NFV blah

Chi-Hsuan Li[†], Che-Wei Lin[‡], Sheng-Jung Wu[†]

[†] Department of Computer Science, National Tsing Hua University
No.101, Sec. 2, Guangfu Rd., East Dist., Hsinchu City 300, Taiwan
[‡] Institute of Communication Engineering, National Tsing Hua University
No.101, Sec. 2, Guangfu Rd., East Dist., Hsinchu City 300, Taiwan

Abstract IEICE (The Institute of Electronics, Information and Communication Engineers) provides a template file for the IEICE Technical Report.

Key words IEICE, IEICE technical report, LaTeX, template

1 Introduction

In these days, the network functions have took its place in modern life, play an important role in providing critical performance, security and policy compliance capabilities. However, network service is usually achieved by the hardware and using a dedicated device to deploy and integrate a variety of network functions, which causes the lack of flexibility in service delivery. When the network service provider wants to introduce the new service, customers will tend not to replace with new equipment and the service provider must also cost high capital expenses (CapEx) and operation expenses (OpEx) to replace the physical equipment. As a result, the network service is difficult to promote and update.

It was against this background, the concept of network function virtualization (NFV) was put forward, expecting to solve these problem. NFV transforms how network operators architect their infrastructures with the virtualization technology to separate software instance from hardware platform, and implements virtualized network functions (VNFs) through software techniques. And NFV also innovates in the service delivery, by deploying these functions over general virtualized compute infrastructures, which are used to replace the traditional network appliances. When these functions are no longer a large degree depent on specific physical devices, the management of VNFs can be easier and the deployment will achieve flexibility and scalability.

With Software Defined Network, which represents the idea and technology of decoupling the control plane from the underlying data plane with, the evolution virtualization of the network functions is a big step forward. At the beginning, SDN simply contribute the network infrastructure with enabling dynamic control and configuration of network nodes. Afterwards, the network infrastructure layer become an implementation part of the VNF functionality.

This concept of SDN-enabled VNF fits perfectly the needs of service providers, especially providers of Customer Premises Equipment (CPE). In the current physical Consumer Premises Equipment (pCPE) model, service providers have to deploy multiple discrete devices. The range of services region are diverse, which make installation taking long time and high CAPEX, and when service providers want to promote new services, the customers have low aspiration to purchase

Though the technology of SDN-enabled VNF, the concept of virtual Consumer Premises Equipment (vCPE) has been discussed. With vCPE platform, service providers will be able to provide services through internet and the customer may need to buy only one low-cost device. As a result, it will shorten installation time, reduce the maintenance cost of pCPE, and increasing the purchase intention of customers.

In this paper, we proposed a vCPE framework architecture to provide SDN-enabled NFV service. The proposed framework is especially suitable for service provider to deliver their service. Consequently, there is a great demand to use the vCPE framework to lower the devices cost as well as maintenance cost.

The rest of the paper is organized as follows: Some related work will be described in Section 2. The proposed vCPE system architecture and implementation details are presented in Section 3. The section 4 introduced some SDN-enabled VNF we have implemented. The Section 5 depicts the experimental environment that verifies the proposed architecture and mechanisms. Lastly, the conclusion and future works are presented in Section 6.

2 Related Work

3 System Design and Implementation

3.1 NFV Design Overview

With the concept of SDN-enabled VNFs in Fig. 1, the network functions have been achived by the synergies between compute and network infrastructures. The former is mainly responsible for dealing with stateful processing, and the latter is used for stateless processing component.

3.1.1 Stateful Processing Componemt

This component have to perform more complex algorithm, keep the state associated with the VNF and provide interface for service providers or customers to configure and update the behavior of the stateless datapath processing component, since software is good at these tasks. It's worth noting that we use southbound APIs of SDN controller to handle the interface between the stateful and stateless component with OpenFlow protocol, which was originally designed for this.

3.1.2 Stateless Processing Componemt

Stateless processing component, are implemented by SDN datapath resources, which is optimized for data plane traffic processing. Since SDN datapath have decoupled the control plane and data plane, so it can accept the control message from the stateful processing component.

3.2 Architecture of the vCPE Platform

3.2.1 System Overview

See Fig 2

3.2.2 System Implementation

1) **Infrastructure Orchiestrator**

2) **Infrastructure Controller**

3) **VNF Orchiestrator**

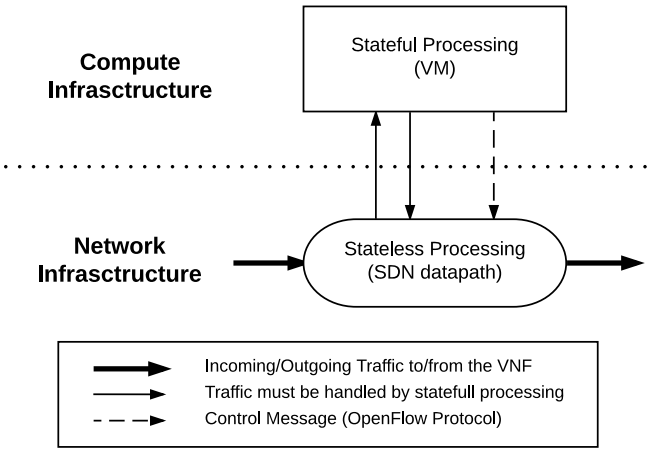

Figure 1: NFV design overview

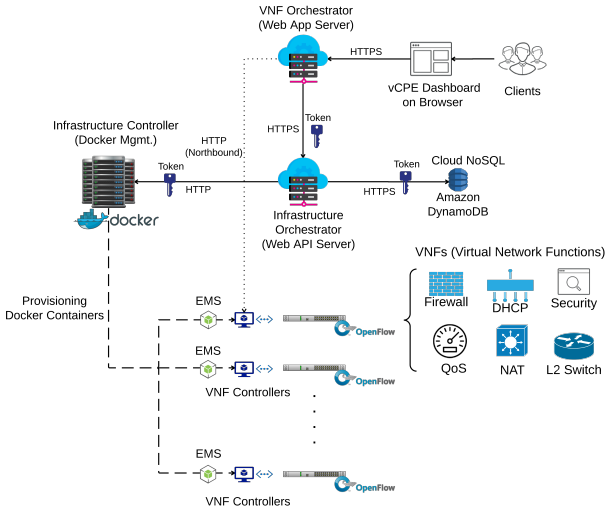

Figure 2: vCPE platform overview

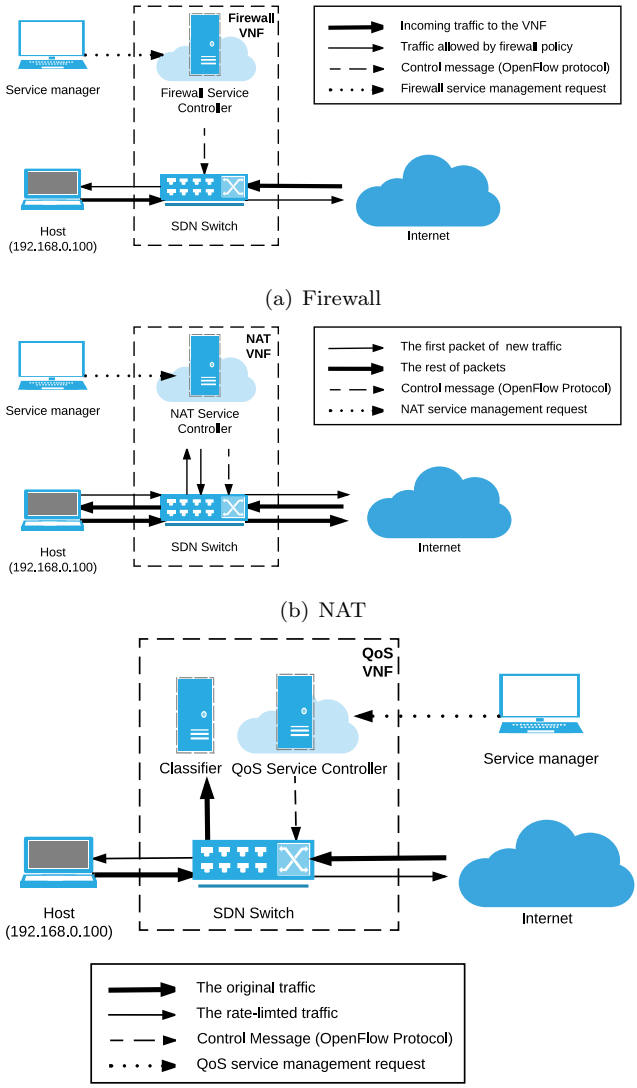

Figure 3: The use case of Virtualized Network Functions

4) **VNF Controllers**

4 Virtualized Network Functions and Use Cases

Considering SDN-enabled architecuture approach, the NFVs can be furthur classified into three categories with different state complexity when processing incomming traffic:

1) **Policy-based service:** In this category, the network infrastructure takes the primary responsibility of handling incoming traffic and the traffic need only stateless processing which is following certain policy given by the service management. Firewall, network mirror tap and load balancing are belongs to this type.

2) **Dynamic-configured service:** The first a few packets are needed to send to stateful processing component for initializing configuration, and the rest packets can be handled by SDN datapath with the previous configured setting. For example, NAT and DHCP services are belongs to this category.

3) **QoS service:** In this approach, our NFV is collaborate with a traffic classification system. The whole incoming traffic will be redirect to the compute infrastructure, where the classifiers run on. When we get the information abount current connection and traffic, we can take responding action, like dropping the connection of malware or limited the throuput of streaming application.

Following we would introduce the SDN-enabled VNFs we have implemented based on these three categories.

4.1 Policy-based service

(Fig 3(a)) We take firewall as an example to illustrate policy-based service. Our virtual firewall service provides the usual packet filtering. It filters each packet

based on information contained in the packet itself, using a combination of the packet's source address, destination address and its protocol, which can be transformed into a list of OpenFlow rules.

In this case, the target traffic need only stateless processing which can be done solely by the SDN datapath resource. Therefore, we just provide policy management interface to handle the request from service manager in the software controller component. When the firewall network service was requested to update firewall policy, the controller will map the request to OpenFlow protocol messages and send to the SDN switch, which will follow the policy to process the incoming traffic.

For example, if the service manager choose to block certain protocol for source IP address 192.168.0.100, he can send the request to controller via management API. The request will be mapped to the form of OpenFlow protocol, adding rules which is Match-Fields with the drop action, and then send to the datapath resource. As a result, when the host 192.168.0.100 want to send SSH request, it will be blocked by the SDN switch.

4.2 Dynamic-configured service

Our virtualized Network Address Translation (NAT) service is a typical examples of this kind of service. (Fig 3(b)) The NAT service is a network function which remapping one IP address space into another by modifying network address.

When client open up a new TCP or UDP connection, the first packet come the SDN switch will be redirected to service controller, since SDN switch have no idea how to handle this packet. Thc controller would pick an unused source port number for remapping this connection, and ask the datapath to handle the Set-Field action, which is used to rewrite packet header fields to easily convert the forwarding policy into OpenFlow entries.

When the rest of packets from private network host arrives at the SDN switch, the packet header fields are modified by Set-Field action. For outgoing packet, the source IP address and source port number are modified to fit the public IP of NAT and remap a port number of NAT. For incoming packet, the destination IP and destination port number are modified accordingly. Below are some examples to present the flow tables for NAT service with ingoing packets and outgoing packets.

4.3 Traffic monitoring service

Fig 3(c)