University of Colorado
Interdisciplinary Telecommunications Program
CYBR 5700 Graduate Projects I

Containerized, Intelligent Network Functions on Host

Project Definition Document (PDD)

Approvals

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

Project Customers

Name: Dr. Levi Perigo
Email: Levi.Perigo@colorado.edu

Team Members

Name: Afure Martha Oyibo Email: maoy4508@colorado.edu	Name: Kiran Yeshwanth Email: kiye0434@colorado.edu
Name: Manesh Yadav Email: maya8094@colorado.edu	Name: Prarthana Shedge Email: prsh1271@colorado.edu
Name: Soumya Velamala Email: sove4216@colorado.edu	

1.0 Problem or Need

Service networks are becoming more prominent and intricate with the advancement of technology. Managing a traditional network is a cumbersome task for network administrators due to the ever-changing network architecture. Smartphones and Internet of Things (IoT) are pushing the barrier even further. Traditional network architectures that have minimal network visibility are challenging to scale, operate, manage, and cannot cope with the rising demand. These networks often use vendor-specific hardware which is costly, lack flexibility, and have a short product life-cycle due to the rapid progress in innovation.

Software-Defined Networking (SDN) is shifting the paradigm by providing automated services resulting in better network management, more straightforward implementation, and flexible solutions. SDN and Network Function Virtualization (NFV) provide the network services as abstracted, orchestrated flows controlled through centralized software using Virtual Network Functions (VNFs). NFV Edge [1] is the capability to bring network services such as routing, firewall, Network Address Translator (NAT), Quality of Service (QoS), and voice services to the edge of the network. The current implementation of NFV Edge centers on the Wide Area Network (WAN) edge, and mobile edge clouds as a standalone NFV solution integrated with traditional networks or as an SDN-NFV solution.

Containerized, Intelligent Network Functions on Host project focuses on simplifying the SDN architecture by deploying VNFs as customized containerized applications on the hosts. The containers have access to the kernel and operate on a single instance, which reduces the kernel overhead and enhances performance as compared to Virtual Machines (VMs). This solution reduces the latency, throughput, provides enhanced QoS, faster service delivery, easier adoption of DevOps practices, and improved backward compatibility. It provides better scalability, reduced network complexity, and enhanced flexibility at a reduced cost by leveraging compute resources available on the hosts. [2] The application layer, with the help of the SDN controller, can push the flow entries on the containerized VNF at the host as per the requirement. For example, Intrusion Detection Services can be installed at the hosts to block perilous packets from affecting the network. Using a simple Layer2 (L2) backbone network reduces network complexity, and increases scalability, as shown in the Containerized, Intelligent Network Functions on Host Architecture in Figure 1.

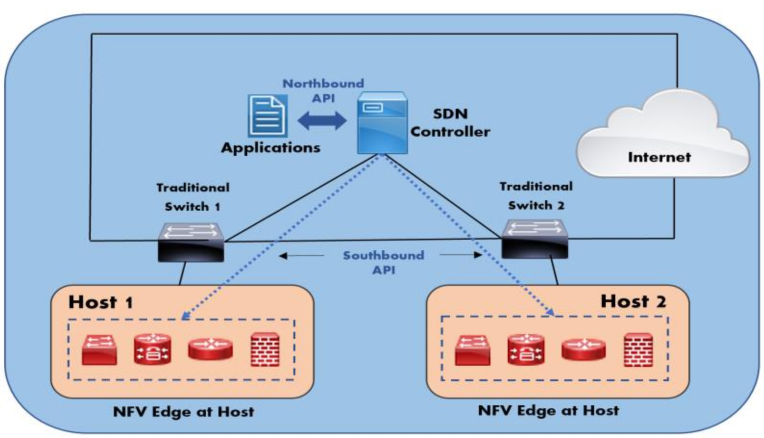

Figure 1: Containerized, Intelligent Network Functions on Host Architecture

2.0 Previous Work

SDN combined with Containerized, Intelligent Network Functions on Host aims at reducing the network overhead at the backbone/core by providing easy to install network services as containers at the hosts. Although there has been research in the field of NFV Edge, this project (Containerized, Intelligent Network Functions on Host) is a novel concept with limited research.

In 2016, Yang B. and others [3] studied the performance of multimedia applications such as Voice over IP (VoIP), video streaming for mobile phones based on the placement of NFV based Mobile Edge Computing (MEC) services by varying the allocation of resources along with load balancing in order to check the latency and QoS. In 2016, Boubendir A. and others [4] devised a solution that uses network Application Programming Interfaces (APIs) to enable third-party NFV based applications on the customer network using containers and Open Platform for NFV (OPNFV). In 2017, Li S. and others [5] devised an NFV based platform for MEC services to check the QoE for the different web-based services. In 2017, Nam Y, Song S, and Chung J-M [6] studied different service chaining models for MEC using NFV devices to calculate the latency and end-to-end service time for Edge Computing in Radio Access Networks (RANs).

A few research publications have been made in the field of Containerized, Intelligent Network Functions on Host. In 2018, Gedia D. and Perigo L. [2] studied the advantages of using containers for VNFs instead of VMs in terms of Central Processing Unit (CPU) utilization, memory utilization, throughput, and latency parameters. In 2019, a fellow group of Interdisciplinary Telecommunications Program (ITP) graduate students [7] devised an SDN based VoIP application using containers that provided enhanced performance compared to the traditional VoIP applications.

This project takes inspiration from the previous research on creating container-based NFV applications for different network services and forming service chains to ensure seamless performance. It fills the research gap in this area by providing a detailed analysis of the performance across various parameters like latency, throughput, QoS, and compatibility of the network topology functions with the help of containers.

3.0 Specific Objectives

3.1 Level 1 (Configuration Tests): Test the deployment of VNFs in the container(s). Create test cases to corroborate reachability to other devices in the network and to design test cases to validate flow entries and flow-table pipelining.

3.2 Level 2 (Performance Tests): Test the network performance parameters by subjecting it to various service tests. The network topology is to be subjected to different traffic using distinct service types, transmission rates and packet sizes to check QoS, and throughput. Evaluate the resource utilization of the containers to ensure that the host has sufficient capacity.

3.3 Level 3 (Failover Mechanism Tests): Test the network's ability to reallocate and use redundant resources after failures arising from device failures, link failures, network connectivity, congestion, improper configuration, and security breach. Verify the failover to conventional TCP/IP stack in case of SDN controller/VNF failure.

4.0 Functional Requirements

An Open vSwitch (OVS) VNF is created using containers on host devices, which supports flows for different network functionalities such as a switch, router, NAT, and a firewall. VNFs are to be created using containers to support standalone network functionalities such as VoIP and commercial firewall applications. The primary task of this project is to create different service chains using these VNFs and the flow table entries supported by them, as depicted in the Functional Block Diagram depicted in Figure 2. An SDN controller helps in automating, abstracting, orchestrating, and provisioning of the solution by using north and south-bound APIs. The application layer pushes the flow entries using the SDN controller to these VNFs as per the service chain scenario.

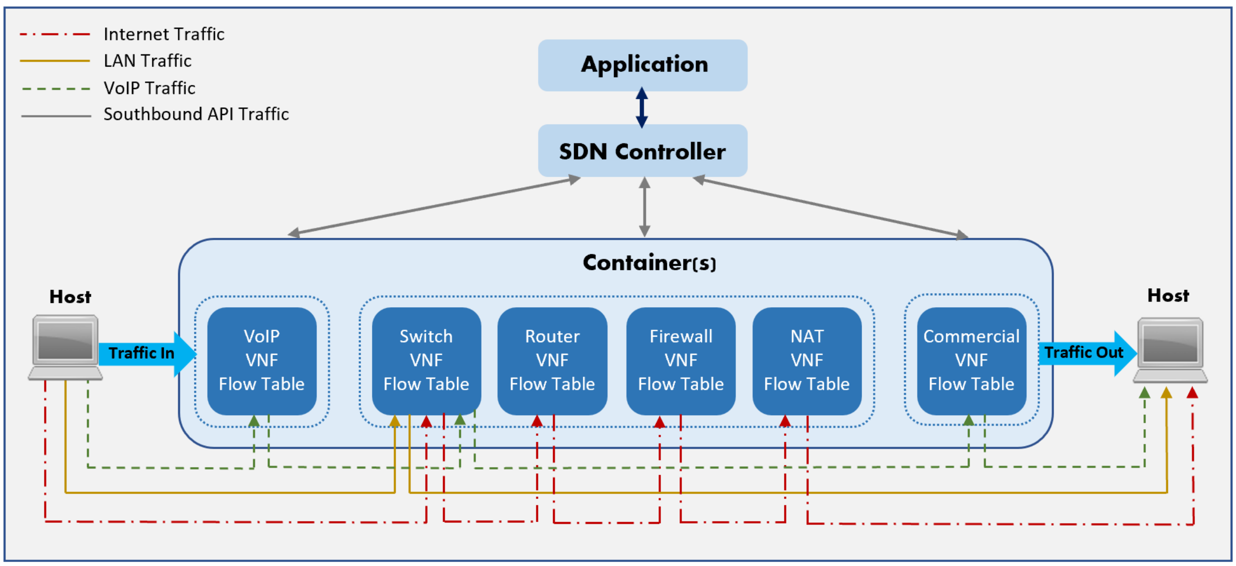

Figure 2: Functional Block Diagram (FBD)

A VM running using hypervisor (Testbed VM) is used to create a testing framework to evaluate the NFV Edge network for configuration, performance, and failover mechanism tests. A packet analyzer is used at the switch interfaces to live capture the traffic to be used by the Testbed VM for analysis. The test results are sent to the application layer to store it in the Database. With the help of a Data Visualization tool, the web interface displays the test results, as shown in the Concept of Operations (CONOPS) diagram in Figure 3.

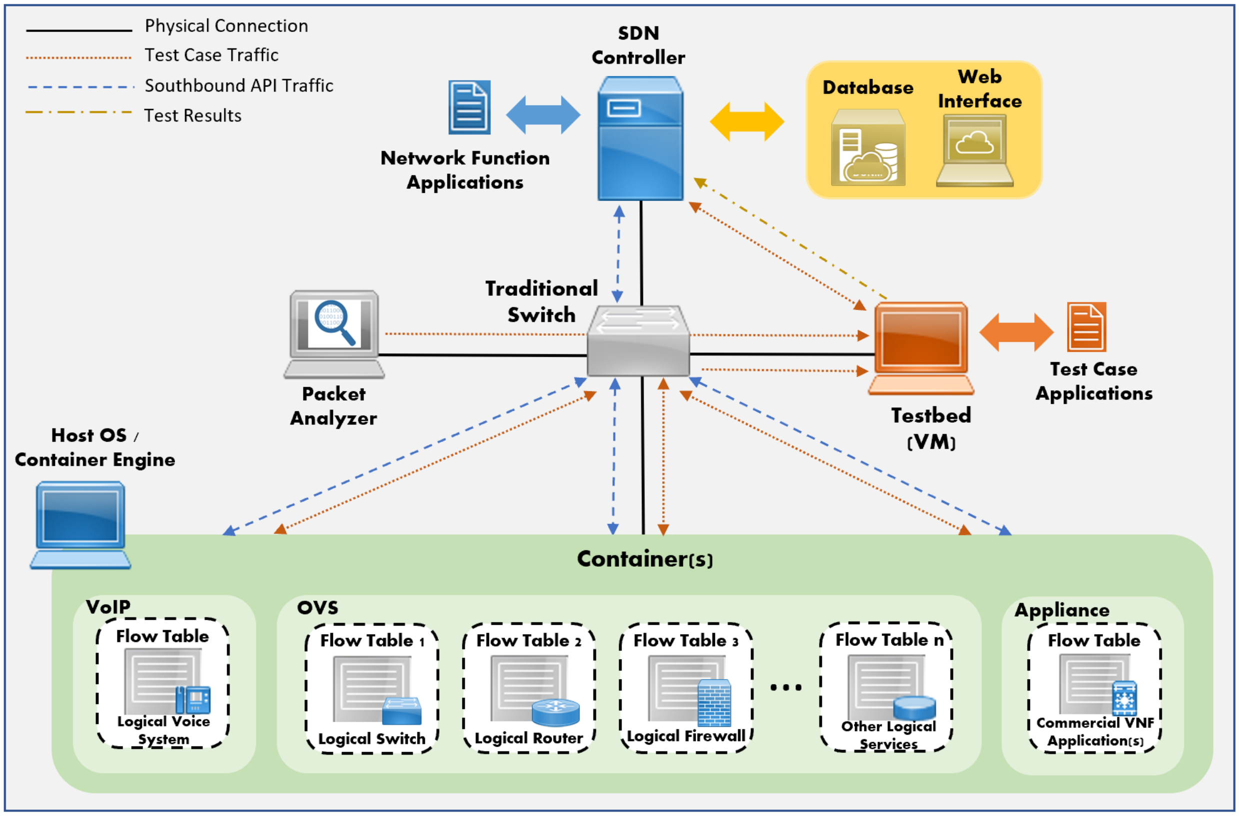

Figure 3: Concept of Operations (CONOPS)

5.0 Critical Project Elements

Technical:
CPE 1.1 - VNFs creation: Create multiple VNFs corresponding to OVS, VoIP, and other commercial network services using containers running on the host**.**
CPE 1.2 - Configure and deploy SDN infrastructure: Deploy and configure SDN Controller to install flows in the VNFs and manage the entire network. A packet analyzer is deployed on the edge switch to capture live traffic.
CPE 1.3 - Test VM creation: Emulate a test VM using the hypervisor to perform operational, performance, and functional tests.
CPE 1.4 - Service chain creation: Create multiple service chains with multiple VNFs to simulate and test different scenarios.
CPE 1.5 - Storage and display of test results: A database is created to store the test results in the application, perform data analysis by parsing it to a data visualization tool and display it on a web interface.

Logistical:
CPE 2.1 - Hardware devices: Using laptops/desktops with the SDN Controllers and SDN compatible switches in the Telecommunications laboratory for implementation.
CPE 2.2 - Containers/Controllers: Using open-source software and applications to create the network topology.

CPE 2.3 - Knowledge and concepts: Need to understand the functionality and uses of containers, SDN, VNFs to establish NFV Edge on a host. Determine a web Graphical User Interface (GUI) framework and learn the functionality to display the output from test results on the web interface.

6.0 Team Skills and Interests

Afure Martha Oyibo: Interned at Facebook as a Network Deployment Engineer. She has 3 years of combined experience as a Network and Systems Engineer. She is proficient in traditional networking and has good hands-on experience with systems and network automation. Her interest lies in DevOps, virtualization technologies, high performing, and low latency system designs in the network domain.

Kiran Yeshwanth U: Kiran interned at Charter Communications as a Product Development - Software-Defined Networking in Wide Area Network (SD-WAN) engineer. With 4 years of experience as a Network Engineer (Wireless) at Ericsson, he has worked on various Radio Frequency (RF) technologies such as Long-Term Evolution (LTE), Wideband Code Division Multiple Access (WCDMA), and Global System for Mobile (GSM). His interest lies in SDN, DevOps, and python programming.

Manesh Yadav: Manesh interned at Nuage Networks from Nokia as a Software Engineer Quality Assurance (QA) and has worked in the SD-WAN domain. He has 6 years of work experience in Networking (Wireless domain) technologies like GSM, WCDMA, and LTE. He has sound knowledge in traditional routing and switching, python programming, and Linux system administration. His interest lies in SDN, network automation, and machine learning.

Prarthana Shedge: Prarthana interned as a Network Engineer at Facebook, where she gained hands-on experience in network automation and DevOps. She also has 3 years of experience in designing, provisioning, and troubleshooting the traditional Service Provider network. Her interest lies in SDN, NFV, and low-latency networks.

Soumya Velamala: Soumya interned at Nuage Networks as Software Engineer QA for SD-WAN product. She has worked for 5 years as a Network Operations Engineer and is proficient in provisioning, designing, and troubleshooting traditional Internet Service Provider (ISP) networks. She is interested in network automation, SDN, and cloud technologies.

Critical Project Elements	**Teams member(s) and associated skills/interests**
1.1 VNF creation	Manesh, Kiran
1.2 Configure and deploy SDN infrastructure	Soumya, Martha
1.3 Test VM creation	Martha, Manesh
1.4 Service chain creation	Manesh, Prarthana
1.5 Storage and display of test results	Prarthana, Kiran
2.1 Hardware devices	All Team Members
2.2 Containers / Controllers	Soumya, Prarthana
2.3 Knowledge and concepts	All Team Members

7.0 Resources

Critical Project Elements	Resource/Source
1.1 VNFs creation	Dr. Levi Perigo
1.2 Configure and deploy SDN infrastructure	Dr. Levi Perigo
1.3 Test VM creation	Dr. Levi Perigo
1.4 Service chain creation	Dr. Levi Perigo
1.5 Storage and display of test results	Dr. Levi Perigo
2.1 Hardware devices	Dr. Levi Perigo
2.2 Containers / Controllers	Dr. Levi Perigo
2.3 Knowledge and concepts	Dr. Levi Perigo

8.0 References

[1] Han, B., Gopalakrishnan, V., Ji, L., and Lee, S., "Network Function Virtualization: Challenges and Opportunities for Innovations", Feb 2015. [online]. Available: http://www.ttcenter.ir/ArticleFiles/ENARTICLE/3431.pdf

[2] Gedia, D., and Perigo, L., "Performance Evaluation of SDN-VNF in Virtual Machine and Container", Proc. IEEE Conf. Network Function Virtualization and Software Defined Networks (NFV-SDN), Nov 2018.

[3] Yang, B., et al., "Seamless Support of Low Latency Mobile Applications with NFV-Enabled Mobile Edgecloud", Proc. IEEE Int'l. Conf. Cloud Networking (Cloudnet), Oct 2016.

[4] Boubendir, A., Bertin, E., and Simoni, E., "On-Demand, Dynamic and At-The-Edge VNF Deployment Model Application to Web Real-Time Communications" Proc. 12th Int'l. Conf. Network and Service Management (CNSM), Nov 2016.

[5] Li, S., et al., "QoE Analysis of NFV-Based Mobile Edge Computing Video Application", Proc. IEEE Int'l. Conf. Network Infrastructure and Digital Content (IC-NIDC), July 2017.

[6] Nam, Y., Song, S., and Chung, J-M., "Clustered NFV Service Chaining Optimization in Mobile Edge Clouds", IEEE Commun. Lett., vol. 21, Feb 2017.

[7] Kuthiala, A., Kohli, S., Agera, N., Volety, N., A., P., Gandotra, R., and Perigo, L., "SDVoIP – Software Defined Voice over Internet Protocol", ITP Capstone Project, May 2019.