HP VAN SDN Controller Release Notes

Abstract

This document contains supplemental information for HP VAN SDN Controller Release 2.3.

HP Part Number: 5998-6078
Published: August 2014
Edition: 2

The HP VAN SDN Controller license text is in /opt/sdn/legal/EULA.txt. The HP VAN SDN Controller incorporates materials from several Open Source software projects. Therefore, the use of these materials by the HP VAN SDN Controller is governed by different Open Source licenses. Refer to /opt/sdn/legal/HP-SDN-CONTROLLER-OPENSOURCE-LIST.pdf for a complete list of the materials used.

Acknowledgements

Warranty

For the software end user license agreement and the hardware limited warranty information for HP Networking products, visit http://www.hp.com/networking/support.

Open Source Software

For information on licenses for the open source software used by the HP VAN SDN Controller, see the *HP VAN SDN Controller Open Source and Third-Party Software License Agreements*.

For information on acquiring the open source code for the HP VAN SDN Controller, send an email to **HPN-Open-Source-Query@lists.hp.com**.

HP Security Policy and Release Notes

A Security Bulletin is the first published notification of security vulnerabilities and is the only communication vehicle for security vulnerabilities.

Fixes for security vulnerabilities are not documented in manuals, release notes, or other forms of product documentation.

A Security Bulletin is released once all vulnerable products still in support life have publicly available images that contain the fix for the security vulnerability.

To find any Security Bulletins for the HP VAN SDN Controller, visit the HP Networking manuals web page:

www.hp.com/networking/support

To initiate a subscription to receive further HP Security Bulletin alerts via email, go to:

http://h41183.www4.hp.com/signup_alerts.php?jumpid=hpsc_secbulletins

Contents

Description

For the latest version of these release notes and HP VAN SDN Controller 2.2 user guides, see the HP Networking support search site.

1. Open your browser and go to: www.hp.com/support/manuals.
2. Use the tools provided to search by product name (for example, VAN SDN Controller) or product number.

   Detailed information about the selected product displays, including a list of Support options in the left column.

Overview

The base SDN Controller Appliance serves as a delivery vehicle for SDN solutions. It aims to provide a platform for developing various flavors of network controllers, e.g. data center, public cloud, private cloud, campus edge networks, etc. This includes being an open platform for development of experimental and special-purpose network control protocols using a built-in OpenFlow controller.

The SDN Controller Appliance meets certain minimum scalability requirements and provides the ability to achieve higher scaling and high-availability requirements via a scale-out teaming model. In this model, the same set of policies are applied to a region of network infrastructure by a team of such appliances, which coordinate and divide their control responsibilities into separate partitions of the control domain for scaling, load-balancing and fail-over purposes.

The principal software stack of the appliance uses OSGi framework (Equinox) and a container (Virgo) as a basis for modular software deployment and to enforce service provider/consumer separation. The software running in the principal OSGi container may interact with other components running as other processes on the appliance. Preferably, such IPC interactions occur using a standard off-the shelf mechanism, e.g. RabbitMQ, but they may exploit any means of IPC best suited to the external component at hand.

Regardless of the specific personality of the controller, the software stack consists of two major tiers. The upper Administrator tier hosts functionality related to policy deployment, management, personae interactions and external application interactions, i.e. slow-path, deliberating operations. The lower Controller tier, on the other hand, hosts policy enforcement, sensing, device interactions, flow interactions, i.e. fast-path, reflex, muscle-memory like operations. The interface(s) between the two tiers provide a design firewall and are elastic in that they can change along with the personality of the overall controller appliance. Also, they are governed by a rule that no enforcement-related synchronous interaction crosses from the Controller to Administrator tier.

The Administration tier of the controller appliance hosts a web-layer through which software modules installed on the appliance can expose REST API to other external entities. Similarly, modules can extend the available web-based GUI to allow network administrators and other personae to directly interact with the features of the software running on the SDN Controller Appliance.

New in release 2.3

Features

- As long as the keystone server is configured to use UUID token_format, current versions of OpenStack Keystone (specifically the Icehouse version) are supported.
- The Keystone server can be local or remote.
- Keystone configuration (default user/roles/tenants) is no longer taken care of by the installer. The configuration of Keystone is now decoupled from the controller.
- Using environment variables, the location of the Keystone server can be provided at installation. 1-way SSL can be used and set up in the same fashion.
    - The environment variables required to use a custom Keystone server (without variables, localhost is assumed and checked to be running Keystone):
        - AUTH_ENDPOINT - In the format http(s)://<ip address>:<port>/v2.0.
        - AUTH_TOKEN - Not required. Assumed to be ADMIN (Keystone package default).
        - TRUSTSTORE - Required if using https in endpoint.
        - TRUSTSTORE_PASS - Required if using https.
- The Keystone configuration can be modified after install if there is a need to change (endpoint, truststore, etc). The controller must be restarted for this to take effect.
- When installing a new system, the user will not be able to log in to the controller until Keystone users/roles/tenants are configured. These do not have to be sdn/skyline (which they probably won't be in a deployed environment)

REST enhancements

- Performance
- Full duplex
- Asynchronous resources

AppStore Features

- Link to Launch AppStore from Applications view
- Link to Login to AppStore allows you to view a list of purchased applications

- AppShelf supports install or upgrade of applications
- Support for proxy configuration
    - /etc/init/sdnc.conf

```
env JAVA_OPTS="-Xms512m -Xmx4096m -XX:MaxPermSize=512m-Dhttps.proxyHost=web-proxy.rose.hp.com
-Dhttps.proxyPort=8088-Dhttp.nonProxyHosts=127.0.0.1|localhost|15.255.121.172|15.255.126.13|15.255.127.5|15.255.123.6-DHPWS_DEV=true"
```

- http.nonProxyHosts value must be provided for the team member controllers and the team's north-bound IP.
- Support for single switch from production to development AppStore:
    - /etc/init/sdnc.conf

```
env JAVA_OPTS="-Xms512m -Xmx4096m-XX:MaxPermSize=512m
-Dhttps.proxyHost=web-proxy.rose.hp.com-Dhttps.proxyPort=8088-Dhttp.nonProxyHosts=127.0.01.
|localhost|15.2551.21.72|15.2551.261.3|15.2551.27.5|15.2551.23.6-DHPWS_DEV=true"
```

- This toggles the browser/client URLs as well as the server-side URLs.
- Three new demo applications submitted to development AppStore portal.
    - "<id>_v<version>.zip" naming convention workaround (i.e. com.hp.sdn.test.alertgen_v1.0.0.zip) is used.
    - Two are signed; one unsigned.
        - application.zip verification for submitted and downloaded applications
- More verbose error handling in UI
- The user interface added a **Refresh** button: the browser refresh forces users to re-authenticate with the HPWS portal.

Licensing features

- UI dialog added to allow copy-paste of uninstall key after a license has been deactivated

Application installer

- Capability between debian and zip-based apps extended
- Support signing of application .zip packages (in addition to the zip's internal Java artifacts)
    - By default the check for signed application zip files is disabled
    - To Turn on signed zip verification
        - UI console "Configurations" - under "AppManager"
            - Set "verifyZips" parameter to true
- The public certificate used to sign the zip file must be installed in "sdnjar_trust.jks" (in /opt/sdn/admin).
- UI enhancements added to the error handling dialogs.
- UI added **Refresh** button
- The use case of upload via REST API and then installing via the UI is not supported. The UI performs an upload and deploy atomically. If the upload is performed via the REST API, the install must be performed via the REST API. If the application is uploaded via the REST API, but not followed by an install, the UI will show the application as "Staged" and the only operation that can be performed from the UI is uninstall.

Supportability/Manageability

- JVM metrics are now being persisted over time using the metrics subsystem, which provide some built-in troubleshooting capabilities that were only accessible before when using supplemental tools (e.g. JConsole, VisualVM, profilers, etc). Some supplemental tools cannot be run on headless systems, some affect the operation of the system they're monitoring, and none are a part of the SDN Controller's basic system requirements. In contrast, these JVM metrics are always available for both pre-release troubleshooting and profiling and for post-release troubleshooting and analysis.
    - There are about forty–five metrics that encompass various measures of memory, NIO, threads, garbage collection, and pertinent operating system values (e.g. CPU and file descriptor use).
    - Each is persisted every minute, kept by default for a week, and can be retrieved using the /metrics REST API. Thus the metrics can be used to monitor the JVM's changing consumption of resources over time as it runs, and because the metric values are persisted they can be used for post-crash or post-hang investigations leading up to a failure. The metrics are also preserved after a controller restart, so even after a failed controller has been restarted some forensic analysis is still possible.
    - The last "snapshot" of metric values can be seen as part of the support report via the /support REST API for an on-demand view of resource consumption.
- The metrics can be used to guide troubleshooting investigations and analyses. They may also be used to "profile" various controller builds against one another to gauge performance changes between builds, or to gauge the impact of running a specific application or combination of applications on the controller's JVM.

Network services

Improvements have been added for both link manager and node manager.

- Introduced com.hp.sdn.link.LinkService API and implementation of SupplierService API.
- Introduced com.hp.sdn.node.NodeService API and implementation of SupplierService API.
- Link and node discovery out-of-the-box are handled by OpenFlow Link Discovery and OpenFlow Node Discovery apps.
- Support added for 3rd party node and link supplier applications via LinkSupplierService and NodeSupplierService API's.
- Removed com.hp.sdn.ctl.linkdisco.LinkService API.
- Removed com.hp.sdn.ctl.nodemgr.NodeService API.
- Slight adjustments to /net/links REST API: Removing link_state from response and changing link type from uppercase to lowercase.
- NodeService utilizes new teaming infrastructure.
- NodeManagerComponent configuration refactored (split) into OfArpDiscoveryComponent, OfDhcpDiscoveryComponent, and OfIpDiscoveryComponent. Any tests which were configuring "arp.age", "dhcp.age", or "ip.learn" will need to configure via these new components.
- LinkService utilizes new teaming infrastructure

Appliance

- OVA for field and partner use

Data Model

- Device holds onto interfaces.
  - Each InterfaceId is unique only in the context of a device.
- REST APIs exist to retrieve, create, and delete devices.
- REST APIs exist to retrieve interfaces.

DeviceService is implemented as a publicly accessible OSGi component along with DeviceId. This is fully implemented with the device objects being cached for the time being.

NOTE: Although the Device interface supports the connection for interfaces, no implementation yet exists to allow a caller to retrieve or create interfaces.

Core Controller

Java API changes

- ControllerService - added methods to register initial flow contributors.
- InitialFlowContributor - provides new interface for flow contributors.
- SequencedPacketListener - event() callback now returns void (not boolean).
- MessageContext - added isSent() predicate.
- MessageContext.PacketOut - added send() method.
- OfmMutableFlowMod - added methods to clear actions/instructions.

UI enhancements

- FlowClass meta data is now included in the Datapath/Flows view detail panes.

Datapath connect sequence

- The datapath initial connection mechanism has been expanded to use the Device Service to determine a type for the datapath, and to install initial flows contributed by other subsystems.
- The complete sequence looks like this:
  1. Datapath connects to the controller
  2. OpenFlow handshake performed (Hello, Hello, FeaturesRequest, FeaturesReply)
  3. Extended handshake performed (MP-request for Description, Ports, TableFeatures)
  4. Event emits `DATAPATH_CONNECTED`.
  5. Device Service determines device type.
  6. Flow tracker sends FlowMod/DELETE (all tables) command to datapath.
  7. Flow tracker generates "core" contributed flowmods (via device driver subsystem).
  8. Flow tracker collects initial flowmods from contributors (NodeManager, LinkManager).
  9. Initial flowmods validated (via flow-class subsystem).
  10. Initial flowmods adjusted (via device driver subsystem).
  11. Initial flowmods are sent to the datapath, along with a barrier request.
  12. On receipt of the barrier reply, emits `DATAPATH_READY`.

Device Drivers

- VLAN handler is in place.
  - Executed on the fly (not persisted).
  - Reads pre-configured VLANs on the device (cannot create or change via the device driver).
- Manual discovery is now used on all devices to get supplemental information via SNMP.
  - If snmp is not on, fields such as serial number, etc, are copied from the DataPathInfo object.
- H3C devices are now supported by the FlowMod facet.
- Generic Event dispatch mechanism is now in place.
  - Receives events and passes them through to the listeners; no adjustments or analysis is done on them.
- SNMP driver now uses credentials and finds its own, if possible, if none are given.
- All flow mod adjustment now goes through the device driver's flow mod facet.
  - Includes any flow created and sent to FlowTrk through sendConfirmedFlowMod.
  - Generation of default flows, when an OF instance connects, is done through the device drivers.
    - When an app gets installed, if it has default flows that switches should have upon connection, it can register itself as an InitialFlowContributor and send in flows that should be installed as default.
    - The SDN controller installs the basic flows (table misses, forward normal, or steal, based on hybrid setting).
    - NodeManager and LinkManager each contribute their ARP/DHCP and BDDP flows.
- Interface facet is complete.
  - Based on OF messages, interfaces can be added, deleted, and updated in association with a device.
- IpDiscovery reference implementation and basic supplier is implemented.

The device driver framework is in place and consists of xml files specifying known device types plus the specific implementation of each of the facets they support.

Currently defined device types include:

- 3800
- 3500
- 2920
- 5400
- 8200

These device types are readable through a REST API in the DeviceDriverDemoApp.

Currently accessible facets include:

- GenericDeviceIdentity
- HpDeviceIdentity
- DefaultOpenflow facet

- Openflow facet for chassis switches in v2 only mode
- Openflow facet for v1 supported mode

Current drivers:

- SNMPDriver is written to obtain sysoid and serial number, etc. from the switch for identification.
- Device types are "evolved" by gathering enough information so that a device can be mapped to a specific device type in an xml file.

NOTE: The SNMPDriver currently supports v2 only and does not have an interface definition. It has to be instantiated before it can be used.

NOTE: Device Keys can be specified through a REST API accessible through the RSDOC and DeviceDriverDemoApp, allowing gets, posts, and deletes.

NOTE: These keys are currently used as community names only (credentials).

The following interfaces have been defined to support future facet development: VLAN, VXLAN, interface, flowmod

OfDeviceDiscovery acts as the supplier of device information. It is an OSGi component which gets invoked when a data path event is received by an external module. With the data path information, it determines the device type and then stores the device info about that specific device with its device type. It uses the device identity facet to evolve to the best device type. The device type itself holds information about which facets and which implementations of those facets are supported.

OfDeviceDiscovery collects all the interfaces of the device that are part of the OF instances and associates them onto the device. Port events such as add, remove or modify are handled by OfDeviceDiscovery.

HA and teaming updates

The new HA framework “happy path” will be available for use to allow the platform services to be updated to the new APIs. The systems status field helps improve troubleshooting in teamed environments.

- The following modules containing public APIs have been removed:
  - hp-util-dbus
    - package: com.hp.dist.bus
  - hp-util-dkvs
    - package: com.hp.dist.keystore
  - hp-util-dlock
    - package: com.hp.dist.lock
  - hp-util-dsync
    - package: com.hp.dist.sync
- The following public services have been removed:
  - com.hp.sdn.ha.HAService
  - com.hp.sdn.team.TeamConfigurationService
  - com.hp.sdn.team.TemInformationService

- com.hp.sdn.team.TeamConfigBootstrapService
- com.hp.sdn.team.TeamingService

- The following modules containing public APIs have been added:
  - hp-util-dcord-api
    - package: com.hp.util.dcord
- The following public services have been added:
  - com.hp.util.dcord.CoordinationService: Replaces com.hp.sdn.ha.HAService and com.hp.sdn.team.TeamingService.
  - com.hp.sdn.teaming.TeamAdminService: Replaces com.hp.sdn.team.TeamConfigurationService, com.hp.sdn.team.TemInformationService, com.hp.sdn.team.TeamConfigBootstrapService and com.hp.sdn.team.TeamingService.

NOTE: As part of the transition, com.hp.sdn.teaming.TeamAdminService has been renamed to com.hp.sdn.team.TeamAdminService.

- The following REST interfaces for Teaming have been removed:
  - GET /team/status - Status for team/systems should be retrieved directly from /system interface
  - GET /team/version - Version is included in response for general GET /team request
    - com.hp.sdn.rs.TeamConfigResource has been removed and replaced with com.hp.sdn.rs.TeamAdminResource for this REST change

Backup/Restore

The new callback framework allows applications to register/unregister callbacks to perform their own backup/restore activities.

New Java API

BackupRestoreService and BackupRestoreListener

REST API changes

/backups has been deleted and all backup/restore related functionality has been removed from /systems. The new set of APIs is as follows:

- /backup – GET for downloading the backup file
- /backup/status – GET for retrieving the status of the backup
- /backup/checksum – GET for retrieving the checksum of the backup file currently on the controller
- /backup – POST to start the backup
- /restore/backup – POST to upload the backup on to the controller
- /restore/status – GET to retrieve the backup status
- /restore – POST to start the restore

Security notes

- Team communication currently does not support cryptographic authentication of team members. Firewall rules are automatically added when the team is created to allow team communication to occur only between team members.
- Team communication currently does not support cryptographic encryption between team members. To help protect this communication, the interfaces used for team communication should be on a trusted network or dedicated VLAN with appropriate Access Control protections.

VNI reservation service

A VNI reservation service has been added. The service is intended to provide a simple mechanism for marking VNIs that are used or intend to be used by applications so that applications do not attempt to use/configure a VNI that is already in use by another application. A lightweight implementation of the service is available, but is only for a single node, and does not include teaming, persistence, or REST API.

Data persistence and teaming for VNI reservations are available and a private/sideways REST API has been added for intra-team coordination.

Controller issues and suggested actions

Release 2.3

- **HP SDN App Store access:** Until the App Store becomes available, the following buttons in the controller Application display do not access App Store features:
  - **Log in to view applications…**
  - **Launch App Store**
- **Recovering from Partial Team Creation:** If the team is not successfully created in all controllers, it is not possible to fix the failed controllers without disbanding the team. To recover from this failure it is recommended to delete the team, fix the problem in the controllers where the create operation failed, and try again.
- Recovering from Partial Team Deletion: If the team is not successfully deleted in all controllers, the failed controllers might go to suspended mode because they might not have quorum. That is, they won't be able to connect to those controllers where the operation was a success. To recover from this failure it is recommended to delete the team on each failed controller so configuration files are removed and the controllers transition to standalone mode.
- **A team misconfiguration makes a controller inoperable (CR 152738):** If a misconfiguration causes the IP Connectivity to fail and the controllers are unable to communicate with each other once the NB_IP is programmed, the cluster becomes inoperable. NB_IP and the member IPs (used for cluster communication) are in the same subnet. Since IP routing is not possible between eth0 where NB_IP was programmed, the behavior is as expected.
- **Config Component LinkDiscovery requires an App disable/enable to take effect (CR 152842):**—com.hp.sdn.disco.of.link.impl.OpenflowLinkDiscoveryComponent changes only take place when you manually disable and enable through OpenFlow Link Discovery undo applications.

Release 2.2

Issues occurring when the controller is configured with hybrid.mode=false:

- **ARP, ICMP request, and other communication to a switch data-plane IP fails** (CR147704)—Currently, the controller does not support direct communication with controlled switches. The only supported communication is through the controlled switches and not to the controlled switches. When a host on the network sends an ARP request, the controller assumes

that the ARP request is intended for another host on the network (and not a controlled switch). The controller instructs the controlled switch to forward the packet elsewhere in the network and does not instruct the controlled switch to directly respond to the packet.

To resolve this:

1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
2. See the user documentation for information on how this change affects the controlled network.

- **Data plane traffic to or from a host indirectly connected to a controlled switch is not forwarded at line-rate** (CR148324)—A host is indirectly connected to a controlled switch when there is an uncontrolled switch between the edge-most controlled switch and the host. When a host is connected to the controlled network in this manner, the controller does not learn where the host is located because the controller assumes that no hosts will appear on infrastructure ports. Since the controller does not learn where the host is located, any traffic that flows to or from this host cannot be paved, and is therefore handled by the controller at each hop through the controlled network. If a single packet to or from such a host needs to cross a number of controlled switches, then the controller will be consulted those many times for the same packet. The actual throughput rate depends upon the load of other processing on the controller and the number of hops that such flows take through the controlled network.

  To resolve this:

  1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
  2. See the user documentation for information on how this change affects the controlled network.
  3. If step 1 cannot be implemented, then connect all hosts to ports of controlled switches that are not connected to other controlled switches.

     Do not connect multiple controlled switches to ports on the same VLAN of a router, especially a gateway router.

- **Packets are not properly forwarded through a controlled router** (CR148326)—The controller is not aware of the data plane MAC addressing or IP addressing of a controlled L3 router. The controller is also unaware of whether a packet received by the data plane of a controlled switch should be switched, routed, or consumed. Additionally, the controller is not aware of the data plane subnetting, static routes, or routing protocol information that provides information necessary to properly route traffic.

  To resolve this:

  1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
  2. See the user documentation for information on how this change affects the controlled network.
  3. If step 1 cannot be implemented, then change the configuration of the OpenFlow instance on the router so that it is no longer controlled by the controller.

- **Traffic between two hosts crossing a switch configured with a multi-VLAN and the aggregate OpenFlow instance in a controlled network is not forwarded at line-rate** (CR148389)—HP switches, including those from the 2920/3500/3800/5400/6200/6600/8200 series, support an aggregate OpenFlow instance. This instance contains all VLANs on the controlled switch. When the controller attempts to pave a path across such a switch configured with this aggregate instance, the controller does not send the VLAN ID to which the path paving flow-mod applies. Any switch configured with multiple VLANs in an aggregate instance will reject the flow-mod because the ingress VLAN was not specified. Since the flow-mod is never accepted by the switch, all future forwarding decisions for such packets are delegated to the controller by the controlled switch. The controller makes the forwarding decision for every packet which needs to cross a controlled switch with a multi-VLAN aggregate OpenFlow instance. If a single packet needs to cross a number of such switches, then the controller will

be consulted those many times for the same packet. The actual throughput rate depends upon the load of other processing on the controller and the number of hops that such flows take through the controlled network.

To resolve this:

1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
2. See the user documentation for information on how this change affects the controlled network.
3. If step 1 cannot be implemented, then change the switch configuration to use an OpenFlow instance per VLAN instead of a single aggregate instance for all VLANs.

- **Traffic between two hosts in a partially-controlled network is not forwarded at line-rate** (CR148385)—The controller is responsible for the forwarding decision of every packet that enters a switch it controls. When the controller observes a packet for any given flow, it attempts to pave the path through the network through the OpenFlow forwarding rules for that flow, so that all future packets of the same flow are handled by the switch according to the forwarding rule. In cases where two controlled switches are separated by a multi-hop link, the controller does not pave the path across that multi-hop link because it paves only a single path and the controller cannot be guaranteed that multiple paths do not exist if multiple multi-hop links exist.

  As a result, the controller will not pave any paths across a multi-hop link. The controller makes the forwarding decision for every packet which needs to cross a multi-hop link. If a single packet needs to cross a number of multi-hop links, then the controller will be consulted those many times for the same packet. The actual throughput rate depends on the load of other processing on the controller and the number of hops that such flows take through the controlled network.

  To resolve this:

  1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
  2. See the user documentation for information on how this change affects the controlled network.
  3. If step 1 cannot be implemented, then connect switches in such a manner that there are no uncontrolled switches (multi-hop links) between controlled switches in the controlled network.

- **All IPv6 traffic is dropped** (CR148658)—The HP VAN SDN Controller does not recognize the devices that use only the IPv6 addresses on the control plane. The controller and the devices with which it communicates must use the IPv4 addresses on the control plane. IPv6 traffic running in the data plane of an OpenFlow-hybrid network is supported when the controller is operating with **hybrid.mode** set to **true** (the default). Under these conditions, the data plane forwarding decision for IPv6 packets is made without involvement by the default controller applications. However, if **hybrid.mode** is set to **false**, all packets are sent to the controller and the default controller applications drop all the IPv6 packets.

  Similar to any protocol that is not supported by the default controller applications, if the IPv6 data plane traffic support is required, then write and install the application in the controller to provide switches with the desired flows to let the controller observe and direct the forwarding decision.

  To resolve this:

  1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
  2. See the user documentation for information on how this change affects the controlled network.

- **Host is unable to ping some other hosts on the network when multiple VLANs are used when the controller is configured with hybrid.mode=false (OpenFlow-only)** (CR148179)—When

mutiple VLANs are used with a controller that is configured for **hybrid.mode=false**, a host might not be able to ping some other hosts on the network.

To resolve this:

1. Change the **hybrid.mode** setting of the controller to **true** and restart the controller.
2. See the user documentation for information on how this change affects the controlled network.

Other issues in release 2.2

- **Controller loses knowledge of hosts on the network, but those hosts are actively communicating through or with the network infrastructure** (CR148453)—The default setting for the controller's ARP timeout is 5 minutes. The default setting of **ip arp-age** for ProVision switches is 20 minutes. If the controller setting does not match the setting on the controlled switches, then information learned by the controller from ARP traffic is aged out of the controller's knowledge base before it is aged out of the infrastructure. The controller needs to wait for up to 15 minutes to see ARP traffic again for a given host. If the controller is unaware of the host's location on the network, other problems may occur with traffic forwarded to or from that host depending upon the applications and forwarding decisions that the controller is responsible for.

  To resolve this:

  1. Change the controller's **arp.age** setting on the controller UI under **com.hp.sdn.ctl.nodemgr.impl.NodeManager** to be greater than or equal to the **ip arp-age** setting of the controlled switches.
  2. If step 1 cannot be implemented, then change the **ip arp-age** setting of the controlled switches to match the **arp.age** setting on the controller.

- **HP VAN SDN Controller becomes unresponsive when the HP Network Optimizer SDN Application session database has around 1M entries** (CR148578)—The HP VAN SDN Controller crashes in JVM and becomes unresponsive when the HP Network Optimizer SDN Application session database has about 1M entries in it. Read the *HP Network Optimizer Release Notes* to know more about the JVM memory size.
- **When trying to access OpenFlow Topology using Firefox browser, error 500 occurs** (CR147973)—When trying to access the OpenFlow Topology using the Firefox browser, the "Server Error - 500: Internal Server Error" error message appears. The topology appears after that error is closed.
- **HA teaming—After failover or failback all the links learned are not shared across all the team members all the time** (CR146171)—In an HA teaming configuration, after failover or failback, all the links learned are not shared across all the team members all the time. This behavior can occur when there is a very large number of links between switches (over12,000) and a large number of switches (over 500) in the controller domain.
- **Pin All option does not pin all data paths and nodes** (CR146165)—On the HP VAN SDN Controller UI, in Topology viewer, the **Pin All** option under **View** does not pin all data paths and nodes.
- **Cassandra fails to free disk space occupied by removed records** (CR146155)—The Cassandra database fails to free disk space occupied by removed records.
- **Topology viewer displays moving switch** (CR146636)—When connecting a single physical switch to the controller and bringing it up in the Topology viewer on the GUI, it occasionally

shows the switch moving around the screen before any end hosts have been discovered on the switch.

- **When performing the backup and restore operations, the restore operation is not logged** (CR148809)—When you perform the backup and restore operations and check the audit logs for these operations, the restore operation is not logged.
- **Schema disagreement exception for Cassandra observed in log files** (CR148457)—The schema disagreement exception for Cassandra occurs when a keyspace is created on all nodes of the controller. This happens when a schema configuration is in progress on one node of the cluster and the same schema is configured on another node of the same cluster. This issue is intermittent and has no impact on the functioning of the controller.
- **The OpenFlow Topology view of any single controller does not display the entire team-wide topology** (CR148644)—The OpenFlow Topology view shows the switches and the respective end-nodes that are connected to the controller. In a controller team environment, the entire team-wide topology is not shown in the OpenFlow Topology view of any single controller.
- **NIO direct buffers will not be garbage collected before running out of space** (CR148470)—There is a possibility that the NIO direct buffers will not be garbage collected before running out of space because of the way the JVM garbage collection is implemented with respect to NIO direct buffers. The NIO buffer garbage collection is triggered to run only when the normal Java heap garbage collection runs. If the normal Java heap remains steady and never invokes the garbage collection, the NIO directly allocated buffers will never be freed.

Release 2.1

- **Never released.**

Release 2.0

- **REST call to get all the ports takes 5 seconds when the number of ports is 40k** (CR141008)—The REST call to get all ports takes more time as the number of ports increases.
- **Auxiliary connections established by the device to the controller are not visible via REST or the UI** (CR140089)—Manage the device auxiliary connections to the controller from the switch by telneting to the switch.
- **When using Internet Explorer 9 or Internet Explorer 10, the controller console is blank** (CR138915)—Currently, IE 9 is not supported, and IE 10 has limited support. In IE 10, OpenFlow Topology is unavailable.
- **When restarting the sdncservice database, exceptions are reported in logs** (CR141589)—The following errors are expected only during the initialization phase, and don't describe any unexpected behavior:

  ```
  [2013-10-03 11:31:39.890] INFO t Resolve Thread (Bundle 81)
  System.outInternal Exception: org.postgresql.util.PSQLException:
  ERROR: relation "X" already exists...
  ```

  ```
  [2013-10-03 11:31:39.899] INFO t Resolve Thread (Bundle 81)
  System.out[EL Warning]: ServerSession(2136794997)--Exception
  [EclipseLink-4002] (Eclipse Persistence Services -
  2.4.2.v20130514-5956486):
  org.eclipse.persistence.exceptions.DatabaseException
  ```

- **Topology Map fails to display network-wide computed trees** (CR137780)—The topology viewer in HP VAN SDN Controller 2.0 topology UI shows only the devices discovered by the controller pointed to by the browser, not the entire topology discovered by a team of controllers.

- **OpenFlow topology GUI is not optimal when a large number of hosts or devices are connected** (CR140798)—The HP VAN SDN Controller 2.0 topology UI is not intended to represent large topologies consisting of hundreds of elements.
- **On team reboot, suppressed ports are lost** (CR137854)—Suppressed ports information (specifying the ports on which want to stop LLDP traffic) is not stored in persistence, and is lost whenever the controller reboots. HP recommends that you maintain a backup of your suppressed ports configuration.
- **Comware switch OpenFlow behavior** (CR138462)—When a flow is pushed to a extensibility table with `apply_actions` as the instruction type, and the retrieved Flow Statistics using Multipart Request `apply_actions` is correct, the CLI always shows `Write actions`.
- **Tagged link between two devices is not discovered** (CR138547)—If a link exists between a pair of ports tagged to two different VLAN instances in Aggregate mode in OpenFlow, links are discovered correctly in the HP VAN SDN Controller between only one of the VLAN instances. The link between the other instances is not discovered. This issue does not occur in Virtualized mode.
- **Link Discovery displays link across two OpenFlow instances** (CR139375)—The Link Discovery application in the HP VAN SDN Controller shows links between two OpenFlow instances when the same port is tagged to two different s associated with the two OpenFlow instances in Comware (5900) devices.
- **Group/meter create/update/delete require optional and redundant command attribute** (CR141930)—When using the REST API for creating/updating/deleting a group/meter, including the JSON request in the command field is required.
- **During installation through the Ubuntu software center, the HP VAN SDN Controller Debian package displays a "Package is of bad quality" error message** (CR141745)—HP VAN SDN Controller installation through the Ubuntu software center is not supported.
- **Output of REST API to fetch application with name parameter is inconsistent** (CR141736)—The HP VAN SDN Controller Application Manager REST API for fetching applications currently fetches all applications. There is no support for a filter based on query parameters.
- **Jconsole utility is not available in openjdk-7–jre-headless, which is a dependency before installing the HP VAN SDN Controller**—If you create metrics in the HP VAN SDN Controller, and then try to open the Jconsole to see the metrics, a message displays saying that Jconsole cannot be found.
- **HP VAN SDN Controller failed to delete clean start memento /tmp/HPN Van Controller.clean, error message in the log.log file** (CR142605) —This is a temporary file that is only created during a restore or upgrade during normal operation when the OSGi Virgo container is restarted with the —clean option.
- **Zookeeper warning logged during initialization** (CR138057)—The HP VAN SDN Controller 2.0 includes ZooKeeper connection logs. Failed connection attempts are normal during Controller initialization and configuration changes and can be ignored.
- **Delete failed error exception is noticed from teaming module in the log.log file** (CR142603)—The HP VAN SDN Controller 2.0 tries to cleanup internal data structures during Controller initialization and configuration changes and logs error message if data is not found. This is normal during Controller initialization and configuration changes and can be ignored.
- **Backup/Restore fails when manual upload/download of backup files when file owner changes to anyone other than sdn user** (CR138689)—Any manual operation on the VAN SDN Controller, other than using the REST APIs through curl can change the file attributes from sdn username to Operator username. The backup/restore fails if you perform a manual upload/download of files. To avoid this, always perform any manual operation via curl using the REST APIs.

- **Keystone-related SDN user password changes are not restored properly** (CR141586)—If the default sdn password is changed from **`skyline`** using keystone and the backup operation is done, you cannot login with the new password after restore operation. You must still use the default password (**`skyline`**) for login.
- **Installation guide calls for AMD64 processor but processors from other manufacturers can be used**—The hardware requirements listed in the *HP VAN SDN Controller Installation Guide* incorrectly specify that an AMD64 server or desktop machine is required. No specific processor manufacturer is required. You can use x86-64 processors from other manufacturers.
- **Flow is not added or retrieved correctly** (CR138494)—The `_VID: PRESENT` bit is not set for ID, which indicates an incorrectly formed `_VID` match field from a Match structure in a `MultipartReply / FlowStats.element`.
- **Unable to modify _vid in table 101** (CR140524)—A switch error occurs if you push Flow Mod with the `vlan_vid` value as part of `set-fields`.
- **Connected Links disappear between the OpenFlow switches when spanning tree is enabled** (CR140755)—When enabling spanning tree in OpenFlow switches, controller-sent LLDP packets are not being forwarded from `STP_BLOCKED` ports, causing the discovered links to be deleted and link rediscovery to not occur on the ports.
- **Openflow 1.3 badly formed MultipartReply/FLOW stats message** (CR142663) —java.lang.IllegalStateException: "Timed-out waiting for response" might appear when the REST API is invoked to list all flows on a datapath from the SDN Controller. This occurs due to the information sent from the switch firmware.
- **When RESTAPi /stats/ports/ is executed with dpid & portid filter, an exception is thrown with response code 500** (CR142114) —HTTP/1.1 500 Internal Server Error might appear when the REST API is invoked to get stats for a port (GET /stats/ports) in a datapath from the SDN Controller. This appears due to the incorrect information from the Openflow devices (Switch CR 131910:dpctl stats-port <port> command displays empty stats.)
- **The RESTAPi /datapaths/{dpid}/ports/{port_id}/action fails to change the state of the port** (CR140753)—The port state does not get changed when the REST API is invoked to change the state of a port in a datapath from the SDN Controller due to the design in the Provision switches.

Documentation feedback

HP is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback@hp.com). Include the document title and part number, version number, or the URL when submitting your feedback.