Between Catalyst 4000, 5000, and 6000 Family Switches Using

Table of Contents

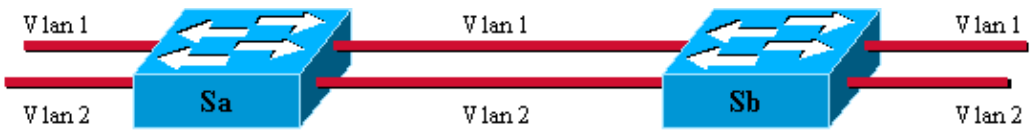
Trunking Between Catalyst 4000, 5000, and 6000 Family Switches Using 802.1q Encapsulation

Introduction

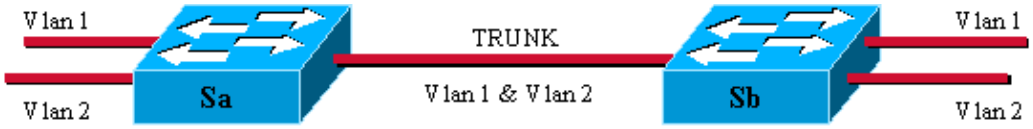
This document introduces the concept of trunking between two Ethernet switches and focuses on the IEEE 802.1q trunking standard. After a brief description of the 802.1q trunking mechanism, the implementation on the Catalyst 4000, 5000, and 6000 range of switches will be described.. A full example is provided, along with some common errors related to 802.1q trunking configuration.

What is a Trunk?

In Cisco's terminology, a trunk is a point−to−point link carrying several VLANs. The purpose of a trunk is to save ports when creating a link between two devices implementing VLANs, typically two switches. In the following diagram, we can see two VLANs that we want available on two switches, Sa and Sb. The first easy method to implement is to create two physical links between the devices, each one carrying the traffic for a VLAN:



Of course, this solution does not scale. If we wanted to add a third VLAN, we would need to sacrifice two additional ports. This design is also inefficient in terms of load sharing; the traffic on some VLANs may not justify a dedicated link. A trunk will bundle virtual links over one physical link, as shown in the next diagram:



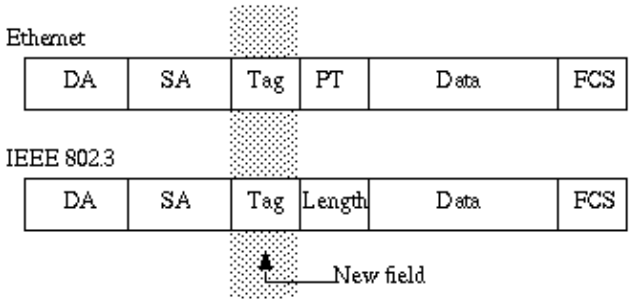
Here, the unique physical link between the two switches is able to carry traffic for any VLAN. In order to achieve this, each frame sent on the link is tagged by Sa so that Sb which VLAN it belongs to. Different tagging schemes exist. The most common for Ethernet segments are:

- ISL (this is the original Cisco proprietary InterSwitch Link protocol)
- 802.1q (the IEEE standard we will focus on in this document)

Basic Characteristics of 802.1q Trunking

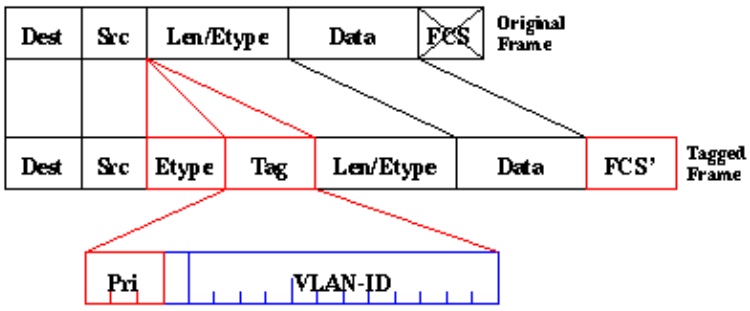
Tagging Mechanism

802.1q uses an internal tagging mechanism. Internal means that a tag is inserted within the frame (with ISL, the frame is encapsulated instead):



Note that on a 802.1q trunk, one VLAN is NOT tagged. This VLAN, named the native VLAN, must be configured the same on each side of the trunk. This way, we can deduce to which VLAN a frame belongs when we receive a frame with no tag.

The tagging mechanism implies a modification of the frame; the trunking device inserts a 4−byte tag and recomputes the frame check sequence (FCS):

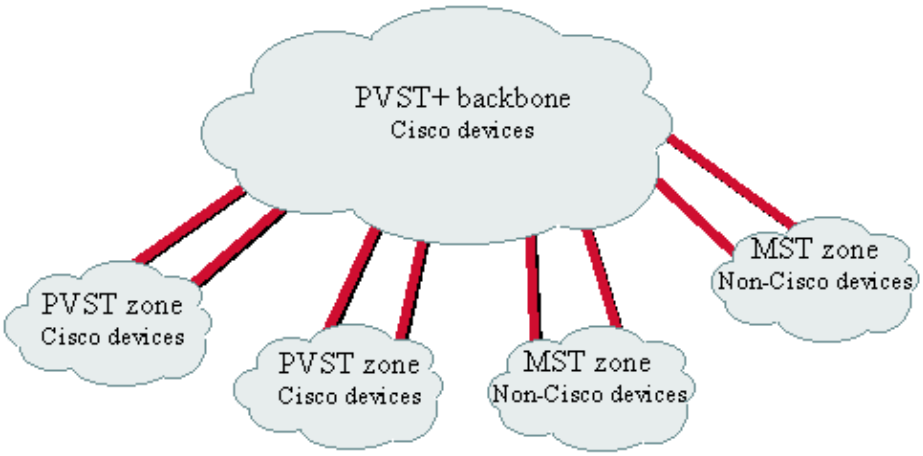


The EtherType field identifying the 802.1q frame is 0x8100. In addition to the 12−bit VLAN−ID, 3 bits are reserved for 802.1p priority tagging.

Also, note that inserting a tag into a frame that already has the maximum Ethernet size creates a 1522 byte frame that can be considered as a "baby giant" by the receiving equipment. The 802.3 committee is extending the maximum standard frame size to address this issue.

Spanning Tree Consideration

802.1q standard is more than just a tagging mechanism. It also defines a unique spanning tree instance running on the native VLAN for all the VLANs in the network. Such a Mono Spanning Tree(MST) network lacks some flexibility compared to a Per VLAN Spanning Tree network (PVST) that runs one instance of Spanning Tree Protocol (STP) per VLAN. Cisco developed PVST+ to allow running several STP instances (even over a 802.1q network) by using a tunneling mechanism. Although beyond the scope of this document, it can be briefly described as utilizing a Cisco device to connect a MST zone (typically another vendor's 802.1q−based network) to a PVST zone (typically a Cisco ISL based network). There is no specific configuration to enter in order to achieve this. Ideally, a mixed environment should look like the following diagram:



No direct trunk can be established between a MST and PVST zone.
There has to be a PVST+ zone in between.

Cisco's Implementation

In the current implementation, Cisco devices support only VLAN numbers up to 1005. This restriction, introduced to match the number of VLANs available with ISL, is allowed by the 802.1q standard. Cisco implemented a VLAN mapping feature in CatOS 5.1 to simplify interoperability with other vendors' devices, but it is seldom necessary.

Cisco also adapted its Dynamic ISL (DISL) protocol and turned it into Dynamic Trunking Protocol (DTP). DISL can negotiate ISL trunking on a link between two devices; DTP can, in addition, negotiate the type of trunking encapsulation (802.1q or ISL) that will be used as well. This is an interesting feature as some Cisco devices support only ISL or 802.1q, whereas some are able to run both.

In Cisco implementation, a trunk is a point−to−point link, although it is possible to use the 802.1q encapsulation on an Ethernet segment shared by more than two devices. Such a configuration is seldom needed but is still possible by disabling DTP negotiation.

Configuring 802.1q Trunks

Hardware/Software Requirements

>From a software point of view, the first appearance of 802.1q encapsulation was with 4.1 CatOS software. In this release, trunking configuration had to be hardcoded; DTP only appeared with CatOS 4.2. See the next section, dedicated to DTP.

Not all Catalyst ports support 802.1q encapsulation. Currently, while Catalyst 4000s only support 802.1q , ports of the Catalyst 6000 families are able to use 802.1q or ISL encapsulation. Depending on the module, Catalyst 5000 trunk capable ports are able to use 802.1q encapsulation, ISL encapsulation, or both. The best way to check this out is to use the show port capabilities command. The trunking capacity is explicitly stated:

```
Sa> (enable) sh port capa 1/1
Model                   WS-X5530
Port                    1/1
Type                    1000BaseSX
Speed                   1000
Duplex                  full
Trunk encap type        802.1Q,ISL
Trunk mode              on,off,desirable,auto,nonegotiate
Channel                 no
Broadcast suppression   percentage(0-100)
Flow control            receive-(off,on,desired),send-(off,on,desired)
Security                no
Membership              static
Fast start              yes
Rewrite                 no
```

DTP Modes

When configuring a port for trunking, two parameters can be set: the trunking mode and the encapsulation type (if DTP is supported on that port).

- The **trunking mode** defines how the port will negotiate the set up of a trunk with its peer port. Here is a list of the possible settings:

Trunking Mode	**DTP frames sent**	**Description**	**Final state** (local port)
on	YES, periodic	The local port advertises the remote it is going to the trunking state.	Trunking, unconditionally.
auto	YES, periodic	The local port advertises the remote it is able to trunk but does not request to go to the trunking state.	The port will end up in trunking state only if the remote wants to, that is, the remote mode is ***on*** or ***desirable***.
desirable	YES, periodic	The local port advertises the remote it is able to trunk and ask to go to the trunking state.	If the port detects that the remote is able to trunk (remote in **on**, **desirable** or **auto** mode), it will end up in trunking state, else will stay non-trunking.
nonegotiate	NO	Local port goes to unconditionally trunking, with no DTP notification	Trunking, unconditionally.

		to the remote.	
off	YES	Disable trunking on the port. DTP frames are only sent out when the port is transitioning to non−trunking.	Non trunking, unconditionally.

Be careful that some modes (*on*, *nonegotiate*,*off*) explicitly specify in which state the port will end up. A bad configuration can lead to a dangerous inconsistent state where one side is trunking and the other is not.

A port in *on*, *auto*, or *desirable* sends DTP frames periodically. A trunking port in *auto* or *desirable* goes back to non−trunking if it does not receive a DTP update from its neighbor in five minutes.

Note that if you are running 4.1 CatOS software, you will need to disable any form of negotiation by using the ***off*** or ***nonegotiate*** mode when configuring 802.1q trunking.

- The **encapsulation type** allows the user to specify whether 802.1q or ISL should be used when setting up the trunk. Of course, the parameter is only relevant if the module you are using is able to use both. The parameter can have three different values:

Encapsulation type	Description
ISL	Sets the port encapsulation to ISL.
dot1q	Sets the port encapsulation to 802.1q.
negotiate	This encapsulation is only available in **auto** or **desirable** trunking modes. • If the remote has a **negotiate** encapsulation type, the trunk will eventually be set up with ISL. • If the remote is configured for ISL or 802.1q or only able to do ISL or 802.1q, then the trunking encapsulation used will be the one of the remote port.

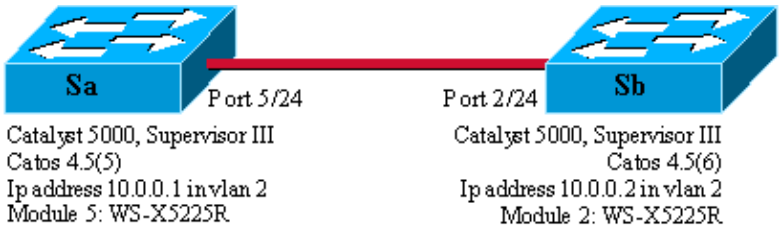
See Results of Possible Fast Ethernet and Gigabit Ethernet Trunk Configurations for a list of all the possible resulting configurations.

Note that no negotiation will take place between two switches in different VTP domain (VLAN Trunk Protocol).

Step−by−Step Example

Network Diagram

The following example is based on a very simple lab setup involving two Catalyst 5000s linked together via trunk capable ports. You need a cross−over cable in order to inter−connect two switches.



Minimal Setup of a 802.1q Trunk with Connectivity Tests

1. Check that the ports' status are up but not trunking
2. Set an IP address on the sc0 management interfaces
3. Check connectivity between Sa and Sb
4. Configure the same VTP domain on both switches
5. Create a VLAN 2 in each switch
6. Change the management interfaces to VLAN 2
7. Check if connectivity is broken between the two switches
8. Check the port capabilities
9. Configure the trunk encapsulation to be 802.1q
10. Verify the trunk is up
11. Check connectivity

Setting the Native VLAN

12. Use the **set vlan** command
13. Check the result

Specifying VLANs Allowed on the Trunk

14. Create additional VLANs
15. Removing VLANs from the trunk
16. Reactivate a VLAN

Step 1: Connect a terminal to the console of your switches. See the document Connecting a Terminal to the Catalyst 5000 if necessary.
First, check the status of the port involved in the setup. Use the command **show port 5/24** on Sa (**show port 2/24** on Sb) and check that the status is connected:

```
Sa> (enable) show port 5/24
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
 5/24                    connected  1          normal a-full a-100 10/100BaseTX

<snip>
```

We have default value for that kind of port. It came negotiating 100 MB full−duplex and it is assigned to VLAN 1. Entering a **show trunk 5/24** command clearly tells us that the port is not trunking and has a default mode auto and encapsulation negotiate.

```
Sa> (enable) show trunk 5/24
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 5/24     auto         negotiate      not-trunking  1
```

```
<snip>
```

Step 2 : Use the **set interface sc0 10.0.0.1** command on switch Sa and **set interface sc0 10.0.0.2** on switch Sb to assign an IP address to our two switches. A **show interface** confirms that the management interface is now correctly set in the default VLAN 1:

```
Sa> (enable) set interface sc0 10.0.0.1
Interface sc0 IP address set.

Sa> (enable) show interface
sl0: flags=51<UP,POINTOPOINT,RUNNING>
        slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
        vlan 1 inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
Sa> (enable)
```

If you have the output of a **show interface** command from your Cisco device, you can use to display potential issues and fixes. To use , you must be a registered user, be logged in, and have JavaScript enabled. You can use Output Interpreter to display potential issues and fixes. To use Output Interpreter, you must be a registered user, be logged in, and have JavaScript enabled.

Step 3 : A **ping 10.0.0.2** command from switch Sa proves that switch Sb can now be reached:

```
Sa> (enable) ping 10.0.0.2
10.0.0.2 is alive
Sa> (enable)
```

Step 4 : Now, assign the same VTP domain to both switches. As we saw, having the same VTP domain is mandatory to use DTP negotiation. Enter the **set vtp domain cisco** command on both switches to configure them with the domain name "cisco" :

```
Sa> (enable) set vtp domain cisco
VTP domain cisco modified
Sa> (enable)
```

Step 5: Enter the command **set vlan 2** on both switches in order to create the VLAN 2. If the switches were already linked by a trunk, we would only need to enter the command on one switch and the other would learn it automatically via VTP. As we don't have a trunk yet, there is no VTP communication between Sa and Sb:

```
Sa> (enable) set vlan 2
Vlan 2 configuration successful
Sa> (enable)
```

Step 6: We are now going to move the management interface of both switches into VLAN 2. This way, we are going to show that there is no communication between Sa and Sb before a trunk is established. Enter the command **set int sc0 2** on each switch to move the sc0 interface in VLAN 2. Issue a **show interface** to check that the command is effective:

```
Sa> (enable) set int sc0 2
Interface sc0 vlan set.
Sa> (enable) sh int
sl0: flags=51<UP,POINTOPOINT,RUNNING>
        slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
        vlan 2 inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
Sa> (enable)
```

Step 7: Now, the **ping 10.0.0.2** to Sb fails from Sa, proving that there is no connectivity in VLAN 2 between the switches:

```
Sa> (enable) ping 10.0.0.2
no answer from 10.0.0.2
Sa> (enable)
```

Step 8: Before we start configuring a trunk, we can check with the **show port capabilities** command that both ports are able to implement 802.1q trunking:

```
Sa> (enable) sho port capa 5/24
Model                   WS-X5225R
Port                    5/24
Type                    10/100BaseTX
Speed                   auto,10,100
Duplex                  half,full
Trunk encap type        802.1Q,ISL
Trunk mode              on,off,desirable,auto,nonegotiate
Channel                 5/23-24,5/21-24
Broadcast suppression   percentage(0-100)
Flow control            receive-(off,on),send-(off,on)
Security                yes
Membership              static,dynamic
Fast start              yes
Rewrite                 yes
Sa> (enable)
```

Step 9: Now the trunk on Sa must be configured. We have seen in **Step 1** that both ports were in the default trunking mode auto, encapsulation type negotiate. A combination auto−auto does not bring a trunk up. This is normal, each side is willing to become trunk, but will only do it if the remote requests it. Considering the default configuration:

- We just need to change the trunk mode to desirable on one side to bring the trunk up. This is because a port in desirable mode notifies its neighbor that it wants to go trunking. As the remote (in auto mode) goes to trunking if prompted to, this is enough to bring the trunk up.
- We also need to specify which encapsulation we want to use. This is because both ports are ISL capable and this encapsulation is chosen first when both ends are in *negotiate* mode.

The syntax of the command is: **set trunk <mod/port> [on | off | desirable | auto | nonegotiate] [vlan_range] [isl | dot1q | negotiate]**. Enter **set trunk 5/24 dot1q desirable** on switch Sa:

```
Sa> (enable) set trunk 5/24 dot1q desirable
Port(s) 5/24 trunk mode set to desirable.
Port(s) 5/24 trunk type set to dot1q.
1997 May 07 17:32:01 %DTP-5-TRUNKPORTON:Port 5/24 has become dot1q trunk
1997 May 07 17:32:02 %PAGP-5-PORTFROMSTP:Port 5/24 left bridge port 5/24
1997 May 07 17:32:13 %PAGP-5-PORTTOSTP:Port 5/24 joined bridge port 5/24
```

Step 10: The console log of the previous command clearly shows that the port moved to trunking, but we can also check by issuing a **show trunk 5/24** command on Sa and a **show trunk 2/24** command on Sb. You can see a subtle difference between the two outputs:

- The port on Sa is in desirable mode, whereas the Sb port is in auto mode.

- More interesting, the encapsulation is dot1q on Sa whereas it is **n−dot1q** on Sb. This is to show that Sb negotiated its encapsulation to dot1q. If we did not specify an encapsulation on Sa, both ports would have ended up in n−isl encapsulation:

```
Sa> (enable) show trunk 5/24
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 5/24     desirable    dot1q          trunking      1

Port      Vlans allowed on trunk
--------  ---------------------------------------------------------------------
 5/24     1-1005

Port      Vlans allowed and active in management domain
--------  ---------------------------------------------------------------------
 5/24     1-2

Port      Vlans in spanning tree forwarding state and not pruned
--------  ---------------------------------------------------------------------
 5/24     1-2
Sa> (enable)
Sb> (enable) sh trunk 2/24
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/24     auto         n-dot1q        trunking      1

<snip>
```

If you have the output of a **show trunk** command from your Cisco device, you can use to display potential issues and fixes. To use , you must be a registered user, be logged in, and have JavaScript enabled. You can use Output Interpreter to display potential issues and fixes. To use Output Interpreter, you must be a registered user, be logged in, and have JavaScript enabled.

Step 11: Now, we can check that VLAN 2 is now going through our trunk, simply pinging Sb from Sa:

```
Sa> (enable) ping 10.0.0.2
10.0.0.2 is alive
Sa> (enable)
```

Step 12: The command **set vlan 2 5/24** is used to assign a port to a specific VLAN. In the case of a trunking port, it changes the native VLAN to VLAN 2. Of course, we need to do the same on Sb with **set vlan 2 2/24** :

```
Sa> (enable) set vlan 2 5/24
VLAN 2 modified.
VLAN 1 modified.
VLAN  Mod/Ports
---- ---------------------
2     5/24

Sa> (enable)
```

Before we change the native VLAN on Sb, there is now an inconsistency between Sa and Sb configuration. The two ends of the trunk don't have the same native VLAN configuration. Here, some warning messages are displayed on Sb console. Note that the switch reporting the inconsistency may vary depending on which one is the root bridge for VLANs 1 and 2.

```
Sb> (enable) 2000 Dec 07 16:31:24 %SPANTREE-2-RX_1QPVIDERR: Rcved pvid_inc BPDU on 1Q port
vlan 1.
2000 Dec 07 16:31:24 %SPANTREE-2-TX_BLKPORTPVID: Block 2/24 on xmtting vlan 2 for inc peer
vlan.
2000 Dec 07 16:31:24 %SPANTREE-2-RX_BLKPORTPVID: Block 2/24 on rcving vlan 1 for inc peer v
2.
```

```
Sb> (enable)
Sb> (enable) set vlan 2 2/24
VLAN 2 modified.
VLAN 1 modified.
VLAN  Mod/Ports
----  ------------------------
2     2/24
Sb> (enable) 2000 Dec 07 16:31:46 %SPANTREE-2-PORTUNBLK: Unblock previously inc port 2/24
vlan 1.
2000 Dec 07 16:31:48 %SPANTREE-2-PORTUNBLK: Unblock previously inc port 2/24 on vlan 2.
```

The native VLAN mismatch has been corrected and everything goes back to normal.

Step 13: Now, let's simply check the result of these commands on our trunk, using the **show trunk 5/24** command:

```
Sa> (enable) sh trunk 5/24
 Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 5/24     desirable    dot1q          trunking      2

<snip>
```

Step 14: When you create a new trunk, it carries by default all the existing VLANs in the network. We are going to see how to restrict the list of allowed VLANs on a trunk. First, we need to create two additional VLANs (3 and 4), entering **set vlan 3** and **set vlan 4** commands on Sa for instance. It is only necessary to enter the command on one switch, VTP will propagate this information to the other.

Note: This part of the configuration is absolutely the same whether 802.1q or ISL encapsulation is used.

```
Sa> (enable) set vlan 3
Vlan 3 configuration successful
Sa> (enable) set vlan 4
Vlan 4 configuration successful
```

Step 15: The command **clear trunk <module/port> <vlan−list>** allows you to remove one or several VLANs from a given trunk. Here, the four VLANs we created were defined on our trunk. Remove VLAN 2 and VLAN 3 using the commands **clear trunk 5/24 2−3** on Sa, and **clear trunk 2/24 2−3** on Sb. You can check the result of the clear command using the **show trunk 5/24** command. Only VLANs 1 and 4 are now crossing the trunk between Sa and Sa. A ping between Sa and Sb will then now fail:

```
Sa> (enable) clear trunk 5/24 2−3
Removing Vlan(s) 2−3 from allowed list.
Port 5/24 allowed vlans modified to 1,4−1005.
Sa> (enable) show trunk 5/24
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 5/24     desirable    dot1q          trunking      2

Port      Vlans allowed on trunk
--------  ---------------------------------------------------------------------
 5/24     1,4−1005

Port      Vlans allowed and active in management domain
--------  ---------------------------------------------------------------------
 5/24     1,4

Port      Vlans in spanning tree forwarding state and not pruned
```

```
--------  ---------------------------------------------------------------------
 5/24     1,4
```

Step 16: To add a VLAN back on a trunk, use the **set trunk <module/port> <vlan−list>** command:

```
Sa> (enable) set trunk 5/24 2
Adding vlans 2 to allowed list.
Port(s) 5/24 allowed vlans modified to 1−2,4−1005.
Sa> (enable) show trunk
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 5/24     desirable    dot1q          trunking      2

Port      Vlans allowed on trunk
--------  ---------------------------------------------------------------------
 5/24     1−2,4−1005

Port      Vlans allowed and active in management domain
--------  ---------------------------------------------------------------------
 5/24     1−2,4

Port      Vlans in spanning tree forwarding state and not pruned
--------  ---------------------------------------------------------------------
 5/24     1−2,4
```

VLAN 2 is now flowing again on our trunk (ping Sa to Sb possible).

Common Errors

Different Native VLANs

This is a frequent configuration error. The native VLAN configured on each end of a 802.1q trunk must be the same. Remember that a switch receiving a non−tagged frame will assign it to the native VLAN of the trunk. If one end is configured for native VLAN 1 and the other to native VLAN 2, a frame sent in VLAN 1 on one side will be received on VLAN 2 on the other. You are then merging VLAN 1 and 2. There is no reason why you would want that and it may imply some connectivity issues in your network.

A Cisco device will usually warn you on a native VLAN mismatch. See **Step 12** for the kind of error messages you will get on the console in this case. Always check that the native VLAN is the same on your switches' trunk configuration.

Different VTP Domains

When you create a trunk between two switches and you are using DTP negotiation, double check that the VTP domain configured on both switches is the same. Negotiation will not take place between two switches that are in different VTP domains. In the following example, we took the working trunking configuration described above:

- Sa in trunking mode desirable, encapsulation dot1q.
- Sb in trunking mode auto, encapsulation negotiate.
- Same native VLAN, same VLANs allowed on each side.

The only difference is that we assigned VTP domain "**c**" on Sa and VTP domain "**cisco**" on Sb:

```
Sa> (enable) sh trunk
No ports trunking.
```

```
Sa> (enable) sh trunk 5/24
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 5/24     desirable    dot1q          not-trunking  1

Port      Vlans allowed on trunk
--------  --------------------------------------------------------------------
 5/24     1-1005

Port      Vlans allowed and active in management domain
--------  --------------------------------------------------------------------
 5/24     1

Port      Vlans in spanning tree forwarding state and not pruned
--------  --------------------------------------------------------------------
 5/24


Sb> (enable) sh trunk
No ports trunking.
Sb> (enable) sh trunk 2/24
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/24     auto         negotiate      not-trunking  1

Port      Vlans allowed on trunk
--------  --------------------------------------------------------------------
 2/24     1-1005

Port      Vlans allowed and active in management domain
--------  --------------------------------------------------------------------
 2/24     1

Port      Vlans in spanning tree forwarding state and not pruned
--------  --------------------------------------------------------------------
 2/24
Sb> (enable)
```

We can see that the trunk did not come up. When you are seeing that kind of issue, check the VTP domain configured on the switches using the **show vtp domain** command:

```
Sa> (enable) sh vtp domain
Domain Name                    Domain Index VTP Version Local Mode  Password
------------------------------ ------------ ----------- ----------- ----------
c                              1            2           server      -

Vlan-count Max-vlan-storage Config Revision Notifications
---------- ---------------- --------------- -------------
8          1023             0               disabled

Last Updater    V2 Mode  Pruning  PruneEligible on Vlans
--------------- -------- -------- -------------------------
10.0.0.1        disabled disabled 2-1000


Sb> (enable) sh vtp domain
Domain Name                    Domain Index VTP Version Local Mode  Password
------------------------------ ------------ ----------- ----------- ----------
cisco                          1            2           server      -

Vlan-count Max-vlan-storage Config Revision Notifications
---------- ---------------- --------------- -------------
8          1023             20              disabled
```

```
Last Updater    V2 Mode  Pruning  PruneEligible on Vlans
--------------- -------- -------- -------------------------
10.0.0.1        disabled disabled 2-1000
```

Now, we will put switch Sa in VTP domain "cisco", using the **set vtp domain cisco** command. After a few seconds, the trunk is negotiated and up again:

```
Sa> (enable) set vtp domain cisco
VTP domain cisco modified
Sa> (enable) 1997 May 13 13:59:22 %DTP-5-TRUNKPORTON:Port 5/24 has become dot1q trunk
1997 May 13 13:59:22 %PAGP-5-PORTFROMSTP:Port 5/24 left bridge port 5/24
1997 May 13 13:59:33 %PAGP-5-PORTTOSTP:Port 5/24 joined bridge port 5/24
```

If you want to keep different VTP domains, but still create a trunk between two switches, then you have to hardcode trunking on each side of the trunk (using nonegotiate/on).

Trunking Mode Incompatible with the Encapsulation Type

This is a common issue that started to be raised to the Technical Assistance Center (TAC) when the first modules able to support both 802.1q and ISL shipped. People were used to configuring a trunk by entering **set trunk <*mod/port*> on**, or **set trunk <*mod/port*> nonegotiate**. The problem is that by default the encapsulation type is set to `negotiate`. The negotiate encapsulation type is only supported by `auto` or `desirable` trunking modes. The `on` and `nonegotiate` encapsulation types do not perform any negotiations between switches and must be hard set to ISL or 802.1q encapsulation when they are configured. The following is a log of what is happening on the switch in that case:

```
Sa> (enable) set trunk 5/24 on
Failed to set port 5/24 to trunk mode on.
Trunk mode 'on' not allowed with trunk encapsulation type 'negotiate'.
Sa> (enable) set trunk 5/24 noneg
Failed to set port 5/24 to trunk mode nonegotiate.
Trunk mode 'nonegotiate' not allowed with trunk encapsulation type
'negotiate'.
Sa> (enable)
```

This makes sense because if you don't negotiate with the remote, how would you know which kind of encapsulation (802.1q or ISL) to use in order to bring up the trunk? Two possibilities:

- Use the desirable mode. In this case, you will negotiate the encapsulation mode with the remote:

```
Sa> (enable) set trunk 5/24 desi
Port(s) 5/24 trunk mode set to desirable.
Sa> (enable) 1997 May 09 17:49:19 %DTP-5-TRUNKPORTON:Port 5/24 has become isl trunk
```

- Specify the encapsulation you want to use:

```
Sa> (enable) set trunk 5/24 isl on
Port(s) 5/24 trunk mode set to on.
Port(s) 5/24 trunk type set to isl.
Sa> (enable) 1997 May 09 17:50:16 %DTP-5-TRUNKPORTON:Port 5/24 has become isl trunk
```

Commands Used in the Document

Command Summary

- **ping**
- **set interface**
- **set trunk**
- **set vlan**
- **set vtp domain**
- **show interface**
- **show port**
- **show port capabilities**
- **show trunk**
- **show vtp domain**

Tools Information

For additional resources, refer to Cisco TAC Tools for LAN Technologies.

Related Information

- **LAN Technologies Top Issues**
- **LAN Technologies Technical Tips**
- **ISL Trunking on Catalyst 5000 and 6000 Family Switches**
- **Configuring Trunks (Catalyst 5000 Cisco Documentation)**
- **Understanding and Configuring VLAN Trunk Protocol (VTP)**

Updated: Jun 13, 2002 Document ID: 14970