2nd International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)

# Top-K Routing Algorithm Based on Bandwidth Utilization in SDN

Erwei Yea, Zhaohua Longband Ruifang Dongc

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 40065, China.

<sup>a</sup> 15310264077@163.com, <sup>b</sup> longzh@cgupt.edu.cn, <sup>c</sup> 2524668099@gg.com

**Keywords:** SDN; Top-K algorithm; Bandwidth Utilization; Load Balancing.

**Abstract.** Because of the tight coupling between the traditional network control layer and the data layer, the path allocation algorithm has poor globality and lack of real-time, and distribution of network traffic is unbalanced, which leads to network congestion. To solve the problem, according to the advantages of SDN centralized control and transparency, a Top-K routing algorithm based on Bandwidth utilization (Top-KRA-BU) is proposed. The algorithm calculates K available paths from the source nodes to the destination nodes in real time, and based on the bandwidth utilization, the K paths are evaluated, and the optimal forwarding path is selected. The experimental results show that the routing algorithm is superior to shortest path first (SPF) routing algorithm in the network bandwidth utilization.

### 1. Introduction

Software Defined Network (SDN) [1, 2] is a new Software-defined network architecture technology. The most feature of SDN is that it has loosely coupled control plane and data plane, supports centralized network control and implements the underlying network to the upper application of the transparent. The characteristics of SDN centralized control ensure the dynamic and flexibility of the traffic scheduling policy in the network, which is not available in the traditional TCP / IP network architecture [3]. However, the data flow is transmitted through the shortest path routing according to the destination address, and a large amount of traffic may flood into the shortest path leading to congestion and reducing the performance of overall network. At the same time, compared with the shortest path, there may be a number of similar or equal paths in the network. The Top-KRA-BU is proposed in this paper which finds a number of reachable routes from the source nodes to the destination nodes, and selects the route with the highest bandwidth utilization for data forwarding. The experimental results show that the routing algorithm I can improve the bandwidth utilization of the network.

### 2. Architecture

The existing SDN controllers that support the OpenFlow protocol [4], such as Floodlight, RYU, etc, can provide a basic data forwarding strategy for switches supporting the OpenFlow protocol. However, there are serious problems with this basic forwarding strategy. Namely, when there are multiple reachable paths in the network, a large amount of traffic flows into the shortest path and the other paths are idle, resulting in low overall network bandwidth utilization, and even lead to network congestion

Aiming at the above problems, this paper studies the load balancing problem of SDN from the perspective of applied research and proposes the Top-KRA-BU for SDN network. In this routing



algorithm, the RYU [5] controller is used for centralized control. The controller obtains the topology of the SDN network. When the data stream enters the OpenFlow [6] switch, the controller searches for the shortest K paths between the source node and the destination node, and the path with the highest bandwidth utilization is selected as the forwarding path. The overall design architecture based on RYU controller is as follows:

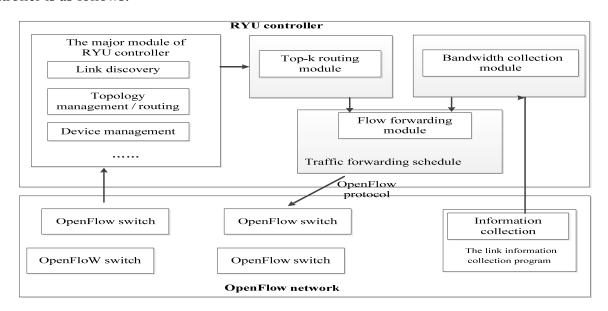


Fig. 1 Structure of RYU controller

# 3. Design and Implementation

The design of Top-KRA-BU model includes three modules: bandwidth collection module, Top-K routing module, flow forwarding module are the core of load balancing model. This article will detail the design of these three modules.

### 3.1 Bandwidth Collection Module.

To achieve full control of the OpenFlow switch, real-time bandwidth of the network link must be collected. In order to achieve this goal, the bandwidth collection program based on sFlow [7] and port sampling technology is used to run on RYU controllers and switches that supports the OpenFlow protocol. The program obtains the bandwidth of switch interface at a certain time interval, and sends the collected bandwidth to the controller.

Based on the collected interface information of the switch, the used bandwidth of the link connected with the interface is calculated, and the formula is as follows:

$$BandWidth_{node} = \frac{(\Delta i finOctets + \Delta i foutOctets)*8}{T}$$
 (1)

 $BandWidth_{ifName}$  is the used bandwidth in T time.  $\Delta ifinOctets$  is the number of bytes received by the interface in T time.  $\Delta ifoutOctets$  is the number of bytes sended by the interface in T time. The maximum remaining capacity of a path is the smallest remaining capacity of all the nodes in the path, that is, for each complete  $Path = \{node_1, node_2, \dots, node_n\}$ , The remaining bandwidth of the link is calculated as equation 2,  $BW_{nodei}^{total}$  represents the total bandwidth,  $BW_{nodei}^{used}$  represents the node's used bandwidth.



$$BW_{path} = \min\{BW_{nodei}^{total} - BW_{node_i}^{used} \mid 1 \le i \le n\}$$
(2)

The bandwidth utilization of the path is defined as the ratio of the bandwidth to the maximum bandwidth in the path. It can calculate as equation 3.

$$\eta(t) = \frac{BW_{path}}{BW_{total-bandwidth}} *100\%$$
 (3)

### 3.2 Design of Top-K Routing Algorithm.

The Top-K routing algorithm is a K-shortest path algorithm which is improved and implemented on the basis of MPS [8,9] algorithm. In finding the shortest path, the length of the link is a constraint, and the minimization of the cost of the path is used as the cost of the path. The algorithm is described in detail as follows:

```
u lti-Pahts()
 # p a t h s
                s e t s
            \pi_{\ j} - \pi_{\ i} + c _{i\,j} //a n y ( i , j ) \in
          shortestpath from s to t;
      \leftarrow \{ p_{\perp} \} ; k \leftarrow 0 ;  h \ i \ l \ e \ ( \ X \neq \varnothing \quad \& \quad \& \quad k \leq K \ ) \ d \ o 
                path in X such that \overline{c}(p) is m in im um;
               X - { p }
      if(loopless(p)) then
           p_k \leftarrow p
         hile ((v_i \neq t) & & loopless(sub_p(s,v_i))) do
                  s\ u\ b_{p}\ (s\ ,v_{i}\ )
            ndw hile
          if(v_i is the tail node of a_n)
              v_{\ j} \ \leftarrow \ \ h \ e \ a \ d \ \ n \ o \ d \ e \ \ o \ f \ \ a_{\ n}
                        m in (\overline{c}_{jt}); //shortest path from v_j to taccording to
                       X \ \cup \ ( \ p_{-s\, v} \ \infty \ a_{-n} \ \infty \ p_{-j\, t} \ ) ;
        endw hile
      endw hile
   return paths
```

Fig. 2 Pseudo code of Top-K routing algorithm

#### 3.3 Flow Forwarding Module.

The flow forwarding module needs to formulate the forwarding policy dynamically. Therefore, when the new flow reaches the OpenFlow switch, the RYU controller will select the path with highest bandwidth utilization for forwarding based on path.

When a new flow arrives at the OpenFlow switch, the OpenFlow switch encapsulates the data flow into the Packet-in message and sends it to the controller. The controller completes the processing of the new data flow through the following steps:

- (1) resolve the source address and the destination address of the packet;
- (2) Find the OpenFlow switch directly connected to it based on the source address and destination address:
  - (3) According to the network topology and the link cost, calculate K available paths of switches:
- (4) According to the statistical data collected by the link parameter collecting module, the comprehensive score of k paths is calculated, and the path with the highest score is selected and allocated to the data flow for use:
  - (5) Create a flow rule and send it to all switches on the path.



# 4. Experiment and Analysis of Result

### 4.1 Experimental Environment.

All experiments in the paper are deployed on mininet 2.0 and RYU controllers. Two win 7 computers are the hardware experimental platform. RYU controllers is deployed on one Win 7 computer, another Win 7 computer installs Linux system and installs Mininet 2.0 open source software.

# 4.2 Experimental Topology.

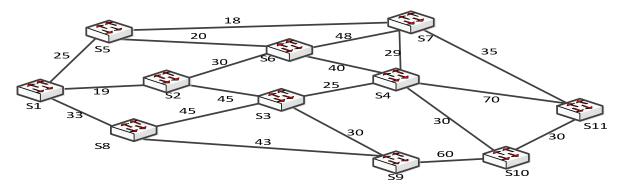


Fig.3 Topology of testing load balancing model performance

There are 11 switches in the experimental Topology 2. Each switch is connected to a host (not shown in the figure). The number on the links represents the cost of the links.

### 4.3 Analysis of Experimental Result.

Experimental topological is used to verify the overall performance of Top-KRA-BU, reflecting the performance of the routing algorithm through average bandwidth utilization and comparing with SPF.

Fig. 4 is the comparison graph of the SPF and Top-KRA-BU model in bandwidth utilization. The parameters are set as follows: The bandwidth of the link is 10 Mbps. K is set to 5, the frequency of data acquisition is 3 s.

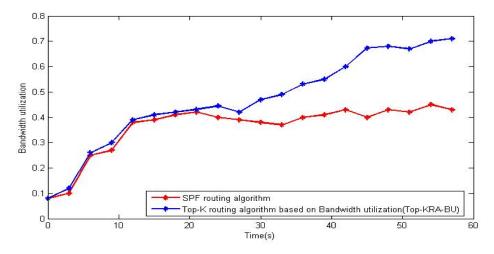


Fig. 4 The comparison graph of the SPF and Top-KRA-BU model

It can be seen from Figure 4 that the bandwidth utilization of Top-KRA-BU model is higher than the bandwidth utilization of SPF. In the initial stage, because the traffic in the links are not very large, the bandwidth utilization of the two algorithms is not much different. After 10s, as a large amount of flow is injected into the network links (not exceeding the maximum bandwidth), the bandwidth utilization of Top-K routing algorithm model is stable between 60% and 70%, while the bandwidth utilization of SPF is stable between 30% and 40%. Therefore, the Top-KRA-BU model in this paper is superior to the SPF routing algorithm in bandwidth utilization.



### 5. Conclusion

In this paper, based on the characteristics of SDN networks, the Top-KRA-BU is proposed. Experiments show that the routing algorithm of load balancing can maximize the use of network bandwidth and improve the overall network performance. This routing algorithm can be used as an internal module of the SDN controller or as an API for the SDN controller. At the same time, it is suitable for a variety of network topologies and is especially suitable for SDN data center network topologies.

#### References

- [1]. Zhang CK, Cui Y, Tang HY, Wu JP. State-of-the-Art survey on software-defined networking (SDN). Ruan Jian Xue Bao/Journal of Software, 2015, 26(1):62–81 (in Chinese).
- [2]. Lin Li and Qiaozhi Xu. Load Balancing Researches in SDN: A Survey. IEEE International Conference on Electronics Information and Emergency Communication, 2017(7), 403-408.
- [3]. ZUO Qing-Yun, CHEN Ming, ZHAO Guang-Song, et.al. OpenFlow-based SDN Technologies[J]. Journal of Software, 2013,24(5):1078-1097.
- [4]. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, and Rexford J, et al. OpenFlow: enabling innovation in campus networks [J]. ACM SIGCOMM Computer Communication Review. 2008, 38(2):69-74.
- [5]. RYU.[EB/OL].[2017-9-20].http://ryu.readthedocs.io/en/latest/.
- [6]. OpenFlow Switch Specication Version 1.6 [EB/OL]. https://www.opennetworking.org/sdn-resources/onf-specifications; Sept.2016.
- [7]. sFlow-RT.[EB/OL].[2014-9-27].http://www.inmon.com/products/sFlow-RT.php.
- [8]. Mao Shao-wu,Zhang Huan-guo,Huang Chong-chao,Wu Wang-Chun.A New Fault-Tolerance Mechanism in Communications Based on K Shortest Path Algorithm[J]. Journal of Wuhan University (Natural Science Edition).2013(6):534-8.
- [9]. ZHAO meng-ya, LONG zhao-hua, JIANG guiquan, WANG qi, QIN xiaohuan.Research of load balancing mechanism based on OpenFlow. Computer Engineering and Design.2015, 36(9).