AOS-CX 10.06 Virtual Switching Extension (VSX) Guide

6400, 8320, 8325, 8360, 8400 Switch Series

aruba

a Hewlett Packard
Enterprise company

Part Number: 5200-7727
Published: November 2020
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Chapter 1 About this document

This document describes features of the AOS-CX network operating system. It is intended for administrators responsible for installing, configuring, and managing Aruba switches on a network.

Applicable products

This document applies to the following products:

- Aruba 6400 Switch Series (JL741A, R0X26A, R0X27A, R0X29A, R0X30A)
- Aruba 8320 Switch Series (JL479A, JL579A, JL581A)
- Aruba 8325 Switch Series (JL624A, JL625A, JL626A, JL627A)
- Aruba 8360 Switch Series (JL700A, JL701A, JL702A, JL703A, JL706A, JL707A, JL708A, JL709A, JL710A, JL711A)
- Aruba 8400 Switch Series (JL375A, JL376A)

Latest version available online

Updates to this document can occur after initial publication. For the latest versions of product documentation, see the links provided in **Support and other resources**.

Command syntax notation conventions

Convention	Usage
`example-text`	Identifies commands and their options and operands, code examples, filenames, pathnames, and output displayed in a command window. Items that appear like the example text in the previous column are to be entered exactly as shown and are required unless enclosed in brackets (`[ ]`).
`example-text`	In code and screen examples, indicates text entered by a user.
Any of the following: • *`<example-text>`* • `<example-text>` • *`example-text`* • *example-text*	Identifies a placeholder—such as a parameter or a variable—that you must substitute with an actual value in a command or in code: • For output formats where italic text cannot be displayed, variables are enclosed in angle brackets (`< >`). Substitute the text—including the enclosing angle brackets—with an actual value. • For output formats where italic text can be displayed, variables might or might not be enclosed in angle brackets. Substitute the text including the enclosing angle brackets, if any, with an actual value.

Table Continued

Convention	Usage
|	Vertical bar. A logical `OR` that separates multiple items from which you can choose only one. Any spaces that are on either side of the vertical bar are included for readability and are not a required part of the command syntax.
`{ }`	Braces. Indicates that at least one of the enclosed items is required.
`[ ]`	Brackets. Indicates that the enclosed item or items are optional.
… or	Ellipsis:
`. . .`	• In code and screen examples, a vertical or horizontal ellipsis indicates an omission of information. • In syntax using brackets and braces, an ellipsis indicates items that can be repeated. When an item followed by ellipses is enclosed in brackets, zero or more items can be specified.

About the examples

Examples in this document are representative and might not match your particular switch or environment.

The slot and port numbers in this document are for illustration only and might be unavailable on your switch.

Understanding the CLI prompts

When illustrating the prompts in the command line interface (CLI), this document uses the generic term `switch`, instead of the host name of the switch. For example:

```
switch>
```

The CLI prompt indicates the current command context. For example:

`switch>`

Indicates the operator command context.

`switch#`

Indicates the manager command context.

`switch(`*`CONTEXT-NAME`*`)#`

Indicates the configuration context for a feature. For example:

```
switch(config-if)#
```

Identifies the `interface` context.

Variable information in CLI prompts

In certain configuration contexts, the prompt may include variable information. For example, when in the VLAN configuration context, a VLAN number appears in the prompt:

```
switch(config-vlan-100)#
```

When referring to this context, this document uses the syntax:

```
switch(config-vlan-<VLAN-ID>)#
```

Where *`<VLAN-ID>`* is a variable representing the VLAN number.

Identifying switch ports and interfaces

Physical ports on the switch and their corresponding logical software interfaces are identified using the format:

member/slot/port

On the 6400 Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Specifies physical location of a module in the switch chassis.
  - Management modules are on the front of the switch in slots 1/1 and 1/2.
  - Line modules are on the front of the switch starting in slot 1/3.
- *port*: Physical number of a port on a line module.

For example, the logical interface `1/3/4` in software is associated with physical port 4 in slot 3 on member 1.

On the 83xx Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Line module number. Always 1.
- *port*: Physical number of a port on a line module

For example, the logical interface `1/1/4` in software is associated with physical port 4 in slot 1 on member 1.

NOTE: If using breakout cables, the port designation changes to x:y, where x is the physical port and y is the lane when split to 4 x 10G or 4 x 25G. For example, the logical interface 1/1/4:2 in software is associated with lane 2 on physical port 4 in slot 1 on member 1.

On the 8400 Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Specifies physical location of a module in the switch chassis.
  - Management modules are on the front of the switch in slots 1/5 and 1/6.
  - Line modules are on the front of the switch in slots 1/1 through 1/4, and 1/7 through 1/10.
- *port*: Physical number of a port on a line module

For example, the logical interface `1/1/4` in software is associated with physical port 4 in slot 1 on member 1.

Identifying modular switch components

- Power supplies are on the front of the switch behind the bezel above the management modules. Power supplies are labeled in software in the format: *member/power supply*:
  - *member*: 1.
  - *power supply*: 1 to 4.
- Fans are on the rear of the switch and are labeled in software as: *member/tray/fan*:

    - *member*: 1.
    - *tray*: 1 to 4.
    - *fan*: 1 to 4.
- Fabric modules are not labeled on the switch but are labeled in software in the format: *member/module*:
    - *member*: 1.
    - *member*: 1 or 2.
- The display module on the rear of the switch is not labeled with a member or slot number.

VSX

Aruba Virtual Switching Extension (VSX) is virtualization technology for aggregation/core switches running the AOS-CX operating system. This solution lets the switches present as one virtualized switch in critical areas. Configuration synchronization is one aspect of this VSX solution where the primary switch configuration is synced to the secondary switch. This solution allows for a pseudo single plane of glass configuration and helps keep key configuration pieces in synchronization as operational changes are made. Since the solution is primarily for high availability, it is expected that most of the configuration policy is the same across both peers.

VSX virtualizes the control plane of two aggregation switches to function as one device at layer 2 and as independent devices at layer 3. From a datapath perspective, each device does an independent forwarding lookup to decide how to handle traffic. Some of the forwarding databases, such as the MAC and ARP tables, are synchronized between the two devices using a proprietary VSX control plane. Some of the forwarding databases are built independently by each switch.

Benefits of VSX

VSX has similar benefits as Virtual Switching Framework (VSF), however, VSX also offers better high availability required in core and data center environments. VSX binds two AOS-CX switches of the same model type to operate as one device for layer 2. VSX also operates as independent nodes for layer 3.

- Control plane:
  - Dual control plane for better resiliency
  - Unified management (synchronized configuration and easy troubleshooting)
  - Live software upgrade with near zero downtime
  - In-chassis redundancy for the 8400 series switches and device level redundancy for all other platforms, such as for the 832x and 6400 series switches.
- Layer 2 distributed LAGs (aggregation switches to access switches):
  - Loop-free L2 multipathing (active-active)
  - Rapid failover
  - Simple configuration
  - No Spanning Tree Required
- Layer 3 distributed LAGs (core switches to aggregate switches)
  - Distributed Layer 3 over VSX pair (various options: Routed Only Ports (ROPs), Switched Virtual Interfaces (SVIs), or LAG SVIs)
  - Unified datapath (active-active first hop gateway)
  - Layer 3 ECMP + Layer 2 VSX LAG (highly fault tolerant) with active-forwarding
- Active Gateway:

- ◦ Active-Active first hop gateway (VIP)
- ◦ No VRRP/HSRP
- ◦ Simple configuration (one command)
- ◦ No gateway protocol overhead
- ◦ DHCP relay redundancy
- ◦ IP multinetting support

VSX solution topology overview

- **Active gateway support:** Active gateways can be configured for active-active routing. VRRP can be used, as an alternative, for active-standby routing.
- **ISL links assigned to higher bandwidth:** An ISL link has a higher bandwidth compared to VSX links. When planning the topology, consider sizing the ISL link according to the traffic volume required for the east-west traffic of a single-homed VSX during a failover scenario.
- **Increasing resiliency:** When creating a LAG with multiple ports on each chassis-based switch, it is a best practice to create the LAG with members from multiple line cards. This technique increases the points of resiliency.
- **Same VLAN configurations:** Both VSX switches have the same VLAN configurations. Make sure that no topology loop is formed because an ISL is added as a member to all the VLANs by default. You can make configuration synchronization automatic between the VSX switches by enabling VSX synchronization.
- **Upstream device from VSX switches:** Connections to the upstream device from the VSX switches have sufficient bandwidth to handle traffic from all VSXs.

NOTE: Core-1 and Core-2, shown in the following figure, can be third-party devices, as long as they support LACP for downstream connectivity to the VSX LAG. VSX Synchronization syncs from the primary switch (shown as Agg-1 in the following diagram) to the secondary switch (shown as Agg-2 in the following diagram).

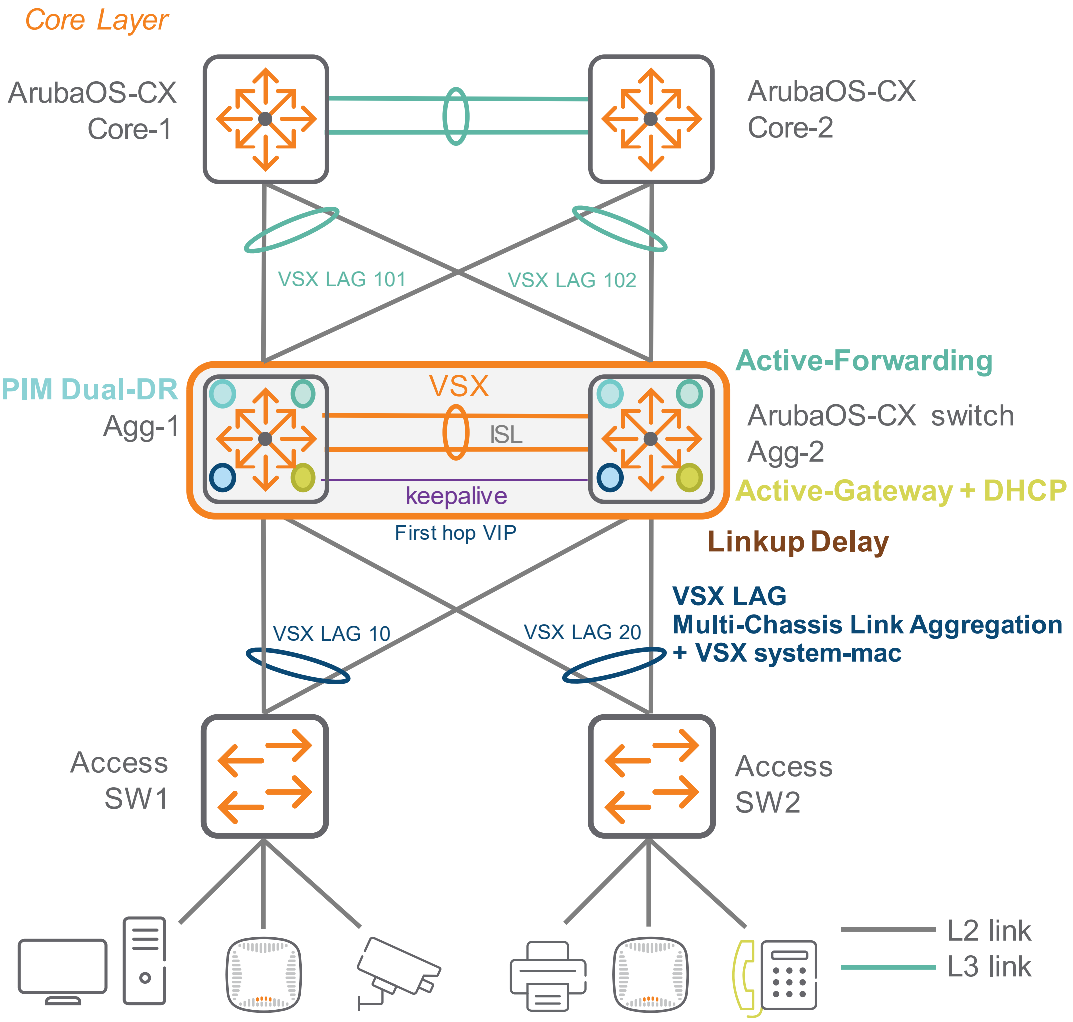
To configure Core-1 and Core-2 with AOS-CX, see **Configuring core 1 and core 2 for VSX**.

To configure the aggregate 1 and aggregate 2, see **Configuring the two aggregate VSX switches**.

To configure the access switch, see **Configuring an AOS-CX switch as an access switch**.

After setting up the VSX topology, **enable VSX synchronization for a feature**. VSX synchronization can be enabled globally for some features, and VSX synchronization can be enabled at the context level for other features.

Figure 1: *Sample VSX solution topology*



More information

Benefits of active forwarding and active gateway
Active gateway over VSX
Active gateway configurations
Active forwarding

VSX LAG

VSX LAGs span both aggregation switches. The two switches appear as one device to partner downstream or upstream devices or both when forming a LAG with the VSX pair. The two switches synchronize their databases and states over a user configured link referred to as an Inter-Switch Link (ISL).

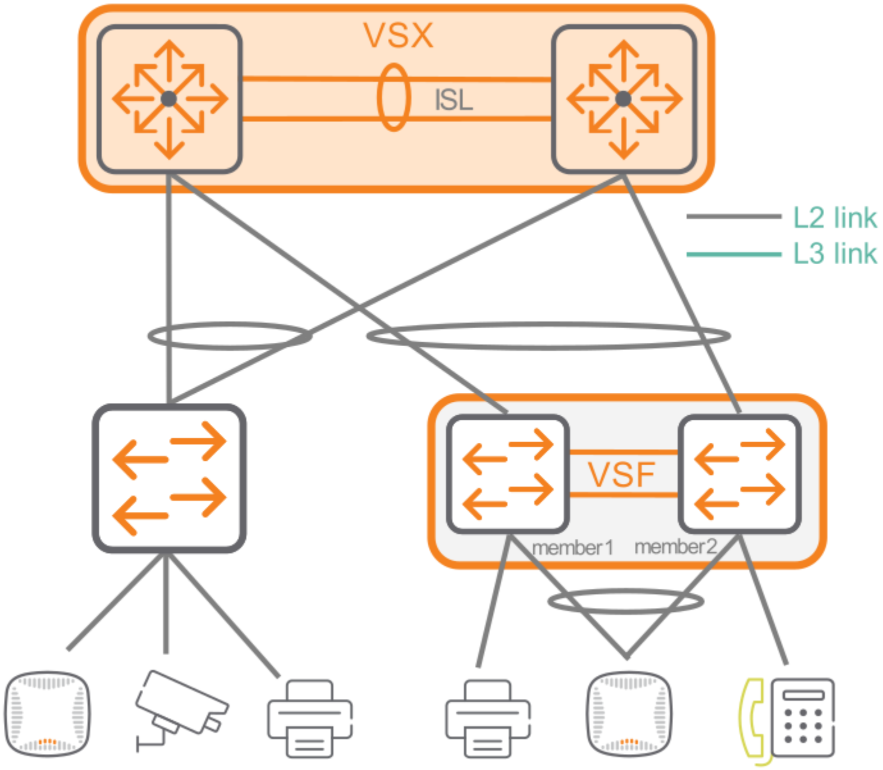
VSX LAGs are preferable to point-to-point transit VLANs for upstream connectivity when the routed only port is not an option, such with the case of multiple VRFs. This configuration reduces the number of transit VLANs and associated SVIs, simplifying operations and troubleshooting. Enable active forwarding and active gateway to further optimize the traffic path. When you enable active forwarding and active gateway, north-south and south-north traffic bypasses the ISL link.

More information

Benefits of active forwarding and active gateway
Active gateway over VSX
Active gateway configurations
`interface lag multi-chassis`
Active forwarding

VSF

Virtual Switching Framework (VSF) technology virtualizes multiple physical devices into one virtual fabric which provides high availability because of the significant reduction in recovery time simplified network design and management. VSF is ideal for campus access. VSF lets supported switches connected to each other through Ethernet connections (copper or fiber) to behave like a single chassis switch.



VSF versus VSX

VSF	VSX
Single control plane	Dual control plane
Single management plane with commander pushing configuration on all members	Dual management plane with "opt-in" configuration synchronization
Dual port state: Enabled	Default port state: Disabled
Layer 2 ports	Layer 3 ports
Ideal for campus access	Ideal for campus agg/core

The common system MAC address

The common system MAC address is used for preventing traffic disruptions when the primary switch is restored after the secondary switch. A primary switch might be restored after the secondary switch in scenarios, such as:

- A primary switch hardware replacement.
- A power outage with the primary switch restored after the secondary switch is restored.

When the primary switch is restored after the secondary switch, a traffic disruption might occur when the ISL starts to sync because the MAC system address changes from the secondary switch to the primary switch for the LACP. To avoid the traffic disruption, set the common system MAC address by entering the `system-mac <MAC-ADDR>` command. This command creates a common system MAC address between the two VSX switches. This common system MAC address prevents a traffic disruption when the secondary switch comes up before the primary switch. If the common system MAC access is enabled, the secondary switch uses the common system MAC address instead of its own system MAC address, which prevents a traffic loss.

The system MAC address also maintains the same MSTP bridge ID across VSX switches, which act as a single switch.

More information

`system-mac`

VSX solution requirements

- **All VSX switches in an environment must have identical settings for the following:**
  - The VLAN membership for all VSX trunk ports.
  - The loop protection configuration on a VLAN that is part of a VSX LAG.
- **Available ports:** Make sure that the VSX LAG interface on both the VSX primary and secondary switches has a member port configured and enabled. Make sure that you also have a non-VSX port that is available for the ISL.
- **Mutually exclusive features:**
  - VSX active-forwarding and VSX active-gateway on the same VLAN interface
  - VSX active-gateway and VRRP at SVI context
  - VSX and MVRP

  NOTE: VSX active-gateway and VRRP can co-exist at global level

- **Profiles for 832x series switches:** All switches must be assigned either in profile L3-agg or L3-core.
- **Support for Inter-Switch links (ISLs):** VSX LAG does not support layer 3 processing, such as a routed port; however, multiple Virtual Switch Interfaces (VSI) can be configured on the switch in association with the VLANs carried over the given VSX LAG.
- **Support for Layer 3:** VSX LAG as a route only port is not supported. To enable Layer 3, create an SVI associated to a given VLAN that is enabled on the VSX LAG.
- **VLAN support:** The same list of VLANs that are trunked over the VSX LAGs must be configured on the primary and secondary VSX switches in the global configuration. The list of VLANs can be synced to the

secondary switch if the `vsx-sync` command is used in the VLAN context. Also verify that the VLAN set is also permitted on the ISL on the primary and secondary VSX switches. To configure VLAN trunking on the ISL, enter the `vlan trunk allowed [<VLAN-LIST> | all]` command. If a native VLAN is defined, the switch automatically runs the `vlan trunk allowed all` command to ensure that the default VLAN is allowed on the trunk. To allow only specific VLANs on the trunk, enter the `vlan trunk allowed <VLAN-LIST>` command, for example: `vlan trunk allowed 2,3,4`

  For steps about creating the ISL within a VSX LAG, see **Configuring the two aggregate VSX switches**.

- **VSX active-forwarding, VSX active-gateway, and VSX LAG are supported with BFD.**
- **VSX switches and software versions:** Both VSX peer switches must use the same software version in most situations; however during an upgrade, one switch can run a different version than the peer with some limitations, such as no VSX synchronization support.

More information

Switch roles
Keepalive
Keepalive scenario
Inter-Switch Link (ISL)
`inter-switch-link {<PORT-NUM> | lag <LAG-ID>}`
`vsx-sync`
Active gateway configurations
Enabling VSX synchronization at the context level
`show vsx status`
Keepalive response in ISL failure scenarios
Keepalive configurations
Recommended network configuration for keepalive
ISL configurations
Enabling VSX synchronization at the global level

VSX components

VSX has the following components:

- Active-standby DHCP server
- Common system MAC address
- DHCP forwarder redundancy
- Inter-Switch Link (ISL)
- IGMP snooping
- Keepalive
- Multiple Spanning Tree Protocol (MSTP)
- Split recovery mode
- Switch roles

Inter-Switch Link (ISL)

In the VSX solution topology, an Inter-Switch Link (ISL) is a layer 2 interface between two VSX peer switches. Each VSX switch must be configured with an ISL link connected to its peer VSX switch. It is recommended that this link is peer-to-peer and used for both datapath traffic forwarding and control path VSX protocol exchange. The ISL interface is by default a member of all VLANs on the device. You can change ISL membership through the command line, but you must ensure VLANs that contain VSX LAG members are not excluded from the ISL.

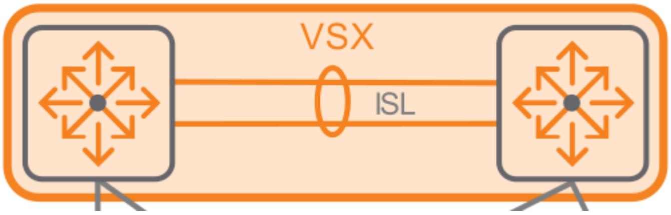
In the datapath, traffic is forwarded natively with no additional encapsulation, unlike VSF. ISL is capable of sending control path data, which requires oversize packets. The ISL MTU is automatically set to the required size to accommodate oversize packets, and cannot be manually overwritten to avoid generating an unintended outage. The token counters of ISL interface show this oversize control path data as part of the ISL operation. The ISL link is the main pipeline for synchronizing data, such as from the following components, during VSX stack join and also permanently between VSX peers:

- ARP table
- LACP states for VSX LAGs
- MAC table
- MSTP states

The ISL uses version control and provides backward compatibility regarding VSX synchronization capabilities.

The ISL can span long distances (transceiver dependent). The traffic that passes over VSX links has no additional encapsulation.

All ISL ports must have the same speed. The speed can be 1G, 10G, 25G, 40G, or 100G, with 40G and 100G being the preferred speeds. For example: 2x40G.



ISL configurations

Task	Command	Example

Configuring an ISL port.	**inter-switch-link**	`switch(config)# vsx`
		`switch(config-vsx)# inter-switch-link lag 100`

Deleting an ISL port.	**no inter-switch-link**	`switch(config)# vsx`
		`switch(config-vsx)# no inter-switch-link`

Configuring ISL dead interval.	**inter-switch-link dead-interval**	`switch(config)# vsx`
		`switch(config-vsx)# inter-switch-link dead-interval 10`

Table Continued

Task	Command	Example

Restore default ISL dead interval.	**no inter-switch-link dead-interval**	`switch(config)# vsx`
		`switch(config-vsx)# no inter-switch-link dead-interval`

Configuring the ISL hello interval.	**inter-switch-link hello-interval**	`switch(config)# vsx`
		`switch(config-vsx)# inter-switch-link hello-interval 3`

Restoring default ISL hello interval.	**no inter-switch-link hello-interval**	`switch(config)# vsx`
		`switch(config-vsx)# no inter-switch-link hello-interval`

Configuring ISL holdtime.	**inter-switch-link hold-time**	`switch(config)# vsx`
		`switch(config-vsx)# inter-switch-link hold-time 2`

Restoring default ISL holdtime.	**no inter-switch-link hold-time**	`switch(config)# vsx`
		`switch(config-vsx)# no inter-switch-link hold-time`

Configuring the amount of time in seconds that the device waits for the ISL interface to link up after a reboot.	**inter-switch-link peer-detect-interval**	`switch(config)# vsx`
		`switch(config-vsx)# inter-switch-link peer-detect-interval 180`

Default values:

- Dead interval: 20 seconds
- Hello interval: 1 second
- Hold time: 0 seconds
- Peer detect interval: 300 seconds

Switch roles

Each VSX switch must be configured with a role – primary or secondary. The roles do not indicate which device is forwarding traffic at a given time as VSX is an active-active forwarding solution. The roles are used to determine which device stays active when there is a VSX split, such as when the ISL goes down, and for determining the direction of configuration-sync. If the VSX ISL goes down, the primary switch keeps forwarding traffic while the secondary switch blocks ports from participating in the VSX LAGs.

VSX switch reboot

After a VSX switch reboots, it has no entries for ARP, MAC, and routes. If downstream VSX LAG ports are activated before all this information is relearned, traffic is dropped. To avoid a traffic drop, VSX LAGs on the rebooted switch stay down until the restoration of LACP, MAC, ARP databases, and MSTP states if MSTP is used.

The learning process for the VSX LAGs has two phases:

- **Initial sync phase:** The LACP states, MAC address table, ARP table, and potentially MSTP states are downloaded from the forwarding switch to the freshly rebooted switch.
- **Link-up delay phase:** The downloaded entries are installed into the ASIC. Router adjacencies with core nodes and learned upstream routes are also established.

  The link-up delay phase is configurable with the `linkup-delay-timer` *`<DELAY-TIMER>`* command. The default value is 180 seconds. Set the link-up delay timer to the maximum value of 600 seconds for a network with many MAC addresses, a large ARP table, or a large routing table.

When both VSX switches reboot, the link-up delay timer is not used because both switches are trying to relearn the LACP states, MAC address table, and ARP table.

To get upstream router adjacencies established during the link-up delay, the upstream LAGs have to be excluded from the scope of the link-up delay. Run the `linkup-delay-timer exclude lag-list` *`<LAG-LIST>`* for identifying the LAGs for exclusion.

For example, assume that you have a topology similar to the one in **VSX solution topology overview**, the upstream LAGs (LAG 101 and LAG 102), would need to be identified by the `linkup-delay-timer exclude lag-list` *`<LAG-LIST>`* for exclusion before a VSX switch reboot.

More information

`linkup-delay-timer`
`linkup-delay-timer exclude lag-list`

Periodic synchronization

Each VSX node synchronizes every second the following with its VSX peer through ISLP:

- Learned MAC addresses
- LACP states
- STP states

In a VSX scenario if all traffic from core to access flows through one switch only, the other device will also learn about ARP/ND from its VSX peer. There is no functional impact to the normal datapath based learning. VSX split and the rejoin scenario, such as periodic synchronization, resumes after the bulk synchronization.

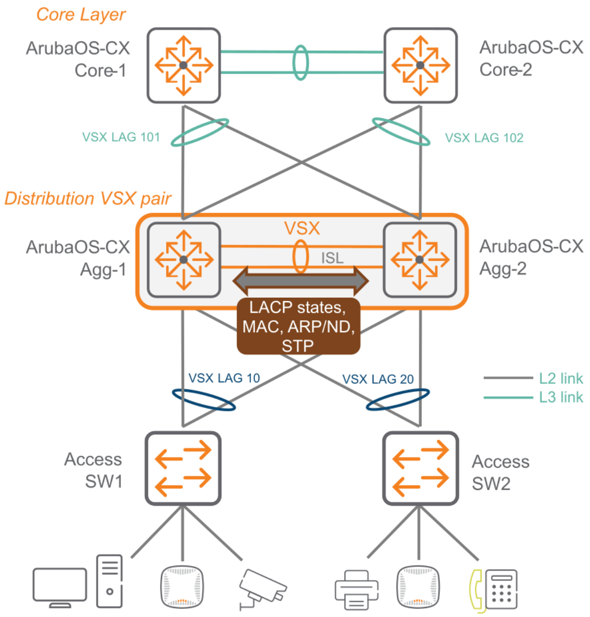
The IVRL induced neighbor entries are not synced either through the initial ARP synchronization or periodic synchronization. The local IVRL will induce the learning on each side triggered by either a local data-path ARP learned or the ongoing sync based-ARP learned in the source VRF.

IMPORTANT: If you enter one or more of the following commands on one VSX switch but not on the other VSX switch or any configuration is incorrect on one switch , the ARP entries on both switches will become unsynchronized:

- `interface vlan`
- `shutdown` for a VLAN
- `no shutdown` for a VLAN

If you run into this situation, correct the configuration and run the `clear arp` command, which clears the ARP entries on both VSX nodes. After you run the `clear arp` command, the ARP entries are synchronized on both switches.

The following image shows how periodic synchronization synchronizes LACP states, MAC, ARP/ND, and MSTP. The image references the VSX synchronization between aggregate 1 (the primary VSX switch) and aggregate 2 (the secondary VSX switch).



BFD and VSX support

BFD supports VSX LAG, active gateway, and active forwarding.

For 832x series switches: Several TCAMs (ternary content-addressable memory) are used to avoid decreasing the time-to-live (TTL) to 254 on BFD single-hop packets received/sent on ISL interfaces.

For 8400 series switches: To account for the TTL decrement on active forwarding, the BFD daemon supports packets with TTL equal to 254 on sessions running on ports with this functionality active.

More information

BFD reports a link being down before the LAG rebalances

Chapter 3
Upgrading to the latest version of VSX

This chapter provides information about upgrading customer configurations to the latest version of the Virtual Switching Extension (VSX).

- If you are upgrading from version 10.00, see the *Aruba Virtual Switching Extension (VSX) Guide for 10.02* for steps on how to upgrade VSX to version 10.02. Then, see the steps in this guide on how to upgrade VSX from version 10.02.
- If you are upgrading from version 10.01/10.02, see the *Aruba Virtual Switching Extension (VSX) Guide for 10.03* for steps on how to upgrade VSX to version 10.03. Then, see the steps in this guide on how to upgrade VSX from version 10.03.

Upgrading VSX from 10.03/10.04/10.05 to 10.06

Upgrade from AOS-CX 10.03/10.04/10.05 to 10.06 by:

- **Running the `vsx update-software` command:** Follow the steps in **Upgrading switches by using the `vsx update-software` command** for required steps before and after running the `vsx update-software vsx update-software` command.

  This command downloads new software from the TFTP server and verifies the download. After a successful verification, the command installs the software to the alternative software bank of both the VSX primary and secondary switches. The command then reboots them in sequence, the VSX secondary switch followed by VSX primary switch. For example if a switch has booted with the primary flash memory, then the command will install the software to secondary flash memory.
- **Running the `vsx update-software boot-bank` command:** This command upgrades the VSX pairs using the specified boot bank on both the devices. Before running this command, copy the new software by adding the TFTP URL into the software bank of both primary and secondary VSX switches. Use this command in cases where the scheduled maintenance window is minimum or to avoid TFTP server timeout. For more information about this command, see **`vsx update-software boot-bank`**.
- **Running Aruba NetEdit to upgrade to the latest version of VSX.** Refer to the Aruba NetEdit documentation.

Upgrading switches by using the `vsx update-software` command

Prerequisites

1. Ensure that there is a scheduled maintenance window. There will be a minimal disruption of service until the upgrade is completed.
2. If you have enabled loop protect, enter the `show loop-protect` command for verifying that the action on loop detection has a value of `TX disable` on the VSX interface. If the setting has a different value, reset the value to `TX disable` by entering the `loop-protect action tx-disable` command:

```
switch(config)# interface lag 2 multi-chassis
switch(config-if)# loop-protect action tx-disable
switch(config-if)# exit
switch(config)# exit
```

3. The `vsx update-software` command provides the option to save the configuration on the primary and secondary VSX switches; however, you can save the configuration manually by using one of the following methods:

   - To copy the running configuration into the startup configuration:

     ```
     switch# copy running-config startup-config
     ```

     If the startup configuration is already present, the command overwrites the pre-existing startup configuration.

   - To copy the running configuration into a checkpoint that has not been created yet:

     ```
     switch# copy running-config checkpoint <CHECKPOINT-NAME>
     ```

4. Check the status of the `show vsx brief` command and validate the ISL is in-sync and the keepalive is established.
5. Check the output of the `show lacp interfaces multi-chassis` command and note which LACP interfaces are in a forwarding state of up.

Procedure

1. Enter the `vsx update-software` command. Prefix the path, for downloading the software, with `tftp://`, as shown in the following example:

   ```
   switch# vsx update-software tftp://192.168.1.1/XL.10.0x.xxxx vrf mgmt
   Do you want to save the current configuration (y/n)? y
   The running configuration was saved to the startup configuration.

   This command will download new software to the %s image of both the VSX primary and secondary systems,
   then reboot them in sequence. The VSX secondary will reboot first, followed by the primary.
   Continue (y/n)? y
   VSX Primary Software Update Status     : <VSX primary software update status>
   VSX Secondary Software Update Status   : <VSX secondary software update status>
   VSX ISL Status                         : <VSX ISL status>
   Progress [...........................................................................................]
   Secondary VSX system updated completely. Rebooting primary.
   ```

   This command gives you the option to save the running configuration on the primary and secondary VSX switches. After the command saves the running configuration, it downloads new software from the TFTP server and verifies the download. After a successful verification, the command installs the software to the alternative image of both the VSX primary and secondary switches.

   The command displays the status of the VSX primary and secondary switches during the upgrade. The command also refreshes the progress bar as the image update progresses. Do not interrupt the VSX primary CLI session until the software updates completes; however, software update process can be stopped. If you stop the upgrade when the secondary switch has already installed the image in its flash memory or the secondary switch has started the reboot the process, it comes up with the new software. The primary switch continues to have with older software. You can stop the software update process by pressing **ctrl+c**.

2. Run the `show vsx brief` command on both switches. Verify that ISL is `In-Sync` by running the `show vsx brief` command on both switches. Verify in the output of the command that the keepalive state is `Keepalive-Established`.
3. Validate on both switches that the downstream LACP links are all forwarding correctly by entering the `show lacp interfaces` command.
4. Save the running configuration to the startup configuration:

```
switch# write memory
Success
```

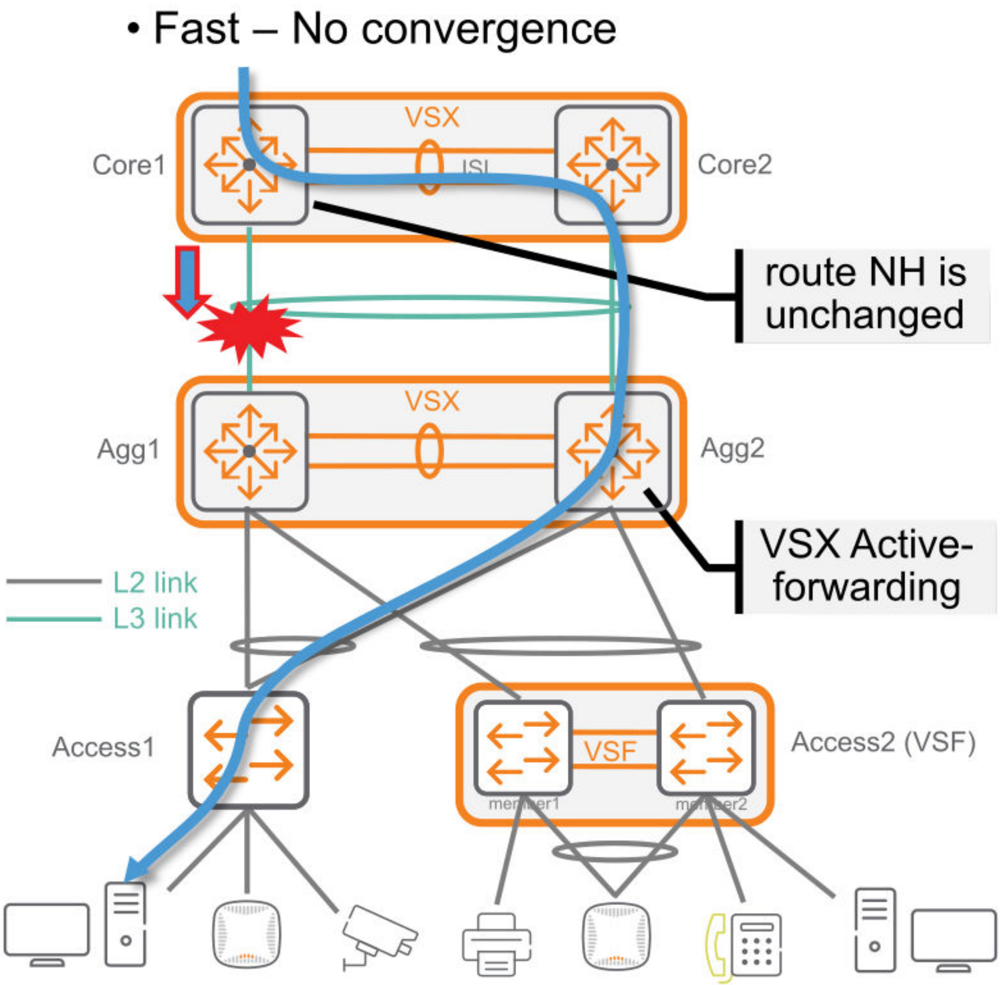
More information

vsx update-software

VSX in the core layer

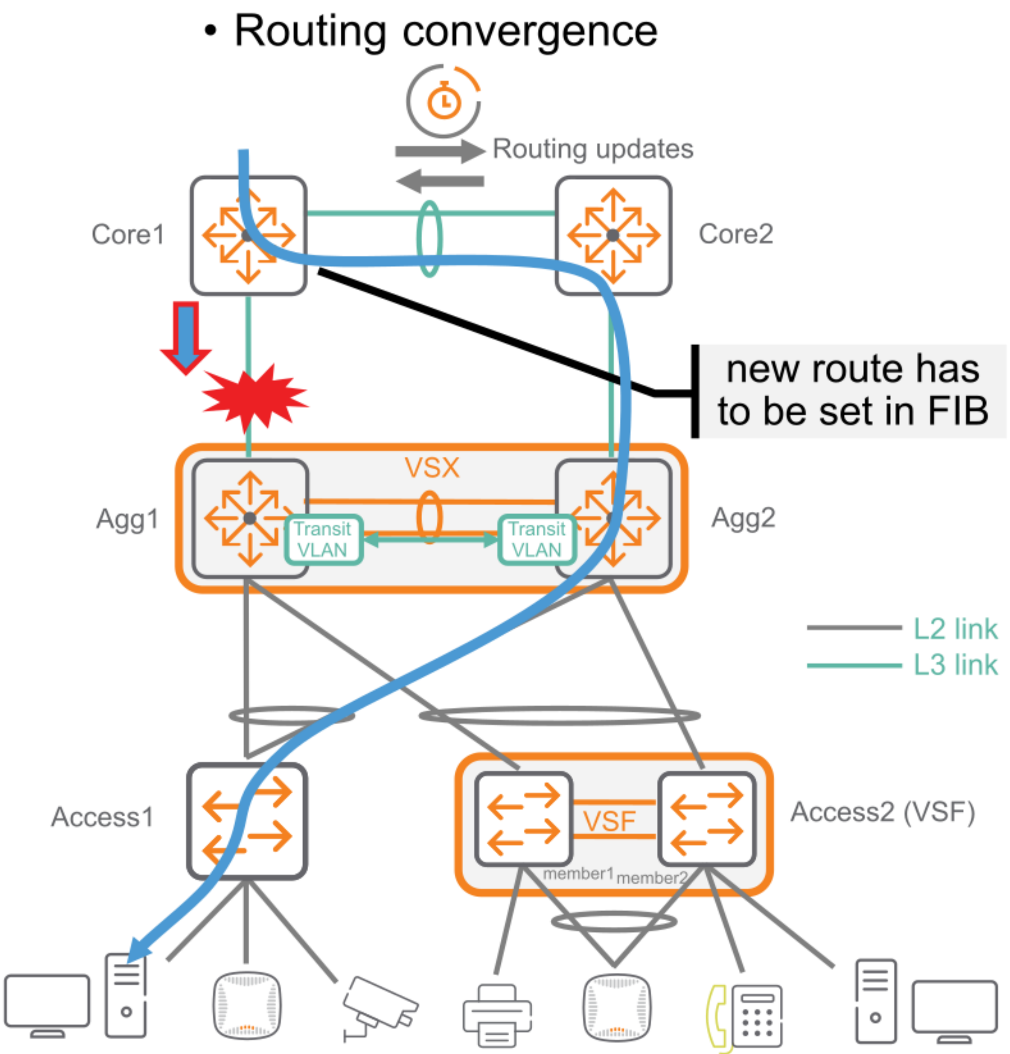
When mobility controllers are attached to the core layer, a VSX LAG must be in the core level. Layer 2 is only at the distribution layer with the core layer being layer 2 and layer 3. This configuration is for IPv6, and the configuration reduces the number of transit VLANs and the associated SVIs for many VRFs. It also minimizes the shortest path first (SPF) calculation. The number of fibers is reduced and there is fast failover because of the simplified topology (no square-routing).

Figure 2: *VSX LAG in the core (recommended)*



The following image shows an example that SPF is slower when VSX is not in the core because of routing convergence.

Figure 3: *VSX LAG not in the core*



Configuring core 1 and core 2 for VSX

The steps in this section are for configuring core 1 and core 2 for VSX, as displayed in **Figure 2: VSX LAG in the core (recommended)**.

After completing these steps, configure the aggregate switches in your network topology, as described in **Configuring the two aggregate VSX switches**. Then, enable VSX configuration synchronization for a feature, as described in **Enabling VSX configuration synchronization**.

A VSX LAG supports a maximum of four member links per switch segment. A VSX LAG across a downstream switch can have at most a total of eight member links. Run the `show capacities` command for the maximum number of VSX LAGs supported for your type of switch.

The core can be third-party devices, as long as they support LACP for downstream connectivity to the VSX LAG. VSX synchronization syncs from the primary switch (aggregate 1) to the secondary switch (aggregate-2).

NOTE: When creating a VSX LAG, select an equal number of member links in each segment for load balancing, such as four member links (one segment) and four member links (another segment). Do not create a VSX LAG with four member links in one switch and two member links on another segment. A switch can have a maximum of four member links.

Procedure

1. Access the prompt on the switch you want to make the primary core switch.
2. If the switch lacks a hostname, create one:

   ```
   switch(config)# hostname <HOSTNAME>
   ```

3. Create the required VLANS:

   ```
   switch(config)# vlan 1-20
   ```

4. Enable OSPFv2:

   ```
   switch(config)# router ospf 1
   switch(config-ospf-1)# redistribute connected
   switch(config-ospf-1)# area 0.0.0.0
   ```

5. Enable OSPFv3:

   ```
   switch(config)# router ospfv3 1
   switch(config-ospfv3-1)# redistribute connected
   switch(config-ospfv3-1)# area 0.0.0.0
   switch(config-ospfv3-1)# exit
   ```

   OSPFv2 and OSPFv3 are not required to be activated simultaneously. Activate OSPFv2 and OSPFV3 according to the needs of the environment.

6. Create a loop back interface and enable OSPFv2/v3:

   ```
   switch(config)# interface loopback 1
   switch(config-loopback-if)# ip address 3.3.3.3/24
   switch(config-loopback-if)# ip ospf 1 area 0.0.0.0
   switch(config-loopback-if)# exit
   ```

7. Enable OSPFv2/v3 on the physical port:

   ```
   switch(config)# interface 1/2/43
   switch(config-if)# no shutdown
   switch(config-if)# ip address 192.168.10.5/24
   switch(config-if)# ipv6 address 2001:11::3/64
   switch(config-if)# ip ospf 1 area 0.0.0.0
   switch(config-if)# ipv6 ospfv3 1 area 0.0.0.0
   switch(config-if)# exit
   ```

8. Create a VLAN for the host network:

   ```
   switch(config)# vlan 200
   switch(config-vlan-200)# interface vlan 200
   switch(config-if-vlan)# ip address 192.168.10.6/16
   switch(config-if-vlan)# ipv6 address 2001:200::1/64
   switch(config-if-vlan)# exit
   ```

9. Enable the port for host communication:

   ```
   switch(config)# interface 1/1/48
   switch(config-if)# no shutdown
   ```

```
switch(config-if)# no routing
switch(config-if)# vlan access 200
switch(config-if)# exit
```

10. Enter `vsx`:

```
switch(config)# vsx
switch(config-vsx)#
```

11. Enter the **`role primary`** command for assigning the primary role to a switch. If you have already gone through these steps for configuring the primary switch and you are now configuring the secondary switch, enter the **`role secondary`** command.

    Setting the primary role on a switch:

```
switch(config-vsx)# role primary
```

    Setting the secondary role on a switch:

```
switch(config-vsx)# role secondary
```

12. Configure a layer 2 interface as an ISL:

```
switch(config-vsx)# inter-switch-link lag 100
```

    In this instance, an ISL was configured over LAG 100.

    NOTE: Before you enter this command, verify that the interface is layer 2 and the LAG is not a VSX LAG.

13. Keepalive helps the core switches continue to stay insynch during an ISL failure. When creating the keepalive path, make sure that the path does not go over the ISL or a VSX LAG. Keepalive can be configure two ways for core 1 and core 2. One way is to enable keepalive between core 1 and core 2 as a direct link. A second way is to create a keepalive path for a loopback interface through the upstream that lacks a VSX LAG.

```
switch(config)# int loopback 0
switch(config-loopback-if)# ip address 192.168.1.1/32
switch(config-loopback-if)# ip ospf 1 area 0
switch(config-loopback-if)# exit
switch(config)# vsx
switch(config-vsx)# keepalive peer 192.168.1.2 source 192.168.1.1 vrf <KA-VRF-
NAME>
switch(config-vsx)# exit
switch(config)# int loopback 0
switch(config-loopback-if)# vrf attach <KA-VRF-NAME>
```

    NOTE: The source of the keepalive interface can be a supported layer 3 interface through the loopback interface, SVI, or layer 3 interface. The source must be reachable to the VSX peer through layer 3. The path can be over the core or direct path. The keepalive path must not be over the ISL. See **Recommended network configuration for keepalive**.

14. Change the context to the `switch(config)#` context:

```
switch(config-vsx)# exit
switch(config)#
```

15. Configuring a LAG interface as an ISL:

```
switch(config)# interface lag <LAG-ID>
```

For example, configuring LAG 100 as an ISL LAG:

```
switch(config)# interface lag 100
switch(config-lag-if)# vsx
switch(config-vsx)# inter-switch-link lag 100
```

16. Repeat the previous steps for the secondary core switch.

17. Enter the **show vsx configuration inter-switch-link** command for confirming the properties of the VSX LAG, such as confirming if the ISL is in-sync.

```
switch# show vsx configuration inter-switch-link
Inter Switch Link    : 1/1/43
Hello Interval       : 1 Seconds
Dead Interval        : 20 Seconds
Hold Time            : 0 Seconds
System MAC           : 10:00:00:00:00:01
Device Role          : primary
Multichassis LAGs    : lag100
```

More information

keepalive peer

interface lag multi-chassis

Configuring the two aggregate VSX switches

The steps in this section are for configuring the two aggregate VSX switches, as described in **VSX solution topology overview**. VSX switches do not automatically have VSX configuration synchronization enabled. After completing the steps in this section, enable VSX configuration synchronization for a feature, as described in **VSX configuration synchronization**. VSX synchronization sync configuration information from the primary switch (Aggregate-1) to the secondary switch (Aggregate-2). After completing the steps in this section, enable VSX configuration synchronization for a feature, as described in **VSX configuration synchronization**.

A VSX LAG supports a maximum of four member links per switch segment. A VSX LAG across a downstream switch can have at most a total of eight member links. Run the `show capacities` command for the maximum number of VSX LAGs supported for your type of switch.

NOTE:

- When creating a VSX LAG, select an equal number of member links in each segment for load balancing, such as four member links (one segment) and four member links (another segment). Do not create a VSX LAG with four member links in one switch and two member links on another segment. A switch can have a maximum of four member links.
- Make sure that the VSX LAG interface on both the VSX primary and secondary switches has a member port configured and enabled.
- Make sure that you also have a non-VSX port that is available for the ISL.

Procedure

1. Access the prompt on the switch you want to make the primary aggregate switch.

2. If the switch does not have a hostname, create one:

```
switch(config)# hostname <HOSTNAME>
```

3. Create the required VLANS:

```
switch(config)# vlan 1-20
```

4. Create the ISL interface:

```
switch(config)# interface lag 128
switch(config-lag-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-lag-if)# vlan trunk native 1
switch(config-lag-if)# lacp mode active
```

When a native VLAN is defined (as shown this example), the switch automatically executes the `vlan trunk allowed all` command to ensure that the default VLAN is allowed on the trunk. In this example, LAG 128 is being used as the ISL.

The same list of VLANs that are trunked over the VSX LAGs must be configured on the primary and secondary VSX switches in the global configuration. The list of VLANs can be synced to the secondary switch if the `vsx-sync` command is used in the VLAN context. Also verify that the VLAN set is also permitted on the ISL on the primary and secondary VSX switches. To configure VLAN trunking on the ISL, enter the `vlan trunk allowed [<VLAN-LIST> | all]` command. If a native VLAN is defined, the switch automatically runs the `vlan trunk allowed all` command to ensure that the default VLAN is allowed on the trunk. To allow only specific VLANs on the trunk, enter the `vlan trunk allowed <VLAN-LIST>` command, for example: `vlan trunk allowed 2,3,4`

5. Add a physical interface into the LAG:

```
switch(config)# interface 1/4/28
switch(config-if)# no shutdown
switch(config-if)# lag 128
switch(config)# interface 1/4/32
switch(config-if)# no shutdown
switch(config-if)# lag 128
```

6. Enable the interface for keepalive communication:

```
switch(config)# interface 1/1/5
switch(config-if)# ip address 192.168.100.1/24
```

7. Go to the `vsx` context:

```
switch(config)# vsx
switch(config-vsx)#
```

8. Enter the **role primary** command for assigning the primary role to a switch. If you have already gone through these steps for configuring the primary switch and you are now configuring the secondary switch, enter the **role secondary** command.

Setting the primary role on a switch:

```
switch(config-vsx)# role primary
```

Setting the secondary role on a switch:

```
switch(config-vsx)# role secondary
```

9. Enable ISL:

```
switch(config-vsx)# inter-switch-link lag 128
```

In this example, ISL is being enabled for LAG 128.

NOTE: Before you enter this command, verify that the interface is layer 2 and the LAG is not a VSX LAG.

10. Enable keepalive:

```
switch(config-vsx)# keepalive peer 192.168.100.2 source 192.168.100.1
```

In this example, 192.168.100.2 is the peer IP address and 192.168.100.1 is the source IP address.

11. Enable the multichassis interface:

```
switch(config)# interface lag 1 multi-chassis
switch(config-lag-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-lag-if)# vlan trunk native 1
switch(config-lag-if)# vlan trunk allowed 11
```

12. Add physical interfaces into the multichassis interface:

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# lag 1
```

13. Create an active gateway SVI:

```
switch(config)# interface vlan 11
switch(config-if-vlan)# ip address 192.168.100.5/16
switch(config-if-vlan)# ipv6 address 2001:DB8::2/64
switch(config-if-vlan)# active-gateway ip 192.168.100.2 mac 00:00:00:00:00:01
switch(config-if-vlan)# active-gateway ipv6 2001:DB8::3 mac 00:00:01:00:00:01
```

14. Enable uplink communication for OSPFv2:

```
switch(config)# router ospf 1
switch(config-ospf-1)# redistribute connected
switch(config-ospf-1)# area 0.0.0.0
```

The `redistribute connected` command is optional in this example. See the *Command-Line Interface Guide* for your switch and software version for more information about the `redistribute connected` command.

15. Enable uplink communication for OSPFv3:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# redistribute connected
switch(config-ospfv3-1)# area 0.0.0.0
```

The `redistribute connected` command is optional in this example. See the *Command-Line Interface Guide* for your switch and software version for more information about the `redistribute connected` command.

16. Create the loopback interface and enable OSPFv2:

```
switch(config)# interface loopback 1
switch(config-loopback-if)# ip address 192.168.0.1/32
switch(config-loopback-if)# ip ospf 1 area 0.0.0.0
```

17. Enable OSPFv2/v3 on the physical port:

```
switch(config)# interface 1/4/30
switch(config-if)# no shutdown
switch(config-if)# ip address 192.168.10.0/31
switch(config-if)# ipv6 address 2001:11::1/64
switch(config-if)# ip ospf 1 area 0.0.0.0
switch(config-if)# ipv6 ospfv3 1 area 0.0.0.0
```

18. Repeat the previous steps for the secondary aggregate switch.
19. View the running configuration by entering the following on the primary and secondary switches:

    ```
    switch# show running-config
    ```

20. Verify that the ISL link is in-sync, the role of the switch, and the keepalive state (if enabled) by entering the following on the primary and secondary switches:

    ```
    vsx-primary# show vsx brief
    ISL State                             : In-Sync
    Device State                          : Peer-Established
    Keepalive State                       : Keepalive-Established
    Device Role                           : primary
    Number of Multi-chassis LAG interfaces : 2
    ```

21. Verify the VSX status by entering the following on the primary and secondary switches:

    ```
    switch# show vsx status
    VSX Operational State
    ---------------------
      ISL channel             : In-Sync
      ISL mgmt channel        : operational
      Config Sync Status      : in-sync
      NAE                     : peer_reachable
      HTTPS Server            : peer_reachable


    Attribute          Local                       Peer
    ------------       --------                    --------
    ISL link           1/1/43                      1/1/43
    ISL version        2                           2
    System MAC         48:0f:cf:af:70:84           48:0f:cf:af:c2:84
    Platform           8320                        8320
    Software Version   10.0x.xxxx                  10.0x.xxxx
    Device Role        primary                     secondary
    ```

22. Verify the LACP interface status by entering the following on the primary and secondary switches:

    ```
    switch# show lacp interfaces
    ```

23. Verify the uplink (layer 3 communication) by entering the following on the primary and secondary switches:

    ```
    switch# show ip ospf neighbors
    ```

Configuring an AOS-CX switch as an access switch

An access switch can be any switch that supports LACP or static link aggregation. The steps in this section are specifically for an AOS-CX switch. For non-AOS-CX switches, refer to the documentation for your switch about how to enable LACP or static link aggregation.

Prerequisites

These steps assume that you have an AOS-CX switch.

Procedure

1. If the switch lacks a hostname, create one:

   ```
   switch(config)# hostname <HOSTNAME>
   ```

   For example:

   ```
   switch(config)# hostname Finance01
   ```

2. Create a VLAN:

   ```
   switch(config)# vlan <VLAN-ID>
   ```

   For example:

   ```
   switch(config)# vlan 11
   ```

3. Create a LAG:

   ```
   switch(config)# interface lag <ID>
   ```

   For example:

   ```
   switch(config)# interface lag 2
   ```

4. Enable LACP for the LAG:

   ```
   switch(config-lag-if)# lacp mode active
   ```

5. Create either an access interface **or** a trunk interface. You can create both allowed and native trunk interfaces on the access switch.

   - To create an access interface:

     ```
     switch(config-lag-if)# vlan access <VLAN_ID>
     ```

     For example:

     ```
     switch(config-lag-if)# vlan access 5
     ```

   - To create a native trunk interface:

     a. To create an allowed trunk interface:

        ```
        switch(config-lag-if)# vlan trunk allowed <VLAN_LIST>
        ```

        For example:

        ```
        switch(config-lag-if)# vlan trunk allowed 30,50,120
        ```

     b. To create a native trunk interface:

        ```
        switch(config-lag-if)# vlan trunk native <VLAN_ID> [tag]
        ```

        For example:

        ```
        switch(config-lag-if)# vlan trunk native 30 tag
        ```

The trunk parameter enables tagging on a native VLAN. Only incoming packets that are tagged with the matching VLAN ID are accepted. Incoming packets that are untagged are dropped except for BPDUs. Egress packets are tagged.

6. For multiple access switches in your topology, repeat the previous steps.

7. Verify the configuration:

```
switch# show lacp interfaces
```

VSX configuration synchronization

VSX configuration synchronization simplifies VSX solution management, reduces configuration misconfiguration and drift across VSX peer switches. With configuration synchronization enabled, the primary peer configuration is synced to the secondary peer. This synchronization is controlled in an opt-in manner by enabling VSX synchronization on a section of configuration.

If one or more of the following scenarios occur, the secondary switch will receive the configuration update after it fulfills synchronization requirements and is fully enabled:

- The secondary switch is not currently present.
- The secondary switch is not currently connected to the primary switch through the ISL.
- The secondary switch is not currently configured for VSX configuration synchronization at the time VSX configuration synchronization is enabled on the primary switch.

You can only enable a specific configuration for syncing through the `vsx-sync` CLI extension on the primary switch. This extension is blocked on the secondary peer switch except when VSX configuration-synchronization is disabled or the ISL link is down.

Features supporting VSX

You can enable VSX synchronization at:

- **The global level:** See **Enabling VSX synchronization at the global level** for a listing of features supporting VSX synchronization at the global level.
- **The context level:** See **Enabling VSX synchronization at the context level** for a listing of features supporting VSX synchronization at the context level.

VSX synchronization requirements

- Software image versions must be the same on both switches.
- The output from the `show vsx status` command must show `in-sync` for Config Sync Status.
- Primary and secondary roles configured.
- An interswitch link must be configured.
- When enabling VSX synchronization under a physical interface, a VLAN interface, or a VSX LAG, create on the secondary switch the physical interface, VLAN interface, or VSX LAG with the same name and routing setting as on the primary switch. For example, if the primary switch has a physical interface of 1/1/1, you must create another physical interface of 1/1/1 on the secondary switch. Also, if the primary VSX switch has routing enabled, the secondary switch must have routing enabled. Once the name and routing information is the same, VSX synchronization synchronizes the additional configuration information from the primary VSX switch to the secondary VSX switch.

IMPORTANT: It is recommended to:

- Enable keepalive for preventing traffic loss during an ISL link failure.
- Assign a common system MAC to prevent traffic loss in cases when the secondary VSX switch is restored before the primary VSX switch.

Enabling VSX synchronization at the global level

The commands in this table are for enabling VSX synchronization at the global level for a feature.

Feature	Command for enabling	Example

AAA configurations, including user, RADIUS server, and TACACS+ server.	`vsx-sync aaa`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync aaa`

Access List Log Timer configurations.	`vsx-sync acl-log-timer`	`switch(config)# access-list log timer 30`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync acl-log-timer`

ARP security configurations.	`vsx-sync arp-security`	`primary_sw(config)# vsx`
		`primary_sw(config-vsx)# vsx-sync arp-security`
		`primary_sw(config-vsx)# vsx-sync mclag-interfaces`

BFD configurations.	`vsx-sync bfd-global`	`switch(config)# bfd detect-multiplier 1`
		`switch(config)# bfd min-transmit-interval 1000`
		`switch(config)# bfd min-receive-interval 1000`
		`switch(config)# bfd echo-src-ip-address 2.2.2.2`
		`switch(config)# bfd min-echo-receive-interval 1000`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync bfd-global`

Table Continued

Feature	Command for enabling	Example

BGP configurations.	`vsx-sync bgp`	`switch(config)# ip aspath-list list1 seq 10 permit 10`
		`switch(config)# ip community-list expanded com1 seq 10 permit 10`
		`switch(config)# ip extcommunity-list standard ext1 seq 10 permit rt 10:4`
		`switch(config)# ip prefix-list pref1 seq 10 permit any`
		`switch(config)# route-map rm1 permit`
		`switch(config-route-map-rm1-10)# match ip next-hop 1.1.1.1`
		`switch(config)# router bgp 100`
		`switch(config-bgp)# bgp router-id 1.1.1.1`
		`switch(config-bgp)# neighbor 12.1.1.1 remote-as 1`
		`switch(config-bgp)# address-family ipv4 unicast`
		`switch(config-bgp-ipv4-uc)# neighbor 12.1.1.1 activate`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync bgp`

CoPP policy configurations.	`vsx-sync copp-policy`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync copp-policy`

DCBx configurations (8325 and 8360 series switches).	`vsx-sync dcb-global`	`switch(config)# lldp dcbx`
		`switch(config)# dcbx application iscsi priority 7`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync dcb-global`

DHCPv4 and DHCPv6 relay configurations.	`vsx-sync dhcp-relay`	`switch(config)# interface 1/1/1`
		`switch(config-if)# ip helper-address 192.168.10.1`
		`switch(config-if)# ip helper-address 192.168.20.1`
		`switch(config)# interface 1/1/2`
		`switch(config-if)# ip helper-address 192.168.30.1`
		`switch(config)# dhcp-relay option 82`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync dhcp-relay`

Table Continued

Feature	Command for enabling	Example

DHCPv4 server configurations, including external storage configurations.	`vsx-sync dhcp-server`	`switch(config)# dhcp-server external-storage dhcp-dbs file dhcpv4_lease_file delay 600`
		`switch(config)# dhcp-server vrf default`
		`switch(config-dhcp-server)# pool test`
		`switch(config-dhcp-server-pool)# range 10.0.0.20 10.0.0.30`
		`switch(config-dhcp-server-pool)# default-router 10.0.0.1 10.0.0.10`
		`switch(config-dhcp-server-pool)# static-bind ip 10.0.0.1 mac 24:be:05:24:75:73`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync dhcp-server`

DHCPv6 server configurations, including external storage configurations.	`vsx-sync dhcpv6-server`	`switch(config)# dhcpv6-server external-storage dhcpv6-dbs file dhcpv6_lease_file delay 600`
		`switch(config)# dhcp-server vrf default`
		`switch(config-dhcp-server)# pool test`
		`switch(config-dhcpv6-server-pool)# range 2001::1 2001::10 prefix-len 64`
		`switch(config-dhcpv6-server-pool)# option 22 ipv6 2001::12`
		`switch(config-dhcpv6-server-pool)# static-bind ipv6 2001::11 client-id 1:0:a0:24:ab:fb:9c`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync dhcpv6-server`

DNS configurations.	`vsx-sync dns`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync dns`

EVPN configurations.	`vsx-sync evpn`	`switch(config)# vlan 2`
		`switch(config-vlan-2)# vsx-sync`
		`switch(config)# evpn`
		`switch(config-evpn)# vlan 2`
		`switch(config-evpn-vlan-2)# rd 5:5`
		`switch(config-evpn-vlan-2)#  route-target export 1:1`
		`switch(config-evpn-vlan-2)#  route-target import 1:1`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync evpn`

Global classifier policy configurations.	`vsx-sync policy-global`	`switch(config)# apply policy testPolicy in`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync policy-global`

IP ICMP configurations.	`vsx-sync icmp-tcp`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync icmp-tcp`

Table Continued

Feature	Command for enabling	Example

LLDP configurations.	`vsx-sync lldp`	`switch(config)#  lldp reinit 6`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync lldp`

Loop protect configurations, such as transmit-interval and re-enable-timer.	`vsx-sync loop-protect-global`	`switch(config)# loop-protect transmit-interval 10`
		`switch(config)# loop-protect re-enable-timer 300`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync loop-protect-global`

MAC lockout configurations.	`vsx-sync mac-lockout`	`switch(config)# mac-lockout 10:10:10:10:10:10`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync mac-lockout`

ND snooping configurations	`vsx-sync nd-snooping`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync nd-snooping`

Static neighbor configurations.	`vsx-sync neighbor`	`DUT-1 (config-vsx)# show run in vlan127`
		`interface vlan127`
		`        ip address 137.1.1.1/16`
		`        ipv6 address 7f00::1/64`
		`        arp ipv4 137.1.1.35 mac 00:12:01:00:00:1a`
		`        arp ipv4 137.1.1.70 mac 00:12:01:00:00:3d`
		`        exit`
		`DUT-1(config-vsx)`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync neighbor`

OSPF configurations.	`vsx-sync ospf`	`switch(config)# router ospf 1`
		`switch(config-ospf-1)# area 0`
		`switch(config-ospf-1)# area 1 nssa`
		`switch(config-ospf-1)# area 2 stub`
		`switch(config-ospf-1)# redistribute connected route-map map1`
		`switch(config)# router ospfv3 1`
		`switch(config-ospfv3-1)# max-metric router-lsa on-startup`
		`switch(config-ospfv3-1)# bfd all-interfaces`
		`switch(config-if)# ip ospf 1 area 0`
		`switch(config-if)# ip ospf hello-interval 33`
		`switch(config-if)# ipv6 ospfv3 1 area 0`
		`switch(config-if)#  ipv6 ospfv3 dead-interval 55`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync ospf`

Table Continued

Feature	Command for enabling	Example

Route map configurations.	`vsx-sync route-map`	`switch(config)#  ip aspath-list list1 seq 10 permit 10`
		`switch(config)# ip community-list expanded com1 seq 10 permit 10`
		`switch(config)# ip extcommunity-list standard ext1 seq 10 permit rt 10:4`
		`switch(config)# ip prefix-list pref1 seq 10 permit any`
		`switch(config)# route-map rm1 permit`
		`switch(config-route-map-rm1-10)# match ip next-hop 1.1.1.1`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync route-map`

QoS Configurations, such as CoS map, DSCP map, and trust policy.	`vsx-sync qos-global`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync qos-global`

sFlow configurations.	`vsx-sync sflow`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync sflow`

sFlow global configurations.	`vsx-sync sflow-global`	`switch(config)# sflow collector 1.1.1.1`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync sflow-global`

SNMP configurations.	`vsx-sync snmp`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync snmp`

SSH configurations.	`vsx-sync ssh`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync ssh`

Static routes.	`vsx-sync static-routes`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync static-routes`

STP configurations.	`vsx-sync stp-global`	`switch(config)# spanning-tree config-name abc`
		`switch(config)# spanning-tree config-revision 1`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync stp-global`

Time-related configurations, including NTP and time zone configurations.	`vsx-sync time`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync time`

Table Continued

Feature	Command for enabling	Example

UDP forwarder configurations.	`vsx-sync upd-forwarder`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync upd-forwarder`

VRF configurations	`vsx-sync vrf`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync vrf`

VSX configurations:
• ISL:
◦ Dead interval.
◦ Hello interval.
◦ Hold time.
◦ Peer detect interval.
• Keepalive:
◦ Dead interval.
◦ Hello interval.
◦ UDP port number.
• The delay time setting for the link-up delay timer.
• The split recovery setting.
• The system MAC address.
	`vsx-sync vsx-global`	`switch(config)# vsx`
		`switch(config-vsx)# inter-switch-link dead-interval 15`
		`switch(config-vsx)# inter-switch-link hello-interval 2`
		`switch(config-vsx)# inter-switch-link hold-time 1`
		`switch(config-vsx)# vsx-sync vsx-global`

VSX LAG interfaces.	`vsx-sync mclag-interfaces`	`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync mclag-interfaces`

VRRP configurations.	`vsx-sync vrrp`	`switch(config)# router vrrp enable`
		`switch(config-if)# vrrp 1 address-family ipv4`
		`switch(config-if-vrrp)# address 1.1.1.100 primary`
		`switch(config-if-vrrp)# timers advertise 1000`
		`switch(config-if-vrrp)# no shutdown`
		`switch(config-if)# vrr 1 address-family ipv6`
		`switch(config)# vsx`
		`switch(config-vsx)# vsx-sync vrrp`

Enabling VSX synchronization at the context level

The commands in this table are for enabling VSX synchronization at the context level, such as for an access list, an interface, or a LAG.

IMPORTANT: When enabling VSX synchronization under a physical interface, a VLAN interface, or a VSX LAG, create on the secondary switch the physical interface, VLAN interface, or VSX LAG with the same name and routing setting as on the primary switch. For example, if the primary switch has a physical interface of 1/1/1, you must create another physical interface of 1/1/1 on the secondary switch. Also, if the primary VSX switch has routing enabled, the secondary switch must have routing enabled. Once the name and routing information is the same, VSX synchronization synchronizes the additional configuration information from the primary VSX switch to the secondary VSX switch.

Feature	Command for enabling	Example

Access lists associated with interface or LAG.	**`vsx-sync access-lists`**	Enabling VSX synchronization for access lists associated with interface 1/1/1:
		`switch(config)# interface 1/1/1`
		`switch(config-if)# vsx-sync access-lists`
		Enabling VSX synchronization for access lists under interface LAG 2:
		`switch(config)# interface lag 2`
		`switch(config-lag-if)# vsx-sync access-lists`
		`switch(config-lag-if)# apply access-list ip test1 in`

An access list context.	**`vsx-sync`**	`switch(config)# access-list ip ITBoston`
		`switch(config-acl-ip)# vsx-sync`

One or more active gateways associated with an interface.	**`vsx-sync active-gateways`**	Enabling VSX sync for active gateways under interface VLAN 5:
		Enter on the primary switch:
		`switch(config)# interface vlan 5`
		`switch(config-if-vlan)# vsx-sync active-gateways`
		Enter on the secondary switch:
		`switch(config)# interface vlan 5`

A class context.	**`vsx-sync`**	`switch(config)# class ip ITHouston`
		`switch(config-class-ip)# vsx-sync`

A policy context.	**`vsx-sync`**	`switch(config)# policy ITPaloAlto`
		`switch(config-policy)# vsx-sync`

Table Continued

Feature	Command for enabling	Example

An IRDP association under an interface enabled for syncing.	`vsx-sync irdp`	`switch(config)# interface 1/1/1`
		`switch(config-if)# ip irdp`
		`switch(config-if)# ip irdp minadvertinterval 550`
		`switch(config-if)# ip irdp maxadvertinterval 850`
		`switch(config-if)# ip irdp holdtime 900`
		`switch(config-if)# vsx-sync irdp`

QoS associated with an interface or LAG.	`vsx-sync qos`	Enabling VSX synchronization for QoS associations under interface 1/1/5:
		`switch(config)# interface 1/1/5`
		`switch(config-if)# vsx-sync qos`
		Enabling VSX synchronization for QoS under interface LAG 3:
		`switch(config)# interface lag 3`
		`switch(config-lag-if)# vsx-sync qos`

A QoS queue-profile.	`vsx-sync`	`switch(config)# qos queue-profile qprofile1`
		`switch(config-queue)# vsx-sync`
		`switch(config-queue)# map queue 0 local-priority 7`
		`switch(config-queue)# map queue 1 local-priority 6`
		`switch(config-queue)# map queue 2 local-priority 5`
		`switch(config-queue)# map queue 3 local-priority 4`
		`switch(config-queue)# map queue 4 local-priority 3`
		`switch(config-queue)# map queue 5 local-priority 2`
		`switch(config-queue)# map queue 6 local-priority 1`
		`switch(config-queue)# map queue 7 local-priority 0`

Table Continued

Feature	Command for enabling	Example

A QoS schedule-profile.	**vsx-sync**	`switch(config)# qos schedule-profile sprofile1`
		`switch(config-schedule)# vsx-sync`
		`switch(config-schedule)# dwrr queue 0 weight 1`
		`switch(config-schedule)# dwrr queue 1 weight 10`
		`switch(config-schedule)# dwrr queue 2 weight 20`
		`switch(config-schedule)# dwrr queue 3 weight 30`
		`switch(config-schedule)# dwrr queue 4 weight 40`
		`switch(config-schedule)# dwrr queue 5 weight 50`
		`switch(config-schedule)# dwrr queue 6 weight 60`
		`switch(config-schedule)# dwrr queue 7 weight 70`

Port filters under an interface.	**vsx-sync portfilter**	`switch(config)# interface 1/1/1`
		`switch(config-if)# vsx-sync portfilter`
		`switch(config)# interface lag 1`
		`switch(config-lag-if)# vsx-sync portfilter`

Policies under an interface.	**vsx-sync policies**	Enabling VSX sync for policies under interface VLAN 5:
		Enter on the primary switch:
		`switch(config)# interface vlan 5`
		`switch(config-if-vlan)# vsx-sync policies`
		Enter on the secondary switch:
		`switch(config)# interface vlan 5`

Rate limits associated with interface or LAG.	**vsx-sync rate-limits**	Enabling VSX synchronization for rate limits with interface 1/1/1:
		`switch(config)# interface 1/1/1`
		`switch(config-if)# vsx-sync rate-limits`
		Enabling VSX synchronization for rate limits under interface LAG 3:
		`switch(config)# interface lag 3`
		`switch(config-lag-if)# vsx-sync rate-limits`

Table Continued

Feature	Command for enabling	Example
VLANs association under an interface enabled for syncing.	**vsx-sync vlans**	switch(config)# **interface 1/1/1** switch(config-if)# **vsx-sync vlans**
VSX active-forwarding for an interface VLAN.	**vsx active-forwarding**	switch# **interface vlan 3** switch(config-if-vlan)# **vsx active-forwarding** switch(config-vsx)#

More information

Benefits of active forwarding and active gateway

`role {primary | secondary}`

`interface lag multi-chassis`

Enabling VSX synchronization of STP configurations between VSX peer switches

Prerequisites

- The VSX switches support several STP modes, such as MSTP and RPVST. Confirm that these STP configurations are identical on the VSX switches.
- You must be in the global configuration context: `switch(config)#`

Procedure

1. Enter:

```
switch(config)# vsx
switch(config-vsx)# vsx-sync stp-global
```

2. Enter:

```
switch(config-vsx)# vsx-sync vsx-global
```

3. Enter:

```
switch(config-vsx)# vsx-sync mclag-interfaces
```

Ways to view the status of VSX

You view the status of VSX by multiple techniques:

- **From the web UI:** See the VSX page topic in the *Introduction to the Web UI Guide*.
- **From the REST API:** See the *REST API Guide*.
- **From the CLI:** See **VSX commands**.

Consistency checking between VSX switches

Use the following commands to verify that all configurations are in-sync between VSX switches. These commands are helpful in troubleshooting configuration mismatches across VSX peer switches.

Task	Command
Displaying the VSX global configuration consistency between two VSX switches. Use this command to troubleshoot configuration mismatches across VSX peer switches.	**`show vsx config-consistency`**
Displaying VSX LACP configuration consistency between two VSX switches. Use this command to troubleshoot configuration mismatches across VSX peer switches.	**show vsx config-consistency lacp [*<LAG-NAME>*]**

Viewing the show commands for both VSX switches from one switch

You can view the outputs of the show command for the primary and secondary VSX switches from one switch. When you enter a show command with the `vsx-peer` parameter, the command displays the output from the peer device.

For example, the following command was entered on the primary switch. The `vsx-peer` parameter indicates to the software to display the output as if the command was entered on the secondary switch.

```
switch# show vsx status vsx-peer
VSX Operational State
---------------------
  ISL channel             : In-Sync
  ISL mgmt channel        : operational
  Config Sync Status      : in-sync
  NAE                     : peer_reachable
  HTTPS Server            : peer_reachable

Attribute           Local               Peer
------------        --------            --------
```

```
ISL link            lag1                lag1
ISL version         2                   2
System MAC          e0:07:1b:cb:72:e4   98:f2:b3:68:79:2e
Platform            8320                8320
Software Version    10.0x.xxxx          10.0x.xxxx
Device Role         secondary           primary
```

NOTE: The show commands that display the file system contents, such as `show logging` or `show core-dump`, do not support the `vsx-peer` parameter.

If the switches lack the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed.

Link-up delay

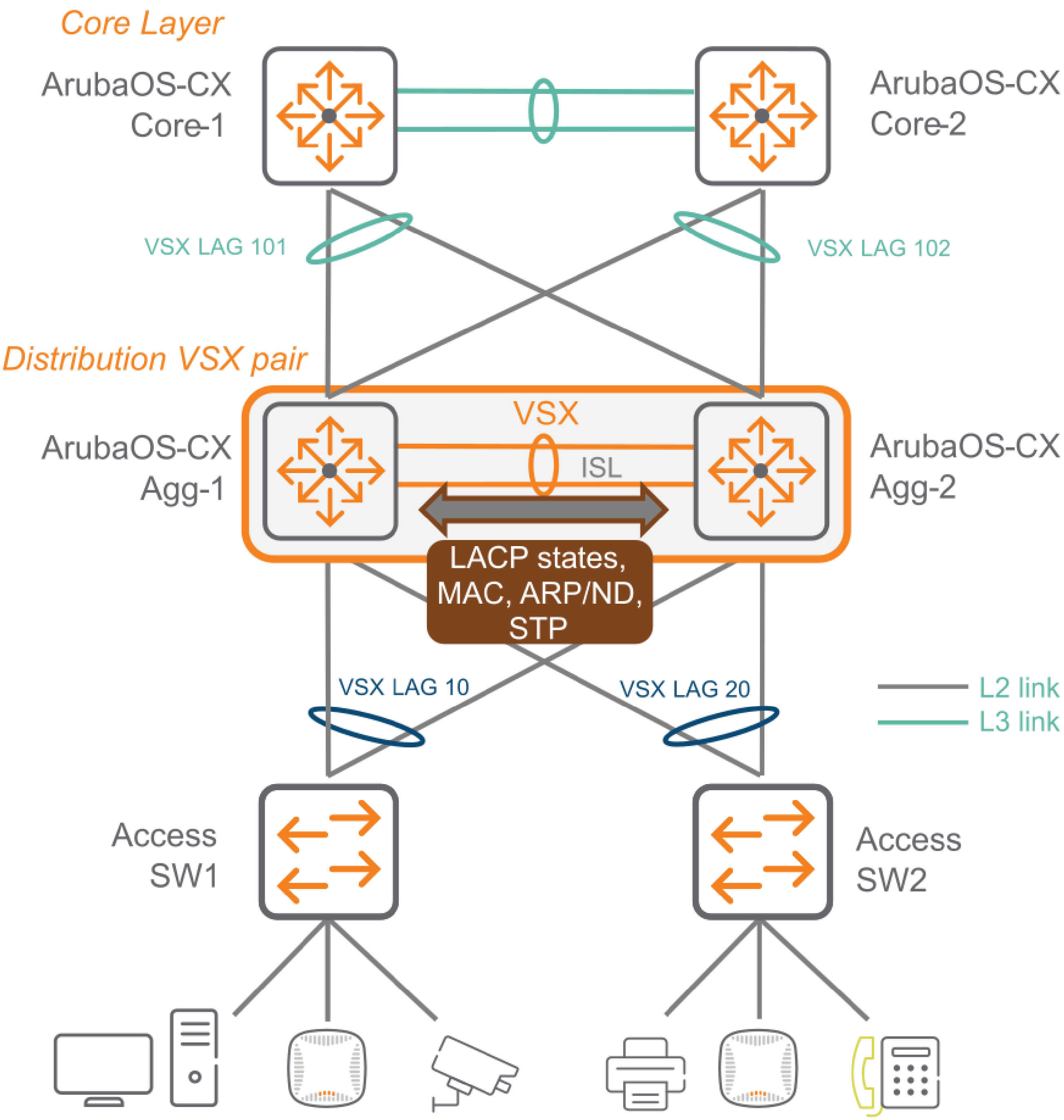
When a VSX device is rebooted, it has no entries for MAC, ARP, routes. If downstream VSX LAG ports are activated before the information is relearned, traffic is dropped. To avoid a traffic drop, VSX LAGs on the rebooted device stay down until the restore of LACP, MAC, ARP, and MSTP databases.

The learning process has two phases:

- Initial synchronization phase:
  - This phase is the download phase where the rebooted node learns all the LACP+MAC+ARP+STP database entries from its VSX peer through ISLP.
  - The initial synchronization timer, which is not configurable, is the required time to download the database information from the peer.
- Link-up delay phase:
  - This phase is the duration for:
    - Installing the downloaded entries to the ASIC.
    - Establishing router adjacencies with core nodes and learning upstream routes.
  - The link-up delay timer default value is 180 seconds.
  - Depending on the network size, ARP/routing tables size, you might be required to set the timer to a higher value (maximum 600 seconds).

When both VSX devices reboot, the link-up delay timer is not used.

To get upstream router adjacencies established during the link-up delay, the upstream LAG (for example LAG 101) has to be excluded from the scope of the link-up delay. Even if the upstream VSX node is not excluded from the link-up delay timer, OSPFv2/OSPFv3 neighborship forms, when active-forwarding is enabled on a VLAN. While the link-up delay timer is running, all SVIs that contain VSX LAG members are kept in a pseudo-shutdown state.



More information

linkup-delay-timer

The link-up delay timer during an ISL failure

Configure the link-up delay timer and exclude LAGs so that if an ISL goes down, the downed ISL does not impact the state of the VLAN, and its SVI that is not a part of a VSX LAG. This SVI is part of at least one orphan port (besides the ISL LAG which is not a VSX LAG).

The following scenario explains what happens:

- **During the ISL going down (before the initial synchronization):** As long as the secondary VSX node has a port that is a member of a VSX LAG, the associated SVI of the VLAN (transported by the VSX LAG) turns to OFF/SHUT on the VSX secondary node. This situation occurs regardless of orphan ports carrying the given VLAN.

- **During the running of the link-up delay timer (after the initial synchronization):**

  As long as the secondary VSX node has a port that is a member of a VSX LAG, the associated SVI of the VLAN (transported by the VSX LAG) turns to OFF/SHUT on the VSX secondary node. This situation occurs regardless of orphan ports carrying the given VLAN.

  The associated SVI of the VLAN transported by VSX LAG restores to ON/UP on the VSX secondary node, only if the following two conditions are met:

  - The VSX LAG is excluded from the link-up delay timer by the following command: `linkup-delay-timer exclude lag-list`
  - The given VLAN is not allowed on a VSX LAG that is not in the part of the exclusion set.

The following example shows how a network was configured so an SVI that was not part of a VSX LAG (SVI 16 in this case) was restored. This example also shows the link-up delay timer and the exclusion of LAGs.

The network in the following example was configured as:

- VLAN 16 is set on 1/1/5 (access).
- VLAN 10 tagged as VSX LAG 11. *
- LAG 1 is not a VSX LAG.*
- LAG 2 and LAG 11 are VSX LAGs.

*This information is not accessible from the following example.

```
switch# show vlan 10

--------------------------------------------------------------------------------------------------------------
VLAN  Name                              Status  Reason                Type      Interfaces
--------------------------------------------------------------------------------------------------------------
10    test_vlan10                       up      ok                    static    1/1/7,lag1-lag2,lag11-lag12,
                                                                                lag14,lag16,lag112
switch# show vlan 15

--------------------------------------------------------------------------------------------------------------
VLAN  Name                              Status  Reason                Type      Interfaces
--------------------------------------------------------------------------------------------------------------
15    ZTP_VLAN                          up      ok                    static    lag1-lag2

switch# show vlan 16

--------------------------------------------------------------------------------------------------------------
VLAN  Name                              Status  Reason                Type      Interfaces
--------------------------------------------------------------------------------------------------------------
16    VLAN16                            up      ok                    static    1/1/5,lag1

switch# show vlan 200

--------------------------------------------------------------------------------------------------------------
VLAN  Name                              Status  Reason                Type      Interfaces
--------------------------------------------------------------------------------------------------------------
200   interco_vlan                      up      ok                    static    lag1-lag2

switch# show run vsx
vsx
    system-mac 00:00:00:01:01:01
    inter-switch-link lag 1
    role secondary
    keepalive peer 192.168.10.1 source 192.168.10.2 vrf KeepAlive
    linkup-delay-timer exclude lag 2
    linkup-delay-timer 60


switch# show vsx status linkup-delay
Configured linkup delay-timer                                : 60 seconds
Initial sync status                                          : Completed
Delay timer status                                           : Running
```

```
Linkup Delay time left                                      : 0 minutes 58 seconds
Interfaces that will be brought up after delay timer expires : lag11-lag12,lag14,lag16,lag112
Interfaces that are excluded from delay timer               : lag2


switch# show int vlan10

Interface vlan10 is down
 Admin state is up
 Description:
 Hardware: Ethernet, MAC Address: 94:f1:28:1d:ad:00
 IPv4 address 10.10.10.3/26
 active gateway 10.10.10.1 00:00:00:00:11:01
 active gateway 2002:0a0a:0a00::1 00:00:00:00:66:01
switch# sh int vlan15

Interface vlan15 is up
 Admin state is up
 Description:
 Hardware: Ethernet, MAC Address: 94:f1:28:1d:ad:00
 IPv4 address 10.10.15.12/24
switch# sh int vlan16

Interface vlan16 is up
 Admin state is up
 Description:
 Hardware: Ethernet, MAC Address: 94:f1:28:1d:ad:00
 IPv4 address 10.10.16.2/24
switch# sh int vlan200

Interface vlan200 is up
 Admin state is up
 Description:
 Hardware: Ethernet, MAC Address: 94:f1:28:1d:ad:00
 IPv4 address 10.10.212.6/29
```

As expected, SVI 10 is in pseudo-shut during the link-up delay. SVI 10 was a part of LAG 2 which is in exclusion. SVI 16 is up, as expected because SVI 16 was not part of a VSX LAG.

More information

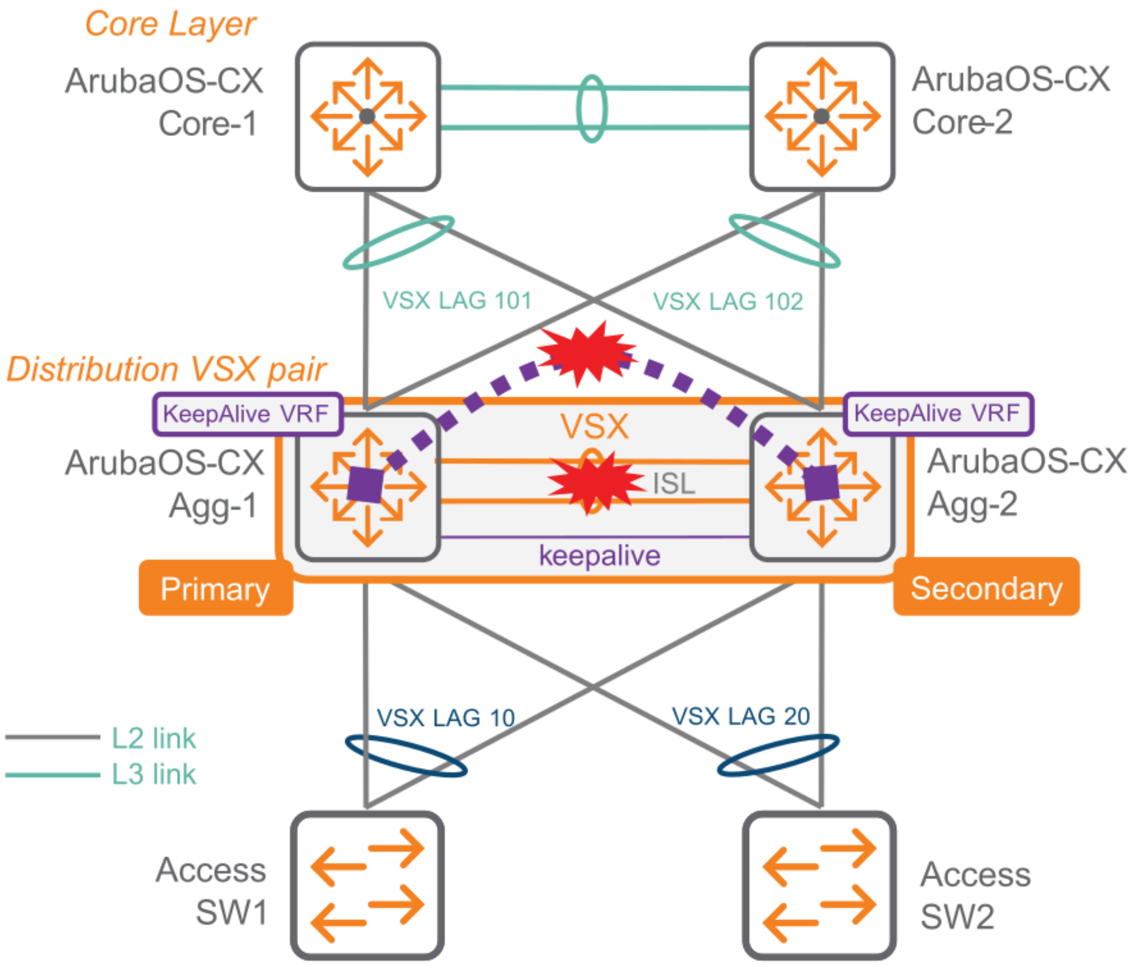
`linkup-delay-timer`

`linkup-delay-timer exclude lag-list`

Split brain scenario

A split brain scenario occurs when both keepalive and the ISL is down, as shown in the follow figure. When the ISL is restored, there is no reboot of the secondary switch. If split recovery is enabled (the default setting), the secondary VSX LAGs are brought up after the time set by the `linkup-delay-timer` command.

Figure 4: *Split brain scenario*



More information

Failure scenarios and split recovery

Keepalive

Keepalive is a layer 3 interface that is used to exchange heartbeats between VSX peer switches. The heartbeats are exchanged by using the User Datagram Protocol (UDP) and port 7678 (default). During an ISL failure, VSX switches use their keepalive connection to determine if both VSX switches are up and running. This configuration helps the VSX switches find alternative paths to the ISL link in the network so the two VSX switch can continue to stay in-sync.

Configure each VSX peer switch with a keepalive connection to the other VSX peer switch. This connection is established over a routed network (IPv4 currently) and is not required to be a dedicated peer-to-peer link unlike ISL. Keepalive packets are UDP-based.

Make sure that the VSX peer switches have layer 3 reachability for keepalive interfaces through directly connected interfaces or routed through the upstream layer 3 network. Source of keepalive interfaces can be a layer 3 interface (router port), a loopback interface, or a Switch Virtual Interface (SVI). An SVI is a logical layer 3 interface configured per VLAN (one-to-one mapping) that performs all layer 3 processing for packets to or from all switch ports associated with that VLAN.

NOTE: With respect to the keepalive path, it is highly recommended to separate keepalive traffic from the ISL link.

Use a dedicated layer 3 link and as a best practice, also use a dedicated VRF, as shown in **Recommended network configuration for keepalive**.

Keepalive packets can be sourced from the supported layer 3 interface; however, the packet must not be transported over the ISL.

In the case of 6400 and 8400 switch series, it highly recommended to use keepalive and ISL on different line cards. A single point of failure on line card that has keepalive and ISL configuration might cause split brain.

Keepalive response in ISL failure scenarios

- **ISL link is down but the switches are still up and running:** In this case, VSX switches use their keepalive connection to determine that they are both up and running. Once that is determined, the user-configured primary VSX switch keeps its multichassis (VSX) LAG links up and the secondary VSX switch forces its VSX LAG links to go down with the appropriate reason. Once the ISL link is up, the MAC and ARP tables of the primary switch are synchronized to the secondary switch. Then, the configured delay timer starts. Once the delay timer expires, the secondary VSX switch brings up its VSX LAG links.
- **ISL link and one of the VSX switches is down:** The running switch sees that the ISL and keepalive connection are both down. Independent of the user configured role (primary or secondary), the switch that is up continues to keep its VSX LAG links up. Subsequently when the peer switch returns, the ISL link comes up first. Then, the returned VSX peer switch synchronizes its MAC and ARP tables from the peer switch that stayed up. After the synchronization completes, the delay time starts. Once the delay timer expires, the VSX peer switch brings up its VSX LAG links.

More information

VSX switch reboot

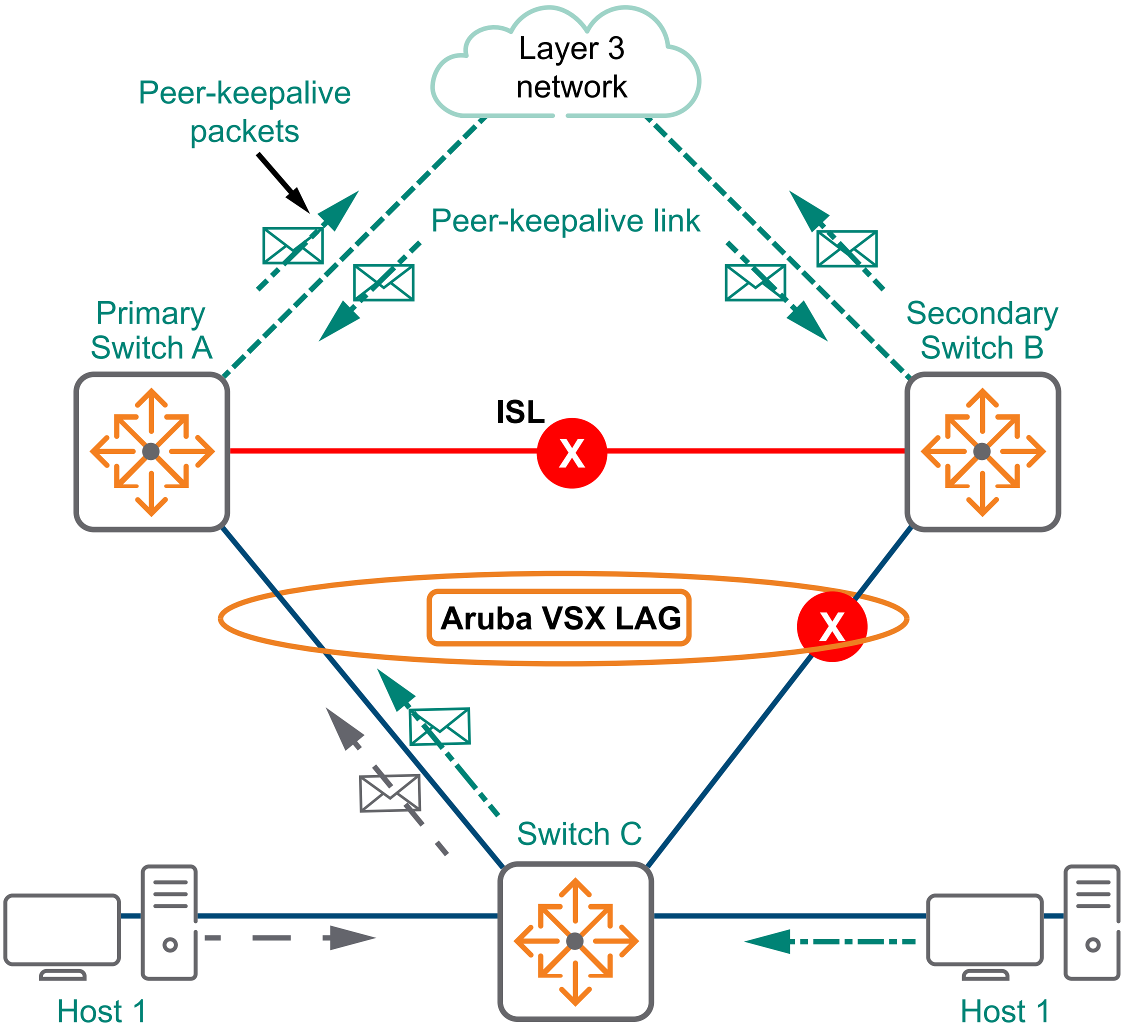
`linkup-delay-timer`

`linkup-delay-timer exclude lag-list`

Keepalive scenario

The following diagram illustrates a scenario in which both VSX switches are up, but the ISL link is down. The switches cannot exchange information.

The keepalive functionality brings down the link between Switch B and Switch C in the following diagram. The traffic is forced to go from Switch C to Switch A and then through the Layer 3 network to access Switch B. The keepalive path is over the Layer 3 network. Traffic traveling from Switch B to Switch A is also forced to go through the Layer 3 network.



NOTE: Do not have the keepalive path go over ISL. Use a direct-link connection for keepalive. If the keepalive path uses ISL as its only path and an ISL link failure occurs, the VSX switches would be out of sync without the keepalive functioning.

Keepalive configurations

Task	Command	Example
Configuring keepalive peer source and VRF.	`keepalive peer <IP-ADDR> source <IP-ADDR> [<VRF-NAME>]`	`switch(config-vsx)# keepalive peer 192.168.1.1 source 192.168.1.5 vrf vrf1`
Unconfiguring keepalive.	`no keepalive`	`switch(config-vsx)# no keepalive`
Configuring keepalive UDP port.	`keepalive udp-port`	`switch(config-vsx)# keepalive udp-port 2000`

Table Continued

Task	Command	Example
Restoring default keepalive UDP port.	**no keepalive udp-port**	`switch(config-vsx)# no keepalive udp-port`
Configuring keepalive hello interval.	**keepalive hello-interval**	`switch(config-vsx)# keepalive hello-interval 3`
Restoring default keepalive hello interval.	**no keepalive hello-interval**	`switch(config-vsx)# no keepalive hello-interval`
Configuring keepalive dead interval.	**keepalive dead-interval**	`switch(config-vsx)# keepalive dead-interval 10`
Restoring default keepalive dead interval.	**no keepalive dead-interval**	`switch(config-vsx)# no keepalive dead-interval`

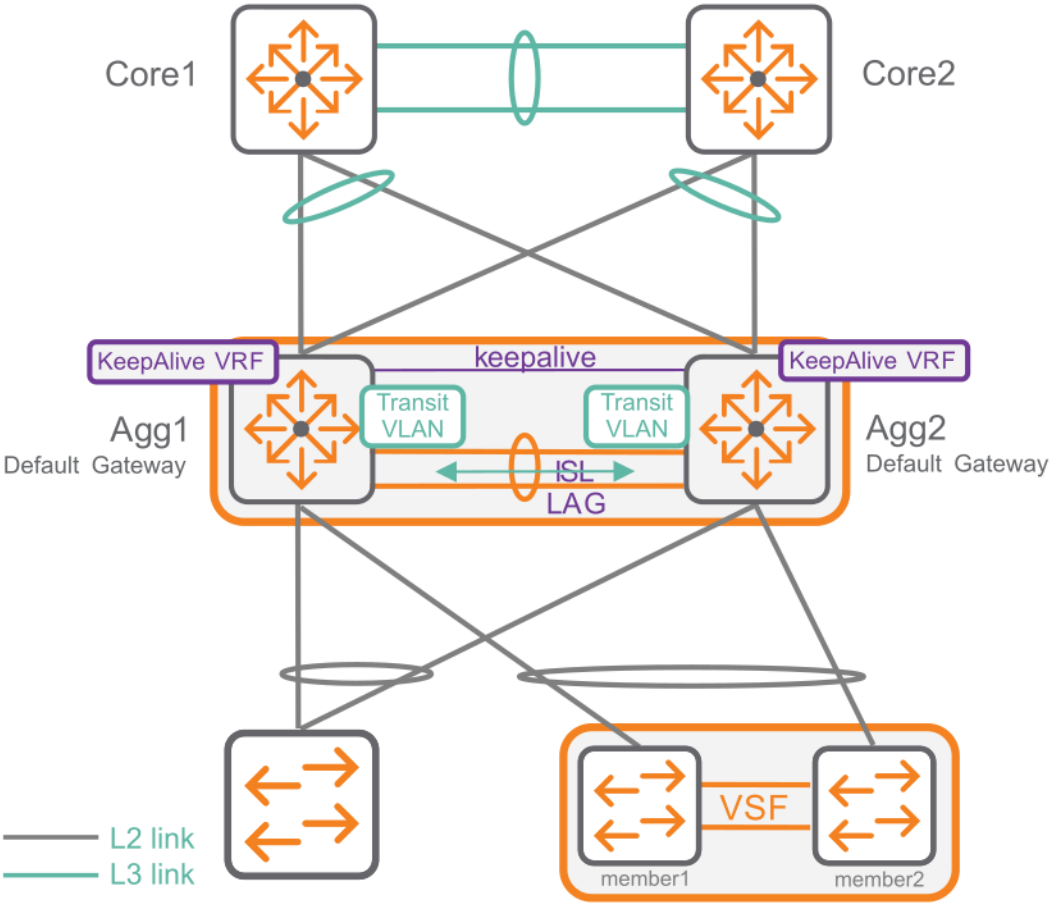
Default values:

- Keepalive dead interval: 3 seconds
- Hello interval: 1 second
- UDP port for the keepalive protocol: 7678

Recommended network configuration for keepalive

Directly connect the keepalive link, as shown in the following figure. Avoid keepalive communication over the ISL circuits.

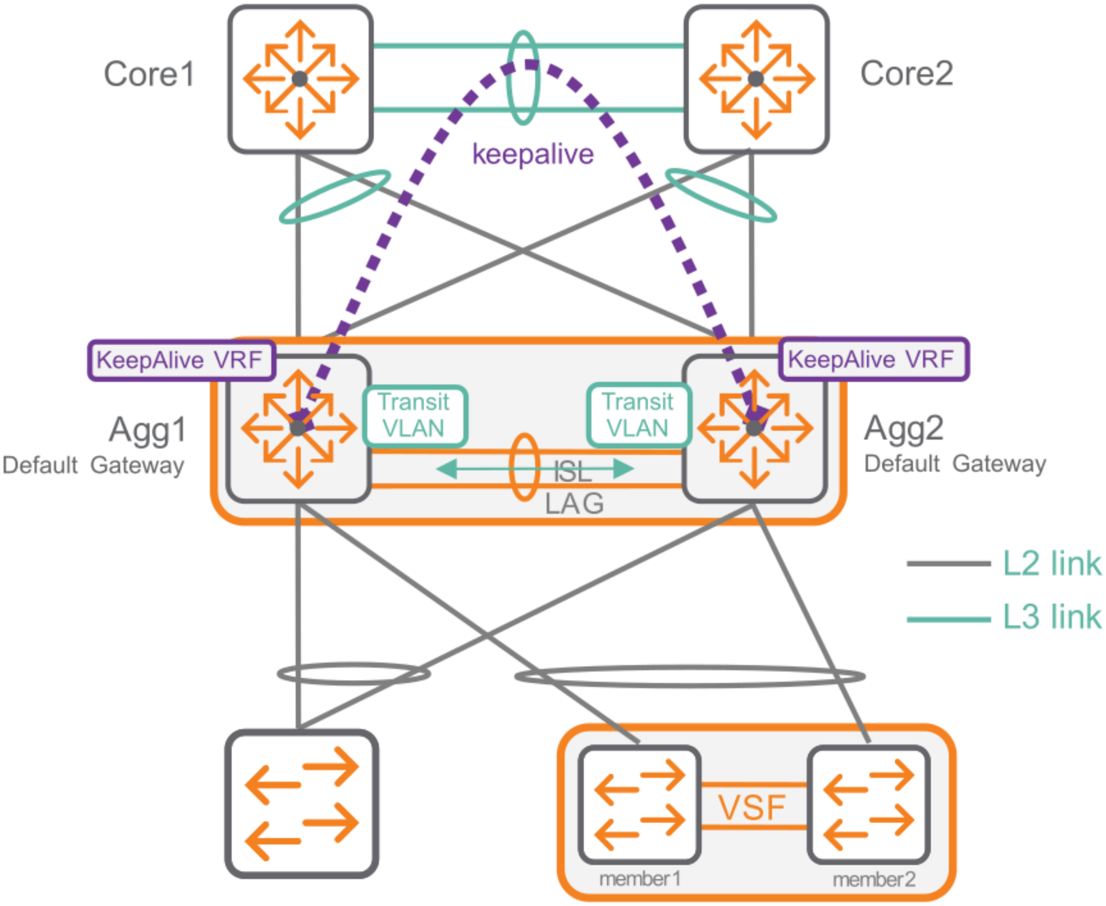
Figure 5: *Recommended configuration for keepalive*



Do not configure keepalive to go through the VSX LAG uplinks, as shown in the following image. This scenario is not supported because:

- VSX LAG on the secondary will clear because split detection.
- Keepalive communication will stop between the VSX switches.

Figure 6: *Not supported configuration for keepalive*



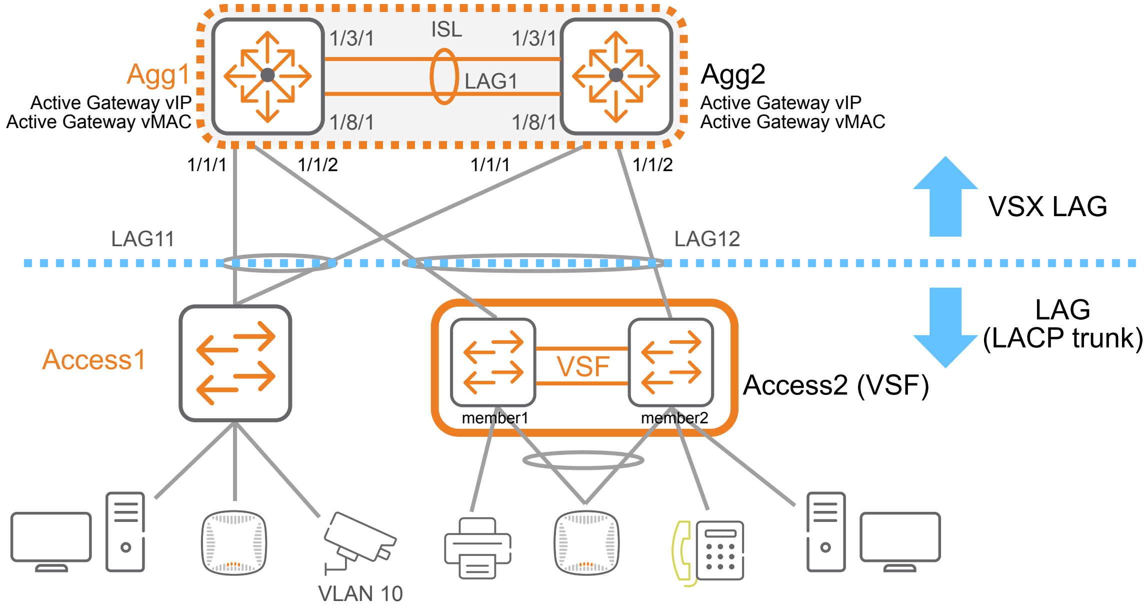
Active gateway and active forwarding

Active-active layer 2

VSX LAGs span two switches and operate in active-active mode. Traffic between the access layer and aggregation layer switches can be forwarded to any of the active links. There are no loops and no need for spanning tree protocol or blocked ports.

From a datapath perspective, each VSX switch that gets a packet always uses its local links of the LAG to forward traffic to the destination. The VSX switch only uses the ISL link if the local LAG links are down.

Figure 7: *Layer 2 configuration*



The following shows the configuration details from the figure:

```
interface lag 11 multi-chassis
    description access-sw1
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 5,10,15,20
    lacp mode active

interface lag 12 multi-chassis
    description access-sw2
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 5,10,15,20
    lacp mode active

interface 1/1/1
    no shutdown
    lag 11
interface 1/1/2
    no shutdown
    lag 12
```

Active-active layer 3 default gateway

VSX aggregation switches can be configured with a shared virtual IP (VIP) and a shared virtual MAC address (VMAC) on the layer 3 VLAN interface.

The VIP/VMAC serves as the default gateway for the access layer. The two switches then share the router MAC and function as an active-active gateway for the IP subnet. The first VSX device that receives traffic from the access layer (based on the hashing algorithm over the LAG interface) routes it across to the other subnets.

Active gateway over VSX

Active gateway is a first hop redundancy protocol that eliminates a single point of failure. The active gateway feature is used to increase the availability of the default gateway servicing hosts on the same subnet. An active gateway improves the reliability and performance of the host network by enabling a virtual router to act as the default gateway for that network.

If you have enabled active gateway, VRRP is not required. Active gateway is similar to VRRP in that routed traffic from the VSX node is sourced from the switch interface MAC and not the virtual MAC address (VMAC). Each active gateway sends a periodic broadcast hello packet to avoid VMAC aging on the access switches. The switch views the active gateway IP as a self IP address.

Active gateway is preferable over VRRP because with VRRP traffic is still pushed over the ISL link, resulting in latency in the network.

VMACs and active gateway

There can be only one virtual MAC address (VMAC) each for IPv4 and IP6, and the VIP and VMAC must be the same on both VSX switches.

You can have a maximum of 16 different VMACs per VSX pair. You can configure the same VMAC for both IPv4 and IPv6. For example: You can have a maximum of eight VMACs for IPv4, simultaneously having a maximum of eight VMACs for IPv6.

NOTE: Only 15 VMACs are supported on 6400 switch series.

If a VMAC is different for IPv4 and IPv6, the switch creates two different interfaces, one for IPv4 and another for IPv6:

```
interface vlan2
active-gateway ip mac 0a:0b:0c:0d:0e:0e
active-gateway ipv6 mac 00:00:00:00:00:01

0020a0b0c0d0e0eLink encap:Ethernet HWaddr 0A:0B:0C:0D:0E:0E

002000000000001Link encap:Ethernet HWaddr 00:00:00:00:00:01
```

If a VMAC is the same for IPv4 and IPv6, only one kernel interface is created for both IPv4 and IPv6:

```
interface vlan3
active-gateway ip mac 00:00:00:00:00:01
active-gateway ipv6 mac 00:00:00:00:00:01

003000000000001Link encap:Ethernet HWaddr 00:00:00:00:00:01
```

NOTE: Do not use peer system MAC address as an active-gateway VMAC. If same MAC address is used, the VSX synchronization will try to sync the configuration on secondary switch and cause traffic disruptions.

Requirements

- Before configuring active gateway, confirm that an IP address is on the SVI that is in the same subnet as the active gateway IP you are trying to configure. If an active gateway IP does not have an SVI IP with the same subnet, the CLI allows the configuration, but the active gateway IP will not be programmed in the kernel, resulting the active gateway to be unreachable.
- An active gateway can be configured only over an SVI. If active gateway and SVI IP addresses are the same, make sure that SVI IP addresses are consistent across VSX switches. If you have a VSX square

topology that contains two pairs of VSX switches, make sure that you do not have the same IP address across all four VSX nodes in the square topology.

- Active gateway configuration must be the same in both the VSX peer switches.
- Having same VMAC and different active gateway IP addresses on different VSX segments in a square topology is not supported. Ensure that you have either same VMAC and same active gateway IP addresses or different VMAC and different active gateway IP addresses configured on two different VSX segments. For 8320 and 8325 switch series, when VMAC and active gateway IP addresses are same, make sure that the SVI status is identical on both the VSX segments.
- If a system has active forwarding enabled, reduce one VMAC from the total number of VMACs supported in the system. An active gateway can have a maximum of 14 "unique" MAC addresses per system, both IPv4 and IPv6 addresses are included in the count.
- If a system has active forwarding disabled, an active gateway can have a maximum of 16 "unique" MAC addresses per system, both IPv4 and IPv6 addresses are included in the count.
- With IP multinetting, a maximum of 32 IPv4 active gateway and a maximum of 31 IPv6 active gateway can be configured. A recommended configuration is a multidimension scale (MD) scale and a maximum network limit, along with four IPv4 active gateways and four IPv6 active gateways per SVIs with a maximum of 512 SVIs per chassis.

  An MD scale is when the VSX active-gateway along with other supported features, such as layer 2, layer 3, and multi-VRF are enabled and the system response/stability is validated against them.
- Link local IPv6 virtual IP address of an active gateway address is multicasted for router advertisement so that the IPv6 address can be chosen as a default gateway.
- Active gateway configuration must be the same in both the VSX peer switches.
- Disable IP ICMP redirect when IP multinetting is enabled.
- Disable ICMP redirect when routing is enabled through an active gateway SVI where egress port belongs to same VLAN as ingress.

Example of IPv4 and IPv6 active gateways on an SVI

Assume that you have IPv4 and IPv6 active gateways on an SVI. Each SVI uses a MAC address for IPv4 and one for IPv6. The configuration of the VSX with an active-gateway consumes a second MAC address per SVI.

```
switch# sh int vlan10

Interface vlan10 is up
Admin state is up
Description: ACCESS switch mgmt
Hardware: Ethernet, MAC Address: 98:f2:b3:68:71:fe
IPv4 address 10.1.1.253/24
Rx
       L3:
            0 packets, 0 bytes
Tx
       L3:
            0 packets, 0 bytes

switch# sh run int vlan141
interface vlan141
   description USER VLAN 10.141.0.0/16
   ip address 10.141.255.253/16
   ip ospf 1 area 0.0.0.0
   ip pim-sparse enable
    ip igmp enable
    ip igmp version 2
   exit
switch# config
switch(config)# int vlan10
switch(config-if-vlan)# active-gateway ip 10.1.1.254 mac 00:00:00:10:11:12
switch# sh int vlan10

Interface vlan10 is up
Admin state is up
Description: ACCESS switch mgmt
Hardware: Ethernet, MAC Address: 98:f2:b3:68:71:fe
IPv4 address 10.1.1.253/24
active gateway 10.1.1.254           00:00:00:10:11:12
Rx
       L3:
            0 packets, 0 bytes
Tx
       L3:
            0 packets, 0 bytes
```

IP multinetting over VSX

IP multinetting is the assignment of more than one IP interface to a single VLAN that is used to enable a router to provide default gateway service to different address ranges associated with a single VLAN.

When using IP multinetting in an environment with VSX enabled, you must configure multiple active gateway IP addresses per SVI so that you can reach multiple networks on the same VLAN. Make sure that you configure an IP address for either the primary or secondary VSX switch on the SVI with the same subnet.

The maximum number of supported active gateways per switch is 4,000. Since a maximum of 31 secondary IPv4 addresses can be configured on an SVI, 32 IPv4 active gateways (along with the primary IPv4 address) can be configured per SVI with IP multinetting support. This support is also the same for IPv6 addresses.

IMPORTANT: Disable IP ICMP redirect when IP multinetting is enabled.

Multiple active gateways IP addresses can be programmed on the same active gateway kernel interface, as shown in the following example:

```
interface vlan3
ip address 10.0.0.1/24
ip address 20.0.0.1/24 secondary
active-gateway ip mac 00:00:00:00:00:01
active-gateway ip 10.0.0.3
active-gateway ip 20.0.0.3

003000000000001@vlan3: <NO-CARRIER,BROADCAST,UP,M-DOWN> mtu 1500 qdisc noqueue
state LOWERLAYERDOWN group default qlen 1000
link/ether 00:00:00:00:00:01 brd ff:ff:ff:ff:ff:ff
inet 10.0.0.3/32 scope global 003000000000001
valid_lft forever preferred_lft forever
inet 20.0.0.3/32 scope global 003000000000001
valid_lft forever preferred_lft forever
```

Active gateway VMAC and VIPs can be configured separately:

```
interface vlan3
ip address 10.0.0.1/24
ip address 20.0.0.1/24 secondary
active-gateway ip mac 00:00:00:00:00:01
active-gateway ip 10.0.0.3
active-gateway ip 20.0.0.3
```

Active gateway configurations

Task	Command	Example
Configuring a virtual IPv4 and IPv6 address for an interface VLAN.	active-gateway {ip | ipv6} [*<IP-ADDRESS>*] [mac *<MAC-ADDRESS>*]	switch(config)# **vlan 2** switch(config)# **interface vlan 2** switch(config-if-vlan)# **ip address 10.0.0.1/24** switch(config-if-vlan)# **active-gateway ip 10.0.0.2 mac 00:00:00:00:00:01** switch(config-if-vlan)# **ipv6 address aa:bb::cc:dd/24** switch(config-if-vlan)# **active-gateway ipv6 2001:DB8::/32 mac 00:00:00:01:00:01**
Unconfiguring active gateway for active-active routing.	no active-gateway {ip | ipv6} [*<IP-ADDRESS>*] [mac]	switch(config-if-vlan)# **no active-gateway ip**

See **IP multinetting over VSX** for additional examples of IP multinetting.

VRRP with VSX configuration

VRRP is similar to active gateway in that it is a first hop redundancy protocol that eliminates a single point of failure. One VSX switch acts as a VRRP master and the other switch acts as the VRRP backup. Both VSX switches route the traffic. The active gateway/VRRP configuration must be consistent across the two VSX switches.

Although active gateway and VRRP are no longer globally exclusive in a VSX configuration, active gateway and VRRP are still exclusive on an SVI. A workaround is to configure VRRP on one SVI (SVI A), and configure active-gateway on the other SVI (SVI B).

NOTE: Active gateway is preferable to VRRP because VRRP traffic is still pushed over the ISL link, resulting in latency.

Sample VRRP configuration

IPV4:

```
switch(config)# vlan 1-10
switch(config)# router vrrp enable
switch(config)# interface vlan2
switch(config-if-vlan)# ip address 192.168.1.253/16
switch(config-if-vlan)# no shutdown
switch(config-if-vlan)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# address 192.168.1.253 primary
switch(config-if-vrrp)# no shutdown
switch(config-if-vrrp)# exit
switch(config-if-vlan)# exit
switch(config)#
```

IPV4 and IPV6:

```
switch(config)# vlan 1-10
switch(config)# router vrrp enable
switch(config)# interface vlan3
switch(config-if-vlan)# ip address 172.3.0.1/16
switch(config-if-vlan)# ipv6 address 2002:3::1/64
switch(config-if-vlan)# ip ospf 1 area 0.0.0.0
switch(config-if-vlan)# ipv6 ospfv3 1 area 0.0.0.0
switch(config-if-vlan)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# address 172.3.0.10 primary
switch(config-if-vrrp)# no shutdown
switch(config-if-vrrp)# exit
switch(config-if-vlan)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# address fe80::3 primary
switch(config-if-vrrp)# no shutdown
switch(config-if-vrrp)# exit
switch(config)#
```

Active forwarding

Active forwarding is an optimization for layer 3 unicast traffic flowing from the upstream (core) to the downstream (access) through the VSX peers (aggregate). Active forwarding prevents the bridged traffic from switching over the ISL. It also minimizes latency and the ISL bandwidth.

Active forwarding requirements

- Active forwarding is enabled on a SVI facing core network on a VSX environment.
- Active forwarding is supported on SVI only.
- Active forwarding and active gateway are mutually exclusive features. You cannot enable both active forwarding and active gateway on the same SVI.
- Although the CLI itself does not limit the number of active forwarding SVIs; the maximum number of configured active forwarding SVIs is 256.
- Active forwarding is supported on more than one SVI per VRF.
- Active forwarding cannot be configured when ICMP redirect is enabled.

Traffic flow scenario

Active forwarding mitigates the suboptimal path scenarios because of undeterministic layer 3 hashing and layer 2 hashing, as described in the following ECMP (equal-cost multi-path routing) scenario.

This scenario describes a situation when active forwarding is not used. In a VSX environment, a core network is connected to a VSX pair, forming an OSPF adjacency over a VSX LAG. The VSX LAG has ECMP routes to the access network. The core has ECMP routes to choose between either the VSX primary switch or the VSX secondary switch for traffic flowing from the core to the access network. Assume that ECMP picked the VSX primary switch. This traffic is now subjected to the hashing algorithm over the VSX LAG interface. Based on the chosen hashing algorithm, the layer 2 interface might route the traffic to the VSX secondary switch. The secondary VSX switch then bridges this traffic over the ISL to the primary VSX switch. The primary VSX switch in turn routes the traffic toward the access network, which causes extra overhead with ISL bandwidth and network latency.

If active forwarding was enabled in the previous scenario, the traffic destined for the access network would not be bridged over the ISL. The traffic would flow from north to south instead, resulting in less network latency. For more information about the benefits of active forwarding, along with a diagram, see **Benefits of active forwarding and active gateway**.

Sample Active forwarding configuration

```
Primary# configure terminal
Primary(config)# no ip icmp redirect
Primary(config)# interface vlan 1000
Primary(config-if-vlan)# vsx active-forwarding
Primary(config-if-vlan)# end
```

Deployment options for upstream connectivity with active-active forwarding

Aggregate core links can be configured in one of the following ways:

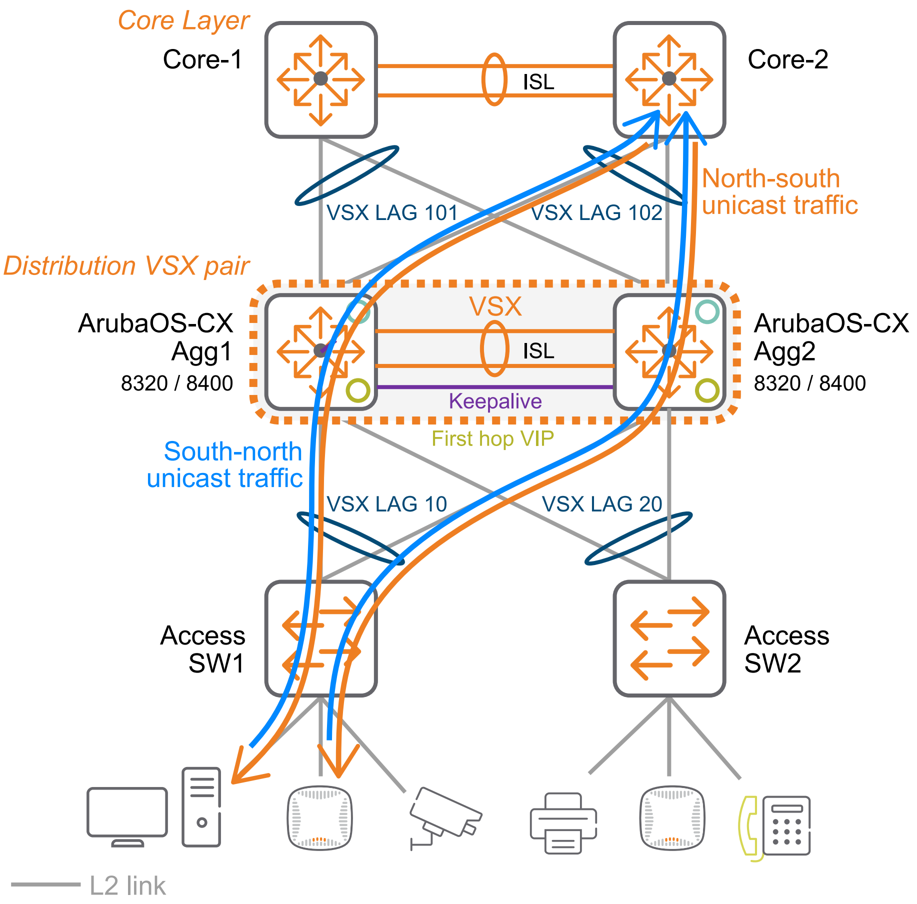
- **Layer 3-LAG/routed ports:** Simple VLAN-free configuration best suited when the network runs on a single VRF domain. With multiple VRFs in the network, one would need multiple routed ports, one per VRF.
- **P2P SVI links:** Each aggregate-core link is on its own VLAN. The layer 2 links can carry traffic for multiple SVIs and therefore multiple VRFs can be carried over the same link.
- **VSX multichassis LAGs:** The aggregate-core links can be multichassis layer 2 links carrying traffic for multiple SVIs and VRFs. This configuration provides for layer 2 LAG and layer 3 ECMP-based active-active forwarding for traffic from core to access.

In these configurations, the two VSX switches run as independent control planes (OSPF/BGP) and present themselves as different routers in the network. In the datapath however, they function as a single router and support active-active forwarding.

Benefits of active forwarding and active gateway

The enabling of active forwarding and active gateway reduces latency in the network by bypassing the ISL link for north-south and south-north traffic, resulting in one less hop.

When active forwarding is enabled, the north-south unicast traffic bypasses the ISL link for Agg1 and Agg2. Just as the south-north traffic bypasses the ISL link for Agg1 and Agg2 when active gateway is enabled, as shown in the following figure.



Virtual active gateway

A virtual active gateway is created by configuring the same IPv4 address in both the interface-VLAN context and the active-gateway context.

Virtual active gateway enables the user to configure the same IPv4 address as both the interface VLAN (SVIs) address as well as the active gateway address. A virtual active gateway is useful when the primary purpose of the SVI is to provide just a first hop gateway service to its clients and does not need a separate set of IPv4 addresses on each device and a virtual IPv4 address to serve the gateway functionality.

Supported services on a virtual active gateway SVI

- DHCP Relay
- ARP
- VRF
- ACLs
- Dual stack (IPv6 requires active gateway to have different real and virtual IP addresses).
- IPv6 Active GW (with real SVI IPv6)
- VSX Active-GW multinet for IP

Unsupported services for a virtual active gateway SVI

- Layer 3 IP Services, such as OSPF, BGP, IGMP, and BFD.
- DHCP Option 82
- PING from the VSX device to downstream clients with SRC IP as the active gateway IP.
- IPv6 virtual active gateway

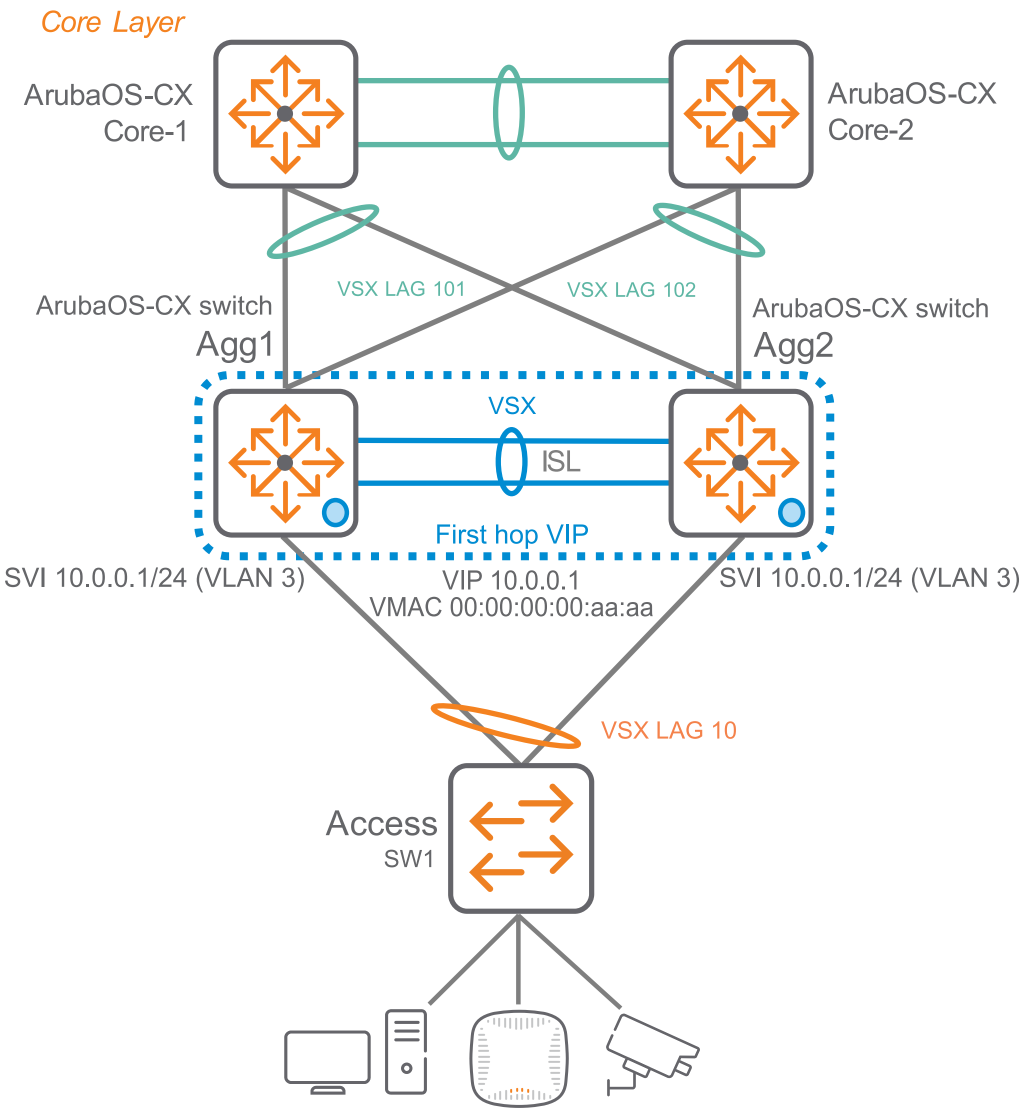
Sample virtual active gateway configuration

A virtual active gateway configuration is created in this example and shown in the following figure.

```
switch(config)# vlan 3
switch(config-vlan-3)# interface vlan 3
switch(config-if-vlan)# ip address 10.0.0.2/24
switch(config-if-vlan)# active-gateway ip 10.0.0.3 mac 00:00:00:00:aa:aa
```

See the *Command-Line Interface Guide* for your switch and software version for more information about the CLI commands.

Figure 8: *Sample virtual active gateway configuration*



More information

active-gateway

Active-standby DHCP relay

When VSX synchronization is enabled for DHCP relay, only the primary VSX node relays DHCP requests to the upstream DHCP server. The secondary VSX node forwards over the ISL to the primary VSX switch the DHCP requests received from the downstream endpoints. The secondary VSX switch takes over DHCP-relay service upon primary failure detection (ISL down and keepalive down). Upstream DHCP servers receive a single DHCP request. Downstream clients receive a single DHCP offer.

More information

vsx-sync dhcp-relay

DHCP relay failure if the SVI is down on the primary switch

Only the primary VSX node relays DHCP requests to the upstream DHCP server. Shutting down the associated SVI on the primary VSX node prevents any DHCP requests to be relayed.

More information

`vsx-sync dhcp-relay`

Split recovery mode

Split recovery mode prevents traffic loss when the ISL goes out-of-sync and keepalive subsequently fails. When the ISL goes out-of-sync and keepalive is established, the secondary VSX LAGs are brought down. If keepalive then also fails, this situation causes a split condition. In this case, if split recovery mode is enabled, the secondary switch restores its VSX LAGs so they are up. The secondary VSX node brings up the VSX LAGs after 10 keepalive packets are missed, approximately 10 seconds after keepalive goes down.

The `no split recovery` command disables split recovery mode. When split recovery mode is disabled during a split condition, the secondary switch keeps it VSX LAGs down.

More information

Failure scenarios and split recovery
`split recovery`
Split brain scenario

IGMP snooping

VSX switches can be configured for IGMP snooping on downstream VLANs facing the access switches. When enabled, the IGMP group database is independently constructed on each VSX switch. Multicast traffic to these groups is appropriately pruned/optimized.

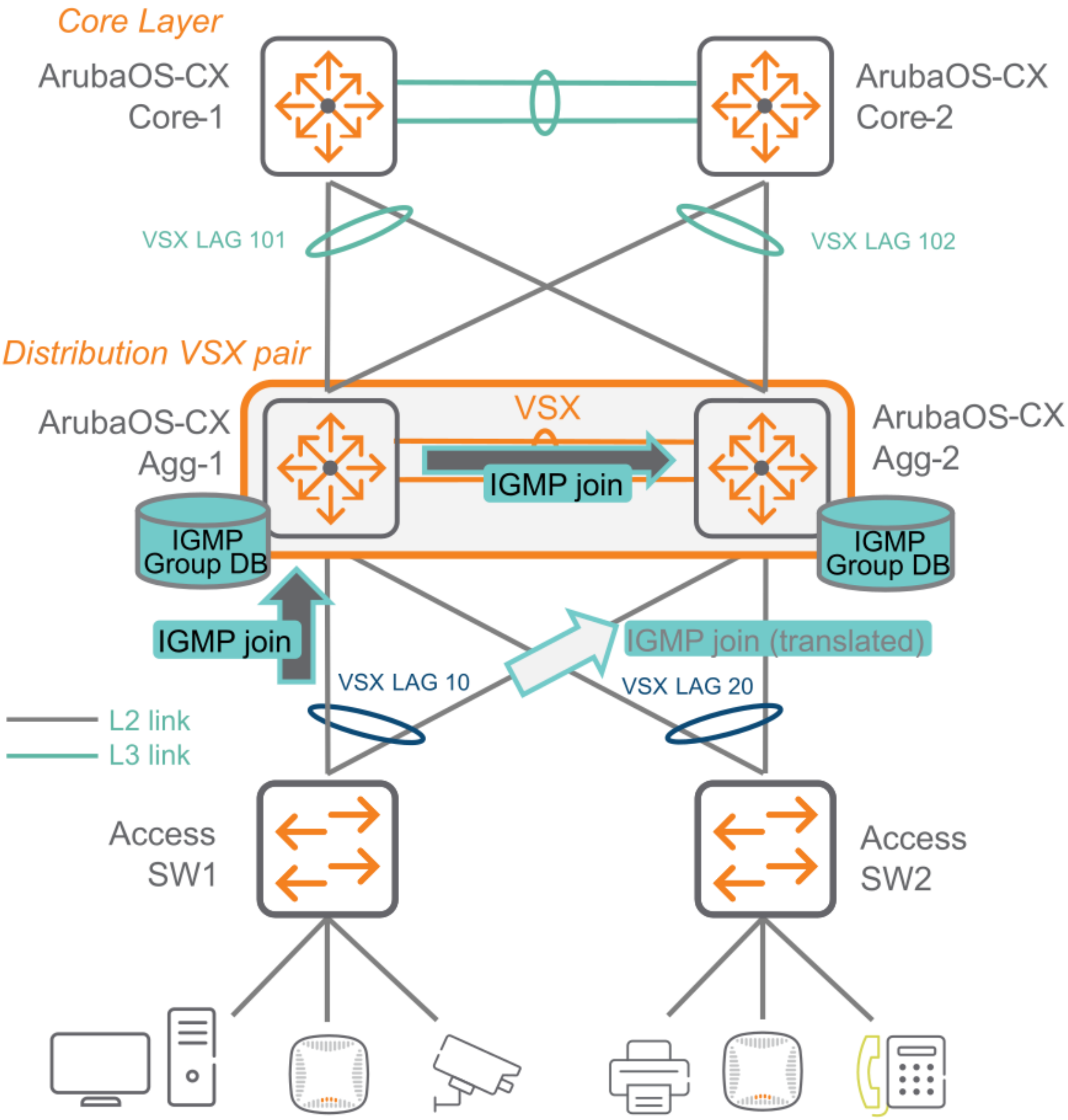
Each VSX switch has an identical IGMP group database:

- Each VSX node individually learns any JOIN/LEAVE message received from a downstream VSX LAG.

  For example: Agg-1 learns on downlink from SW1, whereas Agg-2 learns on the ISL as the ISL is always included as a forwarding port for IGMP, as shown in the following figure.
- The VSX IGMP process translates the received IGMP from the ISL into an IGMP join message from the VSX LAG.

Multicast traffic to these IGMP groups is pruned/forwarded based on the individual IGMP group database on each VSX node. ISLP does not synchronize IGMP groups between VSX peers. The IGMP database construction is a data-plane based process.

If a VSX node reboots, it must relearn all the IGMP groups. The VSX switch floods multicast traffic within the VLANs that have active physical ports being forwarded. It then sends an All Hosts Query message. When the VSX node receives all join messages, it relearns and recreates the IGMP groups database.



DHCP relay backup

When the two VSX switches are configured for DHCP relay on their VLAN interfaces, only the primary switch actively relays DHCP client requests to the upstream server. The secondary switch acts as a backup. If the primary VSX switch goes down, the secondary switch takes over, such in the case with ISL and keepalive both down. Even though both primary and secondary switches receive the DHCP request, the primary switch takes precedence.

The secondary VSX node forwards over the ISL to the primary VSX switch the DHCP requests received from the downstream endpoints. The upstream DHCP servers receive a single DHCP request. The downstream clients receive a single DHCP offer.

Both devices do not end up relaying DHCP requests to the server as duplicates. That scenario is usually the case with typical aggregation switches running VRRP-based redundancy.

IMPORTANT: If SVI is disabled on the primary VSX node and the primary goes down, the secondary switch will not take over and no DHCP requests will be relayed.

IP multicast routing

Multicast PIM routing provides fast failover. For each VSX downstream VLAN, both VSX switches as a PIM Designate Router (DR). One node is the actual DR, the other node is the proxy DR.

From the PIM protocol view point (join, prune, register. The role of the proxy DR is equal to the role of the actual DR. The proxy DR also sends PIM join messages to the upstream PIM router. Any VSX node receives a copy of IGMP join on the SL. Both the DR and proxy DR maintain the same multicast tables and build the shortest path tree.

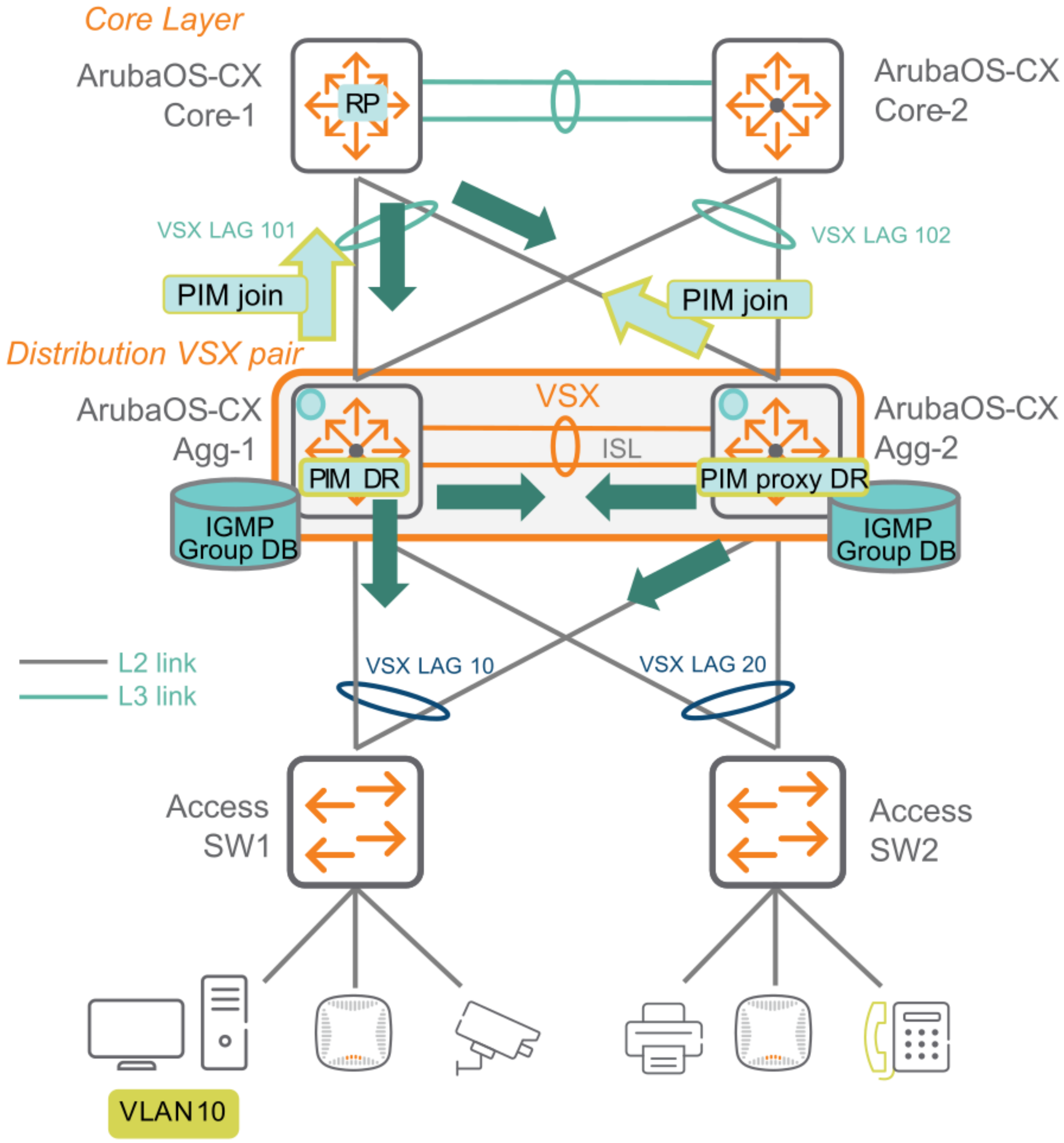
The proxy DR does not route traffic to downstream nodes. The proxy DR only acts as a bridge, all `mroute` entries present in the DB for downstream VLAN is being set as bridge entries in the proxy DR ASIC, for example pointing to the DR VSX node. Only the actual DR performs multicast routing and forward traffic destined to groups to its downstream VLANs in the data-path. There is no PIM asset mechanism as PIM forwarder is the DR.

DR/Proxy DR election is done per VLAN. Election process is first DR priority, then the highest IP address.

Multicast PIM routing is enabled through the `active-active` command. For example:

```
switch(config)# router pim
switch(config-pim)# active-active
```

See the *Command-Line Interface Guide* for your switch model and software version for more information about the `active-active` command.



Recommended values for system MAC and active gateway VMAC

It is highly recommended to use unicast MAC Address when assigning system-mac or active-gateway virtual MAC address. There are four ranges reserved for private use of unicast. The values are:

- x2-xx-xx-xx-xx-xx
- x6-xx-xx-xx-xx-xx
- xA-xx-xx-xx-xx-xx
- xE-xx-xx-xx-xx-xx

x can be any hexadecimal value.

Table 1: *Recommended values*

Function	System-mac	Active gateway Virtual MAC
Access	02:00:00:00:XX:00	12:00:00:00:XX:0Y
Aggregation	02:01:00:00:XX:00	12:01:00:00:XX:0Y
Core	02:02:00:00:XX:00	12:02:00:00:XX:0Y

In the above table, XX represents the unique cluster ID in the function and Y represents the virtual MAC ID (0 to F).

NOTE: Do not use multicast and broadcast MAC address as system-mac address.

Without a spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages, leading to a “broadcast storm” that can bring down the network. STP ensures that only one active path exists between any two nodes in a spanning tree instance. A spanning tree instance comprises a unique set of VLANs, and belongs to a specific spanning tree region. A region can comprise multiple spanning tree instances (each with a different set of VLANs), and allows one active path among regions in a network.

NOTE: Spanning-tree guards and filters are not allowed for configuration on the ISL.

Supported STP modes

The VSX switches support the following spanning tree protocols (STPs) with VSX:

- **MSTP**: Multiple Spanning Tree Protocol.
- **RPVST**: Rapid per-VLAN Spanning Tree Protocol.

How STP works with VSX

Both VSX switches appear as a single common Spanning Tree Bridge ID to STP partner devices upstream and downstream that participate to the same Spanning Tree domain. STP can be enabled on VSX switches and any nonrouting ports. Both VSX LAGs and non-VSX LAGs can participate in STP topology to avoid any loops.

STP on VSX uses the same bridge ID with the same MAC address on VSX LAGs and non-VSX LAGs, orphan ports. This MAC address is referred to as a common Bridge ID which consists of Spanning Tree priority and the switch MAC Address. The STP port state is the same for VSX LAG ports in VSX peer switches.

The Spanning Tree protocol runs independently on VSX nodes, which conforms to the dual-control plane VSX architecture. The primary VSX node is responsible to run the protocol for the VSX LAGs. In the normal state, the primary is "operational primary" and the secondary is "operational secondary". If a primary VSX node failure occurs, the secondary VSX node becomes the STP operational primary. When the primary VSX node goes back up, it takes back ownership of the STP operational primary role.

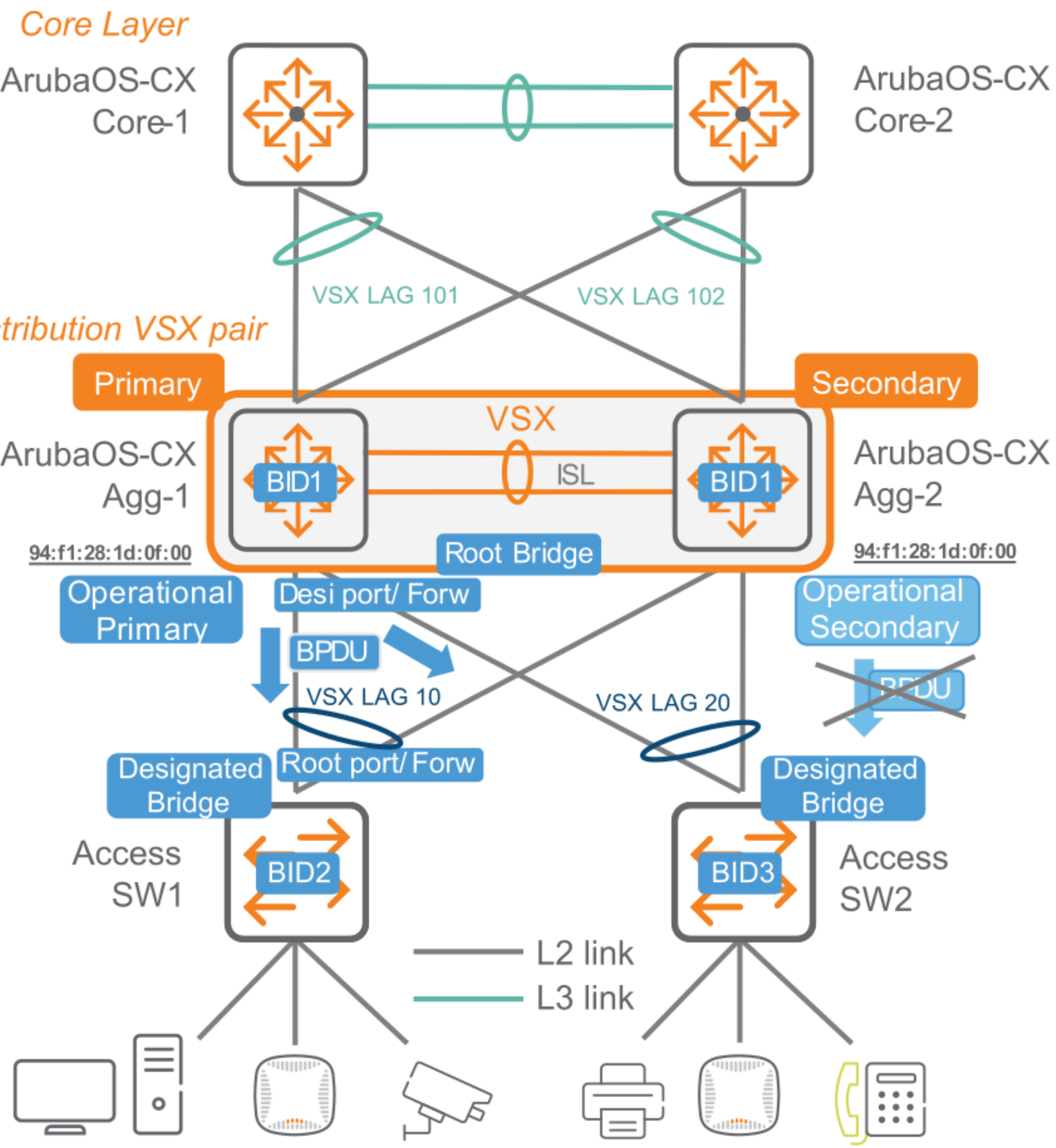
On VSX LAG ports, STP runs only from the operational primary, shown in the following figure. The operational secondary, also shown in the following figure, holds precomputed STP information for ready-state switch over thanks to STP states synchronization. The operational primary does STP state synchronization to the operational secondary for links member of the VSX LAG. That happens as a part of the initial sync (LACP, MAC, ARP, MSTP). During the switch-over, the new operational primary sends the BPDU downstream or upstream within 6 seconds (the default) of the spanning tree BPDU failure detection timer: 3x hello-timer (2s per default).

ISL is always part of STP, nonblocking and it sends and receives BPDUs.

IMPORTANT:

- Do not use the same system STP address for the other nodes. For the internal Spanning Tree protocol between VSX nodes, the Bridge_ID of the primary and secondary VSX nodes are derived from (-1, +1) from the `system-mac <MAC-ADDR>` command. For example, if the system MAC address is 00:00:00:00:00:10, then the other system MAC addresses cannot be 00:00:00:00:00:09, 00:00:00:00:00:10, and 00:00:00:00:00:11.
- You must have identical STP configurations on the primary and secondary VSX switches.
- It is recommended to have common system MAC addresses configured under the VSX context for stable STP convergence and stability.

Figure 9: *Sample STP on VSX configuration*



This figure shows MSTP with a VSX configuration showing BID1 ports as blocking.

More information

`system-mac`

MSTP

MSTP configurations

VSX at the distribution layer with MSTP enabled

In the following figure, the VSX pair is configured as a root switch. All the ports of the VSX LAGs, non-VSX LAGs, and orphan ports are in a forwarding state. Bridge Protocol Data Units (BPDUs), generated by a VSX pair, are the same on all ports, including VSX LAG, non-VSX LAG, and orphan. All switches must be in the same MSTP region consisting of the same configuration name and revision number, as set by the `spanning-tree config-name` *<CONFIG-NAME>* and `spanning-tree config-revision` *<REVISION-NUMBER>* commands.

Switches in the multiple MSTP region consist of different configuration name and revision number. The following example is the extract of MSTP multiple region configuration:

VSX switch

```
Primary# configure terminal
Primary(config)# spanning-tree config-name mstp1
Primary(config)# spanning-tree config-revision 1
```

Non-VSX switch

```
Core# configure terminal
Core(config)# spanning-tree config-name mstp2
Core(config)# spanning-tree config-revision 2
```

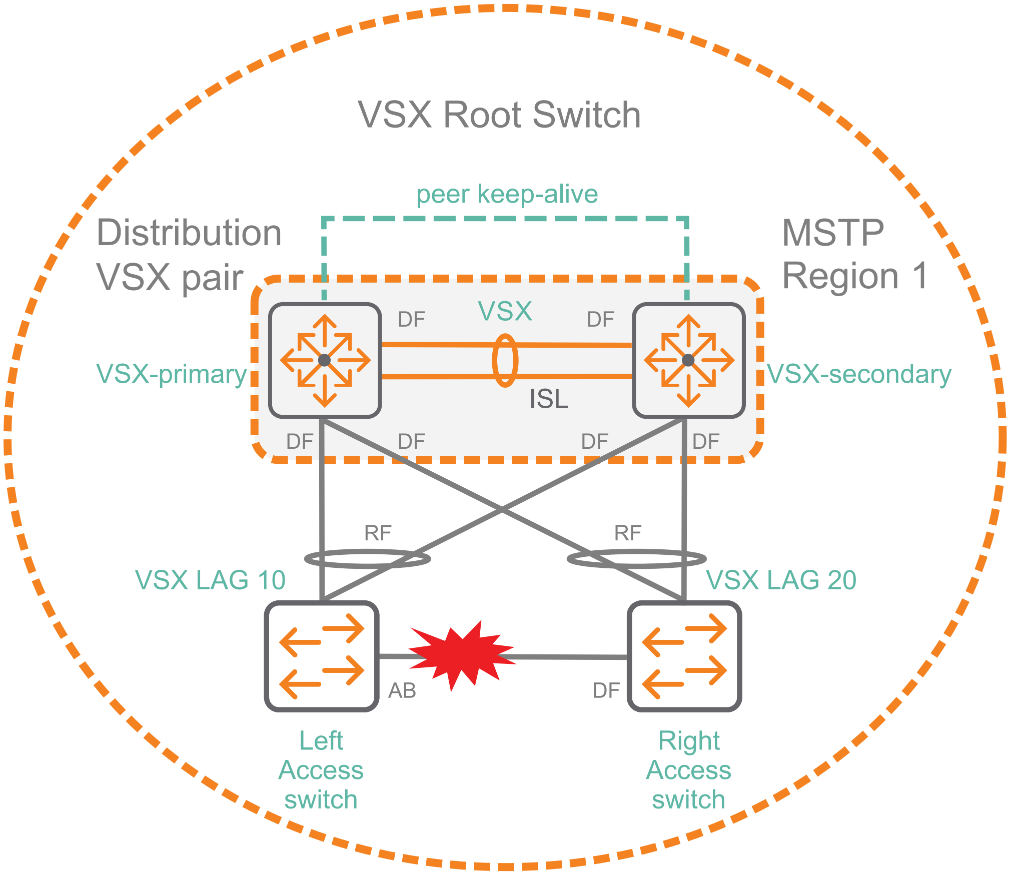
NOTE: The configuration should be similar on both VSX primary and secondary switches.

Table 2: *Definitions of the abbreviations used in the figures provided in this topic*

Abbreviation	Definition
AB	Alternate blocking; the port is in a blocked state.
DF	Designated forwarding; the port is in a forwarding state.
RF	Root forwarding; the port is in a forwarding state.

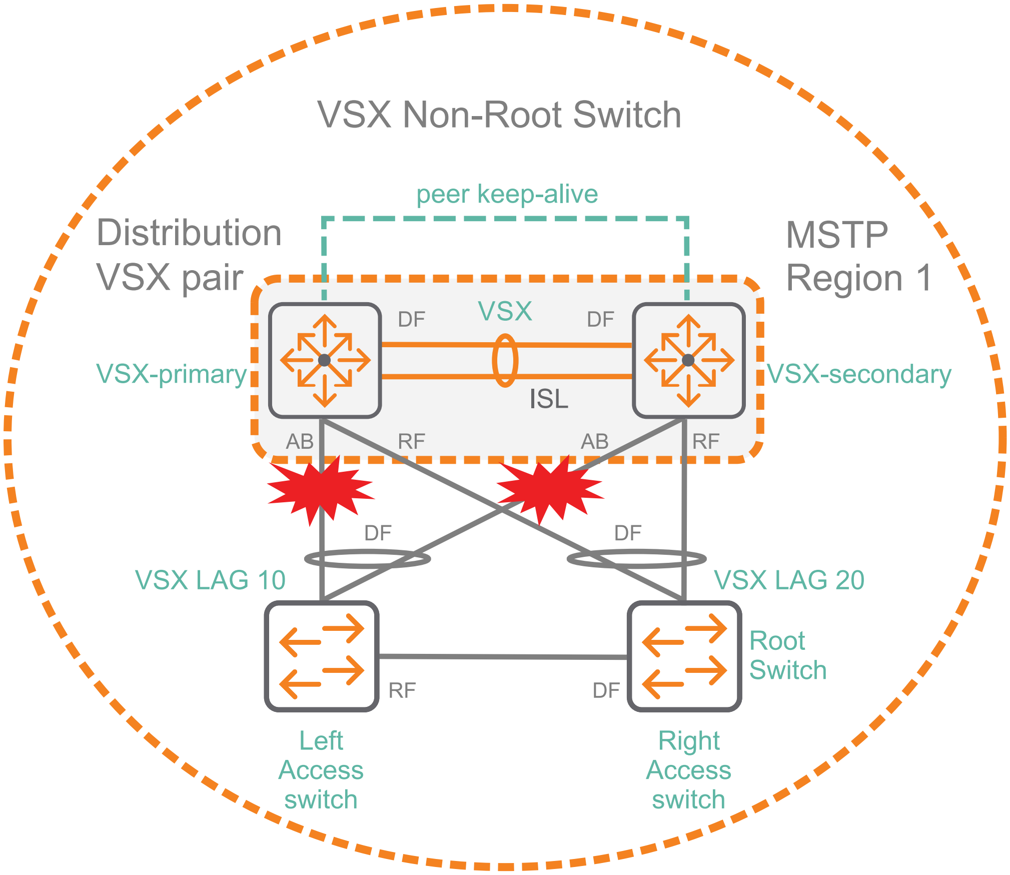
See **Sample configurations for MSTP on VSX** for the configuration for the topologies displayed in the figures in this topic.

Figure 10: *MSTP VSX pair as a root switch*



In the following figure, the VSX pair is not a root switch for STP topology. One of the VSX LAG ports is in the blocking state for resolving an L2 network loop. The VSX LAG port is in a blocking state on both VSX peer switches.

Figure 11: *MSTP VSX pair as a nonroot switch*

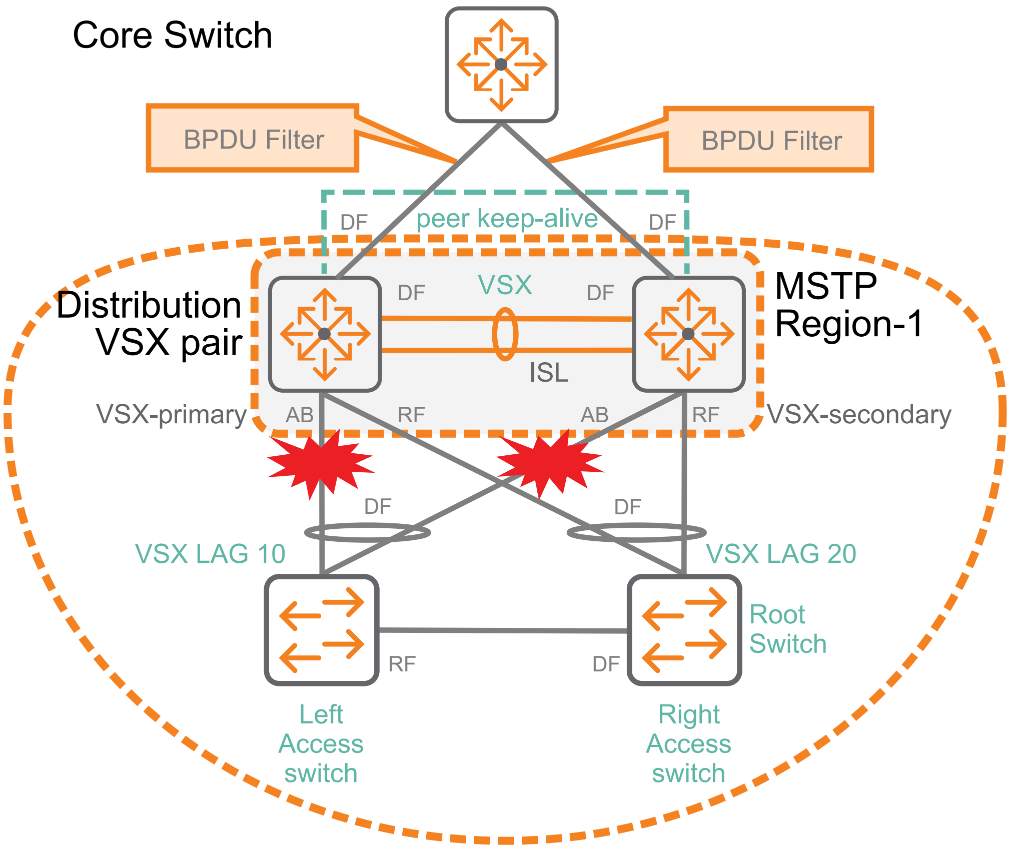


Distribution VSX pair connected to the core switch (SVI solution)

In the following figure, the VSX switch could be either a root switch or a nonroot switch for STP topology. One of the uplinks connected from the distribution layer to the core switch is in a blocking state because the MSTP is enabled in a VSX pair connected to a core switch, but the SVP configured without MSTP is enabled.

This configuration might also cause the flooding of the MSTP BPDUs (VLAN unaware) based on the VLAN configuration. VLANs must be configured differently on both ports to avoid flooding back to another VSX pair. Configure the BPDU filter on L2 ports connected to the core switch so that these ports will be in a forwarding state.

Figure 12: *Distribution layer with VSX and MSTP connected to the core switch*



More information

BFD reports a link being down before the LAG rebalances

Sample configurations for MSTP on VSX

IMPORTANT: For scaled MSTP on VSX configurations, configure all MSTP global and port configurations and then enable MSTP.

The following configurations are shown in **Figure 10: MSTP VSX pair as a root switch** and in **Figure 11: MSTP VSX pair as a nonroot switch**.

Configurations on the VSX primary switch

The following example is an extract from a configuration:

```
vlan 1-512
spanning-tree
spanning-tree priority 2
spanning-tree config-name Region-One
spanning-tree config-revision 1
spanning-tree instance 1 vlan 1,65,129,193,257,321,385,449
spanning-tree instance 2 vlan 2,66,130,194,258,322,386,450
spanning-tree instance 3 vlan 3,67,131,195,259,323,387,451
interface mgmt
no shutdown
```

```
ip dhcp
interface lag 10 multi-chassis
no shutdown
no routing
vlan trunk native 1
vlan trunk allowed all
lacp mode active
interface lag 20 multi-chassis
no shutdown
no routing
vlan trunk native 1
vlan trunk allowed all
lacp mode active
spanning-tree port-priority 1
interface 1/1/1
no shutdown
lag 10
interface 1/1/2
no shutdown
lag 20
interface 1/1/47
no shutdown
no routing
vlan trunk native 1 tag
vlan trunk allowed all
interface 1/1/48
no shutdown
ip address 1.1.1.1/24
vsx
inter-switch-link 1/1/47
system-mac 02:02:02:02:02:02
role primary
keepalive peer 1.1.1.2 source 1.1.1.1
linkup-delay-timer 30
```

Configurations on the VSX secondary switch

The following example is an extract from a configuration:

```
vlan 1-512
spanning-tree
spanning-tree priority 2
spanning-tree config-name Region-One
spanning-tree config-revision 1
spanning-tree instance 1 vlan 1,65,129,193,257,321,385,449
spanning-tree instance 2 vlan 2,66,130,194,258,322,386,450
spanning-tree instance 3 vlan 3,67,131,195,259,323,387,451
interface mgmt
no shutdown
ip dhcp
interface lag 10 multi-chassis
no shutdown
no routing
vlan trunk native 1
vlan trunk allowed all
lacp mode active
interface lag 20 multi-chassis
no shutdown
no routing
vlan trunk native 1
vlan trunk allowed all
lacp mode active
spanning-tree port-priority 1
interface 1/1/3
no shutdown
no routing
vlan trunk native 1 tag
vlan trunk allowed all
interface 1/1/4
no shutdown
lag 10
interface 1/1/45
no shutdown
lag 20
interface 1/1/46
no shutdown
ip address 1.1.1.2/24
interface 1/1/47
no shutdown
no routing
vlan access 1
vsx
inter-switch-link 1/1/3
system-mac 02:02:02:02:02:02
role secondary
keepalive peer 1.1.1.1 source 1.1.1.2
linkup-delay-timer 30
```

Configurations on left-access-switch

The following example is an extract from a configuration:

```
vlan 1-512
spanning-tree
spanning-tree config-name Region-One
spanning-tree config-revision 1
spanning-tree instance 1 vlan 1,65,129,193,257,321,385,449
spanning-tree instance 2 vlan 2,66,130,194,258,322,386,450
spanning-tree instance 3 vlan 3,67,131,195,259,323,387,451
interface mgmt
no shutdown
ip dhcp
interface lag 10
no shutdown
no routing
vlan trunk native 1
vlan trunk allowed all
lacp mode active
interface 1/1/5
no shutdown
lag 10
interface 1/1/6
no shutdown
lag 10
interface 1/1/43
no shutdown
no routing
vlan trunk allowed all
```

Configurations on right-access-switch

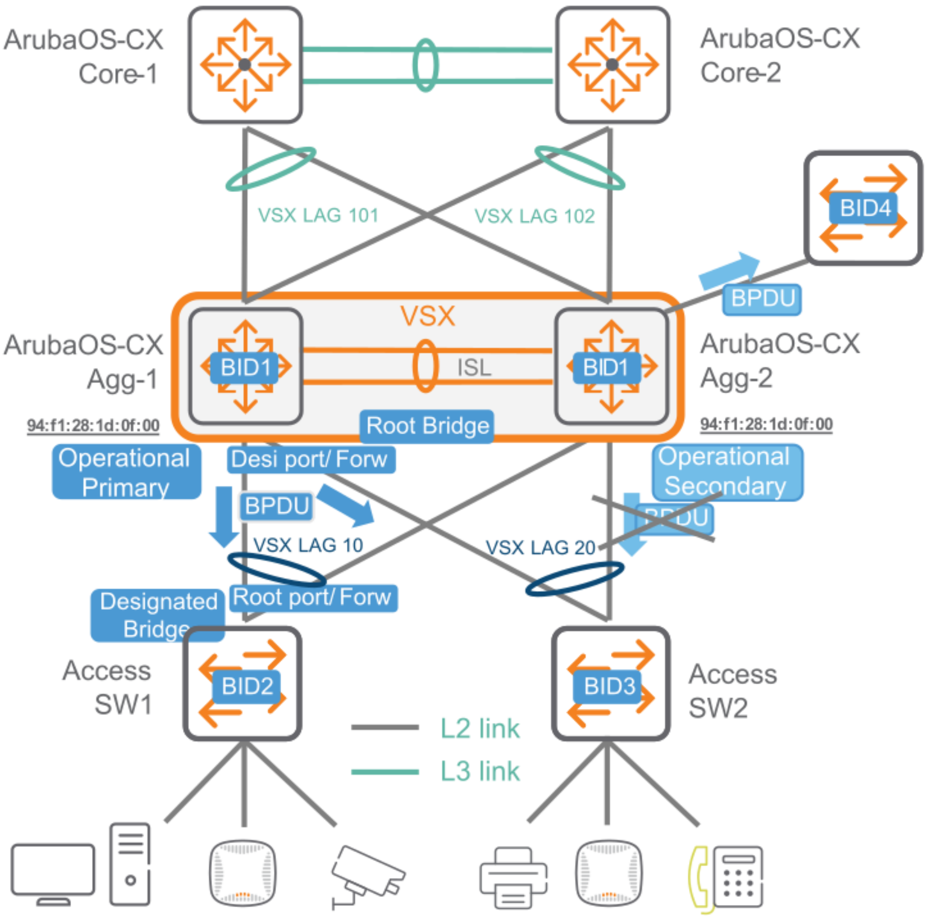
The following example is an extract from a configuration:

```
vlan 1-512
spanning-tree
spanning-tree config-name Region-One
spanning-tree config-revision 1
spanning-tree instance 1 vlan 1,65,129,193,257,321,385,449
spanning-tree instance 2 vlan 2,66,130,194,258,322,386,450
spanning-tree instance 3 vlan 3,67,131,195,259,323,387,451
interface mgmt
no shutdown
ip dhcp
interface lag 20
no shutdown
no routing
vlan trunk native 1
vlan trunk allowed all
lacp mode active
interface 1/1/7
no shutdown
lag 20
interface 1/1/8
no shutdown
lag 20
interface 1/1/41
no shutdown
no routing
vlan trunk allowed all
```

VSX and MSTP loop-protect configurations (physical and logical views)

The figures in this topic show the physical and logical views for VSX and MSTP loop-protect configurations with MSTP as the default instance.

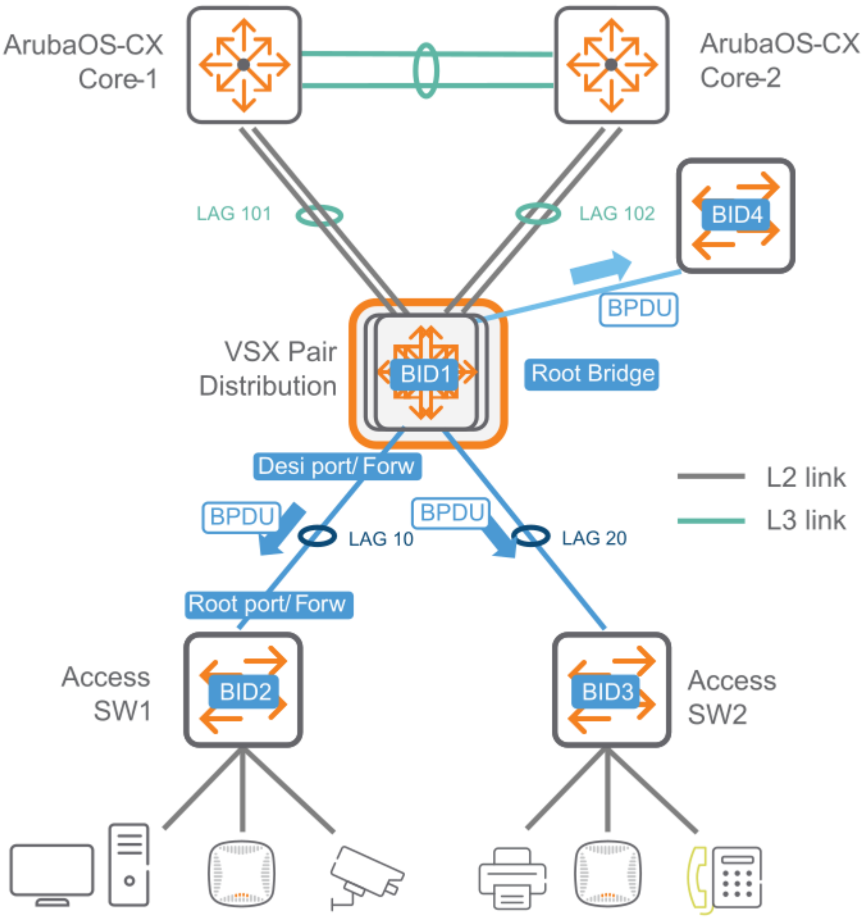
Figure 13: *Physical view of the VSX and MSTP loop-protect configurations*



The configuration from the previous figure is shown in its logical view, so that you can see how the network views the configuration. For example, the following figure shows that the VSX distributed pair as one switch.

The ports on Agg-2 are blocking traffic. The logical view in the next figure shows that the traffic is distributed so that the traffic continues to flow.

Figure 14: *Logical view of the VSX and MSTP loop-protect configurations*



STP interoperability with Loop-Protect in VSX

When both loop protect and STP are enabled on the switch:

- If switch first detects STP, STP blocks the port to stop the loop and loop protect feature will not come into effect.
- If switch first detects loop protect, loop protect blocks the port to stop the loop and STP will not take any effect as there are no loops.
- If loop protect has re-enable timer enabled, the port will be unblocked once the timer is expired. In this case, whichever protocol detects the loop first will block the port.

Show commands for MSTP

IMPORTANT: Before running the show commands, make sure that you have enabled STP synchronization between VSX peer switches. See **Enabling VSX synchronization of STP configurations between VSX peer switches**.

Task	Action
Verify that all switches are in the same MSTP region with the instance mapping to VLAN.	Enter the `show spanning-tree mst-config` command.
View the latest topology changes of the VSX peer.	**1.** Synchronize the time by entering the NTP (`vsx-sync time`) command. **2.** Enter the `show spanning-tree mst <0-64> vsx-peer` command.
Verify that the following global parameters are the same on VSX switches: • STP mode • STP region configuration for MSTP (config-name and config-revision) • STP instance to VLAN mapping • STP instance priority	**1.** Enter the `show running-config spanning-tree` command. **2.** Enter the `show running-config spanning-tree vsx-peer` command.

MSTP with VSX guidelines

- Path cost is not allowed to be configured on the ISL port.
- Layer 2 link connected parallel to ISL link is blocked by MSTP.
- Do not configure port-specific spanning tree configurations on the ISL.
- Multiple instances are supported though (default + 64).
- Topology changes for VSX LAGs are accounted on the active multichassis LAG role only.
- MSTP is supported in both VSX and non-VSX environments.
- The common bridge ID continues to be used even after the VSX split brain scenario is identified.
- STP configurations on VSX LAG ports must be the same on VSX switches. Use `vsx-sync mclag-interfaces` command for syncing STP and LAG interface configurations.
- Run the `show running-config spanning tree` and `show running-config spanning tree vsx-peer` commands for verifying that the following global parameters are the same on VSX switches:
    - STP mode.
    - STP region configuration for MSTP (config-name and config-revision)
    - STP instance to VLAN mapping
    - STP instance priority

Alternatively, you can also use `vsx-sync stp-global` to sync all the above mentioned global commands.

RPVST

A Rapid Per VLAN Spanning Tree (RPVST) system creates one STP instance per VLAN. You are required to create the RPVST instance explicitly.

For example to create an RPVST instance:

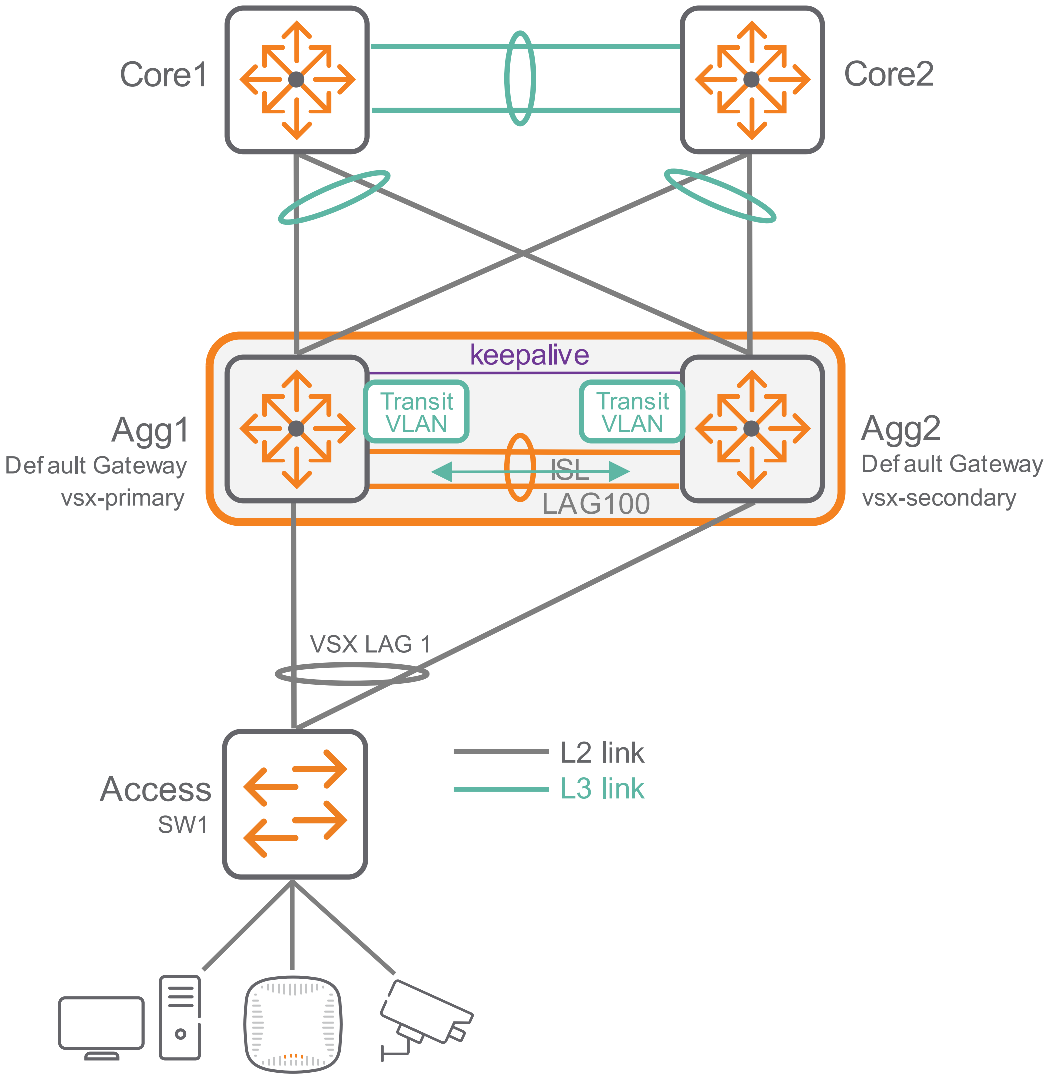
```
switch(config)# spanning-tree vlan 1
```

To create multiple RPVST instances, enter a range:

```
switch(config)# spanning-tree vlan 1-100
```

Sample RPVST configuration with VSX

The following figure shows a sample RPVST configuration.



The configuration for this figure is provided in the following sections.

VSX Primary Configuration

```
configure
hostname vsx-pri
vlan 1,10,20
```

```
spanning-tree mode rpvst
spanning-tree
spanning-tree vlan 1,10,20
interface mgmt
    no shutdown
    ip dhcp
interface lag 1 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed all
    lacp mode active
interface lag 100
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed all
    lacp mode active
interface 1/1/1
    no shutdown
    lag 100
interface 1/1/3
    no shutdown
    lag 1
interface 1/1/2
    no shutdown
    routing
    ip address 1.1.1.1/24
vsx
    system-mac 04:04:04:04:04:04
    inter-switch-link lag 100
    role primary
    keepalive peer 1.1.1.2 source 1.1.1.1
```

VSX secondary configuration

```
configure
hostname vsx-sec
vlan 1,10,20
spanning-tree mode rpvst
spanning-tree
spanning-tree vlan 1,10,20
interface mgmt
    no shutdown
    ip dhcp
interface lag 1 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed all
    lacp mode active
interface lag 100
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed all
    lacp mode active
```

```
interface 1/1/7
    no shutdown
    lag 100
interface 1/1/9
    no shutdown
    lag 1
interface 1/1/8
    no shutdown
    routing
    ip address 1.1.1.2/24
vsx
    system-mac 04:04:04:04:04:04
    inter-switch-link lag 100
    role secondary
    keepalive peer 1.1.1.1 source 1.1.1.2
```

Access switch configuration

```
configure
hostname l2-access
vlan 1,10,20
spanning-tree mode rpvst
spanning-tree
spanning-tree vlan 1,10,20
interface mgmt
    no shutdown
    ip dhcp
interface lag 1
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed all
    lacp mode active
interface 1/1/10
    no shutdown
    lag 1
interface 1/1/11
    no shutdown
    lag 1
```

VSX switch with RPVST, as root and nonroot

In the following figure, the VSX pair is configured as a root switch. All the ports of the VSX LAGs, non-VSX LAGs, and orphan ports are in a forwarding state. Bridge Protocol Data Units (BPDUs), generated by a VSX pair, are the same on all ports, including VSX LAG, non-VSX LAG, and orphan.

Table 3: *Definitions of the abbreviations used in the figures provided in this topic*

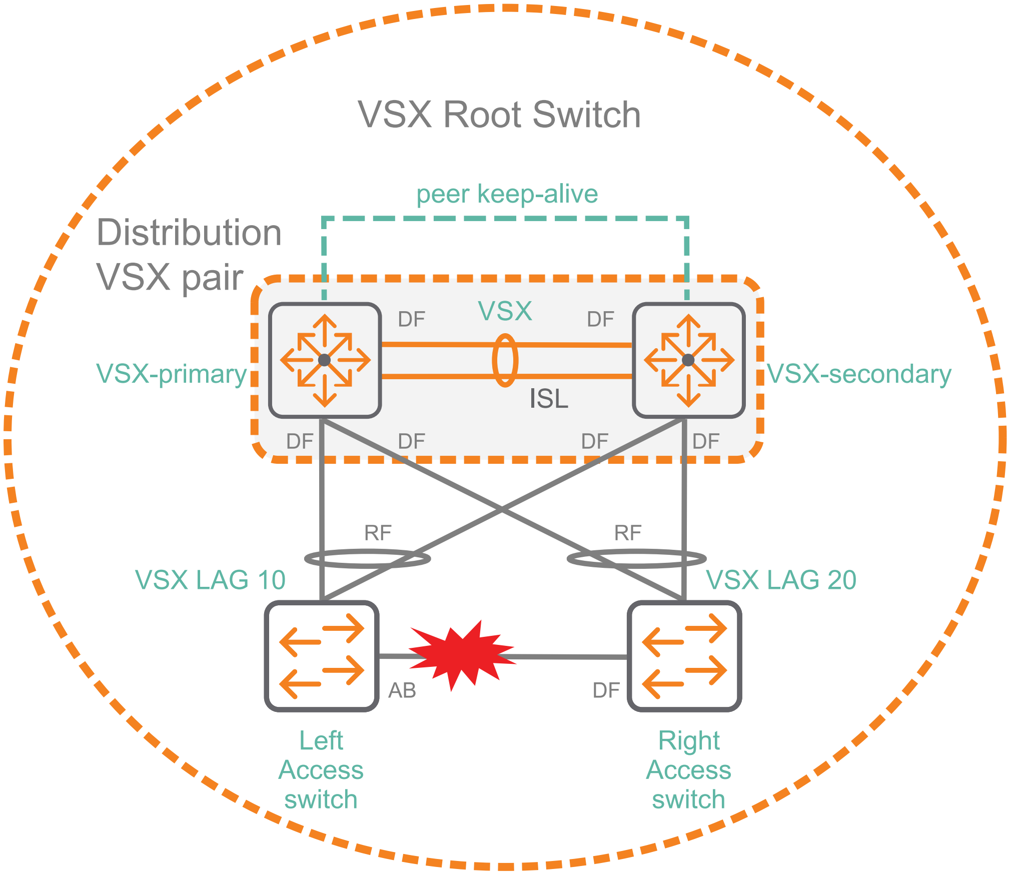
Abbreviation	Definition
AB	Alternate blocking; the port is in a blocked state.
DF	Designated forwarding; the port is in a forwarding state.
RF	Root forwarding; the port is in a forwarding state.

To make the VSX switch root for one or more RPVST instances, set the switch to the lowest bridge identifier for the tree:

- **For one RPVST instance:** `switch(config)# spanning-tree vlan 1 priority 1`
- **For more than one RPVST instance:** `switch(config)# spanning-tree vlan 1-100 priority 1` or `switch(config)# spanning-tree vlan 10,20,30 priority 1`

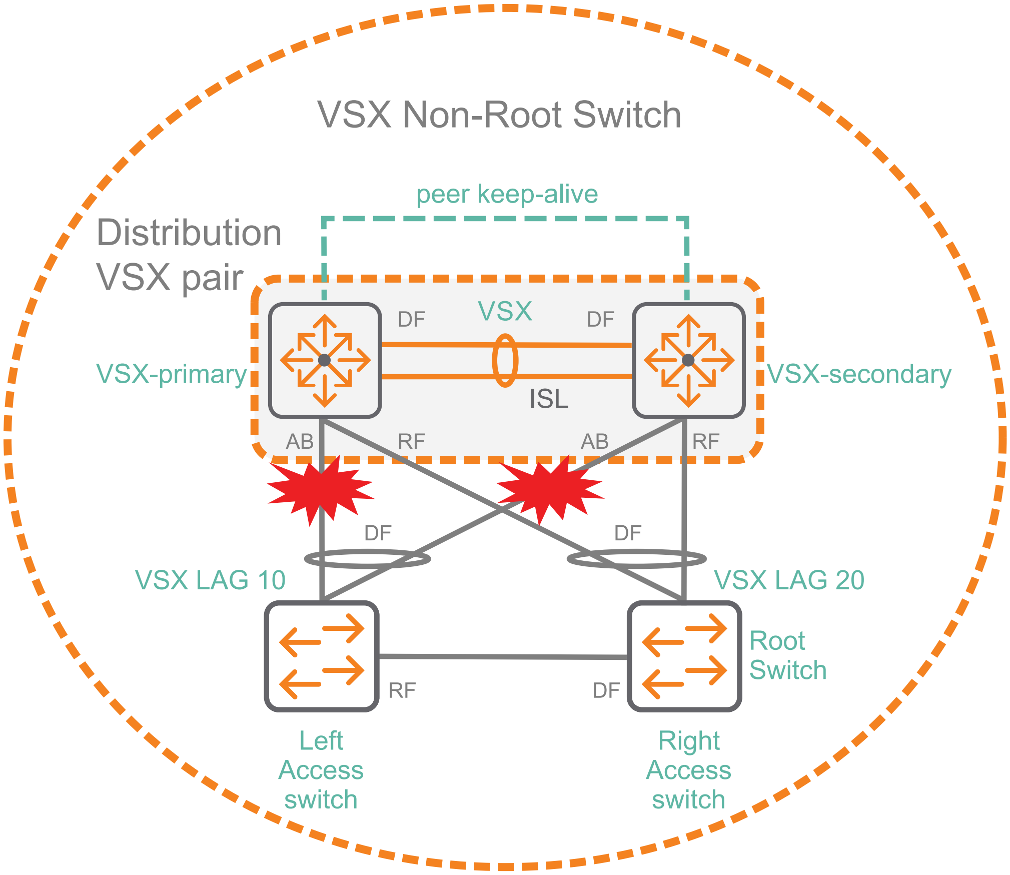
  The priority parameter has a range of 0 to 15 for setting the priority of the RPVST. The priority value is configured as a multiple of 4,096 (Default: 8). For example, when priority parameter is set as 1, the priority value is 4,096. When the priority parameter is set to 2, the priority value is 8,192. By default the priority parameter is 8, so the default priority value is 32,768.

Figure 15: *RPVST VSX pair as a root switch*



In the following figure, the VSX pair is not a root switch for STP topology. One of the VSX LAG ports is in the blocking state for resolving an L2 network loop. The VSX LAG port is in a blocking state on both VSX peer switches.

Figure 16: *RPVST VSX pair as a nonroot switch*



Configuring a VSX switch as root for one or more RPVST instances

Procedure

1. For a single RPVST instance, enter for example:

```
switch(config)# spanning-tree vlan 1 priority 1
```

2. For multiple RPVST instances, enter a range for example:

```
switch(config)# spanning-tree vlan 1-100 priority 1
```

```
switch(config)# spanning-tree vlan 10,20,30 priority 1
```

More information

VSX switch with RPVST, as root and nonroot

Show commands for RPVST

IMPORTANT: Before running the show commands, make sure that you have enabled STP synchronization between VSX peer switches. See **Enabling VSX synchronization of STP configurations between VSX peer switches**.

Task	Action
View information on the RPVST instance of the specified VLAN.	`switch# show spanning-tree vlan <VLAN-ID>`
View information on the RPVST instance of the specified VLAN and displays details on the RPVST instance for the VLAN.	`switch# show spanning-tree vlan <VLAN-ID> detail` The output of this command shows the value of the `Multi-Chassis` role. When a switch has the `Multi-Chassis` role set to `active`, the switch performs the STP operation. When a switch has the `Multi-Chassis` role set to `standby`, the switch relays the information to the switch with the active role for STP tasks. For an example of the output from this command, see **How the `Multi-Chassis` role works**.
View information on the RPVST instance of the specified VLAN on the peer VSX switch.	`switch# spanning-tree vlan <VLAN-ID> vsx-peer`
View information on the RPVST instance of the specified VLAN and displays details on the RPVST instance for the VLAN on the peer VSX switch.	`switch# show spanning-tree vlan <VLAN-ID> detail vsx-peer`
Verify that the following global parameters are the same on VSX switches: • STP mode • RPVST instance creation • RPVST instance priority configuration	**1.** Enter the `show running-config spanning-tree` command. **2.** Enter the `show running-config spanning-tree vsx-peer` command.
View a summary of the port roles or root information.	`switch# show spanning-tree summary {port | root}`

How the `Multi-Chassis` role works

The switch performs the STP operation on the switch that has the `Multi-Chassis` role set to active. The switch with the role set to standby relays the information to the switch with the active role for STP tasks.

The primary VSX switch has the `Multi-Chassis` role set to active by default, just as the secondary VSX switch has the `Multi-Chassis` role set to standby by default. The `Multi-Chassis` role on the secondary VSX switch changes from standby to active if the primary VSX switch goes down.

The `show spanning-tree vlan <VLAN-ID> detail` command provides information about the value of the `Multi-Chassis` role. In the following example, the primary switch has the `Multi-Chassis` role set to active, and the secondary switch has the `Multi-Chassis` role set to standby.

Example of the `Multi-Chassis` role with the active value:

```
VSX-Primary# show spanning-tree vlan 2 detail

VLAN2
Spanning tree status : Enabled Protocol: RPVST
  Root ID    Priority   : 32768
             MAC-Address: 38:21:c7:66:24:00
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID  Priority  : 32768
             MAC-Address: 38:21:c7:66:24:00
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

Port         Role           State        Cost           Priority   Type
------------ -------------- ------------ ------------ ---------- ----------
1/3/1        Designated     Forwarding   1              128        point_to_point
lag2         Designated     Forwarding   20000          64         point_to_point

Topology change flag          : True
Number of topology changes    : 3
Last topology change occurred : 47 seconds ago

Port 1/3/1
Designated root has priority               :32768 Address: 38:21:c7:66:24:00
Designated bridge has priority             :32768 Address: 38:21:c7:66:24:00
Designated port                            : 1153
Number of transitions to forwarding state  : 1
Bpdus sent 28, received 28
TCN_Tx: 2, TCN_Rx: 0

Port lag2
Designated root has priority               :32768 Address: 38:21:c7:66:24:00
Designated bridge has priority             :32768 Address: 38:21:c7:66:24:00
Designated port                            : 770
Multi-Chassis role                         :active
Number of transitions to forwarding state  : 1
Bpdus sent 28, received 3
TCN_Tx: 2, TCN_Rx: 2

VSX-Secondary# show spanning-tree vlan 2 detail

VLAN2
Spanning tree status : Enabled Protocol: RPVST
  Root ID    Priority   : 32768
             MAC-Address: 38:21:c7:66:24:00
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

  Bridge ID  Priority  : 32768
             MAC-Address: 38:21:c7:66:24:00
             Hello time(in seconds):2  Max Age(in seconds):20
             Forward Delay(in seconds):15

Port         Role           State        Cost           Priority   Type
------------ -------------- ------------ ------------ ---------- ----------
```

```
1/3/1        Designated     Forwarding   1              128        point_to_point
lag2         Designated     Forwarding   20000          64         point_to_point

Topology change flag          : False
Number of topology changes    : 2
Last topology change occurred : 35 seconds ago

Port 1/3/1
Designated root has priority              :32768 Address: 38:21:c7:66:24:00
Designated bridge has priority            :32768 Address: 38:21:c7:66:24:00
Designated port                           : 129
Number of transitions to forwarding state : 2
Bpdus sent 24, received 22
TCN_Tx: 1, TCN_Rx: 2

Port lag2
Designated root has priority              :-32768 Address: 38:21:c7:66:24:00
Designated bridge has priority            :-32768 Address: 38:21:c7:66:24:00
Designated port                           :770
Multi-Chassis role                        :standby
Number of transitions to forwarding state : 3
Bpdus sent 0, received 0
TCN_Tx: 0, TCN_Rx: 0
```

RPVST with VSX guidelines

- Path cost is not allowed to be configured on the ISL port.
- Do not configure port-specific spanning tree configurations on the ISL.
- Do not have redundant links to the ISL.
- Topology changes for VSX LAGs are accounted on the **active** multichassis LAG role only.
- RPVST is supported in both VSX and non-VSX environments.
- The common bridge ID continues to be used even after the VSX split brain scenario is identified.
- STP configurations on VSX LAG ports must be the same on VSX switches.
- To find the maximum supported RPVST instances that can be configured, enter the following command: `show capacities rpvst`

Loop protect can be enabled on VSX. Loop protect is a switch feature, which is used to identify and prevent layer 2 loops in a network. The loop protect feature blocks the port based on the configured action, these actions may be:

- Tx-Disable
- Tx-Rx-Disable
- Do-Not-Disable

See the *Layer 2 Bridging Guide* for information about loop protect. To set up loop protect with VSX, loop protect must be enabled on the interfaces on the primary and secondary VSX switches.

How loop protect works over VSX

Assume that you have the loop protect feature enabled on lag 1/1/1 on the primary VSX switch and loop protect enabled on lag 1/1/2 on the secondary VSX switch. When a loop occurs, loop protect notifies the secondary VSX switch that a loop is on the network and interface1/1/2 was blocked. When you enter `show interface 1/1/2` on the secondary switch, the output from the command indicates that the interface was blocked by VSX when in fact the loop protect feature blocked the interface to stop the loop.

If you enter `show lacp interfaces` on the downstream switch, the forwarding state of the blocked interfaces is displayed as `down`, as shown in the following example:

```
switch(config)# show  lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
1/3/1   lag1       130   1     ALFNCD  f8:60:f0:06:87:00 65534  1    up
1/3/2   lag1       131   1     ALFNCD  f8:60:f0:06:87:00 65534  1    up
1/7/3   lag2       388   1     ALFOE   f8:60:f0:06:87:00 65534  2    lacp-block
1/10/46 lag2       623   1     ALFOE   f8:60:f0:06:87:00 65534  2    lacp-block


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/3/1   lag1       206   1     ALFNCD  f8:60:f0:06:49:00 65534  1
1/3/2   lag1       1130  1     ALFNCD  f8:60:f0:06:49:00 65534  1
1/7/3   lag2       0     65534 PLFOEX  00:00:00:00:00:00 65534  0
1/10/46 lag2       0     65534 PLFOEX  00:00:00:00:00:00 65534  0
```

Interface `lag2`, which was shown as `lacp-blocked` in the previous example is shown as `down` on the primary VSX switch, as shown in the following example:

```
switch(config)# show loop-protect

Status and Counters - Loop Protection Information

Transmit Interval             : 5 (sec)
Port Re-enable Timer          : Disabled

Interface lag1
  Loop-protect enabled        : Yes
  Loop-Protect enabled VLANs  : 1-100
  Action on loop detection    : TX disable
  Loop detected count         : 1
  Loop detected               : Yes
    Detected on VLAN          : 10
    Detected at               : 2019-09-27T00:12:55
  Interface status            : up

Interface lag2
  Loop-protect enabled        : Yes
  Loop-Protect enabled VLANs  : 2021-2121
  Action on loop detection    : TX disable
  Loop detected count         : 1
  Loop detected               : Yes
    Detected on VLAN          : 2103
    Detected at               : 2019-09-27T00:13:14
  Interface status            : down

switch(config)# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
1/4/14  lag1(mc)   206   1     ALFNCD  f8:60:f0:06:49:00 65534  1    up
1/5/15  lag2(mc)                                                     down


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/4/14  lag1(mc)   130   1     ALFNCD  f8:60:f0:06:87:00 65534  1
1/5/15  lag2(mc)
```

Interface `lag2` is also shown as `down` on the secondary VSX switch, as shown in the following example:

```
switch(config)# show loop-protect

Status and Counters - Loop Protection Information

Transmit Interval             : 5 (sec)
Port Re-enable Timer          : 15 (sec)
```

```
Interface lag1
  Loop-protect enabled        : Yes
  Loop-Protect enabled VLANs  : 1-100
  Action on loop detection    : TX disable
  Loop detected count         : 0
  Loop detected               : No
  Interface status            : up

Interface lag2
  Loop-protect enabled        : Yes
  Loop-Protect enabled VLANs  : 2021-2121
  Action on loop detection    : TX disable
  Loop detected count         : 0
  Loop detected               : No
  Interface status            : down

switch(config)# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
1/3/2   lag1(mc)   1130  1     ALFNCD  f8:60:f0:06:49:00 65534  1    up
1/9/3   lag2(mc)                                                     down


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/3/2   lag1(mc)   131   1     ALFNCD  f8:60:f0:06:87:00 65534  1
1/9/3   lag2(mc)
```

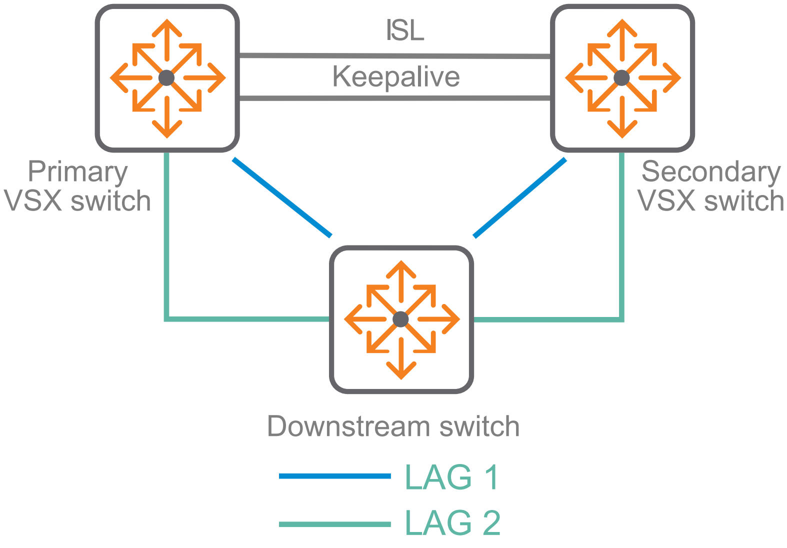
Setting up loop protect over VSX

Procedure

1. Create the VSX LAG.
2. Enable loop protect on the primary and secondary VSX switches. See the *Layer 2 Bridging Guide* for information about how to enable loop protect on a switch.

An example configuration of loop protect over VSX

The following figure is a simplified configuration. Most network configurations will have more than one downstream switch.



Both LAGs are configured as VLAN trunk allowed 1-2000

The following sections provide information about the configurations on the switches before and after configuring the loop protect feature.

VSX configurations before enabling loop protect

This section provides configuration information for the primary VSX switch, secondary VSX switch, and downstream switch before loop protect is enabled.

VSX primary switch before enabling loop protect

```
hostname Primary
module 1/1 product-number jl363a
cli-session
    timeout 0
ssh server vrf mgmt
vlan 1-2000
interface mgmt
    no shutdown
    ip dhcp
interface lag 1 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface lag 2 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface 1/1/1
    no shutdown
    lag 1
interface 1/1/2
    no shutdown
    lag 2
```

```
interface 1/1/3
    no shutdown
    no routing
    vlan trunk native 1 tag
    vlan trunk allowed all
interface 1/1/30
    no shutdown
    ip address 10.1.1.1/24
vsx
    inter-switch-link 1/1/3
    role primary
    keepalive peer 10.1.1.2 source 10.1.1.1
```

LACP interface configuration

```
Primary# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
1/1/1   lag1(mc)   1     1     ALFNCD  04:09:73:62:c8:00 65534  1    up
1/1/2   lag2(mc)   2     1     ALFNCD  04:09:73:62:c8:00 65534  2    up


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/1/1   lag1(mc)   18    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  1
1/1/2   lag2(mc)   20    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  2
```

VSX configuration

```
Primary# show vsx brief
ISL State                              : In-Sync
Device State                           : Peer-Established
Keepalive State                        : Keepalive-Established
Device Role                            : primary
Number of Multi-chassis LAG interfaces : 2
```

VSX secondary switch before enabling loop protect

```
hostname Secondary
module 1/1 product-number jl363a
cli-session
    timeout 0
ssh server vrf mgmt
vlan 1-2000
interface mgmt
    no shutdown
    ip dhcp
```

```
interface lag 1 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface lag 2 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface 1/1/1
    no shutdown
    lag 1
interface 1/1/2
    no shutdown
    lag 2
interface 1/1/3
    no shutdown
    no routing
    vlan trunk native 1 tag
    vlan trunk allowed all
interface 1/1/30
    no shutdown
    ip address 10.1.1.2/24
vsx
    inter-switch-link 1/1/3
    role secondary
    keepalive peer 10.1.1.1 source 10.1.1.2
```

LACP interface configuration

```
Secondary# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
       Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
1/1/1   lag1(mc)   1001  1     ALFNCD  04:09:73:62:c8:00 65534  1    up
1/1/2   lag2(mc)   1002  1     ALFNCD  04:09:73:62:c8:00 65534  2    up


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/1/1   lag1(mc)   19    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  1
1/1/2   lag2(mc)   31    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  2
```

VSX configuration

```
Secondary# show vsx brief
ISL State                               : In-Sync
Device State                            : Peer-Established
Keepalive State                         : Keepalive-Established
Device Role                             : secondary
Number of Multi-chassis LAG interfaces : 2
```

Downstream switch before enabling loop protect

```
hostname Downstream
cli-session
    timeout 0
ssh server vrf mgmt
vlan 1-2000
interface mgmt
    no shutdown
    ip dhcp
interface lag 1
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface lag 2
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface 1/1/17
    no shutdown
    lag 1
interface 1/1/18
    no shutdown
    lag 1
interface 1/1/19
    no shutdown
    lag 2
interface 1/1/30
    no shutdown
    lag 2
```

LACP interface configuration

```
Downstream# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
```

```
1/1/17  lag1       18    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  1    up
1/1/18  lag1       19    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  1    up
1/1/19  lag2       20    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  2    up
1/1/30  lag2       31    1     ALFNCD  e0:07:1b:cb:e1:5a 65534  2    up


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/1/17  lag1       1     1     ALFNCD  04:09:73:62:c8:00 65534  1
1/1/18  lag1       1001  1     ALFNCD  04:09:73:62:c8:00 65534  1
1/1/19  lag2       2     1     ALFNCD  04:09:73:62:c8:00 65534  2
1/1/30  lag2       1002  1     ALFNCD  04:09:73:62:c8:00 65534  2
```

VSX configurations after enabling loop protect

This section provides configuration information for the primary VSX switch, secondary VSX switch, and downstream switch after loop protect is enabled.

VSX primary switch after enabling loop protect

The following configuration shows that loop protect is enabled.

```
hostname Primary
module 1/1 product-number jl363a
cli-session
    timeout 0
ssh server vrf mgmt
vlan 1-2000
interface mgmt
    no shutdown
    ip dhcp
interface lag 1 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
    loop-protect
    loop-protect vlan 1-2000
interface lag 2 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
    loop-protect
    loop-protect vlan 1-2000
interface 1/1/1
    no shutdown
    lag 1
interface 1/1/2
    no shutdown
    lag 2
interface 1/1/3
    no shutdown
    no routing
    vlan trunk native 1 tag
    vlan trunk allowed all
interface 1/1/30
    no shutdown
    ip address 10.1.1.1/24
vsx
    inter-switch-link 1/1/3
    role primary
    keepalive peer 10.1.1.2 source 10.1.1.1
```

VSX secondary after before enabling loop protect

```
hostname Secondary
module 1/1 product-number jl363a
cli-session
    timeout 0
ssh server vrf mgmt
vlan 1-2000
interface mgmt
    no shutdown
    ip dhcp
interface lag 1 multi-chassis
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
    loop-protect
    loop-protect vlan 1-2000
interface lag 2 multi-chassis
    no shutdown
    no routing
   vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
    loop-protect
    loop-protect vlan 1-2000
interface 1/1/1
    no shutdown
    lag 1
interface 1/1/2
    no shutdown
    lag 2
interface 1/1/3
    no shutdown
    no routing
    vlan trunk native 1 tag
    vlan trunk allowed all
interface 1/1/30
    no shutdown
    ip address 10.1.1.2/24
vsx
    inter-switch-link 1/1/3
    role secondary
    keepalive peer 10.1.1.1 source 10.1.1.2
```

Downstream switch after enabling loop protect

```
hostname Downstream
cli-session
    timeout 0
ssh server vrf mgmt
vlan 1-2000
interface mgmt
    no shutdown
    ip dhcp
interface lag 1
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface lag 2
    no shutdown
    no routing
    vlan trunk native 1
    vlan trunk allowed 1-2000
    lacp mode active
interface 1/1/17
    no shutdown
    lag 1
interface 1/1/18
    no shutdown
    lag 1
interface 1/1/19
    no shutdown
    lag 2
interface 1/1/30
    no shutdown
    lag 2
```

Best practices for loop protect over VSX

- Enable loop protect on both primary and secondary VSX switches.
- Do not enable loop protect on the ISL link for the primary and secondary VSX switches.
- If you enable an action for loop protect, such as `do-not-disable`, and another action, such as `Tx-Rx-Disable`, is already in effect, loop protect must be disabled and then re-enabled.
- If you run the `loop-protect action do-not-disable` command, on every transmit interval, the loop is detected and the detection is reported through an SNMP trap and an event log message.

  You can view the events for just the loop protect feature by entering the `show events -d hpe-lpd` command.
- The total number of VLANs across ports is (ports x VLANs) = 4094 ports per VLAN. Loop protect can be configured on a maximum of 4094 VLANs across all interfaces without updating CoPP policies for loop protect. If your network configuration requires you to configure more VLAN, update your CoPP policies values for loop protect to ensure that you allocate more resources. You can assign a maximum of 10,000 VLANs across all the interfaces.

Chapter 10
EVPN VSX support

Ethernet VPN (EVPN) is supported with VSX . The two VSX pairs act as independent BGP routing entities to the other VXLAN tunnel endpoints (VTEPs) or spines for control packets. However, in the datapath, both of them act as a single logical VTEP. This is achieved by using different IP addresses for establishing the BGP session and using a common IP as next-hop to represent the VTEP.

For more information on EVPN VSX support, see the *EVPN VSX support* chapter in the *VXLAN Guide*.

Upstream connectivity options

This firmware supports the following upstream connectivity options:

- **Routed Only Port (ROP):** A physical port on a switch that process all Layer 3 functions for packets to or from the said port without any binding to VLAN processing. See **Figure 17: ROP with a single VRF in the VSX environment**.
- **Switched Virtual Interface (SVIs) (multiple VRFs):** An SVI is a logical Layer 3 interface configured per VLAN (one-to-one mapping) that performs all Layer 3 processing for packets to or from all switch ports associated with that VLAN. See **Figure 18: SVI (multiple VRFs) in a VSX environment**.
- **VSX LAG SVIs with multiple VRFs.** See **Figure 19: VSX LAG and layer 3 ECMP**

Figure 17: *ROP with a single VRF in the VSX environment*

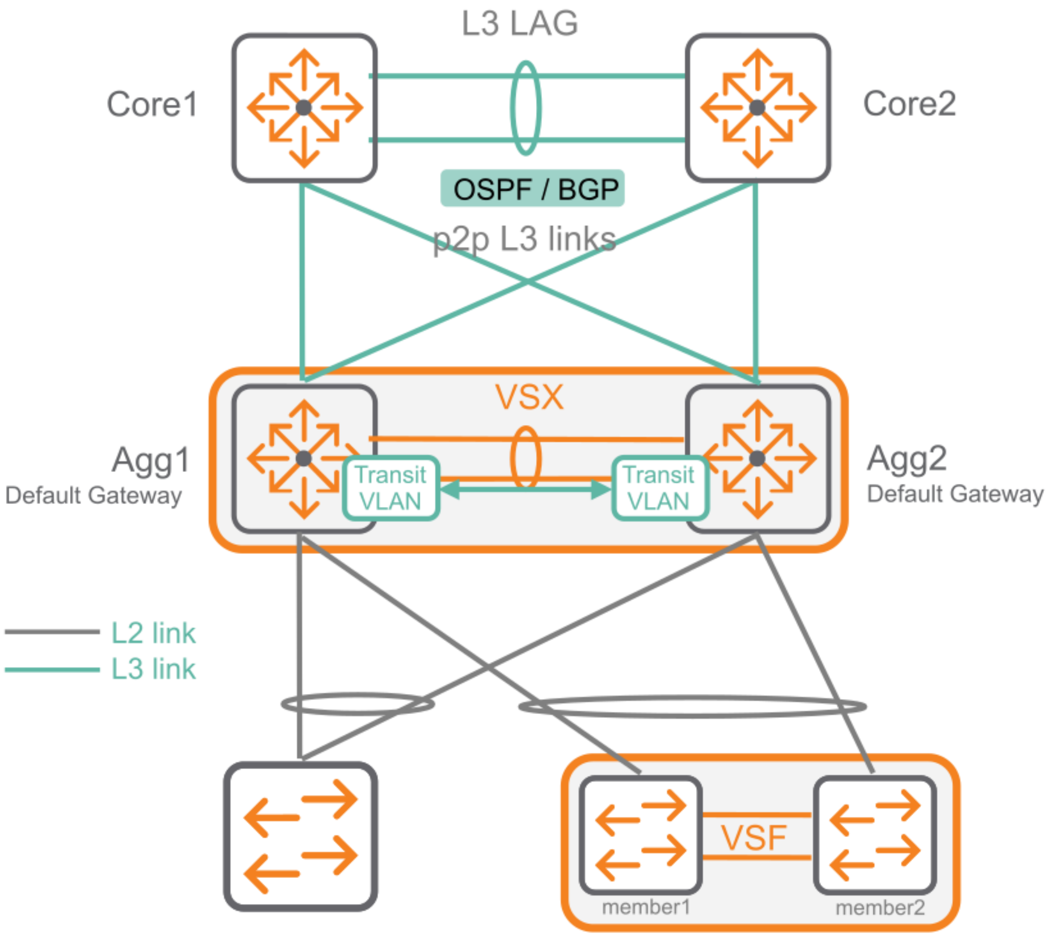


Figure 18: *SVI (multiple VRFs) in a VSX environment*

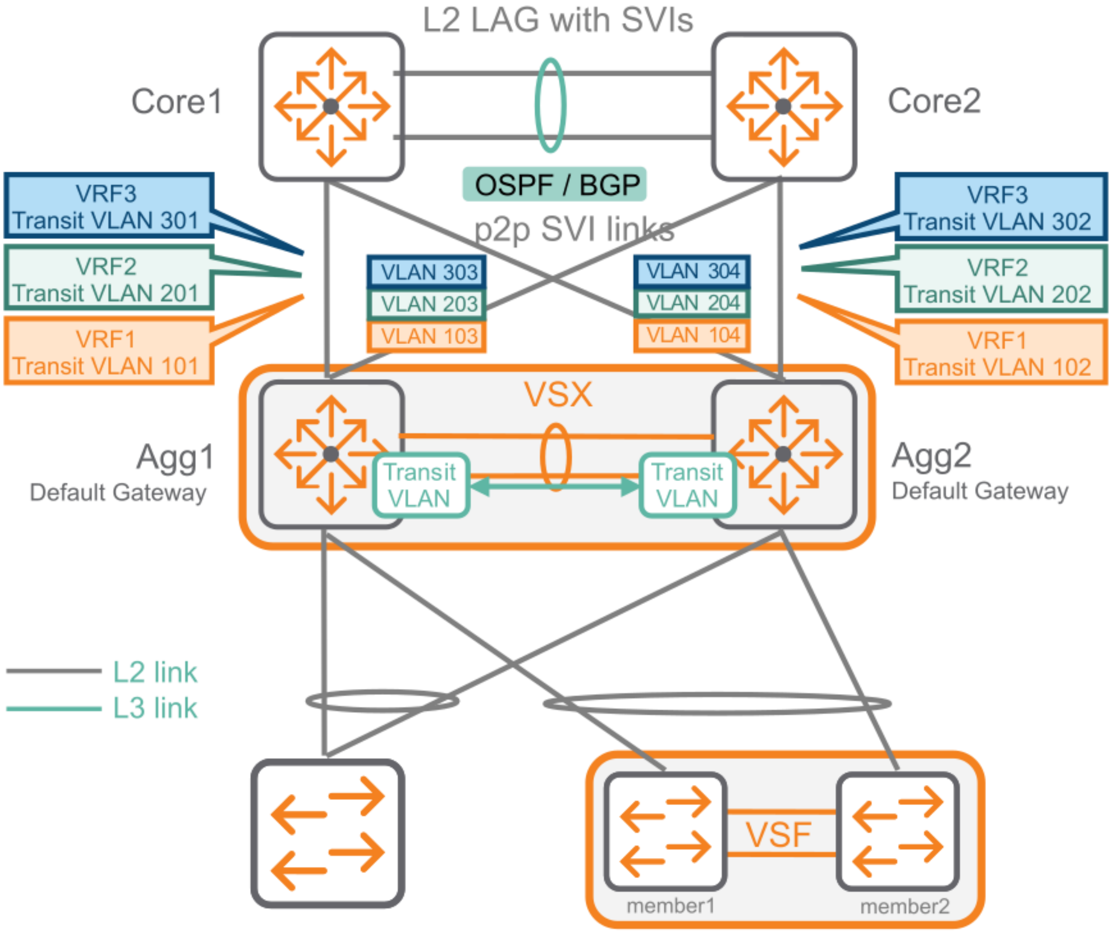
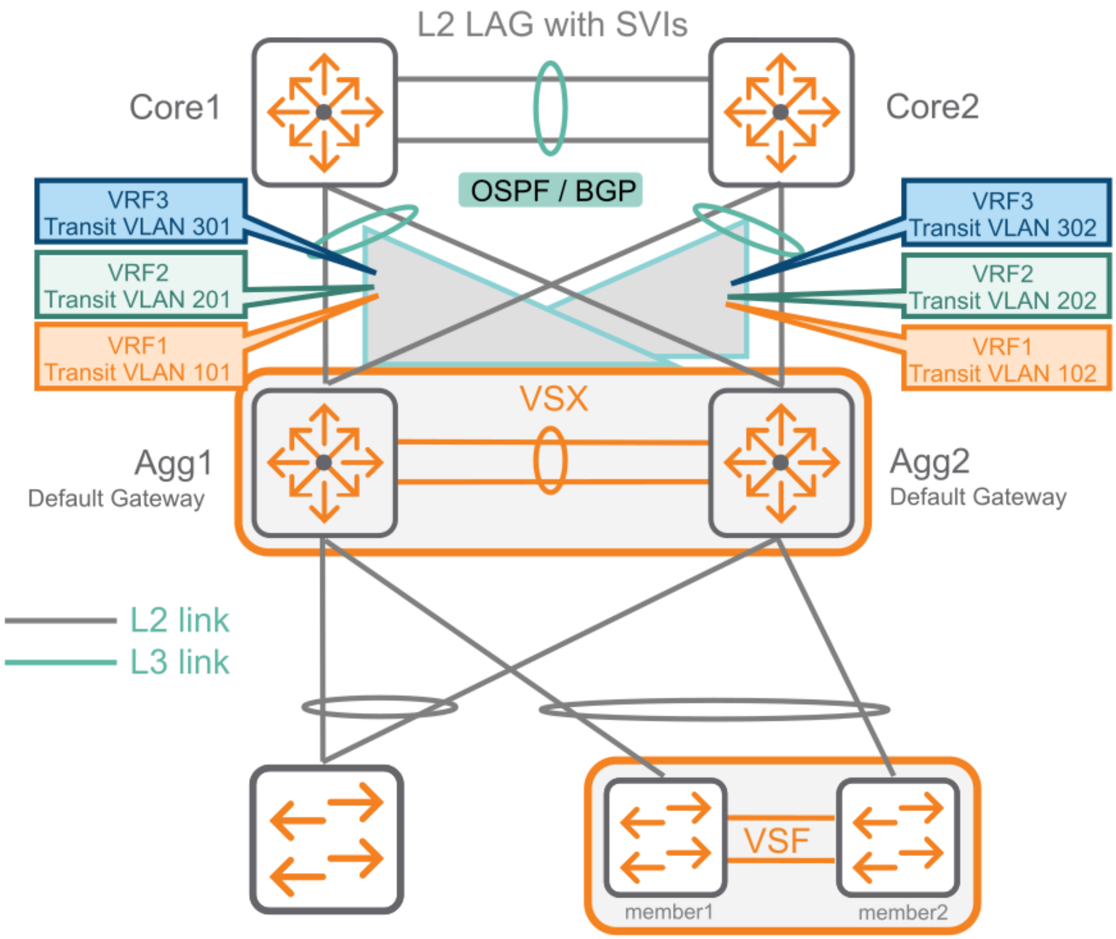


Figure 19: *VSX LAG and layer 3 ECMP*



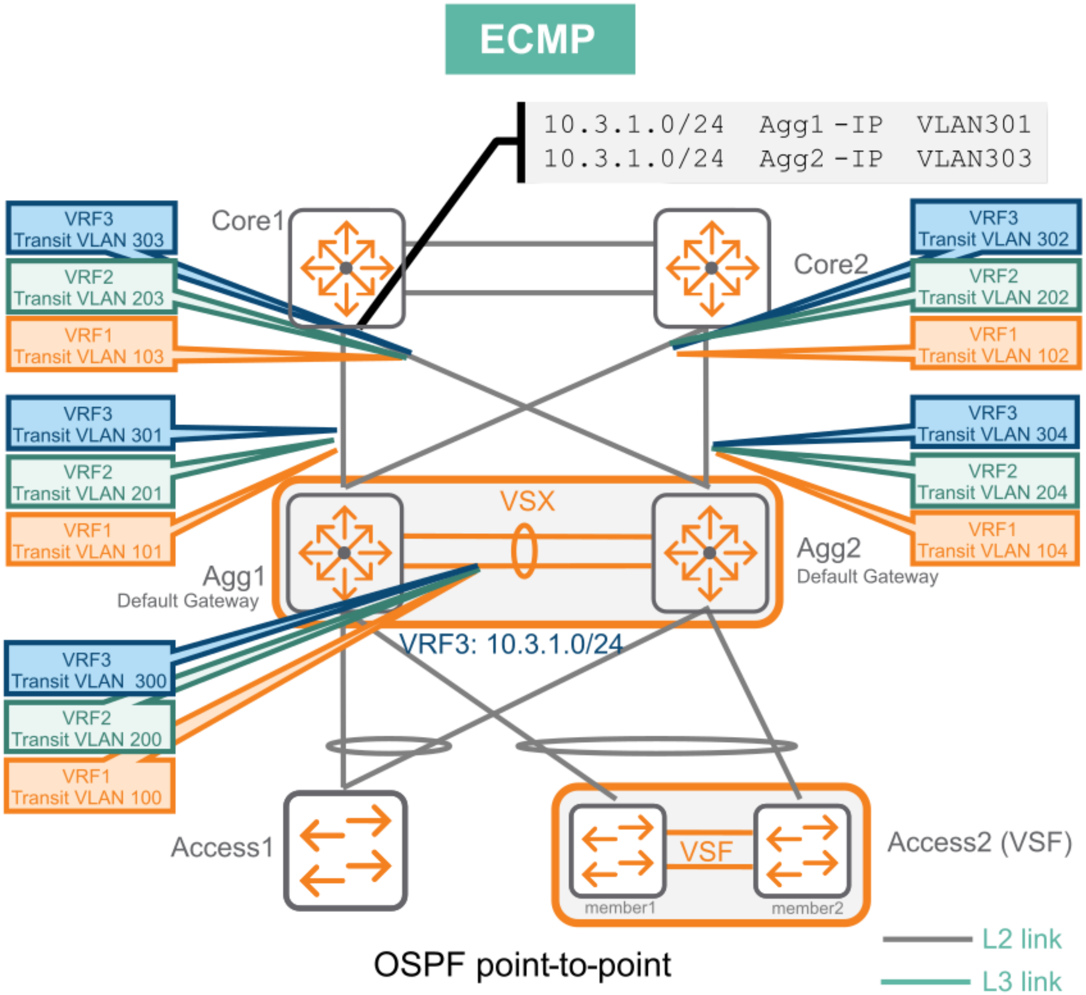
Upstream routing over VSX LAG SVI links

This section shows two configurations for upstream routing over VSX LAG SVI links:

- ECMP
- ECMP and VSX LAG
- Active gateway as next-hop router

The ECMP and VSX LAG configuration is the preferred configuration because LAGs introduce simplicity by reducing the number of transit VLANs and associated SVIs. This simplified configuration results in a minimized Sender Policy Framework (SPF) calculation time. The following figure shows that Core1 and Core2 are not in a VSX LAG, but Agg1 and Agg2 are in a VSX LAG. This figure introduces the requirement for MSTP because all the links between the aggregate and core are bridged (trunk ports with multiple VLANs).

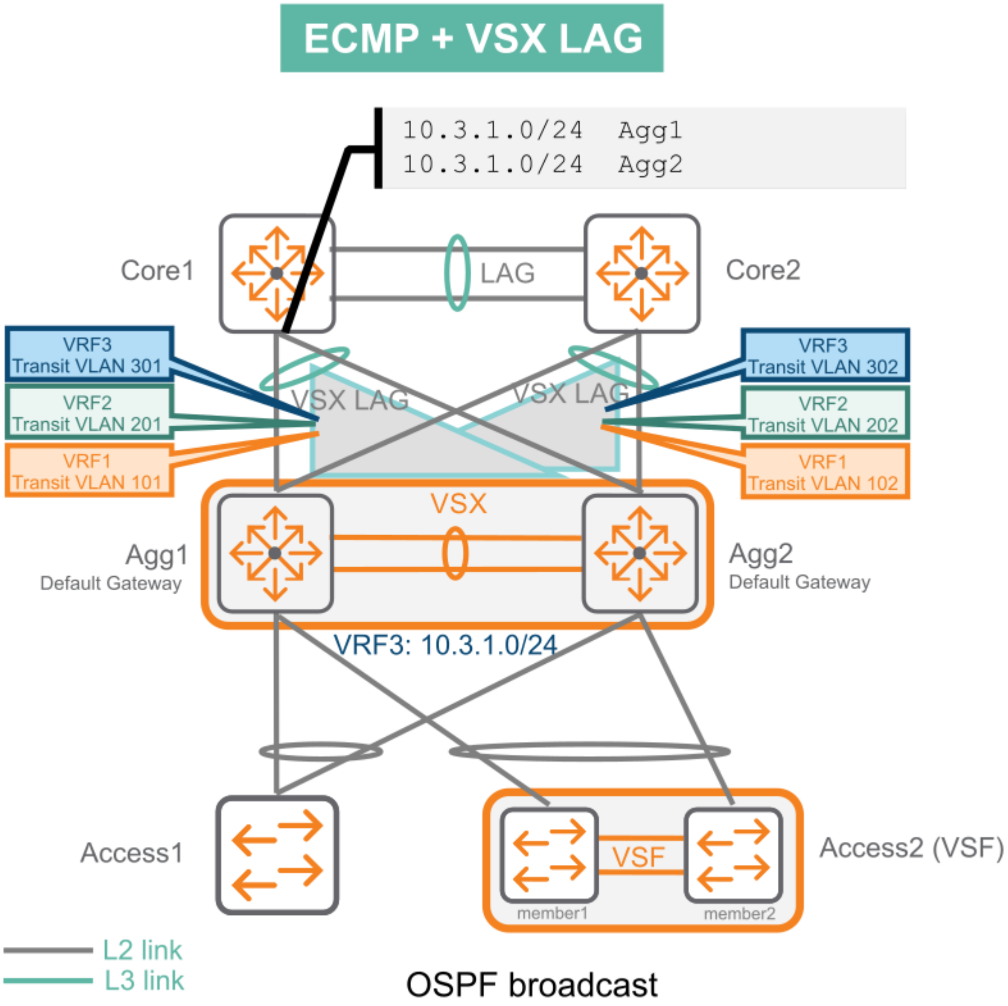
Figure 20: *ECMP in a VSX environment*



```
10.3.1.0/24   Agg1-IP   VLAN301
10.3.1.0/24   Agg2-IP   VLAN303
```

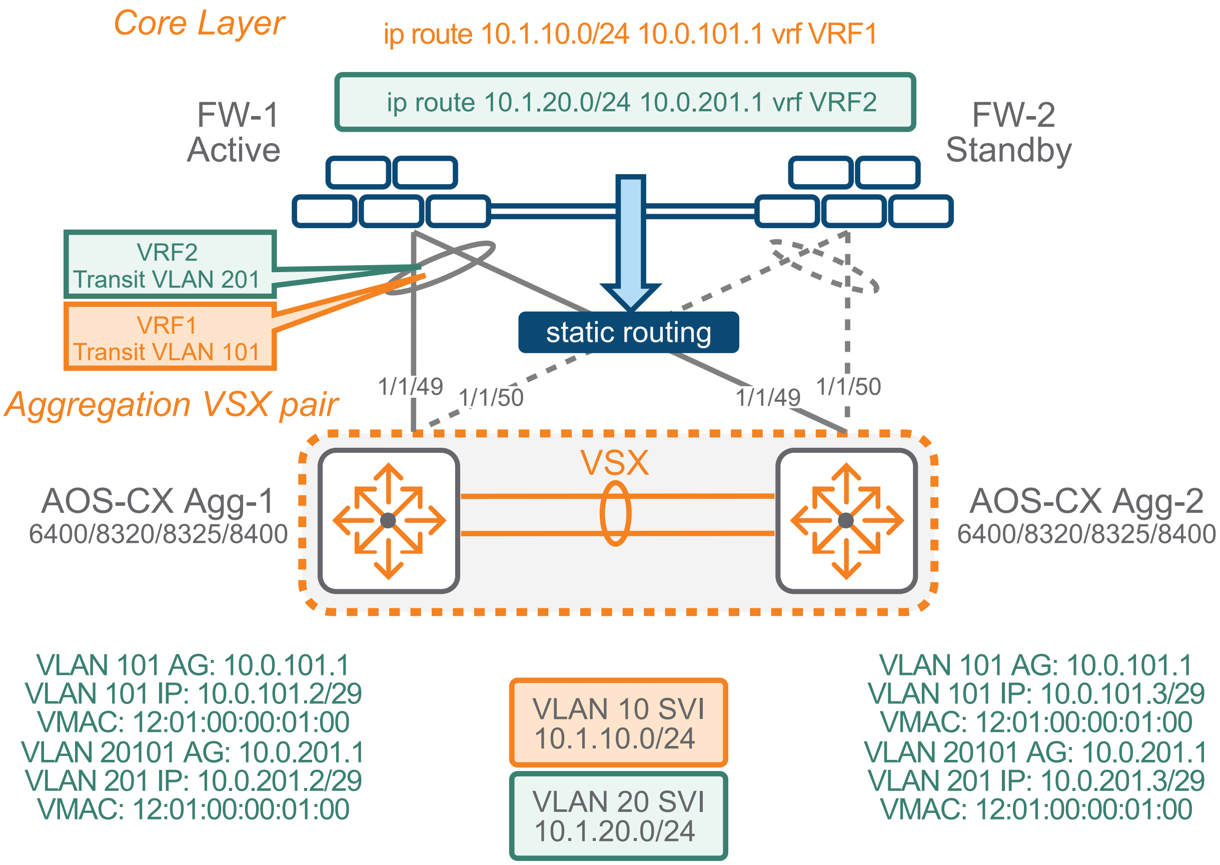
The following figure differs from the previous figure in that Core1 and Core2 are in a VSX LAG, which provides load balancing for ECMP. The transit VLANs shown in the following figure are per VRF.

Figure 21: *ECMP and VSX LAG in a VSX environment*



If ECMP is not supported or firewall does not support dynamic routing protocols, active gateway can be used as next-hop router. The following figure shows the specific use case of active/standby firewall with active gateway as the next-hop router.

Figure 22: *Active gateway as a next-hop router*



active-gateway

Syntax

```
active-gateway {ip | ipv6} [<IP-ADDRESS>] [mac <MAC-ADDRESS>]

no active-gateway {ip | ipv6} [<IP-ADDRESS>] [mac]
```

Description

Configures a virtual IP and virtual MAC for an interface VLAN

The `no` form of this command removes the active gateway for active-active routing.

Command context

`config-if-vlan`

Parameters

`ip`

Specifies the configuration of an IPv4 address.

`ipv6`

Specifies the configuration of an IPv6 address.

`<IP-ADDRESS>`

Specifies the IPv4 or IPv6 address.

- Syntax for IPv4: `A.B.C.D`
- Syntax for IPv6: `A:B::C:D`

`<MAC-ADDR>`

Specifies the Virtual MAC address. Syntax: `xx:xx:xx:xx:xx:xx`

Authority

Administrators or local user group members with execution rights for this command.

Usage

Before configuring active gateway, confirm that an IP address is on the SVI that is in the same subnet as the active gateway IP you are trying to configure. If an active gateway IP does not have an SVI IP with the same subnet, the CLI allows the configuration, but the active gateway IP will not be programmed in the kernel, resulting the active gateway to be unreachable.

Active forwarding cannot be configured when ICMP redirect is enabled. Enter the `no ip icmp redirect` command for disabling ICMP redirect.

It is highly recommended that you use an IPv6 link-local address as a gateway (VIP) on the active gateway IPv6 configuration.

If VRRP or active forwarding is configured on an SVI, active gateway cannot be configured. Active gateway with overlapping networks is not allowed. Maximum of 16 unique virtual MACs are supported in a system.

The maximum number of supported active gateways per switch is 4,000. Since a maximum of 31 secondary IPv4 addresses can be configured on an SVI, 32 IPv4 active gateways (along with the primary IPv4 address) can be configured per SVI with IP multinetting support. This support is also the same for IPv6 addresses.

NOTE: Do not use peer system MAC address as an active-gateway VMAC. If same MAC address is used, the VSX synchronization will try to sync the configuration on secondary switch and cause traffic disruptions.

Examples

Configuring active-gateway when the IP address is different from the SVI IP address on both VSX peers (valid for IPv6 and IPv4):

Switch 1:

```
switch1(config-if-vlan)# ip address 192.168.1.250/24
switch1(config-if-vlan)# active-gateway ip 192.168.1.253 mac 00:00:00:00:00:01
switch1(config-if-vlan)# active-gateway ipv6 fe80::01 mac 00:00:00:01:00:01
```

Switch 2:

```
switch2(config-if-vlan)# ip address 192.168.1.251/24
switch2(config-if-vlan)# active-gateway ip 192.168.1.253 mac 00:00:00:00:00:01
switch2(config-if-vlan)# active-gateway ipv6 fe80::01 mac 00:00:00:01:00:01
```

Configuring active-gateway when the IP address is the same as the SVI IP address on both VSX peers (valid for IPv4 only):

Switch 1:

```
switch1(config-if-vlan)# ip address 192.168.1.250/24
switch(config-if-vlan)# active-gateway ip 192.168.1.250 mac 00:00:00:00:00:01
```

Switch 2:

```
switch2(config-if-vlan)# ip address 192.168.1.250/24
switch2(config-if-vlan)# active-gateway ip 192.168.1.250 mac 00:00:00:00:00:01
```

Configuring only the active gateway address:

```
switch(config-if-vlan)# ip address 192.168.1.250/24
switch(config-if-vlan)# active-gateway ip 192.168.1.250
```

Configuring only the active gateway IP MAC address:

```
switch2(config-if-vlan)# ip address 192.168.1.250/24
switch2(config-if-vlan)# active-gateway ip mac 00:00:00:01:00:01
```

Removing the active gateway for active-active routing (IPv6 and IPv4):

```
switch(config-if-vlan)# no active-gateway ip
switch(config-if-vlan)# no active-gateway ipv6
```

Removing the active gateway for active-active routing for an IP address:

```
switch(config-if-vlan)# no active-gateway ip 192.168.1.250
```

Removing the active gateway for active-active routing for virtual MAC addresses:

```
switch(config-if-vlan)# no active-gateway ip mac
```

config-sync disable

Syntax

```
config-sync disable
```

```
no config-sync disable
```

Description

Pauses VSX synchronization.

The `no` form of this command restarts VSX synchronization.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Pauses VSX configuration synchronization:

```
switch(config)# vsx
switch(config-vsx)# config-sync disable
```

Enables the VSX configuration synchronization:

```
switch(config)# vsx
switch(config-vsx)# no config-sync disable
```

inter-switch-link {*<PORT-NUM>* | lag *<LAG-ID>*}

Syntax

```
inter-switch-link {<PORT-NUM> | lag <LAG-ID>}
```

```
no inter-switch-link
```

Description

Configures a physical port or a LAG as an interswitch link port. Only one port or LAG can be configured to act as an ISL. Once a port is configured as an ISL, it becomes a part of all VLANs in a system.

The `no` form of this command clears the configuration of the interswitch link port from a physical port or a LAG.

Command context

```
config-vsx
```

Parameters

<PORT-NUM>

Specifies a physical port on the switch. Use the format `member/slot/port` (for example, `1/3/1`). Sets the port to act as ISL

<LAG-ID>

Specifies the LAG ID. Run the `show capacities` command for the maximum number of VSX LAGs supported for your particular type of switch.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring port 1/1/1 as an interswitch link port:

```
switch(config-vsx)# inter-switch-link 1/1/1
```

Configuring LAG 100 as an interswitch link port:

```
switch(config-vsx)# inter-switch-link lag 100
```

Clears the interswitch link port:

```
switch(config-vsx)# no inter-switch-link
```

inter-switch-link dead-interval

Syntax

```
inter-switch-link dead-interval <DEAD-INTERVAL>
```

```
no inter-switch-link dead-interval
```

Description

Sets the dead interval for the interswitch link protocol. The dead interval is the amount of time to wait for hellos from a peer before declaring the peer to be dead. The default dead interval time is 20 seconds.

The `no` form of this command resets the interswitch link dead interval to the default of 20 seconds.

Command context

`config-vsx`

Parameters

<DEAD-INTERVAL>

Specifies the dead interval in seconds. Required. Range: 2 to 20 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the dead interval for the interswitch link protocol to 10 seconds:

```
switch(config)# vsx
switch(config-vsx)# inter-switch-link dead-interval 10
```

Setting the dead interval for the interswitch link protocol to the default:

```
switch(config)# vsx
switch(config-vsx)# no vsx inter-switch-link dead-interval
```

inter-switch-link hello-interval

Syntax

`inter-switch-link hello-interval <HELLO-INTERVAL>`

`no inter-switch-link hello-interval`

Description

Configures the interswitch link hello-interval. The hello interval determines the frequency of a hello packet exchange to confirm the control plane of the peer is alive. The default hello-interval is 1 second.

The `no` form of this command sets the interswitch link hello-interval to the default of 1 second.

Command context

`config-vsx`

Parameters

<HELLO-INTERVAL>

Specifies hello interval in seconds. Range: 1 to 5 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring the interswitch link hello-interval to 3 seconds:

```
switch(config)# vsx
switch(config-vsx)# inter-switch-link hello-interval 3
```

Resetting the interswitch link hello-interval to the default of 1 second:

```
switch(config)# vsx
switch(config-vsx)# no inter-switch-link hello-interval
```

inter-switch-link hold-time

Syntax

`inter-switch-link hold-time <HOLD-INTERVAL>`

`no inter-switch-link hold-time`

Description

Sets the holdtime for the interswitch link protocol. A port is treated as down only when it stays down for the configured holdtime interval. The default holdtime is 0 seconds.

The no form of this command sets the interswitch link protocol holdtime to the default of 0 seconds.

Command context

config-vsx

Parameters

<HOLD-INTERVAL>

Specifies the hold interval in seconds. Required. Range: 0 to 3 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the holdtime for interswitch link protocol to 2 seconds:

```
switch(config)# vsx
switch(config-vsx)# inter-switch-link hold-time 2
```

Setting the interswitch link protocol holdtime to the default of 0 seconds:

```
switch(config)# vsx
switch(config-vsx)# no inter-switch-link hold-time
```

inter-switch-link peer-detect-interval

Syntax

inter-switch-link peer-detect-interval *<PEER-DETECT-INTERVAL>*

no inter-switch-link peer-detect-interval

Description

Sets the amount of time in seconds that the VSX switch waits for the ISL interface to link up after a reboot. If the ISL link does not come up within this time window, the VSX switch declares itself as split from its peer. The default peer detect interval is 300 seconds.

The no form of this command sets the interswitch link protocol peer detect interval to the default of 300 seconds.

Command context

config-vsx

Parameters

<PEER-DETECT-INTERVAL>

Specifies the peer detect interval in seconds. Required. Range: 60 to 600 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Usage

After a VSX switch reboots, the switch waits 5 minutes by default to receive a hello packet before it declares itself to be out-of-sync. The inter-switch-link peer-detect-interval *<PEER-DETECT-INTERVAL>*

command lets you change how long the switch waits to receive the hello packet before the switch declares itself to be out-of-sync.

Examples

Setting the peer detect interval to 180 seconds:

```
switch(config)# vsx
switch(config-vsx)# inter-switch-link peer-detect-interval 180
```

Restoring the peer detect interval to the default (300 seconds):

```
switch(config)# vsx
switch(config-vsx)# no inter-switch-link peer-detect-interval
```

interface lag multi-chassis

Syntax

```
interface lag <LAG-ID> multi-chassis [static]

no interface lag <LAG-ID>
```

Description

Configures a given LAG as a dynamic multichassis LAG (VSX LAG), which supports a maximum of four member links per switch segment. A VSX LAG across a downstream switch can have at most a total of eight member links.

The `no` form of this command removes a VSX LAG.

Command context

`config`

Parameters

<LAG-ID>

Specifies the LAG ID. Run the `show capacities vsx` command for the maximum number of VSX LAGs supported for your particular type of switch; however, the maximum VSX LAG value considers that one port is used for the ISL, which is not a VSX LAG. Required.

static

Specifies the multichassis LAG as static. Optional.

Authority

Administrators or local user group members with execution rights for this command.

Usage

A VSX LAG across a VSX pair can have at most a total of eight interfaces.

NOTE:

- When creating a VSX LAG, select an equal number of member links in each segment for load balancing, such as four member links (one segment) and four member links (another segment). Do not create a VSX LAG with four member links in one switch and two member links on another segment. A switch can have a maximum of four member links.
- Make sure that the VSX LAG interface on both the VSX primary and secondary switches has a member port configured and enabled.
- Make sure that you also have a non-VSX port that is available for the ISL.

You cannot change the mode of a multichassis LAG without removing the multichassis LAG first. To change a pre-existing VSX LAG to a static VSX LAG, first remove the VSX LAG with the `no interface lag <LAG-ID>` command. Then, enter the `interface lag <LAG-ID> multi-chassis static` command.

Examples

Configuring LAG 100 as a VSX LAG:

```
switch(config)# interface lag 100 multi-chassis
```

Removing LAG 100 as a VSX LAG:

```
switch(config)# no interface lag 100
```

Specifying LAG 100 as a static VSX LAG:

```
switch(config)# interface lag 100 multi-chassis static
```

More information

Sample configurations for MSTP on VSX

ip icmp redirect

Syntax

`ip icmp redirect`

`no ip icmp redirect`

Description

Enables the sending of ICMPv4 and ICMPv6 redirect messages to the source host. Enabled by default.

The `no` form of this command disables ICMPv4 and ICMPv6 redirect messages to the source host.

Command context

`config`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling ICMP redirect messages:

```
switch(config)# ip icmp redirect
```

Disabling ICMP redirect messages:

```
switch(config)# no ip icmp redirect
```

keepalive dead-interval

Syntax

`keepalive dead-interval <DEAD-INTERVAL>`

`no keepalive dead-interval`

Description

Sets the dead-interval for keepalive protocol. The dead interval is the amount of time to wait for hellos from a peer before declaring the peer to be dead. The default dead-interval is 3 seconds.

The `no` form of this command sets the interswitch link dead-interval to the default of 3 seconds.

Command context

`config-vsx`

Parameters

`dead-interval <DEAD-INTERVAL>`

Specifies the dead-interval in seconds. Range: 2 to 20 seconds

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the dead-interval for keepalive protocol to 10 seconds:

```
switch(config)# vsx
switch(config-vsx)# keepalive dead-interval 10
```

Setting the dead-interval for keepalive protocol to the default:

```
switch(config)# vsx
switch(config-vsx)# no keepalive dead-interval
```

keepalive hello-interval

Syntax

`keepalive hello-interval <HELLO-INTERVAL>`

`no keepalive hello-interval`

Description

Sets the hello-interval for keepalive protocol. The hello interval determines the frequency of a hello packet exchange to confirm the peer is alive. The default hello-interval is 1 second.

The `no` form of this command sets the hello-interval for keepalive protocol to the default of 1 second.

Command context

`config-vsx`

Parameters

`hello-interval <HELLO-INTERVAL>`

Specifies the hello-interval in seconds. Range: 1 to 5 seconds

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the hello-interval for keepalive protocol to 3 seconds:

```
switch(config)# vsx
switch(config-vsx)# keepalive hello-interval 3
```

Resetting the hello-interval for keepalive protocol to the default:

```
switch(config)# vsx
switch(config-vsx)# no keepalive hello-interval
```

keepalive peer

Syntax

`keepalive peer <PEER-IP-ADDR> source <SOURCE-IP-ADDR> [vrf <VRF-NAME>]`

`no keepalive`

Description

Sets the source and peer IP addresses for keepalive packets in a specified VRF. If a VRF is not specified, it sets to the default VRF.

The `no` form of this command removes the source and peer IP addresses and VRF for the keepalive protocol. VSX continues to work.

Command context

`config-vsx`

Parameters

`peer <PEER-IP-ADDR>`

Specifies the peer IPv4 address. Syntax: A.B.C.D

`source <IP-ADDR>`

Specifies the source IPv4 address. The source IP address is the IP address assigned to the keepalive interface on the switch. For example, if you are entering this command on the primary switch, the source IP address would be the IP address assigned to the keepalive interface on the primary switch. Syntax: A.B.C.D

`vrf <VRF-NAME>`

Specifies the VRF name. If you are entering this command on the primary switch, the peer IP address is the IP address assigned to the keepalive interface for the secondary switch. If you are entering this command on the secondary switch, the peer IP address is the IP address assigned to the keepalive interface for the primary switch. Syntax: String

Authority

Administrators or local user group members with execution rights for this command.

Usage

To configure the keepalive feature, enter this command once on the primary switch and once on the secondary switch. The keepalive feature is recommended for redundancy. If the ISL link goes down, the keepalive connection keeps the traffic moving so that the peer and secondary switches can continue to communicate. The keepalive connection is established over a routed network, and it does not have to be a dedicated peer-to-peer link unlike ISL.

Examples

Setting the source and peer IP addresses for keepalive in the default VRF:

```
switch(config)# vsx
switch(config-vsx)# keepalive peer 192.168.1.1 source 192.168.1.5
```

Setting the source and peer IP addresses for keepalive in the vrf1:

```
switch(config)# vsx
switch(config-vsx)# keepalive peer 10.0.0.1 source 10.0.0.2 vrf vrf1
```

Removing the source and peer IP addresses and VRF for the keepalive protocol:

```
switch(config)# vsx
switch(config-vsx)# no keepalive
```

keepalive udp-port

Syntax

`keepalive udp-port <PORT-NUM>`

`no keepalive udp-port`

Description

Sets the UDP port for the keepalive protocol.

The `no` form of this command sets the UDP port for keepalive protocol to the default of 7678.

Command context

`config-vsx`

Parameters

`udp-port <PORT-NUM>`

Specifies UDP port number. Range: 1024-65535

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the UDP port for keepalive protocol to 2000:

```
switch(config)# vsx
switch(config-vsx)# keepalive udp-port 2000
```

Setting the UDP port for keepalive protocol to the default of 7678:

```
switch(config)# vsx
switch(config-vsx)# no keepalive udp-port
```

lacp fallback

Syntax

`lacp fallback`

`no lacp fallback`

Description

Sets LACP fallback on a VSX LAG port. When no LACP partner is detected, the VSX LAG port makes members of the VSX LAG function as nonbonded interfaces. To create a VSX LAG, use the `interface lag multi-chassis` command.

The `no` form of this command sets the VSX LAG to a block state when no LACP partner is detected.

Command context

`config-lag-if`

Authority

Administrators or local user group members with execution rights for this command.

Usage

LACP fallback is supported only when there is a single link from the downstream or peer device to each VSX node.

WARNING: Even though this command appears to be accepted on a standard/non-VSX LAG, the fallback feature works only on a VSX LAG (multichassis LAG) interface.

Examples

Enabling LACP fallback:

```
switch(config)# interface lag 1
switch(config-lag-if)# lacp fallback
```

Disables LACP fallback:

```
switch(config)# interface lag 1
switch(config-lag-if)# no lacp fallback
```

linkup-delay-timer

Syntax

```
linkup-delay-timer <DELAY-TIMER>
```

```
no linkup-delay-timer
```

Description

Configures the VSX link-up delay timer. The VSX delay timer feature lets you configure the delay timer, which delays bringing downstream VSX links up, following a VSX device reboot or an ISL flap.

The `no` form of this command restores the VSX link-up delay timer to a default of 180 seconds.

Command context

```
config-vsx
```

Parameters

<DELAY-TIMER>

Specifies the VSX LAG bring-up delay in seconds. Range: 0 to 600 seconds

Authority

Administrators or local user group members with execution rights for this command.

Usage

The recommended delay timer setting is determined by the number of MAC addresses, ARPv4, and routes. The link-up delay timer might need to be set to a higher value for larger networks, depending on the ARP and routing table size.

Table 4: *Recommended delay timer settings for 832x series switches*

MAC	ARPv4	Routes	Recommended delay timer setting
16K	16K	10K	120
32K	32K	10K	120
47K	47K	10K	150
47K	47K	10K	250
47K	47K	10K	250
47K	69K	10K OSPF	420

Table 5: *Recommended delay timer settings for 8400 series switches*

MAC	ARPv4	Routes	Recommended delay timer setting
40K	40K	512	300
32K	32K	512	180
48K	48K	512	480
48K	48K	20K IPv4 + 20K IPV6	600
48K	48K	10K IPv4 + 10K IPv6	480
32K	-	10K IPv4	180

Examples

Setting the VSX link-up delay timer to 35 seconds:

```
switch(config)# vsx
switch(config-vsx)# linkup-delay-timer 35
```

Setting the VSX link-up delay timer to the default:

```
switch(config)# vsx
switch(config-vsx)# no linkup-delay-timer
```

linkup-delay-timer exclude lag-list

Syntax

```
linkup-delay-timer exclude lag-list <LAG-LIST>
```

```
no linkup-delay-timer exclude lag-list <LAG-LIST>
```

Description

Configures the VSX link-up delay timer exclude list. It excludes the bringing up of specified downstream VSX LAGs, following a device reboot or an ISL flap.

The `no` form of this command unconfigures the VSX link-up delay timer exclude list.

Command context

`config-vsx`

Parameters

<LAG-LIST>

Specifies a range or a set of LAG interfaces to exclude. For example: `1` or `1-10` or `1,2,3` or `1,2-10`. Range: 1-128 characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Specifying LAGs to exclude LAG 100:

```
switch(config)# vsx
switch(config-vsx)# linkup-delay-timer exclude lag-list 100
```

Unconfiguring the VSX link-up delay timer exclude list for LAG 100:

```
switch(config)# vsx
switch(config-vsx)# no linkup-delay-timer exclude lag-list 100
```

neighbor <IP-ADDRESS> vsx-sync-exclude

Syntax

`neighbor <IP-ADDRESS> vsx-sync-exclude`

Description

Excludes VSX sync for the BGP neighbor.

Command context

`config-bgp`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Excluding VSX sync for the BGP neighbor:

```
switch(config-bgp)#  neighbor 1.1.1.1 vsx-sync-exclude
switch#
```

role {primary | secondary}

Syntax

`role {primary | secondary}`

`no role`

Description

Configures the VSX device role.

The `no` form of this command removes the device role of the switch in VSX and causes the interswitch link to be out-of-sync.

Command context

`config-vsx`

Parameters

`{primary | secondary}`

Selects the VSX role to either primary or secondary for the device.

Authority

Administrators or local user group members with execution rights for this command.

Usage

VSX has no default role defined for the device. The device role assigns the device as the primary or secondary for VSX synchronization. For ISL to be in-sync, one device in VSX must be configured as the primary and the other device must be configured as the secondary.

Examples

Setting the VSX role to primary:

```
switch(config)# vsx
switch(config-vsx)# role primary
```

Removing the device role:

```
switch(config)# vsx
switch(config-vsx)# no role
```

show active-gateway

Syntax

```
show active-gateway [vsx-peer]
```

Description

Displays the gateway information configured on SVIs, such as:

- Number of active-gateway interface VLANs
- Number of IPv4 active-gateway interface VLANs
- Number of IPv6 active-gateway interface VLANs
- Per virtual MAC address
  - IPv4 reference count and its interface VLANs
  - IPv6 reference count and its interface VLANs

Command context

Operator (>) or Manager (#)

Parameters

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
primary# show active-gateway
Number of active-gateway interface VLANs           : 265
Number of IPv4 active-gateway interface VLANs      : 264
Number of IPv6 active-gateway interface VLANs      : 1
VMAC 00:00:00:01:01:16 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan192-223
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:11 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan32-63
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:17 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan224-255
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:18 :
     IPv4 ref count        : 6
     IPv4 interface VLANs  : vlan256-259,300-301
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:13 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan96-127
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:12 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan64-95
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:20 :
     IPv4 ref count        : 1
     IPv4 interface VLANs  : vlan4040
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:14 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan128-159
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:10 :
     IPv4 ref count        : 31
     IPv4 interface VLANs  : vlan1-31
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:01:01:15 :
     IPv4 ref count        : 32
     IPv4 interface VLANs  : vlan160-191
     IPv6 ref count        : 0
     IPv6 interface VLANs  : none
VMAC 00:00:00:03:00:12 :
     IPv4 ref count        : 1
     IPv4 interface VLANs  : vlan2000
     IPv6 ref count        : 1
     IPv6 interface VLANs  : vlan4000
VMAC 00:00:00:01:01:19 :
     IPv4 ref count        : 1
     IPv4 interface VLANs  : vlan4000
     IPv6 ref count          : 0
     IPv6 interface VLANs    : none
```

show active-gateway <*IFNAME*>

Syntax

`show active-gateway <IFNAME> [vsx-peer]`

Description

Displays the gateway information per SVI, such as:

- Active-Gateway IPV4 and its MAC address
- Active-Gateway IPV6 and its MAC address

Command context

Operator (>) or Manager (#)

Parameters

<IFNAME>

Specifies the VSX interface name.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show active-gateway vlan2000
Active-gateway IPv4 MAC address           : 00:00:00:01:01:18
Active-gateway IPv4 address
    173.6.1.10
 173.7.1.10
Active-gateway IPv6 MAC address           : 00:00:00:03:00:12
Active-gateway IPv6 address
    173::2
 173::3
```

show interface <VLAN-NAME>

Syntax

`show interface <VLAN-NAME> [vsx-peer]`

Description

Displays a virtual IPv4/IPv6 and MAC configured for active-active routing.

Command context

Operator (>) or Manager (#)

Parameters

<VLAN-NAME>

Specifies the VLAN name. Syntax: string

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show int vlan100

Interface vlan100 is up
 Admin state is up
 Hardware: Ethernet, MAC Address: 48:0f:cf:af:c1:9e
 IPv4 address 192.168.1.1/24
 active gateway 192.168.1.253 00:00:00:00:00:01
 active gateway fe80::01 00:00:00:01:00:01
 RX
       L3:
            ucast: 0 packets, 0 bytes
            mcast: 8 packets, 812 bytes
 TX
       L3:
            ucast: 2 packets, 80 bytes
            mcast: 0 packets, 0 byte
```

show lacp aggregates

Syntax

`show lacp aggregates [<LAG-NAME>] [vsx-peer]`

Description

Displays a specified LAG or all configured LAGs along with VSX LAGs.

Command context

Operator (>) or Manager (#)

Parameters

<LAG-NAME>

Specifies the LAG name. Optional. Syntax: string

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying all configured LAGs along with VSX LAGs:

```
switch# show lacp aggregates

Aggregate name    : lag100 (multi-chassis)
Interfaces        : 1/1/44
Peer interfaces   : 1/1/44
Heartbeat rate    : Slow
Hash              : l3-src-dst
Aggregate mode    : Active
```

Displaying a specified LAG:

```
switch# show lacp aggregates lag100

Aggregate name    : lag100 (multi-chassis)
Interfaces        : 1/1/44
Peer interfaces   : 1/1/44
Heartbeat rate    : Slow
Hash              : l3-src-dst
Aggregate mode    : Active
```

show lacp interfaces

Syntax

`show lacp interfaces [<IFNAME>] [vsx-peer]`

Description

Displays an LACP configuration of the physical interfaces, including VSXs. If an interface name is passed as argument, it only displays an LACP configuration of a specified interface.

Command context

Operator (>) or Manager (#)

Parameters

<IFNAME>

Optional: Specifies an interface name.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

This example displays an LACP configuration of the physical interfaces. One of the interfaces has the `lacp-block` forwarding state. If a VSX switch has loop protect enabled on an interface and a loop occurs, VSX blocks the interface to stop the loop. The forwarding state of the blocked interface is set to `lacp-block`.

```
switch# show lacp interfaces
State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------------
Intf   Aggr    Port    Port   State   System-id          System  Aggr Forwarding
       name    id      Pri                               Pri     Key  State
------------------------------------------------------------------------------------
1/1/1  lag10   17      1      ALFOE   70:72:cf:37:a3:5c  20      10   lacp-block
1/1/2  lag128  69      1      ALFNCD  70:72:cf:37:a3:5c  20      128  up
1/1/3  lag128  14      1      ALFNCD  70:72:cf:37:a3:5c  20      128  up
1/1/4  lag128                                                         down
1/1/5  lag20                                                          up

Partner details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggr    Partner Port     State     System-id       System   Aggr
       name    Port-id Pri                                Priority Key
------------------------------------------------------------------------------
1/1/1  lag10   0       65534    PLFOEX  00:00:00:00:00:00 65534    0
1/1/2  lag128  69      1        PLFNCD  70:72:cf:8c:60:a7 65534    128
1/1/3  lag128  14      1        PLFNCD  70:72:cf:8c:60:a7 65534    128
1/1/4  lag128
1/1/5  lag20
```

Displaying static LAG:

```
switch# show lacp interfaces
State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggr   Port    Port   State   System-id          System  Aggr Forwarding
       Name   Id      Pri                               Pri     Key  State
------------------------------------------------------------------------------
1/1/1  lag10                                                         up
1/1/2  lag10                                                         up

Partner details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggr   Port    Port   State   System-id          System  Aggr
       Name   Id      Pri                               Pri     Key
------------------------------------------------------------------------------
1/1/1  lag10
1/1/2  lag10
```

Displaying an LACP configuration of the 1/1/1 interface:

```
switch# show lacp interfaces 1/1/1

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state


Aggregate-name : lag1
-------------------------------------------------
                       Actor            Partner
-------------------------------------------------
Port-id            | 28                 | 31
Port-priority      | 1                  | 1
Key                | 1                  | 1
State              | ALFNCD             | ALFNCD
System-id          | 98:f2:b3:68:40:a0  | 98:f2:b3:68:60:a6
System-priority    | 65534              | 65534
```

Displaying an LACP configuration after loop-protect is enabled on the primary VSX switch:

```
switch# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
1/4/14  lag1(mc)   206   1     ALFNCD  f8:60:f0:06:49:00 65534  1    up
1/5/15  lag2(mc)                                                     down


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/4/14  lag1(mc)   130   1     ALFNCD  f8:60:f0:06:87:00 65534  1
1/5/15  lag2(mc)
```

Displaying an LACP configuration after loop-protect is enabled on the secondary VSX switch:

```
switch# show lacp interfaces

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr Forwarding
        Name       Id    Pri                             Pri    Key  State
------------------------------------------------------------------------------
```

```
1/3/2   lag1(mc)   1130  1     ALFNCD  f8:60:f0:06:49:00 65534  1    up
1/9/3   lag2(mc)                                                     down


Partner details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggr       Port  Port  State   System-ID         System Aggr
        Name       Id    Pri                             Pri    Key
------------------------------------------------------------------------------
1/3/2   lag1(mc)   131   1     ALFNCD  f8:60:f0:06:87:00 65534  1
1/9/3   lag2(mc)
```

show lacp interfaces multi-chassis

Syntax

```
show lacp interfaces multi-chassis [<IFNAME>] [vsx-peer]
```

Description

Shows all configured VSX remote interface details. The interface that has the ALFNCD status has been synced with the partner and is ready for flow distribution.

Command context

Operator (>) or Manager (#)

Parameters

<IFNAME>

Specifies the VSX interface name. Optional.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show lacp interfaces multi-chassis

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

 Actor details of all interfaces:
------------------------------------------------------------------------------
Intf    Aggregate  Port    Port     State   System-ID         System   Aggr
         name       id      Priority                           Priority Key
 ------------------------------------------------------------------------------
1/1/2   lag100(mc) 2       1        ALFNCD  08:00:09:13:06:7c 65534    100


 Partner details of all interfaces:
```

```
------------------------------------------------------------------------------
Intf    Aggregate  Partner Port     State   System-ID         System   Aggr
         name       Port-id Priority                           Priority Key
 ------------------------------------------------------------------------------
1/1/2   lag100(mc) 2       1        ALFNCD  08:00:09:05:24:f6 65534    10


 Remote Actor details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggregate  Port    Port     State   System-ID         System   Aggr
        name       id      Priority                           Priority Key
 ------------------------------------------------------------------------------
1/1/2  lag100(mc) 1002    1        ALFNCD  08:00:09:13:06:7c 65534    100


 Remote Partner details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggregate  Partner Port     State   System-ID         System   Aggr
        name       Port-id Priority                           Priority Key
 ------------------------------------------------------------------------------
1/1/2  lag100(mc) 3       1        ALFNCD  08:00:09:05:24:f6 65534    10
```

show running-config interface

Syntax

`show running-config interface`

Description

Displays all configured interface commands, including VSX commands.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show running-config interface
interface lag 100 multi-chassis
    no shutdown
    no routing
    lacp mode active
interface 1/1/1
    no shutdown
    no routing
interface 1/1/2
    no shutdown
    lag 100
interface 1/1/3
    no shutdown
    ip address 192.168.1.2/24
interface vlan100
    no shutdown
    ip address 192.168.1.1/24
    active-gateway ip 192.168.1.253 mac 00:00:00:00:00:01
    active-gateway ipv6 fe80::01 mac 00:00:00:01:00:01
```

show running-config vsx

Syntax

`show running-config vsx`

Description

Displays the configured VSX commands.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show running-config vsx
vsx
    system-mac 10:00:00:00:00:01
    inter-switch-link hello-interval 2
    inter-switch-link dead-interval 3
    inter-switch-link hold-time 3
    inter-switch-link peer-detect-interval 300
    role primary
    keepalive udp-port 1500
    keepalive hello-interval 2
    keepalive dead-interval 4
    keepalive peer 192.168.1.1 source 192.168.1.2
    inter-switch-link 1/1/43
interface lag 100 multi-chassis
   no shutdown
   no routing
   vlan access 1
   lacp mode active
interface 1/1/44
   no shutdown
   lag 100
```

show running-config vsx-sync

Syntax

`show running-config vsx-sync`

Description

Displays the lines of running-configuration that VSX configuration synchronization is enabled on. The command also provides a rolled-up view of configuration expected to be synced. This command can be run from the primary or secondary peer.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Displaying the running configuration on which VSX synchronization is enabled:

```
switch# show running-config vsx-sync
Current vsx-sync configuration:
vlan 3
    vsx-sync
access-list ip test1
    vsx-sync
    !
policy test2
    vsx-sync
    !
```

show running-config vsx-sync peer-diff

Syntax

```
show running-config vsx-sync peer-diff
```

Description

Displays the difference between the configuration of features enabled for VSX synchronization on the primary and secondary switches.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Usage

Use this command for diagnosing errors. This command provides visibility into which configuration lines did not synchronize from the primary peer to the secondary peer. This command can be run from the primary or secondary peer. The output is displayed in the GNU diff unified format.

Example

Displaying the running configuration on which VSX synchronization is enabled:

```
switch# show running-config vsx-sync peer-diff
--- /tmp/running-config-vsx.83e 2018-05-01 17:03:38.083281976 +0000
+++ /tmp/peer-running-config-vsx.83e    2018-05-01 17:03:38.077281976 +0000
@@ -1,4 +0,0 @@
-access-list ip sync
-    vsx-sync
-     !
-     10 permit any any any
```

show vsx active-forwarding

Syntax

`show vsx active-forwarding [interface <INTERFACE-VLAN>] [vsx-peer]`

Description

Shows all the VSX active-forwarding configured interface VLANs or the VSX active-forwarding peer information for a particular interface VLAN.

Command context

Operator (>) or Manager (#)

Parameters

interface *<INTERFACE-VLAN>*

Specifies the interface VLAN name. Syntax: string

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying a list of VSX active-forwarding enabled interfaces:

```
switch# show vsx active-forwarding
List of VSX active-forwarding enabled interfaces:
vlan30
vlan32
vlan33
```

Displaying the VSX active-forwarding peer information for `vlan30`:

```
switch# show vsx active-forwarding interface vlan30
Interface vlan30 has VSX active-forwarding enabled.
Interface vlan30 Peer Data:
Peer MAC: 94:f1:28:21:22:00
Peer IPv6 Addresses:
    fe80::96f1:28ff:fe21:2200
```

show vsx brief

Syntax

`show vsx brief [vsx-peer]`

Description

Displays the brief VSX status.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Usage

The `show vsx brief` command displays the ISLP device protocol states under the "Device State" heading.

Table 6: *ISLP device protocol states*

Device state	Definition
Peer-Established	The VSX switch is in a steady state. VSX LAGs are up.
Sync-Primary	ISL connectivity to the peer VSX switch is restored, and the VSX switch is syncing states to the peer VSX switch. VSX LAGs are up.
Sync-Secondary	ISL connectivity to the peer VSX switch is restored, and the VSX switch is learning states from the peer VSX switch. VSX LAGs are down.
Sync-Secondary-Linkup-Delay	The VSX switch has learned its states from the peer VSX switch, and the VSX switch is monitoring for hardware to be programmed. VSX LAGs are down.
Split-System-Primary	The VSX switch has lost ISL connectivity to the peer VSX switch. The VSX switch is operating as the primary VSX switch. VSX LAGs are up.
Split-System-Secondary	The VSX switch has lost ISL connectivity to the peer VSX switch. The VSX switch is operating as the secondary VSX switch. VSX LAGs are down.
Waiting-For-Peer	The VSX switch is waiting for connectivity to the peer VSX switch.

Example

Displaying the brief VSX status for the switch you are logged into:

```
vsx-primary# show vsx brief
ISL State                                    : In-Sync
Device State                                 : Peer-Established
Keepalive State                              : Keepalive-Established
```

```
Device Role                            : primary
Number of Multi-chassis LAG interfaces : 2
```

Displaying the brief VSX status for the peer (secondary) switch while entering the command on the primary switch:

```
vsx-primary# show vsx brief vsx-peer
ISL State                              : In-Sync
Device State                           : Peer-Established
Keepalive State                        : Keepalive-Established
Device Role                            : secondary
Number of Multi-chassis LAG interfaces : 2
```

Displaying the brief VSX status for the peer (primary) switch while entering the command on the secondary switch:

```
vsx-secondary# show vsx brief vsx-peer
ISL State                              : In-Sync
Device State                           : Peer-Established
Keepalive State                        : Keepalive-Established
Device Role                            : primary
Number of Multi-chassis LAG interfaces : 2
```

show vsx config-consistency

Syntax

```
show vsx config-consistency [vsx-peer]
```

Description

Displays the VSX global configuration consistency between two VSX switches.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user groups members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

The following example shows a comparison between the two VSX switches.

```
switch# show vsx config-consistency
Configurations                          Local                              Peer
------------------                      ------                             ------
Software Version                        XL.10.0x.xxxxAE                    XL.10.0x.xxxxAE
System MAC                              94:f1:28:ef:25:00                  f4:03:43:80:28:00
System Profile                          Advanced                           Advanced
ISL hello interval                      1                                  1
ISL dead interval                       20                                 20
ISL hold interval                       0                                  0
```

```
Keepalive hello interval                1                                 1
Keepalive dead interval                 3                                 3
Keepalive UDP port                      7678                              7678


VSX VLAN List
-------------
Local ISL VLANs : 1,100
Peer ISL VLANs  : 1,10

VSX Active Forwarding
---------------------
Interface VLANs      : 2, 5-9
Peer Interface VLANs : 2, 5-10
```

show vsx config-consistency lacp

Syntax

```
show vsx config-consistency lacp [<LAG-NAME>] [vsx-peer]
```

Description

Displays VSX LACP configuration consistency between two VSX switches.

Command context

Operator (>) or Manager (#)

Parameters

<LAG-NAME>

Specifies the LAG name. Optional. Syntax: string

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show vsx config-consistency lacp
Configurations                          Local                  Peer
------------------                      ------                 ------
Name                                    lag100                 lag100
Loop protect enabled                    false                  true
Hash scheme                             l2-src-dst-hash        l2-src-dst-hash
Qos cos override                        0                      0
Qos dscp override                       0                      0
Qos trust

VSX VLAN list
1
Peer VSX VLAN list
1,10

STP link-type                          point-to-point       point-to-point
STP port-type                          admin-network        admin-network
STP bpdu-filter                        Disabled             Disabled
STP bpdu-guard                         Disabled             Disabled
STP loop-guard                         Disabled             Disabled
STP root-guard                         Disabled             Disabled
STP tcn-guard                          Disabled             Disabled


Configurations                          Local                  Peer
------------------                      ------                 ------
Name                                    lag111                 lag111
Loop protect enabled                    false                  false
Hash scheme                             l2-src-dst-hash        l2-src-dst-hash
Qos cos override                        0                      0
Qos dscp override                       0                      0
Qos trust
VSX VLAN list
1
Peer VSX VLAN list
1

STP link-type                          point-to-point       point-to-point
STP port-type                          admin-network        admin-network
STP bpdu-filter                        Disabled             Disabled
STP bpdu-guard                         Disabled             Disabled
STP loop-guard                         Disabled             Disabled
STP root-guard                         Disabled             Disabled
STP tcn-guard                          Disabled             Disabled
------------------------------------------------------
```

show vsx configuration

Syntax

```
show vsx configuration {inter-switch-link | keepalive} [vsx-peer]
```

Description

Displays the ISL configuration or keepalive protocol configuration in VSX.

Command context

Operator (>) or Manager (#)

Parameters

{inter-switch-link | keepalive}

Selects `inter-switch-link` or `keepalive`.

inter-switch-link

Displays the ISL configuration in VSX.

keepalive

Displays the keepalive protocol configuration in VSX.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying the ISL configuration in VSX:

```
switch# show vsx configuration inter-switch-link
Inter Switch Link     : 1/1/43
Hello Interval        : 1 Seconds
Dead Interval         : 20 Seconds
Hold Time             : 0 Seconds
Peer detect interval : 300 Seconds
System MAC            : 10:00:00:00:00:01
Device Role           : primary
Multichassis LAGs     : lag100
```

Displaying the keepalive protocol configuration in VSX:

```
switch# show vsx configuration keepalive
Keepalive Interface      : 1/1/1
Keepalive VRF            : test1
Source IP Address        : 192.168.1.1
Peer IP Address          : 192.168.1.2
UDP Port                 : 7678
Hello Interval           : 1 Seconds
Dead Interval            : 3 Seconds
```

show vsx configuration split-recovery

Syntax

```
show vsx configuration split-recovery [vsx-peer]
```

Description

Displays the state of the split recovery mode.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show vsx configuration split-recovery
Split Recovery Mode    : Enabled
```

show vsx ip data-path

Syntax

```
show vsx ip data-path [<IP-ADDR> | <IP-ADDR>/<MASK>] [vrf <VRF-NAME>] [vsx-peer]
```

Description

Displays the datapath of the IPv4 route present on local and VSX peer devices.

Command context

Operator (>) or Manager (#)

Parameters

<IP-ADDR> | <IP-ADDR>/<MASK>]

Selects one of the following: *<IP-ADDR>* or *<IP-ADDR>/<MASK>*

<IP-ADDR>

Specifies the datapath for an IPv4 address based on the parameters provided.

<IP-ADDR>/<MASK>

Specifies the datapath for an IPv4 address and its specified subnet. Optional. Syntax: A.B.C.D/M

vrf <VRF-NAME>

Shows the IPv4 datapath for a specified VRF.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Example

Displaying the datapath on a VSX switch for 192.0.2.0:

```
switch# show vsx ip data-path 192.0.2.0

IPv4 Data Path Information For 192.0.2.0

Local Device
------------
Route : 192.0.2.0/32
    Egress L3 Interface : 1/1/2
    Next Hop MAC Address    : 08:00:09:ea:d7:d1
    Egress Port    : 1/1/2

    Egress L3 Interface : 1/1/3
    Nexthop Hop MAC Address    : 08:00:09:8e:59:1d
    Egress Port    : 1/1/3

Peer Device
------------
Route : 192.0.2.0/32
    Egress L3 Interface : loopback1
```

Displaying the datapath on a VSX switch for 198.51.100.0/32:

```
switch# show vsx ip data-path 198.51.100.0/32

IPv4 Data Path Information For 198.51.100.0/32

Local Device
------------
Route : 198.51.100.0/32
    Egress L3 Interface : 1/1/4
```

Displaying the datapaths on a VSX switch for 198.51.100.1:

```
switch# show vsx ip data-path 198.51.100.1

IPv4 Data Path Information For 198.51.100.1

Local Device
------------
Route : 198.51.100.1/32
    Egress L3 Interface : 1/1/4

Peer Device
------------
Route : 198.51.100.0/24
    Egress L3 Interface : 1/1/2
    Next Hop MAC Address   : 08:00:09:db:21:e8
    Egress Port   : 1/1/2
```

show vsx ip route

Syntax

```
show vsx ip route [<IP-ADDR> | <IP-ADDR>/<MASK> | unique]
     [vrf <VRF-NAME> | all-vrfs] [vsx-peer]
```

Description

Displays a specified LAG or all configured LAGs along with VSX LAGs.

Command context

Operator (>) or Manager (#)

Parameters

<IP-ADDR> | <IP-ADDR>/<MASK> | unique]

Selects one of the following: *<IP-ADDR>*, *<IP-ADDR>/<MASK>*, or unique

<IP-ADDR>

Specifies the route information for an IPv4 address based on the parameters provided.

<IP-ADDR>/<MASK>

Specifies the route information for an IPv4 address and its specified subnet. Optional. Syntax: A.B.C.D/M

unique

Specifies routes that are present only on the primary switch or only on the secondary switch. The routes that are present on both the primary and secondary switch are excluded. Optional. Syntax string.

vrf *<VRF-NAME>* | all-vrfs

Selects the VRF name or all VRFs.

<VRF-NAME>

Shows the IPv4 route information for a specified VRF.

all-vrf

Shows the IPv4 route information for all VRFs.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying IPv4 routes on a VSX switch:

```
switch# show vsx ip route

IPv4 Forwarding Routes

'[x/y]' denotes [distance/metric]

192.0.2.0/32, vrf default
    via  192.0.2.1,  [1/0],  static on vsx1
    via  192.0.2.2,  [1/0],  static on vsx2
```

Displaying IPv4 routes on a VSX switch:

```
switch# show vsx ip route

IPv4 Forwarding Routes

'[x/y]' denotes [distance/metric]

192.0.2.3/24, vrf default
    via  1/1/3,  [0/0],  connected on vsx1
    via  192.0.2.2,  [110/2],  ospf on vsx2
192.0.2.4/32, vrf default
    via  1/1/3,  [0/0],  local on vsx1
192.0.2.5/24, vrf default
    via  1/1/4,  [0/0],  connected on vsx1
    via  192.0.2.2,  [110/3],  ospf on vsx2
192.0.2.6/32, vrf default
    via  1/1/4,  [0/0],  local on vsx1
192.0.2.7/32, vrf default
    via  192.0.2.8,  [110/1],  ospf on vsx1
    via  192.0.2.1,  [110/1],  ospf on vsx1
    via  loopback1,  [0/0],  local on vsx2
```

Displaying IPv4 unique routes on a VSX switch:

```
switch# show vsx ip route unique

IPv4 Forwarding Routes

'[x/y]' denotes [distance/metric]

192.0.2.0/32, vrf default
    via  192.0.2.2,   [1/0],   static on vsx2
192.0.2.9/32, vrf default
    via  192.0.2.1,   [1/0],   static on vsx1
```

Displaying IPv4 routes on a VSX switch for 192.0.2.10:

```
switch# show vsx ip route 192.0.2.10

IPv4 Forwarding Routes

'[x/y]' denotes [distance/metric]

192.0.2.10/32, vrf default
    via  192.0.2.1,   [1/0],   static on vsx1
    via  192.0.2.2,   [1/0],   static on vsx2
```

show vsx ipv6 data-path

Syntax

`show vsx ipv6 data-path [<IPv6-ADDR> | <IPv6-ADDR>/<MASK>] [vrf <VRF-NAME>] [vsx-peer]`

Description

Displays the datapath of the IPv6 route on local and peer VSX devices.

Command context

Operator (>) or Manager (#)

Parameters

<IPV6-ADDR> | <IPV6-ADDR>/<MASK>]

Selects one of the following: *<IPV6-ADDR>* or *<IPV6-ADDR>/<MASK>*

<IPV6-ADDR>

Specifies the datapath for an IPv6 address based on the parameters provided.

<IPV6-ADDR>/<MASK>

Specifies the datapath for an IPv6 address and its specified subnet. Optional. Syntax: A.B.C.D/M

vrf *<VRF-NAME>*

Shows the IPv6 datapath for a specified VRF.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying an IPv6 datapath on a VSX switch:

```
switch# show vsx ipv6 data-path 1000::

IPv6 Data Path Information For 1000::

Local Device
------------
Route : 1000::/64
    Egress L3 Interface : 1/1/2

Peer Device
------------
Route : 1000::/64
    Egress L3 Interface : 1/1/2
```

Displaying an IPv6 datapath on a VSX switch:

```
switch# show vsx ipv6 data-path 2000::
IPv6 Data Path Information For 2000::

Local Device
------------
Route : 2000::/64
    Egress L3 Interface : 1/1/2
    Next Hop MAC Address    : 08:00:09:0e:0c:1b
    Egress Port    : 1/1/2
```

Displaying IPv6 datapath for 3000::/64 on a VSX switch:

```
switch# show vsx ipv6 data-path 3000::/64
IPv6 Data Path Information For 3000::/64

Local Device
------------
Route : 3000::/64
    Egress L3 Interface : 1/1/2
    Next Hop MAC Address    : 08:00:09:0e:0c:1b
    Egress Port    : 1/1/2
```

show vsx ipv6 route

Syntax

```
show vsx ipv6 route [<IPv6-ADDR> | <IPv6-ADDR>/<MASK> | unique]
     [vrf <VRF-NAME> | all-vrfs] [vsx-peer]
```

Description

Displays a specified LAG or all configured LAGs along with VSX LAGs.

Command context

Operator (>) or Manager (#)

Parameters

***<IPV6-ADDR>* | *<IPV6-ADDR>*/*<MASK>* | unique]**

Selects one of the following: *<IPV6-ADDR>*, *<IPV6-ADDR>*/*<MASK>*, or `unique`

<IPV6-ADDR>

Specifies the route information for an IPv4 address based on the parameters provided.

<IPV6-ADDR>*/*<MASK>

Specifies the route information for an IPv4 address and its specified subnet. Optional. Syntax: A.B.C.D/M

unique

Specifies routes that are present only on the primary switch or only on the secondary switch. The routes that are present on both the primary and secondary switch are excluded. Optional. Syntax string.

vrf *<VRF-NAME>* | all-vrfs

Selects the VRF name or all VRFs.

<VRF-NAME>

Shows the IPv4 route information for a specified VRF.

all-vrf

Shows the IPv4 route information for all VRFs.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying IPv6 routes on a VSX switch:

```
switch# show vsx ipv6 route

IPv6 Forwarding Routes

'[x/y]' denotes [distance/metric]

1000::/64, vrf default
    via  1/1/2,  [0/0],  connected on vsx1
    via  1/1/2,  [0/0],  connected on vsx2
1000::1/128, vrf default
    via  1/1/2,  [0/0],  local on vsx1
```

Displaying IPv6 unique routes on a VSX switch:

```
switch# show vsx ipv6 route unique
IPv6 Forwarding Routes

'[x/y]' denotes [distance/metric]

1000::1/128, vrf default
```

```
    via  1/1/2,  [0/0],  local on vsx1
1000::2/128, vrf default
    via  1/1/2,  [0/0],  local on vsx2
3000::/64, vrf default
    via  1000::2,  [1/0],  static on vsx1
```

Displaying IPv6 routes on a VSX switch for 2000::/64:

```
switch# show vsx ipv6 route 2000::/64
IPv6 Forwarding Routes

'[x/y]' denotes [distance/metric]

2000::/64, vrf default
    via  1000::2,  [1/0],  static on vsx1
    via  1000::1,  [1/0],  static on vsx2
```

show vsx status

Syntax

`show vsx status [inter-switch-link | keepalive | linkup-delay] [vsx-peer]`

Description

Displays global VSX status or a specified status determined by the selected parameter.

Command context

Operator (>) or Manager (#)

Parameters

`[inter-switch-link | keepalive | linkup-delay]`

Selects one of the following: `inter-switch-link`, `keepalive`, or `linkup-delay`

`inter-switch-link`

Specifies the display of the ISL status in VSX.

`keepalive`

Specifies the display of the VSX keepalive protocol status.

`linkup-delay`

Specifies the display of the VSX link-up delay information, such as the:

- Configured link-up delay timer.
- Delay timer status.
- Initial sync status.
- LAGs on which the delay timer is running.
- Status of the LAGs excluded from the link-up delay timer.
- Time remaining for the interfaces to be brought up.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Displaying the global VSX status:

```
switch# show vsx status
VSX Operational State
---------------------
  ISL channel             : In-Sync
  ISL mgmt channel        : operational
  Config Sync Status      : in-sync
  NAE                     : peer_reachable
  HTTPS Server            : peer_reachable


Attribute           Local                       Peer
------------        --------                    --------
ISL link            1/1/43                      1/1/43
ISL version         2                           2
System MAC          48:0f:cf:af:70:84           48:0f:cf:af:c2:84
Platform            8320                        8320
Software Version    10.0x.xxxx                  10.0x.xxxx
Device Role         primary                     secondary
```

Displaying the ISL status in VSX:

```
switch# show vsx status inter-switch-link
State                     : In-Sync
Link Status               : up
Mgmt state                : operational

Inter-switch link Statistics
----------------------------
Hello Packets Tx          : 4572
Hello Packets Rx          : 4573
Data Packets Tx           : 80634
Data Packets Rx           : 80637
Mgmt Packets Tx           : 25946
Mgmt Packets Rx           : 25167
Mgmt Packet Drops         : 0
```

Displaying the VSX keepalive protocol status:

```
switch# show vsx status keepalive
Keepalive State           : Keepalive-Established
Last Established          : Thu Jun  8 09:03:01 2018
Last Failed               : Thu Jun  8 09:04:02 2018
Peer System Id            : 58:1f:cf:af:a0:84
Peer Device Role          : primary

Keepalive Counters
Keepalive Packets Tx      : 322
Keepalive Packets Rx      : 121
Keepalive Timeouts        : 0
Keepalive Packets Dropped : 14
```

Displaying the VSX link-up delay status while ARP/MAC VSX synchronization is in progress:

```
switch# show vsx status linkup-delay

Configured linkup delay-timer                               : 180 seconds
Initial sync status                                         : In-progress
Delay timer status                                          : Waiting-to-start
Linkup Delay time left                                      :
Interfaces that will be brought up after delay timer expires : lag20,lag30-lag31
Interfaces that are excluded from delay timer               : lag2
```

Displaying the VSX link-up delay status with ARP/MAC VSX synchronization completed with the delay timer running:

```
switch# show vsx status linkup-delay

Configured linkup delay-timer                               : 180 seconds
Initial sync status                                         : Completed
Delay timer status                                          : Running
Linkup Delay time left                                      : 1 minutes 22 seconds
Interfaces that will be brought up after delay timer expires : lag20,lag30-lag31
Interfaces that are excluded from delay timer               : lag2
```

Displaying the VSX link-up delay status with ARP/MAC VSX synchronization completed and the delay timer expired:

```
switch# show vsx status linkup-delay

Configured linkup delay-timer                               : 180 seconds
Initial sync status                                         : Completed
Delay timer status                                          : Completed
Linkup Delay time left                                      :
Interfaces that will be brought up after delay timer expires : lag2
Interfaces that are excluded from delay timer               :
```

Displaying the global VSX status for the peer switch:

```
vsx-primary# show vsx status vsx-peer
VSX Operational State
---------------------
  ISL channel             : In-Sync
  ISL mgmt channel        : operational
  Config Sync Status      : in-sync
  NAE                     : peer_reachable
  HTTPS Server            : peer_reachable

Attribute           Local               Peer
------------        --------            --------
ISL link            lag1                lag1
ISL version         2                   2
System MAC          e0:07:1b:cb:72:e4   98:f2:b3:68:79:2e
Platform            8320                8320
Software Version    10.0x.xxxx          10.0x.xxxx
Device Role         secondary           primary
```

Displaying the status for an out-of-sync status for VSX.

```
switch# show vsx status linkup-delay

Configured linkup delay-timer                               : 20 seconds
Initial sync status                                         :
```

```
Delay timer status                                           :
Linkup Delay time left                                       :
Interfaces that will be brought up after delay timer expires :
Interfaces that are excluded from delay timer                :
```

show vsx status config-sync

Syntax

```
show vsx status config-sync [vsx-peer]
```

Description

Displays VSX configuration synchronization status for peers. This command can be run from the primary or secondary peer to view the configuration synchronization state.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show vsx status config-sync
Admin State                : Enabled
Operational State          : Operational
Error State                : None
Recommended remediation : N/A
Current Time               : Wed Jul 18 23:41:07 2018
Last Sync Time             : Wed Jul 18 23:38:26 2018
```

NOTE: The Admin State parameter can be configured individually on each of the switches on the VSX pair. Hence difference in values does not imply a failure.

More information

ISL is out-of-sync

split recovery

Syntax

```
split-recovery
```

```
no split-recovery
```

Description

Enables split recovery mode. Split recovery mode is enabled by default.

The `no` form of this command disables split-recovery mode.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

Split recovery mode prevents traffic loss when the ISL goes out-of-sync and keepalive subsequently fails. When the ISL goes out-of-sync and keepalive is established, the secondary VSX LAGs are brought down. If keepalive then also fails, this situation causes a split condition. In this case, if split recovery mode is enabled, the secondary switch restores its VSX LAGs so they are up.

When split recovery mode is disabled during a split condition, the secondary switch keeps it VSX LAGs down.

Examples

Enabling split recovery mode:

```
switch(config-vsx)# split-recovery
```

Disabling split recovery mode:

```
switch(config-vsx)# no split-recovery
```

More information

Split brain scenario
Failure scenarios and split recovery
ISL is out-of-sync and keepalive is down

system-mac

Syntax

```
system-mac <MAC-ADDR>

no system-mac <MAC-ADDR>
```

Description

Sets the MAC address as the VSX system MAC address to be used by control plane protocols, such as STP and LACP. A pair of VSX switches must have the same VSX system MAC.

The `no` form of this command unconfigures the VSX system MAC address to be used by control plane protocols.

Command context

```
config-vsx
```

Parameters

<MAC-ADDR>

Specifies the MAC address in a colon separated format, such as XX:XX:XX:XX:XX:XX, for control plane protocols.

Authority

Administrators or local user group members with execution rights for this command.

Usage

The `system-mac` *`<MAC-ADDR>`* command is highly recommended for preventing traffic disruptions when the primary VSX switch restores after the secondary VSX switch, such as during:

- A primary switch hardware replacement.
- A power outage with the primary switch restore after the secondary switch restore.

When the primary switch is restored after the secondary switch, a traffic disruption might occur when the ISL starts to sync. This situation occurs because the MAC system address changes from the secondary switch to the primary switch for the LACP. To avoid the traffic disruption, set the common system MAC address by entering the `system-mac` *`<MAC-ADDR>`* command. This command creates a common system MAC address between the two VSX switches. This common system MAC address prevents a traffic disruption when the secondary switch comes up before the primary switch. If the common system MAC access is enabled, the secondary switch uses the common system MAC address instead of its own system MAC address, which prevents a traffic loss.

The system MAC address also maintains the same MSTP bridge ID across VSX switches, which act as a single switch.

Examples

Setting a MAC address as the VSX system MAC address to be used by control plane protocols:

```
switch(config-vsx)# system-mac 02:01:00:00:01:00
```

Unconfiguring a VSX system MAC address to be used by control plane protocols:

```
switch(config-vsx)# no system-mac 02:01:00:00:01:00
```

NOTE: Null system MAC address such as 00:00:00:00:00:00 is not allowed.

vsx

Syntax

```
vsx
```

```
no vsx
```

Description

Creates the VSX context on the switch.

The `no` form of this command disables the VSX context on the switch and removes all related configuration settings.

Command context

`config`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Creating the VSX context on the switch:

```
switch(config)# vsx
switch(config-vsx)#
```

Removing the VSX context and all VSX configuration settings from the switch:

```
switch(config-vsx)# no vsx
VSX configuration will be deleted.
Do you want to continue (y/n)? y
switch(config)#
```

vsx active-forwarding

Syntax

`vsx active-forwarding`

`no vsx active-forwarding`

Description

Configures VSX active-forwarding on an interface VLAN.

The `no` form of this command unconfigures VSX active-forwarding on a VLAN interface.

Command context

`config-if-vlan`

Authority

Administrators or local user group members with execution rights for this command.

Usage

Active forwarding cannot be configured when ICMP redirect is enabled. The ICMP redirect setting is global not per SVI. Enter the `no ip icmp redirect` command for disabling ICMP redirect at the `switch(config)#` prompt.

If a system has active forwarding enabled, an active gateway can have a maximum of 14 "unique" MAC addresses per system, including IPv4 and IPv6 addresses.

If a system has active forwarding disabled, an active gateway can have a maximum of 16 "unique" MAC addresses per system, including IPv4 and IPv6 addresses.

Examples

Successfully enabling VSX active-forwarding:

```
switch# interface vlan 3
switch(config-if-vlan)# vsx active-forwarding
switch(config-vsx)#
```

Unconfiguring VSX active-forwarding:

```
switch# interface vlan 3
switch(config-if-vlan)# no vsx active-forwarding
switch(config-vsx)#
```

vsx-sync

Syntax

```
vsx-sync

no vsx-sync
```

Description

Enables VSX synchronization for the entire context for the following features from the primary VSX node to the secondary peer switch:

- Access list context
- Classifier context
- Object group context
- Policy-based routing profile context
- Policy context
- QoS queue profile context
- QoS schedule profile context
- VLAN context

The `no` form of this command disables VSX synchronization for the entire context for a feature, but it does not remove the feature configurations from the secondary peer. Any subsequent configuration changes made under the specific configuration context are not synchronized to the secondary peer switch.

Command context

`config-acl-<ACL-TYPE>`

`config-addrgroup-ip`

`config-addrgroup-ipv6`

`config-class-<CLASS-TYPE>`

`config-policy`

`config-portgroup`

`config-pbr-action-list-<ACTION-LIST-NAME>`

`config-queue`

`config-schedule`

`config-vlan-<ID>`

Authority

Administrators or local user group members with execution rights for this command.

Usage

Make sure that you are in the correct context for the feature that you are trying to enable VSX synchronization:

Feature context for enabling VSX synchronization	Command for accessing correct context for the `vsx-sync` command*
Access list context for an ACL type, such as IPv4, IPv6, or MAC.	`access-list` *<ACL-TYPE> <ACL-NAME>*
Class context for a class type, such as IPv4, IPv6, or MAC.	`class` *<CLASS-TYPE> <CLASS-NAME>*
Object group context for IPv4	`object-group ip address` *<OBJECT-GROUP-NAME>*
Object group context for IPv6	`object-group ipv6 address` *<OBJECT-GROUP-NAME>*
Object group context for ports	`object-group port` *<OBJECT-GROUP-NAME>*
Policy-based routing profile context	`pbr` *<ACTION-LIST-NAME>*
Policy context	`policy` *<POLICY-NAME>*
QoS queue profile context	`qos queue-profile` *<QUEUE-PROFILE-NAME>*
QoS schedule profile context	`qos schedule-profile` *<SCHEDULE-PROFILE-NAME>*
VLAN context	`vlan` *<ID>*

*The commands listed in this column are entered at the `switch(config)#` prompt, as shown in the following examples.

Examples

Enabling VSX synchronization for this IPv4 access list context to the secondary peer:

```
switch(config)# access-list ip ITBoston
switch(config-acl-ip)# vsx-sync
```

Enabling VSX synchronization for this IPv6 access list context to the secondary peer:

```
switch(config)# access-list ipv6 ITRoseville
switch(config-acl-ipv6)# vsx-sync
```

Enabling VSX synchronization for this MAC access list context to the secondary peer:

```
switch(config)# access-list mac ITBangalore
switch(config-acl-ipv6)# vsx-sync
```

Enabling VSX synchronization for this IPv4 class context to the secondary peer:

```
switch(config)# class ip ITengineering
switch(config-class-ip)# vsx-sync
```

Enabling VSX synchronization for this object group context for IPv4:

```
switch(config)# object-group ip address group1
switch(config-addrgroup-ip)# 1.1.1.1
switch(config-addrgroup-ip)# vsx-sync
```

Enabling VSX synchronization for this QoS queue profile context to the secondary peer:

```
switch(config)# qos queue-profile test_queue_profile
switch(config-queue)# vsx-sync
```

Enabling VSX synchronization for this QoS schedule profile context to the secondary peer:

```
switch(config)# qos schedule-profile test_queue_profile1
switch(config-schedule)# vsx-sync
```

Enabling VSX synchronization for this PBR profile context to the secondary peer:

```
switch(config)# pbr engineering
switch(config-pbr-action-list-engineering)# vsx-sync
```

Enabling VSX synchronization for this policy context to the secondary peer:

```
switch(config)# policy ITPaloAlto
switch(config-policy)# vsx-sync
```

Enabling VSX synchronization for this VLAN context to the secondary peer:

```
switch(config)# vlan 1
switch(config-vlan-1)# vsx-sync
```

Disabling VSX synchronization for this IPv4 class context to the secondary peer:

```
switch(config)# class ip ITengineering
switch(config-class-ip)# no vsx-sync
```

Disabling VSX synchronization for this object group context for IPv4:

```
switch(config)# object-group ip address group1
switch(config-addrgroup-ip)# no vsx-sync
```

Disabling VSX synchronization for this QoS queue profile context to the secondary peer:

```
switch(config)# qos queue-profile test_queue_profile
switch(config-queue)# no vsx-sync
```

Disabling VSX synchronization for this QoS schedule profile context to the secondary peer:

```
switch(config)# qos schedule-profile test_queue_profile1
switch(config-schedule)# no vsx-sync
```

Disabling VSX synchronization for this PBR profile context to the secondary peer:

```
switch(config)# pbr engineering
switch(config-pbr-action-list-engineering)# no vsx-sync
```

Disabling VSX synchronization for this policy context to the secondary peer:

```
switch(config)# policy ITPaloAlto
switch(config-policy)# no vsx-sync
```

Disabling VSX synchronization for this MAC access list context to the secondary peer:

```
switch(config)# access-list mac ITBangalore
switch(config-acl-ipv6)# no vsx-sync
```

Disabling VSX synchronization for this VLAN context to the secondary peer:

```
switch(config)# vlan 1
switch(config-vlan-1)# no vsx-sync
```

vsx-sync {[access-lists] [qos] [rate-limits] [vlans] [policies] [irdp] [portfilter]}

Syntax

```
vsx-sync {[access-lists] [qos] [rate-limits] [vlans] [policies] [irdp] [portfilter]}

no vsx-sync {[access-lists] [qos] [rate-limits] [vlans] [policies] [irdp] [portfilter]}
```

Description

Enables VSX synchronization for the following for a logical interface or a LAG instance:

- Access lists
- IRDP configurations
- QoS
- Rate limits
- Port filter configurations
- VLAN associations

This command enables VSX synchronization for individual associations and to the combination of associations to the interface context. To synchronize the associations, you must configure the same interface on the peer switch.

IMPORTANT: When enabling VSX synchronization under a physical interface, under a VLAN interface, or a VSX LAG, create on the secondary switch the physical interface, VLAN interface, or VSX LAG with the same name and routing setting as on the primary switch. For example, if the primary switch has a physical interface of 1/1/1, you must create another physical interface of 1/1/1 on the secondary switch. Also, if the primary VSX switch has routing enabled, the secondary switch must have routing enabled. Once the name and routing information is the same, VSX synchronization synchronizes the additional configuration information from the primary VSX switch to the secondary VSX switch.

The `no` form of this command disables VSX synchronization, but it does not remove the feature configurations from the secondary peer.

Command context

```
config-if

config-lag-if
```

Parameters

{[access-lists] [qos] [rate-limits] [vlans] [policies] [irdp] [portfilter]}

Specifies one or more of the features for which to enable VSX synchronization.

access-lists

Specifies the access lists that are associated under the interface enabled for VSX syncing.

qos

Specifies the QoS associated under the interface enabled for VSX syncing.

rate-limits

Specifies the rate limits that are associated under the interface enabled for VSX syncing.

vlans

Specifies the VLANs that are associated under the interface enabled for VSX syncing.

policies

Specifies the classifier policies that are associated under the interface enabled for VSX syncing.

irdp

Specifies the Internet Router Discovery Protocol (IRDP) configurations that are associated under the interface enabled for VSX syncing.

portfilter

Specifies the port filter configurations that are associated under the interface enabled for VSX syncing.

Authority

Administrators or local user group members with execution rights for this command.

Example

Enabling VSX synchronization for VLANs associated with logical interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# vsx-sync vlans
```

Enabling VSX synchronization for access lists associated with logical interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# vsx-sync access-lists
```

Enabling VSX synchronization for access lists and policies that are associated with logical interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# vsx-sync access-lists policies
```

Enabling VSX synchronization for QoS that are associated under logical interface 1/1/5:

```
switch(config)# interface 1/1/5
switch(config-if)# vsx-sync vlans qos
```

Enabling VSX synchronization for rate limits that are associated under logical interface 1/1/5:

```
switch(config)# interface 1/1/5
switch(config-if)# vsx-sync rate-limits
```

Enabling VSX synchronization for rate limits, VLANs, QoS, access lists, policies associated with logical interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# vsx-sync rate-limits vlans qos access-lists policies
```

Enabling VSX synchronization for VLAN 1 under interface LAG 1:

```
switch(config)# interface lag 1
switch(config-lag-if)# vsx-sync vlans
switch(config-lag-if)# vlan trunk native 1
```

Enabling VSX synchronization for an access list under interface LAG 2:

```
switch(config)# interface lag 2
switch(config-lag-if)# vsx-sync access-lists
switch(config-lag-if)# apply access-list ip test1 in
```

Enabling VSX synchronization for a QoS under interface LAG 3:

```
switch(config)# interface lag 3
switch(config-lag-if)# vsx-sync qos
switch(config-lag-if)# apply qos schedule-profile test
```

Enabling VSX synchronization for a rate limit under interface LAG 4:

```
switch(config)# interface lag 4
switch(config-lag-if)# vsx-sync rate-limits
switch(config-lag-if)# rate-limit broadcast 23 kbps
```

Enabling VSX synchronization for a policy named test under interface LAG 5:

```
switch(config)# interface lag 5
switch(config-lag-if)# vsx-sync policies
switch(config-lag-if)# apply policy test in
```

Enabling VSX synchronization for a policy named test1, a rate limit of 23 kbps, a QoS named test, VLAN 1, and an access list named test1 under interface LAG 6:

```
switch(config)# interface lag 6
switch(config-lag-if)# vsx-sync policies rate-limits qos vlans access-lists
switch(config-lag-if)# apply policy test1 in
switch(config-lag-if)# rate-limit broadcast 23 kbps
switch(config-lag-if)# apply qos schedule-profile test
switch(config-lag-if)# vlan trunk native 1
switch(config-lag-if)# apply access-list ip test 1 in
```

Enabling VSX synchronization for a port filter:

```
switch(config)# interface 1/1/1
switch(config-if)# vsx-sync portfilter
```

```
switch(config)# interface lag 1
switch(config-lag-if)# vsx-sync portfilter
```

Disabling VSX synchronization for access lists and policies under logical interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# no vsx-sync access-lists policies
```

Disabling VSX synchronization for access lists and policies under interface LAG 2:

```
switch(config)# interface lag 2
switch(config-if)# no vsx-sync access-lists policies
```

Enabling VSX synchronization of IRDP configurations under logical interface 1/1/1. The first five lines in the example configure IRDP and the last line enables VSX synchronization for IRDP configurations associated under interface 1/1/1:

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp
switch(config-if)# ip irdp minadvertinterval 550
switch(config-if)# ip irdp maxadvertinterval 850
switch(config-if)# ip irdp holdtime 900
switch(config-if)# vsx-sync irdp
```

vsx-sync {[active-gateways] [policies]}

Syntax

```
vsx-sync {[active-gateways] [policies]}

no vsx-sync {[active-gateways] [policies]}
```

Description

Enables VSX sync of active gateways or policies associated under an interface. To synchronize the associations, you must configure the same `interface vlan` on the peer switch.

The `no` form of this command removes VSX synchronization for active gateways or policies associated under an interface, but it does not remove the feature configurations from the secondary peer switch.

Command context

`config-if-vlan`

Parameters

`{[active-gateways] [policies]}`

Specifies one or more of the features for which to enable VSX synchronization.

`access-gateways`

Specifies that active gateways associated with an interface are enabled for VSX syncing.

`policies`

Specifies that policies associated with an interface are enabled for VSX syncing.

Authority

Administrators or local user group members with execution rights for this command.

Usage

Configure an SVI on the secondary switch; however, you do not need to run the `vsx-sync active-gateways` command on the secondary VSX switch.

NOTE: Do not use peer system MAC address as an active-gateway VMAC. If same MAC address is used, the VSX synchronization will try to sync the configuration on secondary switch and cause traffic disruptions.

Examples

Enabling VSX synchronization for an active gateway associated with VLAN 1:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# vsx-sync active-gateways
```

Enabling VSX synchronization for policies associated with VLAN 1:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# vsx-sync policies
```

Enabling VSX synchronization for active gateways and policies associated with VLAN 1:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# active-gateway ip 10.10.10.10 mac 23:24:25:26:27:28
switch(config-if-vlan)# active-gateway ipv6 fd12:3456:789a:1::1 mac fd12:3456:789a:1::1 23:24:25:26:27:28
switch(config-if-vlan)# vsx-sync active-gateways policies
```

Disabling VSX synchronization for active gateways associated with VLAN 1:

```
switch(config)# interface vlan 1
switch(config-if-vlan)# no vsx-sync active-gateways
```

vsx-sync aaa

Syntax

```
vsx-sync aaa

no vsx-sync aaa
```

Description

Enables VSX synchronization of all AAA configurations, including user, RADIUS server, and TACACS+ server, on the primary VSX node to the secondary peer switch. To synchronize AAA configurations associated with a particular VRF, you must configure the same VRF on the peer switch.

The `no` form of this command removes VSX synchronization of global AAA configurations, but it does not remove the existing global AAA feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the AAA configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# vsx-sync aaa
```

Disabling VSX sync for the AAA configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync aaa
```

vsx-sync acl-log-timer

Syntax

```
vsx-sync acl-log-timer

no vsx-sync acl-log-timer
```

Description

Enables VSX synchronization of access list log timer configurations on the primary VSX node to the secondary peer.

The `no` form of this command removes VSX synchronization of access list log timer configurations to the secondary peer. However, it does not remove the previously synced configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the access list log timer configurations:

```
switch(config)# access-list log timer 30
switch(config)# vsx
switch(config-vsx)# vsx-sync acl-log-timer
```

Disabling VSX sync for the access list log timer configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync acl-log-timer
```

vsx-sync arp-security

Syntax

```
vsx-sync arp-security

no vsx-sync arp-security
```

Description

Enables VSX synchronization of the ARP security configurations on the primary VSX switch to the secondary peer switch. After you enter `vsx-sync arp-security`, you must enter `vsx-sync mclag-interfaces` for enabling VSX synchronization for the ARP security feature.

The `no` form of this command removes VSX synchronization of ARP security configurations on VLAN mode and LAG interface mode to the secondary peer switch. However, it does not remove the existing ARP security configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling of VSX synchronization for ARP security feature configurations to secondary peer:

```
primary_sw(config)# vsx
primary_sw(config-vsx)# vsx-sync arp-security
primary_sw(config-vsx)# vsx-sync mclag-interfaces
```

Disabling the VSX synchronization for ARP security feature configurations to secondary peer:

```
primary_sw(config)# vsx
primary_sw(config-vsx)# no vsx-sync arp-security
switch(config-vsx)# no vsx-sync mclag-interfaces
```

vsx-sync bfd-global

Syntax

```
vsx-sync bfd-global

no vsx-sync bfd-global
```

Description

Enables syncing of global BFD configurations, such as `echo-src-ip-address`, `detect-multiplier`, `min-transmit-interval`, and `min-receive-interval`, on the primary VSX node to the secondary peer.

IMPORTANT: This command enables VSX synchronization only at the top level and not at the context level.

The `no` form of this command disables the syncing of global BFD configurations to the secondary peer, but it does not remove the existing global BFD feature configurations from it.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX synchronization for various global BFD configurations:

```
switch(config)# bfd detect-multiplier 1
switch(config)# bfd min-transmit-interval 1000
switch(config)# bfd min-receive-interval 1000
switch(config)# bfd echo-src-ip-address 2.2.2.2
switch(config)# bfd min-echo-receive-interval 1000
switch(config)# vsx
switch(config-vsx)# vsx-sync bfd-global
```

Disabling VSX synchronization for global BFD configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync bfd-global
```

vsx-sync bgp

Syntax

```
vsx-sync bgp

no vsx-sync bgp
```

Description

Enables syncing of BGP configurations on the primary VSX switch to the secondary peer switch.

The `no` form of this command disables syncing BGP, as path lists, community lists, prefix lists, and route map configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Usage

The following BGP configurations are synchronized: as path lists, community lists, prefix lists, and route map configurations. To maintain the uniqueness of a switch in the autonomous system, the BGP router ID, BGP cluster ID, and BGP neighbor update-source are not synchronized. This exclusion is required for BGP functionality to work seamlessly even with VSX topology.

Several settings are also not synced. The `neighbor <IP address> shutdown` setting is not synced because syncing that setting would cause both the primary and secondary VSX nodes towards the core to go down. In route map configurations, the following settings are also not synced from the primary VSX switch to the secondary VSX switch, because the next-hop is always set differently for the primary and secondary VSX peers:

- `set ip nexthop <IP-ADDR>`
- `set ipv6 nexthop global <IP-ADDR>`

If the next-hop must be same for both primary and secondary VSX peers, configure the same value on the individual switches.

Examples

Enabling VSX sync for the BGP configurations:

```
switch(config)#  ip aspath-list list1 seq 10 permit 10
switch(config)# ip community-list expanded com1 seq 10 permit 10
switch(config)# ip extcommunity-list standard ext1 seq 10 permit rt 10:4
switch(config)# ip prefix-list pref1 seq 10 permit any
switch(config)# route-map rm1 permit
switch(config-route-map-rm1-10)#  match ip next-hop 1.1.1.1
switch(config)#  router bgp 100
switch(config-bgp)# bgp router-id 1.1.1.1
switch(config-bgp)# neighbor 12.1.1.1 remote-as 1
switch(config-bgp)# address-family ipv4 unicast
switch(config-bgp-ipv4-uc)# neighbor 12.1.1.1 activate
switch(config)# vsx
switch(config-vsx)# vsx-sync bgp
```

Disabling VSX sync for the BGP configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync bgp
```

vsx-sync copp-policy

Syntax

```
vsx-sync copp-policy

no vsx-sync copp-policy
```

Description

Enables VSX synchronization of CoPP policy configurations on the primary VSX node to the secondary peer switch.

The `no` form of this command removes VSX synchronization of global CoPP configurations, but it does not remove the existing global CoPP configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first three lines in the following example show the setting of several policy configurations. The last two lines of the example show the enabling of VSX synchronization for CoPP policy configurations.

```
switch(config)# copp-policy mypolicy
switch(config-copp)# class arp-broadcast drop
switch(config-copp)# no class arp-unicast
switch(config)# vsx
switch(config-vsx)# vsx-sync copp-policy
```

Disabling VSX synchronization for global CoPP configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync copp-policy
```

vsx-sync dcb-global (8325 and 8360 series switches only)

Syntax

```
vsx-sync dcb-global

no vsx-sync dcb-global
```

Description

Enables VSX synchronization of global DCBx configurations from the primary VSX node to the secondary peer.

The `no` form of the command disables VSX synchronization of global DCBx configurations to the secondary peer; however, it does not remove the existing DCBx feature configurations from the secondary peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Usage

The following commands are synced from primary VSX node to secondary VSX node:

- `lldp dcbx`
- `dcbx application`

Examples

The first two lines in the following example show the setting of global DCBx configurations. The last two lines in the example show the enabling of VSX synchronization for global DCBx configurations.

```
switch(config)# lldp dcbx
switch(config)# dcbx application iscsi priority 7
switch(config)# vsx
switch(config-vsx)# vsx-sync dcb-global
```

Disabling VSX synchronization for global DCBx configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync dcb-global
```

vsx-sync dhcp-relay

Syntax

```
vsx-sync dhcp-relay

no vsx-sync dhcp-relay
```

Description

Enables VSX synchronization of DHCPv4 and DHCPv6 relay configurations on the primary VSX node to the secondary peer.

The `no` form of the command disables the VSX synchronization of DHCPv4 and DHCPv6 relay configurations to the secondary peer; however, it does not remove the existing DHCPv4 and DHCPv6 relay configurations from the secondary VSX peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

This example enables VSX synchronization for DHCPv4 relay configurations. The first six lines in the example show DHCPv4 relay configurations. The last two lines show how to enable VSX synchronization for the DHCP relay configurations:

```
switch(config)# interface 1/1/1
switch(config-if)# ip helper-address 192.168.10.1
switch(config-if)# ip helper-address 192.168.20.1
switch(config)# interface 1/1/2
switch(config-if)# ip helper-address 192.168.30.1
switch(config)# dhcp-relay option 82
switch(config)# vsx
switch(config-vsx)# vsx-sync dhcp-relay
```

This example enables VSX synchronization for DHCPv6 relay configurations. The first seven lines in the example show DHCPv6 relay configurations. The last two lines show how to enable VSX synchronization for the DHCP relay configurations:

```
switch(config)# dhcpv6-relay
switch(config)# interface 1/1/1
switch(config-if)# ipv6 helper-address unicast 2001:db8:0:1::
switch(config-if)# ipv6 helper-address multicast FF01::1:1000 egress 1/1/2
switch(config)# interface 1/1/2
switch(config-if)# ipv6 helper-address unicast 2001:db8:0:2::
switch(config)# dhcpv6-relay option 79
switch(config)# vsx
switch(config-vsx)# vsx-sync dhcp-relay
```

Disabling VSX synchronization for DHCP relay configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync dhcp-relay
```

vsx-sync dhcp-server

Syntax

```
vsx-sync dhcp-server

no vsx-sync dhcp-server
```

Description

Enables VSX synchronization of all DHCPv4 server configurations, including external storage configurations, on the primary VSX node to the secondary peer. To synchronize DHCPv4 server configurations associated with a particular VRF, configure the same VRF on the peer device. Only the primary VSX node answers DHCP service requests, and leases can only be exported from the primary VSX node.

The `no` form of the command disables VSX synchronization of DHCPv4 server configurations to the secondary peer; however, it does not remove the existing DHCPv4 server feature configurations from the secondary peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first six lines in the following example show the setting of a DHCPv4 server configuration. The last line of the example shows the enabling of VSX synchronization for global DHCPv4 server configurations.

```
switch(config)# dhcp-server external-storage dhcp-dbs file dhcpv4_lease_file delay 600
switch(config)# dhcp-server vrf default
switch(config-dhcp-server)# pool test
switch(config-dhcp-server-pool)# range 10.0.0.20 10.0.0.30
switch(config-dhcp-server-pool)# default-router 10.0.0.1 10.0.0.10
switch(config-dhcp-server-pool)# static-bind ip 10.0.0.1 mac 24:be:05:24:75:73
switch(config)# vsx
switch(config-vsx)# vsx-sync dhcp-server
```

Disabling VSX synchronization for global DHCPv4 server configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync dhcp-server
```

vsx-sync dhcpv6-server

Syntax

```
vsx-sync dhcpv6-server

no vsx-sync dhcpv6-server
```

Description

Enables VSX synchronization of all DHCPv6 server configurations, including external storage configurations, on the primary VSX node to the secondary peer. To synchronize DHCPv6 server configurations associated with a particular VRF, configure the same VRF on the peer device.

The `no` form of the command disables VSX synchronization of DHCPv6 server configurations to the secondary peer; however, it does not remove the existing DHCPv6 server feature configurations from the secondary peer.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first six lines in the following example show the setting of a DHCPv6 server configuration. The last two lines of the example show the enabling of VSX synchronization for global DHCPv6 server configurations.

```
switch(config)# dhcpv6-server external-storage dhcpv6-dbs file dhcpv6_lease_file delay 600
switch(config)# dhcpv6-server vrf default
switch(config-dhcp-server)# pool test
switch(config-dhcpv6-server-pool)# range 2001::1 2001::10 prefix-len 64
switch(config-dhcpv6-server-pool)# option 22 ipv6 2001::12
switch(config-dhcpv6-server-pool)# static-bind ipv6 2001::11 client-id 1:0:a0:24:ab:fb:9c
switch(config)# vsx
switch(config-vsx)# vsx-sync dhcpv6-server
```

Disabling VSX synchronization for global DHCPv6 server configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync dhcpv6-server
```

vsx-sync dns

Syntax

```
vsx-sync dns

no vsx-sync dns
```

Description

Enables VSX synchronization of the global DNS configurations on the primary VSX node to the secondary peer switch. To synchronize DNS configurations associated with particular VRF, you must configure the same VRF on the peer switch.

The `no` form of this command removes VSX synchronization for global DNS configurations, but it does not remove the feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first line in the following example shows the setting of a DNS configuration. The last two lines of the example show the enabling of VSX synchronization for global DNS configurations.

```
switch(config)# ip dns domain-name domain.com
switch(config)# vsx
switch(config-vsx)# vsx-sync dns
```

Disabling VSX synchronization for global DNS configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync dns
```

vsx-sync evpn

Syntax

```
vsx-sync evpn

no vsx-sync evpn
```

Description

Enables syncing of all EVPN context-related configurations on primary VSX node to the secondary peer switch.

The `no` form of this command disables syncing EVPN configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer switch.

NOTE: As a prerequisite, VLAN vsx-sync must be enabled separately for the VLAN configurations inside EVPN context to get synced.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the EVPN configurations:

```
switch(config)# vlan 2
switch(config-vlan-2)# vsx-sync
switch(config)# evpn
switch(config-evpn)# vlan 2
switch(config-evpn-vlan-2)# rd 5:5
switch(config-evpn-vlan-2)# route-target export 1:1
switch(config-evpn-vlan-2)# route-target import 1:1
switch(config)# vsx
switch(config-vsx)# vsx-sync evpn
```

Disabling VSX sync for the EVPN configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync evpn
```

vsx-sync icmp-tcp

Syntax

```
vsx-sync icmp-tcp

no vsx-sync icmp-tcp
```

Description

Enables VSX synchronization of IP ICMP configurations, including `ip icmp unreachable`, `ip icmp redirect`, and `ip icmp throttle` configurations, on primary VSX node to the secondary peer. To synchronize `ip icmp unreachable`, `ip icmp redirect`, and `ip icmp throttle` configurations, associated with particular VRF, configure the same VRF on the peer device.

The `no` form of the command disables the VSX synchronization of IP ICMP configurations to the secondary peer. However, it does not remove the existing IP ICMP configurations from the secondary VSX peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX synchronization for IP ICMP configurations:

```
switch(config)# vsx
switch(config-vsx)# vsx-sync icmp-tcp
```

Disabling VSX synchronization for IP ICMP configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync icmp-tcp
```

vsx-sync keychain

Syntax

`vsx-sync keychain`

`no vsx-sync keychain`

Description

Enables synchronizing of key chain configurations on primary VSX node to the secondary peer. There is no configuration synchronization from secondary to primary peer.

If any additional modification or configuration is made on the primary for the key chain set of features, the features will be auto-synchronized.

The `no` form of the command disables synchronizing key chain configurations to the secondary peer. But it does not remove the previously synchronized configurations from the secondary peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling synchronizing of key chain configurations on primary VSX node to the secondary peer:

```
switch(config)# keychain ospf_keys
switch(config-keychain)# key 1
switch(config-keychain-key)# send-lifetime start-time 10:10:10 10/25/2019 end-time 10:10:10 11/25/2019
switch(config-keychain-key)# accept-lifetime duration infinite
switch(config)# vsx
switch(config-vsx)# vsx-sync keychain
```

Disabling synchronizing key chain configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync keychain
```

vsx-sync lldp

Syntax

`vsx-sync lldp`

`no vsx-sync lldp`

Description

Enables VSX synchronization of the LLDP configurations on the primary VSX node to the secondary peer.

The `no` form of this command disable VSX synchronization of LLDP configurations to the secondary peer, but it does not remove the existing LLDP feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first line in the following example shows the setting of an LLDP configuration. The last two lines of the example show the enabling of VSX synchronization for LLDP configurations.

```
switch(config)#  lldp reinit 6
switch(config)# vsx
switch(config-vsx)# vsx-sync lldp
```

Disabling VSX synchronization of LLDP configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync lldp
```

vsx-sync loop-protect-global

Syntax

`vsx-sync loop-protect-global`

`no vsx-sync loop-protect-global`

Description

Enables the VSX synchronization of global loop protect configurations, such as transmit-interval and re-enable-timer, on the primary VSX node to the secondary peer switch. To enable VSX synchronization at the context level for this feature, enter the `vsx-sync mclag-interfaces` command at the context level.

The `no` form of this command removes VSX synchronization of global loop protect configurations, but it does not remove the existing global loop protect feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first two lines in the following example show the setting of global loop protect configurations. The last two lines of the example show the enabling of VSX synchronization for global loop protect configurations.

```
switch(config)# loop-protect transmit-interval 10
switch(config)# loop-protect re-enable-timer 300
switch(config)# vsx
switch(config-vsx)# vsx-sync loop-protect-global
```

Disabling VSX synchronization of global loop protect configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync loop-protect-global
```

vsx-sync mac-lockout

Syntax

```
vsx-sync mac-lockout

no vsx-sync mac-lockout
```

Description

Enables VSX synchronization of the MAC Lockout configurations on the primary VSX node to the secondary peer.

The `no` form of this command disables syncing MAC Lockout configurations to the secondary peer. However, it does not remove the existing MAC Lockout feature configurations from the secondary peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX synchronization for MAC Lockout configurations:

```
switch(config)# mac-lockout 10:10:10:10:10:10
switch(config)# vsx
switch(config-vsx)# vsx-sync mac-lockout
```

Disabling VSX synchronization for MAC Lockout configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync mac-lockout
```

vsx-sync mclag-interfaces

Syntax

```
vsx-sync mclag-interfaces

no vsx-sync mclag-interfaces
```

Description

Enables the VSX synchronization of VSX LAG interface associations and attributes on the primary VSX switch to the secondary peer switch. The Usage section in this topic provides a listing of specific associations and attributes that are synchronized to the secondary switch.

The `no` form of this command removes VSX synchronization of global VSX LAG and attributes, but it does not remove the existing VSX LAG feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Usage

The VSX LAG interface associations and attributes that support VSX synchronization are for example:

Interface associations:

- Access lists
- Policies
- QoS
- Port filters
- Rate limits
- VLANs

Supported attributes:

- LAG description
- LACP
- Loop protect
- QoS trust
- sFlow
- STP

This configuration overrides the existing VSX synchronization associations created under the VSX LAG interface context. Also with this configuration, the system blocks further configuration of VSX synchronization associations under the VSX LAG context.

Examples

The first four lines in the following example show the creation and configuration of a VSX LAG. The last two lines of the example show the enabling of VSX synchronization for VSX LAG interface associations and attributes.

```
switch(config)# interface lag 1 multi-chassis
switch(config-lag-if)# access-list ip MY_IP_ACL in
switch(config-lag-if)# rate-limit broadcast 50 kbps
switch(config-lag-if)# qos trust cos
switch(config-lag-if)# exit
switch(config)# vsx
switch(config-vsx)# vsx-sync mclag-interfaces
```

Disabling the VSX synchronization of VSX LAG interface associations and attributes:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync mclag-interfaces
```

vsx-sync nd-snooping

Syntax

```
vsx-sync nd-snooping

no vsx-sync nd-snooping
```

Description

Enables VSX synchronization of ND snooping configurations on the primary VSX node to the secondary peer switch.

To synchronize ND snooping configurations associated with a particular VLAN and interface, configure the same VLAN and interface on the peer device.

The `no` form of this command disables syncing ND snooping configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the ND snooping configurations to the secondary peer:

```
switch(config)# interface 1/1/3
switch(config-if)# no routing
switch(config-if)# nd-snooping trust
switch(config)# vlan 2
switch(config-vlan-2)# nd-snooping
switch(config-vlan-2)# nd-snooping ra-drop
switch(config-vlan-2)# nd-snooping prefix-list 2001::2/64
switch(config)# vsx
switch(config-vsx)# vsx-sync nd-snooping
```

Disabling VSX sync for the ND snooping configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync nd-snooping
```

vsx-sync neighbor

Syntax

```
vsx-sync neighbor

no vsx-sync neighbor
```

Description

Enables VSX synchronization of IPv4 and IPv6 static neighbors configuration on primary VSX node to the secondary peer. There is no configuration sync from secondary to primary peer. If any new modification or additional configuration is made on the primary node for IPv4 and IPv6 static neighbors configuration, they will be auto-synced.

The no form of this command VSX synchronization of IPv4 and IPv6 static neighbors configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the IPv4 and IPv6 static neighbors configurations:

```
DUT-1 (config-vsx)# show run in vlan127
interface vlan127
        ip address 137.1.1.1/16
        ipv6 address 7f00::1/64
        arp ipv4 137.1.1.35 mac 00:12:01:00:00:1a
        arp ipv4 137.1.1.70 mac 00:12:01:00:00:3d
        exit
DUT-1(config-vsx)
switch(config)# vsx
switch(config-vsx)# vsx-sync neighbor
```

Disabling VSX sync for the IPv4 and IPv6 static neighbors configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync neighbor
```

vsx-sync ospf

Syntax

`vsx-sync ospf`

`no vsx-sync ospf`

Description

Enables syncing of OSPF (including OSPFv2 and OPSFv3), route map, and key chain configurations on the primary VSX switch. There is no configuration sync from secondary to primary peer.

To synchronize OSPF configurations at the port level context, configure the same port on the peer device.

The no form of this command disables syncing of OSPF, route map, and key chain configurations to the secondary peer. But it does not remove the previously synced configurations from the secondary peer switch.

NOTE: The OSPF router ID is not synchronized. This exclusion is needed because the router ID uniquely identifies the router. The two OSPF routers with the same router ID do not form an adjacency between them.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the OSPF configurations to the secondary peer:

```
switch(config)# router ospf 1
switch(config-ospf-1)# area 0
switch(config-ospf-1)# area 1 nssa
switch(config-ospf-1)# area 2 stub
switch(config-ospf-1)# redistribute connected route-map map1
switch(config)# router ospfv3 1
switch(config-ospfv3-1)# max-metric router-lsa on-startup
switch(config-ospfv3-1)# bfd all-interfaces
switch(config-if)# ip ospf 1 area 0
switch(config-if)# ip ospf hello-interval 33
switch(config-if)# ipv6 ospfv3 1 area 0
switch(config-if)# ipv6 ospfv3 dead-interval 55
switch(config)# vsx
switch(config-vsx)# vsx-sync ospf
```

Disabling VSX sync for the OSPF configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync ospf
```

vsx-sync policy-global

Syntax

```
vsx-sync policy-global

no vsx-sync policy-global
```

Description

Enables VSX synchronization of global classifier policy configurations on the primary VSX node to the secondary peer switch.

The `no` form of this command disables VSX synchronization of global policy configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the global policy configurations to the secondary peer:

```
switch(config)# apply policy testPolicy in
switch(config)# vsx
switch(config-vsx)# vsx-sync policy-global
```

Disabling VSX sync for the global policy configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync policy-global
```

vsx-sync qos-global

Syntax

```
vsx-sync qos-global
```

```
no vsx-sync qos-global
```

Description

Enables the VSX synchronization of global QoS configurations, such as CoS map, DSCP map, and trust policy, on the primary VSX node to the secondary peer switch. To enable VSX synchronization at the context level for this feature, enter the `vsx-sync qos` command at the context level.

The `no` form of this command removes VSX synchronization of global QoS configurations, but it does not remove the existing global QoS feature configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first five lines in the following example show the setting of global QoS configurations. The last two lines of the example show the enabling of VSX synchronization for global QoS configurations.

```
switch(config)# qos cos-map 1 local-priority 0
switch(config)# qos cos-map 0 local-priority 1
switch(config)# qos cos-map 2 local-priority 2
switch(config)# qos dscp-map 2 local-priority 3
switch(config)# qos trust dscp
switch(config)# vsx
switch(config-vsx)# vsx-sync qos-global
```

Disabling VSX synchronization of global QoS configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync qos-global
```

vsx-sync route-map

Syntax

```
vsx-sync route-map
```

```
no vsx-sync route-map
```

Description

Enables syncing of all As Path lists, community lists, prefix lists, and route map configurations on primary VSX node to the secondary peer switch. There is no configuration sync from the secondary to primary peer.

The no form of this command disables syncing of As Path lists, community lists, prefix lists, and route map configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the route map configurations:

```
switch(config)#  ip aspath-list list1 seq 10 permit 10
switch(config)# ip community-list expanded com1 seq 10 permit 10
switch(config)# ip extcommunity-list standard ext1 seq 10 permit rt 10:4
switch(config)# ip prefix-list pref1 seq 10 permit any
switch(config)# route-map rm1 permit
switch(config-route-map-rm1-10)# match ip next-hop 1.1.1.1
switch(config)# vsx
switch(config-vsx)# vsx-sync route-map
```

Disabling VSX sync for the route map configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync route-map
```

vsx-sync sflow

Syntax

```
vsx-sync sflow

no vsx-sync sflow
```

Description

Enables VSX synchronization of the sFlow configurations on the primary VSX node to the secondary peer.

The no form of this command removes VSX synchronization of global sFlow configurations, but it does not remove the existing global sFlow feature configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

To maintain compliance with sFlow collector functionality for non-VSX topology, the vsx-sync sflow command on primary VSX peer is expected to sync all sFlow configurations, except for the agent-ip configuration. This exclusion is required for sFlow collector functionality to work seamlessly even with VSX topology. To synchronize sFlow configurations associated with particular VRF, you must configure the same VRF on the peer device.

Examples

The first line in the following example shows the setting of an sFlow configuration. The last two lines of the example show the enabling of VSX synchronization for sFlow configurations.

```
switch(config)# sflow agent-ip 10.0.0.100
switch(config)# vsx
switch(config-vsx)# vsx-sync sflow
```

Disabling VSX synchronization of global sFlow configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync sflow
```

vsx-sync sflow-global

Syntax

```
vsx-sync sflow-global
```

```
no vsx-sync sflow-global
```

Description

Enables VSX synchronization of the sFlow global configurations on the primary VSX node to the secondary peer.

To synchronize sFlow configurations associated with a particular VRF, you must configure the same VRF on the peer device.

The `no` form of this command disables VSX synchronization of global sFlow configurations, but it does not remove the existing sFlow feature configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

To maintain compliance with sFlow collector functionality for non-VSX topology, the `vsx-sync sflow` command on primary VSX peer is expected to sync all sFlow configurations, except for the `agent-ip` configuration. This exclusion is required for sFlow collector functionality to work seamlessly even with VSX topology. VSX syncs only the global sFLow configurations and not the sFlow configurations under physical or LAG interfaces.

Examples

The first line in the following example shows the setting of an sFlow configuration. The last two lines of the example show the enabling of VSX synchronization for sFlow configurations.

```
switch(config)# sflow collector 1.1.1.1
switch(config)# vsx
switch(config-vsx)# vsx-sync sflow-global
```

Disabling VSX synchronization of global sFlow configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync sflow-global
```

vsx-sync snmp

Syntax

```
vsx-sync snmp

no vsx-sync snmp
```

Description

Enables VSX synchronization of SNMP configurations on the primary VSX node to the secondary peer. To synchronize SNMP configurations associated with a particular VRF, you must configure the same VRF on the peer device.

The `no` form of this command removes VSX synchronization of global SNMP configurations, but it does not remove the existing global SNMP feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX synchronization for SNMP configuration:

```
switch(config)# vsx
switch(config-vsx)# vsx-sync snmp
```

Disabling VSX synchronization for SNMP configuration:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync snmp
```

vsx-sync ssh

Syntax

```
vsx-sync ssh

no vsx-sync ssh
```

Description

Enables VSX synchronization of SSH server configurations on the primary VSX node to the secondary peer switch. To synchronize SSH configurations associated with particular VRF, you must configure the same VRF on the peer device.

The `no` form of this command removes VSX synchronization of global SSH configurations, but it does not remove the existing global SSH feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first line in the following example shows the setting of an SSH server configuration. The last two lines of the example show the enabling of VSX synchronization for SSH server configurations.

```
switch(config)# ssh certified-algorithms-only
switch(config)# vsx
switch(config-vsx)# vsx-sync ssh
```

Disabling VSX synchronization for global SSH server configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync ssh
```

vsx-sync static-routes

Syntax

```
vsx-sync static-routes

no vsx-sync static-routes
```

Description

Enables VSX synchronization of static route configurations on the primary VSX node to the secondary peer switch. To synchronize static route configurations associated with particular VRF, you must configure the same VRF on the peer switch.

The `no` form of this command removes VSX synchronization of global static route configurations, but it does not remove the existing global static route feature configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX synchronization for static routes:

```
switch(config)# vsx
switch(config-vsx)# vsx-sync static-routes
```

Disabling VSX synchronization for static routes:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync static-routes
```

vsx-sync stp-global

Syntax

```
vsx-sync stp-global
```

```
no vsx-sync stp-global
```

Description

Enables the VSX synchronization of global STP configurations on the primary VSX node to the secondary peer switch. Use the `vsx-sync mclag-interfaces` command to sync context level spanning trees. To enable VSX synchronization at the context level for this feature, enter the `vsx-sync mclag-interfaces` command at the context level.

The `no` form of this command removes VSX synchronization of global STP configurations, but it does not remove the existing global STP feature configurations from the secondary peer switch.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first two lines in the following example show the setting of global STP configurations. The last two lines of the example show the enabling of VSX synchronization for global STP configurations.

```
switch(config)# spanning-tree config-name abc
switch(config)# spanning-tree config-revision 1
switch(config)# vsx
switch(config-vsx)# vsx-sync stp-global
```

Disabling VSX synchronization of global STP configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync stp-global
```

vsx-sync time

Syntax

```
vsx-sync time
```

```
no vsx-sync time
```

Description

Enables VSX synchronization of time-related configurations, including NTP and time zone configurations, on the primary VSX node on the secondary peer switch. To synchronize NTP configurations associated with a particular VRF, you must configure the same VRF on the peer switch.

The `no` form of this command removes VSX synchronization of global time-related configurations, but it does not remove the existing global time-related feature configurations from the secondary peer switch.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

The first two lines in the following example show the setting of time-related configurations. The last two lines of the example show the enabling of VSX synchronization for time-related configurations.

```
switch(config)# ntp authentication
switch(config)# clock timezone utc
switch(config)# vsx
switch(config-vsx)# vsx-sync time
```

Disabling VSX synchronization for time-related configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync time
```

vsx-sync udp-forwarder

Syntax

```
vsx-sync udp-forwarder

no vsx-sync udp-forwarder
```

Description

Enables VSX synchronization of UDP forwarder configurations on the primary VSX node to the secondary peer.

The `no` form of the command disables the VSX synchronization of UDP forwarder configurations to the secondary peer; however, it does not remove the existing udp-forwarder configurations from the secondary VSX peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX synchronization for UDP forwarder configurations:

```
switch(config)# vsx
switch(config-vsx)# vsx-sync udp-forwarder
```

Disabling VSX synchronization for UDP forwarder configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync udp-forwarder
```

vsx-sync vrf

Syntax

```
vsx-sync vrf

no vsx-sync vrf
```

Description

Enables VSX synchronization of VRF configurations on the primary VSX node to the secondary peer switch.

To synchronize VRF configurations, there is no need to configure the same VRF on the secondary peer.

To synchronize VRF configurations associated with a particular L3 interface, no special or extra configuration is required on the primary VSX node.

The `no` form of this command disables syncing VRF configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer.

Command context

```
config-vsx
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the VRF configurations to the secondary peer:

```
switch(config)# vrf finance
switch(config-vrf)# rd 5:15
switch(config-vrf)# address-family ipv4 unicast
switch(config-vrf-ipv4-uc)# route-target import 4:8
switch(config-vrf-ipv4-uc)# route-target export 4:9
switch(config-vrf-ipv4-uc)# route-target both 4:10
switch(config-vrf-ipv4-uc)# exit-address-family
switch(config-vrf)# address-family ipv6 unicast
switch(config-vrf-ipv6-uc)# route-target import 6:8
switch(config-vrf-ipv6-uc)# route-target export 6:9
switch(config-vrf-ipv6-uc)# route-target both 6:10
switch(config-vrf-ipv6-uc)# exit-address-family
switch(config)# interface 1/1/1
switch(config-if)# vrf attach finance
switch(config)# vsx
switch(config-vsx)# vsx-sync vrf
```

Disabling VSX sync for the VRF configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync vrf
```

vsx-sync vrrp

Syntax

```
vsx-sync vrrp

no vsx-sync vrrp
```

Description

Enables VSX synchronization of all VRRP configurations on the primary VSX node to the secondary peer switch. There is no configuration sync from secondary to primary peer.

To synchronize VRRP configurations at the port level context, the same port must be configured on the peer device with IP address.

The `no` form of this command disables syncing VRRP configurations to the secondary peer, but it does not remove the previously synced configurations from the secondary peer.

NOTE: BFD IP is the IP address of VRRP peer device. Hence it cannot be synced.

In the owner scenario, in case the priority is synced, both VSX primary and secondary devices will have 255 as their priority. If the primary device goes down and comes up again, the secondary device will still act as the master in spite of the primary device being the owner. Hence priority cannot be synced.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling VSX sync for the VRRP configurations to the secondary peer:

```
switch(config)# router vrrp enable
switch(config-if)# vrrp 1 address-family ipv4
switch(config-if-vrrp)# address 1.1.1.100 primary
switch(config-if-vrrp)# timers advertise 1000
switch(config-if-vrrp)# no shutdown
switch(config-if)# vrr 1 address-family ipv6
switch(config)# vsx
switch(config-vsx)# vsx-sync vrrp
```

Disabling VSX sync for the VRRP configurations to the secondary peer:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync vrrp
```

vsx-sync vsx-global

Syntax

`vsx-sync vsx-global`

`no vsx-sync vsx-global`

Description

Enables VSX synchronization of global VSX configurations on the primary VSX node to the secondary peer.

The `no` form of the command disables VSX synchronization of global VSX configurations to the secondary peer; however, it does not remove the existing VSX feature configurations from the secondary peer.

Command context

`config-vsx`

Authority

Administrators or local user group members with execution rights for this command.

Usage

The following commands are synced from primary VSX node to secondary VSX node:

- `inter-switch-link dead-interval` *`<DEAD-INTERVAL>`*
- `inter-switch-link hello-interval` *`<HELLO-INTERVAL>`*
- `inter-switch-link hold-time` *`<HOLD-INTERVAL>`*
- `inter-switch-link peer-detect-interval` *`<PEER-DETECT-INTERVAL>`*
- `keepalive dead-interval` *`<DEAD-INTERVAL>`*
- `keepalive hello-interval` *`<HELLO-INTERVAL>`*
- `keepalive udp-port` *`<PORT-NUM>`*
- `linkup-delay-timer` *`<DELAY-TIMER>`*
- `split-recovery`
- `system-mac` *`<MAC-ADDR>`*

Examples

The first three lines in the following example show the setting of global VSX configurations. The last line in the example shows the enabling of VSX synchronization for global VSX configurations.

```
switch(config)# vsx
switch(config-vsx)# inter-switch-link dead-interval 15
switch(config-vsx)# inter-switch-link hello-interval 2
switch(config-vsx)# inter-switch-link hold-time 1
switch(config-vsx)# vsx-sync vsx-global
```

Disabling VSX synchronization for global VSX configurations:

```
switch(config)# vsx
switch(config-vsx)# no vsx-sync vsx-global
```

vsx update-software

Syntax

`vsx update-software` *`<REMOTE-URL>`* `[vrf` *`<VRF-NAME>`*`]`

Description

This command lets you update the software.

Command context

Manager (#)

Parameters

<REMOTE-URL>

Specifies the TFTP URL for downloading the software. Syntax: `tftp://{<IP-ADDRESS>|<HOSTNAME>}[:<PORT>][;blocksize=<VAL>]/<FILE-NAME>`

`vrf` ***<VRF-NAME>***

Specifies the VRF name for downloading the software. Optional

Authority

Administrators or local user group members with execution rights for this command.

Usage

This command gives you the option to save the running configuration on the primary and secondary VSX switches. After the command saves the running configuration, it downloads new software from the TFTP server and verifies the download. After a successful verification, the command installs the software to the alternative image of both the VSX primary and secondary switches.

The command displays the status of the VSX primary and secondary switches during the upgrade. The command also refreshes the progress bar as the image update progresses. Do not interrupt the VSX primary CLI session until the software updates completes; however, software update process can be stopped. If you stop the upgrade when the secondary switch has already installed the image in its flash memory or the secondary switch has started the reboot the process, it comes up with the new software. The primary switch continues to have with older software. You can stop the software update process by pressing **ctrl+c**.

Example

Updating the software using TFTP:

```
switch# vsx update-software tftp://192.168.1.1/XL.10.0x.xxxx vrf mgmt
Do you want to save the current configuration (y/n)? y
The running configuration was saved to the startup configuration.

This command will download new software to the %s image of both the VSX primary and secondary systems,
then reboot them in sequence. The VSX secondary will reboot first, followed by the primary.
Continue (y/n)? y
VSX Primary Software Update Status     : <VSX primary software update status>
VSX Secondary Software Update Status   : <VSX secondary software update status>
VSX ISL Status                         : <VSX ISL status>
Progress [..........................................................................................]
Secondary VSX system updated completely. Rebooting primary.
```

vsx update-software boot-bank

Syntax

`vsx update-software boot-bank {primary | secondary}`

Description

Upgrades the VSX pairs using the specified bank on both the devices. This command compares whether the image versions are same in both the primary and secondary switches and reboots them in sequence, the VSX secondary switch followed by VSX primary switch.

NOTE: Before executing this command, download the software image and install in the required boot banks.

Command context

Manager (#)

Parameters

boot-bank

Specifies the boot bank where the image is pre-staged .

{primary | secondary}

Selects either primary or secondary VSX switch for the software upgrade.

Authority

Administrators or local user group members with execution rights for this command.

Usage

This command gives you the option to save the running configuration on the primary and secondary VSX switches. After the command saves the running configuration, it downloads new software from the TFTP server and verifies the download. After a successful verification, the command installs the software to the alternative image of both the VSX primary and secondary switches.

The command displays the status of the VSX primary and secondary switches during the upgrade. The command also refreshes the progress bar as the image update progresses. Do not interrupt the VSX primary CLI session until the software updates completes; however, software update process can be stopped. If you stop the upgrade when the secondary switch has already installed the image in its flash memory or the secondary switch has started the reboot the process, it comes up with the new software. The primary switch continues to have with older software. You can stop the software update process by pressing **ctrl+c**.

Example

Selecting primary bank for upgrade:

```
switch# vsx update-software boot-bank primary

Do you want to save the current configuration (y/n)? y
The running configuration was saved to the startup configuration.

This command will upgrade both VSX primary and secondary systems, using pre-staged
image 'X' installed in secondary bank on both devices, then reboot
them in sequence. The VSX secondary will reboot first, followed by primary.
Continue (y/n)? y
VSX Primary Software Update Status     : Reboot started
VSX Secondary Software Update Status   : Image updated successfully
VSX ISL Status                         : Up
Progress [..........................................................................................]
Secondary VSX system updated completely. Rebooting primary.
```

Selecting secondary bank for upgrade:

```
switch# vsx update-software boot-bank secondary

Do you want to save the current configuration (y/n)? y
The running configuration was saved to the startup configuration.

This command will upgrade both VSX primary and secondary systems, using pre-staged
image 'X' installed in secondary bank on both devices, then reboot
them in sequence. The VSX secondary will reboot first, followed by primary.
Continue (y/n)? y
VSX Primary Software Update Status     : Reboot started
VSX Secondary Software Update Status   : Image updated successfully
VSX ISL Status                         : Up
Progress [..........................................................................................]
Secondary VSX system updated completely. Rebooting primary.
```

ISL is out-of-sync

Solution 1

Cause

Mismatch with the ISL version or switch platform or both.

Action

1. Run the `show vsx status` command.

   In the following example, the ISL channel is shown as in-sync; however, if the ISL channel was not in-sync, a different status would be provided.

```
switch# show vsx status
VSX Operational State
---------------------
  ISL channel               : In-Sync
  ISL mgmt channel          : operational
  Config Sync Status        : in-sync
  NAE                       : peer_unreachable
  HTTPS Server              : peer_unreachable

Attribute           Local                       Peer
------------        --------                    --------
ISL link            1/1/43                      1/1/43
ISL version         2                           2
System MAC          48:0f:cf:af:70:84           48:0f:cf:af:c2:84
Platform            8320                        8320
Software Version    10.0x.xxxx                  10.0x.xxxx
Device Role         primary                     secondary
```

```
switch(config)# user admin password
Changing password for user admin
Enter password:************
Confirm password:************
```

2. If there is an ISL version mismatch, update the software so the ISL version is the same on the local and peer VSX switch.
3. If the switches have mismatching platforms, create an ISL link that connects two VSX switches with the same platform.

Solution 2

Cause

The role is not configured on any of the VSX switches or the same role is configured on both VSX switches.

Action

1. Run the `show vsx status` command.

If the roles are set incorrectly, the command displays the `role inconsistent` status.

2. Set the roles correctly so that one of the VSX switches has the primary role and the other switch has the secondary role. To set a switch role, enter the `role {primary | secondary}` command.

Solution 3

Cause

The ISL interface is down on any one switch in the VSX pair.

Action

1. Check ISL state and ISL link status by entering: `switch#` **`show vsx status inter-switch-link`**

   In the following example, the ISL state and link status are shown as in-sync and up; however, if the ISL interface is down, a different status would be provided.

   ```
   switch# show vsx status inter-switch-link
   State                     : In-Sync
   Link Status               : up
   Mgmt state                : operational

   Inter-switch link Statistics
   ----------------------------
   Hello Packets Tx          : 4572
   Hello Packets Rx          : 4573
   Data Packets Tx           : 80634
   Data Packets Rx           : 80637
   Mgmt Packets Tx           : 25946
   Mgmt Packets Rx           : 25167
   Mgmt Packet Drops         : 0
   ```

2. Re-enable the ISL interface by going to that interface context and entering `no shutdown`:

   ```
   switch(config)# interface 1/1/1
   switch(config-if)# no shutdown
   ```

ISL is in blocking state

Symptom

The VSX LAGs are shown as `Blocking` in the output of the `show spanning-tree detail` command.

```
switch# show spanning-tree detail
Spanning tree status : Enabled Protocol: MSTP

MST0
Root ID Priority : 32768
      MAC-Address: 02:02:02:02:02:02
      This bridge is the root
      Hello time (in seconds):2 Max Age (in seconds):20
      Forward Delay (in seconds):15

Bridge ID Priority: 32768
     MAC-Address: 02:02:02:02:02:02
     Hello time (in seconds):2 Max Age (in seconds):20
     Forward Delay (in seconds):15

Port   Role     State    Cost    Priority Type
```

```
------ -------- -------- ------ -------- -------
lag1   Disabled Blocking 20000  64       shared
lag100 Disabled Blocking 20000  64       shared

Topology change flag : False
Number of topology changes : 0
Last topology change occurred: 3876 seconds ago
Timers: Hello expiry 0, Forward delay expiry 0

Port lag1 id 321
Designated root has priority : 32768 Address: 02:02:02:02:02:02
Designated bridge has priority : 32768 Address: 02:02:02:02:02:02
Designated port id : 0
Multi-Chassis role : active

Number of transitions to forwarding state: 0
Bpdus sent 0, received 0

Port lag100 id 420
Designated root has priority : 32768 Address: 02:02:02:02:02:02
Designated bridge has priority : 32768 Address: 02:02:02:02:02:02
Designated port id : 0
Number of transitions to forwarding state: 0
Bpdus sent 0, received 0
```

Cause

- Mismatch MSTP configurations on VSX peer switches.
- Switches are not in the same MSTP region within the VSX environment.
- STP configurations on VSX LAG ports must be the same on VSX switches.
- The VSX pair is configured as a nonroot switch.

Action

1. Run the following commands for determining what is causing the ISL to be in a blocking state:

   ```
   switch# show running-config spanning-tree
   switch# show spanning-tree mst-config
   switch# show vsx status
   ```

   When you run the `show vsx status` command, verify that the ISL is in-sync.

2. Verify that the VSX peer switches are in the active and standby role when the ISL is the `in-sync` state by entering the `show spanning-tree detail` command:

   ```
   switch# show spanning-tree detail
   Spanning tree status : Enabled Protocol: MSTP

   MST0
   Root ID Priority : 32768
         MAC-Address: 02:02:02:02:02:02
         This bridge is the root
         Hello time (in seconds):2 Max Age (in seconds):20
         Forward Delay (in seconds):15

   Bridge ID Priority: 32768
        MAC-Address: 02:02:02:02:02:02
        Hello time (in seconds):2 Max Age (in seconds):20
        Forward Delay (in seconds):15
   ```

```
Port   Role     State    Cost   Priority Type
------ -------- -------- ------ -------- -------
lag1   Disabled Blocking 20000  64       shared
lag100 Disabled Blocking 20000  64       shared

Topology change flag : False
Number of topology changes : 0
Last topology change occurred: 3876 seconds ago

Timers: Hello expiry 0, Forward delay expiry 0

Port lag1 id 321
Designated root has priority : 32768 Address: 02:02:02:02:02:02
Designated bridge has priority : 32768 Address: 02:02:02:02:02:02
Designated port id : 0
Multi-Chassis role : active
Number of transitions to forwarding state: 0
Bpdus sent 0, received 0
```

3. Verify if MSTP configurations are the same on the VSX peer switches by entering the following commands:

```
switch# show running-config spanning-tree
switch# show running-config spanning-tree vsx-peer
```

4. In a converged network, if any of MSTP ports are disabled by loop protect because different instances have different root switches, remove loop protect configuration from those ports.
5. Preferably, enable loop-protect on only edge ports or ports connected to STP unaware switches.

   The admin path cost configured on downstream switches results in the VSX pair seeing the root switch as equal cost to the root switch from both VSX pair switches.

Traffic drop on a VSX LAG interface

Action

1. Verify that the VSX LAG interface is in-sync with both peer and down stream switches by entering:

```
switch# show lacp interfaces multi-chassis
```

2. Verify that the VLAN membership of the VSX is the same on both VSX switches by entering:

```
switch# show vsx config-consistency lacp <LAG-NAME>
```

3. Verify that all MAC addresses are programmed correctly on both VSX switches by entering:

```
switch# show mac-address-table
switch# show mac-address-table count
```

Traffic loss after the ISL has been out-of-sync and keepalive is down

Symptom

Traffic loss is seen after the ISL has been out-of-sync and keepalive is down.

Cause

If the ISL becomes out-of-sync and keepalive is established, the secondary VSX LAGs are brought down. If keepalive then fails and you have split recovery mode enabled (default setting), the secondary switch brings up its VSX LAGs. This split condition leads to traffic loss because of the asymmetric traffic flow.

Action

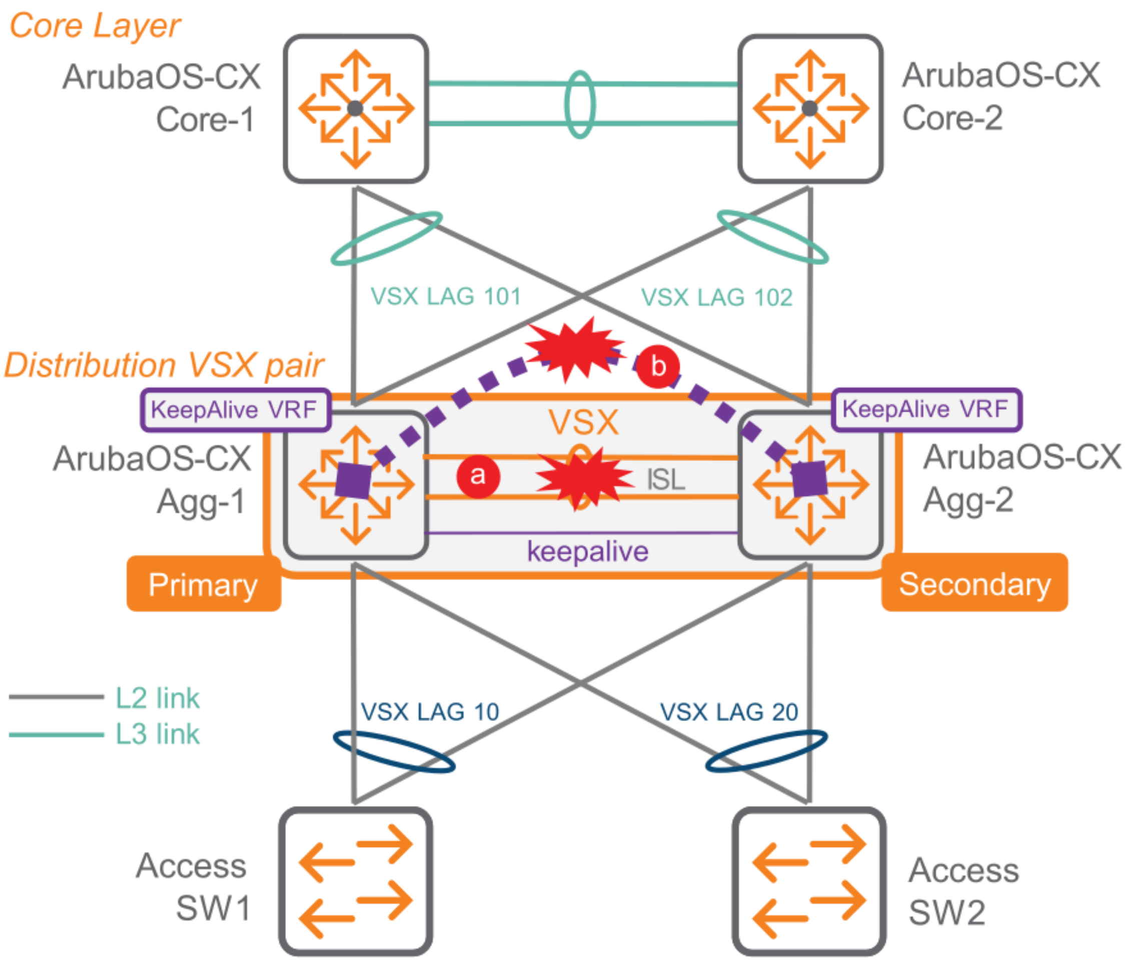
1. Disable split recovery mode by entering the following command on the secondary VSX switch:

   ```
   switch(config)# vsx
   switch(config-vsx)# no split-recovery
   ```

   This command shuts down the secondary link to stop the asymmetric traffic flow.

2. Apply the same setting on the primary VSX switch for consistency and in case there is a primary/secondary role swap in the configuration.

Failure scenarios and split recovery



Failure scenarios	Result with split recovery off	Result with split recovery on (default)
Keepalive is down and ISL is up. This scenario is shown with label "b" in the figure.	No impact, except for the loss of the split detection.	No impact, except for the loss of the split detection.
ISL is down, but keepalive is up. This scenario is shown with label "a" in the figure.	Secondary VSX switch brings down VSX LAG member ports.	Secondary VSX switch brings down VSX LAG member ports.
ISL is down, but keepalive is up, as shown with label "a" in the figure. Then, after sometime, keepalive also goes down, as shown with label b.	Secondary VSX switch brings down VSX LAG member ports. Then, the secondary VSX LAGs stay down.	Label a: The secondary VSX switch brings down VSX LAG member ports. Label b: The secondary VSX switch restores VSX LAG member ports.
At the same time, ISL goes down and keepalive goes down, as shown with label "a" and "b" in the following figure. Then, keepalive is restored, as shown with label b.	Label a and b: All VSX LAG ports stay up. Label b: Then, the secondary VSX switch brings down the VSX LAG member ports.	Label a and b: All VSX LAG ports stay up. Label b: Then, the secondary VSX switch brings down the VSX LAG member ports.

Active gateway is unreachable

Symptom

You are unable to ping the active gateway.

Action

1. Verify that kernel interface is created for active gateway:

```
003000000000001@vlan3: <NO-CARRIER,BROADCAST,UP,M-DOWN> mtu 1500 qdisc noqueue state
LOWERLAYERDOWN group default qlen 1000
    link/ether 00:00:00:00:00:01 brd ff:ff:ff:ff:ff:ff
```

2. Verify that the active gateway IP is programmed correctly on kernel interface:

```
003000000000001@vlan3: <NO-CARRIER,BROADCAST,UP,M-DOWN> mtu 1500 qdisc noqueue state
LOWERLAYERDOWN group default qlen 1000
     link/ether 00:00:00:00:00:01 brd ff:ff:ff:ff:ff:ff
     inet 10.0.0.3/32 scope global 003000000000001
     valid_lft forever preferred_lft forever
     inet 20.0.0.3/32 scope global 003000000000001
     valid_lft forever preferred_lft forever
```

3. Verify that the active gateway VMAC is correctly configured in the hardware. The command used is platform-specific. Refer to the hardware documentation for your switch.

   The following example from the 8400 series switch:

```
8400X:/home/admin# ovs-appctl l3pd/show router-mac -a
```

BFD reports a LAG as down even when healthy links are still available

Symptom

The Bidirectional Forward Detection (BFD) feature reports a Link Aggregation (LAG), as being down, even though there are healthy LAG links available. The LAG, containing the downed link, will eventually rebalance the traffic to its other links.

Cause

This notification occurs when the minimum BFD control packet reception interval is set at a faster rate than the Link Aggregation Control Protocol (LACP) rate and LAG rebalancing occurs. BFD assumes that the link is down without realizing that the LAG is rebalancing the traffic load.

Action

1. Set the minimum BFD control packet reception interval to a slower rate than the LACP rate or set the LACP rate to a faster rate than the minimum BFD control packet reception interval.

    a. To find the current settings of the minimum BFD control packet reception interval, enter the `show running-config` command.

       The minimum BFD control packet reception interval setting is listed as `bfd min-receive-interval` in the command output and the measurement is in ms.

    b. To find the current rate of LACP, enter the `show lacp aggregates` command.

       The LACP rate is listed as the `Heatbeat rate` in the command output.

    c. To change the minimum BFD control packet reception interval, enter the `bfd min-receive-interval` command.

    d. To change the LACP rate, enter the `lacp rate {fast | slow}` command.

Accessing Aruba Support

Aruba Support Services	**https://www.arubanetworks.com/support-services/**
Aruba Support Portal	**https://asp.arubanetworks.com/**
North America telephone	1-800-943-4526 (US & Canada Toll-Free Number) +1-408-754-1200 (Primary - Toll Number) +1-650-385-6582 (Backup - Toll Number - Use only when all other numbers are not working)
International telephone	**https://www.arubanetworks.com/support-services/contact-support/**

Be sure to collect the following information before contacting Support:

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Other useful sites

Other websites that can be used to find information:

Airheads social forums and Knowledge Base	**https://community.arubanetworks.com/**
Software licensing	**https://lms.arubanetworks.com/**
End-of-Life information	**https://www.arubanetworks.com/support-services/end-of-life/**
Aruba software and documentation	**https://asp.arubanetworks.com/downloads**

Accessing updates

To download product updates:

Aruba Support Portal

https://asp.arubanetworks.com/downloads

If you are unable to find your product in the Aruba Support Portal, you may need to search My Networking, where older networking products can be found:

My Networking

https://www.hpe.com/networking/support

To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

https://support.hpe.com/portal/site/hpsc/aae/home/

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

To subscribe to eNewsletters and alerts:

https://asp.arubanetworks.com/notifications/subscriptions (requires an active Aruba Support Portal (ASP) account to manage subscriptions). Security notices are viewable without an ASP account.

Warranty information

To view warranty information for your product, go to **https://www.arubanetworks.com/support-services/product-warranties/**.

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at **https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts**

Additional regulatory information

Aruba is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements, environmental data (company programs, product recycling, energy efficiency), and safety information and compliance data, (RoHS and WEEE). For more information, see **https://www.arubanetworks.com/company/about-us/environmental-citizenship/**.

Documentation feedback

Aruba 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-switching@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.