Load Balancing using SDN Methodology

Kavana H M [1] Kavya V B [2]
B. V. B College of Engg & Tech
Hubli, Karnataka

Madhura B [3] Neha Kamat [4]
B. V. B College of Engg & Tech
Hubli, Karnataka

***Abstract:* Software-Defined Networking (SDN) is an emerging technology in the field of networking. Nowadays clients are using large amount of data which needs to be handled by the network which creates a lot of traffic. For a single server it becomes difficult to handle all the load. The solution to this i to use multiple servers. The requests are sent to the load balancer. The client requests are then forwarded to the servers depending on the load balancing strategy used. Earlier hardware was used which turned out to be expensive and inefficiency. Traditional load balancer are vendor locked, nonprogrammable because network administrators cannot create their own algorithms. On the other hand SDN load balancers are programmable and allow you to design and implement your own load balancing strategy. Additional use of SDN load balancer is it does not need dedicated hardware. In this paper we are implementing and comparing our algorithm using Floodlight controller ,Mininet and testing our results via Wireshark Network Analysis Tool.**

Keywords: Software Defined Networking, Mininet, Floodlight, Load Balancing, Wireshark.

I. INTRODUCTION

There are several limitations in the traditional networking methods which required to be overcome to meet today's reqirements. Software Defined Network (SDN) is a major change in the field of networking. It has witnessed a major evaluation in the network world. In SDN data plane and control plane are separated.

Traditional methods consisted of tight coupling between data and control plane The data planes of SDN are cheap commodity silicon devices. OpenFlow protocol is used by the control plane to communicate with data plane. Network applications such as switching, routing, firewall and load balancer which run on the top of control plane. The most common protocol used in SDN networks which helps to communicate the controller with all the network elements (NE) is Open flow. It is an open standard that provides a standardized hook to allow researchers to run experiments, without requiring vendors to expose the internal workings of their network devices. The SDN is the architecture dividing the layers whereas Open Flow is just a protocol proposed to convey the messages from the control layer to the network elements. To increase the bandwidth, throughput and for better working load balancing must be used in networking. Load balancing is the ability to balance the load across several internet connections. The load balancing capability helps to balance the load across various sessions like web, email etc. To increase the total amount of bandwidth available it spreads out the bandwidth used by each user.

Nowadays we require dynamic network management resources for higher performance and high speed for the data transmission. Our method of load balancing using SDN uses a dynamic load balancing algorithm implemented in the SDN controller. The algorithm distributes the upcoming and incoming traffic flows. It helps in achieving the best possible resource utilization of each of the links present in a network.

We have used the Floodlight controller as the SDN controller, and the network is emulated using Mininet software along with wireshark to assess the performance. The aim of the project is to implement dynamic load balancing using SDN data centres for achieving better results and higher performance. To evaluate and validate the functionality of the proposed algorithm is our objective.

A. SDN

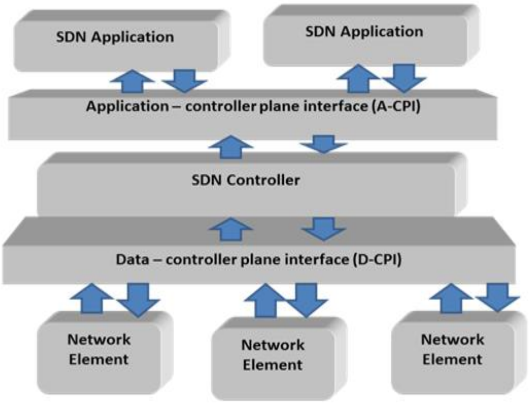
Software defined networking is a new upcoming architecture where it is easily manageable and very cost effective which is useful in today's complex applications. In Software defined networking there is a separation of control plane and the data forwarding plane. [1] Here the controller plane job is done by the controller. It performs various functions like finding routes, minimizing cost and also scale up to the increased workload. The network is centralized in software defined networking so the whole network appears to the application as a single logical switch.

SDN architecture also help in supporting a set of API's which includes routing, security, access control ,finding bandwidth and traffic management. SDN are being controlled by SDN controllers and the SDN applications not by the network consoles or commands which requires a lot of overhead. With using SDN the enterprise will be able to look after the whole network from a single logical point which improves the scalability and functionality of the network. And here there is no need of understanding and processing the protocols but just to follow the instructions given by SDN controller. Open flow is a interface designed for SDN where it is completely centralized and has a good performance, traffic control from multiple sellers. Open flow also helps in improving network reliability and security as there is a centralized management of network devices.

Even end users using the application have a better experience. Open flow has many advantages like there is no need to equip costly load balancing devices where the single load balancing switch or the controller controls the whole workload of the network. It also helps to combine both the packet and circuit networks together to decrease the CAPEX/OPEX of the enterprises. The unused switches or the unused links can be switched off which saves a lot of energy.

Open flow also helps in easy control of network and helps in saving resources. The traditional networks has some limitations as the quality of service and security of the network devices has to be configured manually where as in SDN there are a set of common API's which abstracts the networking details. The workloads from IT change frequently but the traditional network cannot fully satisfy its needs because the IT may demand for increase in bandwidth but by using traditional network to change the bandwidth capacity there has to be increase in end points and services which may lead to redesign of network which may become expensive and complex. But in case of SDN there can be innovation and it helps to change the network capacities and services without need to configure individual devices. SDN also provides the common programming environment which helps enterprises, operators and sellers good programmability and good revenues. So using SDN has more advantages compared to using traditional networks.

II . SYSTEM MODEL


SDN Application
SDN Application
Application – controller plane interface (A-CPI)
SDN Controller
Data – controller plane interface (D-CPI)
Network Element
Network Element
Network Element

The above figure shows the SDN components. There are three layers infrastructure, control and application layer. The infrastructure layer consists of network elements and it interacts with controller plane using controller plane interface. SDN application interacts with controller via Application controller plane interface. SDN controller translates the network requirements to network elements through controller.

III . PREVIOUS WORK

A load balancing mechanism using round robin method. In this method the packets are sent in a circular fashion. The switches are arranged in a circular fashion and if there are five switches in the network then first the packet is sent to switch one and this continues till switch five, if all the packets did not reach the destination then the remaining packets are again sent from switch one and it continues.

IV. PROPOSED METHODOLOGY

We have used the software tool mininet and floodlight controller and our basic aim is to achieve the efficient load balancing using SDN. So by using the above described software tools we have implemented the load balancing. We have used the shortest path first and to implement is we have used python language.

A. The programming language used: Python

In our project we have used python in mininet to convert the algorithm into code for balancing the load.It is a powerful high level language which can be used in any of the projects.

We have used python programming language so that it will be easy to debug the code and so that it should be well understood by the people. This language is supported in linux, windows.

B. The algorithm used: Shortest path First

1)Find information about hosts connected.
2)Using shortest route concept finding the information.
3) Find total link cost for all the routes.
4) Get current transmission rate.
5) Selecting the best path.
6) Push the traffic into each switch in the current best path and go to step 2.

C. Working

First we have to get the information of the hosts connected. once its done then we have to use the shortest path concept .

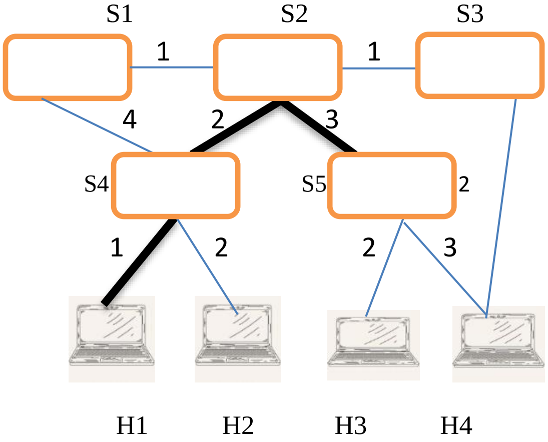


Fig1 : Data center network topology

In the fig we can see that there are 5 switches and four hosts and each link is assigned certain cost.H3 wants to send packets to h1. So according to the code to balance the load it should choose that link which has the least cost, so if packet has to go to host 1 then it will go according to the path highlighted. This will be done for all the hosts. Here the total link cost is taken for all the paths between the hosts and then according to the path the best path is selected this all is done by the floodlight controller, it controls the traffic. This is how there will be no traffic in sending the packets and hence load is balanced.

V. SIMULATION/EXPERIMENTAL RESULTS

A.n Before load balancing and after load balancing

Transferring(Gbytes)	Bandwidth(Gbits)
15.7	13.5
21.9	18.8
24.6	21.1
22.3	19.1
39.8	34.2
Average =24.86	Average=21.34

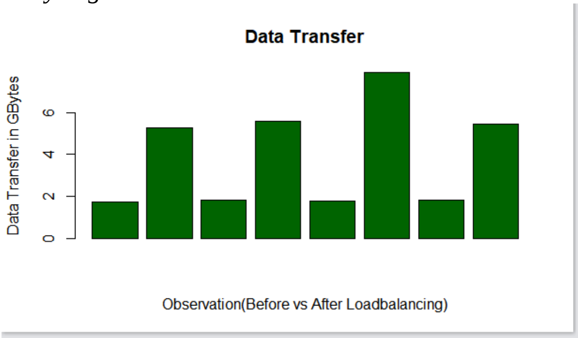
Table1:Before load balancing

Transferring(Gbytes)	Bandwidth(Gbits)
38.2	32.8
27.6	32.3
40.5	34.8
40.8	35.1
16.5	14.2
Average=32.72	Average=29.84

Table2:After load balancing

This is calculated for the transferring of packets from host3 to host1.Likeewise we can calculate of other hosts as well and can check the average .from table its clear that after load balancing the transfer of packets is more and precise.

B. Analyzing the result



VI. CONCLUSION

Though there are many ways in transferring the packets to different destinations but software defined network is the efficient one as it requires least hardware. Balancing the load in the network is very important in today’s huge crowed so its necessary to pass the packets in a no time.

VII. FUTURE WORK

In future it can be further extended using different and more efficient load balancing algorithms and even can connect still more controllers in circular fashion

VIII. REFERENCES

[1] Chen, Yu-Jia, Yi-Hsin Shen, and Li-Chun Wang. "Traffic-Aware Load Balancing for M2M Networks Using SDN." Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on. IEEE, 2014.

[2] Koryachko, Vyacheslav, Dmitry Perepelkin, and Vladimir Byshov. "Approach of dynamic load balancing in software defined networks with QoS." Embedded Computing (MECO), 2017 6th Mediterranean Conference on. IEEE, 2017.

[3] Chen, Kun-Ting, Chien Chen, and Po-Hsiang Wang. "Network aware load-balancing via parallel VM migration for data centers." Computer Communication and Networks (ICCCN), 2014 23rd International Conference on. IEEE, 2014.

[4] Ye, Kejiang, et al. "Live migration of multiple virtual machines with resource reservation in cloud computing environments." Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.

[5] Tsygankov, Mykola, and Chien Chen. "Network aware VM load balancing in cloud data centers using SDN." Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on. IEEE, 2017

[6] Bhandarkar, Smriti, and Kotla Amjath Khan. "Load Balancing in Software-defined Network (SDN) Based on Traffic Volume." Advances in Computer Science and Information Technology (ACSIT) Print ISSN (2015): 2393-9907.

[7] Al-Najjar, Anees, Siamak Layeghy, and Marius Portmann. "Pushing SDN to the end-host, network load balancing using OpenFlow." Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International onference on. IEEE, 2016.

[8] Zhang, Junjie, et al. "Load balancing for multiple traffic matrices using SDN hybrid routing." High Performance Switching and Routing (HPSR), 2014 IEEE 15th International Conference on. IEEE, 2014.

[9] Yu, Jinke, et al. "A load balancing mechanism for multiple SDN controllers based on load informing strategy." Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE, 2016.