Performance Analysis Of Software-Defined Network And POX Controller

Dhyani Joshi[1*], Mr. Shiv Sutar[2]

***Corresponding Author: -** Dhyani Joshi

[1,2]School of Computer Engineering & Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India

Email: [1]dhyanijoshi1912@gmail.com, Email: [2]shiv.sutar@mitwpu.edu.in

Abstract

In a new era of technology world of software-defined networks is running faster nowadays. The Internet demands the growing popularity of computer networks. This traditional network system does not satisfy the recent computer needs. SDN includes so many other facilities like it is allowing operators of the network and while exploiting the network, users can remotely access and control network devices. In the comparison of this network is much better than any other controller. There are so many software-defined networks like POX, NOX, ONOS, Open Day light, etc. These controllers are for the NFV infrastructure of a data center. These controllers provide some policy and centralized management. Advantages of the software-defined network – programmability, virtualization of tasks, and management of the network in an easy way that provides SDN platform and on the other side POX controller based on a Python and open source OpenFlow SDN controller for faster development. To create a prototype of network applications formly it comes from pre-installed with the Mininet-virtual machine. POX controller can turn dumb OpenFlow devices into the hub, switch, load balancer, and firewall devices. In this paper performance metrics such as service Delay, utilized bandwidth, received packets, and bytes were measured and recorded using network monitoring tools like iperf to analyze the functionality of the POX controller as well as to evaluate the operating performance of the POX controller for SDN environment. The results of this research were the recommendation of using POX controllers for the development and prototyping of network control systems.

Keywords: SDN (Software-defined network), POX Controller, Mininet-wifi

DOI Number: 10.14704/nq.2022.20.9.NQ440059 **Neuro Quantology 2022; 20(9):552-560**

1. Introduction

New technologies had been made it possible to reduce the costs associated with bringing training. It is always increasing theuse of the learning environment and it helps to train to underwrite business goals.

An innovative concept called software-defined networking has been developed to easily such limitations in traditional networks. The conventional networking infrastructure is made up of several networking gadgets together with switches, routers, and intermediate gadgets, and every device is software-specific united circuits are hooked up to carry out dedicated duties. From all these duties, the main goal is that to decouple the control plane from the data plane and whichconsists of forwarding devices.

An outcome of suitable control logic based on application- specific requirements can be implemented on physical devices in real-time. Infrastructure, control, and application are the three layers that made up SDN in a broad intellect. There are some multiple application programs like– northbound, and southbound. The northbound API connect with the application and control layers to permit them to communicate. The network is represented as an application layer through the northbound API. In charge of, the southbound API is between the infrastructure layer and control layer. It is with different types of rules for forwarding devices like routers, and switches. The eastbound and westbound APIs are in control of bridging the gap between various controllers so that they can makechoices in concert.

So, for the use of POX controller is that it is work as a remote controller. It provides better result than this SDN simple framework. Talk about another controller like NOX, POX,Open Daylight controller etc. About all this controller POX controller is best out of all. So, in this paper I used this POX controller to create clear result. It provides

OpenFlow is an ordinary protocol that is based on the software-defined networking architecture. So basically, this network works on an Openflow Protocol. So, an SDN controller and the network agent communicate through this protocol.

In the solution of IoT devices that SDN is controlling the traffic networks using these devices. And secondly, it reduces the cost of that hardware devices. So, for that, we are doing a simulation using the Mininet-wifi simulator. In this paper, we are going to analyze how we can connect software defined- network and the Internet of things using the Openflow protocol.

2. Literature Survey

There are different types and ways of research papers and journals about software defined network and the Internet of things also. Some so many authors who write about these emerging technologies. So, here discussion of some papers andwhat they include:

[1]In this paper, they are giving a deliberate outline of SDN-based innovations in IoT in various angles, for example, edge, access, center, and server farm organizing. They are mainly used the sensor network to manage the resource of the sensor node efficiently.

[2]This paper presents a software model based on SDN. In comparison to traditional networks, SDN improves network performance while lowering hardware utilization and providing better security and privacy.

[3]The IoTSim-Osmosis toolset for osmotic computing is described in this study. Multiple Osmotic systems can be modeled and simulated in one setting with IoTSim-Osmosis. It enables IoT, edge, and cloud ecosystems to be integrated, as well as SD-WAN networking techniques. IoT devices can transfer data using various wireless technologies while the edge can incorporate virtualized devices and SDN-aware infrastructure, thanks to this toolset.

[4]In this paper, in comparison to other existing systems and approaches, they presented a novel system concept and execution utilizing SDN to get IoT gadgets that utilizationHTTP to alleviate and stay away from security assaults withoutthe need to alter the IoT gadgets in this work.

[5]This study contributes to the development of a 6LoWPAN emulation topology, which is commonly used in IoT deployments. 6LoWPAN has been demonstrated to have low power consumption and bandwidth. Thus, we utilized the test system program Mininet-IoT to lead reproduction examination and execution investigation to layout IoT correspondence organizations and survey QoS factorsincluding jitter, bundle misfortune, throughput, and idleness.

[6]Mininet-WiFi is a tool that was introduced for simulating wireless OpenFlow and SDN setups, permitting high-fidelity experiments that mimic real-world settings of the network. This tool adds virtual remote stations or passageways to the familiar Mininet emulator while holding the first software-defined capacities and lightweight virtualization programming engineering. We go over the benefits and downsides of Mininet-likely Wifi's purposes.

[7]From this paper, they create a user application to rule Data-center network and analyze Software defined network performance. They look at networks of 8 to 512 hosts and compare performance parameters like latency, packet drop rate, and bandwidth. M VMware workstation, Mininet and v Switch, and OpenFlow controller are the tools used. The Python programming language is utilized to connect the tools.

[8]In this paper, the main key contains between software-defined approaches used in a Data-center Networks and those used in IoT

environments and in IoT settings then presents a vision for a layered SDN controller in IoT settings. They show how to use the layered view to match appropriate low-level resources to tasks with high-level meanings.

3. Proposed Work

In comparison to a traditional network, software- defined networking is gaining popularity among technologists. SDN is a networking technology that allows network nodes aremanaged by programming instead of traditional system management methods.

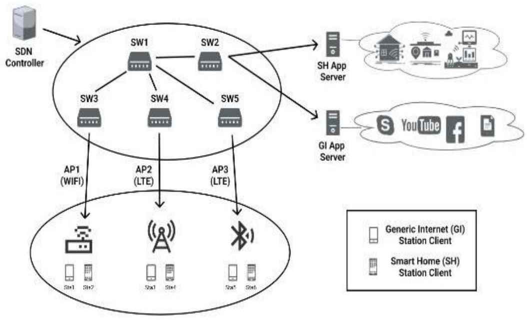

Fig. 1 Proposed Network

SDN innovation can help an organization get ready for a fruitful and dependable IoT arrangement. It gives the adaptability and nimbleness that the Internet of Thingsrequests. Moreover, it gives an open climate for application designers to make creative apparatuses and programming to allthe more likely to coordinate the IoT.

As previously indicated, the infrastructure for this article's implementation is comprised of decoupled control and data planes. A software application serves as the control plane in most cases. High-level languages like Python and C are commonly used to write these software programs. Unlike typical networks, where the data plane is not programmable,the data plane in this network is programmable hardware.

The control planes are made up of logic that governs how routers and switches forward packets. It also includes procedures for configuring intermediary boxes such as firewalls and load balancers.

Because the data plane is in charge of sending packets over the network, it includes routing tables and network hardware. To summarise the functionality of SDN, the control plane functions are responsible for determining the shortest pathways, the data plane functions are in charge of the handling of packets and routing for them from input-output ports. Software and hardware can evolve independently, and control and data planes are separated. Furthermore, it allows high-level programs to program the controller. Debugging the network is simple with this form of control enforcement.

4. Software defined-network

Design a network technique that is a software-defined network it allows the network to be controlled and programmed through the software applications. This network technology manages to enable operators.

Based on software controller and APIs it is used to communicate with the hardware infrastructure and network with the direct traffic. So, this architecture is different from the traditional network and this administrates with the hardware devices. Software-defined Networks can use software to establish and control a virtual network — or traditional hardware.

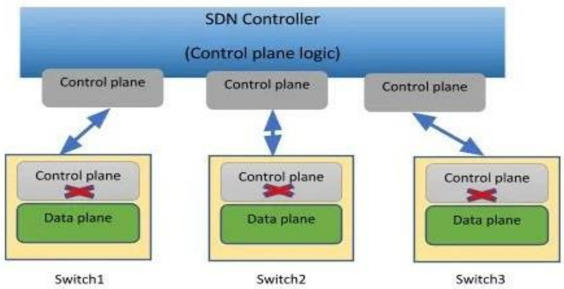

Fig. 2 Working of SDN

Here are the fundamentals of SDN: The product is isolated from the equipment in SDN. The control plane indicates the traffic, which is moved to programming. However, the information plane, which onwards traffic, stays in equipment. This allows a network that utilizes programming characterized systems administration to program and regulate the whole organization from a private mark of control as opposed to the gadget by gadget.

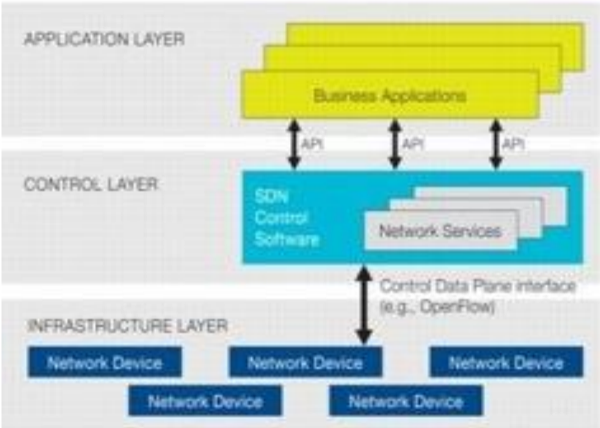

Fig 3. Architecture of SDN

The application layer is where organizations' conventionalnetwork applications or operations are stored. Some examples are like Load balancing, Intrusion detection systems, and firewalls. Software-defines network's brain is a serves of centralized SDN controller software that is comprised control layer.

In the network, the physical switch is from the infrastructure layer. So, It is routed through these switches to its final destination.

5. POX Controller

POX is a Python based open source OpenFlow/Software Defined Networking (SDN) Controller. This controller mainly give or used for faster development and prototyping of new network. POX controller already (pre-installed) when we installed Mininet Virtual Machine. So, using this controller we can turn dumb openflow devices into switch, hub, firewalldevices.

POX controller is based on a Python program so when it started it can invoke to implement networking functions and commands. It is simple to use this controller. At its core, it's a platform for the rapid development and prototyping of networkcontrol software using python. At a very basic level, it's one of a faster and growing number of frameworks like NOX, Floodlight etc for helping an openflow controller.

6. OpenFlow

The OpenFlow protocol is used to communicate through the Software-defined controller and a network device. OpenFlow protocol was formed by Stanford University academics in 2008, and Google implemented it in their backbone network in 2011-2012. The most recent version is V1.5.

OpenFlow is also called a programmable network protocol used to communicate between OpenFlow switches and controllers in an SDN context. Network device programming and hardware devices are separated from the OpenFlow and resulting in this approach to building a centralized, programmable network that can quickly adapt to changing network demands.

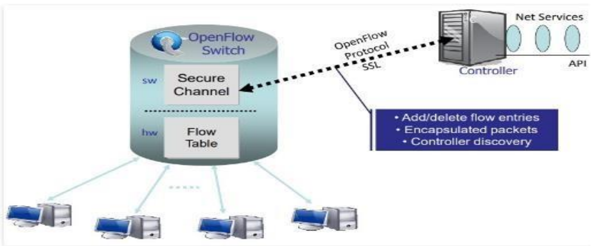

Fig. 4 Architecture of OpenFlow Switch

It is an OpenFlow-enabled data switch and it communicates over the OpenFlow channel among an external controller. It uses more than one flow table and a group table tolook up and forward packets. The controller connects with the OpenFlow switch, which the controller manages through the OpenFlow switch protocol.

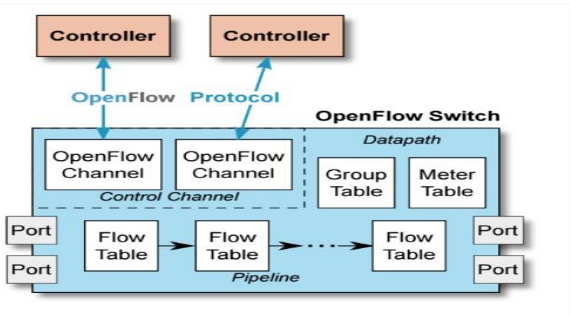

Fig. 5 Working of OpenFlow Switch

Only three basic pieces work together to make an OpenFlow switch work: firstly, the flow table indicates switches, the controller and this protocol that is allows the controller to communicate with switches. On switches, flow tables are created. The OpenFlow protocol is used by controllers to communicate with switches and to apply policies to flows. The controller could create network paths that are optimized for specific qualities like speed, hop count, or latency.

Only a controller and a switch communicate using the OpenFlow protocol. There are no OFF messages between two switches in a network that are linked to the controller via different ports.

Why is OpenSwitch the new craze?

By decoupling the control function from the hardware, it is expected to govern, the OpenFlow switch is designed to assuretraffic management and engineering uniformity. In a world where network variety and demands are expanding. The majority of clients are working on enabling OpenFlow on physical switches and transitioning to an OpenFlow switch.

7. Mininet-wifi

A software that creates prototype in a large network only on one single machine it's called Mininet. It is network emulator so it virtually creates hots, links, switches, and controllers.

Mininet helps with things like research, development, learning, prototyping, testing, and debugging on a laptop or other PC by providing a full experimental network. Several components are needed in the infrastructure of SDN architecture to produce it. The first thing we require is a controller machine running a network operating system, such NOX or POX. Some features of the Mininet:

• It provides a simple and cheap way for testing networks for OpenFlow application development;

• It allows that multiple researchers independently work on the same network topology;

• It allows the testing of a large and complex topology,without even the necessity of a physical network;

• It includes tools to debug and run tests across the network;

• It supports numerous topologies, and includes a basic set oftopologies;

• It provides simple Python API's for creating and testingnetworks.

8. Simulation

In the process of simulation, it creates of this SDN topology using Mininet simulator. Mininet have some diferent numbers of switches, hosts and controller. POX controller is remote controller so it is launched and on the other side, SDN topology was run with different parameters. In addition of, iperf tool is to measure the bandwidth performance.

9. Network Topology

Basically, its like a linear topology that adopted in this reserach which consists of 9 switches and hosts. Switches are connected with openflow controller that enabled switches and support openflow controller. In addition, this nodes perform the role of hosts devices and links created between nodes and switches without any delay. So this is one side of the network.And the other side of network is that POX controller. So all these things are carried out in use of the Mininet which shownon network topology in Fig. 6.So after creating topology, SDNbecome virtual and ready with POX that able to connect remotely on the host machine.It also called as Mininet Virtual Machine (VM).

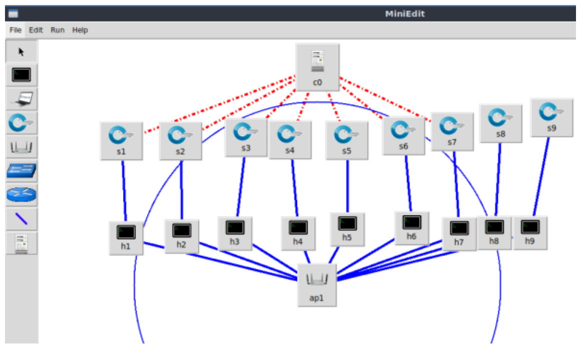

Fig. 6 Network Topology

10. Implementation and Results

In the part of implementation, the performance of POX controller was measured through Mininet in the study. In essence, POX is used whenever the controller is invoked for any purpose because it is the Mininet default controller.All of the switches used here can function with the OpenFlow protocol, which makes switches easy to support and ultimately enables them to successfully interact with the controller in terms of flow. Certain factors were taken into consideration when testing the evaluation of the POX controller's performance. such factors as network speed, allocated CPU load, packet loss, latency, and throughput. So, at basic level of Mininet, Firstly we tried bandwidth network that check bandwidth between sender and receiver. At some point, it increased or decreased based on our internet bandwidth also.

11. Network Bandwidth

Network bandwidth is most important factor for the performance analysis. In the tree topology it consists of 9 switch and 9 hosts. Firstly, the 5mbps of bandwidth was assigned to the switches and then it measured by the iperf tool. In Table No. 1 it describes that packet speed is increase by the300 mbps. It also depends on the speed of wifi that we have used to run this iperf tool. We can also check with some more switched and hosts. So, compare to other network it increased more speed to send packet one host to another host. Network bandwidth is a measurement indicating the maximum capacity of a wired or wireless communications link to transmit data over a network connection in a given amount of time. The more bandwidth a data connection has, the more data it can send and receive at one time. In concept, bandwidth can be compared to the volume of water that can flow through a pipe.

12. Video Streaming

So as per other paper there were some CPU allocation process, some utilization etc. So, in terms of new I did the one of the features of online video streaming. It shows the video stream like a CCTV camera like it shows sender to receiver side. We added some steps to did this streaming. Firstly, We have to open Mininet-wifi and create a topology like Figure.6 . Host 1 is a sender and host 2 is a receiver.

So from the start, we have to open two host. One would be sender and other would-be receiver. After that we have to set ipaddress to both host 10.0.0.1 and 10.0.0.2 likewise. A routing protocol plays major role in data communication. For this video streaming RTP protocol was used. RTP stands for real- time transport protocol. A protocol that designed to handle real-time traffic like audio-video of the internet. This protocol is support different types of streaming methods like MPEG, MJPEG etc. We were using this MPEG streaming method. It stands for Moving picture exports group.

One the Sender side first opens VLC media player through the sender terminal. After opens VLC player go to media stream. Then add the path of video that we would like to stream or shows on cctv camera. Then as I said earlier we have to select file type that is RTP/MPEG transport stream. So next was that click on Add option in this they asked for the receiverip address that is 10.0.0.2 and port number would be 5004.

This 5004-port number is used for Microsoft windows Media Streaming Service. It accepts incoming RTP client connectionand delivers data to the clients. Then we set the profile which indicates transcoding option. In that we select H.264 + MP3(MP4). It is designed to be used particular in any CCTVsystem.

Second the Receiver side, we have to open VLC as like sender side we opened. We just have to add URL in media Open Network Stream. URL is like rtp://@:5004. Then we have toclick play button and sender side Stream button. After that itplays the video from sender to receiver.

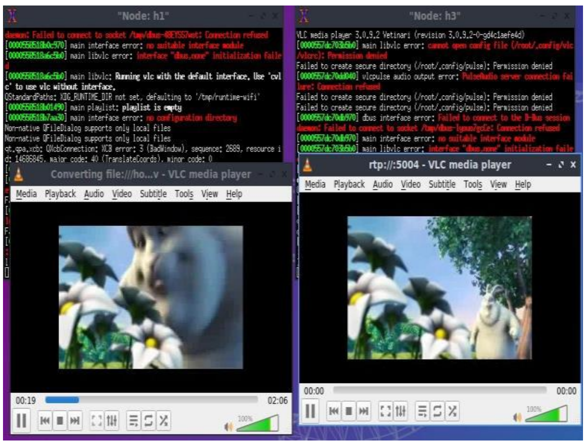

Fig. 7 Video Streaming

According to Table 1 I took different sizes of bandwidth gap like 5, 20, 50, 100, 200, 300. And Hosts are h1, h2, h3, h4, h5,h6, h7, h8, h9. While play this video, we can check bandwidthbetween host and receiver. How this video goes through whichspeed. Table 1 shows results of this video streaming. This table described that how this bandwidth increased when it not has many error while streaming.

When the bandwidth speed was low it means it there was something packet loss over there. When the speed on high there is no packet loss there. It increased in the 50, 100 and 200 mbps in each host. So there is we can say in this speed ofbandwidth is good compare to other. We will also add more mbps in the table upto 500. But at that we have to add more hosts and switched and create a big network.

Table 1: Network Bandwidth

Host	h1						h2						h3					
	5	20	50	100	200	300	5	20	50	100	200	300	5	20	50	100	200	300
h1	-						17	11	15.1	15.6	15.3	12.1	12	14	12	13.1	16.6	12.3
h2	10.7	16.9	15.9	10.0	13.4	12.8	-						12.4	9.19	13.6	15.3	17.1	12.9
h3	12.7	16.2	13.9	13.7	12.9	12.2	10.3	12.3	13.6	10.9	12.9	11.3	-					

Table 2: Network Bandwidth

Host	h4						h5						h6					
	5	20	50	100	200	300	5	20	50	100	200	300	5	20	50	100	200	300
h4	-						16.7	11.8	15.7	15.1	16.7	12.1	13.1	16.2	16.7	13.5	16.0	11.6
h5	16.4	12.4	14.2	12.9	12.8	16.6	-						16	10.4	10.7	10.8	13.1	11.9
h6	12.4	11.9	13.6	11.7	12.9	16.1	10.7	13.2	8.93	16.4	12.7	11.9	-					

Table 3: Network Bandwidth

Host	h8						h9					
	5	20	50	100	200	300	5	20	50	100	200	300
h8	-						12.6	14.6	12.3	15.0	10.9	11
h9	13.6	11.1	11.5	13.8	16.3	11.6	-					

Fig. 8 Graph of network bandwidth

In Fig. 8 refers to the I/O graph that is statistics IO graph forthe slow transfer. For that we can visual what we can see that pattern looks like because that trace file is got a lot of

parameters. So it's a from of bits. We can also change that from bits to packets. The graph is like up-down because whenyou lose data you have to resend it until you figure it out. So the pattern is like low and high like low means wait packet is lose and high means resend of packet that lose.

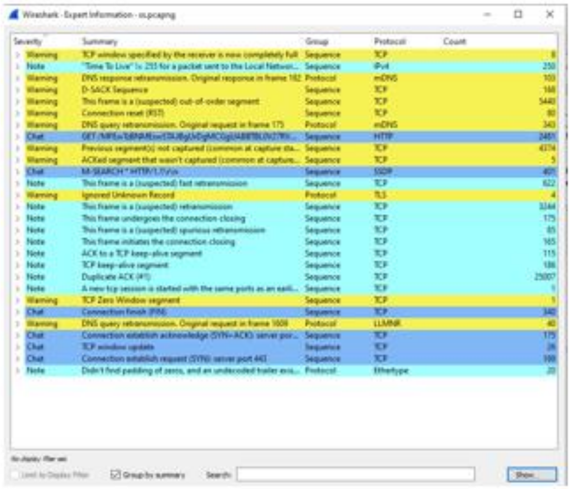

Fig. 9 Expert Information

In addition to, Fig. 9 contains that there's waring, notes, errors,chats that shows the summary of packets, group of packets. It also counts the number of packets that is from warning, notes or other thing.

13.Conclusion and Future Work

For the architecture of the Future of the Internet, the conceptfor Software Defined Networks holds out a lot of potential. Despite the fact that its potential is only now being fully understood, it has already attracted a lot of attention from academia and business. This paradigm is quite promising, as evidenced by the fact that there are many devices on the market using OpenFlow protocol that has been commerciallyenabled.

This essay discussed the current state of computer networks and the challenges they face in keeping up with the demands of emerging network applications. It was observed that the current architecture is saturated and it has difficulties to meet the new market demands. In the summary of the paper, main focus on the bandwidth network that have some accuracy between 50 to 200 mbps. POX controller is used iperf tool for traffic analysis. Iperf tool had been used for assessment parameters such as bandwidth, delay, window size and packet loss. It is for both TCP and UDP traffic. It is GUI interface so it can be installed on Linux and windows system. In the video streaming part, instead of hardware components I tried through the simulation. Hardware components casts are to high. About video streaming, it is a like CCTV streaming. It is one of the feature of Home automation that if sender wants to see the recent CCTV footage so it is possible. Beside, we are checking their bandwidth using iperf tool.

In the future work, we can also check some parameters like packet loss, CPU allocation etc. For check this more parameters we have to create such a big network for the organization. Considering the fulfillment of the SDN paradigm and all of the diagnosed challenges in this take a look at, it is discovered that there may be a massive area for the improvement of recent research projects focused on SDN.

Refernces

Bera, Samaresh, Sudip Misra, and Athanasios V. Vasilakos. "Software-defined networking for internet of things:A survey." IEEE Internet of Things Journal 4, no. 6 (2017): 1994-2008.

Al Shuhaimi, F., Jose, M. and Singh, A.V., 2016, September. Software defined network as solution to overcomesecurity challenges in IoT. In 2016 5th International Conference on Reliability, Infocom Technologies andOptimization (Trends and Future Directions) (ICRITO) (pp.491-496). IEEE.

Alwasel, K., Jha, D.N., Habeeb, F., Demirbaga, U., Rana, O., Baker, T., Dustdar, S., Villari, M., James, P., Solaiman, E. and Ranjan, R., 2021. IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum. Journal of Systems Architecture, 116, p.101956.

Al Hayajneh, A., Bhuiyan, M.Z.A. and McAndrew, I.,2020. Improving internet of things (IoT) security with software-defined networking (SDN). Computers, 9(1), p.8.

Setiawan, D.Y., Hertiana, S.N. and Negara, R.M., 2021, January. 6LoWPAN Performance Analysis of IoT Software-DefinedNetwork-Based Using Mininet-Io. In 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS) (pp. 60-65). IEEE.

Fontes, R.R., Afzal, S., Brito, S.H., Santos, M.A. and Rothenberg, C.E., 2015, November. Mininet-WiFi: Emulating software-defined wireless networks. In 2015 11th International Conference on Network and Service Management (CNSM) (pp.384-389). IEEE.

Mir, N.F., Kotte, J.N. and Pokuri, G.A., 2016, June. Implementation of virtualization in software defined networking (SDN) for data center networks. In Proc. ICN (p.136).

Qin, Zhijing, et al. "A software-defined networking architecture for the internet-ofthings." 2014 IEEE network operations and management symposium (NOMS). IEEE, 2014.

Kreutz, D., Ramos, F.M. and Verissimo, P., 2013, August. Towards secure and dependable software-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp.55-60).

Sivaraman, Vijay, et al. "Network-level security and privacy control for smart-home IoT devices." 2015 IEEE 11th International conference on wireless and mobile computing, networking, and communications (WiMob). IEEE, 2015

Noman, Haeeder Munther, and Mahdi Nsaif Jasim. "POX controller and open flow performance evaluation in software defined networks (SDN) using mininet emulator." IOPconference series: materials science and engineering. Vol. 881.No. 1. IOP Publishing, 2020.

Prete, Ligia Rodrigues, et al. "Simulation in an SDN network scenario using the POX Controller." 2014 IEEE Colombian Conference on Communications and Computing (COLCOM). Ieee, 2014.

Noman, Haeeder Munther, and Mahdi Nsaif Jasim. "POX controller and open flow performance evaluation in software defined networks (SDN) using mininet emulator." IOPconference series: materials science and engineering. Vol. 881.No. 1. IOP Publishing, 2020.

Al-Hasnawi, Sally A., and Mahmood K. Ibrahem."EMULATION OF THE POX CONTROLLER AS A LOAD

BALANCER." Iraqi Journal of Information and Communication Technology 4.2 (2021): 9-22.

De Oliveira, Rogério Leão Santos, et al. "Using mininet for emulation and prototyping software-defined networks." 2014 IEEE Colombian conference on communications and computing (COLCOM). IEEE, 2014.

Muragaa, Wisam H., Kamaruzzaman Seman, and Mohd Fadzli Marhusin. "A POX Controller Module to Prepare a List of Flow Header Information Extracted from SDN Traffic." International Journal of Computer and Systems Engineering 11.12 (2017): 1305-1308.

Goransson, Paul, Chuck Black, and Timothy Culver. Software defined networks: a comprehensive approach. Morgan Kaufmann, 2016.