CONFIGURATION GUIDE

Brocade SLX-OS
Layer 2 Configuration Guide, 16r.1.00

Supporting the Brocade SLX 9850 Router

53-1004418-01
31 August 2016

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

The product described by this document may contain open source software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit http://www.brocade.com/support/oscd.

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.
Click the **Support** tab and select **Document Library** to access documentation on MyBrocade or www.brocade.com You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com
- By sending your feedback to documentation@brocade.com

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/solution provider.

About this document

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Brocade Communications Systems, Inc. for SLX-OS Release 16r.1.00, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release:

- Brocade SLX 9850-4 router
- Brocade SLX 9850-8 router

To obtain information about other Brocade OS versions, refer to the documentation specific to that version.

802.1d Spanning Tree Protocol

Spanning Tree Protocol overview

The Spanning Tree Protocol (STP) prevents Layer 2 loops in a network by providing redundant links. If a primary link fails, the backup link is activated and network traffic is not affected. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs.

The IEEE 802.1d Spanning Tree Protocol (STP) runs on bridges and switches that are 802.1d-compliant. This and the following chapters addresses both basic STP and its variants.

These variants are Rapid STP (RSTP), Multiple STP (MSTP), Per-VLAN Spanning Tree Plus (PVST+), and Rapid-PVST+ (R-PVST+)

When the spanning tree algorithm is run, the network switches transform the real network topology into a spanning tree topology. In an STP topology any LAN in the network can be reached from any other LAN through a unique path. The network switches recalculate a new spanning tree topology whenever there is a change to the network topology.

For each LAN, the switches that attach to the LAN choose a designated switch that is the closest to the root switch. The designated switch forwards all traffic to and from the LAN. The port on the designated switch that connects to the LAN is called the designated port. The switches decide which of their ports is part of the spanning tree. A port is included in the spanning tree if it is a root port or a designated port.

Spanning Tree Protocol configuration notes

Enabling the he Spanning Tree Protocol (STP) creates a loop free topology of Ethernet LANs connected by bridge devices.

The Brocade device supports STP as described in the IEEE 802.1d-1998 specification.

The STP is disabled by default on the Brocade device. Thus, any new VLANs you configure on the Brocade device have STP disabled by default.

Optional features

The following STP configuration features are optional:

- *Root guard*
- *BPDU guard* and *BPDU filter*
- *PortFast*

STP states

Each Layer 2 interface participating in a spanning tree is in one of five states.

A network topology of bridges typically contains redundant connections to provide alternate paths in case of link failures. The redundant connections create a potential for loops in the system. As there is no concept of time to live (TTL) in Ethernet frames, a situation may arise where there is a permanent circulation of frames when the network contains loops. To prevent this, a spanning tree connecting all the bridges is formed in real time.

Every Layer 2 interface running STP is in one of these states:

- *Blocking* — The interface does not forward frames. Redundant ports are put in a blocking state and enabled when required. This is a transitional state after initialization.
- *Listening* — The interface is identified by the spanning tree as one that should participate in frame forwarding. This is a transitional state after the blocking state.
- *Learning* — The interface prepares to participate in frame forwarding. This is a transitional state after the listening state.
- *Forwarding* — The interface forwards frames. This is a transitional state after the learning state.
- *Disabled* — The interface is not participating in spanning tree because of shutdown of a port or the port is not operationally UP. Any of the other states may transition into this state.

Bridge ports must exchange control frames to build a spanning tree, these are called Bridge Protocol data units (BPDU).

BPDUs

To build a spanning tree for the bridge topology, the bridges must exchange control frames called Bridge Protocol data units (BPDU).

To construct a spanning tree requires knowledge of the all the participants. The bridges have to determine the root bridge and compute the port roles (root, designated, or blocked) with only the information that they have. To ensure that each bridge has enough information, the bridges use BPDUs to exchange information about bridge IDs and root path costs.

A bridge sends a BPDU frame using the unique MAC address of the port itself as a source address, and a destination address of the STP multicast address 01:80:C2:00:00:00.

BPDUs are exchanged regularly (every 2 seconds by default) and enable switches to keep track of network changes and to start and stop forwarding through ports as required.

When a device is first attached to a switch port, it does not immediately forward data. It instead goes through a number of states while it processes inbound BPDUs and determines the topology of the network. When a host is attached, after a listening and learning delay of about 30 seconds, the port always goes into the forwarding state. The time spent in the listening and learning states is determined by the forward delay. However, if instead another switch is connected, the port may remain in blocking mode if it would cause a loop in the network.

There are four types of BPDUs in the original STP specification:

- Configuration BPDU (CBPDU) is used for spanning tree computation.
- Topology Change Notification (TCN) BPDU is used to announce changes in the network topology.
- RSTP BPDU is used for RSTP
- MSTP BPDU is used for MSTP

TCN BPDUs

TCN BPDUs are used to inform other switches of port changes.

TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

Consider these configuration rules:

- TCN BPDUs are sent per VLAN.
- TCN BPDUs are sent only in those VLANs in which a topology change is detected.
- TCN BPDUs are sent only in those VLANs for which the bridge is not the root bridge.
- If a topology change is detected on a VLAN for which the bridge is the root bridge, the topology change flag is set in the configuration BPDUs that is sent out.

For a given link, in conjunction with the above rules, a TCN BPDU is sent out as follows:

- On an access port, only standard IEEE TCN BPDU is sent out. This TCN BPDU corresponds to a topology change in the access VLAN.
- On a trunk port, if VLAN 1 is allowed (either untagged or untagged), a standard IEEE TCN BPDU is sent for VLAN 1.
- On a trunk port, if the native VLAN is not 1, an untagged TCN BPDU is sent to Cisco/Brocade proprietary MAC address for that VLAN.
- On a trunk port, a tagged TCN BPDU is sent to Cisco/Brocade proprietary MAC address for a tagged VLAN.

As part of the response to TCN BPDUs, the Topology Change and Topology Change Acknowledgment flags are set in all configuration BPDUs corresponding to the VLAN for which the TCN was received.

When a topology change is detected on a trunk port, it is similar to detecting topology changes in each VLAN that is allowed on that trunk port. TCN BPDUs are sent for each VLAN as per the rules.

STP configuration guidelines and restrictions

Follow these configuration guidelines and restrictions when configuring STP and STP variants.

- Only one form of a standing tree protocol, such as STP or RSTP, can be enabled at a time.
- When any form of STP is enabled globally, that form of STP is enabled by default on all switch ports.
- LAGs are treated as normal links for any form of STP.

STP features

Root guard

Root Guard can be used to predetermine a root bridge location and prevent rogue or unwanted switches from becoming the root bridge.

At times it is necessary to protect the root bridge from malicious attack or even unintentional misconfigurations where a bridge device that is not intended to be root bridge becomes root bridge causing severe bottlenecks in data path. These types of mistakes or attacks can be avoided by configuring root guard feature on ports of the root bridge.

The root guard feature provides a way to enforce the root bridge placement in the network and allows STP and its variants to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires. Errors are triggered if any change from the root bridge placement is detected.

When root guard is enabled on a port, it keeps the port in designated FORWARDING state. If the port receives a superior BPDU, which is a root guard violation, it sets the port into a DISCARDING state and triggers a Syslog message and an SNMP trap. No further traffic will be forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or wrongly configured STP or RSTP bridges.

Root guard should be configured on all ports where the root bridge should not appear. In this way, the core bridged network can be cut off from the user network by establishing a protective perimeter around it.

Once the port stops receiving superior BPDUs, root guard automatically sets the port back to a FORWARDING state after the timeout period has expired.

BPDU guard

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow.

In a valid configuration, edge port configured interfaces do not receive BPDUs. If an edge port configured interface receives a BPDU, an invalid configuration exists, such as connection of an unauthorized device. The BPDU guard feature provides a secure response to invalid configurations because the administrator must manually put the interface back in service.

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Brocade device port to which the end station is connected. The STP BPDU guard shuts down the port and puts it into an "error disabled" state. This disables the connected device’s ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a message is displayed to warn the network administrator of an invalid configuration.

The BPDU guard feature provides a secure response to invalid configurations because the administrator must manually put the interface back in service with the **no shutdown** command if error disable recovery is not enabled by enabling the **errdisable-timeout** feature. The interface can also be automatically configured to be enabled after a timeout. However, if the offending BPDUs are still being received, the port is disabled again.

Expected behavior in an interface context

When this feature is enabled on an interface the device is expected to put the interface in Error Disabled state when BPDU is received on the port when edge-port and BPDU guard is enabled on the switch interface. When the port ceases to receive the BPDUs, it does not automatically switch to edge port mode, you must configure **error disable timeout** or **no shutdown** on the port to move the port back into edge port mode.

BPDU filter

BPDU filtering allows you to avoid transmitting BPDUs on PortFast or edge port enabled ports that are connected to an end system.

When you enable edge port on the switch, spanning tree places ports in the forwarding state immediately, instead of going through the listening, learning, and forwarding states. By default, spanning tree sends BPDUs from all ports regardless of whether edge port is enabled.

Error disable recovery

A port is placed into an error-disabled state when:

- A BPDU guard violation or loop detection violation occurs
- The number of inError packets exceeds the configured threshold
- An EFM-OAM enabled interface receives a critical event from the remote device (functionally equivalent to a disable state)

Once in an error disable state, the port remains in that state until it is re-enabled automatically or manually.

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, you can specify the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds. By default, the timeout feature is disabled.

PortFast

PortFast allows an interface to transition quickly to the forwarding state.

Consider the following when configuring PortFast:

- Do not enable port fast on ports that connect to other devices.
- Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat this configuration for every port connected to workstations or PCs.
- Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.
- If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown**/**no shutdown** command.
- PortFast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

STP per VLAN group

STP per VLAN group overview

STP per VLAN group provides scalability. It allows you to both group VLANs and apply the same STP parameter settings to all the VLANs in the group.

STP per VLAN group overcomes the limitations of scalability alternatives.

- Standard STP — You can configure up to 254 instances of standard STP on a Brocade device. In large configurations you may require even more instances than standard STP provides. STP per VLAN group overcomes this limitation by allowing you to aggregate STP instances.
- Single STP — Single STP allows all the VLANs to run STP. However, each VLAN runs the same instance of STP. This results in numerous blocked ports that do not pass any Layer 2 traffic. STP per VLAN group uses all available links by load balancing traffic for different instances of STP on different ports. A port that blocks traffic for one spanning tree forwards traffic for another spanning tree.

The following figure shows an example of a STP per VLAN group implementation.

FIGURE 1 STP per VLAN group example

A master VLAN contains one or more member VLANs. Each of the member VLANs in the STP group runs the same instance of STP and uses the STP parameters configured for the master VLAN. In this example, the switch is configured with VLANs 3, 4, 13, and 14.

- VLANs 3 and 4 are grouped in master VLAN 2, which is in STP group 1.

- VLANs 13 and 14 are grouped in master VLAN 12, which is in STP group 2. These VLANs share a different spanning tree.

All the ports are tagged. The ports must be tagged so that they can be in both a member VLAN and the member's master VLAN. For example, ports 1/1/1 - 1/1/4 are in member VLAN 3 and also in master VLAN 2 (since master VLAN 2 contains member VLAN 3).

STP load balancing

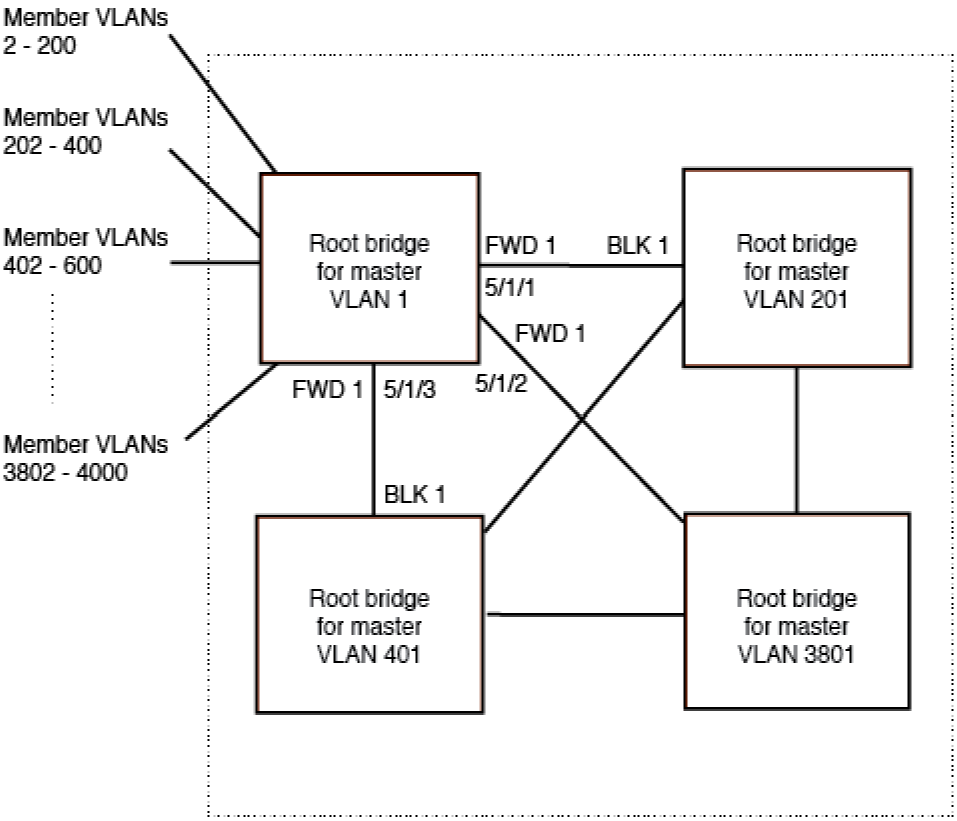
You can use the STP priorities to load balance the STP traffic

Each STP group has different STP priorities. You can set the STP priorities in the same STP group to different values on each device. This makes each of the devices, the root bridge for a different STP group. This type of configuration distributes the traffic evenly across the devices and also ensures that ports that are blocked in one STP group spanning tree are used by another STP group spanning tree for forwarding. Refer to STP load sharing configuration example on page 21 for an example using STP load sharing.

STP load sharing

The following figure shows another example of a STP per VLAN group implementation.

FIGURE 2 More complex STP per VLAN group example



In this example, each of the devices in the core is configured with a common set of master VLANs, each of which contains one or more member VLANs. Each of the member VLANs in an STP group runs the same instance of STP and uses the STP parameters configured for the master VLAN.

The STP group ID identifies the STP instance. All VLANs within an STP group run the same instance of STP. The master VLAN specifies the bridge STP parameters for the STP group, including the bridge priority. In this example, each of the devices in the core is configured to be the default root bridge for a different master VLAN. This configuration ensures that each link can be used for forwarding some traffic. For example, all the ports on the root bridge for master VLAN 1 are configured to forward BPDUs for master VLAN spanning tree. Ports on the other devices block or forward VLAN 1 traffic based on STP convergence. All the ports on the root bridge for VLAN 2 forward VLAN 2 traffic, and so on.

All the ports are tagged. The ports must be tagged so that they can be in both a member VLAN and the member's master VLAN. For example, port 1/1/1 - and ports 5/1/1, 5/1/2, and 5/1/3 are in member VLAN 2 and master VLAN 1 (since master VLAN a contains member VLAN 2).

Configuring the root bridge master VLANs for STP load sharing

These commands configure the root bridge for master VLAN 1 as shown in the Figure 2 on page 18. The first group of commands configures the master VLANs. Notice that the STP priority is set to a different value for each VLAN. In addition, the same VLAN has a different STP priority on each device. This provides load balancing by making each of the devices a root bridge for a different spanning tree.

1. Configure master VLANs.
   a) Configure master VLAN 1.

```
device# configure terminal
device(config)# vlan 1
device(config-vlan-1)# spanning-tree priority 1
device(config-vlan-1)# tagged ethernet 1/1/1 ethernet 5/1/1 to 5/1/3
```

   Set the bridge priority for each master VLAN to the highest priority (1) on one of the devices in the STP per VLAN group configuration. By setting the bridge priority to the highest priority, you make the device the default root bridge for the spanning tree. To ensure STP load balancing, make each of the devices the default root bridge for a different master VLAN.

2. Configure master VLAN 201.

```
device(config-vlan-1)# vlan 201
device(config-vlan-201)# spanning-tree priority 2
device(config-vlan-201)# tagged ethernet 1/1/2 ethernet 5/1/1 to 5/1/3
```

3. Configure master VLAN 401.

```
device(config-vlan-201)# vlan 401
device(config-vlan-401)# spanning-tree priority 3
device(config-vlan-401)# tagged ethernet 1/1/3 ethernet 5/1/1 to 5/1/3
...
```

   Continue in this manner to the last master VLAN.

4. Configure master VLAN 3801.

```
...
device(config-vlan-3601)# vlan 3801
device(config-vlan-3801)# spanning-tree priority 20
device(config-vlan-3801)# tagged ethernet 1/1/20 ethernet 5/1/1 to 5/1/3
device(config-vlan-3801)# exit
```

Configuring member VLAN groups for STP load sharing on page 19.

Configuring member VLAN groups for STP load sharing

Configure the root bridge master VLANs for STP load sharing.

The next group of commands configures VLAN groups for the member VLANs. Notice that the VLAN groups do not contain the VLAN numbers assigned to the master VLANs. Also notice that no STP parameters are configured for the groups of member VLANs. Each group of member VLANs inherits its STP settings from its master VLAN.

1. Create VLAN group 1 and assign VLANS 2 through 200 to it.

   ```
   device# configure terminal
   device(config)# vlan-group 1 vlan 2 to 200
   device(config-vlan-group-1)# tagged ethernet 1/1/1 ethernet 5/1/1 to 5/1/3
   ```

2. Create VLAN group 2 and assign VLANS 202 through 400 to it.

   ```
   device(config-vlan-group-1)# vlan-group 2 vlan 202 to 400
   device(config-vlan-group-2)# tagged ethernet 1/1/2 ethernet 5/1/1 to 5/1/3
   ```

3. Create VLAN group 3 and assign VLANS 402 through 600 to it.

   ```
   device(config-vlan-group-2)# vlan-group 3 vlan 402 to 600
   device(config-vlan-group-2)# tagged ethernet 1/1/3 ethernet 5/1/1 to 5/1/3
   ...
   ```

4. Continue in this manner until all but the master VLANs (configures in the previous topic) are assigned to a group.

   ```
   ...
   device(config-vlan-group-19)# vlan-group 20 vlan 3082 to 3282
   device(config-vlan-group-20)# tagged ethernet 1/1/20 ethernet 5/1/1 to 5/1/3
   device(config-vlan-group-20)# exit
   ```

Configuring groups for STP load sharing on page 20

Configuring groups for STP load sharing

Configuring member VLAN groups for STP load sharing on page 19.

These commands configure the STP groups. Each STP group in this configuration contains one master VLAN, which contains a VLAN group. This example shows that an STP group also can contain additional VLANs (VLANs not configured in a VLAN group).

1. Configure STP group 1.

   ```
   device# configure terminal
   device(config)# stp-group 1
   device(config-stp-group-1)# master-vlan 1
   device(config-stp-group-1)# member-group 1
   device(config-stp-group-1)# member-vlan 4001 4004 to 4010
   ```

2. Configure STP group 2.

   ```
   device(config-stp-group-1)# stp-group 2
   device(config-stp-group-2)# master-vlan 201
   device(config-stp-group-2)# member-group 2
   device(config-stp-group-2)# member-vlan 4002 4003 4011 to 4015
   ```

3. Configure STP group 3.

   ```
   device(config-stp-group-2)# stp-group 3
   device(config-stp-group-3)# master-vlan 401
   device(config-stp-group-3 # member-group 3
   ...
   ```

   Continue in this manner until all STP groups are configured.

4. Configure STP group 20.

   ```
   ...
   device(config-stp-group-19)# stp-group 20
   device(config-stp-group-20)# master-vlan 3081
   device(config-stp-group-20)# member-group 20
   ```

STP load sharing configuration example

1. The first group of commands configure the master VLANs.
2. The next section of commands configure VLAN groups for the member VLANs.
3. The final commands configure the STP groups.

```
device# configure terminal
device(config)# vlan 1
device(config-vlan-1)# spanning-tree priority 1
device(config-vlan-1)# tag ethernet 1/1/1 ethernet 5/1/1 to 5/1/3
device(config-vlan-1)# vlan 201
device(config-vlan-201)# spanning-tree priority 2
device(config-vlan-201)# tag ethernet 1/1/2 ethernet 5/1/1 to 5/1/3
device(config-vlan-201)# vlan 401
device(config-vlan-401)# spanning-tree priority 3
device(config-vlan-401)# tag ethernet 1/1/3 ethernet 5/1/1 to 5/1/3
...
device(config-vlan-3601)# vlan 3801
device(config-vlan-3801)# spanning-tree priority 20
device(config-vlan-3801)# tag ethernet 1/1/20 ethernet 5/1/1 to 5/1/3
device(config-vlan-3801)# exit

device(config)# vlan-group 1 vlan 2 to 200
device(config-vlan-group-1)# tag ethernet 1/1/1 ethernet 5/1/1 to 5/1/3
device(config-vlan-group-1)# vlan-group 2 vlan 202 to 400
device(config-vlan-group-2)# tag ethernet 1/1/2 ethernet 5/1/1 to 5/1/3
device(config-vlan-group-2)# vlan-group 3 vlan 402 to 600
device(config-vlan-group-2)# tag ethernet 1/1/3 ethernet 5/1/1 to 5/1/3
...
device(config-vlan-group-19)# vlan-group 20 vlan 3082 to 3282
device(config-vlan-group-20)# tag ethernet 1/1/20 ethernet 5/1/1 to 5/1/3
device(config-vlan-group-20)# exit

device(config)# stp-group 1
device(config-stp-group-1)# master-vlan 1
device(config-stp-group-1)# member-group 1
device(config-stp-group-1)# member-vlan 4001 4004 to 4010
device(config-stp-group-1)# stp-group 2
device(config-stp-group-2)# master-vlan 201
device(config-stp-group-2)# member-group 2
device(config-stp-group-2)# member-vlan 4002 4003 4011 to 4015
device(config-stp-group-2)# stp-group 3
device(config-stp-group-3)# master-vlan 401
device(config-stp-group-3)# member-group 3
...
device(config-stp-group-19)# stp-group 20
device(config-stp-group-20)# master-vlan 3081
device(config-stp-group-20)# member-group 20
```

STP parameters

STP bridge parameters

Bridge priority

Use this parameter to specify the priority of a switch and to determine the root bridge.

Each switch has a unique bridge identifier called the Bridge ID,.The Bridge ID is an 8-byte value that is composed of two fields: a 2 byte bridge priority field and the 6 byte MAC address field.

The value for the bridge priority ranges from 0 to 61440 in increments of 4096. The default value for the bridge priority is 32768. You use the **bridge-priority** command to set the appropriate values to designate a switch as the root bridge. A default Bridge ID may appear as *32768.768e.f805.5800.*.

If the bridge priorities are equal, the bridge with the lowest MAC address is elected the root bridge.

The root bridge should be centrally located and not in a "disruptive" location. Backbone switches typically serve as the root bridge because they usually do not connect to end stations. All other decisions in the network, such as which port to block and which port to put in forwarding mode, are made from the perspective of the root bridge.

Bridge Protocol data units (BPDUs) carry information between switches. All the switches in the layer 2 network, participating in xSTP, gather information on other switches in the network through an exchange of BPDUs. As the result of exchange of the BPDUs, the switch with the lowest bridge ID is elected as the root bridge

When setting the bridge forward delay, bridge maximum aging time, and the hello time parameters keep in mind that the following relationship should be kept:

```
(2 × (forward-delay — 1)) ≥ max-age ≥ (2 ×(hello-time + 1))
```

Bridge forward delay

You can set this parameter to specify how long an interface remains in the listening and learning states before forwarding spanning tree instances.

NOTE
This parameter can be set in STP, RSTP, MSTP, PVST+, and R-PVST+ mode.

Configure this parameter to specify how long an interface remains in the listening and learning states before the interface begins forwarding all spanning tree instances. The valid range is from 4 through 30 seconds. The default is 15 seconds.

You may also specify the forward delay for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge maximum aging time

You can use this setting to configure the maximum length of time that passes before an interface saves its BPDU configuration information.

NOTE
This parameter can be set in STP, RSTP, MSTP, PVST+, and R-PVST+ mode.

When configuring the maximum aging time, you must set the max-age to be greater than the hello time. The range is 6 through 40 seconds. The default is 20 seconds.

You may specify the maximum aging for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge hello time

You can use this setting to configure how often the switch interface broadcasts hello BPDUs to other devices.

> NOTE
> This parameter can be set in STP, RSTP, MSTP, PVST+, and R-PVST+ mode.

Use the **hello-time** command to configure the bridge hello time. The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The range is from 1 through 10 seconds. The default is 2 seconds.

You may also specify the hello time for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Error disable timeout parameter

Configure this parameter to enable a timer that brings a port out of the disabled state.

This parameter can be set in STP, RSTP, MSTP, PVST/PVST+, and R-PVST/R-PVST+ mode.

When the STP BPDU guard disables a port, the port remains in the disabled state unless the port is enabled manually. The parameter specifies the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds.

By default, the timeout feature is disabled.

Port-channel path cost parameter

You configure this parameter to specify the port-channel path cost.

This parameter can be set in STP, RSTP, MSTP, PVST/PVST+, and R-PVST/R-PVST+ mode.

There are two path cost options:

custom	Specifies that the path cost changes according to the port-channel's bandwidth.
standard	Specifies that the path cost does not change according to the port-channel's bandwidth.

The default port cost is standard.

Configuring STP

Enabling STP and STP parameters globally

Follow these steps to enable or disable STP.

You can enable STP or STP with one or more parameters enabled.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable STP globally.

   ```
   device(config)# protocol spanning-tree stp
   ```

3. Describe or name the STP.

   ```
   device(config-stp)# description stp1
   ```

   A description is not required.
4. Specify the bridge priority.

   ```
   device(config-stp)# bridge-priority 4096
   ```

   The bridge with the lowest priority is designated the root bridge. Bridge priority is set in increments of 4096, with 0 being the highest priority.
5. Specify the bridge forward delay.

   ```
   device(config-stp)# forward-delay 20
   ```

6. Configure the maximum aging time.

   ```
   device(config-stp)# max-age 25
   ```

7. Configure the maximum hello time.

   ```
   device(config-stp)# hello-time 8
   ```

8. Enable the error disable timeout timer.

   ```
   device(config-stp)# error-disable-timeout enable
   ```

9. Set the error disable timeout timer.

   ```
   device(config-stp)# error-disable-timeout interval 60
   ```

10. Configure the port-channel path cost.

    ```
    device(config-stp)# port-channel path-cost custom
    ```

11. Return to privileged exec mode.

    ```
    device(config-stp)# end
    ```

12. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: Spanning Tree Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------
 Eth 2/32      DES   FWD  2000       128   P2P               No
 Eth 2/66      DES   FWD  2000       128   P2P               No
 Po 7          DES   FWD  2000       128   P2P               No
 Po 8          DES   FWD  2000       128   P2P               No
 Po 21         DES   LIS  500        128   P2P               No
 Po 141        BKUP  BLK  1000       128   P2P               No
 Po 151        DES   FWD  10000      128   P2P               No
 Po 154        DES   FWD  285        128   P2P               No
 Po 172        BKUP  BLK  1000       128   P2P               No
 Po 173        BKUP  BLK  500        128   P2P               No
```

Observe that the settings comply with the formula set out in the STP parameters on page 21 section, as:

(2 × (forward_delay - 1)) ≥ max_age ≥ (2 × (hello_time + 1))

Or in this case .38 ≥ 25 ≥ 18.

STP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# description stpForInterface
device(config-stp)# bridge-priority 4096
device(config-stp)# forward-delay 20
device(config-stp)# max-age 25
device(config-stp)# hello-time 8
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config-stp)# port-channel path-cost custom
device(config-stp)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Enabling STP and STP features on an interface

Follow these steps to enable STP and STP features on an interface.

Globally enable STP and STP parameters.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable port fast on switch ports.
   a) Enter interface configuration mode for the port where you intend to enable port fast.

      ```
      device(config)# interface ethernet 1/1
      ```

   b) Enable port fast.

      ```
      device(conf-if-eth-1/1)# spanning-tree portfast
      ```

   c) Return to global configuration mode.

      ```
      device(conf-if-eth-1/1)# exit
      ```

   Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat these commands for every port connected to workstations or PCs. Do not enable port fast on ports that connect to other devices.
3. Specify port priorities to influence the selection of root or designated ports.
   a) Enter interface configuration mode for the target port.

      ```
      device(config)# interface ethernet 1/11
      ```

   b) Set the port priority.

      ```
      device(conf-if-eth-1/11)# spanning-tree priority 16
      ```

      The range is from 0 through 240 in increments of 16. The default value is 128.
   c) Return to privileged exec mode.

      ```
      device(conf-if-eth-1/11)# end
      ```

   Brocade recommends leaving other STP parameters at their default values.
4. To interoperate with non-SLX devices (such as Multi-Service IronWare devices and FastIron devices) in PVST+/R-PVST+ mode, you may need to configure the interface that is connected to that switch.
   a) Enter interface configuration mode for the port that connects with a non-SLX device.

      ```
      device(config)# interface ethernet 1/12
      ```

   b) Specify the MAC address for the device.

      ```
      device(conf-if-eth-1/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
      ```

   c) Return to privileged exec mode.

      ```
      device(conf-if-eth-1/12)# end
      ```

5. Verify the configuration.

```
device# show spanning-tree

 Spanning-tree Mode: Spanning Tree Protocol

 Root Id: 4096.01e0.5200.0180 (self)
 Bridge Id: 4096.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec


device# show spanning-tree brief

 Spanning-tree Mode: Spanning Tree Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------
 Eth 2/32      DES   FWD  2000       128   P2P               No
 Eth 2/66      DES   FWD  2000       128   P2P               No
 Po 7          DES   FWD  2000       128   P2P               No
 Po 8          DES   FWD  2000       128   P2P               No
 Po 21         DES   LIS  500        128   P2P               No
 Po 141        BKUP  BLK  1000       128   P2P               No
 Po 151        DES   FWD  10000      128   P2P               No
 Po 154        DES   FWD  285        128   P2P               No
 Po 172        BKUP  BLK  1000       128   P2P               No
 Po 173        BKUP  BLK  500        128   P2P               No
```

Observe that the settings comply with the formula set out in the STP parameters on page 21 section, as: `(2 × (forward_delay - 1)) ≥ max_age ≥ (2 × (hello_time + 1))` or in this case `38 ≥ 25 ≥ 18`.

6. Save the settings by copying the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

STP on an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1# spanning-tree portfast
device(config-if-eth-1/1)# exit
device(config)# interface ethernet 1/11
device(config-if-eth-1/11)# spanning-tree bpdu-mac 0100.0ccc.cccd
device(config-if-eth-1/11)# exit
device(config)# interface ethernet 1/12
device(config-if-eth-1/12)# spanning-tree priority 16
device(config-if-eth-1/12)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Enabling STP and STP parameters on a VLAN

Follow these steps to enable STP on a single VLAN.

When you configure a VLAN, the VLAN inherits the global STP settings. However, once you begin to define a VLAN, you can no longer configure standard STP parameters globally through the CLI. From that point on, you must configure STP within individual VLANs.

STP is globally enabled.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable spanning tree.

   ```
   device(config)# protocol spanning-tree stp
   ```

3. Specify the bridge forward delay.

   ```
   device(config-stp)# forward-delay 20 vlan 10
   ```

4. Specify the bridge maximum aging time.

   ```
   device(config-stp)# max-age 25 vlan 10
   ```

5. Specify the hello time.

   ```
   device(config-stp)# hello-time 5 vlan 10
   ```

6. Return to privileged exec mode.

   ```
   device(config-stp)# end
   ```

7. Verify the configuration.

   ```
   device# show span vlan 10

   STP instance owned by VLAN 10

   Global STP (IEEE 802.1D) Parameters:

   VLAN Root             Root Root  Prio Max He- Ho- Fwd Last    Chg Bridge
    ID   ID              Cost Port  rity Age llo ld  dly Chang   cnt Address
                                    Hex  sec sec sec sec sec
    10  0000cc4e248bb050 0    Root  0000 20  2   1   15  9329890 0   cc4e248bb050

   Port STP Parameters:

   Port  Prio  Path    State       Fwd    Design  Designated        Designated
   Num   rity    Cost              Trans  Cost    Root              Bridge
         Hex
   1/5   80      0     DISABLED    0      0       0000000000000000  0000000000000000
   1/6   80      19    FORWARDING  1      0       800000e0804d4a00  800000e0804d4a00
   1/7   80      0     DISABLED    0      0       0000000000000000  0000000000000000
   1/8   80      0     DISABLED    0      0       0000000000000000  0000000000000000
   1/13  80      0     DISABLED    0      0       0000000000000000  0000000000000000
   1/14  80      0     DISABLED    0      0       0000000000000000  0000000000000000
   1/15  80      19    FORWARDING  1      0       800000e0804d4a00  800000e0804d4a00
   1/16  80      19    BLOCKING    0      0       800000e0804d4a00  800000e0804d4a00
   1/17  80      0     BLOCKING    0      0       800000e0804d4a00  800000e0804d4a00
   ```

8. Save the configuration.

   ```
   device# copy running-config startup-config
   ```

STP on a single VLAN configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# forward-delay 20 vlan 10
device(conf-stp)# max-age 25 vlan 10
device(config-stp)# hello-time 5 vlan 10
device(config-stp)# end
device# show span vlan 10
device# copy running-config startup-config
```

Re-enabling an error-disabled port automatically

Enable a port to automatically recover from the error-disabled state after the expiration of an error recovery timer.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter STP configuration mode.

   ```
   device(config)# protocol spanning-tree stp
   ```

3. Enable the error-disable-timeout timer.

   ```
   device(conf-stp)# error-disable-timeout enable
   ```

4. Set an interval after which port shall be enabled.

   ```
   device(config-stp)# error-disable-timeout interval 60
   ```

   The interval range is from 0 to 1000000 seconds, the default is 300 seconds.

5. Return to privileged exec mode.

   ```
   device(config-stp)# end
   ```

6. Verify the configuration.

   ```
   device# show spanning-tree
    Spanning-tree Mode: Spanning Tree Protocol

    Root Id: 8000.768e.f805.5800 (self)
    Bridge Id: 8000.768e.f805.5800

    Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
    Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
    Number of topology change(s): 0

    Bpdu-guard errdisable timeout: enabled
    Bpdu-guard errdisable timeout interval: 60 sec
   ```

Automatically re-enable an error-disabled port configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(conf-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config-stp)# end
device# show spanning-tree
```

802.1w Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol overview

Rapid Spanning Tree Protocol (RSTP) provides rapid reconvergence of edge ports, new root ports, and ports connected through point-to-point links.

The STP (802.1d) standard was designed at a time when recovering connectivity after an outage within a minute or so was considered adequate performance. With the advent of Layer 3 switching in LAN environments, bridging competes with routed solutions where protocols such as OSPF are able to provide an alternate path in less time. The RSTP can be seen as evolution of STP standard. It provides rapid convergence of connectivity following the failure of bridge, a bridge port or a LAN. It provides rapid convergence of edge ports, new root ports and port connected through point-to-point links. The port, which qualifies for fast convergence, is derived from the duplex mode of a port. A port operating in full-duplex will be assumed to be point-to-point, while a half-duplex port will be considered as a shared port by default. This automatic setting can be overridden by explicit configuration.

NOTE
RSTP is designed to be compatible and interoperate with STP. However, he benefit of RSTP's fast convergence are lost when interacting with legacy STP (802.1d) bridges since it downgrades itself to STP when it detects it is connected to a legacy bridge.

RSTP parameters on a Brocade device

See the section STP parameters on page 21 for descriptions of parameters that apply to STP, RSTP, MSTP , PVST/PVST+, and R-PVST/R-PVST+.

There is one parameter that can be configured in RSTP that is not available in STP; the transmit hold count. See the section R-PVST example configuration on page 47 for the procedure to configure this parameter.

The edge port (making a port transition directly from initialization to the forwarding state) and auto edge (automatically detects an edge port) features can be enabled in RSTP as well.

Edge port

Configuring the edge port feature makes a port transition directly from initialization to the forwarding state, skipping the listening and learning states.

NOTE
This feature is only for RSTP , R-PVST+, and MSTP.

When edge port is enabled, the port still participates in STP.

When an edge port receives a BPDU, it's a normal spanning tree port and is no longer an edge port. When this feature is enabled on an interface the device is expected to put the port into FORWARDING state immediately.

Configuring RSTP

Enabling RSTP and configuring RSTP parameters

Follow these steps to enable and configure RSTP.

See the section STP parameters on page 21 for parameters applicable to all STP variants.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable RSTP.

   ```
   device(config)# protocol spanning-tree rstp
   ```

3. Designate the root device.

   ```
   device(conf-rstp)# bridge-priority 28582
   ```

   The range is 0 through 61440 and the priority values can be set only in increments of 4096.

4. Configure the bridge forward delay value.

   ```
   device(conf-rstp)# forward-delay 15
   ```

5. Configure the bridge maximum aging time value.

   ```
   device(conf-rstp)# max-age 20
   ```

6. Enable the error-disable-timeout timer.

   a) Enable the timer.

   ```
   device(conf-rstp)# error-disable-timeout enable
   ```

   b) Configure the error-disable-timeout interval value.

   ```
   device(conf-rstp)# error-disable-timeout interval 60
   ```

7. Configure the port-channel path cost.

   ```
   device(conf-rstp)# port-channel path-cost custom
   ```

8. Configure the bridge hello-time value.

   ```
   device(conf-rstp)# hello-time 2
   ```

9. Return to privileged exec mode.

   ```
   device(conf-rstp)# end
   ```

10. Verify the configuration

    ```
    device# show spanning-tree

     Spanning-tree Mode: Rapid Spanning Tree Protocol

     Root Id: 8000.01e0.5200.0180 (self)
     Bridge Id: 8000.01e0.5200.0180

     Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
     Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
     Number of topology change(s): 0

     Bpdu-guard errdisable timeout: enabled
     Bpdu-guard errdisable timeout interval: 60 sec
    ```

11. Save the running configuration to the startup configuration.

    ```
    device# copy running-config startup-config
    ```

Enabling RSTP and configuring RSTP parameters example

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(conf-rstp)# bridge-priority 28582
device(conf-rstp)# forward-delay 20
device(conf-rstp)# max-age 25
device(conf-rstp)# error-disable-timeout enable
device(conf-rstp)# error-disable-timeout interval 60
device(conf-rstp)# port-channel path-cost custom
device(conf-rstp)# hello-time 5
device(conf-rstp# end
device# show spanning-tree
device# copy running-config startup-config
```

Configuring RSTP on an interface

Follow these steps to configure RSTP on an Ethernet interface.

See the section STP parameters on page 21 for parameters applicable to all STP variants.

1. Enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Enter interface configuration mode.

    ```
    device(config)# interface ethernet 1/10
    ```

3. Enable edge port on a switch port.

    ```
    device(conf-if-eth-1/10)# spanning-tree edgeport
    ```

    Configuring edgeport makes a port transition directly from initialization to the forwarding state, skipping the listening and learning states.

4. Enable auto edge on a switch port.

    ```
    device(conf-if-eth-1/10)# spanning-tree autoedge
    ```

    Configuring autoedge enables automatic edge detection.

5. Return to privileged exec mode.

   ```
   device(conf-if-eth-1/10)#  end
   ```

6. Verify the configuration

   ```
   device# show spanning-tree

    Spanning-tree Mode: Rapid Spanning Tree Protocol

    Root Id: 8000.01e0.5200.0180 (self)
    Bridge Id: 8000.01e0.5200.0180

    Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
    Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
    Number of topology change(s): 0

    Bpdu-guard errdisable timeout: enabled
    Bpdu-guard errdisable timeout interval: 60 sec


    Port Eth 1/10 enabled
       Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
       Designated Path Cost: 0
       Configured Path Cost: 20000000
       Designated Port Id: 0; Port Priority: 128
       Designated Bridge: 0000.0000.0000.0000
       Number of forward-transitions: 0
       Version: Spanning Tree Protocol - Received None - Sent STP
       Edgeport: on; AutoEdge: yes; AdminEdge: no; EdgeDelay: 3 sec
       Configured Root guard: off; Operational Root guard: off
       Bpdu-guard: off
       Bpdu-filter: off
       Link-type: point-to-point
       Received BPDUs: 0; Sent BPDUs: 0
   ```

7. Save the running configuration to the startup configuration.

   ```
   device# copy running-config startup-config
   ```

RSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(conf-rstp)# bridge-priority 28582
device(conf-rstp)# forward-delay 20
device(conf-rstp)# max-age 25
device(conf-rstp)# error-disable-timeout enable
device(conf-rstp)# error-disable-timeout interval 60
device(conf-rstp)# port-channel path-cost custom
device(conf-rstp)# hello-time 5
device(config)# interface ethernet 1/10
device(conf-if-eth-1/10)# spanning-tree edgeport
device(conf-if-eth-1/10)# spanning-tree autoedge
device(conf-if-eth-1/10)# end
device# show spanning-tree
device# copy running-config startup-config
```

Per-VLAN Spanning Tree+ and Rapid Per-VLAN Spanning Tree+

PVST+ and R-PVST+ overview

The Per-VLAN Spanning Tree Plus (PVST+) protocol runs a spanning tree instance for each VLAN in the network. The Per-VLAN Rapid Spanning Tree Plus (R-PVST+) protocol runs a rapid spanning tree instance for each VLAN in the network.

Both STP and RSTP build a single logical topology. A single logical topology does not efficiently utilize the availability of redundant paths for multiple VLANs. A typical network has multiple VLANs if a port is set to "blocked/discarding" for one VLAN (under STP/RSTP), it is the same for all other VLANs. PVST+ builds on the STP on each VLAN, and R-PVST+ builds on the RSTP on each VLAN.

Before you configure PVST+ or R-PVST+ see the section STP parameters on page 21.

PVST+ and R-PVST+ guidelines and restrictions

Consider the following when configuring PVST+ and R-PVST+:

- Brocade supports PVST+ and R-PVST+ only. The PVST and R-PVST protocols are proprietary to Cisco and are not supported.
- A port native VLAN is the native VLAN ID associated with a trunk port on a Brocade switch. This VLAN ID is associated with all untagged packets on the port. The default native VLAN ID for a trunk port is 1.
- IEEE compliant switches run just one instance of STP protocol shared by all VLANs, creating a Mono Spanning Tree (MST). A group of such switches running a single spanning tree forms an MST region.
- You can configure up to 128 PVST+ or R-PVST+ instances. If you have more than 128 VLANs configured on the switch and enable PVST then the first 128 VLANs are PVST/+ or R-PVST+ enabled. I
- In PVST/+ or R-PVST+ mode, when you are connected to a Cisco or MLX switch, he Cisco proprietary MAC address to which the BPDUs are sent/processed must be explicitly configured on a per-port basis.
- In PVST/+ or R-PVST+ mode, when you connect to a Cisco switch using a trunk port, the Brocade switch must have a native VLAN configured on the trunk port (same configuration as on the other side).
- Cisco PVST+ BPDUs on a port are not auto-detected, if you change the destination MAC of the BPDUs.
- A Common Spanning Tree (CST) is the single spanning tree instance used by Brocade switches to interoperate with 802.1q bridges. This spanning tree instance stretches across the entire network domain (including PVST, PVST+ and 802.1q regions). It is associated with VLAN 1 on the Brocade switch.
- In order to interact with STP and IEEE 802.1q trunk, PVST evolved to PVST+ to interoperate with STP topology by STP BPDU on the native or default VLAN.
- A group of switches running PVST+ is called a PVST+ region.

Bridge protocol data units in different VLANs

PVST+ uses the spanning tree instance for VLAN 1 to join the CST in the network to build the CST, PVST+ processes and sends standard IEEE Bridge protocol data units (BPDUs) on all the ports in VLAN 1 (access/trunk).

Across IEEE 802.1q trunks, Brocade switches run PVST+. The goal is to interoperate with standard IEEE STP (or RSTP or MSTP), while transparently tunneling PVST+ instance BPDUs across the MST region to potentially connect to other Brocade switches across the MST region.

On trunk ports that allow VLAN 1, PVST+ also sends PVST+ BPDUs to a Cisco-proprietary multicast MAC address (0100.0ccc.cccd) or Brocade-proprietary multicast MAC address (0304.0800.0700) depending on the configuration. By default, the PVST+ BPDUs are sent to Brocade-proprietary multicast MAC address on brocade switches. These BPDUs are tunneled across an MST region. The PVST+ BPDUs for VLAN 1 are only used for the purpose of consistency checks and that it is only the IEEE BPDUs that are used for building the VLAN 1 spanning tree. So in order to connect to the CST, it is necessary to allow VLAN 1 on all trunk ports.

For all other VLANs, PVST+ BPDUs are sent on a per-VLAN basis on the trunk ports. These BPDUs are tunneled across an MST region. Consequently, for all other VLANs, MST region appears as a logical hub. The spanning tree instances for each VLAN in one PVST+ region map directly to the corresponding instances in another PVST+ region and the spanning trees are calculated using the per-VLAN PVST+ BPDUs.

Similarly, when a PVST+ region connects to a MSTP region, from the point of view of MSTP region, the boundary bridge thinks it is connected to a standard IEEE compliant bridge sending STP BPDUs. So it joins the CIST of the MSTP region to the CST of the PVST+ region (corresponding to VLAN 1). The PVST+ BPDUs are tunneled transparently through the MSTP region. So from the Brocade bridge point of view, the MSTP region looks like a virtual hub for all VLANs except VLAN 1.

The PVST+ BPDUs are sent untagged for the native VLAN and tagged for all other VLANs on the trunk port.

On access ports, Brocade switches run classic version of IEEE STP/RSTP protocol, where the BPDUs are sent to the standard IEEE multicast address "0180.C200.0000". So if we connect a standard IEEE switch to an access port on the Brocade switch, the spanning tree instance (corresponding to the access VLAN on that port) of the Brocade switch is joined with the IEEE STP instance on the adjacent switch.

For introductory information about STP BPDUs, see the section BPDUs on page 14.

BPDU configuration notes

In order to build a spanning tree for the bridge topology, the bridges must exchange control frames. These frames are called Bridge Protocol data units (BPDU).

BPDUs are sent to a Cisco-proprietary multicast MAC address 0100.0ccc.cccd or Brocade-proprietary multicast MAC address 0304.0800.0700.By default, the PVST+ BPDUs are sent to Brocade-proprietary multicast MAC address on Brocade switches. These are called SSTP (Single Spanning Tree Protocol) BPDUs. The format of the SSTP BPDU is nearly identical to the 802.1d BPDU after the SNAP header, except that a type-length-value (TLV) field is added at the end of the BPDU. The TLV has 2 bytes for type (0x0), 2 bytes for length, and 2 bytes for the VLAN ID. See Brocade BPDU PVST+ headers/fields on page 37 and BPDU R-PVST+ header and field comparisons on page 37 for an outline of the BPDU header content.

Topology Change Notification (TCN) BPDUs are used to inform other switches of port changes. TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

In PVST+, three types of TCN BPDUs are sent out depending on the type of the link. See Brocade PVST+ TCN BPDU headers/fields on page 39 and Cisco PVST TCN BPDU headers/fields on page 39.

- Standard IEEE TCN BPDU.
- Untagged TCN BPDU sent to the Cisco/Brocade proprietary MAC address.

- Tagged TCN BPDU sent to the Cisco/Brocade proprietary MAC address.

BPDU R-PVST+ header and field comparisons

These tables outline the differences between Brocade R-PVST+ BPDU and Cisco R-PVST+ BPDU header fields.

Brocade R-PVST+ BPDU headers/fields

Header/field	Standard IEEE STP/RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
Source Address (MAC SA)	6B	6B	6B
Destination Address (MAC DA)	0180C2.000000 (6B)	030408.000700 (6B)	030408.000700 (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
Source Service Access Point (SSAP)	42	AA 03	AA 03
Destination Service Access Point (DSAP)	42	AA	AA
Brocade Organizationally Unique Identifier (OUI)	-	02 04 08	02 04 08
PVST PID	-	01 0B	01 0B
Logical Link Control (LLC)	3B	+	+
SubNetwork Access Protocol (SNAP)	-	Yes (2B)	Yes (2B)
IEEE BPDU INFO	35B	35B	35B
Type, Length, Value (TLV) Pad Type Length VLAN ID	-	6B 00 (1B) 00 00 00 02 2B	6B 00 (1B) 00 00 00 02 2B

Cisco R-PVST+ BPDU headers/fields

Header/field	Standard IEEE STP/RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	01000C.CCCCCD (6B)	010002.CCCCCD (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	00 00 0C	00 00 0C
PVST PID	-	01 0B	01 0B
LLC	3B	+	+

Header/field	Standard IEEE STP/RSTP BPDU	R-PVST+ untagged BPDU	R-PVST+ tagged BPDU
	(64B padded)	(64B padded)	(72B padded)
SNAP	-	Yes	Yes
IEEE BPDU INFO	35B	35B	35B
TLV	-	6B	6B
Pad		00 (1B)	00 (1B)
Type		00 00	00 00
Length		00 02	00 02
VLAN ID		2B	2B

Sent BPDUs

On an 802.1q trunk, the PVST+ enabled switch sends the following BPDUs:

1. For all tagged VLANs on the port on which PVST+ is enabled, 802.1q tagged SSTP BPDUs are sent to the Cisco or Brocade MAC address. The 802.1q tag contains the VLAN ID. (VLAN 1 could be tagged on the port. In that case a tagged BPDU for VLAN 1 is sent). The IEEE compliant switches do not consider these BPDUs as a control packet. So they forward the frame as they would forward to any unknown multicast address on the specific VLAN.
2. If PVST+ is enabled on the untagged (native) VLAN of the port, an untagged SSTP BPDU is sent to the Brocade or Cisco MAC address on the native VLAN of the trunk. It is possible that the native VLAN on the Brocade or Cisco port is not VLAN 1. This BPDU is also forwarded on the native VLAN of the IEEE 802.1q switch just like any other frame sent to an unknown multicast address.
3. In addition the above SSTP BPDUs, a standard IEEE BPDU (802.1d) is also sent, corresponding to the information of VLAN 1 on the Brocade or Cisco switch. This BPDU is not sent if VLAN 1 is explicitly disabled on the trunk port.

The following table lists the types of BPDUs sent in case of different port types. The numbers in the third column are the VLAN instance for which these BPDUs are sent/processed.

TABLE 1 Types of BPDUs sent for different port types

Port Configuration	Brocade or Cisco - PVST(+)	VLAN instance
Access - VLAN 1	Standard IEEE BPDU (64B)	1
Access - VLAN 100	Standard IEEE BPDU (64B)	100
Trunk - Native VLAN 1	Standard IEEE BPDU (64B)	1
Allowed VLANs - 1, 100, 200	Brocade or Cisco untagged BPDU (68B)	1
	Brocade or Cisco tagged BPDU (72B)	100
	Brocade or Cisco tagged BPDU (72B)	200
Trunk - Native VLAN 100	Standard IEEE BPDU (64B)	1
Allowed VLANs - 1, 100, 200	Brocade or Cisco untagged BPDU (68B)	100
	Brocade or Cisco tagged BPDU (72B	1
	Brocade or Cisco tagged BPDU (72B)	200
Trunk - Native VLAN 100	Brocade or Cisco untagged BPDU (68B)	100
Allowed VLANs - 100		
Trunk - Native VLAN 100	Brocade or Cisco untagged BPDU (68B)	100
Allowed VLANs - 100, 200	Brocade or Cisco tagged BPDU (72B)	200

TCN headers and fields

Since PVST+ is based on STP, and Rapid-PVST+ is based on RSTP, TCN BPDUs are sent only in PVST+ and not in Rapid-PVST+ mode.

For introductory information about STP BPDUs, see the section TCN BPDUs on page 15.

Brocade PVST+ TCN BPDU headers/fields

Header/field	Standard IEEE STP TCN BPDU (64B with padding)	PVST+ untagged TCN BPDU (64B with padding)	PVST+ tagged TCN BPDU (68B with padding)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	030408.000700 (6B)	030408.000700 ((6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	02 04 08	02 04 08
PVST PID	-	01 0B	01 0B
LLC	3B	8B	8B
SNAP	4B	Entire BPDU with type = TCN 35B	Entire BPDU with type = TCN 35B

Cisco PVST TCN BPDU headers/fields

Header/field	Standard IEEE STP TCN BPDU (64B padded)	PVST untagged TCN BPDU (64B padded)	PVST tagged TCN BPDU (68B padded)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	01000C.CCCCCD (6B)	01000C.CCCCCD (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	00 00 0C	00 00 0C
PVST PID	-	01 0B	01 0B
LLC	3B	8B	8B
SNAP	-	Yes	Yes
IEEE TCN BPDU INFO	4B	Entire BPDU with type = TCN 35B	Entire BPDU with type = TCN 35B

PortFast

PortFast allows an interface to transition quickly to the forwarding state.

Consider the following when configuring PortFast:

- Do not enable port fast on ports that connect to other devices.
- Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat this configuration for every port connected to workstations or PCs.
- Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.
- If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown**/**no shutdown** command.
- PortFast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

Configuring PVST+ and R-PVST+

Enabling PVST+

Use this procedure to enable and set basic parameters for the Per-VLAN Spanning Tree Protocol Plus.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable PVST+.

   ```
   device(config)# protocol spanning-tree pvst
   ```

3. Configure the bridge priority for the common instance.

   ```
   device(config-pvst)# bridge-priority 4096
   ```

   Valid values range from 0 through 61440 in increments of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the transmit hold count parameter.

   ```
   device(config-pvst)# transmit-holdcount 5
   ```

5. Configure the forward delay parameter.

   ```
   device(config-pvst)# forward-delay  11
   ```

6. Configure the hello time parameter.

   ```
   device(config-pvst)# hello-time 2
   ```

7. Configure the maximum age parameter.

   ```
   device(config-pvst)# max-age 7
   ```

8. Return to privileged exec mode

   ```
   device(config-pvst)# end
   ```

9. Verify the configuration.

```
device#  show spanning-tree brief
VLAN 1

 Spanning-tree Mode: Per-VLAN Spanning Tree Protocol

      Root ID      Priority 4097
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 4097
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------

VLAN 100

 Spanning-tree Mode: Per-VLAN Spanning Tree Protocol

      Root ID      Priority 4196
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 4196
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------
```

10. Save the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

PVST+ configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(config-pvst)# bridge-priority 4096
device(config-pvst)# transmit-holdcount 5
device(config-pvst)# forward-delay 11
device(config-pvst)# hello-time 2
device(config-pvst)# max-age 7
device(config-pvst)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

For more information about configuring PVST+ parameters, see STP parameters on page 21. PVST+, R-PVST+, and other types of spanning trees share many tasks with STP.

PVST+ example configuration

Follow the steps to configure PVST+ on a system.

PVST+ and R-PVST+ configuration are identical except for the keyword identifying the spanning tree type in Step 2.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Go to PVST+ configuration mode.

   ```
   device(config)# protocol spanning-tree pvst
   ```

3. Add VLANs.
   a) Add VLAN 100 with a priority of 0.

      ```
      device(config-pvst)# vlan 100 priority 0
      ```

      The bridge priority in configured in increments of 4096.
   b) Add VLAN 201 with a priority of 12288.

      ```
      device(config-pvst)# vlan 201 priority 12288
      ```

   c) Add VLAN 301 with a priority of 20480.

      ```
      device(config-pvst)# vlan 301 priority 20480
      ```

4. Put the interface in Layer 2 mode and set the switching characteristics for interface 1/3.
   a) Enter interface configuration mode.

      ```
      device(config-pvst)# interface ethernet 1/3
      ```

   b) Set the switching characteristics of the interface.

      ```
      device(conf-if-eth-1/3)# switchport
      ```

   c) Set the Layer 2 mode of the interface to trunk.

      ```
      device(conf-if-eth-1/3)#  switchport mode trunk
      ```

   d) Configure VLAN 100 as a member VLAN.

      ```
      device(conf-if-eth-1/3)# switchport trunk allowed vlan add 100
      ```

   e) Configure VLAN 201 as a member VLAN.

      ```
      device(conf-if-eth-1/3)# switchport trunk allowed vlan add 201
      ```

   f) Configure VLAN 301 as a member VLAN.

      ```
      device(conf-if-eth-1/3)# switchport trunk allowed vlan add 301
      ```

5. Enable the spanning tree.

   ```
   device(conf-if-eth-1/3)#  no spanning-tree shutdown
   ```

6. Put the interface in Layer 2 mode and set the switching characteristics for interface 1/0/4.
   a) Enter interface configuration mode.

      ```
      device(config-pvst)# interface ethernet 1/4
      ```

   b) Set the switching characteristics of the interface.

      ```
      device(conf-if-eth-1/4)# switchport
      ```

   c) Set the Layer 2 mode of the interface to trunk.

      ```
      device(conf-if-eth-1/4)# switchport mode trunk
      ```

   d) Configure VLAN 100 as a member VLAN.

      ```
      device(conf-if-eth-1/4)# switchport trunk allowed vlan add 100
      ```

   e) Configure VLAN 201 as a member VLAN.

      ```
      device(conf-if-eth-1/4)# switchport trunk allowed vlan add 201
      ```

   f) Configure VLAN 301 as a member VLAN.

      ```
      device(conf-if-eth-1/4)# switchport trunk allowed vlan add 301
      ```

7. Enable the spanning tree.

   ```
   device(conf-if-eth-1/4)# no spanning-tree shutdown
   ```

8. Return to privileged exec mode.

   ```
   device(conf-if-eth-1/4)# end
   ```

9. Verify the configuration.

```
device# show spanning-tree

VLAN 1

 Spanning-tree Mode: stp Protocol

 Root Id: 0001.01e0.5200.0180 (self)
 Bridge Id: 0001.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 1/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Bpdu-filter: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 1/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Bpdu-filter: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 100

 Spanning-tree Mode: stp Protocol

 Root Id: 0064.01e0.5200.0180 (self)
 Bridge Id: 0064.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 1/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Bpdu-filter: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 1/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Bpdu-filter: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 201

 Spanning-tree Mode: stp Protocol

 Root Id: 30c9.01e0.5200.0180 (self)
 Bridge Id: 30c9.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 1/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Bpdu-filter: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 1/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Bpdu-filter: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 301

 Spanning-tree Mode: stp Protocol

 Root Id: 512d.01e0.5200.0180 (self)
 Bridge Id: 512d.01e0.5200.0180
```

```
        Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
        Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
        Number of topology change(s): 0

        Bpdu-guard errdisable timeout: disabled
        Bpdu-guard errdisable timeout interval: 300 sec

        Port Et 1/3 enabled
           Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
           Designated Path Cost: 0
           Configured Path Cost: 20000000
           Designated Port Id: 0; Port Priority: 128
           Designated Bridge: 0000.0000.0000.0000
           Number of forward-transitions: 0
           Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
           Portfast: off
           Configured Root guard: off; Operational Root guard: off
           Bpdu-guard: off
           Bpdu-filter: off
           Link-type: point-to-point
           Received BPDUs: 0; Sent BPDUs: 0

        Port Et 1/4 enabled
           Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
           Designated Path Cost: 0
           Configured Path Cost: 20000000
           Designated Port Id: 0; Port Priority: 128
           Designated Bridge: 0000.0000.0000.0000
           Number of forward-transitions: 0
           Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
           Portfast: off
           Configured Root guard: off; Operational Root guard: off
           Bpdu-guard: off
           Bpdu-filter: off
           Link-type: point-to-point
           Received BPDUs: 0; Sent BPDUs: 0
```

10. Save the configuration.

```
device# copy running-config startup-config
```

PVST+ configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(config-pvst)# vlan 100 priority 0
device(config-pvst)# vlan 201 priority 12288
device(config-pvst)# vlan 301 priority 20480
device(config-pvst)# interface ethernet 1/3
device(conf-if-eth-1/3)# switchport
device(conf-if-eth-1/3)# switchport mode trunk
device(conf-if-eth-1/3)# switchport trunk allowed vlan add 100
device(conf-if-eth-1/3)# switchport trunk allowed vlan add 201
device(conf-if-eth-1/3)# switchport trunk allowed vlan add 301
device(conf-if-eth-1/3)# no spanning-tree shutdown
device(config-pvst)# interface ethernet 1/4
device(conf-if-eth-1/4)# switchport
device(conf-if-eth-1/4)# switchport mode trunk
device(conf-if-eth-1/4)# switchport trunk allowed vlan add 100
device(conf-if-eth-1/4)# switchport trunk allowed vlan add 201
device(conf-if-eth-1/4)# switchport trunk allowed vlan add 301
device(conf-if-eth-1/4)# no spanning-tree shutdown
device(conf-if-eth-1/4)# end
device# show spanning-tree
device# copy running-config startup-config
```

R-PVST example configuration

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Go to RPVST configuration mode.

   ```
   device(config)# protocol spanning-tree rpvst
   ```

3. Add VLANs.
   a) Add VLAN 100 with a priority of 0.

      ```
      device(config-rpvst)# vlan 100 priority 0
      ```

      The bridge priority in configured in increments of 4096.

   b) Add VLAN 201 with a priority of 12288.

      ```
      device(config-rpvst)# vlan 201 priority 12288
      ```

   c) Add VLAN 301 with a priority of 20480.

      ```
      device(config-rpvst)# vlan 301 priority 20480
      ```

4. Put the interface in Layer 2 mode and set the switching characteristics for interface 1/3.
   a) Enter interface configuration mode.

      ```
      device(config-rpvst)# interface ethernet 1/3
      ```

   b) Set the switching characteristics of the interface.

      ```
      device(conf-if-eth-1/3)# switchport
      ```

   c) Set the Layer 2 mode of the interface to trunk.

      ```
      device(conf-if-eth-1/3)#  switchport mode trunk
      ```

   d) Configure VLAN 100 as a member VLAN.

      ```
      device(conf-if-eth-1/3)# switchport trunk allowed vlan add 100
      ```

   e) Configure VLAN 201 as a member VLAN.

      ```
      device(conf-if-eth-1/3)# switchport trunk allowed vlan add 201
      ```

   f) Configure VLAN 301 as a member VLAN.

      ```
      device(conf-if-eth-1/3)# switchport trunk allowed vlan add 301
      ```

5. Enable the spanning tree.

   ```
   device(conf-if-eth-1/3)#  no spanning-tree shutdown
   ```

6. Put the interface in Layer 2 mode and set the switching characteristics for interface 1/0/4.
   a) Enter interface configuration mode.

      ```
      device(config-rpvst)# interface ethernet 1/4
      ```

   b) Set the switching characteristics of the interface.

      ```
      device(conf-if-eth-1/4)# switchport
      ```

   c) Set the Layer 2 mode of the interface to trunk.

      ```
      device(conf-if-eth-1/4)# switchport mode trunk
      ```

   d) Configure VLAN 100 as a member VLAN.

      ```
      device(conf-if-eth-1/4)# switchport trunk allowed vlan add 100
      ```

   e) Configure VLAN 201 as a member VLAN.

      ```
      device(conf-if-eth-1/4)# switchport trunk allowed vlan add 201
      ```

   f) Configure VLAN 301 as a member VLAN.

      ```
      device(conf-if-eth-1/4)# switchport trunk allowed vlan add 301
      ```

7. Enable the spanning tree.

   ```
   device(conf-if-eth-1/3)# no spanning-tree shutdown
   ```

8. Return to privileged exec mode.

   ```
   device(conf-if-eth-1/4)# end
   ```

9. Verify the configuration.

   ```
   device# show spann
   ```

R-PVST configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# vlan 100 priority 0
device(config-rpvst)# vlan 201 priority 12288
device(config-rpvst)# vlan 301 priority 20480
device(config-rpvst)# interface tengigabitethernet 1/3
device(conf-if-te-1/3)# switchport
device(conf-if-te-1/3)# switchport mode trunk
device(conf-if-te-1/3)# switchport trunk allowed vlan add 100
device(conf-if-te-1/3)# switchport trunk allowed vlan add 201
device(conf-if-te-1/3)# switchport trunk allowed vlan add 301
device(conf-if-te-1/3)# no spanning-tree shutdown
device(config-rpvst)# interface tengigabitethernet 1/4
device(conf-if-te-1/4)# switchport
device(conf-if-te-1/4)# switchport mode trunk
device(conf-if-te-1/4)# switchport trunk allowed vlan add 100
device(conf-if-te-1/4)# switchport trunk allowed vlan add 201
device(conf-if-te-1/4)# switchport trunk allowed vlan add 301
device(conf-if-te-1/4)# no spanning-tree shutdown
device(conf-if-te-1/4)# end
device# show spanning-tree
```

802.1s Multiple Spanning Tree Protocol

MSTP overview

IEEE 802.1s Multiple STP (MSTP) helps create multiple loop-free active topologies on a single physical topology.

MSTP evolved as a compromise between the two extremes of RSTP and R-PVST+, it was standardized as IEEE 802.1s and later incorporated into the IEEE 802.1Q-2003 standard. MSTP configures meshed topology into a loop-free, tree-like topology. When the link on a bridge port goes up, an MSTP calculation occurs on that port. The result of the calculation is the transition of the port into either a forwarding or blocking state. The result depends on the position of the port in the network and the MSTP parameters. All the data frames are forwarded over the spanning tree topology calculated by the protocol.

NOTE
MSTP is backward compatible with STP and RSTP.

Common Spanning Tree (CST)

The single Spanning Tree instance used by the Brocade device, and other vendor devices to interoperate with MSTP bridges. This spanning tree instance stretches across the entire network domain (including PVST, PVST+ and MSTP regions). It is associated with VLAN 1 on the Brocade device.

Internal Spanning Tree (IST)

An MSTP bridge must handle at least these two instances: one IST and one or more MSTIs (Multiple Spanning Tree Instances). Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance known as IST, which extends CST inside the MST region. IST always exists if the device runs MSTP. Besides IST, this implementation supports up to 31 MSTIs.

Common and Internal Spanning Tree (CIST)

The single spanning tree calculated by STP (including PVST+) and RSTP (including R-PVST+) and the logical continuation of that connectivity through MSTP bridges and regions, calculated by MSTP to ensure that all LANs in the bridged LAN are simply and fully connected

Multiple Spanning Tree Instance (MSTI)

One of a number of spanning trees calculated by MSTP within an MST Region, to provide a simply and fully connected active topology for frames classified as belonging to a VLAN that is mapped to the MSTI by the MST Configuration Table used by the MST Bridges of that MST Region.

The Brocade implementation supports up to 32 spanning tree instances in an MSTP enabled bridge that can support up to 32 different Layer 2 topologies. The spanning tree algorithm used by MSTP is RSTP, which provides quick convergence.

MSTP runs multiple instances of spanning tree that are independent of VLANs. It then maps each set of VLANs to each instance (forwarding path), which reduces the number of spanning tree instances needed to support a large number of VLANs. Each MSTP instance has a spanning tree topology independent of other spanning tree instances. With MSTP you can have multiple forwarding paths for data traffic. A failure in one instance does not affect other instances. With MSTP, you are able to more effectively utilize the physical resources present in the network and achieve better load balancing of VLAN traffic.

MST regions

MST regions are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels. Also, one or more VLANs can be mapped to one MST instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances

MSTP regions

Multiple devices must be configured consistently with the same MSTP configuration to participate in multiple spanning tree instances. A group of interconnected devices that have the same MSTP configuration is called an MSTP region.

> NOTE
> Brocade supports 32 MSTP instances and one MSTP region.

MSTP introduces a hierarchical way of managing device domains using regions. devices that share common MSTP configuration attributes belong to a region. The MSTP configuration determines the MSTP region where each device resides. The common MSTP configuration attributes are as follows:

- Alphanumeric configuration name (32 bytes)
- Configuration revision number (2 bytes)
- 4096-element table that maps each of the VLANs to an MSTP instance

Region boundaries are determined by the above configuration. A multiple spanning tree instance is an RSTP instance that operates inside an MSTP region and determines the active topology for the set of VLANs mapping to that instance. Every region has a CIST that forms a single spanning tree instance that includes all the devices in the region. The difference between the CIST instance and the MSTP instance is that the CIST instance operates across the MSTP region and forms a loop-free topology across regions, while the MSTP instance operates only within a region. The CIST instance operates using the devices across the regions. However, if any of the devices operate using 802.1d (STP), the CIST instance reverts to 802.1d. Each MSTP region is viewed logically as a single bridge to other regions.

MSTP guidelines and restrictions

Follow these restrictions and guidelines when configuring MSTP:

- Create VLANs before mapping them to MSTP instances.
- The MSTP **force-version** option is not supported.
- For two or more switches to be in the same MSTP region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same region name.
- MSTP is backward compatible with STP and RSTP.
- Only one MSTP region can be configured on a bridge.
- The Brocade implementation of MSTP supports up to 32 MSTP instances and one MSTP region.

- You must create VLANs before mapping them to MSTP instances.
- A maximum of 4090 VLANs can be configured across the 32 MSTP instances.
- MSTP and Topology groups cannot be configured together.
- MSTP configured over MCT VLANs is not supported.

Interoperability with PVST+ and R-PVST+

Since Brocade or other vendor devices enabled with PVST+ and R-PVST+ send IEEE STP BPDUs in addition to the PVST and R-PVST BPDUs, the VLAN 1 spanning tree joins the Common Spanning Tree (CST) of the network and thus interoperates with MSTP. The IEEE compliant devices treat the BPDUs addressed to the Brocade proprietary multicast MAC address as an unknown multicast address and flood them over the active topology for the particular VLAN.

MSTP global level parameters

To configure a switch for MSTP, first you set the region name and the revision on each switch that is being configured for MSTP. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments.

Each of the steps used to configure and operate MSTP are described in the following:

> NOTE
> The MSTP Region and Revision global parameters are enabled for interface level parameters as described below.

- Set the MSTP region name — Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions. The default MSTP name is "NULL".
- Set the MSTP revision number — Each switch that is running MSTP is configured with a revision number. It applies to the switch, which can have many different VLANs that can belong to many different MSTP regions.
- Enabling and disabling Cisco interoperability — While in MSTP mode, use the **cisco-interoperability** command to enable or disable the ability to interoperate with certain legacy Cisco switches. If Cisco interoperability is required on any switch in the network, then all switches in the network must be compatible, and therefore enabled by means of this command. By default the Cisco interoperability is disabled.

MSTP interface level parameters

Edge port

Configuring the edge port feature makes a port transition directly from initialization to the forwarding state, skipping the listening and learning states.

> NOTE
> This feature is only for RSTP , R-PVST+, and MSTP.

When edge port is enabled, the port still participates in STP.

When an edge port receives a BPDU, it's a normal spanning tree port and is no longer an edge port. When this feature is enabled on an interface the device is expected to put the port into FORWARDING state immediately.

BPDU guard

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow.

In a valid configuration, edge port configured interfaces do not receive BPDUs. If an edge port configured interface receives a BPDU, an invalid configuration exists, such as connection of an unauthorized device. The BPDU guard feature provides a secure response to invalid configurations because the administrator must manually put the interface back in service.

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Brocade device port to which the end station is connected. The STP BPDU guard shuts down the port and puts it into an "error disabled" state. This disables the connected device’s ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a message is displayed to warn the network administrator of an invalid configuration.

The BPDU guard feature provides a secure response to invalid configurations because the administrator must manually put the interface back in service with the **no shutdown** command if error disable recovery is not enabled by enabling the **errdisable-timeout** feature. The interface can also be automatically configured to be enabled after a timeout. However, if the offending BPDUs are still being received, the port is disabled again.

Expected behavior in an interface context

When this feature is enabled on an interface the device is expected to put the interface in Error Disabled state when BPDU is received on the port when edge-port and BPDU guard is enabled on the switch interface. When the port ceases to receive the BPDUs, it does not automatically switch to edge port mode, you must configure **error disable timeout** or **no shutdown** on the port to move the port back into edge port mode.

BPDU filtering

BPDU filtering allows you to avoid transmitting BPDUs on edge port enabled ports that are connected to an end system.

When you enable edge port on the switch, spanning tree places ports in the forwarding state immediately, instead of going through the listening and learning states. By default, spanning tree sends BPDUs from all ports regardless of whether edge port is enabled.

BPDU filtering is configured on a per-port basis.

Expected behavior in an interface context

When this feature is enables the device is expected to stop transmitting BPDUs on the interfaces enabled with edge port. For a BPDU filter configured interface connected to an end station, it is also is expected to put the port in forwarding state immediately when **edge-port bpdu-filter** is enabled on the device interface.

Restricted role

Configuring restricted role on a port causes the port not to be selected as root port for the CIST or any MSTI, even if it has the best spanning tree priority vector.

Restricted role ports are selected as an alternate port after the root port has been selected. It is configured by a network administrator to prevent bridges external to a core region of the network influencing the spanning tree active topology, possibly because those bridges are not under the full control of the administrator. It will protect the root bridge from malicious attack or even unintentional misconfigurations where a bridge device which is not intended to be root bridge, becomes root bridge causing severe bottlenecks in data path. These types of mistakes or attacks can be avoided by configuring 'restricted-role' feature on ports of the root bridge . This feature is similar to the "root-guard" feature which is proprietary implementation of Cisco for STP and RSTP but had been adapted in the 802.1Q standard as "restricted-role". The "restricted-role" feature if configured on an incorrect port can cause lack of spanning tree connectivity.

Expected behavior in an interface context

When this feature is enabled on an interface the device is expected to prevent a port configured with restricted-role feature from assuming the role of a Root port. Such a port is expected to assume the role of an Alternate port instead, once Root port is selected.

Restricted TCN

TCN BPDUs are used to inform other switches of port changes.

Configuring "restricted TCN" on a port causes the port not to propagate received topology change notifications and topology changes originated from a bridge external to the core network to other ports. It is configured by a network administrator to prevent bridges external to a core region of the network from causing MAC address flushing in that region, possibly because those bridges are not under the full control of the administrator for the attached LANs. If configured on an incorrect port it can cause temporary loss of connectivity after changes in a spanning trees active topology as a result of persistent incorrectly learned station location information.

Expected behavior in an interface context

When this feature is enabled on an interface, the device is expected to prevent propagation of topology change notifications from a port configured with the Restricted TCN feature to other ports. In this manner, the device prevents TCN propagation from causing MAC flushes in the entire core network.

Configuring MSTP

Enabling MSTP

Follow this procedure to configure the Multiple Spanning Tree Protocol.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create an MSTP context.

   ```
   device(config)# protocol spanning-tree mstp
   ```

   This command creates a context for MSTP. MSTP is automatically enabled. All MSTP specific CLI commands can be issued only from this context. Executing a **no** on this command deletes the context and all the configurations defined within the context.

3. Specify the region name.

   ```
   device(config-mstp)# region kerry
   ```

4. Specify the revision number.

   ```
   device(config-mstp)# revision 1
   ```

5. Configure an optional description of the MSTP instance.

   ```
   device(config-mstp)# description kerry switches
   ```

6. Configure a bridge priority for the CIST bridge.

   ```
   device(config-mstp)# bridge-priority 4096
   ```

   The range is 0 through 61440 in increments of 4096. The default is 32768.

7. Set the error disable parameters.
   a) Enable the timer to bring the port out of error disable state.

      ```
      device(config-mstp)# error-disable-timeout enable
      ```

   b) Specify the time in seconds it takes for an interface to time out..

      ```
      device(config-mstp)# error-disable-timeout interval 60
      ```

      The range is fro 10 to 1000000 seconds with a default of 300 seconds.

8. Configure forward delay.
   a) Specify the bridge forward delay.

      ```
      device(config-mstp)# forward-delay 16
      ```

   This command allows to specify how long an interface remains in the listening and learning states before the interface begins forwarding. This command affects all MSTP instances. The range of values is from 4 to 30 seconds with a default of 15 seconds.

9. Configure hello time.

   ```
   device(config-mstp)# hello-time 5
   ```

   The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The range is from 1 through 10 seconds with a default of 2 seconds.

10. Configure the maximum age.

    ```
    device(config-mstp)# max-age 16
    ```

    You must set the **max-age** so that it is greater than the **hello-time**. The range is 6 through 40 seconds with a default of 20 seconds.

11. Specify the port-channel path cost.

    ```
    device(config-mstp)# port-channel path-cost custom
    ```

    This command allows to control path-cost of port channel according to bandwidth or not.

    custom - path cost changes according to bandwidth.

    standard - path cost does not change according to bandwidth.

12. Specify the transmit hold count.

    ```
    device(config-mstp)# transmit-holdcount 5
    ```

    Tthe transmit-hold-count which is used to limit the maximum number of MSTP BPDUs that the bridge can transmit on a port before pausing for one second. The range is from 1 to 10 seconds with a default of 6 seconds.

13. Configure Cisco interoperability.

    ```
    device(config-mstp)# cisco-interoperability enable
    ```

    This command enables the ability to interoperate with certain legacy Cisco switches. The default is Cisco interoperability is disabled.

14. Return to privileged exec mode.

    ```
    device(config-mstp)# end
    ```

15. Verify the configuration. The following is an example configuration.

    ```
    device# show spanning-tree mst-config

    Spanning-tree Mode: Multiple Spanning Tree Protocol
     CIST Root Id: 8000.001b.ed9f.1700
     CIST Bridge Id: 8000.768e.f80a.6800
     CIST Reg Root Id: 8000.001b.ed9f.1700

     CIST Root Path Cost: 0; CIST Root Port: Eth 1/2
     CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 19
     Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20;
     Tx-HoldCount: 6
     Number of topology change(s): 139; Last change occurred 00:03:36 ago on Eth 1/2

     Bpdu-guard errdisable timeout: disabled
     Bpdu-guard errdisable timeout interval: 300 sec
     Migrate Time: 3 sec

      Name           : brcd
      Revision Level : 1
      Digest         : 0x9357EBB7A8D74DD5FEF4F2BAB50531AA

      Instance        VLAN
      --------        ----
       0:             1
       1:             10
       2:             20
    ```

16. Save the running configuration to the startup configuration.

    ```
    device# copy running-config startup-config
    ```

MSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# region kerry
device(config-mstp)# revision 1
device(config-mstp)# description kerry switches
device(config-mstp)# instance 1 vlan 7,8
device(config-mstp)# instance 2 vlan 21-23
device(config-mstp)# instance 1 priority 4096
device(config-mstp)# bridge-priority 4096
device(config-mstp)# error-disable-timeout enable
device(config-mstp)# error-disable-timeout interval 60
device(config-mstp)# forward-delay 16
device(config-mstp)# hello-time 5
device(config-mstp)# max-age 16
device(config-mstp)# port-channel path-cost custon
device(config-mstp)# transmit-holdcount 5
device(config-mstp)# cisco-interoperability enable
device(config-mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Configuring MSTP on an interface

Follow these steps to configure and enable MSTP on an Ethernet interface.

1. Enter configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface ethernet 1/5
   ```

3. Enable auto detection of an MSTP edge port.

   ```
   device(conf-if-eth-1/5)# spanning-tree autoedge
   ```

4. Configure the edge port

   ```
   device(conf-if-eth-1/5)# spanning-tree edgeport
   ```

5. Enable the BPDU filter.

   ```
   device(conf-if-eth-1/5)# spanning-tree egdeport bpdu-filter
   ```

6. Enable BPDU guard on the port

   ```
   device(conf-if-eth-1/5)# spanning-tree edgeport bpdu-guard
   ```

7. Set the path cost of a port.

   ```
   device(conf-if-eth-1/5)# spanning-tree cost 2000
   ```

   The path cost range is from 1 to 200000000. Leaving the default adjusts path cost relative to changes in the bandwidth. A lower path cost indicates greater likelihood of becoming root port.

8. Configure the restricted role feature for the port.

   ```
   device(conf-if-eth-1/5)# spanning-tree restricted-role
   ```

9. Configure restricted TCN feature for the port

   ```
   device(conf-if-eth-1/5)# spanning-tree restricted-tcn
   ```

10. Configure the link type.

    ```
    device(conf-if-eth-1/5)# spanning-tree link-type point-to-point
    ```

    The options are point-to-point or shared.

11. Enable port priority.

    ```
    device(conf-if-eth-1/5)# spanning-tree priority 128
    ```

    The range is from 0 to 240 in increments of 16 with a default of 32. A lower priority indicates greater likelihood of becoming root port.

12. Enable spanning tree on the port.

    ```
    device(conf-if-eth-1/5)# no spanning-tree shutdown
    ```

    By default spanning-tree is disabled on port and should be explicitly enabled.

13. Return to privileged exec mode.

    ```
    device(conf-if-eth-1/5)# end
    ```

14. Verify the configuration.

    ```
    device# show spanning-tree interface ethernet 1/5
    Spanning-tree Mode: Multiple Spanning Tree Protocol

     Root Id: 8000.001b.ed9f.1700
     Bridge Id: 8000.01e0.5200.011d

     Port Eth 1/5 enabled
        Ifindex: 411271175; Id: 8002; Role: Designated; State: Forwarding
        Designated External Path Cost: 0; Internal Path Cost: 20000000
        Configured Path Cost: 2000
        Designated Port Id: 8002; Port Priority: 128
        Designated Bridge: 8000.01e0.5200.011d
        Number of forward-transitions: 1
        Version: Multiple Spanning Tree Protocol - Received MSTP - Sent MSTP
        Edgeport: yes; AutoEdge: yes; AdminEdge: no; EdgeDelay: 3 sec
        Restricted-role is enabled
        Restricted-tcn is enabled
        Boundary: no
        Bpdu-guard: on
        Bpdu-filter: on
        Link-type: point-to-point
        Received BPDUs: 86; Sent BPDUs: 1654
    ```

15. Save the configuration.

    ```
    device# copy running-config startup-config
    ```

Enable MSTP on an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 1/5
device(conf-if-eth-1/5)# spanning-tree autoedge
device(conf-if-eth-1/5)# spanning-tree edgeport
device(conf-if-eth-1/5)# spanning-tree egdeport bpdu-filter
device(conf-if-eth-1/5)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-1/5)# spanning-tree cost 64000
device(conf-if-eth-1/5)# spanning-tree restricted-role
device(conf-if-eth-1/5)# spanning-tree restricted-tcn
device(conf-if-eth-1/5)# spanning-tree link-type point-to-point
device(conf-if-eth-1/5)# spanning-tree priority
device(conf-if-eth-1/5)# no spanning-tree shutdown
device(conf-if-eth-1/5)# end
device# show spanning-tree interface ethernet 1/5
device# copy running-config startup-config
```

Enabling MSTP on a VLAN

1. Enter configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the protocol command to enable MSTP configuration.

   ```
   device(config)# protocol spanning-tree mstp
   ```

3. Map a VLAN to an MSTP instance.

   ```
   device(config-mstp)# instance 5 vlan 300
   ```

4. Return to privileged exec mode.

   ```
   device(config-mstp)# end
   ```

5. Verify the configuration.

   ```
   device# show spanning-tree mst

    Spanning-tree Mode: Multiple Spanning Tree Protocol

    CIST Root Id: 8000.609c.9f5d.4800 (self)
    CIST Bridge Id: 8000.609c.9f5d.4800
    CIST Reg Root Id: 8000.609c.9f5d.4800 (self)

    CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
    Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20;
    Tx-HoldCount: 6
    Number of topology change(s): 0

    Bpdu-guard errdisable timeout: disabled
    Bpdu-guard errdisable timeout interval: 300 sec
    Migrate Time: 3 sec

     Name           : NULL
     Revision Level : 0
     Digest         : 0xD5FF4C3F6C18E2F27AF3A8300297ABAA

     Instance        VLAN
     --------        ----
      0:             1
      5:             100
   ```

6. Save the configuration.

   ```
   device# copy running-config startup-config
   ```

Enable spanning tree on a VLAN configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# instance 5 vlan 300
device(config-mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Link Aggregation

Link aggregation overview

Link aggregation allows you to bundle multiple physical Ethernet links to form a single logical trunk providing enhanced performance and redundancy. The aggregated trunk is referred to as a Link Aggregation Group (LAG). The LAG is viewed as a single link by connected devices, the Spanning Tree Protocol, IEEE 802.1Q VLANs, and so on. When one physical link in the LAG fails, the other links stay up. There is no disruption to traffic.

To configure links to form a LAG, the physical links must be of the same speed. Link aggregation can be done by manually configuring the LAG, or by dynamically configuring the LAG using the IEEE 802.3ad Link Aggregation Control Protocol (LACP).

When queuing traffic from multiple input sources to the same output port, all input sources are given the same weight, regardless of whether the input source is a single physical link or a trunk with multiple member links.

> NOTE
> The LAG or LAG interface is also referred to as a *port-channel* in the Brocade SLX 9850 platform.

The benefits of link aggregation are summarized as follows:

- Increased bandwidth (The logical bandwidth can be dynamically changed as the demand changes.)
- Increased availability
- Load sharing
- Rapid configuration and reconfiguration

Each LAG consists of the following components:

- A MAC address that is different from the MAC addresses of the LAG's individual member links.
- An interface index for each link to identify the link to the neighboring devices.
- An administrative key for each link. Only the links with the same administrative key value can be aggregated into a LAG. On each link configured to use LACP, LACP automatically configures an administrative key value equal to the port-channel identification number.

The Brocade SLX 9850 platform supports the following LAG types:

- Static LAG— In static link aggregation, links are added into a LAG without exchanging any control packets between the partner systems. The distribution and collection of frames on static links is determined by the operational status and administrative state of the link.
- Dynamic, standards-based LAG using LACP—Dynamic link aggregation uses LACP to negotiate with links that can be added and removed from a LAG. Typically, two partner systems sharing multiple physical Ethernet links can aggregate a number of those physical links using LACP. LACP creates a LAG on both partner systems and identifies the LAG by the LAG ID. All links with the same administrative key, and all links that are connected to the same partner switch become members of the LAG. LACP continuously exchanges LACPDUs to monitor the health of each member link.

The Brocade SLX 9850 platform supports the following trunk type:

- Static, standards-based LAG

The Brocade SLX 9850 platform supports the following LAG scalability configuration:

- 256 LAGs with each containing up to 64 ports
- 512 LAGs with each containing up to 32 ports

LAG load sharing

Brocade devices can be configured for load sharing over a LAG using:

- Hash-based load sharing

Hash-based load sharing

The Brocade device tries to share the traffic load evenly across the ports in LAG group, while ensuring that packets in the flow are not reordered. Individual flows are assigned a LAG index to identify them. An improved hash-based load sharing algorithm has the following enhancements:

- Better distribution
- Support for 32-port LAGs when the maximum number of LAGs in the system is 512.
- Support for 64-port LAGs when the maximum number of LAGs in the system is 256.
- An increased number of fields in the packet header that can be used for load balancing
- Enhanced load sharing in configurations of ECMP with LAGs.

Configuring LAG hashing

To configure symmetric LAG hashing on a Brocade SLX 9850 device, complete the following tasks.

1. Define where to start the picking headers for the key generation using the **lag hash hdr-start <fwd |term>** command.
   - fwd— start from the header that is used for the forwarding of the packet (inner header). This is the default option.
   - term— start from the last terminated header (outer header)
2. Configure the number of headers to be considered for LAG hashing using the **lag hash hdr-count <count>** command. The default value is 1. There can be a maximum of 3 headers based on the first header selected using the command in the above step.

The following options provide other LAG configurations to achieve specific tasks.

- Configure hash rotate using the **lag hash rotate <rotate-number>** command to provide different options for randomness of hashing. The number can be between 0 and 15. The default value is 3.
- Configure hash normalize by using the **lag hash normalize** command if there is a need to use the same hash in both directions. The normalize option is disabled by default.
- Allow the source port to be included in the hashing configuration using the **lag hash srcport** command. The source port is not used for hashing by default.
- To skip the entire MPLS label stack and pick only the BOS label for hashing, use the **lag hash bos <start | skip>**. The command default is If MPLS header is used for hashing, it will use all labels including BOS label for hashing.
  - start— start from BOS. This is the default option.
  - skip— hash from header next to BOS.
- Enter the **lag hash pwctrlword skip** command to skip password control word in the hashing configuration.

- The following MPLS transit node LSR hashing configuration options are available when using the **lag hash speculate-mpls** command. The default option is using the MPLS labels.
  - enable— Enables Speculative MPLS.
  - inner-eth— Enables inner ethernet header hash for L2VPN.
  - inner-ip-raw— Enables inner IPv4 header hash for L2VPN raw mode.
  - inner-ip-tag— Enables inner IPv4 header hash for L2VPN tag mode.
  - inner-ipv6-raw— Enables inner IPv6 header hash for L2VPN raw mode.
  - inner-ipv6-tag— Enables inner IPv6 header hash for L2VPN tag mode.
- Select the fields to be used for LAG hashing per-header type by entering the **[no] lag hash** *protocol-type packet-fields-to-be-used-for-hashing* command.
- Select the protocol header type using one of the following commands.

  NOTE
  Using the **no** form of the following commands will mask a certain field in the configuration and that field will not be used for load-balance hashing.

  - Ethernet headers: `[no] load-balance hash ethernet <sa-mac> < da-mac> < vlan > < etype>`
  - IPv4 and L4 headers: `[no] load-balance hash ip < src-ip > <dst-ip > < protocol > < src-l4-port> < dst-l4-port >`
  - IPv6 and L4 headers: `[no] load-balance hash ipv6 < ipv6-src-ip > < ipv6-dst-ip> < ipv6-next-hdr> <ipv6-src-l4-port> < ipv6-dst-l4-port>`
  - MPLS: `[no] load-balance hash mpls < label1 > <label2> < label3>`

Load balancing mechanism on different traffic types

The following table provides information about load balancing on different traffic types.

TABLE 2 Load balancing on different traffic types

Traffic type	Header field	Description
Layer 2/ Layer 3 packet load balancing	• Ethernet DA, SA, Etype, Vlan-id • IPv4/v6 dst IP, src IP • L4 Src-Port, Dst-Port	• Ethernet destination address, source address, ethernet type, VLAN ID load balancing • IPv4/v6 destination address, source address load balancing • Layer 4 source and destination port-based load balancing
VPLS/ VLL packet load balancing	CE to PE router traffic can use below fields for load-balancing similar to the Layer 2/ Layer 3 traffic) • Ethernet DA, SA, Etype, Vlan-id • IPv4/v6 dst IP, src IP • L4 Src-Port, Dst-Port PE to CE router traffic can use below fields for load-balancing • Customer (inner) ethernet DA, SA, Etype, Vlan-id • Customer (inner) IPv4/v6 dst IP, Ipv4/Ipv6 src IP, protocol • Customer (inner) L4 Src-Port, Dst-Port	CE to PE router traffic • Ethernet destination address, source address, ethernet type, VLAN ID load balancing • IPv4/v6 destination address, source address load balancing • Layer 4 source and destination port-based load balancing PE to CE router traffic • Customer ethernet destination and source address, ethernet type, VLAN ID load balancing • Customer IPv4/v6 destination address, source address load balancing • Customer Layer 4 source and destination port-based load balancing

TABLE 2 Load balancing on different traffic types (continued)

Traffic type	Header field	Description
MPLS LSR load balancing • Brocade SLX 9850 device provides multiple options to handle different MPLS transit hashing scenarios • The hashing options are mutually exclusive. If one option is enabled, the other option will be disabled.	IP over MPLS traffic going over transit node	Brocade supports speculate-mpls option as default which speculates the IPv4/IPv6 header after the MPLS labels and use the fields for hashing. This hashing scenario is handled by the **lag hash speculate-mpls enable** command in the global mode.
L2VPN (VPLS/VLL) traffic • The hashing options are mutually exclusive. If one option is enabled, the other option will be disabled.	L2VPN tagged mode with IPv4 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ip-tag** command in the global mode. Some sections of the IPv4 source and destination address fields are also used for load-balance hashing.
	L2VPN raw mode with IPv4 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ip-raw** command. Some sections of the IPv4 source and destination address fields are also used for load-balance hashing.
	L2VPN tagged mode with IPv6 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ipv6-tag** command. Some sections of the IPv6 source and destination address fields are also used for load-balance hashing.
	L2VPN raw mode with IPV6 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ipv6-raw** command. Some sections of the IPv6 source and destination address fields are also used for load-balance hashing.

Displaying LAG hashing

Use the **show port-channel load-balance** command to display the configured parameters for LAG hashing.

```
device# show port-channel load-balance
Header parameters
        Ethernet Mask: sa-mac da-mac etype vlan
        ip: src-ip dst-ip protocol src-l4-port dst-l4-port
        ipv6: ipv6-src-ip ipv6-dst-ip ipv6-next-hdripv6-src-l4-port ipv6-dst-l4-port
        mpls: label1 label2 label3

Hash Settings
        hdr-start:FWD, hdr-count:1, bos-start:0, bos-skip:0, skip-cw:0
        normalize:0, rotate:3, include_src_port:0, Disable: L2 0, ipv4 0, ipv6 0, mpls 0

mpls_speculate: Enabled

load-balance-type hash-based
```

Link Aggregation Control Protocol

Link Aggregation Control Protocol (LACP) is an IEEE 802.3ad standards-based protocol that allows two partner systems to dynamically negotiate attributes of physical links between them to form logical trunks. LACP determines whether a link can be aggregated into a LAG. If a link can be aggregated into a LAG, LACP puts the link into the LAG. All links in a LAG inherit the same administrative characteristics.

LACP operates in two modes:

- *Active mode*— LACP initiates the LACPDU exchange regardless of whether the partner system sends LACPDUs.
- *Passive mode* — LACP responds to Link Aggregation Control Protocol Data Units (LACPDUs) initiated by its partner system but does not initiate the LACPDU exchange.

LAG distribution process and conditions

The LAG aggregator is associated with the collection and distribution of Ethernet frames. The collection and distribution process is required to guarantee the following:

- Inserting and capturing control PDUs.
- Restricting the traffic of a given conversation to a specific link.
- Load balancing between individual links.
- Handling dynamic changes in LAG membership.

On each port, link aggregation control does the following:

- Maintains configuration information to control port aggregation.
- Exchanges configuration information with other devices to form LAGs.
- Attaches ports to and detaches ports from the aggregator when they join or leave a LAG.
- Enables or disables an aggregator's frame collection and distribution functions.

LAG configuration guidelines:

- Each link in the Brocade SLX 9850 hardware can be associated with a LAG; a link cannot be associated with more than one LAG. The process of adding and removing links to and from a LAG is controlled statically or dynamically (through LACP).
- The maximum number of port members that may be assigned to a LAG depends on the LAG profile configuration. By default, the Brocade SLX 9850 platform can have a maximum of 256 LAGs with the maximum of 64 ports in each LAG. With the LAG profile 1 , Brocade SLX 9850 platform can have a maximum of 512 LAGs with the maximum of 32 ports in each LAG.
- Interfaces configured as switchport interfaces cannot be aggregated into a LAG. However, a LAG can be configured as a switchport.

Configuring and managing Link Aggregation

The following sections discuss working with the Link Aggregation on Brocade devices.

Configuring a new port channel interface

Follow this procedure to create a new port channel interface at the global configuration mode.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **interface port-channel** command to create a new port channel interface at the global configuration level.

   ```
   device(config)# interface port-channel 30
   ```

   NOTE

   The port-channel interface ranges from 1 to 512.

The following example creates a new port channel interface of 30.

```
device# configure terminal
device(config)# interface port-channel 30
```

Deleting a port channel interface

Follow this procedure to delete a port channel interface and all member interfaces from the specified LAG at the global configuration mode.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **no interface port-channel** command to delete an existing port channel interface at the global configuration level.

   ```
   device(config)# no interface port-channel 30
   ```

   NOTE
   The port-channel interface ranges from 1 to 512.

The following example deletes the existing port channel interface 30 from the specified LAG.

```
device# configure terminal
device(config)# no interface port-channel 30
```

Adding a member port to a port channel

Follow this procedure to add a port to a specific port channel interface at the interface configuration level. If the port channel is not created, the **channel-group** command creates the port channel and also adds a port to the port channel.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface port-channel** command to add a port channel interface at the global configuration level.

   ```
   device(config)# interface port-channel 30
   device(conf-Port-channel-30)#
   ```

3. Configure the **interface ethernet** command to enable the interface.

   ```
   device(conf-Port-channel-30)# interface ethernet 1/5
   device(conf-if-eth-1/5)#
   ```

4. Add a port to the port channel interface as static.

   ```
   device(conf-if-eth-1/5)# channel-group 10 mode on
   ```

5. Add a port to the port channel interface as a dynamic (using LACP), active or passive mode.

   ```
   device(conf-if-eth-1/5)# channel-group 10 mode active

   device(conf-if-eth-1/5)# channel-group 10 mode passive
   ```

The following example is for a static LAG configuration with the mode ON.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 1/5
device(conf-if-eth-1/5)# channel-group 10 mode on
```

The following example adds a port 1/5 to the existing dynamic port channel interface 30 with the mode active.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 1/5
device(conf-if-eth-1/5)# channel-group 30 mode active
```

The following example adds a port 1/5 to the existing dynamic port channel interface 30 with the mode passive.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 1/5
device(conf-if-eth-1/5)# channel-group 30 mode passive
```

Deleting a member port from a port channel

Follow this procedure to delete a member port from a port channel interface at the interface configuration level.

Delete a port from the port channel interface.

```
device(conf-if-eth-1/5)# no channel-group
```

The following example deletes a port 1/5 from the existing port channel interface 30 in a LAG.

```
device# configure terminal
device(config)# interface ethernet 1/5
device(conf-if-eth-1/5)# no channel-group
```

Configuring the minimum number of LAG member links

Follow this procedure to configure the minimum number of LAG member links that should be functional so that the port-channel interface is operationally up.

This configuration allows a port-channel to operate at a certain minimum bandwidth at all times. If the bandwidth of the port-channel drops below the minimum number, then the port-channel is declared operationally DOWN even though it has operationally UP members.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface port-channel** command at the global configuration level.

   ```
   device(config)# interface port-channel 30
   device(conf-Port-channel-30)#
   ```

3. Configure the minimum number of LAG member links at the port-channel interface configuration mode.

   ```
   device(conf-Port-channel-30)# minimum-links 5
   ```

   NOTE
   The number of links ranges from 1 to 64. The default minimum links is 1.

The following example sets min-link 5 to the existing port channel interface 30 in a LAG.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# minimum-links 5
```

Configuring the LACP system priority

The switch must be in privileged EXEC mode.

You configure the LACP system priority on each switch running LACP. LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other switches.

The system priority value must be a number in the range of 1 through 65535. The higher the number, the lower the priority. The default priority is 32768.

To configure the global LACP system priority, perform the following steps:

1. Enter the **configure terminal** command to access global configuration mode.
2. Specify the LACP system priority.

   ```
   device(config)# lacp system-priority 25000
   ```

3. To reset the system priority to the default value.

   ```
   device(config)# no lacp system-priority
   ```

Configuring the LACP port priority

Follow this procedure to configure the LACP port priority of a member port of a specific port-channel interface.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface port-channel** command to add a port channel interface at the global configuration level.

   ```
   device(config)# interface port-channel 30
   device(conf-Port-channel-30)#
   ```

3. Configure the **interface ethernet** command and add the port to the port-channel interface.

   ```
   device(conf-Port-channel-30)# interface ethernet 1/5
   device(conf-if-eth-1/5)#channel-group 30 mode active
   ```

4. Configure the LACP port priority 12 for the member port.

   ```
   device(conf-if-eth-1/5)# lacp port-priority 12
   ```

   NOTE
   The LACP port priority value ranges from 1 to 65535. The default value is 32768.

5. To rest the configured port priority to the default value.

   ```
   device(conf-if-eth-1/5)# no lacp port-priority
   ```

The example sets the port priority as 12.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 1/5
device(conf-if-eth-1/5)# channel-group 30 mode active
device(conf-if-eth-1/5)# lacp port-priority 12
```

Configuring the LACP timeout period

The device must be in privileged EXEC mode.

The LACP timeout period indicates how long LACP waits before timing out the neighboring device. The **short** timeout period is 3 seconds and the **long** timeout period is 90 seconds. The default is **long**.

To configure the LACP timeout period on an interface, perform the following steps:

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command, specifying the interface type and the slot/port.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Enter the **no shutdown** command to enable the interface.
4. Specify the LACP timeout short period for the interface.

   ```
   device(conf-if-eth 1/1)# lacp timeout short
   ```

5. Specify the LACP timeout long period for the interface.

   ```
   device(conf-if-eth 1/1)# lacp timeout long
   ```

Configuring LACP default Up

Follow this procedure to activate an LACP link in the absence of PDUs on the interface mode.

Consider the following when using the **lacp default-up** command:

- The command is available only if the configured interface is a dynamic member of a port-channel interface.
- The command is not supported on static LAGs.
- The command is not supported on port-channel interfaces.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command, specifying the interface type and the slot/port.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Specify LACP default-up for the interface.

   ```
   device(conf-if-eth-1/1)# lacp default-up
   ```

4. Enter the no form of the command to disable the configuration.

   ```
   device(conf-if-eth-1/1)#  no lacp default-up
   ```

Displaying port-channel information

Various show commands are used to display information for a specific port-channel interface.

Before displaying the port-channel information, you should have created a port-channel interface in a LAG to generate details.

1. Use the **show port-channel summary** command to display brief information of all port-channels.

```
device# show port-channel summary
Flags:  D - Down                P - Up in port-channel (members)
        U - Up (port-channel)   * - Primary link in port-channel
        S - Switched
        M - Not in use. Min-links not met
===== =============== ========== ===============
Group Port-channel    Protocol   Member ports
===== =============== ========== ===============
1     Po 1    (D)     None       Eth 2/125 (D)
                                 Eth 4/125 (D)
2     Po 2    (D)     None       Eth 2/126 (D)
                                 Eth 4/126 (D)
10    Po 10   (U)     LACP       Eth 2/4* (P)
                                 Eth 2/18 (P)
100   Po 100  (U)     None       Eth 2/10* (P)
                                          Eth 2/11 (P)
```

2. Use the **show port-channel detail** command to display detailed information of all the port-channels.

```
device# show port-channel detail
 Static Aggregator: Po 1
 Aggregator type: Standard
 Number of Ports: 2
 Member ports:
   Eth 2/125
   Eth 4/125


 Static Aggregator: Po 2
 Aggregator type: Standard
 Number of Ports: 2
 Member ports:
   Eth 2/126
   Eth 4/126


 Static Aggregator: Po 100
 Aggregator type: Standard
 Number of Ports: 2
 Member ports:
   Eth 2/10   *
   Eth 2/11


 LACP Aggregator: Po 10
Aggregator type: Standard
  Actor System ID - 0x8000,76-8e-f8-0a-98-00
  Admin Key: 0010 - Oper Key 0010
  Receive link count: 2 - Transmit link count: 2
  Individual: 0 - Ready: 1
  Partner System ID - 0x8000,76-8e-f8-0a-68-00
  Partner Oper Key 0010
 Number of Ports: 2
 Member ports:
   Link: Eth 2/4 (0x18820016) sync: 1   *
   Link: Eth 2/18 (0x18890084) sync: 1
```

3. Use the **port-channel num** command to display detailed information of a specific port-channel interface

```
device# show port-channel 10
LACP Aggregator: Po 10
 Aggregator type: Standard
  Admin Key: 0010 - Oper Key 0010
  Partner System ID - 0x8000,76-8e-f8-0a-68-00
  Partner Oper Key 0010
 Number of Ports: 2
 Member ports:
   Link: Eth 2/4 (0x18820016) sync: 1   *
   Link: Eth 2/18 (0x18890084) sync: 1   *
```

Displaying LACP system-id information

Follow this procedure to display LACP system ID and priority information.

Enter the **show lacp sys-id** command to display LACP information for the system ID and priority.

```
device# show lacp sys-id
System ID: 0x8000,76-8e-f8-0a-98-00
```

Displaying LACP statistics

Follow this procedure to display LACP statistics for a port-channel interface or for all port-channel interfaces.

Before displaying the LACP port-channel information, it is recommended that you create a port-channel interface in a LAG to generate details.

Enter the **show lacp counters** command to display LACP statistics for a port-channel.

```
device# show lacp counter
Traffic statistics
Port            LACPDUs            Marker           Pckt err     Sent     Recv     Sent     Recv
Sent     Recv
Aggregator Po 3  Eth 1/6                                110      0            0        0
0        0
```

Clearing LACP counter statistics on a LAG

This topic describes how to clear LACP counter statistics on a single LAG.

To clear LACP counter statistics on a LAG, use the following command:

Enter the **clear lacp** *LAG_group_number* **counters** command to clear the LACP counter statistics for the specified LAG group number.

```
device# clear lacp 42 counters
```

Clearing LACP counter statistics on all LAG groups

This topic describes how to clear the LACP counter statistics for all LAG groups.

To clear LACP counter statistics on all LAG groups, use the following command:

Enter the **clear lacp counter** command to clear the LACP counter statistics for all LAG groups.

```
device# clear lacp counter
```

VLANs

802.1Q VLAN overview

IEEE 802.1Q VLANs provide the capability to overlay the physical network with multiple virtual networks. VLANs allow you to isolate network traffic between virtual networks and reduce the size of administrative and broadcast domains.

A VLAN contains end stations that have a common set of requirements that are independent of physical location. You can group end stations in a VLAN even if they are not physically located in the same LAN segment. VLANs are typically associated with IP subnetworks and all the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. VLAN membership is configurable on a per-interface basis.

Configuring VLANs

The following sections discuss working with VLANs on Brocade devices.

Configuring a VLAN

Follow this procedure to configure a VLAN in the Brocade device at the global configuration level.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
device(config)#
```

2. Enter the **vlan** command to create a topology group at the global configuration level.

```
device(config)# vlan 5
device(config-vlan-5)#
```

NOTE
The **no vlan** command removes the existing VLAN instance from the device.

Configuring a switchport interface

Follow this procedure to configure a switchport interface in the device to send and receive data packets.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
device(config)#
```

2. Enter the **interface ethernet** command to configure the interface mode.

```
device(config)# interface ethernet 1/1
```

3. Enter the **switchport** command to configure a switchport interface.

   ```
   device(config-if-eth-1/1)# switchport
   ```

The example configures a switchport interface in the device.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# switchport
```

Configuring the switchport interface mode

Follow this procedure to set the switchport interface as access or trunk. This configuration works only when the interface is set as switchport.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface ethernet** command to configure the interface mode.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Enter the **switchport mode** command to configure a switchport interface.

   ```
   device(config-if-eth-1/1)# switchport mode trunk
   ```

   NOTE
   The default mode is access. Enter the **switchport mode access** command to set the mode as access.

   ```
   device(config-if-eth-1/1)# switchport mode access
   ```

The example sets the switchport interface mode as trunk in the device.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# switchport mode access
```

Configuring the switchport access type

Follow this procedure to change the switchport access VLAN type.

Before changing the access type in the device, you should make sure that the reserved VLANs are not used to configure access VLAN. Using the **no switchport access vlan** command will set the default VLAN as the access VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface ethernet** command to configure the interface mode.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Enter the **switchport access vlan** command to set the mode of a specific port-channel interface.

   ```
   device(config-if-eth-1/1)# switchport access vlan 10
   ```

The example sets the mode of a specific Ethernet interface to *access*.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# switchport access vlan 10
```

The example sets the mode of a specific port-channel interface to *trunk*.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-port-channel-35)# no switchport access 6
```

Configuring VLAN in the trunk mode

Follow this procedure to add or remove VLANs on a Layer 2 interface in the trunk mode. The configuration is also used to configure the VLANs to send and receive data packets.

Before configuring a VLAN in the trunk mode, ensure that the reserved VLANs are not used to configure access VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface ethernet** command to configure the interface mode.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Enter the **switchport trunk allowed vlan** command to set the mode of an Ethernet interface to *access*.

   ```
   device(config-if-eth-1/1)# switchport trunk allowed vlan add 5
   ```

The example sets the mode of a specific Ethernet interface to *access*.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# switchport trunk allowed vlan add 5
```

The example sets the mode of a specific port-channel interface to *trunk*.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-port-channel-35)# switchport trunk allowed all
```

Configuring native VLAN on a trunk port

Follow this procedure to set native VLAN characteristics on a trunk port for classifying the untagged traffic data packets.

Before configuring a VLAN in the trunk mode, ensure that the reserved VLANs are not used to configure access VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface ethernet** command to configure the interface mode.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Enter the **switchport trunk native-vlan** command to set native VLAN characteristics on a trunk port to *access*.

   ```
   device(config-if-eth-1/1)# switchport trunk native-vlan 300
   ```

The example sets the mode of a specific Ethernet interface to *access*.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# switchport trunk native-vlan 300
```

The example sets the mode of a specific port-channel interface to *trunk*.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-port-channel-35)# no switchport trunk native-vlan
```

Enabling VLAN tagging for native traffic

Follow this procedure to enable tagging for native traffic on a specific interface.

Before configuring a VLAN in the trunk mode, ensure that the reserved VLANs are not used to configure access VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **interface ethernet** command to configure the interface mode.

   ```
   device(config)# interface ethernet 1/1
   ```

3. Enter the **switchport trunk tag native-vlan** command to enable tagging for native traffic data VLAN characteristics on a specific interface.

   ```
   device(config-if-eth-1/1)# switchport trunk tag native-vlan
   ```

The example enables tagging for native data on a specific Ethernet interface to *access*.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# switchport trunk tag native-vlan
```

The example enables tagging for native data on a specific port-channel interface to *trunk*.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-port-channel-35)# no switchport trunk tag native
```

Configuring a static MAC address

Follow this procedure to configure static (unicast) addresses to be added to the MAC address table. This configuration also sets the aging time that a MAC address will persist after the last update.

Before configuring a VLAN in the trunk mode, ensure that the reserved VLANs are not used to configure access VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **mac-address-table static** command to add the static address 0011.2222.3333 to the MAC address table with a packet received on VLAN 100.

   ```
   device(config)# mac-address-table static 0011.2222.3333 forward ethernet 1/1 vlan 100
   ```

3. Enter the **mac-address-table aging-time** command to set the aging time as 572 seconds.

   ```
   device(config)# mac-address-table aging-time 572
   ```

   NOTE
   The maximum value supported is 572 seconds. The default time is 300 seconds.

Layer 2 forwarding considerations

The forwarding considerations are as follows:

- There maybe a short delay in the configured mac aging time. The hardware scans the MAC entries every 1/7th of the aging time. For example, if the configured aging time is 300 seconds, there can be a delay of 43 seconds.
- The MAC aging time may take longer than expected. In a scaled environment, the hardware sends only 16K aging events for every aging cycle.

Displaying switchport interface

Follow the procedure to display the detailed Layer 2 information for all interfaces.

Enter the **show interface switchport** to display the detailed Layer 2 information for all interfaces.

```
device# show interface switchport
Interface name : Ten Gigabit Ethernet 0/1/2
Switchport mode : access
Ingress filter : enable
Acceptable frame types : all
Default Vlan : 1
Active Vlans : 1
Inactive Vlans : -
Interface name : Ten port-channel 5
Switchport mode : access
Ingress filter : enable
Acceptable frame types : all
Default Vlan : 1
Active Vlans : 1
Inactive Vlans : 100
```

Displaying switchport interface type

Follow the procedure to display the detailed Layer 2 information for a specific interface.

Enter the **show interface switchport** to display the detailed Layer 2 information for a specific interface.

```
device# show interface ten 1/0/6 switchport
 Interface name          : TenGigabitEthernet 1/0/6
 Switchport mode         : trunk
 Fcoeport enabled        : no
 Ingress filter          : enable
 Acceptable frame types  : vlan-tagged only
 Native Vlan             : 1
 Active Vlans            : 1,5-10
 Inactive Vlans          : -
 MAC learn disable Vlans : -
```

The example displays the detailed Layer 2 information for a port-channel interface.

```
device# show interface port-channel 5 switchport
 Interface name          : Port-channel 5
 Switchport mode         : access
 Fcoeport enabled        : no
 Ingress filter          : enable
 Acceptable frame types  : vlan-untagged only
 Default Vlan            : 1
 Active Vlans            : 1
 Inactive Vlans          : -
 MAC learn disable Vlans : -
```

Verifying switchport interface running configuration

Follow the procedure to display the running configuration information of the Layer 2 properties for a specific interface.

Enter the **show running-config interface** to display the running configuration information for a specific interface.

```
device# show running-config interface TenGigabitEthernet 1/0/6 switchport
interface TenGigabitEthernet 1/0/6
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 5-10
 switchport trunk tag native-vlan
```

The example displays the running configuration information for a port-channel interface.

```
device# show running-config interface Port-channel 5 switchport
interface Port-channel 5
 switchport
 switchport mode access
 switchport access vlan 1
```

Displaying VLAN information

Follow the procedure to display information about a specific VLAN.

Enter the **show vlan** to display information about VLAN 1.

```
device# show vlan 1
VLAN Name State Ports
(u)-Untagged, (t)-Tagged
(c)-Converged
======= ================ ======= ===============================
1 default ACTIVE Te 0/0(t) Te 0/4(t) Te 0/5(t)
Te 0/8(t) Te 0/10(
```

VPLS and VLL Layer 2 VPN services

VPLS overview

Virtual Private LAN Service (VPLS) is a Layer 2 Virtual Private Network (L2 VPN) architecture that provides multipoint Ethernet LAN services.

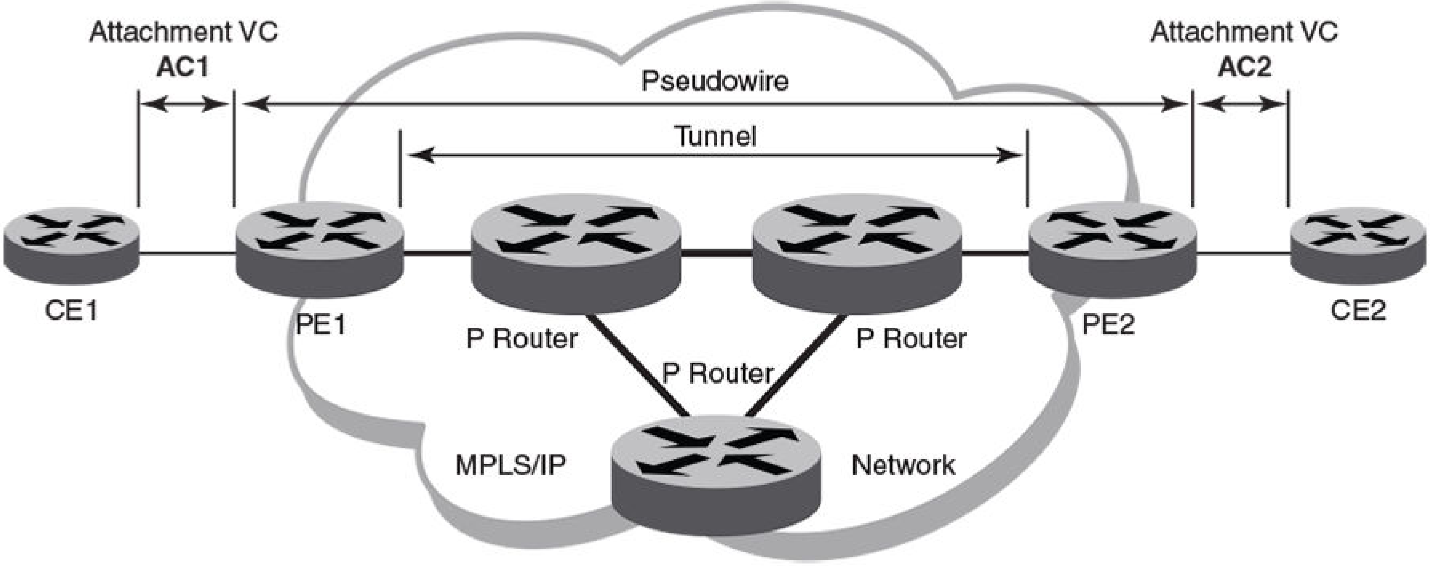
VPLS provides transparent LAN services across provider edge (PE) devices using Internet Protocol (IP) or Multiprotocol Label Switching (MPLS) as the transport technology.

Because it emulates LAN switching, VPLS is considered to be a L2 service that operates over Layer 3 (L3) clouds.

VPLS provides point-to-multipoint (p2mp) functionality.

The following figure shows a VPLS topology in which switched packets traverse a network.

FIGURE 3 VPLS topology with switching between attachment circuits (ACs) and network core



AC1 and AC2 represent L2 connectivity between customer edge (CE) and provider edge (PE) devices.

Pseudowire is a circuit emulation infrastructure that extends L2 connectivity from CE1 to CE2 by way of PE1 and PE2. The tunnel is typically a L3 tunnel on which a L2 circuit is emulated.

In the case of a packet flowing from CE1 to CE2, the packet enters PE1 from CE1 after the forwarding database (FDB) is used to determine the destination MAC address. Then, a virtual connection (VC) label is imposed prior to encapsulation with the tunnel forwarding information, and the packet is sent out onto the wire towards the network core.

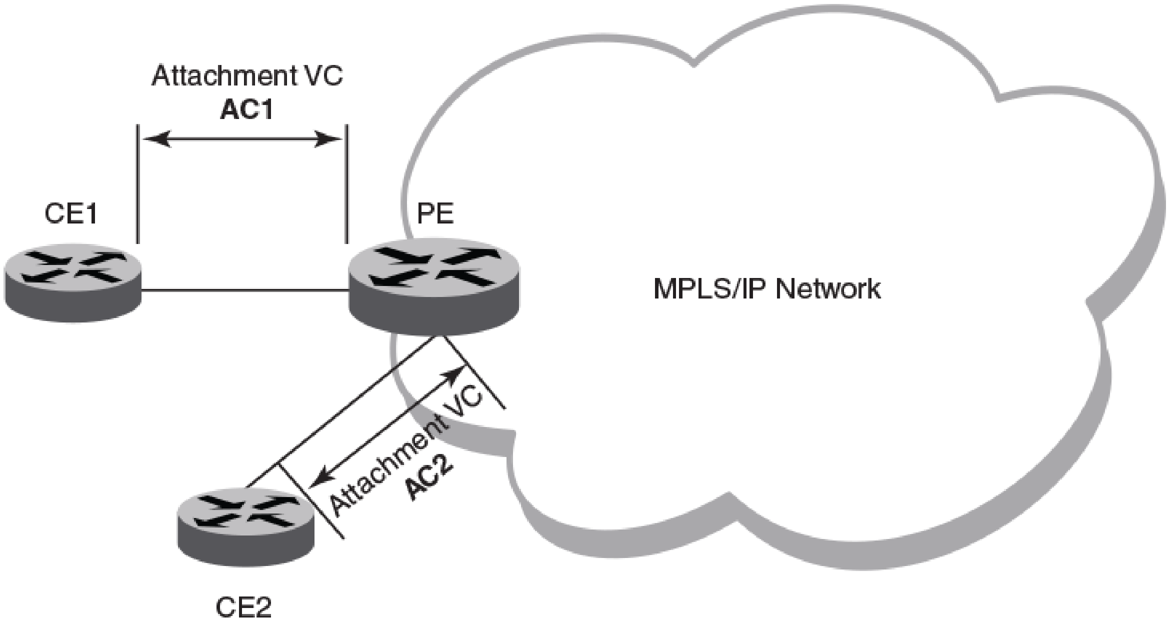
Essentially, the topology in the preceding figure shows a L2 VPN enabling the transport of L2 traffic between two or more native Ethernet networks through an underlying Multiprotocol Label Switching (MPLS) provider network. Customer edge (CE) is the last mile and provider edge (PE) is the first mile node for packets transported towards the provider network. The provider intermediary network is an emulated switch (LAN) or wire (LINE) to the CE. The attachment circuit (AC) represents the logical link between the CE and PE.

An AC may be a port, IEEE 802.1q or IEEE 802.1ad (QinQ)) for Ethernet VPNs. A pseudowire (PW) or emulated wire is used as a transport mechanism to tunnel frames between PEs. A PW is characterized by a circuit identifier, which identifies the destination PE.

MPLS tunnels and paths are established by using routing protocols. PW circuits are established by using signaling.

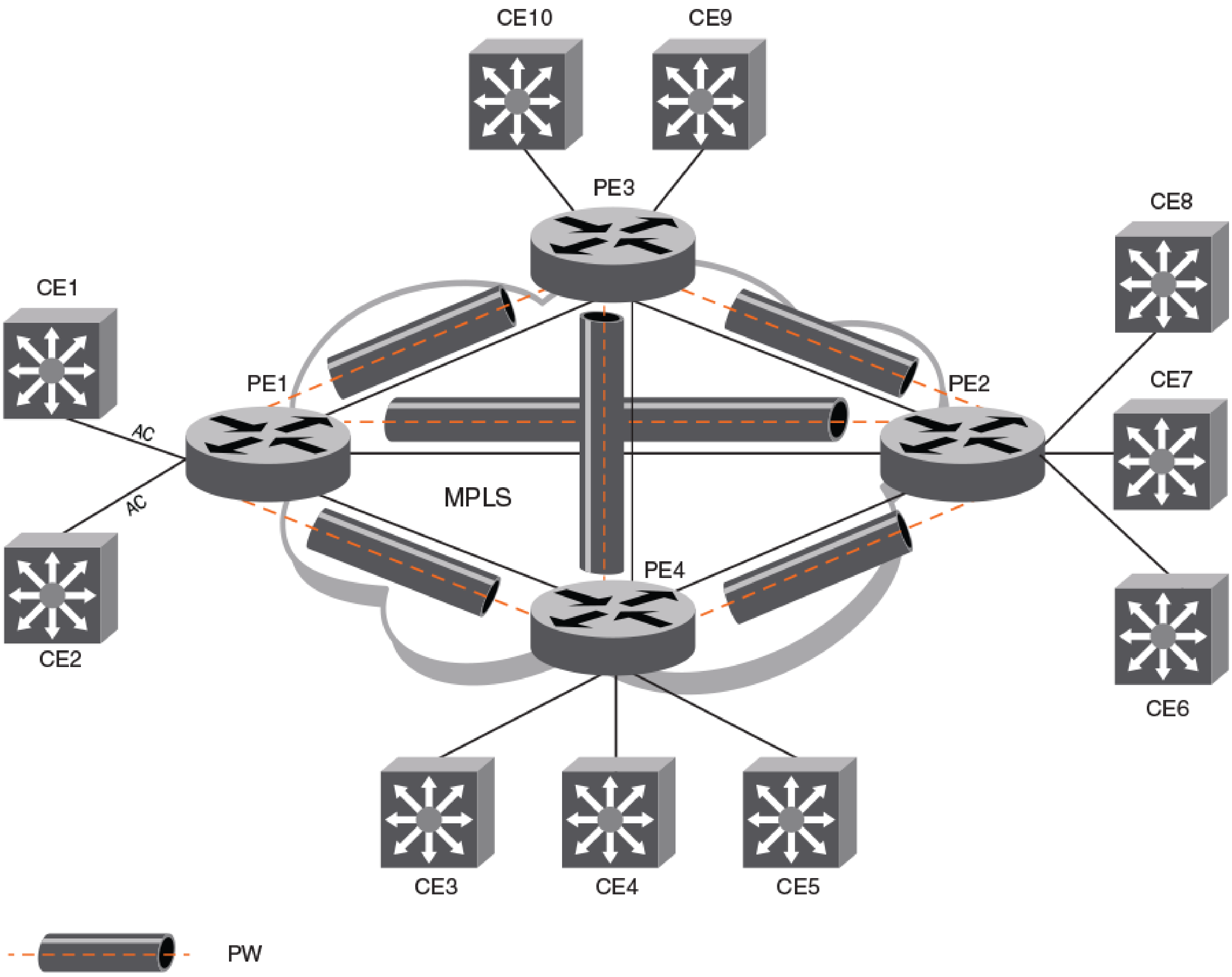
The following figure shows a VPLS topology where switching occurs between two local AC endpoints. This implementation of VPLS does not use VC labels or a pseudowire.

FIGURE 4 VPLS topology with local switching



The following figure shows a common VPLS deployment; an enterprise LAN service. The CE devices represent customer edge devices while the PE devices represent provider edge devices.

FIGURE 5 Enterprise LAN service (VPLS)



VLL

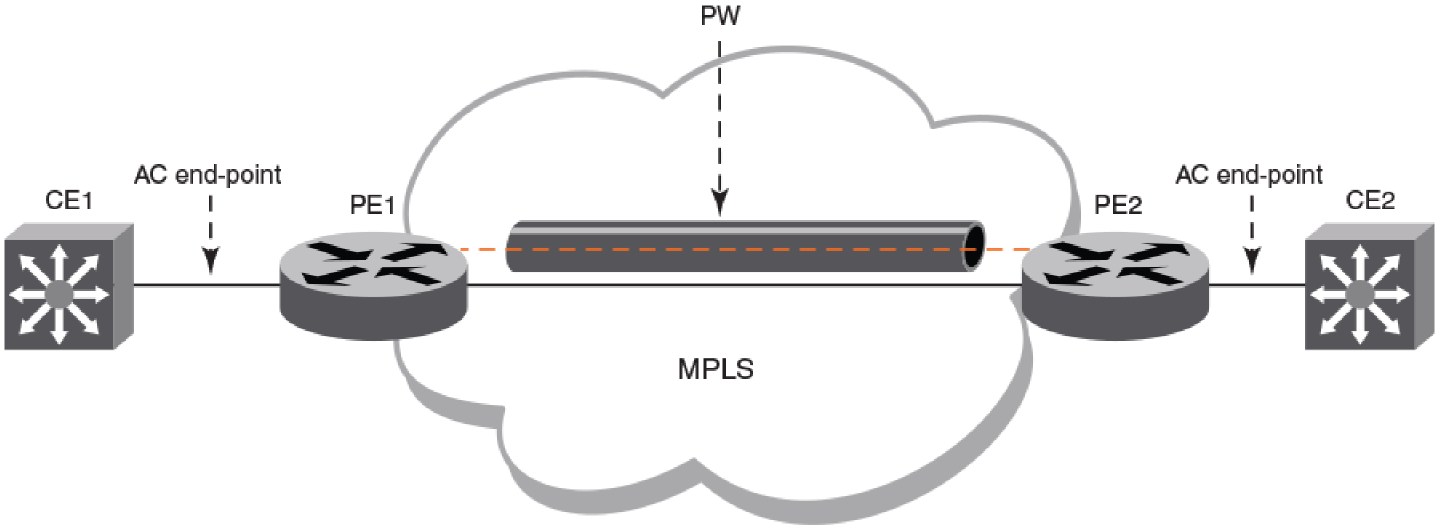
Virtual Leased Line (VLL) is a Layer 2 Virtual Private Network (L2 VPN) architecture that provides point-to-point Ethernet line or Virtual Private Wire Services (VPWS).

A VLL instance is a special type of VPLS deployment.

VLL provides point-to-point (p2p) connectivity between two access networks or endpoints. Typically, a VLL is used to connect two sites that are geographically apart.

The following figure depicts an enterprise VLL service.

FIGURE 6 Enterprise leased line service (VLL)



CE1 and CE2 are the customer edge devices in geographically separate sites.

Pseudowire (PW) is a circuit emulation infrastructure that extends L2 connectivity from CE1 to CE2 by way of PE1 and PE2. The tunnel is typically a L3 tunnel on which a L2 circuit is emulated.

VPLS service endpoints

VPLS supports two types of service-endpoints for VPLS and VLL.

Service endpoints can be categorized as:

- AC endpoints
- PW endpoints

An AC endpoint is a L2 link between a PE device and a CE device. The AC endpoint can be an untagged port, or a tagged port with one or more VLANs. AC endpoints with different VLAN tags can be configured in a single VPLS instance.

A VLL instance interconnects two AC endpoints through a pseudowire, while a VPLS instance forms a full mesh interconnection between multiple AC endpoints through multiple PWs.

The following endpoints are supported:

- port-vlan
- port-vlan-vlan
- PW

Both regular port and port-channel interfaces can be used to form port-vlan, untagged port-vlan, and port-vlan-vlan endpoints.

VPLS service endpoints are represented by logical interfaces (LIFs). By using LIFs, features that apply to regular interfaces, such as QoS, can be applied to VPLS service endpoints.

Local switching

The forwarding behavior of AC endpoints in a VPLS instance is controlled by the switching-mode configuration for local endpoints.

When local switching is enabled, traffic is switched and flooded among AC endpoints in addition to between ACs and PWs. When local switching is disabled, the forwarding between AC endpoints is suppressed.

When an unknown unicast packet is received on an AC endpoint and local switching is enabled, the packet is flooded to all other AC endpoints and PW endpoints in the VPLS instance. When local switching is disabled, the unknown packet is only flooded to the PW endpoints in the domain.

Regardless of the local switching configuration, an unknown unicast packet that is received on a PW endpoint is flooded to all AC endpoints.

By default, local switching is enabled.

In a VPLS instance that does not have a PW peer and where all endpoints are AC endpoints (Local VPLS), local switching must be enabled.

To avoid receipt of traffic with different VLAN tags on local endpoints, it is recommended that local switching is disabled in a bridge domain where the PW profile is configured with the VC mode option of **raw-passthrough**. Raw passthorugh mode is designed to forward packets between two VPLS peer devices and is not intended for use with local switching.

Bridge domains

Bridge domain is an infrastructure that supports the implementation of different switching technologies.

A bridge domain is a generic broadcast domain that is not tied to a specific transport technology. Bridge domains support a wide range of service endpoints including regular L2 endpoints and L2 endpoints over L3 technologies.

Bridge domains switch packets between a range of different endpoint types; for example, attachment circuit (AC) endpoints, Virtual Private LAN Service (VPLS) endpoints, Virtual Leased Line (VLL) endpoints, and tunnel endpoints.

VPLS performs multipoint switching, while VLL performs one-to-one switching.

A bridge domain that is created for a VPLS application is also referred to as a VPLS instance.

The following services are bridge-domain capable:

- VPLS—with multiple AC endpoints and pseudowire (PW) logical interfaces (LIFs)
- Local VPLS—with multiple AC endpoints
- VLL—with one AC endpoint and one PW endpoint

Pseudowires

A pseudowire (PW) is a virtual circuit (VC) formed between two PE devices that connect two attachment circuits (ACs).

An Ethernet pseudowire is logically viewed as an L2 nexthop (VC label) that is reachable through an L3 nexthop (LDP label).

The frames from an AC endpoint packet are sent through an ingress pseudowire interface (which abstracts the transport path and packet encapsulations) towards the remote PE. An egress pseudowire interface then abstracts the packet received from a remote PE and hands it over to the corresponding AC end-point.

A pseudowire interface is unidirectional.

PWs support the following underlying MPLS tunnels:

- LDP – Single Path LSP

- RSVP – Single Path LSP
- RSVP – Pri/Sec (Act LSP)
- RSVP – Pri/Sec (Pas LSP)
- FRR: Adaptive LSP (Make Before Break)
- FRR: Protected & detour (1:1)

PWs do not support the following underlying MPLS tunnels:

- FRR: Protected & Bypass (N:1)
- LDP – Multipath LSP (ECMP)
- LDP over RSVP

Pseudowire operation

The pseudowire setup process establishes the reachability of VPLS bridge domain endpoints across an IP or MPLS cloud.

A pseudowire is operational when the following conditions are met:

- VC signaling is established.
- The L3 reachability of the PW peer is resolved.
- At least one AC endpoint within the bridge domain is up.

A pseudowire is non-operational when the following conditions are met:

- No logical interface is configured for the VPLS instance.
- All AC endpoints are non-operational.

Scalability

Normal pseudowires and load-balanced pseudowires are supported.

An overall maximum of 8000 pseudowires is supported. This includes a maximum of:

- 1000 load-balanced pseudowires (each with a maximum of 16 paths)
- 7000 normal pseudowires

Supported pseudowire features

Pseudowires (PWs) support the following features for each configured PW:

- LSP Load Balancing—Load balancing across a maximum of 16 underlying MPLS tunnels.
- Assigned LSP—A maximum of 32 LSPs can be assigned.
- Specific COS—The underlying MPLS tunnel with the closest CoS value is selected for the transport
- Raw, raw-passthrough, or tagged mode—Can be configured by way of the PW profile that is associated with the bridge domain.
- MTU and MTU check—Can be configured by way of the PW profile that is associated with the bridge domain.
- Uniform and pipe mode for QoS
- Statistics—Egress and ingress statistics are supported but must be enabled in the bridge-domain configuration by using the **statistics** command

Unsupported pseudowire features

Pseudowires (PWs) do not support the following features:

- Auto-discovery of peers
- PW redundancy
- Static PW peers
- VC MAC withdraw
- Status TLV update
- VEoVPLS
- OAM
- Multicast snooping
- Extended counters
- High availability—Process restart
- High availability—ISSU

PW VLAN tag manipulation (vc-mode)

The virtual connection (VC) mode configuration for a pseudowire (PW) profile determines how VLAN tags are manipulated when a packet is received or transmitted on the PW.

The following table describes VC modes that are supported on PWs.

TABLE 3 VC modes supported on pseudowires

VC mode	Description
Raw	At VC label imposition, when a tagged packet is received on a tagged AC endpoint, the VLAN tag is removed before it is sent out on the wire. When an untagged packet is received on an untagged AC endpoint it is encapsulated as is and sent out on the wire.
Raw-passthrough	Enables interoperation with third-party devices. When a packet that is destined for a remote peer is received on either a tagged or untagged AC endpoint, it is encapsulated in an MPLS header and sent on to the MPLS cloud without adding or removing VLAN tags. When a packet that is destined for a local endpoint is received on either a tagged or untagged AC endpoint, the MPLS header is removed before sending it on to the local endpoint; VLAN tags in the original packet are not changed in any way.
Tagged	At VC label imposition, when a tagged packet is received on a tagged AC endpoint, the packet is encapsulated as is and sent out on the wire. This applies to dual-tagged and triple-tagged (or more) endpoints also; that is, tags are neither altered or removed but are sent to the remote peer. When an untagged packet is received on an untagged AC endpoint, a dummy tag is added and it is sent out on the wire.

The VC mode is agreed by PE peer devices during the pseudowire signaling process.

A single VPLS instance can have a mixture of tagged and untagged endpoints.

When the VC mode is changed on a device, the PWs are torn down are re-established except in the cases of a change from raw to raw-passthrough or a change from raw-passthrough to raw. The traffic impact is minimal (because PWs are not torn down and re-established) when the VC mode is changed from raw to raw-passthrough (or vice versa).

VC mode is configured by specifying the **vc-mode** option for the **pw-profile** command.

Supported VPLS features

The following VPLS features are supported:

- Local VPLS (without PWs)
- VPLS—untagged, single-tagged, and dual-tagged endpoints
- Flooding of L2 BPDUs in VPLS
- VPLS tagged, raw, and raw-passthrough modes for virtual circuits
- Dynamic LAG support for VPLS endpoints
- VPLS MTU enforcement
- VPLS static MAC address support for AC endpoints

Unsupported VPLS features

The following VPLS features are not supported:

- Virtual Ethernet (VE) over VPLS
- VPLS over Multi-Chassis Trunking (MCT)

Configuration of VPLS and VLL

Configuration of a VPLS or VLL instance includes configuring a bridge-domain, configuring a virtual connection (VC) identifier, configuring logical interfaces for attachment circuit (AC) endpoints, and configuring peer IP addresses.

To configure a VPLS or VLL instance, you must complete the following tasks:

- Configure a bridge domain.
- Configure a VC identifier.
- Configure logical interfaces for AC endpoints.
- (Optional) Configure a pseudowire (PW) profile.
- Configure peer IP addresses. Configuring peer IP addresses creates PW endpoints.

NOTE

VPLS (or VLL) configuration is separate from the underlying IP or MPLS configuration. MPLS tunnels need to be brought up separately. For further information about the configuration of MPLS tunnels, refer to the *Brocade SLX-OS Multi-Protocol Label Switching (MPLS) Configuration Guide* for the SLX 9850 Router.

QoS treatment in VPLS packet flow

There are default behaviors for Quality of Service (QoS) propagation in VPLS forwarding on PE routers.

On the ingress label-edge router (LER), the final EXP value for the VC label is not dependant on the CoS value in the VC-peer configuration.

By default, for traffic flowing from a CE device to a PE device, 3 bits of the PCP field from the incoming Ethernet frame header are extracted and mapped to an internal CoS value by way of an ingress CoS map. This internal value is then mapped to an outgoing CoS value by way of an egress CoS map. The outgoing CoS value is then inserted into the EXP field in the outgoing VC label. When incoming traffic does not have VLAN tag, the default PCP value that is configured on a port is used.

In the case of traffic received from the network core side, by default the EXP field from the incoming VC label is mapped to an internal CoS value by way of an ingress CoS map. This internal value is then mapped to an outgoing CoS value by way of an egress CoS map. The outgoing CoS value is then inserted into the PCP field in the Ethernet frame header going out to the CE device.

On the egress LER, the CoS value for the VC-peer configuration is not dependant on the final EXP value for the VC label.

The following table shows ingress and egress behavior for different global, tunnel and PW configuration combinations.

TABLE 4 Ingress and Egress LER behavior

Global	Tunnel	PW	AC to PW (per path)	PW to AC (per PW)
Uniform	No CoS	No CoS	Uniform	Uniform
Uniform	CoS	No CoS	Pipe	Uniform
Uniform	No CoS	CoS	Uniform	Pipe
Uniform	CoS	CoS	Pipe	Pipe
Pipe	No CoS	No CoS	Pipe	Pipe
Pipe	CoS	No CoS	Pipe	Pipe
Pipe	No CoS	CoS	Pipe	Pipe
Pipe	CoS	CoS	Pipe	Pipe

Configuring a PW profile

A pseudowire (PW) emulates a point-to-point connection over a packet-switching network. PW profile configuration defines PW attributes. After configuration, a PW profile must be attached to a bridge domain.

A pseudowire profile can be shared across multiple bridge domains. Complete the following task to configure a PW profile.

1. From privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create a PW profile and enter configuration mode for the profile.

   ```
   device(config)# pw-profile pw_example
   ```

3. Configure the virtual connection mode for the profile.

   ```
   device(config-pw-pw_example)# vc-mode tag
   ```

   In this example, tag mode is configured for the PW profile, pw_example.

4. Configure a maximum transmission unit (MTU) of 1300 for the PW profile.

   ```
   device(config-pw-pw_example)# mtu 1300
   ```

5. Enforce an MTU check during PW signaling.

   ```
   device(config-pw-pw_example)# mtu-enforce true
   ```

The following example creates a PW profile named pw_example and configures attributes for the profile.

```
device# configure terminal
device(config)# pw-profile pw_example
device(config-pw-pw_example)# vc-mode tag
device(config-pw-pw_example)# mtu 1300
device(config-pw-pw_example)# mtu-enforce true
```

Configuring a static MAC address over an endpoint in a VPLS instance

A static MAC address can be associated with the logical interface for an attachment circuit (AC) endpoint in a bridge domain.

You can configure a MAC address for a logical interface for an endpoint in a VPLS instance by completing the following task.

1. From privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. From privileged EXEC mode, enter global configuration mode.

   ```
   device(config)# mac-address-table static 0011.2345.6789 forward logical-interface ethernet 1/2.200
   ```

3. Enter privileged EXEC mode.

   ```
   device(config)# exit
   ```

4. (Optional) Verify the configuration.

   ```
   device# show mac-address-table static
   ```

Configuring a VPLS instance

A virtual private LAN Service (VPLS) instance provides multipoint LAN services.

Prior to completing the following task, the underlying L3 configuration of MPLS tunnels must be completed. There is a configuration example at the end of this task that shows all the steps in order.

You can configure a VPLS instance by completing the following task.

1. From privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create a multipoint bridge domain.

   ```
   device(config)# bridge-domain 5
   ```

   By default, the bridge-domain service type is multipoint. In this example, bridge domain 5 is configured as a multipoint service.

3. Configure a virtual connection identifier for the bridge domain.

   ```
   device(config-bridge-domain-5)# vc-id 8
   ```

4. **NOTE**
   Logical interfaces representing bridge-domain endpoints must be created before they can be bound to a bridge domain.

   Bind the logical interfaces for attachment circuit endpoints to the bridge domain.

   ```
   device(config-bridge-domain-5)# logical-interface ethernet 1/6.400
   ```

   In this example, Ethernet logical interface 1/6.400 is bound to bridge domain 5.

5. Repeat Step 4 to bind other logical interfaces for attachment circuit endpoints to the bridge domain.

   ```
   device(config-bridge-domain-5)# logical-interface port-channel 2.200
   ```

   In this example, port channel logical interface 2.200 is bound to bridge domain 5.
6. Configure peer IP addresses to create pseudowire (PW) endpoints.

   ```
   device(config-bridge-domain-5)# peer 10.15.15.15 load-balance
   ```

   In this example, a peer IP address of 10.15.15.15 is configured under bridge domain 5 and specifies load balancing.
7. Repeat Step 6 to configure more peer IP addresses to create PW endpoints.

   ```
   device(config-bridge-domain-5)# peer 10.12.12.12 lsp lsp1 lsp2
   ```

   In this example, a peer IP address of 10.12.12.12 under bridge domain 5 and specifies two label-switched paths (lsp1 and lsp2).
8. (Optional) Configure local switching for bridge domain 5.

   ```
   device(config-bridge-domain-5)# local-switching
   ```

9. (Optional) Enable dropping L2 bridge protocol data units (BPDUs) for bridge domain 5.

   ```
   device(config-bridge-domain-5)# bpdu-drop-enable
   ```

10. (Optional) Configure a PW profile under the bridge domain 5.

    ```
    device(config-bridge-domain-5)# pw-profile 2
    ```

The following example creates bridge domain 5 and configures virtual connection identifier 8 for the bridge domain. It binds ethernet and port-channel logical interfaces to the bridge domain and configures peer IP addresses under the domain. It configures local switching, enables dropping of L2 BPDUs, and configures a PW profile for the domain.

```
device# configure terminal
device(config)# bridge-domain 5
device(config-bridge-domain-5)# vc-id 8
device(config-bridge-domain-5)# logical-interface ethernet 1/6.400
device(config-bridge-domain-5)# logical-interface port-channel 2.200
device(config-bridge-domain-5)# peer 10.15.15.15 load-balance
device(config-bridge-domain-5)# peer 10.12.12.12 lsp lsp1 lsp2
device(config-bridge-domain-5)# local-switching
device(config-bridge-domain-5)# bpdu-drop-enable
device(config-bridge-domain-5)# pw-profile 2
```

Configuring a VLL instance

A virtual leased line (VLL) instance provides point-to-point (peer) LAN services.

Prior to completing the following task, the Ethernet logical interface and pseudowire profiles must be created. There is an example at the end of this task that shows all the steps in order.

You can configure a VLL instance by completing the following task.

1. From privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create a point-to-point bridge domain to use VLL services.

   ```
   device(config)# bridge-domain 3 p2p
   ```

   In this example, bridge domain 3 is created as a point-to-point service. By default, the bridge-domain service type is multipoint.

3. Configure a virtual connection identifier for the bridge domain.

   ```
   device(config-bridge-domain-3)# vc-id 500
   ```

4. Bind the logical interfaces for attachment circuit endpoints to the bridge domain.

   ```
   device(config-bridge-domain-3)# logical-interface ethernet 1/5.15
   ```

   In this example, the Ethernet logical interface 1/5.15 is bound to bridge domain 3.

5. Configure peer IP addresses to create pseudowire (PW) endpoints.

   ```
   device(config-bridge-domain-3)# peer 10.10.10.10
   ```

6. Configure a PW profile under the bridge domain.

   ```
   device(config-bridge-domain-3)# pw-profile to-mpls-nw
   ```

   The following example configures a PW profile 2 under bridge domain 3.

7. Repeat this configuration on the other peer device with appropriate parameters.

8. Enter Privileged EXEC mode.

   ```
   device(config-bridge-domain-3)# end
   ```

9. (Optional) Display information about the configured VLL instance.

   ```
   device# show bridge-domain 3

   Bridge-domain Type: P2P , VC-ID: 3
   Number of configured end-points:2 , Number of Active end-points: 2
   VE if-indx: 0, Local switching: FALSE, bpdu-drop-enable: FALSE
   PW-profile: default, mac-limit: 0
   VLAN: 3, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
   Tagged Ports: eth1/5.15
   Un-tagged Ports:
   Total VLL peers: 1 (1 Operational):
   VC id: 3, Peer address: 10.10.10.10, State: Operational, uptime: 18 sec
   Load-balance: True , Cos Enabled: False,
   Tunnel cnt: 4
   rsvp p105 (cos_enable:Falsecos_value:0)
   rsvp p106 (cos_enable:Falsecos_value:0)
   rsvp p107 (cos_enable:Falsecos_value:0)
   rsvp p108 (cos_enable:Falsecos_value:0)
   Assigned LSPs count:4 Assigned LSPs:p105 p106 p107 p108
   Local VC lbl: 851968, Remote VC lbl: 985331,
   Local VC MTU: 1600, Remote VC MTU: 1500,
   Local VC-Type: 5, Remote VC-Type: 5
   ```

The following example shows the creation of a logical interface and a pseudowire profile in addition to the bridge domain and VLL instance configuration.

```
device# configure terminal
device(config)# interface ethernet 1/5
device(conf-if-eth-1/5)# switchport
device(conf-if-eth-1/5)# switchport mode trunk
device(conf-if-eth-1/5)# switchport trunk tag native-vlan
device(conf-if-eth-1/5)# shutdown
device(conf-if-eth-1/5)# logical-interface ethernet 1/5.15
device(conf-if-eth-lif-1/5.15)# vlan 200
device(conf-if-eth-lif-1/5.15)# exit
device(conf-if-eth-1/5)# exit
device(config)# pw-profile to-mpls-nw
device(config-pw-profile-to-mpls-nw)# mtu 1600
device(config-pw-profile-to-mpls-nw)# mtu-enforce true vc-mode tag
device(config-pw-profile-to-mpls-nw)# exit
device(config)# bridge-domain 3 p2p
device(config-bridge-domain-3)# vc-id 500
device(config-bridge-domain-3)# logical-interface ethernet 1/5.15
device(config-bridge-domain-3)# peer 10.10.10.10
device(config-bridge-domain-5)# pw-profile to-mpls-nw
```

Displaying bridge-domain configuration information

Various show commands can be used to display bridge-domain configuration information.

- Enter the **show bridge-domain** command to display information about all configured bridge domains.

```
device# show bridge-domain

Total Number of bridge-domains: 3
Number of bridge-domains: 3

Bridge-domain 1
-------------------------------
Bridge-domain Type: mp , VC-ID: 5
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx: 1207959555, Local switching: TRUE, bpdu-drop-enable:TRUE
PW-profile: 1, mac-limit: 128000
Number of Mac’s learned:90000,     Static-mac count: 10,
VLAN: 100, Tagged ports: 2(2 up), Un-tagged ports: 0 (0 up)
Tagged ports: Eth 0/2/6, eth 0/2/8
Un-tagged ports:

Total PW peers: 2 (2 Operational)
Peer address: 12.12.12.12, State: Operational, Uptime: 2 hr 55 min
      Load-balance: True , Cos enabled:False,
      Assigned LSP;s:
      Tnnl in use: tnl2[RSVP]
      Local VC lbl: 983040, Remote VC lbl: 983040
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 15.15.15.15, State: Operational, Uptime: 2 hr 55 min
         Load-balance: False , Cos enabled:False,
      Assigned LSP’s: lsp1, lsp2
      Tnnl in use: tnl1[MPLS]
      Local VC lbl: 983041, Remote VC lbl: 983043
      Local VC MTU: 1500, Remote VC MTU: 1500 ,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)

Bridge-domain 2
-------------------------------
Bridge-domain Type: mp , VC-ID: 100
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx:  NA, Local switching: FALSE, bpdu-drop-enable:FALSE
PW-profile: profile_1, mac-limit: 262144
```

```
Number of Mac’s learned:90000,     Static-mac count: 10,
VLAN: 100, Tagged ports: 2(1 up), Un-tagged ports: 0 (0 up)
      Tagged ports: eth 0/2/10, eth 0/1/10
      Un-tagged ports:
VLAN: 150, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
      Tagged ports: eth 0/1/5
      Un-tagged ports:

Bridge-domain 3
-------------------------------
Bridge-domain Type: mp , VC-ID: 200
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx: 120793855, Local switching: FALSE, bpdu-drop-enable:FALSE
PW-profile: 2, mac-limit: 262144
Number of Mac’s learned:90000,     Static-mac count: 10,
Local switching: TRUE,
VLAN: 500, Tagged ports: 2(2 up), Un-tagged ports: 2 (1 up)
Tagged ports:     eth 0/11/6, eth 0/4/3
Un-tagged ports:

Total VPLS peers: 3 (2 Operational)
Peer address: 5.5.5.5, State: Operational, Uptime: 2 hr 35 min
         Load-balance: False , Cos enabled:False,
      Assigned LSP;s:
      Tnnl in use: tnl2[RSVP]
      Local VC lbl: 983050, Remote VC lbl: 983050
      Local VC MTU: 1500,Remote VC MTU: 1500,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 20.20.20.20, State: Operational, Uptime: 0 hr 18 min
         Load-balance: False , Cos enabled:True,
      Assigned LSP’s:
      Tnnl in use: NA,
      Local VC lbl: NA, Remote VC lbl: NA
      Local VC MTU: 1500,Remote VC MTU: 1500,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 10.10.10.10, State: Not-Operational (Tunnel Not Available),
         Load-balance: True , Cos enabled:False,
      Assigned LSP’s: lsp10, lsp15
      Tnnl in use: NA,
      Peer Index:2
      Local VC lbl: NA, Remote VC lbl: NA
      Local VC MTU: 1500,Remote VC MTU: NA ,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: NA
```

- Enter the **show bridge-domain** command specifying the bridge-domain ID to display information about a specific bridge domain. The following example displays information about bridge domain 501.

```
device# show bridge-domain 501

Bridge-domain 501
-------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth1/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 10.9.9.9, State: Operational, uptime: 2 sec
      Load-balance: False, Cos Enabled: False,
      Tunnel cnt: 1
      rsvp  p101(cos_enable:False cos_value:0)
      Assigned LSPs count:0 Assigned LSPs:
      Local VC lbl: 989042, Remote VC lbl: 983040,
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: 5, Remote VC-Type: 5
```

The following example shows information about a bridge domain (501) in which the **load-balance** option is configured for the peer device 10.9.9.9.

```
show bridge-domain 501

Bridge-domain 501
-------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth1/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 10.9.9.9, State: Operational, uptime: 48 sec
      Load-balance: True , Cos Enabled: False,
      Tunnel cnt: 16
      rsvp  p101(cos_enable:False cos_value:0)
      rsvp  p102(cos_enable:False cos_value:0)
      rsvp  p103(cos_enable:False cos_value:0)
      rsvp  p104(cos_enable:False cos_value:0)
      rsvp  p105(cos_enable:False cos_value:0)
      rsvp  p106(cos_enable:False cos_value:0)
      rsvp  p107(cos_enable:False cos_value:0)
      rsvp  p108(cos_enable:False cos_value:0)
      rsvp  p109(cos_enable:False cos_value:0)
      rsvp  p110(cos_enable:False cos_value:0)
      rsvp  p111(cos_enable:False cos_value:0)
      rsvp  p112(cos_enable:False cos_value:0)
      rsvp  p113(cos_enable:False cos_value:0)
      rsvp  p114(cos_enable:False cos_value:0)
      rsvp  p115(cos_enable:False cos_value:0)
      rsvp  p116(cos_enable:False cos_value:0)
      Assigned LSPs count:0 Assigned LSPs:
      Local VC lbl: 989040, Remote VC lbl: 983040,
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: 5, Remote VC-Type: 5
```

The following example shows information about bridge domain 501 in which the **load-balance** option and four assigned label-switched paths (p101, p102, p103, and p104) are configured for the peer device 10.9.9.9.

```
device# show bridge-domain 501

Bridge-domain 501
-------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth1/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 10.9.9.9, State: Operational, uptime: 4 sec
      Load-balance: True , Cos Enabled: False,
      Tunnel cnt: 4
      rsvp  p101(cos_enable:False cos_value:0)
      rsvp  p102(cos_enable:False cos_value:0)
      rsvp  p103(cos_enable:False cos_value:0)
      rsvp  p104(cos_enable:False cos_value:0)
      Assigned LSPs count:4 Assigned LSPs:p101 p102 p103 p104
      Local VC lbl: 989041, Remote VC lbl: 983040,
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: 5, Remote VC-Type: 5
```

- Enter the **show bridge-domain brief** command to display summary information about all configured bridge domains.

```
device# show bridge-domain brief

Total Number of bridge-domains configured: 10
Number of VPLS bridge-domains: 5
Macs Dynamically learned: 50360, Macs statically configured: 0

BDID(VC-ID)    TYPE     Intf(up)      PWs(up)   macs
501(501)       P2MP     5(3)          2(2)      50000
502(502)       P2MP     1(1)          1(1)      10
503(503)       P2MP     10(6)         3(1)      0
504(504)       P2MP     1(1)          1(1)      350
505(505)       P2MP     1(1)          1(1)      0
506(506)       P2P      1(1)          1(1)      0
507(507)       P2P      1(1)          1(1)      0
508(508)       P2P      1(1)          1(1)      0
509(509)       P2P      1(1)          1(1)      0
510(510)       P2P      1(1)          1(1)      0
```

Displaying MAC address information for VPLS bridge domains

Various show commands can be used to display MAC address information for bridge domains.

- Enter the **show mac-address-table bridge-domain** command to display information about MAC addresses in VPLS bridge domains. The following example shows details of all MAC addresses learned on all bridge domains.

```
device# show mac-address-table bridge-domain all

VlanId/BD-Id   Mac-address           Type     State     Ports/LIF/peer-ip
629(B)            0011.2222.5555     Dynamic  Active    eth 1/3.100
629(B)            0011.2222.6666     Dynamic  Inactive  eth 1/1.500
629(B)            0011.2222.1122     Dynamic  Active    10.12.12.12
629(B)            0011.2222.3333     static   Inactive  po 5.700
629(B)            0011.0101.5555     Dynamic  Active    eth 1/2.400

Total MAC addresses : 5
```

- Enter the **show mac-address-table bridge-domain** command specifying a bridge domain to display information about MAC addresses for a specific bridge domain.

```
device# show mac-address-table bridge-domain 1

BD-Name   Mac-address     Type     State     Ports/LIF/Peer-IP
1         0011.2222.5555  Dynamic  Active    eth 1/3.200
1         0011.2222.6666  Dynamic  Inactive  eth 2/2.500

Total MAC addresses : 2
```

Topology Groups

Topology Groups

A topology group is a named set of VLANs and bridge-domains that share a Layer 2 control protocol. Topology groups simplify configuration and enhance scalability of Layer 2 protocols by allowing you to run a single instance of a Layer 2 protocol on multiple VLANs and bridge-domains. One instance of the Layer 2 protocol controls all the VLANs and bridge-domains.

You can use topology groups with the following Layer 2 protocols:

- Per VLAN Spanning Tree (PVST+)
- Rapid per VLAN Spanning tree (R-PVST+)

Master VLAN, member VLANs, and bridge-domains

Each topology group contains a master VLAN and can contain one or more member VLANs and bridge-domains. A definition for each of these VLAN types follows:

- Master VLAN—The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for Rapid per VLAN Spanning tree (R-PVST), the topology group's master VLAN contains the R-PVST configuration information.
- Member VLANs—The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the member VLANs. Member VLANs do not independently run a Layer 2 protocol.
- Member bridge domains—The member bridge domains are similar to VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the bridge domains. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the bridge domains. Bridge domains do not independently run a Layer 2 protocol. In a bridge domain, a single port can have multiple logical interfaces. In this scenario, all the logical interfaces on that port (and bridge domain) will follow the state of master VLAN port.

When a Layer 2 topology change occurs, resulting in a change of port state in the master VLAN, the same port state is applied to all the member VLANs and bridge-domains belonging to the topology group on that port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs and bridge-domains that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Control ports and free ports

A port in a topology group can be a control port or a free port:

- A **control port** is a port in the master VLAN and, therefore, is controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs and bridge-domains is controlled by the master VLAN's Layer 2 protocol. Each member VLAN and bridge-domain must contain all of the control ports. All other ports in the member VLAN and bridge-domain are "free ports."
- **Free ports** are not controlled by the master VLAN's Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs and bridge-domains that are not also in the master VLAN are free ports.

  NOTE
  Because free ports are not controlled by the master port's Layer 2 protocol, they are always in the forwarding state.

Configuration considerations

The configuration considerations are as follows:

- The topology group must contain a master VLAN. The group can also contain individual member VLANs and or member bridge-domains. You must configure the member VLANs or member bridge-domains before adding them to the topology group. Bridge-domains cannot be configured as a master VLAN.
- You cannot delete a master VLAN from the topology group when the member VLANs or bridge-domains are in the topology group.
- The control port membership must match the master VLAN when adding a member VLAN or member bridge-domain.
- If a VLAN enabled with the PVST+ or R-PVST+ protocol is added as a member VLAN of a topology group, the protocol is disabled. The member VLAN is added to the topology group. If the VLAN is removed from the topology group, the protocol is disabled, and you must enable the protocol if required.
- Enabling STP on an interface is only allowed if both master VLAN and member VLAN or bridge-domains are configured on the interface across all topology groups.
- You cannot remove the master VLAN or member VLAN or bridge-domains from an STP enabled interface.
- Topology group configuration is allowed only with PVST+ and R-PVST+ spanning tree configurations.

Configuring a topology group

Follow this procedure to configure a topology group. Brocade SLX devices support creating 128 topology groups in a system.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **topology-group** command to create a topology group at the global configuration level.

   ```
   device(config)# topology-group 1
   device(conf-topo-group-1)#
   ```

   NOTE
   The **no topology-group** command deletes an existing topology group.

Configuring a master VLAN

Follow this procedure to configure a master VLAN in a topology group.

Before configuring a master VLAN, you should have configured a topology group.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **topology-group** command to create a topology group at the global configuration level.

   ```
   device(config)# topology-group 1
   device(conf-topo-group-1)#
   ```

3. Enter the **master-vlan** command to configure a master VLAN in the topology group.

   ```
   device(conf-topo-group-1)# master-vlan 100
   ```

   NOTE
   The **no master-vlan** command removes an existing master VLAN from the topology group.

Adding member VLANs

Follow this procedure to add member VLANs to a topology group. Member VLANs follow the master VLAN protocol states and also no L2 protocol will be running on the member VLANs.

Before adding a member VLAN, you should have created a topology group and configured the master VLAN for that group. The VLAN should not be part of any other topology group. All control ports of master VLAN must also be configured for the member VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **topology-group** command to create a topology group at the global configuration level.

   ```
   device(config)# topology-group 1
   device(conf-topo-group-1)#
   ```

3. Enter the **master-vlan** command to configure a master VLAN in the topology group.

   ```
   device(conf-topo-group-1)# master-vlan 100
   ```

4. Enter the **member-vlan** command to add member VLANs to the topology group.

   ```
   device(conf-topo-group-1)# member-vlan add 200-201
   ```

   NOTE
   The **member-vlan remove** command removes an existing member VLAN from the topology group.

   ```
   device(conf-topo-group-1)# member-vlan remove 200
   ```

Adding member bridge-domains

Follow this procedure to add member bridge domains to a topology group. Member bridge domains follow the master VLAN protocol states and also no L2 protocol will be running on the bridge domains.

Before adding a bridge domain, you should have created a topology group and configured the master VLAN for that group. The bridge-domain should not be part of any other topology group. All control ports of master VLAN must also be configured for the member bridge-domain.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   device(config)#
   ```

2. Enter the **topology-group** command to create a topology group at the global configuration level.

   ```
   device(config)# topology-group 1
   device(conf-topo-group-1)#
   ```

3. Enter the **master-vlan** command to configure a master VLAN in the topology group.

   ```
   device(conf-topo-group-1)# master-vlan 100
   ```

4. Enter the **member-bridge-domain** command to add member bridge-domains to the topology group.

   ```
   device(conf-topo-group-1)# member-bridge-domain add 300
   ```

   NOTE
   The **member-bridge-domain remove** command removes an existing member bridge-domain from the topology group.

   ```
   device(conf-topo-group-1)# member-bridge-domain remove 1
   ```

The example adds 300 as member bridge-domain to the topology group.

```
device# configure terminal
device(config)# topology-group 1
device(conf-topo-group-1)# master-vlan 100
device(conf-topo-group-1)# member-bridge-domain add 300
```

Replacing a master VLAN

For replacing the existing master VLAN of a topology group, use the **master-vlan** command with the new master VLAN.

To avoid temporary loops when the master VLAN is replaced by another VLAN, the following recommendation is made:

- Control ports for both the old and the new master VLAN must match.
- The new master VLAN and the old master VLAN must have same ports in the blocking state to avoid the possibility of temporary loops.

If the recommendation is not followed, and a new master VLAN is configured with a different convergence, the configuration is still accepted.

NOTE
The master VLAN replacement is accepted if both the old and the new master VLANs are spanning-tree disabled.

Displaying topology group information

Follow the procedure to display topology group information for a specified group.

Before displaying the topology group information, you should have configured a topology group and defined the master VLAN.

Enter the **show topology-group** command to display the group information.

```
device# show topology-group 1
Topology Group 1
==============================
Master VLAN : 100
L2 Protocol: R-PVST
Member VLANs : 200 300
Member Bridge-domains: 10
Control Ports : eth 2/1, eth 2/2, po10
Free Ports : VLAN: 200 –eth 2/3, po11
Bridge-domain: 10 –eth 2/3.20, po11.10
```

The example displays information about topology group 1.

The **show running-config** command displays topology group configurations.

```
device# show running-config
topology-group 1
   master-vlan 100
   member-vlan add 200 300
   member-bridge-domain add 10
```

Unidirectional Link Detection

Unidirectional Link Detection overview

Unidirectional Link Detection (UDLD) monitors a link between two Brocade devices and blocks the ports on both ends of the link if there is a unidirectional failure.

UDLD protocol detects and blocks broken unidirectional links in the network. This is done through the exchange of UDLD protocol data units (PDU) between devices on a physical link. Both ends of the link must support the same proprietary UDLD protocol to detect the unidirectional link condition.

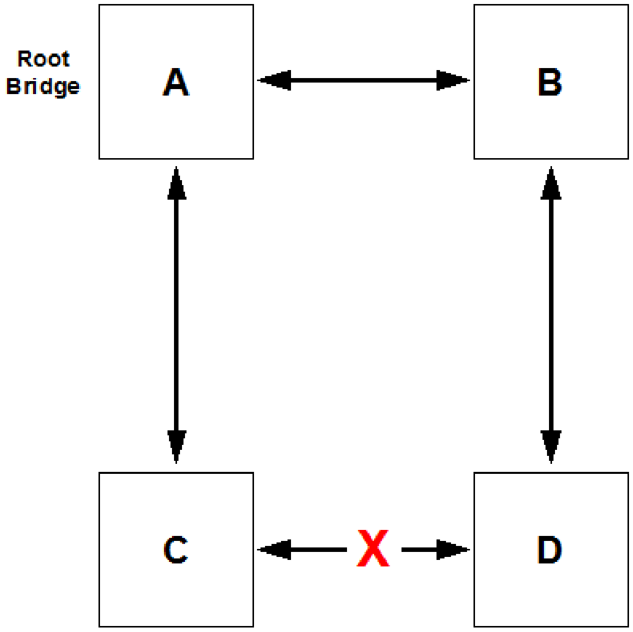
A unidirectional link is assumed when the UDLD stops receiving UDLD PDUs from the other end of the link. The device then blocks the physical link. The physical link will still be up but the line protocol will be down. UDLD PDUs continue to be transmitted and received on the link.

UDLD is disabled by default. To use the UDLD protocol, the protocol must first be enabled globally and then on each individual physical port. When enabled globally and on a physical port, the device starts transmitting UDLD PDUs periodically on the port.

How UDLD works

The following shows a simple four-switch network in which two paths connect to each switch. STP blocks traffic on as many ports as necessary so that only one operational path exists from the STP root bridge to all nodes in the network.

FIGURE 7 Four-switch example for UDLD



In the figure above, STP detects that the port on Switch D that is connected to Switch C should be put into a blocked state. Therefore, no data traffic gets transmitted or received on this port. Data traffic remains blocked as long as Switch D receives bridge protocol data units (BPDUs) from both switches C and B.

If the link between Switch C and Switch D becomes unidirectional (for reasons such as hardware failure or incorrect cabling) in the direction from D to C, Switch D ages out the status that it was receiving BPDUs from Switch C. This eventually causes STP to put the port in a forwarding state, thus allowing all data traffic. This creates a loop for all BUM traffic that enters the network. BUM traffic can go from Switch B to Switch D to Switch C to Switch A, and then back to Switch B.

To prevent this loop from forming, UDLD can be used to detect that the link between Switch C and Switch D has become unidirectional.

UDLD considerations and restrictions

Note the following for UDLD:

- UDLD is used in conjunction with the Spanning Tree Protocol.
- UDLD runs only on physical ports assigned to a port channel.
- UDLD is supported on directly connected switches only.
- The protocol must be running on both ends of the link.
- The default timeout is 2.5 (2500 ms) seconds.
- The feature is not compatible with Cisco UDLD protocol.
- Brocade is a proprietary implementation that interoperates with Multi-Service IronWare devices, FastIron devices, and VDX devices.
- The UDLD interface statistics lose some accuracy after a failover.
- Upon executing the **no protocol udld** command, all of the UDLD global and UDLD interface configuration changes will be removed and the protocol will revert back to its initial disabled state. This means that all interfaces that have been enabled for the protocol stops transmitting and receiving UDLD PDUs.

Configuring UDLD

Follow the steps below to configure basic UDLD on your device.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable the UDLD protocol and enter protocol UDLD configuration mode.

   ```
   device(config)# protocol udld
   ```

3. Change the hello interval to 2000 milliseconds.

   ```
   device(config-udld)# hello 20
   ```

   This changes the interval at which UDLD PDUs are transmitted. The default interval, in counts of one hundred milliseconds is 500 ms.

4. Change the timeout multiplier from the default of 5.

   ```
   device(config-udld)# multiplier 8
   ```

   This changes the timeout multiplier value to affect the UDLD PDU timeout interval. The UDLD timeout interval is the product of the hello time interval at the other end of the link and the timeout multiplier value.

   When the remote is Multi-Service IronWare or FastIron devices, the timeout equals local hello interval × multiplier value.

5. Return to global configuration mode.

   ```
   device(config-udld)# exit
   ```

6. Enter interface configuration mode for an port.

   ```
   device(config)# interface ethernet 5/1
   ```

7. Enable UDLD on the interface.

   ```
   device(config-if-eth-5/1)# udld enable
   ```

8. Repeat the preceding step for each port on which you wish to enable UDLD.

   NOTE
   When the UDLD protocol is enabled on one end of a link, the timeout period might elapse before the UDLD protocol is enabled on the other end of the link. In this case, the link becomes temporarily blocked. When the UDLD protocol is enabled at the other end of the link and a UDLD PDU is received, UDLD automatically unblocks the link.

9. Return to privileged exec mode.

   ```
   device(config-if-eth-5/1)# end
   ```

10. Verify the configuration.

    - Show the UDLD global configuration.

```
device# show udld
UDLD Global Information
  Admin State:       UDLD enabled
  UDLD hello time:   1000 milliseconds
  UDLD timeout:      5000 milliseconds
```

    - Show UDLD status for an interface

```
device# show udld interface ethernet 4/3
Global Admin State:  UDLD enabled

UDLD information for Ethernet 4/3
  UDLD Admin State:                  Enabled
  Interface Operational State:       Bidirectional link
  Remote hello time:                 500 milliseconds
  Local system id:  0x1ecd0fea       Remote system id: 0x1ecd0bea
  Local port :      4/3              Remote port :     2/7
  Local link id:    0x0              Remote link id:   0x0
  Last Xmt Seq Num: 127              Last Rcv Seq Num: 39
```

    - Show UDLD statistics.

```
device# show udld statistics
UDLD Interface statistics for Ethernet 2/7
Frames transmitted:         260
Frames received:            223
Frames discarded:           0
Frames with error:          0
Remote port id changed:     0
Remote MAC address changed: 0
```

    NOTE

    The **show interface** command also indicates whether UDLD is enabled.

11. Save the configuration.

```
device# copy running-config startup-config
```

UDLD configuration example

```
device# configure terminal
device(config)# protocol udld
device(config-udld)# hello 20
device(config-udld)# multiplier 8
device(config-udld)# exit
device(config)# interface ethernet 5/1
device(config-if-eth-5/1)# udld enable
device(config-if-eth-5/1)# end
device# show udld
device# copy running-config startup-config
```

VXLAN visibility

VXLAN visibility

The virtual extensible LAN (VXLAN) visibility feature is used in the transit network devices. In general, a transit device routes the traffic based on the outer destination virtual tunnel endpoints (VTEP) IP address. However, VXLAN visibility feature provides a mechanism for deep packet inspection and classifies the packets on the outer L3 header and the VXLAN header and also on the native inner L3 and L4 header.

Brocade's VXLAN visibility has overlay access-control lists (ACLs) of type VXLAN. It is a collection of filters that defines what action to take on the packets which matches the configured parameter in the filter. VXLAN visibility overlay access lists define filters with parameters that match the outer Layer 3 and Layer 4, VXLAN header and native inner Layer 3 and Layer-4 fields of a packet.

In addition, a VXLAN overlay access control list accepts breakout port as a mirror port or redirect port. You can remove VXLAN visibility rules before changing the breakout port configuration for a port that being used as a mirror port or redirect port in the VXLAN rule.

Overlay access list

An overlay access list is a type of access list used in overlay technologies. Currently, Brocade supports overlay access list of type VXLAN transit.

An overlay access list can be associated to the overlay transit.

Type of overlay access lists

There are two types of overlay access lists such as standard overlay access list and extended overlay access list.

The standard overlay access list is available as part of the ternary content-addressable memory (TCAM) profile. Overlay access lists of this type have limited qualifiers and action.

The extended overlay access list is available as part of the ternary content-addressable memory (TCAM) profile. Overlay access lists of this type have extended qualifiers and action.

Limitations and restrictions

The VXLAN visibility has the following limitation and restrictions.

- Only the overlay access control list of type VXLAN transit is supported.

- The rACL and Openflow are not supported as part of VXLAN visibility extended profile.
- Rule update of the standard or extended mode is not supported.

Creating an overlay access list

1. From the privileged EXEC mode, enter the global configuration mode.

   ```
   device# configure terminal
   ```

2. Use the **overlay access-list type vxlan** command to create an overlay access list.

   Enter the parameter **extended** for creating an overlay access list of type extended or **standard** for creating an overlay access list of type standard.

   ```
   device(config)# overlay access-list type vxlan extended abc_ext
   2016/08/15-23:29:09, [SSMD-1400], 4282, M1 | Active | DCE, INFO, SLX, Overlay access list abc_ext is
   created
   ```

3. Use the **seq** command to insert filtering rules in the overlay access list.

   ```
   device(config-overlay-vxlan-ext-vlx_al)# seq 12 permit dst-vtep-ip-host 10.10.10.1 src-vtep-ip-host
   20.20.20.1 vni-any sflow count
   2016/08/15-23:29:43, [SSMD-1404], 4283, M1 | Active | DCE, INFO, SLX, Overlay access list abc_ext
   rule sequence number 12 is added.
   ```

Binding overlay access list

You must first create an overlay access list that you want to bind to an overlay transit.

1. Use the **overlay-transit** command to create an overlay transit.

   ```
   device(config)# overlay-transit abc_ext
   ```

2. Use the **overlay access-group** command to bind an overlay access list to an overlay transit.

   ```
   device(config-overlay-transit-vxlan1)# overlay access-group abc_ext in
   2016/08/15-23:30:00, [SSMD-1405], 4284, M1 | Active | DCE, INFO, SLX, Overlay access list abc_ext
   configured on interface Global at Ingress by VXLAN VISIBILITY.
   ```

   Only one overlay access list can be bound to an overlay transit.

3. (Optional) Use the **no overlay overlay access-group** command to unbind an overlay access list.

```
device(config-overlay-transit-vxlan1)# no overlay access-group vlx_al in
```

Displaying overlay access list information

Use the following show commands to display the configuration, binding status and statistics pertaining to overlay access list.

1. From the privileged EXEC mode, use the **show access-list overlay transit** command to display which overlay access list is bound with overlay transit.

   ```
   device# show access-list overlay transit tr_name
   Overlay Transit Global Binding
     Inbound access-list is abc_ext (From User)
     Outbound access-list is not set
   ```

2. Use the **show access-list overlay type vxlan** command to display status of individual filters and binding information of the overlay access list.

   ```
   device# show access-list overlay type vxlan acl-name abc_ext
   Number of Rules: 4
   seq 1000 permit  dst-vtep-ip-host 200.1.1.1 src-vtep-ip-host 150.1.1.1 vni 1 vni-mask 0 redirect
   Ethernet 2/65 sflow count 44024774(pkts)/52829728800(bytes)
   seq 1010 permit  dst-vtep-ip-host 200.1.1.2 src-vtep-ip-host 150.1.1.2 vni 2 vni-mask 0 redirect
   Ethernet 2/19 sflow count 44024773(pkts)/52829727600(bytes)
   seq 1020 permit  dst-vtep-ip-host 200.1.1.3 src-vtep-ip-host 150.1.1.3 vni 3 vni-mask 0 redirect
   Ethernet 2/43 sflow count 0(pkts)/0(bytes)
   seq 1030 permit  dst-vtep-ip-host 200.1.1.4 src-vtep-ip-host 150.1.1.4 vni 4 vni-mask 0 redirect
   Ethernet 2/67 sflow count 0(pkts)/0(bytes)
   Transit : transit_name
   ```

3. Use the **show statistics access-listcoverlay type vxlan** command to display statistics for specific an overlay access list.

   ```
   device# show statistics access-list overlay type vxlan abc_ext
   Number of Rules: 2
   seq 1000 permit  dst-vtep-ip-host 200.1.1.1 src-vtep-ip-host 150.1.1.1 vni 1 vni-mask 0 redirect
   Ethernet 2/65 sflow count 0(pkts)/0(bytes)
   seq 1010 permit  dst-vtep-ip-host 200.1.1.2 src-vtep-ip-host 150.1.1.2 vni 2 vni-mask 0 redirect
   Ethernet 2/19 sflow count 44024773(pkts)/52829727600(bytes)
   ```

4. Use the **show running-config overlay access-list type vxlan** command to display the overlay access list configuration.

   ```
   device# show running-config overlay access-list type vxlan
   overlay access-list type vxlan extended abc_ext
    seq 12 permit dst-vtep-ip-host 12.12.1.1 src-vtep-ip-host 33.4.5.6 vni-any count sflow native tag
   none dst-ip-any src-ip-any dst-port-any src-port-any
    seq 123 permit dst-vtep-ip-any src-vtep-ip-any vni-any count!
   ```

Clearing overlay access list statistics

Use the **clear counters access-list overlay type vxlan** command to remove statistics pertaining to a specific overlay access list.

```
device# clear counters access-list overlay type vxlan abc_ext
```