# ASR 9000 Series Common Problems with Spanning Tree Protocols

**Document ID: 116514**

Contributed by Bryan Garland and David Powers, Cisco TAC Engineers.
Sep 19, 2013

## Contents



## Introduction

This document describes common problems encountered when you integrate your current Layer 2 (L2) Spanning Tree networks on Cisco IOS® switches with Cisco Aggregation Services Router (ASR) 9000 Series that run Cisco IOS XR.

## Problem − Port VLAN ID (PVID) inconsistency

Cisco IOS switches that run Per VLAN Spanning Tree Plus (PVST+) block switch ports when they receive a Bridge Protocol Data Unit (BPDU) with an insconsistent PVID. This problem occurs when a device between the switches changes or translates the IEEE 802.1Q tags on the PVST+ BPDUs.

When an ASR 9000 provides L2VPN point−to−point or multipoint service between switches that run PVST+ and rewrites the VLAN tags, these syslog messages might display on the Cisco IOS−based switches:

```
%SPANTREE−2−RECV_PVID_ERR: Received BPDU with inconsistent
peer vlan id 10 on GigabitEthernet0/10 VLAN20.

%SPANTREE−2−BLOCK_PVID_LOCAL: Blocking GigabitEthernet0/10
on VLAN20. Inconsistent local vlan.
```

This issue is due to the PVID tag that is included with the PVST+ BPDUs. This tag is designed in order to detect misconfigurations and avoid accidental loops. But, in this scenario, it causes each end to be blocked and to not allow traffic to pass.

Here is an example:


Here is the configuration for the ASR 9000 Series (a9k1) configuration:

```
2vpn
bridge group bg1
 bridge-domain bd1
 interface TenGigE0/0/0/0.10
 !
 interface TenGigE0/0/0/1.20

interface TenGigE0/0/0/0.10 l2transport
encapsulation dot1q 10
rewrite ingress tag pop 1 symmetric

interface TenGigE0/0/0/1.20 l2transport
encapsulation dot1q 20
rewrite ingress tag pop 1 symmetric
```

# Solution

In order to prevent this problem, you can block the PVST+ BPDUs. This action disables Spanning Tree, and can result in loops if redundant connections are available between the switches.

***Caution***: Use caution when you block BPDUs and effectively disable Spanning Tree.

## BPDU Filter on Switches

The BPDUs are blocked with the BPDU filter feature on the switches. The BPDU filter blocks BPDUs in both directions, which effectively disables Spanning Tree on the port. The BPDU filter prevents inbound and outbound BPDU. If you enable BPDU filtering on an interface, it is the same as if you disable Spanning Tree on it, which can result in Spanning Tree loops.

On switch1 and switch2, enable BPDU filters with this command:

```
interface TenGigabitEthernet1/2
spanning-tree bpdufilter enable
```

## Block PVST+ BPDUs on ASR 9000

This problem is avoided if you configure the ASR9000 in order to drop the PVST+ BPDUs. This is done with an L2 ethernet−services access−list to deny packets destined to the PVST+ BPDU MAC address.

PVST+ BPDU for the non−VLAN 1 (non−native) VLAN are sent to the PVST+ MAC address (also called the Shared Spanning Tree Protocol [SSTP] MAC address, 0100.0ccc.cccd), and tagged with a corresponding IEEE 802.1Q VLAN tag.

This Access Control List (ACL) can be used in order to block the PVST+ BPDUs:

```
ethernet-services access-list l2acl
10 deny any host 0100.0ccc.cccd
 20 permit any any
```

Apply the ACL to the interface configured as l2transport:

```
interface TenGigE0/0/0/0.10 l2transport
encapsulation dot1q 10
rewrite ingress tag pop 1 symmetric
ethernet-services access-group l2acl ingress

interface TenGigE0/0/0/1.20 l2transport
encapsulation dot1q 20
rewrite ingress tag pop 1 symmetric
ethernet-services access-group l2acl ingress
```

# Problem − Switch ports flap between blocking and forwarding when you use multiple types of Spanning Tree Protocols (STPs) through an ASR 9000

The ASR9000 does not do Spanning Tree by default like most Cisco IOS switches. In the Ethernet Virtual Circuit (EVC) model, a BPDU is simply another L2 multicast packet. A common issue encountered is Spanning Tree inconsistencies due to multiple types of STPs that run across an ASR 9000 bridge domain. This appears in a few different ways.

Consider this simple topology:


Assume switch1 runs Multiple Spanning Tree (MST) and switch2 runs PVST+. If a9k1 does not run any form of Spanning Tree, then switch1 sees this as a boundary port. Switch1 falls back to PVST mode for VLANs not in Common Spanning Tree Instance 0 (CST0). If this is the desired design, you should be familiar with MST and PVST interaction as described in the Understanding Multiple Spanning Tree Protocol (802.1s) white paper.

Now assume you run MST on switch1 and on the a9k1 interface that goes to switch1, but you still run PVST+ on switch2. The PVST+ BPDUs passes through the bridge domain and arrives at switch1. Switch1 then sees both MST BPDUs from a9k1 and the PVST+ BPDUs from switch2, which causes Spanning Tree on switch1 port to constantly go from blocking to not blocking and results in traffic loss.

Switch1 reports these syslogs:

```
%SPANTREE-SP-2-PVSTSIM_FAIL: Superior PVST BPDU received on VLAN 2 port Gi2/13,
claiming root 2:000b.45b7.1100. Invoking root guard to block the port
%SPANTREE-SP-2-ROOTGUARD_BLOCK: Root guard blocking port GigabitEthernet2/13
 on MST1.
%SPANTREE-SP-2-ROOTGUARD_UNBLOCK: Root guard unblocking port GigabitEthernet2/13
on MST0.
%SPANTREE-SP-2-PVSTSIM_FAIL: Superior PVST BPDU received on VLAN 2 port Gi2/13,
claiming root 2:000b.45b7.1100. Invoking root guard to block the port
%SPANTREE-SP-2-ROOTGUARD_BLOCK: Root guard blocking port GigabitEthernet2/13
on MST1.
```

The ***show spanning-tree interface*** command output shows that the output constantly changes on the switch1 Cisco IOS device:

```
show spanning-tree interface gig 2/13
Mst Instance Role Sts Cost Prio.Nbr Type
```

```
------------------ ---- --- --------- -------- -------
MST0 Desg BKN*20000 128.269 P2p Bound(PVST) *ROOT_Inc
MST1 Desg BKN*20000 128.269 P2p Bound(PVST) *ROOT_Inc
MST2 Desg BKN*20000 128.269 P2p Bound(PVST) *ROOT_Inc

show spanning-tree interface gig 2/13
Mst Instance Role Sts Cost Prio.Nbr Type
------------------ ---- --- --------- -------- ---------
MST0 Desg FWD 20000 128.269 P2p
MST1 Desg FWD 20000 128.269 P2p
MST2 Desg FWD 20000 128.269 P2p
```

# Solution

There are three options to consider In order to prevent this problem.

- Configure MST on switch2, and enable MST on the a9k1 interfaces to both switch1 and switch2.
- Use an Ethernet−services access−list on a9k1 in order to drop the PVST+ BPDUs either on ingress from switch2 or on egress to switch1.
- Run Per VLAN Spanning Tree Access Gateway (PVSTAG) on the a9k1 interface towards switch2. This causes the a9k1 to consume the PVST+ BPDUs from switch2.

# Problem − Spanning Tree ports blocked due to detection of a self−loop

When a switch receives a Spanning Tree BPDU that it sent on the same interface, it blocks that VLAN due to a self−loop. This is a common problem that occurs when a switch with a trunk port is connected to an ASR 9000 router that provides L2 multipoint services, and the ASR 9000 does not rewrite VLAN tags on the l2transport interfaces in the same bridge domain.

Consider the same simple topology shown previously. But now, for a design reason on the a9k1, multiple VLANs that come from the same switch trunk interface are merged together in one bridge domain.


Here is the a9k1 configuration:

```
l2vpn
bridge group bg1
bridge-domain bd1
interface GigabitEthernet0/1/0/31.2
!
interface GigabitEthernet0/1/0/31.3
!
interface GigabitEthernet0/1/0/31.4
!
interface GigabitEthernet0/1/0/32.2
!
interface GigabitEthernet0/1/0/32.3
!
interface GigabitEthernet0/1/0/32.4

interface GigabitEthernet0/1/0/31.2 l2transport
encapsulation dot1q 2
!
```

```
interface GigabitEthernet0/1/0/31.3 l2transport
 encapsulation dot1q 3
!
interface GigabitEthernet0/1/0/31.4 l2transport
encapsulation dot1q 4
```

This bridges VLANs 2 through 4 together in one bridge domain on the a9k1.

The ASR 9000 EVC model does not rewrite any tags or pop by default. The PVST+ BPDU for ***VLAN2*** comes in on interface ***gig 0/1/0/31.2*** and is forwarded back out on ***gig 0/1/0/31.3*** and ***gig 0/1/0/31.4***. Since the configuration is not a rewrite of ingress pop action, the BPDU returns unchanged. The switch sees this as it gets its own BPDU back, and blocks that VLAN due to a self−loop.

The ***show spanning−tree interface*** command shows the VLAN blocked:

```
6504−A#show spanning−tree interface gig 2/13


Vlan Role Sts Cost Prio.Nbr Type
−−−−−−−−−−−−−−−−−− −−−− −−− −−−−−−−−− −−−−−−−−
VLAN0002 Desg BLK 4 128.269 self−looped P2p
VLAN0003 Desg BLK 4 128.269 self−looped P2p
VLAN0004 Desg BLK 4 128.269 self−looped P2p
```

# Solution

This issue is eliminated through use of the ***ethernet egress−filter strict*** command on the ASR 9000 l2transport interfaces.

This is not a recommended design. However, if this is truly the desired design, then you can use this solution in order to prevent the switch from recieving the BPDU that it sent back in the same interface.

You can use the ***ethernet egress−filter strict*** command on the a9k1 l2transport interfaces or globally. Here is the example of it under the interface:

```
interface GigabitEthernet0/1/0/31.2 l2transport
 encapsulation dot1q 2
 ethernet egress−filter strict
!
interface GigabitEthernet0/1/0/31.3 l2transport
 encapsulation dot1q 3
ethernet egress−filter strict
!
interface GigabitEthernet0/1/0/31.4 l2transport
 encapsulation dot1q 4
ethernet egress−filter strict
```

The ***ethernet egress−filter strict*** command enables strict egress Ethernet Flow Point (EFP) filtering on the interface. Only packets that pass the ingress EFP filter on the interface are transmitted out of this interface. Other packets are dropped at the egress filter. This means that if the packet that egresses does not match the encapsulation ***dot1q*** label configured on the interface, then it is not sent out.

# Related Information

- ***Implementing Multiple Spanning Tree Protocol***
- ***Troubleshooting Spanning Tree PVID− and Type−Inconsistencies***
- ***Understanding Multiple Spanning Tree Protocol (802.1s)***

Updated: Sep 19, 2013 Document ID: 116514