NFV Research Group
Internet-Draft
Intended status: Informational

J. Chin
Cisco Systems

February 3, 2016

A Model Based Approach to Dynamic Service Chaining
draft-chin-nfvrg-model-dynamic-service-chain-01

Abstract

In the context of Network Function Virtualization (NFV), a service chain refers to a group of Virtual Network Functions (VNFs) deployed in a virtual infrastructure manager (VIM) environment, such as OpenStack, to apply a set of networking services to users' traffic passing through it.

Depending on the service requirements, the service chain may consist of a single VNF or multiple VNFs deployed serially, or in parallel, and the Virtual Network Forwarding Graph (VNF-G) describes the topologies of how these VNFs are interconnected together via virtual links to provide different networking services.

After a service chain is deployed, its user may require to modify the VNF-G from time to time, and on-demand, to meet changing service requirements. Therefore, operators may require to support a high degree of flexibility to orchestrate the provisioning and dynamic modifications of VNFs in service chains.

This document presents a hierarchical model based approach to allow the NFV Orchestrator (NFV-O) and the VNF Manager (VNF-M) to support the dynamic provisioning and on-demand modifications of VNFs in service chains.

Status of This Memo

Copyright Notice

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Table of Contents

1. Introduction

Network Function Virtualization (NFV) has provided operators with the potential to reduce cost, while at the same time, it increases business agility, flexibility and accelerate the time to market new services. For the users, NFV promises the delivery of networking services with a quicker turnaround time than traditional hardware based networking services. Users may also expect NFV to give them a greater degree of flexibility in the way they consume networking services.

As traditional networking services are increasingly being supported by Virtual Network Functions (VNFs), it is important for operators to provide their end users with greater flexibility to modify their services "on the fly". For example, a user may have initially deployed a virtual routing function, but now requires another virtual firewall to be added due to new security requirements. As such, the existing service chain needs to be expanded or modified, and ideally carried out with minimal service disruptions.

2. A Hierarchical Model Based Approach to Service Chaining

The purpose of interconnecting Virtual Network Functions (VNFs) together to create a service chain is to provide its user with a set of services which cannot be fulfilled by a single VNF alone. VNFs run in software and they are interconnected together via virtual links, for example by OpenStack Neutron service and OpenVSwitch (OVS).

A service chain of VNFs is functionally equivalent to hardware appliances physically interconnected but by nature there are notable variations in specifications and performance between software and hardware based devices.

Just like in hardware devices, changes to the networking services may require modifications of the network topology, and this may lead to "re-wiring" of network connectivity between software VNFs, and hardware appliances alike.

To handle such topology modifications, the NFV orchestrator (NFV-O) and VNF Manager (VNF-M) must ensure newer VNFs added to the service chain are correctly connected to the existing VNFs and virtual links, and all VNFs in the service chain are properly configured to effect the new services.

When a particular VNF in a service chain is no longer required, the NFV-O must coordinate with the VNF-M to gracefully terminate the VNF without affecting the remaining VNFs in the service chain. The NFV-O must reverse all device configuration changes it made in the service chain to restore the service to its previous state.

The NFV-O should support conditional checks to prevent unsupported topologies in a service chain.

2.1. Anchor VNF

The hierarchical model based approach discussed in this document firstly establishes a single VNF in the service chain as the anchor VNF, which is shown in Figure 1. There are no special requirements for the anchor VNF to support any particular network service or technology, but rather the anchor VNF simply means it is the first VNF to be deployed at the start of service. From the anchor VNF, the service chain can expand eastwards and southwards.

```
        ------------------------------------------> expands eastwards
        *Anchor*
    |   +--------+        +--------+         +--------+
    |   |1st Tier| <----> |2nd Tier| <-----> |3rd Tier| <-----> ...
    |   |  VNF   |        |  VNF   |         |  VNF   |
    |   +--------+        +--------+         +--------+
    |       ^                 ^                  ^
    |       |                 |                  |
    |       V                 V                  V
    |   +--------+        +--------+         +--------+
    |   |1st Tier| <----> |2nd Tier| <-----> |3rd Tier| <-----> ...
    |   |  VNF   |        |  VNF   |         |  VNF   |
    |   +--------+        +--------+         +--------+
    |       ^                 ^                  ^
    |       |                 |                  |
    |       V                 V                  V
    |   +--------+        +--------+         +--------+
    |   |1st Tier| <----> |2nd Tier| <-----> |3rd Tier| <-----> ...
    |   |  VNF   |        |  VNF   |         |  VNF   |
    |    +--------+        +--------+         +--------+
    |       ^                 ^                  ^
    |       |                 |                  |
    |       V                 V                  V
    V      ...               ...                ...
   expands
   southwards
```

Figure 1: Hierarchial service chain with Anchor VNF

2.2. Define the "End State" Service Chain Topology

An "End State" topology typically reflects the maximum number of VNFs in a service chain required to support the most complex NFV services. Operationally, it is a challenge to manage large and complex service chains in a virtual environment, especially if the maximum number of VNFs allowed in a service and its topology are both unknown.

However, the hierarchical model does not impose any restriction on the maximum number of VNFs allowed in any service chain topology. The "end state" topology can be expanded with additional VNFs or the number of VNFs can be reduced anytime to support changing user requirements. This "end state" topology should be translated into a service catalog to be presented in a customer facing service (CFS) portal.

Similarly, there is no restriction imposed on the VNF types in the model - they can be of the same type or entirely different. To use a case in point, we can deploy multiple instances belonging to the same brand of virtual firewall in a service chain to deliver a multi-tier firewall security service. The VNF-M and NFV-O are responsible for onboarding VNFs in the VIM and orchestrating the VNFs device configurations to deliver the desired network service regardless of the VNF types.

To help illustrate the concepts behind our model based approach, we shall study an example in Figure 2 with a maximum number of six VNFs as its "end state" topology, arranged in a 2-dimensional array (i.e 2 rows x 3 columns). The anchor VNF refers to the VNF at A1.

```
             *Anchor*
Ingress |   +--------+   |   +--------+   |   +--------+   |  Egress
  VL    |---| VNF A1 |---|---| VNF B1 |---|---| VNF C1 |---|    VL
segment |   +--------+   |   +--------+   |   +--------+   | Segment
()------|                |                |                |------()
        |   +--------+   |   +--------+   |   +--------+   |
        |---| VNF A2 |---|---| VNF B2 |---|---| VNF C2 |---|
        |   +--------+   |   +--------+   |   +--------+   |
```

Figure 2: "End State" service chain topology with six VNFs

It is necessary to define the different permutations of VNFs allowed in the service chain, such as virtual router, firewall, load balancer, or IPS etc. The NFV-O is responsible to apply the correct device configurations based on the types, brands, and placement of VNFs in the topology.

Figure 3 shows an example of a list of possible VNF permutations required in the service chain to deliver different sets of network services:

anchor VNF A1	VNF B1	VNF C1
router	firewall	IPS
router	firewall	load balancer
firewall	firewall	firewall
firewall	firewall	
IPS	firewall	
IPS	router	
load balancer	router	

Figure 3: Example of a list of VNFs permutations in a service chain

Mapping the list of VNF permutations in Figure 3 above to the six VNF topology example in Figure 2, the "A1" anchor VNF can be a virtual router, firewall, load balancer or IPS; the "B1" VNF, if present, can be a virtual firewall or router; and "C1" VNF, if present, can be a virtual IPS, firewall, or load balancer. For the second row of VNFs - A2, B2, and C2 - its most common deployment scenario is to create an identical VNF of the top row VNF to support High Availability (HA), using either standards based or proprietary protocols. For example, if the anchor VNF is a virtual router, then the VNF at A2 will be the same virtual router and both devices shall be configured to support HA functions in the service chain, e.g. provide gateway redundancy via Virtual Router Redundancy Protocol (VRRP) to devices on ingress VL segment.

A NFV service request may provision a service chain made up of any number of VNFs from one to the maximum six.

In every service chain, there are virtual links connecting the VNFs in the virtual domain to the physical environment. Traffic from users will enter and exit the service chain via the ingress and egress segments. The underlay network, for example, can be VxLAN in a data center environment or IP/MPLS over a WAN environment.

2.3. Start of service chain at the anchor VNF

Referring to the service chain example in Figure 2, the anchor VNF has utmost significance in the model. Regardless of the number of VNFs requested by a user, the NFV-O and VNF-M must always create the anchor VNF at A1 at the start of any service deployment. If the user requests for a single virtual router, then this virtual router will be placed at the A1 position in the service chain topology.

In addition, the NFV-O orchestrating the service must instruct the VNF-M and the VIM to provision three virtual link segments - ingress, egress, and A1-B1 - and attach the anchor VNF to these VL links. Certain types of VNF may use a dedicated heartbeat link for HA purposes. If a heartbeat link is required by a particular VNF, then an additional A1-A2 virtual link must be created. All VNFs are also connected to the VNF-M via a shared management network which is pre-created. The anchor VNF and the virtual link segments, including the optional heartbeat link, are shown in Figure 4.

```
      ========================+====================== out-of-band
                              |                        management
                     (anchor) |       A1-B1          (pre-created)
                  |      +----+---+     |
                  |------| VNF A1 |xx---|
                  |      +----+---+--+ |
       >>>  ------|           |      | |
      Ingress     |         A1-A2    | |
      VL segment  |    (If required  | |
                  |    for dedicated | |
                       heartbeat to  |
                       future A2 VNF |
                                     |
      ===============================+===============================
                                          Egress VL segment >>>
```

Figure 4: Onboarding VNF at A1 and its virtual links

2.4. Initial Configurations and Traffic Flow

After the anchor VNF at A1 is deployed, the VNF-M should apply the initial device configurations (day-0) which will put its virtual network interface connected to the A1-B1 virtual link in the disabled state. The NFV-O will reactivate this virtual network interface when a new VNF at B1 is added to the service chain. Similarly, if the A1-A2 virtual link for HA is required, the NFV-O will place it in the disabled state until a VNF at A2 is added to the service chain.

The NFV-O should apply any necessary additional device configurations (day-1) in the anchor VNF to enable the requested services. The NFV-O may need to orchestrate physical network devices in the outside network to effect end-to-end service chaining to the VNF. At this stage, the service chain with the anchor VNF is considered operational. Depending on the networking service provided and the traffic patterns of the user applications, the traffic flow across the service chain can be uni-directional or bi-directional.Traffic from the outside can enter and exit the service chain via the ingress and egress segments, as shown in Figure 5.

```
              Traffic Flow
    <<<----------------------------+
                  (anchor)  A1-B1  |
    Ingress  |   +--------+   |    |
       VL    |---| VNF A1 |xx-|    ^      >>>> service chain
    segment  |   +--------+-+ |    ^           expands
   ()--------|          V   | |    |
             | service  V   | |    V
       >>>   |  chain   V   | |    V
             | expands  V   | |    |         Traffic Flow
                            |      +-------------------------->>>
    ========================+========================================
                                            Egress VL segment >>>
```

Figure 5: Start of service chain and traffic flow

2.5. Expanding the service eastwards and southwards

Assuming the user now requests for modifications to the existing service and this request requires a new virtual firewall to be deployed and added to the service chain. To fulfill the request, the existing service chain can expand eastwards or southwards away from the anchor VNF at A1. The NFV-O and VNF-M must implement conditional checks to only allow the next VNF to be deployed at either B1 or A2 position.

Supposed the new virtual firewall shall be deployed at B1 position, the NFV-O and NFV-M must now expand the service eastwards. The VNF-M does not need to create the ingress and egress virtual link segments since these are now pre-existing segments previously created during the service deploy of the anchor VNF. Instead, the VNF-M must create a new B1-C1 virtual link segment and a B1-B2 heartbeat link if necessary, as shown in Figure 6. All subsequent service chain expansions eastwards will repeat the same sequence described here.

```
    ------------------------------------------------+
                  (anchor)  A1-B1            B1-C1  |
    Ingress  |   +--------+   |   +--------+   |    |
      VL     |---| VNF A1 |---|---| VNF B1 |xx-|    | >>>> service chain
    segment  |   +--------xx+ |   +--------+-+ |    |        expands
   ()--------|          V   | |          V   | |    |
             | service  V   | | service  V   | |    |
       >>>   |  chain   V   | |  chain   V   | |    |  Traffic Flow
             | expands  V   | | expands  V   | |    |    Direction
                 (HA)       |                |      +---------------->>>
    ====================================================================
                                                  Egress VL segment >>>
```

Figure 6: Expansion of service chain eastwards

Similar to the anchor VNF at A1, the VNF-M will attach the new VNF at B1 to three virtual link segments - the new B1-C1 segment and the existing A1-B1 and egress VL segments. If the new VNF require a dedicated heartbeat link for HA purposes, then the VNF-M must create a B1-B2 virtual link for it as well.

The VNF-M must apply the day-0 configurations to the VNF at B1, and again the VNF-M must put its virtual network interface connected to B1-C1 in the disabled state. After the VNF at B1 has come up online, the NFV-O must do three things.

Firstly, the NFV-O must apply the necessary day-1 device configurations to the VNF at B1 to effect the new network service.

Secondly, it must re-configure the anchor VNF at A1 to activate its virtual network interface connected to A1-B1 and disable its virtual network interface connected to the egress VL segment.

Lastly, it must apply any additional device configurations changes necessary to the anchor VNF at A1 to effect the new service together with the new VNF at B1.

Ideally, the NFV-O should be transaction based to track all the service changes it made in a last-in-first-out fashion. For example, when the VNF at B1 is deleted the NFV-O can reverse all configuration changes it did to the anchor VNF.

If another new VNF is required at A2 position for HA purposes, the service can expand southwards. In this case, VNF-M does not need to create any new virtual link segment but rather, it attaches the new VNF at A2 to the pre-existing virtual link segments previously created by A1 - namely ingress, egress and A1-B1 VL segments. If both VNFs require a dedicated virtual link for heartbeat purposes, then the VNF-M must attach the VNF at A2 to the A1-A2 segment. The NFV-O must apply the day-1 device configurations to the new VNF at A2, and if necessary, apply device configurations changes to the VNF at A1 to activate and synchronize HA services.

In all scenarios, the NFV-O must wait for the VNF-M to bring up the VNF before it apply any configuration changes to service chain. This allows the service chain to be modified with minimal impact to the network service provided by active VNFs.

The sequence of steps described here should be repeated when more VNFs are added to the service chain. For example, after A1 and B1 VNFs are active, a new VNF can now be deployed at either A2, B2, or C1 positions. The dynamic expansion of the service chain can continue until it reaches the "end state" topology, if one is defined. The NFV-O or the customer service portal can impose this topological restriction. Figure 7 shows the service chain all six VNFs deployed.

```
               (anchor)  A1-B1            B1-C1            C1-D1  ..
 Ingress  |   +--------+   |   +--------+   |   +--------+   |
    VL    |---| VNF A1 |---|---| VNF B1 |---|---| VNF C1 |---| <- new
 segment  |   +--------+-+ |   +--------+-+ |   +--------+-+ |   VNFs
()--------|              | |              | |              | |
          |   +--------+ | |   +--------+ | |   +--------+ | |
    >>>   |---| VNF A2 |-|-|---| VNF B2 |-+-|---| VNF C2 |-|-| <- new
          |   +------+-+ | |   +------+-+ | |   +------+-+ | |   VNFs
                     |   |            |   |            |   |
 ====================+===+============+===+===============+==========
                                             Egress VL segment  >>>
```

Figure 7: Dynamic service chain expansion

3. Dependency Relationship in Service Chain

The model based approach presented in this document establishes dependency relationships between VNFs in the service chain topology. The VNF at A1 is the anchor VNF in every service chain, hence all VNFs are dependent on it. What this means is that the NFV-O and VNF-M must not allow the anchor VNF to be deleted before all other VNFs in the service chain are deleted first. And, because our model based approach is designed to expand the service chain eastwards and southwards, every VNF in the service chain topology is dependent on the VNF to its left and on top, if any. For example, the VNF at C1 is dependent on the VNF at B1. Therefore, the VNF at B1 cannot be deleted ahead of the VNF at C1.

For any VNFs in the bottom row, they are only dependent on the VNF in the top most row. For example, the VNF at B1 cannot be deleted ahead of any VNFs at B2 or B3. Figure 8 shows the dependency relationships between the VNFs based on the original example shown in Figure 2.

```
                  (anchor)  A1-B1            B1-C1            C1-D1
    Ingress  |   +--------+   <<  +--------+   <<  +--------+   |
      VL     |---| VNF A1 |---|---| VNF B1 |---|---| VNF C1 |---|
    segment  |   +--------+-+ |   +--------+-+ |   +--------+-+ |
   ()--------|       ^^   | | |       ^^   | | |       ^^   | | |  .....
             |   +--------+ | |   +--------+ | |   +--------+ | |
       >>>   |---| VNF A2 |-|-|---| VNF B2 |-+-|---| VNF C2 |-|-|
             |   +------+-+ | |   +------+-+ | |   +------+-+ | |
                        |   |            |   |            |   |
    ====================+===+============+===+================+=========
                                                   Egress VL segment >>>
    << and ^^ are dependency
```

Figure 7: Dependency Relationships

4. Service Chain Rollback

The NFV-O and VNF-M must support the reversal of the service chain by observing the dependency relationships between the VNFs. When a VNF is deleted in the service chain, the NFV-O must not only instruct the VNF-M to terminate the virtual instance, but it must reverse any configuration changes it previously made in the service chain. For example, when the VNF at C1 is deleted, the NFV-O must now ensure user traffic will exit via the VNF at B1 instead.

5. Caveats

The hierarchical model based approach presented in this document allows a NFV service chain to be dynamically expanded and contracted. However, due to the dependency relationships established by VNFs in the model, there are two caveats we should be aware of.

Firstly, if the user requests to replace the anchor VNF with an entirely different type of virtual device, then the NFV-O and VNF-M have to unprovision the entire service chain and re-deploy the anchor VNF in the VIM, then expand the service chain with the rest of the VNFs.

Secondly, the NFV-O must not allow a particular VNF which is depended upon by another VNF to be deleted. For example, we cannot delete a VNF at B1 while keeping the VNF at C1.

In both scenarios above, the NFV-O and VNF-M have to delete the existing service chain and re-deploy it.

6. IANA Considerations

This draft does not have any IANA considerations.

7. Security Considerations

This draft does not have any Security considerations.

8. Informative References

[1] "OpenStack," http://www.openstack.org/

[2] "OpenStack Neutron," https://wiki.openstack.org/wiki/Neutron

[3] "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6," https://tools.ietf.org/html/rfc5798"

[4] ETSI GS NFV 002 v1.2.1 (2014-12): "Network Function Virtualisation (NFV); Architectural Framework," http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf

[5] ETSI GS NFV 004 v1.1.1 (2013-10): "Network Function Virtualisation (NFV); Virtualization Requirements," http://www.etsi.org/deliver/etsi_gs/NFV/001_099/004/01.01.01_60/gs_NFV004v010101p.pdf

[6] ETSI GS NFV-INF 001 v.1.1.1 (2015-01): "Network Function Virtualisation (NFV); Infrastructure Overview," http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_NFV-INF001v010101p.pdf

Authors' Addresses

Jonathan Chin
Cisco Systems
8 Changi Business Park Avenue 1
#05-51, UE BizHub East
Singapore 486018

Phone: +65 9479-0029
Email: jonachin@cisco.com