

# VLAN Load Balancing Between Trunks Using the Spanning-Tree Protocol Port Priority

Document ID: 10555

## Contents

### Introduction

#### Before You Begin

- Conventions

- Prerequisites

- Components Used

#### Introduction to VLAN Load Balancing Between Trunks

- How STP Decides Which Port to Block

#### Configuring VLAN Load Balancing on Catalyst Switches Running CatOS

- Details on the portvlanpri Command

#### Configuring VLAN Load Balancing on Catalyst Switches Running Integrated IOS

- Details on the port-priority and vlan port-priority Commands

### Conclusion

### Related Information

## Introduction

This document provides the theory behind VLAN load balancing between trunks, and also provides configuration examples for switches running CatOS and Integrated IOS.

## Before You Begin

### Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

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

## Introduction to VLAN Load Balancing Between Trunks

The commands used in this document are available on the Catalyst 4000, 5000, and 6000 families running CatOS and Integrated IOS. The theoretical sections of this document are related to spanning-tree protocol (STP) and are platform independent.

The configuration shown in figure 1 below, in which two switches are directly connected through more than one trunk, is very common for redundancy purposes. If one of the two links fail, the second soon becomes available to transmit traffic. When both links are up, the spanning-tree algorithm (STA) disables one of them in order to avoid a bridging loop between the two switches.

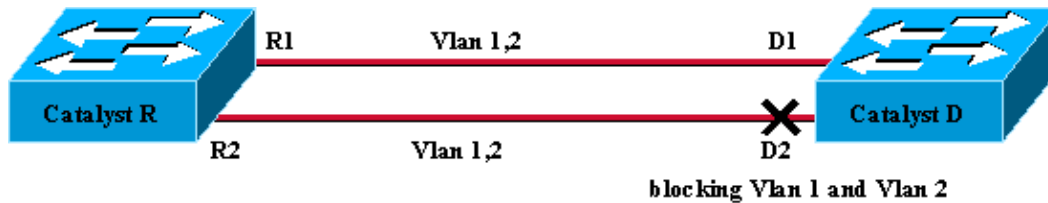


Figure 1

In the figure 1 configuration above, with two FastEthernet trunks linking Catalyst R and Catalyst D, the STP elects the same blocking port for all the VLANs configured. In this case, Catalyst R is the Root bridge and Catalyst D decides to block port D2 for both VLAN 1 and VLAN 2. The major issue with this design is that link R2–D2 is simply sacrificed and there is only 100 Mb/s available between the two switches. To take advantage of both links, you can change the configuration and allow VLAN 1 only on link R1–D1, and VLAN 2 only on link R2–D2.

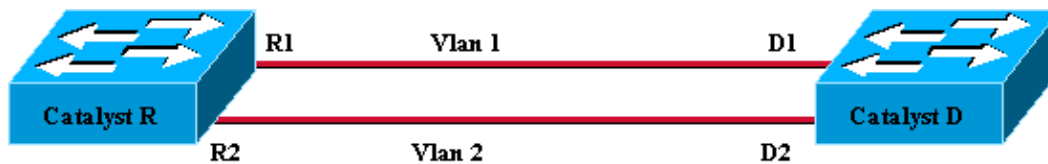


Figure 2

The resulting network, shown in figure 2, lost its redundancy. Now you have both links forwarding at the same time, and you are practically able to use 200 Mb/s between the two switches. However, if one link fails, you completely lose connectivity for one VLAN. The ideal solution is the one described in figure 3:

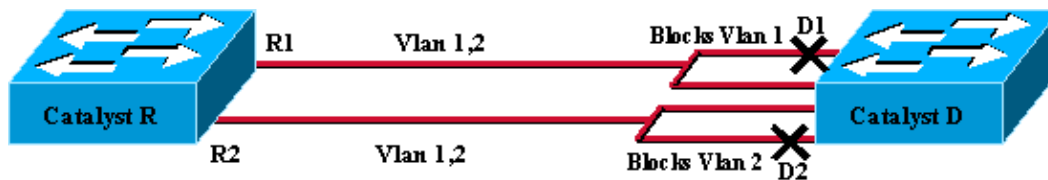


Figure 3

In figure 3, you keep the trunks between the two switches, but port D1 is blocking VLAN 1 and forwarding VLAN 2; port D2 is blocking VLAN 2 and forwarding VLAN 1. This design keeps the best features of figure 1 and figure 2:

- Both links are forwarding, providing 200 Mb/s aggregate connectivity between the two switches.
- If one link fails, the remaining one unblocks the corresponding VLAN and maintains connectivity for both VLANs between the switches.

This document explains how to achieve such a configuration, after a short explanation of the STP operations.

## How STP Decides Which Port to Block

The detailed description of how the STP works is beyond the scope of this document. However, it briefly summarizes how the algorithm decides, in this case, whether a port blocks or forwards. It focuses on the most simple configuration possible with only one VLAN; Catalyst R is the Root bridge in this VLAN and Catalyst D has multiple redundant connections to Catalyst R. Catalyst D blocks all of its ports to Catalyst R but its Root port. How does Catalyst D select its Root port? Bridges running the STP exchange Bridge Protocol Data Units (BPDUs) across the links, and these BPDUs can be strictly classified depending on their content. A BPDU is superior than another if it has:

1. A lower Root Bridge ID.
2. A lower path cost to the Root.
3. A lower Sending Bridge ID.
4. A lower Sending Port ID.

These four parameters are examined in order, that is, you only care about parameter 2 if parameter 1 is the same in the two BPDUs you are comparing. **The port that is elected Root Port on Catalyst D is the port that is receiving the best BPDUs.**

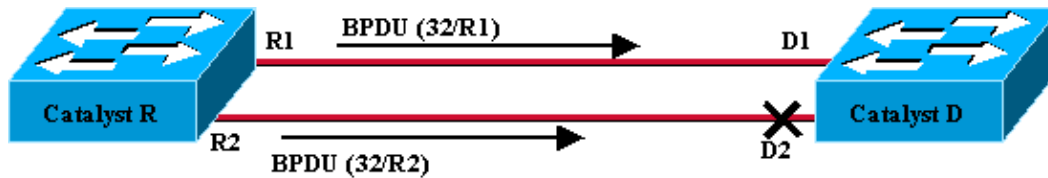


Figure 4

In this specific case, figure 4, all the BPDUs sent by Catalyst R have the same Root Bridge ID, the same path cost to the Root and the same Sending Bridge ID. The only parameter left to select the best one is the Sending Port ID. The Sending Port ID is a 16 bit parameter, split in two fields: the port priority and a port index. The default value for the port priority is 32 and the port index is unique to each port on the switch.

	Port Priority	Port Index
Size in bits	6	10
Default value	32	Fixed unique value

Figure 4 represents the port ID parameter in the BPDUs. In this case, Catalyst D chooses port D1 as its Root port because the port index R2 is inferior to R1. If you want D2 to be eventually forwarding, you need to force it as the Root port. The only way to do this is to reduce R2 port priority value (or increase R1 port priority value). This is what was done on figure 5.

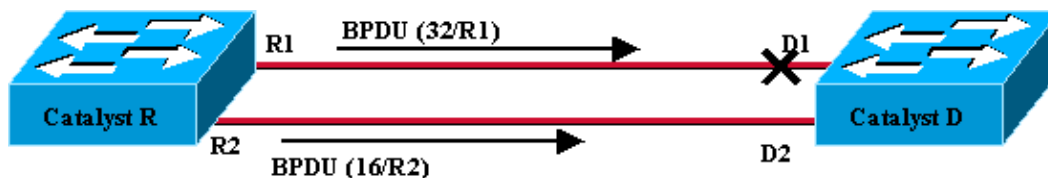


Figure 5

In order to achieve load-balancing between two trunks, you tune on a VLAN basis the port priority on Catalyst R.

## Configuring VLAN Load Balancing on Catalyst Switches Running CatOS

### Lab Diagram

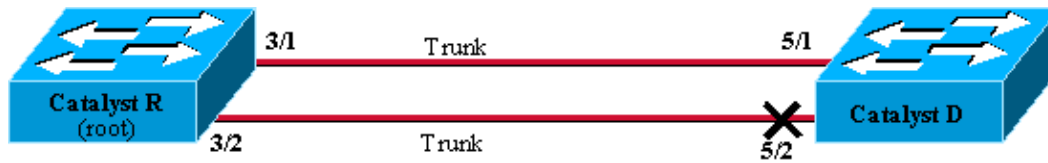


Figure 6

## Show Current STP Status on Catalyst R

Here is the current STP status on Catalyst R. It is the Root for both VLAN 1 and 2, so all its ports are forwarding.

```
Catalyst_R> (enable) show spantree 3/1
Port                Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
3/1                 1    forwarding        19    32 disabled  0
3/1                 2    forwarding        19    32 disabled  0
Catalyst_R> (enable) show spantree 3/2
Port                Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
3/2                 1    forwarding        19    32 disabled  0
3/2                 2    forwarding        19    32 disabled  0
Catalyst_R> (enable)
```

## Show Current STP Status on Catalyst D

On Catalyst D, as expected, port 5/2 is blocked for both VLANs 1 and 2.

```
Catalyst_D> (enable) show spantree 5/1
Port    Vlan  Port-State      Cost  Priority  Fast-Start  Group-Method
-----
5/1     1    forwarding      19    32    disabled
5/1     2    forwarding      19    32    disabled
Catalyst_D> (enable) show spantree 5/2
Port    Vlan  Port-State      Cost  Priority  Fast-Start  Group-Method
-----
5/2     1    blocking        19    32    disabled
5/2     2    blocking        19    32    disabled
Catalyst_D> (enable)
```

## Tune the Port Priority on Catalyst R

You are going to decrease the port priority value for VLAN 1 on port 3/2. This way, the corresponding port 5/2 on Catalyst D receives better BPDUs than the ones that are sent on port 5/1 (that still has a port priority value of 32).

```
Catalyst_R> (enable) set spantree portvlanpri 3/2 16 1
Port 3/2 vlans 1 using portpri 16.
Port 3/2 vlans 2-1004 using portpri 32.
Port 3/2 vlans 1005 using portpri 4.
Catalyst_R> (enable)
```

## Check Result on Catalyst R

You can check that the port priority value has changed for VLAN 1:

```
Catalyst_R> (enable) show spantree 3/1
Port                Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
```

```

3/1          1    forwarding    19      32 disabled  0
3/1          2    forwarding    19      32 disabled  0
Catalyst_R> (enable) show spantree 3/2
Port          Vlan Port-State    Cost  Priority Portfast  Channel_id
-----
3/2          1    forwarding    19      16 disabled  0
3/2          2    forwarding    19      32 disabled  0
Catalyst_R> (enable)

```

## Check Result on Catalyst D

You can see on Catalyst D that, for VLAN 1, port 5/1 is now blocking and port 5/2 is forwarding, as was expected.

```

Catalyst_D> (enable) show spantree 5/1
Port          Vlan Port-State    Cost  Priority  Fast-Start  Group-Method
-----
5/1          1    blocking      19      32 disabled
5/1          2    forwarding    19      32 disabled
Catalyst_D> (enable) show spantree 5/2
Port          Vlan Port-State    Cost  Priority  Fast-Start  Group-Method
-----
5/2          1    forwarding    19      32 disabled
5/2          2    blocking      19      32 disabled
Catalyst_D> (enable)

```

## Details on the portvlanpri Command

There are only two possible values for the port VLAN priority per trunk, and only one of them is able to be configured using the **portvlanpri** command. It means that on a given trunk, you have two groups of VLANs:

- The ones that have the "global" port priority value (32 by default).
- The ones that have a "custom" value entered with the **portvlanpri** command.

An example clarifies this. Consider adding a third VLAN in the example. By default, this VLAN belongs to the group that has the global port priority value (default 32).

```

Catalyst_R> (enable) set vlan 3
Vlan 3 configuration successful
Catalyst_R> (enable) show spantree 3/2
Port          Vlan Port-State    Cost  Priority Portfast  Channel_id
-----
3/2          1    forwarding    19      16 disabled  0
3/2          2    forwarding    19      32 disabled  0
3/2          3    forwarding    19      32 disabled  0
Catalyst_R> (enable)

```

Change the global priority for the port, using the **set spantree portpri** command:

```

Catalyst_R> (enable) set spantree portpri 3/2 48
Bridge port 3/2 port priority set to 48.
Catalyst_R> (enable) show spantree 3/2
Port          Vlan Port-State    Cost  Priority Portfast  Channel_id
-----
3/2          1    forwarding    19      16 disabled  0
3/2          2    forwarding    19      48 disabled  0
3/2          3    forwarding    19      48 disabled  0
Catalyst_R> (enable)

```

Notice that all the VLANs belonging to the "global" group changed their priority to 48. Now assign VLAN 3 to the other "custom" group of VLANs, giving it a value of 8 with the **portvlanpri** command:

```
Catalyst_R> (enable) set spantree portvlanpri 3/2 8 3
Port 3/2 vlans 1,3 using portpri 8.
Port 3/2 vlans 2,4-1004 using portpri 48.
Port 3/2 vlans 1005 using portpri 4.
Catalyst_R> (enable) show spantree 3/2
Port                Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
3/2                 1    forwarding         19    8 disabled  0
3/2                 2    forwarding         19    48 disabled 0
3/2                 3    forwarding         19    8 disabled  0
Catalyst_R> (enable)
```

Notice that all the VLANs in the "custom" group have changed their priority to 8, not just VLAN 3. To put VLAN 3 back in the default group, use the **clear spantree portvlanpri** command:

```
Catalyst_R> (enable) clear spantree portvlanpri 3/2 3
Port 3/2 vlans 1 using portpri 8.
Port 3/2 vlans 2-1004 using portpri 48.
Port 3/2 vlans 1005 using portpri 4.
Catalyst_R> (enable) show spantree 3/2
Port                Vlan Port-State      Cost  Priority Portfast  Channel_id
-----
3/2                 1    forwarding         19    8 disabled  0
3/2                 2    forwarding         19    48 disabled 0
3/2                 3    forwarding         19    48 disabled 0
Catalyst_R> (enable)
```

There is one last constraint on this command. The value assigned to the "global" group must be superior to the one configured on the "custom" group.

```
Catalyst_R> (enable) set spantree portvlanpri 3/2 62 3
Portvlanpri must be less than portpri. Portpri for 3/2 is 48.
```

To summarize:

"global" group	"custom" group
By default, all VLANs belong to this group.	VLANs selected with the <b>set spantree portvlanpri</b> command belong to this group.
The priority for these VLANs is set using the <b>set spantree port priority</b> command.	The priority value for all these VLANs is set by the <b>set spantree portvlanpri</b> command.
The priority value configured for the "global" group must be superior to the one configured for the "custom" group.	The <b>clear spantree portvlanpri</b> allows you to put back a VLAN from this group

## Configuring VLAN Load Balancing on Catalyst Switches Running Integrated IOS

**Note:** This configuration example applies to switches running IOS – Catalyst 2900/3500XL, Catalyst 2950, Catalyst 3550, Catalyst 4000 supervisor III/IV, and Catalyst 6000.

## Lab Diagram

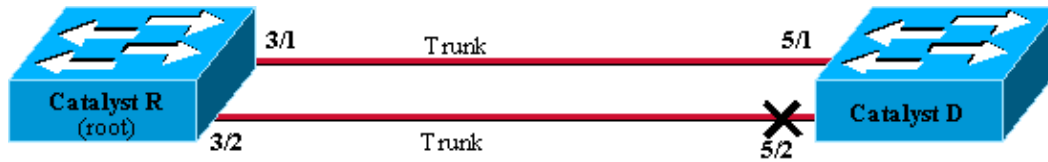


Figure 6

## Show Current STP Status on Catalyst R

Here is the current STP status on Catalyst R. It is the Root for both VLAN 1 and 2, so all its ports are forwarding.

```
Catalyst_R#show spanning-tree interface FastEthernet 3/1
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	128.129	P2p
VLAN0002	Desg	FWD	19	128.129	P2p

```
Catalyst_R#show spanning-tree interface FastEthernet 3/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	128.130	P2p
VLAN0002	Desg	FWD	19	128.130	P2p

## Show Current STP Status on Catalyst D

On Catalyst D, as expected, port 5/2 is blocked for both VLANs 1 and 2.

```
Catalyst_D#show spanning-tree interface FastEthernet 5/1
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Root	FWD	19	128.129	P2p
VLAN0002	Root	FWD	19	128.129	P2p

```
Catalyst_D#show spanning-tree interface FastEthernet 5/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Altn	BLK	19	128.130	P2p
VLAN0002	Altn	BLK	19	128.130	P2p

## Tune the Port Priority on Catalyst R

You are going to decrease the port priority value for VLAN 1 on port 3/2. This way, the corresponding port 5/2 on Catalyst D receives better BPDUs than the ones that are sent on port 5/1 (that still has a port priority value of 128).

```
Catalyst_R#config terminal  
Catalyst_R(config)#interface FastEthernet 3/2
```

```
Catalyst_R(config-if)#spanning-tree vlan 1 port-priority 64
Catalyst_R(config-if)#end
Catalyst_R#
```

## Check Result on Catalyst R

You can check that the port priority value has changed for VLAN 1:

```
Catalyst_R#show spanning-tree interface FastEthernet 3/1
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	128.129	P2p
VLAN0002	Desg	FWD	19	128.129	P2p

```
Catalyst_R#show spanning-tree interface FastEthernet 3/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	64.130	P2p
VLAN0002	Desg	FWD	19	128.130	P2p

## Check Result on Catalyst D

You can see on Catalyst D that, for VLAN 1, port 5/1 is now blocking and port 5/2 is forwarding, as was expected.

```
Catalyst_D#show spanning-tree interface FastEthernet 5/1
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Altn	BLK	19	128.129	P2p
VLAN0002	Root	FWD	19	128.129	P2p

```
Catalyst_D#show spanning-tree interface FastEthernet 5/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Root	FWD	19	128.130	P2p
VLAN0002	Altn	BLK	19	128.130	P2p

## Details on the port-priority and vlan port-priority Commands

There are two ways to define VLAN port priority:

- The "global" port priority value (128 by default) which can be modified per interface by the **port-priority** command
- The "per VLAN" port priority value which can be modified per interface and per VLAN by the **VLAN port-priority** command

An example clarifies this. Consider adding a third VLAN in this example. By default, this VLAN belongs to the group that has the global port priority value (default 128).

```
Catalyst_R#show spanning-tree interface FastEthernet 3/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	64.130	P2p



VLAN0002	Desg FWD 19	128.130	P2p
VLAN0003	Desg FWD 19	128.130	P2p

Change the global priority for the port, using the **spanning-tree port-priority** interface configuration command:

```
Catalyst_R(config)#interface FastEthernet 3/2
Catalyst_R(config-if)#spanning-tree port-priority 160
Catalyst_R#
Catalyst_R#show spanning-tree interface FastEthernet 3/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	64.130	P2p
VLAN0002	Desg	FWD	19	160.130	P2p
VLAN0003	Desg	FWD	19	160.130	P2p

Notice that all the VLANs belonging to the "global" group changed their priority to 160. Now assign VLAN 3 its own priority 48 with the **spanning-tree vlan port-priority** interface command:

```
Catalyst_R(config)#interface FastEthernet 3/2
Catalyst_R(config-if)#spanning-tree vlan 3 port-priority 48
Catalyst_R#
Catalyst_R#show spanning-tree interface FastEthernet 3/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	64.130	P2p
VLAN0002	Desg	FWD	19	160.130	P2p
VLAN0003	Desg	FWD	19	48.130	P2p

Notice that only VLAN 3 has changed its priority to 48. To put VLAN 3 back in the default group, use the **no spanning-tree vlan port-priority** interface command:

```
Catalyst_R(config)#interface FastEthernet 3/2
Catalyst_R(config-if)#no spanning-tree vlan 3 port-priority
Catalyst_R#
Catalyst_R#show spanning-tree interface FastEthernet 3/2
```

Vlan	Role	Sts	Cost	Prio.Nbr	Type
VLAN0001	Desg	FWD	19	64.130	P2p
VLAN0002	Desg	FWD	19	160.130	P2p
VLAN0003	Desg	FWD	19	160.130	P2p

## Conclusion

The VLAN load balancing configuration just completed optimizes the use of redundant trunks between two Catalysts.

Keeping the default STP values leads all redundant links between the two Catalysts to end up in blocking mode. Tuning the STP priority allows several links to be used at the same time, for different VLANs. This increases the overall bandwidth available between the two devices. In case of failure of a link, the STP re-dispatches the VLANs to the remaining trunks as it reconverges.

The only drawback left with this design is that it can only load-balance traffic on a VLAN basis. If in the previous example, you had a 130 Mb/s traffic flowing through VLAN 1 and only 10 Mb/s traffic on VLAN 2, you still drop packets on VLAN 1, even though you have, in theory, 200 Mb/s between Catalyst R and Catalyst D. The EtherChanneling feature addresses this, by providing load balancing between several links on a packet basis. If your hardware supports it, use FastEtherchannel (or GigabitEtherChannel) rather than the

configuration described in this document.

## Related Information

- [Spanning Tree Protocol Support Page](#)
  - [Switches Product Support](#)
  - [LAN Switching Technology Support](#)
  - [Technical Support & Documentation – Cisco Systems](#)
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Aug 30, 2005

Document ID: 10555

---