IPv6 Gap Analysis

Release brahmaputra.1.0 (fe07244)

OPNFV

February 25, 2016

CONTENTS

Project IPv6, http://wiki.opnfv.org/ipv6_opnfv_project

Editors Bin Hu (AT&T), Sridhar Gaddam (RedHat)

Authors Sridhar Gaddam (RedHat), Bin Hu (AT&T)

Abstract

This document provides the users with top-down gap analysis regarding IPv6 feature requirements with OpenStack Liberty Official Release and Open Daylight Beryllium Official Release.

CHAPTER

ONE

IPV6 GAP ANALYSIS WITH OPENSTACK LIBERTY

This section provides users with IPv6 gap analysis regarding feature requirement with OpenStack Neutron in Liberty Official Release. The following table lists the use cases / feature requirements of VIM-agnostic IPv6 functionality, including infrastructure layer and VNF (VM) layer, and its gap analysis with OpenStack Neutron in Liberty Official Release.

Use Case / Requirement	Supported in Liberty	Notes
All topologies work in a multi-tenant environment	Yes	The IPv6 design is following the Neutron tenant networks model; dnsmasq is being used inside DHCP network namespaces, while radvd is being used inside Neutron routers namespaces to provide full isolation between tenants. Tenant isolation can be based on VLANs, GRE, or VXLAN encapsulation. In case of overlays, the transport network (and VTEPs) must be IPv4 based as of today.
IPv6 VM to VM only	Yes	It is possible to assign IPv6-only addresses to VMs. Both switching (within VMs on the same tenant network) as well as east/west routing (between different networks of the same tenant) are supported.
IPv6 external L2 VLAN directly attached to a VM	Yes	IPv6 provider network model; RA messages from upstream (external) router are forwarded into the VMs
IPv6 subnet routed via L3 agent to an external IPv6 network 1. Both VLAN and overlay (e.g. GRE, VXLAN) subnet attached to VMs; 2. Must be able to support multiple L3 agents for a given external network to support scaling (neutron scheduler to assign vRouters to the L3 agents)	1. Yes 2. Yes	Configuration is enhanced since Kilo to allow easier setup of the upstream gateway, without the user being forced to create an IPv6 subnet for the external network.

Continued on next page

Table 1.1 – continued from previous page

Use Case / Requirement	Supported in Liberty	Notes
Ability for a NIC to support both IPv4 and IPv6 (dual stack) address. 1. VM with a single interface associated with a network, which is then associated with two subnets. 2. VM with two different interfaces associated with two different networks and two different subnets.	1. Yes 2. Yes	Dual-stack is supported in Neutron with the addition of `Multiple IPv6 Prefixes` Blueprint
Support IPv6 Address assignment modes. 1. SLAAC 2. DHCPv6 Stateless 3. DHCPv6 Stateful	1. Yes 2. Yes 3. Yes	
Ability to create a port on an IPv6 DHCPv6 Stateful subnet and assign a specific IPv6 address to the port and have it taken out of the DHCP address pool.	Yes	
Ability to create a port with fixed_ip for a SLAAC/DHCPv6-Stateless Subnet.	**No**	The following patch disables this operation: https://review.openstack.org/#/c/129144/
Support for private IPv6 to external IPv6 floating IP; Ability to specify floating IPs via Neutron API (REST and CLI) as well as via Horizon, including combination of IPv6/IPv4 and IPv4/IPv6 floating IPs if implemented.	**Rejected**	Blueprint proposed in upstream and got rejected. General expectation is to avoid NAT with IPv6 by assigning GUA to tenant VMs. See https://review.openstack.org/#/c/139731/ for discussion.
Provide IPv6/IPv4 feature parity in support for pass-through capabilities (e.g., SR-IOV).	**To-Do**	The L3 configuration should be transparent for the SR-IOV implementation. SR-IOV networking support introduced in Juno based on the `sriovnicswitch` ML2 driver is expected to work with IPv4 and IPv6 enabled VMs. We need to verify if it works or not.
Additional IPv6 extensions, for example: IPSEC, IPv6 Anycast, Multicast	**No**	It does not appear to be considered yet (lack of clear requirements)
VM access to the meta-data server to obtain user data, SSH keys, etc. using cloud-init with IPv6 only interfaces.	**No**	This is currently not supported. Config-drive or dual-stack IPv4 / IPv6 can be used as a workaround (so that the IPv4 network is used to obtain connectivity with the metadata service)

Continued on next page

Table 1.1 – continued from previous page

Use Case / Requirement	Supported in Liberty	Notes
Full support for IPv6 matching (i.e., IPv6, ICMPv6, TCP, UDP) in security groups. Ability to control and manage all IPv6 security group capabilities via Neutron/Nova API (REST and CLI) as well as via Horizon.	Yes	
During network/subnet/router create, there should be an option to allow user to specify the type of address management they would like. This includes all options including those low priority if implemented (e.g., toggle on/off router and address prefix advertisements); It must be supported via Neutron API (REST and CLI) as well as via Horizon	Yes	Two new Subnet attributes were introduced to control IPv6 address assignment options: • `ipv6-ra-mode`: to determine who sends Router Advertisements; • `ipv6-address-mode`: to determine how VM obtains IPv6 address, default gateway, and/or optional information.
Security groups anti-spoofing: Prevent VM from using a source IPv6/MAC address which is not assigned to the VM	Yes	
Protect tenant and provider network from rogue RAs	Yes	When using a tenant network, Neutron is going to automatically handle the filter rules to allow connectivity of RAs to the VMs only from the Neutron router port; with provider networks, users are required to specify the LLA of the upstream router during the subnet creation, or otherwise manually edit the security-groups rules to allow incoming traffic from this specific address.
Support the ability to assign multiple IPv6 addresses to an interface; both for Neutron router interfaces and VM interfaces.	Yes	
Ability for a VM to support a mix of multiple IPv4 and IPv6 networks, including multiples of the same type.	Yes	
Support for IPv6 Prefix Delegation.	Yes	Partial support in Liberty
Distributed Virtual Routing (DVR) support for IPv6	**No**	Blueprint proposed upstream, pending discussion.
IPv6 First-Hop Security, IPv6 ND spoofing	Yes	
IPv6 support in Neutron Layer3 High Availability (keepalived+VRRP).	Yes	

CHAPTER

TWO

IPV6 GAP ANALYSIS WITH OPEN DAYLIGHT BERYLLIUM

This section provides users with IPv6 gap analysis regarding feature requirement with Open Daylight Beryllium Official Release. The following table lists the use cases / feature requirements of VIM-agnostic IPv6 functionality, including infrastructure layer and VNF (VM) layer, and its gap analysis with Open Daylight Beryllium Official Release.

Use Case / Requirement	Supported in ODL Beryllium	Notes
REST API support for IPv6 subnet creation in ODL	Yes	Yes, it is possible to create IPv6 subnets in ODL using Neutron REST API. For a network which has both IPv4 and IPv6 subnets, ODL mechanism driver will send the port information which includes IPv4/v6 addresses to ODL Neutron northbound API. When port information is queried it displays IPv4 and IPv6 addresses. However, in Beryllium release, ODL net-virt provider does not support IPv6 features (i.e., the actual functionality is missing and would be available only in the later releases of ODL).
IPv6 Router support in ODL 1. Communication between VMs on same compute node 2. Communication between VMs on different compute nodes (east-west) 3. External routing (north-south)	**No**	ODL net-virt provider in Beryllium release only supports IPv4 Router. In the meantime, if IPv6 Routing is necessary, we can use ODL for L2 connectivity and Neutron L3 agent for IPv4/v6 routing.
IPAM: Support for IPv6 Address assignment modes. 1. SLAAC 2. DHCPv6 Stateless 3. DHCPv6 Stateful	**No**	Although it is possible to create different types of IPv6 subnets in ODL, ODL_L3 would have to implement the IPv6 Router that can send out Router Advertisements based on the IPv6 addressing mode. Router Advertisement is also necessary for VMs to configure the default route.

Continued on next page

Table 2.1 – continued from previous page

Use Case / Requirement	Supported in ODL Beryllium	Notes
When using ODL for L2 forwarding/tunneling, it is compatible with IPv6.	Yes	
Full support for IPv6 matching (i.e., IPv6, ICMPv6, TCP, UDP) in security groups. Ability to control and manage all IPv6 security group capabilities via Neutron/Nova API (REST and CLI) as well as via Horizon.	**No**	Security Groups for IPv6 is a work in progress.
Shared Networks support	**No**	ODL currently assumes a single tenant to network mapping and does not support shared networks among tenants.
IPv6 external L2 VLAN directly attached to a VM.	**ToDo**	
ODL on an IPv6 only Infrastructure.	**ToDo**	Deploying OpenStack with ODL on an IPv6 only infrastructure where the API endpoints are all IPv6 addresses.