Ealuation of Bandwidth Utilization in SDN

Dr. J P Chaudhari[#1], Dr. D K Kirange[*2], Dr. K S Bhagat*[3], Dr. S D Patil**

[#] *Associate Professor, Charusat Space Rsearch and Technology, Changa,Anand (Gujrat)*

[*] *J T Mahajan College of Engg, Faizpur*

[**] *Government Polytechnic Jalgaon*

Abstract— **One of the important factor for measuring the network performance is bandwidth utilization. In this paper we have evaluated the performance of SDN POX controller in terms of bandwidth utilization. We have used the Mininet as an emulator. The performance of the network is evaluated by using three different scenarios; varying the bandwidth allocated from 10 to 50, increasing the tree topology depth i.e. increasing the number of hosts and switches and by increasing the number of controllers. We have used iperf for measuring the throughput. As a result we can conclude that maximum bandwidth can be utilized for all tree scales if we set the bandwidth to 10. Also we can increase the bandwidth utilization by increasing the number of POX controllers in the network. But increasing number of controllers does not support for large scales. Hence we need to find the optimal number of controllers.**

Keywords— ***SDN, POX controller, bandwidth utilization, mininet, bandwidth utilization***

I. Introduction

SDN is the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices. One of the emerging network architecture is Software-Defined Networking (SDN). SDN is more popular due to its following features

1. it is dynamic
2. easily manageable
3. More cost effective
4. Can satisfy the high bandwidth needs of today's network applications

The decoupling of network control and forwarding functions helps to directly program the network control. It is very easy to abstract the underling infrastructure. One of the most widely used foundational element for building SDN applications is OpenFlow.

Figure 1 depicts the SDN architecture. Different features of SDN architecture over traditional networking are:

1. Due to decoupling from forwarding functions, the network control is directly programmable.
2. The global view of the network is centralized in SDN controllers. It appears to the applications and policy engines as the single logical switch.

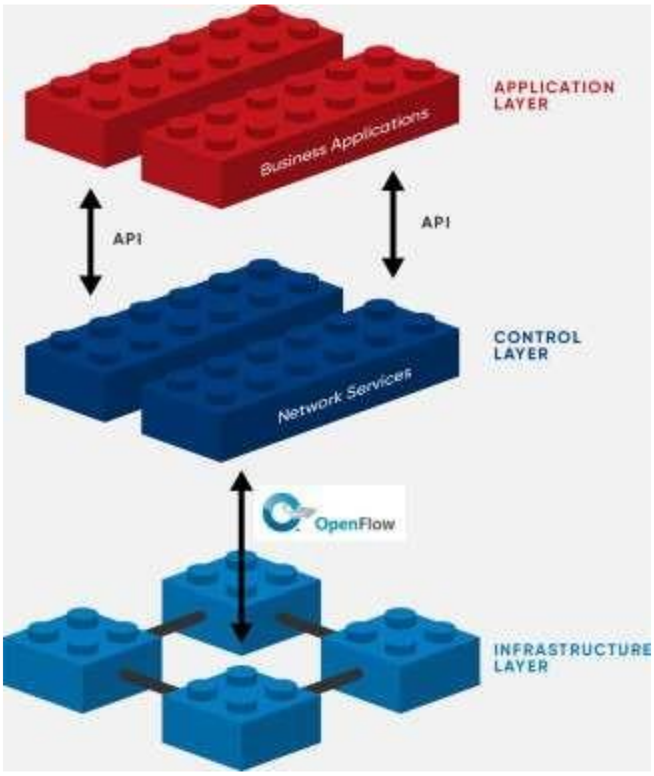

Figure 1: SDN Architecture

3. Dynamic and automated SDN programs allow SDN managers to configure, manage and secure the network resources very quickly.
4. Due to implementations in open standards, the network design and operations are simplified in SDN as instructions are controller specific and not vendor or devices specific.

In this research paper we have evaluated the performance of SDN networks using POX controllers for different bandwidth allocation. The research work is carried out in three scenarios

1. We have allocated the bandwidth varying from 10 to 50
2. Tree topology with different scales is used. By increasing number of switches and hosts we have analyzed the bandwidth loss.
3. By increasing number of controllers in the network we have proved that selecting the optimal number of controllers can improve the network performance in terms of bandwidth utilization.

Rest of the paper is organized as below. Section II deals with the related work in performance evaluation of SDN networks. Section III deals with the simulation setup and preliminaries. Section IV depicts the performance and finally section V concludes the paper.

II. RELATED WORK

In SDN flexible and intelligent network operations are offered due to splitting of control plane and data plane. He flow paths are provided to switches by the intelligent control plane. The control plane also optimizes the network performance. All operations of the data plane management uses the controller in the control plane. Hence, it becomes extremely important to go for the performance of the controller with accurate and effective performance measurement tools. Although a lot of SDN controllers are available in literature, their comparative analysis is not provided. Here authors have presented a qualitative and quantitative analysis of the controllers. 34 SDN controllers have been differentiated based on their performance. Also the detail analysis of capabilities of benchmarking tools used for SDN controllers is provided. In this work three benchmarking tools are used for comparing nine SDN controllers. [1]

The main causes behind the network revolution are heavy traffic, more demand of scalability and bandwidth. SDN when used in data center network paradigm provides a centralized control. T helps to reduce the bandwidth and scalability issues. In this paper authors have developed a user interface for detaining the functions and architecture of DCN. The performance is also analyzed after incorporating SDN controller. The network under consideration consists of 8 hosts connected to 812 parallel hosts. The network performance is evaluated using packet drop rate, bandwidth requirement and latency. Mininet is used as an emulator with python language. [2]

SDN is nowadays widely accepted by stakeholders. Network deployment and management costs is drastically reduced in SDN. Controllers play a major role in SDN. In this study authors have used 4 OpenFlow based controllers for performance evaluation. Different performance parameters considered are latency, bandwidth utilization, packet transmission rate, jitter and packet loss. Different network sizes, topologies and controllers have been used. After performance evaluation, authors have found floodlight controller to be the best. [3]

One of the intelligent component of SDN is its controller. Among the many SDN controllers available in the literature, it is hard to decide which controller to select in which situation. In literature controllers have been compared based on architecture and their efficiency. In this work authors have compared two controllers, Floodlight and opendaylight. The performance parameters used are delay and loss considering different network topologies and scales. The results demonstrated that opendaylight controller performs better as compared to floodlight controller for low loaded networks. For heavily loaded networks, floodlight controller performs best. [4]

Different OpenFlow controllers are available for research and commercial use. But, there is not much more knowledge available about their architectural features to decide which controller to choose. In this paper authors have evaluated the performance of four SDN controllers including NOX, Beacon, Maestro and floodlight. These controllers are multi-threading so they are deployed on on shared memory multicore machines. Different performance metrics used includes, thread scalability, switch scalability and latency. For the performance analysis the guidelines have been provided to design the controller which performs best. [5]

In SDN the forwarding functions are decoupled from the control plane. The controller is used for it. Hence the performance of network depends on the performance of the controller. But lot of SDN controllers are available in literature. Hence it is difficult to select the appropriate controller. This paper studies the performance of different OpenFlow controllers. The benchmarking tool used is CBench. Also the controllers are compared based n features.[6]

Current information and communication technology mostly prefer for SDN due to the flexibility and modularity. One of the important network scenario of SDN is optical transport network. But till today the practical performance of SDN based on DCN has not been evaluated. In this paper authors have evaluated the SDN performance based DCN. Flxi grid optical network testbed with 1000 virtual nodes is used here. The performance is measured using network scalability, bandwidth limitation and restoration time. [7]

The increasing traffic on the internet demands to manage the traffic load. SDN handles the load balancing and decides which servers will handle the traffic or it distribute the traffic among multiple servers. Mininet emulator is used for simulation along with OpenFlow switch. Iperf and Jperf commands are used for performance measure. Different load balancing algorithms used are random, round robin and weighted round robin. The network performance of SDN is increased significantly as compared to other SDN controllers.[8]

III. SIMULATION AND PRELIMINARIES

A. POX Controller

POX is Python based open source controller of SDN .POX is nowadays more commonly used than NOX due to its rapid development and deployment capability.

Different POX features listed by the NOXrepo website [9] are:

1. Python based open source controller
2. Various sample components have been provided for reuse including path selection and topology discovery.
3. Easily deployed anywhere
4. Works on Linux, Mac OS, and Windows.
5. Topology discovery.
6. GUI and Visualization tools of POX are same as NOX
7. POX outperforms NOX applications written in Python.
8. Modern SDN controller.

Figure 2 depicts the architecture of POX controller.

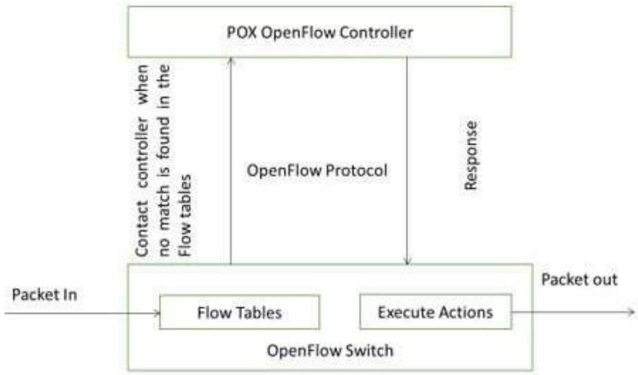

Figure 2: Architecture of POX

As compared to other controller, POX architecture is simple. OpenFlow protocol is used for communication between controller and switches. OF switches behave just like forwarding devices. Controller gives the instruction to switches. After the switch is ON. It contact immediately to the controller. Flow tables are maintained at each switch which are initially empty.

1. Switch will send Packet in message to controller on arrival of the packet.
2. The controller will insert the flow entry in the flow table of the switch and instructs the switch regarding the packet handling.
3. The flow entry is divided into three parts including rule, action and counters.
4. Hence the switch need not to interrupt the controller as packet passes. Switch will only follow the flow entry.
5. If in case the flow entry mismatches with the controller, the packet will be discarded.

POX is beneficial as compared to other controllers in terms of memory space to operate. But if we compare the performance with other controllers in terms of throughput, the POX has low throughput.

B. Mininet

Mininet is an emulator software for creating a realistic virtual network, running real kernel, switch and application code, on a single machine (VM), in seconds. A single command used for creating a virtual network is:

Sudo mn

Mininet CLI is used to

- Easy interaction with the network
- Customize the network
- Share the network
- Easily deploy the network on real hardware
- Tool for development and simulation of SDN networks with openflow

Mininet is most widely used emulator software for development, teaching, and research.

C. Simulation

1. Creating tree topology

Mininet supports the default tree topology. We can create a network using tree with following mininet command,

Sudo mn –topo tree,depth=3 –controller remote

The above command will start a network with a tree topology of depth 3 and fanout 2 (i.e. 8 hosts connected to 6 switches), using Open vSwitch switches under the control of the remote controller.

2. Bandwidth Allocation

We can allocate the bandwidth in tree topology using –link command as below

Sudo mn –topo tree,depth=3 –controller remote –link tc,bw=20
Ere we create a tree topology with depth =3 and bandwidth =20

3. Start POX Controller

Locate the main folder where POX is installed. In pur case it is installed at
/home/ubuntu/pox
To start the controller we move to pox folder by
cd pox
Then
./pox.py log.level --DEBUG forwarding.tutorial_l2_hub

Using this command the POX controller is start by initiating the OpenFlow Protocol Handler and tutorial_hub application.
The above command runs POX controller in DEBUG mode.

4. Use of Multiple POX controllers

We can run multiple POX controllers in the network using different port numbers. We use separate terminal window for each controller. Default port number for the POX controller is 6633. If we wish to add another controller we define the different port as below

./pox.py forwarding.l2_pairs
openflow.of_01 --port=6634

The above command will run the separate POX controller on port 6634.
In the mininet we call multiple controllers with tree topology as below

Sudo mn –topo tree,depth=3
–controller remote, port=6633
–controller remote, port=6634

IV. PERFORMANCE MEASURE

During this experimentation we have used Mininet as an emulator. We have simulated SDN tree topology with POC controller. At each iteration we have increased the depth of the tree (number of switches and hosts in the network). The aim of the experiment is to evaluate the performance of the SDN network for different bandwidth allocation considering different network scales and different number of controllers in the network.

For measuring the throughput of the we have used iperf. Maximum achievable bandwidth on the network can be measured using iperf.

Different steps for performing iperf are

1. Open separate windows for two hsots using
2. Xterm h1 h2
3. Start the TCP server (-s) at h2 with port 5566 (-p). Also, monitor the results every one second (-i).
4. Start the TCP client (-c) at h1. Also, set the transmission duration (-t) to 15 seconds.
5. Obtain the results at h1 and h2.

Table 1 depicts the bandwidth utilization using single POX controller.

TABLE I. BANDWIDTH UTILIZATION FOR SINGLE POX CONTROLLER

Depth Of the tree	10	20	30	40	50
1	8.29	13.7	22.7	26.9	36.7
2	8.29	13.5	19.9	25.5	36.4
3	8.61	12.9	21.4	26	32.3
4	9.02	13.5	14.9	25.7	30.4
5	8.28	12.5	16	27.6	34.6
6	8.26	12.5	21.1	27.2	36.8
7	8.81	12.6	26.7	28.5	35.5
Average	8.508	13.02	20.38	26.7	34.67

Table II depicts the percentage bandwidth utilization from the above results for single POX controller.

TABLE II. % BANDWIDTH UTILIZATION FOR SINGLE POX CONTROLLER

Depth Of the tree	10	20	30	40	50
1	82.9	68.5	75.66	67.2	73.4
2	82.9	67.5	66.33	63.7	72.8
3	86.1	64.5	71.33	65	64.6
4	90.2	67.5	49.66	64.2	60.8
5	82.8	62.5	53.33	69	69.2
6	82.6	62.5	70.33	68	73.6
7	88.1	63	89	71.2	71
Average	85.08	65.14	67.95	66.9	69.34

As can be concluded from table I and table II, maximum bandwidth utilization occurs for bandwidth allocation =10. The performance does not vary with increasing the scales of the tree topology.

Figure 3 depicts the percentage bandwidth loss for single POX controller. As depicted in figure 3, the bandwidth loss is less for bandwidth allocation of 10. The maximum bandwidth loss occurs at bandwidth allocation of 20 and 30.

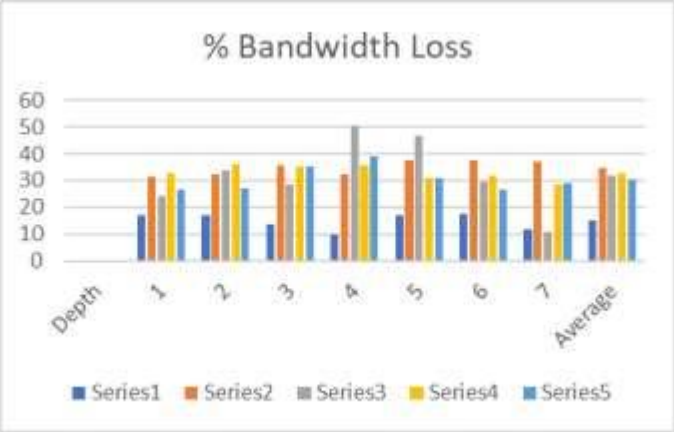

Figure 3: % Bandwidth loss for Single POX controller

Bandwidth utilization suin
g two POX controllers is depicted in table III.

TABLE III. BANDWIDTH UTILIZATION FOR TWO POX CONTROLLERS

Depth Of the tree	10	20	30	40	50
1	8.21	17.9	26.7	34.1	42.7
2	9.91	18.7	26.2	34.9	42
3	8.21	18.7	26.7	34.7	41.8
4	8.93	14.7	20.1	28.3	35.2
Average	8.815	17.5	24.92	33	40.42

If we use two POX controllers in the tree topology we were able to scale the network upto depth of 4 only. Further expansion of the tree topology was not possible.
Table IV depicts the % bandwidth utilization of the network for two POX controllers.

TABLE IV. % BANDWIDTH UTILIZATION FOR TWO POX CONTROLLERS

Depth Of the tree	10	20	30	40	50
1	82.1	89.5	89	85.2	85.4
2	99.1	93.5	87.33	87.2	84
3	82.1	93.5	89	86.7	83.6
4	89.3	73.5	67	70.7	70.4
Average	88.15	87.5	83.08	82.5	80.85

As depicted in table III and table IV the maximum bandwidth utilization of the network using two POX controllers occurs if we allocate the bandwidth=10.

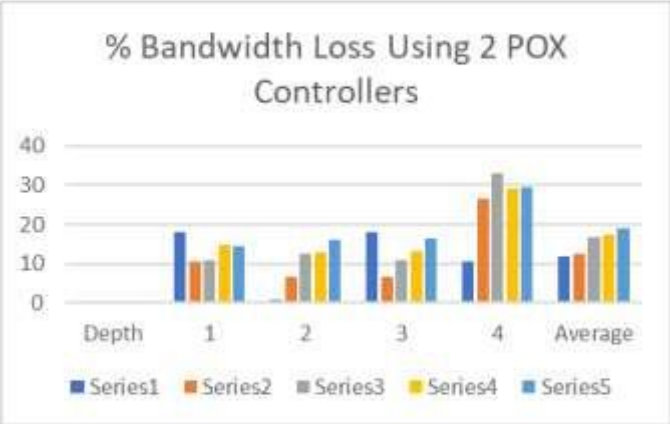

Figure 4: Percentage bandwidth loss using two POX controllers

As depicted in figure 4 the bandwidth loss is less for bandwidth allocation =10. The bandwidth loss increases for bandwidth allocation of 20,30,40 and 50.
Table V shows the bandwidth utilization for three POX controller network.

TABLE V. BANDWIDTH UTILIZATION FOR three POX CONTROLLERS

Depth Of the tree	10	20	30	40	50
1	9.88	18.6	26.7	34.7	42
2	9.83	18.8	26.8	35.3	40.7
3	9.78	19.1	26.9	35.4	42.8
Average	9.83	18.83	26.8	35.1	41.83

Table VI depicts the percentage bandwidth utilization for the SDN network using three POX controllers. As shown in table V. we were able to scale the network upto depth of 3 only for 3 POX controllers.

TABLE VI. %BANDWIDTH UTILIZATION FOR three POX CONTROLLERS

Depth Of the tree	10	20	30	40	50
1	98.8	93	89	86.7	84
2	98.3	94	89.33	88.2	81.4
3	97.8	95.5	89.66	88.5	85.6
Average	98.3	94.16	89.33	87.8	83.66

As depicted in table V and table VI maximum bandwidth utilization occurs for the bandwidth allocation of 10. The bandwidth utilization goes on decreasing for the increasing bandwidth allocation.
Figure 5 depicts the percentage bandwidth loss for the SDN network using three POX controllers. As depicted in figure, bandwidth loss is less for the bandwidth allocation of 10/ The bandwidth loss goes on increasing for the consequent bandwidth allocation.

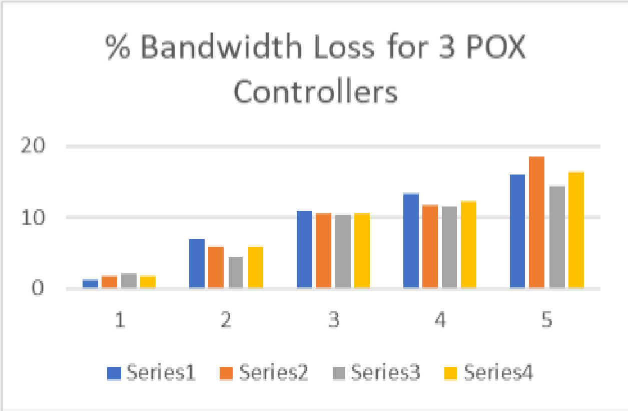

Figure 5: % Bandwidth loss for 3 POX controllers

As we have evaluated the performance of the network for different number of controllers, we found the maximum bandwidth utilization at bandwidth allocation of 10. Table VII depicts the performance of the network using different POX controllers for bandwidth allocation =10.

TABLE VII. %BANDWIDTH UTILIZATION FOR three POX CONTROLLERS

Depth Of the tree	***10***	***20***	***30***
1	10	10	10
2	82.9	82.1	**98.8**
3	82.9	99.1	**98.3**
Average	86.1	82.1	**97.8**

Figure 6 depicts the percentage bandwidth utilization for multiple POX controllers.

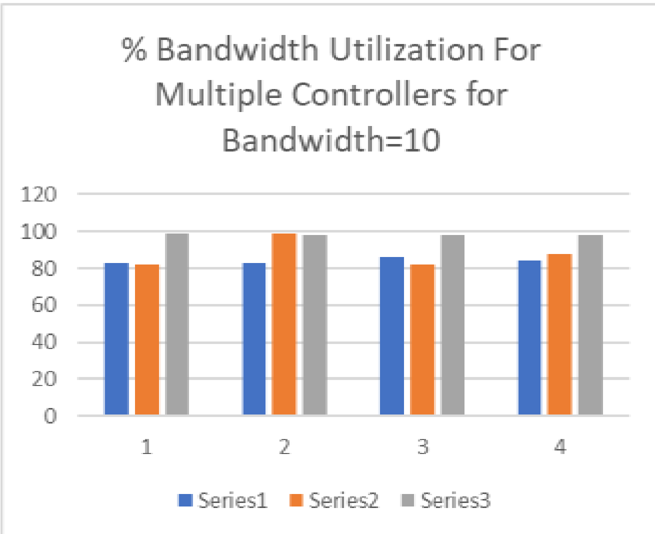

Figure 6: Percentage Bandwidth Utilization for multiple Controllers

As depicted in figure 6 and table VII, the performance of the network increases as we increase the number of controllers. But increasing the number of POX controllers in the network reduces the scalability of the network. Hence optimal selection of number of controllers can improve the network performance.

V. CONCLUSION

SDN plays important role in todays network revolution. A lot of researchers have been attracted towards SDN due to it's flexibility, open source controllers and separation of control plane from forwarding plane. In this paper we have evaluated the performance of SDN POX controller. We have used Mininet as the network emulator. Tree topology with different scales is used. The experimentation is carried out in three scenarios:

1. By increasing the bandwidth allocation from 10,20,30,40 to 50. We have concluded from the results that maximum bandwidth utilization occurs at bandwidth allocation of 10. If we allocate the bandwidth 20, 30 ,40 or 50 the bandwidth loss increases.
2. By increasing the depth of tree i.e. by increasing the number of switches and hosts in the network. We have concluded that increase in number of hosts and switches does not affect the performance of network in terms of bandwidth utilization / bandwidth loss.
3. We have concluded that by increasing number of POX controllers in the network we can increase the throughput of the network. But increase in the number of controllers decreases the network scalability. Hence it is important to find the optimal number of networks.

In future we will try to consider different network performance parameters including Jitter, Packet loss and average delay. Also we will try to find the algorithm for optimal number of SDN controllers in the network sothat maximum throughput can be achieved.

REFERENCES

[1] Liehuang Zhu, Md Monjurul Karim, Kashif Sharif, Fan Li, Xiaojiang Du, and Mohsen Guizani, "SDN Controllers: Benchmarking & Performance Evaluation",
arXiv:1902.04491, Networking and Internet Architecture (cs.NI); Distributed, Parallel, and Cluster Computing

[2] Jayneel Vora ; Shriya Kaneriya ; Sudeep Tanwar ; Sudhanshu Tyagi, "Performance Evaluation of SDN based Virtualization for Data Center Networks", 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)

[3] Sangeeta Mittal, "Performance Evaluation of Openflow SDN Controllers", International Conference on Intelligent Systems Design and Applications ISDA 2017: Intelligent Systems Design and Applications pp 913-923

[4] Shiva Rowshanrad, Vajihe Abdi, Manijeh Keshtgari, "Performance Evaluation Of Sdn Controllers: Floodlight And Opendaylight", IIUM Engineering Journal , VOL. 17 NO. 2 (2016)

[5] Syed Abdullah Shah, Jannet Faiz, Maham Farooq, Aamir Shafi, Syed Akbar Mehdi, "An Architectural Evaluation of SDN Controllers", Rice University, Tech. Rep., December 2011

[6] Lusani Mamushiane, Albert Lysko, Sabelo Dlamini, "A Comparative Evaluation of the Performance of Popular SDN Controllers", CSIR Pretoria, South Africa

[7] Zhao Y, He R, Chen H, Zhang J, Ji Y, Zheng H, Lin Y, Wang X., "Experimental performance evaluation of software defined networking (SDN) based data communication networks for large scale flexi-grid optical networks.", Opt Express. 2014 Apr 21;22(8):9538-47. doi: 10.1364/OE.22.009538.

[8] Dr. Mohammed Najm Abdulla and Zahraa Alaa Baqer, "Performance Evaluation for Load Balancing in SDN Controller", Journal of Advanced Research in Dynamical and Control Systems,Volume 11 , 04-Special Issue

[9] https://noxrepo.github.io/pox-doc/html/