Lab 2 – Implement Rapid Spanning Tree Protocol (IEEE 802.1w) and STP tool kit

Name: Rayyan Mojaddedi

Name: Xindong Lin

Name: Darien Ramirez

Name: Shawn Rae



**Topology:**



**DLS1**

**ALS1**

**Vlan 99:10.1.99.103/24**

**Gig1/0/3**

**Gig1/0/1**



**ALS2**

**Vlan 99:10.1.99.104/24**

**Gig1/0/7**

**ALL Switch-to-Switch connections are 802.1Q trunks**

**Gig0/0/1**

**10.1.100.1/24**

**Gig1/0/11**

**10.1.100.2/24**

**DLS2**

**Vlan99: 10.1.99.102/24**

**Gig1/0/1**

**Gig1/0/1**

**Gig1/0/2**

**Gig1/0/2**

**Gig1/0/3**



**Gig1/0/11**

**Unused**

**switch**



**Loopback0**

**10.10.10.10/32**

**R1**

**DLS1**

**VLAN99: 10.1.99.101/24**

1. Objectives

* Implement Rapid Spanning Tree protocol (802.1w)
* Implement STP tool kit components
* Implement EIGRP IPv4 routing

1. Background

**Note:** This lab uses Cisco Catalyst 3850 and 2960 switches

Required Resources

* 2 Cisco 2960 switches
* 2 Cisco 3850
* 1 ISR router
* Ethernet and console cables

1. Observe default Spanning Tree behavior
   1. Configure 802.1Q trunks on all 4 Switches

Enable Trunking on DLS1 ports. Use native VLAN 666, trunk mode static.

Perform this step on all four (4) switches, use the topology to enable trunking only on required links.

**SHUTDOWN the UNUSED LINKS/TRUNKS based on the above Topology**

An example from DLS1:

DLS1(config)# **int range gigabitEthernet1/0/1-3**

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)#

**“NOTE:** 1- SHUTDOWN USUSED TRUNKS/LINKS on ALL Switches”

2- ALS switches do NOT support command “**switchport trunk encap dot1q” so do not attempt to**

**enter command on ALS switches.**

Finally, configure all four switches as VTP version 2, mode: Server, in domain name: “**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 2**

DLS1(config)#

* 1. Configure VLANs and Enable Management VLAN 99 (Switched Virtual Interface) IP addressing

1. **For VLAN 99, enable SVI interface with IP addresses on all Switches as shown in Topology diagram.**

**IP addresses will be used to verify connectivity between switches over the Management VLAN 99 SVI interface, note that the SVI interface will be in Down state until VLAN 99 is enabled in next step.**

Configure VLANs on ALL switches DLS1 example is shown below:

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)#

**Verify VTP status and VLANs are configured on ALL switches, ALS2 example is shown below:**

**ALS2#** **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 : 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

<output omitted>

**ALS2# show vlan brief | i active**

1 default active

99 MANAGEMENT active

100 SERVERS active

110 GUEST active

120 OFFICE active

666 NATIVE\_DO\_NOT\_USE active

ALS2#

* 1. Identify and modify the root bridge

**Enable Rapid Spanning Tree (RSTP) IEEE 802.1w on ALL Switches, also verify that SVI 99 is in UP/UP state**

**DLS1(config)# spanning-tree mode rapid-pvst**

**Configure “spanning-tree mode rapid-pvst” on ALL other switches**

Modify DLS1 and DLS2 so that DLS 1 is elected the primary root bridge for **VLANs 99, 100, 666** 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 99,100,666

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

**On DLS1:**

DLS1# **conf t**

Enter configuration commands, one per line. End with CNTL/Z.

DLS1(config)# **spanning-tree vlan 99,100,666 root primary**

DLS1(config)# **spanning-tree vlan 110,120 root secondary**

DLS1(config)# **exit**

DLS1#

**On DLS2**

DLS2# **conf t**

Enter configuration commands, one per line. End with CNTL/Z.

DLS2(config)# **spanning-tree vlan 99,100,666 root secondary**

DLS2(config)# **spanning-tree vlan 110,120 root primary**

DLS2(config)# **exit**

DLS2#

**Verification from DLS1 (Ignore VLAN1 as it is not shown below):**

**DLS1#sh spanning-tree root**

Root Hello Max Fwd

Vlan Root ID Cost Time Age Dly Root Port

---------------- -------------------- --------- ----- --- --- ------------

VLAN0099 24675 6c41.6ace.3380 0 2 20 15

VLAN0100 24676 6c41.6ace.3380 0 2 20 15

VLAN0110 24686 6c41.6ace.1400 4 2 20 15 Gi1/0/3

VLAN0120 24696 6c41.6ace.1400 4 2 20 15 Gi1/0/3

VLAN0666 25242 6c41.6ace.3380 0 2 20 15

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#sh spanning-tree bridge**

Hello Max Fwd

Vlan Bridge ID Time Age Dly Protocol

---------------- --------------------------------- ----- --- --- --------

VLAN0001 32769 (32768, 1) 6c41.6ace.3380 2 20 15 rstp

VLAN0099 24675 (24576, 99) 6c41.6ace.3380 2 20 15 rstp

VLAN0100 24676 (24576, 100) 6c41.6ace.3380 2 20 15 rstp

VLAN0110 28782 (28672, 110) 6c41.6ace.3380 2 20 15 rstp

VLAN0120 28792 (28672, 120) 6c41.6ace.3380 2 20 15 rstp

VLAN0666 25242 (24576, 666) 6c41.6ace.3380 2 20 15 rstp

**Ensure that SVI 99 is in UP/UP state and you can ping SVI IP addresses across the switches, also ensure that VLAN 99 is created properly on ALL switches, otherwise SVI will not be in UP/UP state.**

1. Implement STP tool kit components
   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 Multiple Spanning Tree Protocol (MSTP (802.1s)). 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.

**If you are using NETLAB, ensure that you only have the ports unshut based on the Topology diagram, i.e. shutdown the unused ports on ALS and DLS switches to avoid confusing results**

On ALS1, issue the command **debug spanning-tree events** , then configure Gig1/0/11 to be in VLAN 99.

**state Connect Gig1/0/11 to any other unused switch and verify it is in UP/UP**

**shutdown** the interface, then **unshut** 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 Gig1/0/11**

ALS1(config-if)# **swi mode access**

ALS1(config-if)# **swi access vlan 99**

ALS1(config-if)# **shut**

ALS1(config-if)# **no shut**

ALS1(config-if)#

\*Mar 1 00:36:02.448: RSTP(120): initializing port GigabitEthernet1/0/11

\*Mar 1 00:36:02.448: RSTP(120): GigabitEthernet1/0/11 is now designated

\*Mar 1 00:36:02.457: RSTP(120): transmitting a proposal on GigabitEthernet1/0/11

\*Mar 1 00:36:02.465: STP[120]: Generating TC trap for port FastEthernet0/11

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 shutdown the interface and now we will add the **spanning-tree portfast** command to the interface (the debug is still running):

ALS1(config-if)# **shut----------------------🡪 shutdown GIG1/0/11**

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 FastEthernet0/11 but will only

have effect when the interface is in a non-trunking mode.

* **Now unshut the interface:**

ALS1(config-if)#

ALS1(config-if)# **no shut**

ALS1(config-if)#

\*Mar 1 00:38:14.032: RSTP(120): initializing port GigabitEthernet1/0/11

**\*Mar 1 00:38:14.032: RSTP(120): GigabitEthernet1/0/11 is now designated**

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

**Notice in output above that with PortFast configured, no proposals are sent out of interface GigabitEthernet1/0/11; the port goes into Forwarding state immediately.**

**ALS1#show span detail | beg VLAN0099----🡪 Look for GIG1/0/11 output**

**!**

**output trimmed**

**!**

VLAN0099 is executing the rstp compatible Spanning Tree protocol

**Port 11 (GigabitEthernet1/0/11) of VLAN0099 is designated forwarding**

Port path cost 4, Port priority 128, Port Identifier 128.11.

Designated root has priority 24675, address 6c41.6ace.3380

Designated bridge has priority 32867, address 08d0.9f48.d900

Designated port id is 128.11, designated path cost 4

Timers: message age 0, forward delay 0, hold 0

Number of transitions to forwarding state: 5

Link type is point-to-point by default

BPDU: sent 188, received 18

**Note(1)**: PortFast should never be enabled on ports connected to another switches. Doing so could cause a switching loop.

**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 Layer 2 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(edge) ports using the **spanning-tree portfast default** global configuration command.

With this command configured, each port that operates in the access (edge) 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.

* 1. 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 error-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 Gig1/0/1**

**ALS2(config-if)#spanning-tree bpduguard enable**

**ALS2(config-if)#**

\*Mar 1 00:51:17.067: %SPANTREE-2-BLOCK\_BPDUGUARD: Received BPDU on port GigabitEthernet1/0/1 with BPDU Guard enabled. Disabling port.

**\*Mar 1 00:51:17.067: %PM-4-ERR\_DISABLE: bpduguard error detected on Gig1/0/1, putting Gig1/0/1 in err-disable state**

\*Mar 1 00:51:18.082: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1 changed state to down

\*Mar 1 00:51:19.097: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1,changed state to down

**Note that the interface will be placed into “err-disabled” state**

**ALS2#sh int Gig1/0/1**

GigabitEthernet1/0/1 is down, line protocol is down **(err-disabled)**

As you can see, the interface is almost immediately “err-disabled”. Revert the configuration settings and issue the **shutdown** and **no shutdown** commands on GigabitEthernet1/0/11 to bring it back up.

ALS2(config)# **int GigabitEthernet1/0/1**

ALS2(config-if)# **shut**

ALS2(config-if)# **spanning-tree bpduguard disable**

ALS2(config-if)# **no shut**

ALS2(config-if)# **exit**

ALS2(config)#

**ALS2(config-if)#**

\*Mar 1 00:54:34.979: %LINK-5-CHANGED: Interface GigabitEthernet1/0/1, changed state to administratively down

\*Mar 1 00:54:37.185: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to down

\*Mar 1 00:54:40.516: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up

\*Mar 1 00:54:41.522: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed state to up

* 1. 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.

**If you enable BPDU filter globally then any interface with portfast enabled will not send or receive any BPDUs. When you receive a BPDU on a portfast enabled interface then it will lose its portfast status, disables BPDU filtering and STP operates normally, sending or receiving BPDUs on the port, as it would with any other STP port**

If BPDU filter is configured on an interface**, BPDU Filter causes the port to stop sending and processing received BPDUs altogether.** **This is the equivalent of disabling Spanning-tree on that interface.** 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.

Now configure BPDU Filter on **ALS1** interface GigabitEthernet1/0/11 (GIGETHERNET1/0/11 is an access port just ensure it is in UP/UP state)

**ALS1# conf t**

Enter configuration commands, one per line. End with CNTL/Z.

ALS1(config)# **int GigabitEthernet1/0/11**

ALS1(config-if)# **no shut**

ALS1(config-if)# **spanning-tree bpdufilter enable**

ALS1(config-if)# **exit**

ALS1(config)#

Verify that BPDU filter is enabled on interface:

**ALS1#show spanning-tree int GigabitEthernet1/0/11 detail**

Port 11 (GigabitEthernet1/0/11) of VLAN0099 is designated forwarding

Port path cost 4, Port priority 128, Port Identifier 128.11.

Designated root has priority 24675, address 6c41.6ace.3380

Designated bridge has priority 32867, address 08d0.9f48.d900

Designated port id is 128.11, designated path cost 4

Timers: message age 0, forward delay 0, hold 0

Number of transitions to forwarding state: 1

Link type is point-to-point by default

**Bpdu filter is enabled**

BPDU: sent 4, received 3

**Shutdown int GigabitEthernet1/0/11 on ALS1:**

ALS1(config)# **int GigabitEthernet1/0/11**

ALS1(config-if)# **shut**

* 1. Implement and Observe Root Guard

Root Guard helps prevent a root switch 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.

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

**DLS1#sh spanning-tree root**

**Root Hello Max Fwd**

**Vlan Root ID Cost Time Age Dly Root Port**

**---------------- -------------------- --------- ----- --- --- ------------**

VLAN0001 32769 6c41.6ace.3380 0 2 20 15

VLAN0099 24675 6c41.6ace.3380 0 2 20 15

**VLAN0100 24676 6c41.6ace.3380 0 2 20 15**

VLAN0110 24686 6c41.6ace.1400 4 2 20 15 Gi1/0/3

VLAN0120 24696 6c41.6ace.1400 4 2 20 15 Gi1/0/3

VLAN0666 25242 6c41.6ace.3380 0 2 20 15

DLS1#

From the ALS1 side of things, the root port is interface G1/0/1.

**ALS1#show span root | inc VLAN0100**

**VLAN0100 24676 6c41.6ace.3380 4 2 20 15 Gig1/0/1**

**ALS1#**

Configure Root Guard on DLS1 interface Gig1/0/1 (you may immediately see errors with another VLAN, like VLAN 1. 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 gig1/0/1**

DLS1(config-if)# **spanning-tree guard root**

DLS1(config-if)# **exit**

DLS1(config)#

Mar 30 02:34:30.623: %SPANTREE-2-ROOTGUARD\_CONFIG\_CHANGE: Root guard enabled on port GigabitEthernet1/0/1.

**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, you will see error message:**

**Mar 30 02:35:34.989: %SPANTREE-2-ROOTGUARD\_BLOCK: Root guard blocking port GigabitEthernet1/0/1 on VLAN0100.**

**Check the spanning tree interface status for Gig1/0/1:**

**DLS1#show spanning-tree interface gig1/0/1 | inc VLAN0100**

**VLAN0100 Desg BKN\*4 128.1 P2p \*ROOT\_Inc**

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#sh spanning-tree inconsistentports**

Name Interface Inconsistency

-------------------- ------------------------ ------------------

**VLAN0100 GigabitEthernet1/0/1 Root Inconsistent**

Number of inconsistent ports (segments) in the system : 1

DLS1#

To clear this, go back to ALS1 and issue the command

**ALS1(config)# 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 30 02:38:45.393: %SPANTREE-2-ROOTGUARD\_UNBLOCK: Root guard unblocking port GigabitEthernet1/0/1 on VLAN0100.**

**DLS1#show spanning-tree interface gig1/0/1 | inc VLAN0100**

**VLAN0100 Desg FWD 4 128.1 P2p**

DLS1#

For completeness, remove Root Guard from Gig1/0/1 on DLS1

DLS1# **conf t**

Enter configuration commands, one per line. End with CNTL/Z.

DLS1(config)# **int gig1/0/1**

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

DLS1(config-if)# **exit**

DLS1(config)#

1. Enable Classic EIGRP IPv4 on DLS1, DLS2 and R1
2. Enable port Gig1/0/11 on DLS1 switch as Layer 3 switched port, configure ip address “10.1.100.2/24

**DLS1#**

**interface gig1/0/11**

**no switchport**

**ip address 10.1.100.2 255.255.255.0**

**no shut**

1. Enable IP address on R1 Loopback0, Gig 0/0/1 interface as shown in the topology

**R1#**

**interface gig0/0/1**

**ip address 10.1.100.1 255.255.255.0**

**no shut**

**!**

**int loopback0**

**ip add 10.10.10.10 255.255.255.255**

**!**

**end**

1. **Ensure that SVI 99 on all switches is in UP/UP state**

1. Enable EIGRP AS 4 on DLS1, DLS2 SVI 99 and R1

**“Note” you will need to enable ip routing first on DLS switches**

**R1#**

**router eigrp 4**

**network 10.0.0.0**

**eigrp router-id 10.10.10.10**

**DLS1#**

**ip routing**

**!**

**router eigrp 4**

**network 10.0.0.0**

**eigrp router-id 5.5.5.5**

**DLS2#**

**ip routing**

**!**

**router eigrp 4**

**network 10.0.0.0**

**eigrp router-id 6.6.6.6**

1. Verify proper EIGRP neighbor relationship is established between DLS1/R1 and DLS1/DLS2

**DLS1#sh ip eigrp nei**

**EIGRP-IPv4 Neighbors for AS(4)**

H Address Interface Hold Uptime SRTT RTO Q Seq

(sec) (ms) Cnt Num

**1 10.1.99.102 Vl99** 11 00:01:29 1 100 0 3

**0 10.1.100.1 Gig1/0/11** 12 00:02:25 1591 5000 0 3

1. Configure ALS1 and ALS2 switches to use SVI 99 IP address “10.1.99.101/24” of DLS1 switch as their IP default gateway address

**ALS-2(config)#ip default-gateway 10.1.99.101**

**ALS-1(config)#ip default-gateway 10.1.99.101**

1. Verify that you can ping R1 loopback address from ALS switches

**ALS-1#ping 10.10.10.10**

Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1006 ms

ALS-1#

**ALS-2#ping 10.10.10.10**

Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1006 ms

ALS-2#

+++++++++++++++++++++End of Lab+++++++++++++++++++++

**DEMO Verifications**

* **If you have verified all the above steps, Demo to Instructor**
* **Submit the text document to BrightSpace with all the names of your team**

|  |  |  |
| --- | --- | --- |
| **Node** | **Show/Action command** | **Expected output** |
| ALS1  (5 points) | ping 10.10.10.10 |  |
| ALS2  (5 points) | ping 10.10.10.10 |  |