# Design and Implementation of Network Monitoring and Scheduling Architecture Based on P4

Junjie Geng
gjjtianxia369@163.com

Jinyao Yan
jyan@cuc.edu.cn

Yangbiao Ren
gemini_ren@cuc.edu.cn

Yuan Zhang
yuanzhang@cuc.edu.cn

Communication University of China, Beijing, China

## ABSTRACT

Network monitoring is an important part of network operation. In this work, we proposed a network monitoring and scheduling architecture based on P4 which monitors and visualizes the network state information. We evaluate the proposed scheme on INT. Preliminary results show that the congestion can be avoided by our scheduling method in the testing scenario.

## CCS CONCEPTS

• Networks • Information systems

## KEYWORDS

Networks status information, INT, Traffic scheduling, P4

## 1 Introduction

A variety of monitoring technologies are developed to monitor network traffic, including SNMP/RMON [1], NetFlow/SFlow [2], OpenTM [3] and so on.

Among them, SDN is a new way for network monitoring and management, however, it is still far from what we want. Most exiting SDN-based methods achieve network status monitoring through extra probe packets, polling, etc., making the network monitoring costs too much. To address the heavy overhead in SDN, P4 [4] is proposed. P4 is a programming language mainly used for data planes to provide instructions to the data forwarding plane equipment (such as switches, network cards, firewalls, filters, etc.) in indicate how to handle data packets.

Inband Network Telemetry (INT) [5] is a framework designed to allow the collection and reporting state of the data plane, without requiring intervention or work by the control plane. It is a powerful new network-diagnostics mechanism implemented in P4.

In this paper we design and implement a network monitoring and traffic scheduling architecture based on P4. We realize many functions in our P4 monitoring and scheduling system, such as network status detection and positioning, and network status information visualization. Furthermore, we have proposed and implemented a traffic scheduling scheme based on the network status information and the P4 routing architecture. Evaluations show that the basic functions we proposed have been verified in the experiment.

## 2 Network Monitoring and Scheduling Architecture

This section describes the design details of our network monitoring architecture which integrates network status monitoring and network management based on P4. As shown in figure 1, we get the status information of switch through the Software defined monitoring based on the INT, and then we use the acquired state information to realize the function of network management, such as the forwarding logic design, traffic scheduling.

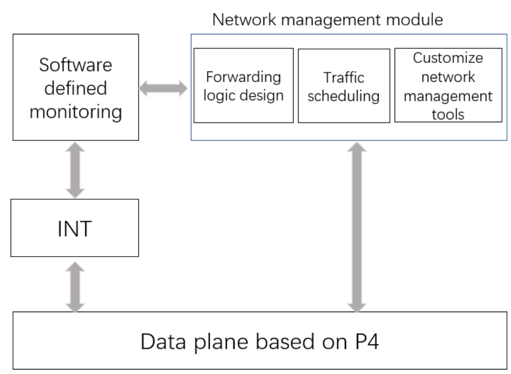


**Figure 1 Network Monitoring and Scheduling Architecture**

The implementation method and function of each module are as follows:

**Software defined monitoring module** obtains the status information of each switch in the data plane through the P4 INT architecture. These information includes switch id, ingress port id, hop latency, queue length, etc. We can get more network status information through extending the INT function according to the metadata information provided by the P4 switch. Through the status monitoring module, we can realize the functions including network fault monitoring and positioning, visualization of network status information.

**The network management module** includes three submodules: forwarding logic design, traffic scheduling module, and customize network management tools. The forwarding logic design can compute and download the forwarding table to implement the forwarding logic. The traffic scheduling module can avoid congestion using the network status information, the module generates a JSON configuration file through the P4 program by P4 compiler, and directly configures the P4 switch to implement the traffic scheduling function. Network manager can develop network management application using the customize network management tools.

## 3 Experiment Design and Results

We use mininet to create the experiment topology with bmv2 as the software switch. Our experiment topology is shown in Figure 2. We use fat-tree like data center network topology as the experimental topology in this paper.

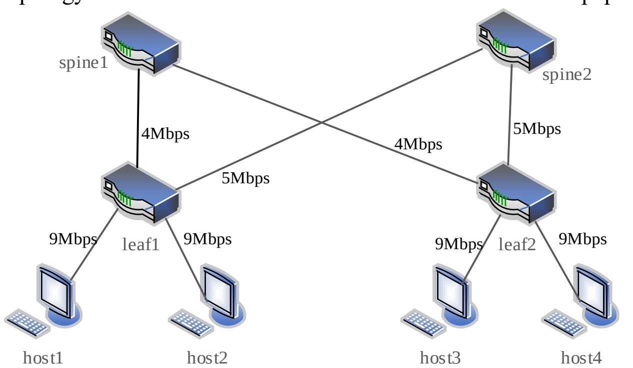


**Figure 2 Network topology**

Four hosts (host1 to host4) are INT Sources/Sinks. VTEP running on host is responsible for encapsulating and de-encapsulating Vxlan GPE headers. Switches are INT transit inserting INT metadata.

We send a UDP flow with 4M/s from host1 to host3 via iperf. There are two paths from host1 to host3: host1→leaf1→spine1→leaf2→host3 with 4Mbps available bandwidth and host1→leaf1→spine2→leaf2→host3 with 5Mbps available bandwidth.

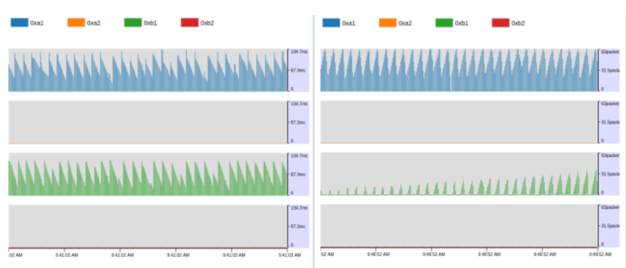

**Figure 3 Congestion monitored**

Figure 3 shows that network congestion has occurred and monitored by the software defined monitoring module. Both hop latency (left figure) and queue occupancy (right figure) are very high when packets passing through host1→leaf1→spine1→leaf2→host3 with available bandwidth of 4Mbps.

Then, the scheduling module in our proposed architecture schedules the flow to the other path with 5Mbps available bandwidth. Figure 4 shows that congestion has been effectively avoided. Both hop latency and queue occupancy are close to zero.

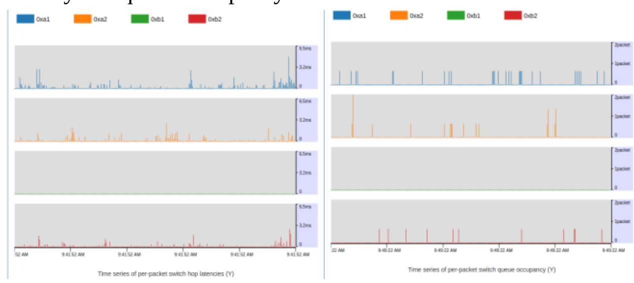

**Figure 4 Congestion eliminated**

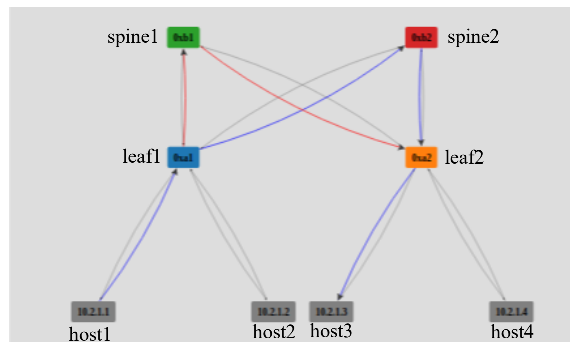


**Figure 5 The path before/after scheduling**

In addition, we visualize the transmission path of the packets by capturing the switch id. As shown in Figure 5, it can be directly seen that the transmission path is changed from host1→leaf1→spine1→leaf2→host3 to host1→leaf1→spine2→leaf2→host3 (from orange to blue links in the spine).

## 4 Conclusion and Future work

In this work, we proposed a network monitoring and scheduling architecture based on P4 which monitors the network state information (switch id, hop latency and queue occupancy) without introducing additional detection packets, implements visualization using these state information, and real-time control/schedule traffic according to the network state.

In the future, we will conduct experiments for realistic applications on hardware P4 switches .

## REFERENCES

[1] Gerald A. Winters, Daniel A. Muntz, Toby J. Teorey.Using RMON Matrix Group Extensions to Analyze Internetworking Problem. Journal of Network and Systems Management, 1998, Vol.6 (2), pp.179-196

[2] Bingdong Li; Jeff Springer; George Bebis; Mehmet Hadi Gunes. A survey of network flow applications.Journal of Network and Computer Applications. 2013.

[3] Amin Tootoonchian, Monia Ghobadi, Yashar Ganjali. OpenTM: Traffic Matrix Estimator for OpenFlow Networks. Passive and Active Measurement. 2010.

[4] Pat Bosshart, Dan Daly, Glen Gibb, Martin lzzard, Nick Mckeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, David Walker. P4: Programming Protocol-Independent Packet Processors. ACM SIGCOMM, 2014.07.

[5] Changhoon Kim, Parag Bhide, Ed Doe. In-band Network Telemetry. June 2016. Conference Name:ACM Woodstock conference.