IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

Edge distributed cloud middleboxes

Resma K. S.
resma.vin@gmail.com
PES Bangalore South Campus, Bangalore, Karnataka

Dr. Sharvani G. S.
sharvanigs@rvce.edu.in
R V College of Engineering, Bangalore, Karnataka

ABSTRACT

The cloud and the related technologies are growing in importance and technology. The Network Function Virtualization is a very important domain of the cloud platform. It provides network abstraction. This paper introduces the NFV and the existing implementations of the virtual middleboxes in the current scenario. The paper also proposes a new implementation of virtual middle boxes. This proposal focus on the distribution of the virtual middlebox managers to the edges of the network. The advantages of such a system is also illustrated.

Keywords*— Network function virtualization, Virtual middleboxes, Network middlebox manager, MANO, Network edges*

1. INTRODUCTION

Cloud native technology is becoming a necessity in most of the network applications. The need to become technically competent makes it important to learn and upgrade in cloud technologies such as Network Function Virtualization [4], Software Defined Networking, Hypervisors, and Containers. Network function virtualization replaces the costly dedicated hardwires with generic servers that use software to provide a bunch of virtualized network functions otherwise called as Virtual Middleboxes (VMB) [1]. It decouples the control plane from the underlying hardware. The implementation of network functions is done on general purpose hardware blades called as COTS (Common Off the Shelf Components) The existing systems of the Virtual Middleboxes otherwise called as VMBs have either a centralized architecture or a distributed architecture wherein a Virtual Middlebox manager located at the core data center manages and controls all the VNF in the CSP/PoP network. Hence in this paper, a new system is proposed which is an edge distributed VMBM. This paper will explain to you what is Network Function Virtualization, how Softwarization, Virtualization, and Orchestration is achieved in NFV. The existing architectures such as the centralized and distributed architecture of the NFV is briefly introduced. This paper will propose a new architecture called Edge Distributed Cloud Middleboxes (EDCMB). This new architecture can bring in a lot of advantages in comparison to the existing architectures because of the VMBs located at the edge devices.

2. RELATED WORK

With the advancement of cloud-based technology, a lot of changes has happened with the domain of network middleboxes. The traditional networks were based on hardware-based middleboxes. They are now replaced with virtualized middleboxes. These were then deployed into cloud networks. There are many existing implementations of middle boxes in an NFV infrastructure. They are as follows:

(a) Centralized model: In the centralized model [5] figure 1, the complete virtual network functions are located at the data communication provider point of the present data center (CSP/PoP) and benefiting the customer location equipment. The VMBs are deployed using the existing networks. The carrier Ethernet is ideal for providing access to centralized VMBs from customer premises. The basic network equipment like switch/Router is placed at the customer premises. This framework is heavily dependent on network performance.

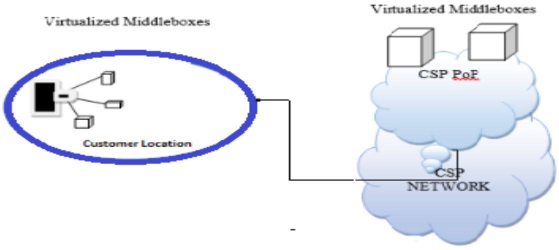


Fig. 1: Centralized model

The entire control of the VMBs is provided by the virtual middlebox managers located at the core data center network. This is a completely stateful implementation. The VMB is utilized from a single service provider. As a result of the performance fault tolerance and resource utilization etc. are compromised to some extent. An improvement to the centralized model is the decentralized model.

(b) Decentralized model: In figure 2, in a decentralized model virtualized functionality are located at the customer premises [5]. No VMBs are located in the Data center. Traffic handling and offloading are facilitated using hardware-based processing. The customer location requires augmented equipment.

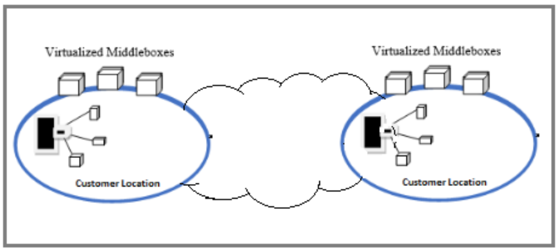


Fig. 2: Decentralized model

(c) Distributed Model: Distributes model [5] figure 3, is the one in which the virtualized middleboxes are located at both the customer premises as well as at the CSP/PoP. This model gives a better network performance and reliability compared to both the previous models.

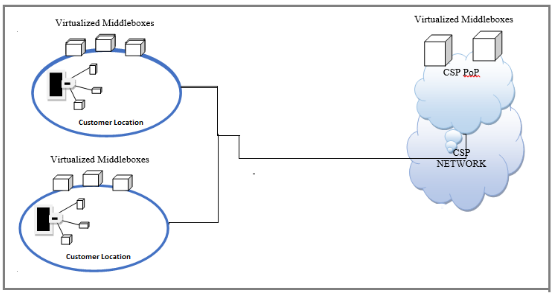


Fig. 3: Distributed model

3. NETWORK FUNCTION VIRTUALIZATION ARCHITECTURE

3.1 Introduction to NFV

Network functions virtualization (NFV) [5] is all about virtualizing network services which are traditionally run on proprietary, dedicated hardware. The concept was introduced by a group of service providers, working towards the acceleration of the deployment of the latest network services so as to increase their revenue and growth objectives. The drawbacks pertaining to hardware-based appliances led them to the application of standard IT virtualization technologies to their networks. With NFV, network functions such as Intrusion Detection System, Network Address Translator, routing, load balancing, and firewalls are converted as packages of virtual machines (VMs) on commodity hardware. Individual virtual network functions, or VNF/ VMBs, are an essential component of NFV [2] architecture offers new ways of designing, deploying and managing networking services. NFV decouples the network functions from the proprietary hardware so that they can be run as software. NFV is designed to consolidate and deliver the services of networking components such as virtual servers, storage, and even other network requirements in order to support a fully virtualized infrastructure. It will utilize standard IT virtualization technologies that run on high-volume services, switches and storage hardware to virtualize network functions. It can be applied to any data plane processing or control plane function in both wireless and wired network infrastructures. If the network functions get virtualized the cost gets reduced greatly and operations get simplified.

3.2 NFV design

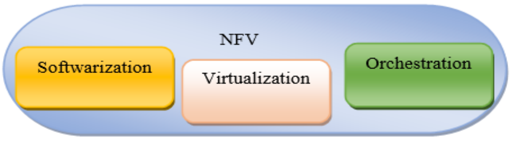


Fig. 4: NFV illustration

The 3 main components of NFV design (figure 4) are:

- Softwarization: Softwarization is the process of deploying a software application instead of traditional hardware. This helps the network to be more scalable, flexible, maintainable and cost-effective.
- Virtualization: Virtualization is the process of separating the operating system from the underlying hardware. It is also separating the application from the hardware. The virtualization results in improved resource utilization

- Orchestration and Automation: is the process of automatically programming the behavior of the network. It helps the network smoothly coordinates with the hardware and the software elements to support applications and services. Orchestration and automation help in the automatic allocation of network resources for the changing network policy.

3.3 NFV infrastructure

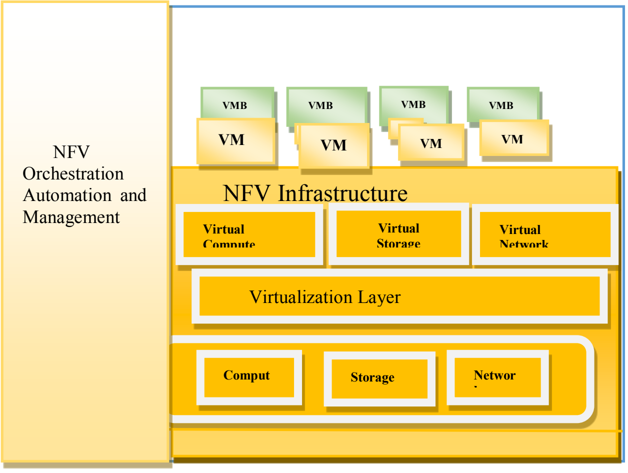


Fig. 5: NFV components

This is the layer which handles the hardware. It hosts all the computer storage and network components and it abstracts the virtual components for the consumption of the virtual machines. COTS-based hardware components or servers are deployed in this NFV infrastructure layer. In order to scale up these can be deployed in a bulk in multiple locations. Virtualized network function layer is the actual layer where we are going to host the applications. These applications are the virtual network function which is otherwise referred to as middleboxes. These middle boxes are run as software. A single VMB can be deployed over multiple virtual machines. We can host multiple application. For managing and controlling the entire piece. This layer controls the entire cloud. Now let's see how this cloud infrastructure works out is. All the middleboxes are deployed as software modules in VMB. The resources can be allotted to these virtual nodes on the basis of their requirements such as the storage, compute and networks. Hence depending upon the requirement of application the resource can be distributed and it differs for each individual component. The resource allotted for a particular node is not consumed by any other node. This allocation of resource is done by the virtualization layer which is situated in the infrastructure layer.

3.4 NFV architecture in detail

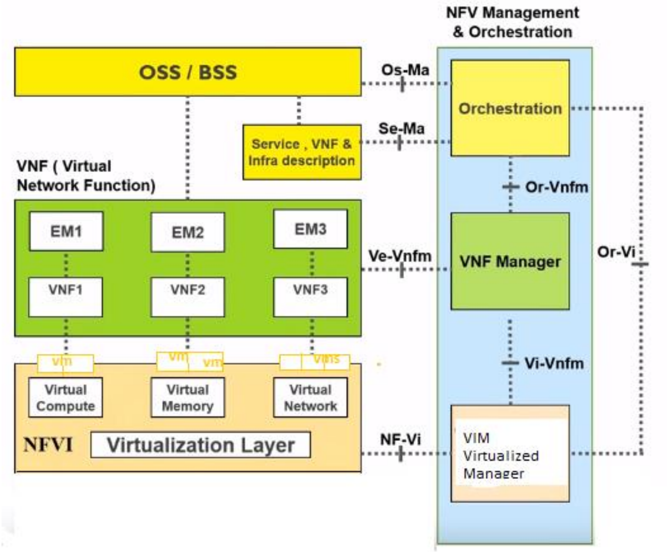


Fig. 6: NFV architecture (Referred from ETSI)

(a) Layer 3: NFVI and VIM

Now let us see NFV architecture in detail (Figure 6), to start with let us see with the last layer which consists of Network Function Virtualization Infrastructure (NFVI) and virtualized Infrastructure Manager (VIM). The role of NFVI is to host the hardware and manage the physical parts. Blades of generic hardware can be deployed. NFVI is a combination of a physical networking card which is NIC card for I/O, the computing, the storage resources exposed as common networking or NFVI. These resources can be at one place or it can be geographically located at different locations. This layer also contains a critical component by the name of hypervisor which is responsible for abstracting the physical resources to virtual resources. The hypervisor is the most important key to enable virtualization. It acts as a platform for the VMs to be created. It is a software installed on top of computer hardware creating a virtualization layer. Hypervisors are of two types: Type 1 hypervisors are also called as Bare Metal Hypervisors. These hypervisor has its own operating system and can be installed directly on the computer hardware. So it creates the virtualization layer on top of which virtual machines are created. Example EXSi server. Type 2 hypervisors are also called as Hosted Hypervisor. These type of hypervisor is a software application which can be installed on top of computer hardware. If the system is aided with any operating system such as Windows, Linux, etc. then the hypervisor can be installed on top of that operating system. That creates the virtualization layer on which the virtual machines can be created. Examples of hosted hypervisors are VM ware workstation professional, VM ware workstation player, VM ware fusion for the Apple IOS 10 operating system. The virtualization layer abstracts the resources so that they can be logically partitioned and provided to the VMB providing their specific network function. The VIM manages and controls the NFVI, it also manages the events of the NFVI which includes the virtual and physical parts of the resources.

(b) Layer 2: VMB and VMB Manager

This layer is the critical key component of the virtualization architecture. To understand it better some insight need to be put on network functions (NF) otherwise called as middleboxes. Middleboxes refers to firewalls, intrusion detection systems, and address translators, etc., which are traditionally hardware components tightly coupled with the underlying hardware, whereas the virtualized network functions are deployed as software applications and are decoupled from the underlying hardware. These NF can run on one or more virtual machines, based on the requirement of the applications the number of VMs can be increased or decreased.

The VMB manager is responsible for the things related to the FCAPS, the operations and the management of the VMB such as the monitoring, configuration and logging and all kind of alerting and performance related to monitoring, operation and maintenance of the VMBs. It also manages the lifecycle of VMB which includes the creation, deletion, and migration, etc. The FCAPS and ONM of the application such as a link down, KPI degradation, etc. are done by the EM (Element Management). This is specifically the operation of the application. ONM of the application means, if the link of any NF is going down, if there is any problem with the network middlebox application, then the alerting, linking and interfaces, etc. type of operations are done by the EM. EMs are Element Management. The EMs take care of the FCAPS operations. There are three layers of FCAPS operations:

- The first layer is the FCAPS of the NFVi and VMs or hardware it is done by VIM,
- The second one is the FCAPS or operations of VMB which includes everything related to the creation, operations, deletion and maintenance of the VF or middleboxes which is done by the VMB manager.
- The third layer is the FCAPS or operation of the application related to NF application virtualization. This is being done by the EM.

(c) Layer 1: Orchestration and Automation

Orchestration is the top most node, it is the key to any type of automation expected out of NFV. This is otherwise called as NFVO. This is the part of NFV framework. It helps in increasing the interoperability of SDN, resource orchestration, and network service orchestration. It is a central component of NFV architecture. It binds together different services to create an end to end resource coordinated service in a dispersed NFV environment. It will manage the global view and flow of resources, scale down and up of resources, hardware resources, keep track of individual VN functions are also taken care and the entire end to end network service creation. It makes sure that there is always adequate resources for the computer the storage etc. to provide service request by the user. The orchestrator has got the ability to coordinate, authorize, release the resources required by a particular flow. It takes cares about global governance in an automated way. Usually, there is only one orchestrator to take care of the entire NFV architecture.

4. PROPOSED SYSTEM ARCHITECTURE - EDGE DISTRIBUTED CLOUD MIDDLEBOXES (EDCM)

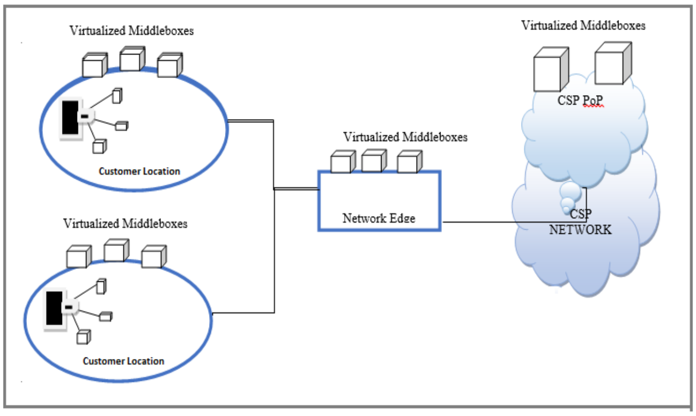


Fig. 7: EDCM Virtualized middleboxes distributed between Customer Location, CSP/PoP and network edges

Considering the disadvantages of these existing frameworks an Edge Distributed Cloud Middlebox (EDCM) is proposed, figure 7. In this architecture, the network middleboxes are distributed between CSP's PoP and Customer Location Equipment. The VMBs can be deployed based on optimal feasibility, performance, reliability, scalability and cost considerations. The VMBs can be deployed dynamically in an ordered, configured and chained manner as per the changing requirements. In this proposed architecture the network middleboxes will be distributed between the Customer Service Provider's Point of Presence, Cloud Edge and Customer Premise Equipment. An edge device is a hardware component that controls data flow at the network boundaries. The services provided by edge devices varies with the hardware implementations, but mostly they serve as network entry or exit points. The important functionalities of these edge devices include the transmission, routing, processing, monitoring, filtering, translation and storage of data passing between networks. With the evaluation of edge NFV, the edge devices are also moved into the virtualization environment. With this type of VMB models, the management model also needs to be modified.

4.1 Responsibilities of VMB manager

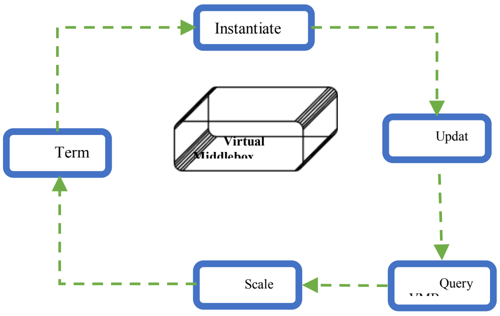


Fig. 8: Responsibilities of VMB manager

From figure 8, the role of VMB manager is very important in a VMB framework. The responsibilities of VMB Manager (VMBM) includes the instantiation and termination of VMB, the scaling, and healing of the resources, providing interfaces to the vendor specific element management module, VMB image management, and updating, etc. In most of the networks, the VMBM are found to be deeply coupled with other components. But in this proposed architecture they are intended to be loosely coupled and will have distributed implementation scope. VMBs are critical to realizing the business benefits outlined by the NFV architecture. The functionality of VMB is not autonomous, they require VMBMs. VMBMs are critical for instantiation, scaling, changing operations, termination of VMBs, updating and upgrading VMBs, adding new resources, communicating the states of VNBs to other functional blocks in the NMB-MANO architecture.

4.2 The triangular relationship

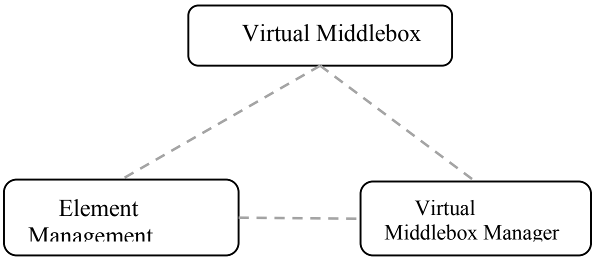


Fig. 9: VMB-VMM-EM

In figure 9, The element management manages the VMBs in coordination with VMB managers. The element management in coordination with the VMB manages the lifecycle of the VMBs. There exist a direct channel of communication between the VMB manager and VMBs for the management of VMB and Element management provides an indirect channel between VMBM and VMBs to manage the VMB. Element Management is more about FCAPS management. FCAPS stands for fault, configuration, accounting, performance, and security. The existing VMB are monolithic, with minimum reuse, they are stateful and possess complex orchestration. The proposed VMBs are going to be providing stateless services, ease of scaling and deployment, micro-services such as composability, reusability, etc.

4.3 EDCMB management architecture

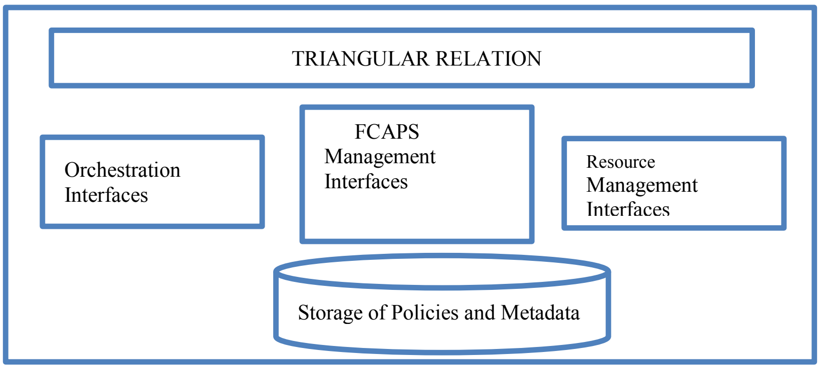

TRIANGULAR RELATION
Orchestration Interfaces
FCAPS Management Interfaces
Resource Management Interfaces
Storage of Policies and Metadata

Fig. 10: EDCMB architecture

The EDCMB figure 10, focus on the improvement of FCAPS management hence in the above architecture more emphasis is given to the attainment of fault management, configuration management, performance management, and security management. The responsibilities of the VMB manager is divided from the perspective of FCAPS, resource management, Storage and orchestration. These VMBMs should be able to provide services from multiple providers. It should be able to provide distributed security management at each node and resource allocated. The services provided by the VMB managers include the on-demand scale and scale out of resources, auto healing of the VMBs should be able to store and forward security policies monitoring policies and rules for the security up gradation. It should also be storing the keys and the certificates of the cryptography. Regular interaction with the orchestration interface in order to adapt to the policy changes.

4.4 Flow of management in EDCMB system

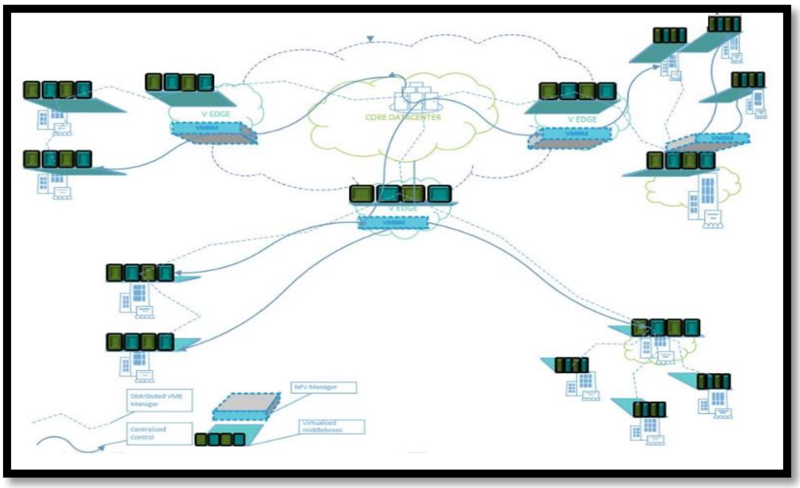


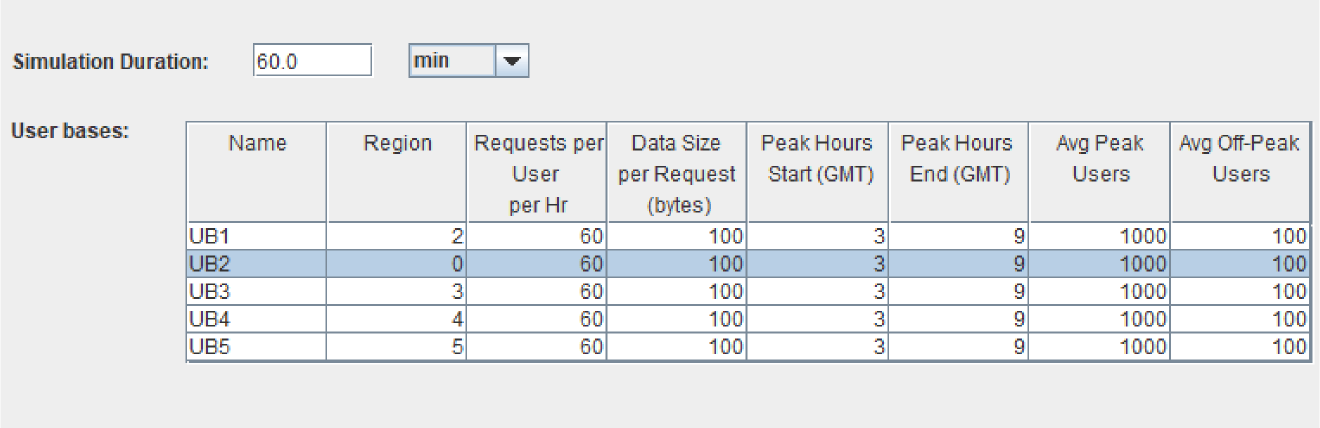
Fig. 11: Flow of control

In addition to the VMBM at the core data center, in the proposed EDCM system figure 11, the VMBM will be located at the edges of the network. These managers serve geographically closely located VMBs. This control flow is termed as distributed control. In addition to this, the centralized VMBM controls the entire clusters in communication with the core data center.

4.5 Experimental setup

For the experimental setup, the software platform used is eclipse OXYGEN.1 with Java programming language, and for the hardware involved is Intel® Core ™ i3 Processor with 6GB. The simulation was set for a duration of 1 hour, 5 different user locations. The number of request per user, data size for each request, start, and end of peak hours are all given in the snapshot (Table 1). Three data centers are created with the given configurations. In the experimental setup user grouping factor in user base is set as 10. Request grouping factor for the data center is made as 10 as well. Executable request length per request is made as 50 bytes. The load balancing policy used is round robin

Table 1: Main simulation configuration

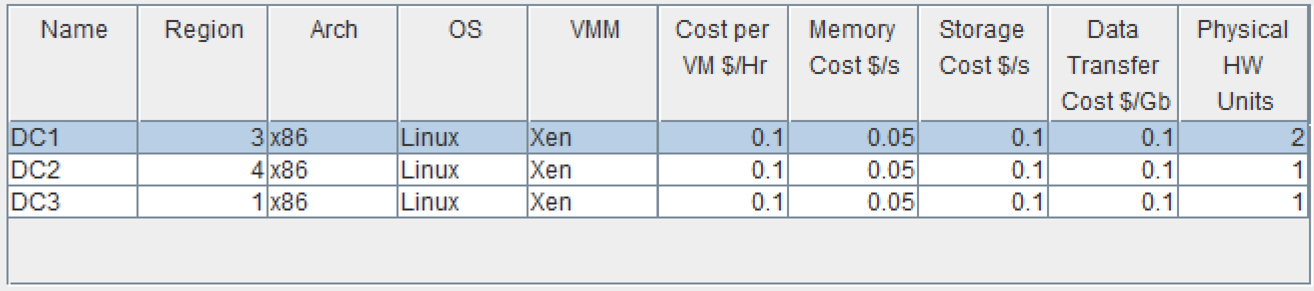


Simulation Duration: 60.0 min

User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	2	60	100	3	9	1000	100
UB2	0	60	100	3	9	1000	100
UB3	3	60	100	3	9	1000	100
UB4	4	60	100	3	9	1000	100
UB5	5	60	100	3	9	1000	100

Table 2: Data center configuration



Name	Region	Arch	OS	VMM	Cost per VM $/Hr	Memory Cost $/s	Storage Cost $/s	Data Transfer Cost $/Gb	Physical HW Units
DC1	3	x86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC2	4	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC3	1	x86	Linux	Xen	0.1	0.05	0.1	0.1	1

4.6 Results and discussions

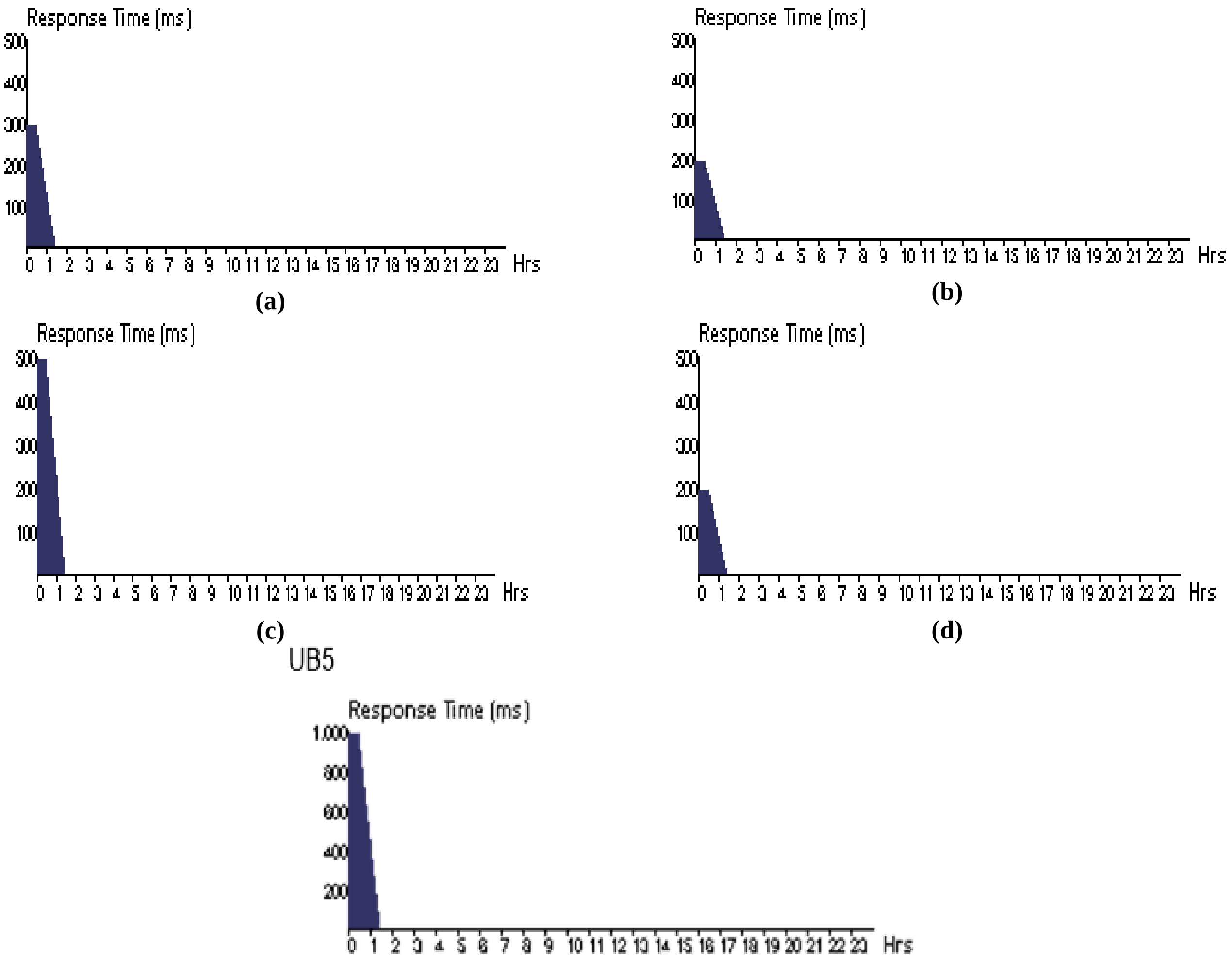
Graph figure 12, of the performance of the system in terms of average CPU load on decentralized cloud and edge, the distributed cloud is shown below. This graph is obtained from the values obtained from the table, in both the environments the load on CPU increases as the utilization on the system increases. In terms of the response time both the system, performance is compared. Table

1 is obtained by applying the topology in a dynamic simulator. The number of VMBs is increased gradually for both the models and the network performance is measured in terms of response time. The simulator [3] simulates these conditions and average performance is tabulated every 60 seconds. The table entries are converted to its graphical representation as in figure:

Table 3: System Performance

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	298.95	235.63	352.61
UB2	199.98	151.11	251.14
UB3	500.29	365.11	655.14
UB4	200.57	160.14	257.14
UB5	215.37	172.14	258.6

The below graphs shows the data center service timings for different user base UB1, UB2, UB3, and UB4 UB5 respectively



(a)

(b)

(c)

(d)

(e)

Fig. 12 (a-e): Data center request servicing times

4.7 ADVANTAGES OF EDCMB

(a) Provides efficient, localized and unique services for VMBs.
(b) Can utilize services from different VMBs from different VMB vendors.
(c) The implementation of VMB management across the trusted network domains.
(d) It can provide faster instantiation and increases resource utilization.
(e) The VMB software upgrade can be done in an agile fashion.
(f) Provides E2E automation of the resource allocation, recovery, scale up/down, modifications, etc.

5. CONCLUSIONS

The paper elaborated about network function virtualization and virtualized network middleboxes. We proposed an architecture for the implementation of middleboxes. The architecture is called as Edge Distributed Cloud Middleboxes abbreviated as EDCBM. In this architecture, the Virtual Middleboxes are placed at the edge devices in contrast to the existing systems with centralized and distributed architecture. A comparative study of these systems is performed using a stress test to analyze the CPU performance with varying number of middleboxes. The result shows that the implementation of such an architecture can improve the performance of the network. Hence improve the fault, configuration, accounting, performance, security (FCAPS) of the network.

6. REFERENCES

[1] Justine Sherry UC Berkeley Shaddi Hasan UC Berkeley Colin Scott UC Berkeley, Arvind Krishnamurthy University of Washington Sylvia Ratnasamy UC Berkeley Vyas Sekar Intel LabsMaking Middleboxes Someone Else’s Problem: Network Processing as a Cloud Service

[2] Ruozhou Yu, Guoliang Xue, Vishnu Teja Kilari, and Xiang Zhang,” Network Function Virtualization in the Multi-Tenant Cloud” IEEE Network • May/June 2015, 0890-8044/15

[3] Webserver Stress Tool - Performance, stress & load test. (2016). Paessler.com. Retrieved 15 August 2016, from https://www.paessler.com/tools/webstress

[4] Sridhar Pothuganti, Trinath Somanchi, Distributed VNF Management: Architecture and Use-cases - https://www.youtube.com/watch?v=r5RdTaqXhWI&t=211s

[5] Introduction to NFV Network function Virtualization Basics - NFV Architecture and ETSI - NFV MANO

[6] ETSI, N. (2014). GS NFV-MAN 001 V1. 1.1 Network Function Virtualization (NFV); Management and Orchestration.

[7] Amazon, Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2/, 2013.

[8] Basta A, Kellerer W, Hoffmann M, et al. Applying NFV and SDN to LTE mobile core gateways, the functions placement problem[C] Proceedings of the 4th workshop on All things cellular: operations, applications, and challenges. ACM, 2014: 33-38.

[9] Bolla R, Lombardo C, Bruschi R, et al. DROPv2: energy efficiency through network function virtualization [J]. Network, IEEE, 2014, 28(2): 26-32.

[10] Buyya R, Yeo C, Venugopal S, Broberg J, and Brandic I, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation computer systems, vol. 25, no. 6, pp. 599C616, 2009.