Slope: A Visual Multi-Controller Management Platform For SDN

Haisheng Yu, Heng Qi, Yinping Li, Junxiao Wang, Sheng Chen, Keqiu Li*
School of Computer Science and Technology
Dalian University of Technology, Dalian, China

ABSTRACT

This demo proposes a visual multi-controller management platform for one or more SDN domains.

Keywords

Slope, Software Defined Networking, Multi-Controller Integration

1. INTRODUCTION

Software defined networking (SDN) has achieved great success as a centralized control model, but it still has some significant deficiencies. One limitation is that all the SDN controllers have their own configuration and management way. A good SDN management platform requires standardized communication mechanism and easy-to-use user interface. Hence, heterogeneous controllers are supposed to provide unified API for users and administrators. How such API is perfectly defined has not been well addressed so far.

Thus, this demo proposes an integrated multi-controller platform (Slope) for one or more SDN domains:(1) we design a many-to-one based high performance network information collection mechanism. We define what network information should be collected and how such information is efficiently exchanged between Slope and SDN controllers. (2)For multiple domains, we design a peer-to-peer based network information exchange mechanism.

The source code and documentations can be found at http://202.118.75.100:8000/sdn. The video of demo is available at https://www.youtube.com/watch?v=81ogRWiXiGM.

2. SLOPE

Understanding how Slope realizes a control platform requires knowing the function of each layer. There are four layers in a network controlled by Slope, and they have very distinct roles.

Application Layer. The application layer determines how a specific application use a network. A graphical user interface (GUI) is needed for SDN controller, then the GUI will change SDN control platform from a software to a networking operating system.

Decision Layer. Decision Layer is the core component of Slope, and it is a newly established layer between SDN applications and controllers. Decision layer makes all decisions driving network control, including reachability, load balancing, access control, security, and interface configuration.

Dissemination and Discovery Layer. The Dissemination and Discovery Layer provides a robust and efficient communication substrate that connects routers/switches with decision elements.

Data Layer. The data layer is the carrier of SDN network, includes network switches, and any other network elements that support an interface allowing Dissemination Layer to read and write the state controlling the element's behavior (such as forwarding table entries). The function of the data layer is mainly confined to packet forwarding and simple processing.

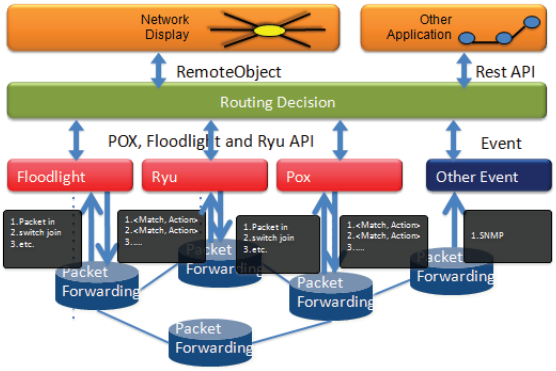

Figure 1: Slope has five components, such as network display, routing decision, dissemination and discovery, forwarding layer and data storage.

3. DEMO IMPLEMENTATION

Slope has five components, such as network display, routing decision, dissemination and discovery, forwarding layer and data storage(see Figure 1).

The network display is responsible for displaying and managing flow entry, traffic and topology information in SDN. The network display is written in flex language. The routing decision aims at collecting network topology information and statistics, and updating flow entry according to the change of network topology information or configuration. Figure 2 shows what the network display looks like.

The dissemination and discovery send network topology and statistics to the routing decision, and distribute flow entry to switches.

*Corresponding author: keqiu@dlut.edu.cn

The function of the data layer is mainly confined to packet forwarding and simple processing. Figure 3 shows that we use four Pica8 P-3297 switches[3], a NetFPGA card, an ONetSwitch20 card[2], and some OpenvSwitches on Ubuntu 12.04 as our forwarding devices.

The network display interacts with the routing decision by one of java communications flex three ways: RemoteObject. The SDN controllers provide APIs in the interaction with routing decision.

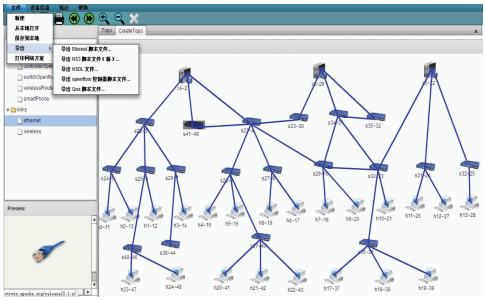

Figure 2: the Slope Network Display

Slope Mechanism: We implemented the Slope and successfully integrated three heterogeneous controllers: Ryu[5], Floodlight[1] and Pox[4]. The demo shows how Slope connects three SDN controllers and collects network topology from these controllers, and how Slope manages flow entries through customized web forms.

Use Case1: Network Information Collection. Slope collects network information from a group of controllers with no interconnection and generate a domain-wide network view, and collects network information from other Slope instances to generate a global-wide network view. A domain includes these information: domain basic information, controller, flow, host, link, switch, switch port and switch port statistics.

Use Case2: Setup Routing Paths. To simplify the manual process of setup routing path, we create a web form with validation in the network display. The routing decision translates all the inputs from the network display into OpenFlow[6] flow entries and setup routing paths.

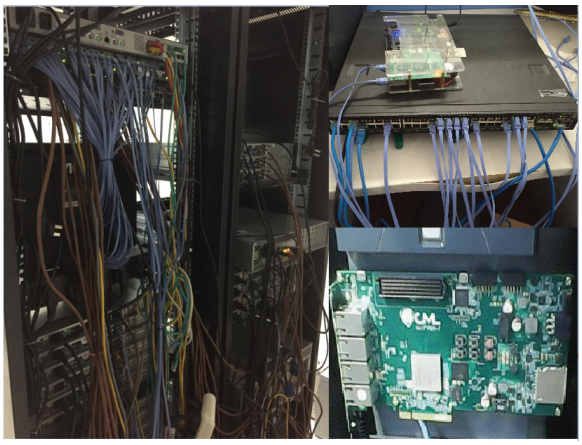

Figure 3: The testbed includes four Pica8 P-3297 switches, a NetFPGA card, an ONetSwitch20 card, and some OpenvSwitches.

Use Case3: Network Topology Backup and Recovery. To reproduce and diagnosis network when something wrong happens, we backup network topology into a script and recovery the network topology in a test environment.

4. EVALUATION

Each Slope instance has to connect to the controllers it manages. To stress this interface, we connect three controllers to a single Slope instance and ran apache benchmark to test Slope's performance as a general platform. The performance of a Slope instance will depend on the capacity for processing updates. To measure this throughput, we ran an apache benchmark which repeatedly acquired exclusive access to a Slope instance.

Figure 4 shows that thread modification percentage of the requests complete in a certain time. When there is only one thread, a Slope instance finishes 95% of the requests in 5ms. When the threads increase, a Slope instance uses much less time to complete a request.

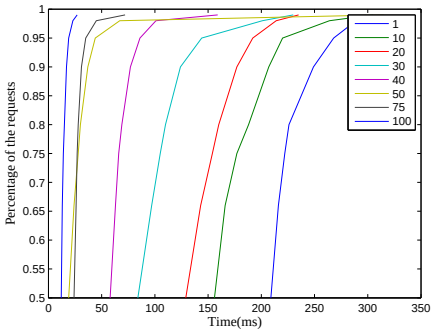

Figure 4: Thread modification percentage of the requests complete in a certain time(ms).

Acknowledgments

This work is supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010); the State Key Program of National Natural Science of China(Grant No. 61432002); NSFC Grants 61173161, 61173162 and 61272417; the Fundamental Research Funds for the Central Universities; Prospective Research Project on Future Networks from Jiangsu Future Networks Innovation Institute.

5. REFERENCES

[1] floodlight project. http://www.projectfloodlight.org/floodlight/.

[2] meshsr company. http://www.meshsr.com/.

[3] pica8 company. http://www.pica8.org/.

[4] pox project. http://www. noxrepo.org/pox/about-pox/.

[5] ryu project. http://osrg.github.io/ryu/.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, pages 69–74, 2008.