

## Improvement in Multi-domains Routing Model Based on SDN

Huihui Ma<sup>1, a</sup>, Yu Lu<sup>2, b</sup>

<sup>1</sup> Department of Information Engineering, Ordnance Engineering College Shijiazhuang, Hebei, China

<sup>2</sup> Department of Information Engineering, Ordnance Engineering College Shijiazhuang, Hebei, China

<sup>a</sup>email: zhiqiuyiye92@163.com, <sup>b</sup>email: 15832169209@126.com.

**Keywords:** Software defined networks; multi-domains; topology; routing.

**Abstract.** There are some defects in the typical SDN network with centralized control. When the network structure is very complex, there are a lot of switches in the network, and the data processing pressure of the SDN controller is very high. In order to solve this problem, a multi-domains SDN topology are designed. Setting up a separate controller into each domain reduces the amount of data processing of a single controller. The problem of negotiation among each domain controllers is research are solved. In order to reduce the frequent interaction between the control layer and the infrastructure layer, pre-rooting is used to divide the partial controller function of the controller to the SDN switches. The new construction method of SDN data transmission path can effectively reduce the message processing time and improve the data transmission speed.

### Introduction

In the typical SDN routing mechanism, the central controller masters the topology information of the whole network [1-3]. When a new data flow arrives at the network, the controller calculates the optimal path of the whole network and sends the flow table to the SDN switch. The typical SDN has the following drawbacks: When the network size is getting bigger, the route of the data flow needs to go through more switches. The network node changes need the controller to regain the network topology. The interaction between the controller and the SDN switches (OpenFlow switches) increases linearly with the increase of the number of SDN switches [4]. Large-scale networks increase the information processing pressure on the switches. In view of the above flaws, there have been some corresponding solutions proposed a multi-domains SDN structure [5]. A large-scale SDN network contains multiple domains and each domain contains a controller responsible for the management of the switches within the domain. The controller controls the data flow within the domain routing. There is also a center controller which is responsible for managing all the controllers in the entire network. The interaction between the controllers in each domain is also dependent on the central controller. The cross-domain data flows are calculated by the central controller, and the controllers in the domains are only issued Flow table. The scheme similarly divides the SDN into a multi-domains network [6]. But there is no central controller in the network. Data flow cross-domain routing is done by intra-domain controller negotiation. In this paper, the above method is used as a reference to achieve the simplification and speed improvement of the routing process. The method is improved and a simplified SDN routing mechanism is designed.

### Architecture of Multi-domains SDN

The entire SDN is divided into multiple domains. There is one border switch in a domain and the rest of the switches are normal switches. All normal switches can be connected directly or indirectly to the border switch. The domain and domain are connected through the border switch. The border switches become the skeleton of the entire network. Each domain contains an intra-domain controller that is responsible for routing and streaming of data streams in the domain. All border switches in the

network are controlled by a central controller. Here is a five-domain SDN network, only draws the SDN switch, no display controller connection. Solid circles represent normal switches, and open circles represent border switches (Fig. 1).

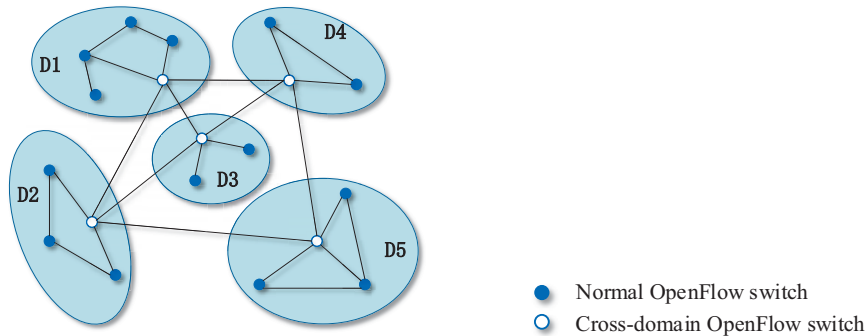


Fig. 1: Division of the network domain

### Packet Routing Type

Packet forwarding includes direct forwarding and indirect forwarding. Direct forwarding refers to the source node and the destination node in the same physical network. The message from the source node can reach the destination node directly through the physical link, and does not need to go through other nodes. If the nodes that need to forward packets and destination nodes are not in the same physical network, the forwarding between them is called indirect forwarding. At this point, there will be one or more intermediate nodes between the source node and the destination node. When the packet forwarding needs to pass through multiple intermediate nodes, the forwarding path selection problem is involved. According to the SDN architecture design, the route of the packet is divided into intra-domain routing and cross-domain routing. The following is description of direct forwarding, intra-domain routing and cross-domain routing features.

**Direct Forwarding.** In the direct forwarding the source host is connected to the destination switch on the same switch. The data transmission only need to pass through the one SDN switch. The packet can be forwarded directly through the connected switch (Fig. 2). In the direct forwarding, the domain switches and border switches can not reflect the difference.

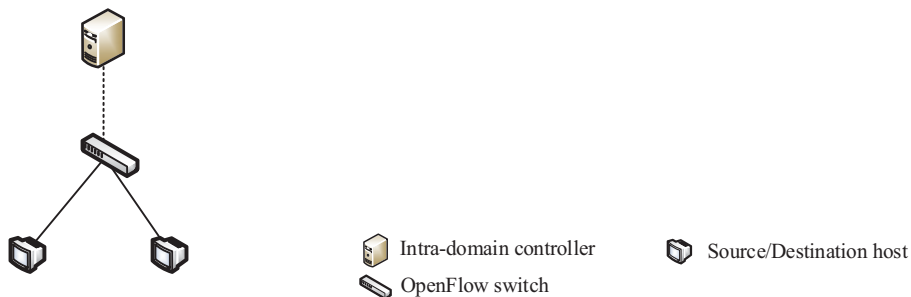


Fig. 2: Direct forwarding

**Intra-domain Routing.** Intra-domain routing refers to the source host and destination host is in the same region, and the data stream need to go through a number of intermediate nodes (Fig. 3). The nodes is in the same region, so packet forwarding can be completed within a domain. In the intra-domain routing, the data flow does not have to go through the border switch. The border switch have the same effect as the normal switch and can be used as an normal switch to complete the routing function.

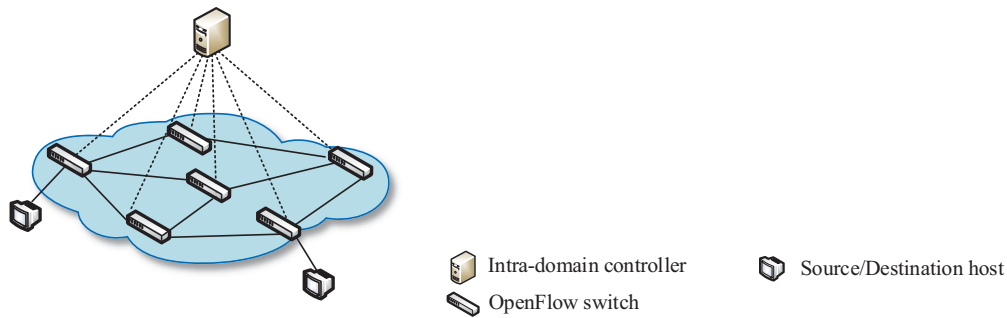


Fig. 3: Intra-domain routing

**Cross-domain routing.** The source host and the destination host are in different domains, and the data flow must flow through the border switch to be forwarded to other domains. As shown (Fig. 4), the border switch belongs to the domain, and controlled by the domain controller. The border switch also forms a separate network with other border switches in the network, which is controlled by a central controller. In fact, in large-scale networks, most data streams are cross-domain routing [7].

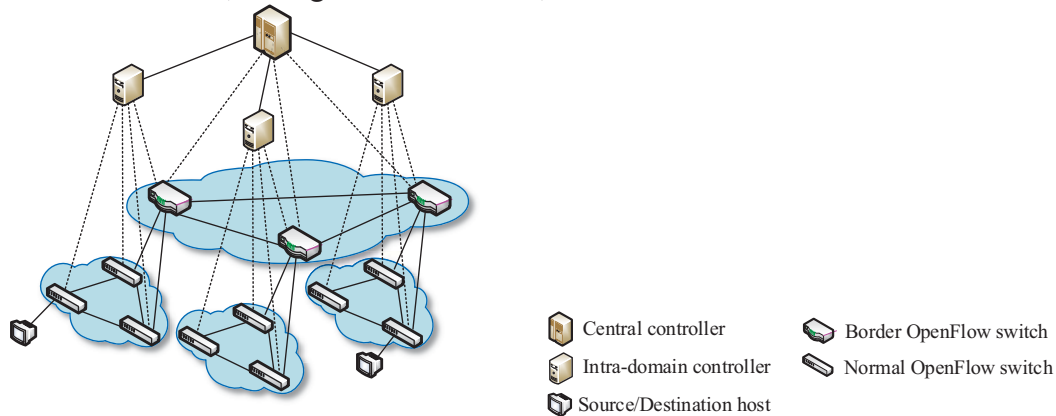


Fig. 4: Cross-domain routing

## Routing Process

**Initialization.** The main goal of the multi-domains SDN routing is to simplify the routing process and implement the fast forwarding of packets. The part control functions of the controller down to the switch reduces the information exchange between the controllers and the switches, which can effectively improve the packet forwarding speed. The process of initializing is to pre-acquire and store the information needed in the routing process, mainly including the following two aspects:

1) Intra-domain pre-routing: In cross-domain routing, the switches in the same domain need to forward the message to the border switch firstly to other domain forwarding. In this process, all the normal switches have the same destination, the border switch of the domain. So the optimal path of all the normal switches to the border switch in one domain can be calculated the to form a flow table stored in the domain switch. In this way, when forwarding cross-domain routing, the forwarding of packets in the source domain can reduce the number of accesses to the controller, thus the speed of data transmission can be increased.

2) The information exchange of the domain controller and the central controller: The domain controller can be informed the information of all the switches in this domain, including normal switches and a border switch. The controllers in each domain summarize the normal switches and border switches that are owned by the domain to form packets to the central controller, so that the central controller mastered the correspondence between all normal switches and border switches. The central controller stores the correspondence in table so that it is queried during routing.

As an example, Figure 1 shows the network topology. After initialization, the central controller only need to maintain the network topology of the border switch (Fig. 5), coupled with the

Correspondence table of the border switch with the domain. The central switch control the entire network can be achieved. The simplification of the network topology reduces the computational complexity of the controller in calculating the optimal routing path, which has certain advantages for improving the speed of the packets forwarding.

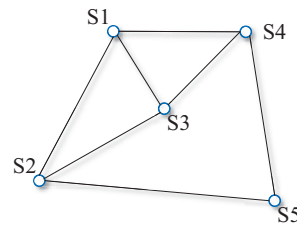


Fig. 5: Boundary switch topology

**Message Forwarding.** When a new data flow is generated in the network, there is no flow entry in the switch that matches the data flow. The switch directly connected to the source host resolves the packet and extracts the packet header. If the source host and the destination host are connected to the same switch and belong to the direct forwarding, the switch will directly establish the channel on the physical link according to the flow table sent by the switch. If the source host and the destination host in the same domain, that is, the source switch and the destination switch in the same controller, belonging to the domain routing, the switch first caches the message and generates a packet\_in message to the controller [8]. Based on the network topology, the controller calculates the optimal path by using the routing algorithm, and sends the flow table to the switches on the optimal path to complete the routing of the message.

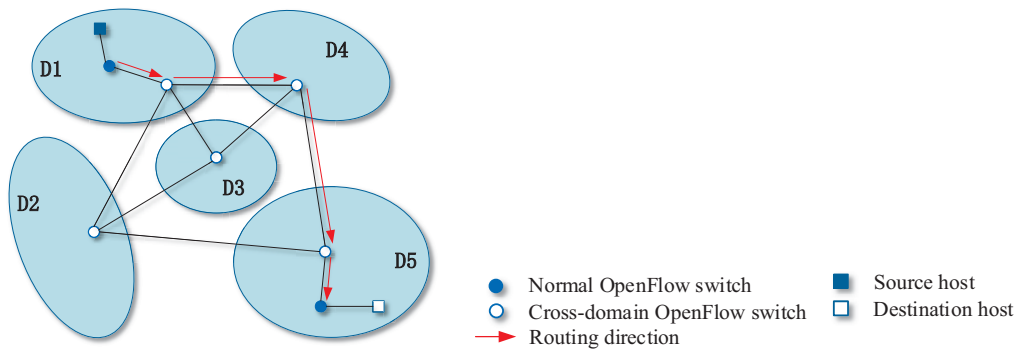


Fig. 6: Cross - domain routing process

When the source host and the destination host is located in the same domain, the message needs to be cross-domain routing to achieve forwarding. Cross-domain routing involves the following three steps: routing in source domain, inter-domain routing, routing in destination domain. The cross-domain routing packets need to be sent from the intra-domain switch to the border switch of the domain, forward through the border switch, and finally reach the border switch of the destination host. The border switch of the destination domain forwards the packet to the intra-domain switch and finally reaches the destination switch to forward the packet. The specific process is as follows (Fig. 6):

1) Source-domain routing: After receiving the new message and determining the cross-domain route, the switch connected to the source host forwards the packet in the domain according to the best path in the pre-route. The routing process does not require the request controller. Finally the message is forwarded to the border switch of the domain.

2) Inter-domain route: The border switch changes the destination address of the packet to the ID of the destination boundary switch. However, the central controller is not connected to the final destination switch. Therefore, it is not possible to know directly which border switch connected to the destination switch. However, after the initialization, the central controller has been cached the correspondence of each domain border switch and the switch within the domain, so it can get to know the domain switch ID. The source domain switch cache this packet and generates a packet\_in message to the central controller. The central controller calculates the optimal path based on the

network topology of the border switch, and sends the flow table to all the border switches on the path. So the message can be routed to the border switch in the destination domain. At the last route, the destination address of the packet is changed to the original destination address.

3) Destination route: After the packet reaches the destination domain, it performs intra-domain routing. The border switch forwards the packet to the intra-domain switch. The domain controller in the destination calculates the optimal path according to the topology of the domain and send the flow table to implement the forwarding function.

## Summery

SDN solves the problem of the traditional network equipment curing function. SDN has obvious advantages over traditional networks. But when the network size is large, SDN also has some defects. The increase of network equipment makes a large increase in control information, so the controller information processing capability has been brought big challenge. In this paper, a distributed management architecture for equipment deployment is introduced. The network devices of each domain have a separate controller, which simplifies the network topology and reduces the information processing burden of the controller. The scheme has a hierarchical processing to the network equipment layer [9]. The border switch for each domain is connected to the normal switches in the same domain, and is also the connection between domain and domain. The switches is clear and easy to manage. Frequent control information transfer between switches and controllers is an important factor limiting of network transmission performance. Pre-routing can reduce the generation and transmission of control information to a certain extent [10]. The increase, decrease, and movement of network devices have a certain impact on the centralized management of the network. Too frequent equipment changes increases the workload of the controller. Multi-domains network structure avoid a wide range of data updates. It can realize the link data updating of the whole network by the local update, so it has good network expansibility.

## References

- [1] Siamak Azodolmolky. Software Defined Networking with OpenFlow[M]. China Machine Press, 2014:44-64.
- [2] Greene K, TR10. Software-Defined Networking, MIT Technology Review, 2009.
- [3] Kreutz D, Ramos FMV, Verissimo P, Azodolmolky S, and Uhlig S. Software-Defined Networking: A Comprehensive Survey[J]. In Proceedig of the IEEE, 2014, 103(1): 14-76.
- [4] Fernandez MP. Comparing OpenFlow Controller Paradigms Scalability: Reactive and Proactive[C]. IEEE Advanced Information Networking and Applications, 2013: 1009-1016.
- [5] Phan XT, Thoai N and Kuonen P. A collaborative model for routing in multi-domains OpenFlow networks[C]. In IEEE ComManTel'13, 2013: 278-238.
- [6] Egilmaz, Hilmi E, Seyhan Civanlar, et al. Adistributed Qos routing architecture for scalable video streaming over multi-domain OpenFlow networks[C]. In IEEE ICIP'13, 2012: 2237-2240.
- [7] Lin P, Bi J, Hu H. ASIC: an architecture for scalable intra-domain control in openflow[C]. In Proceedings of the 7th International Conference on Future Internet Technologies, 2012: 21-26.
- [8] OpenFlow switch specification Version 1.4.0. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>.
- [9] Kleinrock L, Kamoun F. Hierarchical routing for large networks performance evaluation and optimization[J]. Computer Networks, 1977: 155-174.
- [10]Curtis, Andrew R, Tourrilhes JC, Yalagandula J, Sharma P and Banerjee S. DevoFlow: Scaling

flow management for high-performance networks[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 254-265.