

2nd Edition



CISCO IOS

IN A NUTSHELL

*A Desktop Quick Reference
for IOS on IP Networks*

O'REILLY®

James Boney



14

Switches and VLANs

Switches are enhanced versions of bridges. Bridges were introduced in the mid-1980s to improve network performance. They solved a basic network problem: reducing network collisions by segmenting networks. On an Ethernet segment, only one machine may transmit at once. If more than one machine tries to communicate, a collision occurs on the segment. When a collision occurs, the machines that were trying to communicate go into a random wait period before attempting to transmit again. As the number of devices on a segment increases, so does the number of collisions. And the more collisions on a segment, the worse the network performs. By using a bridge, a network is separated into segments called *collision domains*, which reduce the number of devices—and collisions—per segment.

Switches improve on bridges in one important way; switches allow the network to be partitioned into logical smaller segments called Virtual LANs or VLANs. These VLANs allow you to create even smaller domains, decreasing the likelihood of collisions and improving network performance.

Bridges and switches act almost the same way when it comes to learning MAC addresses and forwarding packets based on those addresses. Both switches and bridges implement a loop prevention protocol called *spanning tree*, described later in this chapter.

When switches were first introduced, the companies selling switches announced the death of the router! We now know that switches and routers must work together in a modern network, a fact that is confirmed by Cisco's continued purchase and development of switching technology. This chapter describes Cisco switches, with an emphasis on IOS-enabled switches.

Switch Terminology

Before we delve into the details on switches, we need to define some basic terminology.

Layer-2 and Layer-3 Switching

Most of this chapter focuses on the functionality of a layer-2 switch, a switch that operates at the Data Link layer of the OSI networking model (see the appendix for more details on the OSI model). In other words, it switches frames at the MAC address level.

However, a new switch technology is coming to its own, layer 3 switching. From the name, it's easy to guess that these switches operate at the Networking layer of the OSI model, which means it switches based on IP address. As you can see, the line between routers and switches is becoming even less defined. However, you must remember that just because a switch can operate at layer 3 and above, that doesn't make it a router. In most cases, it means that the switch can do more advanced things, like filtering based on IP access lists. However, as we said earlier, Cisco has been releasing new devices that have both routing and switching capabilities; the lines between these devices are indeed blurring.

Learning MAC Addresses

In order to improve network performance, a switch needs to discover which hosts are connected to each port. Once it has that information, it can send the traffic for a specific host out only one interface instead of clogging the rest of the ports with unnecessary traffic. In other words, the switch sends traffic only to the host that needs it.

In order to do so, the switch must learn which port a host is on. It does this by taking the source MAC address of incoming packets. As the switch learns new MAC addresses, it adds them to the address table, which you can view with the command `show mac-address-table`.

When a switch doesn't know which port a host is on, it sends the traffic out all its active ports. It continues flooding traffic out all ports until the host finally replies. At that point, the switch can add the host's MAC address to its port table. Once the host is in the table, all traffic destined for that host is sent out only that port. Each port can support multiple MAC addresses. For example, if you have a hub plugged into a port, the switch will continue to store MAC addresses for all devices on the hub.

VLAN

A VLAN is a virtual local area network, a network segment defined by a switch or router. The switch connects all ports associated with a VLAN by its internal backbone, which is located inside the switch's hardware.

You can assign any series of ports on a switch to a VLAN. For example, on a 12-port switch, we could assign ports 1–6 to VLAN 2 and ports 7–12 to VLAN 3.

Without extra configuration, these VLANs are logically separated. Devices in VLAN 2 cannot access devices in VLAN 3 and vice-versa. Each VLAN is basically a separate subnet.

Every VLAN is assigned a number, which identifies it with not only the local switch but other switches on the network. In the example we just mentioned, we said we had VLAN 2 and VLAN 3. When we talk about trunking later in this chapter, we'll see that two or more switches can be joined together and share VLAN information as if they were all on one switch.

Broadcast Domain

By default, a router doesn't forward broadcast packets. Since they don't forward broadcast packets, routers create broadcast domains. A broadcast domain is the area to which a broadcast is limited. Switches, by contrast, do forward broadcasts. A VLAN is by definition a broadcast domain, so even though a switch forwards broadcasts among devices in a particular VLAN, it doesn't forward them to other VLANs.

Collision Domain

As described earlier, a collision domain is defined by the number of devices on a particular network segment. The more devices you have on a segment, the more collisions that will occur. Luckily, each port of a switch is considered a separate collision domain. If you add only one device per switch port, this works very well. However, if you plug a hub into a switch port and then plug multiple devices into the hub, you just created a collision domain among the hosts on the segment, and the switch can't do anything to prevent collisions.

Spanning Tree Protocol

Switches and bridges implement the spanning tree protocol (STP). This protocol has one primary purpose: loop prevention. A loop is basically a network transmission that keeps getting forwarded to other segments until it comes back to the original switch, which in turn forwards it again.



Throughout the explanation of spanning tree, you'll see the name bridge. That's because spanning tree was first developed for bridges. For the purposes of our discussion, whenever you see the word "bridge," substitute the word "switch."

To better illustrate a loop, consider the diagram in Figure 14-1, which shows Switch 1 forwarding a broadcast.

Without STP, loops can easily occur because switches have no knowledge of which frames they've already forwarded. In this example, Switch 1 forwards the broadcast to Switch 2 and Switch 3. These switches forward the broadcast back out every port except for the port on which the broadcast was originally received. The broadcast then makes it back to Switch 1, which happily broadcasts the same frame because it has no way of knowing it has already sent that broadcast out. As the cycle repeats, more copies of the same broadcast are flooded onto the network. This scenario is called a broadcast storm.

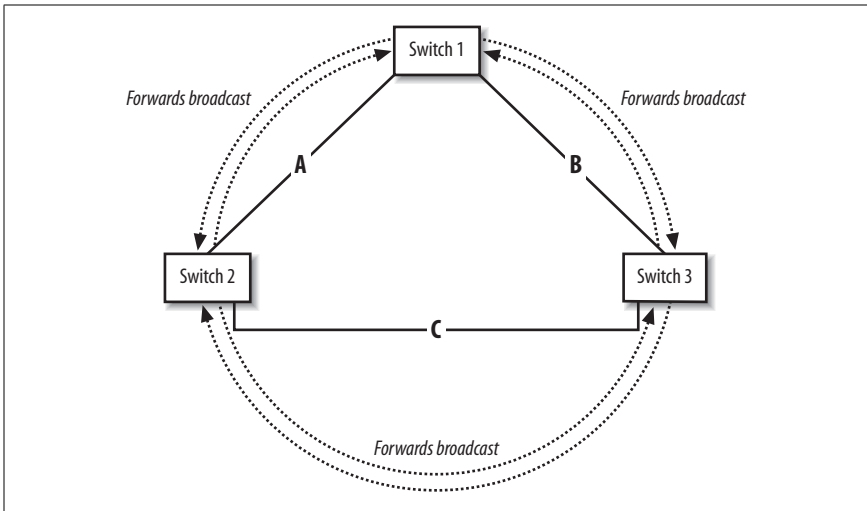


Figure 14-1. Broadcast storm

STP prevents this very situation. In a nutshell, STP builds a tree structure out of our network by removing redundant links. At the logical center of this new tree structure is the *root bridge*. Every switch on the network can access any nonlocal MAC address by forwarding frames toward the *root switch*. This tree structure—with its removal of redundant links—provides us with a loop-free network.

To understand how STP achieves this—from a very high level—we must first explain some STP terminology, including Port States and BPDUs.

Spanning Tree Port States

Consider that every active switch port can have one of the following states:

- Disabled
- Blocking
- Listening
- Learning
- Forwarding

A port is considered disabled if it has no link status or has been shut down with an IOS command. Once a port is enabled (e.g., a cable is plugged in), the port is immediately placed into the blocking state, which allows the network to stabilize before making any changes to the network.

In the blocking state, the port does not participate in frame forwarding. The port remains in the blocking state for the duration of the forward-delay timer, which is 20 seconds. If the port does not hear any messages from another switch during this period, the port switches to the listening state.

Once in the listening state, port learning and frame forwarding are still both disabled. Instead, the switch is listening for messages from other switches in order

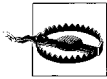
to try to determine how the network topology is configured. The listening state lasts for 15 seconds, after which the switch moves to the learning state.

In the learning state, the switch listens for station location information to add into its Filtering Database (MAC address table.). Once this state is complete, the switch port goes into the forwarding state, which is the normal operating mode of a switch in which it forwards frames.

Here are the possible transitions of port states:

- initialization → blocking
- blocking → listening or disabled
- listening → learning or disabled
- learning → forwarding or disabled
- forwarding → disabled

As you can see from this list, the disabled state can occur at any time. When a change occurs on the network, each port repeats the blocking → listening → learning → forwarding cycle. The switch cannot place a port into the disabled state by itself. Only the administrator can move a port into and out of the disabled state.



Have you ever unplugged your laptop from the network and then plugged it back in a few moments later, only to find that your network connection won't come back for about a minute? Chances are, you are plugging into a switch. That delay is STP doing its work on your port, moving it from blocking to listening to learning to forwarding. Once the port goes back into the forwarding state, your network connection is back. You can get around this delay, however; see the description of port-fast later in this section.

Bridge Protocol Data Units

Every switch that speaks STP uses Bridge Protocol Data Units (BPDUs). BPDUs are messages that switches (and bridges) pass back and forth to each other in order to discover the STP network topology. Every switch sends out one of these multicast messages approximately every 2 seconds. These communications continue even after the STP network topology has been determined. If a change is detected on the network, the switches need to reconfigure the STP network.

With BPDUs, the switches establish (or *elect*) a few things on the network:

- root bridge
- root port
- designated port

STP selects the root bridge

Selecting the *root bridge* is an important process. All switches (like nearly all people) start out thinking they are the root bridge. As switches send out BPDUs, they attach their associated Bridge ID (BID). The switch with the lowest BID wins and becomes the root bridge. Part of the BID message contains the switch's MAC

address and a configurable priority value. If left to the default, the switch with the lowest MAC address has the winning BID. However, you can force a switch to win the election by simply setting its priority value to a lower number than the other switches.

The root bridge selection is important because all other STP calculations are based on that choice. The root bridge becomes the logical center of our new tree structure. And as we already said, any switch on the network can reach any nonlocal MAC address by forwarding frames toward this root bridge.

Selecting a root port and a designated port

Every switch that is not the root bridge must elect a root port. The root port is the port with the lowest “cost” back to the root bridge. Table 14-1 shows the costs associated with various link types.

Table 14-1. Sample path costs

Link type	Cost
Gigabit Ethernet	4
Fast Ethernet	19
Ethernet	100

One problem with the selection of the root port is that this might not be the best or closest path to your intended destination, as we will see. In other words, just because the selected path is closest to the root bridge doesn’t mean it’s the closest to where you want to go.

A single *designated port* is elected for each LAN segment. One port on one switch is selected as the best path back to the root bridge. Unlike the root port, which is selected for every non-root switch, only one designated port is selected per segment. Basically, this port is the one that is placed in a forward state for the segment while the other ports on the segment are placed in blocking state. All ports that do not fall into the category of *root port* or *designated port* are put into blocking mode. By doing this, every segment (or LAN) is connected to every other segment on the network by only one path.

In Figure 14-2, the previous example has been updated with the root bridge, root ports, and designed ports. As you can see, the port connected to Switch 2 from Segment C is in forward state while the port connected to Switch 3 from Segment C is in the blocking state. This gives us only one path from Switch 2 to Switch 3, which is shown with the dotted line. Like we said before, we have only one path to the root bridge, and it’s not exactly the best path to our destination, which in this case is Switch 3.

To put it another way, in order to get to our intended destination (Switch 3), we have to take the longer path to follow STP rules. While this isn’t the best path, it’s a small price to pay for a loop-free network.

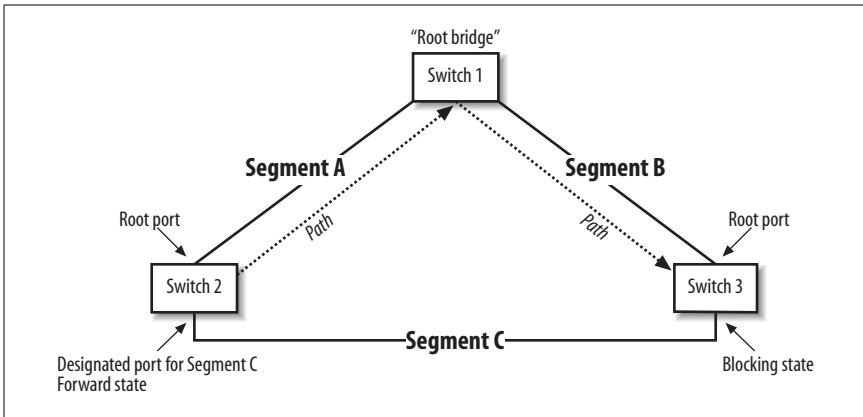


Figure 14-2. Loop-free network, thanks to spanning tree



Remember, even though Switch 3 is blocking on the Segment C port, it's still able to listen to BPDUs from Switch 2. The BPDU that it “hears” lets Switch 3 know that Switch 1 is the root bridge. And in order to get to the root bridge, Switch 2 has the best (designated) port on the segment that they share.

In summary, STP calculates three things to ensure a loop-free network:

- Root bridge election
- Root port on each non-root switch
- Designated port on each segment

By doing this, STP removes redundant links from our network. STP selects a root bridge and tells every other switch how to get back to it. Every switch can access any other nonlocal address simply by forwarding frames toward the root switch. These frames traverse the tree until they reach the final destination on our network.

In the next section, we'll learn how STP recovers if something breaks in our network. With convergence, STP rebuilds our network tree if a link goes down. In other words, one of the previously disabled redundant links will automatically become active.

Convergence in STP

Although our STP network topology has been selected, the switches keep communicating with BPDUs in case something changes. If something does change, like a switch is added or a current switch goes down, STP repeats the state cycle in order to converge the network. For example, if Switch 2 went down, Switch 3 would detect this and repeat the blocking, learning, listening state cycle, which would result in Switch 3 putting its port into the forwarding state. Once that happened, traffic would again flow from Segment C. The downside is that it takes about 50 seconds for this convergence to occur. (That works out to be 20 seconds in the blocking state if the root bridge can no longer be reached, 15 seconds for the listening state, and 15 more seconds for the learning state.)

Speeding up STP convergence

To most people, waiting 50 seconds for the switches to converge during a network change is unacceptable, so Cisco has provided a few methods to speed up STP convergence. The two methods that we cover are `portfast` and `uplinkfast`.

The `portfast` command tells the switch to enter the forwarding state immediately, bypassing the listening and learning states. This command should be used only on ports that are directly connected to a single device such as a server, workstation, or other end-user device (e.g., a network printer.) You should never use this on a port that connects to another switch because doing so will definitely break STP by introducing bridging loops.

```
interface fa0/11
  description port to bobs PC
  spanning-tree portfast
```

On the (older) 1900 and 2820 series switches, the `portfast` command is called `spantree start-forward`, which is actually more descriptive of what the command does by putting the switch immediately into the forwarding state.

The `uplinkfast` command causes an immediate switchover to another available root port when the current root port fails. The new root port is immediately switched from the blocking state to the forwarding state. By doing this, we bypass the STP calculation of selecting a new root port. This command should be used only on switches that will never be selected as the root bridge because the command changes the bridge priority to 19152, a value that assures that it will never be selected as the root bridge.

```
interface fa0/11
  spanning-tree uplinkfast
```

On the 1900 and 2820 series switches, this command is called `spantree uplink-fast`.

show spanning-tree

The `show spanning-tree` command gives you the output and status of spanning tree for the switch. In the highlighted sections of the sample output, we can see that the selected root port is port 8. If this switch were the root bridge, this line would read “We are the root of the spanning tree” because there is no root port on the root bridge. Why? Since the root port is always the port that leads toward the root bridge, we won’t find any such ports on the root bridge itself.

As for the interface listing, we can see that Interface Fa0/1 is in the forwarding state. Other states you might see are disabled and blocking. Finally, the other important item is the BPDUs counter, which tells us the number of BPDUs that were sent and received on this interface.

```
switch2#show spanning-tree
Spanning tree 1 is executing the IEEE compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 0030.80ae.ce40
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 32768, address 0030.809b.9f80
```

Root port is 8, cost of root path is 19

Topology change flag not set, detected flag not set, changes 15
Times: hold 1, topology change 35, notification 2
hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0

Interface Fa0/1 (port 1) in Spanning tree 1 is FORWARDING

Port path cost 100, Port priority 128
Designated root has priority 32768, address 0030.809b.9f80
Designated bridge has priority 32768, address 0030.80ae.ce40
Designated port is 1, path cost 19
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 211437, received 0

IOS on Switches

All Cisco switches use either IOS or CatOS as the user interface. IOS-enabled switches include the 2900XL, 2950, 3550, and 1900 series devices. These devices use the same IOS interface that routers use, with a few slight differences.

On the 1900 series (which is at the end of its product lifecycle), you must take an extra step to get to the IOS interface, navigating through a menu prompt. You press K to get to the IOS interface. For example:

```
Catalyst 1900 Management Console
Copyright (c) Cisco Systems, Inc. 1993-1998
All rights reserved.
Enterprise Edition Software
Ethernet Address:      00-B0-64-A7-85-00
```

```
PCA Number:           73-3121-04
PCA Serial Number:    FAB040231K4
Model Number:         WS-C1924-A
System Serial Number: FAB0403VOKG
Power Supply S/N:     PHI03430376
PCB Serial Number:    FAB040231K4,73-3121-04
-----
```

```
1 user(s) now active on Management Console.
```

```
User Interface Menu
```

```
[M] Menu
[K] Command Line
```

```
Enter Selection: K
```

```
CLI session with the switch is open.
To end the CLI session, enter [Exit].
```

```
switch1>
```

Many higher-end Catalyst series switches (4000, 5000, 5500, 6000, 6500) still run the CatOS (Catalyst OS). This interface is only vaguely similar to IOS. Like IOS, CatOS has an enable mode and show commands. However, that's where the similarities end. The CatOS uses set commands to configure the router. For example:

```
switch> (enable) set system name switch2
switch2> (enable) set ip route 0.0.0.0/0.0.0.0 172.16.1.3
switch2> (enable)
```

As you can see, the enable mode is a bit different. However, most of the commands will be familiar although their syntax and naming is totally different in most places. The good news is that Cisco is working on IOS capability for the newer models of CatOS switches. For example, the 6500 series can be configured with your choice of either CatOS or IOS. (You can easily guess which choice I would make.)

Basic Switch Configuration

Commands for IOS-enabled switches (e.g., 2900XL, 2950, 1900, and 3550) are almost identical to those on the IOS routers, which makes them fit nicely into the scope of this book. To connect a switch to the network, we must first configure the management port, as described next.

Configuring the Management Port (VLAN 1)

In order to access a switch remotely with telnet (SSH is not yet available on switches), ping, or SNMP, we need to supply a few basic pieces of information, including the IP address, subnet mask, and a default gateway.

We won't be applying the IP address to an interface; in fact, you should never configure a physical switch interface with an IP address unless it's a layer 3 switch. Instead, we apply the address to a logical interface. If you look at a default configuration of one of the IOS-based switches that we mentioned previously, you will see the physical ports, such as interface fastethernet0/1 through interface fastethernet0/24. You will also see interface VLAN 1. This does not correspond to a physical port; it is a logical port. By default, VLAN 1 is the management VLAN. Different protocols, which help to manage the network between devices such as CDP or VTP, travel over the management VLAN. When we assign an IP address to a switch, we configure it on the logical VLAN interface. You can configure the default management VLAN 1 with an IP address; however, any VLAN to which you assign an IP address becomes the management VLAN.

Figure 14-3 illustrates a very simple network, consisting only of a router and a switch. In our example network, you can imagine that we just replaced a hub with our new switch. Now that we have swapped our hub for a new switch, we need to configure it.

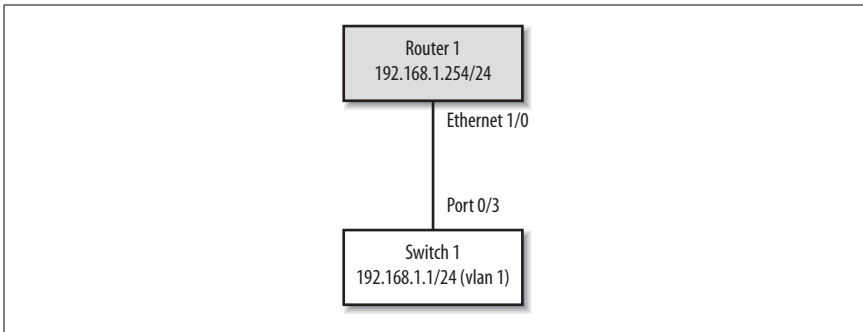


Figure 14-3. Basic switch connection to the network

Simple Switch Configuration

In this configuration, we are going to set VLAN 1 (the management VLAN for the switch) to 192.168.1.1/24 and our default gateway to the IP address of the router, which is 192.168.1.254/24.

Here's the configuration file for Switch 1 from Figure 14-3:

```

! Set the hostname
hostname switch1

! Configure the management VLAN interface
interface VLAN1
  description Our management VLAN for the switch
  ip address 192.168.1.1 255.255.255.0
  no shutdown
!
! Set the default gateway
ip default-gateway 192.168.1.254
!
! Configure the port that connects to router1
interface fastethernet 0/3
  description Connection to router1
  no shutdown
  
```



No matter how trivial they may seem, interface descriptions are always important. Configuring an interface always requires a good description, not for the switch's sake but for your own sanity.

Now that we have our management VLAN configured, we can ping back and forth from the router to the switch.

```
switch1#ping 192.168.1.254
```

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to 192.168.1.254, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/6 ms
```

The output of the `show VLAN brief` command shows all the interfaces currently in VLAN 1, which is what we'd expect since we haven't yet configured them into different VLANs:

```
switch1#show VLAN brief
VLAN Name                Status    Ports
-----
1    default                active   Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12,
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16,
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
```

The `show mac-address-table` command displays all the MAC addresses the switch has learned so far, which, in this case, is the MAC address of the router:

```
switch1#show mac-address-table
Dynamic Address Count:          1
Secure Address Count:          0
Static Address (User-defined) Count: 0
System Self Address Count:     47
Total MAC addresses:           48
Maximum MAC addresses:         2048
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-----
00b0.64f3.5ae0      Dynamic      1    FastEthernet0/3
```

Auto Detection

On our interfaces, we have the ability to leave the speed and duplex settings in auto negotiate, which means that the device will try to detect and set them automatically. Unfortunately, these auto-sensing features are notoriously bad at choosing the correct setting. Furthermore, an incorrect duplex setting can result in serious network latency and intermittent connectivity.

At half duplex, it is possible for both devices to sense that the wire is available and to transmit at the exact same time, which results in a collision. Collisions are considered normal. However, more than a one percent ratio of errors to total traffic signals indicates that something else might be wrong.

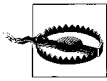
At full duplex, the collisions counter is not active. A duplex mismatch causes impaired collision handling. It's always a good idea to set these values explicitly in interface configuration with the `speed` and `duplex` commands.

```
interface fastethernet 0/3
  speed 100
  duplex full
```

To verify the settings, use the `show interface` command, just as you would on a router:

```
switch#show interface fastethernet0/3
FastEthernet0/3 is up, line protocol is up
Hardware is Fast Ethernet, address is 0030.809b.9f83 (bia 0030.809b.9f83)
```

```
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive not set
Full-duplex, 100Mb/s, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
...
```



If a device is slowing down your network, check the speed and duplex settings for the switch and the device. If you see increasing CRC errors, alignment errors, or runts on your network, it could be a duplex mismatch.

Sample VLAN Configuration

In our previous example, we configured only the management VLAN for the switch (VLAN 1). To make our network more realistic, let's split our network up into four VLANs: VLAN 1, VLAN 2 (Human Resources), VLAN 3 (Development), and VLAN 4 (Sales).

Figure 14-4 shows how these VLANs will be configured in our network.

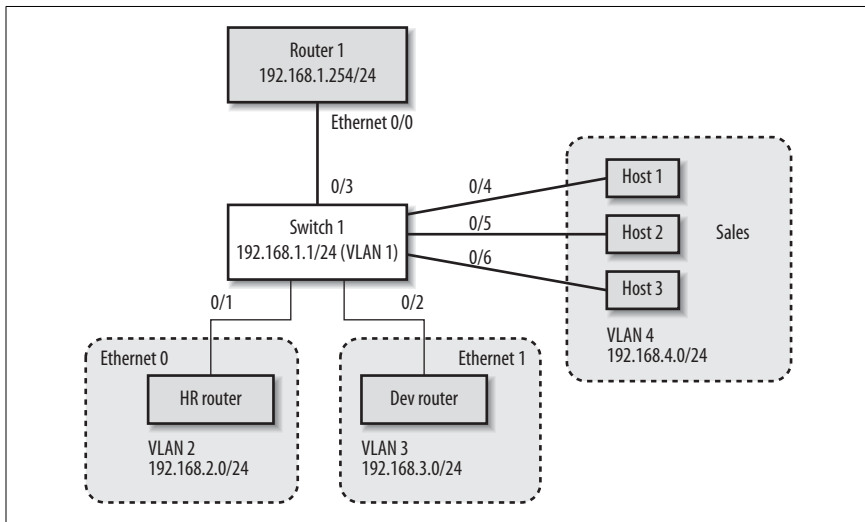


Figure 14-4. Splitting up the network with VLANs

As you can see, VLAN 2 (Human Resources) and VLAN 3 (Development) each have routers for their subnets while VLAN 4 (Sales) is simply composed of three hosts that are directly connected to the switch.

VLAN Interface Commands

To make an interface a member of a VLAN, use the `switchport access` command on each interface. In very simple terms, this command assigns an interface to each VLAN. The following configuration shows the switch commands for the network illustrated in Figure 14-4:

```

!
interface FastEthernet0/1
  description HR router (VLAN 2)
  switchport access VLAN 2
!
interface FastEthernet0/2
  description Development router (VLAN 3)
  switchport access VLAN 3
!
! This is our router connection from before
! no VLAN setting here - yet!
interface FastEthernet0/3
  description Connection to Router1
!
interface FastEthernet0/4
  description Sales1 (VLAN 4)
  switchport access VLAN 4
!
interface FastEthernet0/5
  description Sales2 (VLAN 4)
  switchport access VLAN 4
!
interface FastEthernet0/6
  description Sales3 (VLAN 4)
  switchport access VLAN 4
!

```

Now, when we run `show VLAN brief`, we see that the interfaces are in the VLANs that we expect them to be in:

```

Switch1#show VLAN brief
VLAN Name                Status    Ports
-----
1   default                 active    Fa0/3, Fa0/7, Fa0/8, Fa0/9,
                                 Fa0/10, Fa0/11
2   VLAN0002                active    Fa0/1
3   VLAN0003                active    Fa0/2
4   VLAN0004                active    Fa0/4, Fa0/5, Fa0/6

```

That's great! We've configured our VLANs just the way we want them. However, there's a big problem with this network. As you might recall from our previous discussion, each VLAN is a separate subnet, which means that VLANs 2, 3, and 4 are all logically separated. Router 1 can access only VLAN1 in this configuration, which means that the other VLANs can't access each other or Router 1. So what do we do? We need to make Router 1 a member of all VLANs. To do that, we need to employ trunking.

Trunking

Trunking allows us to connect two devices so that they can share each other's VLANs. For example, suppose you have more than one switch on your network, and you want to configure each switch so that it has ports in VLANs 1, 2, and 3. One (unrecommended) way you could accomplish this is to run a cable for each VLAN to each switch. In this way, each interface would have a port in all the

VLANs. Not only is this a waste of ports and cables, but it's terribly confusing. A much better method is to use trunking between the switches, which allows the devices to share their VLAN information easily.

Consider our previous example, with four VLANs configured on our switch. Like we just said, we could use a separate interface on the router for each VLAN. However, trunking makes this much cleaner. To better understand this concept, look at Figure 14-5.

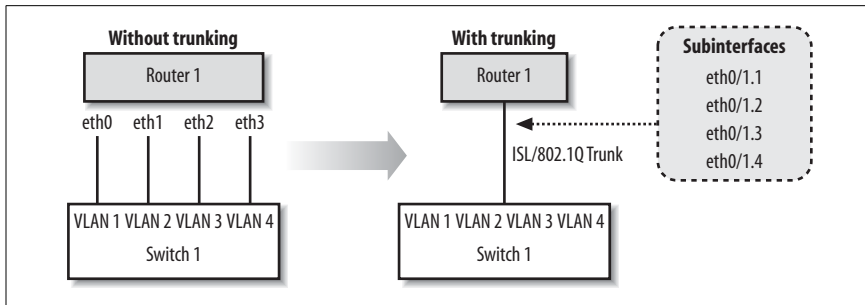


Figure 14-5. The benefits of trunking

Figure 14-5 shows two options for connecting our VLANs to the router. On the left side, which is labeled “without trunking,” you can see that we have used an extra interface and cable on the router to connect to each of the VLANs. What a mess. The much cleaner side, which is labeled “with trunking,” shows that we can use subinterfaces on the already used FastEthernet 0/1 interface and enable a trunk to the switch. Each subinterface on the router is given a VLAN id. The router tags outgoing packets with the appropriate VLAN id before sending them out the trunk. When the switch receives the packets on the trunk port, it can identify the VLAN tag and easily forward to the correct VLAN.

There are two trunking protocols that we can use:

ISL

Inter-Switch Link is a Cisco proprietary protocol, which means it only works on Cisco devices. ISL works on Ethernet, FDDI and Token Ring networks.

802.1Q

This protocol is an industry standard that works well in mixed environments.



Trunking works only on ports with speeds of 100 Mbps or greater, which means you can't run trunking on a 10 Mbps Ethernet connection.

By default, all ports on a switch are access ports. In order to use a port as a trunk, you must configure it with the `switchport mode trunk` command. We can also set the encapsulation type with the `switchport trunk encapsulation` command, which is set to ISL by default:

```
! This is our router connection from before
interface FastEthernet0/3
description Connection to router 1
```

```
switchport mode trunk
! Set the trunking to 802.1q instead of ISL
switchport trunk encapsulation dot1q
!
```

Restricting VLANs on a Trunk

By default, VLANs 1–1005 for ISL and 1-4095 for 802.1q are allowed to pass over a trunk. If you want to restrict which VLANs traverse the trunk port, you can use the following commands to add and remove VLANs from the allowed list:

```
switchport trunk allowed VLAN remove
switchport trunk allowed VLAN add
```

In a sense, the allowed list behaves like an access-list. We can verify the allowed VLAN list with the command `show interface switchport allowed-VLAN`. Here we can see that all VLANs are automatically allowed:

```
switch1#show interface fa0/8 switchport allowed-VLAN
"ALL"
```

If we had VLANs 2–200 but we only wanted allow VLANs 150–155 through the trunk, we could set it up like this:

```
interface fastethernet0/8
  switchport mode trunk
  switchport trunk allowed VLAN remove 2-200
  switchport trunk allowed VLAN add 150-155
```

These commands first remove all our defined VLANs from the list. We add the VLANs we want to permit. We can now verify that this is the truly the case with the show command:

```
switch1#show interface fa0/8 switchport allowed-VLAN
"1,150-155,201-1005"
```



VLAN 1 and VLANs 1002–1005 are reserved and cannot be removed. The switch automatically adds those VLANs to the allowed list.

Finishing Our Previous Network

In the previous section, we configured our switch for the multiple VLANs shown in Figure 14-4. Now that we know about trunking, we can finish our example by enabling trunking on the router, which allows the router to have a subinterface in each of the switch’s VLANs. This configuration is often referred to a “router on a stick.”

Here is Router 1’s configuration:

```
hostname Router1
!
interface FastEthernet0/0
  no ip address
!
interface FastEthernet0/0.1
```

```

description VLAN1 - management VLAN
encapsulation isl 1
ip address 192.168.1.254 255.255.255.0
no ip redirects
!
interface FastEthernet0/0.2
description HR VLAN 2
encapsulation isl 2
ip address 192.168.2.254 255.255.255.0
no ip redirects
!
interface FastEthernet0/0.3
description Development VLAN 3
encapsulation isl 3
ip address 192.168.3.254 255.255.255.0
no ip redirects
!
interface FastEthernet0/0.4
description Sales VLAN 4
encapsulation isl 4
ip address 192.168.4.254 255.255.255.0
no ip redirects

```

Here is Switch 1's configuration. The only thing that has changed in the enabling of the trunk on FastEthernet0/3 interface, highlighted in bold. The rest of the configuration is shown for completeness.

```

hostname switch1
!
interface VLAN1
ip address 192.168.1.1 255.255.255.0
no ip route-cache
!
interface FastEthernet0/1
description HR router (VLAN 2)
switchport access VLAN 2
!
interface FastEthernet0/2
description Development router (VLAN 3)
switchport access VLAN 3
!
interface FastEthernet0/3
description ISL trunk back to Router1
switchport mode trunk
!
interface FastEthernet0/4
description Sales1 (VLAN 4)
switchport access VLAN 4
!
interface FastEthernet0/5
description Sales2 (VLAN 4)
switchport access VLAN 4
!
interface FastEthernet0/6
description Sales3 (VLAN 4)
switchport access VLAN 4

```

To verify our configuration, we can ping the HR and Development routers from Router 1:

```
Router1#ping 192.168.2.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:  
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms  
Router1#ping 192.168.3.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:  
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

On the switch, we can verify that the port to Router 1 is indeed trunking:

```
switch1#show interface fastethernet0/3 switchport
```

```
Name: Fa0/3
```

```
Switchport: Enabled
```

```
Administrative mode: trunk
```

```
Operational Mode: trunk
```

```
Administrative Trunking Encapsulation: isl
```

```
Operational Trunking Encapsulation: isl
```

```
Negotiation of Trunking: Disabled
```

```
Access Mode VLAN: 0 ((Inactive))
```

```
Trunking Native Mode VLAN: 1 (default)
```

```
Trunking VLANs Enabled: ALL
```

```
Trunking VLANs Active: 1-4
```

```
Pruning VLANs Enabled: NONE
```

Added Port Security

One method to secure a port is to limit the number of MAC addresses that can be detected. This feature keeps users from plugging in extra devices (with the use of a hub or switch).

To enable this feature on a 2900 or 3500 series switch, use the port security and port security max-mac-count commands. In the following example, we restrict the port to only 1 MAC address:

```
interface FastEthernet 0/2  
  port security  
  port security max-mac-count 1
```

We can verify our settings with the following command:

```
switch1#show port security fa0/2
```

Secure Port	Secure Addr Cnt (Current)	Secure Addr Cnt (Max)	Security Reject Cnt	Security Action
-----	-----	-----	-----	-----
FastEthernet0/2	1	1	0	Send Trap

To take this example a little further, we could have the switch automatically send us an SNMP trap (assuming we have SNMP set up to forward to our network management station). Or, we could have the switch just shut down the port:

```
interface FastEthernet 0/2
  port security
  port security max-mac-count 1
  port security action shutdown
  port security action trap
```

On enable port security on a 2950 or 3500 switch, the commands are a bit different:

```
interface FastEthernet 0/2
  ! enable port security
  switchport port-security
  ! set the number of mac addresses
  switchport port-security maximum 1
  ! set the action to shutdown (other options are protect and restrict)
  switchport port-security violation shutdown
```

VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) allows switches to communicate about VLANs across trunk ports (see Figure 14-6). VTP makes administration of multiple switches much easier. Once you configure a switch for a VTP domain and set its mode (to either client or server), the switches automatically begin sharing VLAN information from the server.

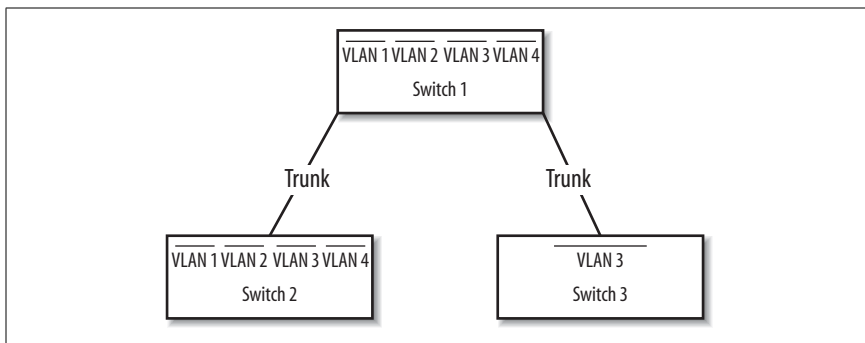


Figure 14-6. Trunks between switches with VTP management

VTP Modes

Switches configured with VTP have three modes: server, client, and transparent. All switches default to server mode when they are first configured for VTP. A VTP server switch can add, remove, and modify VLANs in the VLAN database. Once a change is made on a VTP server, the change is sent to all other VTP switches inside the VTP domain.

A VTP client switch pretty much just does what it's told by the VTP server switch, as long as the server is in the same VTP domain. A client cannot add, remove, or modify VLANs in the VLAN database.

In transparent mode, the switch acts as a go-between. The switch passes along VTP updates received by server switches, but the switch does not process them. A transparent switch is allowed to add, modify, and delete VLANs, but these changes remain local to the switch and are not sent out to other members of the VTP domain.

VLAN Database

To access the VLAN database and to configure VTP, use the VLAN database global command, which places you in VLAN configuration mode. In other words, this command is issued at the global command level, not in configuration mode, as this example shows:

```
switch1#VLAN database
switch1(VLAN)#?
VLAN database editing buffer manipulation commands:
  abort  Exit mode without applying the changes
  apply  Apply current changes and bump revision number
  exit   Apply changes, bump revision number, and exit mode
  no     Negate a command or set its defaults
  reset  Abandon current changes and reread current database
  show   Show database information
  VLAN   Add, delete, or modify values associated with a single VLAN
  vtp    Perform VTP administrative functions.

switch1(VLAN)#
switch1(VLAN)#exit
APPLY completed.
Exiting....
switch1#
```

Notice two important things in this example. First, when we exited VLAN configuration mode, our changes were immediately applied. Second, we entered that command from global command (enable) mode, not configuration mode.

Why from global command level? In short, I don't understand Cisco's rationale for this choice. Maybe it's the same reason that VLAN database configuration commands are not kept with the rest of the router configuration. It doesn't make a lot of sense; you just have to know that's where it is.

As I said, VLAN database configurations are not stored with the regular configuration commands, which are stored in the startup configuration. On a 2900 series router, you can see a *VLAN.dat* file in the output of a `dir` command on the flash contents:

```
switch1#dir flash:
Directory of flash:

 2  -rwx      4388  Mar 01 2004 00:31:53  VLAN.dat
 8  -rwx      656   Mar 01 2004 00:29:08  config.text
```

You also see the *config.text* file, which is the switch's startup configuration. *VLAN.dat* is the file in which our VLAN configurations are actually stored.

While this state of affairs is a bit confusing, it is changing. On newer devices and newer versions of IOS, Cisco has begun to move VTP settings into the regular configuration mode. For example, on the newer 3550 switches, you get this message when you type in the `vlan database` command:

```
% Warning: It is recommended to configure VLAN from config mode, as VLAN
database mode is being deprecated.
```

Configuring VTP

The following are the most commonly used VTP configuration commands. All of these are demonstrated in the configuration example later in this section.

Setting the VTP mode

Every device starts out thinking it's a VTP server. It's up to you to tell it whether it's client, server, or transparent with the `vtp server`, `vtp client`, or `vtp transparent` commands.

Setting the VTP domain

All VTP devices operate only within their domain. For clients and servers to talk to each other, you need to configure the VTP domain with the `vtp domain` command.

Setting the VTP password

Setting the VTP password is optional. However, it provides a bit of security so that someone on your network can't hook up a Cisco switch and start creating havoc with your VTP databases. The command to use is `vtp password`.

Creating a VLAN

You can create a VLAN by simply using the `VLAN id name text` command. *id* is the VLAN number and *text* is the name you wish to give to the VLAN.

Configuration example

In our network in Figure 14-4, we had only one switch. However, let's say we wanted to hook up another switch to our network called Switch 2. Switch 2 will have VLANs 3 and 4 on it. We'll connect these switches together using port 0/8 on both switches. Then we'll configure those two ports as trunks.

Set port 0/8 to trunking on both switches:

```
int fastethernet 0/8
  switchport mode trunk
  no shutdown
```

And on Switch 2, we'll configure the VLAN 1 interface:

```
switch2(config)#interface VLAN1
switch2(config-if)#ip address 192.168.1.2 255.255.255.0
switch2(config-if)#no shutdown
```

When we first connect the switches, they both think that they are VTP servers. On Switch 1, we'll set it as server (which it already is) and configure the VTP domain.

```
switch1#VLAN database
switch1(VLAN)#vtp server
Device mode already VTP SERVER.
switch1(VLAN)#vtp domain xyzcorp
Changing VTP domain name from NULL to xyzcorp
switch1(VLAN)#vtp password vtpass
Setting device VLAN database password to vtpass.
switch1(VLAN)#exit
APPLY completed.
Exiting....
```

Now, configure and name the VLANs on the VTP Server, which is Switch 1:

```
switch1#VLAN database
switch1(VLAN)#VLAN 2 name HR
VLAN 2 modified:
  Name: HR
switch1(VLAN)#VLAN 3 name Development
VLAN 3 modified:
  Name: Development
switch1(VLAN)#VLAN 4 name Sales
VLAN 4 modified:
  Name: Sales
switch1(VLAN)#exit
APPLY completed.
Exiting....
```

After configuring the VLANs in the database, the output of show VLAN brief now displays the names we just assigned:

```
switch1#show VLAN brief
VLAN Name                Status    Ports
-----
1   default                 active    Fa0/7, Fa0/9, Fa0/10, Fa0/11,
                                Fa0/12
2   HR                     active    Fa0/1
3   Development            active    Fa0/2
4   Sales                   active    Fa0/4, Fa0/5, Fa0/6
```

Next, we'll configure Switch 2 as a VTP client using pretty much the same commands as we used for Switch 1. The only major difference is that this switch is a client.

```
switch2#VLAN database
switch2(VLAN)#vtp client
Setting device to VTP CLIENT mode.
switch2(VLAN)#vtp domain xyzcorp
Changing VTP domain name from NULL to xyzcorp
switch2(VLAN)#vtp password vtpass
Setting device VLAN database password to vtpass.
switch2(VLAN)#exit
In CLIENT state, no apply attempted.
Exiting....
```

Now, when we run `show VLAN brief` on Switch 2, we find that the VLAN names have propagated:

```
Switch2#show VLAN brief
VLAN Name                               Status    Ports
-----
1    default                               active    Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                           Fa0/5, Fa0/6, Fa0/7, Fa0/9,
                                           Fa0/10, Fa0/11, Fa0/12

2    HR                                     active

3    Development                             active

4    Sales                                   active
```

VTP works only over trunks. Therefore, if you see the VLANs come across to the second switch, you know that you must have a valid trunk. The command is the best test of our trunking configuration. Also, as you can see, we haven't configured any ports for the VLANs yet, so everything is still in VLAN 1.

Finally, a good VTP show command:

```
switch2#show vtp counters
VTP statistics:
Summary advertisements received      : 3
Subset advertisements received      : 2
Reuquest advertisements received    : 0
Summary advertisements transmitted  : 4
Subset advertisements transmitted   : 2
Request advertisements transmitted   : 2
Number of config revision errors    : 0
Number of config digest errors      : 0
Number of V1 summary errors         : 0

VTP pruning statistics:
Trunk      Join Transmitted Join Received  Summary advts received from
-----
Fa0/8      0                0                0
non-pruning-capable device
```

Backing Up the VLAN Database

Earlier, we described how the VLAN database is stored separately from the device's starting configuration in a file called `VLAN.dat`. Even though this will probably change in the future, you can use it to your advantage now. If you want to, you can back up your VLAN database before you make any changes. If you want to revert to the previous configuration, use the following commands.

To back up the VLAN database:

```
switch2#copy flash:VLAN.dat flash:VLAN.bak
Source filename [VLAN.dat]?
Destination filename [VLAN.bak]?
4388 bytes copied in 0.131 secs
```

To recover a backed up version:

```
switch2#copy flash:VLAN.bak flash:VLAN.dat
Source filename [VLAN.bak]?
Destination filename [VLAN.dat]?
4388 bytes copied in 0.131 secs
switch2#reload
```

Switch Monitor Port for IDS or Sniffers

In order to configure an Intrusion Detection System (IDS) such as Snort (<http://www.snort.org>) or a sniffer for a switch, you need to select the interfaces or VLANs that you want to monitor. This monitoring is done with Switch Port Analyzer or (SPAN).

While the setup of SPAN differs by switch model, the same concepts are common to all switches. You select the interfaces or VLANs that you want the current port to “monitor.” Any traffic sent and received out the monitored interfaces or VLANs should also be sent to your monitor port.

For example, let’s assume we want to plug an IDS box into our switch on port fastethernet0/9. Our incoming Internet connection from the firewall is plugged into fastethernet0/1. This means that we want to send all incoming and outgoing traffic for fastethernet0/1 out to our IDS, which is on fastethernet0/9.

For the 2900xl/3500xl series devices, this is fairly straightforward:

```
interface FastEthernet0/9
port monitor FastEthernet0/1
```

With this configuration, any packet transmitted or received by fastethernet0/1 is copied (mirrored) out interface fastethernet0/9. That way, our IDS box can listen to all incoming and outgoing packets and look for signs of intrusion.

We can verify this with show port monitor:

```
switch2#show port monitor
Monitor Port      Port Being Monitored
-----
FastEthernet0/9  FastEthernet0/1
```

On 2940, 2950, 2955, 2970, 3550, 3560, 3750 and most other series switches, you need to employ the global monitor command:

```
! Set up fastEthernet 0/1 as our SOURCE port
monitor session 1 source interface fastEthernet 0/1
! Setup fastEtherent 0/9 as our DESTINATION port
monitor session 1 destination interface fastethernet 0/9
```

On a 2950, we can have only one monitor session and we can monitor only source interfaces.

To see the monitor configuration, use the show monitor command

```
# show monitor session 1
Session 1
-----
```

Source Ports:
RX Only: None
TX Only: None
Both: Fa0/1
Destination Ports: Fa0/9

Troubleshooting Switches

Here are some common troubleshooting techniques to try when configuring VLANs:

- Verify physical connections and layer 2 (data link layer).
 - Is the cable plugged into the correct port?
 - Is the link light on, and if so, what color is it? (green = forwarding, yellow = blocking, blinking yellow = error)
 - Use `show interface` command to verify link state (up or down).
 - Verify duplex and speed settings. (Remember that autonegotiation is unreliable.)
 - Use CDP to see if the Cisco devices can see each other (see Chapter 3 for details).
 - Verify that VLAN 1 has been configured and that a default gateway has been configured.
- If your VLANs can't see each other or your edge routers, verify your router and switch configurations.
 - Is the router a member of all VLANs?
 - Do you need trunking?
 - If you have a router in each VLAN, verify the router's configuration.
- Verify VLAN configuration.
 - Is the port in the correct VLAN?
 - Is there an "allowed" statement in the trunking configuration?
 - Use `show VLAN`.
 - Use `show interface switchport`.
 - Use `show spanning-tree`.
- If two switches don't seem to be sharing VLAN information or are not forwarding frames, verify the VTP configuration.
 - Is trunking enabled between the two switches?
 - Are both switches using the same trunk encapsulation (ISL, dot1q, etc.)?
 - Use the `show interface fastethernet0/1 switchport` command to verify the trunk encapsulation.
 - Use the `vtp status` command to verify the domain name and revision number.