Mitigating DoS Attacks in SDN Using Offloading Path Strategies

Tai-Siang Huang, Po-Yang Hsiung, Bo-Chao Cheng

Department of Communications Engineering, National Chung-Cheng University, Taiwan
x030060@gmail.com, paul100dtj@gmail.com, bcheng@ccu.edu.tw

Abstract

Software-Defined Networks (SDNs) were created to facilitate the management and control of the network. However, the security problem is still unresolved. To avoid the DoS attacks caused by links exceeding the bandwidth load (such as traffic flooding and security loopholes), the most simple mitigation solution is to offload the data by transferring it to other links. However, the transfer of information could lead to high bandwidth loads on other links. To overcome this problem, this paper proposes a method called "Avoid Passing High Utilization Bandwidth (APHUB)," which aims to (1) prevent the unloaded data putting additional load on the links when passing through the high bandwidth and (2) find a suitable new path. A comparison of the maximum bandwidth utilization using the proposed method with that of other algorithms showed that this method consistently produced the smallest bandwidth utilization; we thus consider it a better mitigation method than those presented previously.

Keywords: Software defined network, Offload, Path selection, Maximum utilization

1 Introduction

With the development of modern networks, software defined network (SDN) technology is gradually being developed to make it easier and more convenient for managers to control and operate the networks. However, security problems in the network are growing so fast that the design and updates for the defense strategies can hardly keep up due to the physical limits of the security devices. As a result, excessive congestion occurs in some links because of DoS attacks, leading to the loss of data packets and other problems. In the worst case, hackers might use this defect to cause a transmission inefficiency in the entire network 0. To avoid this problem, many studies have developed various load-balancing methods.

To avoid high bandwidth utilization caused by overuse in the same link, load-balancing methods transfer data by passing links with high bandwidth utilization to links with lower bandwidth utilization. This method is used mainly in decentralized IP networks. The disadvantage of this method is that it cannot ensure that, after transferring to the links with lower bandwidth utilization, the next switch still has links with low bandwidth utilization available.

One load-balanced method uses SDNs to monitor the switches and collect information about the usage of all links in the entire network so that it can calculate and reallocate the entire network bandwidth usage between all links. The new balanced load is achieved based on the results of the calculation.

Though the two methods mentioned above solve the basic problem of high bandwidth utilization, they cannot ensure that, in the path from the source to the destination, the data always passes through the links that have the smallest bandwidth utilization. Therefore, this study aims to find a path that ensures the smallest maximum bandwidth utilization ($\hat{u}$) in all paths which is demonstrated by the following formula:

$$\hat{u} = Max\{P_i[U]\}, \ i = 1, 2, \ldots, n$$

Find the P which $\hat{u}$ is minimum → Min $\hat{u}$,

where P is the path, û is the maximum bandwidth utilization, i is one of the possible paths from source to destination, and $P_i[U]$ is the set of bandwidth loads for each link between the switches selected by this path ($P_i[U] = [u_1, u_2, \ldots, u_m]$).

The remainder of this article is structured as follows: The following section reviews previous literature to introduce existing flow control methods and compares them with the method proposed in this study. The third section presents the proposed method with an example of its operation. The fourth section lists the experimental results with the analysis and explanations, and the fifth section presents a summary of this article and discusses future developments.

2 Related Work

In 2016, Benjamin Baron et al. 0 used vehicle traffic for the centralized control of quality data unloading and used existing roads and road networks instead of data networks to mitigate the delays in traffic flow. This system takes advantage of the mobility of the vehicle by transmitting delay-tolerant traffic through

*Corresponding Author: Bo-Chao Cheng; E-mail: bcheng@ccu.edu.tw
DOI: 10.3966/160792642019072004026

the road network and uses the daily routine of the vehicle to reduce the traffic burden of the traditional data network. The author proposed the SDN-based structure in which a controller and a set of fixed wireless data storage devices make up the unloading point and are used as a forwarding engine. The controller receives the request to offload all or part of the data transmission and selects the vehicle flow for a series of offloading points that meet the transmission performance requirements in terms of bandwidth and latency. The controller solves the traffic distribution problem using Max-Min fairness allocation, calculates the unloading point sequence, connects to the unloading point, installs the forwarding state, and configures the scheduling policy.

As the amounts of users and data traffic on mobile networks has grown in recent years, the 3G/4G base station (BS) or access point (AP) network service speed have declined, resulting in low QoS. To solve this problem, Jang and Chang 0 proposed a new approach called "Flow Management on Mobile Data Using SDN (FMSDN)", which uses software to define the characteristics of the network, depending on the different circumstances, to manage the flow and control of data, and compare the methods to the enforced handover and horizontal handover theories. The purpose of this method is to prevent information loss from BS or AP, low QoS, or other issues caused by the overly huge data flow. However, the question remains whether a switch bandwidth can bear the huge amount of data in the huge information flow when passing from BS or AP down to SDN.

In the dynamic load balanced, the lack of strict routing synchronization increases the tendency of transient loops. In 2016, Lee et al. 0 attempted to make the router achieve a dynamic load balance while avoiding the occurrence of transient loops in the IP network using two methods: Local Traffic Rerouting (LTR) and Global Traffic Rerouting (GTR). The only difference between the two methods is “Local” and “Global”; the core algorithms are the same. The focus is to prevent the link bandwidth usage from getting too large and leading to network congestion and packet loss. The use of LTR or GTR can dynamically adjust the flow of information and achieve a balanced load. In the algorithm, the way the links are connected to use the directed acyclic graph (DAG) ensures that the transient loops are avoided. In the path search, the search takes the shortest path, and the selection of the next node is relaxed. The threshold is used to dynamically adjust the path selection.

Lan, Wang, and Hsu 0 proposed an algorithm called Dynamic Load-balanced Path Optimization (DLPO). This algorithm is optimized from LABERIO 0 and contains two main stages. The first stage is the initialization of the path in which the DLPO attempts to find a temporary path based on the available bandwidth of the bottleneck link for each path, and in all possible paths between the source and target hosts. The path with the largest available bandwidth on the bottleneck link will be selected as a temporary path. The second stage is the optimization of the dynamic path in which the DLPO changes the traffic path during traffic transmission to balance the link utilization and solve the congestion problem in the data center network. The DLPO load balancing consists of two algorithms: the Multi-link DLPO algorithm and the Single-link DLPO algorithm. The Multi-link DLPO algorithm can quickly balance the link utilization in the network to address some congestion paths, and the Single-link DLPO algorithm can reroute the traffic to avoid using high utilization links to solve the congestion path that the Multi-link DLPO algorithm cannot handle.

3 Approach

In this article, we propose an algorithm called “APHBU,” which collects current information, controls the status of the switch, and calculates an appropriate path to offload data. This method follows four steps:

Detection. The algorithm first detects whether the original path has a link with a high bandwidth utilization. The switch notifies the control layer, which recalculates the new path and passes it back to the switch.

Sorting. The algorithm recalculates the path and uses the features of the SDN to collect the information downstream of the switch, mainly for each link load conditions, and to sort all of the links from small to large. This sorting procedure is used to create the Order List (L) in preparation for the next algorithm.

Connection path establishment. The algorithm defines a Select List (S), which starts with NULL; through the L, after DELETE the load link one by one from the smallest, add INSERT it into the S, and determine whether a path of S can be found between the source and the destination; if not, repeat the action of DELETE and INSERT to re-judge, until successful. By connecting the S path, the connected tree is formed.

Assurance in the minimum bandwidth utilization in the path. After forming the connected tree, there may be plural paths between the source and the destination. To improve the path, the shortest path algorithm is applied to the connected tree.

4 Example

In this subsection, we take a basic network as an example, to compare the path found by APHUB with those found by other algorithms. First, as shown in the diagram of the network, Figure 1, A is the source and H is the destination, and the number associated with the links are current link bandwidth utilizations. The results of three algorithms—APHUB, Shortest Path Smoothing (SPS), and Minimum First Smoothing (MFS)—are compared.

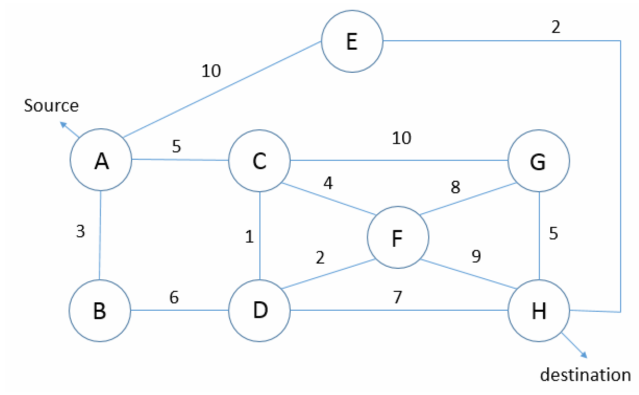

Figure 1. Example network

4.1 APHUB

First, it removes all links from the network, and sorts the links by current link bandwidth utilization to create a sorted list L = [Link(C, D), Link(D, F), Link(E, H), Link(A, B), Link(C, F), Link(A, C), Link(B, D), Link(D, H), Link(F, G), Link(F, H), Link(A, E), Link(C, G)].

Next, it joins the links one by one and decide whether A can be connected to H successfully. When the link comes to Step 7, as shown in Figure 2, the connection is successful, and the formed structure is recorded as a connected tree. A multiple path is found, and the shortest path is applied to the connected tree. The best path is found as A → C → D → H, and the maximum link utilization is the last link, namely, Link (D, H), whose value is 7.

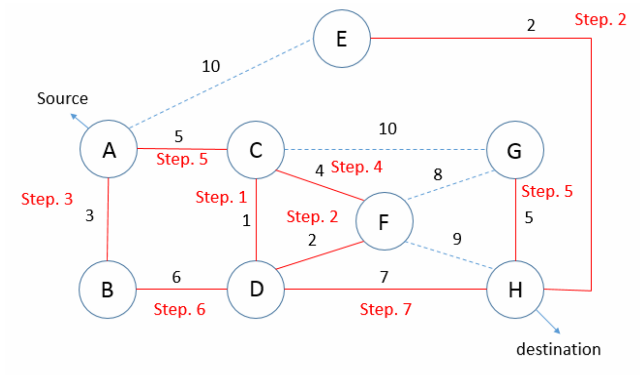

Figure 2. APHUB outcomes

4.2 Shortest Path Smoothing (SPS)

SPS is a typical path search algorithm, which selects the route by finding all the paths connecting A → H and by summing the utilizations in all paths shown in Table 1. The path with the smallest utilization is selected, as shown in Figure 3. The selected path is A → E → H, but the maximum utilization is Link (A, E), whose value is 10.

Table 1. The results of the three algorithms

Algorithm	Path	$\hat{u}$
APHUB	[Link (A, C), Link (C, D), Link (D, H), Link (G, H)]	Link (D, H) = 7
Shortest Path Smoothing (SPS)	[Link (A, E), Link (E, H)]	Link (A, E) = 10
Minimum First Smoothing (MFS)	[Link (A, B), Link (B, D), Link (D, C), Link (C, F), Link (F, G), Link (G, H)]	Link (F, G) = 8

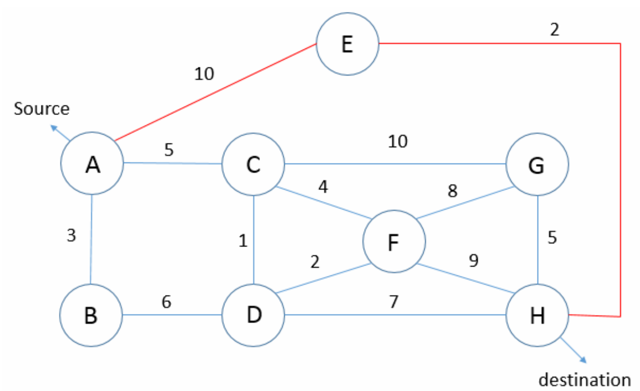

Figure 3. SPS outcome

4.3 Minimum First Smoothing (MFS)

MFS is an intuitive algorithm that directly selects the minimum utilization of the link to forward; the time complexity is low and can quickly determine the path.

Step 1. Of the three links after A, the minimum bandwidth link, Link (A, B), is selected.
Step 2. After reaching B, as the connection cannot move backward, only one link is available, namely, Link (B, D), so the connection continues here.
Step 3. After reaching D, there are three links, and the minimum bandwidth link, (D, C), is selected.
Step 4. After reaching C, there are three links, the minimum bandwidth link, Link (C, F), is selected.
Step 5. After reaching F, there are three links, the minimum bandwidth link is (C, D). However, as D has already been visited, the second smallest Link (F, G) is selected.
Step 6. After reaching C, there are two links, the minimum bandwidth link, Link (G, H), is selected.

To arrive at the destination, the selected path is A→B→D→C→F→G→H. In the path, the maximum link bandwidth utilization is Link (F, G), whose value is 8 (as shown in Figure 4).

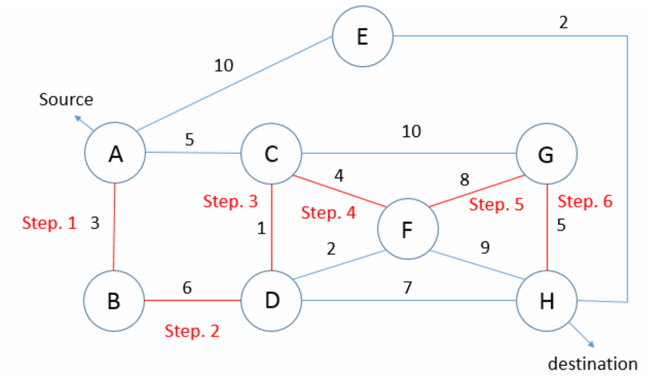

Figure 4. Step-by-step diagram of MFS

Consolidating the results of the three algorithms results in Table 1. The path found by APHUB has the minimum value of $\hat{u}$.

5 Experiment

This section compares the algorithms presented in the examples to compare the simulations in terms of the maximum link utilization of each û and the average size of the bandwidth. To make the simulation easier, this article uses C language to perform the experiment (as sown in Table 2). Table 3 shows the pre-set parameters before the experiment is carried out.

Table 2. Experimental simulation environment

SW/HW	Description
Processor	Intel Core i7 3.40 GHz
RAM	4GB
Operating System	Microsoft Windows 7
Programming Language	C Language

Table 3. Parameters in the experiments

Parameter	Quantity
Number of switches	100~500
Offload	5~35 (%)
Output Link	1~10
Utilization	0~99 (%)
Average bandwidth utilization in the network	30~70 (%)

In this lab, the assumed specification for each switch is the same, but output link and bandwidth utilization are different. The topology of the network is a random connection, and the value of bandwidth utilization is given by the Gaussian discrete method, so that the value belongs to the normal distribution. In this experiment, we discuss how different averages will lead to the relation between the average and the maximum bandwidth utilization. In the experiment, three path search methods, APHBU, MFS, and SPS are discussed.

The relation between the maximum link bandwidth load rate and the average bandwidth of the entire network link bandwidth is shown in Figure 5. It can be seen from the figure that APHUB can effectively keep the maximum link bandwidth utilization in the path below the average value of the network link bandwidth utilization. For MFS and SPS, the situation is as follows:

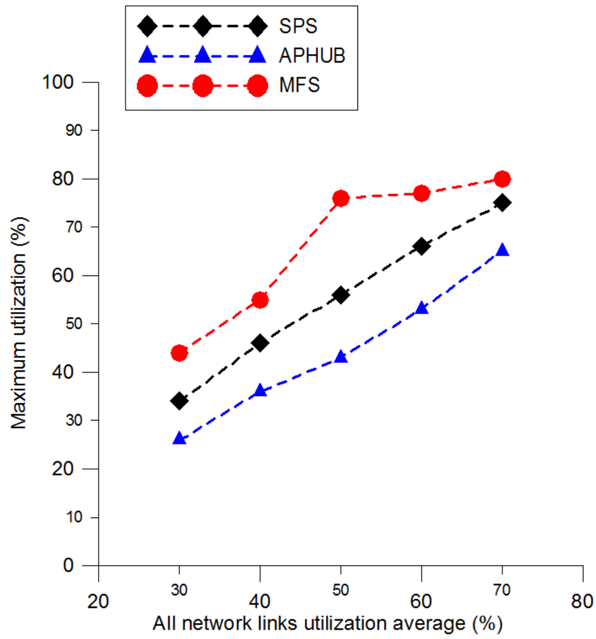

Figure 5. The relation between the average bandwidth utilization and the maximum bandwidth utilization

(1) As mentioned above, the maximum utilization of the MFS maximum load is the largest among the three, much larger than the average, or is unable to let the data offload connect to the link with maximum utilization, thus resulting in failure.

(2) The maximum utilization of the SPS is significantly smaller compared to that of MFS, but the maximum link bandwidth utilization of its path is still not comparable to that of APHUB, and the minimum value cannot be achieved.

6 Conclusion

The purpose of this paper is to prevent links to high bandwidth utilization when the data path is reproduced and to ensure that the path is optimized so that the highest link utilization is the smallest of all paths. In addition, the new data path can ensure that all links in the path of the data flow are low load links, thus balancing the load. As the link in the path of the largest utilization joins last, it can be clearly and quickly known which link requires further analysis.

Acknowledgements

This research is partially supported by Ministry of Science and Technology, Taiwan under Grant no. MOST 105-2221-E-194-014-MY2 and MOST 106-3114-E-006-003.

References

[1] R. Kandoi, M. Antikainen, Denial-of-service Attacks in OpenFlow SDN Networks, *IFIP/IEEE International Symposium on Integrated Network Management*, Ottawa, Canada, 2015, pp. 1322-1326.

[2] B. Baron, P. Spathis, H. Rivano, M. D. de Amorim, Y. Viniotis, M. Ammar, Centrally-Controlled Mass Data Offloading Using Vehicular Traffic, *IEEE Transactions on Network and Service Management*, Vol. 14, No. 2, pp. 401-415, June, 2017.

[3] H.-C. Jang, C.-H. Chang, Context Aware Mobile Data Offload Using SDN, *Telecommunication Networks and Applications Conference (ITNAC)*, Dunedin, New Zealand, 2016, pp. 185-190.

[4] K.-Y. Li, C.-W. Chen, S. S. W. Lee, Dynamic Load Balanced Routing in IP Networks, *6th International Conference on Information Communication and Management*, Hertfordshire, UK, 2016, pp. 216-221.

[5] Y.-L. Lan, K. Wang, Y.-H. Hsu, Dynamic Load-balanced Path Optimization in SDN-based Data Center Networks, *10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Prague, Czech Republic, 2016, pp. 1-6.

[6] H. Long, Y. Shen, M. Guo, F. Tang, LABERIO: Dynamic Load-balanced Routing in OpenFlow-enabled Networks, *IEEE Advanced Information Networking and Applications Conference*, Barcelona, Spain, 2013, pp. 290-297.

Biographies

Tai-Siang Huang received a M.S. degree in degree in Communications Engineering at National Chung-Cheng University in 2017. His research interests include network security and path planning algorithms.

Po-Yang Hsiung received B.S. (2015) from Yuan Ze University, Taiwan, and M.S. (2017) from National Chung Cheng University, Taiwan. His research focuses on cell phone security, fog computing, energy consumption and aims at collusion attacks.

Bo-Chao Cheng is a Professor of Department of Communications Engineering at National Chung-Cheng University. Cheng received a Ph.D. degree in CIS from New Jersey Institute of Technology in 1996. After graduations, he also worked for Transtech Network (2000-2002), Bellcore (1998-2000) and Racal DataCom (1996-1998) respectively. His broad interests include network security, network management and real-time embedded system design.