

A COMPARISON OF LOAD BALANCING STRATEGY IN SOFTWARE DEFINED NETWORKING

Prabhjot Kaur
M.Tech (Scholar)

Department of Computer Science & Engineering,
Punjabi University, Patiala
prabh.kr92@gmail.com

Dr. Abhinav Bhandari
Assistant Professor

Department of Computer Science & Engineering
Punjabi University, Patiala
bhandarinitj@gmail.com

Abstract— SDN is the present big thing in network industry. This report includes the SDN architecture, benefits and simulation best practice using Mininet and POX controller. It also explains and helps in understanding how the controller can be implemented on AWS EC2 Linux machine and creating load balancer using POX load balancer algorithm. Cloud based SDN load balancer helps in balancing the load on machines running in different or same data centers. Software defined network load-balancers are programmable and allow it to implement and develop your own load balancing stages. Other benefits of software defined network load balancer is it don't require dedicated software. In this paper, we implement and comparing the various method like round robin and scheduling algorithm with already designed random methods using NS-2 Simulator.

Keywords— SDN (Software Defined Networking), EC2 (Elastic Compute Cloud), VLAN (Virtual Local Area network) and AWS (Amazon Web Services).

I. INTRODUCTION

Today internet applications require fast networking which caused large amount of traffic and deploys number of distinct, dynamic applications and services. This management makes network highly complex and is more difficult for the network administrator to handle concepts like network virtualization also interchangeable data is added [1]. The configuration and installation of the network demands skilled personnel. The network nodes such as switches and routers has complicated interactions. System based approaches requires simulation of elements but due to programming interfaces it is difficult to access [2].

The term software defined networking (SDN) was coined few years back but its idea is changed since 1996 as driven by user control management need to forward in the network nodes. There are many projects such as Ipsilon, The Storm and Internet Engineering Task Force (IETF) Forwarding and Control Element Separation, 2000. Latest, Ethane (2007) and Open Flow (2008) required the usage of the software defined network. Open Flow facilitates the flow table by enabling entries which are defined by external server.

In SDN (Software Defined Network), controller plays important role to improve the control plane and monitor the behaviour of network as supervisor provides software interfaces of system and maintenance of global viewpoint of network. In the manner new functions and management job can be done by the several applications.

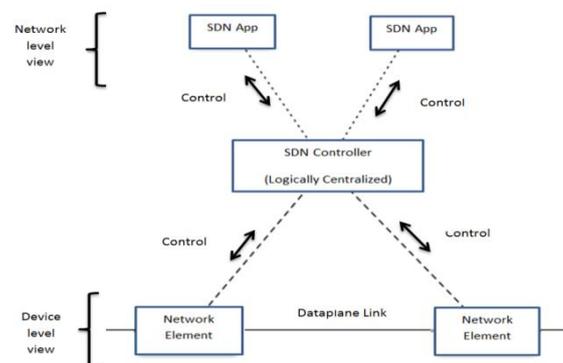


Fig.1. Programmatically Abstracted Interaction with Network

Data plane forwards conventional flows allowing to the protocols resulting from the supervisor and restrictions are put by the controller on all flow through consecutively of user-defined control requests. Benefits of software defined network are lost due to the failure of controller in the network [3].

A. Characteristics of SDN

The features of software defined network are Separating switch plane after data-plane, network view, central supervisor, open boundaries between the equipment and network program ability [2]. System devices include control plane and data plane that decide how network traffic is handled, and traffic forwarding according to the control plane decision respectively [1]. Communication network is made up of end users, hosts connected by the infrastructure of network and hosts share the infrastructure and use routers, communication links, and switches to carry data [4].

The research work is partitioned into different section. The first section is about the basic information about SDN. The section II involves the architecture of SDN. Third section

composed the literature survey and the fourth section is about the proposed work and methods. It involves load balancing and the available protocols. Further the result is discussed and at last the conclusion and future scope is given.

II. SDN ARCHITECTURE

The information technology have incapability to acquire the bestest implementation via the system such as the the traditional software devices. For instance the routers, switches and the load balances. Subsequently, for enhancing the security the firewalls are considered to get the things properly managed. The requests of the system customers or the clients are expands one by one. There is the necessity to manage the every single gadget completely. The number of gadgets are varied and come under the thousands and more. The process to manage and to set the protocols and other devices are boring as well as difficult job. For instance the need of switches are because to include and to diminish the gadgets from the particular system, ACL (Access Control List), other switches and the firewalls to enhance the security. The process is entirely unpredictable, long and certain problems occurred [5].

In the system gadgets, there are three planes as the data, control and the applications. The basic employment of IP (Information Plane) is transmitted the data. The basic employment of software defined network is controlled the transmitted plane. It is finalized the application plane via driving the stream rules under the transmitted plane. In these system gadgets, the information, control and application planes are managed as shown in fig.2. The overall setyp is easy and simple due to the deattachment of the information and the control plane [6]. The open flow conventions are used to symbolize SDN (Software efined Network). The programming characterized organizing is the fundamental convention which trained for transmission plance and control plane. For this purpose, the message trade is used via Open Flow convention [7].

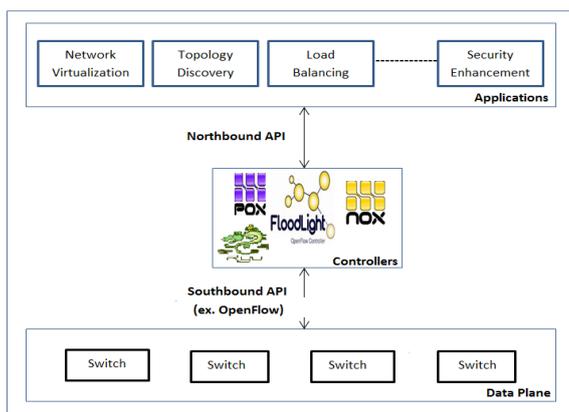


Fig.2. SDN Architecture

In the figure 3 the idea of SDN is shown that is detachment of the control layer and merging into a single point of network. It refers to the processing of every single device in the network pass the packets of data from the particular place to the next coming place under the choices of SDN controller.

Through the API particular controller controls the particular switch and the application layer SDN applications is directed with the controller. Following terms are the layers.

- 1) *Data or Information Layer*: The hardware exists that are physically connected in the layer of the network and the programming is processed on the system gadgets which gives the basic control plane interface. This is later associated to the upper dimensions of the southbound API.
- 2) *Control Layer*: The control layer is the most essential layer in the network and it specifically saced the topology track which contains the cotroller and later it used for the communication of different devices in the particular framework. When the data is shuffled from the state of the network to the upper layers via using the Northbound API. The controls worked to generate the sensible directions for the gadgets.
- 3) *Application layer*: It defines the features, services and policies. Applications create new features with big decisions according to the network changes. Whenever the topology or the feature changes the applications get the control to change the behaviour of the network which can done from a single point.

B. North-bound API

The controller gives the Northbound API and to manage communication by the applications they have to use the API to get to the controller. The whole SDN network is controlled by this method. It is applied by restful API as through this it makes easier to control with the usage of HTTP methods such as POST, PUT, GET and DELETE.

C. South-bound API

The South-bound API gives the correspondence interface among the controller and the different network approaches. This is the basic and the central for the software defined network (SDN) controller while the administration of switches is done through the API by the controller. The controller of SDN has the responsibility of seeing the network. The configurations are essential flows for every switch in the network. The switches are talking about the transmission of packets. In some situations the switches are become incapable to take the final decision then the process is set to the default action.

D. SDN Controllers

A. Three layers in SDN Architecture

The Software Defined Network controllers coined in the short term as SDN controller and are considered as the mind of the system. The software defined network controller covered the essential control points in the network that worked with the stream control and the switches via the south-bound API. The business logic is completed by using the north-bound API. In this way, the controllers are assumed as the smartest networks. The software defined networks are reliant upon the servers mainly to guide the changes in which the data packets are send and depends upon the open flow protocol.

- 1) *NOX*: NOX is both an exceptionally old form of the controller and it is relied on the modules design to create the fundamental applications of SDN. Further, NOX gives the bestest modules correction openflow that are later increased. The centre of NOX gives partner techniques and APIs to interfacing with OpenFlow switches, including an association handler and occasion motor. Extra parts that advantage that Programming Interface is available, including host following, steering, topologies along with the Python line executed as a covering for the module Programming Interface.
- 2) *POX*: POX is the more up to date; the dependency is upon the python which is associated to NOX. Further, the improvements are returned NOX for its basic C++ and to generate the well defined Python base such as Python 2.7. Consequently, it composed of the abnormal states with the capability of topology charts as well as the virtualization.
 - a) It has a Pythonic OpenFlow interface.
 - b) It has also reusability which are the examples to determine the topology and to discover.
 - c) POX capable to process everywhere and it is compressed with the free PyPy implementation specifically for the easiest utilization.
 - d) It explicitly focused on the Linux operating system, Max operating system and Windows.
 - e) It bolsters are the easiest and the simple Graphical user interferences and contains the perception apparatuses like NOX.
 - f) It makes all around contrasted with NOX services written in Python.
- 3) *Ryu*: Ryu is part based and the open source introduced by NTT labs. The framework implemented entirely in the programming of Python. Ryu informed the administration supports parts created in different dialects. The open flow wired bolster composed the Nicira expansions, occasion the administrators, informing, in memory state the board, application the administrators, foundation administrations benefit. For instance NETConf libraries and the netlow libraries. Ryu likewise bolsters has the REST interface to its OpenFlow tasks.
- 4) *Flood-Light*: this is the rapidly fascinating in the SDN controllers that initialized to impact from the beginning of Big switch systems specifically for the open source

systems. the dependency is on the Beacon. The flood-light is the apache authorized and relied on the Java with the open flow controllers. The framework of flood-light with the API is utilized differently with the big switch systems and it is seen in BNC (Big Network Controller).

- 5) *Open Day-light*: An Open Day-light venture is come under the open souce phases specifically for SDN (Software Defined Networking) which used the open conventions to give access for automatic control andchecking of system gadgets. Similarly, the other terms of software defined networks, Open Day-light under the Open Flow, and it manages the arrangements as a feature of its stage.

III. LITERATURE SURVEY

Sukhveer Kaur et al., [8] introduced an SDN application for executing server load balancing. The basic idea of SDN consists of logical centralized network control in SDN controller. To solve the issue of unnecessary latency they implemented direct routing based load balancing algorithm.

Mu Sub et al., [4] proposed a two LVS based technique to decrease the problem of over-head of the supervisors and forward loop, and to reduce controller synchronization. When load of a server exceeds an amount more than the threshold the LVS handles the synchronization state.

Luo et al. [9] proposed PTAA that can erase the problem of balancing in a way of effective parallel. Addressed load balancing of in band manage rush and evaluated the performance confirmation of demonstrating communication of PTAA with least eighty percent delay reduction through an approach which is fast and had low complexity. The location of the controller is additional used for the finding the optimize network end to end delay.

D. Perepelkin et al. [10] developed software application and visualized design environment in software defined networking with quality of service. The paper has focused on set pavement of reserved routes that are set accordingly to the indicator of QoS and paved route based on paired shifting with balance the load. Giving to the indicators of way difference the alignment and optimization of load is done.

S. Attarha et al., [11] to avoid the over utilization of link in the environment of software defined networking proposed an efficient algorithm. The manager is accountable for the monitoring the condition of the system and decide switches to be used occasionally. The new movement is transmitted towards the possible direct path but in case when path is not blocked with traffic otherwise controller find a back-up path. To resolve the issue of traffic overloading the algorithm selects the lowest possible number of flows and if not they are directed to the backup paths.

U. Mahlab et al., [12] proposed a strategy called discontinuity aware load adjusting for optical systems that upgrade the heap dissemination over the system. The localization minimizes cost for the new link development. To measure load imbalance, proposed entropy based metric.

IV. PROPOSED WORK

In this section, the load balancing and several kinds of validation protocols are described.

A. Load Balancing

Load balancing is necessary constraint for initiative networks so it can provide high availability and scalability. Load balancing works as an aware routing protocol software defined network (SDN). The entity that assists the availability and scalability that leads to obtain the minimal response time of application. To solve this problem load balancing techniques method increases the network efficiency [13]. Present framework method of balance the load isn't useful aimed at the framework of Software Defined Network (SDN), where the main station or manager of network basics is completed by programs. To manage the load of server is easy which is connected to the controller of the framework [14]. It also helps the clients from connecting directly to the backend servers which secure the structure of internal network and prevents from attack on the network [15].

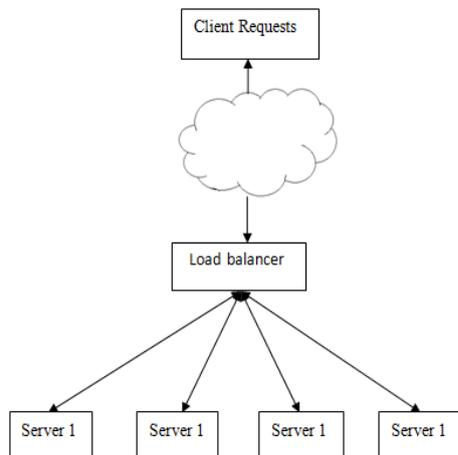


Fig.3. Load Balancing Architecture

- 1) *Load Balancing Algorithm and Methods:* Load balancing utilizes different calculations, called load balancing techniques, to characterize the criteria that the ADC application uses to choose the administration to which to divert every customer demand. Unlike load balancing systems utilize distinctive criteria.
- a) *Least Connection Method:* The default strategy, when a virtual server is arranged to utilize the minimum

connection, it chooses the administration with least dynamic associations.

- b) *Round Robin Method:* This strategy constantly pivots a rundown of administration that is appended to it. At the point when the virtual server gets a demand, it appoints the association with the main administration in the rundown, and after that moves that support to the base of the list.
- c) *Least Response Time Method:* This technique chooses the administration with the least dynamic associations and the most minimal normal reaction time.
- d) *Custom Load Method:* When utilizing this technique, the load balancing machine picks an administration that isn't dealing with any dynamic exchanges. On the off chance that the majority of the administrations in the load balancing setup are taking care of dynamic exchanges, the application chooses the administration with the least load.

A. Various Available Protocol Validation Tools

There are many validation tools available in the market through which the process of validating is performed. Some of the tools are:

- 1) *AWS:* Amazon Web Services (AWS) is an entire and exhaustive, changing distributed computing stage given via Amazon. This will blend IaaS (Infrastructure As A Service), PaaS (Platform As A Service) and SaaS (Packaged Software As A Service) advertising. Amazon Web Services boosted in 2006 via thye internal organizations of Amazon. It worked for managing and processing of online marketing activities. Amazon web services (AWS) is the fundamental organization which represented the compensation to distribute the computational models which serves the clients with the availability of stockpiling and quantity.
- a) *Computer:* The Elastic Compute Cloud (EC2) gives virtual systems called occasions to ascertain measures. The elastic compute clouds (EC2) benefit gives access to the heaps of the present kinds via changing the measurements and dimensions, customized to the particular remaining task at hand types and applications, for example, memory-serious and upgraded figuring employments. AWS additionally gives an Auto Scaling device to vigorously scale capacity to keep up occurrence wellbeing and execution.
- b) *Storage:* S3 is given by Amazon which stands for the simple storage services. S3 gives the better stockpiling for the information, reinforcement and for the better segments. Information Technology (IT) proficient stores information and documents like the simple storage services (S3). It is implemented on 5GB and pails to save the data. The businesses must be organized with extra caches to expand the storage level and to utilize the amazon storage for long term.

- c) *Databases*: AWS gives oversight data-bases benefit done by ARDS (Amazon Relational Database Service), that incorporates alternatives (Oracle, SQL Server, MySQL, MariaDB) and a private superior data-base as namely as AA (Amazon Aurora). Amazon web services offered No SQL data-bases via using Amazon Dynamo database.
- d) *Alternarite Services*: AWS has the objective for the industry generation such as SaaS alternatives. It is beneficial for empowering the virtual video gatherings, requests and content established talks over the gadgets.
- 2) *Amazon Elastic Compute Cloud (EC2)*: Amazon EC2 gives the essential registering capability mainly in AWS cloud that are the amazon web services. The development of the elastic compute cloud described the different equipment for the better construction and to solve the problems quickly.
- 3) *Mininet*: The term mininet is the emulator which utilized for assembling the extreme systems on the constrained terms in the particular PC (Personal Computer) and VM (Virtual Machine). It gives the permissions for the data packets to transmit successfully with the encryption of data. Additionally, it gives more availability and the better sense without accessing more knowledge and effort. It gives more access for the better utilization, implementation and the versatility [16].

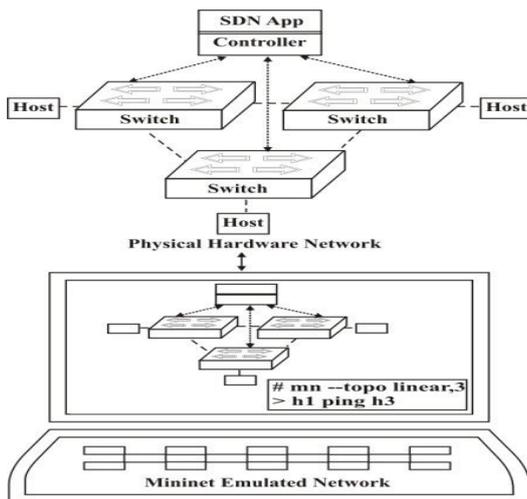


Fig.4. Emulating Real Networks in Mininet [16]

The mininet is the simplest and most preferable open source programming which performed the survey on the open flow processes and to describe the software defined network (SDN) controllers. It has the tendency to generate SDN systems which executed the controllers for tests [17].

- 4) *Mininet Topologies (MT)*: MT contains the large number of the basic topologies for instance the minimal singles, reversed and the linear [18]. It is tolerated and proven approach for the terminals hosts and mainly for the several kinds of switches for the upcoming mininets. The switches are initialized from S1 to SN. The hosts are named as H1 and HN. The interfaces are combined with

the host names via using Ethernet to establish with zero. The first interfaces are known as H1-eth0 and the third interfaces are known as H2- eth2. The first switch is named as S1-eth1 and the numbering is start with 1 [16].

V. RESULT AND DISCUSSION

Mininet is open source software and used to simulate a SDN. As SDN separates the two planes from each other thus making network devices fully programmable which means the network will work upto the user need. Mininet provides OVS controllers and switches. The primary feature that separates SDN devices from traditional network devices is the modification of protocols and functions is the points which differentiate SDN devices from traditional ones. It also supports the OpenFlow protocol that act as an interface between the planes. The OpenFlow controls the flow of packets on the instructions of API. Supporting various topologies and allow to customize the topologies. A little knowledge is required to work with the command line interface.

A. Features of Mininet

- 1) It is inexpensive network for developing applications of OpenFlow.
- 2) Enables developers to make parallel work that can independently done on the same topology.
- 3) Supports regression tests at the system level.
- 4) Enables complex topology testing.
- 5) Includes a CLI to debug or run the network-wide assessments.
- 6) Supports random customized topologies, and contains a simple set of parameterized topologies.
- 7) Can be used without programming.
- 8) Provides Python API to create network.

- B. *Test results*: To well plan routing processing of the clients to the particular servers mainly in the imporved approaches and there are numerous load balancing.

Load Balancing Strategies:

- 1) *Random*: Traffic towards web servers go into random manner. You cannot control or predict about the load balancing ratio. It is not recommendable for most of the environment. SDN controller runs this random load balancing application and diverts the traffic to web-servers in random manner.
- 2) *Weighted Round-Robin*: In this type, we can add weight to the web servers. More the weight, better the chance that traffic will be forwarded to that. This type of method is used mainly when there are servers with different capacities in the network all we want is to transfer traffic to servers with larger capacity. Every server gotten the requests from the clients which are fixed.

3)Round-Robin: This method makes one-by-one shift when traffic from clients go towards web servers. This method can be used when all the servers running web application have same set of capacity [RAM, Processing Power, and HDD].

C. VPS (Virtual Private Server)

VPS is server that exists in on a server with respect to the virtual machines. With a VPS, we have our own operating system installation.

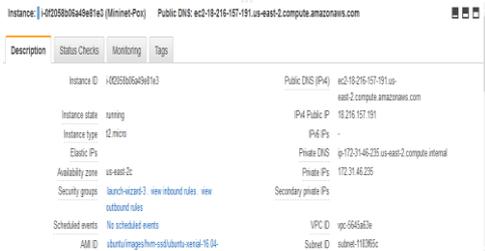


Fig. 5. VPS IP address assigned to the instance by AWS

Now we will create topology with one controller, one switch and 6 hosts in Fig.6.

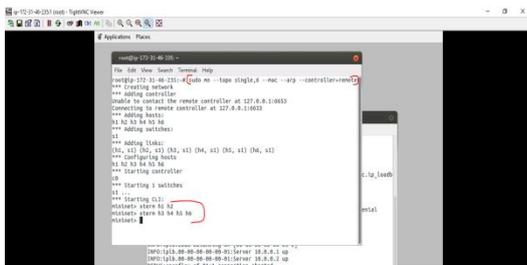


Fig. 6. Mininet Topology Creation

We have 2 two hosts as “h1”, “h2” that we run as server and from another “h3”, “h4”, “h5”, and “h6” we request HTTP service by using curl command.

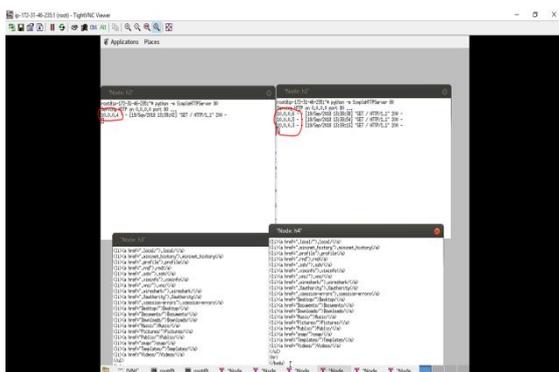


Fig. 7 Server Setup on h1, h2 and Traffic from Client Node h3, h4

The rewriting of the destination IP address is done by the load balancer. The decision of where to send a request is

made by the round robin using a circular queue. The request to the server with low load is based on the load based policy while load is the frequency of the request that is pending.

Using Mininet , a custom topology is created. After that the server is to set up. HTTP server is made with 80 port number of the node one and two. The servers should have their individual IP addresses shown in the figure 13.

Now, traffic is sent to the server by curl command used by the http clients shown in the figure 13. From this a webpage is got from the server’s IP. Round robin algorithm helps the client to get server in a circular manner. Figure 14 shows the different nodes and the direction of the traffic to the servers.

To run ip_loadbalancer we need pass arguments. ./pox.py misc.ip_loadbalancer -ip=<Service IP> -servers=<Server1 IP>,<Server2 IP>,... [-dpid=<dpid>]

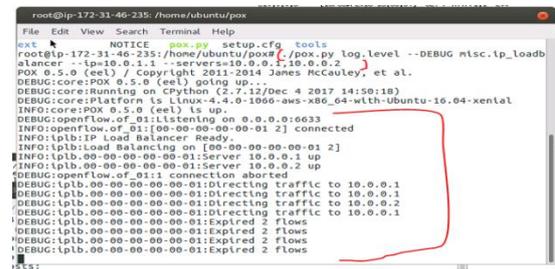


Fig. 8. Directing Traffic

ip_load_balancer is a TCP load balancer which takes a service ip and the list of internet protocol addresses of the servers. Client requests are directed to various pre-defined http servers while the round robin chooses the server.

Traffic will be provided to that this type of method is used namely when those are servers wise different capacities in the network and all we want is to tend more traffic to server with larger capacity.

Round Robin method makes one-by-one shift when traffic form clients go towards web-servers. These methods can be used when the entire server running web-application same have set of capacity like as a user, processing power and HDD.

D. Result:

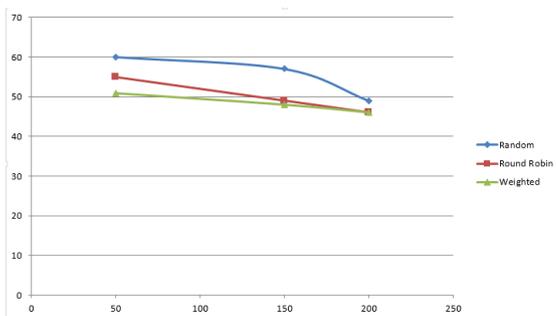


Fig 9. Comparison

The above figure shown that the comparison between proposed and existing work. Random algorithm best perofrmnace analysis based on one at a time transfer in minimum time and heavy or weighted algorithm to calculated the weight accordingly data transferring and Round Robin Algorithm defined used the same link activated in equal memory used. In this comparison define the load balancing the customer requirement.

VI. CONCLUSION AND FUTURE SCOPE

SDN has changed the network industry and the plethora of benefits that it provides makes it a super success in Data Center and service provider industry. Load Balancing is a need in Data Centers where multiple servers are running same application for better redundancy and processing. Pox comes by default in mininet which can be used the controller and with mininet based switches (OVS). The basic protocol is the openflow and utilized among controller besides infrastructure layer. With load balancing, all the servers in the server zone works parallel and which automatically makes our data center network much more efficient than before. AWS is amazon's cloud which runs application, host virtual machine and stores data. POX controller is s implemented in the AWS EC2 cloud over a Linux machine and we made a cloud based controller which is acting as a load balancer for other servers. Lots of research work is still on in the SDN and as a result lots of new applications and technologies are coming up as a part of research like 5G, SD-WAN etc. I want to take my research further in SDN based LBs clustered in the cloud, which can be used in application and networks infrastructure forming the cloud and makes the cloud based infrastructure much more efficient than using physical traditional network infrastructure.

REFERENCES

- [1] Sood, M. (2015). SDN and Mininet: Some Basic Concepts. *International Journal of Advanced Networking and Applications*, 7(2), 2690.
- [2] Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., ... & Rao, N. (2013). Are

we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7), 36-43.

- [3] Xie, J., Guo, D., Hu, Z., Qu, T., & Lv, P. (2015). Control plane of software defined networks: A survey. *Computer communications*, 67, 1-10.J.
- [4] Zehua, G., Mu, S., Yang, X., Zhemin, D., & Luo, W. (2014). Improving the performance of load balancing in software-defined networks through load variance-based synchronization [J]. *Computer Networks*, 68, 95-109.
- [5] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3), 1617-1634.
- [6] Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17(1), 27-51..
- [7] Ramos, F. M., Kreutz, D., & Verissimo, P. (2015). Software-defined networks: On the road to the softwarization of networking. *Cutter IT journal*.
- [8] Kaur, S., & Singh, J. (2016). Implementation of Server Load Balancing in Software Defined Networking. In *Information Systems Design and Intelligent Applications* (pp. 147-157). Springer, New Delhi.
- [9] Lin, S. C., Wang, P., & Luo, M. (2016). Control traffic balancing in software defined networks. *Computer Networks*, 106, 260-271.
- [10] Perepelkin, D., & Byshov, V. (2017, April). Visual design environment of dynamic load balancing in software defined networks. In *Radioelektronika (RADIOELEKTRONIKA), 2017 27th International Conference* (pp. 1-4). IEEE.
- [11] Attarha, S., Hosseiny, K. H., Mirjalily, G., & Mizanian, K. (2017, May). A load balanced congestion aware routing mechanism for Software Defined Networks. In *Electrical Engineering (ICEE), 2017 Iranian Conference on* (pp. 2206-2210). IEEE.S.
- [12] Mahlab, U., Omiyi, P. E., Hundert, H., Wolbrum, Y., Elimelech, O., Aharon, I., & Zarakovsky, S. (2017, July). Entropy-based load-balancing for software-defined elastic optical networks. In *Transparent Optical Networks (ICTON), 2017 19th International Conference on* (pp. 1-4). IEEE.U.
- [13] Kumari, P., & Thakur, D. (2017). Load Balancing in Software Defined Network.
- [14] Raghul, S., Subashri, T., & Vimal, K. R. (2017, March). Literature survey on traffic-based server load balancing using SDN and open flow. In *Signal Processing, Communication and Networking (ICSCN), 2017 Fourth International Conference on*(pp. 1-6). IEEE.

- [15] Pathak, P., & Mahajan, E. K. (2015). A Review on Load Balancing In Cloud Computing. International Journal Of Engineering And Computer Science, 4(06).
- [16] Kaur, K., Singh, J., & Ghumman, N. S. (2014, February). Mininet as software defined networking testing platform. In International Conference on Communication, Computing & Systems (ICCCS) (pp. 139-42).
- [17] Lantz, B., Heller, B., & McKeown, N. (2010, October). A network in a laptop: rapid prototyping for software-defined networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (p. 19). ACM.
- [18] Sharma, K. K., & Sood, M. (2014). Mininet as a container based emulator for software defined networks. International Journal of Advanced Research in Computer Science and Software Engineering, 4(12).