Automated Scaling of Virtual Network Functions

Anjali Shanbhog

Electronics and Communication Engineering
RV College of Engineering
Bangalore, India

Rohini S. Hallikar
Electronics and Communication Engineering
RV College of Engineering
Bangalore, India

Abstract— Using a combination of hardware (such as routers, access networks, content servers, etc.) and software, traditional network topologies produced certain network functionalities. With NFV, a variety of virtualized network tasks are provided by generic servers instead of expensive, purpose-built dedicated hardware (VNFs). This work talks about the concept of Network function Virtualization and its architecture. The importance of the LCM operations including the scaling operation is also discussed. An approach has been proposed to automate the scaling of VNFs to dynamically decrease or increase the number of VNF instances under times of high demand or very less demand.
In a try to finish the deployment of instances in the quickest time, the dynamic threshold algorithm would be used to control the number of instances that are deployed. The dynamic threshold has upper and lower limits that would change as and when the traffic varies and load increases or decreases.

Key Words— Network Function Virtualization, Auto Scaling, Virtual Network Functions, Network Function Virtualization Management and Orchestration

---

I. INTRODUCTION

Traditional network architectures create specific network functions using a combination of hardware for router, Access Network, Content server, etc. and software. A huge variety of dedicated hardware appliances are becoming more prevalent on the networks of network operators. Figure 1 shows the traditional Network Appliance approach.

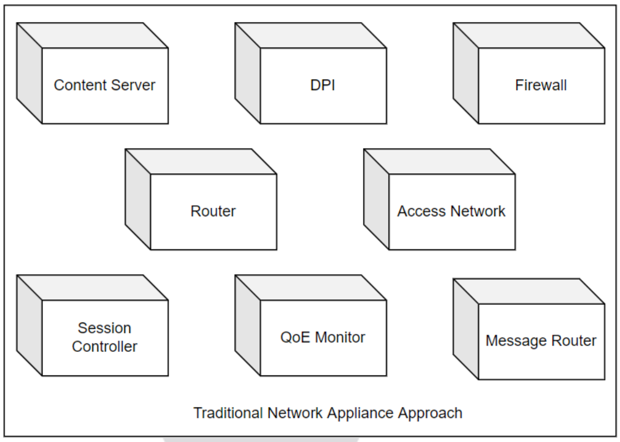

Figure 1: Traditional Network Approach

Finding the space and power to support these boxes is becoming more and more challenging as a result of rising energy costs, difficulties with capital investments, and a lack of skills needed to design, integrate, and manage increasingly complex hardware-based appliances. This difficulty is worsened by the need to launch new network services, which frequently require yet another variety.

Hardware-based appliances have a lifespan of five to ten years, which adds to the service provider's burden of going through the cycle of procuring, then working on the design, integrating and lastly, deploying with little to no revenue gains. As technology and service innovation speeds up, hardware lifecycles are getting shorter, which slows the rollout of new revenue-generating network services and limits innovations in a network-oriented world.
By combining various network equipment types on industry-standard big switches, volume servers and storage that may be deployed in, Network Nodes, datacenters and customers, Network Functions Virtualization solves these problems. A network with NFV approach has been shown in Figure 2.

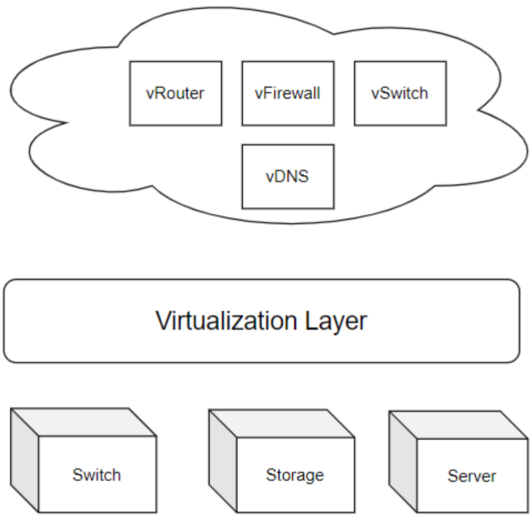

Figure 2: NFV Approach

II. LITERATURE REVIEW

[1] Introduces the idea of NFV by discussing the necessity of virtualizing network functions and how doing so helps network operators by lowering costs and power usage. Additionally, it describes how this permits the use of a single platform for a variety of operations. The authors of are confident that if a more standardised approach and common architecture are used, as this would aid in the interoperability of hardware devices, solutions to these technical challenges, such as obtaining high quality and efficient virtualised network devices that are compatible with various hardware vendors, managing different virtual network devices, and being sure that the network is safe and secure from network attacks, can be made available. In [2], the author focuses on achieving NFV deployment using a" Common Platform" approach across transport, core networks with a high-performance hypervisor and access transport. The approach the author has come up with helps in providing logical networking functions like routing, switching and firewalling. The key areas where virtualisation is important for customer service points which include Logical Routing, Logical Switching, Logical Load Balancing and Logical Firewall have also been highlighted here.
The authors of [4] survey the state-of-the-art in NFV and point up potential areas for further research in this field. State-of-art NFV Implementations have also been summarised in this paper which includes the functionality provided, the platform used and the driving standards. The common driving standard is the European Telecommunications Standards Institute (ETSI) which is an independent standardization organization in the field of information and communications.
The functional requirements for NFV Orchestrator, Virtualised Network Function Lifecycle management, functional requirements of VNF Management and Virtualised Infrastructure Manager, and the architectural level requirements have been described in detail along with the functional requirements of their operations and functions in [5]. The lifecycle management of network services is taken care of by the Network Function Virtualization Orchestrator(NFVO) which also performs the management and orchestration functions if VNFs. Authors of [3] also explain how the upcoming," telecommunications cloud" structure may be made up of different cloud operating systems and hypervisors and also the dynamic properties like reacting to adhoc changes of constraints and parameters. The significance of these changes in the future of telecommunications has also been talked about.

III. NFV ARCHITECTURE

NFV reduces the hardware usage and virtualizes it using a software layer. The hardware used in the architecture is the bare metal hardware that can be purchased from different vendors. The NFV Architectural Framework consists of these main components as shown below in Figure 3.

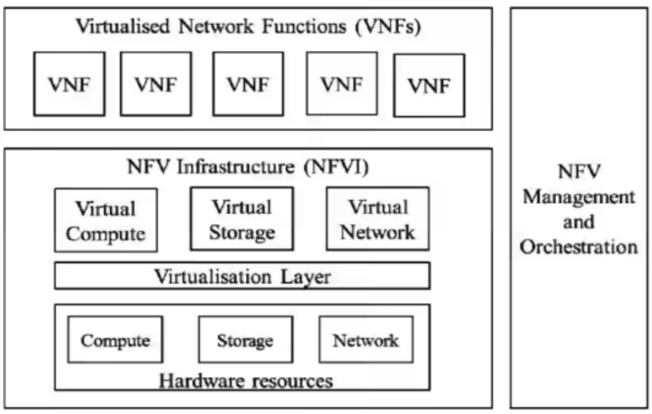

Figure 3: NFV Architecture

- NFVI: The resources needed to facilitate the implementation of the virtualized network functions are provided by this layer. Every NFVI block can be considered as a NFVI branch with numerous branches and one can install and manage numerous nodes from various places.

- VNF: VNFs are the central building block. The network function that can be operated over NFVI is implemented in software as a VNF. It is a bunch of software that runs in one or more virtualization containers which are typically Virtual Machines or VMs. Each VNF Component performs a set of tasks for the VNF. Each VNF that is running is called a VNF Instance. VNF Package contains software and additional files needed for the management and contains the VNF Descriptor. VNF Descriptor is an artifact defined by NFV and contains meta data that asks NFVO to plan and deploy a VNF.

IV. NFV MANO

NFV MANO: The NFV MANO looks after the orchestration and the lifecycle management of all the components which support the virtualization of infrastructure and of the Virtual Network Functions. The main functionality of MANO is to improve the management tasks that are virtualization specific and important for the NFV framework. Figure 4 shows the NFV Management and Orchestration architecture.

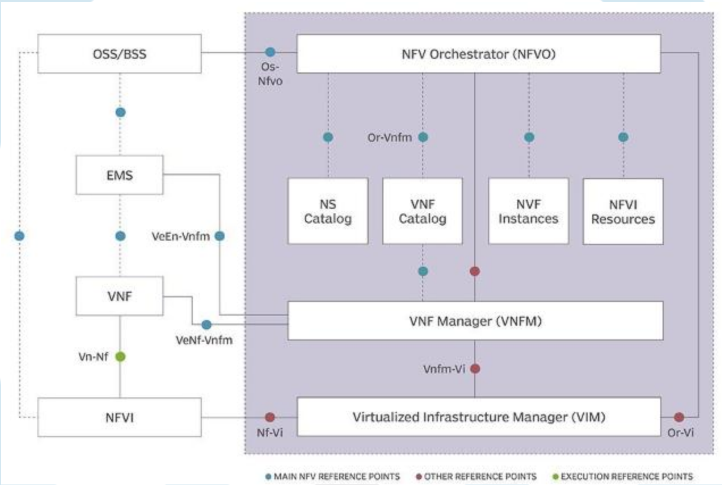

Figure 4: NFV MANO Architecture

NFV MANO consists of three main functional blocks. They are:

• Orchestrator: It is a primary element of the NFV MANO architectural framework. It is essential for the instantiation of the VNFs and network services. It performs resource and network service orchestration

• VNF Manager: Its job involves looking after or managing the lifecycle of VNFs. It is also involved in instantiating, scaling, and terminating VNFs

• Virtual Infrastructure Manager: Resources in the NFVI are managed by VIM. It also maintains an inventory of the VMs connected to physical Resources

V. LIFECYCLE MANAGEMENT OPERATIONS

Life Cycle Management Operations are the operations responsible for managing the VNF lifecycle control the distribution of virtualized resources to a VNF instance and/or changing the state of the VNF instance. Life Cycle management of a VNF controls processes to:

- Bring a VNF Instance into existence.
- Maintain and modify virtualization related aspects of the instance during the operation.
- Remove the VNF Instance from existence

  The NFVO manages the service and resource orchestration. VNFM asks permission from the NFVO to conduct the Life Cycle Management Operations. The NFVO lets the VNFM know in which part the resources can be assigned.

VI. SCALING OF VNFs

Scaling functionality changes the amount of virtual resources assigned to a VNF. Horizontal scaling deals with adding or deleting of virtual machines while vertical scaling deals with the upgradation of the configuration of a VM. Scale in, scale out functionality of a VNF adds or deleted the quantity of Virtual Machines which were previously escalated over a certain VNF. The purpose of scaling in is to maximize resource usage by removing extra capacity from a cluster by deleting one or more VNF instances. The network should have more VNFs instantiated than the minimum permitted instances. In manual scaling, the scale out or scale in operation is initiated by the ScaleVnfRequest and is met using ScaleVnfResponse.

VII. PROPOSED APPROACH

Auto scaling can dynamically increase or decrease the capacity according to varying demand, for example, compute resources. In order to preserve performance and manage the load, auto scaling can dynamically expand the number of VNF instances under times of high demand. It can assure that enough VNF instances are there to manage the load. To keep running VNF services performing, the deployed VNF must perform the scaling operations. For auto scaling of VNFs, the NFVO senses that the triggering conditions that are required to meet the scale out and scale in has been met.
The way in which the scaling is triggered can be explained using the following steps:

1. Check if the scaling is required
2. When a trigger is occurred, NFVO decides what action is to be taken
3. If the CPU usage is greater than a certain threshold (say >75%), a VM is added. If the CPU usage is less than a certain threshold (say <25%), a VM is deleted.

The below figure 5 shows a proposed architecture for auto scaling which monitors the CPU usage.

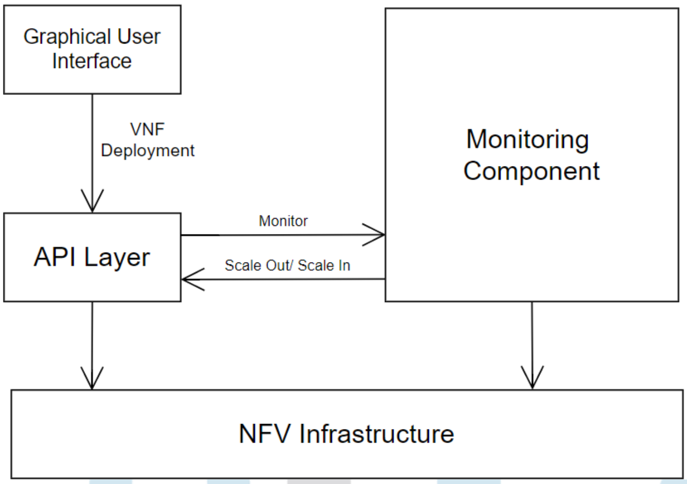

Figure 5: Proposed Architecture for Auto Scaling

A VNF can be deployed by the user from the Graphical user interface. The scaling operations are present in the API layer. The monitoring component monitors all the VNFs deployed by the API layer in the form of VMs The API layer shares the details of the VM that are to be monitored, to the monitoring component.The monitoring component checks if the CPU usage is according to specified threshold. If the CPU usage is less than a certain threshold (25%) or crosses a certain threshold (75%), the monitoring component will inform the API layer directly and the API layer triggers and performs the operations in the NFV Infrastructure. Whenever the API layer initiates operations due to high CPU consumption, the NFV Infrastructure creates the VM and performs the scale out operation. This helps in running the VNF services properly as the load would get distributed among the VMs.
The CPU usage can be monitored with respect to time as shown in Figure 6. If the CPU usage crosses the set threshold, the automated scaling would get triggered.

Figure 6: CPU usage vs Time graph

VIII. EXPECTED RESULT

Once the CPU usage exceeds the threshold, it is expected to scale out. Consider the Figure 7, which shows the initial condition, where the CPU usage is within the limits. The number of VMs can be seen under VNF Component which has been highlighted. Under the VNF Component, under the scale policy, only two VMs are present.

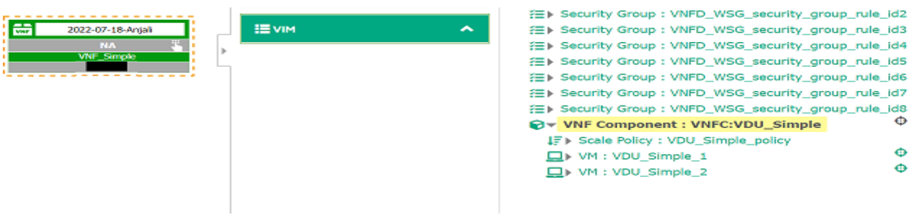

Figure 7. Before scale out operation

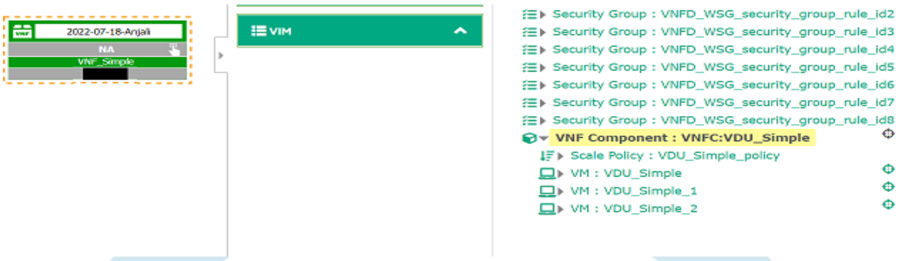

Figure 8. After scale out operation

In Figure 8, under the scale policy, three VMs are present. On increase in CPU usage, the number of VMs is expected to increase as shown in Figure 7 and Figure 8.

IX. CONCLUSION

This work talks about the NFV framework, its architecture and the MANO architecture as well, which is a part of it. The lifecycle management operations have also been discussed about in this work. Using the approach that has been proposed in this work, the user can modify the threshold according to their needs. This would help in managing the load and keeping the performance intact.

X. REFERENCES

1. White Paper." Network Functions Virtualization- Introductory White Paper" SDN and OpenFlow World Congress, 2012
2. Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Steven Latr´e, Marinos Charalambides, and Diego Lopez " Management and Orchestration challenges in networkfunctions virtualization " IEEE Communications Magazine, 2016
 White Paper. "Network Functions Virtualization (NFV) – Update White Paper" SDN and OpenFlow World Congress, 2013
3. ETSI "Network Functions Virtualization(NFV) Release 3; Management and Orchestration; Functional requirements specification" ETSI GS NFV-IFA 010 V3.6.1,2022
4. Srini Beereddy, Kiran Sirupa "NFV use case Delivering virtual CPE with multi-vendor VNF orchestration- 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), 2015
 White Paper. "Network Functions Virtualization (NFV) – White Paper 3" SDN and OpenFlow World Congress, 2014
5. ETSI "Network Functions Virtualization (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the management of NFV-MANO ETSIGD NFV-SOL 009, 2022
6. Bo Han; Vijay Gopalakrishnan; Lusheng Ji; Seungjoon Lee. "Network Functions Virtualization: Challenges and Opportunities forInnovations"IEEECommunicationsMagazine,2015
7. Dharma Rajan "Common Platform Architecture for Network Function Virtualiza-tion Deployments " 4th IEEE InternationalConferenceonMobileCloudComputing,ServicesandEngineering,2016
8. Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck "Network Function Virtualization: State-of-the-ArtandResearchChallenges"IEEECommunicationsSurveysTutorials,2016
9. Steven Van Rossem, Manuel Peuster, Luıs Conceicao , Hadi Razzaghi Kouchaksaraei, Wouter Tavernier, Didier Colle, ario Pickavet and Piet Demeester "A network service development kit supporting the end-to-end lifecycle of NFV-based elecom services " IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN),2017
10. Nikos Makrisy, Christos Zarafetas, Alexandros Valantasis, Thanasis Korakisy and Leandros Tassiulasz "Integrating NFV-MANO with Wireless Services: Experiences and Testbed Development" Development- IEEE Conference on NetworkFunctionVirtualizationandSoftwareDefinedNetworks(NFV-SDN),2018
11. Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba "nf.io: A file system abstraction for NFV orchestration-," IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN),2015

12. Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, Raouf Boutaba, Fellow, IEEE, and Otto Carlos Muniz andeira Duarte ”Orchestrating VirtualizedNetwork Functions” IEEE Transactions on Network and Service Management,2016

13. Montida Pattaranantakul, Ruan He, Zonghua Zhang , Ahmed Meddahi , and Ping Wang ”Leveraging Network Functions Virtualization Orchestrators to AchieveSoftware-Defined Access Control in the Cloud,” IEEE Transactions onDependableandSecureComputing,2021

14. OASIS,“TOSCA simple profile for Network Functions Virtualization (NFV) Version 1.0,”May 2017

15. R. Jain and S. Paul, “Network virtualization and software defined networking for cloud computing: A survey,” IEEE Communication Magazine., vol. 51, no. 11, pp. 24–31, Nov. 2013