

Performance Assessment for different SDN-Based Controllers

Mohammad Nowzin Amin Sheikh, I-Shyan Hwang, Elaiyasuriyan Ganesan, Razat Kharga
 Department of Computer Science and Engineering
 Yuan Ze University, Chung-Li 32003, Taiwan, 32003
 s1089109@mail.yzu.edu.tw, ishwang@saturn.yzu.edu.tw, s1069107@mail.yzu.edu.tw, s1089107@mail.yzu.edu.tw

Abstract—Software Defined Networking (SDN), a new network architecture in which the control plane and data plane are split, has evolved as a result of technological advancements. A controller is recognized as the key element in the control plane which is responsible for performing all major operations for managing the data plane. Hence, the controller's performance itself is very significant. In this paper, we have evaluated the performance of different SDN-based controllers in terms of response time and throughput. Our analysis indicates that when the number of operations is small, NOX performs better in both cases. As the number of operations increases, NOX has a better response time but ONOS has better throughput. Mininet and Wireshark have been used for simulation purposes. Finally, we concluded about the popular controller performance after studying the test data, which is important when picking the right SDN controller.

Keywords—SDN Controllers, Performance.

I. INTRODUCTION

Software-Defined Networking (SDN) is a revolutionary computer network architecture. In network devices, SDN divides the control and data planes unlike traditional networks. The control plane is the system that efficiently displays all devices and logic that describe how to deliver information or data in the data plane and where to store it [1]. This novel approach proposes the use of a centralized controller to regulate the behavior of all forwarding components in the network. We employed popular controllers in our investigation to evaluate better performance.

In recent years, SDNs have witnessed enormous growth and deployment in various sorts of networks. Datacenter, the Cellular and wide-area [2], Internet of Things (IoT) and wireless networks, as well as security and privacy domain, are all using them. The controller is an important part of the control plane since it is in charge of performing all major data plane tasks. As a result, the controller's performance is crucial. Some of the most well-known controllers include *internal controller*-Network Operating System (NOX), and *external controllers*-Floodlight, Open Daylight (ODL), Pythonic Network Operating System (POX), Open Network Operating System (ONOS), and RYU [1-5].

A few numbers of controller performance in terms of response time and throughput have been considered in previous studies [1-3,6]. To the best of our knowledge, no previous effort has taken into account the performance of so many common controllers. We examined the performance of various common SDN-based controllers in terms of response time and throughput, which is the contribution of this work.

II. SDN CONTROLLERS

The controller is an important part of the control plane since it is in charge of performing all major data plane tasks. The most common SDN controllers are described in this section.

NOX is an internal controller of the SDN ecosystem. It's a platform for creating network control apps in particular. NOX, along with OpenFlow, was first introduced to the community in 2009, originally developed by Nicira Networks and now owned by VMware [2]. However, the NOX is unable to provide the necessary functionalities for QoS-assured SDN to improve Internet services, such as QoS-aware virtual system insertion, end-to-end QoS evaluation, and coordinated efforts among control components in other space networks [3].

The Floodlight Open SDN Controller is an Apache-licensed Java-based enterprise-class OpenFlow controller and allows developer to simply adjust software and construct apps. It is backed by a developer community that includes several Big Switch Networks engineers [2]. The Open Networking Foundation manages the OpenFlow open standard. It defines a protocol that allows a remote controller to change the behavior of a network device using a well-defined "forwarding instruction set."

Open Daylight is a multi-protocol controller infrastructure that is highly available, adaptable, scalable, and built for SDN deployments. It has a model-driven service abstraction that allows users to design applications that work with a variety of hardware and southbound protocols. It serves as an external controller for SDN, assisting not only solution suppliers and developers, but also all users in delivering interoperable, programmable networks to all service providers, businesses, colleges, and other organizations throughout the world [4].

POX is a user-friendly SDN controller that comes packaged with the Mininet SDN network emulator and is used as a learning and prototyping tool in education and research. When POX is started, POX components are Python programs that implement networking functionality. POX includes a few ready-to-use stock components [3]. In academic network research, POX is frequently used to build SDN applications such as network protocol research. Example code for imitating a learning switch and a network-wide switch is accessible as a side effect of its broad academic use, and it can be used as starter code for many programming projects and experiments.

The most popular open source SDN controller for constructing next-generation SDN/NFV solutions is ONOS. It manages network segments such as switches and connections and runs programming projects or modules to deliver correspondence administrations to end-users and neighboring systems (SDN). Common ON.Lab applications include segment routing, multi-layer SDN control, topology watcher, way computation, and SDN-IP peering. It improves a variety of southbound routing, multi-layer SDN control, topology watcher, way protocols, such as OpenFlow, NetConf, and others, for communication with a variety of internet devices and hide the unpredictability of conventions from different parts of the controller stage [3].

RYU Controller is an open-source SDN controller that improves the system's agility by making it easier to monitor and alter how traffic is handled. It has programming segments and application program interfaces (APIs) that allow creating new system executives and control applications easy for designers. This segmented architecture enables organizations to modify arrangements to match their specific needs. RYU supports a variety of conventions for managing system devices, including OpenFlow, NetConf, and OF-Config and, so on [3].

III. EXPERIMENT SETUP

In our study we have measured throughput and response time where throughput is measured as the maximum amount of data that can be processed between two nodes in one millisecond and response time is measured as total amount of time in which one operation is responded. For throughput calculation we have used number of operations vs response rate (flow/ms) method. We have taken four operations at a time and calculated the average value for both the response time and throughput.

For simulation purposes, we have considered the architecture are shown in Fig. 1 and described in ref [2,3]. Nine routers are connected to three cloud interfaces in the architecture. These multiple routers are connected to eight IP-addressed hosts. The routers in every network are connected. Three networks are linked to the SDN controller, which also serves as a response time and throughput monitor. After connecting each network's routers, the network will be connected.

The architecture is designed in the CISCO environment and implemented in the Mininet environment in Linux Operating System. The topology is created using Python language and the Ping command has been used to check the connection establishment. The packet has been analyzed using Wireshark.

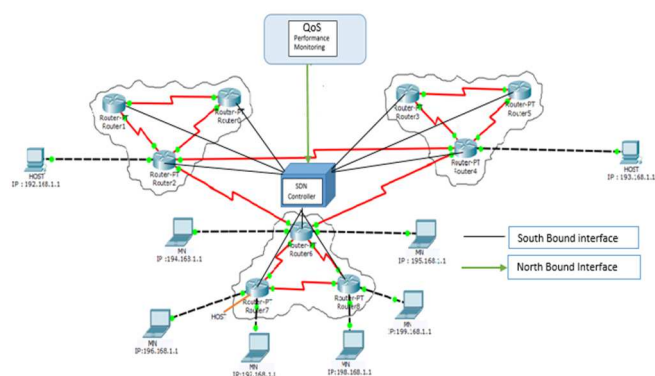


Fig. 1. SDN Architecture and its real-time loop [2,3].

IV. RESULT AND DISCUSSION

Performance of SDN controllers may vary due to numerous reasons. Mostly it depends on the right selection of controllers when the number of operations or the number of switches increases or decreases [6]. Network topology in which we are working also affects the performance as well.

For performance evaluation the fact that we have taken into consideration is the number of operations increase or decrease. From Fig. 2, we see that NOX has the least response time and ODL has the worst response time. It is independent

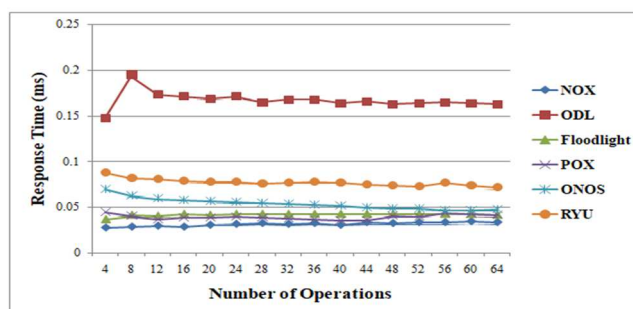


Fig. 2. Response Time of Popular Controllers.

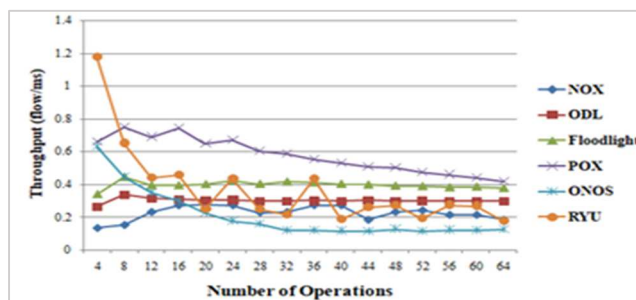


Fig. 3. Throughput of Popular Controllers.

of the number of operations. ODL has the 2nd worst response time where Floodlight and POX have nearly the same response time. ONOS and RYU show average response times. From Fig. 3, we see that throughput is dependent on the number of operations. When the number of operations is less NOX and ODL performs well in terms of throughput but as the number of operations increases, ONOS and RYU start performing better. NOX and Floodlight have the worst throughput whether the number of operations is more or less.

From the above analysis, we suggest picking ONOS or RYU controllers when there is a need to pick the best-performing controller. Further research needs to be carried to finally establish our assumption. We have a plan to observe the performance of these controllers by providing different loads on different scenarios. There are plans to work with switch capacity, number of queues impact, quality of experience (QoE) evaluation, and bandwidth isolation in the future.

REFERENCES

- [1] A. Hossain, M.N.A. Sheikh, S.S.M. Rahman, S. Biswas, Md.A.I Arman, "Enhancing and measuring the performance in software defined networking," *International Journal of Computer Networks & Communications*, vol. 10, no. 5, Sept. 2018.
- [2] M.N.A. Sheikh, M. Halder, S.K. S. Kabir, R.N. Mohalder, "Performance evaluation on software defined networking through external controller Floodlight and internal controller NOX," *International Journal of Scientific & Engineering Research*, vol. 9, no. 7, July 2018, ISSN 2229-5518.
- [3] M.N.A. Sheikh, "SDN-based approach to evaluate the best controller: internal controller NOX and external controllers POX, ONOS, RYU," *Global Journal of Computer Science and Technology*, vol. 19, no. 1, 2019.
- [4] S. Badotra, J. Singh, "Open daylight as a controller for software-defined networking," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 1105-1111, 2017.
- [5] L. Zhu, Md. M. Karim, K. Sharif, F. Li, X. Du, M. Guizani, "SDN controllers: benchmarking & performance evaluation," arXiv preprint arXiv:1902.04491, 2019.
- [6] S. Islam, M.A.I. Khan, S.T. Shorno, S. Sarker and M.A. Siddik, "Performance evaluation of SDN controllers in wireless network," *1st International Conference on Advances in Science, Engineering and Robotics Technology*, 2019, pp. 1-5.