

OpenRouteFlow: Enable Legacy Router as a Software-Defined Routing Service for Hybrid SDN

Tao Feng, Jun Bi

Institute for Network Sciences and Cyberspace, Tsinghua University
Department of Computer Science and Technology, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology
fengt09@mails.tsinghua.edu.cn, junbi@tsinghua.edu.cn

Abstract—Hybrid SDN is an important direction for the evolution of SDN. It is one of the challenges to be addressed for hybrid SDN to open the legacy router and enable the routing view as a software defined routing service. In this paper, OpenRouteFlow is presented, which opens legacy routing protocols and flow sampling information as a routing view service for network applications by software update. OpenRouteFlow tries to realize the decoupling of network visualization and network control. It provides path-oriented and traffic-oriented subscription and publication services for different scenarios of network control. In a hybrid network consisted of OpenFlow switches and legacy routers, OpenRouteFlow is able to map OpenFlow control rules into ACL or RIB in order to support flexible software defined capacities in legacy routers.

Keywords—legacy router; hybrid SDN; OpenFlow; routing service

I. INTRODUCTION

In the enterprise network, access switches form the access networks, the access networks after being connected with three-layer switches or routers form the bearer network and the bearer network and access network jointly form the wide area network. In the access network, the two-layer switching is adopted to realize the host interconnection. OpenFlow switch is adopted to replace the existing legacy two-layer switch, which is easy from the view of the migration cost and maintenance work[1]. In addition, after use of OpenFlow switch for networking, the centralized control mode can be used for forwarding control of switch in the access network to realize the network control of filtration, load balancing and fine grit. However, in the bearer network consisted of three-layer switch or router, the use of OpenFlow can change the operation mechanism of current network routing protocol, leading to the reconfiguration of networking structure and increase of management cost and bringing unstable risk to the current network services. Therefore, under the conditions of maintaining the stability of the existing network service and reducing the upgrading and maintenance cost, it is an important direction for SDN network evolution to realize the coexistence of legacy routing control plane and SDN control plane by the routing protocol of OpenRouter. The hybrid SDN in which the OpenFlow control and legacy routing control planes coexist has the following advantages: firstly, the hybrid SDN network

supporting SDN and legacy routing protocol improves the flexibility of the route compared with the pure IGP network[2]; secondly, it has the ability of deploying the advanced network (such as anti-spoofing and traffic engineering)[3]; thirdly, it combines the flexibility of SDN and the expansibility and robustness of IGP[4]; fourth, it can realize the smooth migration with pure SDN deployment[5].

However, to get the advantages of hybrid SDN, we need to solve some challenges: Firstly, lack of unified routing view and control protocol. In OpenFlow 1.3 [6], hybrid OpenFlow is presented and suggested to adopt ship in the night, which separates the OpenFlow flow and legacy network flow by VLAN or a port. However, there is still distributed routing protocol in the router and the goal of building the router-centered hybrid SDN is to how to realize the coordinated interaction between robust routing control plane and SDN control plane. Secondly, considering the deployment cost, the network operator is not likely to exchange the existing router hardware to support SDN. How to provide software-based network control ability by exposing the route information and supporting OpenFlow protocol based on the low cost of software upgrading is a feasible way of OpenRouter. Thirdly, the network status information of all flows cannot be transmitted due to the transmission bandwidth limit of the control channel of an existing router. At the same time, all the flows cannot be cached in a router due to TCAM hardware capacity [7]. How to get a balance between network visualization and network control ability is an important problem to be considered in the practical application of hybrid SDN.

In this paper, OpenRouteFlow is presented, which does not only maintain the legacy distributed routing networking mode, but also opens the routing information and flow information in the router to form the associated view of route and flow view in order to realize the decoupling of network visualization and network control. The architecture of OpenRouteFlow is designed in Section II. Section III depicts how to expose route status in a legacy router. The design of OpenRouteFlow controller and the calculation of route view are presented in Section IV.

II. OPENROUTEFLOW ARCHITECTURE

address information in bulk. TVL produced by the upgrading information of the three status can be filled into a same RouteStatusBuffer mixedly. OpenRouteFlow module opens a SampleBuffer (default is 1300 bytes) for the Client Socket, for transmitting the packets of sampling information in bulk. OpenRouteFlow module establishes an ORF-ACL table for the Client Socket, which is empty during initialization. When the listening socket on OpenRouteFlow is disconnected with the corresponding TCP of the Client Socket, ACL rules issued by this TCP can only be cleared.

2) *Route status information for synchronization with ORFC*

When OpenRouter first establishes connection with ORFC, OpenRouteFlow Agent module handles ORF-INTERFACE table, ORF-INTERFACE-ADDRESS table and ORF-INTERFACE-FIB table in bulk. If the number of the entries in the table is more than zero, the entries will be packaged into TLV format and then transmitted to RouteStatusBuffer. When RouteStatusBuffer is filled or the timer is overtime, these TLVs will be transmitted to ORFC as a ROUTE_STATUS message and RouteStatusBuffer will be cleared. It repeats until the contents of local information tables are transmitted; in the process, OpenRouteFlow Agent module will stop handling the upgrading information of RTM.

B. *Push Model of Status Information Release*

1) *Route status advertisement*

OpenRouteFlow module registers the concerned event type in the RTM module through the registration mechanism. OpenRouteFlow module receives the routing, interface or interface address information transmitted by RTM. In the TCP connection established by OpenRouteFlow module and ORFC, OpenRouteFlow module packages the information into TLV format and then sends it to the corresponding RouteStatusBuffer of each TCP connection. If ORFC and OpenRouteFlow are disconnected, TLV is not transmitted. OpenRouteFlow module stores the information at local ORF-INTERFACE table, ORF-INTERFACE-ADDRESS and ORF-INTERFACE-FIB table or deletes the information from the tables according to the add or delete tag.

2) *Sampling information advertisement*

After receiving the sampling packets from sFlow module, OpenRouteFlow module analyzes the field information defined in the structural body of sampling message, including the access and port index, source IP and destination IP of sampling packets, partial sampling packets (optional according to the users' needs) packages the information into the TVL structure of sampling information and then transmits the information to ORFC through a ORF_SAMPLE_PACKET message.

3) *Information Push associated with Timer and Buffer*

As the frequent transmission of route status information and flow information will lead to the blocking of the control channel between router and controller, an information push mechanism associated with the timer and buffer zone is designed. After receiving TLV by RouteStatusBuffer, it is required to check whether the corresponding latency timer of RouteStatusBuffer is started; if no, it should be started. If the sum of the used capacity of RouteStatusBuffer and size of

received TLV is more than the total capacity of RouteStatusBuffer, TLV in RouteStatusBuffer is transmitted to ORFC after being packaged into a ORF_ROUTE_STATUS message and the RouteStatusBuffer is cleared; then, the received TLV is transmitted to RouteStatusBuffer and the RouteStatusBuffer timer is reset; otherwise, TLV is directly pushed into the buffer zone, without transmission, and the RouteStatusBuffer timer is also reset. After receiving the data, RouteStatusBuffer should transmit the data within maximum seconds, which is determined by the user. When the time of RouteStatusBuffer timer is up, the timer is stopped, TLV in the RouteStatusBuffer is transmitted to ORFC after being packaged into an ORF_ROUTE_STATUS message and RouteStatusBuffer is cleared.

C. *Subscription of Interest Flow*

When all the flows statistic information are retransmitted to the controller, it will lead to blocking of control channel and package loss. In sFlow Agent, the interest flow can be configured in order to reduce the message load of control channel. For the path-oriented flow subscription, packets sampling function of sFlow in the router of relevant path is started to transmit the head information of packets to the path calculation module. The path calculation module can calculate the forwarding path information of the message according to the routing view and transmit it to the required control application. For the traffic-oriented flow subscription, sFlow packets statistics function in the router of relevant path is started to give a notification to the required control application about the flow larger than the threshold value by monitoring the Top-k of each flow.

D. *Forwarding Control Compatible to The Existing Hardware*

In the existing router, forwarding control is realized by two types of hardware: forwarding table and filtering table. In the hardware realization of router, the forwarding table and filtering table are realized mainly by TCAM. Due to the cost and power consumption of TCAM, the forwarding table adopts the one-dimensional TCAM to accommodate the large number of routing rules and the filtering table adopts the multi-dimensional TCAM to support the flow-oriented filtering, QoS and redirect and other advanced flow control operation. To give full play to the advantages of the two types of forwarding hardware, OpenRouteFlow provides OpenFlow protocol extension-based routing control information and ACL control information.

1) *Route control*

The external routing control application can transmit routing rules to OpenRouter through OpenRouteFlow protocol with Route_Mod messages. A Route_Mod message is consisted of router ID (consistent with OpenFlow protocol, adopt the MAC address of interface from router to controller), destination address, next-hop information and router priority. By analyzing the information, OpenRouter transmits the value in the match field and action field of routing rule information to RTM module and then RTM transmits the user defined rule and current dynamic routing protocol rule to FIB table after selection.

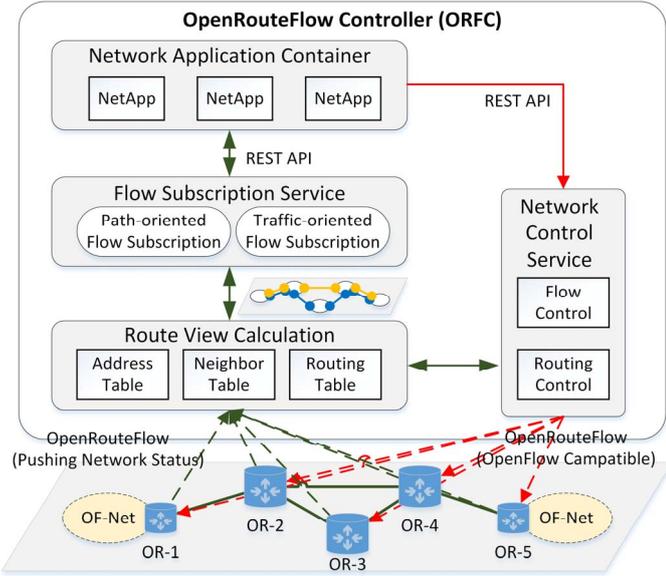


Figure 3. OpenRouteFlow controller architecture

2) ACL&QoS control

To be compatible to flow processing in OpenFlow protocol, the support for Flow_Mod information in OpenFlow protocol is retained in OpenRouteFlow protocol. After receiving a Flow_Mod message, OpenRouteFlow will analyze the match field and action field in the information according to the OpenFlow standard. According to the different value of action field, the match field is mapped into ACL and QoS operation of router. For example, the multi-dimension match field analysis in OpenFlow is mapped into the match field of ACL and the drop operation of OpenFlow is mapped into the deny operation of ACL.

IV. DESIGN OF OPENROUTEFLOW CONTROLLER

An OpenRouteFlow controller consists of route view calculation, flow subscription service and network control service, as shown in Fig. 3. The module of route view calculation receive route status information and flow statistic information from OpenRouter in order to generate route view. Network application can subscribe route view by means of path-oriented flow subscription and traffic-oriented flow subscription with REST APIs. Meanwhile, network application can control routing policies and flow process in OpenRouter with network control service.

A. Calculation for Route-Flow View

In the actual network, the precondition of network control is the accurate and timely network topography and network state awareness. Therefore, the network view is divided into three levels: connection view, routing view and flow view, as shown in Fig. 4. The connection view shows the link relationship between network devices, which is the fundamental network view of IP network. The routing view shows the message forwarding relationship between routers running the routing protocol, which is a logical view. The route-flow view, based on the routing view, shows the

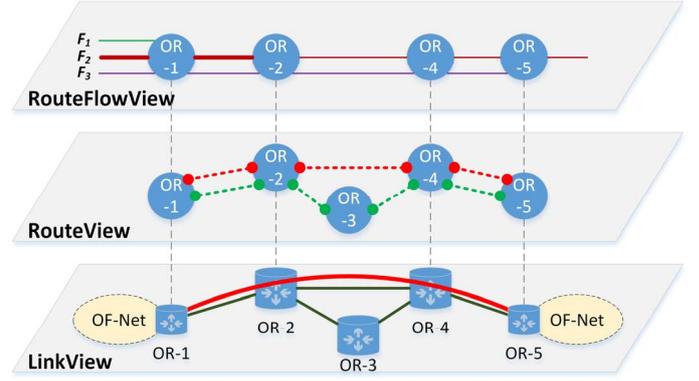


Figure 4. Route-flow view

application services on the routing path, which is a view that the flow is overlapped on the route.

1) Initial conditions

In the current routers, there are three types of basic information for matching the destination address of message and generating the next hop of route: interface, interface address and routing table.

The interface can be a physical port or logical port, such as VLAN interface.

$$IF^r := [if_1^r, \dots, if_i^r, \dots, if_n^r]$$

Where, r is a router in the network. When the message is retransmitted through router, the interface is divided into outbound interface and inbound interface. The inbound interface of r is defined as $in_if_i^r$ and the outbound interface is defined as $out_if_i^r$.

The interface address is used to find the interconnecting path of two routers upon forwarding and addressing when an interface is configured with IP address.

$$IF_IPADDR^r := [(if_1^r, ipaddr_1^r), \dots, (if_i^r, ipaddr_i^r), \dots, (if_n^r, ipaddr_n^r)]$$

The routing table is the set of routing rules generated by the active routing protocol in the router, which is used for deciding the packets forwarding.

$$ROUTE_TALBE^r := [ROUTE_1^r, \dots, ROUTE_i^r, \dots, ROUTE_n^r]$$

The routing rules are consisted of prefix and next hop of IP address.

$$ROUTE^r := [(ip_prefix_1^r, nexthop_1^r), \dots, (ip_prefix_n^r, nexthop_n^r)]$$

Where, the next hop of route $nexthop_i^r$ is consisted of local outbound interface and router interface address:

$$nexthop^r := (out_if_i^r, ipaddr_j^{r+1})$$

2) Calculating route view

When the interface, interface address and listing table are known, the neighbor table is calculated. The neighbor table of a router consists of the number of the router, the outbound

interface of the router, the number of the neighbor router and the inbound interface of the neighbor router.

$$NBR_TABLE^r := \left[(r, out_if^r, nbr, in_if^{nbr}), \dots, (r, out_if^n, nbr, in_if^{nbr}) \right]$$

The neighbor table is mainly used to get the process of inbound interface from a router to the next-hop router. For the calculation of inbound interface, that is, inbound interface from the last-hop router to the next-hop router, multiple interface of a router may use the same local link address when the next-hop address is the local link address of interface, so the next-hop address can not distinguish the inbound interface. Therefore, an indirect method can be adopted: the inbound interface from a router to the next-hop router can be obtained by inquiring the neighbor table of the router.

Theorem 1 When the path between directly connected routers is asymmetric, the inbound interface from router r_1 to router r_2 is consistent with the outbound interface from router r_2 to router r_1 .

According to Theorem 1, the establishment of neighbor table is divided into two steps: first, in the routing table, the number of neighbor router can be obtained from the next-hop address of all entries and the outbound interface from the router to the neighbor router can be obtained according to the outbound interface data of each routing entry. Second, in the existing incomplete neighbor table of router r (neighbor router number nbr to the outbound interface of the neighbor out_if^r , lacking of the inbound interface to the neighbor router in_if^{nbr}), for each entry, the entry of neighbor router r is found in the neighbor table of nbr router, the entry of out_if^r is the entry of in_if^{nbr} while the entry of in_if^{nbr} is the entry of out_if^r . The algorithm of calculating neighbor table of a router is shown in Algorithm 1.

B. Subscription Service of Interest Flow

After formation of routing view, the external application can realize the interest flow subscription by using the route-flow view provided by OpenRouteFlow. OpenRouteFlow provides two kinds of interest flow subscription services: path-related flow subscription and flow-related flow subscription.

1) Path-oriented flow subscription

The path-oriented flow subscription service refers to the path returning to the flow when given a flow. When a packet enters a router from an inbound interface of router, the router finds the next information of routing table according to the destination address of the packet and then conduct forwarding. The input of path-related flow subscription is (inbound interface and destination address of the packet). The forwarding path of the packet can be obtained by routing view calculation:

$$PATH := src_ipaddr \rightarrow (in_if^1)nextHop^1 \rightarrow (in_if^i)nextHop^i \rightarrow (in_if^n)nextHop^n$$

Algorithm 1 Calculate Neighbour Table of A Router

```

1: procedure CALCULATENBR(ROUTE_TABLE)
2:   for all ROUTE  $\in$  ROUTE_TABLE[r] do
3:     NBR_TABLE.r_nbr  $\leftarrow$  ROUTE.nextHop.addrir+1
4:     NBR_TABLE.ifoutr  $\leftarrow$  ROUTE.nextHop.ifir
5:   end for
6:   for all ENTRYr  $\in$  NBR_TABLE[r] do
7:     r_tmp  $\leftarrow$  ENTRY.r_nbr
8:     for all ENTRYr_tmp  $\in$  NBR_TABLE[r_tmp] do
9:       ENTRYr.ifinr_nbr  $\leftarrow$  ENTRYr_tmp.ifoutr_tmp
10:      ENTRYr.ifoutr_nbr  $\leftarrow$  ENTRYr_tmp.ifinr_tmp
11:    end for
12:  end for
13: end procedure

```

A forwarding path can be formed on each middle-hop router through these path information and the specific forwarding path calculation is as follows:

Step 1: in the routing table of all routers, for each routing entry $ROUTE^r$: if the entry type is “direct connection”, it means that the prefix of the entry $ROUTE^r$ *ip_prefix* is the prefix of the local sub-net; if the entry type is “indirect connection”, inquire the number of next-hop router by the next-hop address in the entry $ROUTE^r$ *nextHop_ipaddr* (address of an interface of the next-hop router), record the current router number r , next-hop router number nbr and the prefix of the entry.

Step 2: know the source router number r , next-hop router number nbr and destination prefix $ipaddr_{dst}$. Find the corresponding router of nbr , find the entry with $ipaddr_{dst}$ as the destination prefix in the routing table of the router: if the entry type is “direct connection”, it refers to arrive at the destination; find the complete path, add a temporary entry in the subscription table of the current router, in which *in_if* means the inbound interface from the last-hop router to the current router (router represented by nbr) with the neighbor table; if the entry type is “indirect connection”, add a permit temporary entry in the filtration entry of the current router; inquire the next router number through the next-hop address in the entry (address of an interface of the next-hop router), record the current router number, next-hop router number and the prefix of the entry; then repeat *step 2*.

Finally, if the temporary filtration entries generated in *step 2* are to be converted into formal filtration entries, the first source router number r should be converted into the local prefix of source router and the prefix table of local sub-net should be used.

2) Traffic-oriented flow subscription

The route-flow view is to conduct statistics of the packets transmitted by each flow in the router by using the sampling function of the router. The flow statistic message is consisted of inbound port, source IP address, destination IP address, TCP port number and transmitted packets quantity. The next-hop information is obtained by associating the inbound port of the statistic packets with the inbound port of the router in the neighbor table of routing view, the next-hop router is obtained

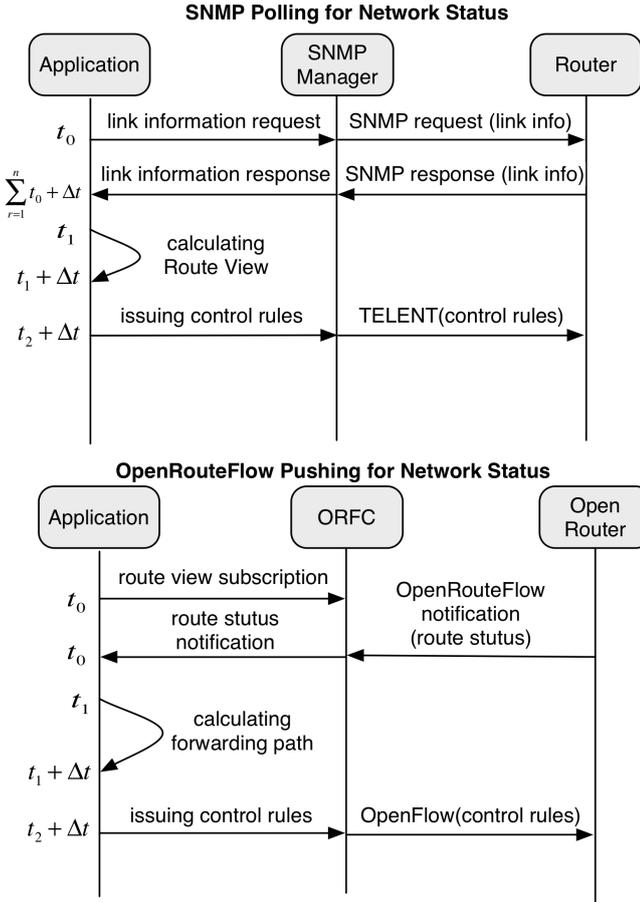


Figure 5. Comparison of polling and pushing of routing status

according to the outbound interface of the next-hop information and then the flow information of the flow of the router is collected. By the above-mentioned recursive method, the set of all the flows of the flow on the forwarding path can be calculated.

V. EVALUATION

A. Evaluation of Polling and Pushing of Routing Status

The main way of the retrieval of network status information is SNMP in a legacy network. A SNMP manager can poll network information from the SNMP agent in a network device periodically. Fig. 5 shows the time comparison between SNMP and OpenRouteFlow. A network control application with SNMP need send request messages to each network device.

This will spend $\sum_{r=1}^n t_0 + \Delta t$, where t_0 is the delay to generate a request message for a device. Δt is the delay to transmit a request message to a device. However, OpenRouteFlow controller is a listener for network status. Once OpenRouter pushes network status information, the controller can calculate network view as soon as possible.

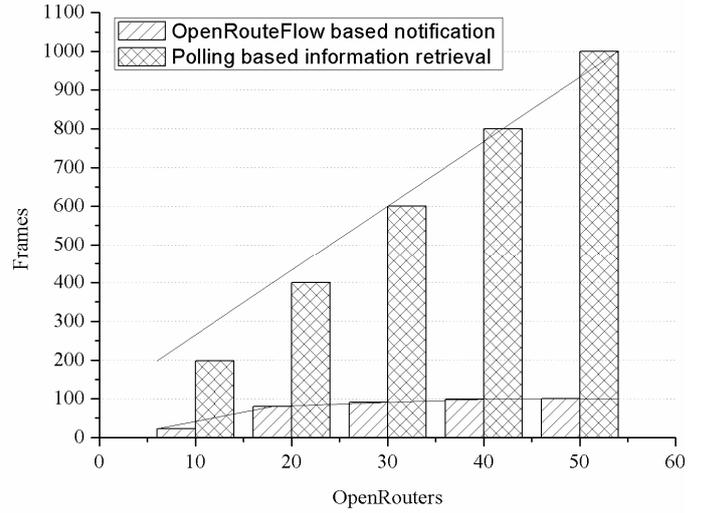


Figure 6. Protocols overhead of route status transmission.

B. Protocols Overhead of Route Status Transmission

To compare SNMP overhead with OpenFlow overhead, we implement the OpenRouter prototype with Click Router. We compares the overhead of OpenRouteFlow and SNMP with different numbers of router. Fig. 6 depicts number of frames sent when SNMP polling is used in the retrieval of route table entries, as well as when OpenRouteFlow protocol is used. While the reduction in overhead with the OpenRouteFlow protocol is substantial. The application with OpenRouteFlow sets up a connection with each router via TCP. Each router will send routing table entries when routing table changes in a router. The application with SNMP with each router for all of routing table entries with SNMP OID request and response. Moreover, if SNMP based application has to decrease the intervals of polling messages, network status may not be acknowledged as soon as possible and result in a positive false in the calculation of route view.

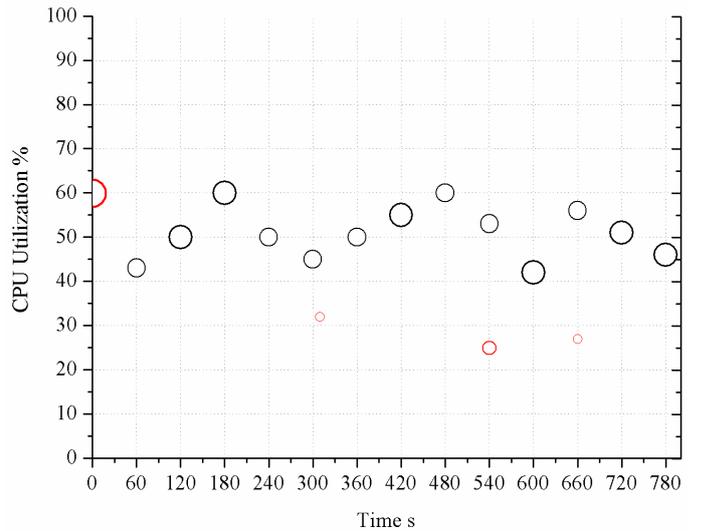


Figure 7. CPU Utilization evaluation of OpenRouteFlow Controller.

C. OpenRouteFlow Controller CPU Utilization evaluation

The network view generation of OpenRouteFlow will consume CPU of server. In the routing view generation algorithm, OpenRouteFlow controller triggers push when OpenRouter perceives the change of route status. In Fig. 7, the view calculation method according to the changes of route status can greatly reduce the CPU load compared with pull model view calculation method. The bubbles with black edge represent the CPU utilization of pulling network information periodically. The bubbles with red edge represent that of pushing network status information when the routing table of an OpenRouter is changed. From the figure, OpenRouteFlow with pushing network status information will not occupy CPU frequently more than the way of pulling. The diameter of a bubbles represents the difference of the CPU utilization at the time on the retrieval of network information and at previous slot. Because of incremental computing of OpenRouteFlow, the changes in CPU utilization is obvious with respect to the way of pulling.

D. Evaluation of Flow Subscription Service

We refactor an application of intra-domain source address validation [15] based on improvement of early work [16, 17]. We develop ORFC and provide flow subscription services for network control applications with REST APIs based on Floodlight which is a popular OpenFlow controller in JAVA. OpenRouteFlow is enable to provide route view and flow subscription service so that the network control application of the intra-domain source address validation can decouple the calculation for filter tables with network status retrieval. OpenRouteFlow based intra-domain source address validation can listen the notification of routing table change from OpenRouter. The OpenRouteFlow solution is more sensitivity to network status than SNMP polling in a legacy router so that OpenFlow based intra-AS source address validation can reduce false positives of packets filtering. Fig. 7 shows filtering rules delay cumulative probability distribution.

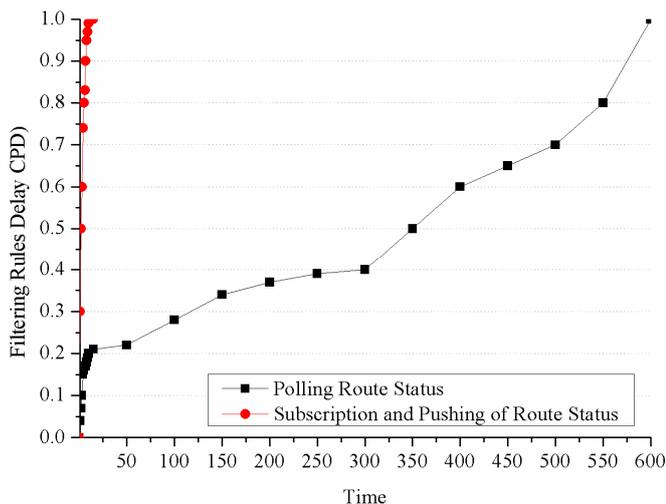


Figure 8. Filtering Rules Delay CPD.

VI. RELATED WORKS

It is one of the important directions for the academic research of SDN technology to support the routing operation by OpenRouter and OpenFlow to realize hybrid SDN network.

A. Open Practice of Network Equipment

RouteFlow [8], or QuagFlow [11], runs the routing protocol in multiple virtual machine of external computer. The virtual machine of each route corresponds to an OpenFlow switch in the data plane. RouteFlow controller realizes the packets forwarding control by mapping the routing rules as the flow rules of OpenFlow. In this paper, OpenRouteFlow, starting from the existing router, realizes the information opening of distributed routing protocol in the router. XORP [9] is an open-source router platform. The modular routing protocol realizes the expandability of router software by exchanging route information through forwarding engine abstract (FEA) and forwarding engine. The OpenRouter design of OpenRouteFlow and XORP are the router platform-oriented software fix. XORP focuses on decoupling the routing protocol and forwarding engine by forwarding engine abstract and OpenRouteFlow focuses on shielding the routing protocol in the router and providing a unified routing view for external control application to realize the decoupling of network visualization and controllability. According to the data center flow scheduling delay demand and OpenFlow protocol overhead analysis, DevoFlow [10] designs flow rule colony, flow thread trigger and advertisement mechanism, realizing the discovery and control of important flow and reducing the network expense brought by OpenFlow protocol and centralized control mechanism. OpenRouteFlow refers to the threshold trigger and push idea in terms of information transmission mechanism to reduce the flow load of control channel and expands this idea to the sending of route status.

B. Hybrid SDN

In OpenFlow 1.3 [6], hybrid OpenFlow is presented and suggested to adopt ship in the night, which separates the OpenFlow flow and legacy network flow by VLAN or a port. [12] is the general theoretical framework for the co-existence of control plane which is presented to solve the forwarding abnormalities. Panopticon [13] solves the deployment problem of SDN in the enterprise network by exposing an abstract of logical SDN, that is, solve the problem with regard to how to deploy and maintain the network consistency of SDN switch and legacy switch when benefited from the simplified management and stronger flexibility of SDN. To adopt legacy equipment and technology in OpenFlow network, LegacyFlow [14] introduces Legacy Datapath between OpenFlow control plane and legacy equipment to realize the configuration of flow table.

VII. CONCLUSION

Hybrid SDN is an important direction for the evolution of SDN, in which it is one of the challenges to be solved for hybrid SDN to open the router and regard the routing view as a service in order to realize the software defined route. In this paper, OpenRouteFlow is presented, which does not maintain

the legacy distributed routing networking mode, but also release the routing information and flow information in the router to form the associated view of route and flow and realize the decoupling of network visualization and network control.

OpenRouteFlow system is consisted of OpenRouter and OpenRouteFlow controller. The OpenRouter, on the basis of maintaining the hardware structure of router, embeds the OpenRouteFlow agent into the legacy router software platform by software upgrading to realize the dynamic push of routing change status and flow statistic information and be compatible to the OpenFlow control instruction. OpenRouteFlow controller can calculate the routing view according to the dynamic routing changes and provide path-oriented and flow-oriented flow subscription service for external applications.

The contributions of OpenRouteFlow are as follows: Firstly, the design of OpenRouteFlow integrates the centralized control in SDN with legacy distributed routing protocols, which improves the visualization and flexibility of legacy routing plane; Secondly, OpenRouteFlow provides a global routing view and flow view for the deployment of advanced network control functions; Thirdly, the implementation of OpenRouteFlow just needs software upgrading without hardware modification, which reduces the investment cost for SDN deployment; Fourthly, OpenRouteFlow protocol is compatible to OpenFlow protocol, which can realize the smooth migration to pure SDN deployment.

ACKNOWLEDGMENT

This work is Supported by the National High-tech R&D Program ("863" Program) of China (No.2013AA013505), the National Science Foundation of China (No.61472213) and National Research Foundation of Korea (NRF-2014K1A1A2064649). Jun Bi is the corresponding author.

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.*, 38(2): 69–74, 2008.

[2] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proc. INFOCOM*, 2013.

[3] T. Koponen et al., "Network virtualization in multi-tenant datacenters," in *Proc. NSDI*, 2014.

[4] O. Tilmans and S. Vissicchio, "IGP-as-a-Backup for Robust SDN Networks," in *Proc. CNSM*, 2014.

[5] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM CCR*, vol. 44, no. 2, pp. 70–75, 2014.

[6] "OpenFlow Switch Specification (version 1.3.4)," March 2014. [Online]. Available: <https://www.opennetworking.org/>

[7] M. Casado, T. Koponen, D. Moon, and S. Shenker. Rethinking Packet Forwarding Hardware. In *Proc. HotNets*, Oct. 2008.

[8] Marcelo R. Nascimento, Christian E. Rothenberg, Marcos R. Salvador, Carlos N. A. Correa, Sidney C. de Lucena, and Mauricio F. Magalhaes. Virtual routers as a service: the routeflow approach leveraging softwaredefined networks. In *Proceedings of the 6th International Conference on Future Internet Technologies, CFI '11*, pages 34–37, New York, NY, USA, 2011. ACM.

[9] M. Handley, O. Hudson, and E. Kohler, "XORP: An open platform for network research," in *Proc. SIGCOMM Workshop on Hot Topics in Networking*, October 2002.

[10] M. Handley, O. Hudson, and E. Kohler, "XORP: An open platform for network research," in *Proc. SIGCOMM Workshop on Hot Topics in Networking*, October 2002.

[11] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, and M. F. Magalhães. QuagFlow: partnering Quagga with OpenFlow. *SIGCOMM Comput. Commun. Rev.* 41, 4 (August 2010), 441-442, 2010

[12] S. Vissicchio, L. Cittadin, O. Bonaventure, G. G. Xie. On the Co-Existence of Distributed and Centralized Routing Control-Planes. *IEEE INFOCOM*, April, 2015

[13] Levin D, Canini M, Schmid S, et al. Toward transitional SDN deployment in enterprise networks[J]. *Proceedings of the Open Networking Summit (ONS)*, 2013.

[14] Farias F N N, Salvatti J J, Cerqueira E C, et al. A proposal management of the legacy network environment using openflow control plane[C]//*Network Operations and Management Symposium (NOMS)*, 2012 IEEE. IEEE, 2012: 1143-1150.

[15] Yao G, Bi J, Xiao P. VASE: Filtering IP spoofing traffic with agility[J]. *Computer Networks*, 2013, 57(1): 243-257.

[16] T. Feng, J. Bi, G. Yao, P. Xiao. InSAVO: Intra-AS IP Source Address Validation Solution with OpenRouter. In *proceedings of INFOCOM*, 2012.

[17] Yao G, Bi J, Feng T, et al. Performing software defined route-based IP spoofing filtering with SEFA[C]//*Computer Communication and Networks (ICCCN)*, 2014 23rd International Conference on. IEEE, 2014: 1-8.