International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

Dynamic Load Balancing In Software Defined Networking

Gaurav Tiwari, V Deeban Chakaravarthy, Aditi Rai

Abstract: ***In this research paper we propose an algorithm of Dijkstra's algorithm of finding the shortest path from one point to other. It generates a SPT (shortest path tree) with given source as root. Two sets are maintained, shortest path tree included vertices is contained in set one, other set includes vertices which are not included in shortest path tree. At all steps of the algorithm, a vertex can be found which is in the other set and has minimum distance from the source. Computer networking has scaled great heights in past few years. And traditional networks had failed the s of many of the people. People want quality of service and hence various network applications provide quality of service to customers. In this paper we try to balance load using software defined networking with the use of RYU controller using mininet tool. Nowadays a lot of data is sent by various people using various networks. There should be no data loss and nd there should be no delay in time taken for data transmission. We develop new mechanism for guaranteed latency requirements of those applications who give guaranteed delivery of data.***

I. INTRODUCTION

Now a days a lot of data is being transferred in software defined networking. There is a lot of traffic that has increased to a great extent. There is an urgency to deal with this type of congestion problem using an efficient algorithm networks can be improved. Software defined networking helps in decoupling dara and control plane. Thus the network administrator manage their network by using an external SDN controller.A special type of controller is used that has the capacity to change the forwarding behavior of the network. This behavior has to be changed directly. The network framework that is depicted by SDN is flexible to a great extent and the network administrator can take the profit of the programmability of SDN enabled switches. The use of controller reduces the cost, simplifies operations, optimizes resource usage by using by algorithm and also simplified the software upgrade. There are various protocols of SDN but OpenFlow is the best one by using this protocol the controller con put flow entries to the respective switches and thus controls the forwarding rule of all the switches.

II. MININET INSTALLATION

Mininet is a emulator of network which creates a network of virtual host , switches, controllers and links.

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
Gaurav Tiwari, Btech Department of Computer Science and Engineering, SRM University
V Deeban Chakaravarthy, Assistant Professor (Sr.G) , Department of Computer Science and Engineering, SRM University
Aditi Rai, Btech Department of Computer Science And Engineering, SRM University

Mininet hosts runs standard linux network software and its switches support open flow for highly flexible custom routing and Software Defined Networking.

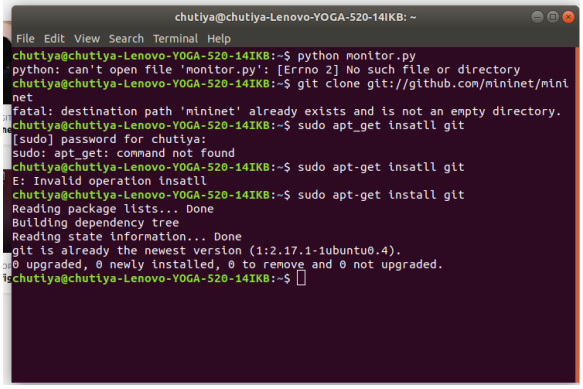

Fig. 1: Mininet installation

III. TOPOLOGY

Now a days, the software defined networking has gained a lot of attention in the software industry. It is an efficient method for network information, collection and managing network management. As already told these algorithms decide the path of data transmission of the particular traffic. Thus it wholly relies on status of network and the time of transmission, but when the status of network will change the path also should be automatically updated, if it doesn't happens then there will be an imbalance of load. To stop this various protocols are used. We have used the following steps to install mininet and thus created the topology

- Created a topology using mininet tool
- Used a controller named ryu
- Connected mininet with ryu controller
- Tested and pinged host to host connection using command pingall or sudo mn --topo single, 3 --mac --switch ovs --controller remote.
- Analysed the packet using wireshark and obatined the various protocols used to transferr data and latency and network strength was calculated.
- Now algorithm will be formulated oncE we get parameters for RYU architecture like packet count, packets received etc for further implementation of dynamic load balencing.,

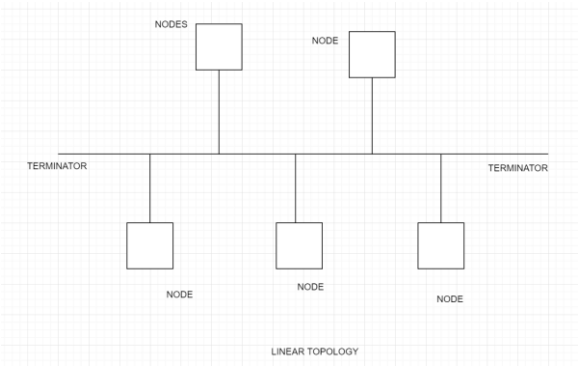

Fig. 2: Linear topology

We have used linear topology in ryu controller, and we have found out the bandwidth and latency using the number of packets that have passed through it in the given amount of time. We have also found out the protocols in which data is being transferred using the wireshark. After developing our project using ryu controller we have decided to use pox controller and have implemented it using a fat tree topology. Fat tree topology is the simplest of all topology as it consists of a root and non blocking switches and non blocking links. The diagram is explained as below.

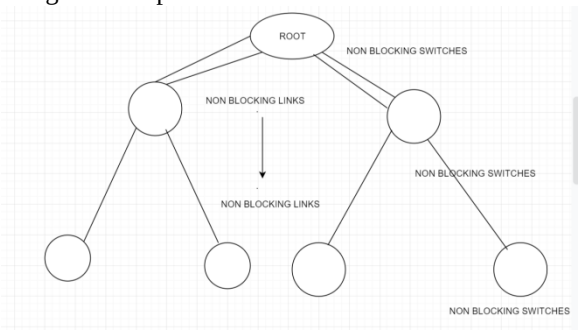

Fig. 3: Fat tree topology

The network applications cant change their style to meet the requirements of end users. Here the problem arises Software defined network has proposed the new paradigm for computer networks.

IV. ALGORITHM

The other problem that we deal is that of latency. Data should be sensed and signals should be transmitted within the given frame of time so that the system works effectively. We have used dijkstra algorithm to find the shortest path from among the given range of path so that the data is transferred in the given amount of time. This algorithm plays a crucial role in sending and collecting the data and transferring it to the destination port address. It can be depicted in the following diagram.

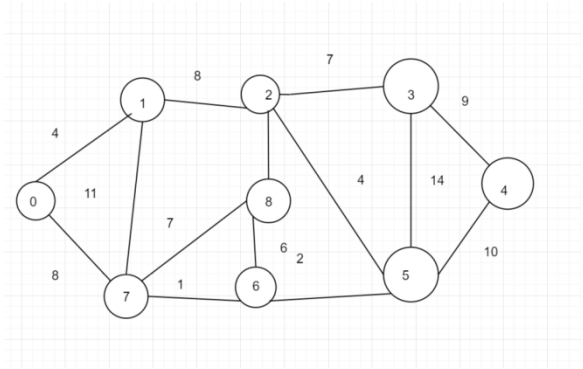

Fig. 4: Dijkstra Algorithm Topology

The network topology that is created with Mininet Emulator and which provides network flows, switches and hosts. POX is used to discover the network topology created in Mininet. OpenFlow switches created in Mininet forms the data plane of the network. POX uses Dijkstra's Algorithm to find the shortest path between a source and a destination. An POX Application is created to redirect the traffic flowing through the OpenFlow switches. Latency is found out and also the bandwidth. Threshold bandwidth is found out and finally packet split is seen. sssThere are some factors that these packets have to being a queue and at the same time there is a lot of traffic. The higher priority queue and switches are to be allowed to go first. We can understand it in other words such as if bandwidth can be used by others if it is not used by the reserved traffic. But the priorities of the reserved traffic is high and hence there is no delay in control and sensor traffic in the congested network and routers and thus we are able to keep end-to-end latency in limit. New routing paths can be explored and these applications can use the concept of SDN. There are new transmission data for the transfer of data in big data transfer such as cloud computing etc. We don't use one TCP protocol for the transfer of data but we use multiple TCP protocols to transfer the data effectively. We have a good knowledge about network topology and we have SDN controller that may be RYU or Pox. Thus we can develop algorithm and hence can improve the overall throughput of the system and make the system more effective.

V. PROBLEMS SOLVED

Here we use few mechanisms to overcome the problems that have arisen due to traditional network system and provide better quality of service to selected networks. The main works of our project is:

- Improve the quality of service to improve the demands that are associated with the bandwidth. We use SDN and monitor the status of the network and and then set the path for bandwidth demanding traffic flows.

- Provide quality of service to the devices that are sensitive to latency. Different class of traffic flows are adjusted and accommodated with different levels of traffic requirements.

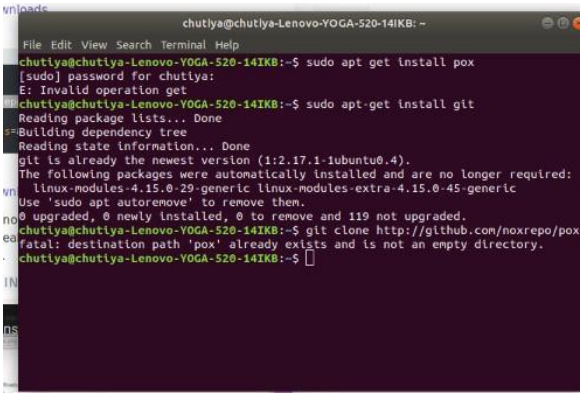

Fig. 5: Starting of pox controller

Applications require multiple point communication so that we can reduce the traffic in network. Many technologies have been used as a method to reduce the traffic. It is done in order to reduce the traffic and make the communication comfortable with the users and the customers. But the data is distributed in various routers. So there comes a lot of problem thus we have come up with a new algorithm that is dijkstra. In this we found the shortest path and thus we can transfer the data in shortest time.Since more number of devices are in connection with the internet and this number grows day by day. Thus it has become hard and bitter.Lot of data is transferred and from one router to the other router. And this architecture is becoming really complicated thus it increases the network congestion. We have used our algorithm and found the shortest path and hence have quite worked on this domain. There are many problems and we have come out with a solution to those problem of network congestion. Software defined networking helps to data layers devices and separates them from the network control . In this paper we have introduced the software defined networking and the research work with various methods that we can relate with SDN. We have also plotted the graph between latency and bandwidth and have shown that latency per unit of bandwidth. 1 mbps of bandwidth corresponds to 3000 PLT(ms). But this bar graph declines slowly as we have increased the bandwidth and after a certain period of time it reaches a stable situation where it maintains the stability of the system and it continues till infinity. We have one more graph of bandwidth and propagation time. It is the time required to send the data and the message from one point to other. As we have increased the bandwidth then we see that the graph varies with it a lot.

VI. DATA ANALYSIS

Data Analysis is done using wireshark. It is installed in ubuntu using the command sudo apt-get install wireshark. It shows the data transferred from packet to another packet through various protocols such as TCP, UDP,OVSK etc. It also shows the graph of rate at which the data is transferred and shows the amount of packet dropped and received by the server. It also shows the congestion in network and load in the network.

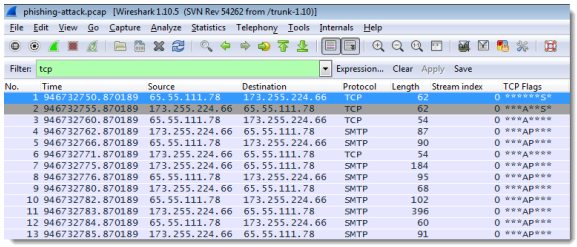

Fig. 6: Wireshark analysis

VII. POX INSTALLATION

The pox is installed by the command sudo apt get install pox.If u don't have the git package it should be installed first and then all other things should be done. U can install git by the command sudo apt get install git Photon is an interpreter which is a object-oriented language, suitable for many purposes. Photon has a clear, intuitive syntax, powerful structures having high-level data , and a flexible dynamic type system. Python could be used interactively, in stand-alone scripts, for larger programs, or as an extension language for existing application. The language can be run on Linux, Ubuntu, Macintosh, and Windows machines. There are also a number of system-specific extensions. Python has a large library of standard modules written in this language. As compared to C programs, python programs are much more shorter, and consequently can be written much faster.

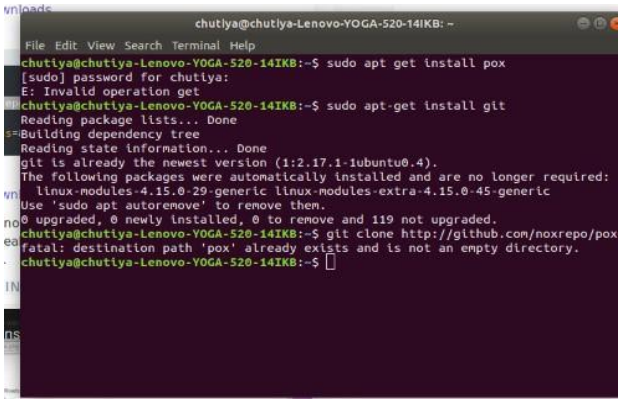

Fig. 7: Pox installation

VIII. FAT TREE TOPOLOGY

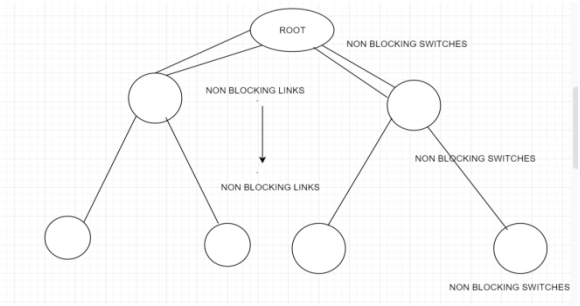

Fig. 8: Fat tree topology

Software Defined Networks the most commonly used network topology is the Fat-tree topology. Fat-tree topology is used in this work since the branches nearer the top of the hierarchy are thicker than branches further down the hierarchy. Since the branches are data links the varied thickness of the data links allows for more efficient and technology-specific use.

IX. GRAPHS OBTAINED

So the first graph that we have obtained is of latency and bandwidth it shows that when we increase the bandwidth the the latency decreases . Initially latency was more when bandwidth was 1 mbps but later it increases so latency decreases and after a certain amount of time it became constant.

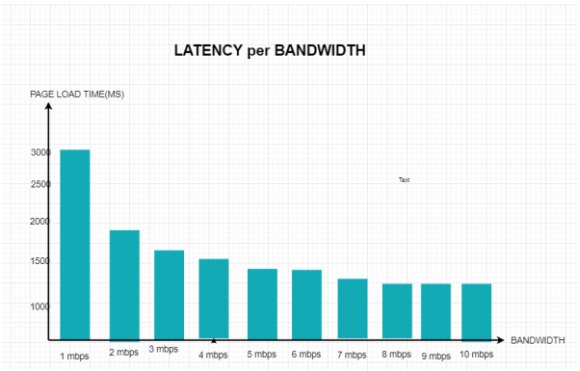

Fig. 9: Latency per bandwidth

The second graph that we have obtained is of bandwidth and propagation. We can see that there is a steep increase in the slope of graph as when bandwidth and propagation time increases. Since there are also many factors that have influenced the graph so it is not a complete straight line.

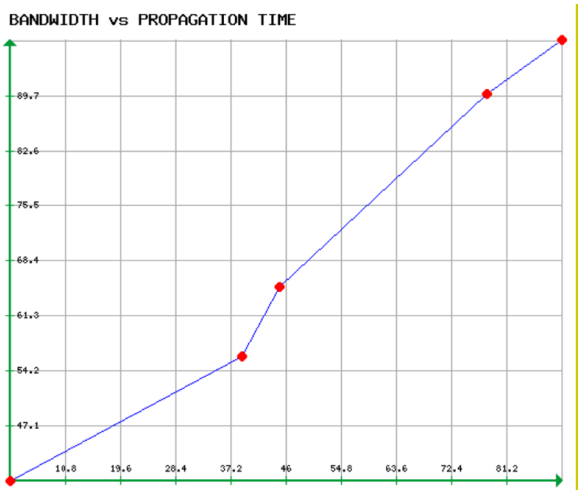

Fig. 10: Comparison of bandwidth with propagation time

Now let's describe the traditional networks that are used or we can say are relevant in the research of SDN.In traditional networks, networks are defined with OSI model. In OSI model we have 7 different layers and each layer uses many protocols. every layer is connected with the above layer and also with the below layer so that the communication can be done easily. But as a whole all the layers work independently. And they transfer data and information through them using a methodology called as API. It is a interface of programming. Since those 7 layers use various protocols so they can be replaced easily by those who have same configuration as them and offers as the service provided it is the same. A data of high layer is engulfed by the data in the lower level because a device in a particular layer only checks the header of that particular layer only and it doesn't checks for any other layer. When it goes to the other layer then it checks the configuration of that layer and match it with that of its. But while it checks for the configuration of that particular layer it does not disturbs the upper or lower layer. Those devices who are simple in nature only use the lowest layer of the OSI model but the devices which are complicated us the upper layers as well. So we conclude that the devices that work in the upper oe we can say in higher layers are more complicated than those at lower levels.

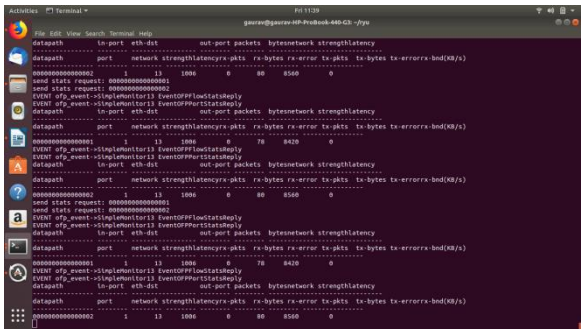

Fig. 11: Bandwidth and latency calculation in monitor file in ryu controller

When we tried to find out the simplest elements that active in current network then we found out that they are the layer 2 switches. We can guess the configuration from the name itself but let me elaborate it. They are the devices that are related with the layer 1 and then at the layer 2 switches only. They don't disturb the other higher levels and check the configuration of only those at those layers. They transfer the data between data link layer and VLAN . They don't transfer data anywhere else. These devices cannot be connected with multiple networks simultaneously. But we can say that switches are the device that are self learning. Routers are more mannered when it comes with forwarding the traffic among multiple networks. Routing tables are stored in the memory of routers which are made half manually and half automatically.

X. DEPLOYMENT MODULE

Here we have used a terminal so that we can issue commands to the operating system and in our case it is ubuntu. There is network congestion and link capacity in traffic streams. If here we give minimum bandwidth then the receiver will receive the required data

● Latency should be maximum. It is the total time taken for the packet to travel from the source to destination. Thus these delay should be minimized .

● Packet loss ratio should be maximum. When there is any type of network congestion then the packets will be delivered with a delay.We can rely on TCP for this however the packets that are dropped affect the congestion signal and hence result in delay of other packets.

● Jitter should be maximum. Jitter is the variance in the latency. Sometimes packet with different latency go in same traffic flow. This happens when network conditions change. This is not good and hence can affect the quality of the product

XI. IMPLEMENTATION

It was first used in a research paper by ONF. It is a basically new concept and architecture that helps in decoupling the control of network from actually forwarding the data in the network. It helps in centralizing the control to a particular or we can say single network and finds out the infrastructure lying underneath. When we come at the bottom of this type of SDN architecture that is the infrastructure layer so that consists SDN network devices in other words SDN switches. In traditional network there were layer 2 switches that were concerned with the data transfer with only those two layer. Well , here is nothing like that because SDN believes on a flow table made by controller and its uses a switch that has been installed using a API that is shared. The flow table is bit more complex than the traditional method of forwarding of data and the packets that are sent can be divided bad sub divided into various flows that are based on information that is obtained from OSI layer 2 to OSI layer 4. Our final result is shown below. And with our research we calculated that

- Creating switches, network topology and calculated out bandwidth
- Created out hosts and bridges
- We created number of source host and destination host and inserted number of packets , then we have split the packets by using the algorithm and calculated out threshold bandwidth, latency, size of packet and the propagation time.
-

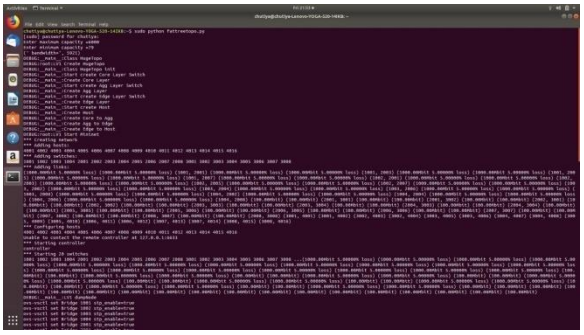

Fig. 12: Creation topology switch establishing controller bridges

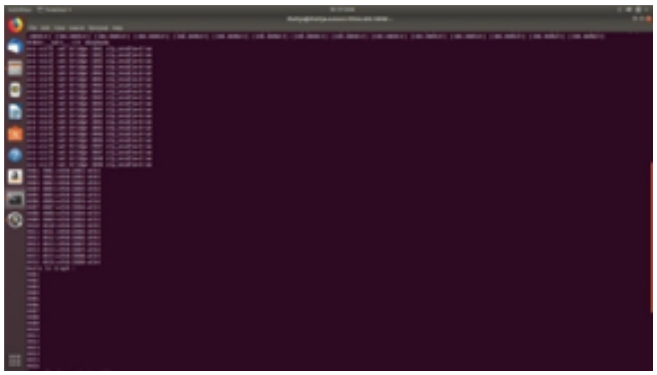

Fig. 13: Creation host

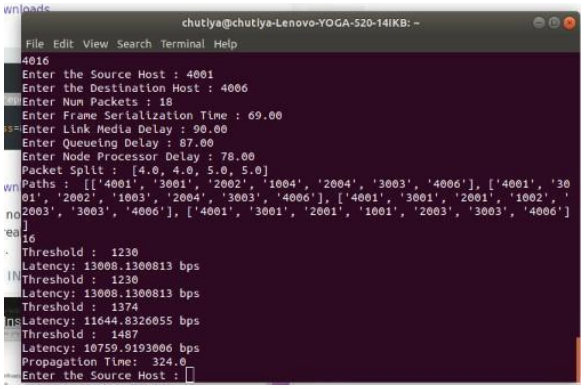

Fig. 14: Result calculation of bandwidth latency in pox controller with our implementation model

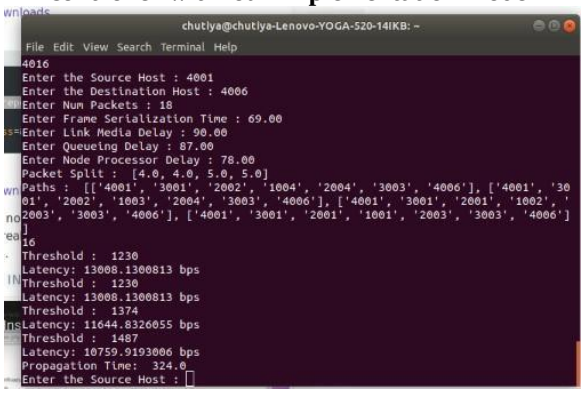

Fig. 15: Data division is shown along with the path of dataflow to calculate the parameters

XII. RESULT

The simulation results show that the proposed algorithm is better than the two naive algorithms in the term of end-to-end latency, as shown below.

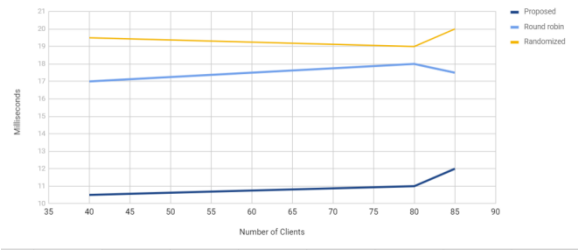

Fig. 16: End-to-End Latency

We have also compared the Dijkstra algorithm with round-robin and randomized algorithm that does not consider the nearest available server. The simulation results show that the Dijkstra algorithm is better than the round robin algorithms in the term of end-to-end latency. The superiority is because of the shortest paths (nearest server) consideration and also the congestion control.

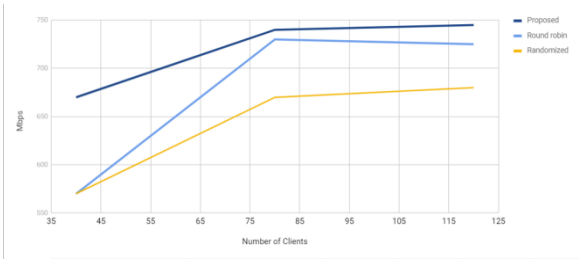

Fig. 17: Throughput

To evaluate the capability of the proposed algorithm, measurement and comparison of throughput was conducted. Throughput is defined as the rate of successful messages delivered over a communication channel. Figure shows that the proposed algorithm has higher throughput than the round robin and randomized algorithm. The round robin and randomized algorithm may deflect request to the far server and even the link is congested. It causes the throughput is lower than the proposed algorithm.

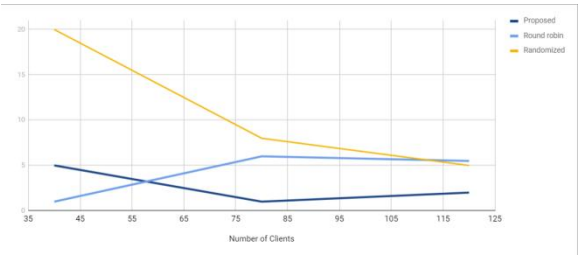

Fig. 18: Standard Deviation Load of Servers

Standard deviation is used to measure the server's load variation. Standard deviation is used to measures the amount of dispersion or variation from the average value. In this experiment, the proposed algorithm shows smallest result (good) in the server load variation.

XIII. CONCLUSION

We have analysed the traffic and have suggested various methods to overcome those . mininet tool is used with RYU controller. Bandwidth has been minimized and latency has been maximized . jitter is also maximized. The network has been analysed through wireshark and graph has been plotted. This paper has proposed Dijkstra's algorithm which is a load-balancing algorithm that takes advantage of the shortest path. To find the nearest server for a requesting client, Dijkstra's shortest path algorithm was used. The Dijkstra's algorithm considers the edge weights and the node weights for a graph derived from the underlying SDN topology. We firstly tried to test the results in linear topology in sdn setup with ryu controller with which we were able to find results on different parameters as represented above.Then we started to work on pox with mininet to establish different algorithm setup to find efficiency and find the improvement possible setup was based on mininet controller where the host switches were made connected then data flow was done and analysed in fields of latency bandwidth and throughput did work on dijkstra round robin algorithm and obtained the results which shows that due to Dijkstra shortest paths and logical weighted graph factor dijkstra certain has an edge over round robin in field of latency and bandwidth in the present environment simulated and represented above.And In field it is much time and cost saving in case of logical work 0of Dijkstra rather than randomized algorithms.

REFERENCES

1. Nunes B., Marc Mendonca X. Nguyen, Katia Obraczka and Thierry Turletti. 2014. A survey of software defined networking: Past, present, and future of programmable networks. 1-18.
2. Kreutz D., Ramos F.M., Verissimo P.E., Rothenberg C.E., Azodolmolky S. and Uhlig S. 2015. Software Defined networking: A comprehensive survey. Proceedings of the IEEE. 103(1): 14-76.
3. Zhou W., Li L., Luo M. and Chou W. 2014, May. REST API design patterns for SDN northbound API. In Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on (pp. 358-365). IEEE.
4. Kaur S., Kumar K., Singh J. and Ghumman N.S. 2015, March. Round-robin based load balancing in Software Defined Networking. In Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on (pp. 2136-2139). IEEE.
5. Yahya W., Basuki A. and Jiang J.R. 2015. The Extended Dijkstra's-based Load Balancing for OpenFlow Network. International Journal of Electrical and Computer Engineering. 5(2): 289.
6. Wang L. and Lu G. 2016, January. The dynamic sub topology load balancing algorithm for data center networks. In Information Networking (ICOIN), 2016 International Conference on (pp. 268-273). IEEE.
7. ONOS:https://wiki.onosproject.org/display/ONOS/Wiki+Ho me.
8. Mininet: http://mininet.org/.
9. Yang Liu, Jogesh K. Muppala, Malathi Veeraraghavan. 2014. A Survey of Data Center Network Architectures.

AUTHOR(S) BIOGRAPHY

Gaurav Tiwari, Btech Department of Computer Science and Engineering, SRM University

V Deeban Chakaravarthy, Assistant Professor (Sr.G) , Department of Computer Science and Engineering, SRM University

Aditi Rai, Btech Department of Computer Science And Engineering, SRM University