International Journal of Engineering & Technology, 7 (2.4) (2018) 190-195

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Study of optimization parameters for service chaining in cloud environment

Prathamesh Purohit [1] *, Ruturaj Kadikar [2], M. Susila [3], B. Amutha [2]

[1] *Department of Information and Telecommunication Engineering*
[2] *Department of Computer Science Engineering*
[3] *Department of Telecommunication Engineering SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.*
**Corresponding author E-mail: pratham2992@gmail.com*

Abstract

Recently, the access to content over the internet has increased in a significant way because of advancements in communication networks and it is growing towards integration of 5G technology in near future. Therefore, to improve the quality of experience for viewing the content over the internet requires dynamic allocation and adaptation of network resources in an optimized manner. Traditional IP networks are vertically integrated hence flexibility in network resources management is very less. Software-defined networking (SDN) as an emerging technology, which comes with the promise of the solution to dynamically govern various network resources by breaking this chain or hierarchy of vertical integration. Network function virtualization along with service chain optimization provides the solution to enhance the Quality of Experience (QoE) and Quality of Service (QoS). In this paper, we are proposing an approach to improve the QoE by ameliorating the service chain and data center parameters.

Keywords: *Software-Defined Networking; Network Function Virtualization; Quality of Service; Quality of Experience*

1. Introduction

Service Function Chaining (SFC) furnishes a very simplified management and configuration of the network, so that service providers can easily visualize a number of policies on security, assess control, packet modification, traffic engineering and Quality of service (QoS) [1] [2] [3]. In service chain, a single network can be used in various ways or multiple network policies can be deployed by creating virtual chains of multiple network components. A service chain in networking consists of a set of services such as firewalls, video optimizer, parental controls or application delivery controllers (ADCs). These components are interconnected through the network to support the application [2].

Service functionalities such as antivirus, firewalls or video optimizers can be planted at the different points of the network as shown in Figure 1. The operational nodes in the network will have one or more service function chains.

Previously in traditional networks, these service functions were dependent on the actual location of the operational node and the service functions available at that location i.e. actual service function for any node is not fixed.

Building a service chain to support new applications takes more efforts and more time in traditional networks, i.e., it requires network devices which have to be cabled in the correct sequence. Each service functionality requires a specialized set of hardware devices and each device needs to be configured individually with its own command set. In this scenario, if one component fails then the entire network gets affected and chances of errors increments. Over a period of time if there is an increase of load in the application then building an immediately reconfigured chain will not help in estimating future demands and over-provisioning of services to support growth. To support maximum level of demand which might only occur at the particular time of the year, devices needed to be sized, i.e. for extra capacity extra capital investment required.

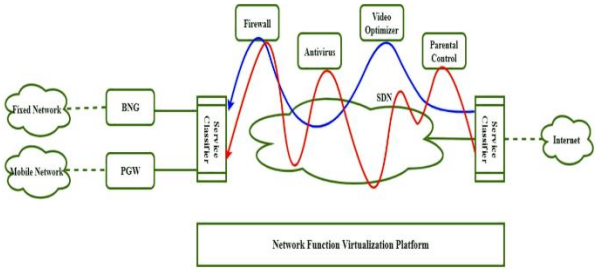

Fig. 1: Service Function Chaining.

In order to tackle issues in a traditional network, Software Defined Networking (SDN) and Network Function Virtualization (NFV) combine, provides the best solution by creating virtual chains of the network resources. SDN and NFV can make application provisioning and service chain processes shorter and easier.

SDN is an emerging technology, in which Control plane and Data plane are decoupled and Data plane is remotely controlled by the Control Plane. Decoupling of Control plane and Data plane increases the flexibility of the network which is advantageous in Service Chaining. The content delivery in SDN network is flow-based which was initially destination based in traditional networks [5] i.e. in the traditional network each packet contains the address of the destination and forwarding of these packets only depends on the destination address included in the header.

In traditional networks, middlebox services are not totally automated, but introducing NFV can make these networks fully auto-

mated. NFV has upper hand on the traditional network in terms of network service deployment flexibility. NFV implementations using commercial high volume servers provides system administrators flexibility to decide where and when to launch a service.

A survey from Markit's market tracker has shown that revenue for datacenter and virtual security was USD $2.4 billion in 2015 and at the end of 2020, it will boost to USD $3.9 billion i.e. nearly 62% growth in just 5 years. These numbers indicate the rising popularity of SDN and NFV technology in service provisioning [2].

In present SDN-NFV enabled networks, service functions are not location dependent i.e. service node can adapt any service function as per the payload. Software configured service chaining is proficient to determine the type of payload and is capable of determining the best progression of resources for that payload i.e. service chains can be highly dynamic or based on pre-defined service templates. NFV provides resilience to scale up or scale down service functions in the network, it helps in reducing capital and operating expenditure involved in networking infrastructure and management.

This paper is arranged as follows. Section II provides previous research on SDN and Service Function Chaining. Section III gives an overviewon components of OpenStack environment and implementation of OpenStack. It is followed by results and analysis of the deployed environment in section IV. The conclusion is covered in section V.

2. Literature survey

As stated by Benson et al. [4] the traditional IP networks are labyrinth and tough to operate/ regulate, as data plane and control plane are coupled together and embedded in the same networking device. This traditional networking structure is highly distributed. The information to stream in the form of digital packets depends on switches, routers and key technologies running inside these devices like Transport Network Protocols (TNP), Distributed Control Protocol (DCP).

In Software-Defined Network architecture, the control plane and data plane are highly decoupled as per Kreutz et al.[5]. As shown in Fig. 2 the data plane is controlled by the remotely controlled control plane. In SDN Architecture (Fig. 2) NetworkController Platform has two interfaces northbound Interface and southbound interface. The southbound interface connects a particular component (e.g. SDN Controller) with lower-level components in the network (e.g. switches, routers). Traditional IP networks are very complex, difficult to manage and difficult to configure as per predefined policies. As described by Kreutz et al.[5] bundled data & control plane and vertical integration of current networks reduce the flexibility and progression of the network. Software-defined networkingprovides a solution for these networks, by breaking vertical integration of networks and control logic is taken out of the network's middle-box services e.g. switch, router, access points etc.

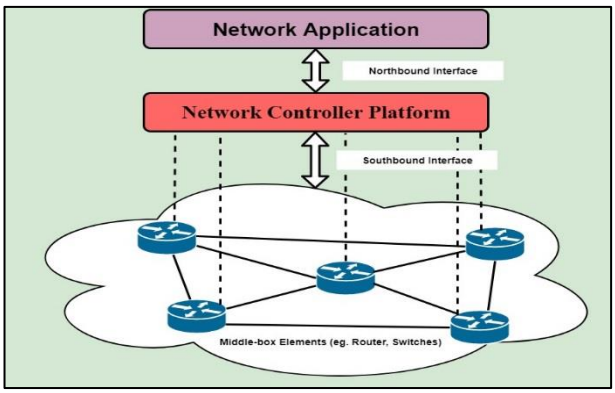

Fig. 2: SDN Architecture.

Academic experiment like SDN has gained momentous assiduity in the industry and in past few years, it has emerged and developed as a commercial success. Google has deployed an SDN technology to interconnect their data centers across the globe this is a great example of the commercial success of SDN technology. Middlebox services like network switches, routers, firewall, and intrusion detection system play a crucial role in the network as mentioned by Kreutz[5]. In traditional network these middlebox services are not entirely programmed or computerized i.e. they are semi-manual. While configuring these traditional networks they are error-prone to human inadvertence. Consequently, to address this issue, an idea of Network Function Virtualization (Fig. 3) was proposed in the system which at that point expanded the adaptability and flexibility of network services deployment.

A described by Qiang et al. [6] evolution of SDN and NFV technology has produced substantial harmony to address challenges in networking. Combination of SDN and NFV principle forms the Software-defined network virtualization (SDNV) in which control and data plane are decoupled (which is SDN Principle) along with decoupling of service functions of the infrastructure (NFV Principle).

Lijun et al. [7] have proposed Least Busy Path Algorithm (LBPA), which can be utilized for strengthening the system capacities in an ideal method to guarantee network services are conveyed according to request and policies. The content delivery in the traditional network was destination-based which is flow-based in current SDN enabled network as specified by Jun et al. By using Flow-based routing NFV along with SDN transforms the network into flexible and programmable manner to deliver content on-demand basis. But for Flow-based routing, it is very important to have Stateful information of the network to process these flows accurately. Hence, for this OpenFlow protocol is used in SDN.

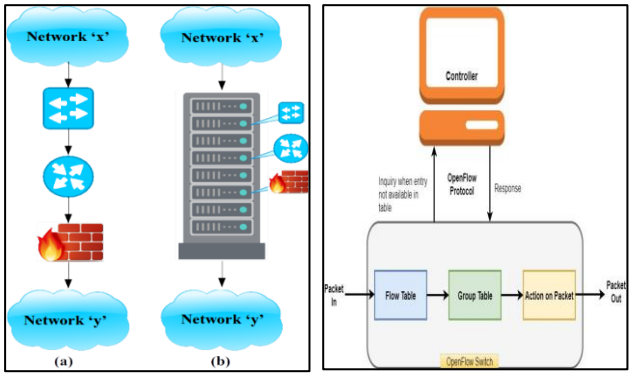

Fig. 3: Traditional Network vs NFV Networkfig.4 Working of Open flow.

Open Flow was primitively defined by Open Networking Foundation (ONF) and it is mainly used in the northbound interface of the network as explained by Lin et al.[8] By practicing/implementing/executing Open Flow protocol in SDN environment, SDN controller can directly communicate/ interact with forwarding equipment/ networking plane e.g. switches, hypervisors, firewalls, and routers. To identify and categorize incoming packets Open v Switch can be programmed as pre-requisite. As shown in Fig.4 incoming traffic is processed by Open Flow switch through the pipeline and for this, it uses one or more flow tables or group tables. By checking flow entries from the tables it determines what action should be taken on a packet. The switch will either drop the packet or ask the controller to take appropriate action on the packet if it fails to match flow entries in the table. The controller will acknowledge with 'Flow Modification Message' to switch, which will direct the switch on how to process this packet. At the same time switch will modify its flow and group table to process the packet independently, when a packet from the same source will arrive next time.

As mentioned by Jun Bi et al. [9] OpenFlow is heavily reliant on SDN controller, hence it has some stipulations/limitations to support advanced networking function. To alleviate this issue an Open vSwitch (OVS) to support forwarding processor with SDN

controller can be used, by which Stateful Data Plane Abstraction (SDPA) performance can be improved, also SDPA architecture is copiously compatible with Open Flow In Stateful architecture, every packet will go through the controller, this causes slight latency in packet delivery but overhead and throughput remains nearly unchanged. This latency can be reduced by using Policy-based Dynamic Service Chaining along with NFV as suggested by Scheid et al. [10]

To further improve the throughput with the secure data flow, packets can be forwarded to the controller after fixed number packets are exchanged between the nodes. This will reduce the latency in Stateful architecture and improves security provided due to this periodic checking of packets.

As suggested by Azodolmolky [11] to provide service requirements in more flexible and timely manner, SDN and NFV capabilities can be combined together to develop Next Generation Service Overlay Architecture (NGSOA). Service-oriented architecture (SOA) has been adopted by many operators and service providers as it gives the option of re-using existing infrastructure to develop and add services to existing infrastructure. As proposed by Martini et al. [12] Context-aware data delivery can be increased i.e. capability of collecting, processing and routing content as per context can be increased by dynamically managing middlebox services like routers, switches, and firewalls.

Akyildiz et al. [13] indicated that Real-time multimedia applications usually require QoS assurances of bandwidth or delay but are tolerant to packet loss. Low delay and high bandwidth requirements are major challenges, and efficient QoS routing mechanisms are urgently needed to realize multimedia applications. In recent time, many QoS routing protocols and mechanisms are proposed for optimizing the network resource utilization by avoiding traffic congestion and balancing network traffic. Most of the mechanisms and protocols focus on calculating the optimized path with best network conditions without considering QoS requirements of applications. An application may consist of various types of flows, such as text, voice, and video. A decision on the path would be made for each flow according to its QoS requirements and also network conditions.

As proposed by Kumar et al. [14] to fulfill QoS requirements, Service function chaining (SFC) is one of the technique which can be followed in order to take maximum advantage of network resources. These service functions can be placed at the different points of the network. In this, service/operational nodes in the network will have one or more service function chains. Previously, in traditional networks, these service functions were dependent on the actual location of the service node and the service functions available at that location i.e. actual service function for any node was not fixed. In present SDN-NFV enabled networks, service functions are not location dependent i.e. service node can adapt any service function as per the payload e.g. video data, voice data, numeric data, alphanumeric data and for selective traffic.OpenStack is a platform that has enabled the implementation of above technologies in the network. As stated by Belmiro [15] the OpenStack is a group of open source projects and it provides an operating platform for organizing public and private clouds on a massive scale. OpenStack controls a huge numberof network resources and storage in the datacenters. It can be managed through dashboard or OpenStack API (Application Programmable Interface).

As stated by Buyya et al.[16] Software-Defined Networking address the shortcoming of traditional networks, also it deals with recurrent and immediate changes in data centers. For this, network resource management should be open and modernization-friendly and also it is important that developers should be familiar with these test-beds. To address this Son et al. [17] proposed CloudSim-SDN, which helps developers to validate their policies on simulation environment with compute and network resources. It is designed in such a way that developers can dynamically manage the network and compute resources, which helps them to implement policies in the real-time environment.

3. Implementation

In past few years, cloud technology emerges in, a significant way as it provides various services to users e.g. Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS) etc. Demand for these services has been increasing day by day and hence cloud providers are mending ways to maximize use of available infrastructure resources. To analyze new policies some simulation platform is required which is provided by CloudSim therefore CloudSim is included in this implementation.

3.1. Cloud Sim

In order to maximize use of infrastructure different algorithm, policies, application methodologies, and modeling technique are being developed by the cloud providers. Testing these technologies on the real-time running environment is not possible. Hence, to test algorithms, policies and application methodologies, cloud developers require a test-bed. CloudSim provides a test-bed for simulating these techniques, by which developers/providers can ratify their designed framework and by optimizing it further, more resource utilization can be achieved. Hence, we have included/ comprehended CloudSim in our implementation. For CloudSim implementation server with following configuration is used- Intel®Xeon(R) CPU E5-2690 v4 @ 2.60GHz × 56, RAM 125.5GB, 64 bit OS and 4.6 TB Hard Disk. For optimizing network resources default program of "DataCenter" available in CloudSim package is used. This program is modified as per created OpenStack environment on which policies are going to be implemented. For optimization purpose, following parameters are considered in CloudSim- Requests per user per Hour, Number of Virtual Machines (VMs), Memory (RAM), Bandwidth and Service broker policy. Optimization of above parameters is to reduce the overall response time and data processing time.

The number of parallel requests at each VM decides the time required to process each request. As the number of parallel request goes on increasing, overall data processing time also increases. Response time is also dependent on the number of VMs available to process received requests or data. Bandwidth is another major factor that plays an important role in calculating overall response and data processing time. In Section IV (Results) we will discuss the relation between above parameters.While running CloudSim simulation due to the random nature of the processes the results may vary for different simulations.

3.2. Open stack

For creating, managing and developing a large group of virtual private servers, OpenStack platform is used. It is an open source Infrastructure as a service. To simplify the highly available OpenStack environments and after deployment to manage them, Fuel can be used. OpenStack distributors provide the choices of databases, performance components, message queuing and orchestration tools. Networks are the foundation of our connected world. These networks need to evolve if they are to keep pace with the demands of ever-changing business models and technology innovation. Traditional systems are vertically integrated, so they are difficult to upgrade and manage. Network Functions Virtualization (NFV) is a vision of the network that takes advantage of advances in SDN, dynamic cloud architecture, and modern software provisioning techniques. Formerly networking tasks were performed by dedicated hardware.

NFV uses software to perform tasks in the network instead of using dedicated hardware devices. Open Platform for NFV (OPNFV) enables the development and evolution of NFV components across various open source systems. To accelerate enterprise and service provider networks, it creates reference platform for system-level integration, deployment, and testing.

OPNFV enables network system to use different controllers on a single platform. OPNFV takes compute virtualization, storage

virtualization and network virtualization of different SDN controllers and composes them into a single system.

3.3. Methodology of network deployment

Fig 5 shows the methodology of network deployment.
Which contains following components-

- Jump Server: managing devices in one security zone.
- Public Network: providing Internet access.
- Pxe Network: connecting jump server and other nodes.
- OpenStack Controller: managing overall network and policies.
- Compute Nodes: allocating and managing computing resources.
- Management Network: connecting the controller and compute node.
- Private Network: accessing storage traffic.

3.4. Nodes

- Controller Node

SDN Controllers are the "brains" of the network. In fact, SDN controller works as the Operating System of the network. It acts as a strategic control point in the SDN network; it manages flow control to the switches/routers via southbound Interface and the applications and business logic via northbound Interface to deploy intelligent networks (IN).

- The controller promotes automated network management and makes it easier to integrate applications if the control plane is taken off the network hardware and running it as software. To communicate with switches and routers in the network controller uses OpenFlow and OVSDB (Open vSwitch Database) protocols.

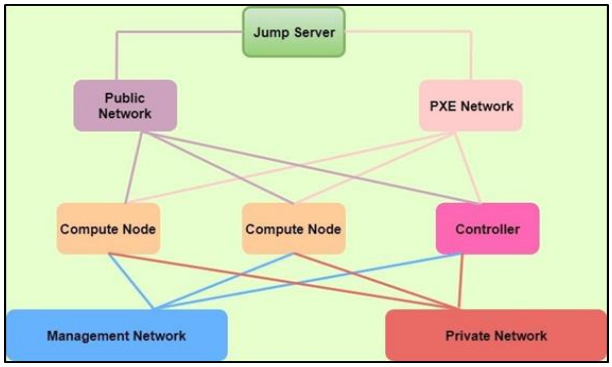

Fig. 5: Methodology of Network Deployment.

- Storage Node

It is usually a physical server, which contains one or more hard-disks (HDD) or solid-state devices (SSD). A storage node can be a virtual machine with access to one or more HDDs or SSDs. A number of storage nodes are clustered together and managed through software as a single pool of storage. It is preferred to have more than one storage node to provide redundancy.

- Compute Node

The compute node is the system that actually runs hypervisor. This is the system where virtual instances are run. Virtual Machines (VM) get CPU, storage, network and memory resources from compute node through the hypervisor.

- Master Node

Master maintains a registry/record of all nodes. When the system administrator registers a node, master allocates an unused subnet from the cluster network and stores this subnet in the registry. When a node is deleted, master deletes the subnet from the registry and considers the subnet available to be allocated again. The cluster network has total 256 subnets available to assign to nodes. The address range and size of the cluster network is configurable, as is the host subnet size.

3.5. Networks

- Reboot Execution Environment (PXE) Network

It is an industry standard client/server network interface that allows networked systems that are not loaded with an operating system and has to be configured remotely using their network card.

- Public Network

It provides Internet access to existing network to communicate with external nodes. It also provides an external network for VMs.

- Private Network

It is nothing but internal network between the controller and compute node. It carries tenant traffic inside the network.

- Management Network

It is an internal network between the controller and compute node used to manage controller and all compute nodes inside that network. It also manages private API (Application Programming Interface) that enables developers to use backend data application facility within the network.

- Storage Network

It is also an internal network between the controller and compute node used to manage storage inside that network. It provides a network for storage traffic.

3.6. Services

- Identity Service (Keystone)

This service provides a single point of integration for managing authentication, authorization and service catalog. Users and services can locate other services by using a collection of other services in an OpenStack deployment.

- Compute Service (Nova)

This service serves control over instances & networks and allows to manage access to the cloud through users and projects. It interacts with other OpenStack services e.g. Keystone for authentication, Horizon for the user and administration interfaces, and Image by limiting access by projects and users. It limits the number of floating IPs, fixed IPs, instances, volumes, amount of RAM and CPUs.

- Object Storage Service (Swift)

It is used for storing VM images and data.

- Networking Service (Neutron)-

Neutron manages all networking aspects and features for the virtual networking infrastructure. Access layer of neutron service aspects the physical networking of infrastructure.

- Image Service (Glance)

This service enables users to discover, register and retrieve VM images. It accepts RESTful (REpresentational State Transfer) API requests for disk or service images.

- Block Storage Service (Cinder)-

It is used for adding additional persistent storage to VMs.
The following table compares Object and Block storage services.

Table 1: Comparison of Object Storage and Block Storage

Object Storage (Swift)	Block Storage (Cinder)
Used for storing VM images and data.	Used for adding additional persistent (continuous) storage to VMs.
Fixed until deleted.	Fixed until deleted.
Accessible from anywhere.	Access linked with a VM
Encryption is available.	Encryption is available.
Easily scalable for future growth.	Sizing based on needs.
e.g.: 10s of TBs of data set storage.	e.g.: 1 TB extra hard drive.

Along with these services, additional service called Horizon (Dashboard) has been deployed to create projects, instances, for managing users, for creating security groups etc. For deploying above environment server with the following configuration is used Intel®Xeon(R) CPU E5-2690 v4 @ 2.60GHz × 56, RAM 125.5GB, 64 bit OS and 4.6 TB Hard Disk.

The created network environment can be easily visualized, planned, tested, and troubleshoot using network emulators without interacting with the hardware.

As per the availability of hardware, one can increase or decrease the number of nodes in the network. With the inbuilt graphical interface, it becomes easy to connect all types of virtual interfaces. In order, to achieve better QoE and QoS service chaining is used. Software-defined networking can create service chain of connected network services like routers, switches, firewalls, load balancers, data centers, etc.In service chaining single network can be used in multiple ways or different network policies can be implemented by creating virtual chains of different network components. Services which can be connected to the network by using software are generally represented as service chaining. SFC is very convenient in the NFV environment in which new services can be merged, as software-only, running on service hardware. By using SFC and NFV together, a large number of virtual network functions can be connected in the environment. These connections can be controlled NFV orchestration layer.

4. Results and discussions

For scenario mentioned in Section III, simulation is performed on CloudSim to observe overall response time and Datacenter processing time. In this, the number of VMs (100) is kept constant, and the number of parallel requests is varied for 3 different bandwidth (10Gbps, 20Gbps, and 40Gbps) as shown in Table 2. From Fig. 6 and Fig.7, it is observed that the number of parallel requests increased, but the data processing time for Datacenter was not affected significantly. Data processing time has increased by only 0.01mSec and Overall response time from the Datacenter has increased by just 0.02mSec which is acceptable. Multiple VMs can perform parallel processing of requests better than a server with single VM performing serial processing on requests.

Though increasing number of VMs will increase parallel requests, it is not the best solution, as the number of VMs will increase, the bandwidth required to carry output or response from VMs will also increase. Hence it is always important to maintain the ratio between the Number of VMs, Bandwidth and Overall response time.

Fig 8 shows the relation between VM, bandwidth and overall response time. By comparing response for 20Gbps and 40Gbps bandwidth and respectively for 100 and 1000 VMs, it is observed that the overall time response is increasing as the number of VMs are increasing. As bandwidth is increased for the same number of VMs, overall time response is reducing.

Table 2: Iitime Response Analysis

BW(Gbps)	Parallel Requests	Overall Response Time(ms)	Data Processing Time(ms)
10	100	200.19	0.12
20	100	200.15	0.12
40	100	200.13	0.11
10	500	200.08	0.16
20	500	200.07	0.17
40	500	200.07	0.16

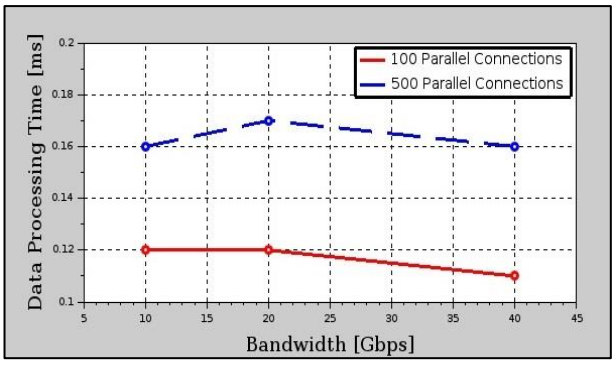

Fig. 6: Data Processing Time for Multiple Parallel Connections.

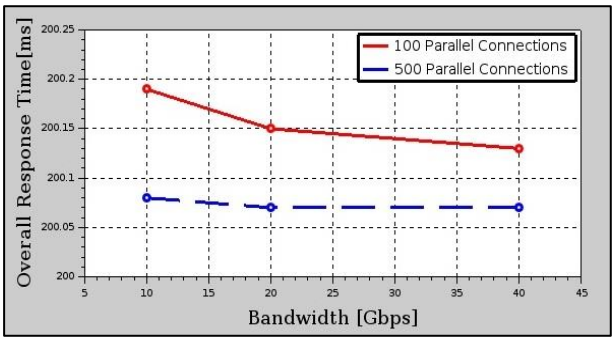

Fig. 7: Overall Response Time for Multiple Parallel Connections.

Table 3: Relation between Vm, Bandwidth and Overall Response Time

VM	BW(Gbps)	Overall Response Time(ms)
10	10	406.26
10	20	410.52
50	10	493.75
50	20	415.01
50	40	409.52
50	60	398.57

Fig 7 shows the relation between VM, bandwidth and overall response time. By comparing response for 20Gbps and 40Gbps bandwidth and respectively for 100 and 1000 VMs, it is observed that the overall time response is increasing as the number of VMs are increasing. As bandwidth is increased for the same number of VMs, overall time response is reducing.

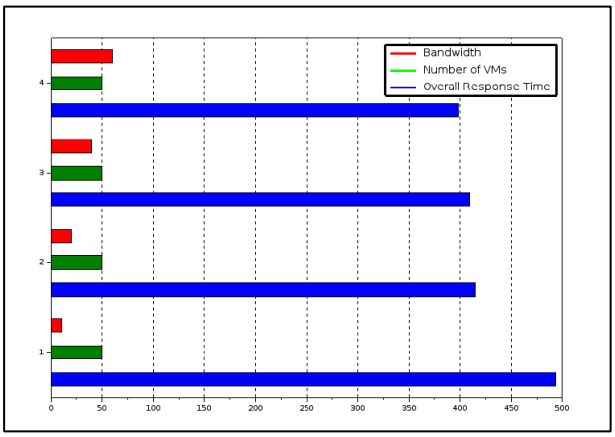

Fig. 8: Relation between VM, Bandwidth and Overall Response Time.

To measure overall time response from the data center to user base simulation is performed for the bandwidth of 20Gbps, 40Gbps, 60Gbps, 80Gbps and 100Gbps, VMs are set to 100 and parallel requests are varied from 1000 to 5000.

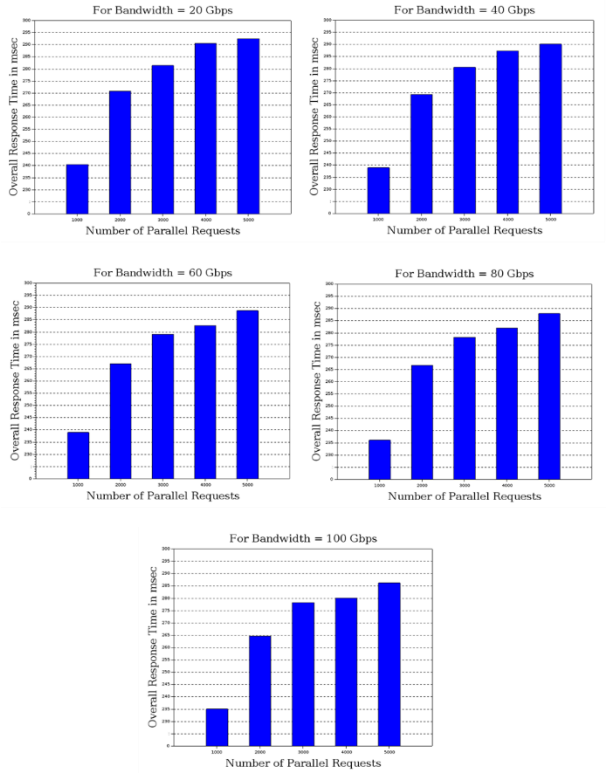

Fig. 9: Depict the Overall Time Response Taken for Varying Parallel Requests with Different Bandwidths.

It is noticed that the overall time response is directly proportional to number of parallel requests but inversely proportional to available bandwidth. As we can observe in the graph for 20Gbps bandwidth overall time response for 5000 requests is nearly 292msec, at the same time for 100Gbps bandwidth it is 286msec i.e. less time required to get back to user base than required for 20Gbps.
VMs handle multiple requests simultaneously, process them and return responses to users. This entire process takes some amount of time and the duration of time is depended on number of requests that VMs need to process. Hence as the number of requests increases, VMs take more time to process and delays to respond to user base.

5. Conclusion

Implementation of service function chaining using SDN is easier and cost-effective as compared to traditional networks. Also, the QoS and QoE factors are better than the traditional networks. From the simulation, it is observed that by using parallel processing we can accommodate more user requests at a time. But only performing parallel processing of requests cannot curtail delay, the processor should be fast enough toaccumulate all the responses without adding delay. Bandwidth is another factor to take into account in reducing delay, more bandwidth is required to accommodate the increased number of responses.Flexible and policy-based service chaining can provide better results to achieve QoS and QoE for online video streaming. As a next step, we intend to optimize additional network parameters like data size per request, service policy, and the memory of VMs in the Datacenter and Load balancing policies across VMs in the Datacenter. We envisioned flexible NFV Service chaining deployment and redeployment according to different user QoS requirement in a real-time environment.

Acknowledgement

We would like to express our gratitude to the research group, Software Defined Research Lab, Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, for their kind assistance and providing us with the required resources.

References

[1] V. Sekar, N. Egi, S. Ratnasamy, M. K. Reiter, and G. Shi, “Design and implementation of a consolidated middlebox architecture,” in Proceedings of the ninth USENIX conference on NSDI, 2012, pp. 24–24.
[2] Huang, H., Guo, S., Wu, J. and Li, J., 2017. Service chaining for hybrid network function. IEEE Transactions on Cloud Computing.
[3] J. Halpern and C. Pignataro, “Service function chaining (sfc) architecture,” IETF, Tech. Rep., 2015.
[4] Benson, T., Akella, A. and Maltz, D.A., 2009, April. Unraveling the Complexity of Network Management. In NSDI (pp. 335-348).
[5] 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), pp.14-76.
[6] Duan, Q., Ansari, N. and Toy, M., 2016. Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks. IEEE Network, 30(5), pp.10-16.
[7] Xie, L., Jiang, Y., Wang, B., Xiong, G. and Cheng, G., 2016. An approach for network function combination based on least busy placement algorithm. China Communications, 13(Supplement 1), pp.167-176.
[8] Lin, P.C., Lin, Y.D., Wu, C.Y., Lai, Y.C. and Kao, Y.C., 2016. Balanced service chaining in software-defined networks with network function virtualization. Computer, 49(11), pp.68-76.
[9] Bi, J., Zhu, S., Sun, C., Yao, G. and Hu, H., 2016. Supporting virtualized network functions with stateful data plane abstraction. IEEE Network, 30(3), pp.40-45.
[10] Scheid, E.J., Machado, C.C., Franco, M.F., dos Santos, R.L., Pfitscher, R.P., Schaeffer-Filho, A.E. and Granville, L.Z., 2017, May. INSpIRE: Integrated NFV-baSed Intent Refinement Environment. In-Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on (pp. 186-194). IEEE.
[11] Azodolmolky, S., 2013. Software defined networking with OpenFlow. Packt Publishing Ltd.
[12] Martini, B., Paganelli, F., Mohammed, A.A., Gharbaoui, M., Sgambelluri, A. and Castoldi, P., 2015, April. SDN controller for context-aware data delivery in dynamic service chaining. In Network Softwarization (NetSoft), 2015 1st IEEE Conference on (pp. 1-5). IEEE.
[13] Akyildiz, I.F., Melodia, T. and Chowdhury, K.R., 2007. A survey on wireless multimedia sensor networks. Computer networks, 51(4), pp.921-960.
[14] Surendra Kumar, San Ramon, CA (US); Nagaraj Bagepalli, Fremont, CA (US); Abhijit Patra, Saratoga, CA (US); Paul Quinn, Wellesley, MA (US); James Guichard, New Boston, NH (US); JayaramanIyer, San Jose, CA (US), (Jul.30, 2015), US 2015/0215172 A1, CIS- TECHNOLOGY, INC., San Jose, CA (US).
[15] Belmiro Moreira, April 2014, “OpenStack training”.
[16] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I., 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the fifth utility. Future Generation computer systems, 25(6), pp.599-616.
[17] Son, J., Dastjerdi, A.V., Calheiros, R.N., Ji, X., Yoon, Y. and Buyya, R., 2015, May. Cloudsimsdn: Modeling and simulation of software-defined cloud data centers. In Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on (pp. 475-484). IEEE.
[18] A. Kapadia and N. Chase, Understanding OPNFV: Accelerate NFV Transformation using OPNFV. 2017.