Toward a Network and Traffic Monitoring and Management Platform for Software Defined Networks (SDN)

Nguyen Tai Hung

Hanoi University of Science and Technology – No. 1, Dai Co Viet Str., Hai Ba Trung, Ha Noi, Viet Nam

Received: April 17, 2019; Accepted: June 24, 2019

Abstract

BKMON is a software tool designed to provide an independent monitoring solution for SDN network controllers with scalable capabilities. Therefore, it extends monitoring capabilities based on the network controller by providing visual graphs with information on each switch or port of the switch or even on specific traffic flow. Network monitoring helps to avoid interruption and system's failure during the operational process, thus, physical errors such as cable breaks or network node failures can be detected and fixed quickly. Those examples show how important network monitoring systems are and furthermore, the complexity and extensibility of the current network makes monitoring an extremely difficult task. Finally, the deployment of new technologies, like SDN, in the network makes it even more challenging. BKMON was defined and prototyped in the direction that will partly provide solutions to this scientific and technological challenge.

Keywords: SDN, network monitoring, openflow, northbound API, RESTFul.

1. Introduction

In any size of the computer networks, management and monitoring have always played an important role in process of operating, exploiting and ensuring network's efficiency. Determining critical state parameters of network components is essential for evaluation of network system. In practice, network monitoring usually means regular measurement and collection of many network-related parameters such as bandwidth usage or delay between nodes. Moreover, other parameters based on network nodes such as connection status or packet loss are important indicators to indicate if everything in the network works as expected. Nowadays, monitoring tools furtherly strive to provide information that can be later used to evaluate the Quality of Service (QoS), detect faults and improve security. Traffic monitoring in traditional networks is a well-researched area, where many applications were developed in order to aid network administrators in their day-to-day operations. Some of them include solutions built with the use of packet sampling or Deep Packet Inspection (DPI) implemented in software or hardware. However, while trusted by in industry the computational requirements of traffic monitoring continue to impact network performance significantly.

Software Defined Network (SDN) [1] enables highly programmable networks with a high level of transparency unlike traditional networks allowing the management of flows on each component in the network. Network impairments can be calculated using statistical information provided by each SDN enabled switch. Unfortunately, borrowing monitoring techniques developed for traditional networks and placing them in SDN, may cause a large overhead due to large amount of queries exchanged between the controller and switches in the network. New techniques should be developed in order to reduce the amount of resources required to monitor application specific traffic. One approach to minimize the overhead is to filter out traffic based on feedback arriving directly from the business application, so that the application becomes the main source of information used to decide on which flows in the network should be monitored. Based on this feedback we can identify switches that require monitoring. The management decisions are still made by the controller, not the service itself, however, the controller gains greater visibility in the network. Ideally the monitoring and decision making should be completed without the need for packet classification. This paper presents BKMON: an SDN monitoring platform equipped with out-of-the-box techniques used to identify traffic unit like: packets, flows and ports together with its statistic collected from SDN switches through Openflow protocol.

* Corresponding author: Tel.: (+84) 903217248
Email: hung.nguyentai@hust.edu.vn

2. The survey of previous works on SDN network management

As said above, network monitoring is generally an important topic but is often underestimated. Network systems are often monitored with multiple mechanisms; For example, measuring latency using ICMP (Internet Control Message Protocol) or querying network node via SNMP (Simple Network Management Protocol). However, these applications need to be adjusted and tested in distributed ways. Therefore, a centralized monitoring server system like Zabbix [2] or Nagios [3] is needed.

In recent years, there are many open source SDN controllers available but OpenDaylight and Floodlight are the most popular ones. Although they support general network monitoring capabilities such as gathering information about switches and traffic through them. However, the form of information provided is not a readable for normal people and also do not provide understanding of current network use. So many previous studies have tried to overcome this with many proposals and solutions for monitoring network based on SDN technology. But most of previous studies focused on measuring and proposing different processes to increase the accuracy during measurement period. In addition, a number of studies have introduced a control module to expand Controller and interaction directly with packet forwarding process. Some other studies only describe few researches samples, not suitable for testing and not implemented with utility software. For example, a few selected researches done before about this SDN monitoring mechanisms and tools are as below.

Jarschel et al. [4] shows the benefits of application control plane, where applications running on top interact with the network itself. The application-state information is used by the controller to choose the best path in the network. Authors were able to achieve reduced bandwidth consumption in the initial ramp-up stages of the transmission when applying to the YouTube streaming application, which is very dynamic, just like VoIP traffic. The initial results were satisfactory however, this approach requires dedicated machine or array of machines used to gather helpful information from the running applications and additional computing resource in the control plane to perform packet analysis task. sFlow, can be used to monitor business-critical applications such as voice, data, video, without having to employ multiple monitoring applications for that purpose. sFlow is implemented in hardware (network switches/routers) and hence it can operate at line speeds without impacting the switch performance considerably. The sampling is done at the hardware ASICs, which makes it simple and accurate. The packet flow sampling mechanism carried out by each sFlow instance must ensure that any packet observed at a data source has an equal chance of being sampled, irrespective of the packet flow to which it belongs. Taking a sample involves either copying the packet's header, or extracting features from the packet. The biggest drawback of sFlow, as well as Cisco's NetFlow is that all devices in the network need to support those technologies for a comprehensive and complete network analysis. OpenNetMon [5] - OpenFlow based controller module uses the idea of per-switch monitoring to enable fine-grained traffic engineering. In order to obtain network metrics, probe packets are send every measuring round. However, it is not possible to measure the performance of each link on a per-flow basis, but only the performance of probe packets being injected onto each path in the network, where there could be many flows in one path. In another work [7] authors propose the per-link monitoring and investigate the ability for the SDN controller to report intermediate MOS (iMOS) for a given set of calls by calculating accurate loss rates and using the simplified Emodel formula detailed in the paper. That work uses OpenFlow switch statistics request, and evaluates the cost associated with sending such messages to all switches in the network. It establishes that the overhead is very low, yet it linearly increases as the number of switches in the network increase. The limitation of that work is that every active and non-active switch is queried every second, and even that it outperforms OpenNetMon as no probe packets are injected, other methods for flow selection should be developed in order to reduce the overhead.

In contrast, BKMON software presented on this article introduces a monitoring method (northbound API) of the controller. This software is fully separate from network components and acts as an extension to monitor network usage.

3. BKMON – an Integrated Architecture for Monitoring SDN Networks

The main idea of BKMON is to implement a network monitoring software that is not only independent of Controller and Switches in the network but also has the ability to monitor and provide specific visualization. Therefore, BKMON does not enforce the flow measurement method which depends on network equipment such as switch, port and traffic frequency, defined by the OpenFlow standard and can be queried and collected by any SDN controller. BKMON is a business application on the network application layer of the SDN model described in Fig. 1.

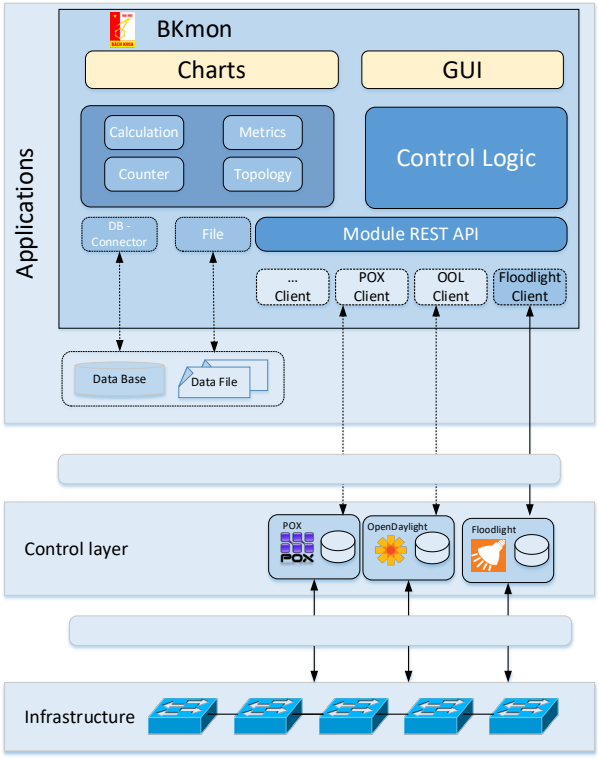

Fig. 1. BKMON Architecture

It interacts with the interface (northbound of SDN controller) through a special application programming interface or API. The architecture of BKMON is based on the classification of software architecture. The lowest layer is the data access layer, which includes databases (SQL database), I/O files and REST (Representation state transfer). This layer provides the basic functions needed for communication with SDN controller. Most open source implementations of SDN controller provide a REST-based Northbound API interface, enabling the creation of an independent language programming interface for network applications. This interface is also used to develop BKMON and allow it to connect to SDN controller. Therefore, the architecture of BKMON contains a logical class called Module REST API, which defines the methods and data models provided by specific communication modules with related SDN controller. The above layer includes data models, which calculates performance indicators using the statistical data provided by SDN controller.

SDN controller also provides information through OpenFlow protocol from network nodes. However, the data model is formed by three main factors; the first is the network topology that includes all network devices (hosts, controller, switches and their connections). Secondly, counters contain statistical data. The final component is the metric, which is used to estimate network performance. Topology and counter models are mainly based on the OpenFlow v1.3 specifications. Therefore, the function of data model transformation obtained by SDN controller does not standardize REST API into BKMON's data model provided by specific REST clients in the data access layer. The top layer contains a graphical interface for user interaction and for graphically show of the collected data. It is a collection of GUI (graphic user interface), tables, graphs or series of options for monitoring and managing network components and traffic. Many tables and charts present flow graphics and performance indicators in real time.

4. Results and discussion

During the development of BKMON, virtual testing environment based on simulating network on Mininet and using Floodlight SDN controller. This testing environment includes two virtual machines (VMs) with Ubuntu version 14.04 Linux operating system added in VMware Workstation Hypervisor on Windows 10 Professional system. The first virtual machine contained a Mininet emulator. Mininet is conFig.d with a topology tree with two levels as described in Fig. 1. During the entire development process, this topology is used on regular basis to get the return parameters from SDN controller. SDN controller is packaged in the second virtual machine and developed directly on the Windows server operating system using Java and Eclipse. At the last stage, BKMON is fed into the real network where SDN Controller will connect to the Gateway with real network traffic to be able to measure and monitor the real network traffic.

Floodlight controller is chosen to deploy BKMON, because it is widely used in research and development environment. However, OpenDaylight was also tested and showed that there is no major technical problem because it also uses REST API for Northbound. So most of the functional blocks of BKMON are built on the Java language. BKMON's function is demonstrated in Fig. 2.

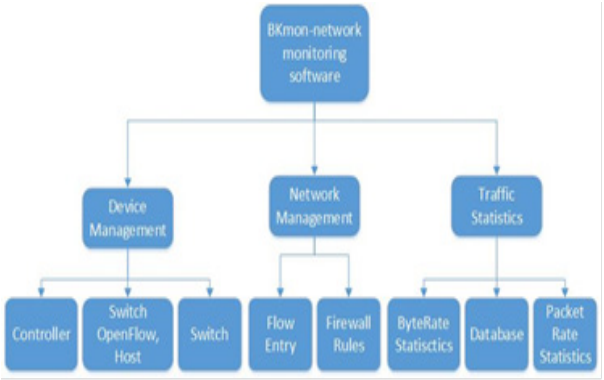

Fig. 2. Functionalities of BKMON

Basically, BKMON has three main functions of device monitoring, traffic monitoring and network management

In device monitoring category, we managed the following information.

- Controller monitoring: IP address, memory usage, status (yes as working normally), list of modules used by controller.

- Other devices monitoring: list down information of all devices found by controller, which consists of Switch OpenFlow and Host.

- Switch monitoring: Real time statistics about switch's physical information, traffic on ports and flows. We use two Openflow 1.3 messages to collect these statistics from respective counters on switches, which are: StatsReq and StatsRes [9]

For network management function, it does following features:

- Collecting information to build and show the network topology

- Switch's flow entries management: help us PUSH or DELETE a static flow in switch's flow table

- Firewall rules management: provide many options to set up our own rules.

For traffic statistics: This funtions support users to identify byte rate, packet rate on all ports or specified port in real-time; Also, BKMON uses the SQLite database platform built into firefox browser to stores parameters of all flows that go through switches for later statistical analysis and dashboard showing. In addition, BKMON also saves all switches's metric in the network that scanned by controller.

Figures below show the GUI and charts of various counters of traffic and switches collected by BKMON.

Fig. 3. Login Interface of BKMON

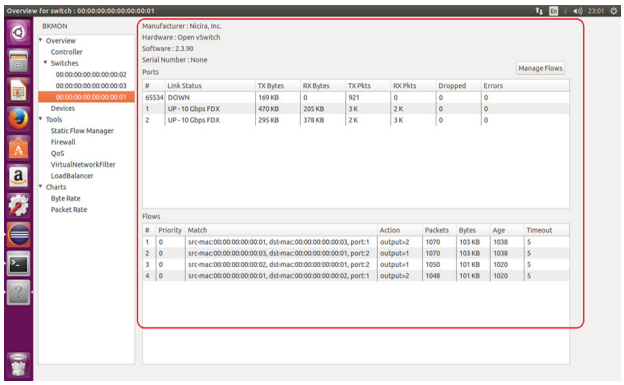

Fig. 4. Main GUI of BKMON

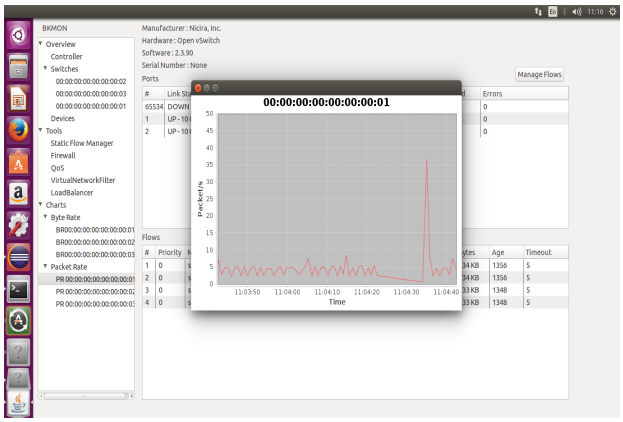

Fig. 5. Chart of packets counter of specific port of a Switch

5. Conclusion

Network management and monitoring the daily network operation are always important topics in research. Furthermore, managing and monitoring the network that utilizes new technologies is extremely difficult. BKMON software is designed and built with the goal of providing a tool for easy managing the components in the network as well as monitoring the status of all hosts, network devices and especially the traffic in the network. BKMON currently only runs in one network domain controlled by a single controller, but it is fully configurable to connect to more controllers and thus allows it to monitor in the inter-domain network environment. In addition, BKMON can also be further invested to develop it in to the commercial products with more advanced and pragmatically features such as mobile alerting when some emergency events occur, etc. This paper provides design descriptions, functions that makes up BKMON and shows the results of running BKMON test in the lab of the Information Technology Center at the Central Party Committee Office.

Acknowledgements

The results of work presented on this paper are possible due to the sponsorship of the National Research Project titled of "Research and development of Internet of Things (IoT) platform, application to management of high technology, industrial zone", coded as KC.01.17/16-20"

References

[1] https://www.opennetworking.org/images/stories/downloads/sdn-resources/IEEE-papers/evolution-of-sdn-and-of.pdf

[2] https://www.zabbix.com/

[3] https://www.nagios.org/

[4] M. Jarschel, F. Wamser, T. Hohn, T. Zinner, and P. Tran-Gia, Sdnbased application-aware networking on the example of youtube video streaming, in EWSDN. IEEE, 2013, pp. 87–92.

[5] N. L. Van Adrichem, C. Doerr, and F. A. Kuipers; Opennetmon: Network monitoring in openflow software-defined networks, in NOMS. IEEE, 2014.

[6] C. Thorpe, C. Olariu, and A. Hava; imos: Enabling voip qos monitoring at intermediate nodes in an openflow sdn, in SDS. IEEE, 2016.

[7] B. Siniarski, C. Olariu, P. Perry, and J. Murphy; Openflow based voip qoe monitoring in enterprise sdn, in IM. IEEE, 2017.

[8] Grover, N., Agarwal, N., Kataoka, K. liteflow; Lightweight and distributed flow monitoring platform for sdn. In: Network Softwarization (NetSoft), 2015 1st IEEE Conference on. 2015, p. 1–9.doi:10.1109/NETSOFT.2015.7116160

[9] http://flowgrammable.org/sdn/openflow/message-layer/statsrequest/