cp(c, s_2), \cdots, cp(c, s_n)\}$

Based on Theorem 1, we present User Network Centralized Trust-based In-Band (UN-CTIB) routing algorithm shown in Algo. 2. The input still consists of topology graph G, connection node c and the Link Trust Level of every link in G. S is the set of normal switches. Firstly, we calculate the Maximum Spanning Tree of G, T, using the Link Trust Level as the weight number of every link. Then by traversing T starting with connection node c, we can obtain the most trusted control path between the connection node and every normal switch.

To demonstrate the algorithm more clearly, we use a sample topology to explain. In Fig.7, there are seven switches in the data plane, including connection node c and normal switches. The numbers besides links are Link Trust Level, as the weight number of edges. Firstly, we calculate the Maximum Spanning Tree of this topology graph. The red links in Fig.7 are the edges of MST. Based on this MST, we can get the control paths for s_1 to s_6: (c, s_2, s_1), (c, s_2), (c, s_2, s_3), (c, s_2, s_4), (c, s_2, s_3, s_6, s_5), (c, s_2, s_3, s_6).

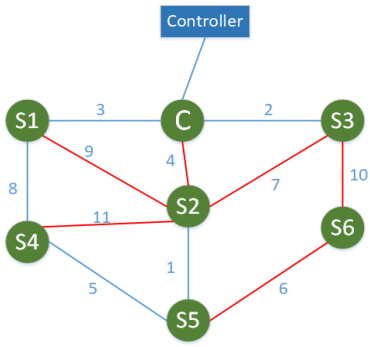

FIGURE 7. This sample topology consists of one controller and seven switches. Switch c is the connection node and the numbers besides links are Link Trust Level. The red links are the edges of MST.

TABLE 1. Computation Time of DC-CTIB and U-CTIB in Fat-Tree Topology.

	Fat-Tree (k=2)	Fat-Tree (k=4)	Fat-Tree (k=6)
DC-CTIB	0.219 ms	0.989 ms	6.348 ms
U-CTIB	3.908 ms	74.730 ms	3812.026 ms

TABLE 2. Computation Time of UN-CTIB in Random User Topology. E-Num represents the number of edges. N-Num represents the number of nodes.

	E-Num=100	E-Num=200	E-Num=400	E-Num=800
N-Num=50	0.830 ms	1.465 ms	1.835 ms	2.643 ms
N-Num=100		1.511 ms	1.947 ms	3.639 ms
N-Num=200			2.754 ms	4.898 ms
N-Num=400				6.185 ms

F. SIMULATION

We simulate the algorithm using the Python package Networkx [24], [25] on a personal computer with an Intel Core i5-7500 CPU and 16GB RAM. We emulate the Data Center network and User network topologies and test the path selection results and the computation times. We first build Fat-Tree topologies (k = 1,2,3) and randomly generate the LTLs for all the links. We run U-CTIB and DC-CTIB on these topologies. The computation time is shown in Tab.1. We can find that the computation time of DC-CTIB is much less than U-CTIB.

Then we generate stochastic user network topologies with different scales and random LTLs. Tab.2 shows the computation time of UN-CTIB. E-Num and N-Num represent the number of edges and the number of nodes. We choose not to display the U-CTIB computation time in the table, due to that the value is too large. Under the network condition of 200 edges and 50 nodes, the computation time of U-CTIB has exceeded 4000ms. Therefore, UN-CTIB improves computational efficiency compared to U-CTIB.

V. IMPLEMENTATION AND EXPERIMENTS ANALYSIS

In this section, we first introduce an implementation of our centralized control mechanism for in-band control traffic.

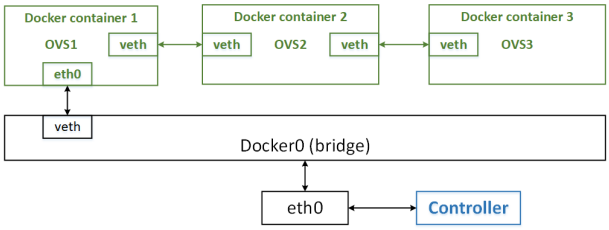

FIGURE 8. To implement Centralized CTIB control, we need a SDN in-band network environment. For isolating the direct control traffic between normal switches and the controller, we use Docker container network to build in-band topology as this figure shows.

Then based on this network environment, we build DC fat-tree network and user network topologies. In two kinds of topologies, we design experiments to test the QoS of in-band control traffic when using CTIB centralized control and Layer 2 switching. It should be noted that all experimental results shown in this section are the average of randomly repeated experimental results for three times. By analyzing the experimental results, we conclude that centralized in-band traffic control can improve the QoS of in-band control traffic.

A. IMPLEMENTATION

We build an in-band SDN network environment on a server. We create a VM with Ubuntu 14.04 operating system, which has 4-core Intel CPU and 16-GB RAM. This VM contains an SDN controller and a network topology. We choose ONOS (Open Network Operating System) [26] as the SDN controller to manage the data plane. For data plane switches, we choose OVS (Open VSwitch) [12] to forward data traffic.

Then we explain how to isolate the in-band control traffic in a single VM. At first, we connected all the switches through network tools like Mininet, which virtualizes multiple switches in the same namespace. However, we found that the control traffic of OVS is sent to the Linux Kernel protocol stack by default, and OVSs in the same namespace share the same Linux Kernel. This means that if there are multiple OVSs in the same namespace, the control traffic of one OVS cannot be forwarded through the data plane, but directly enter the Linux Kernel protocol stack, which means this method cannot support in-band connection. To solve this problem, we use the Docker container network to avoid the direct connection between the controller and the normal switch. We put every OVS into a Docker container and build the network topology with the help of Docker container network [27]. We explain the isolation of control traffic using a simple in-band network example shown in Fig.8. This VM contains three containers and a controller. OVS1 is the connection node that directly connects to the controller using Docker bridge mode. To isolate direct control traffic between OVS2/OVS3 and Controller, the eth0 ports in Docker container 2 and 3 are removed. OVS2 and OVS3 are connected to OVS1 by two pairs of veth.

Based on this in-band SDN network demo, we implement CTIB control in the ONOS controller by adding a CTIB application module and operate the following experiments.

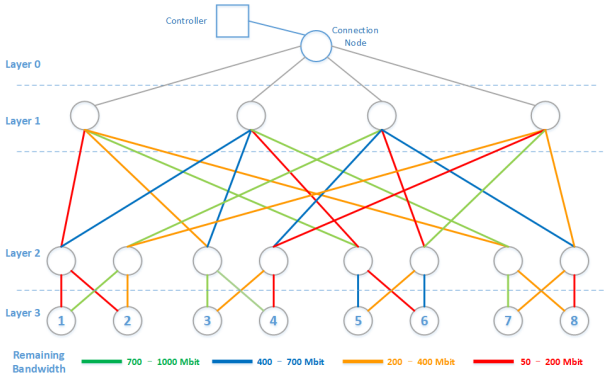

FIGURE 9. This figure shows a Fat-Tree (k = 4) topology. The colors of links represent the intervals of link remaining bandwidth.

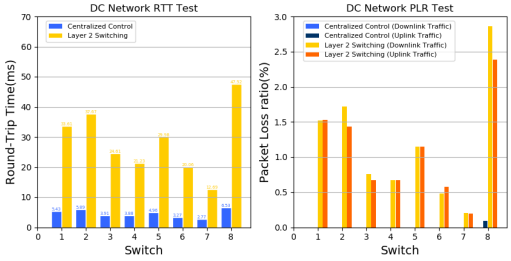

FIGURE 10. Through the experiment results, we can verify that in DC network topology, DC-CTIB centralized control can make better use of unoccupied network resources and provide better QoS for in-band control traffic than Layer 2 switching.

B. EXPERIMENT IN FAT-TREE (K = 4) TOPOLOGY

Firstly, we test the performance of DC-CTIB. We build fat-tree (k = 4) topology and set the maximum link bandwidth as 1 Gbit. We divide the link remaining bandwidth into four levels and mark them with different colors. Green, blue, orange and red respectively represent 700 to 1000Mbit, 400 to 700Mbit, 200 to 400Mbit and 50 to 200Mbit. The minimum remaining bandwidth is 50Mbit which, according to the former statistic, will not cause the congestion of control traffic. With the constraint that the four kinds of remaining bandwidth in the topology are equally proportional, we generate the remaining bandwidth of each link randomly. The experiment topology is shown in Fig.9. Then we implement DC-CTIB and Layer 2 switching to manage in-band control traffic. We first test the RTT (Round-Trip Time) between each bottomed switch and the connection node. Then we send downlink control traffic (the traffic from a controller to a switch) and uplink control traffic (the traffic from a switch to a controller) between the controller and every bottomed switch at a rate of 120Kbit/s for 100 seconds and record the PLR (Packet Loss Ratio).

Fig.10 shows the experimental results. First, through the RTT test results, we can find that the RTT using DC-CTIB is smaller than the RTT using Layer 2 switching. The average RTT of the eight bottomed switches using CTIB and Layer 2 forwarding are respectively 4.58ms and 28.42ms. In the PLR test, the PLR using DC-CTIB is almost 0. Since the data size

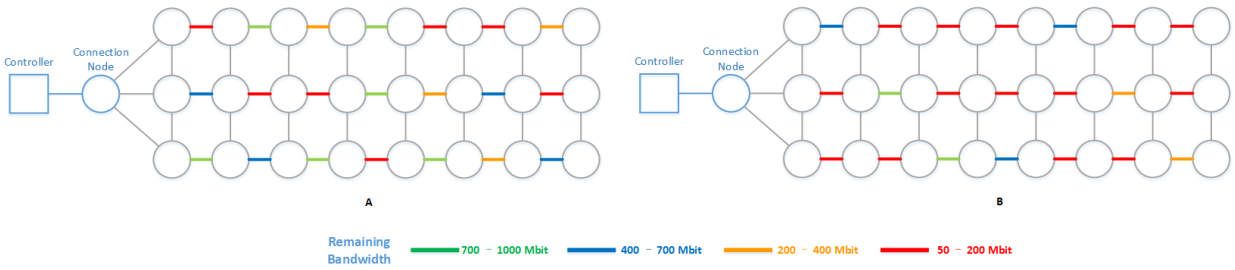

FIGURE 11. In this designed user network topology, all switches are arranged in three rows and we set two link states for experiments.

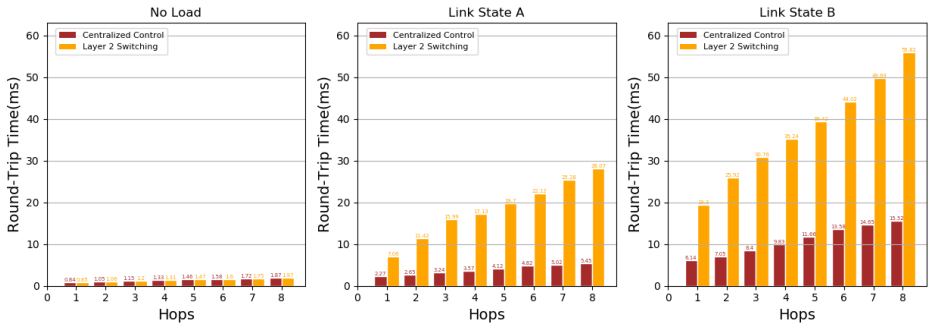

FIGURE 12. The RTT of UN-CTIB control and Layer 2 switching in the reference group and link states A and B.

of control traffic is small and it is transmitted on the most trusted path selected by DC-CTIB, the packet loss ratio can be kept at an extremely low level. On the other hand, Layer 2 forwarding is easy to cause packet loss when the remaining bandwidth of links is not sufficient enough. Through this experiment, we can verify that in DC network topology, DC-CTIB centralized control can make better use of unoccupied network resources and provide better QoS for in-band control traffic than Layer 2 switching.

C. EXPERIMENT IN DESIGNED USER NETWORK TOPOLOGY

In order to verify the functionality of UN-CTIB in the user network, we have designed a specific topology for testing. The topology graph is shown in Fig.11, and the topology is arranged in a rectangle. As demonstrated, all switches are arranged in three rows and adjacent nodes on the same column and row are connected. In Fig.11 we still use different colors to represent the remaining bandwidth situations of links and the maximum bandwidth of each link is 1 Gbit/s. We set up two scenarios to test the performance of UN-CTIB under different load conditions of the network. In link state A (Fig.11 A), we randomly select one of the three links between each adjacent two-row switch to randomly generate the remaining bandwidth in the range of 50-200 Mbit. The other two links between every two columns of switches randomly generate remaining bandwidth in the range of 0-800 Mbit. In link state B (Fig.11 B), we randomly select two of the three links between each adjacent two-row switch to randomly generate the remaining bandwidth in the range of 50-200 Mbit. Another link between each of the two columns of switches randomly generates the remaining bandwidth in the range of 0-800 Mbit. Link states A and B respectively simulate the network conditions of the user network under moderate load and heavy load. We also set up a reference group as the baseline for comparison, which has no data traffic in the network.

We implement UN-CTIB centralized control and Layer 2 switching in these three link states. Then we send downlink control traffic and uplink control traffic between the controller and every switch at a rate of 120Kbit/s lasting 100 seconds and test the average RTT (Round-Trip Time) and PLR (Packet Loss Ratio) of same-column switches. The experimental results are shown in Fig.12 and Fig.13. In the reference group, the control traffic RTTs are both very small using UN-CTIB and Layer 2 switching, and the packet loss rate remains at 0. In link state A, the RTT using Layer 2 switching is much larger than using centralized control. For switches of the same hop, the Layer 2 switching RTT is approximately five times that of UN-CTIB. In the case of PLR test, packet loss occurs at the fifth hop using Layer 2 switching, and the packet loss rate increases with the growing of hops. When using UN-CTIB, packet loss occurs at the eighth hop. In link state B, as the network load increases, the RTT rises significantly. We can see that UN-CTIB controls RTT at a relatively low level, about one-third of the RTT using Layer 2 switching. In terms of the packet loss ratio, when using Layer 2 switching, the

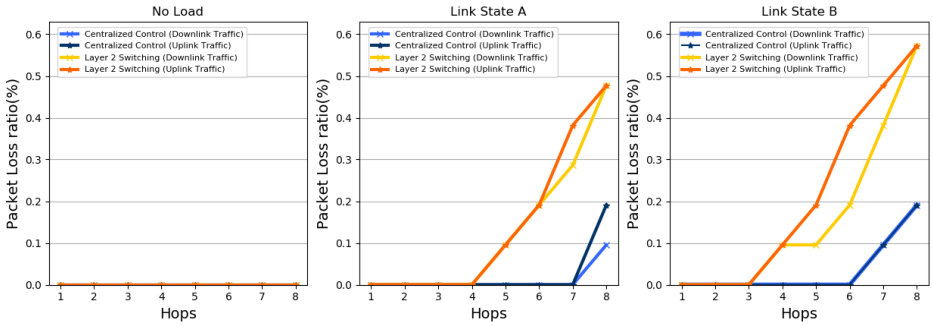

FIGURE 13. The PLR of UN-CTIB control and Layer 2 switching in the reference group and link states A and B.

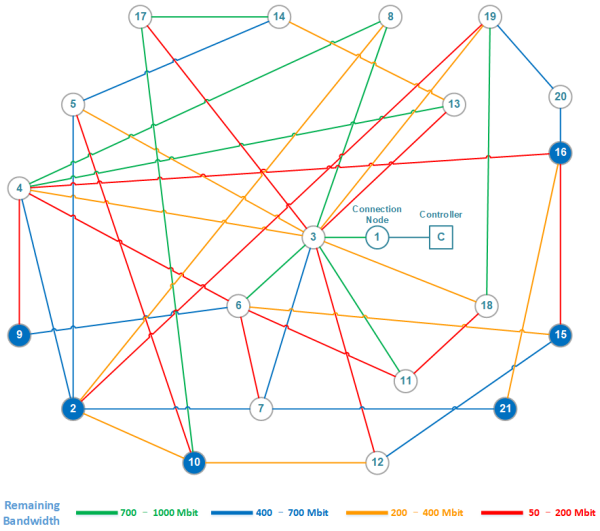

FIGURE 14. This random topology is generated by Brite topology generator, which consists of 20 normal switch nodes and 40 links. Node 1 is the connection node. The remaining bandwidth of each link is marked by colors.

fourth-hop switch has packet loss. When utilizing centralized control, the packet loss appears in the seventh hop.

This comparison experiment shows that UN-CTIB has better performance than Layer 2 switching under different network load conditions. As the network load increases, centralized control minimizes the impact of link resource reduction on control traffic by selecting the most reliable control path. We can conclude that UN-CTIB can improve the QoS of in-band control traffic.

D. EXPERIMENT IN RANDOM USER NETWORK TOPOLOGY

Considering the lack of randomness in the user topology that we design in the former experiment, we add a random topology experiment to verify the QoS of in-band control traffic using centralized control in random topologies. We use the Brite topology generator [28] to generate a random topology as shown in Fig.14. The topology consists of 20 normal switch nodes and 40 links. Node 1 is the connection node.

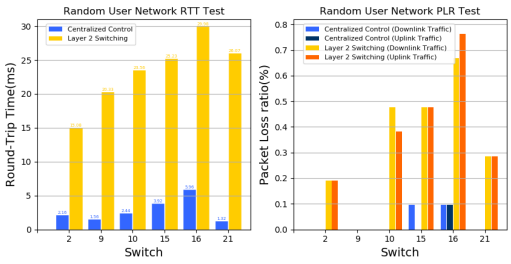

FIGURE 15. The experiment results verify that UN-CTIB can optimize the path selection and provide better QoS for in-band control traffic than Layer 2 switching in random user topology.

We randomly generate the remaining bandwidth of links, as indicated by the color in Fig.14. We select six normal switches in the topology as test nodes, which are marked by blue in Fig.14. We send downlink control traffic and uplink control traffic between the controller and each test node at a rate of 120Kbit/s for 100 seconds. Then we record the RTT and the PLR between the connection node and each test node.

The experimental results are shown in Fig.15. In the RTT test, the average RTT of the six test nodes using centralized control is 2.89ms, and when utilizing Layer 2 switching, the value is 23.37ms. It is apparent that the centralized control RTT is much lower than the Layer 2 switching RTT. In the PLR test, when UN-CTIB operates, only two switches have packet losses, and the PLRs are less than 0.1%. However, when Layer 2 switching is working, five switches lose packets, and the PLRs all exceed 0.1%. This experiment verifies that UN-CTIB can optimize the path selection and provide better QoS for in-band control traffic than Layer 2 switching in random user topology.

VI. CONCLUSION

In the SDN in-band network, the connection between the control and data planes needs to be established through the data plane switches. In current implementations of the in-band mode, the forwarding of in-band control traffic depends on Layer 2 switching which cannot adapt to the network state changes. Thus, the in-band control traffic is vulnerable and has no guarantee of QoS.

In this article, we propose Centralized Trust-based In-Band (CTIB) control mechanism for SDN in-band control channel to improve the in-band control traffic QoS. By analyzing, we find that in-band control traffic has little data size and a high requirement for path trust level. According to these features, we design CTIB routing algorithm to select the most trusted control paths for switches. Furthermore, we enhance the computational efficiency of CTIB routing algorithm in DC and User network scenarios. In the final part, we introduce the implementation of CTIB control mechanism in an in-band SDN network built on a server. Based on this network environment, we respectively compare the performance of CTIB control and Layer 2 switching for in-band control traffic in DC and User network topologies. By analyzing the experiment results, we conclude that CTIB control can effectively improve the QoS of in-band control traffic.

APPENDIX
PROOF FOR THEOREM 1

Proof: Let $G(V, E)$ be the topology graph. Let $T(V, E_1)$ represents the Maximum Spanning Tree of G, $E_1 \in E$.

$$E_2 = E - E_1.$$

E_2 represents the links not belong to T. Let wl be the least trusted link in T.

$$wl = \arg\min_{l}(LTL(l), (l \in E_1)).$$

E_2 consists of 2 parts:

$$\begin{aligned} E_2 &= E_3 + E_4, \\ E_3 &= \{l, l \in E_2 \wedge LTL(l) < LTL(wl)\}, \\ E_4 &= \{l, l \in E_2 \wedge LTL(l) > LTL(wl)\}. \end{aligned}$$

E_3 consists of links that have less Link Trust Level than wl. E_4 is the set of links having bigger Link Trust Level than wl. For T is the Maximum Spanning Tree, according to Kruskal Algorithm, $l(l \in E_4)$ belongs to a ring H_l and has the least trust level in H_l. And all the links in H_l except l are in T.

For normal switch s_i, the path between c and s_i in T is $p_T(c, s_i)$. We assume that there is a path between c and s_i in G, $p_G(c, s_i)$, differing from $p_T(c, s_i)$. We compare the Path Trust Level of $p_T(c, s_i)$ and $p_G(c, s_i)$. For $p_T(c, s_i)$,

$$PTL(p_T(c, s_i)) = LTL(wl).$$

For $p_G(c, s_i)$, let L be the set of links that belong to $p_G(c, s_i)$ and E_2:

$$L = \{l, l \in p_G(c, s_i) \wedge l \in E_2\}.$$

There are 3 situations about L. Situation 1:

$$L \subset E_3.$$

Situation 2:

$$(L \cap E_3 \neq \varnothing) \wedge (L \cap E_4 \neq \varnothing).$$

Situation 3:

$$L \subset E_4.$$

In situation 1 and 2,

$$\exists l, l \in p_G(c, s_i) \wedge l \in E_3.$$

We can get

$$PTL(p_G(c, s_i)) < PTL(p_T(c, s_i)).$$

In situation 3, each link in L, $l(l \in L)$, can be replaced by the other links in H_l, which are in T and have higher LTL. Therefore we can obtain

$$PTL(p_G(c, s_i)) < PTL(p_T(c, s_i)).$$

In three situations, $p_T(c, s_i)$ has higher PTL than $p_G(c, s_i)$. Hence, we conclude that $p_T(c, s_i)$ is the most trusted path between c and s_i in G. □

REFERENCES

[1] N. Mckeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[2] Y. Li and M. Chen, "Software–defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[3] S. Jain, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, A. Vahdat, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, and J. Zhou, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.

[4] *ONF OpenFlow Switch Specification-Version 1.3.0*, Open Netw. Found., Menlo Park, CA, USA, 2012.

[5] *ONF OpenFlow Switch Specification-Version 1.4.0*, Open Netw. Found., Menlo Park, CA, USA, 2013.

[6] A. Jalili, H. Nazari, S. Namvarasl, and M. Keshtgari, "A comprehensive analysis on control plane deployment in SDN: In-band versus out-of-band solutions," in *Proc. IEEE 4th Int. Conf. Knowl.-Based Eng. Innov. (KBEI)*, Dec. 2017, pp. 1025–10314.

[7] *IEEE Standard for Ethernet*, IEEE Standard 802.3-2018, 2018.

[8] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "In-band control, queuing, and failure recovery functionalities for openflow," *IEEE Netw.*, vol. 30, no. 1, pp. 106–112, Jan. 2016.

[9] C. Suo, I.-C. Tsai, and C. H.-P. Wen, "ERIC: Economical & reconfigurable hybrid-band control for software-defined datacenter network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2016, pp. 214–219.

[10] A. Raza, A. Gohar, and S. Lee, "MPTCP based in-band controlling for the software defined networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2017, pp. 163–167.

[11] A. Raza and S. Lee, "Gate switch selection for in–band controlling in software defined networking," *IEEE Access*, vol. 7, pp. 5671–5681, 2019.

[12] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, and P. Shelar, "The design and implementation of open vswitch," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2015, pp. 117–130.

[13] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[14] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: A high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, 2009.

[15] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: A scalable and fault-tolerant network structure for data centers," in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, 2008.

[16] Dell, *DELL EMC Netw. Z9100-ON Ser. SWITCHES*. [Online]. Available: http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/Dell-Networking-Z9100-spec-sheet.pdf

[17] Edge-Core. *AS7712-32XAS7716-32X Series Switch*. [Online]. Available: https://www.edge-core.com/_upload/images/AS7712_AS7716-32X_DS_R17_20191226.pdf

[18] *Link Layer Discovery Protocol*, IEEE Standard 802.1AB-2005, 2005.

[19] U. Lamping and E. Warnicke, "Wireshark user's guide," *Interface*, vol. 4, no. 6, 2004.

[20] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust–based routing and intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 2, pp. 169–183, Jun. 2012.
[21] N. Marchang and R. Datta, "Light-weight trust-based routing protocol for mobile ad hoc networks," *IET Inf. Secur.*, vol. 6, no. 2, pp. 77–83, 2012.
[22] J. E. Beasley and N. Christofides, "An algorithm for the resource constrained shortest path problem," *Networks*, vol. 19, no. 4, pp. 379–394, Jul. 1989.
[23] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, "Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 388–415, 1st Quart., 2018.
[24] A. Hagberg, D. Schult, P. Swart, D. Conway, L. Séguin-Charbonneau, C. Ellison, B. Edwards, and J. Torrents. (2013). *Networkx. High Productivity Software for Complex Networks*. [Online]. Available: https://networkx.lanl.gov/wiki
[25] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python Sci. Conf. SciPy*, 2008, pp. 11–15.
[26] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.
[27] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, p. 2, 2014.
[28] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proc. 9th Int. Symp. Modeling, Anal. Simulation Comput. Telecommun. Syst. (MASCOTS)*, 2001, pp. 346–353.

WENTAO FAN received the B.S. degree in information and communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, where he is currently pursuing the Ph.D. degree. His research interests include future network architecture, software defined networks, segment routing, and P4.

FAN YANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications. He is currently a Lecturer with the Beijing University of Posts and Telecommunications. His research interests include software defined networks, high-performance routing, and switching technologies. He has participated in the National 973 Planning Project Research on Service-Oriented Future Internet Architecture and Mechanism, National 863 Project Service-Oriented SDN Architecture and Key Technology Research, Future Network System and Structural Research for Service-Oriented Resource Intelligent Scheduling funded by National Key Laboratory, etc., presided over German Telecom & Huawei's Next-Generation IP Routing Equipment Research, Huawei Ultra-High-Speed Network Processor Search Algorithm Research, and other enterprise projects. He has published more than 30 SCI/EI articles and national invention patents. He has received the 2013 National Technology Contribution Individual Award, the Huawei Golden Network Award, and the China Communication Society Science and Technology First Prize.