AGENDA

• ETSI NFV MANO
• IETF SFC
• Existing solutions
• Container service chaining solution
• Demo
ETSI NFV Management and Orchestration (MANO)
MANO ARCHITECTURE
ARCHITECTURE IN EXAMPLE

Portal

Tacker

NFV Orchestration (NFVO)

NFV Service

VNF Catalogue

NFV Instance

NFVI Resource

FW API

FW

Server + HV

Virtualised OpenStack Manager (VIM)

OpenStack

Tacker
**NOT PART OF MANO**

- **NFVI** – NFV Infrastructure that includes physical (server, storage etc.), virtual resources (Virtual Machines, Containers) and software resources (hypervisor) in an NFV environment.

- **VNF** – Virtual Network Function is the virtualized network element like Router VNF, Firewall VNF etc.

- **EM** – Entity Manager is responsible for the FCAPS for the functional part of the VNF.

- **OSS/BSS** include collection of systems/applications that a service provider uses to operate its business.
VIM

- manages life cycle of virtual resources in one NFVI domain
- creates, maintains and tears down VMs, Containers from physical resources in an NFVI domain
- there may be multiple VIMs in an NFV architecture, each managing its respective NFVI domain
**VNFM**

- manages life cycle of VNFs
- creates, maintains and terminates VNF instances which are installed on the VMs, Containers
- there may be multiple VNFMIs managing separate VNFs
- there may be one VNFM managing multiple VNFs
NFVO

- coordinates, authorizes, releases and engages NFVI resources by engaging with the VIMs directly through their north bound APIs
- creates end to end service among different VNFs (that may be managed by different VNFMNs)
CATALOGUES

- NFV service (NS) catalogue
- VNF Catalogue
- NFV Instance repository
- NFVI Resource repository
VNFFG
NFV – MANO SOLUTIONS

- Open Source MANO (OSM)
- ONAP
  - OPEN-O
  - open source ECOMP
- CORD
- Gigaspaces Cloudify
- Open Baton
- Tacker
RESOURCES

http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/010/02.01.01_60/gs_NFV-IFA010v020101p.pdf

http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/009/01.01.01_60/gs_NFV-IFA009v010101p.pdf
IETF Service Function Chaining (SFC)
SERVICE FUNCTION CHAINING

• The definition and instantiation of an ordered set of service functions and subsequent "steering" of traffic through them is termed Service Function Chaining (SFC).

• SFC is complementary to MANO VNFFG
CLASSIFIER & CLASSIFICATION

- **Classifier** - is an element that performs classification.

- **Classification** - Locally instantiated matching of traffic flows against policy for subsequent application of the required set of network service functions. The policy may be customer/network/service specific.
SERVICE FUNCTION

• is responsible for specific treatment of received packets
• can be realized as a virtual element or be embedded in a physical network element
• one or more service functions can be involved in the delivery of added-value services
• firewalls, WAN and application acceleration, Deep Packet Inspection (DPI), Lawful Intercept (LI), server load balancing, NAT, HTTP Header Enrichment functions, and TCP optimizer
SERVICE FUNCTION FORWARDER

- is responsible for forwarding traffic to one or more connected service functions according to information carried in the SFC encapsulation, as well as handling traffic coming back from the SF.

- is responsible for delivering traffic to a classifier when needed and supported, transporting traffic to another SFF (in the same or different type of overlay), and terminating the Service Function Path (SFP).
ARCHITECTURE COMPONENTS AFTER CLASSIFICATION

- **SFC encapsulation** - provides, at a minimum, SFP identification, and is used by the SFC-aware functions, such as the SFF and SFC-aware SFs.

- **SFC-aware Service Function** (SFC-aware SF) – is network function which can process SFC encapsulation. It is equivalent to VNF in MANO.

- **SFC-unaware Service Function** (SFC-aware SF) – is network function which cannot process SFC encapsulation. It is equivalent to VNF in MANO.

- **Service Function Forwarder** (SFF) – forwards traffic among SFs and SFFs, equivalent to Virtual Link (VL) in MANO.

- **SFC proxy** – is used in case when SF is SFC-unaware so proxy can modify SFC encapsulation as SFC-aware SF would do.
SERVICE FUNCTION CHAIN (SFC)

- Defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification.
- An example of an abstract service function is "a firewall".
SERVICE FUNCTION PATH (SFP)

• is a constrained specification of where packets assigned to a certain service function path must go
• provides a level of indirection between the fully abstract notion of service chain, and the fully specified notion of exactly which SFF/SFs the packet will visit.
• by allowing the control components to specify this level of indirection, the operator may control the degree of SFF/SF selection authority that is delegated to the network.
RENDERED SERVICE PATH (RSP)

• represents visiting a specific sequence of SFFs and SFs. This sequence of actual visits by a packet to specific SFFs and SFs in the network is known as the Rendered Service Path (RSP).
EXAMPLE OF TRAFFIC STEERING BY USING SFC
TECHNIQUES USED FOR PATH IDENTIFICATION IN SFC

- Network Service Header (NSH)
- VLAN SFC
- Ethernet MAC Chaining
- SFC using MPLS-SPRING
NETWORK SERVICE HEADER (NSH)

- a new service plane protocol specifically for the creation of dynamic service chains and is composed of the following elements:
  - Service Function Path identification
  - Transport independent service function chain
  - Per-packet network and service metadata or optional variable type-length-value (TLV) metadata.

VLAN SERVICE FUNCTION CHAINING

- Uses combination of sMAC, VLAN, Rx Port for path identification and VLAN rewrite
- Assumptions about Service Functions:
  - Each service function node is assumed to be a bump-in-the-wire
  - Ethernet device with the following properties:
    - the device has two interfaces, logically subscriber-side and Internet-side;
    - the device forwards Ethernet packets between the interfaces without modifying any aspect of the Ethernet header;
    - if the devices needs to inject packets that it has created for a particular connection, it uses Ethernet MAC addresses and VLANs previously observed for the connection;
    - the device may be capable of intersecting an Ethernet 802.1q trunk, in which case it can reside on more than one service chain.

ETHERNET MAC CHAINING

- MAC chaining addresses are terminated at each SFF and replaced by a new set of MAC chaining addresses used to forward through the next SF in the chain.
- MAC Chain forwarding is performed by a SFF using DA and SA address swapping. The operation of a SFF has characteristics of a router in that it uses information in the packet to determine a new link destination, however unlike a router the new link decision is based on the previous MAC address rather than the IP address.

SERVICE FUNCTION CHAINING USING MPLS-SPRING

- each SF and SFF has own segment ID which is encoded as MPLS label

- the service classifier attaches a segment list of (i.e., SID(SFF1)->SID(SF1)->SID(SFF2)->SID(SF2)) which indicates the corresponding SFP to the packet. This segment list is actually represented by a MPLS label stack.

- SFF and SFC encap-aware SF pops top label before sending the packet
RESOURCES

https://datatracker.ietf.org/wg/sfc/documents/
Existing open-source solutions
EXISTING OPEN-SOURCE SOLUTIONS

OPNFV SFC

- Uses OVS 2.5.90 (Intel Patch)
- OpenDaylight Boron
- OpenStack Mitaka
- OpenStack Tacker project (customized)
- Direct API communication between Tacker and OpenDaylight
- Latest release: Colorado

https://wiki.opnfv.org/display/sfc

Tacker + SFC Overview: Proof of Concept (Direct ODL API)

EXISTING OPEN-SOURCE SOLUTIONS

OPNFV SFC
Tacker + SFC
Newton Implementation

Operator / OSS / BSS
Horizon(GUI)  CLI

Tacker

VNFM
NFVO

networking-sfc driver

Heat
Nova
Neutron
Networking-
sfc

SFC
Netvirt-sfc
Neutron NB
ODL Controller
OVSDB

Compute Node 1
VNF
vFirewall
HTTP Client
HTTP Server

OVS

Templates
VNFD

Templates
VNFFGD

EXISTING OPEN-SOURCE SOLUTIONS

OPNFV SFC

OTHER SOLUTIONS WITH THE SAME INTENT

• **OpenStack SFC** – ML2 with OVS driver is used instead of using ODL
  https://docs.openstack.org/developer/networking-sfc.ovs_driver_and_agent_workflow.html

• **ONOS SFC** – ONOS is used instead of ODL
  https://wiki.onosproject.org/pages/viewpage.action?pageId=4163192
Container service chaining solution
REQUIREMENTS

- Lightweight SF
- Simple for debugging
- Traffic steering without packet modification
- Avoid encapsulation overhead
PACKET FLOW FROM USER TO WEB

Assumptions:
• Each SF has two interfaces
• SFF has two physical interfaces (one towards access, the other towards aggregation)
• Traffic classification for SFC is based on VLAN
• SF chain is symmetric
• All SFs from the SF chain are located on single node
SFC DATA PLANE AND CONTROL/MANAGEMENT PLANE

**SFC data plane** (green lines)
- process traffic between Access and Aggregation
- Traffic is redirected on SFF to service functions

**Control/management plane** (blue lines)
- Allows connections to compute node and containers in order to configure SFF and SF
FUNCTIONAL COMPONENTS

- MANO components for SF (VNF) orchestration (NFVO, VNFM, VIM)
- SFC components for traffic steering (SDNC, SFF, SF)
- SFC port agent – creates and wire interfaces for data plane
- SFC Orchestrator – high level abstraction and glue between SFC and MANO
REAL COMPONENTS

- **OpenDaylight** - Open Source SDN Platform used for application development. It will run SFC wiring logic.
- **VPP** - Vector Packet Processing technology – an open source high performance virtual switch/router running on commodity CPUs
- **Cloudify** - open source cloud orchestration framework. It allows you to model applications and services and automate their entire life cycle.
- **Kubernetes/Docker** - open-source system for automating deployment, scaling, and management of containerized applications
Create SFC

1. Create container (template, location, etc.)
2. Spin up container (flavor, image)
3. Attach SFC interfaces
4. Update interface IDs
5. Boot up SFC (initial configuration)
6. Configure interfaces, bridge-domains, etc.
7. Container
CONTAINER SERVICE CHAINING SOLUTION

NETWORK DATA ANALYTICS AND FEEDBACK LOOP

Controller → Management agent → VPP

Control Plane

Data Plane

Feedback loop

Network data analytics

Data analytics platform

Spark notebook → Spark

Hadoop

Kafka → Gobblin
REFERENCES

- https://www.mirantis.com/blog/which-nfv-orchestration-platform-best-review-osm-open-o-cord-cloudify/
Demo
Thank you!