Newer Approach for Implementing Network Services

Manan D. Thakkar[1], Rakesh D. Vanzara[2], Anand Mankodia[3], Amrutbhai N. Patel [4]
[1]Assistant Professor, Computer Engineering, S.P.B.Patel Engineering College
[2] Associate Professor, Information and Technology , Ganpat University-U V Patel College of Engineering
[3] Assistant Professor, Electronics & Communication Engineering , Ganpat University-U V Patel College of Engineering
[4] Assistant Professor, Electronics & Communication Engineering , Ganpat University-U V Patel College of Engineering

Abstract- *Network Function Virtualization (NFV) is an upcoming and revolutionary approach in network area that offers network services by means of software, which should be placed in a Virtual Machine or in a Data Centre. Ultimate goal of NFV is to transform architecture of network by using virtualization technique for many network devices. One complementary approach to NFV is that of Software Defined Network (SDN). Purpose of SDN is to separate network's control and forwarding planes to provide central view of network for network services. In day to day life, network operators suffer with many kind of problems like economy constraints, limited man power and problems in handling complex network infrastructure. NFV aims to solve such problems by virtually implementing the network functions like firewall, DNS, routing, by considering the basic function offered by routers, this paper presents the virtualization of routing functions.*

Keywords- NFV, SDN, NID, virtualization, virtual machines, Routing Function Virtualization (RFV), OpenFlow

I. INTRODUCTION

Now a day, many business areas face various constraints and economical limitations. Network designer and operator also have to face several problems other than technical problems like man power, economy constraint, number of network nodes etc. [1]. To handle such kind of problems there is a strong need of simple, dynamic, and easily manageable network and economic network solution for betterment of future networking technologies [1]. SDN and NFV are one of the most revolutionary approaches proposed since last few years. At this moment, there is no actual implementation of NFV concept, but it is considered as an approach that will be able to improve several network aspects [1]. Further, everyone talks about SDN, NFV as an emerged new idea that provides a new point of view of the management of networks [2]. In most modern approach of developing a network, Network Functions (NFs), also known as "middle boxes" [3], are playing most important role ranging from mobile networks, enterprise networks, to data-center networks. NFs play its role in improvement of network performance, improving security parameters or in traffic monitoring [3].

The objective of NFV is to transform the approach by which network operators can design their networks [4]. In reality, NFV is a new functional area which is not yet implemented. As per its definition, every network element is having a group of dissimilar functions which can be potentially extracted and implemented into external entities that can later be moved, instantiated, duplicated, and managed individually[5]. NFV and SDN are highly related but complementary approach, and independent of each other. Network designer and operators are highly interested in such a type of approach as it allows mechanism to develop and implement network services. However, NFV is still just functional concept, not yet implemented fully. SDN can be considered as one the most promising approach [6] that implemented in recent years. OpenFlow is a configuration protocol used to enable architecture of SDN [7]. At present, OpenFlow is used as most promising solution within SDN, which allows separation of the Control Plane from the Data Plane [8] and allows communication between controller and switch. OpenFlow controller provides centralized approach by connecting with multiple switches, and allows the management of all switches. This approach will open wide range of opportunity that can be implemented in different scenarios [7].

A. How NFV and SDN differ?

SDN: It separates the brain (control) of the network and forwarding (muscle) planes and provides a centralized approach of distributed network to provide more efficient network services [9].

NFV: Its main focus is on optimization of network services by decoupling the services like firewall, load balancing, DNS etc., from the hardware appliances and implementing their service by software at service provider's environment [9].

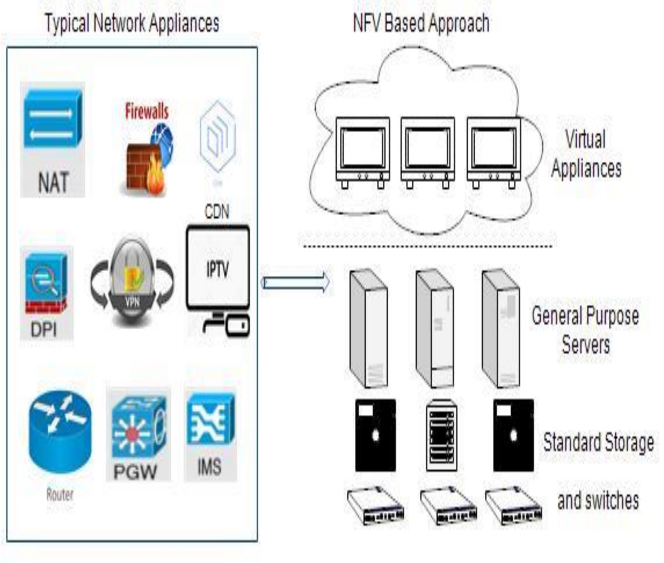

Figure 1. Network Services implementation by Traditional approach and by NFV [10]

Figure 1 shows the implementation of network services by virtualizing it. As shown in Figure 1, that the functions like NAT, Firewall, Router, etc. are implemented virtually through a different solution [10].

Functionalities of NFV can be implemented without incorporating mechanism of SDN. Although NFV and SDN are independent, if implemented together can potentially provide good solution [9]. NFV is able to incorporate SDN by providing infrastructure upon which the SDN can run. There is a strong need to understand, how NFV and SDN actually could work together. Figure 2 shows the traditional approach of how currently router service is implemented by using router at client side [9].

NFV can be applied to this situation and after the application of NFV, the traditional approach looks like as shown in Figure 3. The only thing that reside at customer site is a Network Interface Device (NID) for providing a boundary point as well as for measuring performance [9].

Finally, SDN is included which separate the control and data, as shown in Figure 4. Now, the data packets are forwarded by an optimized data plane, while the routing (control plane) function is running either in a virtual machine or in a rack mount server [9].

The rest of the paper is structured as follows: Design and Architectural view are explained in section II. Section III elaborates communication between controller and RFV. Finally, conclusion is presented in section IV.

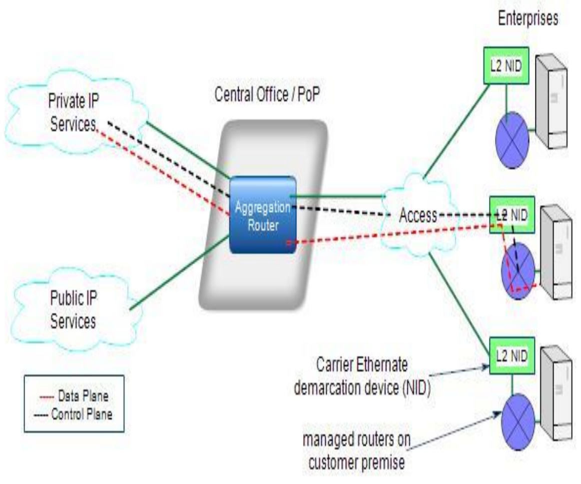

Figure 2. Traditional approach to provide router service [9]

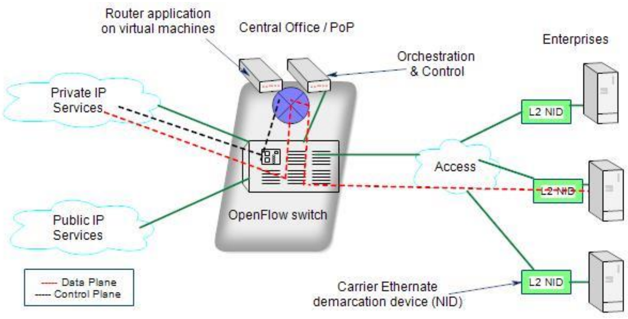

Figure 3. Managed router service using NFV [9]

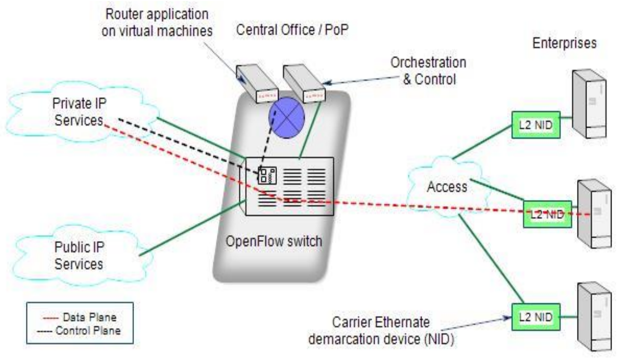

Figure 4. Managed router service using NFV and SDN [9]

II. DESIGN AND ARCHITECTURAL VIEW

Purpose of this paper is to present, how to implement NFV in an OpenFlow network. The function of router is virtualized by implementing its services in an external machine [11]. For this purpose there is a need to modify OpenFlow controller along with implementation of routing function [11]. Management of this service is done in a centralized manner which provide a distributed management. Apart from this, its task is to maintain a routing table of the network and provide routes to the switches. This routing function can be handled by using two techniques: proactive and reactive routing [11].

The first step for the implementation of NFV is to store or implement the functions of networks in virtual machine (VM). While designing, its proposal needs to be implemented in a Naas platform implemented in a VM, which offers some characteristics to provide virtualization function [11]. The things proposed by OpenNaas helps in the design and implementation of proposed prototype. After that, most important thing is the handling of the stored data for routing function. A normal routing table contains the source and destination address of the packet, along with information about next hop. After defining routing function with its database defined, the next step is to make data accessible from the OpenFlow Controller (OFC).The best way to make communication among OFC and routing database is by using a simple interface Representational State Transfer (REST) provided by Web Services [11].

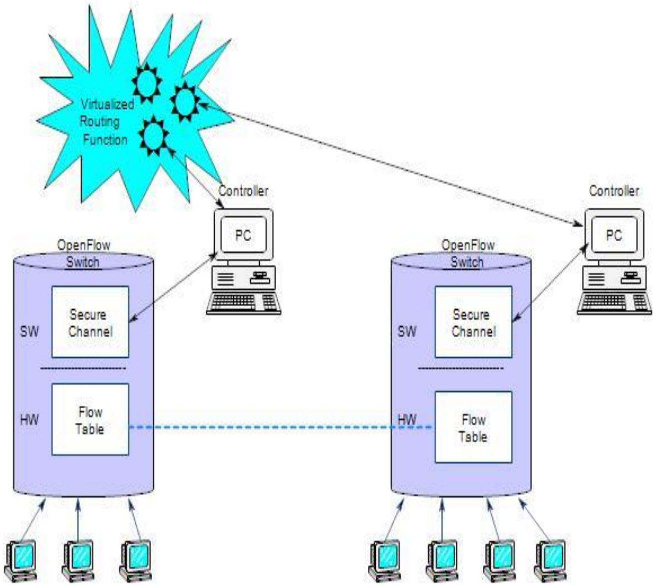

Figure 5. Architecture [11]

The experiment over here is between two hosts located in different networks. To make this communication there is need to make communication between three elements: the switches, the controllers and the routing machines [11]. The switches are enabled by OpenFlow and connected with OFC. The managements of unknown packet whose address done not there with switch, at this moment it is handled by OFC [11] the original image, lower left corner of the vertical details, upper right corner of the horizontal details, lower right corner of the component of the original image detail (high frequency). You can then continue to the low frequency components of the same upper left corner of the 2nd, 3rd inferior wavelet transform.

Using this architecture the control is centralized in the routing machine and the function of each controller is reduced [11]. This centralized function can be implemented in different VMs, and can be distributed through the network. Each virtual machine that contains the routing function can manage different routing tables and several routing algorithms [11].

Figure 4 depicts the architecture of this proposal. By using OpenFlow, the control plane is separated from its data plane by using the mechanism of SDN [11]. The data plane only manages the packets that match with an entry in the Flow Table of the switch. The packets which do not match with any flow entry of data plane are transferred to the controller using the OpenFlow protocol with a Secure Socket Layer (SSL) [11].

Mininet [12] is a software package which allow to design an OpenFlow network and test the behaviour of the network. This package includes several examples of scripts, easy to use and at this moment is the best way to develop and add new features to an OpenFlow network [12].Each network designs in Mininet is done by a script file which contains the definition of the network with the IP of the remote controllers, IP of the defined hosts and the MAC address of each switch. In addition, this definition contains the connection between all of these elements (e.g. switch connected to a controller, hosts connected with a switch) [12].

Each host created in Mininet is defined in a Linux Containers (LXC) which provide processes (and groups of processes) with exclusive ownership of interfaces, ports, and routing tables (such as ARP and IP) [12].

III. COMMUNICATION BETWEEN THE CONTROLLER AND RFV

To implement this communication, new protocol is needed, which makes the communication between the OFC and the routing machine. This protocol is responsible for the fast response in order to avoid more delay. This protocol defines the message exchange, their header and when they should be used [11]. When a switch needs a route and consult the routing machine, the packet goes to the controller and the controller needs to make a petition to the routing machine [11].

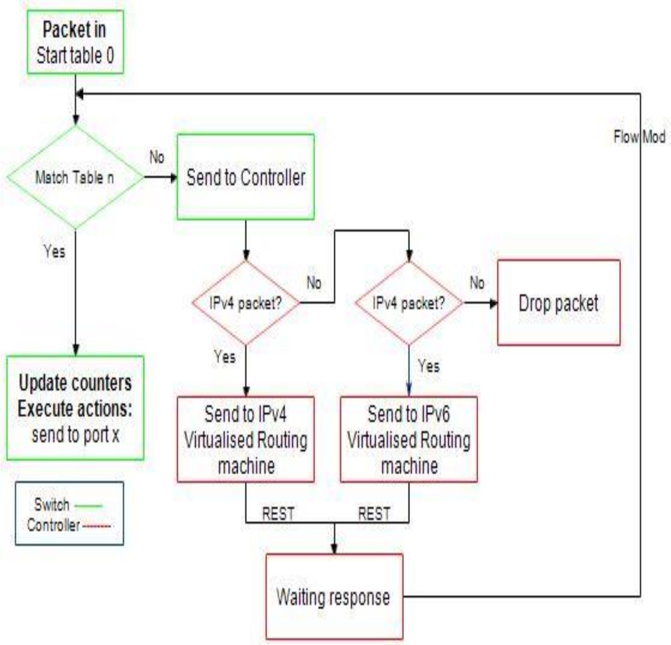

Figure 6. Flow of Communication between Controller and RFV [11]

IV. CONCLUSION AND FUTURE WORK

The basic idea behind this paper is to explain the virtualization of the routing function between two hosts. Purpose of this is to simplify the routing function and to avoid overhead in order to find the best route. From this point, it can become interesting to implement the basic network functionality which router or switch offers in a virtualized manner. Apart from that it will help to provide centralized control for managing all routers, which will simplify the requirements of man power, will reduce economic constraints. It will also reduce hardware component requirements by implementing its function virtually. Also it is important to keep in mind the imminent utilization of IPv6 in place of IPv4 and the migration to this technology. It presents a virtualization of routing function as a proposal of NFV. It also shows the contribution in network area by using scenario regarding how routing function can be implemented in OpenNaaS platform and using OpenFlow as a SDN.

As still implementation of network functionality and developing network architecture virtually is awaited to deploy and use, so first need to implement this concept and get results as we are achieving at this moment with physical implementation and then after to look for betterment of performance. There is need to design efficient routing protocol in order to make quick communication between controller and RFV.

REFERENCES

[1] ONF Market Education Committee: Software-defined Networking: The New Norm For Networks, ONF White Paper (2012)

[2] White paper on "Network Functions Virtualisation". http://portal.etsi.org/NFV/NFV_White_Paper.pdf ,2012. [Onlineaccessed : 30th Dec-2016]

[3] Li, Xin, and Chen Qian. "A survey of network function placement." 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2016.

[4] Guerzoni, R.: Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action, Introductory White Paper, SDN and OpenFlow World Congress. 2012

[5] Eric Keller, Soudeh Ghorbani, Matt Caesar, Jennifer Rexford: Live Migration Of An Entire Network (and its hosts). In Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, pages 109{114, New York, NY, USA, 2012. ACM

[6] O. M. E. Committee. Software-de_ned networking:The new norm for networks. Open Networking Foundation, 2012.

[7] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner: Openow: Enabling Innovation in Campus Networks, SIGCOMM Comput. Commun. Rev., 38(2):69{74, March 2008}

[8] Liu, Jiang, Juan Liu, and Renchao Xie. "Reliability-based controller placement algorithm in software defined networking." Computer Science and Information Systems 00 (2016): 14-14.

[9] https://www.sdxcentral.com/articles/contributed/nfv-and-dn-whats-the-difference/ 2013/03/[Online accessed, 23-Dec-2016]

[10] Gladisch, Andreas, and Wolfgang Kellerer. "Software defined networking and network function virtualization." it-Information Technology 57.5 (2015): 265-266.

[11] Josep Batalle Oronich: Experimentation on Virtualized Routing Function Migration Using OpenFlow" –Master report in Proceedings of the 2013 Internet network management conference on Research on enterprise networking, IEEE Communications Magazine, vol. 51, no. 2, pp. 114–119, 2013

[12] Bob Lantz, Brandon Heller, and Nick McKeown: A Network In A Laptop: Rapid Prototyping For Software Defined Networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets-IX, pages 19:1{19:6, New York, NY, USA, 2010 ACM