

## Towards a P4-based Dynamic In-Band SDN Control Channel

Takumi Yoshida Masahiro Shibata Masato Tsuru

Computer Science and Systems Engineering, Kyushu Institute of Technology

## 1. Introduction

For the decentralization and virtualization of computing resources, Software Defined Networking (SDN), which enables flexible and dynamic control of packet forwarding and processing, is an essential technology. To connect one or more controllers with a number of distributed switches in SDN, an in-band control channel is of importance. However, the traditional static schemes such as L2VLAN are insufficient for a reliable and efficient SDN control plane especially in unreliable wireless networks. Therefore, our final goal is to develop a framework for dynamic in-band control channels in which the control messages can piggyback on data packets. As the first step, we report a feasibility study of the in-band control channel using Programming Protocol Independent Packet Processors (P4) [1], which can be implemented flexibly and vendor-independently for packet processing functions of switches.

## 2. Implementation

A short control message is delivered from a switch to a controller by embedding the message into the option field of normal IPv4 data packets as shown in Fig.1. Switch ID is to identify the switch. Control ID is to identify the control message from the switch. Option type and Option length are defined for IPv4 option header management. As the result, a total of 20 bytes (160 bits) are added to an IPv4 packet to which any IPv4 option header has not been added before. The embedded control message should be forwarded to a controller and also removed from the original data packet before reaching a receiver host.

## 3. Prototype system testing and results

We implemented a P4-based in-band control channel on each switch that can (i) embed some information on the switch to application data packets passing through the switch selectively and (ii) extract (remove) the information from the data packet and forward it as a control message to the controller. The data packet selection can be based on the packet type and a predefined probability.

In the network configuration shown in Fig.2, the sender and receiver host (H1 and H2), two switches (S1 and S2) and the controller are implemented on off-the-shelf PCs with Ubuntu 16.04 LTS. UDP packets as application data are sent from H1 to H2 using iperf. In addition to the standard packet forwarding in the ingress part, the P4 switch developed here has two new functions in the egress part: one is to add the control message to a data packet (at S1 and S2) and the other is to remove the control message

from the packet and send it to the controller in a proper form (at S2). The latter function is performed at a switch directly connected to the controller in general. A typical operation in S2 is as follows. When a data packet with embedded control message arrives at S2, the clone of the packet is generated. Then the original packet is sent to H2 after the control message is removed. On the other hand, the cloned packet is sent to the controller as a control packet after its payload is removed and its destination address is changed to the controller's address. The size of the control packet is 54 bytes, including the Ethernet header (14 bytes) and the IPv4 header (40 bytes). The controller uses raw-socket to directly receive IPv4 packets and extract the control message.

Through our experiments as a functional test, we verified the controller successfully received the control messages from S1 and S2 while the application flow from H1 to H2 works normally. In the next step, a performance test is essential. A split embedding over multiple data packets should also be developed for a long control message.

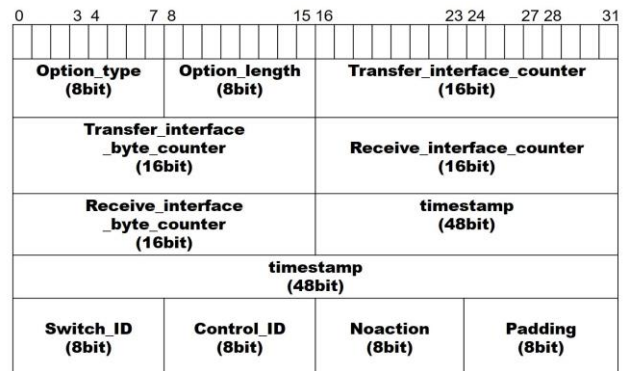


Fig. 1 Format of IPv4 option header.

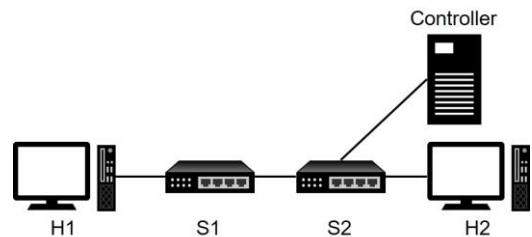


Fig. 2 Experimental network configuration.

## Reference

- [1] P. Bosshart, et al., "P4: Programming Protocol-Independent Packet Processors," SIGCOMM CCR, 44(3):88–94, 2014.