Troubleshoot Spanning Tree PVID- and Type-Inconsistencies

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

Theory Behind PVID- and Type-Inconsistencies

Troubleshoot

Related Information

Introduction

This document describes how to troubleshoot two Spanning Tree Protocol (STP) inconsistencies, Port VLAN ID (PVID) and Type.

Prerequisites

Requirements

Cisco recommends that you have knowledge of STP concepts.

Components Used

This document is not restricted to specific software or hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

In Layer 2 (L2) networks, there can be only one path between any two devices. Redundancy is supported with Spanning-Tree Protocol (STP), which detects and blocks redundant paths and thereby avoids forwarding loops. Certain misconfigurations can lead to STP failure and cause network outage. To prevent downtime, some enhancements were implemented so that STP detects certain cases of misconfiguration, and the relevant port is put into an inconsistent state.

There can be different types of STP inconsistencies:

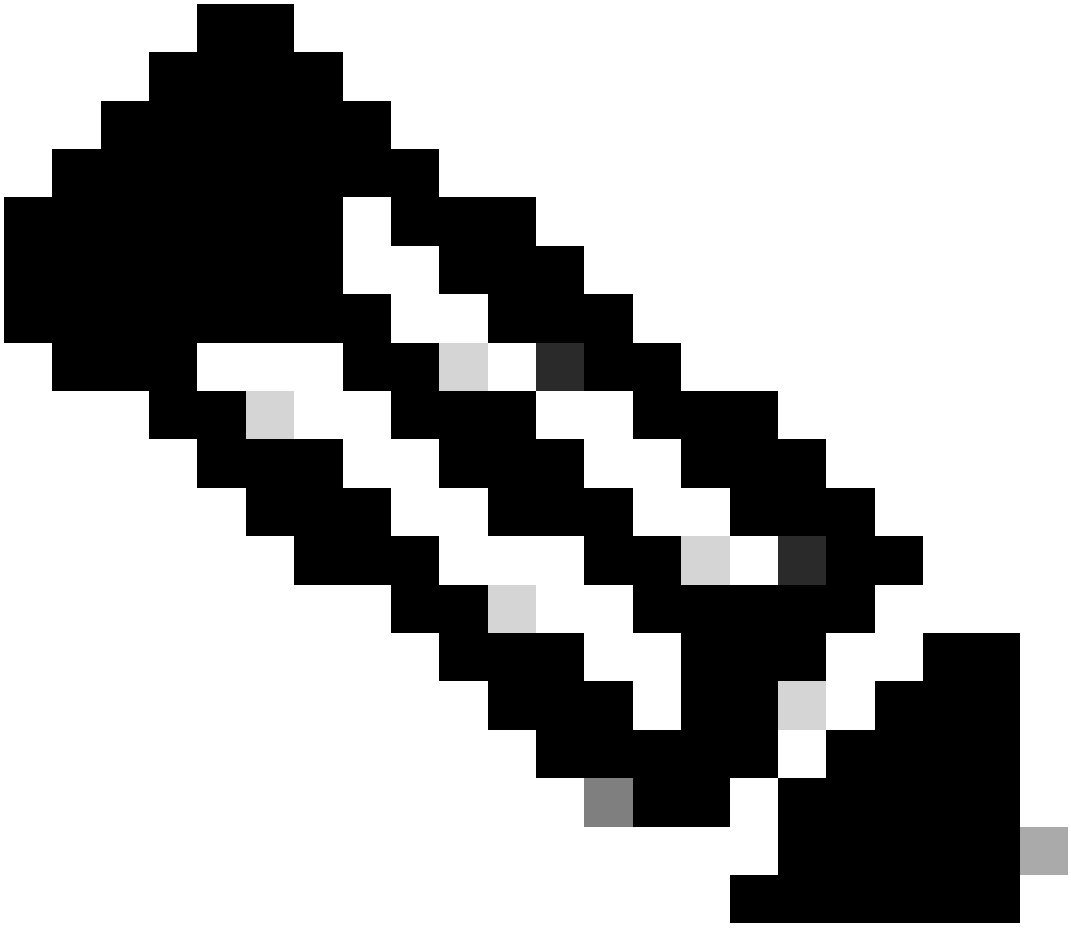
- Loop inconsistency—This is detected by the Loop Guard feature. For more information, refer to Configure STP with Loop Guard and BPDU Skew Detection.
- Root inconsistency—This is detected by the Root Guard feature. For more information, refer to

Enhance Spanning Tree Protocol with Root Guard.

- EtherChannel inconsistency—This is detected by the EtherChannel consistency detection feature. For more information, refer to Understanding EtherChannel Inconsistency Detection.
- Port VLAN ID (PVID) inconsistency—A per-VLAN spanning tree (PVST+) Bridge Protocol Data Unit (BPDU) is received on a different VLAN than it was originated: (`Port VLAN ID Mismatch` or `*PVID_Inc`).
- Type inconsistency—A PVST+ BPDU is received on a non-802.1Q trunk.

Theory Behind PVID- and Type-Inconsistencies

Cisco Catalyst switches implement PVST that use Inter-Switch Link (ISL) trunks. With the support of IEEE 802.1Q and ISL trunking, a way was needed for interoperation between PVST and the IEEE 802.1Q concept of a single spanning tree for all VLANs. The PVST+ feature was introduced to address this requirement.



Note: From the STP point of view, IEEE 802.1D is not VLAN-aware and IEEE 802.1Q is VLAN-aware, but it uses a single STP instance for all VLANs. That is, if the port is blocking then it is blocking for all VLANs on that port.

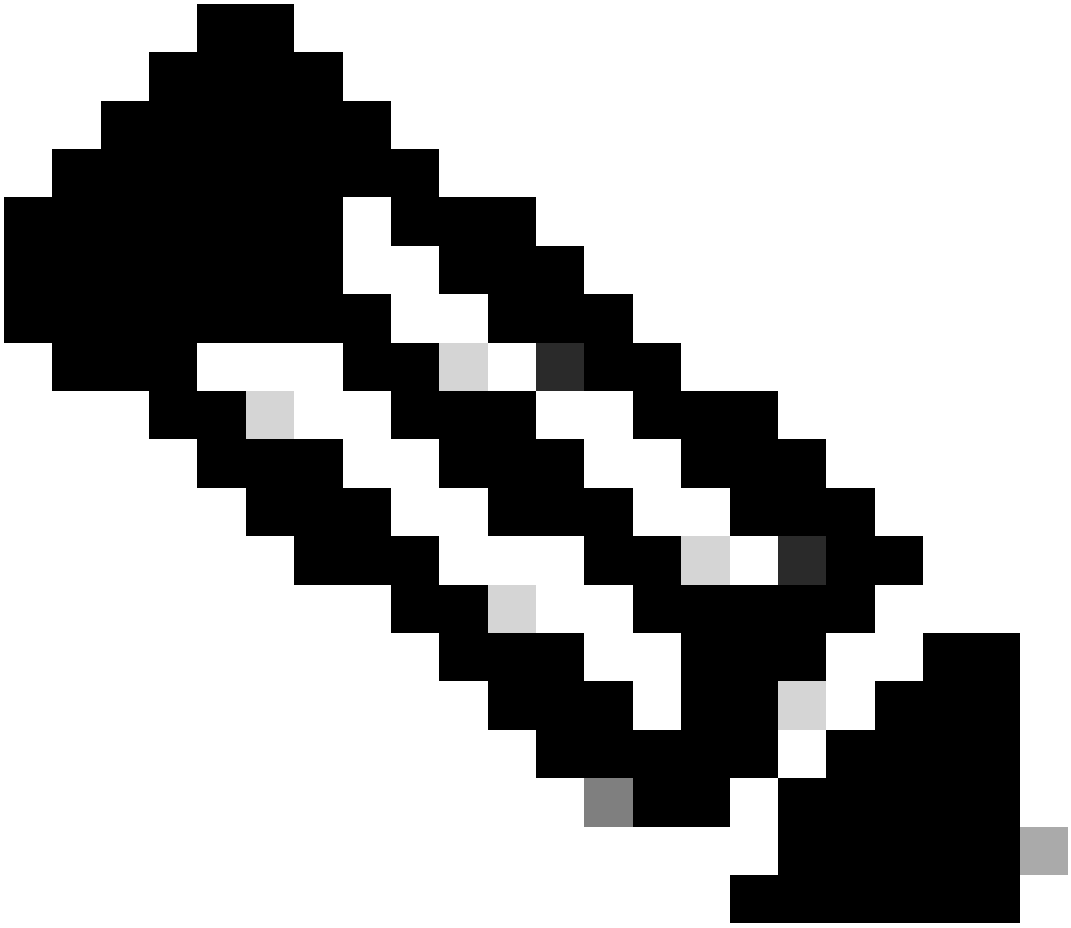
The same is true for forwarding.

This list shows how PVST+ interoperates with IEEE 802.1Q or IEEE 802.1D, if the Native VLAN on an IEEE 802.1Q trunk is VLAN 1:

- VLAN 1 STP BPDUs are sent to the IEEE STP MAC address (0180.c200.0000), untagged.
- VLAN 1 STP BPDUs are also sent to the PVST+ MAC address, untagged.
- Non-VLAN 1 STP BPDUs are sent to the PVST+ MAC address (also called the Shared Spanning Tree Protocol (SSTP) MAC address, 0100.0ccc.cccd), tagged with a corresponding IEEE 802.1Q VLAN tag.

If the Native VLAN on an IEEE 802.1Q trunk is not VLAN 1:

- VLAN 1 STP BPDUs are sent to the PVST+ MAC address, tagged with a corresponding IEEE 802.1Q VLAN tag.
- VLAN 1 STP BPDUs are also sent to the IEEE STP MAC address on the Native VLAN of the IEEE 802.1Q trunk, untagged.
- Non-VLAN 1 STP BPDUs are sent to the PVST+ MAC address, tagged with a corresponding IEEE 802.1Q VLAN tag.



Note: Native VLAN STP BPDUs are sent untagged.

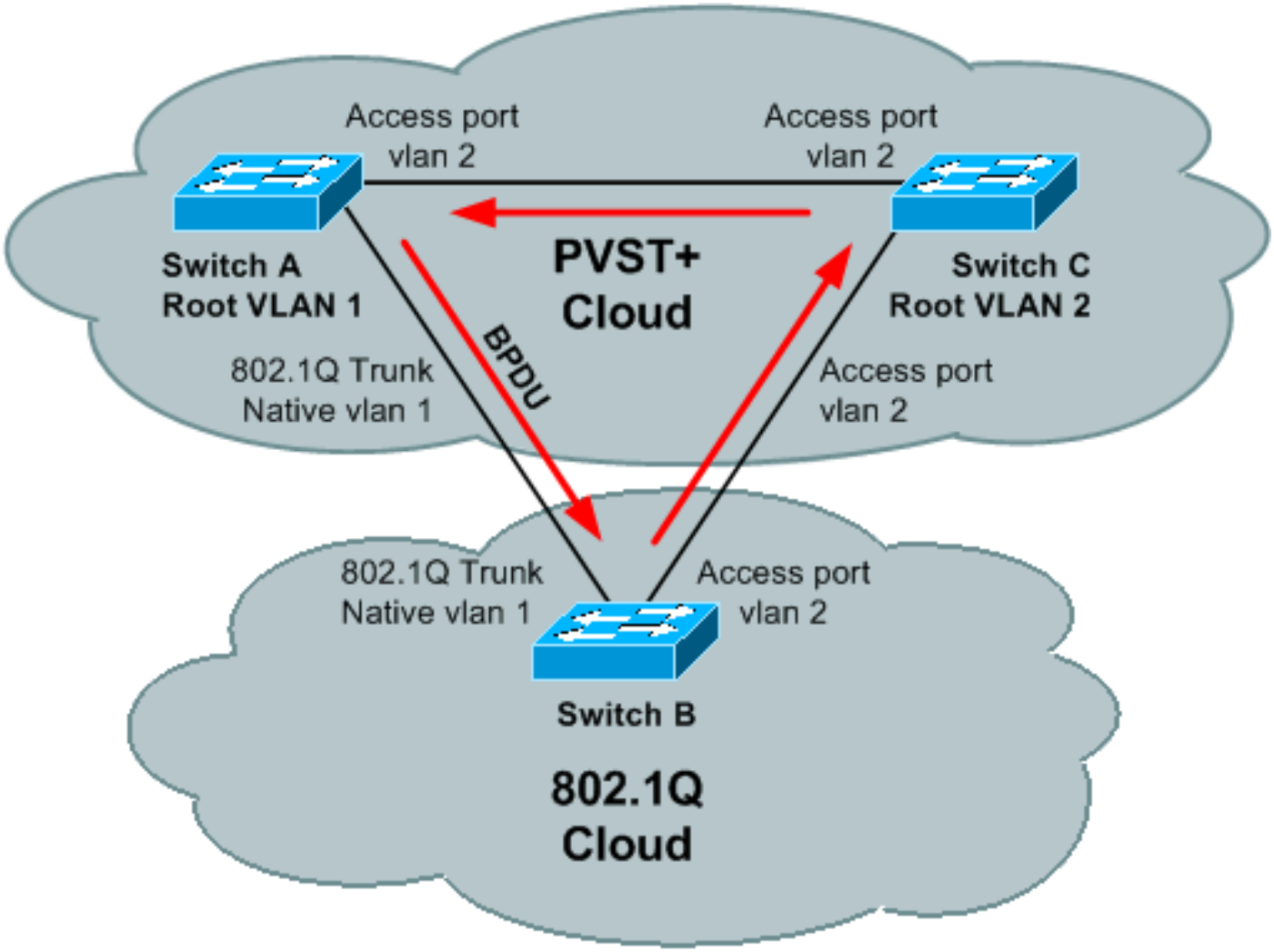
This way, VLAN 1 STP of PVST+ merges with STP of IEEE 802.1D or 802.1Q, while other VLANs are tunneled through the cloud of IEEE 802.1D or 802.1Q bridges.

For example, the IEEE 802.1D or 802.1Q cloud looks similar to a wire to the PVST+ VLANs other than 1.

For STP to operate correctly, observe certain rules when you connect PVST+ bridges to IEEE 802.1D or 802.1Q bridges. The main rule is that PVST+ bridges must connect to IEEE 802.1D or 802.1Q bridges through an IEEE 802.1Q trunk with a consistent Native VLAN on all bridges that connect to the cloud of IEEE 802.1Q or 802.1D bridges.

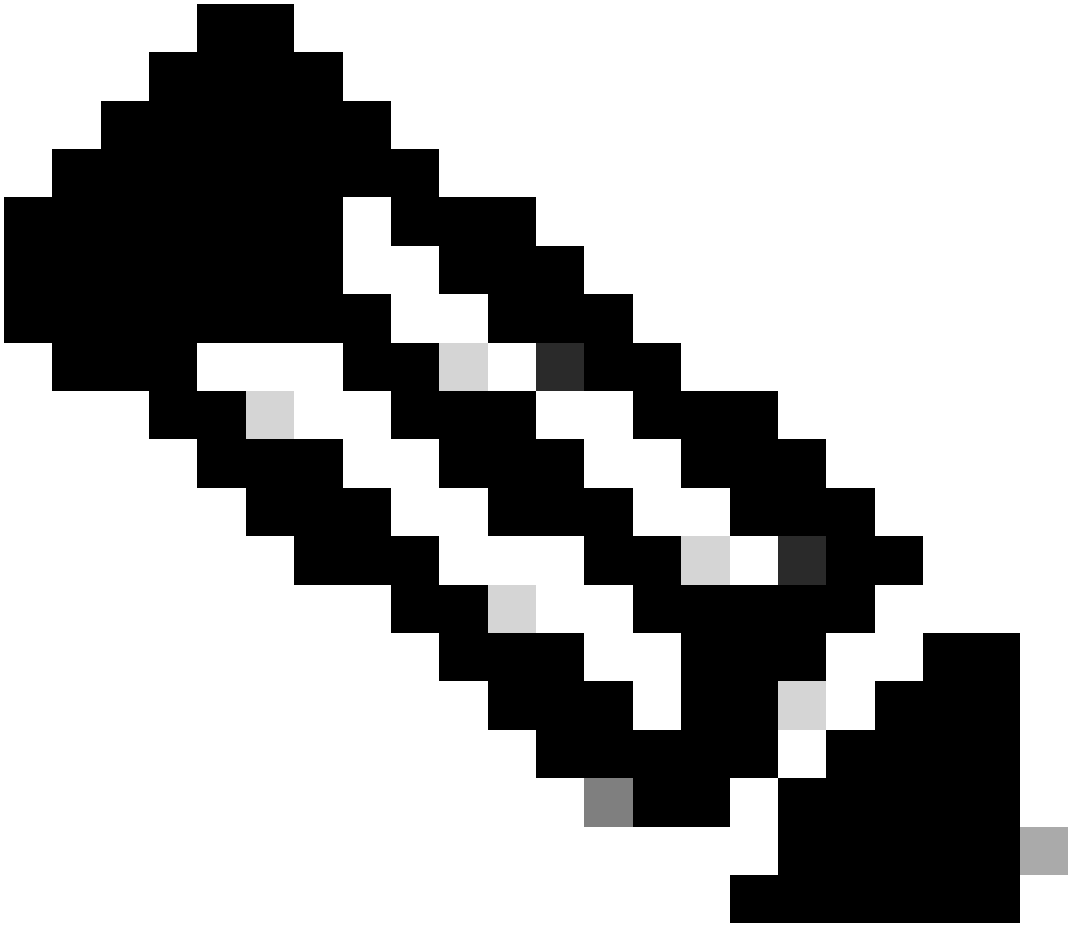
The PVST+ BPDU contains a VLAN number that allows PVST+ bridges to detect whether the previous rule is not respected. When a Catalyst switch detects a misconfiguration, the corresponding ports are put into a PVID-inconsistent or type-inconsistent state, which effectively blocks the traffic in the corresponding VLAN on a corresponding port. These states prevent forwarding loops that are caused by misconfiuration or were miswired.

To illustrate the need for inconsistency detection, consider this topology, where switches A and C run PVST+ STP and switch B runs 802.1Q STP:



If the BPDU of the root in VLAN 1 is better than the BPDU of the root in VLAN 2 then there is no blocking port in the VLAN 2 topology. The BPDU of VLAN 2 never makes a full circle around the topology; it is replaced by the VLAN 1 BPDU on the B-C link, because B runs only one STP merged with VLAN 1 STP of PVST+.

Thus, there is a forwarding loop. Fortunately, switch A sends PVST+ BPDUs of VLAN 2 (to the SSTP address that is flooded by switch B) towards switch C. Switch C can put port C-B into a type-inconsistent state, which prevents the loop.



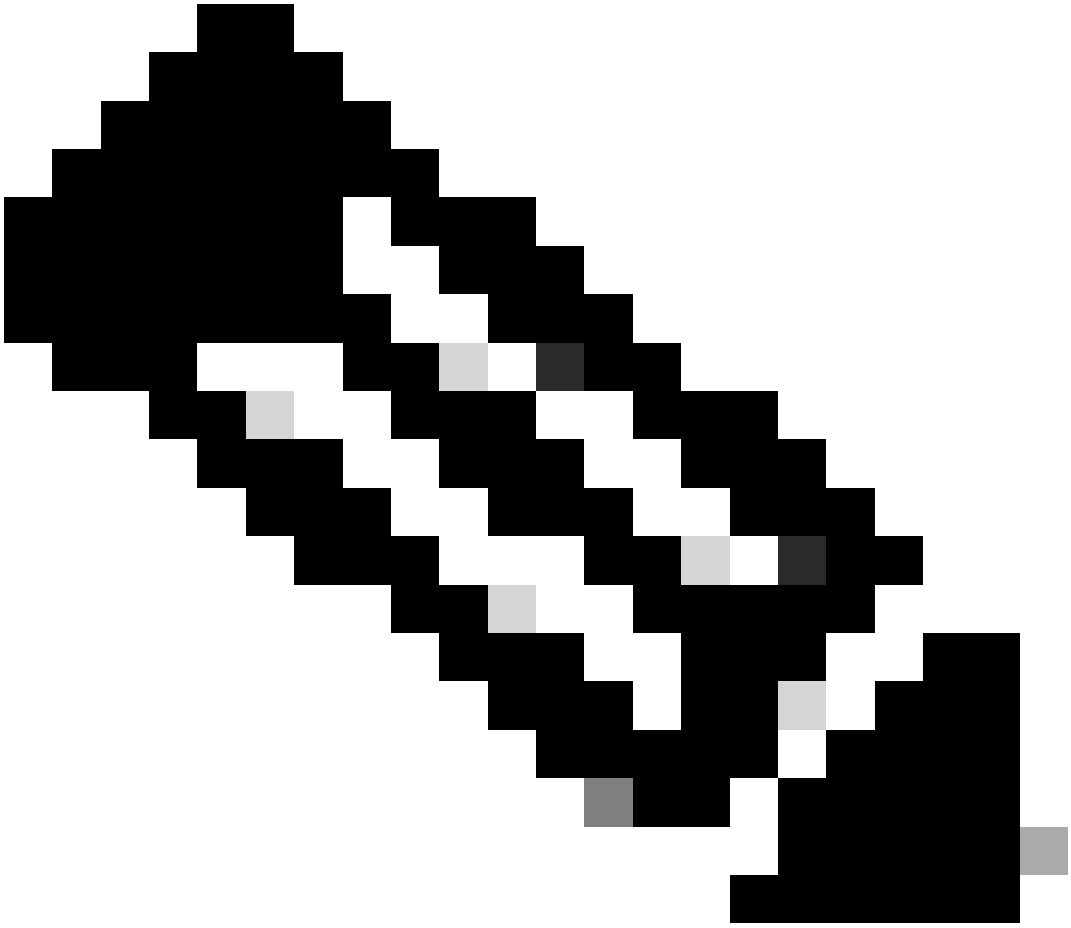
Note: In some command outputs, *-inconsistent STP state is referred to as broken.

When STP inconsistency is detected, switches send these syslog messages:

```
%SPANTREE-2-RECV_1Q_NON_TRUNK: Received IEEE 802.1Q BPDU on non trunk
FastEthernet0/1 on vlan 1.
%SPANTREE-2-BLOCK_PORT_TYPE: Blocking FastEthernet0/1 on vlan 1.
Inconsistent port type.

%SPANTREE-2-RX_1QPVIDERR: Rcved pvid_inc BPDU on 1Q port 3/25 vlan 1
%SPANTREE-2-RX_BLKPORTPVID: Block 3/25 on rcving vlan 1 for inc peer vlan 10
%SPANTREE-2-TX_BLKPORTPVID: Block 3/25 on xmtting vlan 10 for inc peer vlan
```

In that example, VLAN 1 is where the BPDU was received, and VLAN 10 is where the BPDU was originated. When inconsistency is detected, both VLANs are blocked on the port where this BPDU is received.



Note: Messages can vary based on the type and version of the Cisco IOS® Software Release that is in use.

Notice, if the port no longer receives inconsistent BPDUs, the *-inconsistent state is cleared and STP changes the port state based on normal STP operation. A syslog message is sent to indicate the change:

```
%SPANTREE-SP-2-UNBLOCK_CONSIST_PORT: Unblocking FastEthernet0/1 on vlan 1.
Port consistency restored.
```

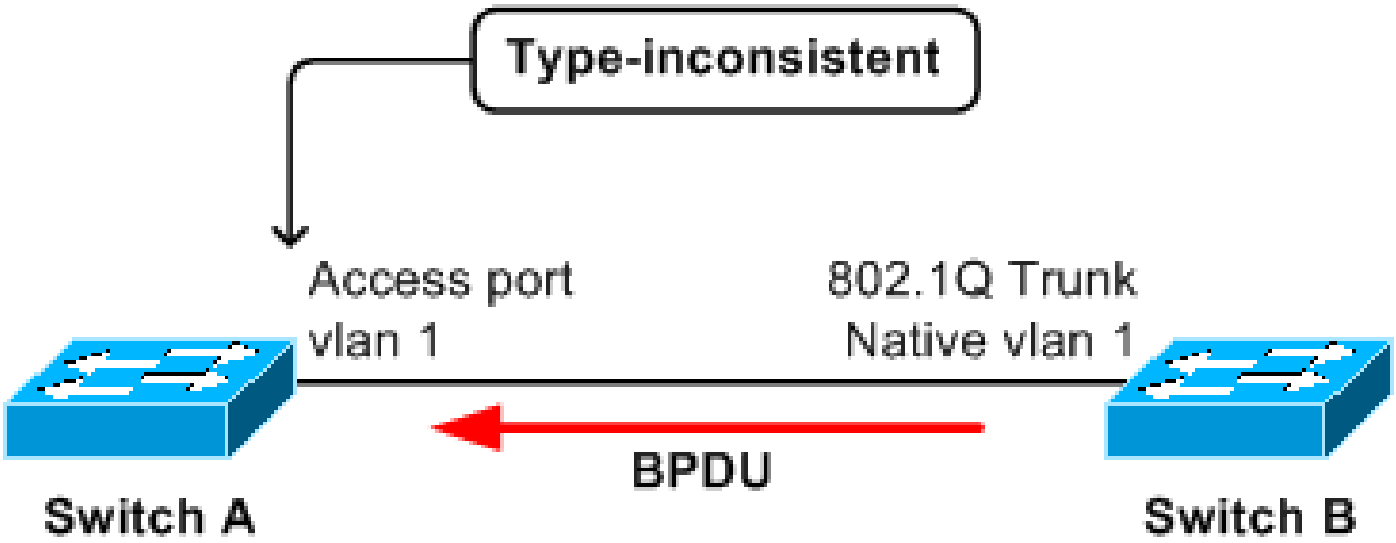
For more details on PVST+ operation, refer to Spanning Tree from PVST+ to Rapid-PVST Migration Configuration Example.

Troubleshoot

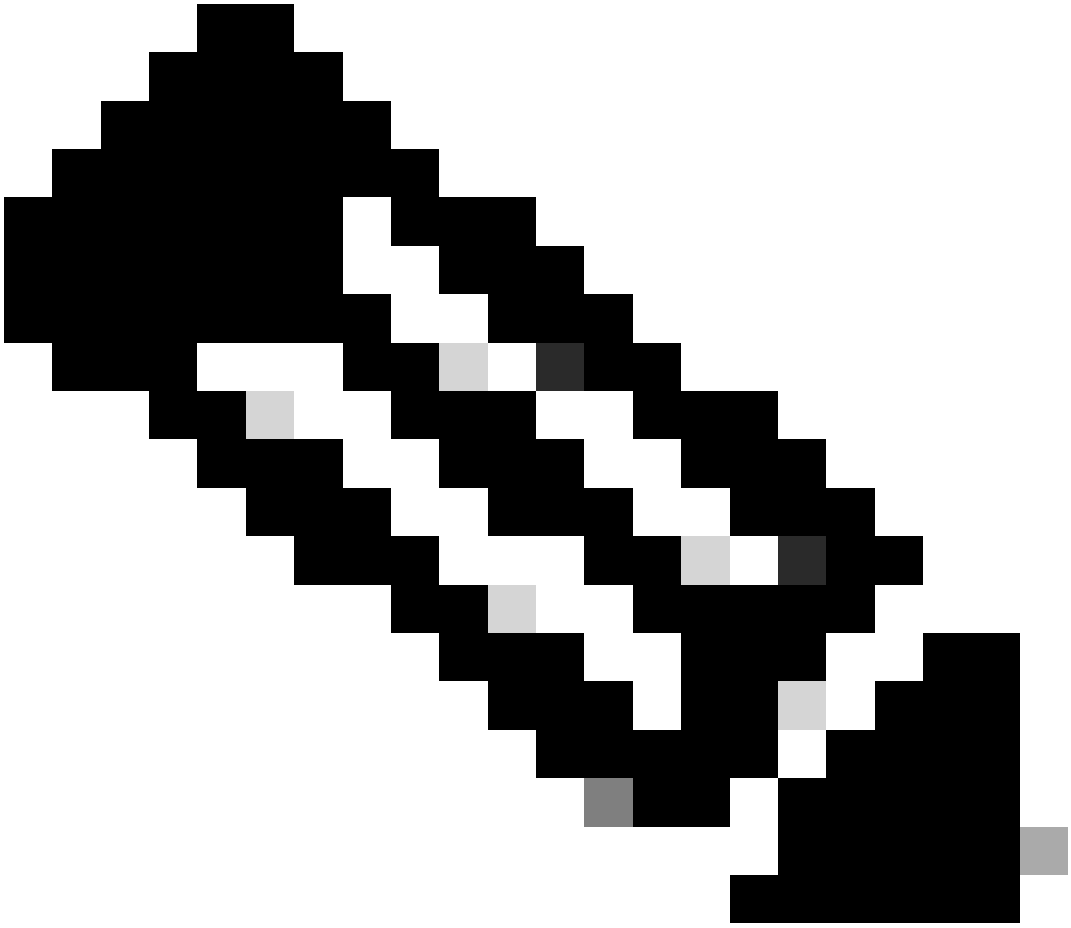
To see the list of inconsistent ports, recent Cisco IOS-based STP implementation supports the **show spanning-tree inconsistentports** command.

In the majority of cases, the reason for detection of STP inconsistency on the port is apparent:

- Access port receives an IEEE 802.1Q-tagged SSTP BPDU.

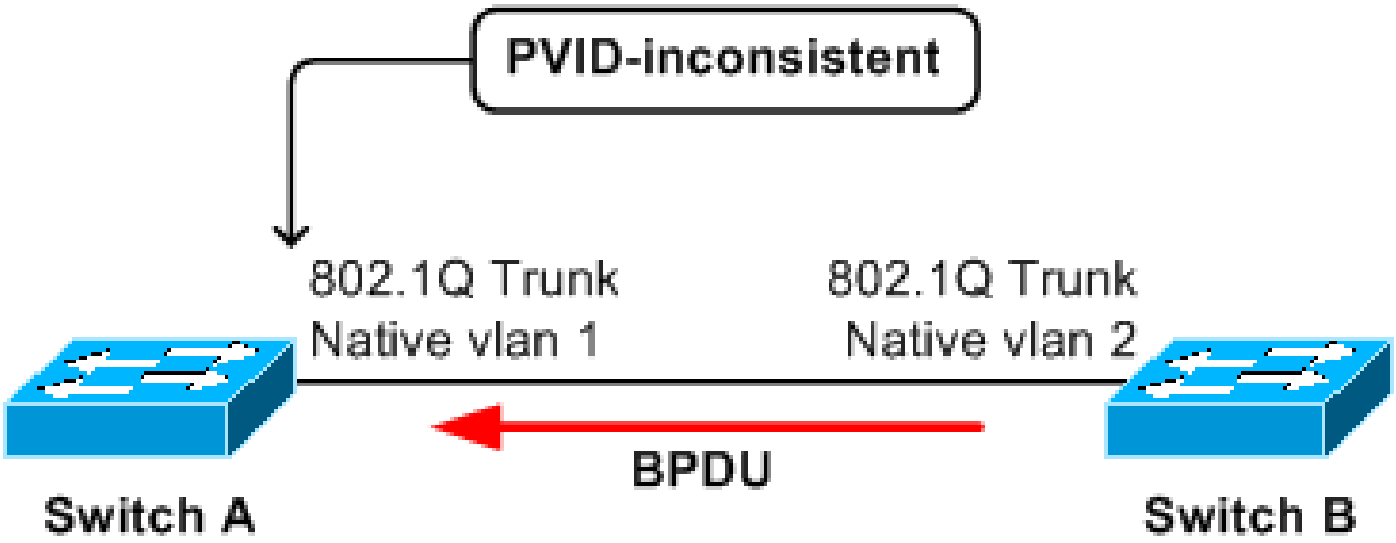


In this scenario, access port on bridge A receives, from bridge B, a tagged PVST+ BPDU from STP of a VLAN other than 1. The port on A can be put into type-inconsistent state.



Note: The switches need not be connected directly; if they are connected through one or more IEEE 802.1D or IEEE 802.1Q switches—or even hubs—then the effect is the same.

- IEEE 802.1Q trunking port receives an untagged SSTP BPDU with a VLAN type, length, value (TLV) that does not match the VLAN where the BPDU was received.



In this scenario, the trunk port on A receives a PVST+ BPDU from STP of VLAN 2 with a tag of VLAN 2. This triggers the port on A to be blocked in both VLAN 1 and VLAN 2.

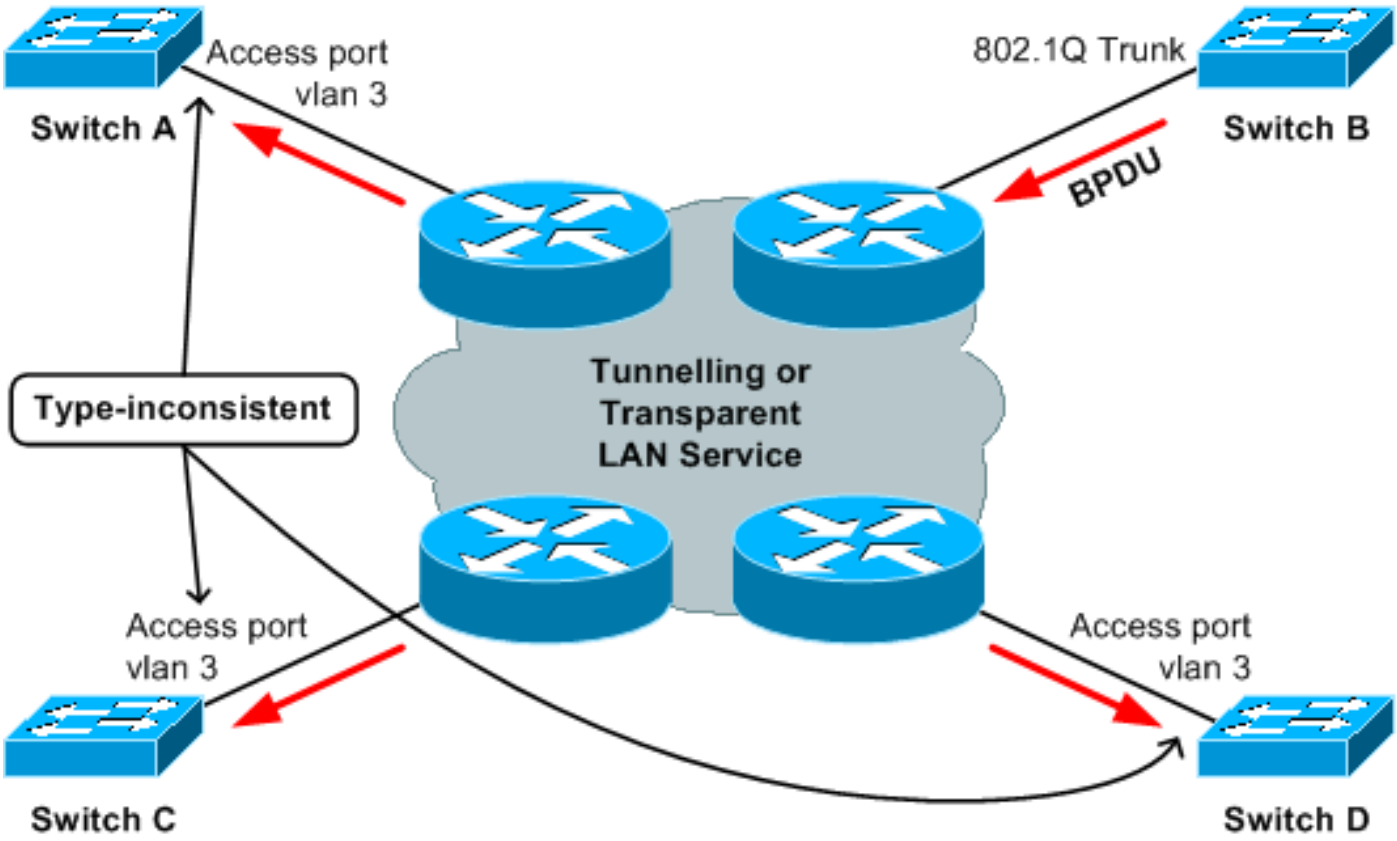
If devices on both ends of a point-to-point link are Cisco Catalyst switches, an examination of the local and remote port configuration typically reveals the configuration mismatch:

- The port is configured for IEEE 802.1Q trunking on one side but the other side is access port.
- IEEE 802.1Q trunks are on both sides, but the Native VLANs are different.

In these cases, fix the configuration mismatch to resolve the STP inconsistency.

In some cases, it is more difficult to identify the reason:

- A BPDU is received from a shared media with multiple devices.
- A BPDU is received, from the switch cloud, which implements an IEEE 802.1D or 802.1Q STP model while PVST+ switches are connected to the cloud.
- A BPDU comes from behind some tunnel (for example, Data Link Switch Plus [DLSw+] cloud, L2 protocol tunneling, EoMPLS, Virtual Path Links [VPLs], LAN Emulation [LANE], and others).



In this example, switch B is misconfigured and injects an SSTP BPDU into the cloud. This causes the ports on switches A, C, and D to become type-inconsistent.

The issue is that the device that originates the offending BPDU is not directly connected to the affected switches.

Thus, with many devices on the trunk, it can consume time to troubleshoot all of them.

Fortunately, there is a systematic approach to troubleshoot this issue:

1. Establish the **source MAC address** andsending **bridge ID** of the BPDU. This must be done while the issue occurs.

2. Find the bridge that originates the offending BPDU. This can be done at some later time, not necessarily when the issue occurs.

For Step 1, there are normally two options: use a packet analyzer or enable debug to see the dump of received BPDUs.

For more details about the use of a debug to dump STP BPDUs, refer to the Use STP Debug Commands section of Troubleshoot STP Issues on Catalyst Switches.

This is an example of debug output that shows received BPDU:

```
*Mar 14 19:33:27: STP SW: PROC RX: 0100.0ccc.cccd<-0030.9617.4f08 type/len 0032
*Mar 14 19:33:27:     encap SNAP linktype sstp vlan 10 len 64 on v10 Fa0/14
*Mar 14 19:33:27:     AA AA 03 00000C 010B SSTP
*Mar 14 19:33:27:     CFG P:0000 V:00 T:00 F:00 R:8000 0050.0f2d.4000 00000000
*Mar 14 19:33:27:     B:8000 0050.0f2d.4000 80.99 A:0000 M:1400 H:0200 F:0F00
*Mar 14 19:33:27:     T:0000 L:0002 D:0001
```

Once you know the source MAC address and sending bridge ID, you need to find the device to which this MAC address belongs. This can be complicated by the fact that switches typically do not learn a source's MAC addresses from BPDU frames.

If you issue the **show mac-address-table addressBPDU_mac_address** command (for Cisco IOS-based switches) then, typically, no entry is found.

One way to find the offending MAC address is to collect, from all switches that are connected to the cloud, output from the **show spanning-tree** command.

These command outputs include information about the bridge ID of each bridge.

```
<#root>

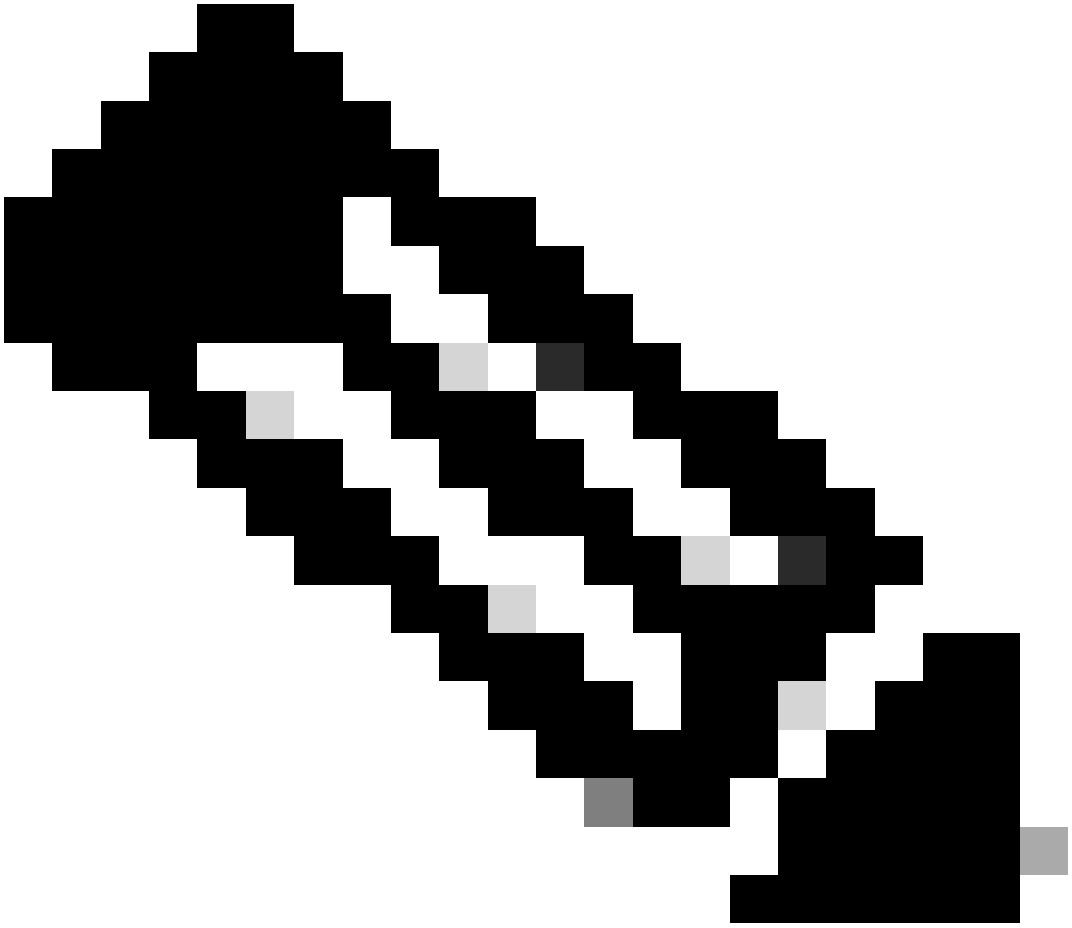
Boris#

show spanning-tree


!--- Use with Cisco IOS.

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    0
             Address     0007.4f1c.e847
             Cost        131
             Port        136 (GigabitEthernet3/8)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     00d0.003f.8800

!--- Output suppressed.
```



Note: Based on the model, software version, and configuration, a switch can have multiple bridge ID MAC addresses. Fortunately, all of the addresses can typically be in a certain range (for example, from 0001.1234.5600 to 0001.1234.5640). If you know one bridge ID MAC address, then you can check whether the sent bridge ID MAC address (found in Step 1) falls within the range of given bridge ID MAC addresses. You can also use network management tools to collect the bridge IDs of all bridges.

Once you have found the bridge that has sent the offending BPDU, you need to verify the configuration of the port that is attached to the cloud: ensure that it is consistent (trunking as opposed to non-trunking and Native VLAN) with other switches that are also connected to the same cloud.

It could happen that the bridge sends proper BPDUs, but they are incorrectly modified inside the tunneling cloud. In that case, you can see that the offending BPDU that enters the cloud is consistent with the configuration of the other bridges, but the same BPDU becomes inconsistent when it exits the cloud (for example, the BPDU exits the cloud in a different VLAN, or becomes tagged or untagged). In such a case, it can help to check whether the source MAC address of the offending BPDU belongs to the same bridge as the sending bridge ID.

If this is not the case, then you can try to locate the bridge that owns the source MAC address of the BPDU and verify its configuration.

To locate the switch that owns the source MAC address of the BPDU, you can use the same approach (to find the bridge ID), except now the **show module** command output is inspected (for Catalyst 4000, and 6000). For other Catalyst switches, you can examine the output of the **show interface** command to see the MAC addresses that belong to the ports.

```
<#root>

Cat4000-#

show module


!--- Use for Catalyst 4000,5000,6000

Mod  Ports Card Type                                  Model              Serial No.
----+-----+----------------------------------------+------------------+-----------
 1      2  1000BaseX (GBIC) Supervisor(active)    WS-X4515           ZZZ00000001
 5     14  1000BaseT (RJ45), 1000BaseX (GBIC)     WS-X4412-2GB-T     ZZZ00000002

 M MAC addresses                     Hw  Fw           Sw               Status
--+--------------------------------+---+------------+----------------+---------
 1 000a.4172.ea40 to 000a.4172.ea41 1.2 12.1(12r)EW  12.1(14)E1, EARL Ok
 5 0001.4230.d800 to 0001.4230.d80d 1.0                                Ok

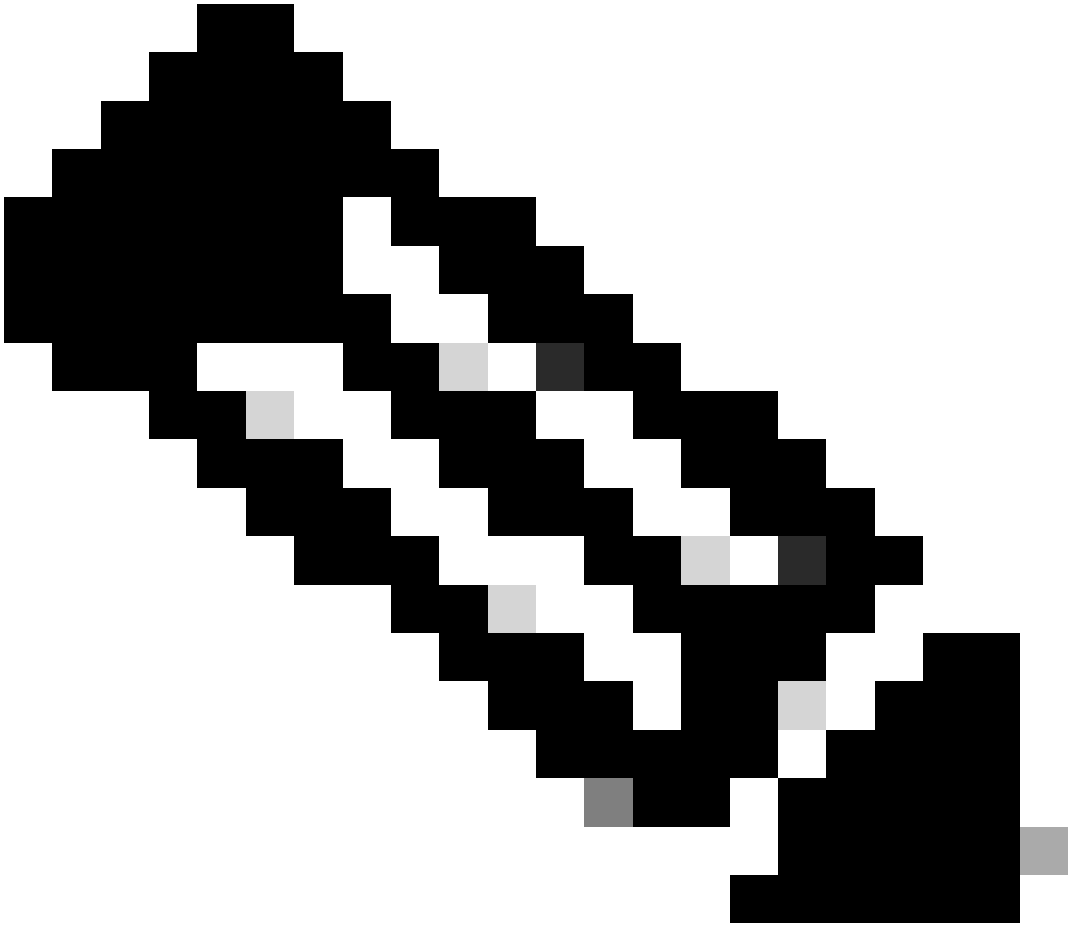
!--- Output suppressed.

cat3550#

show interface | i bia


  Hardware is Gigabit Ethernet, address is 0002.4b28.da80 (bia 0002.4b28.da80)
  Hardware is Gigabit Ethernet, address is 0002.4b28.da83 (bia 0002.4b28.da83)
  Hardware is Gigabit Ethernet, address is 0002.4b28.da86 (bia 0002.4b28.da86)
  Hardware is Gigabit Ethernet, address is 0002.4b28.da88 (bia 0002.4b28.da88)
  Hardware is Gigabit Ethernet, address is 0002.4b28.da89 (bia 0002.4b28.da89)

!--- Output suppressed.
```



Note: If the cloud is DLSw+, refer to Understanding and Configuring DLSw and 802.1Q

Related Information

- LAN/Spanning Tree Protocol Product Support
- Technology Support
- Cisco Technical Support & Downloads