

# Performance Study of a Software Defined Network Emulator

Jose M. Jimenez, Oscar Romero, Albert Rego, Avinash Dilendra, Jaime Lloret

Universidad Politécnic de Valencia  
Camino Vera s/n 46022, Valencia (Spain)

email: jojiher@dcom.upv.es, oromero@dcom.upv.es, alrema91@gmail.com, avinash.dilendra@gmail.com, jlloret@dcom.upv.es

**Abstract**—Generally, network researchers use applications that allow them to emulate or simulate networks. It is desired to obtain very close results between the ones given in a virtual and the ones obtained when the real network hardware is implemented. In this paper, we compare the experimental results obtained when a virtual network is generated by using Mininet versus a real implemented network. We have compared them varying the Maximum Transmission Unit (MTU) on Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) packets. Ethernet, Fiber Distributed Data Interface (FDDI), and Wireless Local Area Network 802.11 (WLAN 802.11) MTUs have been used in our experimental tests.

**Keywords**- SDN; OpenFlow; Mininet; MTU; virtualization; bandwidth; jitter.

## I. INTRODUCTION

In the field of computer networks, the researches usually use programs that allow us to emulate or simulate networks. This is because, in most cases, we do not have the necessary devices needed to create complex networks. There are emulators and simulators as Omnet++ [1], OPNET [2], NS-2 [3], NS-3 [4] Netsim [5], GNS3 [6], etc. that are frequently used to create computer networks.

Deployment of network is very quick in virtual environment, even if it is needed a large number of resources, which is always practically almost impossible to implement with real hardware. Problem solving or troubleshooting capability is still easier than real implementations. Note that a network researcher has to keep in mind that the results obtained from a virtual network should be similar from those obtained by the real hardware network. If there is a significant difference between results of virtual network and real network, then the research work should not be taken into consideration. As a network test bed gives almost the same results than the real implemented network, then it saves a large amount of time, complexity and a lot of resources.

In general, network devices perform the transport and the control function. But, configuring a great amount of devices and changing the configuration efficiently to work properly, it means a big challenge for networking professionals.

Today's, computer network world is able to offer a large amount of functionalities suited to the requirements of users. A new technology, named Software Defined Networking

(SDN) [7] appears to increase the efficiency and reduce the cost of network configuration.

Figure 1 shows the components of SDN in a layered structure. The first layer consists of some frequently used tools of monitoring and depuration. The tool "Oftrace" is used for analyzing and parsing Openflow message from network dump. "Oftrace" provides a library which analyzes and parses the message from TCP dump or Wireshark [8]. Loops or cyclic path can cause critical problems in SDN. "Oflops" is a tool to catch the loop mechanism in the software defined networks. It mentions the data packets in the loop which are not able to leave the network [9]. "Openseer" is a CGI script which helps to plot that data effectively in SDN [10]. In Controllers Layer there are few controllers which are used in SDN. More often, controllers are called the Brain of Network which controls and manages the software defined network. Floodlight, Open Daylight, Beacon, Nox are among the frequently used controllers in SDN [11]. Flow Visor ensures that multiple isolated logical networks can share the same topology and hardware resources of a network. It places as a transparent proxy between OpenFlow switches and OpenFlow controllers. The isolated logical network is named slice of the network and flow visor is named slicing software in SDN [12]. In SDN environment, OpenFlow switches are used to forward the packets. OpenFlow switches are either a software program or a hardware device which is compatible to OpenFlow protocols. Some of the commercial switches are available in market like HP, Nec, Juniper, etc. [13]. Mininet is used to create realistic virtual network within seconds on a single machine that could be able to run real kernel, switch and application code [14].

There are few emulators and simulators which are frequently used to run and control the technology SDN from a single screen, like NS-3, Estinet 9.0 [15], OmNet ++, Mininet, etc.

In this paper, we show the comparison among the obtained results from the virtual networks and from the real implemented networks. With the assessment of these results we are able to find the significant differences, which may be very useful for the researchers who all are performing their research work in Networking Industry.

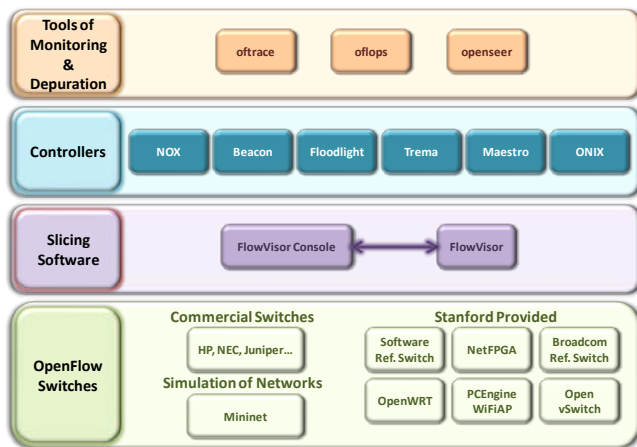


Figure 1. Key component of SDN in layered structure.

We have performed different experiments over Mininet and real implementation to have a good understanding of the network behavior in both scenarios. To do a detailed study, we must send data packets of different properties and compare the results.

We used the data packets with different Maximum Transmission Unit (MTU) on IPv4 and IPv6. These sizes of packets are usual for Ethernet version 2, Ethernet with LLC, PPOE, WLAN, Token Ring and FDDI.

The rest of the paper is structured as follows. In Section 2, we discuss existing related works. In Section 3, we introduced all resources that we used in our test bench. Measurement results and our discussion and analysis are shown in Section 4. Section 5 shows the conclusion and future works.

## II. RELATED WORKS

In the past, a few researchers have accomplished their work in the area of SDN and investigated the performance of multimedia delivery over SDN. Furthermore, in the last years, emulators have been developed in order to provide an easy way to manage virtual networks and perform the research experiments. These emulators reduce the costs associated to the hardware needed to build the network. Inside the SDN research, the emulators have a great importance because of the great number of tests and the specific hardware that are necessary.

In the following section, we are going to discuss about some previous research work that helps us to get a deep understanding of SDN. Then, we will describe the previous researches in which emulators provide a useful way to test the experiments.

Recently, in our previous research article [16], we tried to evaluate the performance of multimedia streaming delivery over Mininet compared to real network implementation. We considered different properties of multimedia delivery, i.e., bandwidth, delay, jitter, and we found some significant differences over mininet and real test network. Kreutz et al. [17] discussed the SDN, and analyzed the significance of SDN over traditional networking. Authors explained about the key components of SDN by using a bottom-up layered

approach and focused on challenges, troubleshooting and debugging in SDN. Noghani et al. [18] introduced a framework based on SDN that could enable the network controller to deploy IP multicast between source and subscribers. The network controller was also able to control the distributed set of sources where multiple description coded (MDC) video content is available by using a simple northbound interface. Due to this SDN-based streaming multicast framework for medium and heavy workload, the Peak Signal-to-Noise Ratio (PSNR) of the received video is increasing considerably. Authors noticed that the received video, which had a very poor quality before, was having a significant increase in the quality of video now. Nam et al. [19] proposed a mechanism to solve the congestion problem and improve the video quality of experience (QoE). Authors tried to develop an SDN based application to improve the quality of video that can monitor conditions of network in real time streaming, and change routing paths dynamically by multi-protocol label switching (MPLS).

Egilmez et al. [20] gives a unique design of an Openflow controller for multimedia delivery over SDN with end to end Quality of Service (QoS) support. The authors tried to optimize routes of multimedia dynamically. After experiments over real test network, the authors found better results than HTTP based multi-bitrate adaptive streaming. They ensured that OpenQoS can guarantee the video delivery with little or no video artifacts experienced by the end-users. In another publication, Egilmez et al. [21] gave new distributed control plane architectures for multimedia delivery over large-scale, multi-operator SDN. The extension included in the design of architecture was: (a) to acquire network topology and the state information by topology aggregation and link summarization, (b) to propose an optimized framework for flowing based end to end over multi-domain networks, and (c) two distributed control plane designs by addressing the messaging between controllers for scalable and secure routing between two domains. By applying these extensions on layered video streaming, authors obtained a better quality of received video, reduced cost and memory overhead. This architecture was effectively scalable for large networks. Kassler et al. [22] tried to negotiate the service and parameter for network communication between end users, and assign multimedia delivery paths in network according to prefixed service configuration. The idea behind this system was to centralized multi-user optimization of path assignments, which provide the better quality of experience by considering network topology, link capacities, delay and account service utility. Due to optimization, the system was able to use Openflow to set up forwarding paths in network.

## III. TEST BENCH

In this section, we are going to introduce the SDN emulator and the real network topology used in our test bench.

### A. Devices and equipment

In this subsection, we explain the devices and equipment used to perform our study.

The real topology is composed by the following equipment:

- 1 Switch Cisco Catalyst WS-C3560-24PS-E [23] that runs an IOS C3560-IPSERVICESK9-M, Versión 12.2 (53) SE2, release software (fc3). It has 24 Fast Ethernet and 2 Gigabit Ethernet interfaces and 16 Mbytes of flash memory;
- 1 Desktop PC that has an Intel Core Quad Q9400 CPU @2.66 Ghz processor, 6 Gb of RAM memory, 1 Network Interface Card (NIC) Intel 82579V Gigabit Ethernet and Windows 7 Professional - 64 bits operative system;
- 1 Desktop PC that has an Intel Core i5-2400 CPU @3.10 Ghz, 4 Gb RAM memory, 1 NIC Intel 82579V Gigabit Ethernet and Windows 7 Enterprise - 64 bits as operating system.

To design and develop the virtualized topology we have used a laptop composed by an Intel i7-4500UCPU @ 2.70 Ghz processor, 16 Gb RAM memory, 1 10/100/1000 Mbit/s NIC, and Ubuntu 14.04 - 64 bits as operating system.

**B. Software used**

With Mininet, we can create a realistic virtual network, running real kernel, switch and application code, on a single machine. The machine can be a virtual machine, or a machine virtualized through the cloud, or a native machine. For our study, we have used Mininet version 2.2.1, with a native installation on Ubuntu 14 as shown in Figure 2.

We used a software application named gt, developed by us, to send traffic with different MTU and bandwidths.

In both, real and virtualized topologies, to capture and analyze the received traffic, we have used Wireshark [24], version 1.10.

**C. Characteristics of traffic transmited**

In our work, we send traffic with different MTUs that represents the packet sizes in different standards. Table I shows different sizes of MTU that was sent in our network topologies.

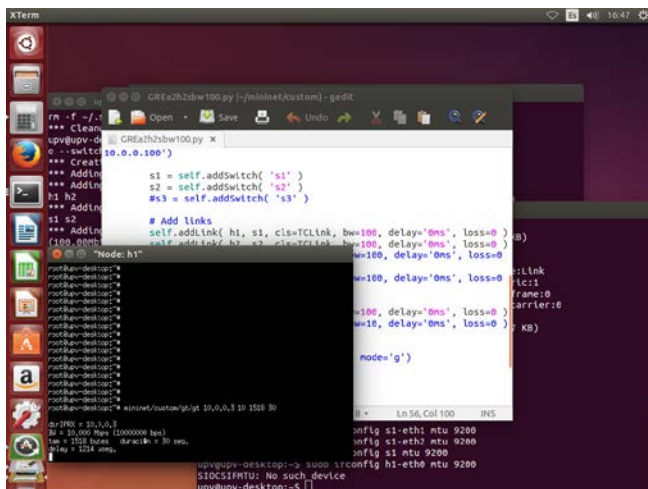


Figure 2. Host running in Mininet.

As can be observed in Table I, sizes of MTU that was sent in our topology do not have standard values. This is because of the need to establish a GRE tunnel in the real topology, to connect the two hosts that have been created in Mininet, thus changing the frame size. Traffic was transmitted through UDP protocol. To calculate the jitter (J), we use the expression presented in RFC 4689 (Terminology for Benchmarking Network-layer Traffic Control Mechanisms) [25]. Therefore, we use the formula (1), where  $S_i$  is the transmission timestamp from packet  $i$ , and  $R_i$  is the reception timestamp of arrival packet  $i$ . For two consecutive packets  $i$  and  $j$ .

$$J = |(R_j - S_j) - (R_i - S_i)| \tag{1}$$

**D. Physical topology**

The real topology consists of two computer connected by straight-through cable, using one switch, as shown in Figure 3. The data transfer rates used is 10 Mbps.

In the virtualized network, we used a computer with Mininet, where we set up the same topology as the real one.

**IV. MEASURAMENT AND DISCUSSION**

This section, shows the results obtained in both cases, when traffic is being delivered over the real network and in the virtual topology using Mininet. Here, we present measures of traffic, with bandwidth links at 10 Mbps, and sending traffic at 10 Mbps. The parameters observed are bandwidth and jitter of packets with three different MTUs: 1518, 4370 and 7999, corresponding to the size of packets for Ethernet, FDDI and WLAN 802.11 traffic.

1) *MTU - 1518: We present the results related of bandwidth and jitter obtained from the real and virtual topologies.*

In Figure 4, we can see the bandwidth consumption values of the real topology and the values obtained in the virtual topology.

TABLE I. MTU PACKETS IN TOPOLOGIES

Frame Differentiation	
Media	MTU (bytes)
Ethernet wit LLC and SNP, PPPoE	1518
FDDI	4370
WLAN 802.11, Ethernet Jumbo Frame	7999

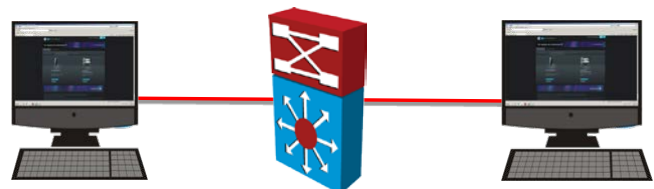


Figure 3. Real topology.

The data have similar values for both topologies when the transmission is stabilized. Although, in real topology is less than in virtual topology. The mean value of bandwidth in real topology is 9.5 Mbps while for virtual topology is 10 Mbps. The maximum and minimum values for real and virtual topologies are different, 9.9 Mbps and 20.8 Mbps for maximum and 6.7 Mbps and 9.9 Mbps for minimum. Observe that in the virtual topology, at the beginning of the transmission we obtain bandwidth values higher than 10 Mbps, meaning that in this situation the emulator is not accurate since the maximum bandwidth for a emulated 10 Mbps physical link should be 10 Mbps. After a few transmitted packets, the measured bandwidth is already providing more accurate values.

In Figure 5, we can see the jitter values of the real topology and the values obtained in the virtual topology. The values of the real topology are higher than those from the virtual topology. The mean value of jitter in real topology is 0.690 ms while for virtual topology is 0.001 ms. The maximum values real and virtual topologies are different, 3.169 ms and 0.607 ms. The minimum values for both topologies are the same, 0 ms.

2) MTU - 4370: We present the results related of bandwidth and jitter obtained for the real and virtual topologies.

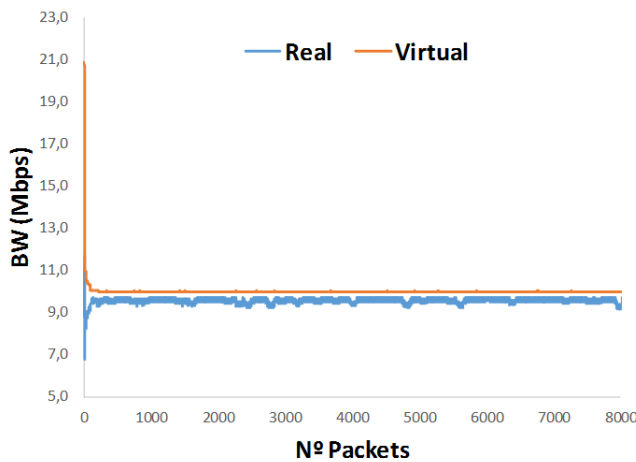


Figure 4. BW at 1518.

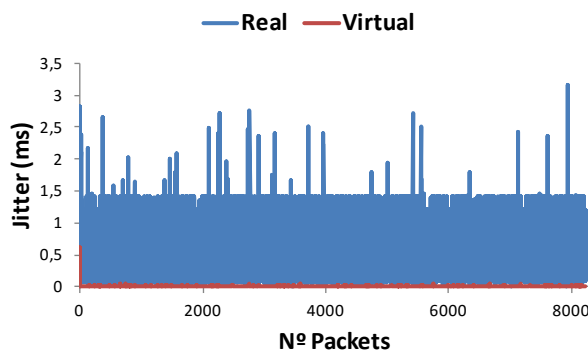


Figure 5. Jitter at 1518.

In Figure 6, we can see the bandwidth consumption values of the real topology and the values obtained in the virtual topology. The data have similar values for both topologies, when the transmission is stabilized, although in real topology is less than in virtual topology. The mean value of bandwidth in real topology is 9.5 Mbps while for virtual topology is 10 Mbps. The maximum and minimum values for real and virtual topologies are different, 9.8 Mbps and 31.5 Mbps for maximum, and 4.2 Mbps and 9.9 Mbps for minimum. As in the previous case, MTU 1518 bytes, in the virtual topology, we can observe that the bandwidth values are not realistic at the beginning of the transmission. After several transmitted packets, the values obtained are already close to the real network values.

In Figure 7, we can see the jitter values of the real topology and the values obtained in the virtual topology. The values of the real topology are higher than those from the virtual topology. The mean value of jitter in real topology is 0.228ms while for virtual topology is 0.002 ms. The maximum values for real topology are different, 9.189 ms and 1.277 ms. The minimum values for real topology and virtual topology are the same, 0 ms.

3) MTU - 7999: We present the results related of bandwidth and jitter obtained for the real and virtual topologies.

In Figure 8, we can see the bandwidth consumption values of the real topology and the values obtained in the virtual topology. The data have similar values for both topologies, when the transmission is stabilized, although in real topology is less than in virtual topology. The mean value of bandwidth in real topology is 9.5 Mbps while for virtual topology is 10 Mbps. The maximum and minimum values for real topology and virtual topology are different, 9.9 Mbps and 23 Mbps for maximum and 3.9 Mbps and 10 Mbps for minimum. Once again, the virtual topology is not providing realistic bandwidth values at the beginning of the transmission and, after transmitting a few packets, the bandwidth values are quite similar to those from the real network.

In Figure 9, we can see the jitter values of the real topology and the values obtained in the virtual topology. The values of the real topology are higher than those from the virtual topology. The mean value of jitter in real topology is 0.345 ms while for virtual topology is 0.001 ms. The maximum and minimum values for real topology and virtual topology are different, 25.091 ms and 0.844 ms for maximum and 0.037 ms and 0 ms for minimum.

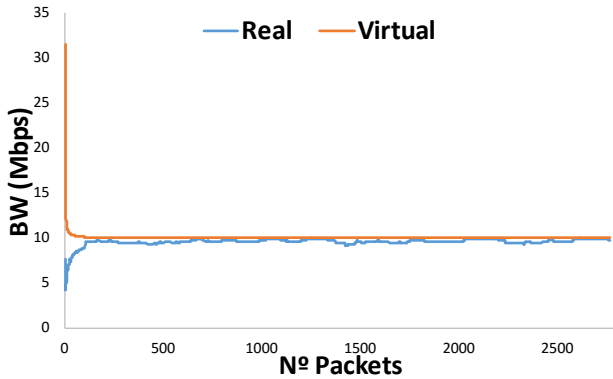


Figure 6. BW at 4370.

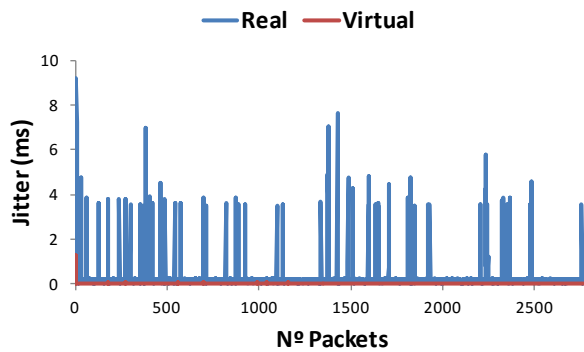


Figure 7. Jitter at 4370.

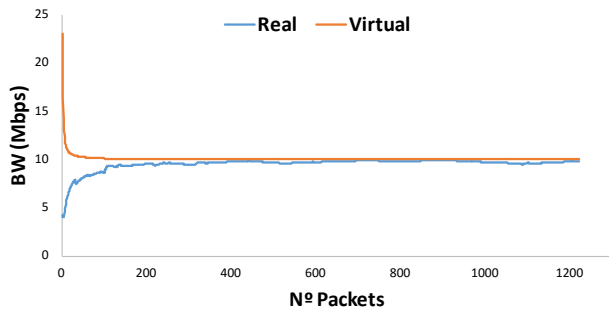


Figure 8. BW at 7999.

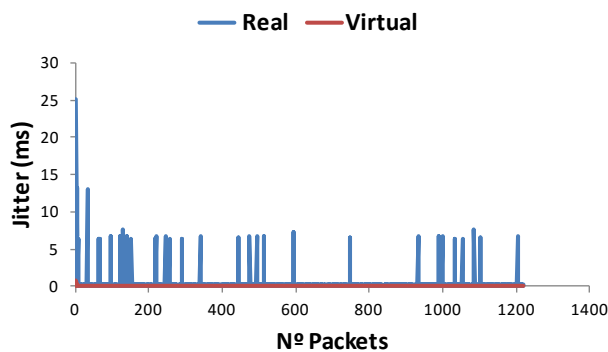


Figure 9. Jitter at 7999

## V. CONCLUSION

In this paper, we have studied the transmission of different MTU sizes which correspond to Ethernet, FDDI, and WLAN 802.11 (also Jumbo Ethernet frames) packets. It can be seen that the variation of the bandwidth between the real and virtual topologies are very low. But, the results obtained for the jitter show that there are major deviations, although, we are working with a very low time scale, as we are dealing with milliseconds.

In our future work, we will compare real and virtual networks using more complex topologies, and by using Openflow compatible equipment.

## ACKNOWLEDGMENT

This work has been supported by the “Ministerio de Economía y Competitividad”, through the “Convocatoria 2014. Proyectos I+D - Programa Estatal de Investigación Científica y Técnica de Excelencia” in the “Subprograma Estatal de Generación de Conocimiento”, Project TIN2014-57991-C3-1-P and the “Programa para la Formación de Personal Investigador – (FPI-2015-S2-884)” by the “Universitat Politècnica de València”.

## REFERENCES

- [1] Omnet++. Available at <https://omnetpp.org/> [Last access January 14, 2016]
- [2] OPNET is now part of Riverbed. Available at <http://es.riverbed.com/products/performance-management-control/opnet.html> / [Last access January 14, 2016]
- [3] The Network Simulator - ns-2. Available at <http://www.isi.edu/nsnam/ns/> [Last access January 14, 2016]
- [4] NS-3. Available at NS-3 website: <https://www.nsnam.org/> [Last access January 14, 2016]
- [5] NetSim NETWORK SIMULATOR. Available at <http://www.boson.com/netsim-cisco-network-simulator> [Last access January 14, 2016].
- [6] GNS3 The software that empowers networks professionals. Available at <https://www.gns3.com/> [Last access January 14, 2016].
- [7] Software-Defined Networking: A Perspective from within a Service Provider Environment. Available at: <https://tools.ietf.org/pdf/rfc7149.pdf> [Last access January 14, 2016]
- [8] Liboftrace. Available at <http://archive.openflow.org/wk/index.php/Liboftrace> [Last access January 14, 2016]
- [9] OFLOPS. Available at <https://www.sdxcentral.com/projects/oflops/> [Last access January 14, 2016]
- [10] OpenSeer. Available at <http://archive.openflow.org/wk/index.php/OpenSeer> [Last access January 14, 2016]
- [11] What are SDN Controllers (or SDN Controllers Platforms)?. Available at <https://www.sdxcentral.com/resources/sdn/sdn-controllers/>[Last access January 14, 2016]
- [12] OpenFlow network virtualization with FlowVisor. Available at [https://www.os3.nl/\\_media/2012-2013/courses/rp2/p28\\_report.pdf](https://www.os3.nl/_media/2012-2013/courses/rp2/p28_report.pdf) [Last access January 14, 2016]
- [13] OpenFlow switch. Available at <http://searchsdn.techtarget.com/definition/OpenFlow-switch> [Last access January 14, 2016]

- [14] Mininet An Instant Virtual Network on your Laptop (or other PC). Available at <http://mininet.org> [Last access January 14, 2016]
- [15] EstiNet Technologies Inc. Available at EstiNet Technologies website: <http://www.estinet.com/index.php> [Last access January 14, 2016]
- [16] J. M. Jimenez, O. Romero, A. Rego, A. Dilendra and J. Lloret, "Study of Multimedia Delivery over Software Defined Networking" in *Network Protocols and Algorithm*, vol. 7, No. 4, 2015, pp. 37-62., 2015, doi:10.5296/npa.v7i48794
- [17] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey", *Proceedings of the IEEE*, Volume103, Issue 1, Jan. 2015, pp. 14-76. 2015, <http://dx.doi.org/10.1109/JPROC.2014.2371999>
- [18] K. A. Noghani and M. O. Sunay, "Streaming Multicast Video over Software-Defined Networks" *Proceedings of the IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) (2014)*, 28-30 Oct. 2014, pages 551-556, 2014, doi: 10.1109/MASS.2014.125
- [19] H. Nam, K. Kim, J. Y. Kim and H. Schulzrinney, "Towards QoE-aware Video Streaming using SDN" *Global Communications Conference (GLOBECOM)*, Dec. 2014, pp 1317-1322, 2014, doi: 10.1109/GLOCOM.2014.7036990
- [20] H. E. Egilmez, S. T. Dane, K. Tolga Bagci and A. Murat Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks" *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, 3-6 Dec. 2012, Hollywood (USA), pp 1-8, 2012
- [21] H. E. Egilmez, and A. M. Tekalp, "Distributed QoS Architectures for Multimedia Streaming Over Software Defined Networks" *Multimedia, IEEE Transactions on*, Volume:16, Issue: 6, Sept 2014, pages: 1597 – 1609, 2014; doi:10.1109/TMM.2014.2325791
- [22] A. Kessler, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, and P. Dely, "Towards QoE-driven Multimedia Service Negotiation and Path Optimization with Software Defined Networking" *Software, Telecommunications and Computer Networks (SoftCOM), IEEE*, Sept 2012, Split (Croatia), pages: 1-5, 2012, ISBN: 978-1-4673-2710-7
- [23] Z. Jingjing, C. Di, W. Weiming, J. Rong, and W. Xiaochun, "The Deployment of Routing Protocols in Distributed Control Plane of SDN" in *The Scientific World Journal*, Volume 2014, Article ID 918536, 8 pages, 2014, <http://dx.doi.org/10.1155/2014/918536>
- [24] Wireshark software. Available at Wireshark website: <https://www.wireshark.org/> [Last access January 14, 2016]
- [25] Terminology for Benchmarking Network-layer Traffic Control Mechanisms. Available at <https://www.ietf.org/rfc/rfc4689.txt.txt> [Last access January 14, 2016]