Joint Optimization of Flow Entry Aggregation and Routing Selection in Software Defined Wireless Access Networks

Zhipeng Zhao[1], Bin Wu[1](✉), Jie Xiao[2], and Zhenyu Hu[3]

[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
{zzpeng, binw}@tju.edu.cn
[2] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China
[3] Henan Electric Power Survey & Design Institute, Henan, China

Abstract. Software Defined Wireless Access Networks (SD-WAN) produce large amounts of traffic, which has caused limited flow table space in the TCAM. To solve the issue, this paper enables routing to overlap as many as possible with a minimum transmission delay, and thus traffic can be allocated to common paths to achieve flow entry aggregation. However, overlapping routing and traffic delay are two conflicting factors. An Integer Linear Program (ILP) is first proposed to minimize the total cost by tradeoff of the two factors. Through numerical experiments, this paper shows the effectiveness of the proposed method.

Keywords: ILP · SD-WAN · Flow entry aggregation
TCAM (ternary content addressable memory)

1 Introduction

Wireless Access Network (WAN) can efficiently expand the coverage and capacity of networks by easily accessing access nodes [1]. According to a Cisco report, the proportion of worldwide wireless access users increased 120% in 2016 and had reached 96.5% by 2017, so large numbers of nodes and large amount of traffic degrade the network performance [2]. Software Defined Network (SDN) is a new type of network innovation architecture, which was first proposed by the Clean Slate research group in Stanford University [3]. It separates control plane from data plane by Openflow technology, thus realizing flexible control of network traffic and providing a good platform for the innovation of WAN [4]. In SD-WAN, control layer is mainly responsible by the SDN controller for scheduling resource in data plane and maintaining state information of the network. Data layer is made of Openflow switches which are responsible for data processing, data forwarding and status collection based on flow tables [5].

To achieve high-performance data forwarding and processing, current commercial switch is usually implemented by ternary content addressable memory (TCAM), which

S. Chellappan et al. (Eds.): WASA 2018, LNCS 10874, pp. 834–839, 2018.
https://doi.org/10.1007/978-3-319-94268-1_75

can parallelly match all rules formed by aggregating routes through a certain method [6, 7]. TCAM is very expensive due to the characteristics of 'high cost', 'high power consumption', and 'large silicon space occupation' [8, 9]. In addition, the fine-grained definition of SDN traffic increases the number of rules. Therefore, current Openflow switches face a severe problem of insufficient flow table space, which restricts the application of SD-WAN [10–12].

Existing methods to solve the above problem can be classified into three categories: (1) Considering the differences in the occurrence time, duration time and traffic of different hosts, and offloading the processing rules that are not currently used to reduce size. Typical works include: TFO [2], Cacheflow [3], AHTM [4] and TimeoutX [5]. (2) Analyzing redundant rules in the flow table and aggregating some of the rules without changing the semantics of the original rules. Typical works include ORTC [6], SMALTA [7], and bitweaving [8]. (3) Using decentralized routing among different switches to ensure that the flow table is not overflowed. Typical works include: OBS [9], Palette [10] and vCRIB [11]. The first method involves offload overhead. The second method separates routing from flow aggregation without achieving a joint design, and thus leads to a poor performance. The third method can minimize average traffic delay and achieve load balance through decentralized routing, but cannot easily generate overlapping routes.

In this paper, we jointly consider flow entry aggregation and routing selection to reduce the number of rules in flow table and average traffic delay. An Integer Linear Program (ILP) is formulated to find the optimal solution by leveraging the two factors.

The rest of this paper is organized as follows. System model is proposed in Sect. 2. ILP is formulated in Sect. 3 to achieve optimal joint design. Performance of the proposed algorithm is shown in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 System Model

In this paper, we consider a SD-WAN system model, which consists of a SDN controller and a number of Openflow switches. Assuming that one wireless access user with a given amount of traffic is required to be transmitted from one source to its destination switch. In this case, no direct wireless transmission paths between the source switch and the destination switch exist, so the optimal routing algorithm should be designed. Moreover, under the assumption that the flow table spaces of all switches are the same, the routing algorithm should be jointly designed with flow entry aggregation to further reduce the system cost. We denote s_i as the i^{th} relay switch, $1 \leq i \leq n$, where n is the number of relay switches. For convenience, s_1 and s_n are denoted as the source switch and the destination switch. We introduce binary connection variables to characterize the connection status between SDN switches in this paper. c_{ij} is denoted as the connection variable between s_i and s_j. We set $c_{ij} = 1$ representing that s_i is the adjacent node of s_j, and set $c_{ij} = 0$, when $0 \leq i, j \leq n+1, i \neq j$. In this paper, we assume that the network topology of the SDN is given, so c_{ij} is a given constant. Figure 1 shows the system model in this paper.

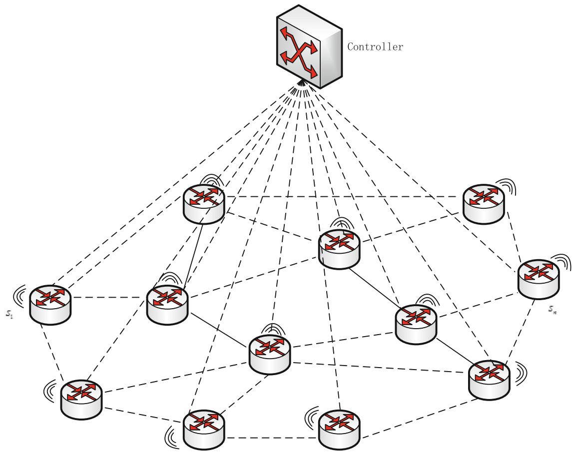

Fig. 1. A SD-WAN system model.

3 ILP Formulation

The number of rules in the flow tables and the traffic delay lead to the total cost at a SD-WAN network. To reduce the number of rules in the flow table, it is inevitable to increase the traffic delay. Therefore, we set a traffic delay cost scaling factor λ which represents the total delay at a SD-WAN network. In our design, the number of rules and the transmission delay can be properly controlled by the variable λ.

3.1 Notation List

Input:

N: The set of all nodes in a SD-WAN network G (N, L).
L: The set of all bidirectional edges in a SD-WAN network G (N, L).
d_{ij}: Wireless traffic delay between nodes i and j in the SD-WAN network. In this paper, it is regarded as the distance between nodes i and j, and $d_{ij} = d_{ji}$.
q_{sd}: The requested traffic from switch s to switch d.
λ: The cost scaling factor representing the transmission delay in SD-WAN network.
δ: The cost of each entry in the flow table.

Variables:

X_{ij}^{sd}: Binary variables. It is defined by $\{i, j, s, d \in N | i < j, s < d\}$. If some traffic start from the source switch s to the destination switch d and cross the wireless transmission path from the node i to the node j, it takes 1. Otherwise, it takes 0.
Y_{ij}^{s}: Binary variables. It is defined by $\{i, j, s, d \in N | i < j, s < d\}$. If some traffic start form the source switch s and cross the wireless transmission path from the node i to the node j, it takes 1. Otherwise, it takes 0.

3.2 ILP Formulation

$$\min\{\lambda \sum_{s\in n}\sum_{d\in n}\sum_{i\in n}\sum_{j\in n} X_{ij}^{sd} d_{ij} q_{sd} + \delta \sum_{i\in n}\sum_{j\in n}\sum_{s\in n} Y_{ij}^{s}\} \tag{1}$$

Subject to

$$\sum_{j\in n} X_{ij}^{sd} - \sum_{j\in n} X_{ij}^{sd} = 1, s, d \in N, i = s, i \neq j \tag{2}$$

$$\sum_{j\in n} X_{ij}^{sd} - \sum_{j\in n} X_{ij}^{sd} = -1, s, d \in N, j = d, i \neq j \tag{3}$$

$$\sum_{j\in n} X_{ij}^{sd} - \sum_{j\in n} X_{ij}^{sd} = 0, i, s, d \in N, i \neq s \neq d \tag{4}$$

$$\sum_{s\in n}\sum_{d\in n}\sum_{i\in n}\sum_{j\in n} X_{ij}^{sd} - \sum_{i\in n}\sum_{j\in n}\sum_{s\in n} Y_{ij}^{s} \leq 0 \tag{5}$$

Objective (1) minimizes the total cost. The first term is the cost of traffic delay, and the delay of all traffic is computed by using λ. The second term represents the cost of the flow table. In this paper, we assume these routes forwarded from the same output port with the same source node can be aggregated as a rule. Therefore, we define the sum number of entries after aggregating as the flow table cost, which is computed by using δ. Constraints (2)–(4) formulate only one wireless multi-hop transmission path between any two nodes in the network. Constraint (5) formulates the relationship between two variables.

4 Performance Evaluation

Experiments are carried out based on the topology with 11 nodes and 26 edges as shown in Fig. 2. We define the transmission delay between an arbitrary pair of nodes in Table 1. Our simulations are run in MATLAB 7.11 in an 8 GHz computer with 8 GB memory. In order to efficiently verify the performance of the proposed algorithm, we obtain the average result from randomly generated 300000 data samples which are denoted by the traffic matrix and compare the rules variation and the total cost, with the increase of the total number of flows.

In Fig. 3, we compare the total number of rules of Min Hop and ILP with the increase of the total number of flows. When the number of flow is less than 5, Min Hop algorithm is slightly better than STR, and once the number of flow surpasses 5, the average increment percentages of aggregation performance of the ILP is 40% better than Min Hop. When the number of flows is 15, the number of rules of the ILP does not change, but the number of flows of the Min Hop algorithm need to reach 50, implying that our proposed ILP is more suitable for the high performance network containing large traffic. Figure 4 shows the total cost with the increase of the total number of flows

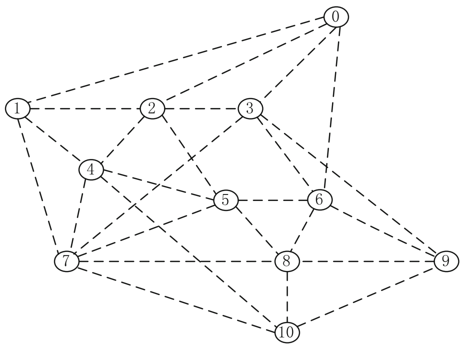

Fig. 2. A network topology G (11, 26).

Table 1. Transmission delay.

Traffic	Delay	Traffic	Delay	Traffic	Delay	Traffic	Delay
(0, 1)	131	(2, 3)	660	(4, 7)	330	(7, 8)	600
(0, 2)	760	(2, 4)	210	(4, 10)	730	(7, 10)	720
(0, 3)	390	(2, 5)	390	(5, 6)	730	(8, 9	730
(0, 6)	740	(3, 6)	340	(5, 7)	400	(8, 10)	320
(1, 2)	550	(3, 7)	109	(5, 8)	350	(9, 10)	720
(1, 4)	390	(3, 9)	660	(6, 8)	565		
(1, 7)	450	(4, 5)	220	(6, 9)	320		

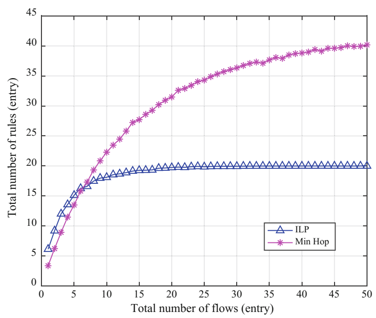

Fig. 3. The total number of rules vs the total number of flows.

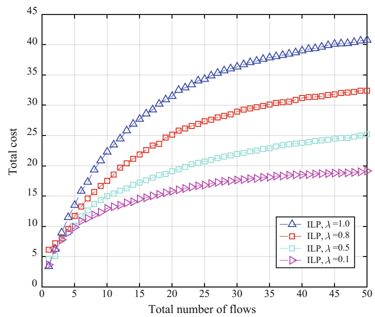

Fig. 4. The total cost vs the total number of flows under different λ.

character with different values of parameter λ and the total cost increases with the increase of parameter λ, indicating the parameter λ can adjust the total cost by the tradeoff of the delay cost and hardware cost.

5 Conclusion

We addressed the flow entry shortage issue in SD-WAN. By jointly considering flow entry aggregation and routing selection, we formulated an ILP to minimize the total cost with a proper set of parameters. Through numerical experiments, we showed that our proposed method can reduce the number of flow entries while reducing traffic delay.

Acknowledgment. This research is supported by the National Key R&D Program (Grant No. 2016YFB0201403), the Natural Science Fund of China (NSFC project No. 61701054). It is also supported by Tianjin Key Laboratory of Advanced Networking (TANK), School of Computer Science and Technology, Tianjin University, Tianjin, P. R. China.

References

1. Abujoda, A., Dietrich, D., Papadimitriou, P., et al.: Software-defined wireless access networks for internet access sharing. Comput. Netw. **93**(P2), 359–372 (2015)
2. Sarrar, N., Uhlig, S., Feldmann, A., et al.: Leveraging Zipf's law for traffic offloading. SIGCOMM Comput. Commun. Rev. **42**(1), 16–22 (2012)
3. Katta, N., Alipourfard, O., Rexford, J., et al.: Infinite cache flow in software-defined networks. In: 3rd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN), Chicago, Illinois, USA, pp. 175–180 (2014)
4. Zhang, L., Lin, R., Xu, S., et al.: AHTM: achieving efficient flow table utilization in software defined networks. In: IEEE Global Communications Conference (GLOBECOM), Austin, Texas, USA, pp. 1897–1902 (2014)
5. Zhang, L., Wang, S., Xu, S., et al.: Timeoutx: an adaptive flow table management method in software defined networks. In: IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, pp. 1–6 (2015)
6. Draves, R., King, C., Venkatachary, S., et al.: Constructing optimal IP routing tables. In: IEEE INFOCOM, New York, USA, vol. 1, pp. 88–97 (1999)
7. Uzmi, Z.A., Nebel, M., Tariq, A., et al.: Smalta: practical and near-optimal FIB aggregation. In: 7th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT), Tokyo, Japan, pp. 29:1–29:12 (2011)
8. Meiners, C., Liu, A., Torng, E.: Bit weaving: a non-prefix approach to compressing packet classifiers in TCAMs. IEEE/ACM Trans. Netw. **20**(2), 488–500 (2012)
9. Kang, N., Liu, Z., Rexford, J., et al.: Optimizing the "one big switch" abstraction in software-defined networks. In: 9th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT), Santa Barbara, California, USA, pp. 13–24 (2013)
10. Kanizo, Y., Hay, D., Keslassy, I.: Palette: distributing tables in software-defined networks. In: IEEE INFOCOM, Turin, Italy, pp. 545–549 (2013)
11. Moshref, M., Yu, M., Sharma, A., et al.: Scalable rule management for datacenters. In: 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI), Lombard, IL, USA, pp. 157–170 (2013)
12. Sheu, J.P., Chen, Y.C.: A scalable and bandwidth-efficient multicast algorithm based on segment routing in software-defined networking. In: IEEE International Conference on Communications, pp. 1–6. IEEE (2017)