

Performance Evaluation of Software Defined Network Using ONOS Controller

*Aidah Alnaeim, #Raied Ibrahim

*#Electronic Engineering, Garden City University, Khartoum-Sudan, *aidahalnaeim@gmail.com,
#raiedin2018@gmil.com

Abstract: The advanced in software defined network (SDN) technology around the globe design and management added many merits and characteristics that facilitated complex operations on traditional network , it also eases the dependence only on the controller and its software to cover the management of the network from the single point, which was performed in this paper by using the ONOS controller to solve all operations carried out by several devices and to aggregate the functions of all servers devices which are combined into one software program with different modules to be activated as needed inside the ONOS controller to operate and running some of its useful Applications and do the proof of performance by measuring the quality of service parameters of SDN compared to traditional networks.

Keywords: SDN, Controller, ONOS, QOS, Point Intent, Mininet.

I. INTRODUCTION

SDN is the Most Trending topic in the field of research right now, it is a technology that enables innovation in how we design, programme and manage networks devices.

The effort is to make computer networks more programmable and manageable as it imposes a centralized architecture using a single controller that makes all the decisions.

so the aim of this paper is to implement and establish a software defined network (SDN) using ONOS controller with putting acceptable efforts in exploring its features and applications to improve the SDN network and increase its Availability and also show that importance of SDN technology in a mend mending the quality of Service process. The technical and business benefits of SDN are well understood. Google, Facebook, Microsoft and others have successfully demonstrated the benefits of deploying SDN in both WAN and data center networks.

Software Defined Networking has become one of the more popular ways for organizations to deploy applications. this technology has been instrumental in allowing organizations to deploy applications at a faster rate, and reduce the overall cost of deployment. SDN gives administrators the ability to manage and provision network services from a centralized location. The benefits of this setup are such that more organizations than ever before are starting to ask „what is SDN“ and making the transition.

II. SDN VS TRADITIONAL NETWORK

Traditional networks rely on physical infrastructure such as switches and routers to make connections and run properly. In contrast, a software-based network allows the user to

control the allocation of resources at a virtual level through the control plane. Rather than interacting with physical infrastructure, the user is interacting with software to provision new devices.

From this perspective, an administrator can ascertain network paths and actively configure network services. An SDN also has more ability to communicate with devices throughout the network than a traditional switch. The core difference between the two can be summed up as virtualization. SDN virtualizes your entire network. Virtualization creates an abstract version of your physical network which allows resources to be provisioned from a centralized location.

In a traditional network, the data plane tells your data where it needs to go. Likewise, under the traditional network model, the control plane is located within a switch or router. The location of the control plane is particularly inconvenient because administrators don't have easy access to dictate traffic flow (especially when compared to an SDN).

Under an SDN the control plane becomes software based and can be accessed through a connected device. This means that an administrator can control the flow of traffic from a centralized user interface with greater scrutiny. This gives users more control over how their network functions. You can also change your network's configuration settings from the comfort of a centralized hub. Managing configurations in this way is particularly beneficial with regards to segmentation of the network as the user can process many configurations promptly.

The reason why SDN has become an alternative is that it allows administrators to provision resources and bandwidth instantaneously. it does so while eliminating the requirement

to invest in more physical infrastructure. in contrast, a traditional network would need new hardware if its network capacity was to increase. The traditional model is to buy more equipment, not to press a button on a screen. [8]

III. ONOS CONTROLLER

ONOS is the open source SDN networking operating system for Service Provider networks architected for high performance, scale and availability. The ONOS ecosystem comprises the ONF, organizations that are funding and contributing to the ONOS initiative, and individual contributors. These organizations include AT&T, China Unicom, Comcast, Google, NTT Communications Corp., SK Telecom Co. Ltd., Verizon, Ciena Corporation, Cisco Systems, Inc., Ericsson, Fujitsu Ltd., Huawei Technologies Co. Ltd., Intel Corporation, NEC Corporation, Nokia, Radisys and Samsung. See the full list of members, including ONOS' collaborators, and learn how you can get involved with ONOS at onos.project.org. ONOS is an independently funded software project supported by The Linux Foundation, the nonprofit advancing professional open source management for mass collaboration to fuel innovation across industries and ecosystems.

ONOS Controller (Open Network Operating System) is a Network Operating System, written in Java and provides a distributed SDN applications platform. The system is designed to operate as a cluster of nodes.

ONOS is the SDN controller for building next generation.

The most important benefit of ONOS is that it provides a useful and usable platform for software programs designed for a particular application or use case as its GUI is very functional as it has so many features and useful applications and it is easy to explore.

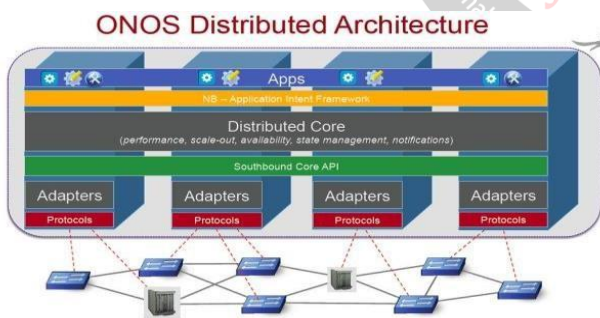


Figure 1, ONOS Distributed Architecture.

IV. SDN ENVIRONMENT

In this Paper, a network simulation was established using Virtual Box to mimic virtual machines started by Mininet Emulator and ONOS controller to improve the SDN Network and Increase its Availability By Running a Useful Application like INTENT Application Which allows ONOS to provide. This test ensures the functionality of intent framework application. The script focuses on the northbound with high level verification.

This test uses a topology shown in the figure below emulated via Mininet. The topology has 7 switches, each end switch consist of multiple hosts that has different configuration. The remaining core switches are used to test rerouting by bringing links up and down. The test combines single and multi node testing using OF1.3 OVS.

The second phase of the project is to testing the quality of services and performance evaluation of ONOS controller compared it with traditional network.

We started by install the latest network drivers available for your NIC hardware, and then install the Hyper-V role on all hosts from oracle. Oracle VM Virtual Box is a free, open source, crossplatform application for creating, managing and running virtual machines (VMs). Virtual machines are computers whose hardware components are emulated by the host computer.

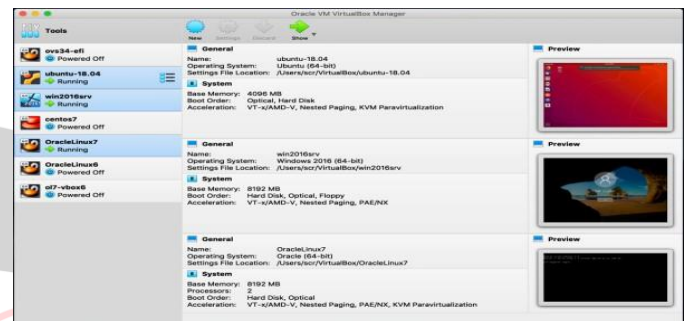


Figure2, oracle virtual Box.

A. Install Mininet

Mininet is an emulator used for OpenFlow & Software Defined Networking experiments. First of all install any version of ubuntu linux. In our case, we are using Ubuntu 14.04 server edition for installing Mininet.

We have to download mininet software from github.com by using "git" utility.

```
$ sudo apt-get install mininet
```

B. Install and Running ONOS Controller Using the Commands Below:

```
$cd onos-2.0.0.0 %to open the onos
$cd bin/ %to activate the onos
$sudo su %to get the permission
```

```
ubuntu@sdnhubvm:~[12:46]$ cd desktop
Desktop
ubuntu@sdnhubvm:~/Desktop[12:46]$ cd onos-2.0.0
ubuntu@sdnhubvm:~/Desktop/onos-2.0.0[12:47]$ cd bin/
ubuntu@sdnhubvm:~/Desktop/onos-2.0.0/bin[12:47]$ sudo su
root@sdnhubvm:/home/ubuntu/Desktop/onos-2.0.0/bin# ./onos-service start
karaf: JAVA_HOME not set; results may vary
```

Figure 3, Running ONOS Controller (CLI).

C. Create a Tree Topology

In the following steps we use commands to establish our network design topology that Contain 8 Hosts and 5 Switches to apply the applications and test the quality of services.

```
$Sudo mn -controller=remote,ip=127.0.0.1,port=6653 -
topo=tree,depth=3,fanout=2
%to Create Tree Topology with 8 Hosts and 7 switches.
```

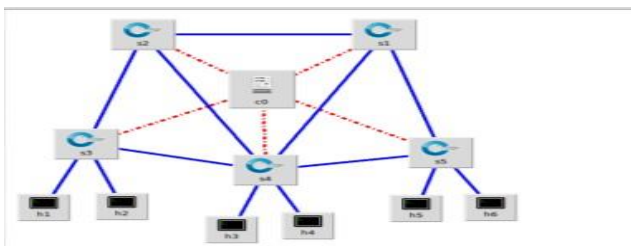


Figure 4, Network Tested Topology.

D. OPEN The ONOS GUI

The ONOS GUI is a *single-page web-application*, providing a visual interface to the ONOS controller (or cluster of controllers).

The following url is ready now to open gui

```
http://localhost:8181/onos/ui/ %to open GUI
```

The ONOS GUI is Very Functional and It Has Many Useful Features.

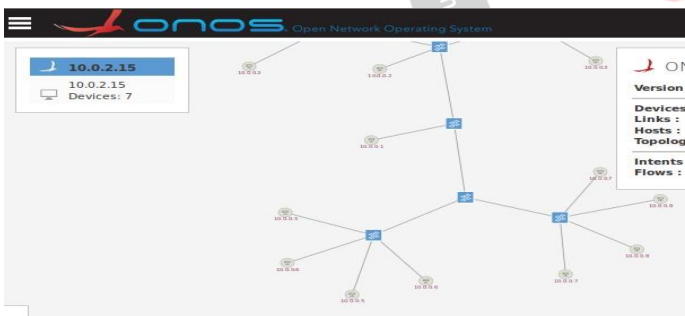


Figure 5, ONOS GUI interface.

E. Apply Point intent on ONOS Controller

The Main Idea of the INTENT application is to enable the network administrator to manage the network with a high-level of abstraction by submitting Intents.

The main purpose from applying intent is abstraction of an application level intent.

For example an intent in setting up a connection between two particular hosts in the network will be handed over to the ONOS core, which is responsible for translating them via Intent Compilation, into Installable Intents, which are Actionable Operations to ONOS.

The compilation process takes into account the network state and behavior requested in the Intent to generate one or more flow rules to implement the requested behavior in the ONOS managed network. These actions are then carried out by the Intent

Installation process, which results in a set of flow rules being installed on one or more selected switches in the network.

First showing all the IP and MAC addresses Must be Done Before applying the intent.

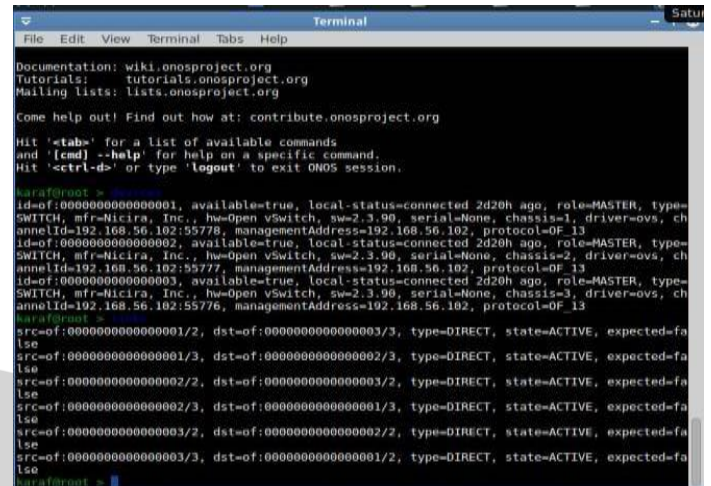


Figure 6, Show all IP and MAC Addresses.

Add point intent	000000000000001/2
	000000000000003/2
Add point intent	source / destination

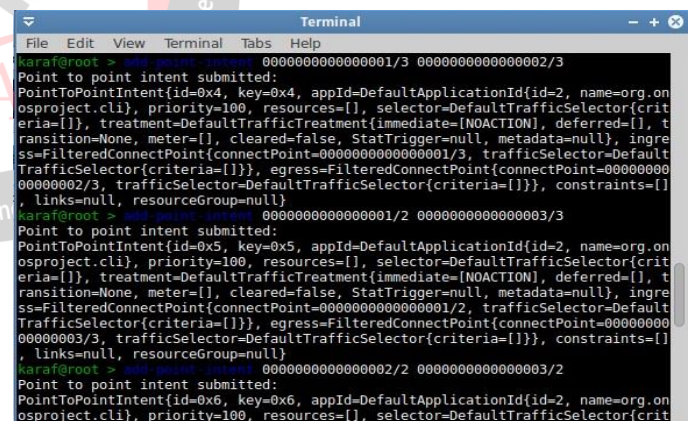


Figure 7, Add Point intent from CLI .

The Figure Below Shows the Point Intent in the GUI

APPLICATION ID	KEY	TYPE	PRIORITY
2: org.onosproject.cli	0x0	PointToPointIntent	100
2: org.onosproject.cli	0x1	PointToPointIntent	100
2: org.onosproject.cli	0x2	PointToPointIntent	100
2: org.onosproject.cli	0x3	HostToHostIntent	100

Figure 8, Add Point intent from gui .

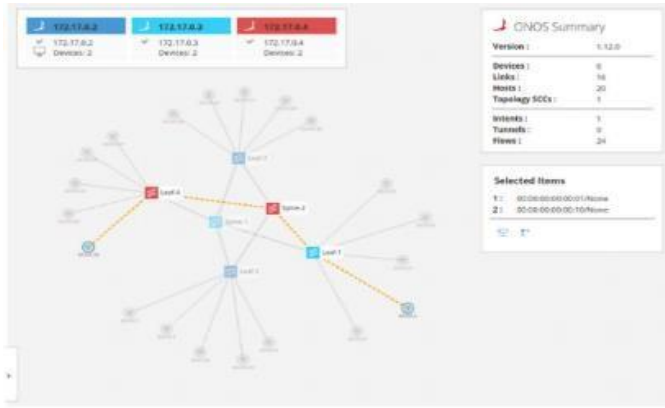


Figure 9, packet path after Add Point intent from gui .

F. Apply Point intent on ONOS Controller That Adding Host-Intent which has the Same Feature as the Point Intent that it Control the traffic and paths But in a Different Technique

```
Add host intent 0000000000000001/2
0000000000000003/2
Add host intent source / destination
```

The figure below shows the host Intent in the GUI



Figure 10, packet path after Add host intent from gui .

V. QUALITY OF SERVICE

The same environment was applied to a conventional network to see the difference in service quality, the main factors influencing service quality and data state have been experimented and measured such as delay , jitter and bandwidth .

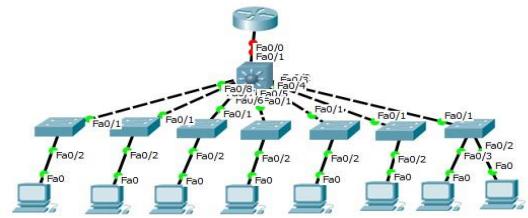


Figure 10, Traditional Network Topology

Table 1, Traditional Network QoS Analysis

QOS parameter	Value
Maximum Interval (ms)	56
Bandwidth (Mbits/s)	1.05
Jitter (ms)	12

the quality of service in ONOS controller in this paper is measured based on the equations shown below:

A. Delay calculator:

Delay is measured by computing the difference between the packets departure and arrival times, subtracting with the estimated switch-to-controller and controller-to-switch times, using the following formula:

$$T_{delay} = T_{arrival} - T_{sent} - T_{controller_to_switch} - T_{switch_to_controller} \quad (1)$$

B. Jitter calculator

The T is represented as following:

TS(x) and TR(y) are proportional to the size of packet. It means they are static value when the packet size is fixed. Based on [14] PT represents the sum of all the propagation time on all links (T total) and forwarding time on all intermediate switches (F total). T total, F total and PT are represented as (2) (3) and (4) respectively. [2]

$$T_{ee} = TS(x) + PT + TR(y) \quad (2)$$

$$T_{total} = \sum_{k=1}^{n+1} T_k \quad (3)$$

$$F_{total} = \sum_{k=1}^n F_k \quad (4)$$

The value of Total can be controlled by routing.

C. Bandwidth calculator:

Defined as the amount of data delivered in a given period of time. It could be obtained using the following given formula

$$max\ throughput = max\ rcv.\ BW / rtt \quad (5)$$

Applying The Quality of Service in ONOS Command line window is Done By Making a Server and Client Hosts then Using The iperf Command To Test The Quality of

Service between them[2]

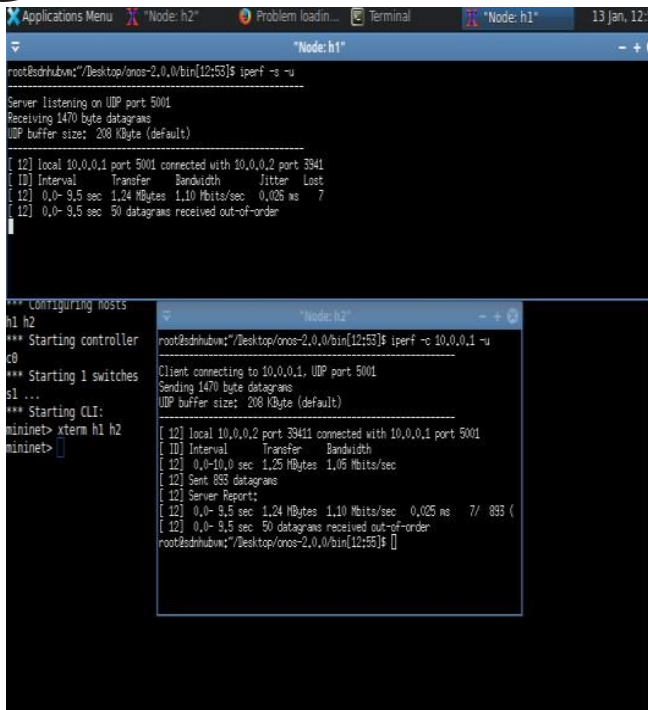


Figure 8, ONOS QoS.

Quality of Service Data analysis Results

Table 2, QoS of ONOS Controller.

Parameter	Value
Maximum Interval (ms)	10
Bandwidth (Mbits/s)	1.05
Jitter (ms)	0.034

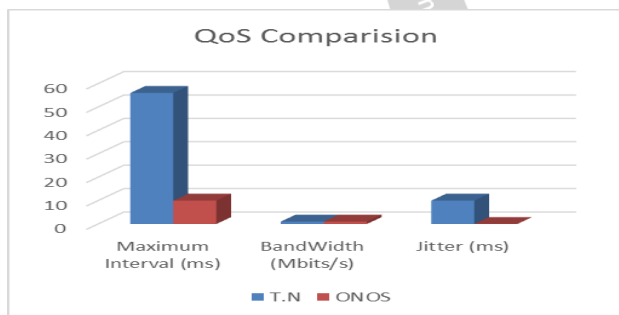


Figure 9, ONOS VS Traditional Network Jitter and Delay .

And When Comparing These Results to The ones

Displayed Earlier By the Traditional Network at the same Bandwidth Value, it Show a huge difference almost 3/100 proportion in the delay Results and a Larger Jitter as Well which explain how much time consuming the traditional network face because of device multiplicity and the noncentralized monitoring system and also Approve ONOS Controller"s High availability and Quality.

VI. CONCLUSION

SDN is considered as a great step towards achieving many features that helps to have sustainable and robust network.

In this paper, previous ideas on the multiplicity and quality of the controller applications were proven upon the traditional network. When using the conventional network, thinking and decision-making is only in the controller and not in the peripheral devices, which reduces the delay in data packets and latency as we shown in the results.

Using ONOS Controller with intent application added more flexibility, so it became possible by click into two hosts to create a dedicated path and control the traffic. This features and applications to ameliorate the network performance and measure the QoS parameters was a True addition as it shows a huge Different in The Performance and the Quality of Service Measures and Offered a very useful Applications which can be a true addition to the SDN Field.

The paper is a comprehensive reference for the operating and running on the ONOS controller

Also proved that SDN is an approach to not only simplify the network but also make it more reactive and reliable to the requirements of business market.

VII. FUTURE WORK

In the coming future we have planned to compare the quality of service between different controllers of SDN. Such as floodlight, RYU and hp.

And also we will Focus on ONOS GUI and all its Features and how to running more than one controller to avoid the problems of single point of attack and failures.

VIII. ACKNOWLEDGMENT

Thanks God for the wisdom and perseverance that has been bestowed upon us during this Work. Last but not least, we would like to thank our Families and Friends for their unconditional support.

IX. REFERENCES

- [1] Horvath, R., Nedbal, D., & Stieninger, M. (2015). "A literature review on challenges and effects of software defined networking. *Procedia Computer Science*" 64, 552-561.
- [2] Azizi, M., Benaini, R., & Mamoun, M. B. (2015). "Delay measurement in OpenFlow-enabled MPLS-TP network. *Modern Applied Science*" 90.
- [3] Chitimalla, D., Kondepudi, K., Valcarenghi, L., Tornatore, M., & Mukherjee, B. (2017). "5G fronthaul-latency and jitter studies of CPRI over Ethernet. *IEEE/OSA Journal of Optical Communications and Networking*" 172-182.
- [4] <https://wiki.onosproject.org/display/ONOS/ONOS> [5] Dumka, A. (Ed.). (2018). "Innovations in Software-Defined Networking and Network Functions Virtualization. IGI Global".
- [6] Blenk, A., Basta, A., Zerwas, J., Reisslein, M., & Kellerer, W. (2016). "Control plane latency with SDN network hypervisors: The cost of virtualization. *IEEE Transactions on Network and Service Management*" 366-380.
- [7] Ahmed, A. M. M., Musa, A. M., & Ahmed, M. A. E. (2017). "Designing Dynamic Consistency for Multi-Controller Software Defined Network Topologies (Doctoral dissertation, Sudan University of Science and Technology)".
- [8] <https://www.comparitech.com/net-admin/software-definednetworking/>