REVIEW COMPARISON AND ANALYSIS OF POX AND RYU CONTROLLER SDN

Shujairi Murtadha

Doctoral Degree Program (1), FEEC BUT

xshuja00@vutbr.cz

Supervised by Vladislav Škorpil

E-mail: skorpil@feec.vutbr.cz

Abstract: SDN it is a technology that changed the concept of Conventional Networks by making all control functions in one central place and centralized decision-making. The Control Plane is an essential and important part of the SDN architecture, as well as working on Controllers, knowing their types, and comparing them in terms of performance and efficiency. this paper deals with the performance comparison of two Controllers POX and RYU.

Keywords:: SDN, POX, RYU

1 INTRODUCTION

Software-Defined Network (SDN) is a new networking paradigm whose architecture shifted from its old distribution model to a centralized design. The latter structure is characterized by the separation of control planes and data. The Control Plane comprises the Controller, while the data encompasses forwarding elements, Routers, and Switches [22]. SDNs have recorded a massive deployment and growth in varying forms of networks. Their utilization is in cellular and wide area networks, internet of things and wireless networks, and datacenter networks [26]. Compared to the old system model, SDN has a particular uniqueness entailing multilayered structure for safeguarding efficient traffic flow and forwarding within its working [18].

The multilayered structure is replacing the traditional complex and inflexible networks with innovative Controllers, fixing different limitations, such as mobility, scalability, security, debugging, and manual configuration. Some of the SDN Controllers include Floodlight, POX, NOX, Beacon, Open Daylight, ONOS, and RYU, and they act as brains of SDN management [11]. This discussion critically analyzes and compares the RYU and POX controllers in which the work distinguishes the two controllers' functionality to highlight their differences and similarities.

2 POX ANALYSIS

POX is an SDN Controller that originated from NOX, an indigenous OpenFlow Controller. Its repository has various branches [14]. According to [4], POX is Apache-licensed and written in Python. Individuals can utilize the controller to understand SDN concepts, for it is easy. The Controller is used for faster prototyping and development of modern network applications. Besides, the design often comes pre-installed by the Mininet tactic machine. A user can use the POX Controller to turn dumb OpenFlow gadgets into a firewall, load balancer, switch, or hub device [8]. The regulator permits easy means of running SDN or OpenFlow experiments. One of its unique attributes is that the system can be passed within varying parameters according to experimental or real topologies, allowing its user to experiment with a Mininet emulator, testbeds, or actual hardware [8]. The advantage of using the POX Controller is its requirement for minimal memory space for its performance compared to other archetypes. The structure has considerable throughput outcomes than other SDN Controllers. POX provides productivity, effective documentation, language support, open-source, and

platform [19]. This model supports visualization tools, graphic user interfaces, and simple architecture [23]. Command-line interface Syntax, together with varying components and options, is required during POX operations [7]. The POX Controller can provide better results than NOX If properly operated.

3 RYU ANALYSIS

The RYU Development Team (2020) considers the RYU Controller a component-based application elaborated on the networking framework. RYU offers freeware aspects that are adequately defined, simplifying designers' ability to develop modern control applications and network management (RYU Development Team, 2020. The RYU Controller has a simple structure due to its Mininet command (RYU project team, n.d.). The design has a unique way of operation. It functions under Apache 2 licenses and an open source, written based on Python, and deployed and supported by NTT cloud information centers. RYU's primary source code is located on GitHub, and the Open RYU society is its supporter and provider [6]. One might also be interested in its composition. RYU applications comprise a variety of components crucial for SDN applications. Under the RYU Controller, an individual can adjust existing apparatuses and develop new elements. The reason is RYU Controller entails a software environment to facilitate building a network distribution within a Mininet system, denoting its advantages when executing operations requiring SDN architecture [9]. According to [16], one of RYU's strengths is its supports for Nicira extensions and numerous southbound procedures for controlling devices like Configuration Protocol, OpenFlow Management, and NETCONF. Still, the controllers also comprise Router Northbound Applications and a Simple Switch [17]. The Northbound software regulates networks via centralized controllers, permitting other systems to be more flexible and dynamic while adjusting to the structure's conditions.

4 COMPARISON

Pox, a modern python-based NOX version, has great-level SDN APIC supporting virtualization and includes a query-able topology graph. Unlike RYU, POX is not entirely written using Python. Some of the RYU components include messaging, reusable libraries, application management, event management, and OpenFlow support (SDN Tutorial, n.d.). These components have only been proved to be more existent within the RYU control and not the POX control. Regrading POX, its features are different Python applications that can be invoked if the controller starts from the command line [10]. Additionally, POX components implement software functionality. The Python-based controller is vastly utilized within the research community because of how easily it can be programmed. The structure provides a platform for rapid development and prototyping software that manages network gadgets within an OpenFlow network (“Mininet and Pox”). The Controller can be remotely connected with various applications like intrusion detection, load balancer, firewall, and routing, and Mininet [3]. The application can be installed directly from its repository, GitHub, and it is built within Mininet. Contrary, the RYU Controller was developed from a Japanese word that meant flow. This regulator, like POX, is Python-based and an OpenFlow with an efficient API via which developers can manage, control, and program applications. Still, OF-conf, OpenFlow, and Netconf, which are RYU protocols, can be utilized to configure network devices [25]. The RYU has its code accessible within its Apache 2.0 license besides well-organized documentation for facilitating diverse SDN contexts [15]. The provided attributes denote exceptional strengths and limitations of SDN scenarios in their different functionalities.

In most cases, RYU is utilized to gather statistical data from switches. Hence, it can be configured as a switch, firewall router, and traffic monitor. Both controllers have the forwarding, control, and application layers. [1] state that the three layers exchange information through the southbound and northbound application programs. Southbound is often between the forwarding and applications layer. At the same time, the northbound is between the control and applications layer. Both regulator's data plan, architecture, control logic, and control plan are implemented using a centralized approach managed by an intelligent design [5]. POX currently exchanges data with Open Flow v1 Switches,

with outstanding support from Open vSwitch. Conversely, RYU uses its components that include messaging supporting architecture developed using other languages [6]. Another exceptionality is that controllers help SDN separate data plane from Control Plane [2]. These attributes indicate the structure facilitating control logic within a central Controller. Regardless of the difference between the two Controllers, they play a crucial role in SDN operations. Figure1 shows the properties of both controllers.

Features	POX	RYU
GUI	Yes	Yes
OpenFlow Version	V1.0	V1.0, v1.2, v1.3 v1.4, v1.5, Niciria extensions
Language Support	Python	Python
REST API	Yes	Yes
Platform Support	Linux, Windows, Mac	Linux
License Provider	Apache	Apache
Learning Curve	Easy	Moderate
Distributed	No	Yes

Table 1: Properties of POX and RYU

5 CONCLUSION

Network operators locate and operate centralized consoles, so their performance is important to network management. In this paper, we study, analyze, and compare two controls (POX, RYU) and their characteristics.

In our future work, we aim to create, complex simulations, and test the efficiency of multiple controllers, using tools that create the load on the controller also using a DDoS attack to find out the efficiency of each of the controllers.

REFERENCES

[1] Abdullah, M. Z., Al-awad, N. A., & Hussein, F. W. (2018). Performance comparison and evaluation of different software defined network controllers. International Journal of Computing & Network Technology, 06(02), 36-41. doi:10.12785/ijcnt/060201

[2] Ali, J., Lee, G., Roh, B., Ryu, D. K., & Park, G. (2020). Software-defined networking approaches for link failure recovery: A survey. Sustainability, 12(10). doi:10.3390/su12104255

[3] Ali, J., Lee, S., & Roh, B. (2018). Performance analysis of POX and Ryu with different SDN Topologies. Proceedings of the 2018 International Conference on Information Science and System - ICISS '18. Retrieved from https://booksc.xyz/book/72816877/06d6d0

[4] Altangerel, G., Chuluuntsetseg, T., & Yamkhin, D. (2019). Performance analysis of IPv6 transition technologies and transition methods. The 13th International Forum on Strategic Technology (IFOST 2018). Retrieved from https://www.researchgate.net/publication/332466286_Performance_analysis_of_SDN_controllers_POX_Floodlight_and_Opendaylight

[5] Asadollahi, S., Goswami, B., & Gonsai, A. M. (2017). Software-defined network, controller comparison. International Journal of Innovative Research in Computer and Communication Engineering, 5(2), 211-217. Retrieved from https://www.researchgate.net/publication/320346909_Software_Defined_Network_Controller_Comparison

[6] Asadollahi, S., Goswami, B., & Sameer, M. (2019). Ryu controller's scalability experiment on software defined networks. 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC). doi:10.1109/ICCTAC.2018.8370397 Retrieved from

[7] Bholebawa, I. Z., & Dalal, U. D. (2017). Performance analysis of SDN/OpenFlow controllers: POX versus floodlight. Wireless Personal Communications, 98(2). doi:10.1007/s11277-017-4939-z

[8] Chauhan, N., & Sood, m. (2019). Performance analysis of POX, open v switch, and opendaylight SDN controllers on the cloud. International Journal of Innovative Technology and Exploring Engineering, 8(8S3), 332-339. Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v8i8s3/H10900688S319.pdf

[9] Eissa, H. A., Bozed, K. A., & Younis, H. (2019). Software Defined Networking. In 2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (pp. 620-625). IEEE. doi:10.1109/STA.2019.8717234

[10] Elmedani, M. K. (2017). Performance evaluation of software-defined networking compares to traditional networks (Master's degree. Sudan University of Science and Technology). Retrieved from http://repository.sustech.edu/bitstream/handle/123456789/18764/Performance%20Evaluation%20of%20Software....pdf?sequence=1&isAllowed=y

[11] Islam, T., Islam, N., & Refat, A. (2020). Node to node performance evaluation through the RYU SDN controller. Wireless Personal Communications. Retrieved from https://www.researchgate.net/publication/338703576_Node_to_Node_Performance_Evaluation_through_RYU_SDN_Controller

[12] Kaur, S., Singh, J., & Ghumman, N. S. (n.d.). Network programmability using the POX controller. Department of Computer Science and Engineering, 134-138. Retrieved from http://www.baburd.com.np/material/NG/Paper2-SDN-POX-Controller.pdf

[13] Mininet and pox. (n.d.). Retrieved from https://pld.cs.luc.edu/courses/netmgmt/sum17/notes/mininet_and_pox.html

[14] Muragaa, W. H., Seman, K., & Marhusin, M. F. (2017). A POX controller module to prepare a list of flow header information extracted from SDN traffic. International Journal of Computer and Systems Engineering, 11(12). Retrieved from https://zenodo.org/record/1132991/files/10008318.pdf

[15] Nazir, F., Humayun, Q., Ahmad, R. B., & Elias, S. J. (2017). Software-Defined Network Testbed Using ZodiacFX a Hardware Switch for OpenFlow. EAI Endorsed Transactions on Scalable Information Systems, 4(14), 1-6. doi:10.4108/eai.25-9-2017.153150

[16] O'Briain, D. (2019). RYU SDN. Testbed manual version 1.6. Institute of Technology Carlow. Retrieved from http://www.obriain.com/training/sdn/RYU_Soft_Testbed_v1.6_odt.pdf

[17] Pham, D., & But, J. (2016). The Ryu action node v1.01. CAIA Technical report 161006A. Retrieved from http://caia.swin.edu.au/reports/161006A/CAIA-TR-161006A.pdf

[18] Quincozes, S. E., Soares, A. A., Oliveira, W., Cordeiro, E., Lima, R. A., Muchaluat-Saade, D., & Albuquerque, C. (2019). Survey and comparison of SDN controllers for teleportation and control power systems. Retrieved from http://dl.ifip.org/db/conf/lanoms/lanoms2019/196234_1.pdf

[19] Rowshanrad, S., Abdi, V., & Keshtgari, M. (2016). Performance evaluation of SDN controllers: Floodlight and OpenDaylight. IIUM Engineering Journal, 17(2), 47-57. doi:10.31436/iiumej.v17i2.615

[20] RYU Development Team. (2020). RYU Documentation. Release 4.34. Retrieved from https://readthedocs.org/projects/ryu/downloads/pdf/latest/

[21] RYU project team. (n.d.). RYU SDN framework. Using OpenFlow 1.3. Retrieved from https://osrg.github.io/ryu-book/en/Ryubook.pdf

[22] Salman, O., Elhajj, I. H., Kayssi, A., & Chehab, A. (2016). SDN controllers: A comparative study. 2016 18th Mediterranean Electrotechnical Conference (MELECON). Retrieved from https://www.researchgate.net/publication/304457462_SDN_controllers_A_comparative_study

[23] Saritakumar, N., Srinivasan, A. V., Albert, E., & Rani, S. S. (2019). Performance evaluation of the pox controller for software-defined networks. International Journal of Innovative Technology and Exploring Engineering, 8(9S2), 679-684. Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v8i9S2/I11390789S219.pdf
[24] SDN Tutorial. (n.d.). SDN controllers. Retrieved from https://cs.wmich.edu/~alfuqaha/spring16/cs6560/modern/lec4-SDN.pdf
[25] Tseng, Y. (2018). Securing network applications in software-defined networks (Diss., Paris Descartes University).
[26] Zhu, L., Karim, M., Sharif, K., Li, F., Du, X., & Guizani, M. (2019). SDN controllers: Benchmarking & performance evaluation. Retrieved from https://arxiv.org/pdf/1902.04491.pdf