Part No. 060397-10 Rev. D
September 2014

OmniSwitch Networking Plug-in for OpenStack

The following covers the OpenStack Havana Release 3.0.

Alcatel·Lucent

www.alcatel-lucent.com

Alcatel·Lucent

Service & Support Contact Information

North America: 800-995-2696

Latin America: 877-919-9526

EMEA: +800 00200100 (Toll Free) or +1(650)385-2193

Asia Pacific: +65 6240 8484

Web: service.esd.alcatel-lucent.com

Email: esd.support@alcatel-lucent.com

Contents

1 OmniSwitch Plug-in for OpenStack Havana Installation

The OmniSwitch OpenStack Networking Plug-in (OONP) for OpenStack Havana offers infrastructure services for OpenStack logical networks by coordinating the orchestration of Alcatel-Lucent OmniSwitches as the underlying physical network. When used in conjunction with the OpenVSwitch plug-in, end-to-end multi-tenant network provisioning through OpenStack Networking (Quantum/ Neutron) is achieved.

The plug-in is intended to be installed in existing OpenStack Havana environments to configure the underlying physical network for its cloud networking operations.

The plug-in was developed and validated using the Ubuntu 12.04.04 LTS Linux operating system, based on the Canonical (Ubuntu) Cloud Archive release of OpenStack Havana. The configuration and plug-in file locations are based on this environment.

In This Chapter

The information described in this chapter include:

What's New in OONP 3.0

The following new features are being introduced in the OmniSwitch OpenStack Networking Plug-in (OONP) for OpenStack Havana 3.0 release. Please refer to "Deployment/Usage Guidelines and Known Issues" on page 2-17 for important guidelines prior to configuring these features.

Support for VLAN Name Based on Tenant ID

A new configuration parameter called `switch_vlan_name_tenantid` is being introduced to have TenantId as the VLAN name that gets created on the OmniSwitch as a result of Tenant network creation. It can be enabled or disabled in the ***omniswitch_network_plugin.ini*** file as below.

```
switch_vlan_name_tenantid = Enable | Disable
```

Note: If both `switch_vlan_name_prefix` and `switch_vlan_name_tenantid` are configured, `switch_vlan_name_tenantid` configuration will take precedence.

QoS Policy Mapping to Tenant Network

This feature allows OpenStack administrators to associate a predefined QoS policy on an OmniSwitch to a tenant network. In this release (OONP_H_R03), the QoS policy lists must be created outside of the OpenStack plug-in and already be applied on the OmniSwitch. This can be done through either the switch management interface such as SNMP (OmniVista NMS), CLI or Webview. The QoS policy management on the OmniSwitch is not done by OpenStack nor the plug-in.

See "Qos Policy Mapping to Tenant Networks" on page 1-6 for configuration information.

What’s New in OONP 2.0

The following new features are being introduced in the OmniSwitch OpenStack Networking Plug-in (OONP) for OpenStack Havana 2.0 release.

OmniSwitch 6860/6860E Support

Support for the OS6860(E) platforms has been added. The plug-in functionalities are the same as the OS6900 and OS10K platforms except the following:

The OS6860(E) does not support the VLAN_TAG based UNP classification method. If the plug-in is configured to use VLAN_TAG based host classification with an OS6860(E) in the topology, the plug-in will display the following ALERT message and use MAC_ADDRESS based UNP classification on 6860(E) platforms. Other devices in the topology will still use VLAN_TAG classification.

```
ALERT: OS6860 10.255.222.34 does not support VLAN-TAG based host classification.
Instead it will use MAC based host classification
```

Device specific switch_access_method

The OONP 1.0 release switch_access_method config parameter is global, meaning it uses the same configured access mechanism (whether TELNET or REST) for all the devices in the topology. OONP 2.0 is specific to each device allowing a different access method to be used per individual device in the topology. If an AOS 6.X device is mistakenly configured to use REST interface, an ALERT message will be logged and TELNET will be used automatically on those devices.

Note. This parameter is added as an additional entry in the existing device configuration. Please refer to the *.ini* file for the proper format.

VPA based host classification method

In addition to supporting UNP profiles on the edge switches for the interfaces connected to the Compute nodes, static 802.1q tagged plain Vlan-Port-Association (VPA) method is also supported. This is useful mainly for supporting an OS6450 as an edge switch.

Note. A new option "VPA" is introduced to "host_classification" parameter. Please refer the *.ini* file.

Switch save config interval

The range of time interval to save the switch configuration periodically is defined to be 600 - 1800 seconds. If any value is configured out of this range, an ALERT message is logged and the minimum value (600 seconds) is used automatically.

OmniSwitch OpenStack Networking Plug-in Architecture

OpenStack Networking provides an extensible architecture that supports a variety of plug-ins for configuring physical networks; however, choosing a network plug-in restricts configuration to only that of the plug-in's target technology. The OmniSwitch OpenStack Networking Plug-in architecture provides an environment where multiple independent plug-ins can be used to configure different network devices from different vendors. Each plug-in uses its own internal mechanism to communicate with their respective network elements. From the OpenStack Neutron-server perspective a single interface is provided through which the required networking services are provided to the OpenStack cloud applications. This allows OpenStack Networking to configure the physical network as well as the virtual switch instances running on the hypervisors. In addition, the OpenStack Networking L3-agent and DHCP-agent are fully supported.

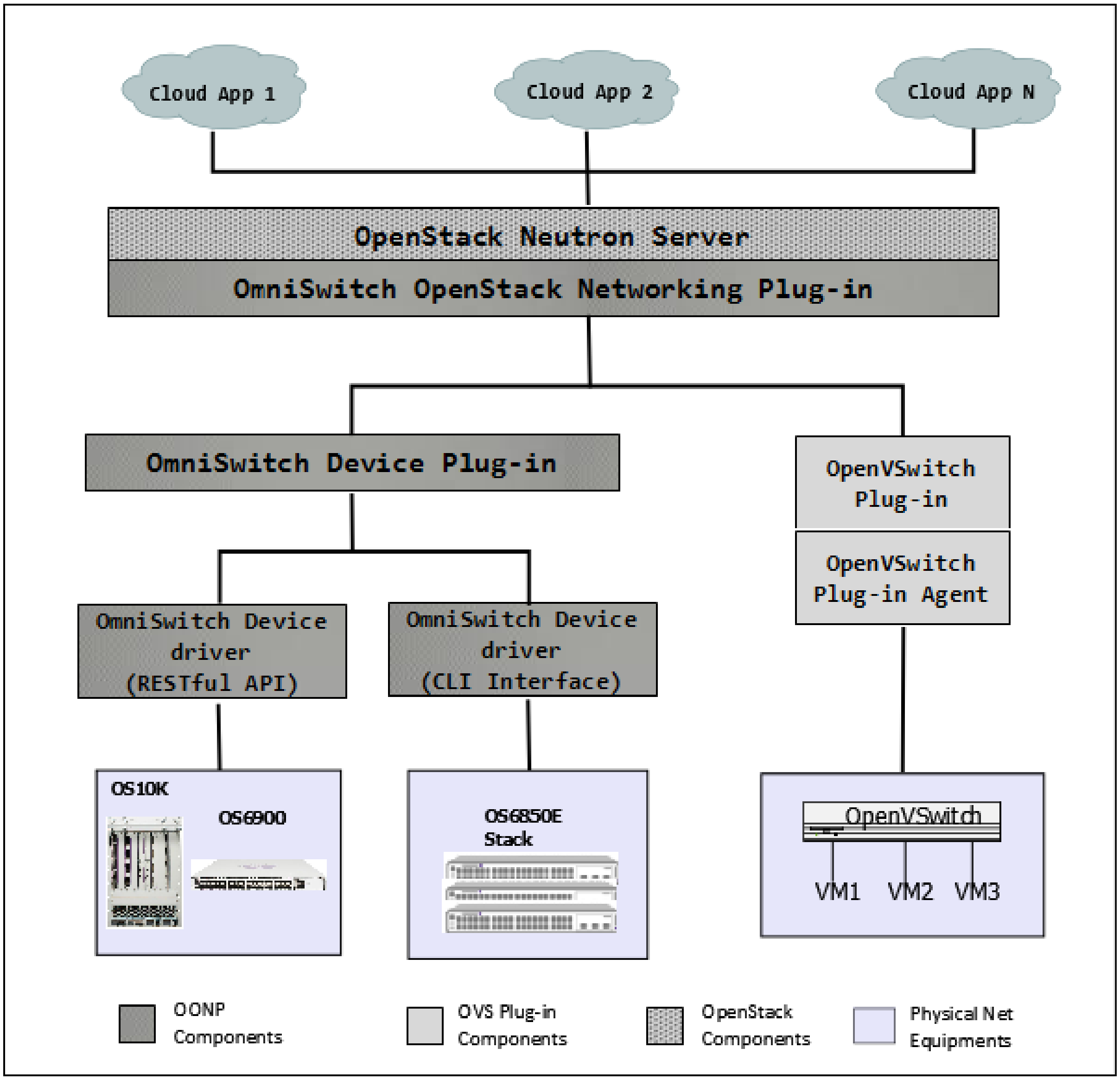
Release OONP_H_R03_7 of the OONP supports the following plug-ins:

- OVSNeutronPluginV2 (for OpenVSwitch)
- OmniSwitchDevicePluginV2 (for OmniSwitch physical hardware)

While the OONP supports multiple plug-ins, only one configuration file is allowed. The OVSNeutronPluginV2 configuration data must be appended to the OmniSwitch Neutron Plug-in configuration file.

The OmniSwitch Neutron plug-in supports the following hardware platforms with its respective AOS software releases.

- OS6900 and OS10K with AOS 732-R01 SW release and above
- OS685X and OS9000 with AOS 645-R02 SW release and above
- OS6250 and OS6450 with AOS 664-R01 and above, limited to the features supported on this platform
- OS6860 and OS6860 with AOS 811-R01.



In this deployment scenario, the OmniSwitchDevicePluginV2 plug-in uses configuration elements from the OpenVSwitch database to ensure configuration consistency and consistent VLAN ID assignment between it and the OVSNeutronPluginV2 plug-in.

OpenStack Networking API support

OpenStack applications using the northbound Neutron 2.0 APIs can manage logical network resources on the underlying physical network without knowing about the networking devices in the physical network.

The plug-in supports all of the standard Neutron 2.0 APIs (listed below).

Network	Subnet	Port
List Network	List Subnet	List Port
Show Network	Show Port	Show Port
Create Network	Create Subnet	Create Port
Update Network	Update Subnet	Update Port
Delete Network	Delete Subnet	Delete Port

Table 1: Neutron 2.0 APIs

VLAN Based Tenant Networks

The plug-in supports the VLAN network type. This means that logical networks (also referred to as “tenant networks”) are realized using VLANs in the OmniSwitch; thus, any operation related to tenant networks performed in OpenStack Networking is translated and configured in the OmniSwitch using VLANs. The VLAN id is obtained from the reserved VLAN range defined in the plug-in configuration file.

VLAN ID assignment is efficiently and intelligently used on switch ports by provisioning and deprovisioning VLANs across switches as virtual machines connected to tenant networks are created and destroyed.

Moreover, connectivity from the compute hosts to the physical network is trunked to allow traffic only from the VLANs configured on the host by the virtual switch.

The compute host VLAN trunk configuration on the OmniSwitch is automated through the use of the Alcatel-Lucent VNP mechanism. VNP classification is performed on the basis of either: the MAC address of the virtual machine; or the VLAN tag. While both methods are supported, only one can be used within an OpenStack installation instance. Network node connectivity (DHCP and L3 services) are automatically managed using VLAN-Port-Assignment.

Qos Policy Mapping to Tenant Networks

This feature allows OpenStack administrators to associate a predefined OmniSwitch QoS policy to a tenant network. There are three different levels of QoS policy-lists that are supported. The OpenStack administrator can choose one of them to associate to the tenant network created from the OpenStack GUI/ CLI. The levels are named Gold, Silver and Bronze with different levels of quality of service and network access. There is no separate GUI/CLI in OpenStack network creation to map these policy lists. Instead, it will be specified as part of the tenant network name string prefix as listed below:

- “GLD_” - Identifies the Gold policy-list will be applied.

- “SIL_” - Identifies the Silver policy-list will be applied.
- “BRN_” - Identifies the Bronze policy-list will be applied.

Mapping of the QoS policies is done in the in the ***omniswitch_network_plugin.ini*** file. The policy list names must already be configured on the OmniSwitch.

- qos_gold_policy_list = <policy-list for Gold level>
- qos_silver_policy_list = <policy-list for Silver level>
- qos_bronze_policy_list = <policy-list for Bronze level>

When a tenant network is created in OpenStack any of the defined prefixes can be used as part of the network name string to indicate which policy-list is to be applied on the OmniSwitch.

Enabling and mapping the policy-name

QoS policy mapping can be enabled and disabled using a configurable parameter called `qos_policy_mapping` in the ***omniswitch_network_plugin.ini*** file as follows.

```
qos_policy_mapping = Enable | Disable
```

An example of the associated QoS mapping parameters in the configuration file is:

- qos_gold_policy_list = corporate_tenants_qos
- qos_silver_policy_list = premium_tenants_qos
- qos_bronze_policy_list = regular_tenants_qos

In this example, the `corporate_tenants_qos` is the QoS policy list name that is predefined on the OmniSwitch that would be applied for Gold tenant networks. When a tenant network is created in OpenStack the prefix can be used as part of the network name string to indicate which policy-list to be applied. For example, if the network name is "GLD_corporate_external", the policy-list associated with Gold level would be applied. In the above case, that would be `corporate_tenants_qos.`

OmniSwitch AOS Configuration

The mapping of a QoS policy-list to a tenant network is achieved by mapping the policy-list to the UNP profile. In the OpenStack plug-in a UNP profile is created and assigned for each tenant network created from OpenStack. For policy-list mapping, in addition to assigning a UNP to a VLAN, the corresponding QoS policy-list name will also be mapped.

The following provides an example AOS configuration for predefining the QoS policy-list:

1 Create one or more QoS policy rules.

```
-> policy rule r1 condition c1 action a1
-> policy rule r2 condition c2 action a2
-> policy rule r3 condition c3 action a3
```

2 Create a QoS policy list.

```
-> policy list corporate_tenants_qos type unp enable
```

3 Assign one or more QoS policy rules to the policy list.

```
-> policy list corporate_tenants_qos rules r1 r2 r3
```

The above steps are done outside of OpenStack. Now that the policy-list is created on the OmniSwitch, once the OpenStack QoS policy mapping is configured, the following is automatically configured on the OmniSwitch through the OpenStack plug-in.

```
-> unp name corporate_user vlan 500 qos-policy-list corporate_tenants_qos
```

Modifying Mapping from OpenStack

QoS policy-list mapping is managed as part of the following network operations:

- **create tenant network -** Corresponding QoS policy-list will be applied to the corresponding tenant UNP profile on the OmniSwitch, if the feature is enabled.
- **delete tenant network -** The tenant UNP profile itself will get deleted along with the associated QoS policy-list, if one was applied.
- **update network -** If the QoS mapping is changed as part of the update network (via a name change), the old policy-list mapping will be removed and new mapping will be applied to the corresponding tenant UNP profile.

OmniSwitch OpenStack Networking Plug-in (OONP) Installation Overview

OmniSwitch OpenStack Networking Plug-in (OONP) provides OpenStack Networking (Neutron) with the ability to manage the Layer-2 configuration of OmniSwitch devices. The OONP plug-in supports the following features:

- 802.1q VLAN based tenant networks.
- Multiple physical topologies - ranging from a single switch to multi-switch based core-edge and spine-leaf topologies.
- Edge port (Host VM connections) automatic configuration based on VLAN Port Assignment (VPA).
- Edge port automatic configuration based on vNP classification using either: VM MAC address; or 802.1q VLAN tag.
- Automated inter-switch (Uplink Port) VLAN configuration using either static VPA or MVRP.

The following product matrix shows which features are supported/used for physical OmniSwitch configuration:

Feature Mode/Product	**OS10K/ OS6900**	**OS6860/6860E**	**OS6850E/ OS6850**	**OS6450**
Switch Access Method	ReST/Telnet	ReST/Telnet	Telnet	Telnet
Edge Port Configuration	VPA/MAC/ VLAN	VPA/MAC	VPA/MAC/ VLAN	VPA/MAC/ VLAN
Uplink Port Configuration	VPA/MVRP	VPA/MVRP	VPA/MVRP	VPA

Table 2: OmniSwitch Configurations Supported

While a mix of switch features is supported by the plug-in, only one Port Configuration method can be used in the OpenStack configuration. For example if you choose to use MVRP to configure Uplink Ports, all switches MUST support the MVRP feature. This is the case for BOTH the Edge Port Configuration and Uplink Port Configuration parameters.

OmniSwitch OpenStack Networking Plug-in Installation

The plug-in is delivered as a *tar.gz* file that contains the python modules, supporting applications, and a default omniswitch_network_plugin.ini file.

The tar file includes two applications that are used during the installation and configuration of the plug-in:

1 The ***omniplugin_install*** shell script will install the actual plug-in modules into the correct Neutron directories; additionally, it will create an initial configuration file that must be modified to reflect the plug-in configuration.

2 The ***omniswitch_setup*** shell script must be used to apply the one-time configuration actions to the OmniSwitch equipment, after the plug-in configuration has been modified with the correct settings.

The plug-in is installed on the OpenStack controller node and can be performed by either:

1 Using the ***omniplugin_install*** installation script, or

2 By manually extracting the contents of the tar.gz file into the neutron installation plug-in directory (typically ***/usr/lib/python2.7/dist-packages/neutron/plugins***)

Note. It is a pre-requisite to have an Operational OpenStack system and root access on Havana controller node.

The ***omniplugin_install*** script will install the plug-in files into the default OpenStack neutron distribution directory, ***/usr/lib/python2.7/dist-packages/neutron/plugins.*** If required, the installation script can be given an optional neutron installation directory that overrides the default of ***/usr/lib/python2.7/dist-packages/neutron.*** The installation script must be located in the same directory as the OONP release package. The package will be copied into the OmniSwitch plug-in directory and then extracted, leaving the original distribution package un-changed. A sample configuration file will be copied into the neutron configuration file directory, ***/etc/neutron/plugins/omniswitch.***

Note. Any existing ***omniswitch_network_plugin.ini*** file will be overwritten.

If you manually extract and install the plug-in, you need to copy the ***omniswitch_network_plugin.ini*** configuration file from the plug-in directory into the neutron configuration file location. For the Ubuntu Cloud archive distribution this will be ***/etc/neutron/plugins/omniswitch***.

Neutron must be configured to use the OONP at this point. The typical installation will use a combination of the OONP along with the OVS plug-in. The plug-in configuration is defined in two (2) files in the Ubuntu environment on the controller node:

1 /etc/default/neutron-server **-** the configuration file is specified using the following entry:

```
NEUTRON_PLUGIN_CONFIG= /etc/neutron/plugins/omniswitch/omniswitch_network_
plugin.ini
```

2 /etc/neutron/neutron.conf **-** the plug-in configuration is specified using the following entries:

```
core_plugin = neutron.plugins.omniswitch.omniswitch_network_plugin.OmniSwitch
NetworkPluginV2
```

OmniSwitch OpenStack Networking Plug-in Configuration

The plug-in requires configuration data about the plug-in operational details, details of the devices, the physical network topology, and the mechanisms to be used to provide the end-to-end connectivity across core and edge network switches.

When the plug-in is installed, the configuration file will have “default” values in it. These values must modified by the user before using the plug-in.

Any changes made in the configuration file during run-time require the Neutron-server to be restarted for the changes to take effect.

Following are the various parameters in the configuration file, its purpose and its default values.

```
omni_plugin = neutron.plugins.omniswitch.omniswitch_device_plugin.OmniSwitchDevice-
PluginV2
```

If you plan to use OVS plug-in the network along with OmniSwitch plug-in, please provide OVS plug-in name below. Otherwise leave it empty.

```
ovs_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
```

This is used by OmniSwitch plug-in for accessing Quantum database connection.

```
mysql://neutron:neutron@192.168.222.25:3306/quantum
reconnect_interval = 2
```

Below is an example ***omniswitch_network_plugin.ini*** file. Additionally refer to the “OmniSwitch OONP Installation and Multi-Switch Edge-Core Topology Example” on page 2-2 chapter for example topologies and their associated configuration parameters.

```
-------------------------------------------------------------------------------
[PLUGIN]
#Defines the sub-plug-ins used.

omni_plugin = neutron.plugins.omniswitch.omniswitch_device_plugin.OmniSwitchDevice-
PluginV2

ovs_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2

# Database access
#Defines the Neutron database location for use by the OmniSwitch plug-in
sql_connection = mysql://neutron:neutron@192.168.222.25:3306/quantum
reconnect_interval = 2

network_vlan_ranges = physnet1:2000:2009

[DEVICE]

# This is used to define the edge switches and which ports the compute/network
# nodes are attached to. The entry may contain 1 or more device definitions.
# The definition is:
# <switch-ip>:<switch-type>:<user-name>:<password>:<command-prompt>:<access_method>
# <node-interaces>:<core-interfaces>, with blank entries for user-name, password,
# or command-prompt specifying default values. A blank value for access_method will
# result in using the global switch_access_method

omni_edge_devices = 192.168.222.33:OS6900:  :  :  :  :1/19,
                    192.168.222.35:OS6900:  :  :  :  :1/16:1/20
```

```
# Used to define 0, or more, core switch configurations; the entry follows the same
# format as that of the omni_edge_device.

omni_core_devices = 192.168.222.34:OS6900:  :  :  :  :1/19 1/20

# This is used to specify which switch and port the DHCP server (network node) is
# connected.

dhcp_server_interface = 192.168.222.33:  :  :  :  :1/18

# HOST_CLASSIFICATION:
# The kind of classification rules to be applied for isolating tenant traffic from
# individual hosts or a subnet; possible values are:
#      MAC_ADDRESS - classify the host based on its mac address.
#      VLAN_TAG - classify the host based on the tenant network id (vlan).
#      (Note: VLAN_TAG is not supported in OS6860 devices;
#       if specified and OS6860 is in the topology, MAC_ADDRESS
#       will be used instead.)
#      VPA - classify the host based on the vlan-port assignment.
#     (Note: VPA can be used for devices that don't support UNP profiles,
#      such as the OS6450.
#     Not applicable when qos_policy_mapping is Enable.)

host_classification = MAC_ADDRESS

# This is used to specify what VLAN trunk configuration mechanism is to be used in
# the core switch inter-connections. <MVRP|VPA>

core_network_config = MVRP

# The default global method to access devices if not overridden by the switch_ac-
cess_method
# parameter of the omni_edge_devices, omni_core_devices, and/or dhcp_server_inter-
face.
# <TELNET|REST>

switch_access_method = TELNET

# SWITCH_VLAN_NAME_TENANTID:
# Use the tenant ID as the vlan-name. Default is Disable. Possible values are:
#               Enable - Use the tenant ID as the vlan-name.
#               Disable - Do not use the tenant ID as the vlan-name.
switch_vlan_name_tenantid = Disable


# SWITCH_VLAN_NAME_PREFIX:
# The user-defined custom string used as a prefix for the vlan-name; applicable
# only when switch_vlan_name_tenantid is Disable and ignored otherwise.
# 32 character Max
switch_vlan_name_prefix = Havana

# SWITCH_SAVE_CONFIG_INTERVAL:
# This is used to specify how often (in seconds) the config changes in the switches
# are to be saved.
# The valid range is 600 - 1800 seconds. If value is out of range, it will default
# to 600 seconds.
switch_save_config_interval = 600
```

```
# QOS_POLICY_MAPPING:
# This is used to associate an existing applied QoS policy in the OmniSwitches to a
# tenant network.
# There are 3 different levels of QOS policy lists supported: Gold, Silver and
# Bronze with distinct levels of quality of service and network access. Which
# policy list to be applied in the OmniSwitches is indicated by a non-configurable
# prefix to the tenant network name upon creation:
#               "GLD_" identifies a Gold QoS policy list
#               "SIL_" identifies a Silver QoS policy list
#               "BRN_" identifies a Bronze QoS policy list
# Default is Disable. Possible values are:
#               Enable - Associate QoS policy to tenant network as identified by
# corresponding prefix in its' name.
#               Disable - Do not associate QoS policy to tenant network, ignore
# values of qos_*_policy_list.
# (Note: not applicable when host_classification is VPA; qos_*_policy_list cannot
# exceed 32 characters.)
# For example,
# qos_policy_mapping = Enable
# qos_gold_policy_list = gold_tenants
# qos_silver_policy_list = silver_tenants
# qos_bronze_policy_list = bronze_tenants
qos_policy_mapping = Disable
qos_gold_policy_list =
qos_silver_policy_list =
qos_bronze_policy_list =
```

OpenStack Networking (Neutron) Configuration

The OpenStack Neutron plug-in provider must be modified to use the OmniSwitchNetworkPluginV2 (defined in the configuration file /etc/neutron/neutron.conf). Use the following definition:

core_plugin = neutron.plugins.omniswitch.omniswitch_network_plugin.OmniSwitchNetworkPluginV2

Please refer OVS plug-in documentation for more details and other parameters that may be used by the OVS plug-in and OVS agent.

```
---------------------------------------------------------------------------
[DATABASE]
sql_connection = mysql://neutron:neutron@192.168.222.25:3306/quantum
reconnect_interval = 2

[OVS]
bridge_mappings = physnet1:br-eth1
network_vlan_ranges = physnet1:2000:2009
integration_bridge = br-int
tenant_network_type = vlan

[AGENT]
polling_interval = 2

[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewall-
Driver
root_helper = sudo /usr/bin/quantum-rootwrap /etc/quantum/rootwrap.conf
-----------------------------------------------------------------------------
```

Prerequisites and Recommendations

The OONP is designed to be used with-in an existing OpenStack Havana environment. It is installed on the node hosting the controller functionality.

Each switch to be managed by the plug-in must have a separate management interface defined, preferably directly attached to the OpenStack management network. If the EMP interface on the OS6900/OS10K is used for management access, it must be configured on the same IP sub-net as the OpenStack management network infrastructure.

The following network infrastructure pre-configuration (or one time configuration) must be performed:

- Enabling VNP on the interfaces connected to the Compute node
- Enabling MVRP on the uplink interfaces, if the MVRP option is used
- The one-time switch configuration can be done manually or by invocation of the **omniswitch_setup script** found in */usr/lib/python2.7/dist-packages/neutron/plugins/omniswitch*. See "OmniSwitch OpenStack Networking Plug-in Installation" on page 1-9

2 Installation and Topology Examples

This chapter presents example OpenStack topologies with the associated OmniSwitch OpenStack Networking Plug-in (OONP) configurations. The configurations presented are based on two (2) variants of topologies:

1. A Single OS6900 switch
2. Multiple OS6900 switches.

Network configuration and management is achieved by using a combination of the OONP to configure the physical switches and OpenVSwitch (OVS) to configure the compute and network nodes.

The OONP supports tenant network isolation using VLANs; therefore, the OpenStack instance must generally be configured to use VLANs. These configurations do not use the flat-dhcp or GRE tunnel based tenant topologies.

The OONP supports multiple OmniSwitch product families as well as a variety of configuration and management methods. Generally, only one classification method and one uplink management method is available for a given OpenStack instance. For example, if VLAN classification is selected as the edge port configuration method it will be used for all edge switch configurations. Likewise if MVRP is chosen as the uplink port configuration method; MVRP will be used for all switches. This results in an OONP configuration that utilizes the common configuration options amongst all of the switches within the physical topology.

The configurations presented in this document are based on the OONP_H_R03_7 release for OpenStack Havana.

In This Chapter

The example installation and topologies described in this chapter include:

OmniSwitch OONP Installation and Multi-Switch Edge-Core Topology Example

This topology utilizes three (3) OmniSwitch 6900s in a core-edge configuration. Due to the homogeneous use of the OS9600 switch in this topology, the advanced configuration features of MVRP and vNP can be utilized.

The physical network is composed of two (2) 'edge switches', which provide end-station connectivity; and one (1) 'core switch'. It should be noted that 'edge-switches' have OpenStack nodes (and possibly other computing devices) connected to them; while 'core-switches' are connected only to other switches; either 'cores' or 'edges'.

The compute node VLAN trunk configuration on the OmniSwitch is automated through the use of the Alcatel-Lucent vNP mechanism. The VM mac-address will be used to identify and create the correct 802.1q VLAN tag configuration on the edge port. (Called mac-address classification)

The Network node VLAN trunk configuration is managed by the OmniSwitch plug-in using VLAN-Port-Assignment (VPA). (This is required because vNP classification occurs on ingress and the dhcp server, located on the network node, must have static VLAN connectivity to receive the tenant VM dhcp DISCOVER broadcast message)

VLAN uplink connectivity between the edge switches and the core switch is learned and configured automatically using MVRP.

This example provides steps on installing and configuring OONP. The physical configuration is composed of: two (2) compute nodes; One (1) network node; and a separate controller node. There are three (3) physical networks (refer to “Figure 1: Multi-Switch Edge-Core Topology” on page 2-4):

- The 'public' network, which has connections to both the controller and network nodes
- The internal management network, which has connections to all devices and nodes
- The 'private' tenant network, which has connections to the compute and network nodes

The following switch configuration assumptions are used in the example:

- The management/admin network is 10.1.2.0/24
- The EMP is used as the management interface
- The tenant VLAN range is 1005-1015
- The OmniSwitch device plug-in will use the factory default credentials to login to the switches (admin:switch)
- The configuration assumes the factory default prompt on all of the switches (->)
- The OmniSwitch device plug-in will use the telnet driver to communicate with the switches.

Refer to the Multi-Switch Topology OONP configuration file example for the following parameters:

- The general OmniSwitch plug-in configuration elements selecting: MAC_ADDRESS qualification; MVRP configuration; and TELNET switch communication methods are shown green.
- The VLAN range configuration, mapping the usage of VLANs 1005-1015 is shown in red.

The switch connection topology in “Figure 1: Multi-Switch Edge-Core Topology” on page 2-4 is mapped directly into the ***omni_edge_device***, ***omni_core_device***, and ***dhcp_server_interface*** configuration parameters, shown in **blue** in the example OONP configuration file. Each parameter may have multiple switch definitions separated by commas ','. The configuration attributes for the switch definition are as follows:

```
<Switch-IP>:<Switch-Type>:<User>:<Password>:<Prompt>:
<Switch-access-method>:<Node-interfaces>:<Core-interfaces>
```

In this example the factory defaults are used for authentication credentials and the default CLI prompt. Additionally the plugin *global switch_access_method* will be used; producing an entry of the following format::

n.n.n.n:OS6900: : : : :X:Y, where;

n.n.n.n is the switch management IP address,

X is the compute or network node endpoint connection port (in slot/port format),

Y is the inter-switch (uplink) connection port.

Note. The device specifications may include multiple interface ports for both the endpoint and inter-switch interfaces separated by spaces. Additionally, the network node is specified separately in the ***dhcp_-server_interface*** parameter and should not be included in the ****_device*** configuration parameters.

The corresponding OVS agent configuration file for this example is shown in configuration example**.** It should be noted that the ***tenant_network_type*** and ***network_vlan_ranges*** parameters duplicate the definitions in the OONP configuration file found on the controller node.

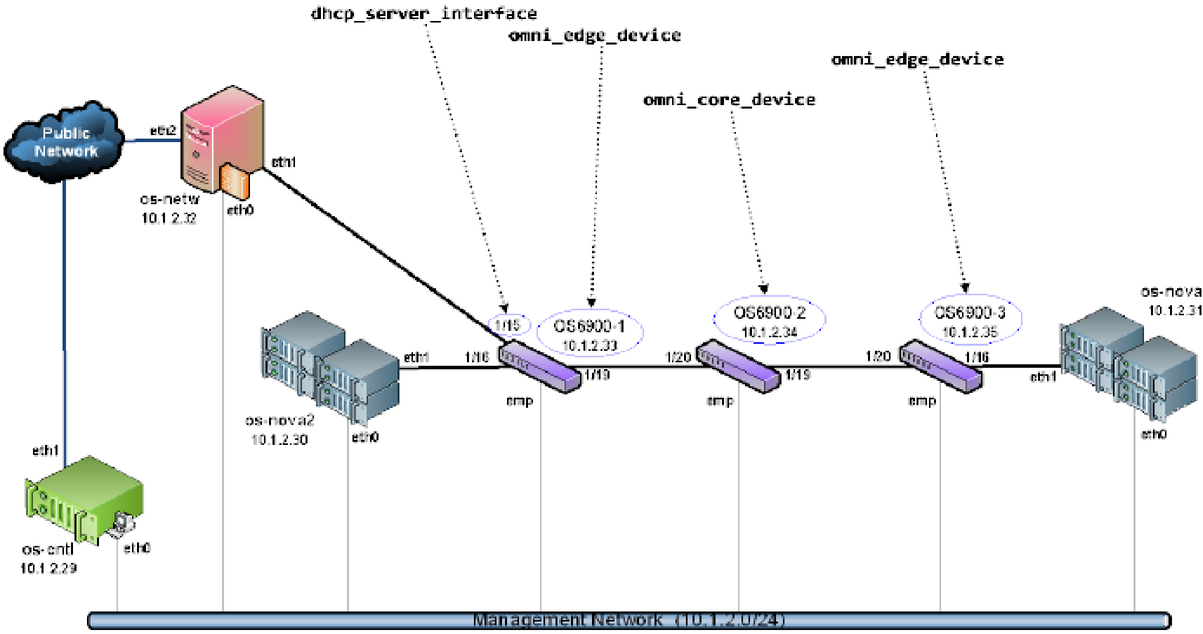


Figure 1: Multi-Switch Edge-Core Topology

```
[PLUGIN]
omni_plugin = neutron.plugins.omniswitch.omniswitch_device_plugin.OmniSwitchDevice-
PluginV2
ovs_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
sql_connection = mysql://quantum:quantum@10.1.2.29:3306/quantum
reconnect_interval = 2
network_vlan_ranges = physnet1:1005:1015

[DEVICE]
omni_edge_devices = 10.1.2.33:OS6900: : : :1/16:1/19,
                    10.1.2.35:OS6900: : : :1/16:1/20
omni_core_devices = 10.1.2.34:OS6900: : : :1/19 1/20
dhcp_server_interface = 10.1.2.33:OS6900: : : :1/15
host_classification = MAC_ADDRESS
core_network_config = MVRP
switch_access_method = TELNET
switch_vlan_name_prefix = Havana
switch_save_config_interval = 600

[DATABASE]
sql_connection = mysql://quantum:quantum@10.1.2.29:3306/quantum
reconnect_interval = 2

[OVS]
bridge_mappings = physnet1:br-eth1
integration_bridge = br-int
```

```
tenant_network_type = vlan
network_vlan_ranges = physnet1:1005:1015

[AGENT]
polling_interval = 2

[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
```

Installing and Configuring the OONP

To install and configure the OONP plug-in do the following.

1 Download the OONP_H_R03_7.tar.gz plug-in file and the ***omniplugin_install*** shell script from Alcatel Lucent service & support website at https://service.esd.alcatel-lucent.com/. Ensure that you download the latest version of the plug-in.

2 Copy both the plug-in package and the installation script into **/tmp** directory.

3 As root run ***./omniplugin_install*** script from /**tmp** directory. When the script is executed, the output is shown as follows:

```
root@os2-cntl:/tmp# ./omniplugin_install OONP_H_R03_7.tar.gz

*********** OmniSwitch Network Plug-in Installation Begin... ***************

Installing plug-in & config file ...
Extracting plug-in package <OONP_H_R03_7> into </usr/lib/python2.7/dist-packages/neutron/plugins/omniswitch> ...
LICENSE
NOTICE
__init__.py
consumer.py
official_build_num
omnidb_setup
omniswitch_clear_db
omniswitch_constants.py
omniswitch_db_v2.py
omniswitch_device_plugin.py
omniswitch_network_plugin.ini
omniswitch_network_plugin.py
omniswitch_neutron_dbutils.py
omniswitch_restful_driver.py
omniswitch_setup
omniswitch_telnet_driver.py
omniswitch_topology_utils.py
Setting file ownerships and permissions...
```

```
Installing plug-in config file ...
  Saving existing plugin configuration file <omniswitch_network_plugin.ini>...

*********** OmniSwitch Network Plug-in Installation End. ***************

root@os2-cntl:/tmp#
```

4 Edit the Neutron-server's configuration file to use the OmniSwitch Network plug-in. Edit "***/etc/neutron/neutron.conf***" and update the '***core_plugin***' parameter as follows.

```
core_plugin = neutron.plugins.omniswitch.omniswitch_network_plugin.OmniSwitchNetworkPluginV2
```

5 Edit "***/etc/default/neutron-server***" to say the plug-in configuration file to Neutron-server. Update the file with:

```
NEUTRON_PLUGIN_CONFIG= /etc/neutron/plugins/omniswitch/omniswitch_network_plugin.ini
```

6 Edit ***omniswitch_network_plugin.ini*** file to match Havana topology port configuration.

7 Restart neutron server on controller (as root run "service neutron-server restart").

One Time Configuration of OmniSwitch for OONP

Basic OmniSwitch configuration must be performed prior to using the plug-in. This configuration is performed by the omniswitch_setup script and requires that the correct operational configuration, including the physical topology, is reflected in the omniswitch_network_plugin.ini file.

As root run ./omniswitch_setup script from /tmp. When the script is installed, the output is shown as follows:

```
root@os2-cntl:/usr/lib/python2.7/dist-packages/neutron/plugins/omniswitch# ./omniswitch_setup
Warning: Plug-in config file is not specified!
Warning: Using default config file </etc/neutron/plugins/omniswitch/omniswitch_network_plugin.ini>
Config file </etc/neutron/plugins/omniswitch/omniswitch_network_plugin.ini> parsed successfully!
Applying One-time config on the OmniSwitches!. Please wait...
MVRP enabled on 10.1.2.33
MVRP enabled on 1/19 of 10.1.2.33
MVRP enabled on 10.1.2.34
MVRP enabled on 1 /19 of 10.1.2.34
MVRP enabled on 1 /20 of 10.1.2.34
MVRP enabled on 10.1.2.35
MVRP enabled on 1/20 of 10.1.2.35
UNP enabled on 1/16 of 10.1.2.33
UNP enabled on 1/16 of 10.1.2.35
One-time config on the OmniSwitches completed!.
root@os2-cntl:/usr/lib/python2.7/dist-packages/neutron/plugins/omniswitch#
```

Verify OONP Configuration Using AOS CLI

To verify OONP configuration, use the AOS CLI on each of the switch. The output will show as follows.

edge1 (.33)

```
OpenStack_6900_edge1-> show configuration snapshot da-unp
! DA-UNP:
unp port 1/16
unp port 1/16 classification enable
OpenStack_6900_edge1->

OpenStack_6900_edge1-> show configuration snapshot mvrp
! MVRP:
mvrp port 1/19 enable
mvrp enable
OpenStack_6900_edge1->
```

egde2 (.35):

```
OpenStack_6900_edge1-> show configuration snapshot da-unp
! DA-UNP:
unp port 1/16
unp port 1/16 classification enable
OpenStack_6900_edge1->

OpenStack_6900_edge2-> show configuration snapshot mvrp
! MVRP:
mvrp port 1/20 enable
mvrp enable
OpenStack_6900_edge2->
```

core1 (.34):

```
OpenStack_6900_core-> show configuration snapshot mvrp
! MVRP:
mvrp port 1/19 enable
mvrp port 1/20 enable
mvrp enable
OpenStack_6900_core->
```

Multi-Switch All Edge Topology Example

This topology consists of three (3) OmniSwitch OS6900s in a mesh interconnect. Each of the switches has at least one (1) OpenStack node connected to it (either compute or network). This configuration does not contain a core switch so the configuration file will not have the omni_core_devices configuration element.

The compute node VLAN trunk configuration on the OmniSwitch is automated through the use of the Alcatel-Lucent vNP mechanism. The incoming 802.1q tag will be used to identify and create the correct 802.1q VLAN tag configuration on the edge port. (Called vlan classification)

As in the previous topology, the Network node VLAN trunk configuration is managed by the OmniSwitch plug-in using VPA.

The inter-switch link 802.1q configuration is also managed using VPA.

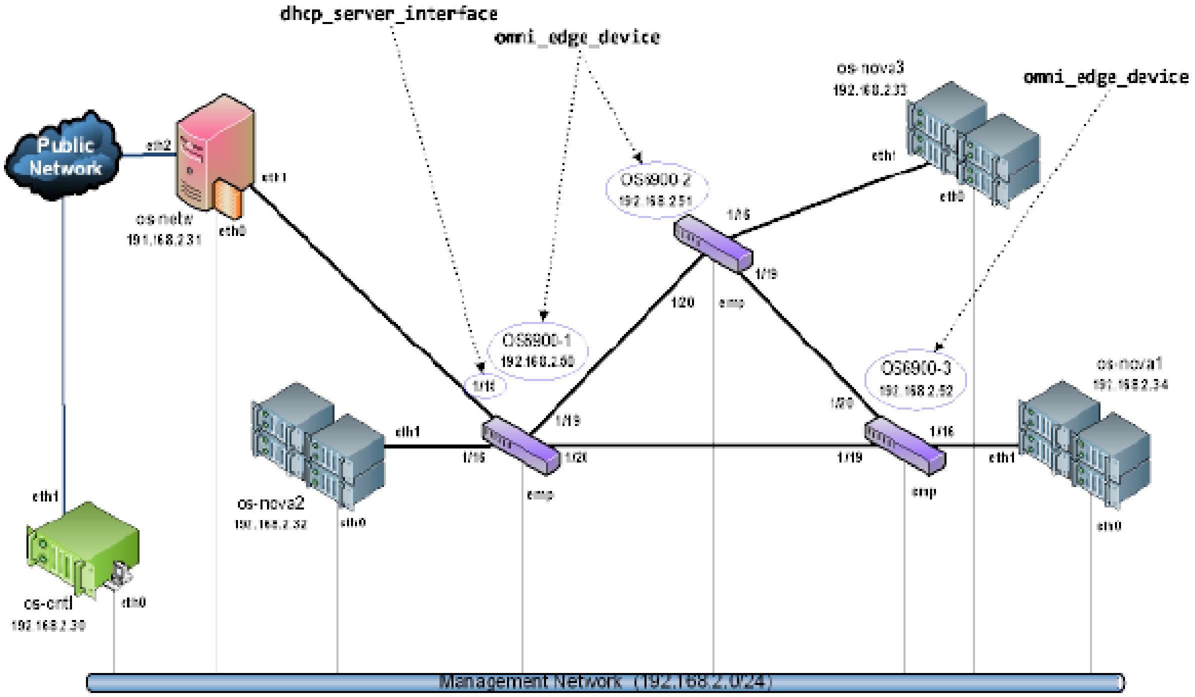


Figure 2: Multi-Switch All Edge Topology

Again, we are starting with a functioning OpenStack Havana configuration (with a functioning Neutron OVS networking infrastructure). The instance is composed of: three (3) compute nodes; network node, and the controller. The instance is composed of three (3) networks:

The inter-switch link 802.1q configuration is also managed using VPA.

- The ‘public’ network, which has connections to both the controller and network nodes
- The internal management network, which has connections to all devices and nodes
- The ‘private’ tenant network, which has connections to the compute and network nodes

The following switch configuration assumptions are used in the example:

- The management/admin network is 192.168.2.0/24

- The EMP is used as the management interface
- The tenant VLAN range is 1005-1015
- The OmniSwitch device plug-in will use the username of OpenStack and the password of secret to login to the switches
- The configuration assumes the factory default prompt on all of the switches (->)
- The OmniSwitch device plug-in will use the telnet driver to communicate with all of the switches (as specified by the plugin global switch_access_method. However, the REST method will be used to control the switch at 192.168.2.51.

Refer to the Multi-Switch Topology OONP configuration file example for the following parameters:

- The general OmniSwitch plug-in configuration elements selecting: VLAN qualification; VPA configuration; and TELNET switch communication methods are shown **green.**
- The VLAN range configuration, mapping the usage of VLANs 1005-1015 is shown in **red.**

The switch connection topology in "Figure 2: Multi-Switch All Edge Topology" on page 2-8 is mapped directly into the ***omni_edge_device*** and ***dhcp_server_interface*** configuration parameters, shown in blue in the example OONP configuration file. Note that ***omni_edge_device*** element is absent from the configuration. The configuration attributes for the switch definitions are as follows:

In this example the authentication credentials of ***openstack:secret*** and the default CLI prompt is used; producing an entry of the following format:

n.n.n.n:OS6900:openstack:secret: : :X:Y , where;

n.n.n.n is the switch management IP address,

X is the compute or network node endpoint connection port (in slot/port format),

Y is the inter-switch (uplink) connection port.

```
[PLUGIN]
omni_plugin = neutron.plugins.omniswitch.omniswitch_device_plugin.OmniSwitchDevicePluginV2
ovs_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
sql_connection = mysql://quantum:quantum@10.1.2.29:3306/quantum
reconnect_interval = 2
network_vlan_ranges = physnet1:1005:1015

[DEVICE]
omni_edge_devices = 192.168.2.50:OS6900:openstack:secret: : :1/16:1/19 1/20,
                    192.168.2.51:OS6900:openstack:secret: : :REST:1/16:1/19 1/20,
                    192.168.2.52:OS6900:openstack:secret: : :1/16:1/19 1/20
dhcp_server_interface = 192.168.2.50:OS6900:openstack:secret: : :1/15

host_classification = VLAN
core_network_config = VPA
switch_access_method = TELNET
switch_vlan_name_prefix = Havana
switch_save_config_interval = 600
```

```
[DATABASE]
sql_connection = mysql://quantum:quantum@192.168.2.29:3306/quantum
reconnect_interval = 2
[OVS]
bridge_mappings = physnet1:br-eth1
integration_bridge = br-int
tenant_network_type = vlan
network_vlan_ranges = physnet1:1005:1015
[AGENT]
polling_interval = 2
[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewall-
Driver
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
```

Single-Switch Topology Example

This configuration utilizes a single OmniSwitch OS6900 to provide the OpenStack tenant network inter-connectivity. The configuration is limited to a single ‘edge’ switch.

The compute node VLAN trunk configuration on the OmniSwitch is automated through the use of the Alcatel-Lucent vNP mechanism. The VM mac-address will be used to identify and create the correct 802.1q VLAN tag configuration on the host port.

The Network node VLAN trunk configuration is managed by the OmniSwitch plug-in using VLAN-Port-Assignment. (This is required because vNP classification occurs on ingress and the dhcp server located on the network node must have VLAN connectivity to receive the tenant VM dhcp DISCOVER broadcast message)

Again, we are starting with a functioning OpenStack Havana configuration (with a functioning Neutron OVS networking infrastructure). The instance is composed of: two (2) compute nodes; and a combined controller/network node. The instance is composed of three (3) networks.

- The ‘public’ network, which has a connection to the controller/network node;
- The internal management network, which has connections to the OmniSwitch devices and nodes;
- The ‘private’ data network, with connections to the compute and controller/network nodes.

The following switch configuration assumptions are used in the example:

- The management/admin network is 192.168.1.0/24
- The EMP is used as the management interface
- The tenant VLAN range is 1005-1015
- The OmniSwitch device plug-in will use the factory default credentials to login to the switch (admin:switch)
- The configuration assumes the factory default prompt on the switch
- The OmniSwitch device plug-in will use the telnet driver to communicate with the switch

Refer to the Single-Switch Topology OONP configuration file example for the following parameters:

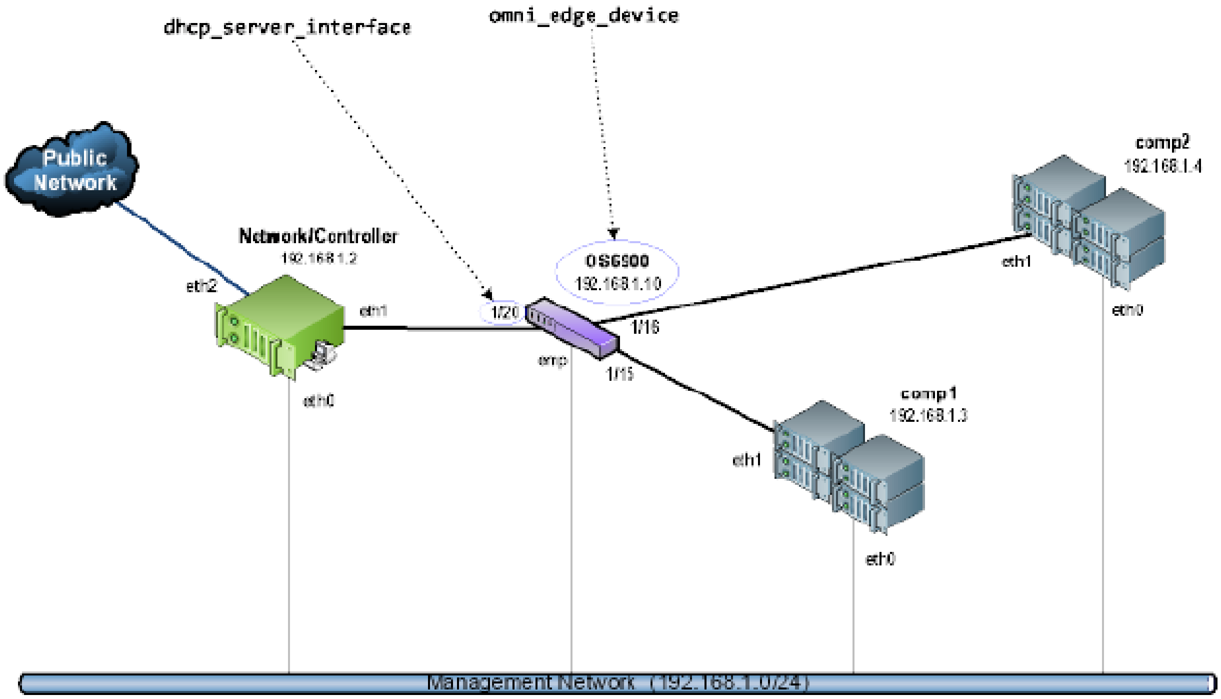


Figure 3: Single Switch Topology

- The general OmniSwitch plug-in configuration elements selecting: MAC_ADDRESS qualification and TELNET switch communication methods are shown **green.**
- The VLAN range configuration, mapping the usage of VLANs 1005-1015 is shown in **red.**

The switch connection topology in “Figure 3: Single Switch Topology” on page 2-11 is mapped directly into the ***omni_edge_devices*** and ***dhcp_server_interface*** configuration parameters, shown in **blue** in the example OONP configuration file.

In this example the factory defaults are used for authentication credentials and the default CLI prompt; additionally, the plugin global switch_access_method will be used, However; this example does NOT make use of core-switch connections, producing an entry of the following format:

n.n.n.n:OS6900: : : : :x: , where;

n.n.n.n is the switch management IP address,

x is the compute or network node endpoint connection port (in slot/port format),

Note the inter-switch (uplink) connection port is left blank.

Note. The device specifications includes multiple interface ports for both the endpoint interfaces separated by spaces ' '. Additionally, the network node is specified separately in the **dhcp_server_interface** parameter and should not be included in the ***_device** configuration parameters.

The corresponding OVS agent configuration file is shown in this example. It should be noted that the ***tenant_network_type*** **and** ***network_vlan_ranges*** parameters mirror the definitions in the OONP configuration file found on the controller node. Also pay special attention to the lack of the ***omni_core_devices*** **and** ***core_network_config*** configuration elements**.**

```
[PLUGIN]
omni_plugin = neutron.plugins.omniswitch.omniswitch_device_plugin.OmniSwitchDevicePluginV2
ovs_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
sql_connection = mysql://quantum:quantum@10.1.2.29:3306/quantum
reconnect_interval = 2
network_vlan_ranges = physnet1:1005:1015

[DEVICE]
omni_edge_devices = 192.168.1.10:OS6900: : : :1/15 1/16:
dhcp_server_interface = 192.168.1.10:OS6900: : : :1/20
host_classification = MAC_ADDRESS
switch_access_method = TELNET
switch_vlan_name_prefix = Havana
switch_save_config_interval = 600

[DATABASE]
sql_connection = mysql://quantum:quantum@10.1.2.29:3306/quantum
reconnect_interval = 2

[OVS]
bridge_mappings = physnet1:br-eth1
integration_bridge = br-int
tenant_network_type = vlan
network_vlan_ranges = physnet1:1005:1015

[AGENT]
polling_interval = 2

[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
```

Virtual-Chassis with LACP Topology

This configuration utilizes a combination of Link Aggregation and Virtual Chassis features to provide network redundancy.The example is implemented using a pair of OmniSwitch OS6900s. The topology outlined is limited to a single 'edge' switch for simplicity; however, this is the most complex configuration presented.

Prior to use the switches must be configured as a virtual chassis.

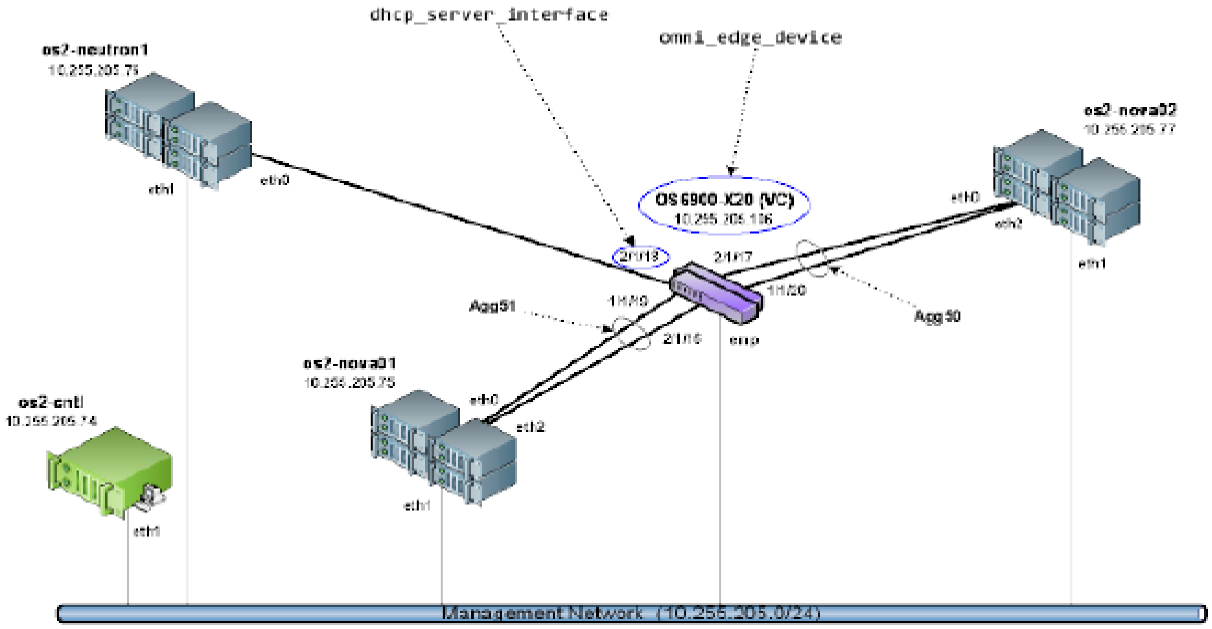


Figure 4: Single-Switch Topology

Link Aggregation features must also be enabled and configured on the switch prior to use by the OpenStack instance. LACP is used to manage the network traffic on the aggregated links. The following commands will achieve the Link Aggregation configuration:

```
linkagg lacp agg 50 size 2 admin-state enable
linkagg lacp agg 50 name "OpenStack LACP testing nova02"
linkagg lacp agg 50 actor admin-key 50
linkagg lacp agg 51 size 2 admin-state enable
linkagg lacp agg 51 name "OpenStack LACP testing nova01"
linkagg lacp agg 51 actor admin-key 51
linkagg lacp agg 52 size 2 admin-state enable
linkagg lacp port 1/1/19 actor admin-key 51
linkagg lacp port 1/1/20 actor admin-key 50
linkagg lacp port 2/1/16 actor admin-key 51
linkagg lacp port 2/1/17 actor admin-key 50
unp linkagg 50
unp linkagg 50 classification enable
unp linkagg 51
unp linkagg 51 classification enable
```

The compute hosts require specialized configurations to support LACP also. While OVS directly supports link aggregation (and LACP), there have been issues with 802.1x VLAN tagging when combined with LACP. Therefore, the linux bonding driver is used to provide LACP support; OVS is configured to use the bond interface. The interface bonding is described by the following **/etc/network/interfaces** configuration elements:

```
auto eth0
allow-bond0 eth0
iface eth2 inet manual
        bond-master bond0
auto eth2
allow-bond0 eth2
iface eth2 inet manual
        bond-master bond0
auto bond0
iface bond0 inet manual
        bond-mode 802.3ad
        bond-slaves none
        bond-miimon 100
        bond-lacp-rate fast
```

The OpenVSwitch configuration required to utilize the bond0 interface defined is:

```
ovs-vsctl show
dbb9cf7f-3c38-42b2-831f-4ebf940d7c43
    Bridge "br-bond0"
        Port "bond0"
            Interface "bond0"
        Port "phy-br-bond0"
            Interface "phy-br-bond0"
        Port "br-bond0"
            Interface "br-bond0"
                type: internal
    Bridge br-int
        Port br-int
            Interface br-int
                type: internal
        Port "int-br-bond0"
            Interface "int-br-bond0"
ovs_version: "1.10.2"
```

The OONP will manage both the compute node 802.1q VLAN configuration using the vNP and mac-address classification mechanisms as in prior configurations.

```
[PLUGIN]
omni_plugin = neutron.plugins.omniswitch.omniswitch_device_plugin.OmniSwitchDevicePluginV2
ovs_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
sql_connection = mysql://neutron:neutron@10.255.205.74:3306/quantum
reconnect_interval = 2
network_vlan_ranges = physnet1:1100:1110

[DEVICE]
omni_edge_devices = 10.255.205.106:OS6900: : : : :50 51:
dhcp_server_interface = 10.255.205.106:OS6900: : : : :2/1/10
host_classification = MAC_ADDRESS
core_network_config = MVRP
switch_access_method = TELNET
switch_vlan_name_prefix = Havana
switch_save_config_interval = 600
```

```
[DATABASE]
sql_connection = mysql://neutron:neutron@10.255.205.74:3306/quantum
reconnect_interval = 2

[OVS]
bridge_mappings = physnet1:br-eth1
integration_bridge = br-int
tenant_network_type = vlan
network_vlan_ranges = physnet1:1100:1110

[AGENT]
polling_interval = 2
[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewall-
Driver
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
```

In the previous examples, switch ports have been defined using the <slot/port> nomenclature; this changes with the use of link aggregation and Virtual Chassis. For connections that use aggregations, the linkagg NUMBER is used, while the non-aggregated links are specified using the <chassis/slot/port> format.

Because this is a mixed topology, with some connections using LACP and some not, the OVS configuration files will be different depending on the node configuration. The compute nodes will use the bonding interface instead of the physical NIC interface. This can be seen in highlighted sections of the OVS configurations for the network and compute nodes.

VC with LACP Compute Node OVS Configuration File

```
[DATABASE]
sql_connection = mysql://neutron:neutron@10.255.205.74:3306/quantum
reconnect_interval = 2

[OVS]
bridge_mappings = physnet1:br-bond0
integration_bridge = br-int
tenant_network_type = vlan
network_vlan_ranges = physnet1:1100:1110

[AGENT]
polling_interval = 2

[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewall-
Driver
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
```

VC with LACP Network Node OVS Configuration File

```
[DATABASE]
sql_connection = mysql://neutron:neutron@10.255.205.74:3306/quantum
reconnect_interval = 2

[OVS]
bridge_mappings = physnet1:br-eth1
integration_bridge = br-int
tenant_network_type = vlan
network_vlan_ranges = physnet1:1100:1110

[AGENT]
polling_interval = 2

[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
```

Deployment/Usage Guidelines and Known Issues

- For large deployments it is recommended to use VNP for access ports and MVRP for the core and aggregation.
- The OONP writes log output to the neutron-server log file, which by default is ***/var/log/neutron/neutron-server.log***. Output includes plug-in initialization messages and switch configuration commands.
- The version of the OVS and the associated plug-in (OVSNeutronPluginV2) used along with OONP is 1.10.0. It is expected that the later version of the OVS plug-in will work with OONP as long as there is no change in the database schema used by OVS plug-in.
- If the admin username or password are incorrect in the OmniSwitch *plugin.ini* file and networks are created using Horizon, the network gets created on Horizon OpenStack web GUI but OONP fails to create the VLANs on the OmniSwitch. The workaround is to use the CLI on the controller to list the networks and then delete the unwanted networks. The Neutron **net-delete** command may need to be run several times to get the network removed and/or the OmniSwitch plug-in may need to be restarted.
- There is no error message in */var/log/server.log* if the variable **switch_save_config_interval** is set to an unsupported value. There is only an information message stating that the switch_save_config_interval is out of the valid range and that it will of valid range(%d - %d)... default to 600" . The plug-in will ignore the incorrect value and set it to the default value of 600.
- The Neutron server log shows OmniSwitch plug-in deletion of UNP MAC rule failure when deleting a network; however, network is deleted correctly. This issue can be confirmed by looking for the same UNP MAC rule that failed in the immediate previous log successful entries. The OmniSwitch plug-in duplicates deletion of a UNP MAC rule. The first time the rule is successfully deleted and the second time it fails since the rule no longer exists. This happens for both RESTful and TELNET drivers. The second set of failed log entries can be ignored.
- If configuring the QoS policy mapping feature in R03, OpenFlow cannot be enabled on the OmniSwitch. OpenFlow was introduced as an early availability feature in AOS Release 7.3.3.R01.
- Traffic between VMs on the same compute node is not subject to the switch QoS policies. This may result in some tenant traffic being un-affected by the switch QoS policies.