Architecture Design and Implementation for P4 INT Web System

Wun-Yuan Huang, Ta-Yuan Chou, Hui-Min Tseng, Nai-Yuan Hu, Te-Lung Liu
Affiliation: National Center for High-Performance Computing, Tainan, Taiwan, R.O.C
{wunyuan, 1203053, 0303118, 2103081, tlliu}@narlabs.org.tw

***Abstract*--This paper presents a design of the P4 INT Web system. its purpose is to gather P4 INT network performance data stored in a database, and display them on the Web UI. The ability of P4 INT can put switch status data into a packet, such as an ingress time, egress time, and switch hop count. Users can implement or look for an application for collecting the INT data to do a network performance measurement, and then record results to the database. However, organizations that have a P4 INT testbed often gather and store INT data with specific methods or with a different database, these organizations have to address the issue of how to get INT data from different databases when they corporate to build a large P4 INT testbed. For the issue, we design a system that can gather and analyze INT data from many kinds of databases, and then show the results on the Web UI.**

I. Introduction

The emergence of SDN brings new transformation and thinking in network technology, which changed the control mode of tradition and split the data and control layer from traditional network equipment. The communication of both layer is usually via an SDN compliant protocol so the data plane can handle packets following the decision results of the control plane. In addition to former feature, SDN has the programmable feature in control plane. So Users can develop the desired functions on an control plane to satisfy the demand of customized networks.

Although the control layer is open, the data layer is still closed. Therefore, ONF proposes P4 Language [1], and it is a framework that allows users to write how to handle packets on the data layer. In addition to the P4 architecture, P4.org also provides a P4Runtime protocol for communication between the control plane and the P4 switch. Therefore, users can use a control plane with P4Runtime to manage P4 switches.

P4.org also develops an In-Band Network Telemetry (INT) [3], which embeds switch status data into a packet header in a P4 Switch. Then, the last P4 Switch of the transmission path can retrieve INT data from the INT Field embedded in the packet header to analyze the network performance. Therefore, a user can use a P4 switch that supports INT packets, coupled with a P4Runtime control platform, such as ONOS, to build a P4 INT network.

Because INT can put network status data in a packet and then get them at the end switch to do a network performance measurement, users can implement an application to collect INT data to monitor the P4 network. However, most implemented applications by users often only consider storing INT data and measurement in their database and do not exchange data with external applications. Therefore, if the organizations cooperate to integrate their P4 testbed to become a large-scale P4 network, they have to address INT data or measurement sharing issues among INT collector applications of each organization. For example, we plan to connect our P4 INT testbed on TWAREN with the P4 INT testbed of iCAIR. However, iCAIR's database and ours are not the same, so we have to design a system to compatible both databases to gather P4 network measurement of two domains. Therefore, this paper will propose a system design to solve this issue.

II. P4 INT Web System Design

A. Architecture Design

In addition to deploying specific databases, most INT data collector applications do not support topology display and flow visualization. Therefore, we design a system that can interact with different databases and support topology display and flow visualization. Fig. 1 is system architecture design. To interact with all kinds of databases, the bottom layer of this architecture is DB Driver Layer, which contains some database drivers. This system can interact with different databases via these drivers and return data to the Analyze Layer. In Analyze Layer, it can process and analyze the data obtained from the database, format it into a specific data format according to the functions provided by the UI Layer, and then visualize the data in the UI Layer.

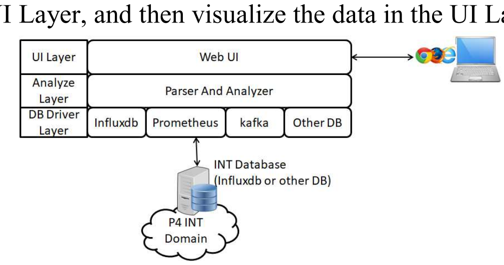

Fig. 1. System Architecture Design.

B. Web UI

This section will describe a beta version of Web UI implemented by us. This Web UI has a Grafana Dashboard framework, Data Query feature, and topology display function to satisfy all requirements for P4 monitoring on TWAREN. Fig. 4 and 5 show this Web UI. The Web UI page will be divided into three parts: information input framework, information display framework, and Grafana Dashboard framework. The information input framework is responsible for obtaining database data and adding data sources in Grafana. The information display framework and Grafana Dashboard framework will display the data from the database and the results of the Grafana Dashboard creation.

C. Server-side Design

Server-side include Analyze Layer and DB Driver Layer.

Fig. 2 is the architecture design for these both layers. This architecture can satisfy the P4 monitoring requirement on TWAREN. And this architecture processing is as follows:

- Step 1: Send "request" to Data Parser.
- Step 2: Forward Parser result to DB Selector. DB Selector will base on Parser result to Communicate with Influxdb or Grafana.
- Step 3: Reply DB info or Grafana data to Data Classification.
- Step 4: Topology info is forwarded to Topology Creator. Others will be forwarded to Normal Data Parser. If the data is received from the Grafana connection, it will communicate with Grafana Server to create a Grafana dashboard and statistical graphs.
- Step 5: Combine topology data and normal data.
- Step 6: Format data in JSON format.
- Step 7: Reply JSON data to Web UI.

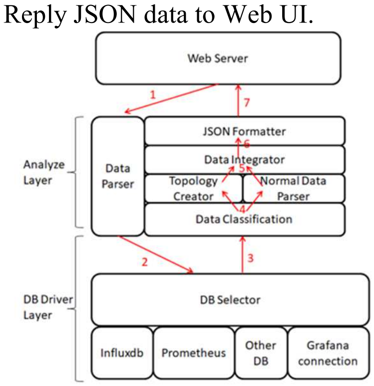

Fig. 2. Server-side processing procedure.

III. Experimental Results

We implemented a beta system according to our design architecture in chapter 2. But this beta system is rough yet. It only supports communication with InfluxDB. We deploy this beta system, BMV2, ONOS, and BPFCollector [4] to build a test environment successfully. Fig. 3 is this test environment. The goal of this environment is to confirm our design is feasible. And the test result is shown in Fig. 4 and 5.

We try to modify this beta system for the multi-domain P4 INT network after deploying it successfully in the experimental single domain network. We also change the topology of Fig. 3 to an experimental multi-domain network of Fig. 6. Finally, we deploy this beta system in the experimental multi-domain network, and we also get all correct INT information and global topology.

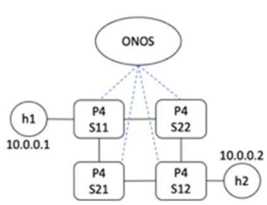

Fig. 3. Experimental Topology.

Fig. 4. Test result for the Web UI of the beta system.

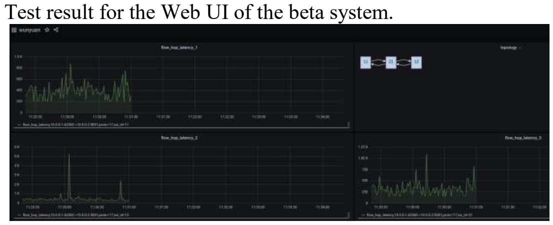

Fig. 5. Test Result for the Grafana feature of the beta system.

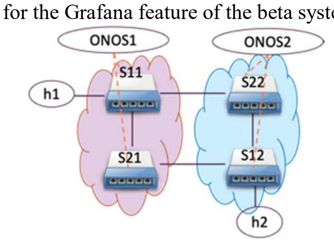

Fig. 6. Experimental muti-domain topology.

IV. Conclusion

With INT technology on P4, Users can collect INT data easily to manage a P4 network. However, most collector applications do not support INT data sharing, and they only store INT data in a specific database. When some organizations want to cooperate to integrate many P4 subnets to build a large-scale P4 network, it is hard to know the global P4 network status for the manager if their collector application cannot share INT data stored in a specific database. So, we propose a system design to compatible all databases to address this issue. In the future, besides further improving our architecture design, we will add all functions we designed in the beta system to benefit the management of the P4 network.

In the future, we will build a P4 testbed with Tofino P4 switch equipment on TWAREN. Also, we will port this beta system to the testbed and try to improve it on the testbed.

Reference

[1] A. Sivaraman, C. Kim, R. Krishnamoorthy, A.t Dixit, and M. Budiu, "DC.p4: Programming the Forwarding Plane of a Data-Center Switch," Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, Article No. 2, 2015.

[2] P4Runtime Specification, https://p4.org/p4-spec/p4runtime/v1.3.0/P4Runtime-Spec.html

[3] In-band Network Telemetry (INT) Dataplane Specification, https://p4.org/p4-spec/docs/INT_v2_1.pdf

[4] INTCollector, https://gitlab.com/tunv_ebpf/BPFCollector