Performance Evaluation on Software Defined Networking through External Controller Floodlight and Internal Controller NOX

Mohammad Nowsin Amin, Monishanker Halder, SK. Shalauddin Kabir, Rathindra Nath Mohalder

Abstract— Software Defined Networking (SDN) is a network design approach that empowers the system to be astutely and midway controlled, or 'customized,' utilizing programming applications. It promises to deliver a solid method to advise Quality of service (QoS) ideas in communication systems. It is an intense innovation that guarantees to give a superior method to present Quality of Service (QoS) approach in the present correspondence systems. SDN linguistically alters the conduct and usefulness of network instruments utilizing single high state program. The acknowledgment of Quality of Service (QoS) ideas ends up conceivable in an adaptable and dynamic way with SDN. It gives a few advantages including, administration and system adaptability, enhanced activities and elite performances. This paper will center on the Quality of service (QOS) like response time, delay, throughput, and other execution estimating parameters of our proposed network architecture utilizing inward controller e.g. Network operating system (NOX) and outer controller e.g. Floodlight. In spite of the fact that ideas of QoS are very much investigated, they were not understood in communication networks because of high usage many-sided quality and acknowledgment costs. It will chiefly focus on the external controller and internal controller performance in the proposed network architecture. These perceptions of switch assortment may give SDN application developer's bits of knowledge while acknowledging QoS ideas in a SDN-based framework.

Index Terms— SDN, QoS, NOX, API, IP, ASIC, BNC, NFV, C-RAN, POX, ODL, MAC, CPU.

———————————— u ————————————

1 INTRODUCTION

Software Defined Networking (SDN) is a novel way to deal with distributed computing that simplifies network administration and empowers automatically proficient system design keeping in mind the end goal to enhance network execution and observing [1].SDN is a developing technology that is dynamic, sensible, financially savvy, and versatile, making it perfect for the high-transfer speed, dynamic nature of the present applications. It separates the system control and sending functions empowering the network control to end up specifically programmable and the basic foundation to be dreamy for applications and system administrations. The OpenFlow convention is a foundational component for building SDN arrangements. SDN is another extend technology that intercept the intelligence of network (i.e. manage) from commencing network devices such as witches, hubs, routers etc. Its primary recognizing issue is that partition of the information plane from the control plane in switches and routers [2]. In Software-defined networking, system administrator can oversee network components by running programming on an outside server. SDN executes functionality and network behavior of system gadgets. The information plane is sent through the system-network, for example, the equipment and parcels that is utilized to make it dynamic, for instance, switches. The control plane is the framework that shows proficiently all gadgets and logic which are in charge of portraying how to sent data or information in information plane and where to store. SDN envelops various sorts of network advancements intended to make the system more adaptable and spry to help the virtual server and storage framework of the cutting edge server farms. We will without a great deal of a comprehend SDN against standard system by a prime representation; accept inside the bundles that we need to pass on a package in standard system, it must change its course item times for finding the perfect way. It is a capable framework to utilize better Quality of Service (QoS) which demonstrates to a system's productivity to accomplish most noteworthy transmission capacity and manage other system execution components like latency, error rate and uptime [3]. SDN it consequently finds the all conceivable and most brief course to deliver the packets. By isolating of control plane from information plane in SDN some controllers are expanded its adaptability in sending new administrations (e.g., virtual private system, distributed computing), program capacity in open API, unwavering quality in merged IP network. SDN is one sort of practical substitute to previous kinds of network administration since it causes developer to utilize the less expensive stock switches which give favored control of movement. Network engineers and designers can help exchanging textures through equipment from various mongers and models comprehensive of coordinate switches with ASICs and those without. Now days, in network framework OpenFlow is the most prevalent specialized particular for SDN and grants to control remote segment of routing tables [4].

· *Mohammad Nowsin Amin Sheikh is serving as an Assistant Professor in the department of Computer Science and Engineering in Jessore University of Science and Technology, Bangladesh, E-mail: nowsin.jstu@gmail.com*
· *Monishanker Halder is serving as a Lecturer in the department of Computer Science and Engineering in Jessore University of Science and Technology, Bangladesh, E-mail: monicsejust@gmail.com*
· *SK.Shalauddin Kabir is a candidate of B.Sc. degree in the department of Computer Science and Engineering in Jessore University of Science and Technology, Bangladesh, E-mail: riponcse32@gmail.com*
· *Rothindronath Mohalder is a candidate of B.Sc. degree in the department of Computer Science and Engineering in Jessore University of Science and Technology, Bangladesh, E-mail: mr.rathi.just@gmail.com*

Network operating systems (NOX) is the primary and essential OpenFlow internal controller of SDN. It distributes as a system control organize, which make strides an abnormal state automatic interface for the advancement and administration of system control requests. NOX was first acquainted with the group in 2009 — at first created by Nicira Networks and now possessed by VMware, alongside OpenFlow. Its applications are various types of focused projects on high level of absence of steadiness in network execute segment, dissimilar to bring down back of calculation course of action [5][6] applications. The network operating system does not deal with the system itself; It furnishes a programming interface with top state objects, (for example, plate stockpiling volume, CPU processing power, memory, connect control, and so forth.) of system assets, which empowers organize application projects to deal with secure and practical complex assignments on a wide assortment of networks [6]. The NOX – whose deliberated execution propelled a few late propositions on enhancing control plane effectiveness has a low stream setup throughputs and expansive stream setup inactivity. Yet, the NOX can't legitimately supply the mandatory capacities for QoS-guaranteed software defined networking (SDN) [7][8] to give better Internet administrations, for instance, QoS-watchful virtual framework embeddings, end-to-end mastermind QoS assessment, and facilitated endeavors among control parts in other space systems.

Floodlight is the leading internal controller of SDN. The Controller, an Apache authorized, Java-based OpenFlow controller, is one of the critical commitments from Big Switch Networks to the open source group. Floodlight's engineering depends on Big Network Controller (BNC), the organization's business advertising. Open Floodlight [9] is a well known execution of an OpenFlow controller, being both allowed to utilize and moderately simple to get up and running. Solomon, et. al. [10] have set up a test connect with an Open Floodlight controller, dealing with a system of switches executed with Open Switch, a free OpenFlow-empowered switch that keeps running on a broadly useful processor. The v1.2 Floodlight discharge expands upon the upgrades made in v1.1, with accentuation on more vigorous and highlight rich center modules, including totally modified Device, Topology, and Forwarding modules. Another Statistics module gives port transmission capacity insights and furthermore serves a multi threading and measurements accumulation case. IPv6, Link idleness, OF-DPA, message audience members, and numerous all the more energizing highlights have additionally been presented [11]. The Floodlight Controller is perfect with Openstack, an arrangement of programming apparatuses that assistance manufactures and oversees distributed computing stages for both open and private clouds.

2 Related Word

Previous works on giving better QoS security utilizing OpenFlow can be partitioned in three classes. To start with, studies about were expanding dynamic QoS in a SDN condition [12], [13], [14]. Second, propelled considers were on switch assortment [15], [16], [17], [18].Third, explore on SDN system to accomplish better execution from QoS were utilizing OpenFlow-qualify switches [19], [20]. A few research endeavors have been examining the utilization of Software-defined networking (SDN) and Network Functions Virtualization (NFV) as answers for the previously mentioned issues. For instance, the idea of Cloud Radio Access Network (C-RAN) has been propelled by China Mobile as a developing umbrella vision including participation radio with high higher range effectiveness, open stage genuine time cloud foundation and BS virtualization innovations [21].

In recent years, various specialists have been done to analyze SDN controllers like POX, NOX, ODL, Floodlight and so forth. Other research paper had ascertained the starting results of benchmarking through Open-Daylight SDN outer Controller with Floodlight controller. Scientist had estimated throughput, idleness and reaction time of Open Daylight SDN Controller and Floodlight under different circumstances [22]. A related task of specific note is Maestro [23] (created in parallel to NOX), which is additionally charged as a "system working framework". Through NOX controller 4D frameworks is to control sending (e.g., FIBs in switches), and in this manner their system see just incorporates the organize foundation (e.g., joins, switches/switches). The Rational [24] and Ethane [25] ventures gave a more extensive class of usefulness by including a namespace for clients and hubs in their system view and monitoring the ties between these names and the low-level MAC and IP addresses.

Shalimov et al. [26] proposed a system to test and look at different open-sources SDN or OpenFlow controllers, for example, floodlight. Execution parameters including inertness and throughput, unwavering quality, adaptability, reaction time, deferral and security were formulated for correlation. The exploration of execution and adaptability were finished with Cbench. Security and dependability test were altered with hcprobe.

3 Description Of The Proposed Architecture

To quantify the Quality of Service (QoS) like reaction time, throughput, delay and other execution estimating parameters, we created SDN based cloud architecture.

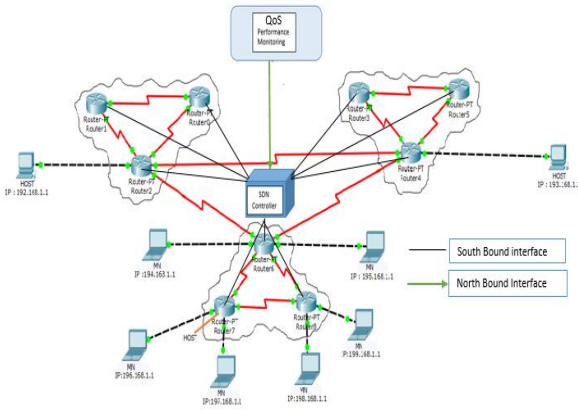

Fig. 1. Proposed_SDN_architecture and its real-time loop

In our developed architecture, there are three cloud interfaces with nine switches. These numerous switches are associated with eight hosts which have possessed IP address. For observing QoS exhibitions from hosts to switches, SDN controllers are utilized like existing controllers and outer controllers.

3.1 Architectural description based on NOX controller

Network operating system (NOX) is one sort of process that makes organization applications. These applications are developed as brought together control programs over abnormal state repudiations of system having which are utilized as an inverse to the appropriated framework calculations above low-level locations [27], [28]. NOX does not manage the framework itself; it gives a programming interface irregular state impressions of framework resources (e.g., memory, circle amassing volume, plate storing volume, CPU dealing with control, associate breaking point and so on.) that enable mastermind application ventures to perform jumbled endeavors safely and profitably on a sweeping heterogeneity of framework organization progressions [27]. For another case, NOX slips by the major capacities with regards to QoS-guaranteed programming described sorting out (SDN) [29] advantage provisioning on bearer survey provider Internet, for instance, QoS-careful virtual framework seating, end-to-end orchestrate QoS estimation, and support among control segments in other space compose.

3.2 Architectural Description Based On Floodlight Controller

Floodlight, which is composed in Java, is an elite, open source OpenFlow controller. Floodlight was created based on Beacon, a trial OpenFlow controller from Stanford University, and it is currently upheld by a vast engineer group. Big Switch Networks backs Floodlight as an organization that basically offers answers for business datacenters. Floodlight offers various highlights and deliberations for controlling an OpenFlow network. For ideal use of assets, Floodlight depends on multithreading and can deal with a few million new streams for each second. The Westbound Java API permits the improvement of custom modules in Java and fast interfacing with the center controller. The modules are stacked by means of a different module framework when the Floodlight controller begins. You would thus be able to use the full usefulness of the controller and OpenFlow API and expeditiously react to occasions on the system, for example, the rise of new bundles or new streams [30]. The Open Flow datapath notwithstanding QoS modules shape the QoS Flow datapath. This datapath is a use space use where lines are arranged in the segment space. The QoS module opens a channel with the bit through Netlink to relate for both use and bit space. In this way, the Floodlight can be instantiated to engage action embellishment and enqueuing of streams in our proposed architecture.

4 Comparison Between The Internal And External Controller On Our Proposed Architecture

4.1 NOX Controller

4.1.1 Response Time of NOX Controller

Figure 2 shows the graph of the calculation of Response Time with NOX of Table 1.

TABLE 1

Calculation Of Response Time With NOX Controller

Number of Clients	Response Time (NOX)
4	0.028
8	0.029
12	0.030
16	0.029
20	0.031
24	0.032
28	0.033
32	0.032
36	0.033
40	0.031
44	0.034
48	0.033
52	0.034
56	0.034
60	0.035
64	0.034

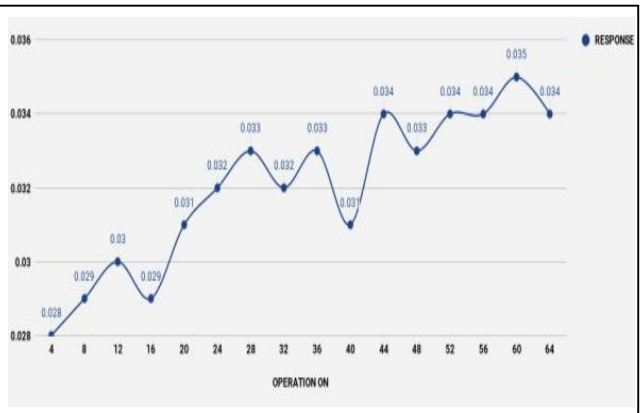

Fig. 2. Response Time of NOX controller

4.1.2 Throughput of NOX Controller

Figure 3 shows the graph of the calculation of Throughput with NOX of Table 2.

TABLE 2

CALCULATION OF THROUGHPUT WITH NOX CONTROLLER

Number of Clients	Throughput (NOX)
4	0.138
8	0.156
12	0.234
16	0.274
20	0.279
24	0.274
28	0.231
32	0.234
36	0.275
40	0.274
44	0.187
48	0.234
52	0.244
56	0.217
60	0.217
64	0.187

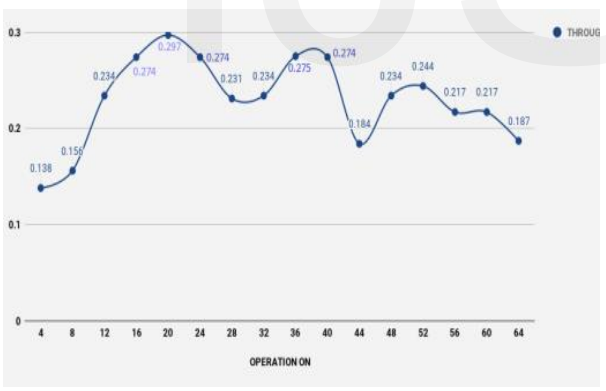

Fig. 3.Throughput of NOX controller

4.2 FLOODLIGHT Controller

4.2.1 Response Time of FLOODLIGHT Controller

Figure 4 shows the graph for the calculation of Response Time with Floodlight of Table 3.

TABLE 3

CALCULATION OF RESPONSE TIME WITH FLOODLIGHT CONTROLLER

Number of Clients	Response Time (NOX)
4	0.028
8	0.029
12	0.030
16	0.029
20	0.031
24	0.032
28	0.033
32	0.032
36	0.033
40	0.031
44	0.034
48	0.033
52	0.034
56	0.034
60	0.035
64	0.034

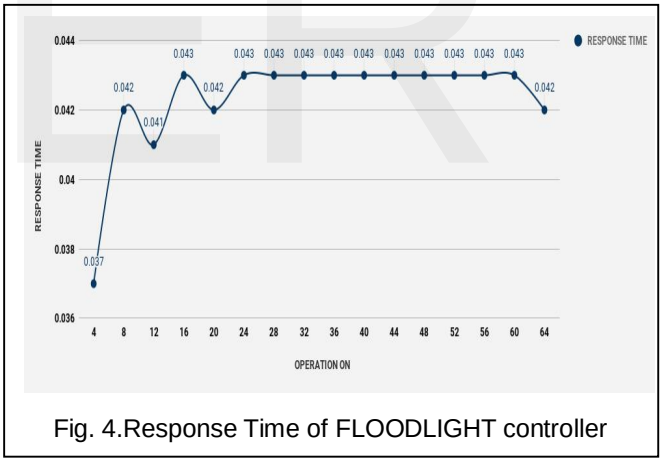

Fig. 4.Response Time of FLOODLIGHT controller

4.2.2 Throughput of FLOODLIGHT Controller

Figure 5 shows the graph of the calculation of Throughput with FLOODLIGHT of Table 4.

TABLE 4

CALCULATION OF THROUGHPUT TIME WITH FLOODLIGHT CONTROLLER

Number of Clients	Throughput
4	0.341
8	0.447
12	0.396
16	0.395
20	0.402
24	0.426
28	0.404
32	0.418
36	0.413
40	0.404
44	0.400
48	0.393
52	0.390
56	0.383
60	0.387
64	0.379

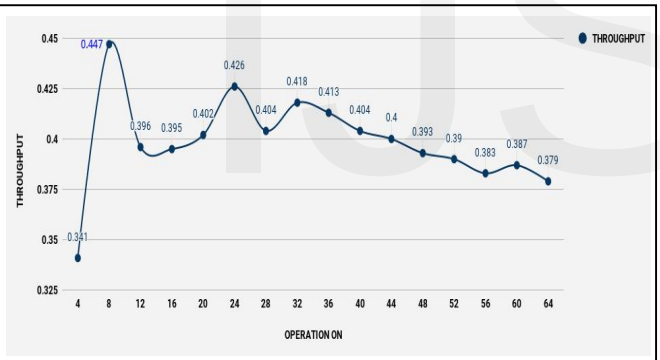

Fig. 5.Throughput of FLOODLIGHT controller

5 COMPARISON BETWEEN NOX CONTROLLER AND FLOODLIGHT CONTROLLER

5.1 Response Time

Response time of external controller Floodlight is better than response time of internal controller NOX on our proposed architecture.

Fig. 6. Response Time of using controllers on proposed architecture

5.2 Throughput

Throughput of external controller Floodlight is better than response time of internal controller NOX on our proposed architecture.

Fig. 7. Throughput of using controllers on proposed architecture

6 CONCLUSION AND FUTURE WORK

SDN (Software Defined Networking) is the most recent buzzword in IT, getting more prevalent consistently. It is an advancing server farm arrangement that moves the greater part of the nearby complexities back to a brought together, controlling gadget or gadgets. With SDN the product that dwells on these controllers settles on the larger amount choices and sends this data down to each physical device. The fundamental objective of this work is enabling included esteem benefits in organize system.SDN use for this work as well as utilized for discovering the security of another reason in various system frameworks. It approves a basic and versatile affirmation of existing dynamic Quality of Service (QoS) parts in the present correspondence organize. In the comparison of response time of proposed architecture Floodlight outer controller is better than internal controller NOX. In the case of throughput outer controller is also better. Later on, we will work with Number of Queues Impact, Bandwidth Isolation, QoE Evaluation and Switch Capacity. We will also work on load balance, security systems, wireless network, Secure Mo-

bility, Cloud Networking and so on. We will work on IOE or IOT like home automation, industry, city, nation and so on through unique Quality of Service (QoS) by joining our proposed SDN architecture with various controllers.

REFERENCES

1. Benzekki Kamal et al.Software-defined networking (SDN): a survey. Security and Communication Networks 9, no. 18 (2016): 5803-5833.
2. International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 11, November 2014.
3. "Control of Multiple Packet Schedulers for Improving QoS on OpenFlow/SDN Networking - IEEE Xplore Document." 12 December 2013.
4. OpenFlow Switch Specification: Version 1.1.0 Implemented. (2011, Feb 28). Open Networking Foundation. Retrieved from https://www.opennetworking.org
5. ArsalanTavakoli et al, "Applying NOX to the Datacenter," in Proc. Of SIGCOMM Hotnet 2009.
6. Natasha Gude et al., "NOX: Towards an Operating System for Networks," editorial note submitted to CCR.
7. A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Towards a next generation data center architecture: scalability and commoditization. In PRESTO, 2008.
8. DimitriStaessens et al., "Software Defined Networking: Meeting Carrier Grade Requirements," in Proc. of IEEE Workshop on Local & Metropolitan Area Networks (LANMAN), 2011.
9. Project Floodlight. [Online] http://www.projectfloodlight.org/floodlight.
10. Solomon, Nir, Francis, Yoav, and Eitan, Liahav. Floodlight OpenFlow DDoS. [Online] September 2013. [Cited: December 27, 2013.] http://www.slideshare.net/YoavFrancis/floodlightopenflow-ddos.
11. OpenFlow Switch Specification Version 1.0.0. OpenFlow. [Online] [Cited: December 22, 2013.] http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf.
12. T. Zinner, M. Jarschel, A. Blenk et al., "Dynamic application-aware resource management using software-defined networking: implementation prospects and challenges," in Proc. of the 2014 IEEE Network Operations and Management Symposium (NOMS '14), Krakow, Poland, 2014, pp. 1–6.
13. P. Georgopoulos, Y. Elkhatib, M. Broadbent et al., "Towards networkwide QoE fairness using OpenFlow-assisted adaptive video streaming," in Proc. of the 2013 ACM SIGCOMM Workshop on Future Human- Centric Multimedia Networking (FhMN 2013), Hong Kong, China, 2013, pp. 15–20.
14. A. Lazaris, D. Tahara, X. Huang et al., "Tango: simplifying SDN control with automatic switch property inference, abstraction, and optimization," in Proc. of the 10th ACM International on Conference on emerging Networking Experiments and Technologies (CoNEXT), Sydney, Australia, 2014, pp. 199–212.
15. Z. Bozakov and A. Rizk, "Taming SDN controllers in heterogeneous hardware environments," in Proc. of Second European Workshop on Software Defined Networks (EWSDN), Berlin, Germany, 2013, pp. 50 – 55.
16. M. Kuzniar, P. Peresini, and D. Kostic, "What you need to know about SDN control and data planes," EPFL, Lausanne, Switzerland, Tech. Rep. EPFL-REPORT-199497, 2014.
17. M. Kuzniar, P. Peresini, and D. Kostic, "What you need to know about sdn flow tables," in Passive and Active Measurement, ser. Lecture Notes in Computer Science, J. Mirkovic and Y. Liu, Eds. Springer International Publishing, 2015, vol. 8995, pp. 347–359.
18. V. Mann, A. Vishnoi, A. Iyer et al., "VMPatrol: dynamic and automated QoS for virtual machine migrations," in Proc. of the 8th International Conference on Network and Service Management (CNSM), Las Vegas, USA, 2012, pp. 174–178.
19. A. Nguyen-Ngoc, S. Lange, S. Gebert et al., "Investigating isolation between virtual networks in case of congestion for a Pronto 3290 switch," in Proc. of the Workshop on Software-Defined Networking and Network Function Virtualization for Flexible Network Management (SDNFlex 2015), Cottbus, Germany, 2015.
20. P. M. Mohan, D. M. Divakaran, and M. Gurusamy, "Performance study of TCP flows with QoS-supported OpenFlow in data center networks," in Proc. of the 19th IEEE International Conference on Networks (ICON), Singapore, Singapore, 2013, pp. 1–6.
21. China Mobile Research Institute, 'C-RAN: the road towards Green RAN', version 2.5, October 2011.
22. Z. K. Khattak, M. Awais and A. Iqbal, "Performance evaluation of OpenDaylight SDN controller," 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS), Hsinchu, Taiwan, 2014, pp. 671-676.
23. Z. Cai, F. Dinu, J. Zheng, A. L. Cox, and T. S. E. Ng. Maestro: A Clean-Slate System for Orchestrating Network Control Components. Under submission, 2008.
24. M. Casado, T. Garfinkel, M. Freedman, A. Akella, D. Boneh, N. McKeown, and S. Shenker. SANE: A Protection Architecture for Enterprise Networks. In Usenix Security Symposium, 2006.
25. M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking control of the enterprise. In SIGCOMM '07, 2007.
26. ShalimovA, ZuikovD, ZimarinaD, PashkovV, SmelianskyR. (2013) Advanced study of SDN/OpenFlow controllers.Proceedings of the 9th Central & Eastern European Software Engineering Conference, ACM, 1-6.
27. M. Betts, S. Fratini, N. Davis, R. Dolin and others, "SDN Architecture". Open Networking Foundation ONF SDN ARCH, Issue 1, June, 2014.
28. M. Joselli et al., "An Architeture with Automatic Load Balancing for Real-Time Simulation and Visualization Systems," Journal of Computational Interdisciplinary Sciences, vol. 1, no. 3, pp. 207–224, 2010.
29. Bert Hubert, Thomas Graf, Gregory Maxwell, Remco Van Mook, Martijn Van Oosterhout, Paul B. Schroeder, Jasper Spaans, and Pedro Larroy. Linux Advanced Routing & Traffic Control HOWTO. Linux Advanced Routing & Traffic Control, http://lartc.org/, April 2004.
30. www.admin-magazine.com/Articles/Floodlight-Welcome-to-the-World-of-Software-Defined-Networking. Accessed on 21.05.2018