OpenRouter: OpenFlow Extension and Implementation based on a Commercial Router

Tao Feng, Jun Bi, and Hongyu Hu
Network Research Center, Tsinghua University
Department of Computer Science, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology (TNList)
fengt09@mails.tsinghua.edu.cn, junbi@tsinghua.edu.cn, and huhongyu@cernet.edu.cn

Abstract—By analyzing challenges of current OpenFlow in production network, we propose three extensions of OpenFlow about FlowTable, control mode and OpenFlow protocol. Based on these extensions, a commercial OpenFlow-enabled router, named OpenRouter, is designed and implemented using only available and existing hardware in a commercial router. OpenRouter brings the abilities of control openness, integration of inside/outside protocols, and flexibility of OpenFlow message structure, low-cost implementation and deployment. We expect OpenRouter may accelerate the large-scale application and deployment of OpenFlow in production network.

I. INTRODUCTION

OPENFLOW is a key technology to enable network innovation. Current researches on OpenFlow are mainly about the applications of OpenFlow switch in campus [1] or in data center, but seldom about OpenFlow with Layer 3 switches or routers in production network.

In production network, Layer 3 switches or routers are dominant devices. Many mature network protocols, such as routing protocols, running inside the devices in distributed control mode. So, there are some challenges of current OpenFlow to deploy in production network. Firstly, to adapt all future protocols and different vendors, we need to make FlowTable more open. Secondly, if a new innovation is mature enough, we need to implement the controller inside the device to improve the efficiency. Thirdly, since it is hard to pre-define all the communication requirements between a controller and a device, we need to make the OpenFlow protocol more flexible. Fourthly, we need to run OpenFlow in today’s router, thus it will make OpenFlow deployment low-cost and deployable.

II. OPENFLOW EXTENSION FOR OPENROUTER

A. FlowTable Extension

Standard and open FlowTable structure combined with all forwarding elements, currently, is effective to control flows for experiment. However, there are many existing available forwarding hardware in production network, such as FIB, ACL and xFlow (NetFlow or sFlow which depends on vendor implementation). From the view of forwarding behavior, these existing forwarding elements can be regarded as a type or subset of FlowTable. FlowTable in OpenRouter, as shown in Fig. 1, can be defined by mandatory, optional or vendor-defined forwarding element.

B. Control Mode Extension

Control mode of current OpenFlow is a centralized control mode that is strong in control efficiency with a global view,
but is weak in scalability and robust. In OpenRouter, distributed control protocols, such as OSPF, still keep running inside the device. The information generated by distributed control protocols inside the device, such as routing information, are shared with outside controller by encapsulated in OpenFlow protocol, as shown in Fig. 2.

C. OpenFlow Protocol Extension

Network data in OpenFlow message are reorganized using TLV (Type, Length, and Value) format. It can not only support transmission of variable-length and optional data, but also achieve the goal of flexible extension of FlowTable structure in future. According to the content of current OpenFlow protocol, three TLV types have been defined: network state TLV (e.g., port status TLV and routing TLV), network configuration TLV (e.g., interface TLV and queue TLV) and FlowTable TLV. To differentiate different types of FlowTable in OpenRouter, “FlowTable Type” field is added to original FlowTable definition with TLV format, as shown in Fig. 3.

III. DESIGN AND IMPLEMENTATION OF OPENERouter

Based on three OpenFlow extensions, we have designed and implemented an OpenRouter prototype using a commercial layer 3 switch which architecture is shown as Fig. 4. An OpenFlow module is embedded into control plane of OpenRouter. OpenFlow protocol is redesigned and reconstructed with TLV. OpenFlow module communicates with Routing Table Management (RTM) and xFlow modules through two added asynchronous messages and a synchronous message to transmit routing information and sampling packets.

FlowTable of OpenRouter is implemented using existing forwarding hardware-FIB, ACL and xFlow, as shown in Fig. 5. FIB, ACL can consist of some headers and actions in FlowTable (OpenFlow version 1.0 [2]). Fine-grained flow match can be processed using ACL while FIB can process simple flow or packet match with forwarding destination. xFlow can take place of some functions of Counter field in FlowTable.

IV. CONCLUSION AND FUTURE WORK

The advantages of an OpenRouter are as follows: 

Openness. OpenRouter brings openness to a traditional commercial router with flexible and customized control abilities offered by OpenFlow.

Integration. Control protocols inside and outside an OpenRouter can be integrated and exchanged network state information.

Flexibility. OpenRouter enable TLV format to encode and encapsulate network data for variable-length data transmission and flexible extension.

Low-Cost. OpenRouter makes use of existing ACL and FIB hardware as a type of FlowTable for low-cost and easy implementation.

We have set up the testbed of an Intra-AS source address validation with 10 OpenRouters to evaluate the performance of OpenRouter, as shown in Fig. 6. In future, we’ll focus on performance problems of OpenFlow in production network. One is FlowTable compression using Bloom filter. The other is FlowTable quick lookup with new multi-dimensional lookup algorithms.

REFERENCES
