QOS and scalability evaluation in SDN and Ethernet networks using OMNET4.6 simulator

Fatima LAASSIRI[1], Mohamed MOUGHIT[1, 2, 4], Noureddine IDBOUFKER[3]

[1] *IR2M Laboratory, FST, Univ Hassan 1 UH1- Settat, Morocco*

[2]*EEA&TI Laboratory, FST, Univ Hassan 2, Mohammedia, Morocco*

[3] *National Schools of Applied Sciences, Univ Cadi Ayyad Marrakech, Morocco*

[4] *National Schools of Applied Sciences Khouribga, Univ Hassan 1, UH1- Settat, Morocco*

[1]laassiri.fati@gmail.com

***Abstract*— Traditional network architectures, for example, Ethernet, are not absolutely adapted to users and, generally, don't meet their requirements. SDN (Software Defined Network) has totally changed the network architecture in terms of control and services automation. In this regard, this article presents a comparison between two technologies the first is SDN and the second is Ethernet with their scalability; the make known paper presents simulation results regarding to SDN and Ethernet performances in terms of QoS requirements (Jitter, latency, lost packets, MOS) under Omnet 4.6++.**

***Keywords*— SDN; Ethernet, Controllers; Switch OpenFlow; Scalability; QoS; Omnet 4.6++.**

I. INTRODUCTION

With the rise to SDN, network architecture has seen an extremely positive change in term of managing network policies. SDN is a current architecture that has come to settle the multifaceted nature of those methodologies by separating control and data plane.

It is in certainty a blend of network and programming frameworks, with a specific end goal to isolate the flagging part (Control plane) of the exchange of user information (Data plane), and making the control plane programmable. Subsequently, we have greater adaptability to deal with the system direct with everything taken into account and the flexibility particularly.

The fundamental constraints of customary system builds might be compressed as takes after;

Difficulty to manage the interest for progressively request on transfer speed. The requirement for more transfer speed is for the most part because of a few factors, the primary one are as per the following;

The blast in the quantity of virtual machines facilitated in a solitary Datacenter;

The legitimization and improvement of the usage of equipment assets by mounted development of virtualization advancements. An equipment stage can oblige tens or even many servers.

This expansion in the volume of system movement traded contrasted with past arrangements;

The multiplication of versatile innovations utilized for interfacing with corporate systems: portable PCs, cell phones, tablets ...;

Convergence of administrations: Applications that give administrations consolidating voice and video are democratized and winding up more open by a huge group of clients;

The speculation of topical applications known to require a huge transmission capacity: video conferencing, CCTV...;

Data facilitating outsourcing which enables organizations to accomplish reserve funds in capital cost for the development of datacenters and vitality investment funds.

Complexity: The expansion or expulsion of any system gear makes the need to refresh the arrangements of numerous different gadgets (QoS);

Difficulty to execute steady approaches (get to control and QoS) because of the need to exclusively design hundreds or thousands of gadgets;

Dependence on merchants: The improvement of new administrations by administrators is obliged the solid reliance on gear makers whose hardware advancement cycle is slower than what is required by the necessities of clients.

The following work presents a simulation that implements the performance of SDN [1] (Software Defined Networking) and Ethernet in terms of QoS (Quality of Service) [2] parameters (Jitter, latency, lost packets, MOS) Omnet 4.6++.

With the happening to SDN, network architecture has seen an extremely positive change when managing QoS. SDN is a current architecture that has come to settle the multifaceted nature of those methodologies by separate control and data plane.

It is in certainty a blend of network and programming frameworks, with a specific end goal to isolate the flagging part (Control plane) of the exchange of information (Data plane), and making the control plane programmable. Subsequently, we have greater adaptability to deal with the system conduct all in all and the versatility specifically.

II. STATE OF THE ART

SDN is these days one of the real improvements influencing the substance of the crucial ideas of system models. They are for the systems administration world what virtualization is for the server world. This advancement comes without a moment to spare to diminish the outstanding hole between the capacity to develop of the virtual machines foundations and the system frameworks. Note here that virtualization and distributed computing combined with versatility as all the more squeezing need are the essential factors that have contributed most to feature the constraints of customary system structures.

Quality of Service (QoS) is the capacity to meet applications requirements in term of jitter, latency, packet lost rate, MOS while optimising network resources. For example, for voice over IP,

It is very important in order that network elements must deliver packets, end to end, in 150ms.

QoS leads to the implementation, of many mechanisms such as identification, classification, traffic control and congestion avoidance.

Ethernet increases the lost packets which presents the possibilities of saturation to the memory of the switch as well as the frames lost implies a recovery of losses by the transport layer which collapses the efficiency of the end-to-end quality of service with a transmission delay increase.

Compared to Ethernet, SDN can contribute to the improvement of quality of service in term of jitter, latency, packet lost rate, MOS …

The present paper is simulation results regarding to SDN and Ethernet performances in terms of QoS requirements (Jitter, latency, lost packets, MOS) Omnet 4.6^{++}.

III. ETHERNET

Ethernet is a universal technology that already dominated local networks well before the development of the Internet. The key to the longevity of this technology is its simplicity. Often criticized, it has always been easier to use and implement than its competitors. This article is an introduction to standards (IEEE 802.3 - 10 Mbps, Fast Ethernet - 100 Mbps, Gigabit Ethernet - 1 Gbps, 10 Gbps) and assistance with the design and realization of local networks. [3]

IV. SOFTWARE DEFINED NETWORK CONCEPTS AND ARCHITECTURE

The logical perspective of the SDN architecture comprises of three layers: [4]

The infrastructure layer demonstrated by the network physical part.

The control layer represented by a controller (CR) which is a consistent element deciding on the way how packet are processed on each network element.

The application layer is responsible of defining and providing the access to services.

The communication between SDN layers is done by via Open Flow (OF) [5] protocol.

A. Northbound access

North interfaces (API) are those that allow programmability of SDN. SDN programmability through these interfaces is a double edgead sword. In fact, they can be used to develop security applications that use the network equipment as check points for compliance with the security policy but they may also constitute a privileged entering point for attackers to introduce malicious applications. We can imagine the devastating impact of such exploits on the network. Access control policy to controllers must align with the strongest policies for access control to systems containing sensitive data. [6]

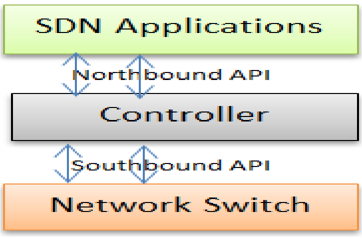

Fig 1. SDN Architecture and interfaces

B. Southbound access

At this level, the authentication needed (in one direction or in both) appears important. Any weakness authentication can be exploited to introduce "rogues" components in the network whether they are controllers or network devices. The OpenFlow 1.4.0 specification states the option to use TLS or UDP/DTLS as transport security protocol.

It also specifies the possibility that the OpenFlow channel rests directly on the TCP protocol without encryption. It simply recommends the use of alternative solutions in this case to protect against eavesdropping. [7]

C. Controller SDN

This is the mind of the SDN show. It gathers data on all networks.

It gives a brought together perspective of the worldwide system and sends charges to all system gadgets. It incorporates the insight of the network.

The controller architecture has evolved from the original single threaded design [8] to the more advanced multithreaded design [9] in recent years.

It contains the instruments, advances, and conventions expected to program the network infrastructure.

The SDN architecture is astoundingly adaptable. It can work with diver's sorts of switches and at various convention layers. SDN controllers and switches can be actualized for Ethernet switches (Layer 2), Internet routers (Layer 3), transport switching (Layer 4), or application layer switching and routing. SDN depends on the regular capacities found on networking devices, which basically include sending parcels in light of some type of flow definition.

Based on the study of available materials on twenty four SDN/Open Flow controllers, we have chosen the following seven open source controllers:

NOX [10] is a multi-threaded C++-based controller written on top of Boost library.

POX [11] is a single-threaded Python-based controller; it is widely used for fast prototyping of network applications in research.

Beacon [12] is a multi-threaded Java-based controller that relies on OSGi and spring frameworks.

Floodlight [13] is a multi-threaded Java-based controller that uses Netty framework.

MUL [14] is a multi-threaded C-based controller written on top of lib event and glib.

Maestro [15] is a multi-threaded Java-based controller.

Ryu [16] is Python-based controller that uses gevent wrapper of lib event.

D. Switch Open Flow

An OpenFlow switch consists of one or more flow tables and a group table. It performs packet look-ups and forwarding. The controller manages the OpenFlow-enabled switch using the OpenFlow protocol over a secure channel. Each flow table in the switch is made up of a set of flow entries in which each flow entry consists of match header fields, counters, and a set of instructions to apply to matching packets [17].

V. IMPLEMENTATION AND SCALABILITY IN OMNET4.6++ (FIG.2, FIG.3, FIG.4, FIG.5)

The present section extant performance evaluation of a SDN and Ethernet while processing SIP (Session Initialization Protocol) based VoIP packets is used.

Performance evaluation is done based on two main scenarios.

TABLE I

SDN AND ETHERNET SCENARIOS

	Scenario 1		**Scenario 2**	
	Case 1: SDN	Case 2: Ethernet	Case 1: Scalability in SDN	Case 2: Scalability in Ethernet
Number of hosts	6 Hosts	6 Hosts	14 Hosts	14 Hosts
Debit	8 kbit/s	8 kbit/s	8 kbit/s	8 kbit/s

SIP is used, which is an application layer tradition. It is convey for end-to-end hailing control to develop a correspondence session between the two systems for the exchanging of data (or streams) over the Internet. This standard is given an exchange technique between UAC (10.0.0.2) and UAS (10.0.0.9) as takes after.

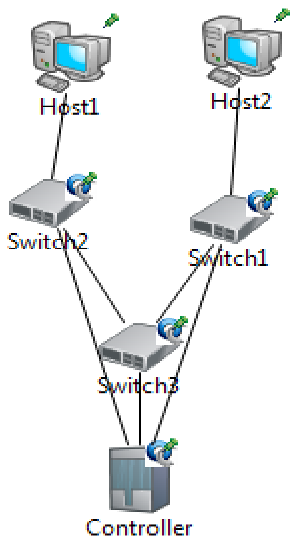

Fig.2 Topology SDN in Omnet 4.6++

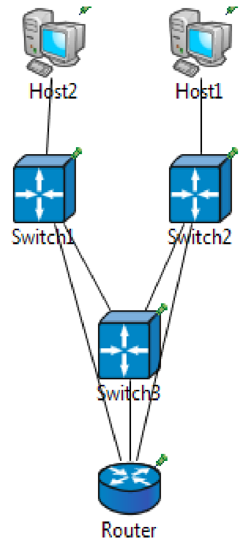

Fig.3 Topology Ethernet in Omnet4.6++

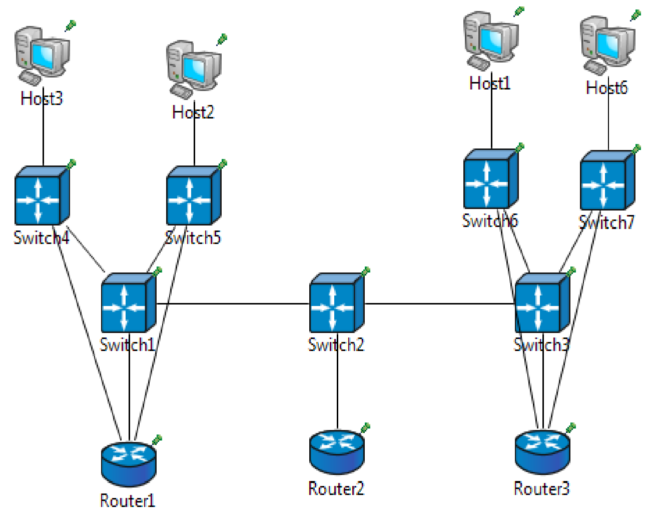

Fig.4 Topology Ethernet with scalability in Omnet4.6 ++

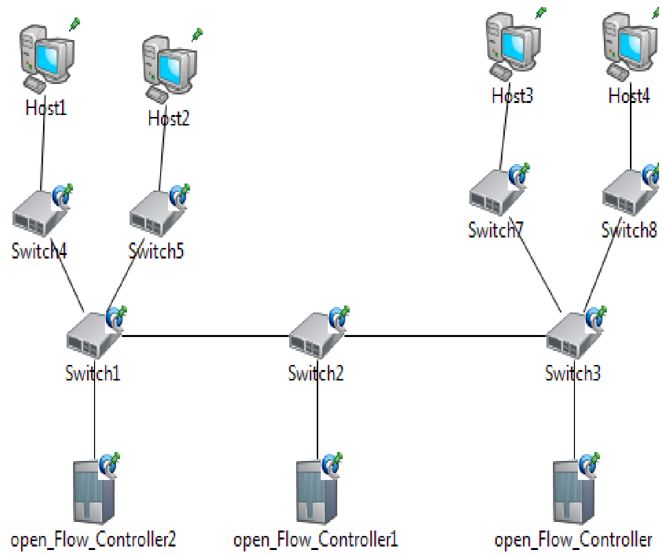

Fig.5 Topology SDN with scalability in Omnet4.6 ++

For this situation, the machine "10.0.0.9" needs to speak with other one "10.0.0.2". It begins with a demand for correspondence by means of the sending of an INVITE request. We take note of the code of the preparing begin with "180 RINGING", and OK which demonstrate to us that the machine "10.0.0.9" acknowledges the foundation of a correspondence session. At last, the machine "10.0.0.9" comes back to its beneficiary an ACK message to affirm this foundation of the association.

VI. DISCUSSION THE RESULTS

A. Settling Time

On the basis of the results obtained, it is observed that the establishment of the calls under SDN is 0,009888 µs by opposition Ethernet 0,860855 µs, as shown in the situation Fig.6. And more than that we have found that SDN does not influence scalability.

QoS over SDN in very depends of network size.

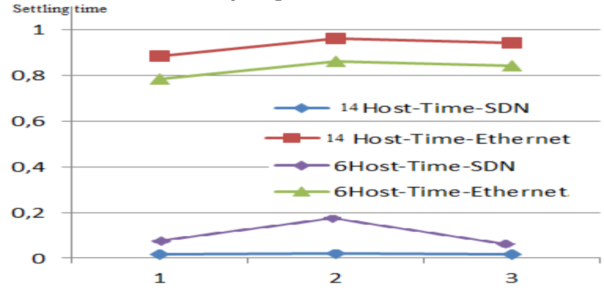

Fig.6 Settling time in SDN and Ethernet networks

VII. QOS PARAMETERS

In this part, we will see every one of QoS parameters (MOS, jitter, latency, lost packets) are done to demonstrate that the SDN is a performant architecture compared to Ethernet as far as in terms quality of service offered to voice application.

We will test the particular parameters of QoS.

A. Latency

The end- delay occurs due to the serialisation, encoding, decoding, propagation delay and the jitter buffering delay. [18]

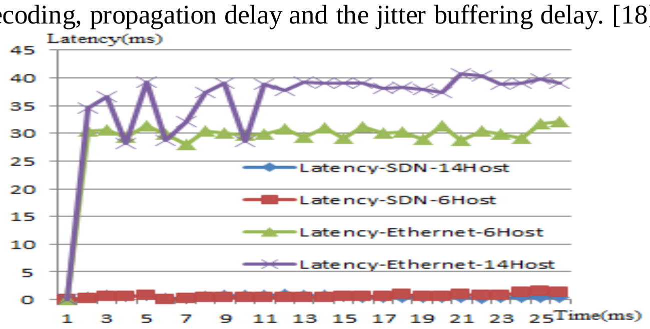

Fig.7 Latency in SDN and Ethernet networks

In a VoIP session of 7 minutes, we observe that the latency of 6 and 14 Hosts with SDN is remain under to 0.1 ms, on the other hand for Ethernet is between 30 ms 41 ms. This shows that SDN is greater than Ethernet several times.

B. Jitter

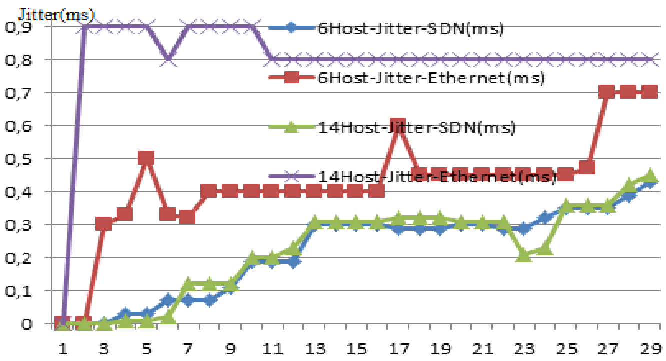

Fig.8 Jitter in SDN and Ethernet networks

Jitter is the variation of the end to end delay between two elements. It is less than 50 ms is adequate for VoIP sessions. In case the transmission time varies exorbitantly in a VoIP call, the nature of the calls is uncommonly degraded. [19]

Fig. 8 shows that SDN architecture with 6 and 14 machines is limited to 0.4 ms whereas Ethernet with their scalability reach to 0.9 ms. Therefore on the latter technology, it is noted that it influences with the number of nodes

C. Lost Packets

Packet lost is the data identified with the nature of your physical line. This includes overhauling the information, and along these lines lessening the throughput. A "normal" bundle lost is under 10%.

From the results we find that the number of packets lost under Ethernet with their scalability is between 15 and 30, which are more remarkable than the SDN which is almost null.

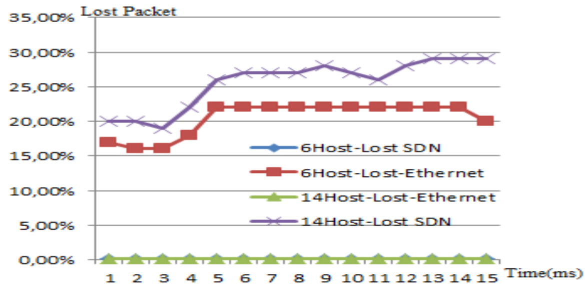

Fig .9 Lost packets in SDN and Ethernet networks

D. MOS: Mean Opinion Score

MOS Mean Opinion Score is a measure of the user quality perception; it is a quality measure that has been used in telephony for decades as a way to assess the human user's opinion of call quality [20]. To calculate the MOS we have used the following formula:

$$MOS = 4 - \ln(loss) - 0{,}7\ln(size)$$

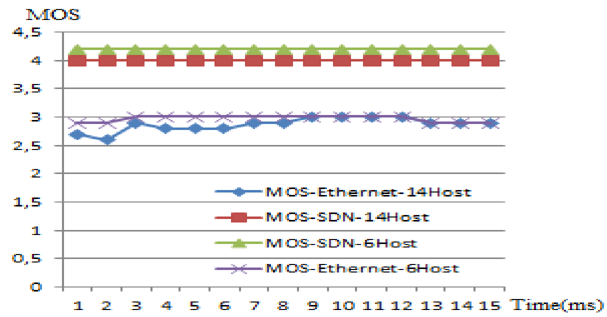

Fig. 8 MOS in SDN and Ethernet networks

TABLE II
RELATIONSHIP OF MOS VALUE TO THE QUALITY OF VOICE RATING

MOS	Quality of Voice Rating
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

From the results obtained, it is found that MOS under SDN with 6 nodes is 4.1 and 14 hosts is 4, unlike the case of Ethernet with 6 machines varies between 2 and 3, with the addition of nodes it decreases. We can therefore conclude that

SDN is more advantageous than Ethernet in terms of MOS.

The quality of transmission of the score under SDN is good because of controller that allows a better transfer of the voice compared to Ethernet that it is between poor and fair.

VIII. CONCLUSION

The concept of SDN is in a decisive turning point which may be crucial for the future of this new technology.

Through various pilot implementations, it was clearly demonstrated that it accelerates innovation and opens networks, which was largely its vocation.

This article presents performance evaluation of Ethernet and SDN in term quality of service. It presents the case of VoIP transport by using Omnet4.6 ++. The QoS evaluation is based on the measure of metrics such as Jitter, Latency, Packet lost and MOS.

As a result, it was absolutely noticed that the performance evaluation under SDN is more efficient than Ethernet.

This paper consists a step for another article that will be an analysis of different SDN architecture to determine the most suitable topology.

REFERENCES

[1] Kiran Vemuri, "Policy carry-over for mobility in Software-Defened Networking", p 9, December 2014

[2] E. Crawley, Argon Networks, R. Nair, Arrowpoint, B. Rajagopalan, NEC USA, H. Sandick, Bay Networks, “A Framework for QoS-based Routing in the Internet”, Network Working Group, August 1998.

[3] Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standard 802.3, 2003

[4] Ungureanu. Oana-Mihaela," Flexible and Programmable Evolved Packet Core: A New SDN-based Model", pp. 24-25, 7 July 2014

[5] Maxence TURY, "Les risques d’OpenFlow et du SDN", maxence.tury@ssi.gouv.fr, ANSSI, May 2016

[6] [7] Jaouad BENABBOU, Noureddine IDBOUFKER, Khalid ELBAAMRANI, Raja ELASSALI, ‘‘SDN Security in OpenFlow Specification 1.4.0’’, Conference Proceedings of ISIVC'2014.

[8] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. Mckeown, and S. Shenker, “NOX: Towards an Operating System for Networks,” in SIGCOMM CCR, 2008.

[9] Advait Dixit, Fang Hao, Sarit Mukherjee, T.V. Lakshman, Ramana Kompella Purdue University, Bell Labs Alcatel-Lucent, “Towards Elastic Distributed SDN Controller” April 2017

[10] POX. http://www.noxrepo.org/pox/about-pox/ , April 2017

[11] Beacon https://openow.stanford.edu/display/Beacon9, December 2015

[12] Floodlight. http://Floodlight.openowhub.org/? December 2015

[13] Mul. http://sourceforge.net/p/mul/wiki/Home/, January 2017

[14] Zheng Cai, Maestro, ‘’Achieving Scalability and Coordination in Centralized Network Control Plan’’, Ph.D. Thesis, Rice University, 2011

[15] Ryu. http://osrg.github.com/ryu/ , January 2017

[16] http://osrg.github.io/ryu/, January 2017

[17] OpenFlow Switch Consortium, OpenFlow Spec, v1.3.0 https://www.opennetworking.org/images/stories/downloads/sdnresources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf, 2012.

[18] SANS Institute InfoSec Reading Room, “Latency and QoS for Voice over IP”, July 2017.

[19] Tobias Hoßfeld, David Hock , Phuoc Tran-Gia1 , Kurt Tutschku , Markus Fiedler, Testing the IQX Hypothesis for Exponential Interdependency between QoS and QoE of Voice Codecs iLBC and G.711, “Testing the IQX Hypothesis for Exponential Interdependency between QoS and QoE of Voice Codecs iLBC and G.711” March 2008.