University of Colorado
Interdisciplinary Telecom Program
TLEN 5700

Project Definition Document

SDN/NFV VNF Service Chaining

Approvals

	Name	Affiliation	Approved	Date
Customer	Dr. Levi Perigo	CU Boulder ITP		
Customer	Brooke Mouland	Equinix		
Course Coordinator	Dr. Kevin Gifford	CU Boulder ITP		

Project Customers

Name: Brooke Mouland Address: 1225 17th St #1690, Denver Email: bmouland@equinix.com	Name: Dr. Levi Perigo Address: ECOT 312, CU Boulder, Boulder Email: levi.perigo@colorado.edu

Team Members

Name: Dashmeet Singh Anand Email: dashmeet.anand@colorado.edu	Name: Hariharakumar Narasimhakumar Email: hariharakumar.narasimhakumar@colorado.edu
Name: Sarang Ninale Email: sarang.ninale@colorado.edu	Name: Rohit Dilip Kulkarni Email: rohit.d.kulkarni@colorado.edu

1.0 Problem or Need

Service Chaining or Service Function Chaining (SFC) is a capability that uses Software Defined Networking (SDN) to define a chain of network services (such as Firewall, NAT, QoS and many more) [1]. These network services are decoupled from traditional networking hardware and written in software (known as Virtual Network Functions - VNFs) to support a fully virtualized infrastructure. Software Defined Networking (SDN) can be used to direct traffic to different service chains based on the traffic type. For instance, an email service could use chains consisting of virus and phishing detection services, and video traffic could be directed to a chain consisting of video optimizer and parental control services. Even though the ability to connect these network services is well understood, there is not enough data on the performance characteristics of chaining multiple VNFs.

Equinix has had a prominent footprint in providing Data-center Solutions and Internet Exchanges between leading networks through their availability. This “SDN/NFV VNF Service Chaining” project aims at testing the performance characteristics of different service chains of both vendor and open source VNFs. This project will test the most optimal order of linking these VNFs together and any throughput and performance losses incurred in adding a VNF to the chain. This project shall stress different service scenarios, which Equinix houses, to fully understand their virtualization capabilities and provide catalogs for their customers.

2.0 Previous Work

Traditionally, special network functionalities such as Network Address Translation (NAT), traffic filtering firewall, packet inspection or manipulation, intrusion detection, etc. were implemented in the form of hardware devices called Middleboxes. These devices offered new features, improved security, and performance [2].

With the highly dynamic network requirements that exist today, the deployment and operation of these middleboxes come at the expense of increased capital and operational costs. To overcome this blockade, Network Function Virtualization (NFV) was seen as a scalable software solution that can virtualize the hardware middleboxes. [3] used the concept of NFV to replace middleboxes with commodity servers running software applications known as Virtual Network Function (VNF).

VNFs have an inherent advantage over their physical counterparts (Middleboxes) as they are easy to deploy, modify and upgrade. [4] focused on the use of VNF to chain network services together and analyzed different possible locations for the services to obtain optimum performance.

Since the advent of VNF based service chaining, further research helped maximize the performance of the chain. [5] implemented VNF service chains using containers to reduce the kernel overhead introduced into the system due to Virtual Machines. The main advantage of container-based VNFs over Virtual Machines is that containers have access to the kernel and operate on a single OS instance.

To our knowledge, there is no existing research literature that provides experimental results on the order of VNFs in a service chain or quantitative analysis of its throughput and performance characteristics. This project will help fill this gap.

3.0 Specific Objectives

Level 1: Create service chains with Virtual Network Functions (VNFs) from various vendors and open source network services required by customer. Create specific and well-defined test cases to test the service chains for throughput and performance. The service chains will be subjected to traffic with varied parameters such as packet size, TCP, UDP, unidirectional, bidirectional traffic, number of VNFs in the service chain and many more.

Level 2: In addition to Level 1 criteria, various combinations and sequences of different services in the chain will be tested in a consistent environment. Each service chain will be evaluated based on two main parameters; throughput and performance (in terms of memory and CPU utilization). Create an abstraction layer to plug-in and test various combinations of service chains with only the knowledge of inputs, outputs and functionality.

Level 3: In addition to Level 2 criteria, data obtained as a result of test cases in Level 2 objective will be stored in the database. An open source tool will be used to query, alert and create a visual representation of the data available in the database. This visual representation will highlight the performance of various service chains.

4.0 Functional Requirements

Multiple VNFs running on hypervisors will be linked to form a service chain. These hypervisors will be hosted on x86 architecture devices. A testing tool will be used to generate traffic with various parameters. This traffic will be input to the service chain and the test results will be stored in databases. Additionally, data analysis tools will be used to provide insights into the performance characteristics of various service chain combinations.

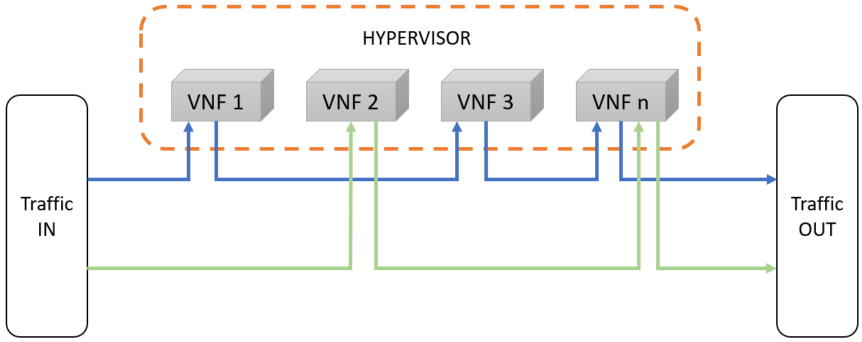

Figure 1: Functional Block Diagram (FBD)

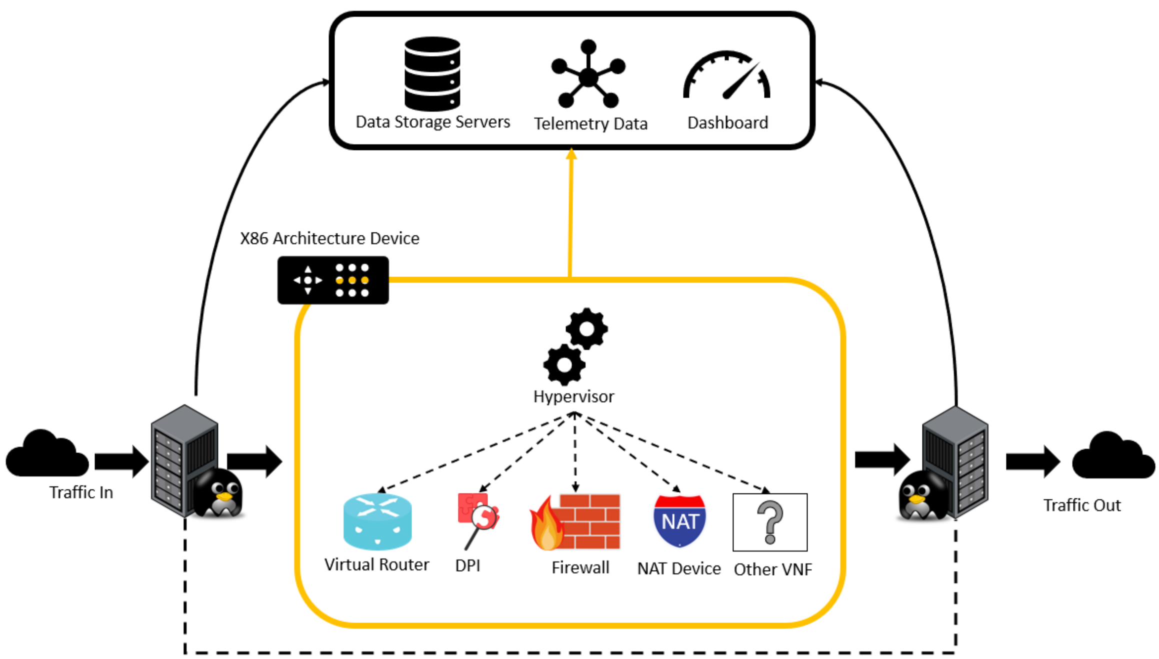

Figure 2: Concept of Operations (CONOPS)

5.0 Critical Project Elements

Technical:

<u>CPE.1.1:</u> Create multiple service chain combinations with VNFs implemented in VMs run on one or multiple hypervisors.

CPE.1.2: Build a consistent test environment and identify test cases. The test environment must remain constant with respect to the order of VNFs and test parameters. This ensures that the test results can be reproduced for a particular combination of VNFs.
CPE.1.3: Save test results in a database and perform data analysis for evaluating performance. The telemetry data generated by the testing tool is sent to a database. Performance monitoring will be done by querying the database and will be used to create a dashboard.

Logistical:

CPE.2.1: Make use of physical hardware devices and software applications to setup servers that will be used to store and analyze data during service chaining.
CPE.2.2: Equinix to provide VNF images, management platform, hypervisor and access to their User Interface (UI) and Application Programmable Interface (API).
CPE.2.2: Gain extensive knowledge of Service chaining, VNFs and Software-Defined Networking to create effective service chains. Get familiar with a testing tool to effectively test the performance and throughput of service chains.

6.0 Team Skills and Interests

Dashmeet Singh Anand: Interned as a Network Operations Engineer at Amazon Web Services. He has prior work experience in Software testing. He is proficient in traditional Routing and Switching technologies and has good hands-on experience with Python. His interest lies in Network Design, Development and Automation.

Hariharakumar Narasimhakumar: Hari interned as a Network Engineer at Facebook. He has experience in Router/Switch technologies and Python Programming. Interested in automation, virtual network functions and Software-Defined networking aspects of the project.

Rohit Dilip Kulkarni: Holds experience as a Network Engineer for Telecomm Service Provider and Interned with Amazon Web Services under Network Operations Team. Keen in learning Network Automation and Virtualization, he is knowledgeable in Python programming and traditional Routing and Switching. His interest lies in Network Development, Scalability and Design.

Sarang Ninale: Interned as Technical Operations Engineer at Jump Trading LLC. Experience in Linux System Administration, Network Automation using Python Programming and Development and Operations. Interests lie in Network Virtualization technologies, Software-Defined Networking and Low-latency Networks.

Critical Project Elements	Team members(s) and associated skills/interests
1.1 Service chain implementation	Hari, Sarang
1.2 Test case creation	Hari, Dashmeet
1.3 Test results storage and processing	Rohit, Sarang
2.1 Software Requirements	Rohit, Dashmeet
2.2 Knowledge gap	All team members

7.0 Resources

Critical Project Elements	Resource/Source
1.1 Service chain implementation	Levi Perigo
1.2 Test case creation	Levi Perigo, Brooke Mouland
1.3 Test results storage and processing	Levi Perigo, Brooke Mouland
2.1 Software Requirements	Levi Perigo, Brooke Mouland
2.2 Knowledge gap	Levi Perigo

8.0 References

[1] "What is Network Service Chaining? Definition", Available: https://www.sdxcentral.com/sdn/network-virtualization/definitions/what-is-network-service-chaining/ [Accessed: 24-Sep-2018]

[2] B. Carpenter, "Middleboxes: Taxonomy and Issues", Ietf.org, 2002, [Online]. Available: https://tools.ietf.org/html/rfc3234. [Accessed: 24- Sep- 2018]

[3] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On Orchestrating Virtual Network Functions in NFV", Arxiv.org, 2015. [Online]. Available: https://arxiv.org/pdf/1503.06377.pdf. [Accessed: 24- Sep- 2018]

[4] S. Mehraghdam, M. Keller, H. Karl, "Specifying and placing chains of virtual network functions - IEEE Conference Publication", Doi.org, 2014. [Online]. Available: https://doi.org/10.1109/CloudNet.2014.6968961. [Accessed: 24- Sep- 2018]

[5] S. Livi, Q. Jacquemart, D. L. Pacheco, G. Urvoy-Keller, "Container-Based Service Chaining: A Performance Perspective - IEEE Conference Publication", Doi.org, 2016. [Online]. Available: https://doi.org/10.1109/CloudNet.2016.51. [Accessed: 24- Oct- 2018].