

## - VLANs and VTP -

### Review of Collision vs. Broadcast Domains

In a previous guide, it was explained that a “collision domain” is a segment where a collision can occur, and that a Layer-2 switch running in Full Duplex breaks up collision domains. Thus, Layer-2 switches create **more collision domains**, which results in **fewer collisions**.

However, Layer-2 switches do not break up broadcast domains, and thus belong to only **one broadcast domain**. Layer-2 switches will forward a broadcast or multicast out every port, excluding the port the broadcast or multicast originated from.

Only Layer-3 devices can break apart broadcast domains. Because of this, Layer-2 switches are not well suited for large, scalable networks. Layer-2 switches make forwarding decisions solely based on Data-Link layer MAC addresses, and thus have no way of differentiating between one *network* and another.

### Virtual LANs (VLANs)

**Virtual LANs (or VLANs)** separate a Layer-2 switch into **multiple broadcast domains**. Each VLAN is its own individual broadcast domain (i.e. IP subnet).

Individual ports or groups of ports can be assigned to a specific VLAN. Only ports belonging to the same VLAN can freely communicate; ports assigned to separate VLANs require a router to communicate. Broadcasts from one VLAN will never be sent out ports belonging to another VLAN.

Please note: a Layer-2 switch that supports VLANs *is not* necessarily a Layer-3 switch. A Layer-3 switch, in addition to supporting VLANs, must also be capable of routing, and caching IP traffic flows. Layer-3 switches allow IP packets to be **switched** as opposed to **routed**, which reduces latency.

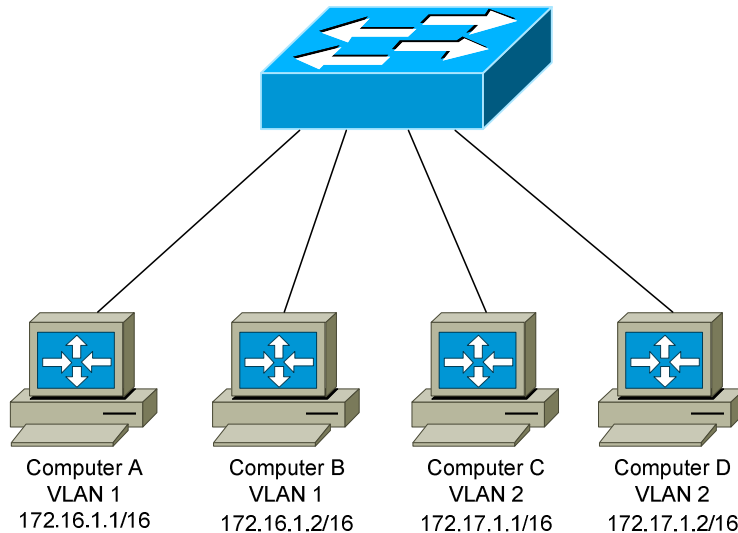
\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VLAN Example

Consider the following example:



Four computers are connected to a Layer-2 switch that supports VLANs. Computers A and B belong to VLAN 1, and Computers C and D belong to VLAN 2.

Because Computers A and B belong to the same VLAN, they belong to the same IP subnet and broadcast domain. They will be able to communicate without the need of a router.

Computers C and D likewise belong to the same VLAN and IP subnet. They also can communicate without a router.

However, Computers A and B will *not* be able to communicate with Computers C and D, as they belong to separate VLANs, and thus separate IP subnets. Broadcasts from VLAN 1 will never go out ports configured for VLAN 2. A router will be necessary for both VLANs to communicate.

Most Catalyst multi-layer switches have integrated or modular routing processors. Otherwise, an external router is required for inter-VLAN communication.

By default on Cisco Catalyst switches, all interfaces belong to **VLAN 1**. VLAN 1 is considered the **Management VLAN** (by default).

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Advantages of VLANs

VLANs provide the following advantages:

**Broadcast Control** – In a pure Layer-2 environment, broadcasts are received by every host on the switched network. In contrast, each VLAN belongs to its own broadcast domain (or IP subnet); thus broadcast traffic from one VLAN will never reach another VLAN.

**Security** – VLANs allow administrators to “logically” separate users and departments.

**Flexibility and Scalability** – VLANs remove the physical boundaries of a network. Users and devices can be added or moved anywhere on the physical network, and yet remain assigned to the same VLAN. Thus, access to resources will never be interrupted.

## VLAN Membership

VLAN membership can be configured one of two ways:

- **Statically** – Individual (or groups of) switch-ports must be manually assigned to a VLAN. Any device connecting to that switch-port(s) becomes a member of that VLAN. This is a transparent process – the client device is unaware that it belongs to a specific VLAN.
- **Dynamically** – Devices are automatically assigned into a VLAN based on its MAC address. This allows a client device to remain in the same VLAN, regardless of which switch port the device is attached to.

Cisco developed a dynamic VLAN product called the **VLAN Membership Policy Server (VMPS)**. In more sophisticated systems, a user’s network account can be used to determine VLAN membership, instead of a device’s MAC address.

Catalyst switches that *participate in a VTP domain* (explained shortly) support up to **1005 VLANs**. Catalyst switches configured in *VTP transparent mode* support up to **4094 VLANs**.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Static VLAN Configuration

The first step in configuring VLANs is to **create** the VLAN:

```
Switch(config)# vlan 100
Switch(config-vlan)# name MY_VLAN
```

The first command creates VLAN 100, and enters VLAN configuration mode. The second command assigns the name *MY\_VLAN* to this VLAN. Naming a VLAN is *not* required.

The list of VLANs is stored in Flash in a database file named **vlan.dat**. However, information concerning which local interfaces are assigned to a specific VLAN is not stored in this file; this information is instead stored in the **startup-config** file of each switch.

Next, an interface (or range of interfaces) must be assigned to this VLAN. The following commands will assign interface *fa0/10* into the newly created *MY\_VLAN*.

```
Switch(config)# interface fa0/10
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 100
```

The first command enters interface configuration mode. The second command indicates that this is an *access* port, as opposed to a *trunk* port (explained in detail shortly). The third command assigns this access port to VLAN 100. Note that the VLAN *number* is specified, and not the VLAN *name*.

To view the list of VLANs, including which ports are assigned to each VLAN:

```
Switch# show vlan
```

VLAN	Name	Status	Ports
1	default	active	fa0/1-9, 11-24
100	MY_VLAN	active	fa0/10
1002	fddi-default	suspended	
1003	token-ring-default	suspended	
1004	fddinet-default	suspended	
1005	trnet-default	suspended	

\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VLAN Port “Types”

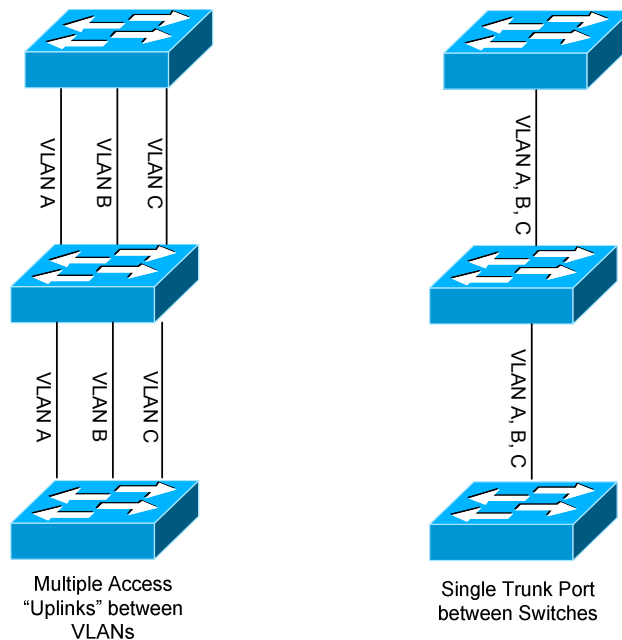
There are two types of ports supported on a VLAN-enabled switch, **access ports** and **trunk ports**.

An **access port** belongs to only one VLAN. Host devices, such as computers and printers, plug into access ports. A host automatically becomes a member of its access port’s VLAN. This is done transparently, and the host is usually unaware of the VLAN infrastructure. By default, all switch ports are access ports.

VLANs can span multiple switches. There are two methods of connecting these VLANs together. The first requires creating “uplink” access ports between all switches, for each VLAN. Obviously, in large switching and VLAN environments, this *quickly* becomes unfeasible.

A better alternative is to use **trunk ports**. Trunk ports do not belong to a single VLAN. Any or all VLANs can traverse trunk links to reach other switches. Only Fast or Gigabit Ethernet ports can be used as trunk links.

The following diagram illustrates the advantage of using trunk ports, as opposed to uplinking access ports:



\* \* \*

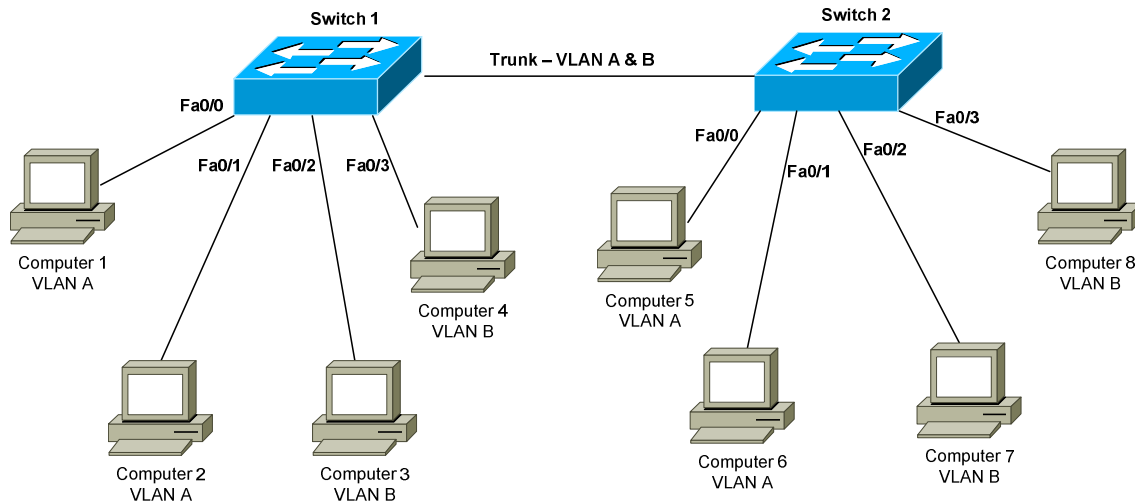
All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VLAN Frame-Tagging

When utilizing trunk links, switches need a mechanism to identify which VLAN a particular frame belongs to. **Frame tagging** places a VLAN ID in each frame, identifying which VLAN the frame belongs to.

Tagging occurs *only* when a frame is **sent out a trunk port**. Consider the following example:



If Computer 1 sends a frame to Computer 2, no frame tagging will occur. The frame never leaves the Switch 1, stays within its own VLAN, and will simply be **switched** to Computer 2.

If Computer 1 sends a frame to Computer 3, which is in a separate VLAN, frame tagging will *still* not occur. Again, the frame never leaves the switch, but because Computer 3 is in a different VLAN, the frame must be **routed**.

If Computer 1 sends a frame to Computer 5, the frame *must* be **tagged** before it is sent out the trunk port. It is stamped with its VLAN ID (in this case, VLAN A), and when Switch 2 receives the frame, it will only forward it out ports belonging to VLAN A (fa0/0, and fa0/1). If Switch 2 has Computer 5's MAC address in its CAM table, it will only send it out the appropriate port (fa0/0).

Cisco switches support two frame-tagging protocols, **Inter-Switch Link (ISL)** and **IEEE 802.1Q**.

\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

### **Inter-Switch Link (ISL)**

ISL is Cisco's proprietary frame-tagging protocol, and supports Ethernet, Token Ring, FDDI, and ATM frames.

ISL *encapsulates* a frame with an additional header (**26 bytes**) and trailer (**4 bytes**), increasing the size of an Ethernet frame up to **30 bytes**. The header contains the 10 byte VLAN ID. The trailer contains an additional 4-byte CRC for data-integrity purposes.

Because ISL increases the size of a frame, non-ISL devices (i.e. non-Cisco devices) will actually *drop* ISL-tagged frames. Many devices are configured with a maximum acceptable size for Ethernet frames (usually 1514 or 1518 bytes). ISL frames can be as large as 1544 bytes; thus, non-ISL devices will see these packets as **giants** (or corrupted packets).

ISL has deprecated in use over time. Newer Catalyst models may not support ISL tagging.

### **IEEE 802.1Q**

IEEE 802.1Q, otherwise known as DOT1Q, is the standardized frame-tagging protocol supported by most switch manufacturers, including Cisco. Thus, switches from multiple vendors can be trunked together.

Instead of adding an additional header and trailer, 802.1Q actually embeds a **4-byte VLAN ID** into the **Layer-2 frame header**. This still increases the size of a frame from its usual 1514 bytes to 1518 bytes (or from 1518 bytes to 1522 bytes). However, most modern switches support 802.1Q tagging and the slight increase in frame size.

Neither ISL nor 802.1Q tagging alter the source or destination address in the Layer-2 header.

### **Manual vs. Dynamic Trunking**

ISL or 802.1Q tagging can be manually configured on Catalyst trunk ports. Catalyst switches can also dynamically negotiate this using Cisco's proprietary **Dynamic Trunking Protocol (DTP)**.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Configuring Trunk Links

To manually configure a trunk port, for either ISL or 802.1Q tagging:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk

Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
```

The first line in each set of commands enters interface configuration mode. The second line manually sets the tagging (or *encapsulation*) protocol the trunk link will use. **Always remember**, both sides of the trunk line must be configured with the **same** tagging protocol. The third line manually sets the *switchport mode* to a trunk port.

The Catalyst switch can **negotiate** the tagging protocol:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk encapsulation negotiate
```

Whichever tagging protocol is supported on both switches will be used. If the switches support both ISL and 802.1Q, **ISL** will be selected.

By default, trunk ports allow **all VLANs** to traverse the trunk link. However, a list of *allowed* VLANs can be configured on each trunk port:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk allowed vlan remove 50-100
Switch(config-if)# switchport trunk allowed vlan add 60-65
```

The first *switchport* command will prevent the trunk port from passing traffic from VLANs 50-100. The second *switchport* command will re-allow the trunk port to pass traffic from VLANs 60-65. In both cases, the *switchport trunk allowed* commands are adding/subtracting from the current list of allowed VLANs, and not replacing that list.

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk allowed vlan all
Switch(config-if)# switchport trunk allowed vlan except 2-99
```

Certain VLANs are reserved and cannot be removed from a trunk link, including **VLAN 1** and system **VLANs 1002-1005**.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Native VLANs

A **native VLAN** can also be configured on trunk ports:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan 42
```

Frames from the native VLAN are **not tagged** when sent out trunk ports. A trunking interface can only be assigned one native VLAN. **Only 802.1Q supports native VLANs**, whereas ISL does not. (More accurately, ISL will tag frames from all VLANs, even if a VLAN is configured as native). The native VLAN should be configured **identically on both sides of the 802.1Q trunk**).

Native VLANs are often configured when plugging Cisco VoIP phones into a Catalyst Switch (beyond the scope of this section). Native VLANs are also useful if a **trunk port fails**. For example, if an end user connects a computer into a trunk port, the trunking status will fail and the interface will essentially become an access port. The user's computer will then be transparently *joined* to the Native VLAN.

Native VLANs provide another benefit. A trunk port will accept *untagged* frames and place them in the Native VLAN. Consider the following example:



Assume that both 802.1Q switches have trunk links configured to the non-802.1Q switch, and that the trunk ports are configured in Native VLAN 42. Not only will the 802.1Q switches be able to communicate with each other, the non-802.1Q switch will be placed in Native VLAN 42, and be able to communicate with any device in VLAN 42 on any switch.

(Please note, that the author of this study guide finds the “benefit” of the above example of Native VLANs to be.....*dubious* at best, and confusing as hell at worst).

By default on all trunking interfaces, the Native VLAN is **VLAN 1**.

\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Dynamic Trunking Protocol (DTP) Configuration

Not only can the frame tagging protocol of a trunk port be auto-negotiated, but whether a port actually *becomes* a trunk can be negotiated dynamically as well using the **Dynamic Trunking Protocol (DTP)**.

To manually set a port to be a trunk:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport mode trunk
```

To allow a port to *dynamically* decide whether to become a trunk, there are two options:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport mode dynamic desirable
```

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport mode dynamic auto
```

If a switchport is set to **dynamic desirable** (the *default* dynamic setting), the interface will actively attempt to form a trunk with the remote switch. If a switchport is set to **dynamic auto**, the interface will passively wait for the remote switch to initiate the trunk.

This results in the following:

- If both ports are manually set to **trunk** - a trunk will form.
- If one port is set to **dynamic desirable**, and the other is set to manual **trunk, dynamic desirable, or dynamic auto** - a trunk will form.
- If one port is set to **dynamic auto**, and the other port is set to manual **trunk or dynamic desirable** - a trunk will form.
- If both ports are set to **dynamic auto**, the link will *never* become a trunk, as both ports are waiting for the other to initialize the trunk.

Trunk ports send out DTP frames **every 30 seconds** to indicate their configured *mode*.

In general, it is best to manually specific the trunk link, and disable DTP using the *switchport nonegotiate* command:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport nonegotiate
```

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Troubleshooting Trunks

When troubleshooting a misbehaving trunk link, ensure that the following is configured identically on both sides of the trunk:

- **Mode** - both sides must be set to *trunk* or *dynamically negotiated*
- **Frame-tagging protocol** - ISL, 802.1Q, or dynamically negotiated
- **Native VLAN**
- **VTP Domain**
- **Allowed VLANs**

If the above parameters are not set identically on both sides, the trunk link will never become active.

To view whether a port is an access or trunk port (such as fa0/5):

```
Switch# show interface fa0/24 switchport
```

```
Name: Fa0/24
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 42

<snip>
```

To view the status of all trunk links:

```
Switch# show interface trunk
```

```
Port      Mode      Encapsulation      Status      Native VLAN
Fa0/24    on        802.1q              trunking    42

Port      Vlans allowed on trunk
Fa0/24    1,100-4094

Port      Vlans allowed and active in management domain
Fa0/24    1,100

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/24    1,100
```

If no interfaces are in a trunking state, the *show interface trunk* command will return no output.

\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VLAN Trunking Protocol (VTP)

In large switching environments, it can become difficult to maintain a consistent VLAN database across all switches on the network. The Cisco-proprietary **VLAN Trunking Protocol (VTP)** allows the VLAN database to be easily managed throughout the network.

Switches configured with VTP are joined to a **VTP domain**. Only switches belonging to the same domain will share VLAN information, and a switch can only belong to a single domain. When an update is made to the VLAN database, this information is propagated to all switches via **VTP advertisements**.

By default, VTP updates are sent out every **300 seconds**, or anytime a **change to the database** occurs. VTP updates are sent across **VLAN 1**, and are only sent out trunk ports.

There are *three* versions of VTP. The key additions provided by **VTP Version 2** are support for Token Ring and Consistency Checks.

VTP Version 1 is default on Catalyst switches, and is **not compatible** with VTP Version 2.

Cisco describes **VTP Version 3** as such: “VTP version 3 differs from earlier VTP versions in that it does not directly handle VLANs. VTP version 3 is a protocol that is only responsible for distributing a list of opaque databases over an administrative domain.”

(If you are confused, don't be alarmed. The author of this guide is not certain what that means either).

Cisco further defines the enhancements that VTP version 3 provides:

- Support for extended VLANs
- Support for the creation and advertising of private VLANs
- Support for VLAN instances and MST mapping propagation instances
- Improved server authentication
- Protection from the “wrong” database accidentally being inserted into a VTP domain.
- Interaction with VTP version 1 and VTP version 2
- Ability to be configured on a per-port basis.

(Reference: [http://www.cisco.com/en/US/tech/tk389/tk689/technologies\\_tech\\_note09186a0080094c52.shtml](http://www.cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a0080094c52.shtml),  
<http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/catos/8.x/configuration/guide/vtp.html#wp1017196>)

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VTP Modes

VTP-enabled switches can operate in one of three **modes**:

- **Server**
- **Client**
- **Transparent**

Only **VTP Servers** can create, modify or delete entries in the shared VLAN database. Servers advertise their VLAN database to all other switches on the network, including other VTP servers. This is the **default mode** for Cisco Catalyst switches. VTP servers can only advertise VLANs 1 - 1005.

**VTP Clients** cannot make modifications to the VLAN database, and will receive all of their VLAN information from VTP servers. A client will also forward an update from a server to other clients out its trunk port(s).

Remember, VTP switches must be in the **same VTP Domain** to share/accept updates to the VLAN database.

A **VTP Transparent** switch maintains its own separate VLAN database, and will neither advertise nor accept any VLAN database information from other switches, even a server. However, transparent switches will forward VTP updates from servers to clients, thus acting as a *pass-through*.

Transparent switches handle this pass-through differently depending on the VTP version:

- **VTP Version 1** – the transparent switch will only pass updates from the *same* VTP domain.
- **VTP Version 2** – the transparent switch will pass updates from *any* VTP domain.

As a best practice, a new switch should be configured as a VTP *client* in the VTP domain, and have its **configuration revision number** (described in the next section) set back to *zero* before being installed into a production network.

There is a specific reason for this: if by some circumstance a new switch's configuration revision number is *higher* than that of the existing production switches, a new VTP switch could conceivably advertise a *blank* or *incorrect* VLAN database to all other switches. This could result in a significant network outage.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VTP Updates

VTP updates contain a 32-bit **configuration revision number**, to ensure that all devices have the most current VLAN database. Every change to the VLAN database increments the configuration revision number by 1.

A VTP switch will only accept or synchronize an update if the revision number is *higher* (and thus more recent) than that of the currently installed VLAN database. This is true *even if the advertising switch is a VTP Client*. Updates with a *lower* revision number are ignored.

**REMEMBER:** a VTP client can update other clients *and* VTP servers in the VTP domain, if its revision number is higher.

The simplest way to reset the configuration revision on a VTP switch is to change the VTP domain name, and then change it back to the original name.

VTP utilizes three **message types**:

- **Summary Advertisement** – sent out every 300 seconds, informing all VTP switches of the current configuration revision number.
- **Subset Advertisement** – sent out when there is a change to the VLAN database. The subset advertisement actually contains the updated VLAN database.
- **Advertisement Request** – sent out when a switch requires the most current copy of the VLAN database. A switch that is newly joined to the VTP domain will send out an Advertisement Request.

A switch will also send out an Advertisement Request if it receives a Summary Advertisement with a configuration revision number higher than its current VLAN database. A Subset Advertisement will then be sent to that switch, so that it can synchronize the latest VLAN database.

A Subset Advertisement will contain the following fields:

- VTP Version
- VTP Domain
- VTP Configuration Revision
- VLAN IDs for each VLAN in the database
- VLAN-specific information, such as the VLAN name and MTU

(Reference: [http://www.cisco.com/en/US/tech/tk389/tk689/technologies\\_tech\\_note09186a0080094c52.shtml](http://www.cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a0080094c52.shtml))

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Configuring VTP

To configure the VTP *domain* (the domain name is case sensitive):

```
Switch(config)# vtp domain MYDOMAIN
```

To configure the VTP *mode*:

```
Switch(config)# vtp mode server
Switch(config)# vtp mode client
Switch(config)# vtp mode transparent
```

The VTP domain can be further secured using a password:

```
Switch(config)# vtp password PASSWORD
```

All switches participating in the VTP domain must be configured with the same password. The password will be hashed into a 16-byte MD5 value.

**By default**, a Catalyst switch uses **VTP version 1**. VTP Version 1 and 2 are not compatible. If applied on a VTP server, the following command will enable VTP version 2 globally on all switches:

```
Switch(config)# vtp version 2
```

To view status information about VTP:

```
Switch# show vtp status
```

```
VTP Version                : 2
Configuration Revision     : 42
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 7
VTP Operating Mode         : Server
VTP Domain Name            : MYDOMAIN
VTP Pruning Mode           : Disabled
VTP V2 Mode                 : Enabled
VTP Traps Generation       : Disabled
MD5 digest                  : 0x42 0x51 0x69 0xBA 0xBE 0xFA 0xCE 0x34
Configuration last modified by 0.0.0.0 at 3-12-09 4:07:52
```

To view VTP statistical information and error counters:

```
Switch# show vtp counters
```

\* \* \*

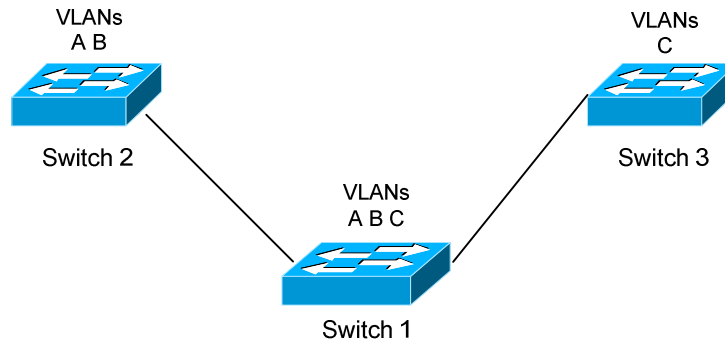
All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## VTP Pruning

**VTP pruning** is a process of preventing unnecessary VLAN broadcast or multicast traffic throughout the switching infrastructure.

In the following example, VTP pruning would prevent VLAN C broadcasts from being sent to Switch 2. Pruning would further prevent VLAN A and B broadcast traffic from being sent to Switch 3.



With VTP pruning, traffic is only sent out the necessary VLAN trunk ports where those VLANs exist.

VTP pruning is **disabled by default** on Catalyst IOS switches. If applied on a VTP server, the following command will enable VTP pruning globally on all switches:

```
Switch(config)# vtp pruning
```

On trunk ports, it is possible to specify which VLANs are *pruning eligible*:

```
Switch(config)# interface fa0/24
```

```
Switch(config-if)# switchport trunk pruning vlan add 2-50
```

```
Switch(config-if)# switchport trunk pruning vlan remove 50-100
```

```
Switch(config)# interface fa0/24
```

```
Switch(config-if)# switchport trunk pruning vlan all
```

```
Switch(config-if)# switchport trunk pruning vlan except 2-100
```

VLAN 1 is *never* eligible for pruning. The system VLANs 1002-1005 are also *pruning-ineligible*.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.