



Network Research Workshop

Proceedings of the Asia-Pacific Advanced Network 2014 v. 38, p. 19-22.

<http://dx.doi.org/10.7125/APAN.38.x> ISSN 2227-3026

A Flow-based Method to Measure Traffic Statistics in Software Defined Network

Chun-Yu Hsu^{1,*}, Pang-Wei Tsai¹, Hou-Yi Chou¹, Mon-Yen Luo² and Chu-Sing Yang¹

¹ Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University / No.1, University Rd., East Dist. Tainan City, Taiwan

² Department of Computer Science & Information Engineering, National Kaohsiung University of Applied Sciences / No.415, Chien Kung Rd., Sanmin District, Kaohsiung City, Taiwan

Mails: sherlyhsush@gmail.com; pwtsai@ee.ncku.edu.tw; hychou@ee.ncku.edu.tw; myluo@kuas.edu.tw; csyang@ee.ncku.edu.tw

* 92633R, EE Department Building, Tze-Chiang Campus, National Cheng Kung University, No.1, University Rd., East Dist. Tainan City, Taiwan 701, Taiwan; Tel.: +886-6-2757575-62357; Fax: +886-6-234-5482

Abstract: Since software defined network became a revolution of network, many new developments and deployments are put forward. Device operating in software defined network environment will need a controller to store control policy. As a result of flow is one of traffic presentation units in software defined network, hence, this paper proposed a prototype of flow-based method to measure traffic statistics which can be used in OpenFlow network. By analysis of flow table, active flows which managed by controller can be listed, and information of each flow can be presented with developed module on the controller. After gathered port statistics from OpenFlow switch and entries in flow table, information of each flow can be presented on a developed graphic interface. By using this method, Traffic statistics of each flow may be more adaptive and realizable for measurement for observation.

Keywords: flow-based; OpenFlow; traffic statistics.

1. Introduction

Since Software Defined Network (SDN) became a revolution of network, many new developments and deployments are put forward. The most significant difference between traditional network and software defined network is design of control- and data-plane. This will make a significant difference on network operation issues such as network management, traffic statistic and so on. Hence, this paper proposed a prototype of flow-based method to measure traffic statistics in OpenFlow network environment. According to collected information from controller, with developed module in this paper, there will be more adaptive and realizable to monitor flow processing in SDN environment.

2. Related Work

2.1. OpenFlow and OpenFlow Controller

In the past, the network is hard to innovation and manage due to each device has its own control logic and has vendor dependency. To break this limitation, OpenFlow[1, 2] was proposed. The core idea of OpenFlow is to decouple the data plane and control plane. OpenFlow enabled devices can be controlled by controllers through secure channel. Controllers instruct devices to forward packets according to the instructions or flow entries. Users can deploy their new idea on the OpenFlow network with a central control logic. We choose POX[3] controller, origin from NOX[4], for deployment.

2.2. Mininet

Mininet[5] is an open source network emulator for prototyping software defined networks. With Python language, Mininet is simple to use and has a great flexibility. It can create fairly components with its lightweight approach which uses an OS-level virtualization. As for switches, they are built upon OpenvSwitch[6] as OpenFlow based devices.

3. Design and Development

3.1. Flow Diagnostic Module

Due to the design concept of OpenFlow, forwarding rules (flow entries) for all controlled switches will be kept in data structure of their controller. This module is used to make analysis of forwarding rules, we can list all flows on entire network which controlled by this controller. Usually these information are stored in a Flow Table. To identify different flows from different switches, our method attaches a distinctive id number to each flow for recognizing them. In the implementation of this flow-based measurement method, some modification has been made to POX controller. Hence, controller will handling flows from different switches correctly.

3.2. Data Collection Module

Usually, there will be several counters stored packet processing record in each OpenFlow device. In our development, we add a data collection function implemented in controller for querying the counter periodically. Accumulated data such as packet count, byte count and dropped-packet number in period of time will be stored in data structure. The interval time is given by administrator. This module can itemizes data in two types: per port and per flow statistics. Per port collection is a basic way for gathering traffic statistics. For most classic network monitor presentations such as MRTG[7], port utilization of each device can be easily appeared. However, this may not be applicable for all OpenFlow devices. Therefore, a Data Collection Module is designed. It collects data from switches by controller periodically and send them for calculation. On the other hand, according to flow information which provided from Flow Diagnostic Module, a data structure for per flow statistics is also prepared to preserve these calculated statistics.

3.3. Statistic Integration Module

Statistic Integration Module is used to aggregate traffic including port, flow and other index items. Collection of Statistic Integration Module will be sent to here, and be refreshed iteratively. For example, port status can be presented with received and transmit bit rate; received and transmit packet rate; packet average size and drop packet number, etc. For per flow status, packet rate and byte rate per flow are the crucial statistics in flow-based mode.

4. Experiment and Verification

This paragraph shows experiment and verification results of proposed method in this paper. The experiments configured in Mininet 2.0.0. Controller devices are configured in OpenFlow 1.1.0 and Linux Kernel 2.6.32. We utilize Mininet to create simulation environment. In experiment, two of the hosts send heavy-load packets flow to another host. Our modules collect data, analysis the traffic load, and present the result in weathermap as shown in Figure.1. The green nodes and the orange nodes represent hosts and switches, while the number beside switch means switch id. Port and flow statistics are attached beside the weathermap. Taking Figure.1 as an example, the detail data for switch will be listed on web interface, including port statistics such as received packet count, transmitted packet count, dropped packet count and bit rate. For more, a flow statistics index is developed to identify the header, action and traffic statistics of each flow, and it will also be presented on web interface.

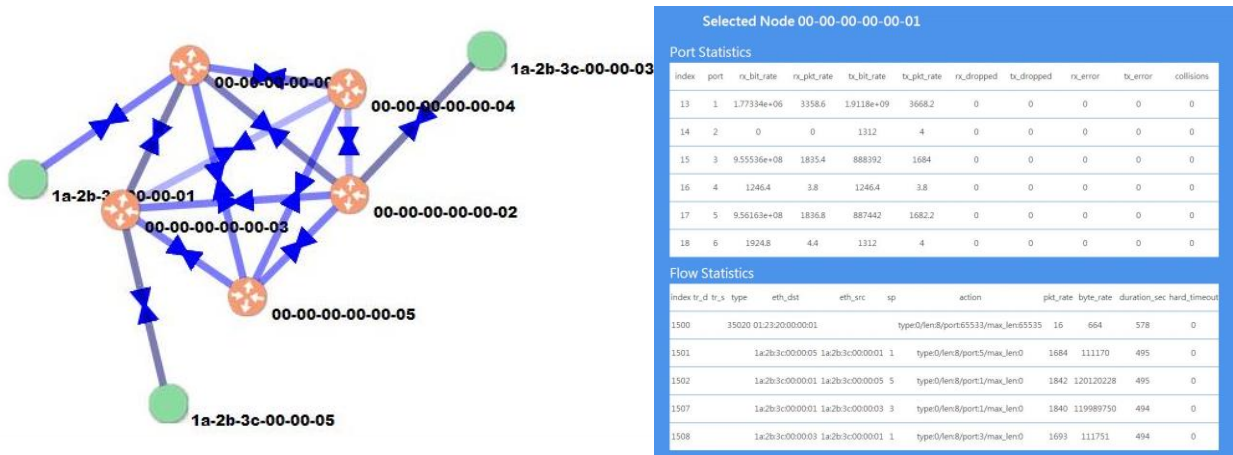


Figure 1. Weathermap and information on web interface.

5. Conclusions

This paper proposed a flow-based method to measure traffic statistics in software defined network environment. According to collected information in POX controller, this design can provide statistics which are related to each flow. By using this prototyping method, Traffic statistics of each flow may be more adaptive and realizable for measurement or observation.

References

1. Nick McKeown; Tom Anderson; Hari Balakrishnan; Guru Parulkar; Larry Peterson; Jennifer Rexford; Scott Shenker; Jonathan Turner. *OpenFlow: Enabling Innovation in Campus Networks*. 2008.
2. *OpenFlow Switch Specification Version 1.1.0 Implemented (Wire Protocol 0x02)*. Available: <http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf>
3. *POX controller*. Available: <http://www.noxrepo.org/pox/about-pox/>
4. N. Gude; T. Koponen; J. Pettit; B. Pfaff; Martin Casado; Nick Mckeown; Scott Shenker. NOX: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, vol. 38, 2008; pp. 105-110.
5. Mininet. Available: <http://mininet.org/>
6. OpenvSwitch. Available: <http://openvswitch.org/>
7. MRTG. Available: <http://oss.oetiker.ch/mrtg/>

© 2014 by the authors; licensee Asia-Pacific Advanced Network. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).