Network Working Group   G. Chen
Internet-Draft   H. Deng
Intended status: Informational   China Mobile
Expires: April 30, 2015   October 27, 2014

IPv6 Considerations for Network Function Virtualization (NFV)
draft-chen-v6ops-nfv-ipv6-00

Abstract

NFV adoption is gaining significant momentum, driven largely by the need to improve service agility and reduce operational cost. IPv6 is a fundamental feature should be enabled. This memo desribes the layered NFV components and typical implementations. The IPv6 considerations have been elaborated to each component in order to consoliate IPv6 demands across entire NFV system.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Network Virtualization Function (NFV) is a new trend for the telcom industry revolution. It leverages IT infrastructure to take over the telcom functions. Driven largely by the need to improve service agility and reduce operational cost, virtualization has been adopted in the NFV architecture. Server, storage, and network resources are abstracted from their physical functions, e.g. processor, memory, I/O controllers, disks, network and storage switches, etc, into pools of functionality which can be managed functionally regardless of their implementation or location. In other words, all servers, storage, and network devices can be aggregated into independent pools of resources to be used as needed, regardless of the actual implementation of those resources.

Depending on the virtualization, NFV system gains good scalability. However, this expansion also can't survive on the exhaunsting IPv4 address space. IPv6 is definitely the only way out to this pressing needs, because the larger IP address space makes it easier to manage large cloud infrastructures. The memo intends to enumurate IPv6 considerations regarding to the different components in the NFV architecture. It's expected early adopters could reconsider the way they design NFV cloud network so as to get more scalable and manageable infrastructure.

2. Overview on IPv6 Considerations in the NFV Architecture

European Telecommunications Standards Institute (ETSI) has defined the NFV architecure framework [GS_NFV_002]. NFV system has been structured from three main working domain in the high-level framework as shown in the Figure 1.

```
+-------------------------------------+   +---------+
| Virtualised Network Functions(VNFs) |   |  NFV    |
| +------+   +------+        +------+ |   |Managment|
| | VNF  |   |  VNF |  ..... | VNF  | |   |  and    |
| +------+   +------+        +------+ |   |Orchest  |
+-------------------------------------+   |-ration  |
+-------------------------------------+   |         |
|       NFV Infrastructure(NFVI)      |   |         |
| +-------+   +-------+   +-------+   |   |         |
| |Virtual|   |Virtual|   |Virtual|   |   |         |
| |Compute|   |Storage|   |Network|   |   |         |
| +-------+   +-------+   +-------+   |   |         |
| +-------------------------------+   |   |         |
| |     Virtualisation Layer      |   |   |         |
| +-------------------------------+   |   |         |
|       Hardware Resource             |   |
| +--------+ +--------+ +--------+    |   |         |
| |Compute | |Storage | |Network |    |   |         |
| |Hardware| |Hardware| |Hardware|    |   |         |
| +--------+ +--------+ +--------+    |   |         |
+-------------------------------------+   +---------+
```

Figure 1: High-Level NFV Framework

We try to document the effort made to enable IPv6 across all components as illustrated in the overall NFV architecture. The Figure 2 lists components, which should take specific considerations to perform IPv6 functions. For each component, a typical implementation has been exemplified. Those implementations are leveraged towards the realization of IPv6-capable NFV system.

NFV Components	Implementations Instance
VI Management	Openstack
Virtual Network	OpenDayLight, OpenVSwitch
Virtualisation Layer	KVM, Librvirt,Linux Kernel
Network Hardware	DPDK
VNF	OpenEPC

Figure 2: IPv6 Relevant NFV Components

3. IPv6 Considerations on VIM

Virtualised Infrastructure Management (VIM) comprises the functionalities that are used to control and manage the interation of a VNF with computing , storage and network resources under its authority, as well as their virtualisation. We can clearly see OpenStack gaining more and more traction. Openstack is composed by several core projects, e.g., Compute (Nova), Network (Neutron), Image (Glance), Object Storage (Swift) and Block Storage (Cinder) and etc. The major concerns of IPv6 capability should be implemented into the Neutron project. Neutron could offers sophisticated networking functionality to coordinate network resources. Numerous IPv6 features could be merged into Neutron.

In general, Neutron is responsible for all topologies work in a multi-tenant environment. IPv6 enable Neutron is able to allow IPv6 address static configuration and auto assignment. The internal IPv6 communications between Virtual Machines (VMs) and external IPv6 interconnection via Neutron and external router/border gateway should be supported. The following considerations faciliate the IPv6 communications goals:

o Address Management: several IPv6 configuration modes such as SLAAC [RFC4862] , DHCPv6 Stateless [RFC3736] and DHCPv6 Stateful [RFC3315] are recommended to be supported. It includes the ability for a user to create a port on a IPv6 subnet and assign a specific IPv6 address or muliple IPv6 addresses to the port and have it taken out the DHCP address pool. Prefix delegation is also expected to be used to automatically configure neutron routers with prefixes so that IPv6 prefixes are obtained and renumbering can be done automatically.

- External IPv6 Interconnections: IPv6 subnet could be routed via Layer 3 (L3) agent to an external IPv6 network. Both VLAN and overlay (e.g. GRE, VXLAN) subnet attached to VMs can be used to support multiple L3 agents for a given external network to support scaling. Neutron scheduler could be used to assign virtual routers to the L3 agents. Openstack takes the concept of floating IP to allow internal servers to be accessed from external networks. That is the normal cases in IPv4. Given the large address space that IPv6 offers, the floating IP may be unnecessary. End-to-end native IPv6 is more desirable than any of the transition solutions.

- Floating IP: Floating IP is used in Openstack to make internal servers to be accessible from external Internet. Floating IP support for IPv6 Addresses could be used for internal IPv6 connecting to external IPv6.

- Security Group: security group is set to interrogate and/or disallow IP flows. Full support for IPv6 TCP/UDP/ICMP in IPv6 security groups are necessary in a IPv6 environment.

- User Interfece and Command Line (CLI): it's important for users to manipulate networks with IPv6 features. During the network, subnet, router creation, it should have the option to allow user to specify the type of address management they would like. This includes the supports via Neutron API (Restful and CLI) as well as via Openstack UI (i.e., Horizon). It's also esstential to enable that feature to be able to specify Floating IPs via Neutron API (restful and CLI) and control and manage all IPv6 security group capabilities via Neutron/Nova API (restful and CLI) .

4. IPv6 Considerations on Vitrual Network

Virtual Networks is used to isolate resources and network overlays. It could be orchestrated by Openstack Neutron to lign network resources to be able to better address the requirements of rich multi-tenant environments. In order to make system more scaleable, Neutron adopts a plug-in model for various 3rd party components to provide the networking service. New technologies (e.g., software-defined networking (SDN)) are emerging to increase the flexibility and agility of the network, decoupling the control from the forwarding plane to make it easier to provision, automate and orchestrate network services. The OpenDaylight provides a plugin and a corresponding agent to enable integration with Neutron. IPv6 demands should also be considered in OpenDaylight softwares including a pluggable controller, interfaces and applications.

The target of IPv6-enable OpenDaylight is to make the overlay and underlay networks in the cloud architecture both being developed with IPv6. The OpenDaylight project, like OpenFlow, is a good initiative to accelarate the IPv6 transition. OpenFlow v1.3 could dynamically learn the Layer 3 IPv6 hosts. This can be facilitated by supporting the IPv6 Neighbor Discovery Protocol (NDP) or supporting DHCPv6. In this case, the OpenDaylight controller should has the ability to perform matching on IPv6 packets and pushes down a flow-table entry to each of the edge devices enabling the forwarding of these packets up to the controller or application to process.

Each of the underlay devices would need to support the optional IPv6 features of OpenFlow and support the required combinations of match/ action on the IPv6 header. This also includes the ability to support masking of address fields. Open vSwitch (OVS) is a typical effort to enable the IPv6 process with the overwhelming superiority, such as flexible controller in user-space and fast datapath in kernel. A IPv6-enable vSwitch should be able to support IPv6 flows via OpenFlow. The flow could be identified by the combination of any IPv6 features, such as IPv6 ND target, IPv6 source address or IPv6 destination address. The implementation of OVS would the dedicated IPv6 module to enable IPv6 forwarding.

5. IPv6 considerations on Virtualisation Layer

The virtualisation layer abstracts the hardware resources and decouples the VNF software from the underlying harware. It enables the software that implements the VNF to use the underlying virtualised infrasturecture. Typically, this type of functionality is provided for computing and storage resources in the form of hypervisors. In order to facilitate the management of different kinds of hypervisors, libvirt virtualization API is created to provide management tool for managing platform virtualization. The Figure 3 elaborates the relations of different components in virtualisation layer. The sub-section will describe the detailed consideration for each one.

```
+---------------------------------------+
|         VIM(e.g., Openstack)          |
+---------------------------------------+
|         Libvirt API                   |
+---------------------------------------+
| KVM  |  Xen |   ESX   |   others...   |
+---------------------------------------+
|  Linux OS   | Others ....             |
+---------------------------------------+
```

Figure 3: Virtualisation Layer Components

5.1. IPv6-enable Libvirt

Libvirt could provide a common and stable layer sufficient to securely manage VNF instances. libvirt provides all APIs needed to do the management, such as provision, create, modify, monitor, control, migrate and stop the instances. IPv6 network configurations can be enabled by the libvirt networking APIs, which is formulated by network XML format. Libvirt define network profile from different elements including general metadata, connectivity and addressing. To enable IPv6, each attributes shoule be configured properly. For the IPv6 addressing, Libvirt could take SLAAC as default and optionally enable DHCP services. Libvirt could configure static routes for IPv6 forwarding, but lack of supports for dynamic routing protocol.

5.2. IPv6-enable KVM

KVM should provied same operations cooresponding to Libvirt. It may be straight forward to enable IPv6 on KVM guests by configure the host machine and interfaces with IPv6 address. The necessary firewall rules could be also added to ip6tables on the host machine. NDIS driver in KVM also should be able to handle the IPv6 packages.

5.3. IPv6-enable Linux

Linux system should have to enable the IPv6 support in the kernel. Some interface configuration file should add IPv6 address information and restart the networking. Other consideration is the MTU setting. The MTU size of the NIC on Linxu defaults to 1500 bytes. It may be good to support Jumbo frames in the cloud infrastructure. Large MTU size not only gives you better network performance, but also provides you with workaround for software issues. It has been observed that many IPv6 packeges may exceed 1500-bytes. Therefore, it's very important to enable jumbo frames to avoid the corruption.

6. IPv6 Considerations on Network Hardware

Network hardware is capable of high-performance packet processing. There are optimized data plane solutions for the IP package processing. The Intel Data Plane Development Kit (DPDK) is a set of optimized software libraries and drivers, that enable high-performance data plane on network elements. The IPv6 demands to DPDK are targeted to support IPv6 forwarding, including IPv6 fragmentation reassembly. For the fast path, it would support IPv6 exact match flow classification.

7. IPv6 Considerations on VNF

The traditional mobile node functions would gradually be migrated to Vitrual Network Function (VNF). Examples of VNF are 3GPP Evolved Packet Core (EPC) network elements, e.g., Mobility Managment Entity (MME), Serving Gateway (SGW), Packet Data Network Gateway (PGW). VNF may remodel the network node functions into the different instances. For examples, the IPv6 relevant functions of SGW/PGW include PDN signaling processing, IPv6 data-plane filtering, classification, forwarding and IPv6 Charging control. Those IPv6 processing should also be supported in the new-built VNF instances.

8. IANA Considerations

This document makes no request of IANA.

9. Security Considerations

TBD

10. References

10.1. Normative References

[GS_NFV_002]
European Telecommunications Standards Institute, ETSI., "Network Functions Virtualisation (NFV); Architectural Framework", March 2009.

10.2. Informative References

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

Authors' Addresses

Gang Chen
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China

Email: phdgang@gmail.com

Hui Deng
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China

Email: denghui@chinamobile.com