SDN Controlled VNF Forwarding Graphs
PoC - Lessons Learned

Linda.dunbar@huawei.com (ONF L4-L7 Group Chair)
Myo.zarny@gs.com (ONF L4-L7 Group Chair)
Cathy.H.Zhang@huawei.com (PoC Coordinator)
Louis.fourie@huawei.com
PoC Background

- Conducted in 2015: ETSI NFV-ONF sponsored proof-of-concept (POC #28) trial on SDN-Controlled VNF Forwarding Graphs conducted by nine vendors
- VNF Forwarding Graphs – Use Case #4 as outlined in the ETSI GS NFV 001 v1.1.1 (13-10) document
PoC Background

• **Roles of Participating Companies**
  - Network Operator/Service Provider: Deutsche Telekom
  - Manufacturer A: Huawei
  - Manufacturer B: Freescale
  - Manufacturer B: RiverBed

• **Goals of the PoC**
  - PoC Project Goal #1: SDN controlled NFV VNF service delivery (insertion/deletion/reconfiguration)
    - SDN/OpenFlow controlling dynamic steering of various flows to their designated VNFs (with different VNF types and from different vendors) in scalable way
    - What/how OpenFlow protocols are used to control forwarding nodes, virtualized network functions, etc.
  - PoC Project Goal #2: NFV MANO (OpenStack, CloudStack, etc.) can leverage the SDN controller’s APIs to:
  - PoC Project Goal #3: NFV MANO interworking with orchestration engine that can translate user’s requests to SDN controllers for coherent service delivery
  - PoC Project Goal #4: interoperability with existing non-SDN infrastructure
Map to NFV Framework
Mapping to NFV ISG Work:
NFV Use Case #4 (VNF Forwarding Graphs);
Requirement: Elasticity & Portability (5.2 & 5.4 of gs-NFV004v010101p)
E2E Arch: “Or-Vi” interfaces

Scenario #1: User Selected Service Chain

Service Chain "Acceleration"
DPI -> QoS Marking -> Acceleration

Service Chain "Green-internet"
URL Filter -> QoS Marking -> FW

Traffic Generator (Wget for HTTP, ftp, ssh; Browsing and Video)

Compute node with switch and SF (potentially with vFW)

X.86 based Compute node with Switch & Riverbed SFs

Orchestration (Huawei)
Controller (Freescale)

Traffic Generator (VxLAN)
Scenario #2: Scale out and Scale In

- Riverbed provide VNF management systems

Service Chain “Acceleration”
- DPI -> QoS Marking -> Acceleration

Service Chain “Green-internet”
- URL Filter -> QoS Marking -> FW

Traffic Generator
- (Wget for HTTP, ftp, ssh; Browsing and Video)

Controller (Freescale)

Orchestration (Huawei)

Classifier

SW1

SW2

WanOpt

DPI

Packet Capture

vFirewall

vFirewall

VNF Management System
Scenario #3: Fail-over

QoS Marking
vFirewall
vURL_filter

Orchestration (Huawei)
Controller (Freescale)

Service Chain “Acceleration”
DPI -> QoS Marking -> Acceleration

Service Chain “Green-internet”
URL Filter -> QoS Marking -> FW

Traffic Generator (Wget for HTTP, ftp, ssh; Browsing and Video)

Classifier
SW1
SW2

QoS Marking
DPI
vURL_filter
vFirewall
vFirewall

SW3
SW4

SF vendor Mgr
Scenario #4: Dynamic by-pass

- **QoS Marking**
- **vFirewall**
- **vURL_filter**

**Orchestration (Huawei)**

**Controller (Freescale)**

- Service Chain "Acceleration"
  - DPI -> QoS Marking -> Acceleration

- Service Chain "Green-internet"
  - URL Filter -> QoS Marking -> FW

**Traffic Generator**
- Wget for HTTP, ftp, ssh; Browsing and Video

**Traffic Generator**

**Classifier**
- Qosmos

**SW1**
- DPI
  - vURL_filter

**SW2**
- vFirewall
LESSONS LEARNED & GAPS IDENTIFIED IN NFV STANDARDIZATION

ETSI GS NFV-EVE 005 http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_NFV-EVE005v010101p.pdf
A.12 POC#28: SDN Controlled VNF Forwarding graph
SF by different Vendors have different capabilities
Service functions by different vendors have different interfaces

1. The same function, different parameter name

2. The same parameter, different meanings

NSF interface of multi-vendor is privatized.
Difficult to automated deployment and unable to realize SDN.
Challenges …

Clients need to control its network functions for their virtual networks.

There are many types of VNFs, the Video Optimization based VNF is very different from FW/IPS/IDS based VNF, or Router/switch based VNF.

Other issues:
- Clients don’t know how their VMs are mapped in the network.
- VMs being moved, which will have different network ports.
- Clients can’t easily view/query the FW policies related to their virtual networks.
Client Specific Policies

Zones:
Yellow zone
Green zone
...

Security Policy:
Yellow ===> Yellow, Green
Green ===> Green
• Prohibited
Green=X ===> Yellow

Config 1:
App 1=IP1
App 2=IP2
App 3=IP3
App 4=IP4
...

Policies for Firewall:
- IP1 ===> IP3
- IP1 ===> IP4
- IP2 =X===> IP3
- IP2 ===> IP4
- IP3 ===> IP1
- IP3 ===> IP2
- IP4 =X===> IP1
- IP4 ===> IP2

Change of the policies:
- IP11 ===> IP12
- IP11 ===> IP14
- IP13 ===> IP12
- IP13 ===> IP14
...
- IP12 =X===> IP11
- IP12 =X===> IP13
- IP14 =X===> IP11
- IP14 =X===> IP13
...

Config 2:
App 1=IP11
App 2=IP12
App 3=IP13
App 4=IP14
...

Zones:
Yellow zone
Green zone
...

Security Policy:
Yellow ===> Yellow, Green
Green ===> Green
• Prohibited
Green=X ===> Yellow
Need Standardized Interfaces

✓ **Interface: A**
  - For Service Function Vendors to register their available service functions & instances.

✓ **Interface: B**
  - The Service Layer defines how clients’ security policies may be expressed and monitored.

✓ **Interface: C**
  - The Capability Layer specifies how to control and monitor NSFs at functional level.
PROGRESS OF STANDARDIZATION
Service Interface (IETF I2NSF)
Under Development

Service Group Policy Model

Tenant Group
- tenant
- ...

ID | Name | Profile | Privilege | ...

Logical combination

Event Group
- Event
- ...

Event Group
- Event
- ...

Condition Group
- condition
- ...

Condition Group
- condition
- ...

Action Group
- action
- ...

Action Group
- action
- ...

characteristic
- Device
- user
- VM
- APP
- Web
- ...

Profile

Privilege

allow
deny
redirect...

state
rate
...

time

state
rate
...
IETF I2NSF Capability Layer (Under development)

- Match values based on packet data
  L2/L3/L4 Packet header
  Packet payload

- Match values based on context
  Ex.: user, Schedule, Region, Target, State, Direction, etc.
  Many can (and should) be standardized, but many also from NSF capabilities

- Egress processing
  Invoke signaling
  Packet forwarding and/or transformation
  Possibility for SDN/NFV integration

- Vendor Unique innovation, Vendor specific
  e.g. IPS:<Profile>
  Profile: signature, Anti-virus, URL filtering, etc.
  Integrated and one-pass checks on the content of packets

Key goal:
- Flexible and comprehensive semantics;
- extensible IM for containing different vendors’ security capabilities, in essence, respective difference or innovation.
Enabler: Application Defined Service Control

- Sliced Network for Specific Applications
- Video Service Ctrl
- QoS
- IPSec Service Service
- Anti-DDoS as a Service
- Security Service Engine
- Network Service Engine
- SDN Controller
- Analytics Service
- Analytics Service
- ICN Service Ctrl
- VPN Customer Ctrl

Shared Network Layer (VNF, routers, switches, ...
Summary

➢ Simply chaining together the VNFs is not enough
➢ Needs methods to categorize VNFs for Operators to utilize the VNFs from different vendors.
➢ Need Standardized interfaces for policies to VNFs and ways to monitor the execution status
OPENSTACK SERVICE CHAINING
Service Chaining in OpenStack (networking-sfc)

OpenStack Based Management Plane

- Service Chain Intent Manager
- Service Instance Catalog Manager
- Neutron Server with Service Chain Extension (networking-sfc)
- SF Instance Manager (OpenStack or 3rd Party)

OVS or SDN Based Control Plane

- Traffic Source
- Classifier
- SF Forwarder (vSwitch)
- FW
- IDS
- Cache
- Traffic Destination
- WOC
- QoS
- Load Balancer
- SF Forwarder (vSwitch)
Neutron Service Chain (networking-sfc)

OpenStack Neutron Server

- OpenStack API for Service Chain (C1)
- Service Chain Plugin (networking-sfc)
- Common Service Chain Driver API (C2)

OVS Service Chain Driver
ONOS Service Chain Driver
ODL Service Chain Driver
Dragonflow Service Chain Driver
OVN Service Chain Driver

ONOS Controller
ODL Controller

Compute Node
- OVS Switch (Classifier)
- Service VM (FW)
- Service VM (IDS)

Compute Node
- OVS Switch (Classifier)
- Service VM (NAT)
- Service VM

Compute Node
- OVS Switch (Classifier)
- Service VM
- Service VM (LB)

Traffic Source
Traffic Destination
Openstack networking-sfc Project

- Started in Liberty cycle
- First phase complete in Mitaka
- Approved specs:
  - Service Chain API
  - System Design and Workflow
  - OVS driver and agent
- Service chain implementation:
  - CLI, Horizon
  - Neutron server: API, DB, Driver Manager, Common Driver API
  - OVS driver and agent
  - ONOS driver
Openstack networking-sfc Project

- Multiple Flow Classifiers per Port Chain
- Port chain update for flexible Service function insertion and removal
- Scale-in/out by updating Port Pair Group to add/remove Port Pairs (service functions)
- Port Chains may span multiple Compute nodes
- Port Chains may span multiple subnets
- [http://docs.openstack.org/developer/networking-sfc/](http://docs.openstack.org/developer/networking-sfc/)
- IRC Meetings [http://eavesdrop.openstack.org/meetings/service_chaining/](http://eavesdrop.openstack.org/meetings/service_chaining/)
Service Chain Objects

- Port Chain - represents a Service Function Chain
  - Sequence of Port Pair Groups
  - List of Flow Classifiers
- Port Pair Group - scale-in/out: defines a load distribution group of functionally equivalent SFs
  - Group of Port Pairs
- Port Pair - represents a single SF
  - Ingress, egress Neutron ports
- Flow Classifier - N-tuple for packet matching
  - Source/destination IP address, TCP/UDP ports, protocol, IP version, source/destination Neutron ports
Service Chain Example

Neutron API Service Chain Extension

Chain Classifier

Source N-Tuple

Destination N-Tuple

Logical Chain Path

IPS1 Neutron Port-pair

IPS2 Neutron Port-pair

FW1 Neutron Port-pair

FW2 Neutron Port-pair

FW3 Neutron Port-pair

IPS2 Neutron Port-pair

WOC1 Neutron Port-pair

WOC2 Neutron Port-pair

Traffic Source

IPS

FW

WOC

Traffic Destination
Service Chain Configuration Model

- Port Chain
- Flow Classifier
- Port Pair Group 1
  - SF Port Pair 1.1
- Port Pair Group 2
  - SF Port Pair 2.1
  - SF Port Pair 2.2
  - SF Port Pair 2.3
- Port Pair Group N
  - SF Port Pair N.1
  - SF Port Pair N.2
THANK YOU