Unidad 3: Paso 5 - Actividad Colaborativa 3

Presentado por:

FABIAN LEONARDO BERMUDEZ

ALVARO ELIU MELO

MARILIN VELASCO VELEZ

Presentado a:

GERARDO GRANADOS ACUÑA

Grupo colaborativo:

208014-1

Universidad Nacional Abierta y a distancia

Noviembre 2017

TABLA DE CONTENIDO

INTRODUCCIÓN

Las redes modernas continúan evolucionando para adaptarse a la manera cambiante en que las organizaciones realizan sus actividades diarias.

Ahora los usuarios esperan tener acceso instantáneo a los recursos de una compañía, en cualquier momento y en cualquier lugar. Estos recursos incluyen no solo datos tradicionales, sino también de video y de voz.

También hay una necesidad creciente de tecnologías de colaboración que permitan el intercambio de recursos en tiempo real entre varias personas en sitios remotos como si estuvieran en la misma ubicación física.

Los switches LAN proporcionan el punto de conexión a la red empresarial para los usuarios finales y también son los principales responsables del control de la información dentro del entorno LAN.

OBJETIVOS

Objetivo general

- Resolver problemas de las VLAN, los enlaces troncales de los switches Cisco, el enrutamiento entre VLAN, VTP y RSTP.

Objetivos específicos

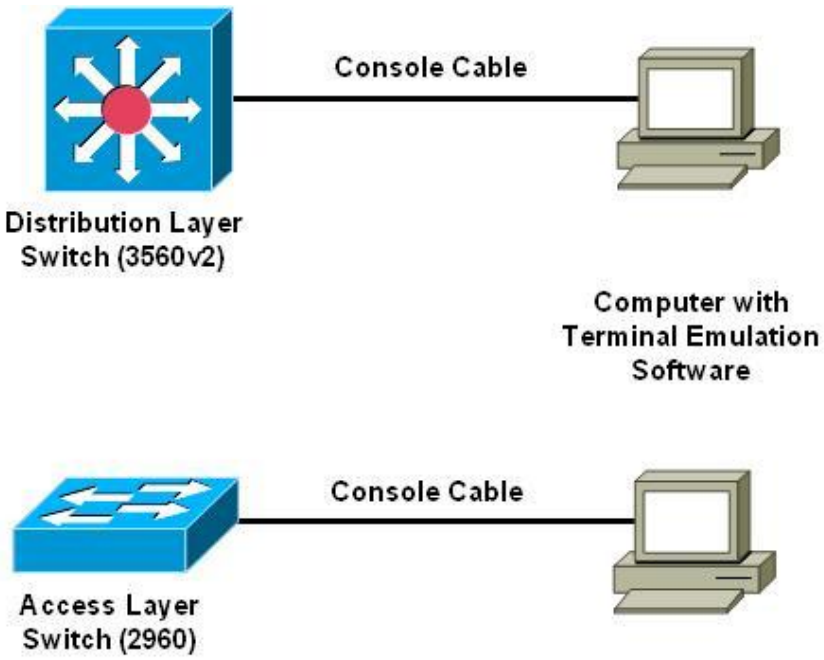
- Configurar redes con VLAN
- verificar redes con VLAN
- Solucionar problemas de las VLAN
- Evaluar los protocolos de Enrutamiento entre VLAN”s

DESARROLLO DE LA ACTIVIDAD

CCNPv7.1 SWITCH

Chapter 1 Lab - Preparing the Switch

Topology



Objectives

- Clear the configuration of **all the switches in your pod**
- Configure the database template used by **all the switches in your pod**
- Save a baseline configuration **for all the switches in your pod**

Background

When working with a switch that has been previously configured, any new commands entered will be merged with the existing configuration, causing unpredictable results. Additionally, if the switch is connected to other switches in the network, you can remove the VLANs but they might be relearned from another switch via VTP. In this lab you prepare your switches for use with future labs. This is accomplished by erasing the startup configuration from NVRAM and deleting the VLAN database. You also ensure that VLANs will not be relearned from another switch after the VLAN database has been deleted. Additionally, your switches may be required to support IPv6 traffic, which it does not

by default. This is accomplished by changing the database template used by the Switch Database Manager.

Note: This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates "dual-ipv4-and-ipv6 routing" and "lanbase-routing", respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

You may use one of the following switches **or a comparable one** with this lab:

- Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Console Cable
- Computer with terminal emulation software

Step 1: Connect to the switch console port and enter privileged EXEC mode.

From a computer running a terminal emulation program, connect to the console port of the switch that you want to clear using a console cable. You should see a console prompt that includes the switch's hostname, followed by a **>** or **#**. The default switch hostname is "Switch."

```
Switch>
```

or

```
Switch#
```

If the prompt ends with a **>**, you are not in privileged EXEC mode. To enter privileged EXEC mode, type **enable**. This might require a password. If you are in a configuration mode, type **exit** or **end**.

If not enabled:

```
Switch> enable
Switch#
```

If in global configuration mode:

```
Switch(config)# exit
Switch#
```

Step 2: Delete the VLAN database file, if present.

A VLAN database file named vlan.dat might exist in FLASH on the switch if it has been previously used in the network. This file holds information about VLANs created on the switch, their IDs, names, types and states, and it also stores the VTP settings. In privileged EXEC

mode, type **dir** or **dir flash:** and press Enter. This will provide a directory listing of the files in FLASH. In particular, note two files in the output: the vlan.dat file that will be removed in this step, and the multiple-fs file that will be explained and removed in Step 3 below.

```
Switch#dir
Directory of flash:/

    3  drwx         512   Mar 1 1993 00:38:22 +00:00  c3560-ipservicesk9-
mz.150-2.SE6
  522  -rwx        4889   Mar 2 1993 01:37:37 +00:00  startup-config
  560  -rwx        3096   Mar 1 1993 02:55:29 +00:00  multiple-fs
  561  -rwx         616  Mar 11 1993 23:00:09 +00:00  vlan.dat
```

In privileged EXEC mode, type **delete flash:vlan.dat** or **delete vlan.dat** (in the shorter form without the flash: prefix, the <TAB> key completion does not work) and press Enter. If you are asked to confirm, press Enter until you are back to the original prompt.

```
Switch# delete vlan.dat
Delete filename [vlan.dat]?              ! Press Enter
Delete flash:vlan.dat? [confirm]         ! Press Enter
Switch#
```

Step 3: Erase the startup config from NVRAM.

On Cisco devices, NVRAM is the common location for configuration files. The goal of this step is to entirely erase the NVRAM contents so that on the next boot, the switch starts in a factory default configuration. There are, however, a few important facts you need to be aware of.

On the switch platforms used in these or similar labs, such as Catalyst 2950, 2960, 3550, 3560, 3650, 3750, 3850, the NVRAM is not truly physically present. Instead, a part of the FLASH memory is used to store the NVRAM contents. In other words, on these switch platforms, the NVRAM is only simulated using a part of the FLASH, as also evidenced by one of lines in the **show version** command output:

```
Switch# show version | include volatile
512K bytes of flash-simulated non-volatile configuration memory.
```

Files that appear to reside in NVRAM (use **dir nvram:** to display its contents) are in fact stored in FLASH. Some of them are stored as standalone files in FLASH, such as flash:config.text that maps to nvram:startup-config, or flash:private-config.text that maps to nvram:private-config and stores sensitive information such as RSA keys, master password encryption key etc. Deleting any of these files from FLASH will cause the corresponding mapped file in simulated NVRAM to also be deleted or its apparent length in NVRAM to be truncated to zero, and vice versa. Other files in NVRAM, such as self-generated X.509 certificates, are all stored in the flash:multiple-fs file.

The **erase startup-config** command commonly used to remove the stored configuration will remove the flash:config.text and flash:private-config.text. However, other contents of the simulated NVRAM, such as X.509 certificates that were automatically created for the HTTPS server run on the switch, will not be removed as they reside in the flash:multiple-fs file unaffected by the **erase startup-config** command. Therefore, to completely erase the simulated NVRAM contents, not only the **erase startup-config** command must be issued, but also the flash:multiple-fs file must be removed.

In privileged EXEC mode, issue the **delete flash:multiple-fs** (or simply **delete multiple-fs**) command, followed by the **erase startup-config** command. Press Enter on each prompt.

```
Switch# delete multiple-fs
Delete filename [multiple-fs]?              ! Press Enter
Delete flash:/multiple-fs? [confirm]       ! Press Enter
Switch# erase startup-config
Erasing the nvram filesystem will remove all configuration files! Continue?
[confirm]                                  ! Press Enter
[OK]
Erase of nvram: complete
Switch#
*Mar  1 00:43:23.286: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
```

Note: Historically, the **write erase** command was used to erase the startup configuration. This command has the same effect as **erase startup-config** command that was implemented in later IOS versions, and is still being used as its shortened version **wr e** is more convenient to write than **erase startup-config**.

Step 4: Change the Switch Database Template.

The Cisco Switch Database Manager (SDM) provides various TCAM allocation templates that can be enabled to support specific roles, depending on how the switch is used in the network. By default the switch is using the “Default Desktop” template. This particular template divides the available TCAM up for use by the different processes and protocols in a manner that most likely supports standard IPv4 unicast and multicast traffic. Use the **show sdm prefer** command to see the details of the current template. Output may differ depending on the specific switch platform.

```
Switch# show sdm prefer
 The current template is "desktop default" template.
 The selected template optimizes the resources in
 the switch to support this level of features for
 8 routed interfaces and 1024 VLANs.

  number of unicast mac addresses:                  6K
  number of IPv4 IGMP groups + multicast routes:    1K
  number of IPv4 unicast routes:                    8K
    number of directly-connected IPv4 hosts:        6K
    number of indirect IPv4 routes:                 2K
  number of IPv6 multicast groups:                  0
  number of IPv6 unicast routes:                    0
    number of directly-connected IPv6 addresses:    0
    number of indirect IPv6 unicast routes:         0
  number of IPv4 policy based routing aces:         0
  number of IPv4/MAC qos aces:                      0.5K
  number of IPv4/MAC security aces:                 1K
  number of IPv6 policy based routing aces:         0
```

```
  number of IPv6 qos aces:                          20
  number of IPv6 security aces:                     25

Switch#
```

Notice in the output that there is NO memory allocated to IPv6 operations.

There are several different SDM templates available for use, each with different amounts of TCAM allocated to different processes and protocols. Use the **`show sdm prefer`** ***`template`*** command to examine the details of a particular database template.

```
Switch# show sdm prefer ?
  access              Access bias
  default             Default bias
  dual-ipv4-and-ipv6  Support both IPv4 and IPv6
  routing             Unicast bias
  vlan                VLAN bias
  |                   Output modifiers
  <cr>

Switch# show sdm prefer routing
 "desktop routing" template:
 The selected template optimizes the resources in
 the switch to support this level of features for
 8 routed interfaces and 1024 VLANs.

  number of unicast mac addresses:                  3K
  number of IPv4 IGMP groups + multicast routes:    1K
  number of IPv4 unicast routes:                    11K
    number of directly-connected IPv4 hosts:        3K
    number of indirect IPv4 routes:                 8K
  number of IPv6 multicast groups:                  0
  number of IPv6 unicast routes:                    0
    number of directly-connected IPv6 addresses:    0
    number of indirect IPv6 unicast routes:         0
  number of IPv4 policy based routing aces:         0.5K
  number of IPv4/MAC qos aces:                      0.5K
  number of IPv4/MAC security aces:                 1K
  number of IPv6 policy based routing aces:         0
  number of IPv6 qos aces:                          20
  number of IPv6 security aces:                     25

Switch#
```

Because our switching network will need to support IPv6 traffic, the SDM template must be changed to one of the dual-ipv4-and-ipv6 templates, using the **`sdm prefer dual-ipv4-and-ipv6`** ***`template`*** global configuration command.

The template options for dual IPv4 and IPv6 operation vary based on the model of switch. As of this writing, the dual-ipv6-and-ipv4 template options on 3560s are **default**, **routing**, and **VLAN**, while the option on 2960s is **default**. The 2960's **lanbase-routing** template also supports connected IPv6 hosts.

Note: The **routing** template is not a valid selection on switches running the LANBASE feature set, even though **routing** may appear as an option at the command line.

```
Switch(config)# sdm prefer dual-ipv4-and-ipv6 routing
Changes to the running SDM preferences have been stored, but cannot take
effect until the next reload.
Use 'show sdm prefer' to see what SDM preference is currently active.
Switch(config)#
```

As you can see in the output above, a restart is required for the change to be effective; the switch cannot re-allocate the TCAM on the fly. You will restart the switch at the end of this lab.

Configure your 3560 switches for the **`dual-ipv4-and-ipv6 routing`** template, and your 2960 switches for the **`lanbase-routing`** template.

Note: Several of the labs in the course end with instructions to reset the switch to its defaults. If you clear the switch (**`delete vlan.dat`**, **`delete multiple-fs`**, **`write erase`**, **`reload`**), the selected SDM template will return to the default, and could require reconfiguration (including a reboot).

Step 5: Reload the device, but do not save the system configuration if prompted.

After clearing the switch configuration, reload the switch by typing **`reload`** and pressing Enter. If you are asked whether to save the current configuration, answer **`no`**. Press Enter to confirm. The switch starts reloading. Your output might look different depending on the switch model that you are using. This step might take a few minutes, because the switch needs time to reload.

```
Switch# reload

System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm]

*Mar 11 23:03:06.985: %SYS-5-RELOAD: Reload requested by console. Reload
Reason: Reload command.
<output omitted>
```

Step 6: Create a Baseline Configuration

To eliminate some of the redundant basic configurations, use a TCL script to build a configuration shell. This can be customized several ways, but the basic text below creates a file named BASE.CFG in FLASH that can be used at the beginning of labs after the switch has been completely cleared. This will save some of the mundane configuration steps

Use the script below and modify it to meet the particulars of the switch you are working on:

```
tclsh
puts [ open "flash:BASE.CFG" w+ ] {
hostname DLS1
ip domain-name CCNP.NET
no ip domain lookup
interface range f0/1-24 , g0/1-2
shutdown
exit
vtp mode transparent
line con 0
no exec-timeout
logging synchronous
exit
end
}
tclquit
```

Step 7: Create a script to automate clearing and reloading of the switch

Use TCL once more to create a script to automate the tasks involved in clearing the switch.

DLS1/DLS2:

```
tclsh
puts [ open "flash:reset.tcl" w+ ] {
typeahead "\n"
copy running-config startup-config
typeahead "\n"
erase startup-config
delete /force vlan.dat
delete /force multiple-fs
ios_config "sdm prefer dual-ipv4-and-ipv6 routing"
typeahead "\n"
puts "Reloading the switch in 1 minute, type reload cancel to halt"
typeahead "\n"
reload in 1 RESET.TCL SCRIPT RUN
}
tclquit
```

ALS1/ALS2:

```
tclsh
puts [ open "flash:reset.tcl" w+ ] {
typeahead "\n"
copy running-config startup-config
typeahead "\n"
erase startup-config
delete /force vlan.dat
delete /force multiple-fs
```

```
ios_config "sdm prefer lanbase-routing"
typeahead "\n"
puts "Reloading the switch in 1 minute, type reload cancel to halt"
typeahead "\n"
reload in 1 RESET.TCL SCRIPT RUN
}
tclquit
```

Now for any lab that requires clearing the switch, issue the privileged exec command **`tclsh reset.tcl`** and the switch will be completely cleared and reload in the proper state.

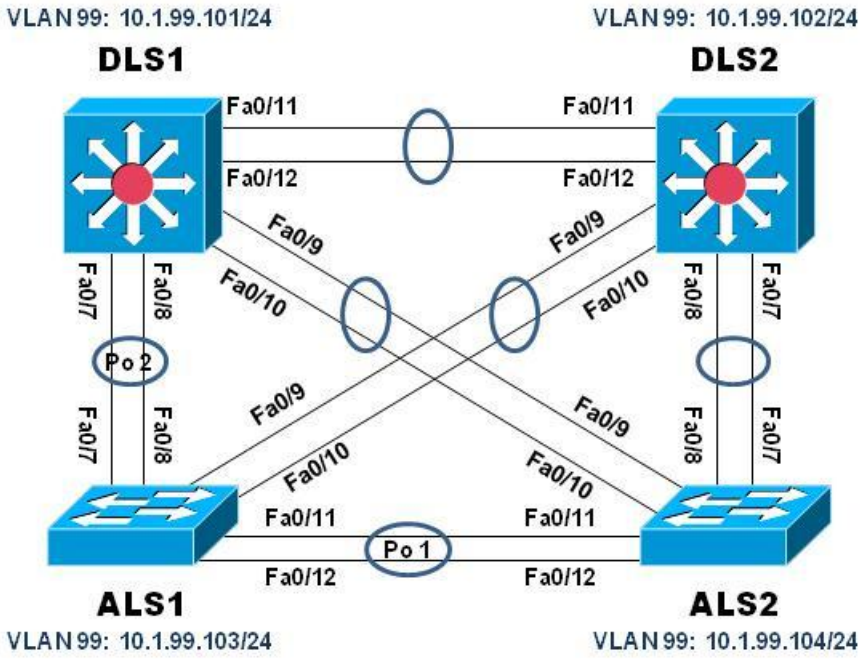
Step 8: End of Lab

At this point, your switches should be at a factory default with the BASE.CFG file in FLASH for future use.

CCNPv7.1 SWITCH

Chapter 3 Lab 3-2 – EtherChannel

Topology



All Switch-to-Switch connections are 802.1Q trunks

Objectives

- Create EtherChannel Links.
- Configure and test load balancing options

Background

Four switches have just been installed. The distribution layer switches are Catalyst 3560 switches, and the access layer switches are Catalyst 2960 switches. There are redundant uplinks between the access layer and distribution layer. Usually, only one of these links could be used; otherwise, a bridging loop might occur. However, using only one link utilizes only half of the available bandwidth. EtherChannel allows up to eight redundant links to be bundled together into one logical link. In this lab, you configure Port Aggregation Protocol (PAgP), a Cisco EtherChannel protocol, and Link Aggregation Control Protocol (LACP), an IEEE 802.3X (formerly IEEE 802.1ad) open standard version of EtherChannel. LACP and PAgP are signaling protocols allowing two switches to negotiate the use of selected physical ports as members of a single EtherChannel bundle. Throughout this lab, we will be using the term EtherChannel to refer to a logical bundling of multiple physical links, and the term Port-channel to refer to a virtual interface that represents an EtherChannel bundle in the Cisco IOS configuration.

Note: This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates “dual-ipv4-and-ipv6 routing” and “lanbase-routing”, respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

- 2 Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- 2 Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Computer with terminal emulation software
- Ethernet and console cables

Part 2: Configure EtherChannel Links

Step 1: Prepare the switches for the lab

The instructions in this lab assume that the switches are running using the final configuration from Lab 3-1 "Static VLANs, Trunking, and VTP".

Step 2: Configure an EtherChannel with Cisco PAgP.

The first EtherChannel created for this lab aggregates interfaces Fa0/11 and Fa0/12 between ALS1 and ALS2. Make sure that you have a trunk link active for those two links with the **`show interfaces trunk`** command.

```
ALS1# show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
```

```
Fa0/7       on              802.1q         trunking      666
Fa0/8       on              802.1q         trunking      666
Fa0/9       on              802.1q         trunking      666
Fa0/10      on              802.1q         trunking      666
Fa0/11      on              802.1q         trunking      666
Fa0/12      on              802.1q         trunking      666
<output omitted>
```

Note: When configuring EtherChannels, it can be helpful to shut down the physical interfaces being grouped on both devices before configuring them into channel groups. Otherwise, the EtherChannel Misconfig Guard may place these interfaces into error disabled state. The interfaces and port channel can be re-enabled after the EtherChannel is configured.

On ALS1, bundle interfaces Fa0/11 and Fa0/12 under the Port-Channel 1 interface with the **`channel-group 1 mode desirable`** command. The **`mode desirable`** option indicates that you want the switch to actively negotiate to form a PAgP link. The Port-Channel interface numbers are locally-significant only. On the 2960, the number can be anything between 1 and 6, and they do not have to match end to end. If it is possible, use the same number on both sides of a port-channel so that coordinating troubleshooting is less complicated. At the very least, clearly document the configuration.

```
ALS1(config)# interface range f0/11-12
ALS1(config-if-range)# shutdown
<output omitted - interfaces logged as shutting down>
ALS1(config-if-range)# channel-group 1 mode desirable
Creating a port-channel interface Port-channel 1

ALS1(config-if-range)# no shutdown
<output omitted - interfaces logged as coming up>
ALS1(config-if-range)# exit
ALS1(config)#
<the following output is seen after ALS2 configuration is complete>
*Mar  1 00:14:01.570: %LINK-3-UPDOWN: Interface Port-channel1,
changed state to up
*Mar  1 00:14:02.576: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Port-channel1, changed state to up
```

After you configure an EtherChannel, a virtual port channel interface is created automatically that represents a logical link consisting of the bundled physical interfaces. The Port-channel interface will automatically inherit the configuration of the first physical interface that was added to the EtherChannel. All configuration changes applied to the port channel interface will then apply to all the physical ports bundled under this interface.

The configuration of the physical interfaces that are bundled into an EtherChannel must be consistent. Otherwise, the bundle may never form or individual links in the bundle may be suspended. Once physical interfaces are added to the EtherChannel bundle, the administrator should not make any configuration changes directly to the physical interfaces. Any necessary adjustments should be made to the appropriate port channel interface.

Therefore, unless explicitly asked to do so in these labs, after physical ports have been bundled in an EtherChannel, apply all further commands to the corresponding port channel interface only.

Before configuring the EtherChannel bundle on ALS2, issue the command **show etherchannel summary** on ALS1 and notice the status of both the bundle and the individual interfaces:

```
ALS1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
-----
1      Po1(SD)          PAgP      Fa0/11(I)    Fa0/12(I)

ALS1#
```

PAgP is preventing the bundle from forming because the other end is not speaking the PAgP protocol.

Using the same commands as above, configure interfaces F0/11 and F0/12 on ALS2 to be in an EtherChannel, and then verify that it is working by issuing the **show etherchannel summary** command on both switches. This command displays the type of EtherChannel, the ports utilized, and port states.

```
ALS1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
```

```
Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
-----
1      Po1(SU)         PAgP      Fa0/11(P)   Fa0/12(P)

ALS1#


ALS2# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
-----
1      Po1(SU)         PAgP      Fa0/11(P)   Fa0/12(P)

ALS2#
```

At this point, the system does not consider interfaces FastEthernet 0/11 and 0/12 as individual trunks, but as a components of interface Port-Channel 1. The output of **`show interface trunk`** illustrates this; F0/11 and F0/12 are not shown while the Port-channel is operational.

```
ALS1# show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/7       on               802.1q         trunking      666
Fa0/8       on               802.1q         trunking      666
Fa0/9       on               802.1q         trunking      666
Fa0/10      on               802.1q         trunking      666
Po1         on               802.1q         trunking      666
<output omitted>
```

Step 3: Configure an EtherChannel with IEEE 802.1X LACP

In 2000, the IEEE passed an open standard version of EtherChannel numbered 802.3ad and referred to as "Link Aggregation". The current version of the standard is numbered 802.1AX. LACP-based EtherChannels are supported by most major network equipment vendors and provide interoperability in multi-vendor environments.

Using the previous commands, configure the link between DLS1 and ALS1 on ports Fa0/7 and Fa0/8 as an 802.1X LACP EtherChannel.

You must use a different port channel number on ALS1 than 1, because you already used that in the previous step. The port channel number you use on DLS1 is locally-significant and can be anything between 1 and 48. If it is possible, use the same number on both sides of a port-channel so that coordinating troubleshooting is less complicated. At the very least, clearly document the configuration.

To configure a port channel as LACP, use the interface-level command **`channel-group`** *`number`* **`mode active`**. Active mode indicates that the switch actively tries to negotiate that link as LACP, as opposed to PAgP

```
DLS1(config)# interface range f0/7-8
DLS1(config-if-range)# shutdown
<output omitted – interfaces logged as shutting down>
DLS1(config-if-range)# channel-group 2 mode active
Creating a port-channel interface Port-channel 2

DLS1(config-if-range)# no shutdown
<output omitted – interfaces logged as coming up>
DLS1(config-if-range)# end
DLS1#
<the following output is seen after ALS1 configuration is complete>
*Mar  1 00:31:29.752: %LINK-3-UPDOWN: Interface Port-channel2, changed
state to up
*Mar  1 00:31:30.758: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-
channel2, changed state to up
```

Verify that EtherChannel is working by issuing the **`show etherchannel summary`** command on both switches. This command displays the type of EtherChannel, the ports utilized, and port states.

```
DLS1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
```

```
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------
-----
2      Po2(SU)         LACP      Fa0/7(P)    Fa0/8(P)

DLS1#

ALS1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------
-----
1      Po1(SU)         PAgP      Fa0/11(P)   Fa0/12(P)
2      Po2(SU)         LACP      Fa0/7(P)    Fa0/8(P)

ALS1#
```

Step 4: Explore Misconfiguration

In this step, you will intentionally misconfigure an EtherChannel bundle on DLS2 with parameters that do not match the distant end switches to observe the results.

To do this, you will configure the interfaces on DLS1 and ALS1 as they should be configured for our final desired configuration. Then you will misconfigure DLS2 by bundling an interface that is connected to DLS1 and an interface that is connected to ALS1 into a single EtherChannel. Because different protocols are being used on the two distant ends, misconfiguration guard will force the interfaces into an error disabled state.

To begin, configure an EtherChannel using LACP on ALS1 interfaces F0/9 and F0/10. Assign this EtherChannel to Port-channel number 3.

```
ALS1(config)# interface range f0/9-10
ALS1(config-if-range)# shutdown
ALS1(config-if-range)# channel-group 3 mode active
Creating a port-channel interface Port-channel 3

ALS1(config-if-range)# no shut
ALS1(config-if-range)# exit
ALS1(config)#
```

Next configure an EtherChannel in "on" mode on DLS1 interfaces F0/11 and F0/12. Assign this EtherChannel to Port-channel number 12.

```
DLS1(config)# interface range f0/11-12
DLS1(config-if-range)# shutdown
DLS1(config-if-range)# channel-group 12 mode on
Creating a port-channel interface Port-channel 12

DLS1(config-if-range)# no shut
DLS1(config-if-range)# exit
DLS1(config)#
```

Now go to DLS2 and configure an EtherChannel using PAgP on interfaces F0/10 and F0/11. Assign this EtherChannel to Port-channel number 40.

```
DLS2(config)# interface range f0/10-11
DLS2(config-if-range)# shutdown
DLS2(config-if-range)# channel-group 40 mode desirable
Creating a port-channel interface Port-channel 40

DLS2(config-if-range)# no shut
DLS2(config-if-range)# exit
DLS2(config)#
```

Wait about three minutes, then issue the command **`show etherchannel summary`** on DLS2. Notice the difference in the individual interface status'.

```
DLS2# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
```

```
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
-----
40     Po40(SD)         PAgP      Fa0/10(I)    Fa0/11(D)

DLS2#
```

Interface F0/10 is attempting to communicate with a distant interface that is configured for LACP. This results in the interface being in a stand-alone state. Interface F0/11 is attempting to communicate with a distant interface that is configured not to use a signaling protocol, so the interface is in a down state.

On DLS1, the configuration mismatch caused Etherchannel Misconfig Guard to put F0/11, F0/12, and Port-channel 12 into an error-disabled state. The messages that displayed at DLS1's console when this happened:

```
*Mar  1 05:43:12.639: %PM-4-ERR_DISABLE: channel-misconfig (STP) error
detected on Fa0/11, putting Fa0/11 in err-disable state
*Mar  1 05:43:12.664: %PM-4-ERR_DISABLE: channel-misconfig (STP) error
detected on Fa0/12, putting Fa0/12 in err-disable state
*Mar  1 05:43:12.698: %PM-4-ERR_DISABLE: channel-misconfig (STP) error
detected on Po12, putting Fa0/11 in err-disable state
*Mar  1 05:43:12.698: %PM-4-ERR_DISABLE: channel-misconfig (STP) error
detected on Po12, putting Fa0/12 in err-disable state
*Mar  1 05:43:12.698: %PM-4-ERR_DISABLE: channel-misconfig (STP) error
detected on Po12, putting Po12 in err-disable state
*Mar  1 05:43:13.654: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/11, changed state to down
*Mar  1 05:43:13.679: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/12, changed state to down
*Mar  1 05:43:13.688: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-
channel12, changed state to down
```

To fix all of this, remove Port-channel 40 on DLS2 and create EtherChannels with the proper configurations to match the distant ends.

```
DLS2(config)# interface range f0/10-11
DLS2(config-if-range)# shut
DLS2(config-if-range)# no channel-group 40 mode desirable
DLS2(config-if-range)# exit
DLS2(config)# interface range f0/9-10
DLS2(config-if-range)# channel-group 3 mode active
Creating a port-channel interface Port-channel 3

DLS2(config-if-range)# no shut
```

```
DLS2(config-if-range)# exit
DLS2(config)# interface range f0/11-12
DLS2(config-if-range)# channel-group 12 mode on
Creating a port-channel interface Port-channel 12

DLS2(config-if-range)# no shut
DLS2(config-if-range)# exit
DLS2(config)# no interface port-channel 40
DLS2(config)# exit
```

Then reset Port-channel 12 on DSL1:

```
DLS1(config)# interface port-channel 12
DLS1(config-if)# shut
DLS1(config-if)# no shut
DLS1(config-if)# end
```

And all of the EtherChannels on DLS2 should be up and operational.

```
DLS2# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
-----
3      Po3(SU)         LACP      Fa0/9(P)    Fa0/10(P)
12     Po12(SU)         -        Fa0/11(P)   Fa0/12(P)

DLS2#
```

Challenge

The topology still has redundant links that you can combine. Experiment with the other port channel modes using the question mark on the physical interface command **`channel-group`**

number **mode** ?. Look at the descriptions and implement the remaining EtherChannels in different ways.

You may find the **desirable**, **auto**, **active**, and **passive** keywords cumbersome and unintuitive to associate with the particular signaling protocol. Try using the **channel-protocol** physical interface command, which limits the keywords in the **channel-group** *number* **mode** command so that only the keywords appropriate to the selected signaling protocol will be accepted.

Using **channel-protocol pagp** will make sure that in subsequent **channel-group** *number* **mode** command, only **desirable** and **auto** keywords are accepted. Conversely, using **channel-protocol lacp** will make sure that in subsequent **channel-group** *number* **mode** command, only **active** and **passive** keywords are accepted.

The end state from this part of the lab is that there are NO single interface trunks; all connections between switches will be port-channel interfaces consisting of two members.

Part 3: Configure and Test EtherChannel Load Balancing

Step 1: Configure the load-balancing method

The load balancing method used to send traffic through an EtherChannel is a global setting on the switch. All EtherChannels on a given switch will use the method selected for that switch. The load balancing method used at either end of an EtherChannel bundle do not have to match.

The available methods as well as the default method used varies by hardware platform. By default, Cisco Catalyst 3560 and Catalyst 2960 switches load-balance using the source MAC address.

```
DLS1# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source MAC address
  IPv4: Source MAC address
  IPv6: Source MAC address

DLS1#

ALS1# show etherchannel load-bal
EtherChannel Load-Balancing Configuration:
        src-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source MAC address
  IPv4: Source MAC address
  IPv6: Source MAC address
```

```
ALS1#
```

Change the load balancing configuration on ALS1 and ALS2 to **src-dst-ip**, which is ideal for most environments. Example from ALS2:

```
ALS2(config)# port-channel load-balance ?
  dst-ip       Dst IP Addr
  dst-mac      Dst Mac Addr
  src-dst-ip   Src XOR Dst IP Addr
  src-dst-mac  Src XOR Dst Mac Addr
  src-ip       Src IP Addr
  src-mac      Src Mac Addr

ALS2(config)#port-channel load-balance src-dst-ip
ALS2(config)#end
ALS2#
```

Step 2: Verify EtherChannel Load Balancing

Once this is configured on the switches, you can use the **test etherchannel load-balance** command . Using this command, you input a source and destination value and the switch will respond with what member interface of the EtherChannel would be used.

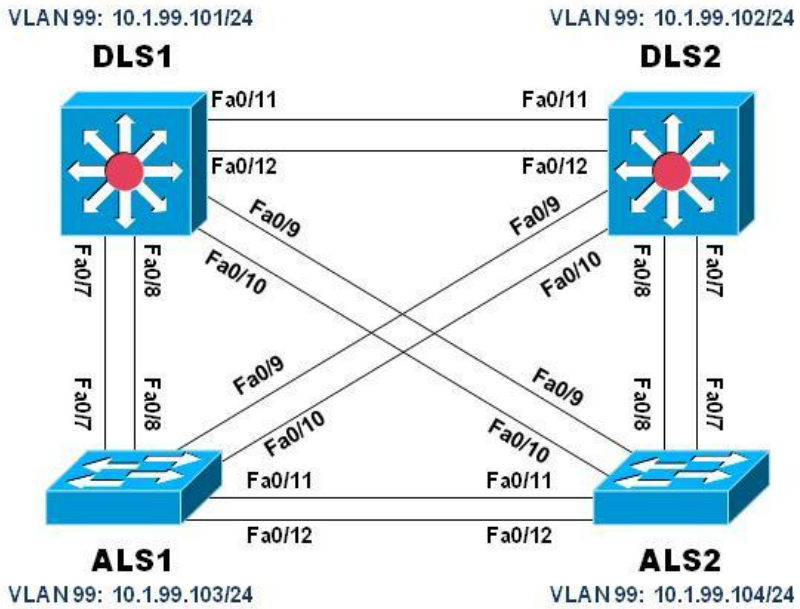
```
ALS1# test etherchannel load-balance interface po 1 ?
  ip    IP address
  ipv6  IPv6 address
  mac   Mac address

ALS1# test etherchannel load-balance interface po 1 ip ?
  A.B.C.D  Source IP address

ALS1# test etherchannel load-balance interface po 1 ip 10.1.99.103 ?
  A.B.C.D  Destination IP address

ALS1# test etherchannel load-balance interface po 1 ip 10.1.99.103
10.1.99.104
Would select Fa0/12 of Po1

ALS1# test etherchannel load-balance interface po 1 ip 10.1.99.103
209.165.200.103
Would select Fa0/11 of Po1

ALS1#
```

Step 3: End of Lab

Do not save your configurations. The equipment will be reset for the next lab.

Lab 3-1, Stactic VLANS, Trunking, and VTP

Topology



All Switch-to-Switch connections are 802.1Q trunks

Objectives

- Setup a VTP v2 Domain.
- Create and maintain VLANs.
- Configure 802.1Q Trunking.
- Setup a VTP v3 Domain.

Background

VLANs logically segment a network by function, team, or application, regardless of the physical location of the users. End stations in a particular IP subnet are often associated with a specific VLAN. VLAN membership on a switch that is assigned manually for each interface is known as static VLAN membership.

Trunking, or connecting switches, and the VLAN Trunking Protocol (VTP) are technologies that support VLANs. VTP manages the addition, deletion, and renaming of VLANs on the entire network from a single switch.

Note: This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates “dual-ipv4-and-ipv6 routing” and “lanbase-routing”, respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

- 2 Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- 2 Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Computer with terminal emulation software
- Ethernet and console cables

Part 1: Prepare for the Lab

Step 1: Prepare the switches for the lab

Use the **`reset.tcl`** script you created in Lab 1 “Preparing the Switch” to set your switches up for this lab. Then load the file BASE.CFG into the running-config with the command **`copy flash:BASE.CFG running-config`**. An example from DLS1:

```
DLS1# tclsh reset.tcl
Erasing the nvram filesystem will remove all configuration files! Continue?
[confirm]
[OK]
Erase of nvram: complete
Reloading the switch in 1 minute, type reload cancel to halt

Proceed with reload? [confirm]

*Mar  7 18:41:40.403: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of
nvram
*Mar  7 18:41:41.141: %SYS-5-RELOAD: Reload requested by console. Reload
Reason: Reload command.
<switch reloads - output omitted>

Would you like to enter the initial configuration dialog? [yes/no]: n
Switch> en
```

```
*Mar  1 00:01:30.915: %LINK-5-CHANGED: Interface Vlan1, changed state to
administratively down
Switch# copy BASE.CFG running-config
Destination filename [running-config]?
184 bytes copied in 0.310 secs (594 bytes/sec)
DLS1#
```

```
DLS1#tclsh reset.tcl
           ^
% Invalid input detected at '^' marker.

DLS1#
```

Copy Paste

Step 2: Configure basic switch parameters.

Configure an IP address on the management VLAN according to the diagram. VLAN 1 is the default management VLAN, but following best practice, we will use a different VLAN. In this case, VLAN 99.

Enter basic configuration commands on each switch according to the diagram.

DLS1 example:

```
DLS1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# interface vlan 99
DLS1(config-if)# ip address 10.1.99.101 255.255.255.0
DLS1(config-if)# no shutdown
```

```
DLS1(config)#interface vlan 99
DLS1(config-if)#ip address 10.1.99.101 255.255.255.0
DLS1(config-if)#no shutdown
DLS1(config-if)#exit
DLS1(config)#
```

Copy Paste

The interface VLAN 99 will not come up immediately, because the broadcast domain it is associated with (VLAN 99) doesn't exist on the switch. We will fix that in a few moments.

(Optional) On each switch, create an enable secret password and configure the VTY lines to allow remote access from other network devices.

DLS1 example:

```
DLS1(config)# enable secret class
DLS1(config)# line vty 0 15
DLS1(config-line)# password cisco
DLS1(config-line)# login
```

```
DLS1(config)#enable secret class
DLS1(config)#line vty 0 15
DLS1(config-line)#password cisco
DLS1(config-line)#login
DLS1(config-line)#
```

Copy Paste

Note: The passwords configured here are required for NETLAB compatibility only and are NOT recommended for use in a live environment.

Part 2: Configure VTP Version 2, VLANs, and Trunking.

Note(2): For purely lab environment purposes, it is possible to configure the VTY lines so that they accept any Telnet connection immediately, without asking for a password, and place the user into the privileged EXEC mode directly. The configuration would be similar to the following example for DLS1:

```
DLS1(config)# enable secret class
DLS1(config)# line vty 0 15
DLS1(config-line)# no login
DLS1(config-line)# privilege level 15
```

```
DLS1(config)#enable secret class
DLS1(config)#line vty 0 15
DLS1(config-line)#privilege level 15
DLS1(config-line)#
```

Copy

A VTP domain, also called a *VLAN management domain*, consists of trunked switches that are under the same administrative responsibility sharing the same VTP domain name. A switch can be in only one VTP domain, and VLAN database contents in the domain are globally synchronized. VLAN information is not propagated until a domain name is specified and trunks are set up between the devices.

There are three versions of VTP available; Version 1 and 2 are able to support normal-range VLANs only, while version 3 can support normal- and extended-range VLANs, as well as the synchronization of other databases. Support for version 3 on the Catalyst platforms used in this lab was added in IOS version 12.2(52)SE. Older IOS versions do not generally support VTP version 3.

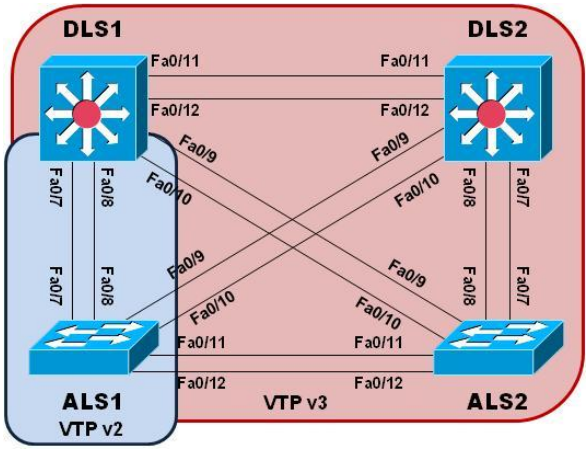
Switches operate in one of four VTP ***modes***. The default VTP mode for the 2960 and 3560 switches is server mode**, however our Lab 1 configuration changes this to transparent**.

VTP Mode	Description
VTP Server	You can create, modify, and delete VLANs and specify other configuration parameters, such as VTP version and VTP pruning, for the entire VTP domain. VTP servers advertise their VLAN configuration to other switches in the same VTP domain and synchronize their VLAN configuration with other switches based on advertisements received over trunk links. VTP server is the default mode.

	In VTP Server mode, VLAN configurations are only stored in the flash:vlan.dat file. While VLANs are manipulated in the configuration mode, the configuration commands do not appear in the running-config.
VTP Client	A VTP client behaves like a VTP server and transmits and receives VTP updates on its trunks, but you cannot create, change, or delete VLANs on a VTP client. VLANs are configured on another switch in the domain that is in server mode. In VTP Client mode, VLAN configurations are only stored in the flash:vlan.dat file. The configuration of VLANs does not appear in the running-config.
VTP Transparent	VTP transparent switches do not participate in VTP. A VTP transparent switch does not advertise its VLAN database nor synchronize its VLAN database based on received advertisements. However, transparent switches forward received VTP messages under two circumstances: either the VTP domain name of the transparent switch is empty (not yet configured), or it matches the domain name in the received VTP messages. In VTP Transparent mode, VLAN configurations are stored both in flash:vlan.dat file and also are present in the running-config. If extended range VLANs are used, however, they are stored in the flash:vlan.dat only if the switch is running VTP version 3.
VTP Off	A switch in VTP Off mode functions in the same manner as a VTP transparent switch, except that it does not forward VTP advertisements on trunks. VTP off is only available on switches that support VTP version 3 although it is not necessary to run VTP version 3 on the switch to be able to put it into the Off mode. In VTP Off mode, VLAN configurations are stored both in flash:vlan.dat file and also are present in the running-config. If extended range VLANs are used, however, they are stored in the flash:vlan.dat only if the switch is running VTP version 3.

In this lab we will demonstrate the configuration and operation of both VTP versions 2 and 3. We will do this by first configuring VTP version 2 between DLS1 and ALS1, and then configuring DLS1, DLS2 and ALS2 with VTP version 3.

Topology



Step 1: Verify VTP status

Issue the **`show vtp status`** command on DLS1

```
DLS1# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 64a0.e72a.2200
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Feature VLAN:
--------------
VTP Operating Mode                : Transparent
Maximum VLANs supported locally   : 1005
Number of existing VLANs          : 5
Configuration Revision            : 0
MD5 digest                        : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD
                                    0x56 0x9D 0x4A 0x3E 0xA5 0x69 0x35 0xBC
```

```
DLS1#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x7D 0x5A 0xA6 0x0E 0x9A 0x72 0xA0 0x3A
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
DLS1#
```

Copy Paste

Because no VLAN configurations were made, all settings except the VTP mode that was changed in Lab 1 are the defaults. This switch is capable of running version 1, 2 or 3 of VTP and runs version 1 by default. All switches in the VTP domain must run the same VTP version. The VTP mode is set to transparent as a result of steps performed in Lab 1. The number of existing VLANs is the five built-in VLANs. Different switches in the Catalyst family support different numbers of local VLANs. The 3560 switch used in this lab supports a maximum of 1,005 VLANs locally, while the 2960 switch used in this lab supports at most 255 VLANs. Lastly, note that the configuration revision is 0.

As you should recall from CCNA, the configuration revision number is compared amongst VTPv1 or VTPv2 switches and the VLAN database from the switch with the highest revision number is adopted by all the other switches in the VLAN management domain. Every time VLAN information is modified and saved in the VLAN database (vlan.dat), the revision number is increased by one when the user exits from VLAN configuration mode.

In VTPv3, revision numbers are still used but they no longer determine the switch whose database is going to apply to the entire domain. Instead, a single designated switch in a VTP domain called the *primary server* is allowed to assert its database in the entire VTP domain, even if its own revision number is lower. Other switches that are not primary servers are not allowed to assert their databases even if their revision numbers are higher.

Multiple switches in the VTP domain can be in VTP server mode. In VTPv1 and VTPv2, any of these server switches can be used to manage all other switches in the VTP domain. In VTPv3, a single primary server for a particular VTP domain is designated to control where changes originate from in the switched network. This enables careful management and protection of the VLAN database.

Step 2: Configure VTP on DLS1.

We will start off this lab by configuring DLS1 for VTP Server mode and setting the VTP domain name and VTP version 2. We will also set a VTP password, which provides some rudimentary protection against automatic VLAN database propagation. Because this password is set, VTPv2 will not allow ALS1 to automatically learn the domain name once trunks are installed.

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
DLS1(config)# vtp version 2
DLS1(config)# vtp mode server
Setting device to VTP Server mode for VLANS.
DLS1(config)# vtp password cisco123
Setting device VTP password to cisco123
DLS1(config)#
*Mar  1 00:29:10.895: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name
changed to SWLAB.
```

```
DLS1(config)#vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
DLS1(config)#vtp version 2
DLS1(config)#vtp mode server
Device mode already VTP SERVER.
DLS1(config)#vtp password cisco123
Setting device VLAN database password to cisco123
DLS1(config)#
```

Copy Paste

Verify these settings by using the **show vtp status** command again.

```
DLS1# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 2
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 64a0.e72a.2200
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode                : Server
Maximum VLANs supported locally   : 1005
Number of existing VLANs          : 5
Configuration Revision            : 0
MD5 digest                        : 0xA7 0xE6 0xAF 0xF9 0xFE 0xA0 0x88 0x6B
                                    0x21 0x6D 0x70 0xEE 0x04 0x6D 0x90 0xF3
```

```
DLS1#show vtp status
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Enabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0xE3 0x6D 0x6B 0xCB 0x9A 0x16 0x24 0xCF
Configuration last modified by 0.0.0.0 at 3-1-93 01:03:37
Local updater ID is 0.0.0.0 (no valid interface found)
DLS1#
```

Copy Paste

Step 3: Configure VLANs on DLS1

Next configure the VLANs that will be required to support the network. There are two ways to create VLANs, either directly via the **vlan** command or by assigning an interface to a non-existent VLAN. For now, you will create the VLANs directly on the switch. Create:

- VLAN 99 to enable the management interface.
- VLAN 999 as a “parking lot” VLAN for unused access ports
  - Suspend this VLAN to prevent ports in the VLAN from every communicating with each other.
- The VLANs required for network operations, which are VLANs 100, 110, and 120.

Suspending a VLAN deserves a special mention. Each VLAN has an operational state associated with it: it can be either active (the default state) or suspended. A suspended VLAN exists but it does not operate. Access ports assigned to a suspended VLAN drop all frames and are unable to communicate, similar to ports put into a nonexistent VLAN. Putting a suspended VLAN back into the active state reinstates normal communication on ports in that VLAN.

To globally *suspend* a VLAN, use the **`state suspend`** command in the VLAN configuration mode. This state is propagated by VTP to all other switches in the VTP domain if VTP is in use.

To locally *shut down* a VLAN, use the **`shutdown`** command in the VLAN configuration mode. This setting is not propagated through VTP.

Do not confuse the **`shutdown`** command in the VLAN configuration mode with the same command available under **`interface Vlan`** mode, which has a different and unrelated meaning. Further discussion on suspending and reactivating VLANs can be found in Part 3, Step 7 of this lab.

```
DLS1(config)# vlan 99
DLS1(config-vlan)# name MANAGEMENT
DLS1(config-vlan)# vlan 100
DLS1(config-vlan)# name SERVERS
DLS1(config-vlan)# vlan 110
DLS1(config-vlan)# name GUEST
DLS1(config-vlan)# vlan 120
DLS1(config-vlan)# name OFFICE
DLS1(config-vlan)# vlan 999
DLS1(config-vlan)# name PARKING_LOT
DLS1(config-vlan)# state suspend
DLS1(config-vlan)# vlan 666
DLS1(config-vlan)# name NATIVE_DO_NOT_USE
DLS1(config-vlan)# exit
```

```
DLS1(config-vlan)#name MANAGEMENT
DLS1(config-vlan)#vlan 100
DLS1(config-vlan)#name SERVERS
DLS1(config-vlan)#vlan 110
DLS1(config-vlan)#name GUEST
DLS1(config-vlan)#vlan 120
DLS1(config-vlan)#name OFFICE
DLS1(config-vlan)#vlan 999
DLS1(config-vlan)#name PARKING_LOT
DLS1(config-vlan)#state suspend
                  ^
% Invalid input detected at '^' marker.

DLS1(config-vlan)#vlan 666
DLS1(config-vlan)#name NATIVE_DO_NOT_USE
DLS1(config-vlan)#exit
DLS1(config)#
```

Copy Paste

The VLANs will not appear in the VLAN database until the **exit** command is issued.

After configuring the VLANs, issue the **show vtp status** command and you will see that the all-important configuration revision number has increased based on these changes to the VLAN database. Note that the revision number you have when performing this lab may be different.

```
DLS1#show vtp status | include Configuration Revision
Configuration Revision          : 6
```

Step 4: Configure trunking on DLS1

VTP will only propagate information over trunks. Cisco switches support Dynamic Trunking Protocol (DTP), which allows automatic negotiation of trunks. The partial output here from ALS1 shows you the default trunking mode:

```
ALS1#show interface f0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
```

```
ALS1#show interfaces fastEthernet 0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Appliance trust: none
ALS1#
ALS1#
```

Copy Paste

Switches that are interconnected and have DTP enabled can form a trunk automatically if either end is in the dynamic desirable mode or static trunk mode on the condition that either both switches use the same VTP domain name or at least one of the switches does not yet have the VTP domain name configured.

The dynamic auto mode on both ends will prevent a trunk from automatically forming; however, this is not really a valid safeguard against unintentional trunk connections as the port can become a trunk if the other side changes to dynamic desirable or static trunk mode.

As a best practice you should configure each interface into either access or trunk mode and use the **`switchport nonegotiate`** interface configuration command to disable the propagation of DTP messages. Never leave ports to operate in the dynamic mode.

Configure the appropriate interfaces on DLS1 to be trunks.

- Because DLS1 is a 3560, you must first specify the encapsulation protocol for the interface. Catalyst 3560 switches support ISL (Inter-Switch Link) and 802.1Q encapsulations for trunk interfaces. In the topology, all the trunks are 802.1Q trunks.
- Change the native VLAN from the default of VLAN 1 to VLAN 666.
- Set the interfaces to be in trunking mode only, and include the **`switchport nonegotiate`** command.
- The **`no shutdown`** command is needed because the Lab 1 configuration has all interfaces shutdown.

```
DLS1(config)# interface range f0/7-12
DLS1(config-if-range)# switchport trunk encapsulation dot1q
```

```
DLS1(config-if-range)# switchport trunk native vlan 666
DLS1(config-if-range)# switchport mode trunk
DLS1(config-if-range)# switchport nonegotiate
DLS1(config-if-range)# no shutdown
DLS1(config-if-range)#
```

```
DLS1(config)#interface range f0/7-12
DLS1(config-if-range)#switchport trunk encapsulation dot1q
DLS1(config-if-range)#switchport trunk native vlan 666
DLS1(config-if-range)#switchport mode trunk




DLS1(config-if-range)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/7, changed state to
down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/7, changed state to
```

Copy Paste

By default, all VLANs are allowed on all trunks. You can explicitly control which VLANs are allowed on a trunk by using the **`switchport trunk allowed vlan`** *`vlan-id`* command on the interface at each end of the trunk.

There are several approaches to deciding what VLANs to allow or disallow to cross the trunk. Common practice is to disallow VLAN 1 and the PARKING_LOT vlan. You could go a step further and disallow any unused VLAN numbers, but you would then have to modify all the trunks should you later add a new VLAN to the network.

In this lab, disallowing the PARKING_LOT VLAN from all trunks is not really necessary since the VLAN has been suspended. Disallowing the VLAN can serve as an additional protection against inadvertent reactivation of this VLAN.

Disallowing VLAN 1, also referred to as VLAN 1 Minimization, excludes VLAN 1 from the trunk but does not restrict layer 2 management traffic (such as CDP, LLDP, VTP, STP, etc) from passing.

Since only these 2 VLANs are being disallowed, the **`except`** option of the command can be used:

```
DLS1(config-if-range)# switchport trunk allowed vlan ?
  WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
  add     add VLANs to the current list
  all     all VLANs
  except  all VLANs except the following
  none    no VLANs
  remove  remove VLANs from the current list

DLS1(config-if-range)# switchport trunk allowed vlan except 1,999
DLS1(config-if-range)#
```

```
DLS1(config-if-range)#switchport trunk allowed vlan except 1,999
                                                            ^
% Invalid input detected at '^' marker.

DLS1(config-if-range)#
```

Copy Paste

Validate these settings by examining the switchport configuration for one of the trunk interfaces:

```
DLS1#show interface f0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 666 (NATIVE_DO_NOT_USE)
Administrative Native VLAN tagging: enabled
<output omitted>
Trunking VLANs Enabled: 2-998,1000-4094
<output omitted>
```

```
DLS1#show interfaces fastEthernet 0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 666
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
  --More--
```

Copy Paste

Step 5: Configure VTP and trunking on ALS1

Next configure VTP and trunking on ALS1. Configure ALS1 to be in VTP Client mode and then configure all of the appropriate trunk interfaces to use a native VLAN of 666 and to be in trunking mode only. The native VLAN number does not have to be the same across your network, but it must match between switches on a given connected trunk. Also, disallow VLANs 1 and 999.

```
ALS1(config)# vtp mode client
Setting device to VTP Client mode for VLANS.
```

```
ALS1(config)# interface range f0/7-12
ALS1(config-if-range)# switchport trunk native vlan 666
ALS1(config-if-range)# switchport mode trunk
ALS1(config-if-range)# switchport nonegotiate
ALS1(config-if-range)# switchport trunk allowed vlan except 1,999
ALS1(config-if-range)# no shutdown
ALS1(config-if-range)# exit
ALS1(config)#
```

```
ALS1#show interfaces fastEthernet 0/7 sw
ALS1#show interfaces fastEthernet 0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Copy Paste

After activating the interfaces, use the **`show interface trunk`** command to see the status of the trunks. You should see interfaces Fa0/7 and Fa0/8 in trunking mode.

```
ALS1# show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/7       on               802.1q         trunking      666
Fa0/8       on               802.1q         trunking      666

Port        Vlans allowed on trunk
Fa0/7       2-998,1000-4094
Fa0/8       2-998,1000-4094

Port        Vlans allowed and active in management domain
Fa0/7       none
Fa0/8       none
```

```
Port        Vlans in spanning tree forwarding state and not pruned
Fa0/7       none
Fa0/8       none
ALS1#
```

```
ALS1#show interfaces trunk
Port        Mode         Encapsulation  Status        Native vlan
Fa0/7       on           802.1q         trunking      666
Fa0/8       on           802.1q         trunking      666
Fa0/9       on           802.1q         trunking      666
Fa0/10      on           802.1q         trunking      666
Fa0/11      on           802.1q         trunking      666
Fa0/12      on           802.1q         trunking      666

Port        Vlans allowed on trunk
Fa0/7       1-1005
Fa0/8       1-1005
Fa0/9       1-1005
Fa0/10      1-1005
Fa0/11      1-1005
Fa0/12      1-1005

Port        Vlans allowed and active in management domain
Fa0/7       1
Fa0/8       1
Fa0/9       1
Fa0/10      1
Fa0/11      1
Fa0/12      1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/7       1
```

Copy Paste

Now if you look at the VTP status on ALS1, you will see the values are at their defaults, even though the trunk is operational. This is because of the VTP password.

```
ALS1# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 64a0.e72a.2200
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Feature VLAN:
--------------
VTP Operating Mode                : Client
Maximum VLANs supported locally   : 255
Number of existing VLANs          : 5
Configuration Revision            : 0
```

```
MD5 digest                          : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD
                                      0x56 0x9D 0x4A 0x3E 0xA5 0x69 0x35 0xBC
```

```
ALS1#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 255
Number of existing VLANs        : 5
VTP Operating Mode              : Client
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x87 0x67 0x56 0x26 0xCB 0xE7 0x65 0x4D
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
ALS1#
```

Copy Paste

Set the VTP password on ALS1 and the VLAN database will be synchronized. However, before you can set the password, the VTP domain name must be manually configured.

```
ALS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ALS1(config)# vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
ALS1(config)# vtp password cisco123
Setting device VTP password to cisco123
ALS1(config)# end
*Mar  1 00:27:21.902: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name
changed to SWLAB.
```

```
ALS1(config)#vtp domain SWLAB
Domain name already set to SWLAB.
ALS1(config)#vtp password cisco123
Setting device VLAN database password to cisco123
ALS1(config)#
%LINK-5-CHANGED: Interface Vlan99, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan99, changed state to up

ALS1(config)#end
ALS1#
%SYS-5-CONFIG_I: Configured from console by console
```

Copy Paste

Now check the VTP status and you will see a revision number matching that of DLS1, and that VLANs 99, 100, 110, 120, 666 and 999 are all in the local VLAN database.

```
ALS1# show vtp status
VTP Version capable             : 1 to 3
```

```
VTP version running             : 2
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 64a0.e72a.2200
Configuration last modified by 0.0.0.0 at 3-1-93 00:04:37

Feature VLAN:
--------------
VTP Operating Mode                : Client
Maximum VLANs supported locally   : 255
Number of existing VLANs          : 11
Configuration Revision            : 6
MD5 digest                        : 0xF3 0x8A 0xEA 0xFA 0x9B 0x39 0x6D 0xF5
                                    0xA6 0x03 0x2F 0xB8 0x16 0xC1 0xE6 0x8C

ALS1# show vlan brief | incl active
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            active
120  OFFICE                           active
666  NATIVE_DO_NOT_USE                active
ALS1#
```

```
ALS1#show vtp status
VTP Version                     : 2
Configuration Revision          : 13
Maximum VLANs supported locally : 255
Number of existing VLANs        : 11
VTP Operating Mode              : Client
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Enabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x26 0x8B 0x75 0x2F 0xAE 0xF9 0x43 0xA3
Configuration last modified by 0.0.0.0 at 3-1-93 01:10:25
ALS1#
```

Copy Paste

VLAN 999 will be missing from the filtered output above because it only includes VLANs in active state and VLAN 999 is suspended. Using the **`show vlan brief`** without filtering would show the VLAN 999.

You will also see that the configured VLANs except VLANs 1 and 999 are allowed over the trunks

```
ALS1#show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
```

```
Fa0/7       on              802.1q         trunking      666
Fa0/8       on              802.1q         trunking      666

Port        Vlans allowed on trunk
Fa0/7       2-998,1000-4094
Fa0/8       2-998,1000-4094

Port        Vlans allowed and active in management domain
Fa0/7       99-100,110,120,666,999
Fa0/8       99-100,110,120,666,999

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/7       99-100,110,120,666
Fa0/8       99-100,110,120,666
ALS1#
```

You will also see that the state of interface VLAN 99 has changed to 'up'.

```
*Mar  1 00:27:52.336: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Vlan99, changed state to up
```

Because ALS1 is in VTP Client mode, local changes to the VLAN database cannot be made:

```
ALS1(config)# vlan 199
VTP VLAN configuration not allowed when device is in CLIENT mode.
ALS1(config)#
```

```
ALS1(config)#vlan 99
VTP VLAN configuration not allowed when device is in CLIENT mode.
ALS1(config)#
```

Copy Paste

At this point, VTP version 2 is working and secured between DLS1 and ALS1. You should now be able to ping DLS1 from ALS1 and vice versa.

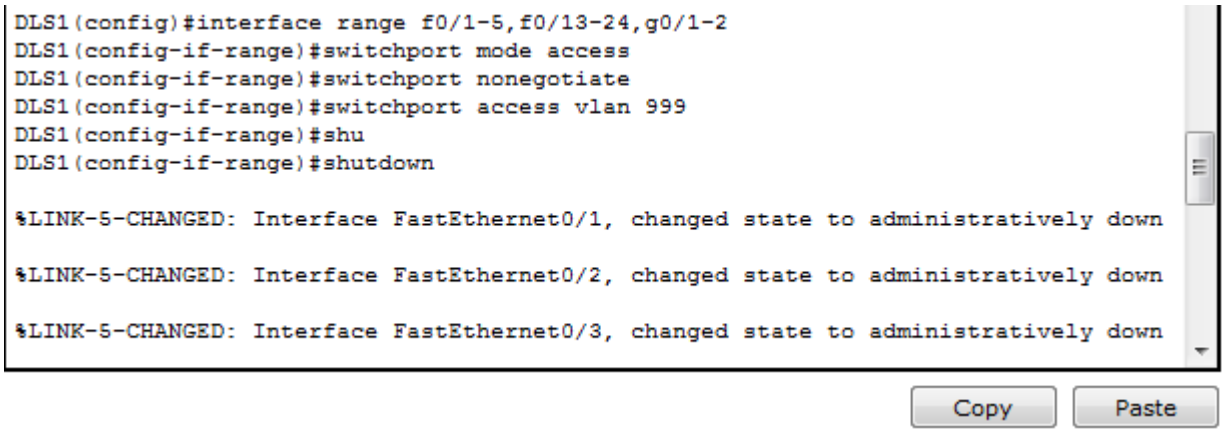
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	ALS1	DLS1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	DLS1	ALS1	ICMP		0.000	N	1	(edit)	(delete)

Step 6: Park unused interfaces

On DLS1 and ALS1, place all of interfaces that will not be used into the PARKING_LOT VLAN and shut them down. For this lab, the interfaces being used on all switches are F0/6 through F0/12.

An example from DLS1:

```
DLS1(config)# interface range f0/1-5,f0/13-24,g0/1-2
DLS1(config-if-range)# switchport mode access
DLS1(config-if-range)# switchport nonegotiate
DLS1(config-if-range)# switchport access vlan 999
DLS1(config-if-range)# shutdown
DLS1(config-if-range)# exit
```



Part 3: Configure VTP version 3, VLANs, and Trunking

In this part of the lab you will configure VTP version 3 to operate across the rest of the switched network. VTP version 3 provides some significant benefits to the network administrator.

1. The concept of a primary server was added. In VTP versions 1 and 2, all VTP server switches are equal; any one of them may add/remove/rename VLANs and change their state. In VTP version 3, only the primary server can do this. There can be at most one *primary* server present in a VTP domain. The role of a *primary* server is a runtime state. It is not a part of the configuration; rather, this state is requested in privileged EXEC mode and is relinquished whenever another switch attempts to become the primary server or when the switch is reloaded.
2. VTP version 3 has the ability to hide the VTP password. On a VTP version 1 or 2 switch, issuing the command **`show vtp password`** will show the password to you in plain text. VTP version 3 allows you to specify that the password be hidden in the output, preventing the password from being inadvertently or maliciously divulged.
3. VTP version 3 can propagate information about extended-range VLANs; VLANs numbered between 1006 and 4094. To support these VLANs with VTP version 2, all switches had to be in transparent mode and the VLANs had to be configured manually on a switch-by-switch basis.
4. VTP version 3 only supports pruning for normal-range VLANs.
5. VTP version 3 supports propagating Private VLAN information. As with extended-range VLANs, the lack of PVLAN support in VTP version 2 required all switches to be transparent mode and manual configuration at each switch.
6. VTP version 3 added support for opaque databases. In other words, VTP version 3 can transport more than just the VLAN database between switches. The only option at this

time is to share the Multiple Spanning Tree (MSTP) database, but room was left for expansion. We will cover MSTP in a later lab.

VTP version 3 is backwards compatible with VTP version 2; at the boundary of the two protocols, a VTP version 3 switch will send out both version 3 and version 2-compatible messages. Version 2 messages received by a version 3 switch are discarded.

Step 1: Configure VTP on DLS1, DLS2, and ALS2

VTP version 3 cannot be configured unless a VTP domain name has been set, so for this step, setting the domain name is not needed on DLS1. Configure VTP version 3 on DLS1, DLS2, and ALS2 using the following parameters

- VTP domain SWLAB (DLS2 and ALS2 only)
- VTP version 3
- VTP mode server (DLS2 and ALS2 only)
- VTP password cisco123 (DLS2 and ALS2 only)

DLS1 Configuration:

```
DLS1(config)# vtp version 3
DLS1(config)#
*Mar  1 00:08:17.637: %SW_VLAN-6-OLD_CONFIG_FILE_READ: Old version 2 VLAN
configuration file detected and read OK.  Version 3
    files will be written in the future.
DLS1(config)# end
DLS1#
```

```
DLS1(config)#vtp version 3
                        ^
% Invalid input detected at '^' marker.

DLS1(config)#
```

Copy Paste

Example configuration on ALS2:

```
ALS2(config)# vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
ALS2(config)# vtp version 3
ALS2(config)# vtp mode server
Setting device to VTP Server mode for VLANS.
ALS2(config)# vtp password cisco123
Setting device VTP password to cisco123
ALS2(config)# end
*Mar  1 18:46:38.236: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name
changed to SWLAB.
*Mar  1 18:46:38.345: %SW_VLAN-6-OLD_CONFIG_FILE_READ: Old version 2 VLAN
configuration file detected and read OK.  Version 3
```

```
    files will be written in the future.
ALS2#
```

```
ALS2(config)#vtp domain SWLAB
Domain name already set to SWLAB.
ALS2(config)#vtp version 3
                         ^
% Invalid input detected at '^' marker.

ALS2(config)#vtp version 2
ALS2(config)#vtp mode server
Device mode already VTP SERVER.
ALS2(config)#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/9
(1), with DLS1 FastEthernet0/9 (666).

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/10
(1), with DLS1 FastEthernet0/10 (666).
vtp password cisco123
Setting device VLAN database password to cisco123
ALS2(config)#
%LINK-5-CHANGED: Interface Vlan99, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan99, changed state to up
end
ALS2#
%SYS-5-CONFIG_I: Configured from console by console
```

Copy Paste

Step 2: Configure trunking on DLS2 and ALS2

In steps 4 and 5 of part 1, we configured and activated all the trunk interfaces on DLS1 and ALS1. Now configure and activate all the trunk interfaces on DLS2 and ALS2.

Example from DLS2:

```
DLS2(config)# interface range f0/7-12
DLS2(config-if-range)# switchport trunk native vlan 666
DLS2(config-if-range)# switchport trunk encapsulation dot1q
DLS2(config-if-range)# switchport mode trunk
DLS2(config-if-range)# switchport nonegotiate
DLS2(config-if-range)# switchport trunk allowed vlan except 1,999
DLS2(config-if-range)# no shutdown
DLS2(config-if-range)# exit
DLS2(config)#
```

Step 3: Verify trunking and VTP on DLS2 and ALS2

Next verify that DLS2 and ALS2 trunking is operational. Here is an example from DLS2

```
DLS2# show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/7       on               802.1q         trunking      666
Fa0/8       on               802.1q         trunking      666
Fa0/9       on               802.1q         trunking      666
Fa0/10      on               802.1q         trunking      666
Fa0/11      on               802.1q         trunking      666
Fa0/12      on               802.1q         trunking      666

Port        Vlans allowed on trunk
Fa0/7       2-998,1000-4094
Fa0/8       2-998,1000-4094
```

```
Fa0/9       2-998,1000-4094
Fa0/10      2-998,1000-4094
Fa0/11      2-998,1000-4094
Fa0/12      2-998,1000-4094

Port        Vlans allowed and active in management domain
Fa0/7       none
Fa0/8       none
Fa0/9       none
Fa0/10      none
Fa0/11      none
Fa0/12      none

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/7       none
Fa0/8       none
Fa0/9       none
Fa0/10      none
Fa0/11      none
Fa0/12      none
```

```
DLS2#show interfaces trunk
Port        Mode         Encapsulation  Status        Native vlan
Fa0/7       on           802.1q         trunking      666
Fa0/8       on           802.1q         trunking      666
Fa0/9       on           802.1q         trunking      666
Fa0/10      on           802.1q         trunking      666
Fa0/11      on           802.1q         trunking      666
Fa0/12      on           802.1q         trunking      666

Port        Vlans allowed on trunk
Fa0/7       1-1005
Fa0/8       1-1005
Fa0/9       1-1005
Fa0/10      1-1005
Fa0/11      1-1005
Fa0/12      1-1005

Port        Vlans allowed and active in management domain
Fa0/7       1
Fa0/8       1
Fa0/9       1
Fa0/10      1
Fa0/11      1
 --More--
```

Copy Paste

Next, validate VTP is operational. Here is an example from DLS2:

```
DLS2# show vtp status
```

```
VTP Version capable             : 1 to 3
VTP version running             : 3
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : e840.406f.7380

Feature VLAN:
--------------
VTP Operating Mode                : Server
Number of existing VLANs          : 5
Number of existing extended VLANs : 0
Maximum VLANs supported locally   : 1005
Configuration Revision            : 0
Primary ID                        : 0000.0000.0000
Primary Description               :
MD5 digest                        :


Feature MST:
--------------
VTP Operating Mode                : Transparent


Feature UNKNOWN:
--------------
VTP Operating Mode                : Transparent

DLS2#
```

```
DLS2#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x87 0x67 0x56 0x26 0xCB 0xE7 0x65 0x4D
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
DLS2#
```

Copy Paste

Notice that the configuration revision number is zero and the number of local VLANs is the default of 5. There has been no update because DLS1's configuration revision number was reset to zero when the VTP version was changed, so at this point DLS2 and ALS2 will not learn

about the configured VLANs because as far as they are concerned, they have the same database as DLS1.

```
DLS1# show vtp status | inc Configuration Revision
Configuration Revision            : 0
```

If we attempt to add VLANs at DLS1, or any of the other VTP version 3 switches, our attempt will not work and we will be told that we cannot add VLANs.

```
DLS1(config)# vlan 111
VTP VLAN configuration not allowed when device is not the primary server
for vlan database.
DLS1(config)#
```

Step 4: Configure the Primary VTP Server

In a VTP version 3 domain, only the "Primary Server" can make changes to the VLAN database. Becoming the primary server requires the **`vtp primary`** privileged EXEC command be executed. When you issue that command, the switch checks to see if there is another switch acting as primary server already, and asks you to confirm that you want to continue.

In the output from DLS2 above, note that the Primary ID field equals 0000.0000.0000. That field will display the base MAC address of the primary server once a device is promoted into that role.

Also note that a separate primary server can be configured independently for each feature supported; VLAN or MST. If no feature is specified, the **`vlan`** feature is assumed.

Lastly, there is a **`force`** option which causes the switch not to check for conflicts in the identity of the primary server. If different switches in the VTP domain identify different switches as the primary server, there is a good chance there are inconsistencies in the VLAN database.

```
DLS1# vtp primary ?
  force  Do not check for conflicting devices
  mst    MST feature
  vlan   Vlan feature
  <cr>

DLS1# vtp primary vlan
This system is becoming primary server for feature vlan
No conflicting VTP3 devices found.
Do you want to continue? [confirm]
DLS1#
*Mar  1 00:42:54.983: %SW_VLAN-4-VTP_PRIMARY_SERVER_CHG: e840.406f.7280 has
become the primary server for the VLAN VTP feature
```

Now verify the primary on DLS2 or ALS2:

```
DLS2# show vtp status | i Primary
Primary ID                          : e840.406f.7280
Primary Description                 : DLS1
DLS2#
```

The promotion of DLS1 to primary increments its configuration revision number to 1, so the VLANs that were previously created on DLS1 are propagated to DLS2 and ALS2 automatically.

```
ALS2# sho vlan brief | inc active
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            active
120  OFFICE                           active
666  NATIVE_DO_NOT_USE                active
```

```
ALS2#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24, Gig0/1, Gig0/2
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            active
120  OFFICE                           active
666  NATIVE_DO_NOT_USE                active
999  PARKING_LOT                      active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
ALS2#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/9
(1), with DLS1 FastEthernet0/9 (666).

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/10
(1), with DLS1 FastEthernet0/10 (666).
```

Copy Paste

VLAN 999 will be missing from the filtered output above because it only includes VLANs in active state and VLAN 999 is suspended. Using the **`show vlan brief`** without filtering would show the VLAN 999.

Step 5: Park unused interfaces.

On DLS2 and ALS2, place all of interfaces that will not be used into the PARKING_LOT VLAN and shut them down. For this lab, the interfaces being used on all switches are F0/6 through F0/12.

An example from DLS2:

```
DLS2(config)# interface range f0/1-5,f0/13-24,g0/1-2
DLS2(config-if-range)# switchport mode access
DLS2(config-if-range)# switchport nonegotiate
DLS2(config-if-range)# switchport access vlan 999
DLS2(config-if-range)# shutdown
DLS2(config-if-range)# exit
```

```
DLS2(config)#interface range f0/1-5,f0/13-24,g0/1-2
DLS2(config-if-range)#switchport mode access
DLS2(config-if-range)#switchport nonegotiate
DLS2(config-if-range)#switchport access vlan 999
% Access VLAN does not exist. Creating vlan 999
DLS2(config-if-range)#shutdown

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%LINK-5-CHANGED: Interface FastEthernet0/2, changed state to administratively down

%LINK-5-CHANGED: Interface FastEthernet0/3, changed state to administratively down
```

Copy Paste

Step 6: Verify VLAN management capability

DLS1 is able to create VLANs, including extended-range VLANs. Note that because ALS1 is running VTP version 2 and its revision number is 6, it will ignore any of the VTPv2 messages sent to it because they have a lower revision number. When a VTP message with an equal revision number but different MD5 checksum is received, ALS1 will report an error. Here we added seven VLANs and then remove 6 of them to push the revision number on DLS1 to 9.

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# vlan 510
DLS1(config-vlan)# name TEST510
DLS1(config-vlan)# exit
DLS1(config)# vlan 511
DLS1(config-vlan)# name TEST511
DLS1(config-vlan)# exit
DLS1(config)# vlan 512
DLS1(config-vlan)# name TEST512
DLS1(config-vlan)# exit
DLS1(config)# vlan 513
DLS1(config-vlan)# name TEST513
DLS1(config-vlan)# exit
DLS1(config)# vlan 514
DLS1(config-vlan)# name TEST514
DLS1(config-vlan)# exit
DLS1(config)# vlan 515
DLS1(config-vlan)# name TEST515
DLS1(config-vlan)# exit
DLS1(config)# vlan 1500
DLS1(config-vlan)# name TEST-EXT-1500
DLS1(config-vlan)# exit
DLS1(config)# no vlan 510-514
DLS1(config)# end
DLS1#
```

```
DLS1(config)#vlan 510
DLS1(config-vlan)#name TEST510
DLS1(config-vlan)#vlan 511
DLS1(config-vlan)#name TEST511
DLS1(config-vlan)#vlan 512
DLS1(config-vlan)#name TEST512
DLS1(config-vlan)#vlan 513
DLS1(config-vlan)#name TEST513
DLS1(config-vlan)#vlan 514
DLS1(config-vlan)#name TEST514
DLS1(config-vlan)#vlan 515
DLS1(config-vlan)#name TEST515
DLS1(config-vlan)#vlan 1500
                   ^
% Invalid input detected at '^' marker.

DLS1(config-vlan)#exit
DLS1(config)#no vl
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/9
(666), with ALS2 FastEthernet0/9 (1).

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/10
(666), with ALS2 FastEthernet0/10 (1).
```

Copy Paste

```
ALS1# show vlan brief | i active
1    default                          active    Fa0/6
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            active
120  OFFICE                           active
515  TEST515                          active
666  NATIVE_DO_NOT_USE                active
ALS1#

ALS2#show vlan brief | inc active
1    default                          active    Fa0/6
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            active
120  OFFICE                           active
515  TEST515                          active
666  NATIVE_DO_NOT_USE                active
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
1005 trbrf-default                    act/unsup
1500 TEST-EXT-1500                    active
ALS2#
```

```
ALS1#show vlan brief

VLAN Name                                Status    Ports
---- ----------------------------------- --------- -------------------------------
1    default                             active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                   Fa0/5, Fa0/6, Fa0/13, Fa0/14
                                                   Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                   Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                   Fa0/23, Fa0/24, Gig0/1, Gig0/2
99   MANAGEMENT                          active
100  SERVERS                             active
110  GUEST                               active
120  OFFICE                              active
510  TEST510                             active
511  TEST511                             active
512  TEST512                             active
513  TEST513                             active
514  TEST514                             active
515  TEST515                             active
666  NATIVE_DO_NOT_USE                   active
999  PARKING_LOT                         active
1002 fddi-default                        active
1003 token-ring-default                  active
1004 fddinet-default                     active
1005 trnet-default                       active
ALS1#
ALS1#
```

Copy Paste

At this point, you should be able to ping each switch from every other switch.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	DLS1	ALS1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	DLS1	ALS2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	DLS1	DLS2	ICMP		0.000	N	2	(edit)	(delete)

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	ALS1	DLS1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	ALS1	ALS2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	ALS1	DLS2	ICMP		0.000	N	2	(edit)	(delete)

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	DLS2	DLS1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	DLS2	ALS1	ICMP		0.000	N	1	(edit)	(delete)
	Successful	DLS2	ALS2	ICMP		0.000	N	2	(edit)	(delete)

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	ALS2	ALS1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	ALS2	DLS2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	ALS2	DLS1	ICMP		0.000	N	2	(edit)	(delete)

Step 7: Change the VLAN status to deactivate ports.

As already briefly discussed in Part 2, Step 3 of this lab, the default status of VLAN 1 and user-created VLANs is "active". A VLAN can be made locally inactive by entering the global configuration command **shutdown vlan** *vlan-id*, where *vlan-id* is the number of the VLAN to be shut down.

Alternatively, the VLAN can be shut down by issuing the **shutdown** command while in VLAN configuration mode and then exiting.

Both these options are equivalent, however, only the **shutdown vlan** command works in while in VTP client mode.

Shut down the Guest VLAN 110 on ALS1, wait a few moments, exit the configuration mode and then issue the **show vlan brief** command. The status should change to “act/lshut”.

a.

```
ALS1(config)# shutdown vlan 110

ALS1# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/6
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            act/lshut
515  TEST515                          active
120  OFFICE                           active
666  NATIVE_DO_NOT_USE                active
999  PARKING_LOT                      suspended Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/13, Fa0/14,
Fa0/15
                                                Fa0/16, Fa0/17, Fa0/18,
Fa0/19
                                                Fa0/20, Fa0/21, Fa0/22,
Fa0/23
                                                Fa0/24, Gi0/1, Gi0/2
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
1005 trbrf-default                    act/unsup
ALS1#
```

Reactivate all ports in ALS1 Guest VLAN 110 using the **no shutdown** command in VLAN configuration mode.

```
ALS1(config)# no shutdown vlan 110
```

As discussed and demonstrated in Part 2, Step 3 of this lab, you can put a VLAN into suspended status by using the **state suspend** command while in VLAN configuration mode on a VTPv2 server switch or on the VTPv3 primary server switch. In a mixed VTP version network, the suspension only works network-wide if it originates from the VTPv3 primary server.

Suspending a VLAN causes all ports in that VLAN throughout the VTP domain to stop transferring data.

Suspend Guest VLAN 110 on DLS1, wait a few moments, exit VLAN configuration mode and then issue the **show vlan brief | include suspended** command. The status should change show the VLAN as suspended.

```
DLS1(config)# vlan 110
DLS1(config-vlan)# state ?
  active    VLAN Active State
  suspend   VLAN Suspended State

DLS1(config-vlan)# state suspend
DLS1(config-vlan)# exit
DLS1(config)#end
DLS1#

DLS1# sho vlan brief | include suspended
110  GUEST                            suspended
999  PARKING_LOT                      suspended Fa0/1, Fa0/2, Fa0/3, Fa0/4
DLS1#
```

Issue the **show vlan brief | include suspended** command on another switch in the network, and you will see that the VLAN status is suspended as well.

b.

```
DLS2# show vlan brief | include suspended
110  GUEST                            suspended
999  PARKING_LOT                      suspended Fa0/1, Fa0/2, Fa0/3, Fa0/4
DLS2#
```

Reactivate VLAN 110 using the **state active** command in VLAN configuration mode.

```
DLS1(config)# vlan 110
DLS1(config-vlan)# state active
DLS1(config-vlan)# exit
DLS1(config)#
```

Issue the **show vlan brief | include suspended** command on another switch in the network, and you will see that the VLAN status is no longer listed.

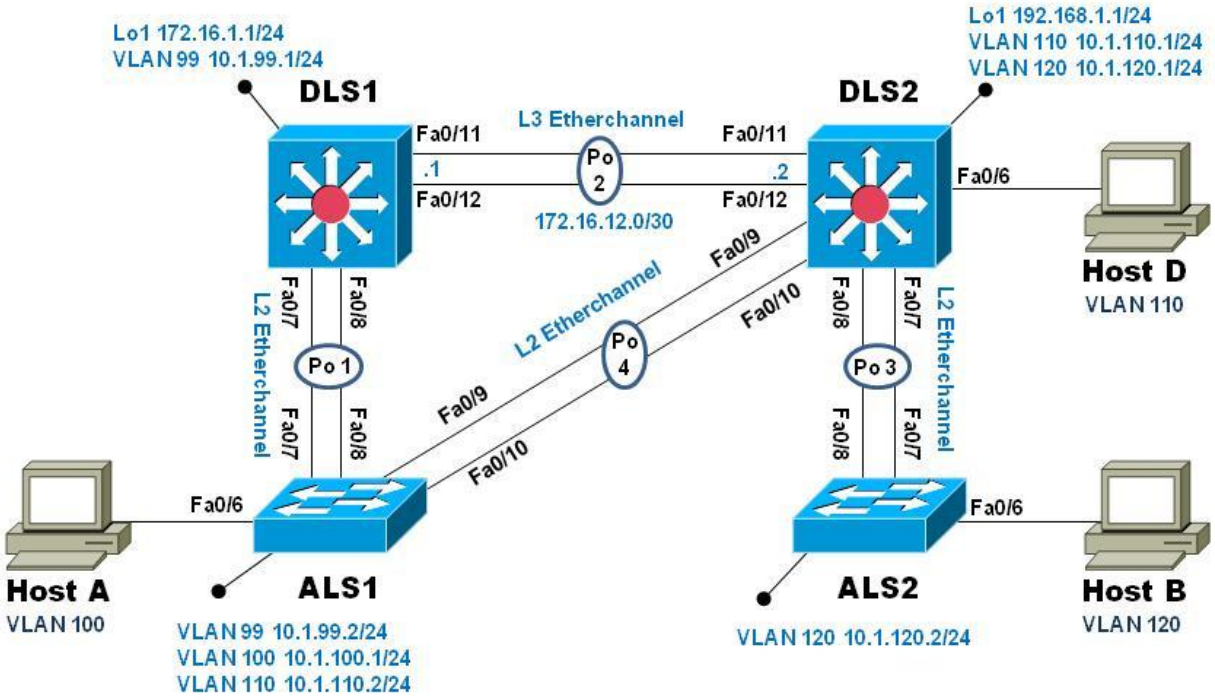
```
DLS2# show vlan brief | include suspended
999  PARKING_LOT                      suspended Fa0/1, Fa0/2, Fa0/3, Fa0/4
DLS2#
```

Step 8: End of Lab

Save your configurations. The equipment should be in the correct end state from this lab for Lab 3-2, EtherChannel.

Lab 5-1, Inter-VLAN Routing

Topology



Objectives

- Implement a Layer 3 EtherChannel
- Implement Static Routing
- Implement Inter-VLAN Routing

Background

Cisco's switching product line offers robust support for IP routing. It is common practice to use only multi-layer switching in the distribution layer of the network, eliminating routers in all but special use cases, usually when a gateway interface is required. Doing so provides many benefits in terms of cost and manageability. In this lab you will configure Inter-VLAN routing on the multi-layer switches in your pod and then a Layer 3 EtherChannel link to interconnect them. You will further configure one of your access-layer switches to support basic routing, and apply static routes so that there is simple path control.

Note: This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates “dual-ipv4-and-ipv6 routing” and “lanbase-routing”, respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

- 2 Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- 2 Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Computer with terminal emulation software
- Ethernet and console cables
- 3 PCs with appropriate software

Part 1: Configure Multilayer Switching using Distribution Layer Switches

Step 1: Load base config

Use the reset.tcl script you created in Lab 1 “Preparing the Switch” to set your switches up for this lab. Then load the file BASE.CFG into the running-config with the command **`copy flash:BASE.CFG running-config`**. An example from DLS1:

```
DLS1# tclsh reset.tcl
Erasing the nvram filesystem will remove all configuration files! Continue?
[confirm]
[OK]
Erase of nvram: complete
Reloading the switch in 1 minute, type reload cancel to halt

Proceed with reload? [confirm]
```

```
*Mar  7 18:41:40.403: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of
nvram
*Mar  7 18:41:41.141: %SYS-5-RELOAD: Reload requested by console. Reload
Reason: Reload command.
<switch reloads - output omitted>

Would you like to enter the initial configuration dialog? [yes/no]: n
Switch> en
*Mar  1 00:01:30.915: %LINK-5-CHANGED: Interface Vlan1, changed state to
administratively down
Switch# copy BASE.CFG running-config
Destination filename [running-config]?
184 bytes copied in 0.310 secs (594 bytes/sec)
DLS1#
```

Step 2: Verify switch management database configuration

At each switch, use the show sdm prefer command to verify the appropriate template is chosen. The DLS switches should be using the "dual ipv4-and-ipv6 routing" template and the ALS switches should be using the "lanbase-routing" template.

If any of the switches are using the wrong template, follow the procedures in Lab 1 to set the correct template and reboot the switch with the **reload** command.

Step 3: Configure layer 3 interfaces on the DLS switches

Enable IP Routing, create broadcast domains (VLANs), and configure the DLS switches with the layer 3 interfaces and addresses shown:

Switch	Interface	Address/Mask
DLS1	VLAN 99	10.1.99.1/24
DLS1	Loopback 1	172.16.1.1/24
DLS2	VLAN 110	10.1.110.1/24
DLS2	VLAN 120	10.1.120.1/24
DLS2	Loopback 1	192.168.1.1/24

VLAN	Name
99	MGMT1

100 (ALS1 only)	LOCAL
110	INTERNODE
120	MGMT2

An example from DLS2:

```
DLS2(config)# ip routing
DLS2(config)# vlan 110
DLS2(config-vlan)# name INTERNODE
DLS2(config-vlan)# exit
DLS2(config)# vlan 120
DLS2(config-vlan)# name MGMT2
DLS2(config-vlan)# exit
DLS2(config)# int vlan 110
DLS2(config-if)# ip address 10.1.110.1 255.255.255.0
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)# int vlan 120
DLS2(config-if)# ip address 10.1.120.1 255.255.255.0
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)# int loopback 1
DLS2(config-if)# ip address 192.168.1.1 255.255.255.0
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)#
```

```
DLS2>EN
DLS2#CONF T
Enter configuration commands, one per line.  End with CNTL/Z.
DLS2(config)#ip routing
DLS2(config)#vlan 110
DLS2(config-vlan)#name INTERNODE
DLS2(config-vlan)#EXIT
DLS2(config)#VLAN 120
DLS2(config-vlan)#name MGMT2
DLS2(config-vlan)#exit
DLS2(config)#int vlan 110
DLS2(config-if)#
%LINK-5-CHANGED: Interface Vlan110, changed state to up

DLS2(config-if)#ip address 10.1.110.1 255.255.255.0
DLS2(config-if)#no shu
DLS2(config-if)#no shutdown
DLS2(config-if)#exit
DLS2(config)#int vlan 120
DLS2(config-if)#
%LINK-5-CHANGED: Interface Vlan120, changed state to up

DLS2(config-if)#ip address 10.1.120.1 255.255.255.0
DLS2(config-if)#no shut
DLS2(config-if)#exit
DLS2(config)#int lo1
```

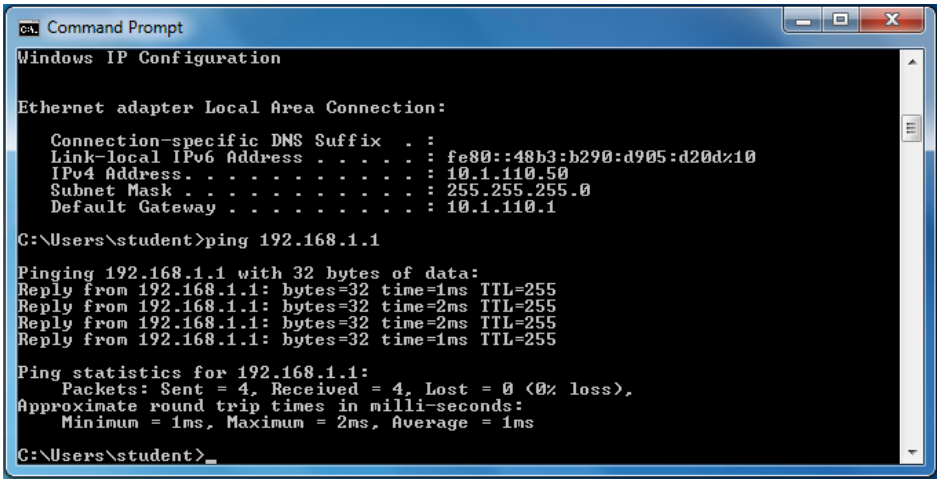
Copy Paste

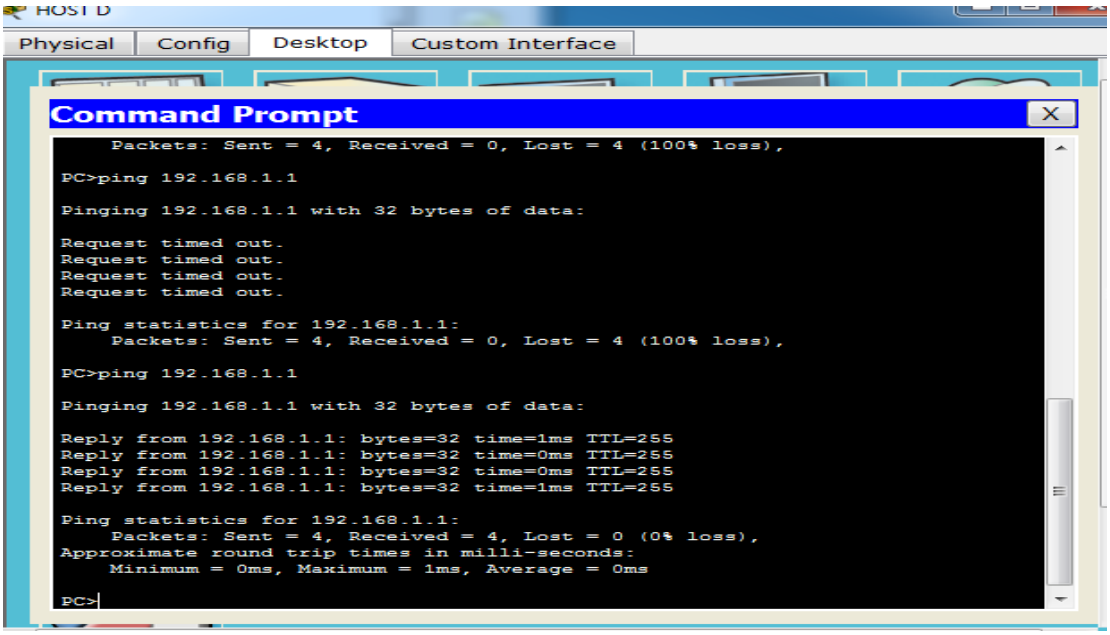
At this point, basic inter-vlan routing can be demonstrated using an attached host. Host D is attached to DLS2 via interface Fa0/6. On DLS2, assign interface Fa0/6 to VLAN 110 and configure the host with the address 10.1.110.50/24 and default gateway of 10.1.110.1. Once you have done that, try and ping Loopback 1's IP address (192.168.1.1). This should work just like an external router; the switch will provide connectivity between two directly connected interfaces. In the output below, the **`switchport host`** macro was used to quickly configure interface Fa0/6 with host-relative commands:

```
DLS2(config)# int f0/6
DLS2(config-if)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled

DLS2(config-if)# switchport access vlan 110
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)#
```

```
DLS2(config-if)#switchport access vlan 110
DLS2(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan110, changed state to up

DLS2(config-if)#switchport host
                               ^
% Invalid input detected at '^' marker.

DLS2(config-if)#no shu
DLS2(config-if)#no shutdown
DLS2(config-if)#^Z
DLS2#
%SYS-5-CONFIG_I: Configured from console by console

DLS2#
```

Copy Paste





Step 4: Configure a Layer 3 EtherChannel between DLS1 and DLS2

Now you will interconnect the multilayer switches in preparation to demonstrate other routing capabilities. Configure a layer 3 EtherChannel between the DLS switches. This will provide the benefit of increased available bandwidth between the two multilayer switches. To convert the links from layer 2 to layer 3, issue the **`no switchport`** command. Then, combine interfaces F0/11 and F0/12 into a single PAgP EtherChannel and then assign an IP address as shown.

DLS1	172.16.12.1/30	DLS2	172.16.12.2/30

Example from DLS1:

```
DLS1(config)# interface range f0/11-12
DLS1(config-if-range)# no switchport
DLS1(config-if-range)# channel-group 2 mode desirable
Creating a port-channel interface Port-channel 2

DLS1(config-if-range)# no shut
DLS1(config-if-range)# exit
DLS1(config)# interface port-channel 2
DLS1(config-if)# ip address 172.16.12.1 255.255.255.252
DLS1(config-if)# no shut
DLS1(config-if)# exit
DLS1(config)#
```

```
DLS1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)#int range fa0/11-12
DLS1(config-if-range)#no switchport
DLS1(config-if-range)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to
down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/12, changed state to
down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/12, changed state to
up

DLS1(config-if-range)#channel-group 2 mode desirable
DLS1(config-if-range)#
Creating a port-channel interface Port-channel 2

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to
down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/12, changed state to
down
```

Copy Paste

Once you have configured both sides, verify that the EtherChannel link is up

```
DLS2# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
-----
2      Po2(RU)         PAgP      Fa0/11(P)    Fa0/12(P)
```

```
DLS2# ping 172.16.12.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/3/9 ms
DLS2#
```

```
DLS2#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

2      Po2(RU)           PAgP    Fa0/11(P) Fa0/12(P)
DLS2#ping 172.16.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms

DLS2#
```

Copy Paste

Step 5: Configure default routing between DLS switches

At this point, local routing is supported at each distribution layer switch. To provide reachability across the layer 3 EtherChannel trunk, configure fully qualified static default routes at DLS1 and DLS2 that point to each other. From DLS1:

```
DLS1(config)# ip route 0.0.0.0 0.0.0.0 port-channel 2
%Default route without gateway, if not a point-to-point interface, may
impact performance
DLS1(config)# no ip route 0.0.0.0 0.0.0.0 port-channel 2
DLS1(config)# ip route 0.0.0.0 0.0.0.0 port-channel 2 172.16.12.2
DLS1(config)#
```

Once done at both ends, verify connectivity by pinging from one switch to the other. In the example below, DLS2 pings the Loopback 1 interface at DLS1.

```
DLS2# show ip route | begin Gateway
Gateway of last resort is 172.16.12.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.12.1, Port-channel2
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.110.0/24 is directly connected, Vlan110
L        10.1.110.1/32 is directly connected, Vlan110
```

```
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.12.0/30 is directly connected, Port-channel2
L        172.16.12.2/32 is directly connected, Port-channel2
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback1
L        192.168.1.1/32 is directly connected, Loopback1
DLS2#
DLS2# ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
DLS2#
```

```
DLS2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.110.0 is directly connected, Vlan110
     172.16.0.0/30 is subnetted, 1 subnets
C       172.16.12.0 is directly connected, Port-channel 2
C    192.168.1.0/24 is directly connected, Loopback1
DLS2#
DLS2#ping 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

DLS2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS2(config)#no shutdown
```

Copy Paste

Step 6: Configure the remaining EtherChannels for the topology

Configure the remaining EtherChannel links as layer 2 PagP trunks using VLAN 1 as the native VLAN.

Endpoint 1	Channel number	Endpoint 2	VLANs Allowed
ALS1 F0/7-8	1	DLS1 F0/7-8	All except 110
ALS1 F0/9-10	4	DLS2 F0/9-10	110 Only
ALS2 F0/7-8	3	DLS2 F0/7-8	All

Example from ALS1:

```
ALS1(config)# interface range f0/7-8
ALS1(config-if-range)# switchport mode trunk
ALS1(config-if-range)# switchport trunk allowed vlan except 110
ALS1(config-if-range)# channel-group 1 mode desirable
Creating a port-channel interface Port-channel 1
```

```
ALS1(config-if-range)# no shut
ALS1(config-if-range)# exit
ALS1(config)# interface range f0/9-10
ALS1(config-if-range)# switchport mode trunk
ALS1(config-if-range)# switchport trunk allowed vlan 110
ALS1(config-if-range)# channel-group 4 mode desirable
Creating a port-channel interface Port-channel 4

ALS1(config-if-range)# no shut
ALS1(config-if-range)# exit
ALS1(config)#end
ALS1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------
-----
1      Po1(SU)         PAgP      Fa0/7(P)    Fa0/8(P)
4      Po4(SU)         PAgP      Fa0/9(P)    Fa0/10(P)

ALS1# show interface trunk
Port        Mode             Encapsulation  Status        Native vlan
Po1         on               802.1q         trunking      1
Po4         on               802.1q         trunking      1

Port        Vlans allowed on trunk
Po1         1-109,111-4094
Po4         110
<output omitted>
ALS1#
```

```
ALS1#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

1      Po1(SD)           PAgP   Fa0/7(I) Fa0/8(I)
4      Po4(SD)           PAgP   Fa0/9(I) Fa0/10(I)
ALS1#
```

Copy Paste

Step 7: Enable and Verify Layer 3 connectivity across the network

In this step we will enable basic connectivity from the management VLANs on both sides of the network.

- Create the management VLANs (99 at ALS1, 120 at ALS2)
- Configure interface VLAN 99 at ALS1 and interface VLAN 120 at ALS2
- Assign addresses (refer to the diagram) and default gateways (at DLS1/DLS2 respectively).

Once that is all done, pings across the network should work, flowing across the layer 3 EtherChannel. An example from ALS2:

```
ALS2(config)# vlan 120
ALS2(config-vlan)# name MGMT2
ALS2(config-vlan)# exit
ALS2(config)# int vlan 120
ALS2(config-if)# ip address 10.1.120.2 255.255.255.0
ALS2(config-if)# no shut
ALS2(config-if)# exit
ALS2(config)# ip default-gateway 10.1.120.1
ALS2(config)# end

ALS2# ping 10.1.99.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.99.2, timeout is 2 seconds:
..!!!
```

```
Success rate is 60 percent (3/5), round-trip min/avg/max = 1/3/8 ms
ALS2#
ALS2# traceroute 10.1.99.2
Type escape sequence to abort.
Tracing the route to 10.1.99.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.120.1 0 msec 0 msec 8 msec
  2 172.16.12.1 0 msec 0 msec 8 msec
  3 10.1.99.2 0 msec 0 msec *
ALS2#
```

Part 2: Configure Multilayer Switching at ALS1

At this point all routing is going through the DLS switches, and the port channel between ALS1 and DLS2 is not passing anything but control traffic (BPDUs, etc).

The Cisco 2960 is able to support basic routing when it is using the LANBASE IOS. In this step you will configure ALS1 to support multiple SVIs and configure it for basic static routing. The objectives of this step are:

- Enable inter-vlan routing between two VLANs locally at ALS1
- Enable IP Routing
- Configure a static route for DLS2's Lo1 network travel via Port-Channel 4.

Step 1: Configure additional VLANs and VLAN interfaces

At ALS1, create VLAN 100 and VLAN 110 and then create SVIs for those VLANs:

```
ALS1(config)# ip routing
ALS1(config)# vlan 100
ALS1(config-vlan)# name LOCAL
ALS1(config-vlan)# exit
ALS1(config)# vlan 110
ALS1(config-vlan)# name INTERNODE
ALS1(config-vlan)# exit
ALS1(config)# int vlan 100
ALS1(config-if)# ip address 10.1.100.1 255.255.255.0
ALS1(config-if)# no shut
ALS1(config-if)# exit
ALS1(config)# int vlan 110
ALS1(config-if)# ip address 10.1.110.2 255.255.255.0
ALS1(config-if)# no shut
ALS1(config-if)# exit
ALS1(config)#
```

```
ALS1(config)#vlan 100
ALS1(config-vlan)#name LOCAL
ALS1(config-vlan)#exit
ALS1(config)#vlan 110
ALS1(config-vlan)#name INTERNODE
ALS1(config-vlan)#exit
ALS1(config)#int vlan 100
ALS1(config-if)#
%LINK-5-CHANGED: Interface Vlan100, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan100, changed state to up

ALS1(config-if)#ip add 10.1.100.1 255.255.255.0
ALS1(config-if)#no shu
ALS1(config-if)#no shutdown
ALS1(config-if)#exit
ALS1(config)#int vlan 110
ALS1(config-if)#
%LINK-5-CHANGED: Interface Vlan110, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan110, changed state to up

ALS1(config-if)#ip add 10.1.110.2 255.255.255.0
ALS1(config-if)#no shut
ALS1(config-if)#exit
ALS1(config)#
```

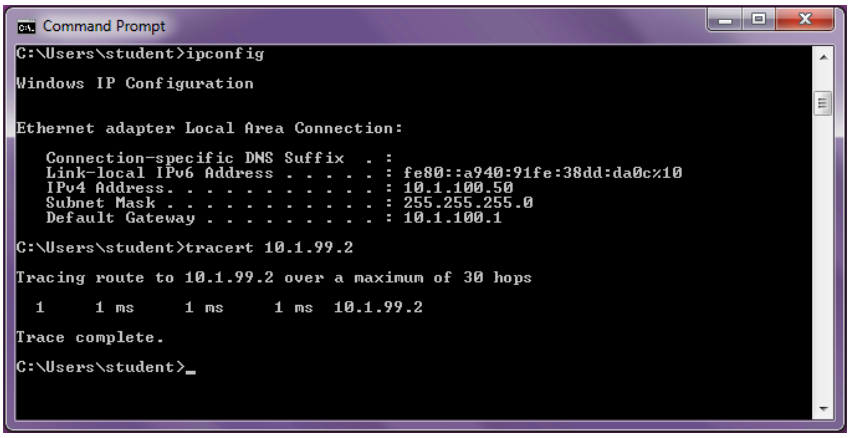
Copy Paste

Step 2: Configure and test Host Access

Assign interface Fa0/6 to VLAN 100. On the attached host (Host A) configure the IP address 10.1.100.50/24 with a default gateway of 10.1.100.1. Once configured, try a traceroute from the host to 10.1.99.2 and observe the results.

In the output below, the **`switchport host`** macro was used to quickly configure interface Fa0/6 with host-relative commands.

```
ALS1(config)# interface f0/6
ALS1(config-if)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled

ALS1(config-if)# switchport access vlan 100
ALS1(config-if)# no shut
ALS1(config-if)# exit
```



The output from the host shows that attempts to communicate with interface VLAN 99 at ALS1 were fulfilled locally, and not sent to DLS1 for routing.

Step 3: Configure and verify static routing across the network

At this point, local routing (at ALS1) works, and off-net routing (outside of ALS1) will not work, because DLS1 doesn't have any knowledge of the 10.1.100.0 subnet. In this step you will configure routing on several different switches:

- At DLS1, configure:
  - a static route to the 10.1.100.0/24 network via VLAN 99
- At DLS2, configure
  - a static route to the 10.1.100.0/24 network via VLAN 110
- At ALS1, configure
  - a static route to the 192.168.1.0/24 network via VLAN 110
  - a default static route to use 10.1.99.1

Here is an example from ALS1:

```
ALS1(config)# ip route 192.168.1.0 255.255.255.0 vlan 110
ALS1(config)# ip route 0.0.0.0 0.0.0.0 10.1.99.1
ALS1(config)# end
ALS1# show ip route | begin Gateway
Gateway of last resort is 10.1.99.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.1.99.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.1.99.0/24 is directly connected, Vlan99
L        10.1.99.2/32 is directly connected, Vlan99
C        10.1.100.0/24 is directly connected, Vlan100
L        10.1.100.1/32 is directly connected, Vlan100
C        10.1.110.0/24 is directly connected, Vlan110
L        10.1.110.2/32 is directly connected, Vlan110
```

```
S     192.168.1.0/24 is directly connected, Vlan110
```

After configuring all of the required routes, test to see that the network behaves as expected.

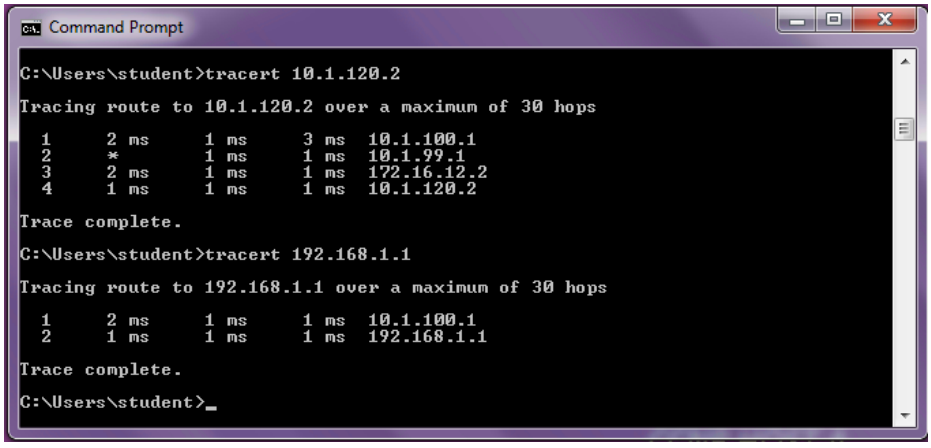
From ALS1, a traceroute to 10.1.120.2 should take three hops:

```
ALS1# traceroute 10.1.120.2
Type escape sequence to abort.
Tracing the route to 10.1.120.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.99.1 0 msec 0 msec 0 msec
  2 172.16.12.2 9 msec 0 msec 0 msec
  3 10.1.120.2 0 msec 8 msec *
ALS1#
```

From ALS1, a traceroute to 192.168.1.1 should take one hop:

```
ALS1# traceroute 192.168.1.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.110.1 0 msec 0 msec *
ALS1#
```

Traces from Host A show an additional hop, but follow the designated path:



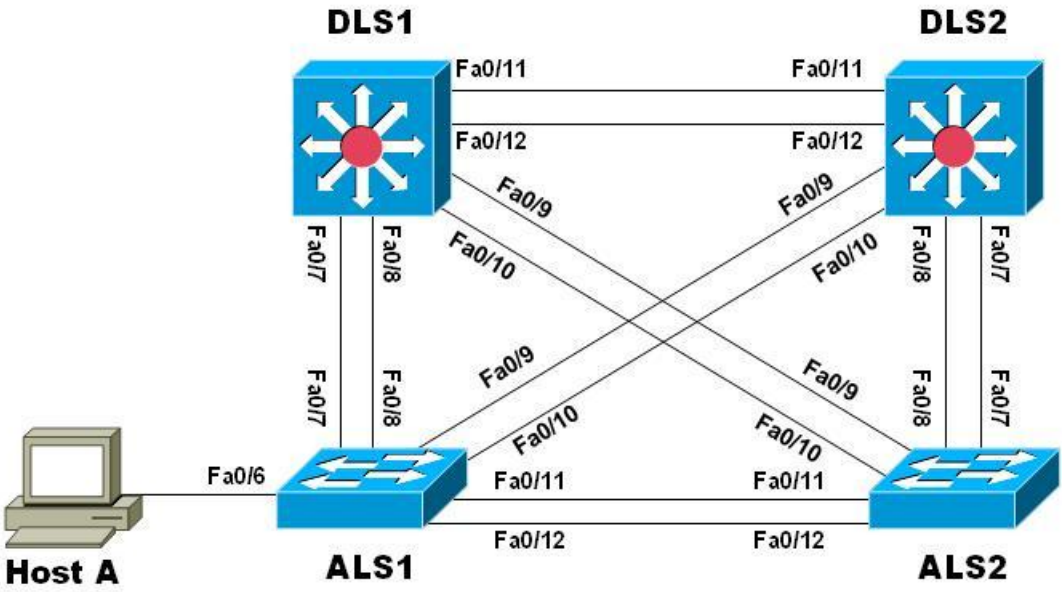
Step 4: End of Lab

Save your configurations. The switches will be used as configured now for lab 5-2, DHCP.

CCNPv7.1 SWITCH

Chapter 4 Lab 4-1– Implement Spanning Tree Protocols

Topology



Objectives

- Observe default Spanning Tree behavior
- Implement Rapid Spanning Tree
- Implement STP tool kit components

Background

The potential effect of a loop in the layer 2 network is significant. Layer 2 loops could impact connected hosts as well as the network equipment. Layer 2 loops can be prevented by following good design practices and careful implementation of the Spanning Tree Protocol. In this lab you will observe and manipulate the operation of spanning tree protocols to help secure the layer 2 network from loops and topology disruptions. The terms "switch" and "bridge" will be used interchangeably throughout the lab.

Note: This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates “dual-ipv4-and-ipv6 routing” and “lanbase-routing”, respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

- 2 Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- 2 Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Computer with terminal emulation software
- Ethernet and console cables
- 1 Windows 7 PC with Wireshark, TCPDump, or another comparable packet capture utility installed

Part 4: Observe default Spanning Tree behavior

Step 1: Load base config and configure trunks

Use the **reset.tcl** script you created in Lab 1 "Preparing the Switch" to set your switches up for this lab. Then load the file BASE.CFG into the running-config with the command **copy flash:BASE.CFG running-config**.

Perform this step on all four switches. An example from DLS1:

```
DLS1# tclsh reset.tcl
Erasing the nvram filesystem will remove all configuration files! Continue?
[confirm]
[OK]
Erase of nvram: complete
Reloading the switch in 1 minute, type reload cancel to halt

Proceed with reload? [confirm]

*Mar  7 18:41:40.403: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of
nvram
*Mar  7 18:41:41.141: %SYS-5-RELOAD: Reload requested by console. Reload
Reason: Reload command.
<switch reloads - output omitted>
Would you like to enter the initial configuration dialog? [yes/no]: n
Switch> en
*Mar  1 00:01:30.915: %LINK-5-CHANGED: Interface Vlan1, changed state to
administratively down
Switch# copy BASE.CFG running-config
Destination filename [running-config]?
184 bytes copied in 0.310 secs (594 bytes/sec)
DLS1#
```

Next, enable interfaces F0/7 through F0/12 as 802.1Q trunk ports. Perform this step on all four switches. An example from DLS1:

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# int ran f0/7-12
DLS1(config-if-range)# switchport trunk encap dot1q
DLS1(config-if-range)# switchport trunk native vlan 666
DLS1(config-if-range)# switchport trunk allowed vlan except 1,999
DLS1(config-if-range)# switchport mode trunk
DLS1(config-if-range)# switchport nonegotiate
DLS1(config-if-range)# no shut
DLS1(config-if-range)# exit
DLS1(config)#
```

Finally, configure all four switches as VTP version 3 servers in domain SWLAB with no password. An example from DLS1:

```
DLS1(config)# vtp mode server
Setting device to VTP Server mode for VLANS.
DLS1(config)# vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
DLS1(config)# vtp version 3
DLS1(config)#
```

Step 2: Configure VLANs

Configure DLS1 as the VTP primary server for VLANs, and then create VLANs. The VLAN database will propagate to the other switches in the network.

```
DLS1# vtp primary vlan
This system is becoming primary server for feature vlan
No conflicting VTP3 devices found.
Do you want to continue? [confirm]
DLS1#
*Mar  1 01:35:22.917: %SW_VLAN-4-VTP_PRIMARY_SERVER_CHG: e840.406f.7280 has
become the primary server for the VLAN VTP feature
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# vlan 99
DLS1(config-vlan)# name MANAGEMENT
DLS1(config-vlan)# vlan 100
DLS1(config-vlan)# name SERVERS
DLS1(config-vlan)# vlan 110
DLS1(config-vlan)# name GUEST
DLS1(config-vlan)# vlan 120
DLS1(config-vlan)# name OFFICE
DLS1(config-vlan)# vlan 999
DLS1(config-vlan)# name PARKING_LOT
DLS1(config-vlan)# state suspend
DLS1(config-vlan)# vlan 666
```

```
DLS1(config-vlan)# name NATIVE_DO_NOT_USE
DLS1(config-vlan)# exit
DLS1(config)#

ALS2# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 3
VTP Domain Name                 : SWLAB
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 5017.ff84.0a80

Feature VLAN:
--------------
VTP Operating Mode                : Server
Number of existing VLANs          : 11
Number of existing extended VLANs : 0
Maximum VLANs supported locally   : 255
Configuration Revision            : 7
Primary ID                        : e840.406f.7280
Primary Description               : DLS1
MD5 digest                        : 0xF3 0xD5 0xF7 0x62 0x3F 0x7C 0x84 0x86
                                    0x41 0xC0 0x4E 0xCA 0x36 0xB8 0x15 0x47
<output omitted>

ALS2# show vlan brief | i active
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
99   MANAGEMENT                       active
100  SERVERS                          active
110  GUEST                            active
120  OFFICE                           active
666  NATIVE_DO_NOT_USE                active
ALS2#
```

Step 3: Identify and modify the root bridge

Use the **show span root** command on all of the switches to find the root switch for all of the VLANs. Note: Your results may vary from the examples.

```
DLS1# show span root
                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0099         32867 5017.ff84.0a80        19    2   20  15  Fa0/9
VLAN0100         32868 5017.ff84.0a80        19    2   20  15  Fa0/9
VLAN0110         32878 5017.ff84.0a80        19    2   20  15  Fa0/9
VLAN0120         32888 5017.ff84.0a80        19    2   20  15  Fa0/9
VLAN0666         33434 5017.ff84.0a80        19    2   20  15  Fa0/9
```

```
DLS1#

ALS2# show span root
                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0099         32867 5017.ff84.0a80         0    2   20  15
VLAN0100         32868 5017.ff84.0a80         0    2   20  15
VLAN0110         32878 5017.ff84.0a80         0    2   20  15
VLAN0120         32888 5017.ff84.0a80         0    2   20  15
VLAN0666         33434 5017.ff84.0a80         0    2   20  15
ALS2#
```

Compare the output of the **show span** command on all of the switches; why did the current root get elected?

```
DLS2#show span vlan 99
VLAN0099
  Spanning tree enabled protocol ieee
  Root ID     Priority    32867
              Address     5017.ff84.0a80
              Cost        19
              Port        11 (FastEthernet0/9)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32867  (priority 32768 sys-id-ext 99)
              Address     e840.406f.7280
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Altn BLK 19        128.9    P2p
Fa0/8               Altn BLK 19        128.10   P2p
Fa0/9               Root FWD 19        128.11   P2p
Fa0/10              Altn BLK 19        128.12   P2p
Fa0/11              Altn BLK 19        128.13   P2p
Fa0/12              Altn BLK 19        128.14   P2p
```

The current root bridge was elected based on the lowest Bridge ID (consisting of the Priority, extended system ID equal to the VLAN ID, and base MAC address values). In the output above, the root's MAC is 5017.ff84.0a80; the local bridge MAC is e840.406f.7280.

With the priority and extended system IDs being identical, the root bridge's MAC is numerically smaller than the local bridge's MAC. The end result is that in a completely un-configured network, one single switch will be elected as the root bridge. The resulting choice of switch may or may not be desirable.

There are two basic ways to manipulate the configuration to control the location of the root bridge.

- The **spanning-tree vlan** *vlan-id* **priority** *value* command can be used to manually set a priority value
- The **spanning-tree vlan** *vlan-id* **root { primary | secondary }** command can be used to automatically set a priority value.

The difference between the two is the **priority** command will set a specific number (multiple of 4096) as the priority, while the **root primary** command will set the local bridge's priority to 24,576 (if the local bridge MAC is lower than the current root bridge's MAC) or 4096 lower than the current root's priority (if the local bridge MAC is higher than the current root bridge's MAC).

The logic behind this operation is straight-forward. The **root primary** command tries to lower the priority only as much as is needed to win the root election, while leaving priorities between 24576 and the default 32768 for use by secondary bridges. The command always takes the entire Bridge ID into account when computing the resulting priority value.

The **spanning-tree vlan** *vlan-id* **root secondary** command will statically set the local bridge's priority to 28,672. In an otherwise unconfigured network where all switch priorities default to 32,768, the **root primary** command will set the priority on the switch to 24,576 (two "steps" lower than the default priority) while the **root secondary** command will set the priority on the secondary root to the 28,672 (one "step" lower than the default priority).

Modify DLS1 and DLS2 so that DLS 1 is elected the primary root bridge for VLANs 99 and 100 and DLS2 is elected the primary root bridge for VLAN 110 and 120. DLS1 should be elected as the secondary root bridge for VLAN 110 and 120, and DLS2 should be elected as the secondary root bridge for VLANs 99, and 100.

You will need to make configuration changes on both DLS1 and DLS2.

An example from DLS1:

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# spanning-tree vlan 99,100 root primary
DLS1(config)# spanning-tree vlan 110,120 root secondary
DLS1(config)# exit
DLS1#
```

Verification from DLS1:

```
DLS1# show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0099         24675 e840.406f.7280         0    2   20  15
VLAN0100         24676 e840.406f.7280         0    2   20  15
VLAN0110         24686 e840.406f.6e00        19    2   20  15  Fa0/11
VLAN0120         24696 e840.406f.6e00        19    2   20  15  Fa0/11
VLAN0666         33434 5017.ff84.0a80        19    2   20  15  Fa0/9
```

```
DLS1#
```

The **show spanning-tree bridge** command also provides detailed information about the current configuration of the local bridge:

```
DLS1# show spanning-tree bridge ?
  address        Mac address of this bridge
  detail         Detailed of the status and configuration
  forward-time   Forward delay interval
  hello-time     Hello time
  id             Spanning tree bridge identifier
  max-age        Max age
  priority       Bridge priority of this bridge
  protocol       Spanning tree protocol
  |              Output modifiers
  <cr>
```

```
DLS1# show spanning-tree bridge

                                                        Hello  Max  Fwd
Vlan                         Bridge ID                   Time  Age  Dly
Protocol
---------------- --------------------------------- ------  ---  ---  -------
-
VLAN0099         24675 (24576,  99) e840.406f.7280    2    20   15  ieee
VLAN0100         24676 (24576, 100) e840.406f.7280    2    20   15  ieee
VLAN0110         28782 (28672, 110) e840.406f.7280    2    20   15  ieee
VLAN0120         28792 (28672, 120) e840.406f.7280    2    20   15  ieee
VLAN0666         33434 (32768, 666) e840.406f.7280    2    20   15  ieee
DLS1#
```

Step 4: Manipulate port and path costs

As the network is implemented right now, there are two paths between each directly connected switch. As the Root Port is elected, path and port costs are evaluated to determine the shortest path to the root bridge.

In the case where there are multiple equal cost paths to the root bridge, additional attributes must be evaluated. In our case, the lower interface number (for example, F0/11) is chosen as the Root Port, and the higher interface number (for example, F0/12) is put into a spanning tree Blocking state.

You can see which ports are blocked with the **show spanning-tree** *vlan-id* command or the **show spanning-tree blockedports** command. For now examine VLAN 110 on DLS1.

```
DLS1# show spanning-tree vlan 110

VLAN0110
  Spanning tree enabled protocol ieee
  Root ID     Priority    24686
              Address     e840.406f.6e00
              Cost        19
              Port        13 (FastEthernet0/11)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28782  (priority 28672 sys-id-ext 110)
              Address     e840.406f.7280
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Desg FWD 19        128.9    P2p
Fa0/8               Desg FWD 19        128.10   P2p
Fa0/9               Desg FWD 19        128.11   P2p
Fa0/10              Desg FWD 19        128.12   P2p
Fa0/11              Root FWD 19        128.13   P2p
Fa0/12              Altn BLK 19        128.14   P2p


DLS1# show spanning-tree blockedports

Name                 Blocked Interfaces List
-------------------- ------------------------------------
VLAN0110             Fa0/12
VLAN0120             Fa0/12
VLAN0666             Fa0/7, Fa0/8, Fa0/10, Fa0/11, Fa0/12

Number of blocked ports (segments) in the system : 7
```

As you can see, VLAN 110 has its Root Port on Fa0/11 and Fa0/12 is an Alternate Blocking Port. Note that despite the switch not yet running Rapid STP, it recognizes the port roles as known by RSTP.

It is possible to manipulate which port becomes the Root Port on non-root bridges by manipulating the port cost value, or by changing the port priority value. Remember that this

change could have an impact on downstream switches as well. For this example, we will examine both options.

Note: The changes you are about to implement are considered topology changes and *could* have a significant impact on the overall structure of the spanning tree in your switch network. Do not make these changes in a production network without careful planning and prior coordination.

The first change you will make will influence the Root Port election based on a change to the port cost. We will further examine the impact of the changes to downstream switches.

To do this, issue the **shutdown** command on interfaces Fa0/9 and Fa0/10 on DLS1 and DLS2. Example from DLS1:

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# int ran f0/9-10
DLS1(config-if-range)# shut
DLS1(config-if-range)# exit
DLS1(config)#
```

Then, examine the VLAN 110 root values on ALS1:

```
ALS1# show span root

                                        Root    Hello Max Fwd
Vlan                 Root ID            Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0099         24675 e840.406f.7280         19    2   20  15  Fa0/7
VLAN0100         24676 e840.406f.7280         19    2   20  15  Fa0/7
VLAN0110         24686 e840.406f.6e00         38    2   20  15  Fa0/7
VLAN0120         24696 e840.406f.6e00         38    2   20  15  Fa0/7
VLAN0666         33434 5017.ff84.0a80         19    2   20  15  Fa0/11
ALS1#
```

The election of the Root Port is based on the lowest total path cost to the root bridge. The root path cost is a sum of all of the Root Port costs between the local bridge and the root bridge. If the total path cost to the root bridge is the same over multiple ports, then the port towards the neighbor switch that has the lowest Bridge ID is chosen as the Root Port.

If the local bridge has multiple connections to a neighbor bridge that is in the lowest-cost path, BDPUs sent from that neighbor are examined and the BPDU containing the lowest sending Port-ID is chosen as the Root Port. In this case the term "sending" refers to the switch and its port that *forwarded* the BPDU.

Notice in the output above that the root bridge for VLAN110 is reachable from ALS1 via Fa0/7 with a total root path cost of 38 (19 for the Fa0/7 trunk between ALS1 and DLS1, and 19 for the trunk between DLS1 and DLS2).

On ALS2, change the spanning tree cost on interface Fa0/7 to 12.

```
ALS2(config)# int f0/7
ALS2(config-if)# spanning-tree cost 12
ALS2(config-if)# exit
ALS2(config)#
```

Now go back to ALS1 and see the impact:

```
ALS1# show span root

                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0099         24675 e840.406f.7280        19    2   20  15  Fa0/7
VLAN0100         24676 e840.406f.7280        19    2   20  15  Fa0/7
VLAN0110         24686 e840.406f.6e00        31    2   20  15  Fa0/11
VLAN0120         24696 e840.406f.6e00        31    2   20  15  Fa0/11
VLAN0666         33434 5017.ff84.0a80        19    2   20  15  Fa0/11
ALS1#
```

ALS1's Root Port changed to F0/11, and the path cost to the Root Bridge changed to 31 (19 + 12).

The change you just made on ALS2 did not impact the Root Port from its perspective; it is still Fa0/7.

Next you will use port priority to modify which port is selected as the Root Port. For this exercise, we will focus on VLAN 100.

On DLS1, use **`show span vlan 100`** to see what the priorities are (default to 128)

```
DLS1# show span vlan 100

VLAN0100
  Spanning tree enabled protocol ieee
  Root ID     Priority    24676
              Address     e840.406f.7280
              This bridge is the root
```

```
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24676  (priority 24576 sys-id-ext 100)
             Address     e840.406f.7280
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Desg FWD 19        128.9    P2p
Fa0/8               Desg FWD 19        128.10   P2p
Fa0/11              Desg FWD 19        128.13   P2p
Fa0/12              Desg FWD 19        128.14   P2p
```

In the output above, focus on interface Fa0/7. Notice that its Port ID is made up of two values, labeled as Prio (Priority) and Nbr (Number): The priority number (128) and the port number (9).

The port number is not necessarily equal to the interface ID. On the 3560s used for creating this lab, port numbers 1 and 2 are assigned to G0/1 and G0/2 respectively, whereas on the 2960s G0/1 and G0/2 area assigned the port numbers 25 and 26. A switch may use any port number for STP purposed as long as they are unique for each port on the switch.

The port priority can be any value between 0 and 240, in increments of 16 (older switches may allow setting the priority in different increments).

Next, examine ALS1 to find the root port for VLAN 100:

```
ALS1# show span root | i VLAN0100
VLAN0100         24676 e840.406f.7280        19    2   20  15  Fa0/7
ALS1#
```

On DLS1, change the port priority value of Fa0/8 to 112:

```
DLS1(config)# int f0/8
DLS1(config-if)# spanning-tree port-priority 112
DLS1(config-if)# exit
```

And then examine the impact on ALS1:

```
ALS1# show span root | i VLAN0100
VLAN0100         24676 e840.406f.7280        19    2   20  15  Fa0/8

ALS1# show span vlan 100

VLAN0100
  Spanning tree enabled protocol ieee
```

```
  Root ID     Priority    24676
              Address     e840.406f.7280
              Cost        19
              Port        8 (FastEthernet0/8)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32868  (priority 32768 sys-id-ext 100)
              Address     64a0.e72a.2200
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Altn BLK 19        128.7    P2p
Fa0/8               Root FWD 19        128.8    P2p
Fa0/11              Desg FWD 19        128.11   P2p
Fa0/12              Desg FWD 19        128.12   P2p
```

Notice that the priority value at ALS1 doesn't change, but the Root Port did, based on DLS1's advertised port priorities.

Step 5: Examine Re-convergence Time

Use the **debug spanning-tree events command** on DLS1 and watch how long re-convergence takes when interface Fa0/11 on DLS1 is shut down (Fa0/11 is DLS1's Root Port for VLAN 110). The output below has been manually filtered for VLAN 110 related messages only:

```
DLS1# debug span eve
Spanning Tree event debugging is on
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# int f0/11
DLS1(config-if)# shut
DLS1(config-if)#
*Mar  1 02:08:50.781: STP: VLAN0110 new root port Fa0/12, cost 19
*Mar  1 02:08:50.789: STP: VLAN0110 Fa0/12 -> listening
*Mar  1 02:08:50.789: STP[110]: Generating TC trap for port
FastEthernet0/11
*Mar  1 02:08:50.7
*Mar  1 02:08:52.769: %LINK-5-CHANGED: Interface FastEthernet0/11, changed
state to administratively down
*Mar  1 02:08:52.786: STP: VLAN0110 sent Topology Change Notice on Fa0/12
*Mar  1 02:09:05.797: STP: VLAN0110 Fa0/12 -> learning
*Mar  1 02:09:20.804: STP[110]: Generating TC trap for port
FastEthernet0/12
*Mar  1 02:09:20.804: STP: VLAN0110 sent Topology Change Notice on Fa0/12
*Mar  1 02:09:20.804: STP: VLAN0110 Fa0/12 -> forwarding
```

```
DLS1(config-if)# do sho span vlan 110

VLAN0110
  Spanning tree enabled protocol ieee
  Root ID     Priority    24686
              Address     e840.406f.6e00
              Cost        19
              Port        14 (FastEthernet0/12)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28782  (priority 28672 sys-id-ext 110)
              Address     e840.406f.7280
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Desg FWD 19        128.9    P2p
Fa0/8               Desg FWD 19        112.10   P2p
Fa0/12              Root FWD 19        128.14   P2p
```

As you can see from the timestamps, it took a full 30 seconds for PVST to settle on Fa0/12 as the root port and move to the Forwarding state on the Designated Ports. When Fa0/11 is reactivated:

```
DLS1(config-if)#no shut
*Mar  1 02:12:28.902: set portid: VLAN0110 Fa0/11: new port id 800D
*Mar  1 02:12:28.902: STP: VLAN0110 Fa0/11 -> listening
*Mar  1 02:12:29.900: STP: VLAN0110 new root port Fa0/11, cost 19
*Mar  1 02:12:29.900: STP: VLAN0110 sent Topology Change Notice on Fa0/11
*Mar  1 02:12:29.900: STP [110]: Generating TC trap for port
FastEthernet0/12
*Mar  1 02:12:29.900: STP: VLAN0110 Fa0/12 -> blocking
*Mar  1 02:12:43.909: STP: VLAN0110 Fa0/11 -> learning
*Mar  1 02:12:58.916: STP[110]: Generating TC trap for port
FastEthernet0/11
*Mar  1 02:12:58.916: STP: VLAN0110 sent Topology Change Notice on Fa0/11
*Mar  1 02:12:58.916: STP: VLAN0110 Fa0/11 -> forwarding
```

The re-convergence process took another full 30 seconds.

Part 5: Implement Rapid Spanning Tree

By default Cisco switches are running Per-VLAN Spanning Tree, which is a Cisco-proprietary protocol derived from the IEEE 802.1D standard.

```
DLS1# show span detail
```

```
VLAN0099 is executing the ieee compatible Spanning Tree protocol
<output omitted>
```

The running configuration shows you the protocol being used

```
DLS1# show run | inc spanning-tree mode
spanning-tree mode pvst
DLS1#
```

The issue with PVST is that its convergence is quite slow. The time for the transition between port states is called *forward-delay* and by default, it is 15 seconds. In addition, the time until a BPDU stored on a port expires is called *max-age* and is 20 seconds by default. Depending on the nature of a topological change, STP requires between 30 and 50 seconds to converge on a new loop-free topology.

Rapid Spanning Tree significantly reduces the time it takes to go from the Discarding (PVST: Blocking) to the Forwarding state.

Configure Rapid Spanning Tree Protocol on DLS1. Use the **`clear spanning-tree detected-protocols`** privileged EXEC command to flush any stored PVST information.

```
DLS1# conf t
DLS1(config)# spanning-tree mode rapid-pvst
DLS1(config)# end
DLS1#clear spanning-tree detected-protocols
DLS1#
```

Then verify the protocol. Use the **`show span vlan 110`** command:

```
DLS1# show span vlan 110

VLAN0110
  Spanning tree enabled protocol rstp
  Root ID     Priority    24686
              Address     e840.406f.6e00
              Cost        19
              Port        13 (FastEthernet0/11)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28782  (priority 28672 sys-id-ext 110)
              Address     e840.406f.7280
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
```

```
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Desg LRN 19        128.9    P2p Peer(STP)
Fa0/8               Desg LRN 19        112.10   P2p Peer(STP)
Fa0/11              Root FWD 19        128.13   P2p Peer(STP)
Fa0/12              Altn BLK 19        128.14   P2p Peer(STP)
```

Take note of the Type field in the output. All of the other switches are still running PVST, which is noted here by the entry Peer(STP).

Configure the rest of the switches to use Rapid Spanning Tree Protocol, then verify the protocol is running. An example from DLS2:

```
DLS2# show span vlan 99

VLAN0099
  Spanning tree enabled protocol rstp
```

To examine the impact of Rapid Spanning Tree on convergence time, use the **debug spanning-tree events command** on DLS1 and watch how long re-convergence takes when interface Fa0/11 on DLS1 is shut down (Fa0/11 is DLS1's Root Port for VLAN 110). The output below has been manually filtered for VLAN 110 related messages only:

```
DLS1# debug spanning-tree events
Spanning Tree event debugging is on
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# int f0/11
DLS1(config-if)# shut
DLS1(config-if)#
*Mar  1 02:18:53.201: RSTP(110): updt roles, root port Fa0/11 going down
*Mar  1 02:18:53.201: RSTP(110): Fa0/12 is now root port
*Mar  1 02:18:53.201: RSTP(110): syncing port Fa0/7
*Mar  1 02:18:53.201: RSTP(110): syncing port Fa0/8
*Mar  1 02:18:53.226: STP[110]: Generating TC trap for port
FastEthernet0/12
*Mar  1 02:18:53.242: RSTP(110): transmitting a proposal on Fa0/7
*Mar  1 02:18:53.242: RSTP(110): transmitting a proposal on Fa0/8
*Mar  1 02:18:55.189: %LINK-5-CHANGED: Interface FastEthernet0/11, changed
state to administratively down
*Mar  1 02:18:56.195: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/11, changed state to down
```

In the output above, the change of Root Port and synchronization to interfaces Fa0/7 and Fa0/8 took less than six-tenths of a second

```
DLS1(config-if)# no shut
DLS1(config-if)#
```

```
*Mar  1 02:22:38.368: RSTP(110): initializing port Fa0/11
*Mar  1 02:22:38.368: RSTP(110): Fa0/11 is now designated
*Mar  1 02:22:38.393: RSTP(110): transmitting a proposal on Fa0/11
*Mar  1 02:22:38.401: RSTP(110): updt roles, received superior bpdu on
Fa0/11
*Mar  1 02:22:38.401: RSTP(110): Fa0/11 is now root port
*Mar  1 02:22:38.401: RSTP(110): Fa0/12 blocked by re-root
*Mar  1 02:22:38.409: RSTP(110): syncing port Fa0/7
*Mar  1 02:22:38.409: RSTP(110): syncing port Fa0/8
*Mar  1 02:22:38.409: RSTP(110): synced Fa0/11
*Mar  1 02:22:38.409: RSTP(110): Fa0/12 is now alternate
*Mar  1 02:22:38.418: STP[110]: Generating TC trap for port
FastEthernet0/11
*Mar  1 02:22:38.435: RSTP(110): transmitting an agreement on Fa0/11 as a
response to a proposal
*Mar  1 02:22:38.435: RSTP(110): transmitting a proposal on Fa0/7
*Mar  1 02:22:38.435: RSTP(110): transmitting a proposal on Fa0/8
*Mar  1 02:22:38.653: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed
state to up
*Mar  1 02:22:39.659: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/11, changed state to up
```

In the output above, Fa0/11 is brought back up, and the change of root port and synchronization to interfaces Fa0/7 and Fa0/8 took one half second. This is a significant improvement from standard spanning tree.

Part 6: Part 3: Implement STP tool kit components

Step 1: Implement and observe PortFast

In both STP and RSTP, a newly connected port must be guaranteed not to create a switching loop before it can become a Forwarding port. This may take up to 30 seconds. However, such a check is not necessary for ports connected to end devices that do not perform switching or bridging, such as workstations, network printers, servers, etc. In RSTP, these ports are called *edge* ports (ports that connect to other switches in the topology are called *non-edge* ports). Edge ports can safely enter the Forwarding state right after they come up, because by definition they do not connect to any device capable of forwarding frames.

Cisco developed a feature called PortFast that essentially allows defining a port as an edge port. Any PortFast enabled port will enter the Forwarding state immediately after coming up, without going through the intermediary non-forwarding states, saving 30 seconds each time a new connection is made to the port. PortFast can be used with all STP versions.

Apart from allowing a port to jump into the Forwarding state as soon as it is connected, the concept of an edge port is extremely important in RSTP and MSTP. Recall that as part of its improvements over legacy STP, RSTP uses a so-called Proposal/Agreement mechanism to rapidly, yet safely enable a link between switches if one of the switches has its Root port on that link.

Upon receiving a Proposal on its Root port, a switch puts all its non-edge Designated ports into the Discarding state, effectively cutting itself off the network and preventing a possible switching loop (this is called the Sync operation). When this is accomplished, the switch sends an Agreement back out its Root port so that the upstream Designated port receiving this Agreement can be immediately put into the Forwarding state. The switch will then start sending its own Proposals on all its non-edge Designated ports that have been just made Discarding, waiting for Agreements to arrive from downstream switches that would allow these ports to instantaneously become Forwarding again.

If end devices are connected to ports not configured as edge (that is, PortFast) ports, these ports will become Discarding during the Sync operation. Because end hosts do not support RSTP and cannot send back an Agreement, they will be cut off from the network for 30 seconds until the ports reach the Forwarding state using ordinary timers. As a result, users will experience significant connectivity outages.

Ports configured as edge ports are not affected by the Sync operation and will remain in the Forwarding state even during the Proposal/Agreement handling. Activating RSTP in a network without properly configuring ports toward end hosts as edge ports will cause the network to perform possibly even more poorly than with legacy STP. With RSTP, proper configuration of ports toward end hosts as edge ports is an absolute necessity. Cisco switches default to all their ports being non-edge ports.

For this lab step to work, Host A must be connected to ALS1's Fa0/6.

On ALS1, issue the command **`debug spanning-tree events`** , then configure F0/6 to be in VLAN 120.

Finally, **`no shut`** the interface, **`shut`** the interface, and observe the syslog

```
ALS1# debug spanning-tree events
Spanning Tree event debugging is on
ALS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ALS1(config)# int f0/6
ALS1(config-if)# swi mo ac
ALS1(config-if)# swi ac vl 120
ALS1(config-if)# no shut
ALS1(config-if)#
*Mar  1 02:26:47.778: RSTP(120): initializing port Fa0/6
*Mar  1 02:26:47.778: RSTP(120): Fa0/6 is now designated
*Mar  1 02:26:47.786: RSTP(120): transmitting a proposal on Fa0/6
*Mar  1 02:26:48.197: %LINK-3-UPDOWN: Interface FastEthernet0/6, changed
state to up
*Mar  1 02:26:48.759: RSTP(120): transmitting a proposal on Fa0/6
*Mar  1 02:26:49.204: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/6, changed state to up
*Mar  1 02:26:50.772: RSTP(120): transmitting a proposal on Fa0/6
*Mar  1 02:26:52.786: RSTP(120): transmitting a proposal on Fa0/6
*Mar  1 02:26:54.799: RSTP(120): transmitting a proposal on Fa0/6
*Mar  1 02:26:56.812: RSTP(120): transmitting a proposal on Fa0/6
ALS1(config-if)# shut
*Mar  1 02:26:58.825: RSTP(120): transmitting a proposal on Fa0/6
```

```
*Mar  1 02:27:02.441: %LINK-5-CHANGED: Interface FastEthernet0/6, changed
state to administratively down
*Mar  1 02:27:03.448: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/6, changed state to down
ALS1(config-if)#
```

As you can see in the output above, RSTP sees the interface come up, recognizes it as a Designated port, and starts sending proposals. Now we will add the **spanning-tree portfast** command to the interface (the debug is still running):

```
ALS1(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on FastEthernet0/6 but will only
 have effect when the interface is in a non-trunking mode.
ALS1(config-if)#
ALS1(config-if)# no shut
ALS1(config-if)#
*Mar  1 02:28:13.534: RSTP(120): initializing port Fa0/6
*Mar  1 02:28:13.534: RSTP(120): Fa0/6 is now designated
*Mar  1 02:28:13.945: %LINK-3-UPDOWN: Interface FastEthernet0/6, changed
state to up
*Mar  1 02:28:14.952: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/6, changed state to up
ALS1(config-if)# shut
*Mar  1 02:28:35.999: %LINK-5-CHANGED: Interface FastEthernet0/6, changed
state to administratively down
*Mar  1 02:28:37.006: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/6, changed state to down
ALS1(config-if)# no shut
ALS1(config-if)#
*Mar  1 02:28:51.434: RSTP(120): initializing port Fa0/6
*Mar  1 02:28:51.434: RSTP(120): Fa0/6 is now designated
ALS1(config-if)#
*Mar  1 02:28:51.761: %LINK-3-UPDOWN: Interface FastEthernet0/6, changed
state to up
*Mar  1 02:28:52.768: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/6, changed state to up
```

Disable the debug using the **undebug all** command.

Notice in output above that with PortFast configured, no proposals are sent out of interface Fa0/6; the port goes into Forwarding state immediately.

```
ALS1# show span detail | beg VLAN0120
 VLAN0120 is executing the rstp compatible Spanning Tree protocol
<output omitted>

 Port 6 (FastEthernet0/6) of VLAN0120 is designated forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.6.
   Designated root has priority 24696, address e840.406f.6e00
   Designated bridge has priority 32888, address 64a0.e72a.2200
   Designated port id is 128.6, designated path cost 31
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast mode
   Link type is point-to-point by default
   BPDU: sent 39, received 0
```

Note(1): PortFast should never be enabled on ports connected to another switches. Doing so could cause a switching loop. RSTP and MSTP have their own mechanisms to put inter-switch links into Forwarding state rapidly.

Note(2): On trunk interfaces, configuring the **`spanning-tree portfast`** command will have no effect. This is a safety precaution, as trunks are usually connected to other switches. However, in situations like inter-VLAN routing using a router-on-stick, or when a trunk is being connected to a server that operates on multiple VLANs simultaneously, it may still be advantageous, and safe, to allow this trunk to be treated as an edge port and become Forwarding as soon as it is connected. In these cases, you can use the **`spanning-tree portfast trunk`** command on a trunk port to force a switch to treat it as an edge port regardless of its operating mode. Be absolutely sure that the device connected to such port is not performing Layer2 switching before using this command.

Note(3): Because the proper configuration of edge ports in RSTP and MSTP is of such great importance for proper network performance, Cisco also provides the way of globally configuring the PortFast on all access ports using the **`spanning-tree portfast default`** global configuration command. With this command configured, each port that operates in the access mode will automatically have PortFast enabled. Trunk ports will not be affected. The logic of this behavior is simple: Usually, trunk ports connect to other switch where PortFast should never be enabled, while access ports usually connect to end devices.

Step 2: Implement and Observe BPDU Guard

PortFast causes an interface to go into Forwarding state immediately. There is a risk that if two PortFast-enabled ports are inadvertently or maliciously connected together, they will both come up as Forwarding ports, immediately creating a switching loop.

The default, expected behavior of a PortFast port that receives a BPDU is for that port to revert to a normal spanning-tree non-edge port. There is the potential that the load on a given switch might be too great to handle the received BPDU properly, prolonging the loop condition.

BPDU Guard adds another layer of protection. Whenever a port protected by BPDU Guard unexpectedly receives a BPDU, it is immediately put into err-disabled state. Any interfaces can be protected with BPDU Guard, but its typical use in on PortFast-enabled ports.

BPDU Guard can be configured globally, or on a per-interface basis. If the BPDU Guard is configured on the global level using the **`spanning-tree portfast bpduguard default`** command, the BPDU Guard will be automatically enabled on all PortFast-enabled ports of the switch. If the BPDU Guard is configured on a particular interface using the **`spanning-tree`**

bpduguard enable command, it will apply to this port unconditionally, regardless of whether it is a PortFast-enabled port.

For this example, we will configure BPDU guard on a trunking interface that is a non-root port on ALS2. Configuring BPDU Guard on an interface that is intended to be a trunk is *not a recommended practice*; we are doing this just to demonstrate the functionality of the tool.

```
ALS2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ALS2(config)# int f0/11
ALS2(config-if)# spanning-tree bpduguard enable
ALS2(config-if)# exit
ALS2(config)#
*Mar  1 02:30:57.792: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port
Fa0/11 with BPDU Guard enabled. Disabling port.
*Mar  1 02:30:57.792: %PM-4-ERR_DISABLE: bpduguard error detected on
Fa0/11, putting Fa0/11 in err-disable state
*Mar  1 02:30:58.798: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/11, changed state to down
*Mar  1 02:30:59.813: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed
state to down
```

As you can see, the interface is almost immediately err-disabled. Revert the configuration settings and issue the **shutdown** and **no shutdown** commands on Fa0/11 to bring it back up.

```
ALS2(config)# int f0/11
ALS2(config-if)# shut
ALS2(config-if)# spanning-tree bpduguard disable
ALS2(config-if)# no shut
ALS2(config-if)# exit
ALS2(config)#
*Mar  1 02:32:06.092: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed
state to up
*Mar  1 02:32:07.107: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/11, changed state to up
```

Step 3: Implement and Observe BPDU Filter

Neither PortFast nor BPDU Guard prevents the switch from sending BPDUs on an interface; if such a behavior is required, BPDU Filter can be used. It can be configured either globally or at a specific interface.

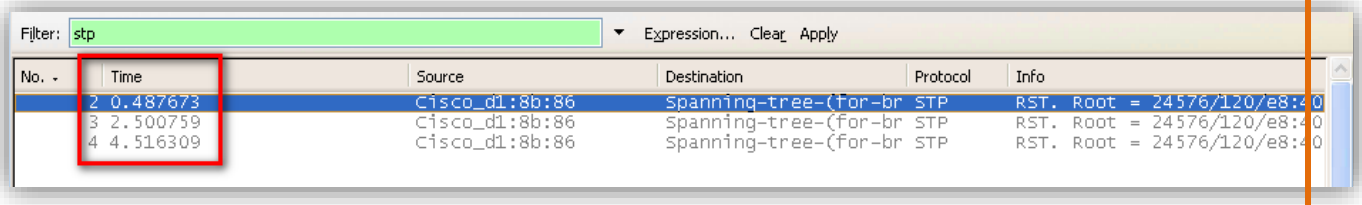
If BPDU Filter is configured on the global level using the **spanning-tree portfast bpdufilter default** global configuration command, the BPDU Filter applies only to PortFast-enabled ports. When these ports come up, they will *send* up to 11 BPDUs and then stop sending further BPDUs. If the BPDU Filter-configured interface *receives* a BPDU *at any time*, the BPDU Filter and PortFast will be deactivated on that port and it will become a normal spanning tree interface. As a result, a globally configured BPDU Filter does not prevent ports from receiving and processing BPDUs; it only attempts to stop sending BPDUs on ports where most probably, there is no device attached that would be interested in processing them.

Configured at an interface, BPDU Filter causes the port to stop sending and processing received BPDUs altogether. This can be used, for example, to split a network into two or more independent STP domains, each having its own root bridge and resulting topology. However, because these domains are no longer protected against mutual loops by STP, it is the task of the network administrator to make sure that these two domains are never connected by more than just a single link.

To test this feature, we will run a packet capture utility on Host A connected to ALS1, configure BPDU Filter on interface Fa0/6, and see that BPDUs stop being transmitted.

Login to Host A, run your packet capture utility and filter the output to show only STP packets. In Wireshark, the filter bar should look something like this:

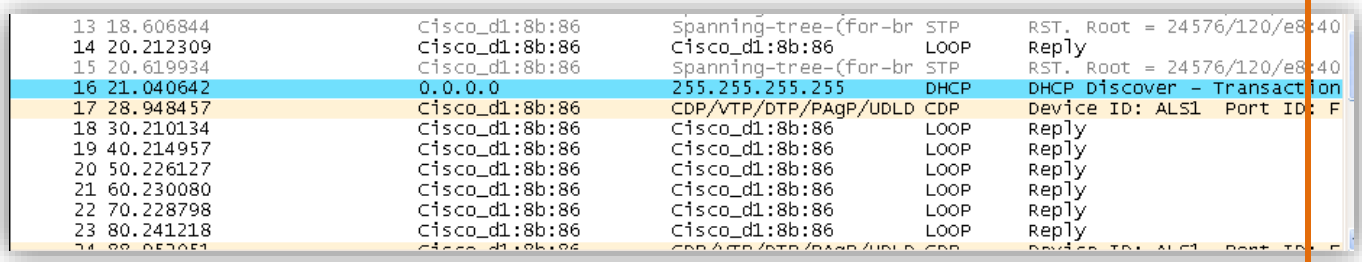
You should see a BPDU being received at your host every 2 seconds.



Now configure BPDU Filter on ALS1 interface Fa0/6

```
ALS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ALS1(config)# int f0/6
ALS1(config-if)# spanning-tree bpdufilter enable
ALS1(config-if)# exit
ALS1(config)#
```

Clear the Wireshark display filter and observe your packet capture window; you will have stopped receiving BPDUs.



Step 4: Implement and Observe Root Guard

Root Guard helps prevent a root switch or Root Port takeover. It is configured on the port to be protected. If a port protected by Root Guard receives a superior BPDU that would normally cause the port to become a Root port, the BPDU will be discarded and the port will be moved to the Root-Inconsistent state. An STP inconsistent state differs from the error disabled state that the port is not disabled entirely; instead, it is only put into the Blocking (Discarding) state and will be put back into its proper role and state once the cause for its inconsistent state disappears. With Root Guard, a port will be reinstated into its appropriate role and state automatically when it stops receiving superior BPDUs.

Note: BPDU Root Guard is a protective mechanism in situations when your network and the network of your customer need to form a single STP domain, yet you want to have the STP root bridge in your network part and you do not want your customer to take over this root switch selection, or back up the connectivity in your network through the customer. In these cases, you would put the Root Guard on ports toward the customer. However, inside your own network, using Root Guard would actually be harmful. Your network can be considered trustworthy and there is no rogue root switch to protect against. Using Root Guard in your own network would cause it to be unable to converge on a new workable spanning tree if any of the primary links failed, and it would also prevent your network from converging to a secondary root switch if the primary root switch failed entirely.

To illustrate the behavior of Root Guard, we will configure it on a designated port on DLS1 for VLAN 100. DLS1 is the root bridge for VLAN 100, so all trunk ports are designated.

```
DLS1# show span root
                                        Root    Hello Max Fwd
Vlan                 Root ID            Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0099         24675 e840.406f.7280         0    2   20  15
VLAN0100         24676 e840.406f.7280         0    2   20  15
VLAN0110         24686 e840.406f.6e00        19    2   20  15  Fa0/11
VLAN0120         24696 e840.406f.6e00        19    2   20  15  Fa0/11
VLAN0666         33434 5017.ff84.0a80        38    2   20  15  Fa0/7
```

From the ALS1 side of things, the root port is interface F0/8. Normally it would be F0/7, but we changed the port priority of F0/8 to 112, and this impacts root port selection at ALS1 when all interfaces are operational:

```
ALS1# show span root | inc VLAN0100
VLAN0100         24676 e840.406f.7280        19    2   20  15  Fa0/8
```

Configure Root Guard on DLS1 interface Fa0/8 (you may immediately see errors with another VLAN, like 666. Ignore these as we are focusing on VLAN 100):

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# int f0/8
DLS1(config-if)# spanning-tree guard root
DLS1(config-if)# exit
```

```
DLS1(config)#
```

Then go to ALS1 and configure it to be the root for VLAN 100 using the priority 16384

```
ALS1(config)# spanning-tree vlan 100 priority 16384
```

Then back at DLS1, check the spanning tree interface status for Fa0/8:

```
DLS1# show spanning-tree interface f0/8 | inc VLAN0100
VLAN0100            Desg BKN*19        112.10   P2p *ROOT_Inc
DLS1#
```

This output has two indicators of the issue. First BKN* is short for "BROKEN", and *ROOT_Inc represents the Root Inconsistent message. A list of all STP inconsistent ports including the reason for their inconsistency can also be requested with the command **show spanning-tree inconsistentports**.

```
DLS1# show spanning-tree inconsistentports

Name                 Interface                Inconsistency
-------------------- ------------------------ ------------------
VLAN0100             FastEthernet0/8          Root Inconsistent
VLAN0666             FastEthernet0/8          Root Inconsistent

Number of inconsistent ports (segments) in the system : 2

DLS1#
```

To clear this, go back to ALS1 and issue the command **no spanning-tree vlan 100 priority 16384**. Once you do this, you will see the following SYSLOG message at DLS1, and the interface will become consistent again.

```
DLS1#
*Mar  1 02:54:06.761: %SPANTREE-2-ROOTGUARD_UNBLOCK: Root guard unblocking
port FastEthernet0/8 on VLAN0100.
DLS1# show spanning-tree interface f0/7 | inc VLAN0100
VLAN0100            Desg FWD 19        128.9    P2p
DLS1#
```

For completeness, remove Root Guard from Fa0/7 on DLS1

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# int f0/8
```

```
DLS1(config-if)# no spanning-tree guard root
DLS1(config-if)# exit
DLS1(config)#
```

Step 5: Implement and Observe Loop Guard

The last tool we will demonstrate is Loop Guard. Its job is to prevent Root and Alternate ports from becoming Designated ports if BPDUs suddenly cease to be received on them

In a normal STP network, all ports receive and process BPDUs, even Blocking (Discarding) ports. This is how they know that the device at the other end of the link is alive and still superior to them. If a Blocked port stops receiving these BPDUs, it can only assume that the device on the other side is no longer there and they are now superior, and should be in Forwarding state for the given segment. An example of when this could occur is the instance where the Rx fiber in an optical cable becomes disconnected, cut, or connected to a different port or device than the corresponding Tx fiber, in essence creating an uni-directional link.

This could cause permanent switching loops in the network, so Loop Guard helps to prevent them.

For this example, we will configure Loop Guard on an Alternate port on ALS2, and then stop sending out BPDUs from the corresponding Designated port on the other end of the link, and observe the behavior.

```
ALS2# show span vlan 100

VLAN0100
  Spanning tree enabled protocol rstp
  Root ID     Priority    24676
              Address     e840.406f.7280
              Cost        31
              Port        7 (FastEthernet0/7)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32868  (priority 32768 sys-id-ext 100)
              Address     5017.ff84.0a80
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
-----
Fa0/7               Root FWD 12        128.7    P2p
Fa0/8               Altn BLK 19        128.8    P2p
Fa0/11              Altn BLK 19        128.11   P2p
Fa0/12              Altn BLK 19        128.12   P2p
```

Here ALS2 tells us that its path to the root is via Fa0/7 (connected to DLS2) with a total cost of 31, while Fa0/11 and 12 (connected to ALS1) are Alternate ports. Fa0/11 and Fa0/12 are alternate ports because the interface cost plus the cost advertised by ALS1 equals 39, which is

greater than the local interface cost plus the cost advertised by DLS2. Fa0/7 has a locally configured cost of 12. That plus the 19 advertised by DLS2 equals 31. You can see these details in the output of **show spanning-tree detail**

Configure Loop Guard on ALS2s Fa0/11 interface:

```
ALS2(config)# int f0/11
ALS2(config-if)# spanning-tree guard loop
ALS2(config-if)# exit
```

Modify the corresponding interface (Fa0/11) on ALS1 to stop sending BPDUs:

```
ALS1(config)# int f0/11
ALS1(config-if)# spanning-tree bpdufilter enable
ALS1(config-if)# exit
```

Shortly after doing this, you should receive the following SYSLOG message on ALS2 for every VLAN that had Fa0/11 as an Alternate port:

```
ALS2#
*Mar  1 03:22:36.795: %SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port
FastEthernet0/11 on VLAN0100.
*Mar  1 03:22:37.802: %SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port
FastEthernet0/11 on VLAN0099.
ALS2# show spanning-tree inconsistentports

Name                 Interface               Inconsistency
-------------------- ----------------------- ------------------
VLAN0099             FastEthernet0/11        Loop Inconsistent
VLAN0100             FastEthernet0/11        Loop Inconsistent

Number of inconsistent ports (segments) in the system : 2

ALS2#
```

Fix this by reversing the configuration at ALS1 and verifying at ALS2:

```
ALS1(config)# int f0/11
ALS1(config-if)# spanning-tree bpdufilter disable
ALS1(config-if)# exit

*Mar  1 03:23:37.227: %SPANTREE-2-LOOPGUARD_UNBLOCK: Loop guard unblocking
port FastEthernet0/11 on VLAN0099.
ALS2# show spanning-tree inconsistentports

Name                 Interface               Inconsistency
-------------------- ----------------------- ------------------
```

```
Number of inconsistent ports (segments) in the system : 0
```

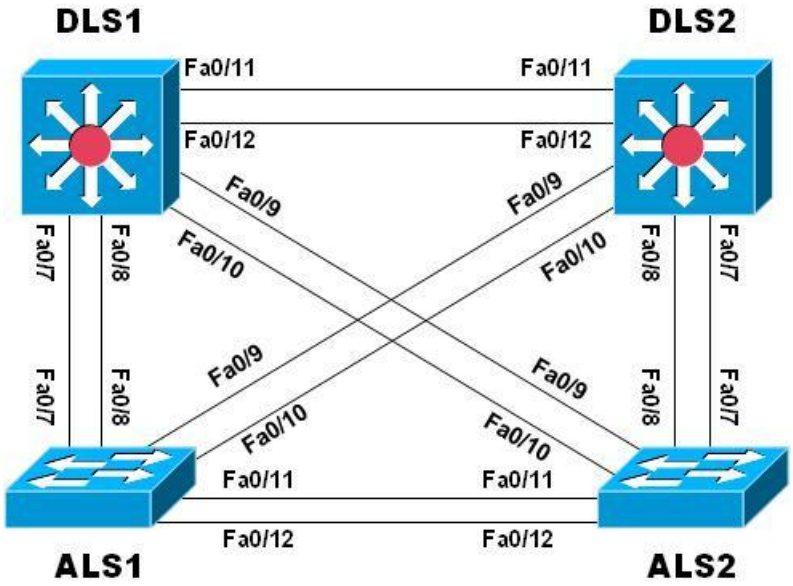
Step 6: End of Lab

Do not save your configurations. The equipment will be reset for the next lab.

CCNPv7.1 SWITCH

Chapter 4 Lab 4-2- Multiple Spanning Tree

Topology



Objectives

- Implement Multiple Spanning Tree
- Leverage VTP version 3 with MST

Background

Cisco's Per VLAN Spanning Tree (PVST) provides a significant step up from standard spanning tree in terms of flexibility, allowing each VLAN to have its own independent spanning tree, thereby make better use of available links in the network. A drawback to PVST is that there is an instance of PVST running for each VLAN in the network, regardless of whether there are actually different spanning-tree topologies required. This presents the potential for overwhelming the switch CPU and memory. Additionally, Cisco switches like those used in these labs allow only a limited number of PVST instances – usually 128. If more than 128 VLANs are created, some of them will not have any STP running, and therefore not have any switching loop protection. PVST and Rapid PVST are simply unusable in that kind of environment. Lastly, PVST and Rapid PVST are Cisco-proprietary protocols and generally unusable in mixed vendor environments.

Cisco was involved in the early development of Multiple Spanning Tree. MST was standardized as IEEE 802.1s in 2002 and merged into 802.1Q in 2005. MST is an open protocol derived from

RSTP, sharing all its rapid convergence properties, and in fact, the only standardized spanning-tree protocol for VLAN-based networks supported by multiple vendors. MST is a compromise between common spanning-tree and per-VLAN spanning tree. An MST instance represents a unique spanning-tree topology. Multiple MST instances can be created to account for each of the required spanning-tree topologies in a network, and an arbitrary number of VLANs can be mapped to a single MST instance.

In this lab you will set up two instances of MST, one for VLANs 99 and 100 and the other for VLANs 110 and 120. All other VLANs will be mapped to the default MST instance (also referred to as IST or Internal Spanning Tree).

Note: This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates "dual-ipv4-and-ipv6 routing" and "lanbase-routing", respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

- 2 Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- 2 Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Computer with terminal emulation software
- Ethernet and console cables

Step 7: Prepare the switches for the lab

Use the reset.tcl script you created in Lab 1 "Preparing the Switch" to set your switches up for this lab. Then load the file BASE.CFG into the running-config with the command copy flash:BASE.CFG running-config. An example from DLS1:

```
DLS1# tclsh reset.tcl
Erasing the nvram filesystem will remove all configuration files! Continue?
[confirm]
[OK]
Erase of nvram: complete
Reloading the switch in 1 minute, type reload cancel to halt

Proceed with reload? [confirm]

*Mar  7 18:41:40.403: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of
nvram
*Mar  7 18:41:41.141: %SYS-5-RELOAD: Reload requested by console. Reload
Reason: Reload command.
<switch reloads - output omitted>
```

```
Would you like to enter the initial configuration dialog? [yes/no]: n
Switch> en
*Mar  1 00:01:30.915: %LINK-5-CHANGED: Interface Vlan1, changed state to
administratively down
Switch# copy BASE.CFG running-config
Destination filename [running-config]?
184 bytes copied in 0.310 secs (594 bytes/sec)
DLS1#
```

Step 8: Configure Trunking

Next configure interfaces F0/7 through F0/12 as 802.1Q trunk ports on all four switches. Additionally, configure all four switches VTP Servers. An example from DLS1:

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# vtp mode server
Setting device to VTP Server mode for VLANS.
DLS1(config)# int ran f0/7-12
DLS1(config-if-range)# switchport trunk encap dot1q
DLS1(config-if-range)# switchport trunk native vlan 666
DLS1(config-if-range)# switchport trunk allowed vlan except 1,999
DLS1(config-if-range)# switchport mode trunk
DLS1(config-if-range)# switchport nonegotiate
DLS1(config-if-range)# no shut
DLS1(config-if-range)# exit
DLS1(config)#
```

Step 9: Configure VTP and VLANs

To simplify the lab configuration, configure VTP version 2 on DLS1 with no password, and configure VLANs for use in the network. This configuration will propagate to the other switches in the network.

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
DLS1(config)# vtp version 2
DLS1(config)# vlan 99
DLS1(config-vlan)# name MANAGEMENT
DLS1(config-vlan)# vlan 100
DLS1(config-vlan)# name SERVERS
DLS1(config-vlan)# vlan 110
DLS1(config-vlan)# name GUEST
DLS1(config-vlan)# vlan 120
DLS1(config-vlan)# name OFFICE
DLS1(config-vlan)# vlan 999
DLS1(config-vlan)# name PARKING_LOT
DLS1(config-vlan)# state suspend
DLS1(config-vlan)# vlan 666
```

```
DLS1(config-vlan)# name NATIVE_DO_NOT_USE
DLS1(config-vlan)# exit
*Mar  1 00:18:41.431: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name
changed to SWLAB.
DLS1(config)#
```

Verify that all of the VLANs propagate and that there is a single root bridge for all of the VLANs.

Step 10: Implement Multiple Spanning Tree

In this step you will implement MST on DLS1 and DLS2; we will ignore ALS1 and ALS2 for now.

Issue the global configuration command **`spanning-tree mode mst`** and then the privileged exec command **`clear spanning-tree detected-protocols`**.

An example from DLS1:

```
DLS1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# spanning-tree mode mst
DLS1(config)#exit
DLS1# clear spanning-tree detected-protocols
DLS1#
DLS1# show spanning-tree

MST0
  Spanning tree enabled protocol mstp
  Root ID    Priority    32768
```

Step 11: Observe default MST configuration

At this point, MST is running with default parameters. On DLS1, issue the command **`show spanning-tree mst configuration`** to see the configuration information:

```
DLS1# show span mst configuration
Name      []
Revision  0     Instances configured 1

Instance  Vlans mapped
--------  ---------------------------------------------------------------
----
0         1-4094
---------------------------------------------------------------------------
----
DLS1#
```

The output diplays:

- The region is un-named
- The revision number is 0

- There is one instance of MST, number 1, and VLANS 1-4094 are mapped to that instance

For MST to work, the region must be named and given a revision number (this revision number does not work like VTP, it is just an administrator-assigned value). All the switches in the same region must have the same region name and revision number, and have the same VLAN-to-instance mapping.

Step 12: Manually Configure MST

Now configure MST **on both DLS1 and DLS2** with the following information (you must configure each switch manually):

- Region Name: CCNP
- Revision Number: 1
- VLAN Mappings: Instance 1: VLAN 99 and VLAN 100

MST region configuration is performed in a special mode under the global configuration that is entered using the **`spanning-tree mst configuration`** command. You have to make the changes and exit from configuration mode to have the changes applied; the changes are not applied until you exit. While in MST configuration mode, you can use the **`show current`** and **`show pending`** commands to see how the configuration stands. From DLS1:

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# spanning-tree mst configuration
DLS1(config-mst)# name CCNP
DLS1(config-mst)# revision 1
DLS1(config-mst)# instance 1 vlan 99,100
DLS1(config-mst)#
DLS1(config-mst)# show current
Current MST configuration
Name      []
Revision  0     Instances configured 1

Instance  Vlans mapped
--------  ---------------------------------------------------------------------
----
0         1-4094
---------------------------------------------------------------------------------
----
DLS1(config-mst)#
DLS1(config-mst)# show pending
Pending MST configuration
Name      [CCNP]
Revision  1     Instances configured 2
```

```
Instance  Vlans mapped
--------  ------------------------------------------------------------
----
0         1-98,101-4094
1         99-100
------------------------------------------------------------------------
----
DLS1(config-mst)#
DLS1(config-mst)#exit
DLS1(config)#end
DLS1#
DLS1# show span mst config
Name      [CCNP]
Revision  1     Instances configured 2

Instance  Vlans mapped
--------  ------------------------------------------------------------
----
0         1-98,101-4094
1         99-100
------------------------------------------------------------------------
----
DLS1#
```

Wait a moment to let the topology settle and then issue the **show spanning-tree mst** command on DLS1:

```
DLS1# show spanning-tree mst


##### MST0    vlans mapped:   1-98,101-4094
Bridge        address e840.406f.7280  priority      32768 (32768 sysid 0)
Root          address e840.406f.6e00  priority      32768 (32768 sysid 0)
              port    Fa0/11          path cost     0
Regional Root address e840.406f.6e00  priority      32768 (32768 sysid 0)
                                      internal cost 200000    rem hops 19
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Desg BLK 200000    128.9    P2p Bound(PVST)
Fa0/8            Desg BLK 200000    128.10   P2p Bound(PVST)
Fa0/9            Desg BLK 200000    128.11   P2p Bound(PVST)
Fa0/10           Desg BLK 200000    128.12   P2p Bound(PVST)
Fa0/11           Root FWD 200000    128.13   P2p
Fa0/12           Altn BLK 200000    128.14   P2p
```

```
##### MST1    vlans mapped:   99-100
Bridge        address e840.406f.7280  priority      32769 (32768 sysid 1)
Root          address e840.406f.6e00  priority      32769 (32768 sysid 1)
              port    Fa0/11          cost          200000    rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Desg BLK 200000    128.9    P2p Bound(PVST)
Fa0/8            Desg BLK 200000    128.10   P2p Bound(PVST)
Fa0/9            Desg BLK 200000    128.11   P2p Bound(PVST)
Fa0/10           Desg BLK 200000    128.12   P2p Bound(PVST)
Fa0/11           Root FWD 200000    128.13   P2p
Fa0/12           Altn BLK 200000    128.14   P2p

DLS1
```

As you can see from the output above, the VLANs are mapped to the correct instance and the root bridge for instance 1 is not the local switch (it is DLS2 in this case).

Notice the type entry **`P2p Bound(PVST)`**. This is the entry shown when the device connected at the other end of the given interface is not running MST; in this case, ALS1 and ALS2 are running the default PVST.

Step 13: Propagate MST configurations with VTP

Manual configuration of MST is not particularly difficult until the network scales to a large size. For switches to form a single MST region, they must match in all region parameters: region name, configuration revision, VLAN-to-instance mappings. Switches that differ in their MST region configuration will form separate regions, each region having its own internal root bridges for the defined MST instances and independent internal topologies. While having multiple regions is not an error per se, and some large networks are even partitioned into multiple regions intentionally, running multiple MST regions as a result of region misconfiguration is undesirable.

VTP version 3 allows for the sharing of the MST database amongst switches, which simplifies this process considerably.

To use VTP version 3 to propagate the MST region configuration to all switches in the VTP domain, convert all switches to VTP version 3 and set them as servers or clients for MST. Then designate one switch as the VTP primary for MST. Do not forget to activate MST on all switches; VTP version 3 will synchronize only the region configuration across all switches and will not affect the STP version running on the switch.

From DLS2:

```
DLS2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS2(config)# vtp version 3
DLS2(config)#
*Mar  1 00:49:27.386: %SW_VLAN-6-OLD_CONFIG_FILE_READ: Old version 2 VLAN
configuration file detected and read OK.  Version 3
    files will be written in the future.
DLS2(config)#
```

```
DLS2(config)# vtp mode server mst
Setting device to VTP Server mode for MST.
DLS2(config)# end
DLS2# vtp primary mst
This system is becoming primary server for feature  mst
No conflicting VTP3 devices found.
Do you want to continue? [confirm]
DLS2#
*Mar  1 00:55:45.217: %SW_VLAN-4-VTP_PRIMARY_SERVER_CHG: e840.406f.7380 has
become the primary server for the MST VTP feature
```

From ALS1 (the same configuration must be applied at ALS2):

```
ALS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ALS1(config)# spanning-tree mode mst
ALS1(config)# vtp version 3
ALS1(config)# vtp mode server mst
Setting device to VTP Server mode for MST.
ALS1(config)# end
```

Note: An identical MST region configuration will be propagated to all switches within a VTPv3 domain, and consequently they will all form a single region. As a result, there is always a one-to-one mapping between a VTPv3 domain and an MST region.

Step 14: Verify Initial MST Configuration

After the entire configuration is done, VTP version 3 will propagate the MST configuration to the other switches. Verify this by checking ALS2:

```
ALS2# show spanning-tree mst configuration
Name      [CCNP]
Revision  1     Instances configured 2

Instance  Vlans mapped
--------  ---------------------------------------------------------------
----
0         1-98,101-4094
1         99-100
-------------------------------------------------------------------------
----
ALS2# show span mst

##### MST0    vlans mapped:   1-98,101-4094
Bridge        address 5017.ff84.0a80  priority      32768 (32768 sysid 0)
Root          this switch for the CIST
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
```

```
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Desg FWD 200000    128.7    P2p
Fa0/8            Desg FWD 200000    128.8    P2p
Fa0/9            Desg FWD 200000    128.9    P2p
Fa0/10           Desg FWD 200000    128.10   P2p
Fa0/11           Desg FWD 200000    128.11   P2p
Fa0/12           Desg FWD 200000    128.12   P2p

##### MST1    vlans mapped:   99-100
Bridge       address 5017.ff84.0a80  priority      32769 (32768 sysid 1)
Root         this switch for MST1

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Desg FWD 200000    128.7    P2p
Fa0/8            Desg FWD 200000    128.8    P2p
Fa0/9            Desg FWD 200000    128.9    P2p
Fa0/10           Desg FWD 200000    128.10   P2p
Fa0/11           Desg FWD 200000    128.11   P2p
Fa0/12           Desg FWD 200000    128.12   P2p
```

Step 15: Modify MST Configuration

To further illustrate the convenience of MST and VTP version 3, add another instance on DLS2, mapping VLANs 110 and 120 to it.

```
DLS2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS2(config)# spanning-tree mst config
DLS2(config-mst)# instance 2 vlan 110,120
DLS2(config-mst)# show pending
Pending MST configuration
Name      [CCNP]
Revision  1     Instances configured 3

Instance  Vlans mapped
--------  ------------------------------------------------------------------
----
0         1-98,101-109,111-119,121-4094
1         99-100
2         110,120
```

```
--------------------------------------------------------------------------------
----
DLS2(config-mst)#
DLS2(config-mst)# exit
DLS2(config)# end
DLS2#
```

And then verify on that the changes propagated to another switch:

```
DLS1# show span mst config
Name      [CCNP]
Revision  1     Instances configured 3

Instance  Vlans mapped
--------  ---------------------------------------------------------------------
----
0         1-98,101-109,111-119,121-4094
1         99-100
2         110,120
--------------------------------------------------------------------------------
----
DLS1# show span mst

##### MST0    vlans mapped:   1-98,101-109,111-119,121-4094
Bridge        address e840.406f.7280  priority      32768 (32768 sysid 0)
Root          address 5017.ff84.0a80  priority      32768 (32768 sysid 0)
              port    Fa0/9           path cost     0
Regional Root address 5017.ff84.0a80  priority      32768 (32768 sysid 0)
                                      internal cost 200000    rem hops 19
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Altn BLK 200000    128.9    P2p
Fa0/8            Altn BLK 200000    128.10   P2p
Fa0/9            Root FWD 200000    128.11   P2p
Fa0/10           Altn BLK 200000    128.12   P2p
Fa0/11           Altn BLK 200000    128.13   P2p
Fa0/12           Altn BLK 200000    128.14   P2p

##### MST1    vlans mapped:   99-100
Bridge        address e840.406f.7280  priority      32769 (32768 sysid 1)
Root          address 5017.ff84.0a80  priority      32769 (32768 sysid 1)
              port    Fa0/9           cost          200000    rem hops 19
```

```
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Altn BLK 200000    128.9    P2p
Fa0/8            Altn BLK 200000    128.10   P2p
Fa0/9            Root FWD 200000    128.11   P2p
Fa0/10           Altn BLK 200000    128.12   P2p
Fa0/11           Altn BLK 200000    128.13   P2p
Fa0/12           Altn BLK 200000    128.14   P2p

##### MST2    vlans mapped:   110,120
Bridge        address e840.406f.7280  priority      32770 (32768 sysid 2)
Root          address 5017.ff84.0a80  priority      32770 (32768 sysid 2)
              port    Fa0/9           cost          200000    rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Fa0/7            Altn BLK 200000    128.9    P2p
Fa0/8            Altn BLK 200000    128.10   P2p
Fa0/9            Root FWD 200000    128.11   P2p
Fa0/10           Altn BLK 200000    128.12   P2p
Fa0/11           Altn BLK 200000    128.13   P2p
Fa0/12           Altn BLK 200000    128.14   P2p
```

Step 16: Manipulate the spanning tree

To this point, we have left election of the root bridge up to the protocol defaults (which are the same as PVST with the exception of port cost values), still based on the physical interface's bandwidth which use much larger numbers.

An example of the show spanning-tree root command at DLS1 provides proof that the root bridge is elsewhere:

```
DLS1# show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance            Root ID         Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
MST0             32768 5017.ff84.0a80         0    2   20  15  Fa0/9
MST1             32769 5017.ff84.0a80    200000    2   20  15  Fa0/9
MST2             32770 5017.ff84.0a80    200000    2   20  15  Fa0/9
DLS1#
```

Port costs, which are summed to find a path cost in the quest for a root bridge, are different in MST:

- 10 Mbps—2,000,000
- 100 Mbps—200,000

- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000

MST uses the same basic commands and values to manipulate the operation.

To manually configure a bridge to be the primary MST root, use the command **spanning-tree mst** *instance-list* **root {***primary* | *secondary***}** global configuration command. You can also manually set the bridge priority using the **spanning-tree mst** *instance-list* **priority** *priority* global configuration command. In the example below, DLS1 is configured as the primary root for instance 0 and 1, and the secondary root for instance 2:

```
DLS1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS1(config)# spanning-tree mst 0-1 root primary
DLS1(config)# spanning-tree mst 2 root secondary
DLS1(config)# end
DLS1#
```

DSL2 is configured with a complementary set of instructions; root primary for instance 2 and root secondary for instances 0 and 1:

```
DLS2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS2(config)# spanning-tree mst 0-1 root secondary
DLS2(config)# spanning-tree mst 2 root primary
DLS2(config)# end
DLS2#
```

The results of these configuration changes are evident using the **show spanning-tree root** command. From ALS1, which shows Fa0/7 (connected to DLS1) as the Root Port for instances 0 and 1 and Fa0/9 (connected to DLS2) for instance 2:

```
ALS1# show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance            Root ID         Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
MST0             24576 e840.406f.7280         0    2   20  15  Fa0/7
MST1             24577 e840.406f.7280    200000    2   20  15  Fa0/7
MST2             24578 e840.406f.6e00    200000    2   20  15  Fa0/9
ALS1#
```

As with PVST, Root Port selection is based on total path cost to the root bridge. Path cost is the sum of Port Costs. You can configure the port costs using the **`spanning-tree mst instance cost`** *`value`* interface configuration command, which sets the cost for that instance alone.

As an implementation example, we will shutdown interfaces Fa0/9-10 on DLS2 and then change the port cost value of ALS2's interface Fa0/7 to a lower number, causing the spanning tree for instance 2 to go through ALS2.

On ALS2:

```
ALS2# config t
ALS2(config)# int f0/7
ALS2(config-if)# spanning-tree mst 2 cost 1000
ALS2(config-if)# exit
ALS2(config)# end
```

On DLS2:

```
DLS2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
DLS2(config)# interface ran f0/9-10
DLS2(config-if-range)# shut
DLS2(config-if-range)# end
```

And then finally examining ALS1:

```
ALS1# show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance            Root ID         Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
MST0             24576 e840.406f.7280         0    2   20  15  Fa0/7
MST1             24577 e840.406f.7280    200000    2   20  15  Fa0/7
MST2             24578 e840.406f.6e00    201000    2   20  15  Fa0/11
ALS1#
```

Step 17: End of Lab

Do not save your configurations. The equipment will be reset for the next lab.

CONCLUSIONES

- Con el desarrollo de estos laboratorios planteados se logró entender el concepto de VLAN.
- Los Laboratorios han permitido entender el concepto de EtherChannel.
- Realizar estos laboratorios simulados nos ha permitido entender cómo funciona el enrutamiento entre VLANs y adicionalmente nos ha dado claridad con respecto a los protocolos VTP, STP.
- En general estos laboratorios nos ha brindado un acercamiento para la resolución de posibles problemas en las redes convergentes.