Software-Defined Network Function Virtualization-A Survey

Arkajit Dhar

November 9, 2018

Abstract

There have been a lot of work in segregating the Network Functions from its proprietary network hardware appliances. Network Function Virtualization (NFV) allows the Networks Functions to be provisioned as a virtualised software commodity. This facilitates the non - dependability of the network functions on proprietary hardware servers thus reducing the capital expenditures (CAPEX) and the operation expenditures (OPEX). On the other hand,Software Defined Networks (SDN) allows the network controlling logic to be centralised and moved out of the forwarding devices such as switches and routers. It helps in traffic steering via enabling network programmability. This paper gives a brief overview of the architecture of SDN coupled with NFV architecture and also discusses few challenges on that regard.

1 Introduction

In current networks, the network functions such as firewall, Load Balancing , Denial of Service, routing are all tied to dedicated hardware (servers) systems. These hardware systems thus require huge overhead to maintain them and also scale-up/down as and when needed. Such overhead induces huge cost in both capital as well as time.In order to address this issue Network Function Virtualization has emerged as the methodology to virtualize every network functions and untie it from its hardware system. This way the Network function could be seen and used as a hardware independent software which could be deployed any time as per requirement. These softwarised Network Functions could also be dynamically controlled and centrally maintained so as to steer the data traffic through each of them as per user / network requirements. This dynamic and central control is facillitated by Software Defined Network infrastructure.

Yong Li et al. [3] lists out the main difference between NFV and SDN as follows:

NFV is the concept of implementing Network Functions as software , whereas SDN is the concept of achieving flexibility in centrally controlled network resources.

NFV decouples the network functions from the hardware , whereas SDN decouples the network control plane from the data plane to create a centralized controller through enabling programmability.

NFV allows the Network Functions to virtualized and deployed as a software. It facilitates the notion of Commodity Off The Shelf (COTS) which reduces the huge CAPEX and OPEX required to maintain a dedicated hardware. However, Network Functions deployed as softwares would still not able to optimally utilize the network resources dynamically according to user requirement, if the data flow is not dynamically or programmatically driven. This condition motivates the integration of NFV architecture with SDN so that multiple Network Functions could be deployed on the network along with the forwarding devices so as to dynamically change or modify the flow as per user requirements.

The paper starts with a background and brief concepts of SDN and NFV laying a foundation for the applications and usefulness of integrating these two concepts together in later chapters. Finally couple of important challenges which could be faced are also laid out in configuring and building such architecture.

2 Concepts: NFV and SDN

2.1 Network Function Virtualization

Network Function Virtualization (NFV) decouples the network functions, such as network address translation (NAT), firewalling, intrusion detection, domain name service (DNS) etc from proprietary hardware appliances so they can run as software. These virtualized network functions run on Hypervisors, which abstracts the Physical hardware on top of which it runs. Thus from maintainability perspective VNFs can be added , moved or changed from one server to another without any manual work. A VM can be spawned in any physical server to host VNF or multiple VMs can be spawned in a single server to host multiple VNFs.

According to [2] the NFV architecture consists of the following components :

Vitualized Network Functions - These are the softwarized network functions which are capable of running on VM hosted on a server through Hypervisor.

NFV Infrstructure (NFVI) - This consist of the various physical resources (storage, cpu, network) on top of which an Hypervisor resides which abstracts these hardware resources to the NFVs.

NFV Management and Orchestration - This manages and orchestrates the lifecycle of both the physical and software resource that are part of the NFV architecture.

2.2 Software Defined Networking

Software Defined Networking aims at decoupling the network control layer from the dataplane layer via direct programming.SDN, as described by Haleplidis, et al.[1], comprises several abstraction layers and distinct planes. Planes refer to the collection of functions and resources that relate to the same functionality. Abstraction layers refer to the abstraction of resources of specific planes and interfaces refer to the APIs between planes

Haleplidis, et al.[1] distinguishes Network Planes into 5 distinct planes:

Forwarding Plane (FP) - Includes resources across all network devices responsible for forwarding packets.

Operational Plane (OP) - Involved in managing the operational states (active or inactive) of network devices.

Control Plane (CP) - Involves the control and directs the networking devices how to forward the packets.

Management Plane (MP) - Involves monitoring and managing and configuring the network devices.

Application Plane (AP)- Involves the applications and services that define network behavior.

Haleplidis, et al. [1] distinguishes Network Layers into 4 distinct layers:

Device and resource Abstraction Layer (DAL) - The device's resource abstraction layer based on one or more models. If it is a physical device, it may be referred to as the Hardware Abstraction Layer (HAL). DAL provides a uniform point of reference for the device's forwarding- and operational-plane resources.

Control Abstraction Layer (CAL) - The control plane's abstraction layer. CAL provides access to the Control-Plane Southbound Interface.

Management Abstraction Layer (MAL) - The management plane's abstraction layer. MAL provides access to the Management-Plane Southbound Interface.

Network Services Abstraction Layer (NSAL) - Provides service abstractions that can be used by applications and services.

3 SDN with NFV architecture

SDN along with the virtues of virtualized Network Functions drives great flexibility in provisioning Network services programmatically thus reducing the servic- providing cost and maximising network resource utilization. Software Defined NFV architecture mainly includes a control module , NFV Platform and forwarding devices. Control Module determinces the logic of packet forwarding handled by the SDN controller and also performs the orchestration of the Network Functions in the NFV platform. Network Functions are implemented on the virtual machines running on commodity servers via the Hypervisor. This platform architecture allows the flexibility of programmable forwarding data devices processing functions such as middleboxes of firewalls, NAT which are running as pure softwares in virtual machines.

SDN with NFV model derives it greatest potential in automating the Network Service Chaining or Service Function Chaining (SFC). In traditional networks ,Network Service Chaining would typically involve network administrator to connect up a series of physical middleboxes such as firewalls to process incoming and out going network traffic, involving a number of manual steps which is tedious , error prone and clumsy. With SDN dynamic traffic steering capability becomes easy. NFV allows for the hardware based middleboxes implementing network services (firewalls, NAT etc) to be virtualized and deployed as a software decoupled from the hardware.

SFC and SDN network service chaining standards are being developed in several industry groups. The European Telecommunications Standards Institute (ETSI) has a

service architecture that uses network forwarding graphs to route traffic between virtual network functions (VNFs) with a network service header.

4 Challenges

Although an active research topic, Yong Li [3] characterizes the following challenges in the SD NFV architecture model till date.

Function virtualization- The virtual machines and the hypervisors which implements the Network Functions are not highly optimized for the processing. Thus obtaining high performance from the standard servers is still a main challlenge for function virtualization.

Function Placement and Traffic Steering - In Software Defined NFV traffic steering is jointly optimized with Network Function deployment. This unified optimization creates a difficult situation as more variables are introduced. These two seperate optimization problems should be tackled together using heuristic algorithms such as mixed integer linear problem (MILP) [4].

Ali Mohammadkhan et al. [4] explains about the problem: A network function could be dynamically instantiated in a vm to host a service courtesy SD NFV flexibility. So a traffic flow can be steered through a switch and the network function and executing the service as and when required. This could lead to a flow to traverse a given link multiple times. Thus the placement of NFs has to be done judiciously such that a flow starts from a entry switch , goes through all neccessary NFs and routing of flows, and ends in a exit switch.

5 Conclusion

Software Defined NFV is still an area of active research. In this paper the basic idea behind the SDNFV architecture have been thoroughly discussed. It has also been shown Network Sevice Chaining as the main application area of Software Defined NFV. Software Defined NFV challenges were also presented.

References

[1] S. Denazis E. Haleplidis and D. Meyer. *Software-Defined Networking (SDN): Layers and Architecture Terminology*. 2015.

[2] ETSI. *Network Functions Virtualization (NFV) Architectural Framework*. 2013.

[3] Yong Li and Min Chen. *Software-Defined Network Function Virtualization: A Survey*. 2015.

[4] Ali Mohammadkhan Sheida Ghapani Guyue Liu Wei Zhang K. K. Ramakrishnan and Timothy Wood. *Virtual Function Placement and Traffic Steering in Flexible and Dynamic Software Defined Networks*. 2015.