Development of a Multilayered Aggregate Switch Architecture for DCN Using Software Defined Networking

Abhilash Reddy Naredla, Jahnavi Tejomurtula, Abhash Malviya, Ashmita Chakraborty
Department of Electrical Engineering
San Jose State University
San Jose, USA

abhilashreddy.naredla@sjsu.edu, jahnavi.tejomurtula@sjsu.edu, abhash.malviya@sjsu.edu, ashmitaparthakamal.chakraborty@sjsu.edu

***Abstract*— Applications today have evolved from single, multiple servers, to being expansively distributed across racks of servers to dynamic models that span across multiple data centers. Further with the expansion of users from business managers to complete organizations including the customers having access to these applications across multiple devices, the stress on existing network infrastructure has increased causing data centers to reach the tipping point. Moreover in all the conventional data centers control, management and data planes reside within the single device making it extremely difficult to configure all the devices in the network individually. This demands for highly agile and flexible data centers, which can be scaled and centrally managed effectively making use emerging technologies like SDN, to save cost and resources. In this paper we propose to study the complexity and feasibility of Data Center Networks expansion scaling them horizontally and vertically. In our proposed method we are creating Fat-tree and multi-tiered data center network topologies with more than one aggregate levels and also increasing the number of edge layer switches. The implementation is done on mininet and by making use of the learning switches we centrally manage the network using SDN controllers. Further we implemented performance tests comparing the efficiency of the topologies at various levels when scaled.**

Keywords—Datacenter; SDN; scaling; performance; Fat-Tree; Multi-tiered

I. INTRODUCTION

The primary requirement of most organizations is business continuity; if there is a system disruption, IT operations become impaired which can impact availability of services to customers. To minimize any chances of disruption, availability of reliable infrastructure is a must. Almost a third of all the IT- related spending is used for data center [1]. Communication inside the datacenter is based on IP protocol based network consisting of routers and switches that transfer traffic between the internal servers to the outside world. Some servers are often used for hosting intranet and other services required for internal users such as email servers, DNS, DHCP and proxy servers. A data center has to be optimized to balance the workload and provide efficient results to the network. Data center network performance can characteristically be illustrated using well-accredited metrics such as bandwidth, reliability, throughput, power consumption, latency and cost.

With the introduction of software-defined networks, the control plane and data plane are separated from each other, with the centralization of state and network intelligence, and abstraction of the underlying network infrastructure from the applications. By separating the control logic out from the switch we are able to use a centralized controller that can view and control the network and routing. SDN gives us a more programmable and customize freedom to the network. It also leads to creation of open interfaces between the control and data plane devices. The most deployed SDN protocol is OpenFlow [2], which allows setting into OF-compliant switches forwarding rules established by centralized SDN controller. A SDN controller like POX [3], Opendaylight [4] etc., allows to redefine and reconfigure network functions with scalability and better use of network resources. We have designed the topologies for Fat tree [5] and Multi-tier DCN with layering of aggregate switches. Layering of the aggregate switches has been done horizontally and vertically for both the fat-tree and multi-tier topology to understand and effectively compare the DCN topologies for communication between the hosts in same rack and communication between the hosts in different racks. Then a comparison between the performance of the fat-tree and multi-tiered architecture was done to test out the better-suited topology for scaling. The aggregation layer is also the connection point for data center firewalls and other services. Thus, it consolidates L2 traffic in a high-speed packet switching fabric and provides a platform for network-based services at the interface between L2 and L3 in the data center. Mininet [6] was used as the network emulator. It uses virtualization mechanisms to create software-defined network, which can be customize and interacted with effectively.

The rest of this paper is organized as follows: section II contains description of related work on data center networking. Section III provides a detail of our architecture and its functionality and section IV presents a performance analysis of the designed architectures. Finally section V concludes the paper.

II. RELATED WORK

A Data Center is a pool of computational, storage and network devices that are connected over a communication channel. It is a medium that interconnects data center resources together. It also enables the connection of multiple host servers to a single system by virtualizing them. Virtualization makes the devices more scalable and mutable without affecting the physical structure. Each of the data center has its own network to connect itself and its components to the Internet. The Data Center Network plays a vital role since they handle the load balancing and the speed of communication.

DCN architecture modeling is viewed as a standout amongst the most vital determinants of system execution, and it plays a noteworthy part in network analysis. Data centers are classified into different architectures based on several factors that include cost, performance, scalability and end-user requirements. Following section will cover the description of Fat tree [5] and multi-tier topology [7] that is implemented in this paper and also the details of horizontal and vertical scaling.

A. *Fat-tree Architecture:*

Fat-tree is tree based network architecture that is usually built in 2-tier or 3-tier. The Fat tree topology is a modified instance of Clos topology in which the Ethernet switches are interconnected. The Servers are connected to the Top of Rack (TOR) switches, which are connected to the aggregate switches that are in turn connected to core switches. Each server block in a Fat-tree DCN is called a pod. The construction of the Fat Tree topology where k is the number of pods; consists of (k/2)^2 number of servers per rack, (k/2) number of TOR switches per pod, (k/2) number of aggregate Layer switches per pod, (k/2)^2 total number of core switches in the topology. Fat Tree provides a potential solution by replacing as many as high-end switches with low-end access level switches at lower level making them handle the major part of communication. By using fat tree topology the network is effectively tailored to utilize the available bandwidth in the communication.

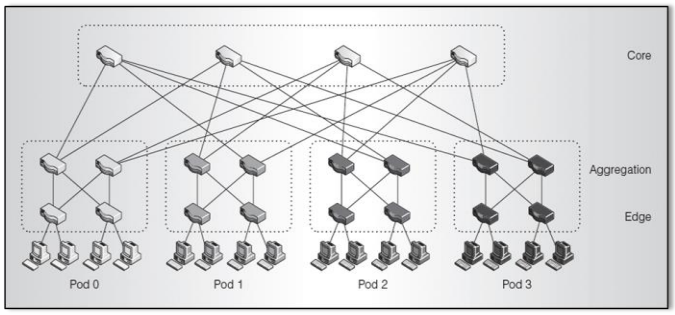

Fig. 1: Generalized Fat Tree Data Center Architecture

B. *Multi-tiered Architecture:*

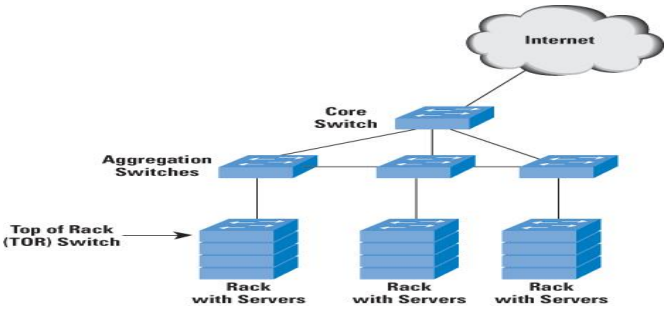

Fig. 2: Generalized multi-tiered architecture

A Multi-tiered architecture is a traditional data center for medium and large-scale industries. There are three layers in a multi-tiered architecture; Core, Aggregate and Access layers. The core layer consists of core switches at the root level, which are used for the communication outside the scope of the architecture. There is a load balancer at this level, which can take care of deploying the traffic as per needs and managing the load on the network. Aggregate layer has the aggregation switches at the mid-level which acts like an interface between the access layer switches and the core switches. They take care of the routing and forwarding of packets. Access level has the switches that are connected directly to the hosts. The major consideration in this architecture is the ratio of incoming traffic to the capacity the network links can handle. These factors are configurable by varying the number of up-links and downlinks for each access level and aggregate level switch.

C. *Vertical scaling:*

One way to maximize flexibility and increasing the design life cycle is to vertically scale the data center network. Vertical scaling is increasing the aggregate layers in the topologies. Scaling vertically will help us to reduce operation costs and increases several efficiencies. It also increases the life span of the data center, vertical scaling will give the data center the ability to grow twice power wise at half the costs of regular build of a data center [8]. It also proves to be highly efficient operation wise with high reliability. It reduces the cost per server-application by increasing the efficiency with increasing density. As demand changes flexibility becomes more important to provide more options supporting more projects.

D. *Horizonatal scaling:*

Scaling of a data center horizontally is scaling-out the data center, it is nothing but increasing the number of servers racks which increases oversubscription and therefore demands for the increase in the number of access layer switches. The interfaces between access layer switches and severs is not a distributed one like multiple aggregate layers so scaling horizontally requires high efficiency switches and routers. Scaling out proves to be a cheaper than scaling vertically but results in high utility costs.

III. NETWORK ARCHITECTURE

Data center is a distributed environment where multiple operations are carried out in a fraction of time. The switches and servers handle a large amount of computing load due to the running applications. In this situation, if there is any server/switch failure, the alternate servers compromise the delay in the system. For more redundancy, increasing the number of switches that handle operations scales the data center [9]. When a data center is scaled, the major change is contributed by server virtualization, which helps consolidating the load onto the newly added, more capable switches. This method is cost and performance efficient. However, there are major challenges in scaling the data center networks. One of them is the resource allocation. Even though there is many servers configured in the system and the resources are not allocated properly, the performance is at stake. Allocating resources to each machine is as important as scaling the network [9]. Failure of resource allocation leads to workload imbalance, loss of resources, server unable to host the needed number of virtual machines and even cause the entire server to crash. Any scaled system would become efficient when the multiple dependent factors are satisfied.

A. *Implemented fat-tree architecture design:*

The Fat tree topology is built using below values from the TABLE 1. The architecture was built for k=4, 5, 6, 7 i.e. 16, 35, 64 and 98 hosts in network. For all the above topologies the architecture was designed by scaling the layer of aggregate switches horizontally and vertically. The connections between any two layers of switches are made as one too many. In horizontal scaling there is only one layer of aggregate switches and for vertically scaling there are 2 layer of aggregate switches for (16, 35 and 64 hosts) and 3 layers of aggregate switches for (98 hosts). Each aggregate switch layer in the vertical scaling is connected to the above and below aggregate layer switches in the same pod. The topology diagram can be seen below.

TABLE 1: Fat-tree architecture design based host and switch count.

No. of Racks(K)	No. of Servers per Rack	No. of TOR Switches per Rack	No. of Aggregate Switches per Rack	Total No. of Core Switches	Total No. of Servers	Total No of TOR Switches	Total No. of Aggregate Switches	Layers for Aggregate Switch
HORIZONTAL								
4	4	2	4	2	16	4	8	1 Layer
5	7	3	6	7	35	15	15	1 Layer
6	8	4	4	16	64	32	32	1 Layer
7	14	4	4	12	98	28	28	1 Layer
VERTICAL								
4	4	2	4	2	16	4	8	2 Layers
5	7	3	6	7	35	15	30	2 Layers
6	8	4	4	16	64	32	32	2 Layers
7	14	4	6	12	98	28	42	3 Layers

B. *Implemented multi-tiered architecture deisgn:*

First thing we have taken into consideration is that for us to compare Multi-Tiered and Fat-Tree architectures it is to be taken care that the oversubscription ratio for both the architectures should be 1:1, whereas Fat-Tree with its K-pod structure makes 1:1 inherent. We design multi-tiered architecture to be having 1:1 oversubscription ratio. For the ratio to be 1:1 the bandwidth over the links all through the three levels of basic aggregate switch must be in the ratio e: 2e, which is the uplink to down link band width provision to be in the ratio 1:2. Unlike Fat-Tree architecture, the Multi-Tiered architecture has a flexible algorithmic structure.

Where K is the number of servers per rack

Number of Top of the Rack switches = k/2

Number of Aggregate switches = k/4

Number of Core switches = k/8

A detailed metrics of the architecture we are implementing in this paper is embedded in the TABLE 2 below. The highlighted row of TABLE 2 depicts how the Data center architectures are being scaled and the column stating the number of aggregate layers denotes the increased number of layers in the Data Center architecture. The edge layer, which consists of the TOR switches can be scaled horizontally by increasing the number, switches which are directly connected to the hosts.

TABLE 2: Multi-tiered architecture design based host and switch count

Number of racks	Number of Servers per rack	Number of TOR switches per rack	Number of Aggregate Switches per block	Number of Aggregate switch blocks	Number of Core Switches in total	Total number of servers	Total No.of TOR Switches	Total No.of Aggregate Switches
4	4	2	1	0	4	16	8	4
4 racks with 4 servers each - No Scaling								
4	4	2	2	3	2	16	8	24
4 racks with 4 servers each - Aggregate Layer scaled to 3 layers								
4	8	4	2	0	2	32	16	8
4 racks with 8 servers each - No Scaling								
4	8	4	2	3	2	32	16	8
4 racks with 8 servers each - Aggregate Layer scaled to 3 layers								
8	8	4	2	0	8	64	32	16
8 racks with 8 servers each - No Scaling								
8	8	4	2	3	4	64	32	16
8 racks with 8 servers each - Aggregate Layer scaled to 3 layers								
2	48	24	12	2	2	96	48	24
2 racks with 48 servers each - Aggregate Layer scaled to 2 layers								
6	16	8	4	0	6	96	72	24
6 Racks with 16 servers each - Edge Layer scaled by increasing TOR switch count by 60%								

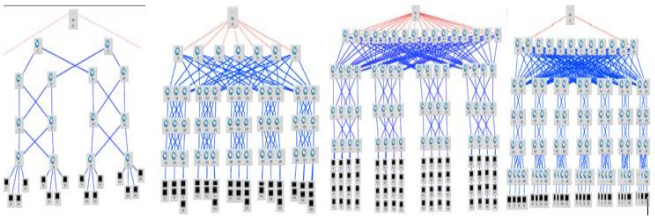

Fig. 3: Fat-tree vertically scaled architectures for 16, 35, 64, and 98 hosts.

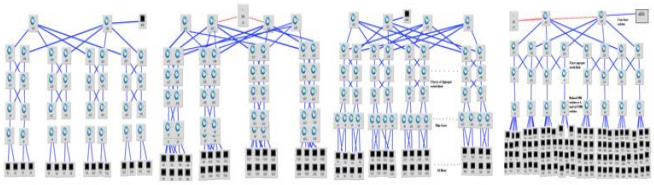

Fig. 4: Multi-tiered vertically scaled architectures for 16, 32, 64, and 98 hosts.

IV. PERFORMANCE ANALYSIS

A. Iperf test:

Iperf (Internet performance test) is a tool, which we used to test the limits of the network [10]. It is used to measure the maximum bandwidth achievable on the IP networks. With the iperf tool we will have the ability to choose the parameters like protocols, timing and buffer sizes. When networks are tested with iperf results involve the parameters like bandwidth, loss and other parameters.

B. Ping test:

Ping is to test the network performance, it is generally used to analyze parameters like total number of packets sent and received, see the percentage of lost packets over different paths, response times both minimum and maximum generally in milliseconds, and also the average response time in milliseconds. Ping test helps us to check the connectivity as well as the response time for each packet. Ping can be done between two hosts and also between all the nodes in network, which it is, can be a unicast or broadcast type [11]. Typically ping test latency is often less than 100 milliseconds.

C. Comparitive analysis of layered and non layered fat-tree architectures:

It was observed from TABLE 3 and TABLE4 that average bandwidth and average delay of non-layered architectures is greater than average bandwidth and average delay of layered architectures for both inter rack communication and intra communication. The only exception was the 35 host layered fat tree architecture in which the intra rack average bandwidth was less than inter rack average bandwidth and the intra rack average delay is higher than inter rack average delay, indicating that the architecture for 35 hosts layered is correctly tailored, to provide higher bandwidth for inter rack communication in layered architectures as compared to non-layered architectures and to provide lower delay for inter rack communication in layered architectures as compared to non-layered architectures. The exception was due to the number of switches and layering in 35 host-layered architecture that provided efficient bandwidth allocation and response time for end-to-end packet transfer between different racks. For 16 hosts layered and non-layered architecture the bandwidth was high that was due to the less load on the network, which has high performance of the mininet. It was observed that average bandwidth for communication between hosts in same rack is higher than average bandwidth for communication between hosts in different racks and also average delay between the hosts in different rack is higher than the average delay between hosts in the same rack.

This is because for inter rack data transfer, the data packet has to go through layers of aggregate switches and one layer of core switches which is indicated in the topology. The layers of aggregate switches increase the number of times a packet has to pass through switches to reach the end racks of the network topology which in turn results in the packet having to contend for available bandwidth along with other Inter Rack communication packets and Switch configuration messages, controller messages in the network. This reduced the bandwidth available for individual packets during inter rack communication and increases the switching time, number of times it has to share the bandwidth. In Intra rack communication, the packets only have to pass through maximum of two TOR switches and sometimes through a single Aggregate switch to reach the distant host connected to the Same Rack. Since the TOR switches forward the packets directly to the end hosts and do not have to deal with forwarding of other packets in inter Rack data transfers, most of their bandwidth is available for forwarding of packets towards destination hosts.

TABLE 3: Average bandwidth of all the fat tree topologies (Inter &Intra)

Average Bandwidth	16 Host Non Layered	16 Host Layered	35 Host Non Layered	35 Host Layered	64 Host Non Layered	64 Host Layered	98 Host Non Layered	98 Host Layered
Average Bandwidth Between Hosts from Different Racks (Inter Rack)	20.66	8.7	3.20	6.81	4.09	3.13	4.11	3.64
Average Bandwidth Between Hosts from Same Racks (Intra Rack)	26.46	24.91	4.73	5.54	4.70	3.75	5.08	4.91

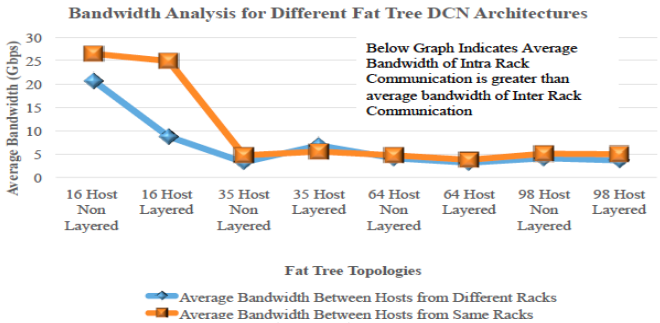

Fig. 5: Average bandwidth of all the fat tree topologies (Inter &Intra)

TABLE 4: Average delay of all the fat-tree topologies (Inter &Intra)

Average Delay	16 Host Non Layered	16 Host Layered	35 Host Non Layered	35 Host Layered	64 Host Non Layered	64 Host Layered	98 Host Non Layered	98 Host Layered
Average Delay Between Hosts from Different Racks (Inter Rack)	7.97	20.58	28.20	43.15	46.61	59.34	27.29	44.24
Average Delay Between Hosts from Same Racks (Intra Rack)	2.48	7.18	59.01	74.66	22.30	31.67	12.9	20.73

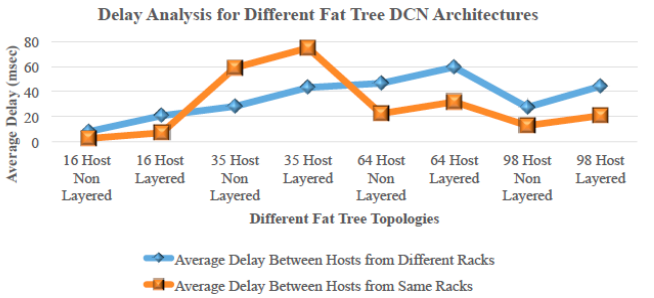

Fig. 6: Average delay of all the fat tree topologies (Inter &Intra)

D. *Comparitive analysis of layered and non layered multi-tiered architectures:*

We compared the overall bandwidth and overall delay of layered and non-Layered architectures. We conclude from TABLE 5 and TABLE 6 that the performance of data Center has increased when it has been scaled compared to the performance when it is non-scaled. It was observed that average bandwidth and delay of non-layered architectures is greater than that of the layered architectures for both inter rack communication and intra communication. The overall bandwidth was increased with the increase in the number of hosts in case of Layered architecture. While the bandwidth has decreased as number of hosts increased in non-layered architecture. Similarly, the delay has decreased as the number of hosts increased in layered architecture and it has linearly increased in case of a non-layered architecture.

TABLE 5: Average bandwidth of all the multi-tiered topologies (Inter &Intra)

Number of Hosts	Average Bandwidth	
	Layered	Non-Layered
16	21.20	23.90
32	14.90	18.90
64	23.40	12.5
96	28.30	18.40

TABLE 6: Average delay of all the multi-tiered topologies (Inter &Intra)

Number of Hosts	Average Delay	
	Layered	Non-Layered
16	1.60	5.60
32	72.60	18.20
64	51.50	39.8
96	42.40	88.10

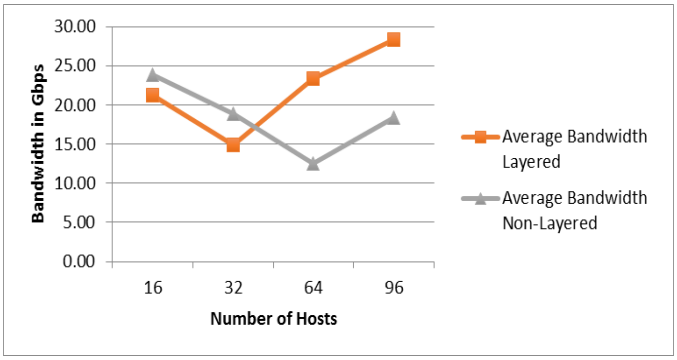

Fig. 7: Average bandwidth of all the multi-tiered topologies (Inter &Intra)

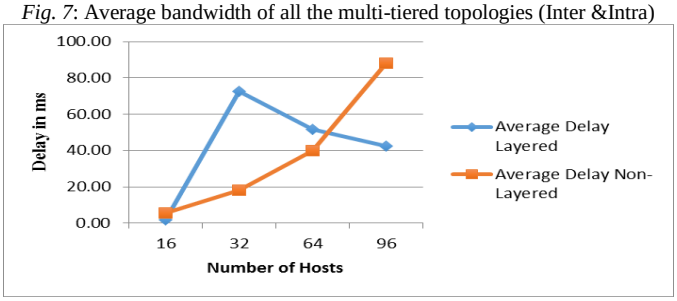

Fig. 8: Average delay of all the multi-tiered topologies (Inter &Intra)

E. *Comparitive analysis of fat-tree and multi-tiered architectures:*

We calculated and compared the overall average bandwidth and overall average delay for all the designed architectures in multi-tiered topology and fat-tree topology. The data was compared in terms of the number of hosts in the network and the type of network (Layered or Non-Layered). For accurate results of comparison, we designed two topologies with same metrics. The below tables summarizes the results of the performance of the architectures.

From the bandwidth analysis, we observed that multi-tiered architecture provides better bandwidth to data paths than fat tree architecture in case of both layered and non-layered architectures. Only for 16 host non-layered Fat Tree architecture, the bandwidth is comparable to Multi-tiered architecture.

TABLE 7: Average delay data of layered and non- layered fat-tree and multi-tiered architectures.

Average Delay	Layered				Non-Layered			
	16 Host	35 Host	64 Host	98 Host	16 Host	35 Host	64 Host	98 Host
Fat Tree Architectures	13.88	58.91	45.5	32.48	5.22	43.6	34.54	20.09
Multi-tiered Architectures	1.6	72.6	51.5	42.4	5.6	18.2	39.8	88.1

TABLE 8: Average bandwidth data of layered and non-layered fat-tree and multi-tiered architectures.

Average Bandwidth	Layered				Non-Layered			
	16 Host	35 Host	64 Host	98 Host	16 Host	35 Host	64 Host	98 Host
Fat Tree Architectures	16.8	6.17	3.44	4.3	23.56	3.96	4.4	4.6
Multi-tiered Architectures	21.2	14.9	23.4	28.3	23.9	18.9	12.5	18.4

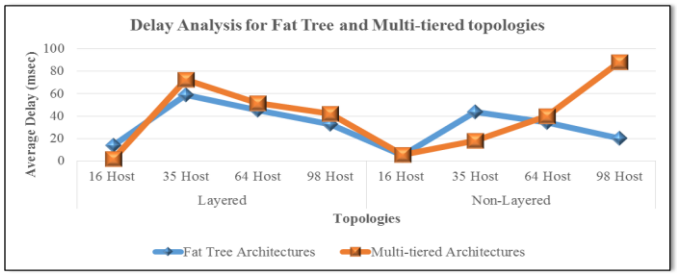

Fig. 9: Average delay analysis for fat-tree and multi-tiered, layered and non-layered architectures.

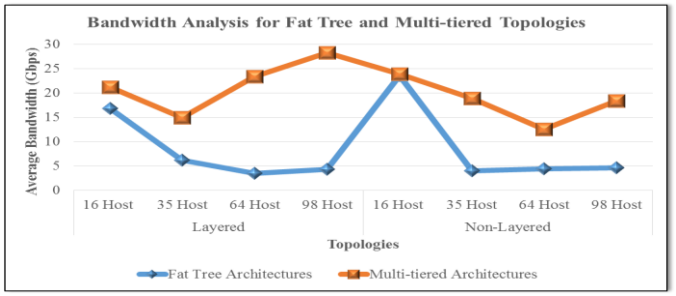

Fig. 10: Average bandwidth analysis for fat-tree and multi-tiered, layered and non-layered architectures.

The difference in bandwidth is because Fat Tree Architecture follows an algorithm for determining the number of switches in the network based on the number of hosts per rack, which results in an increase in the overall number of switches. Scaling this architecture, doubles the number of switch connectivity, which in turn increases the number of times, packet has to pass through switches before reaching its destination. The increased connectivity to multiple core and aggregate switches decreases the available bandwidth as core and aggregate switches also forward controller messages and configuration messages from the switches across the same links through which data packets pass. Any such algorithm does not limit multi-tiered Architecture. It depends only on the size of switches used for TOR and aggregate connections, which allows multiple hosts to be connected to single switch and reduces the number of switches used at TOR, aggregate and core layer, providing better bandwidth from source to destination.

F. Compartive analysis of core to host connectivity for both fat-tree and multi-tiered architectures:

We tested communication between Core Layer Switches and Hosts present in different Racks for both 16 Host Layered and Non Layered Architectures and 64 Host Layered and Non Layered Architectures. We Attached a Host to one of the Core Layer Switches and initiated ICMP Ping requests from Core Switch Host to Host in various racks. The Delay and Bandwidth observed for each of such communication is as shown in table 9.

TABLE 9: Average bandwidth and delay of core to host data of both multi-tiered and fat-tree architectures.

Architecture	Bandwidth	Delay
Multi Tier	24.7	21.425
Fat Tree	8.54	15.97

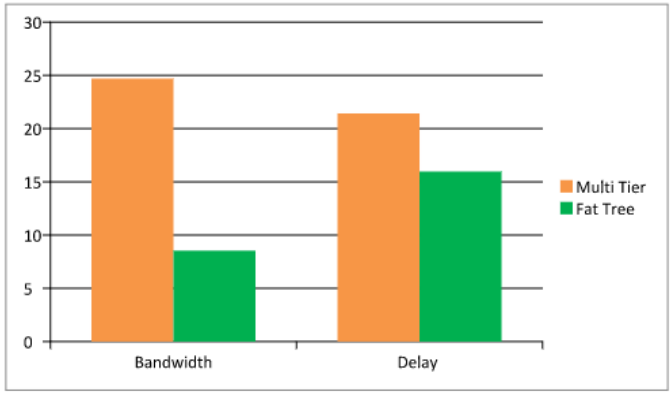

Fig. 11: Average bandwidth and delay of core to host

V. CONCLUSION

In this paper, we designed and implemented Multi-Tier and Fat-Tree Data Center Architectures making use of the forward learning switches for centralized control, to compare and analyze the performances of two different Data Center Networks. We created a combination of architectures with 16, 32, 68 and 96 hosts for end-to-end testing. For the better performance of the network, we have scaled the networks vertically by increasing the number of aggregate layers and horizontally by increasing the number of Edge Layer switches and compared the Delay-Bandwidth values of the scaled and non-scaled networks.

From the Delay-Bandwidth analysis, we conclude that, Multi-Tier architecture performs better when vertically scaled, as it increased the efficiency of network by increasing the bandwidth and decreasing the delay. However, horizontal scaling, in spite of offering a better bandwidth, it has increased the delay and cost of the network which is not suited for a real-time application.

When compared to Fat-Tree architecture, from the analysis we have performed, we observed that Multi-tier is the better suited architecture for scaling as the algorithm allows to alter the number of switches to balance load on the network unlike Fat-Tree which has to bind to the algorithm which rapidly increases number of Switches with increased number of hosts and thus increases network complexity and cost.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Dr. Nader F. Mir, Professor, Department of Electrical Engineering, SJSU for guiding and advising us all along the progress of this Paper. He encouraged by providing all the vital information and resources, helping us to realize the need for this kind of work. Creating a free learning environment he gave us the opportunity to express our views and opinions regarding the paper. We would also like to take this opportunity to thank the Electrical Engineering department, SJSU for providing us the tools and resources necessary for successful completion of our project.

References

[1] Software Defined Networks (EWSDN), 2013 Second European Workshop on, Issue Date: 10-11 Oct. 2013, Written by: Teixeira, J.; Antichi, G.; Adami, D.; Del Chiaro, A.; Giordano, S.; Santos, A.

[2] OpenFlow Home Page. http://www.openflow.org. Last visited: 10-08-2015.

[3] POX Home Page. http://ww.noxrepo.org. Last visited: 10- 08-2015.

[4] OpenDaylight Consortium. http://www.opendaylight.org

[5] M. Al-Fares, A. Loukissas, and A. Vahdat, ``A scalable, commodity datacenter network architecture," in Proc. ACM SIGCOMM, 2008, pp. 63_74.

[6] Mininet Home Page. https://mininet.github.com. Last visited: 0-08-2015

[7] TING WANG1, (Student Member, IEEE), ZHIYANG SU1 , (Student Member, IEEE), YU XIA1 , (Member, IEEE), AND MOUNIR HAMDI1,2 , (Fellow, IEEE) Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hamad Bin Khalifa University, Doha, Qatar " Rethinking the Data Center Networking:Architecture, Network Protocols and Resource sharing ", 2013

[8] A. Greenberg et al., ``VL2: A scalable and _exible data center network," ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 4, pp. 51_62, Oct. 2009.

[9] T. Lam et al., ``NetShare: Virtualizing data center networks across services," Dept. Comput. Sci. Eng., Univ. California, Berkeley, CA, USA, Tech. Rep., 2010.

[10] Barayuga, V.J.D. Inst. of Comput. Studies, Ilocos Sur Polytech. State Coll., Santa Maria, Philippines Yu, W.E.S. Study of Packet Level UDP Performance of NAT44, NAT64 and IPv6 Using Iperf in the Context of IPv6 Migration, INSPEC: 14882156, IEEE conference, Beijing, Oct 2014

[11] Do Nguyet Quang Dept. of Electron. & Commun. Eng., Univ. Tenaga Nasional, Kajang, Malaysia Ong Hang See ; Dao Viet Nga ; Lai Lee Chee ; Chee Yung Xuen ; Karuppiah, S.A.L. ; Fadzil Mohd Siam, M. IEEE paper published in Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on 26-28 Nov. 2012. ISBN 978-1-4673-5832-3.