OSM Hackfest – Session 10
Service Function Chaining

Davide Borsatti (University of Bologna)
What is Service Function Chaining?
A couple words on Service Function Chaining (SFC)

Concatenation (*chaining*) of basic services or (virtualized) network/service functions…

…in order to obtain a composite service, spanning over the whole network domain
A couple words on Service Function Chaining (SFC)
Service Function Chaining over an SDN Domain

Traffic steering is handled by OpenFlow-capable switches, controlled by a SDN Controller
Service Function Chaining over an SDN Domain

Virtual Machines → br-int

Other OpenStack Nodes → br-tun

External networks → br-ex

Virtualized Ethernet Cable
SFC Status in OSM

- Available since OSM Release 3
- SFC only available using: **Openstack Queens (or higher) with Networking-SFC**
- Only supports Asymmetrical Chains
- SFC Encapsulation using MPLS
- Traffic classification based in the following fields:

<table>
<thead>
<tr>
<th>IP Protocol</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP Address</td>
<td>Destination IP Address</td>
</tr>
<tr>
<td>Source Port</td>
<td>Destination Port</td>
</tr>
</tbody>
</table>
Separate ingress and egress ports

Modifying the Information Model and Resource Orchestrator

Status: Complete.
Symmetrical/Asymmetrical Chains

Currently only Asymmetrical Chains are supported. **Status:** In development.
Non-Transparent Service Functions

Currently not supported.  
**Status:** Research ongoing.

VNF# 1  
(SFC Aware)

SFC Proxy

VNF #2  
(SFC Unaware)

SFC Proxy
Branching and Joining

Currently not supported.  
**Status:** Research ongoing.
Service Function Tap

Currently not supported. 
**Status:** Research ongoing.
OSM SFC descriptors

rsp: #redered service paths
   - id: rsp-id
     name: rsp-name
     vnfd-connection-point-ref:
     - member-vnf-index-ref: vnf-index
       vnfd-id-ref: vnf-id
       order: order-int
       vnfd-egress-connection-point-ref: egress-cp
       vnfd-ingress-connection-point-ref: ingress-cp
List of VNFs composing the Rendered Service Path

```
osp: #redered service paths
  - id: rsp-id
    name: rsp-name
    vnfd-connection-point-ref:
      - member-vnf-index-ref: vnf-index
        vnfd-id-ref: vnf-id
        order: order-int
        vnfd-egress-connection-point-ref: egress-cp
        vnfd-ingress-connection-point-ref: ingress-cp
```
**OSM SFC descriptors**

**rsp**: #redered service paths
- **id**: rsp-id
  - **name**: rsp-name
  - **vnfd-connection-point-ref**:  
    - **member-vnf-index-ref**: vnf-index  
    - **vnfd-id-ref**: vnf-id  
    - **order**: order-int  
  - **vnfd-egress-connection-point-ref**: egress-cp  
  - **vnfd-ingress-connection-point-ref**: ingress-cp

*Identifier of the VNFD and its index inside the “constituent-vnfd” list*
**OSM SFC descriptors**

** rsp: #redered service paths  
- id: rsp-id  
  name: rsp-name  
  vnfd-connection-point-ref:  
- member-vnf-index-ref: vnf-index  
  vnfd-id-ref: vnf-id  
  order: order-int  
  vnfd-egress-connection-point-ref: egress-cp  
  vnfd-ingress-connection-point-ref: ingress-cp  

Specifies the order of the different VNFs inside the RSP
OSM SFC descriptors

rsp: #redered service paths
- id: rsp-id
  name: rsp-name
  vnfd-connection-point-ref:
    - member-vnf-index-ref: vnf-index
      vnfd-id-ref: vnf-id
      order: order-int
      vnfd-egress-connection-point-ref: egress-cp
      vnfd-ingress-connection-point-ref: ingress-cp

Specifies from which connection point the traffic will enter and exit the VNF
OSM SFC descriptors

- classifier:
  - id: classifier-id
  name: classifier-name
  **match-attributes:**
  - destination-ip-address: dst-ip
  destination-port: dst-port
  id: match-id
  ip-proto: ip-proto-id
  source-ip-address: src-ip
  source-port: src-ip
  rsp-id-ref: rsp-id
  member-vnf-index-ref: vnf-index
  vnfd-connection-point-ref: vnf-cp
  vnfd-id-ref: vnf-id

Specifies the characteristics of the traffic we want to steer inside the RSP
- classifier:
  - id: classifier-id
    name: classifier-name
    match-attributes:
      - destination-ip-address: dst-ip
        destination-port: dst-port
      id: match-id
      ip-proto: ip-proto-id
      source-ip-address: src-ip
      source-port: src-ip
      rsp-id-ref: rsp-id
      member-vnf-index-ref: vnf-index
      vnfd-connection-point-ref: vnf-cp
      vnfd-id-ref: vnf-id

Desired IP Protocol numeric identifier. Ex:
1  →  ICMP
6  →  TCP
17 → UDP
OSM SFC descriptors

- classifier:
  - id: classifier-id
  - name: classifier-name
  - match-attributes:
    - destination-ip-address: dst-ip
    - destination-port: dst-port
    - id: match-id
    - ip-proto: ip-proto-id
    - source-ip-address: src-ip
    - source-port: src-ip
    - rsp-id-ref: rsp-id
    - member-vnf-index-ref: vnf-index
    - vnfd-connection-point-ref: vnf-cp
    - vnfd-id-ref: vnf-id

Source and Destination IP addresses and port (if TCP or UDP)
OSM SFC descriptors

- classifier:
  - id: classifier-id
  name: classifier-name
  match-attributes:
  - destination-ip-address: dst-ip
  destination-port: dst-port
  id: match-id
  ip-proto: ip-proto-id
  source-ip-address: src-ip
  source-port: src-ip
  rsp-id-ref: rsp-id
  member-vnf-index-ref: vnf-index
  vnfd-connection-point-ref: vnf-cp
  vnfd-id-ref: vnf-id

RSP linked to this classifier
OSM SFC descriptors

- classifier:
  - id: classifier-id
  name: classifier-name
  match-attributes:
    - destination-ip-address: dst-ip
    destination-port: dst-port
    id: match-id
    ip-proto: ip-proto-id
    source-ip-address: src-ip
    source-port: src-ip
    rsp-id-ref: rsp-id
  member-vnf-index-ref: vnf-index
  vnfd-connection-point-ref: vnf-cp
  vnfd-id-ref: vnf-id

VNF and relative connection point from which the “matching” flows will enter the RSP
Hands On scenario

Flow Classifier #1:
- IP Proto: TCP
- Src IP: 10.0.0.11
- Dst IP: 10.0.0.12
- Dst Port: 80

Image:
- ubuntu-16.04
SFC descriptors

Download and onboard on your OSM the following descriptors:

http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/endpoint_vnfd.tar.gz

http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/vnf_mpls_vnfd.tar.gz

http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/sfc_mpls_nsd.tar.gz
SFC descriptors

- classifier:
  - id: class1
    match-attributes:
      - destination-ip-address: 10.10.10.12
      - destination-port: 80
      - ip,proto: 6
      - source-ip-address: 10.10.10.11
      - source-port: 0
      - member-vnf-index-ref: 1
    name: class1-name
    rsp-id-ref: rsp1
    vnfd-connection-point-ref: vnf-data
    vnfd-id-ref: endpoint_vnfd

  rsp: #redered service paths
  - id: rsp1
    name: rsp1-name
    vnfd-connection-point-ref:
      - member-vnf-index-ref: 3
      - order: 0
    vnfd-egress-connection-point-ref: vnf-cp0
    vnfd-id-ref: vnf_nsh_vnfd
    vnfd-ingress-connection-point-ref: vnf-cp0
Segment Routing with OSM Demo
Segment Routing For SFC

- A Source Routing approach to Service Function Chaining

- The whole list of SF of the chain is inserted directly from the ingress node of the SR domain by means of a segment list. Each segment representing a function of the chain

- These segment identifiers can be expressed in two ways:
  - MPLS labels
  - IPv6 addresses
SRv6 descriptors

http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/clean_vnfd.tar.gz

http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/srv6_vnfd.tar.gz

http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/packages/4node_srv6_nsd.tar.gz
SRv6 actions - create-chain

vnf-configuration:
  juju:
    charm: srv6
  config-primitive:
    - name: create-chain
      parameter:
        - name: dest
          data-type: STRING
          default-value: ""
          mandatory: true
        - name: list
          data-type: STRING
          default-value: ""
          mandatory: false
        - name: source
          data-type: STRING
          default-value: ""
          mandatory: false
        - name: proto
          data-type: STRING
          default-value: ""
          mandatory: false
        - name: port-dest
          data-type: STRING
          default-value: ""
          mandatory: false
        - name: port-source
          data-type: STRING
          default-value: ""
          mandatory: false
SRv6 actions - *del-chain, update-chain*

vnf-configuration:
juju:
  charm: srv6
config-primitive:
  - name: del-chain
    parameter:
      - name: dest
        data-type: STRING
        default-value: "
        mandatory: true

vnf-configuration:
juju:
  charm: srv6
config-primitive:
  - name: encap-chain
    parameter:
      - name: dest
        data-type: STRING
        default-value: "
        mandatory: true
      - name: list
        data-type: STRING
        default-value: "
        mandatory: true
SRv6 Hands on - Scenario (4node_srv6_nsd)

Management Network

Endpoint (srv6_vnfd)  
2001:DB8:1234::1

VNF#1 (clean_vnfd)  
2001:DB8:1234::F

VNF#2 (clean_vnfd)  
2001:DB8:1234::FF

Endpoint (clean_vnfd)  
2001:DB8:1234::2

IPv6 Data Network  
(2001:DB8:1234::/64)
SRv6 Hands on - First Chain

SRv6 chain#1:
- IP Proto: TCP
- Src IP: 2001:DB8:1234::1
- Dst IP: 2001:DB8:1234::2
- Dst Port: 80
- VNF List: 2001:DB8:1234::F

IPv6 Data Network (2001:DB8:1234::/64)
SRv6 Hands on - Updated Chain

Management Network

Endpoint (srv6_vnfd)

VNF#1 (clean_vnfd)

VNF#2 (clean_vnfd)

Endpoint (clean_vnfd)

SRv6 chain#2:
- IP Proto: TCP
- Src IP: 2001:DB8:1234::1
- Dst IP: 2001:DB8:1234::2
- Dst Port: 80
- VNF List: 2001:DB8:1234::F, 2001:DB8:1234::FF

IPv6 Data Network (2001:DB8:1234::/64)

© ETSI 2019