International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 06, June 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

INTRODUCTION TO LOAD BALANCING AND STRATEGIES USED IN SOFTWARE DEFINED NETWORKING

[1]Sumit Badotra, [2]Japinder Singh

[1] ,[2]Department of Computer Science and Engineering

[1] ,[2]Shaheed Bhagat Singh State Technical Campus, Ferozepur, Punjab(India)

ABSTRACT

In today's network world as more and more number of clients are attaching with the network and to handle each and everyone's request has become a tedious job to a server. For handling such huge load it is impossible for a single server. In traditional networks there was vendor specific dedicated hardware that used to handle these huge loads of requests from multiple number of clients, client sends the request to load balancer and it forward the requests to the different servers using some load balancer strategy but these vendor locked specific hardware are very costly and inflexible, non programmable to use. In overcome these problems Software Defined Networking has come into act and serve all the problems. The basic idea in this is the separation of control plane and data plane (control plane constitutes the intelligence of the network and this is the place where the controller resides whereas data plane contains the switches and other networking devices, basically the infrastructure used in networking).Networks are now programmable and hence applications like load balancer are no longer be dependent upon dedicated hardware a application of load balancer can be made to run with controller association and OpenFlow switches in the network can behave as a load balancer we don't need to buy any dedicated hardware for specific applications. In this paper different load balancer strategies which are used in load balancer application is provided.

Keywords - Software Defined Networking, OpenFlow, Load balancing, Network Function Virtualization, Open Network Foundation.

I. INTRODUCTION

Technology of traditional network is very complicated and difficult to administer. The devices which are helpful in providing network services are vendor locked in and used on dedicated hardware only. The configuration of each and every network device is the responsibility of network administrator. All the networking devices are not from same vendor so it's even more difficult to manage such diverse different products of different vendors [1]. Manual configuration is done in traditional networks and they are very inflexible to change, and require high operational expenditure for running the network. With the advancement of new technology especially in the

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

areas of mobile, social, and big data centers, high bandwidth and dynamic management of computer networks is required. These growing networks of computers mainly require changes in exciting computer network so that the complexity of handling it can be managed. Hence new emerging technology in this field has come called as Software-Defined Networking (SDN) where network control is separated from forwarding plane (data plane) and is programmable directly, hence eliminate the need of network administer [1].

SDN model consists of infrastructure layer which consists of switching device. To collect network status, temporary stored them in local devices and forward them to controller is the main aim of this layer [1].Control layer is the next layer in SDN architecture and it acts as an interface between infrastructure layer and application layer.

The application layer consists of SDN application that is designed to fulfill requirement of user. This layer is residing above the control layer. Protocol used for the purpose of communication in between control plane and data plane is called as OpenFlow protocol. The logic of control plane is implemented in the software called controller. [2] The decision of sending packets to where and how to make forwarding is all residing in control plane. Flow tables are constituted in OpenFlow they may be one or two in number. Packet handling and switch rules are stored in these flow tables. Figure 1 highlights the SDN architecture.

In every flow based routing, controller independently works and sets up. Exact match is performed for flow entries. It is a requirement for fine grain control. One flow entry can covers a large group of flows in aggregated flow routing. Each category of flows contains one flow entry. Wildcard flow entries are used by it and aggregated on flow based routing [2].

The OpenFlow specification describes the information exchange between the two planes. In this architecture, an OpenFlow contains a flow table consisting of flow table entries.

Due to increase in the rise of networking traffic load heavily and growing very rapidly day by day there is an increase in server overloads and network congestion. Multiple servers to offer enhanced capacity and high reliability requirement is to be achieved in Online services such as web sites, search engines and all social networking sites [2].To distribute requests to multiple servers and load among them load balancer is used. Traditional load balancers are implemented on dedicated hardware, very costly, non-programmable.

Load balancing is a method used for spreading requests across multiple services. This is done using highly expensive vendor specific hardware devices. In enhancing performance of networks by the use of proper utilization of available resources is done by Load balancer.

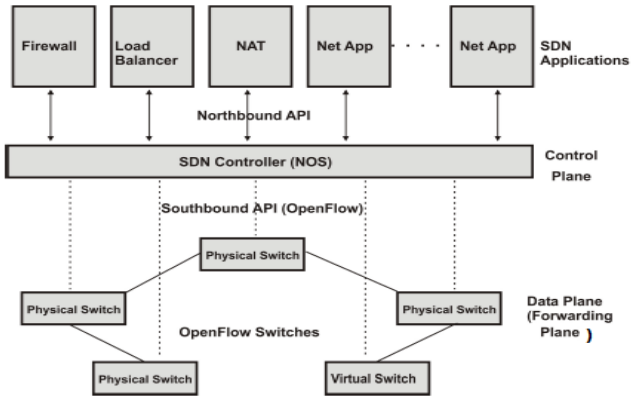

Figure1 architecture of SDN

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 06, June 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

II. LOAD BALANCING

Helping to save power and improve resource utilization in network is all done by Load balancing stratergy.Traditional load balancing technique requires dedicated hardware support which is expensive, lacks of flexibility and is easy to become a single point failure [3].But with the help of SDN load balancing strategies no more dependency is there on dedicated hardware as any switch can behave as a load balancer. Now handling of all heavy traffic loads is easy and manageable.fig.2 describes the architecture for load balancer,s1 is the switch acting as load balancer and connected to h2,h3,h4 hosts h1 is serving as client. Load requests are handled by the load balancer using a corresponding strategy.

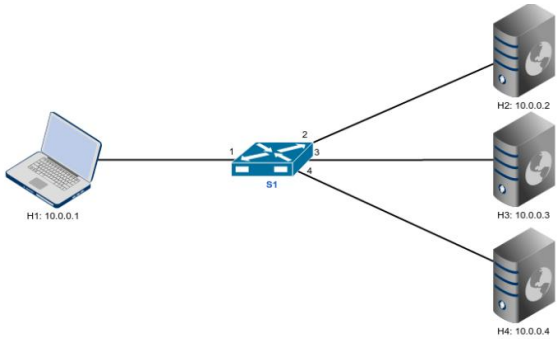

Figure 2 load balancing architecture

III. ARCHITECTURE OF SDN BASED LOAD BALANCER

The architecture of Load balancing is consisted of OpenFlow switch network with a controller and multiple servers connected to the ports of the OpenFlow switch. Each server is assigned static IP address and the controller maintains a list of live servers that are connected to the OpenFlow switch. Web service is running on each server on a well known port 80 [3].

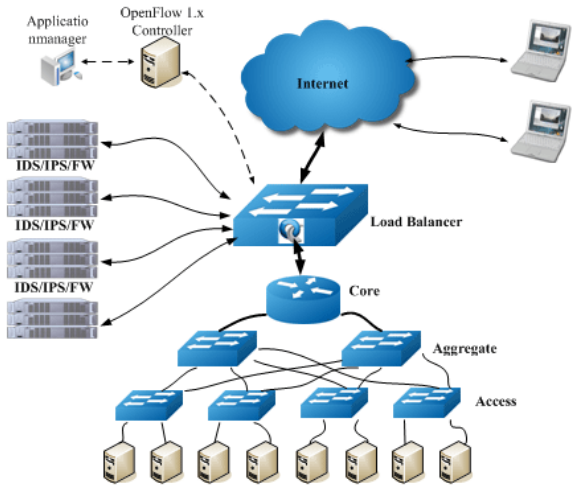

Figure 3 SDN based load balancer

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Controller contains a virtual address. All requests from the clients are sent to the virtual IP address. When the client sends a request packet to the virtual IP, OpenFlow switch uses the information contained in packet header and compare it with flow entries in switch, if the client packet header information matches up with the flow entry, then switch modifies the destination virtual Ip address to the address of one of the servers based on load balancing strategy and forward the packet to that server, If packet does not matches with any flow entry, the switch will forward this packet to Controller. The Controller will decide how to deal with the packet. The Controller adds flow entries to the switch through using OpenFlow [3].

IV. TECHNICAL BACKGROUND

SDN Controller: Five topmost open source controllers which are widely used are: POX, Ryu, Trema, Floodlight, and OpenDaylight. For Proper understanding and realization of the SDN concept we have to choose a suitable controller. The vital element of SDN network is considered to be its controller. It is a platform which manages the flow of control to the routers and switches via Southbound OpenFlow protocol and applications via Northbound APIs [4] .A collection of Pluggable modules is contained by controller which performs different network tasks.Fig.4 describes all about the most used SDN controllers.

	POX	Ryu	Trema	FloodLight	OpenDaylight
Interfaces	SB (OpenFlow)	SB (OpenFlow) +SB Management (OVSDB JSON)	SB (OpenFlow)	SB (OpenFlow) NB (Java & REST)	SB (OpenFlow & Others SB Protocols) NB (REST & Java RPC)
Virtualization	Mininet & Open vSwitch	Mininet & Open vSwitch	Built-in Emulation Virtual Tool	Mininet & Open vSwitch	Mininet & Open vSwitch
GUI	Yes	Yes (Initial Phase)	No	Web UI (Using REST)	Yes
REST API	No	Yes (For SB Interface only)	No	Yes	Yes
Productivity	Medium	Medium	High	Medium	Medium
Open Source	Yes	Yes	Yes	Yes	Yes
Documentation	Poor	Medium	Medium	Good	Medium
Language Support	Python	Python-Specific + Message Passing Reference	C/Ruby	Java + Any language that uses REST	Java
Modularity	Medium	Medium	Medium	High	High
Platform Support	Linux, Mac OS, and Windows	Most Supported on Linux	Linux Only	Linux, Mac & Windows	Linux
TLS Support	Yes	Yes	Yes	Yes	Yes
Age	1 year	1 year	2 years	2 years	2 Month
OpenFlow Support	OF v1.0	OF v1.0 v2.0 v3.0 & Nicira Extensions	OF v1.0	OF v1.0	OF v1.0
OpenStack Networking (Quantum)	NO	Strong	Weak	Medium	Medium

Figure 4 SDN controllers

Working with mininet: Emulation of the real world network is done with the help of Mininet. By giving a single command and is all done by Mininet. For developing, sharing, and experimental Process with OpenFlow and Software-Defined Networking systems Mininet plays a very vital role [5].

The figure 5 shows the creation of the custom topology. Within a single machine emulation is done by mininet with an OpenFlow network and end-hosts are connected within the network. The built-in support to create several common topologies (default and custom) is a great feature of it. The construction of custom topologies is done using a python script. The comfort, advantage and authenticity are provided by the mininet at very low cost [5].

The substitute to Mininet is hardware test beds which are agile, exact but due expensive and shared nature cannot be used for all experiments. Mininet is available freely and it is open source software which emulates the OpenFlow devices and SDN controllers as well. Simulation of SDN networks, a controller (POX, Ryu, OpenDaylight etc.) for experiment is run by mininet [5].Emulation of real world network scenarios is provided by it and few of SDN controllers are by default included with in Mininet VM,for using some other controllers

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

(as per the need) there installation is required. With the help of Mininet ease usability, scalability in the network and accuracy in performance can be achieved

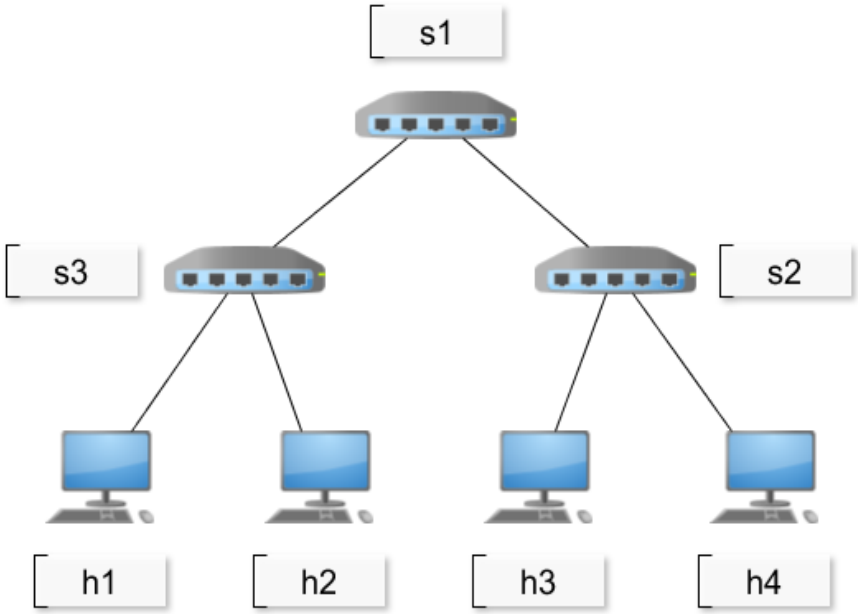

Figure 5 Mininet Topology of Tree

Tree: To per switch (s1, s2 ….s8) this topology contains *m* levels and 2 hosts (H1, H2….H8) are attached

Open Flow switch: An OpenFlow-compliant switch is a network's basic forwarding device. According to its flow table it forwards the packet to its destination [5]. The flow table of a switch contains a set of flow table entries, in which each one is consists of match fields, counters and instructions, as illustrated in Figure 6.Flow table entries are also called flow rule or flow entries.

The **header fields** in a flow table are the first entry which describes that which packets this entry is applicable in for. Constitution of a wildcard-capable match over specified header fields of packets is included in it [5]. For faster packet forwarding with the help of OpenFlow, ternary content addressable memory (TCAM) is required by the switch that allows the fast lookup of wildcard matches. Depending on the OpenFlow specification, e.g., Ethernet, IPv4, IPv6 or MPLS [5]. The header fields can match different the above stated protocols.

Header Fields	Counters	Actions

Figure 6 Flow Table of OpenFlow Switch

Collection of statistics about flows is reserved by the counters. The number of received packets and bytes, as well as the duration of the flow is stored by these **counters** only [5].

How the packets of that flow are handled is decided by the **action**. Some common actions are ―forward drop, ―modify field, etc. [5].

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

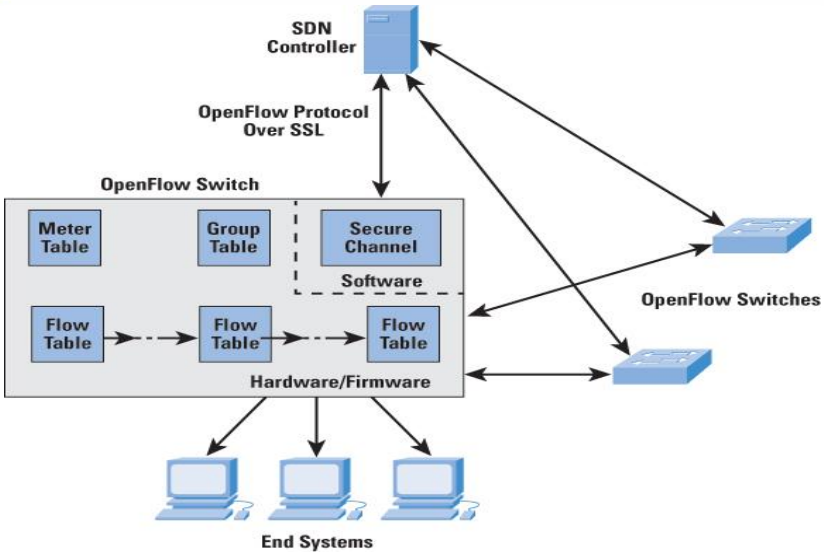

Figure 7 Working with OpenFlow Switches

V. LOAD BALANCER STRATERGIES

To effectively schedule the routing of requests from a client to the respective servers in an optimized way, several load balancing strategy are used that are as follows:

5.1 RANDOM STRATEGY: From a list of live servers, the Load Balancer will randomly choose a server for sending request. This policy has large overheads.

5.2 ROUND-ROBIN: In this, each server receive the request from clients in circular manner. The requests are allocated to various live servers on round robin base.

5.3 WEIGHTED ROUND-ROBIN: In this, each server receive the request from the client based on criteria that are fixed by the site administrator. In other world A Static weight is assigned to each server in Weighted Round Robin (WRR) policy [6].We usually specify weights in proportion to actual capacities. So, for example, if Server 1's capacity is 5 times more than Server 2's, then we can assign a weight of 5 to server 1 and weight of 1 to server 2 [6].

VI. CONCLUSION AND FUTURE SCOPE

SDN load balancer solves many problems of traditional load balancers because traditional load balancer use dedicated hardware. That hardware is expensive and inflexible. But in SDN, OpenFlow device is converted into powerful load balancer by programming the SDN controller. As is many cases with the help of most commercial load balancer, load balancer can also be single point of failure. To eliminate this problem in future one can use multiple controllers instead of single controller. In case of one controller failing, another machine will take over the role of controller and continue routing traffic

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

VII. ACKNOWLEDGEMENTS

For the constant blessings of the almighty god we are very thankful to him. We would like to dedicate our gratitude towards parents, family, friends and all the teachers and staff members of department of Computer Science and Engineering of Shaheed Bhagat Singh State Technical Campus for providing their constant support and encouragement in the development of this paper and in essence, all sentient one beings.

REFERENCES

[1] Jammal, M.; Singh, T.; Shami, A.; Asal, R.; Li, Y.,"Software defined networking: State of the art and research challenges" *Elsevier computer Networks* 72(2014)74-98

[2] Doria, A., Salim, J. H., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal (2010). "Forwarding and control element separation (forces) protocol specification". Internet requests for comments, RFC editor.

[3] Koerner (2012). "Multiple service load-balancing with openflow"., 2012 IEEE 13th international conference on In high performance switching and routing, pages 210-214,IEEE.

[4]Sumit Badotra and Japinder Singh, "Software Defined Networking: An Innovation through Transformation" in National Conference on Computing, communication and Electrical systems (NCCCES-2016)

[5] Sumit Badotra, Japinder Singh." A Review Paper on Software Defined Networking", in Proceedings of International Journal of Advanced Research in Computer Science", 2017

[6] Sabiya, Japinder Singh. "Weighted load balancing using software defined networking". International Journal of Advanced Research in Computer Science and Software Engineering, 2016.