UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

M.Sc. Computer Science – Semester I

SOFTWARE DEFINED NETWORKING

JOURNAL

2022-2023

Seat No. __02

मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr./Ms. **Namrata Ashok Ambarkar** Seat No. **02** for the course of M.Sc. Computer Science - Semester I (CBCS) (Revised) during the academic year 2022-2023 in a satisfactory manner.

Subject In-charge

Head of Department

External Examiner

Index

9	**Implement BGP Communities** **1. Implement MP-BGP** **2. Implement eBGP for IPv4** **3. Implement BGP Path Manipulation**			
10	**Implement IPsec Site-to-Site VPNs** **1. Implement GRE over IPsec Site-to-Site VPNs** **2. Implement VRF Lite**			

PRACTICAL NO -01

Aim:- **Implement IP SLA (IP Service Level Agreement)**

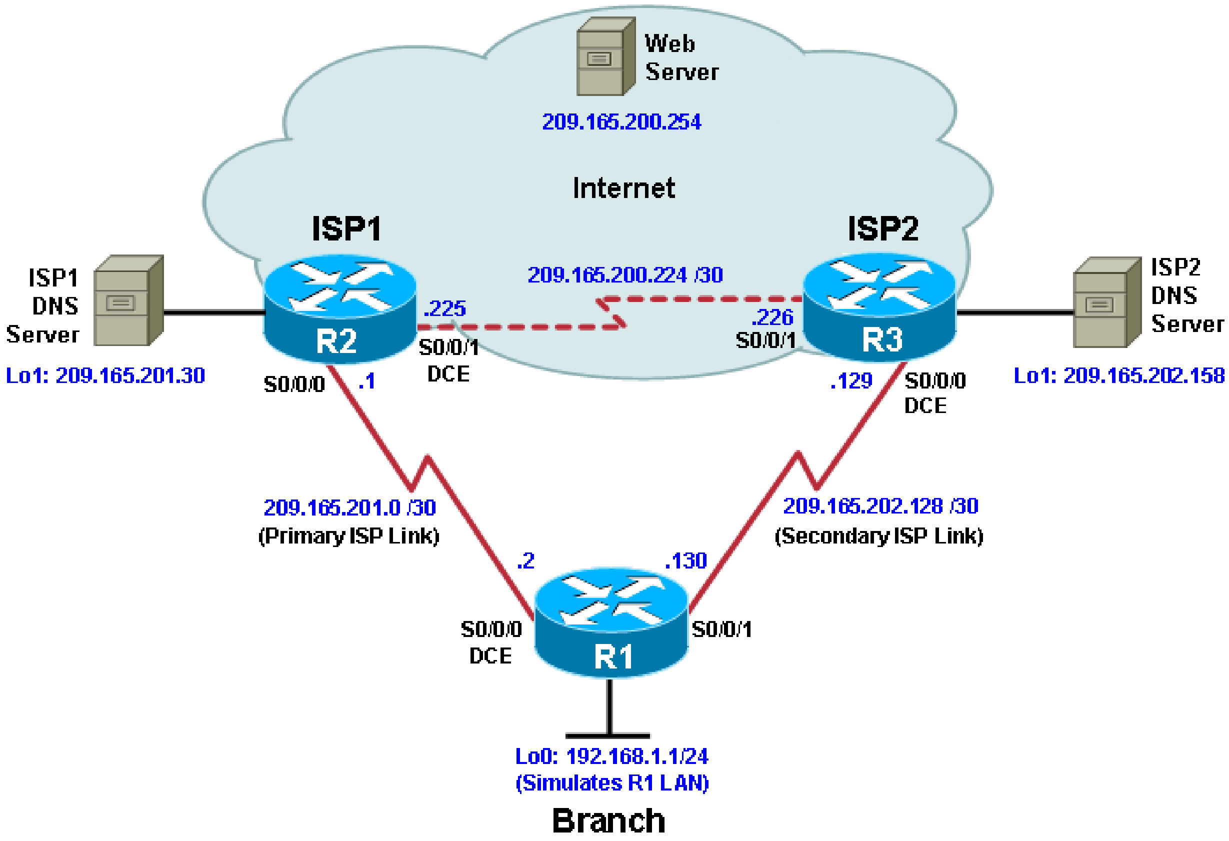
What is IP service level agreement?

IP SLA (Internet protocol service level agreement) is **a feature of the Cisco Internetwork Operating System (Cisco IOS) that allows an IT professional to collect information about network performance in real time**.
When you use IP SLA for a simple ping then you only have to **configure your local router**. However when you want to use it for some more advanced things like sending RTP packets then you have to configure the remote router to respond to your IP SLA traffic.

There are three basic types of SLAs: **customer, internal and multilevel service-level agreements**. A customer service-level agreement is between a service provider and its external customers.

Configure IP SLA Tracking and Path Control

Topology



Objectives

- Configure and verify the IP SLA feature.

- Test the IP SLA tracking feature.
- Verify the configuration and operation using **show** and **debug** commands.

Background

You want to experiment with the Cisco IP Service Level Agreement (SLA) feature to study how it could be of value to your organization.

At times, a link to an ISP could be operational, yet users cannot connect to any other outside Internet resources. The problem might be with the ISP or downstream from them. Although policy-based routing (PBR) can be implemented to alter path control, you will implement the Cisco IOS SLA feature to monitor this behavior and intervene by injecting another default route to a backup ISP.

To test this, you have set up a three-router topology in a lab environment. Router R1 represents a branch office connected to two different ISPs. ISP1 is the preferred connection to the Internet, while ISP2 provides a backup link. ISP1 and ISP2 can also interconnect, and both can reach the web server. To monitor ISP1 for failure, you will configure IP SLA probes to track the reachability to the ISP1 DNS server. If connectivity to the ISP1 server fails, the SLA probes detect the failure and alter the default static route to point to the ISP2 server.

Note: This lab uses Cisco 1941 routers with Cisco IOS Release 15.2 with IP Base. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco IOS Release 15.2 or comparable)
- Serial and Ethernet cables

Step 1: Configure loopbacks and assign addresses.

a. Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear the previous configurations. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to them as well as the serial interfaces on R1, ISP1, and ISP2.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
hostname R1


interface Loopback 0

 description R1 LAN

 ip address 192.168.1.1 255.255.255.0


interface Serial0/0/0

 description R1 --> ISP1
 ip address 209.165.201.2 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description R1 --> ISP2
 ip address 209.165.202.130 255.255.255.252
 bandwidth 128
 no shutdown
```

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255

interface Serial0/0/0
 description ISP1 --> R1
 ip address 209.165.201.1 255.255.255.252
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP1 --> ISP2
 ip address 209.165.200.225 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown
```

Router ISP2 (R3)

```
hostname ISP2

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0
 description ISP2 --> R1
 ip address 209.165.202.129 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP2 --> ISP1
 ip address 209.165.200.226 255.255.255.252
 bandwidth 128
 no shutdown
```

b. Verify the configuration by using the **show interfaces description** command. The output from router R1 is shown here as an example.

```
R1# show interfaces description | include up
Se0/0/0                    up          up      R1 --> ISP1
Se0/0/1                    up          up      R1 --> ISP2
Lo0                        up          up      R1 LAN
R1#
```

All three interfaces should be active. Troubleshoot if necessary.

Step 2: Configure static routing.

The current routing policy in the topology is as follows:

- Router R1 establishes connectivity to the Internet through ISP1 using a default static route.
- ISP1 and ISP2 have dynamic routing enabled between them, advertising their respective public address pools.
- ISP1 and ISP2 both have static routes back to the ISP LAN.

Note: For the purpose of this lab, the ISPs have a static route to an RFC 1918 private network address on the branch router R1. In an actual branch implementation, Network Address Translation (NAT) would be configured for all traffic exiting the branch LAN. Therefore, the static routes on the ISP routers would be pointing to the provided public pool of the branch office.

a. Implement the routing policies on the respective routers. You can copy and paste the following configurations.

Router R1

```
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1

R1(config)#
```

Router ISP1 (R2)

```
ISP1(config)# router eigrp 1

ISP1(config-router)# network 209.165.200.224 0.0.0.3

ISP1(config-router)# network 209.165.201.0 0.0.0.31

ISP1(config-router)# no auto-summary

ISP1(config-router)# exit

ISP1(config)#

ISP1(config-router)# ip route 192.168.1.0 255.255.255.0 209.165.201.2

ISP1(config)#
```

Router ISP2 (R3)

```
ISP2(config)# router eigrp 1

ISP2(config-router)# network 209.165.200.224 0.0.0.3

ISP2(config-router)# network 209.165.202.128 0.0.0.31

ISP2(config-router)# no auto-summary

ISP2(config-router)# exit

ISP2(config)#

ISP2(config)# ip route 192.168.1.0 255.255.255.0 209.165.202.130

ISP2(config)#
```

EIGRP neighbor relationship messages on ISP1 and ISP2 should be generated. Troubleshoot if necessary.

b. The Cisco IOS IP SLA feature enables an administrator to monitor network performance between Cisco devices (switches or routers) or from a Cisco device to a remote IP device. IP SLA probes continuously check the reachability of a specific destination, such as a provider edge router interface, the DNS server of the ISP, or any other specific destination, and can conditionally announce a default route only if the connectivity is verified.

Before implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2 DNS server to verify connectivity. You can copy the following Tcl script and paste it into R1.

```
foreach address {

209.165.200.254

209.165.201.30

209.165.202.158

} {

ping $address source 192.168.1.1

}
```

All pings should be successful. Troubleshoot if necessary.

c. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server. You can copy the following Tcl script and paste it into R1.

```
foreach address {

209.165.200.254

209.165.201.30

209.165.202.158

} {

trace $address source 192.168.1.1

}
```

Through which ISP is traffic flowing?

All traffic is routed to the ISP1 router.

Step 3: Configure IP SLA probes.

When the reachability tests are successful, you can configure the Cisco IOS IP SLAs probes. Different types of probes can be created, including FTP, HTTP, and jitter probes.

In this scenario, you will configure ICMP echo probes.

a. Create an ICMP echo probe on R1 to the primary DNS server on ISP1 using the **ip sla** command.

R1(config)# **ip sla 11**

```
R1(config-ip-sla)# icmp-echo 209.165.201.30

R1(config-ip-sla-echo)# frequency 10

R1(config-ip-sla-echo)# exit

R1(config)#

R1(config)# ip sla schedule 11 life forever start-time now

R1(config)#
```

The operation number of 11 is only locally significant to the router. The **frequency 10** command schedules the connectivity test to repeat every 10 seconds. The probe is scheduled to start now and to run forever.

b. Verify the IP SLAs configuration of operation 11 using the **show ip sla configuration 11** command.

```
R1# show ip sla configuration 11

IP SLAs Infrastructure Engine-III

Entry number: 11

Owner:

Tag:

Operation timeout (milliseconds): 5000

Type of operation to perform: icmp-echo

Target address/Source address: 209.165.201.30/0.0.0.0

Type Of Service parameter: 0x0

Request size (ARR data portion): 28

Verify data: No

Vrf Name:

Schedule:

  Operation frequency (seconds): 10  (not considered if randomly scheduled)

  Next Scheduled Start Time: Start Time already passed

  Group Scheduled : FALSE

  Randomly Scheduled : FALSE

  Life (seconds): Forever

  Entry Ageout (seconds): never

  Recurring (Starting Everyday): FALSE

  Status of entry (SNMP RowStatus): Active

Threshold (milliseconds): 5000

Distribution Statistics:
  Number of statistic hours kept: 2

  Number of statistic distribution buckets kept: 1

  Statistic distribution interval (milliseconds): 20

Enhanced History:

History Statistics:

  Number of history Lives kept: 0

  Number of history Buckets kept: 15

  History Filter Type: None



R1#
```

The output lists the details of the configuration of operation 11. The operation is an ICMP echo to 209.165.201.30, with a frequency of 10 seconds, and it has already started (the start time has already passed).

c. Issue the **show ip sla statistics** command to display the number of successes, failures, and results of the latest operations.

R1# **show ip sla statistics**

```
IPSLAs Latest Operation Statistics


IPSLA operation id: 11

        Latest RTT: 8 milliseconds

Latest operation start time: 10:33:18 UTC Sat Jan 10 2015

Latest operation return code: OK

Number of successes: 51

Number of failures: 0

Operation time to live: Forever



R1#
```

You can see that operation 11 has already succeeded five times, has had no failures, and the last operation returned an OK result.

d. Although not actually required because IP SLA session 11 alone could provide the desired fault tolerance, create a second probe, 22, to test connectivity to the second DNS server located on router ISP2.

R1(config)# **ip sla 22**

```
R1(config-ip-sla)# icmp-echo 209.165.202.158
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)#
R1(config)# ip sla schedule 22 life forever start-time now
R1(config)# end
R1#
```

e. Verify the new probe using the **show ip sla configuration** and **show ip sla statistics** commands.

```
R1# show ip sla configuration 22
IP SLAs Infrastructure Engine-III
Entry number: 22
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.202.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None


R1#


R1# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
```

```
  Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
  Number of statistic hours kept: 2
  Number of statistic distribution buckets kept: 1
  Statistic distribution interval (milliseconds): 20
History Statistics:
  Number of history Lives kept: 0
  Number of history Buckets kept: 15
  History Filter Type: None
Enhanced History:


R1#
R1# show ip sla statistics 22
IPSLAs Latest Operation Statistics


IPSLA operation id: 22
        Latest RTT: 16 milliseconds
Latest operation start time: 10:38:29 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 82
Number of failures: 0
Operation time to live: Forever



R1#
```

The output lists the details of the configuration of operation 22. The operation is an ICMP echo to 209.165.202.158, with a frequency of 10 seconds, and it has already started (the start time has already passed). The statistics also prove that operation 22 is active.

Step 4: Configure tracking options.

Although PBR could be used, you will configure a floating static route that appears or disappears depending on the success or failure of the IP SLA.

a. On R1, remove the current default route and replace it with a floating static route having an administrative distance of 5.

```
R1(config)# no ip route 0.0.0.0 0.0.0.0 209.165.201.1

R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 5

R1(config)# exit
```

b. Verify the routing table.

```
R1# show ip route | begin Gateway

Gateway of last resort is 209.165.201.1 to network 0.0.0.0


S*    0.0.0.0/0 [5/0] via 209.165.201.1

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C        192.168.1.0/24 is directly connected, Loopback0

L        192.168.1.1/32 is directly connected, Loopback0

      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C        209.165.201.0/30 is directly connected, Serial0/0/0

L        209.165.201.2/32 is directly connected, Serial0/0/0

      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C        209.165.202.128/30 is directly connected, Serial0/0/1

L        209.165.202.130/32 is directly connected, Serial0/0/1

R1#
```

Notice that the default static route is now using the route with the administrative distance of 5. The first tracking object is tied to IP SLA object 11.

c. From global configuration mode on R1, use the **track 1 ip sla 11 reachability** command to enter the config-track subconfiguration mode.

```
R1(config)# track 1 ip sla 11 reachability

R1(config-track)#
```

d. Specify the level of sensitivity to changes of tracked objects to 10 seconds of down delay and 1 second of up delay using the **delay down 10 up 1** command. The delay helps to alleviate the effect of flapping objects—objects that are going down and up rapidly. In this situation, if the DNS server fails momentarily and comes back up within 10 seconds, there is no impact.

```
R1(config-track)# delay down 10 up 1

R1(config-track)# exit

R1(config)#
```

e. To view routing table changes as they happen, first enable the **debug ip routing** command.

```
R1# debug ip routing

IP routing debugging is on
```

```
R1#
```

f. Configure the floating static route that will be implemented when tracking object 1 is active. Use the **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1** command to create a floating static default route via 209.165.201.1 (ISP1). Notice that this command references the tracking object number 1, which in turn references IP SLA operation number 11.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1

R1(config)#

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.201.1   0 1048578


Jan 10 10:45:39.119: RT: closer admin distance for 0.0.0.0, flushing 1 routes

Jan 10 10:45:39.119: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.201.1   0 1048578


Jan 10 10:45:39.119: RT: rib update return code: 17

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.201.1   0 1048578


Jan 10 10:45:39.119: RT: rib update return code: 17

R1(config)#
```

Notice that the default route with an administrative distance of 5 has been immediately flushed because of a route with a better admin distance. It then adds the new default route with the admin distance of 2.

g. Repeat the steps for operation 22, track number 2, and assign the static route an admin distance higher than track 1 and lower than 5. On R1, copy the following configuration, which sets an admin distance of 3.

```
R1(config)# track 2 ip sla 22 reachability

R1(config-track)# delay down 10 up 1

R1(config-track)# exit

R1(config)#

R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2

R1(config)#
```

h. Verify the routing table again.

```
R1#show ip route | begin Gateway

Gateway of last resort is 209.165.201.1 to network 0.0.0.0


S*    0.0.0.0/0 [2/0] via 209.165.201.1

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C        192.168.1.0/24 is directly connected, Loopback0

L        192.168.1.1/32 is directly connected, Loopback0

      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C        209.165.201.0/30 is directly connected, Serial0/0/0

L        209.165.201.2/32 is directly connected, Serial0/0/0

      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C        209.165.202.128/30 is directly connected, Serial0/0/1

L        209.165.202.130/32 is directly connected, Serial0/0/1

R1#
```

Although a new default route was entered, its administrative distance is not better than 2. Therefore, it does not replace the previously entered default route.

Step 5: Verify IP SLA operation.

In this step you observe and verify the dynamic operations and routing changes when tracked objects fail. The following summarizes the process:

- Disable the DNS loopback interface on ISP1 (R2).
- Observe the output of the **debug** command on R1.
- Verify the static route entries in the routing table and the IP SLA statistics of R1.
- Re-enable the loopback interface on ISP1 (R2) and again observe the operation of the IP SLA tracking feature.

a. On ISP1, disable the loopback interface 1.

```
ISP1(config-if)# int lo1

ISP1(config-if)# shutdown

ISP1(config-if)#

Jan 10 10:53:25.091: %LINK-5-CHANGED: Interface Loopback1, changed state to
administratively down

Jan 10 10:53:26.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to down

ISP1(config-if)#
```

b. On R1, observe the **debug** output being generated. Recall that R1 will wait up to 10 seconds before initiating action therefore several seconds will elapse before the output is generated.

```
R1#

Jan 10 10:53:59.551: %TRACK-6-STATE: 1 ip sla 11 reachability Up -> Down

Jan 10 10:53:59.551: RT: del 0.0.0.0 via 209.165.201.1, static metric [2/0]

Jan 10 10:53:59.551: RT: delete network route to 0.0.0.0/0

Jan 10 10:53:59.551: RT: default path has been cleared

Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.202.129    0 1048578


Jan 10 10:53:59.551: RT: add 0.0.0.0/0 via 209.165.202.129, static metric [3/0]

Jan 10 10:53:59.551: RT: default path is now 0.0.0.0 via 209.165.202.129

Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.201.1    0 1048578


Jan 10 10:53:59.551: RT: rib update return code: 17

Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.202.129    0 1048578


Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.201.1    0 1048578


Jan 10 10:53:59.551: RT: rib update return code: 17

R1#
```

The tracking state of track 1 changes from up to down. This is the object that tracked reachability for IP SLA object 11, with an ICMP echo to the ISP1 DNS server at 209.165.201.30.

R1 then proceeds to delete the default route with the administrative distance of 2 and installs the next highest default route to ISP2 with the administrative distance of 3.

c. On R1, verify the routing table.

```
R1# show ip route | begin Gateway

Gateway of last resort is 209.165.202.129 to network 0.0.0.0
```

```
S*    0.0.0.0/0 [3/0] via 209.165.202.129
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30 is directly connected, Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
R1#
```

The new static route has an administrative distance of 3 and is being forwarded to ISP2 as it should.

d. Verify the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics


IPSLA operation id: 11
              Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 11:01:08 UTC Sat Jan 10 2015
Latest operation return code: Timeout
Number of successes: 173
Number of failures: 45
Operation time to live: Forever




IPSLA operation id: 22
              Latest RTT: 8 milliseconds
Latest operation start time: 11:01:09 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 218
Number of failures: 0
```

```
Operation time to live: Forever



R1#
```

Notice that the latest return code is **Timeout** and there have been 45 failures on IP SLA object 11.

e. On R1, initiate a trace to the web server from the internal LAN IP address.

```
R1# trace 209.165.200.254 source 192.168.1.1

Type escape sequence to abort.

Tracing the route to 209.165.200.254

VRF info: (vrf in name/id, vrf out name/id)

 1 209.165.202.129 4 msec *  *

R1#
```

This confirms that traffic is leaving router R1 and being forwarded to the ISP2 router.

f. On ISP1, re-enable the DNS address by issuing the **no shutdown** command on the loopback 1 interface to examine the routing behavior when connectivity to the ISP1 DNS is restored.

```
ISP1(config-if)# no shutdown

Jan 10 11:05:45.847: %LINK-3-UPDOWN: Interface Loopback1, changed state to up

Jan 10 11:05:46.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up

ISP1(config-if)#
```

Notice the output of the **debug ip routing** command on R1.

```
R1#

Jan 10 11:06:20.551: %TRACK-6-STATE: 1 ip sla 11 reachability Down -> Up

Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)  :

  via 209.165.201.1   0 1048578


Jan 10 11:06:20.551: RT: closer admin distance for 0.0.0.0, flushing 1 routes

Jan 10 11:06:20.551: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]

Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)  :

  via 209.165.202.129   0 1048578
```

```
Jan 10 11:06:20.551: RT: rib update return code: 17

Jan 10 11:06:20.551: RT: u

R1#pdating static 0.0.0.0/0 (0x0)  :

    via 209.165.202.129   0 1048578


Jan 10 11:06:20.551: RT: rib update return code: 17

Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)  :

    via 209.165.201.1   0 1048578


Jan 10 11:06:20.551: RT: rib update return code: 17

R1#
```

Now the IP SLA 11 operation transitions back to an up state and reestablishes the default static route to ISP1 with an administrative distance of 2.

g. Again examine the IP SLA statistics.

```
R1# show ip sla statistics

IPSLAs Latest Operation Statistics


IPSLA operation id: 11

            Latest RTT: 8 milliseconds

Latest operation start time: 11:07:38 UTC Sat Jan 10 2015

Latest operation return code: OK

Number of successes: 182

Number of failures: 75

Operation time to live: Forever




IPSLA operation id: 22

            Latest RTT: 16 milliseconds

Latest operation start time: 11:07:39 UTC Sat Jan 10 2015

Latest operation return code: OK

Number of successes: 257
Number of failures: 0

Operation time to live: Forever



R1#
```

The IP SLA 11 operation is active again, as indicated by the OK return code, and the number of successes is incrementing.

h. Verify the routing table.

```
R1# show ip route | begin Gateway

Gateway of last resort is 209.165.201.1 to network 0.0.0.0


S*    0.0.0.0/0 [2/0] via 209.165.201.1

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C        192.168.1.0/24 is directly connected, Loopback0

L        192.168.1.1/32 is directly connected, Loopback0

      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C        209.165.201.0/30 is directly connected, Serial0/0/0

L        209.165.201.2/32 is directly connected, Serial0/0/0

      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C        209.165.202.128/30 is directly connected, Serial0/0/1

L        209.165.202.130/32 is directly connected, Serial0/0/1

R1#
```

The default static through ISP1 with an administrative distance of 2 is reestablished.

There are many possibilities available with object tracking and Cisco IOS IP SLAs. As shown in this lab, a probe can be based on reachability, changing routing operations, and path control based on the ability to reach an object. However, Cisco IOS IP SLAs also allow paths to be changed based on network conditions such as delay, load, and other factors.

Before deploying a Cisco IOS IP SLA solution, the impact of the additional probe traffic being generated should be considered, including how that traffic affects bandwidth utilization, and congestion levels. Tuning the configuration (for example, with the **delay** and **frequency** commands) is critical to mitigate possible issues related to excessive transitions and route changes in the presence of flapping tracked objects.

The benefits of running IP SLAs should be carefully evaluated. The IP SLA is an additional task that must be performed by the router's CPU. A large number of intensive SLAs could be a significant burden on the CPU, possibly interfering with other router functions and having detrimental impact on the overall router performance. The CPU load should be monitored after the SLAs are deployed to verify that they do not cause excessive utilization of the router CPU.

Device Configurations (Instructor version)

Router R1

```
hostname R1
!
interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/0
 description R1 --> ISP1
 ip address 209.165.201.2 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown
!
interface Serial0/0/1
 description R1 --> ISP2
 ip address 209.165.202.130 255.255.255.252
 bandwidth 128
 no shutdown
!
! Initial static route to establish ISP1 connectivity
ip route 0.0.0.0 0.0.0.0 209.165.201.1 5
!
ip sla 11
 icmp-echo 209.165.201.30
 frequency 10
!
ip sla schedule 11 life forever start-time now
!
track 1 ip sla 11 reachability
 delay down 10 up 1
```

```
!
ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
!
ip sla 22
 icmp-echo 209.165.202.158
 frequency 10
!
ip sla schedule 22 life forever start-time now
!
track 2 ip sla 22 reachability
 delay down 10 up 1
!
ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
!
end
```

Router ISP1 (R2)

```
hostname ISP1
!
interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255
!
interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255
!
interface Serial0/0/0
 description ISP1 --> R1
 ip address 209.165.201.1 255.255.255.252
 bandwidth 128
 no shutdown
!
```

```
interface Serial0/0/1
 description ISP1 --> ISP2
 ip address 209.165.200.225 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown
!
router eigrp 1
 network 209.165.200.224 0.0.0.3
 network 209.165.201.0 0.0.0.31
 no auto-summary
!
ip route 192.168.1.0 255.255.255.0 209.165.201.2
!
end
```

Router ISP2 (R3)

```
hostname ISP2
!
interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255
!
interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255
!
interface Serial0/0/0
 description ISP2 --> R1
 ip address 209.165.202.129 255.255.255.252
 no fair-queue
 clock rate 128000
 bandwidth 128
 no shutdown
!
interface Serial0/0/1
 description ISP2 --> ISP1
 ip address 209.165.200.226 255.255.255.252
 bandwidth 128
 no shutdown
!
router eigrp 1
 network 209.165.200.224 0.0.0.3
 network 209.165.202.128 0.0.0.31
 no auto-summary
!
ip route 192.168.1.0 255.255.255.0 209.165.202.130
!
end
```

PRACTICAL NO -02

Aim :- **Implement IPv4 ACLs**

1. Standard

2. Extended

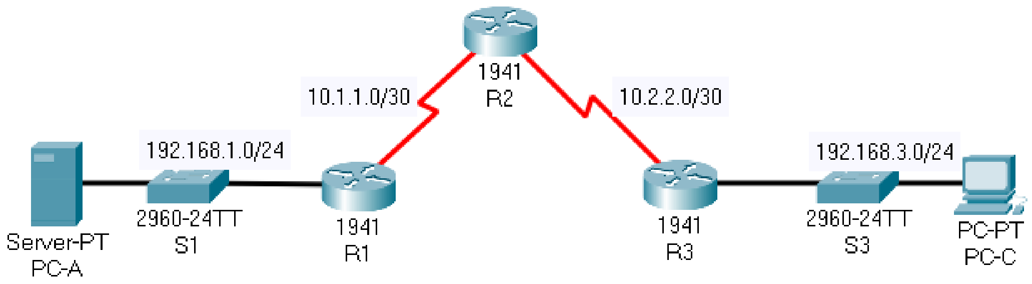
What is standard ACL and extended ACL?

A standard ACL allows or denies traffic access based on the source IP address, while an extended access control list can filter packets with a higher degree of specification. It can determine the types of traffic it allows or blocks beyond just the IP address to include TCP, ICMP, and UDP,
How do you set up an extended ACL?

To configure an extended named ACL, **enter the ip access-list extended command**. The options at the ACL configuration level and the syntax for the ip access-group command are the same for numbered and named ACLs and are described in Extended numbered ACL configuration and Extended numbered ACL configuration.

How to configure standard ACL?

ACL number for the standard ACLs has to be between 1–99 and 1300–1999. Once the access list is created, it needs to be applied to an interface. You do that by **using the ip access-group ACL_NUMBER in|out interface subcommand**. in and out keywords specify in which direction you are activating the ACL.



Packet Tracer - Configure IP ACLs to Mitigate Attacks

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
	Lo0	192.168.2.1	255.255.255.0	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/6
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Verify connectivity among devices before firewall configuration.
- Use ACLs to ensure remote access to the routers is available only from management station PC-C.
- Configure ACLs on R1 and R3 to mitigate attacks.
- Verify ACL functionality.

Background/Scenario

Access to routers R1, R2, and R3 should only be permitted from PC-C, the management station. PC-C is also used for connectivity testing to PC-A, which is a server providing DNS, SMTP, FTP, and HTTPS services.

Standard operating procedure is to apply ACLs on edge routers to mitigate common threats based on source and destination IP address. In this activity, you will create ACLs on edge routers R1 and R3 to achieve this goal. You will then verify ACL functionality from internal and external hosts.

The routers have been pre-configured with the following:

- Enable password: **ciscoenpa55**
- Password for console: **ciscoconpa55**
- SSH logon username and password: **SSHadmin/ciscosshpa55**
- IP addressing
- Static routing

Part 1: Verify Basic Network Connectivity

Verify network connectivity prior to configuring the IP ACLs.

Step 1: From PC-A, verify connectivity to PC-C and R2.

a. From the command prompt, ping **PC-C** (192.168.3.3).

b. From the command prompt, establish an SSH session to **R2** Lo0 interface (192.168.2.1) using username **SSHadmin** and password **ciscosshpa55**. When finished, exit the SSH session.

```
SERVER> ssh -l SSHadmin 192.168.2.1
```

Step 2: From PC-C, verify connectivity to PC-A and R2.

a. From the command prompt, ping **PC-A** (192.168.1.3).

b. From the command prompt, establish an SSH session to **R2** Lo0 interface (192.168.2.1) using username **SSHadmin** and password **ciscosshpa55**. Close the SSH session when finished.

c. Open a web browser to the **PC-A** server (192.168.1.3) to display the web page. Close the browser when done.

Part 2: Secure Access to Routers

Step 1: Configure ACL 10 to block all remote access to the routers except from PC-C.

Use the **access-list** command to create a numbered IP ACL on **R1**, **R2**, and **R3**.

Step 2: Apply ACL 10 to ingress traffic on the VTY lines.

Use the **access-class** command to apply the access list to incoming traffic on the VTY lines.

Step 3: Verify exclusive access from management station PC-C.

a. Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful).

b. Establish an SSH session to 192.168.2.1 from **PC-A** (should fail).

Part 3: Create a Numbered IP ACL 120 on R1

Create an IP ACL numbered 120 with the following rules:

- Permit any outside host to access DNS, SMTP, and FTP services on server **PC-A.**
- Deny any outside host access to HTTPS services on **PC-A.**
- Permit **PC-C** to access **R1** via SSH.

Note: Check Results will not show a correct configuration for ACL 120 until you modify it in Part 4.

Step 1: Verify that PC-C can access the PC-A via HTTPS using the web browser.

Be sure to disable HTTP and enable HTTPS on server **PC-A**.

Step 2: Configure ACL 120 to specifically permit and deny the specified traffic.

Use the **access-list** command to create a numbered IP ACL.

Step 3: Apply the ACL to interface S0/0/0.

Use the **ip access-group** command to apply the access list to incoming traffic on interface S0/0/0.

Step 4: Verify that PC-C cannot access PC-A via HTTPS using the web browser.

Part 4: Modify an Existing ACL on R1

Permit ICMP echo replies and destination unreachable messages from the outside network (relative to **R1**). Deny all other incoming ICMP packets.

Step 1: Verify that PC-A cannot successfully ping the loopback interface on R2.

Step 2: Make any necessary changes to ACL 120 to permit and deny the specified traffic.

Use the **access-list** command to create a numbered IP ACL.

Step 3: Verify that PC-A can successfully ping the loopback interface on R2.

Part 5: Create a Numbered IP ACL 110 on R3

Deny all outbound packets with source address outside the range of internal IP addresses on **R3**.

Step 1: Configure ACL 110 to permit only traffic from the inside network.

Use the **access-list** command to create a numbered IP ACL.

Step 2: Apply the ACL to interface G0/1.

Use the **ip access-group** command to apply the access list to incoming traffic on interface G0/1.

Part 6: Create a Numbered IP ACL 100 on R3

On **R3**, block all packets containing the source IP address from the following pool of addresses: any RFC 1918 private addresses, 127.0.0.0/8, and any IP multicast address. Since **PC-C** is being used for remote administration, permit SSH traffic from the 10.0.0.0/8 network to return to the host **PC-C**.

Step 1: Configure ACL 100 to block all specified traffic from the outside network.

You should also block traffic sourced from your own internal address space if it is not an RFC 1918 address. In this activity, your internal address space is part of the private address space specified in RFC 1918.

Use the **access-list** command to create a numbered IP ACL.

Step 2: Apply the ACL to interface Serial 0/0/1.

Use the **ip access-group** command to apply the access list to incoming traffic on interface Serial 0/0/1.

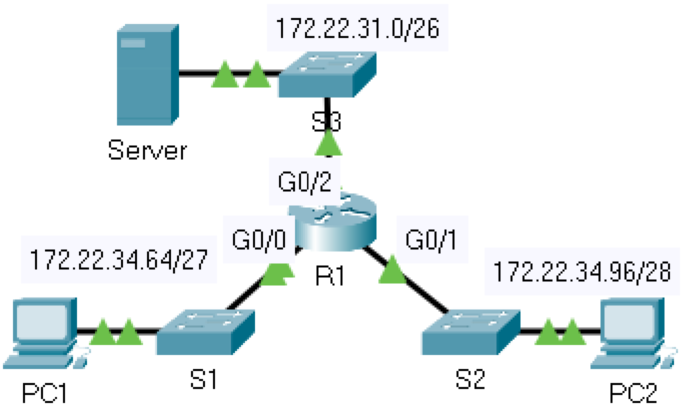
Step 3: Confirm that the specified traffic entering interface Serial 0/0/1 is handled correctly.

a. From the PC-C command prompt, ping the PC-A server. The ICMP echo replies are blocked by the ACL since they are sourced from the 192.168.0.0/16 address space.

b. Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful).

Step 4: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.



Packet Tracer - Configuring Extended ACLs - Scenario 1

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	172.22.34.65	255.255.255.224	N/A
	G0/1	172.22.34.97	255.255.255.240	N/A
	G0/2	172.22.34.1	255.255.255.192	N/A
Server	NIC	172.22.34.62	255.255.255.192	172.22.34.1
PC1	NIC	172.22.34.66	255.255.255.224	172.22.34.65
PC2	NIC	172.22.34.98	255.255.255.240	172.22.34.97

Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Configure, Apply and Verify an Extended Named ACL

Background / Scenario

Two employees need access to services provided by the server. **PC1** needs only FTP access while **PC2** needs only web access. Both computers are able to ping the server, but not each other.

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Step 1: Configure an ACL to permit FTP and ICMP.

a. From global configuration mode on **R1**, enter the following command to determine the first valid number for an extended access list.

```
R1(config)# access-list ?
  <1-99>     IP standard access list
  <100-199>  IP extended access list
```

b. Add **100** to the command, followed by a question mark.

```
R1(config)# access-list 100 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
```

c. To permit FTP traffic, enter **permit,** followed by a question mark.

```
R1(config)# access-list 100 permit ?
  ahp    Authentication Header Protocol
  eigrp  Cisco's EIGRP routing protocol
  esp    Encapsulation Security Payload
  gre    Cisco's GRE tunneling
  icmp   Internet Control Message Protocol
  ip     Any Internet Protocol
  ospf   OSPF routing protocol
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
```

d. This ACL permits FTP and ICMP. ICMP is listed above, but FTP is not, because FTP uses TCP. Therefore,enter **tcp** to further refine the ACL help.

```
R1(config)# access-list 100 permit tcp ?
  A.B.C.D  Source address
  any      Any source host
  host     A single source host
```

e. Notice that we could filter just for **PC1** by using the **host** keyword or we could allow **any** host. In this case, any device is allowed that has an address belonging to the 172.22.34.64/27 network. Enter the network address, followed by a question mark.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 ?
  A.B.C.D  Source wildcard bits
```

f. Calculate the wildcard mask determining the binary opposite of a subnet mask.

11111111.11111111.11111111.11100000 = 255.255.255.224

00000000.00000000.00000000.000**11111** = 0.0.0.31

g. Enter the wildcard mask, followed by a question mark.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 ?
  A.B.C.D  Destination address
  any      Any destination host
  eq       Match only packets on a given port number
  gt       Match only packets with a greater port number
  host     A single destination host
  lt       Match only packets with a lower port number
  neq      Match only packets not on a given port number
  range    Match only packets in the range of port numbers
```

h. Configure the destination address. In this scenario, we are filtering traffic for a single destination, which is the server. Enter the **host** keyword followed by the server's IP address.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 ?
  dscp        Match packets with given dscp value
  eq          Match only packets on a given port number
  established established
  gt          Match only packets with a greater port number
  lt          Match only packets with a lower port number
  neq         Match only packets not on a given port number
  precedence  Match packets with given precedence value
  range       Match only packets in the range of port numbers
  <cr>
```

i. Notice that one of the options is **<cr>** (carriage return). In other words, you can press **Enter** and the statement would permit all TCP traffic. However, we are only permitting FTP traffic; therefore, enter the **eq** keyword, followed by a question mark to display the available options. Then, enter **ftp** and press **Enter**.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ?
  <0-65535>  Port number
  ftp        File Transfer Protocol (21)
  pop3       Post Office Protocol v3 (110)
  smtp       Simple Mail Transport Protocol (25)
  telnet     Telnet (23)
  www        World Wide Web (HTTP, 80)
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp
```

j. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC1** to **Server**. Note that the access list number remains the same and no particular type of ICMP traffic needs to be specified.

```
R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host 172.22.34.62
```

k. All other traffic is denied, by default.

Step 2: Apply the ACL on the correct interface to filter traffic.

From **R1**'s perspective, the traffic that ACL 100 applies to is inbound from the network connected to Gigabit Ethernet 0/0 interface. Enter interface configuration mode and apply the ACL.

```
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# ip access-group 100 in
```

Step 3: Verify the ACL implementation.

a. Ping from **PC1** to **Server**. If the pings are unsuccessful, verify the IP addresses before continuing.

b. FTP from **PC1** to **Server**. The username and password are both **cisco**.

```
PC> ftp 172.22.34.62
```

c. Exit the FTP service of the **Server**.

```
ftp> quit
```

d. Ping from **PC1** to **PC2**. The destination host should be unreachable, because the traffic was not explicitly permitted.

Part 2: Configure, Apply and Verify an Extended Named ACL

Step 1: Configure an ACL to permit HTTP access and ICMP.

a. Named ACLs start with the **ip** keyword. From global configuration mode of **R1**, enter the following command, followed by a question mark.

```
R1(config)# ip access-list ?
```

```
extended  Extended Access List
standard  Standard Access List
```

b. You can configure named standard and extended ACLs. This access list filters both source and destination IP addresses; therefore, it must be extended. Enter **HTTP_ONLY** as the name. (For Packet Tracer scoring, the name is case-sensitive.)

R1(config)# **ip access-list extended HTTP_ONLY**

c. The prompt changes. You are now in extended named ACL configuration mode. All devices on the **PC2** LAN need TCP access. Enter the network address, followed by a question mark.

R1(config-ext-nacl)# **permit tcp 172.22.34.96 ?**

```
A.B.C.D  Source wildcard bits
```

d. An alternative way to calculate a wildcard is to subtract the subnet mask from 255.255.255.255.

```
  255.255.255.255
- 255.255.255.240
----------------
=   0.  0.  0. 15
```

R1(config-ext-nacl)# **permit tcp 172.22.34.96 0.0.0.15 ?**

e. Finish the statement by specifying the server address as you did in Part 1 and filtering **www** traffic.

R1(config-ext-nacl)# **permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www**

f. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC2** to **Server**. Note: The prompt remains the same and a specific type of ICMP traffic does not need to be specified.

R1(config-ext-nacl)# **permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62**

g. All other traffic is denied, by default. Exit out of extended named ACL configuration mode.

Step 2: Apply the ACL on the correct interface to filter traffic.

From **R1**'s perspective, the traffic that access list **HTTP_ONLY** applies to is inbound from the network connected to Gigabit Ethernet 0/1 interface. Enter the interface configuration mode and apply the ACL.

R1(config)# **interface gigabitEthernet 0/1**

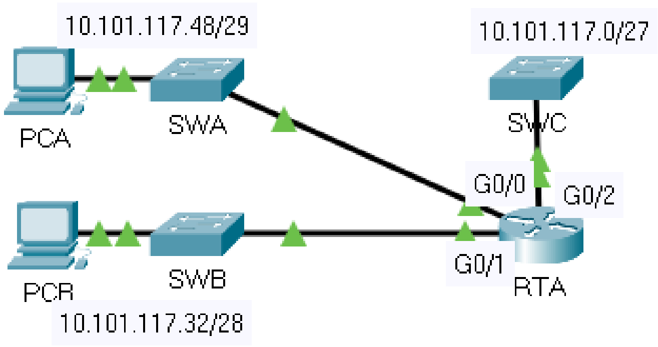
R1(config-if)# **ip access-group HTTP_ONLY in**

Step 3: Verify the ACL implementation.

a. Ping from **PC2** to **Server**. The ping should be successful, if the ping is unsuccessful, verify the IP addresses before continuing.

b. FTP from **PC2** to **Server**. The connection should fail.

c. Open the web browser on **PC2** and enter the IP address of **Server** as the URL. The connection should be successful.



Packet Tracer - Configuring Extended ACLs - Scenario 2

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
RTA	G0/0	10.101.117.49	255.255.255.248	N/A
	G0/1	10.101.117.33	255.255.255.240	N/A
	G0/2	10.101.117.1	255.255.255.224	N/A
PCA	NIC	10.101.117.51	255.255.255.248	10.101.117.49
PCB	NIC	10.101.117.35	255.255.255.240	10.101.117.33
SWA	VLAN 1	10.101.117.50	255.255.255.248	10.101.117.49
SWB	VLAN 1	10.101.117.34	255.255.255.240	10.101.117.33
SWC	VLAN 1	10.101.117.2	255.255.255.224	10.101.117.1

Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Reflection Questions

Background / Scenario

In this scenario, devices on one LAN are allowed to remotely access devices in another LAN using the SSH protocol. Besides ICMP, all traffic from other networks is denied.

The switches and router have also been pre-configured with the following:

- Enable secret password: **ciscoenpa55**
- Console password: **ciscoconpa55**
- Local username and password: **Admin** / **Adminpa55**

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Configure, apply and verify an ACL to satisfy the following policy:

- SSH traffic from devices on the 10.101.117.32/28 network is allowed to devices on the 10.101.117.0/27 networks.
- ICMP traffic is allowed from any source to any destination.
- All other traffic to 10.101.117.0/27 is blocked.

Step 1: Configure the extended ACL.

a. From the appropriate configuration mode on **RTA**, use the last valid extended access list number to configure the ACL. Use the following steps to construct the first ACL statement:

1) The last extended list number is 199.

2) The protocol is TCP.

3) The source network is 10.101.117.32.

4) The wildcard can be determined by subtracting 255.255.255.240 from 255.255.255.255.

5) The destination network is 10.101.117.0.

6) The wildcard can be determined by subtracting 255.255.255.224 from 255.255.255.255.

7) The protocol is SSH (port 22).

What is the first ACL statement?

b. ICMP is allowed, and a second ACL statement is needed. Use the same access list number to permit all ICMP traffic, regardless of the source or destination address. What is the second ACL statement? (Hint: Use the **any** keywords)

c. All other IP traffic is denied, by default.

Step 2: Apply the extended ACL.

The general rule is to place extended ACLs close to the source. However, because access list 199 affects traffic originating from both networks 10.101.117.48/29 and 10.101.117.32/28, the best placement for this ACL might be on interface Gigabit Ethernet 0/2 in the outbound direction. What is the command to apply ACL 199 to the Gigabit Ethernet 0/2 interface?

Step 3: Verify the extended ACL implementation.

a. Ping from **PCB** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addresses before continuing.

b. SSH from **PCB** to **SWC**. The username is **Admin**, and the password is **Adminpa55**.

```
PC> ssh -l Admin 10.101.117.2
```

c. Exit the SSH session to **SWC**.

d. Ping from **PCA** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addresses before continuing.

e. SSH from **PCA** to **SWC**. The access list causes the router to reject the connection.

f. SSH from **PCA** to **SWB**. The access list is placed on **G0/2** and does not affect this connection. The username is **Admin**, and the password is **Adminpa55**.

g. After logging into **SWB**, do not log out. SSH to **SWC** in privileged EXEC mode.

```
SWB# ssh -l Admin 10.101.117.2
```

Part 2: Reflection Questions

1. How was PCA able to bypass access list 199 and SSH to SWC?

2. What could have been done to prevent PCA from accessing SWC indirectly, while allowing PCB SSH access to SWC?

Suggested Scoring Rubric

Activity Section	Question Location	Possible Points	Earned Points
Part 1: Configure, Apply and Verify an Extended Numbered ACL	Step 1a	4	
	Step 1b	4	
	Step 2	4	
Part 1 Total		**12**	
Part 2: Reflection Questions	Question 1	4	
	Question 2	4	
Part 2 Total		**8**	
Packet Tracer Score		**80**	
Total Score		**100**	

PRACTICALNO -03

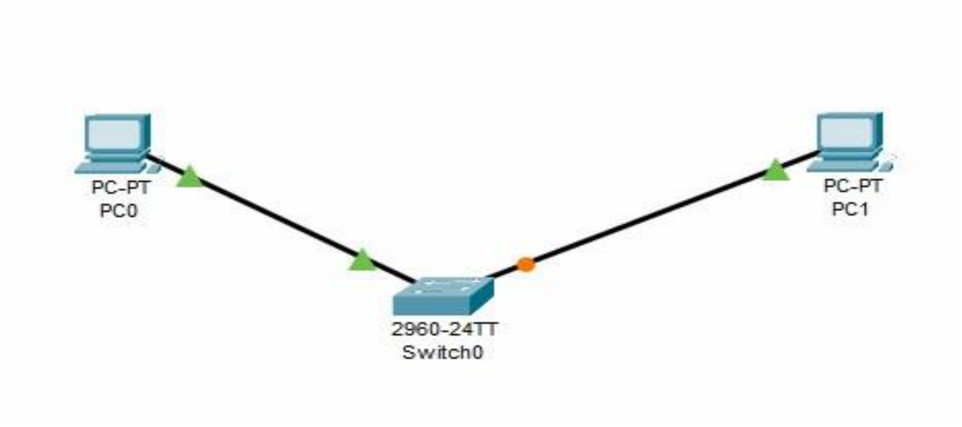
Aim:- **1. Implement SPAN Technologies (Switch Port Analyzer)**
2. Implement SNMP and Syslog
3. Implement Flexible NetFlow

Switched Port Analyzer (SPAN) The SPAN feature, which is sometimes called port mirroring or port monitoring, **selects network traffic for analysis by a network analyzer**. The network analyzer can be a Cisco SwitchProbe device or other Remote Monitoring (RMON) probes.

What is SNMP syslog?

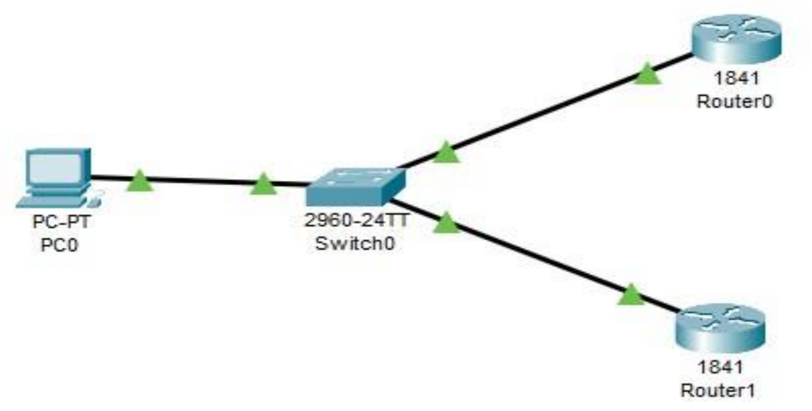
SNMP protocol refers Simple Network Management Protocol. SYSLOG protocol refers to System Logging Protocol. 2. It monitors the network devices over IP networks. It transmits the log messages to Syslog server and monitors it.

Flexible NetFlow **improves on original NetFlow by adding the capability to customize the traffic analysis parameters for your specific requirements**. Flexible NetFlow facilitates the creation of more complex configurations for traffic analysis and data export through the use of reusable configuration components.



```
monitor session 1 source int f0/0
monitor session 1 destination int
f0/2 show monitor session 1 show
monitor detail
```

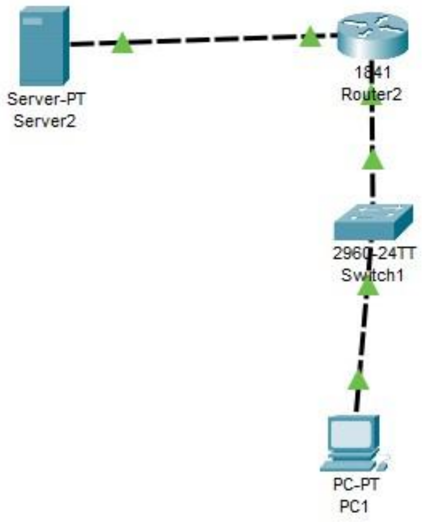
SNMP :



```
snmp-server community read ro
snmp-server community write rw
```

Goto MIB browser and do settings

SYSLOG :



service timestamps log datetime

msec int f0/0.1 check syslog of

server

FLOW SIMULATOR :

int f0/0 ip flow ingress ip flow

egress ip flow-export destination

10.0.0.2 99 ip flow-export source

f0/0 show ip cache flow

PRACTICAL NO -04

Aim:- **1. Implement a GRE Tunnel**
2. Implement VTP
3. Implement NAT

Generic Routing Encapsulation (GRE) is one example of a basic, nonsecure, site-to-site VPN tunneling protocol. GRE is a tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels. GRE creates a virtual point-to-point link to Cisco routers at remote points, over an IP internetwork.

GRE is designed to manage the transportation of multiprotocol and IP multicast traffic between two or more sites that may have only IP connectivity. It can encapsulate multiple protocol packet types inside an IP tunnel.

As shown in Figure 3-22, a tunnel interface supports a header for each of the following:

- *Passenger protocol*: This is the original IPv4 or IPv6 packet that will be encapsulated by the carrier protocol. It could also be a legacy AppleTalk, DECnet, or IPX packet.
- *Carrier protocol*: This is the encapsulation protocol such as GRE that encapsulates the passenger protocol.
- *Transport protocol*: This is the delivery protocol such as IP that carries the carrier protocol.

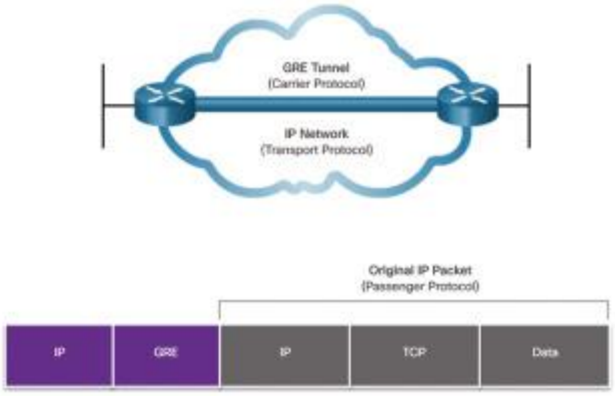


Figure 3-22 Generic Routing Encapsulation

GRE Characteristics (3.4.1.2)

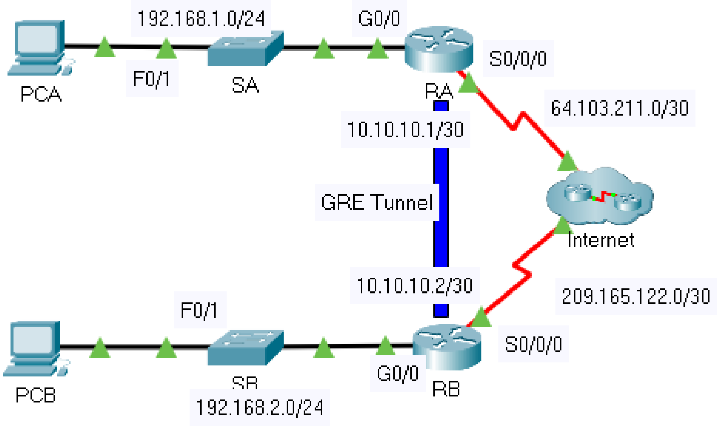
GRE is a tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork. IP tunneling using GRE enables network expansion across a single-protocol backbone environment. It does this by connecting multiprotocol subnetworks in a single-protocol backbone environment.

GRE has these characteristics:

- GRE is defined as an IETF standard (RFC 2784).

- GRE is identified as IP protocol 47 in the Transport protocol IP protocol field.
- GRE encapsulation includes a protocol type field in its header to provide multiprotocol support. Protocol types are defined in RFC 1700 as “EtherTypes.”
- GRE is stateless, which means that, by default, it does not include any flow-control mechanisms.
- GRE does not include any strong security mechanisms to protect its payload.
- The GRE header consumes at least 24 bytes of additional overhead for tunneled packets.
- VTP is a Cisco proprietary protocol is used to exchange VLAN information. This type of protocol was developed to effectively manage the transfer of frames from different VLANs on a single physical line. The full form of VTP is the VLAN Trucking Protocol.
- Using VTP, you can synchronize VLAN information (like VLAN name or VLAN ID) with switches into the same VTP domain.
- For example, let us consider a large size network with 100 switches. Without VTP protocol, if you try to create a VLAN on each Switch, you need to enter VLAN configuration commands on every Switch!
- Trunking protocol VTP allows you to create the VLAN only on a single switch. Similarly, if you want to delete a VLAN, you only require deleting it in one switch. After that, it will automatically circulate to every other switch inside the same VTP domain.

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required. **Network Address Translation (NAT)** is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on a router or firewall.



Packet Tracer - Configure GRE

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
RA	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	64.103.211.2	255.255.255.252	
	Tunnel 0	10.10.10.1	255.255.255.252	
RB	G0/0	192.168.2.1	255.255.255.0	N/A
	S0/0/0	209.165.122.2	255.255.255.252	
	Tunnel 0	10.10.10.2	255.255.255.252	
PCA	NIC	192.168.1.2	255.255.255.0	192.168.1.1
PCB	NIC	192.168.2.2	255.255.255.0	192.168.2.1

Objectives

Part 1: Verify Router Connectivity

Part 2: Configure GRE Tunnels

Part 3: Verify PC Connectivity

Scenario

You are the network administrator for a company which wants to set up a GRE tunnel to a remote office. Both networks are locally configured. You need configure the tunnel and static routes.

Instructions

Part 1: Verify Router Connectivity

Step 1: Ping RA from RB.

a. Use the **show ip interface brief** command on **RA** to determine the IP address of the S0/0/0 port.

b. From **RB** ping the IP S0/0/0 address of **RA.**

Step 2: Ping PCA from PCB.

Attempt to ping the IP address of **PCA** from **PCB**. We will repeat this test after configuring the GRE tunnel. What were the ping results? Explain.

Part 2: Configure GRE Tunnels

Step 1: Configure the Tunnel 0 interface of RA.

a. Enter into the configuration mode for **RA** Tunnel 0.

```
RA(config)# interface tunnel 0
```

b. Set the IP address as indicated in the Addressing Table.

```
RA(config-if)# ip address 10.10.10.1 255.255.255.252
```

c. Set the source and destination for the endpoints of Tunnel 0.

```
RA(config-if)# tunnel source s0/0/0
RA(config-if)# tunnel destination 209.165.122.2
```

d. Configure Tunnel 0 to convey IP traffic over GRE.

```
RA(config-if)# tunnel mode gre ip
```

e. The Tunnel 0 interface should already be active. In the event that it is not, treat it like any other interface.

```
RA(config-if)# no shutdown
```

Step 2: Configure the Tunnel 0 interface of RB.

Repeat Steps 1a – e with **RB**. Be sure to change the IP addressing as appropriate.

Step 3: Configure a route for private IP traffic.

Establish a route between the 192.168.X.X networks using the 10.10.10.0/30 network as the destination.

RA(config)# **ip route 192.168.2.0 255.255.255.0 10.10.10.2**
RB(config)# **ip route 192.168.1.0 255.255.255.0 10.10.10.1**

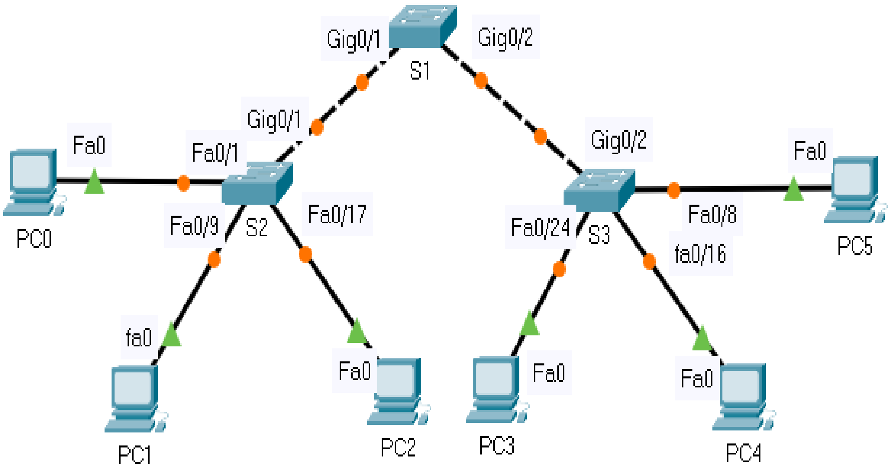
Part 3: Verify Router Connectivity

Step 1: Ping PCA from PCB.

Attempt to ping the IP address of **PCA** from **PCB**. The ping should be successful.

Step 2: Trace the path from PCA to PCB.

Attempt to trace the path from **PCA** to **PCB**. Note the lack of public IP addresses in the output.



Packet Tracer - Configure VTP and DTP

Addressing Table

Device	Interface	IP Address	Subnet Mask
PC0	NIC	192.168.10.1	255.255.255.0
PC1	NIC	192.168.20.1	255.255.255.0
PC2	NIC	192.168.30.1	255.255.255.0
PC3	NIC	192.168.30.2	255.255.255.0
PC4	NIC	192.168.20.2	255.255.255.0
PC5	NIC	192.168.10.2	255.255.255.0
S1	VLAN 99	192.168.99.1	255.255.255.0
S2	VLAN 99	192.168.99.2	255.255.255.0
S3	VLAN 99	192.168.99.3	255.255.255.0

Objectives

Part 1: Configure and Verify DTP

Part 2: Configure and Verify VTP

Background / Scenario

As the number of switches in a network increases, the administration necessary to manage the VLANs and trunks can be challenging. To ease some of the VLAN and trunking configurations, VLAN trunking protocol (VTP) allows a network administration to automate the management of VLANs. Trunk negotiation between network devices is managed by the Dynamic Trunking Protocol (DTP), and is automatically enabled on Catalyst 2960 and Catalyst 3560 switches.

In this activity, you will configure trunk links between the switches. You will configure a VTP server and VTP clients in the same VTP domain. You will also observe the VTP behavior when a switch is in VTP transparent mode. You will assign ports to VLANs and verify end-to-end connectivity with the same VLAN.

Instructions

Part 1: Configure and Verify DTP

In Part 1, you will configure trunk links among the switches, and you will configure VLAN 999 as the native VLAN.

Step 1: Verify VLAN configuration.

Verify the configured VLANs on the switches.

a. On S1, click the **CLI** tab. At the prompt, enter **enable** and enter the **show vlan brief** command to verify the configured VLANs on S1.

b. Repeat step a. on S2 and S3.

What VLANs are configured on the switches?

Step 2: Configure Trunks on S1, S2, and S3.

Dynamic trunking protocol (DTP) manages the trunk links between Cisco switches. Currently all the switchports are in the default trunking mode, which is dynamic auto. In this step, you will change the trunking mode to dynamic desirable for the link between switches S1 and S2. For the link between switches S1 and S3, the link will be set as a static trunk. Use VLAN 999 as the native VLAN in this topology.

a. On switch S1 and switch S2, configure the trunk link to dynamic desirable on the GigabitEthernet 0/1 interface. The configuration of S1 is shown below.

S1(config)# **interface g0/1**

S1(config-if)# **switchport mode dynamic desirable**

b. For the trunk link between S1 and S3, configure a static trunk link on the GigabitEthernet 0/2 interface.

S1(config)# **interface g0/2**

S1(config-if)# **switchport mode trunk**

S3(config)# **interface g0/2**

S3(config-if)# **switchport mode trunk**

c. Verify trunking is enabled on all the switches using the **show interfaces trunk** command.

What is the native VLAN for these trunks currently?

d. Configure VLAN 999 as the native VLAN for the trunk links on S1.

S1(config)# **interface range g0/1 - 2**

S1(config-if-range)# **switchport trunk native vlan 999**

What messages did you receive on S1? How would you correct it?

e. On S2 and S3, configure VLAN 999 as the native VLAN.

f. Verify trunking is successfully configured on all the switches. You should be able ping one switch from another switch in the topology using the IP addresses configured on the SVI.

Part 2: Configure and Verify VTP

S1 will be configured as the VTP server and S2 will be configured as a VTP client. All the switches will be configured to be in the VTP domain **CCNA** and use the VTP password **cisco**.

VLANs can be created on the VTP server and distributed to other switches in the VTP domain. In this part, you will create 3 new VLANs on the VTP server, S1. These VLANs will be distributed to S2 using VTP. Observe how the transparent VTP mode behaves.

Step 1: Configure S1 as VTP server.

Configure S1 as the VTP server in the **CCNA** domain with the password **cisco**.

a. Configure S1 as a VTP server.

```
S1(config)# vtp mode server
Setting device to VTP SERVER mode.
```

b. Configure **CCNA** as the VTP domain name.

```
S1(config)# vtp domain CCNA
Changing VTP domain name from NULL to CCNA
```

c. Configure **cisco** as the VTP password.

```
S1(config)# vtp password cisco
Setting device VLAN database password to cisco
```

Step 2: Verify VTP on S1.

a. Use the **show vtp status** command on the switches to confirm that the VTP mode and domain are configured correctly.

b. To verify the VTP password, use the **show vtp password** command.

Step 3: Add S2 and S3 to the VTP domain.

Before S2 and S3 will accept VTP advertisements from S1, they must belong to the same VTP domain. Configure S2 as a VTP client with **CCNA** as the VTP domain name and **cisco** as the VTP password. Remember that VTP domain names are case sensitive.

a. Configure S2 as a VTP client in the **CCNA** VTP domain with the VTP password **cisco**.

```
S2(config)# vtp mode client
Setting device to VTP CLIENT mode.
S2(config)# vtp domain CCNA
Changing VTP domain name from NULL to CCNA
S2(config)# vtp password cisco
Setting device VLAN database password to cisco
```

b. To verify the VTP password, use the **show vtp password** command.

```
S2# show vtp password
VTP Password: cisco
```

c. Configure S3 to be in the **CCNA** VTP domain with the VTP password **cisco**. Switch S3 will be set in VTP transparent mode.

```
S3(config)# vtp mode transparent
S3(config)# vtp domain CCNA
Changing VTP domain name from NULL to CCNA
S3(config)# vtp password cisco
Setting device VLAN database password to cisco
```

d. Enter **show vtp status** command on all the switches to answer the following question.

Notice that the configuration revision number is 0 on all three switches. Explain.

Step 4: Create more VLANs on S1.

a. On S1, create VLAN 10 and name it Red.

```
S1(config)# vlan 10
S1(config-vlan)# name Red
```

b. Create VLANs 20 and 30 according to the table below.

VLAN Number	VLAN Name
10	Red
20	Blue
30	Yellow

c. Verify the addition of the new VLANs. Enter **show vlan brief** at the privileged EXEC mode.

Which VLANs are configured on S1?

d. Confirm configuration changes using the **show vtp status** command on S1 and S2 to confirm that the VTP mode and domain are configured correctly. Output for S2 is shown here:

How many VLANs are configured on S2? Does S2 have the same VLANs as S1? Explain.

Step 5: Observe VTP transparent mode.

S3 is currently configured as VTP transparent mode.

a. Use **show vtp status** command to answer the following question.

How many VLANs are configured on S3 currently? What is the configuration revision number? Explain your answer.

How would you change the number of VLANs on S3?

b. Change VTP mode to client on S3.

Use show commands to verify the changes on VTP mode. How many VLANs exists on S3 now?

Note: VTP advertisements are flooded throughout the management domain every five minutes, or whenever a change occurs in VLAN configurations. To accelerate this process, you can switch between Realtime mode and Simulation mode until the next round of updates. However, you may have to do this multiple times because this will only forward Packet Tracer's clock by 10 seconds each time. Alternatively, you can change one of the client switches to transparent mode and then back to client mode.

Step 6: Assign VLANs to Ports

Use the **switchport mode access** command to set access mode for the access links. Use the **switchport access vlan** *vlan-id* command to assign a VLAN to an access port.

Ports	Assignments	Network
S2 F0/1 - 8 S3 F0/1 – 8	VLAN 10 (Red)	192.168.10.0 /24
S2 F0/9 - 16 S3 F0/9 – 16	VLAN 20 (Blue)	192.168.20.0 /24
S2 F0/17 - 24 S3 F0/17 – 24	VLAN 30 (Yellow)	192.168.30.0 /24

a. Assign VLANs to ports on S2 using assignments from the table above.

```
S2(config-if)# interface range f0/1 - 8
S2(config-if-range)# switchport mode access
S2(config-if-range)# switchport access vlan 10
S2(config-if-range)# interface range f0/9 -16
S2(config-if-range)# switchport mode access
S2(config-if-range)# switchport access vlan 20
S2(config-if-range)# interface range f0/17 - 24
S2(config-if-range)# switchport mode access
S2(config-if-range)# switchport access vlan 30
```

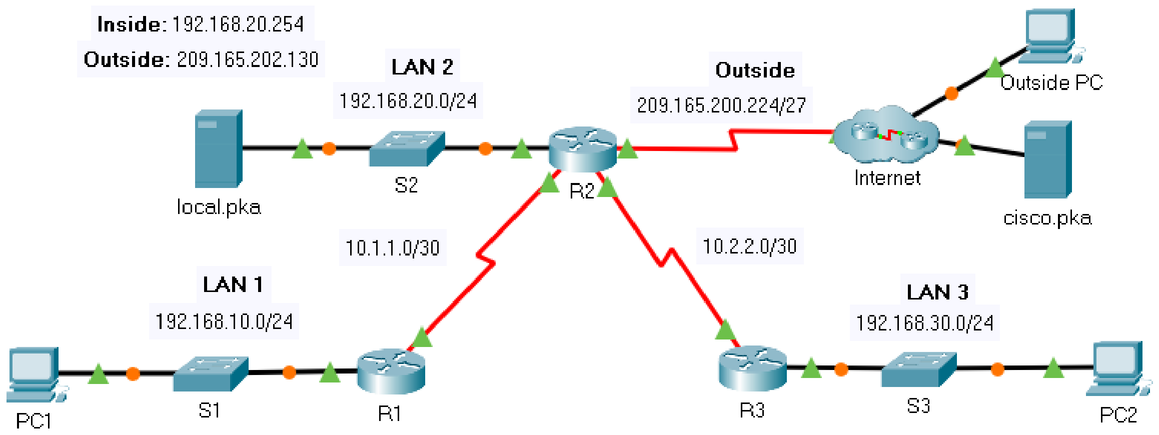
b. Assign VLANs to ports on S3 using assignment from the table above.

Step 7: Verify end to end connectivity.

a. From PC0 ping PC5.

b. From PC1 ping PC4.

c. From PC2 ping PC3.



Packet Tracer - Configure NAT for IPv4

Addressing Table

Device	Interface	IP Address
R1	S0/0/0	10.1.1.1/30
	F0/0	192.168.10.1/24
R2	S0/0/0	10.1.1.2/30
	S0/0/1	10.2.2.1/30
	S0/1/0	209.165.200.225/27
	F0/0/0	192.168.20.1/24
R3	S0/0/1	10.2.2.2/30
	F0/0	192.168.30.1/24
PC1	NIC	192.168.10.10/24
PC2	NIC	192.168.30.10/24
local.pka	NIC	192.168.20.254/24

Device	Interface	IP Address
Outside PC	NIC	209.165.201.14/28
cisco.pka	NIC	209.165.201.30/28

Objectives

- Configure Dynamic NAT with PAT
- Configure Static NAT

Background / Scenario

In this lab, you will configure a router with dynamic NAT with PAT. This will translate addresses from the three internal LANs to a single outside address. In addition, you will configure static NAT to translate an internal server address to an outside address.

Instructions

In this activity you will only configure router R2.

- Use a named ACL to permit the addresses from LAN1, LAN2, and LAN3 to be translated. Specify the LANs in this order. Use the name **R2NAT**. The name you use must match this name exactly.
- Create a NAT pool named **R2POOL**. The pool should use the **first** address from the **209.165.202.128/30** address space. The pool name you use must match this name exactly. All translated addresses must use this address as their outside address.
- Configure NAT with the ACL and NAT pool that you have created.
- Configure static NAT to map the local.pka server inside address to the **second** address from the **209.165.202.128/30** address space.
- Configure the interfaces that will participate in NAT.

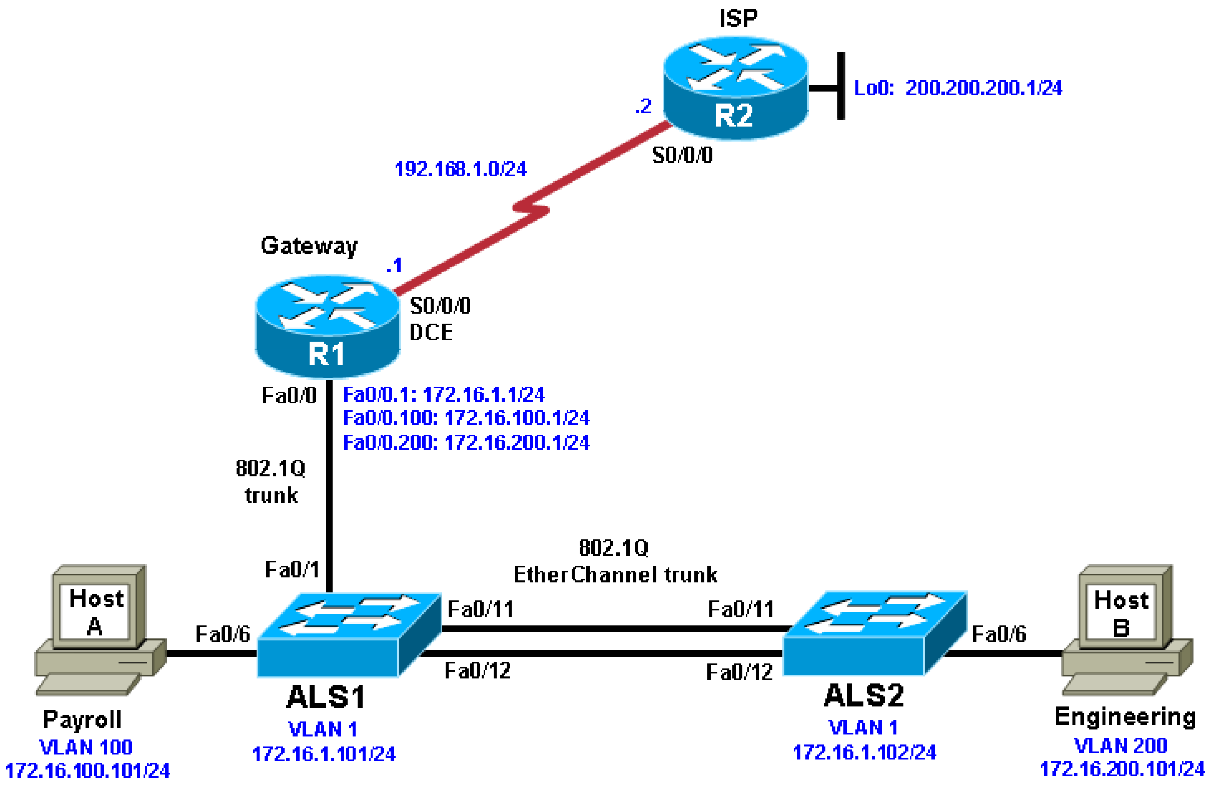
PRACTICAL NO :-5

Aim:- Implement Inter-VLAN Routing

Inter-VLAN routing is **the ability to route, or send, traffic between VLANs that are normally blocked by default**. Switches and VLANs work at the MAC address Layer (Layer 2). Traffic can't be routed between VLANs at Layer 2 based on MAC addresses.

Inter-VLAN Routing with an External Router

Topology



Objective

- Configure inter-VLAN routing using an external router, also known as a router on a stick.

Background

Inter-VLAN routing using an external router can be a cost-effective solution when it is necessary to segment a network into multiple broadcast domains. In this lab, you split an existing network into two separate VLANs on the access layer switches, and use an external router to route between the VLANs. An 802.1Q trunk connects the switch and the Fast Ethernet interface of the router for routing and management. Static routes are used between the gateway router and the ISP router. The switches are connected via an 802.1Q EtherChannel link.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switches are Cisco WS-C2960-24TT-L with the Cisco IOS image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as 2801 or 2811), switches (such as 2950), and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 2 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 2 switches (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Prepare the switches and routers for the lab.

i. Cable the network as shown in the topology diagram. On each switch, erase the startup configuration, delete the vlan.dat file, and reload the switches. Refer to Lab 1-1, “Clearing a Switch” and Lab 1-2, “Clearing a Switch Connected to a Larger Network” to prepare the switches for this lab.

j. Erase the startup configuration and reload the routers.

Step 2: Configure the hosts.

Configure PC hosts A and B with the IP address, subnet mask (/24), and default gateway shown in the topology.

Step 3: Configure the routers.

k. Configure the ISP router for communication with your gateway router. The static route used for the internal networks provides a path for the local network from the ISP. In addition, configure a loopback interface on the ISP router to simulate an external network.

```
Router(config)# hostname ISP
ISP(config)# interface Loopback0
ISP(config-if)# ip address 200.200.200.1 255.255.255.0
ISP(config-if)# interface Serial0/0/0
ISP(config-if)# ip address 192.168.1.2 255.255.255.0
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# ip route 172.16.0.0 255.255.0.0 192.168.1.1
```

l. Configure the Gateway router to communicate with the ISP router. Notice the use of a static default route. The default route tells the router to send any traffic with an unknown destination network to the ISP router.

```
Router(config)# hostname Gateway
```

```
Gateway(config)# interface Serial0/0/0
Gateway(config-if)# ip address 192.168.1.1 255.255.255.0
Gateway(config-if)# clockrate 64000
Gateway(config-if)# no shutdown
Gateway(config-if)# exit
Gateway(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.2
```

m. Verify connectivity from the Gateway router using the **ping** command.

Was this ping successful?

The ping will be successful if the serial connection is set up properly.

Step 4: Configure the switches.

n. Configure the switch hostnames and IP addresses on the management VLAN according to the diagram. By default, VLAN 1 is used as the management VLAN. Create a default gateway on both access layer switches using the **ip default-gateway** *ip_address* command.

The following is a sample configuration for switch ALS1.

```
Switch(config)# hostname ALS1
ALS1(config)# interface vlan 1
ALS1(config-if)# ip address 172.16.1.101 255.255.255.0
ALS1(config-if)# no shutdown
ALS1(config-if)# exit
ALS1(config)# ip default-gateway 172.16.1.1
```

The following is a sample configuration for switch ALS2.

```
Switch(config)# hostname ALS2
ALS2(config)# interface vlan 1
ALS2(config-if)# ip address 172.16.1.102 255.255.255.0
ALS2(config-if)# no shutdown
ALS2(config-if)# exit
ALS2(config)# ip default-gateway 172.16.1.1
```

o. (Optional) Set an enable secret password and configure the vty lines for Telnet access to the switch.

```
ALS1(config)# enable secret cisco
ALS1(config)# line vty 0 15
ALS1(config-line)# password cisco
ALS1(config-line)# login
ALS1(config-line)# end

ALS2(config)# enable secret cisco
ALS2(config)# line vty 0 15
ALS2(config-line)# password cisco
ALS2(config-line)# login
ALS2(config-line)# end
```

p. By default, how many lines are available for Telnet on the access switches?

Sixteen lines are available by default.

Step 5: Confirm the VLANs.

a. Verify that the only existing VLANs are the built-in VLANs. Issue the **show vlan** command from privileged mode on both access layer switches.

ALS1# **show vlan**

```
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gi0/1, Gi0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1
Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Which VLAN is the default management VLAN for Ethernet? What types of traffic are carried on this VLAN?

____

The default management VLAN is VLAN 1. Management traffic usually includes traffic for managing the switches, including VTP frames. VLAN 1 is also the default native VLAN, so untagged frames are assigned to this VLAN by default on 802.1q trunk links.

Step 6: Configure trunk links and EtherChannel on switches.

b. Use the Fast Ethernet 0/11 and 0/12 ports of ALS1 and ALS2 to create an EtherChannel trunk between the switches.

ALS1# **configure terminal**

```
Enter configuration commands, one per line.  End with CNTL/Z.
ALS1(config)# interface range fastEthernet 0/11 - 12
ALS1(config-if-range)# switchport mode trunk
ALS1(config-if-range)# channel-group 1 mode desirable
ALS1(config-if-range)# end

ALS2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ALS2(config)# interface range fastEthernet 0/11 - 12
ALS2(config-if-range)# switchport mode trunk
ALS2(config-if-range)# channel-group 1 mode desirable
ALS2(config-if-range)# end
```

c. Verify the EtherChannel configuration using the **show etherchannel** command.

```
ALS1# show etherchannel 1 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
------+-------------+-----------+-----------------------------------------
1      Po1(SU)        PAgP      Fa0/11(P)   Fa0/12(P)
```

Step 7: Configure VTP.

d. Set up the VTP domain for the access layer switches in global configuration mode. The default VTP mode is server for both switches. Configure ALS2 as a VTP client, and leave ALS1 as a server. Configure the VTP domain name and version on VTP server ALS1.

```
ALS2(config)# vtp mode client
Setting device to VTP CLIENT mode.

ALS1(config)# vtp domain SWLAB
Changing VTP domain name from NULL to SWLAB
%SW_VLAN-6-VTP_DOMAIN_NAME_CHG:  VTP domain name changed to
SWLAB.

ALS1(config)# vtp version 2
```

e. Use the **show vtp status** command to verify the ALS1 VTP configuration and that client ALS2 has learned the new VTP domain information from ALS1.

```
ALS1# show vtp status
```

```
VTP Version                   : running VTP2
Configuration Revision        : 1
Maximum VLANs supported locally : 255
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                    : 0x6A 0x1A 0x90 0xA3 0x10 0xCE 0x86 0xFA
Configuration last modified by 172.16.1.101 at 2-28-10 00:36:24
Local updater ID is 172.16.1.101 on interface Vl1 (lowest numbered VLAN interface
found)
```

```
ALS2# show vtp status
VTP Version                   : running VTP2
Configuration Revision        : 1
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                   : Enabled
VTP Traps Generation          : Disabled
MD5 digest                    : 0x6A 0x1A 0x90 0xA3 0x10 0xCE 0x86 0xFA
Configuration last modified by 172.16.1.101 at 2-28-10 00:36:24
```

Step 8: Configure VLANs and switch access ports.

f. Configure the VLAN 100 named Payroll and VLAN 200 named Engineering on VTP server ALS1.

```
ALS1(config)# vlan 100
ALS1(config-vlan)# name Payroll
ALS1(config-vlan)# vlan 200
ALS1(config-vlan)# name Engineering
```

g. Use the **show vlan brief** command on ALS2 to verify that ALS2 has learned the new VLANs from ALS1.

```
ALS2# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- ----------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/13, 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
100  Payroll                          active
200  Engineering                      active
```

```
1002 fddi-default                 act/unsup
1003 trcrf-default                act/unsup
1004 fddinet-default              act/unsup
1005 trbrf-default                act/unsup
```

h. Configure the switch access ports for the hosts according to the diagram. Statically set the switch port mode to access, and use Spanning Tree PortFast on the interfaces. Assign the host attached to ALS1 Fast Ethernet 0/6 to VLAN 100, and the host attached to ALS2 Fast Ethernet 0/6 to VLAN 200.

```
ALS1(config)# interface fastEthernet 0/6
ALS1(config-if)# switchport mode access
ALS1(config-if)# switchport access vlan 100
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.

ALS2(config)# interface fastEthernet 0/6
ALS2(config-if)# switchport mode access
ALS2(config-if)# switchport access vlan 200
ALS2(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.
```

i. Use the **show vlan brief command** to verify that Fa0/6 is in VLAN 100 on ALS1 and in VLAN 200 on ALS2.

```
ALS1# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- ----------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, 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
100  Payroll                          active    Fa0/6
200  Engineering                      active
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
```

```
1005 trbrf-default              act/unsup

ALS2# show vlan brief

VLAN Name                          Status    Ports
---- ------------------------------ --------- ---------------------------
1    default                       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                             Fa0/5, Fa0/7, Fa0/8, Fa0/9
                                             Fa0/10, 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
100  Payroll                       active
200  Engineering                   active    Fa0/6
1002 fddi-default                  act/unsup
1003 trcrf-default                 act/unsup
1004 fddinet-default               act/unsup
1005 trbrf-default                 act/unsup
```

Step 9: Configure ALS1 trunking to the Gateway router.

Configure switch ALS1 interface Fast Ethernet 0/1 for trunking with the Gateway router Fast Ethernet interface, according to the topology diagram.

```
ALS1(config)# interface fastEthernet 0/1
ALS1(config-if)# switchport mode trunk
ALS1(config-if)# end
```

Note: Optionally, you can apply the **spanning-tree portfast trunk** command to interface Fa0/1 of switch ALS1. This allows the link to the router to rapidly transition to the forwarding state despite being a trunk.

Step 10: Configure the Gateway router Fast Ethernet interface for VLAN trunking.

The native VLAN cannot be configured on a subinterface for Cisco IOS releases earlier than 12.1(3)T. The native VLAN IP address must be configured on the physical interface. Other VLAN traffic is configured on subinterfaces. Cisco IOS release 12.1(3)T and later support native VLAN configuration on a subinterface with the **encapsulation dot1q native** command. If a subinterface is configured using the **encapsulation dot1q native** command, the configuration on the physical interface is ignored. This technique is used in the lab configuration.

j. Create a subinterface for each VLAN. Enable each subinterface with the proper trunking protocol, and configure it for a particular VLAN with the **encapsulation** command. Assign an IP address to each subinterface, which hosts on the VLAN can use as their default gateway.

The following is a sample configuration for the Fast Ethernet 0/0 interface.

```
Gateway(config)# interface fastEthernet 0/0
Gateway(config-if)# no shut
```

The following is a sample configuration for the VLAN 1 subinterface.

```
Gateway(config)# interface fastEthernet 0/0.1
Gateway(config-subif)# description Management VLAN 1
Gateway(config-subif)# encapsulation dot1q 1 native
Gateway(config-subif)# ip address 172.16.1.1 255.255.255.0
```

Note: For enhanced switch security, it is considered best practice to use independent unused VLANs for native and management VLANs.

The following is a sample configuration for the VLAN 100 subinterface.

```
Gateway(config-subif)# interface fastEthernet 0/0.100
Gateway(config-subif)# description Payroll VLAN 100
Gateway(config-subif)# encapsulation dot1q 100
Gateway(config-subif)# ip address 172.16.100.1 255.255.255.0
```

The following is a sample configuration for the VLAN 200 subinterface.

```
Gateway(config-subif)# interface fastEthernet 0/0.200
Gateway(config-subif)# description Engineering VLAN 200
Gateway(config-subif)# encapsulation dot1q 200
Gateway(config-subif)# ip address 172.16.200.1 255.255.255.0
Gateway(config-subif)# end
```

k. Use the **show ip interface brief** command to verify the interface configuration and status.

```
Gateway# show ip interface brief
Interface           IP-Address      OK? Method Status               Protocol
FastEthernet0/0     unassigned      YES unset  up                   up
FastEthernet0/1.1   172.16.1.1      YES manual up                   up
FastEthernet0/1.100 172.16.100.1    YES manual up                   up
FastEthernet0/1.200 172.16.200.1    YES manual up                   up
FastEthernet0/1     unassigned      YES unset  administratively down down
Serial0/0/0         192.168.1.1     YES manual up                   up
Serial0/0/1         unassigned      YES unset  administratively down down
```

l. Use the **show interfaces description** command to verify the interface status and description assigned.

```
Gateway# show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          up             up
Fa0/0.1                        up             up       Management VLAN 1
Fa0/0.100                      up             up       Payroll VLAN 100
Fa0/0.200                      up             up       Engineering VLAN 200
Fa0/1                          admin down     down
Se0/0/0                        up             up
Se0/0/1                        admin down     down
```

m. Use the **show vlans** command on the Gateway router.

```
Gateway# show vlans

Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   FastEthernet0/1.1
```

```
 This is configured as native Vlan for the following interface(s) :
FastEthernet0/1

  Protocols Configured:   Address:           Received:       Transmitted:
       IP            172.16.1.1              198              54
     Other                                 0              29

  277 packets, 91551 bytes input
  83 packets, 15446 bytes output

Virtual LAN ID:  100 (IEEE 802.1Q Encapsulation)

  vLAN Trunk Interface:   FastEthernet0/1.100

  Protocols Configured:   Address:           Received:       Transmitted:
       IP            172.16.100.1                1              25

  0 packets, 0 bytes input
  25 packets, 2350 bytes output

Virtual LAN ID:  200 (IEEE 802.1Q Encapsulation)

  vLAN Trunk Interface:   FastEthernet0/1.200

  Protocols Configured:   Address:           Received:       Transmitted:
       IP            172.16.200.1                1              25

  0 packets, 0 bytes input
  25 packets, 2350 bytes output
```

n. Use the **show cdp neighbor detail** command on the Gateway router to verify that ALS1 is a neighbor. Telnet to the IP address given in the CDP information.

Gateway# **show cdp neighbor detail**

```
-------------------------
Device ID: ISP
Entry address(es):
  IP address: 192.168.1.2
Platform: Cisco 1841,  Capabilities: Router Switch IGMP
Interface: Serial0/0/0,  Port ID (outgoing port): Serial0/0/0
Holdtime : 174 sec

Version :
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M),  Version
12.4(24)T1,
 RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Fri 19-Jun-09 13:56 by prod_rel_team
advertisement version: 2
VTP Management Domain: ''
```

```
-------------------------
Device ID: ALS1
Entry address(es):
  IP address: 172.16.1.101
Platform: cisco WS-C2960-24TT-L,  Capabilities: Switch IGMP
Interface: FastEthernet0/0.1,  Port ID (outgoing port): FastEthernet0/1
Holdtime : 118 sec

Version :
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version
12.2(46)SE, RELE
ASE SOFTWARE (fc2)
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Thu 21-Aug-08 15:59 by nachen

advertisement version: 2
Protocol Hello:  OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=0000000
0FFFFFFFF010221FF000000000000001D46350C80FF0000
VTP Management Domain: 'SWLAB'
Native VLAN: 1
Duplex: full
```

Was the Telnet successful?

Yes, because the subnet given by CDP was a directly connected subnet to the router.

Step 11: Verify inter-VLAN routing on the Gateway router and the host devices.

o. Ping to the 200.200.200.1 ISP loopback interface from either host. Was this ping successful?

Yes, if the hosts are set up with the correct IP addresses and default gateways. This is because the static routes on Gateway and ISP make it possible for pings to make it to ISP and then back to the hosts.

p. Ping from Host A to Host B. Was this ping successful?

Yes, if the hosts are set up with the correct IP addresses and default gateways, because Gateway routes between the VLANs

q. Telnet to the ALS2 VLAN 1 management IP address from the Engineering host. Was this Telnet successful?

Yes, if the hosts are set up with the correct IP addresses and default gateways, because Gateway routes between the VLANs.

If any of the tests failed, make the necessary corrections to the configurations for the router and switches.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configurations (Instructor version)

Router ISP

```
hostname ISP
!
interface Loopback0
 ip address 200.200.200.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.2 255.255.255.0
 no shutdown
!
ip route 172.16.0.0 255.255.0.0 192.168.1.1
!
end
```

Router Gateway

```
hostname Gateway
!
interface FastEthernet0/0
 no shutdown
!
interface FastEthernet0/0.1
 description Management VLAN 1
 encapsulation dot1Q 1 native
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0.100
 description Payroll VLAN 100
 encapsulation dot1Q 100
 ip address 172.16.100.1 255.255.255.0
!
interface FastEthernet0/0.200
 description Engineering VLAN 200
 encapsulation dot1Q 200
 ip address 172.16.200.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.1 255.255.255.0
 clockrate 64000
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.1.2
!
end
```

Note: VLAN and VTP commands do not display in the running configuration when the switch is in client or server mode. It is only displayed in transparent mode.

Switch ALS1

```
hostname ALS1
!
enable secret cisco
!
interface Port-channel1
 switchport mode trunk
!
interface FastEthernet0/1
switchport mode trunk
!
interface FastEthernet0/6
 switchport access vlan 100
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/11
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/12
 switchport mode trunk
 channel-group 1 mode desirable
!
interface Vlan1
 ip address 172.16.1.101 255.255.255.0
 no shutdown
!
ip default-gateway 172.16.1.1
!
line vty 0 4
 password cisco
 login
line vty 5 15
 password cisco
 login
!
end
```

Switch ALS2

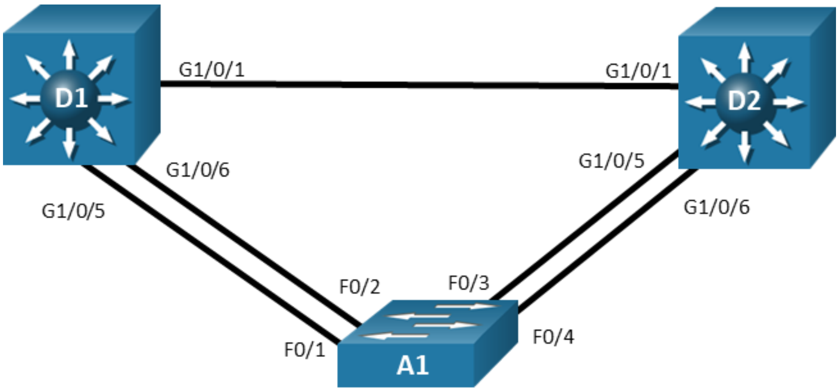
```
hostname ALS2
!
enable secret cisco
!
interface Port-channel1
 switchport mode trunk
!
interface FastEthernet0/6
```

```
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/11
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/12
 switchport mode trunk
 channel-group 1 mode desirable
!
interface Vlan1
 ip address 172.16.1.102 255.255.255.0
 no shutdown
!
ip default-gateway 172.16.1.1
!
line vty 0 4
 password cisco
 login
line vty 5 15
 password cisco
 login
!
end
```

PRACTICAL NO -06

Aim:- Observe STP Topology Changes and Implement RSTP
1. Implement Advanced STP Modifications and Mechanisms
2. Implement MST

Observe STP Topology Changes and Implement RSTP



Addressing Table

Device	Interface	IPv4 Address
D1	VLAN1	10.0.0.1/8
D2	VLAN1	10.0.0.2/8
A1	VLAN1	10.0.0.3/8

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Observe STP Convergence and Topology Change

Part 3: Configure and Verify Rapid Spanning Tree

Background / Scenario

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 the operation of spanning tree protocols to protect the Layer 2 network from loops and topology disruptions. The terms "switch" and "bridge" will be used interchangeably throughout the lab.

Note: This lab is an exercise in deploying and verifying various STP mechanisms. It does not reflect networking best practices.

Note: The switches used with CCNP hands-on labs are Cisco 3650 with Cisco IOS XE release 16.9.4 (universalk9 image) and Cisco 2960+ with IOS release 15.2 (lanbase image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 2 Switches (Cisco 3650 with Cisco IOS XE release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 2960+ with Cisco IOS release 15.2 lanbase image or comparable)
- 1 PC (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each switch.

a. Console into each switch, enter global configuration mode, and apply the basic settings and interface addressing. The startup configuration is provided below for each switch in the topology.

Switch D1

```
hostname D1
spanning-tree mode pvst
banner motd # D1, STP Topology Change and RSTP Lab #
```

```
line con 0
exec-timeout 0 0
logging synchronous
exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface range g1/0/1, g1/0/5-6
 switchport mode trunk
 no shutdown
 exit
vlan 2
 name SecondVLAN
 exit
interface vlan 1
 ip address 10.0.0.1 255.0.0.0
 no shut
 exit
```

Switch D2

```
hostname D2
banner motd # D2, STP Topology Change and RSTP Lab #
spanning-tree mode pvst
line con 0
exec-timeout 0 0
logging synchronous
exit
interface range g1/0/1-24, g1/1/1-4, g0/0
shutdown
exit
interface range g1/0/1, g1/0/5-6
switchport mode trunk
no shutdown
exit
vlan 2
 name SecondVLAN
 exit
interface vlan 1
ip address 10.0.0.2 255.0.0.0
no shut
exit
```

Switch A1

```
hostname A1
banner motd # A1, STP Topology Change and RSTP Lab #
spanning-tree mode pvst
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range f0/1-24, g0/1-2
 shutdown
 exit
interface range f0/1-4
 switchport mode trunk
 no shutdown
 exit
vlan 2
 name SecondVLAN
 exit
interface vlan 1
ip address 10.0.0.3 255.0.0.0
no shut
exit
```

b. Set the clock on each switch to UTC time.

c. Save the running configuration to startup-config.

Note: Outputs and Spanning Tree topologies highlighted in this lab may be different than what you observe using your own equipment. It is critically important for you to understand how Spanning Tree makes its decisions, and how those decisions impact the operational topology of the network.

Part 2: Discover the Default Spanning Tree

Your switches have been configured and interfaces have been enabled, and the Spanning Tree Protocol, operational by default, has already converged onto a loop-free logical network. In this part of the lab, we will discover what that default spanning tree looks like and evaluate why it converged the way it did. We will do this by following the same set of steps that Spanning Tree does. We will find the Root Bridge, then find the Root Ports, and lastly see which ports are **Designated** ports, and which ports are **non-Designated** ports in our topology.

Step 1: Find the root bridge.

Our switches are running the Cisco default PVST+, and we have two VLANs in the network, so we should see two root bridges.

a. On A1, issue the command **show spanning-tree root** and observe what the output tells you about the root bridge. Amongst the lab devices being used to document this lab, A1 shows the root id with a cost of 19 and the root port as interface FastEthernet 0/1 for both VLAN1 and VLAN2.

A1# **show spanning-tree root**

```
                                    Root    Hello Max Fwd
Vlan                Root ID          Cost   Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 d8b1.9028.af80         19     2   20  15  Fa0/1
VLAN0002         32770 d8b1.9028.af80         19     2   20  15  Fa0/1
```

Because we know from the physical topology diagram that A1 is connected to D1 using F0/1, and that interface is a FastEthernet interface, therefore having a cost of 19, D1 is the root bridge for both VLAN 1 and VLAN 2. The question at this point is – why?

b. The root bridge is elected based upon which switch has the highest Bridge ID (BID). The BID is made up of a configurable priority value (which defaults to 32768) and the base MAC address for the switch. Use the command **show spanning-tree root** to gather that information from your switches to support the root bridge decision.

D1# **show spanning-tree root**

```
                                    Root    Hello Max Fwd
Vlan                Root ID          Cost   Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 d8b1.9028.af80          0    2   20  15
VLAN0002         32770 d8b1.9028.af80          0    2   20  15
```

D2# **show spanning-tree root**

```
                                    Root    Hello Max Fwd
Vlan                Root ID          Cost   Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 d8b1.9028.af80          4    2   20  15  Gi1/0/1
VLAN0002         32770 d8b1.9028.af80          4    2   20  15  Gi1/0/1
```

A1# **show spanning-tree root**

```
                                    Root    Hello Max Fwd
Vlan                Root ID          Cost   Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 d8b1.9028.af80         19     2   20  15  Fa0/1
VLAN0002         32770 d8b1.9028.af80         19     2   20  15  Fa0/1
```

The first thing to look at is the priority value. It is 32768 by default. Because we are working with PVST+, a differentiator is added – the priority value is modified with the extended system ID, which is equal to the VLAN number. You can see in the output here that our three devices are using default priorities – 32769 for VLAN 1 (32768 + 1) and 32770 for VLAN 2 (32768 + 2). For each VLAN, the priority values are the same for each of the three switches. When this happens, the rest of the BID is taken into account. The rest of the BID includes the base MAC address. The **lowest** base MAC address is used to break the tie.

c. What are the base MAC addresses for the devices we are using? Issue the command **show version | include MAC** (capitalized exactly like that) on each switch.

```
D1# show version | include MAC
Base Ethernet MAC Address          : d8:b1:90:28:af:80

D2# show version | include MAC
Base Ethernet MAC Address          : d8:b1:90:5d:c3:00
D2#

A1# show version | include MAC
Base ethernet MAC Address        : F0:78:16:47:45:80
```

Amongst the three switches being used to document this lab, D1 has the lowest base MAC address. The OUI portion of each MAC address is the same. The first set of hexadecimal characters are different; 0x28 is a lower number than 0x5d. This is what has caused D1 to be elected as the root bridge.

Step 2: Find the Root Port for each switch.

Each switch will have one single root port. This port represents the lowest path cost to the root bridge. Path Cost is the total of the Port Costs in the path to the root bridge. The Port Cost is based upon the bandwidth value of the port, and it can either be dynamically assigned or statically configured.

a. As we saw in the previous output of **show spanning-tree root** on each switch, the Path Cost can be different amongst switches. In this case, the path cost from A1 to D1 is 19, reflecting connectivity via a FastEthernet port, while the path cost from D2 to D1 is 4, reflecting connectivity via a GigabitEthernet port.

```
D1# show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001          32769 d8b1.9028.af80        0    2   20  15
VLAN0002          32770 d8b1.9028.af80        0    2   20  15

D2# show spanning-tree root
```

```
                          Root    Hello Max Fwd
Vlan             Root ID        Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 d8b1.9028.af80        4     2   20  15  Gi1/0/1
VLAN0002         32770 d8b1.9028.af80        4     2   20  15  Gi1/0/1
```

A1# **show spanning-tree root**

```
                          Root    Hello Max Fwd
Vlan             Root ID        Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 d8b1.9028.af80       19     2   20  15  Fa0/1
VLAN0002         32770 d8b1.9028.af80       19     2   20  15  Fa0/1
```

b. These are direct connections to the root, so port cost and path cost are the same. This can be seen in the output of **show spanning-tree**.

A1# **show spanning-tree**

```
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     d8b1.9028.af80
             Cost        19
             Port        1 (FastEthernet0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     f078.1647.4580
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Root FWD 19        128.1    P2p
Fa0/2               Altn BLK 19        128.2    P2p
<some output omitted>
```

c. Our topology does not really illustrate the difference between port cost and path cost very well, so we will introduce a change in the network to achieve this. At D1, shutdown the g1/0/1 interface. The result of this is that D2 will have to change the port it considers root, and we will then see the difference between port cost and path cost.

```
D1(config)# interface g1/0/1
D1(config-if)# shutdown
```

d. On D2, issue the command **show spanning-tree** and you will see the port cost and path cost values separating themselves.

```
D2# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     d8b1.9028.af80
             Cost        38
             Port        5 (GigabitEthernet1/0/5)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     d8b1.905d.c300
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/5             Root FWD 19        128.5    P2p
Gi1/0/6             Altn BLK 19        128.6    P2p
```

The root path cost is now 38, while the root port cost is 19. For D2 to reach the root bridge D1, it must traverse two FastEthernet links, and 19 times 2 is 38.

Step 3: Identify Designated Ports.

The Spanning Tree Designated Port can be traced back to the early versions of the protocol, which were developed when LAN segments were shared, multiaccess networks. In these networks, there was a very real possibility that there could be users attached to a segment between two switches.

The job of the Designated Port back then was to ensure that users had a way to access the network from a given segment, and there was always one Designated Port on each segment. In the switched networks of today, there are very few shared segments, so the job of the Designated Port is more to help maintain the network topology.

A Designated Port stays active in the topology, both sending BPDUs and learning MAC addresses. Every port on the Root Bridge is a Designated Port. Further, there is one Designated Port on every segment that is not attached directly to the root.

a. If you have not already done so, issue the **no shutdown** command for D1 interface g1/0/1. This will restore our full topology and allow for the non-root attached segment to exist (the links between A1 and D2).

b. On D2, issue the **show spanning-tree** command, and you will see that there are two ports now identified as being in the Designated Port role.

D2# **show spanning-tree**

```
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     d8b1.9028.af80
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     d8b1.905d.c300
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/5             Desg FWD 19        128.5    P2p
Gi1/0/6             Desg FWD 19        128.6    P2p
```

c. And now look at the segments from the A1 side. Issue the **show spanning-tree** command on A1.

A1# **show spanning-tree**

```
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     d8b1.9028.af80
             Cost        19
             Port        1 (FastEthernet0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     f078.1647.4580
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
```

```
Fa0/1          Root FWD 19      128.1    P2p
Fa0/2          Altn BLK 19      128.2    P2p
Fa0/3          Altn BLK 19      128.3    P2p
Fa0/4          Altn BLK 19      128.4    P2p
```

Interfaces F0/3 and F0/4 on A1 are in the Alternate Role, which is the Cisco PVST+ version of the IEEE 802.1D Discarding role. These interfaces are up and receiving BPDUs from the Designated Ports on each segment, but they will not learn MAC addresses or forward traffic until they stop receiving those BDPUs and move to the Designated state.

Why is D2 controlling the Designated Port role on these two segments? Because from the middle of the segment, D2 has a lower cost to the root bridge than does A1. The root cost on D2 is 4, while the root cost on A1 is 19. Therefore, it takes and maintains the Designated Ports for these two segments.

d. You may have noticed in the previous output that the two links from A1 to D1 were not being used.

```
Fa0/1          Root FWD 19      128.1    P2p
Fa0/2          Altn BLK 19      128.2    P2p
```

Each switch can only have a single root port. In this example, F0/2, which is in the Alternate Role, would only take over if F0/1 were to fail. The decision about which interface to use in this scenario is based on the lowest port priority, which defaults to 128.*interface_id*.

Part 3: Implement and Observe Rapid Spanning Tree Protocol

In Part 3, you will implement Rapid Spanning Tree Protocol (RSTP) on all the switches. Using the same basic rules, RSTP speeds up convergence significantly.

a. On D2, issue the **debug spanning-tree events** command, and then issue the **shutdown** command for interface g1/0/1 and observe the output.

```
D2# debug spanning-tree events
D2# config t
D2(config)# interface g1/0/1
D2(config-if)# shutdown
D2(config-if)#
*Dec 24 13:07:10.790: %LINK-5-CHANGED:  Interface GigabitEthernet1/0/1,
changed state to administratively down
*Dec 24 13:07:11.790: %LINEPROTO-5-UPDOWN:  Line protocol on Interface
GigabitEthernet1/0/1, changed state to down
D2(config-if)#
*Dec 24 13:07:28.159: STP: VLAN0001 heard root 32769-d8b1.9028.af80 on
Gi1/0/5
*Dec 24 13:07:28.160:     supersedes 32769-d8b1.905d.c300
```

```
*Dec 24 13:07:28.161: STP: VLAN0001 new root is 32769, d8b1.9028.af80 on port
Gi1/0/5, cost 38
*Dec 24 13:07:28.162: STP: VLAN0001 sent Topology Change Notice on Gi1/0/5
*Dec 24 13:07:28.165: STP[1]: Generating TC trap for port GigabitEthernet1/0/6
*Dec 24 13:07:28.166: STP: VLAN0001 Gi1/0/6 -> blocking
*Dec 24 13:07:28.166: STP: VLAN0002 heard root 32770-d8b1.9028.af80 on
Gi1/0/5
*Dec 24 13:07:28.167:     supersedes 32770-d8b1.905d.c300
*Dec 24 13:07:28.167: STP: VLAN0002 new root is 32770, d8b1.9028.af80 on port
Gi1/0/5, cost 38
D2(config-if)#
*Dec 24 13:07:28.169: STP: VLAN0002 sent Topology Change Notice on Gi1/0/5
*Dec 24 13:07:28.171: STP[2]: Generating TC trap for port GigabitEthernet1/0/6
*Dec 24 13:07:28.171: STP: VLAN0002 Gi1/0/6 -> blocking
D2(config-if)#
```

From the above output, you can see that it took a total of about 17 seconds for spanning tree to adjust to the topology change. Rapid Spanning Tree can adjust much faster.

b. On D1, change the spanning tree mode to rapid-pvst:

```
D1(config)# spanning-tree mode rapid-pvst
D1(config)#
*Dec 24 13:14:48.458: %LINK-3-UPDOWN: Interface Vlan1, changed state to down
*Dec 24 13:14:49.459: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Vlan1, changed state to down
D1(config)#
*Dec 24 13:15:18.452: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Dec 24 13:15:19.453: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Vlan1, changed state to up
D1(config)#
```

From the above output, you can see that it took more than 30 seconds for interface VLAN1 to come back up. There are two conditions that must be met for an SVI to be operational. First, the VLAN the SVI is associated with must exist, and second, the VLAN the SVI is associated with must be in spanning tree forwarding mode on at least one interface. It took 30 seconds to adjust. What happened to “rapid”?

c. On D1, issue the command **show spanning-tree**.

```
D1# show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     d8b1.9028.af80
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     d8b1.9028.af80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi1/0/5             Desg FWD 19        128.5    P2p Peer(STP)
Gi1/0/6             Desg FWD 19        128.6    P2p Peer(STP)
```

The type values tell the story. Rapid spanning tree is backwards compatible with common spanning tree. It achieves this backwards compatibility by falling back to using the timers and settings of common spanning tree. In other words, we will not see the benefits of rapid spanning tree if only one switch is running it.

d. On D2 and A1, change the spanning tree mode to rapid spanning tree.

```
A1(config)# spanning-tree mode rapid-pvst
A1(config)#
Dec 24 13:31:51.023: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
Dec 24 13:31:51.081: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
A1(config)#
```

A1 was the last switch that was configured for RSTP. As you can see, interface VLAN1 was only down for 0.048 seconds. This is the “rapid” in rapid spanning tree.

Device Configs - Final

Switch D1

```
D1# show run
Building configuration...

Current configuration : 8871 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
```

```
!
hostname D1
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
login on-success log
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL SAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
```

```
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 switchport mode trunk
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 switchport mode trunk
```

```
!
interface GigabitEthernet1/0/6
 switchport mode trunk
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 shutdown
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
```

```
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 shutdown
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
 shutdown
!
interface GigabitEthernet1/1/2
 shutdown
!
interface GigabitEthernet1/1/3
 shutdown
!
interface GigabitEthernet1/1/4
 shutdown
!
interface Vlan1
 ip address 10.0.0.1 255.0.0.0
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C D1, STP Topology Change and RSTP Lab ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
```

```
 stopbits 1
line vty 0 15
!
end
```

Switch D2

```
D2# show run
Building configuration...

Current configuration : 8881 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D2
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
login on-success log
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
 mode sso
```

```
!
transceiver type all
 monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL SAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
!
```

```
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 switchport mode trunk
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 switchport mode trunk
!
interface GigabitEthernet1/0/6
 switchport mode trunk
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 shutdown
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
```

```
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 shutdown
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
 shutdown
!
interface GigabitEthernet1/1/2
 shutdown
!
interface GigabitEthernet1/1/3
 shutdown
!
```

```
interface GigabitEthernet1/1/4
 shutdown
!
interface Vlan1
 ip address 10.0.0.2 255.0.0.0
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C D2, STP Toplogy Change and RSTP Lab ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 15
!
end
```

Switch A1

```
A1# show run
Building configuration...

Current configuration : 2008 bytes
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname A1
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
system mtu routing 1500
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface FastEthernet0/1
 switchport mode trunk
!
interface FastEthernet0/2
 switchport mode trunk
!
interface FastEthernet0/3
 switchport mode trunk
!
interface FastEthernet0/4
 switchport mode trunk
!
interface FastEthernet0/5
 shutdown
!
interface FastEthernet0/6
 shutdown
!
interface FastEthernet0/7
 shutdown
!
interface FastEthernet0/8
 shutdown
!
interface FastEthernet0/9
 shutdown
!
interface FastEthernet0/10
 shutdown
!
interface FastEthernet0/11
 shutdown
!
interface FastEthernet0/12
 shutdown
!
```

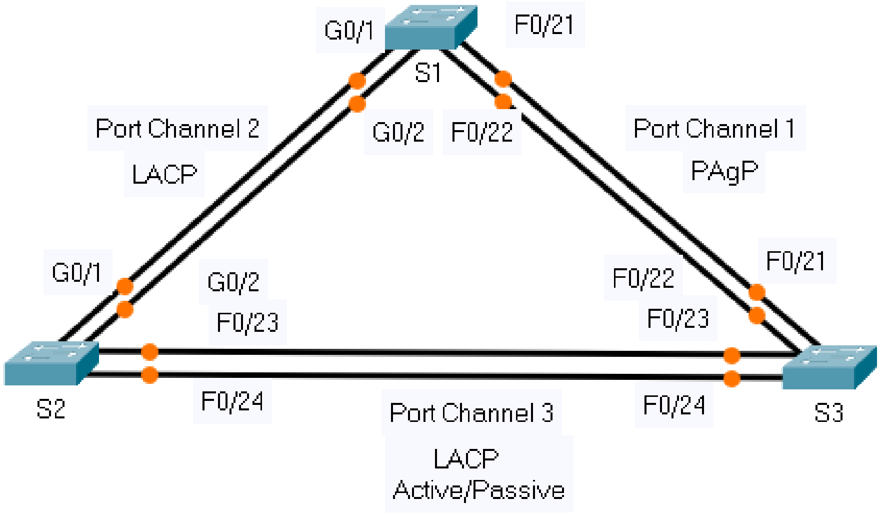
```
interface FastEthernet0/13
 shutdown
!
interface FastEthernet0/14
 shutdown
!
interface FastEthernet0/15
 shutdown
!
interface FastEthernet0/16
 shutdown
!
interface FastEthernet0/17
 shutdown
!
interface FastEthernet0/18
 shutdown
!
interface FastEthernet0/19
 shutdown
!
interface FastEthernet0/20
 shutdown
!
interface FastEthernet0/21
 shutdown
!
interface FastEthernet0/22
 shutdown
!
interface FastEthernet0/23
 shutdown
!
interface FastEthernet0/24
 shutdown
!
interface GigabitEthernet0/1
 shutdown
!
interface GigabitEthernet0/2
 shutdown
!
interface Vlan1
 ip address 10.0.0.3 255.0.0.0
```

```
!
ip http server
ip http secure-server
!
banner motd ^C A1, STP Toplogy Change and RSTP Lab ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
line vty 0 4
 login
line vty 5 15
 login
!
end
```

Practical No -07

Aim:- 1. Implement EtherChannel
2. Tune and Optimize EtherChannel Operations

EtherChannel is a port link aggregation technology in which multiple physical port links are grouped into one logical link. It is used to provide high-speed links and redundancy. A maximum of 8 links can be aggregated to form a single logical link.



Packet Tracer - Configure EtherChannel

Objectives

Part 1: Configure Basic Switch Settings

Part 2: Configure an EtherChannel with Cisco PAgP

Part 3: Configure an 802.3ad LACP EtherChannel

Part 4: Configure a Redundant EtherChannel Link

Background

Three switches have just been installed. There are redundant uplinks between the switches. As configured, only one of these links can 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 will configure Port Aggregation Protocol (PAgP), a Cisco EtherChannel protocol, and Link Aggregation Control Protocol (LACP), an IEEE 802.3ad open standard version of EtherChannel.

Before beginning the configuration, review the EtherChannel Configuration Guidelines and Restrictions listed at the end of this activity.

Port Channel Table

Channel Group	Ports	Protocol
1	S1 F0/21. F0/22 S3 F0/21, F0/22	PAgP
2	S1 G0/1, G0/2 S2 G0/1, G0/2	LACP
3	S2 F0/23, F0/24 S3 F0/23, F0/24	Negotiated LACP

Instructions

Part 1: Configure Basic Switch Settings

a. Assign each switch a hostname according to the topology diagram.

b. Before beginning the link aggregation between switches, verify the existing configuration of the ports that connect the switches to ensure that the ports will successfully join the EtherChannels. Commands that provide information about the state of the switch ports include:

```
S1# show interfaces | include Ethernet
S1# show interface status
S1# show interfaces trunk
```

c. Configure all ports that are required for the EtherChannels as static trunk ports.

Note: If the ports are configured with DTP dynamic auto mode, and you do not set the mode of the ports to trunk, the links do not form trunks and remain access ports. The default mode on a 2960 switch is for DTP to be enabled and set to dynamic auto. DTP can be disabled on interfaces with the **switchport nonegotiate** command.

Part 2: Configure an EtherChannel with Cisco PAgP

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

Step 1: Configure Port Channel 1.

a. The first EtherChannel that is created for this activity aggregates ports F0/21 and F0/22 between **S1** and **S3**. Configure the ports on both switches as static trunk ports.

b. Use the **show interfaces trunk** command to ensure that you have an active trunk link for those two links, and the native VLAN on both links is the same.

```
S1# show interfaces trunk

Port Mode Encapsulation Status Native vlan
F0/21 on 802.1q trunking 1
F0/22 on 802.1q trunking 1
G0/1 on 802.1q trunking 1
G0/2 on 802.1q trunking 1

<output omitted>
```

c. On S1 and S3, add ports F0/21 and F0/22 to Port Channel 1 with the **channel-group 1 mode desirable** command. The **mode desirable** option enables the switch to actively negotiate to form a PAgP link. **Note:** Interfaces must be **shutdown** before adding them to the channel group.

```
S1(config)# interface range f0/21 – 22
S1(config-if-range)# shutdown
S1(config-if-range)# channel-group 1 mode desirable
S1(config-if-range)# no shutdown

S3(config)# interface range f0/21 - 22
S3(config-if-range)# shutdown
S3(config-if-range)# channel-group 1 mode desirable
S3(config-if-range)# no shutdown
```

The message “Creating a port-channel interface Port-channel 1” should appear on both switches when the channel-group is configured. This interface designation will appear as Po1 in command output.

d. Configure the logical interface to become a trunk by first entering the **interface port-channel** *number* command and then the **switchport mode trunk** command. Add this configuration to both switches.

```
S1(config)# interface port-channel 1
S1(config-if)# switchport mode trunk

S3(config)# interface port-channel 1
S3(config-if)# switchport mode trunk
```

Step 2: Verify Port Channel 1 status.

a. Issue the **show etherchannel summary** command on S1 and S3 to verify that EtherChannel is working on both switches. This command displays the type of EtherChannel, the ports utilized, and the port states. Command output is shown for S1.

```
S1# 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
------+-------------+-----------+---------------------------------------
1 Po1(SU) PAgP F0/21(P) F0/22(P)
```

b. If the EtherChannel does not come up, shut down the physical interfaces on both ends of the EtherChannel and then bring them back up again. The **show interfaces trunk** and **show spanning-tree** commands should show the port channel as one logical link.

Part 3: Configure an 802.3ad LACP EtherChannel

Step 1: Configure Port Channel 2.

a. In 2000, the IEEE released 802.3ad, which is an open standard version of EtherChannel. It is commonly referred to as LACP. Using the previous commands, configure the link between **S1** and **S2,** using ports G0/1 and G0/2, as an LACP EtherChannel. You must use a different port channel number on **S1** than 1, because you already used that in the previous step. To configure port channel 2 as LACP, use the interface configuration mode **channel-group** *2* **mode active** command. Active mode indicates that the switch actively tries to negotiate that link as LACP, as opposed to PAgP. The configuration of S1 is shown below.

```
S1(config)# interface range g0/1 - 2
S1(config-if-range)# shutdown
S1(config-if-range)# channel-group 2 mode active
S1(config-if-range)# no shutdown
S1(config-if-range)# interface port-channel 2
S1(config-if)# switchport mode trunk
```

Step 2: Verify Port Channel 2 status.

Use the **show** commands from Part 1 Step 2 to verify the status of Port Channel 2. Look for the protocol used by each port.

Part 4: Configure a Redundant EtherChannel Link

Step 1: Configure Port Channel 3.

There are various options for the **channel-group** *number* **mode** command:

```
S2(config)# interface range f0/23 - 24
S2(config-if-range)# channel-group 3 mode ?
active Enable LACP unconditionally
auto Enable PAgP only if a PAgP device is detected
desirable Enable PAgP unconditionally
on Enable Etherchannel only
passive Enable LACP only if a LACP device is detected
```

a. On switch **S2**, add ports F0/23 and F0/24 to Port Channel 3 with the **channel-group 3 mode passive** command. The **passive** option indicates that you want the switch to use LACP only if another LACP device is detected. Statically configure Port Channel 3 as a trunk interface.

```
S2(config)# interface range f0/23 - 24
S2(config-if-range)# shutdown
S2(config-if-range)# channel-group 3 mode passive
S2(config-if-range)# no shutdown
S2(config-if-range)# interface port-channel 3
S2(config-if)# switchport mode trunk
```

b. On **S3**, add ports F0/23 and F0/24 to Port Channel 3 with the **channel-group 3 mode active** command. The **active** option indicates that you want the switch to use LACP unconditionally. Statically configure Port Channel 3 as a trunk interface.

Step 2: Verify Port Channel 3 status.

a. Use the **show** commands from Part 1 Step 2 to verify the status of Port Channel 3. Look for the protocol used by each port.

b. Creating EtherChannel links does not prevent Spanning Tree from detecting switching loops. View the spanning tree status of the active ports on **S1**.

```
S1# show spanning-tree active
VLAN0001
Spanning tree enabled protocol ieee
Root ID Priority 32769
Address 0001.436E.8494
Cost 9
Port 27(Port-channel1)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
Address 000A.F313.2395
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 20

Interface Role Sts Cost Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po1 Root FWD 9 128.27 Shr
Po2 Altn BLK 3 128.28 Shr
```

Port Channel 2 is not operative because Spanning Tree Protocol placed some ports into blocking mode. Unfortunately, those ports were the Gigabit ports. In this topology, you can restore these ports by configuring **S1** to be **primary** root for VLAN 1. You could also set the priority to **24576**.

S1(config)# **spanning-tree vlan 1 root primary**

or

S1(config)# **spanning-tree vlan 1 priority 24576**

You may have to wait for STP to recalculate the tree topology. Press fast-forward if necessary. Use the **show spanning-tree active** command to verify that the Gigabit ports are now in the forwarding state.

EtherChannel Configuration Guidelines and Restrictions

EtherChannel has some specific guidelines that must be followed in order to avoid configuration problems.

1) All Ethernet interfaces support EtherChannel up to a maximum of eight interfaces with no requirement that the interfaces be on the same interface module.

2) All interfaces within an EtherChannel must operate at the same speed and duplex.

3) EtherChannel links can function as either single VLAN access ports or as trunk links between switches.

4) All interfaces in a Layer 2 EtherChannel must be members of the same VLAN or be configured as trunks.

5) If configured as trunk links, Layer 2 EtherChannel must have the same native VLAN and have the same VLANs allowed on both switches connected to the trunk.

6) When configuring EtherChannel links, all interfaces should be shutdown prior to beginning the EtherChannel configuration. When configuration is complete, the links can be re-enabled.

7) After configuring the EtherChannel, verify that all interfaces are in the up/up state.

8) It is possible to configure an EtherChannel as static, or for it to use either PAgP or LACP to negotiate the EtherChannel connection. The determination of how an EtherChannel is setup is the value of the **channel-group** *number* **mode** command. Valid values are:

    active LACP is enabled unconditionally

passive LACP is enabled only if another LACP-capable device is connected.

desirable PAgP is enabled unconditionally

auto PAgP is enabled only if another PAgP-capable device is connected.

on EtherChannel is enabled, but without either LACP or PAgP.

9) LAN ports can form an EtherChannel using PAgP if the modes are compatible. Compatible PAgP modes are:

desirable => desirable

desirable => auto

If both interfaces are in **auto** mode, an Etherchannel cannot form.

10) LAN ports can form an EtherChannel using LACP if the modes are compatible. Compatible LACP modes are:

active => active

active => passive

If both interfaces are in **passive** mode, an EtherChannel cannot form using LACP.

11) Channel-group numbers are local to the individual switch. Although this activity uses the same Channel-group number on either end of the EtherChannel connection, it is not a requirement. Channel-group 1 (interface po1) on one switch can form an EtherChannel with Channel-group 5 (interface po5) on another switch.

PRACTICAL NO -08

Aim:- OSPF Implementation
1. Implement Single-Area OSPFv2
2. Implement Multi-Area OSPFv2
3. OSPFv2 Route Summarization and Filtering
4. Implement Multiarea OSPFv3

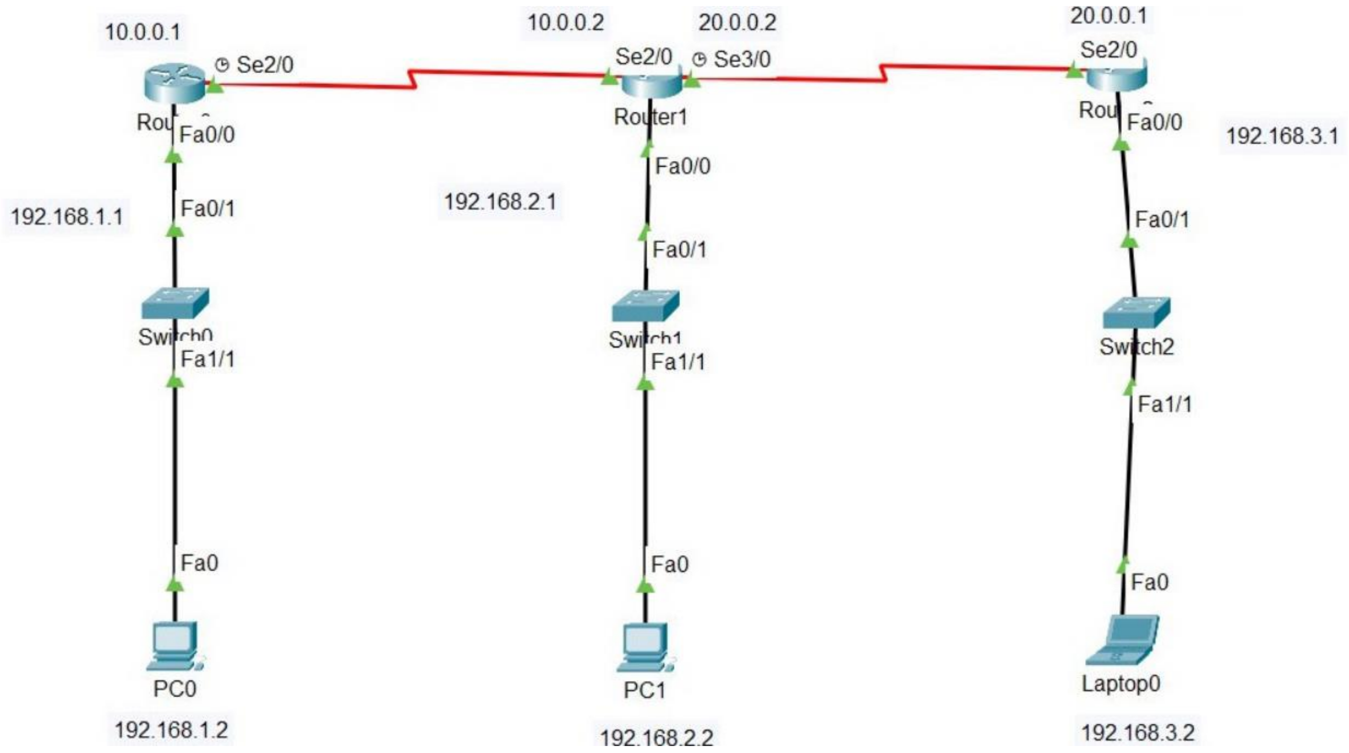
OSPF is an interior gateway routing protocol that uses link-states rather than distance vectors for path selection. OSPF propagates link-state advertisements (LSAs) rather than routing table updates. Because only LSAs are exchanged instead of the entire routing tables, OSPF networks converge in a timely manner.

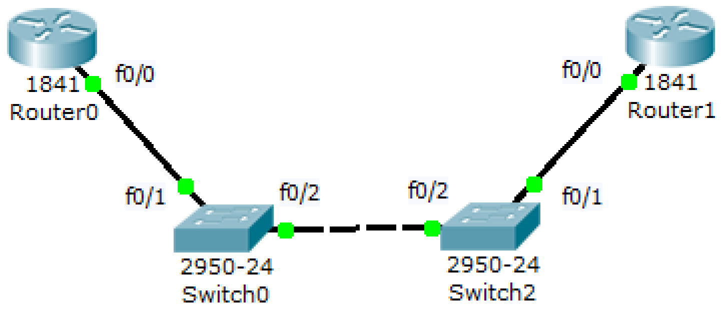
What is OSPFv2 configuration?

Open Shortest Path First Version 2 (OSPFv2) is **a link-state routing protocol that uses link-state advertisements (LSAs) to update neighboring routers about a router's interfaces**. Each router maintains an identical area-topology database to determine the shortest path to any neighboring router.

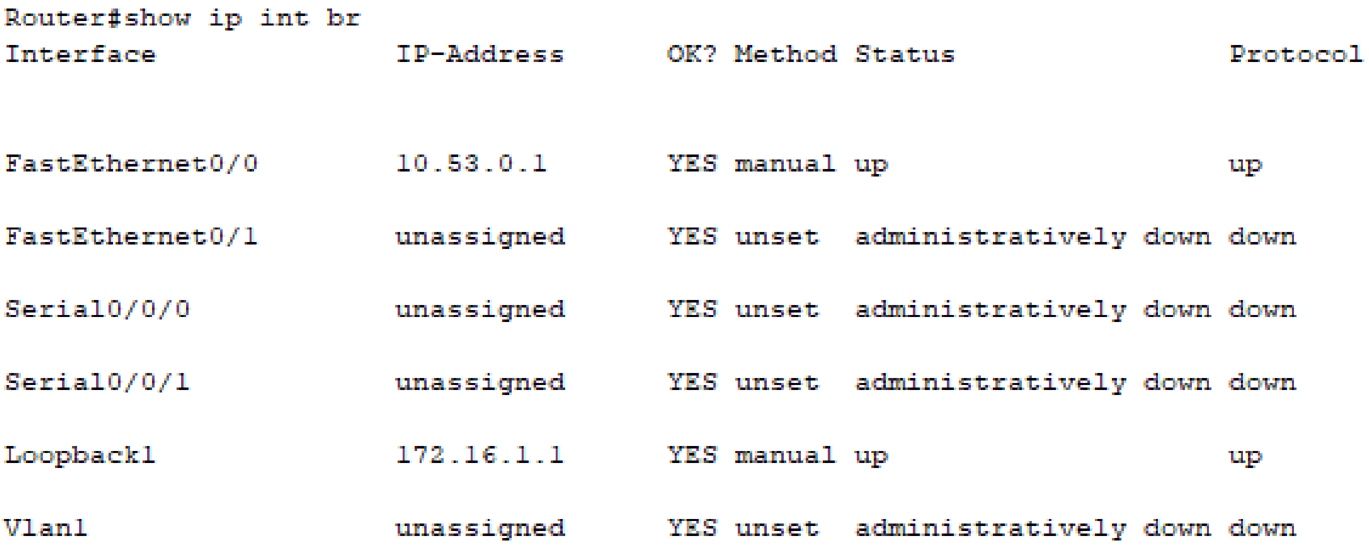
What is OSPFv3 used for?

OSPFv3 is a **routing protocol for IPv4 and IPv6**. It is a link-state protocol, as opposed to a distance-vector protocol. Think of a link as being an interface on a networking device. A link-state protocol makes its routing decisions based on the states of the links that connect source and destination machines.

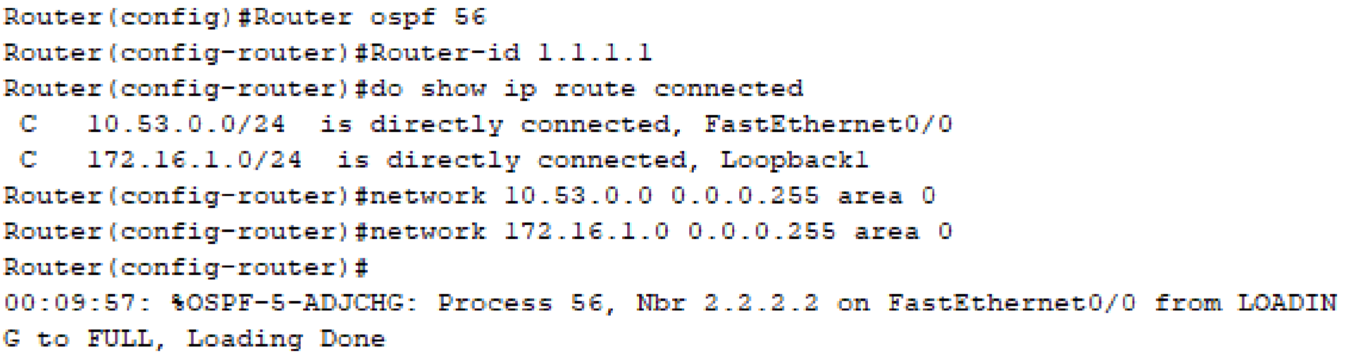

10.0.0.1
Se2/0
Rou
Fa0/0
192.168.1.1
Fa0/1
Switch0
Fa1/1
Fa0
PC0
192.168.1.2
10.0.0.2
Se2/0
20.0.0.2
Se3/0
Router1
Fa0/0
192.168.2.1
Fa0/1
Switch1
Fa1/1
Fa0
PC1
192.168.2.2
20.0.0.1
Se2/0
Rou
Fa0/0
192.168.3.1
Fa0/1
Switch2
Fa1/1
Fa0
Laptop0
192.168.3.2


1841
Router0
f0/0
f0/1
f0/2
2950-24
Switch0
f0/0
1841
Router1
f0/2
f0/1
2950-24
Switch2

Router0 :

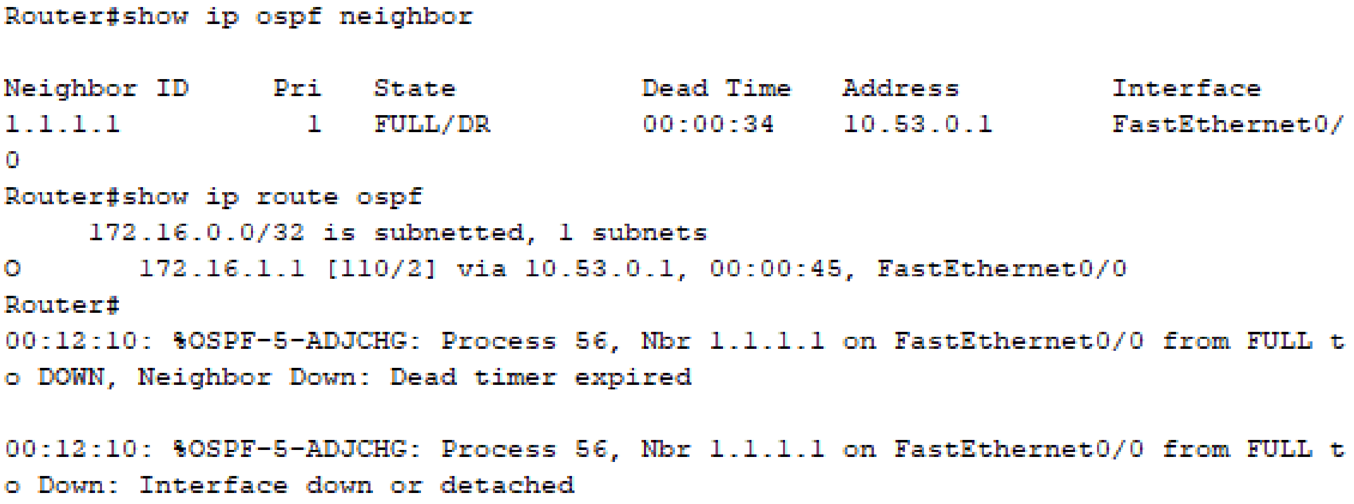


```
Router#show ip int br
Interface              IP-Address      OK? Method Status                Protocol


FastEthernet0/0        10.53.0.1       YES manual up                    up

FastEthernet0/1        unassigned      YES unset  administratively down down

Serial0/0/0            unassigned      YES unset  administratively down down

Serial0/0/1            unassigned      YES unset  administratively down down

Loopback1              172.16.1.1      YES manual up                    up

Vlan1                  unassigned      YES unset  administratively down down
```

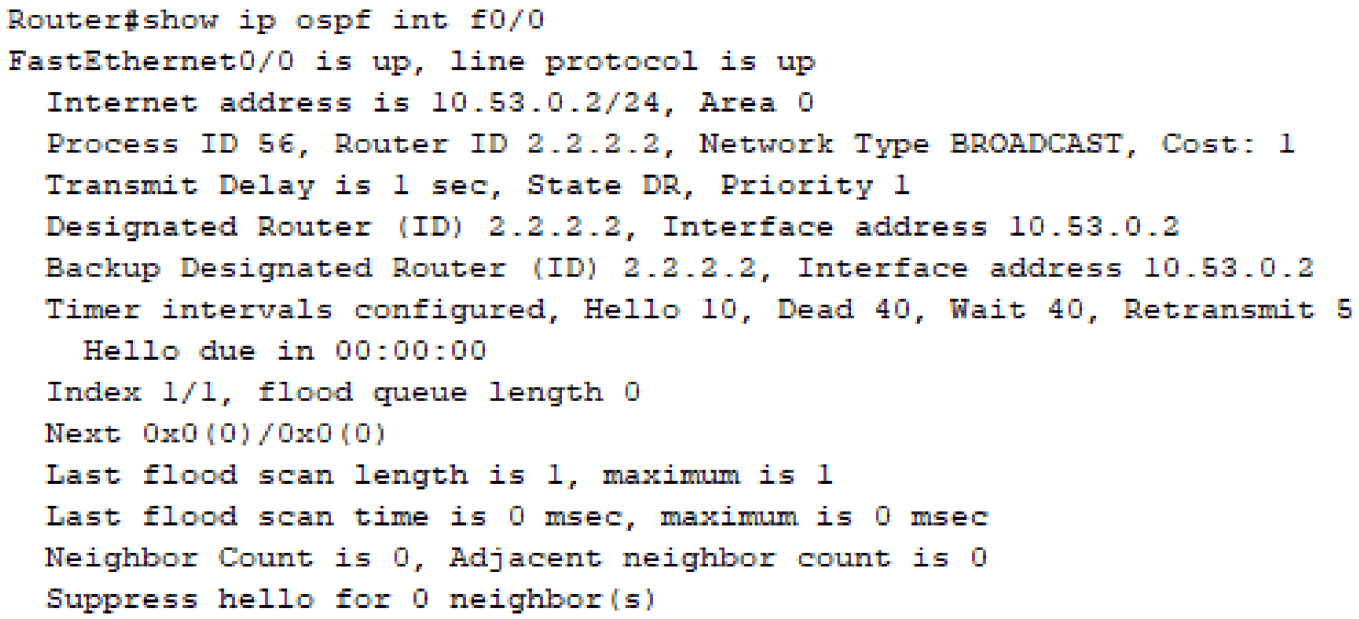


```
Router(config)#Router ospf 56
Router(config-router)#Router-id 1.1.1.1
Router(config-router)#do show ip route connected
 C   10.53.0.0/24  is directly connected, FastEthernet0/0
 C   172.16.1.0/24  is directly connected, Loopback1
Router(config-router)#network 10.53.0.0 0.0.0.255 area 0
Router(config-router)#network 172.16.1.0 0.0.0.255 area 0
Router(config-router)#
00:09:57: %OSPF-5-ADJCHG: Process 56, Nbr 2.2.2.2 on FastEthernet0/0 from LOADIN
G to FULL, Loading Done
```

Router1 :



```
Router#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/DR         00:00:34    10.53.0.1       FastEthernet0/
0
Router#show ip route ospf
     172.16.0.0/32 is subnetted, 1 subnets
O       172.16.1.1 [110/2] via 10.53.0.1, 00:00:45, FastEthernet0/0
Router#
00:12:10: %OSPF-5-ADJCHG: Process 56, Nbr 1.1.1.1 on FastEthernet0/0 from FULL t
o DOWN, Neighbor Down: Dead timer expired

00:12:10: %OSPF-5-ADJCHG: Process 56, Nbr 1.1.1.1 on FastEthernet0/0 from FULL t
o Down: Interface down or detached
```



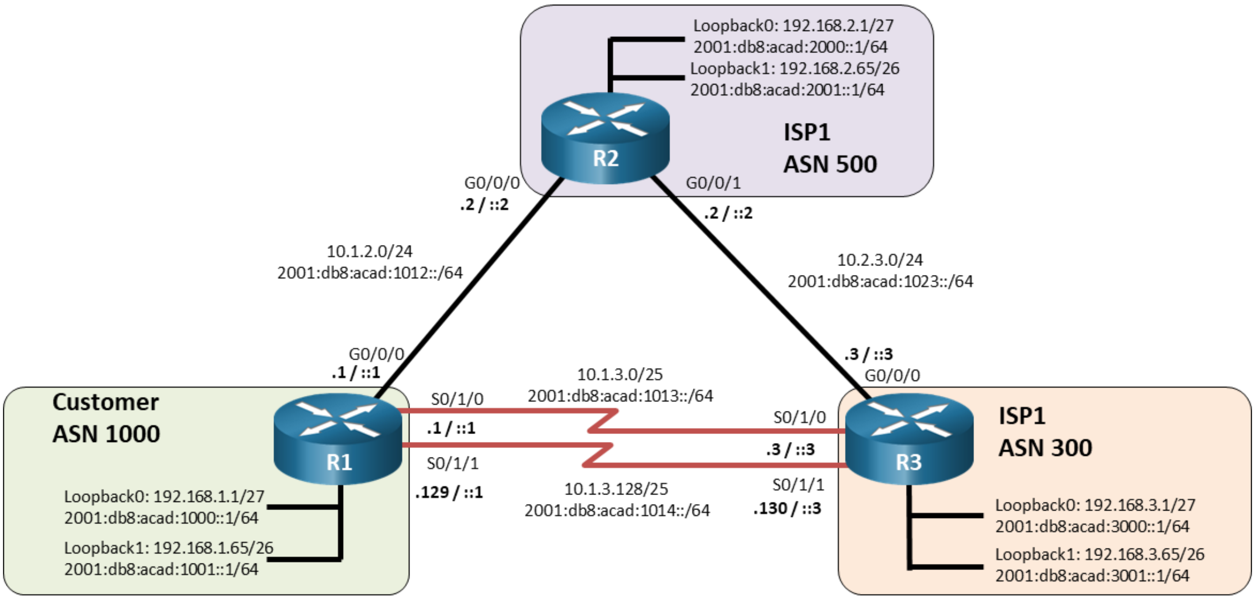
```
Router#show ip ospf int f0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.53.0.2/24, Area 0
  Process ID 56, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 2.2.2.2, Interface address 10.53.0.2
  Backup Designated Router (ID) 2.2.2.2, Interface address 10.53.0.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

PRACTICAL NO-09

Aim:- Implement BGP Communities
1. Implement MP-BGP

2. Implement eBGP for IPv4
3. Implement BGP Path Manipulation

Topology



Addressing Table

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
R1	G0/0/0	10.1.2.1/24	2001:db8:acad:1012::1/64	fe80::1:1
	S0/1/0	10.1.3.1/25	2001:db8:acad:1013::1/64	fe80::1:2
	S0/1/1	10.1.3.129/25	2001:db8:acad:1014::1/64	fe80::1:3
	Loopback0	192.168.1.1/27	2001:db8:acad:1000::1/64	fe80::1:4
	Loopback1	192.168.1.65/26	2001:db8:acad:1001::1/64	fe80::1:5
R2	G0/0/0	10.1.2.2/24	2001:db8:acad:1012::2/64	fe80::2:1

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
	G0/0/1	10.2.3.2/24	2001:db8:acad:1023::2/64	fe80::2:2
	Loopback0	192.168.2.1/27	2001:db8:acad:2000::1/64	fe80::2:3
	Loopback1	192.168.2.65/26	2001:db8:acad:2001::1/64	fe80::2:4
R3	G0/0/0	10.2.3.3/24	2001:db8:acad:1023::3/64	fe80::3:1
	S0/1/0	10.1.3.3/25	2001:db8:acad:1013::3/64	fe80::3:2
	S0/1/1	10.1.3.130/25	2001:db8:acad:1014::3/64	fe80::3:3
	Loopback0	192.168.3.1/27	2001:db8:acad:3000::1/64	fe80::3:4
	Loopback1	192.168.3.65/26	2001:db8:acad:3001::1/64	fe80::3:5

Objectives

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

Part 2: Configure MP-BGP on all Routers

Part 3: Verify MP-BGP

Part 4: Configure and Verify IPv6 Summarization

Background / Scenario

In this lab, you will configure MP-BGP, BGP for IPv4 and IPv6 using address families.

Note: This lab is an exercise in developing, deploying, and verifying various path manipulation tools for BGP, and does not reflect networking best practices.

Note: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 PC (Choice of operating system with a terminal emulation program installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each router.

a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing. A command list for each router is listed below to perform initial configuration.

Router R1

```
hostname R1
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
interface Loopback0
 ip address 192.168.1.1 255.255.255.224
 ipv6 address FE80::1:4 link-local
 ipv6 address 2001:DB8:ACAD:1000::1/64
 no shut
interface Loopback1
 ip address 192.168.1.65 255.255.255.192
 ipv6 address FE80::1:5 link-local
 ipv6 address 2001:DB8:ACAD:1001::1/64
 no shut
interface GigabitEthernet0/0/0
 ip address 10.1.2.1 255.255.255.0
```

```
  ipv6 address FE80::1:1 link-local
  ipv6 address 2001:DB8:ACAD:1012::1/64
  no shut
 interface Serial0/1/0
  ip address 10.1.3.1 255.255.255.128
  ipv6 address FE80::1:2 link-local
  ipv6 address 2001:DB8:ACAD:1013::1/64
  no shut
 interface Serial0/1/1
  ip address 10.1.3.129 255.255.255.128
  ipv6 address FE80::1:3 link-local
  ipv6 address 2001:DB8:ACAD:1014::1/64
  no shut
```

Router R2

```
hostname R2
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
interface Loopback0
 ip address 192.168.2.1 255.255.255.224
 ipv6 address FE80::2:3 link-local
 ipv6 address 2001:DB8:ACAD:2000::1/64
 no shut
interface Loopback1
 ip address 192.168.2.65 255.255.255.192
 ipv6 address FE80::2:4 link-local
 ipv6 address 2001:DB8:ACAD:2001::1/64
 no shut
interface GigabitEthernet0/0/0
 ip address 10.1.2.2 255.255.255.0
 ipv6 address FE80::2:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::2/64
 no shut
interface GigabitEthernet0/0/1
 ip address 10.2.3.2 255.255.255.0
 ipv6 address FE80::2:2 link-local
 ipv6 address 2001:DB8:ACAD:1023::2/64
 no shut
```

Router R3

```
hostname R3
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
interface Loopback0
 ip address 192.168.3.1 255.255.255.224
 ipv6 address FE80::3:4 link-local
 ipv6 address 2001:DB8:ACAD:3000::1/64
 no shut
interface Loopback1
 ip address 192.168.3.65 255.255.255.192
 ipv6 address FE80::3:5 link-local
 ipv6 address 2001:DB8:ACAD:3001::1/64
 no shut
interface GigabitEthernet0/0/0
 ip address 10.2.3.3 255.255.255.0
 negotiation auto
 ipv6 address FE80::3:1 link-local
 ipv6 address 2001:DB8:ACAD:1023::3/64
 no shut
interface Serial0/1/0
 ip address 10.1.3.3 255.255.255.128
 ipv6 address FE80::3:2 link-local
 ipv6 address 2001:DB8:ACAD:1013::3/64
 no shut
interface Serial0/1/1
 ip address 10.1.3.130 255.255.255.128
 ipv6 address FE80::3:3 link-local
 ipv6 address 2001:DB8:ACAD:1014::3/64
 no shut
```

b. Save the running configuration to startup-config.

Part 2: Configure MP-BGP on all Routers

Step 1: Implement eBGP and neighbor relationships on R1 for IPv4 and IPv6.

a. Enable IPv6 routing.

```
R1(config)# ipv6 unicast-routing
```

b. Enter BGP configuration mode from global configuration mode, specifying AS 1000 and configure the router ID.

```
R1(config)# router bgp 1000
R1(config-router)# bgp router-id 1.1.1.1
```

c. Based on the topology diagram, configure all the designated IPv4 neighbors for R1.

```
R1(config-router)# neighbor 10.1.2.2 remote-as 500
R1(config-router)# neighbor 10.1.3.3 remote-as 300
R1(config-router)# neighbor 10.1.3.130 remote-as 300
```

d. Based on the topology diagram, configure all the designated IPv6 neighbors for R1.

```
R1(config-router)# neighbor 2001:db8:acad:1012::2 remote-
as 500
R1(config-router)# neighbor 2001:db8:acad:1013::3 remote-
as 300
R1(config-router)# neighbor 2001:db8:acad:1014::3 remote-
as 300
```

e. Enter address family configuration mode for IPv4 and activate each of the IPv4 neighbors.

```
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.2.2 activate
R1(config-router-af)# neighbor 10.1.3.3 activate
R1(config-router-af)# neighbor 10.1.3.130 activate
R1(config-router-af)# exit
```

f. Enter address family configuration mode for IPv6 and activate each of the IPv6 neighbors.

```
R1(config-router)# address-family ipv6 unicast
R1(config-router-af)# neighbor 2001:db8:acad:1012::2
activate
R1(config-router-af)# neighbor 2001:db8:acad:1013::3
activate
R1(config-router-af)# neighbor 2001:db8:acad:1014::3
activate
R1(config-router-af)# exit
```

Step 2: Implement eBGP and neighbor relationships on R2 for IPv4 and IPv6.

a. Enable IPv6 routing.

```
R2(config)# ipv6 unicast-routing
```

b. Enter BGP configuration mode from global configuration mode, specifying AS 500 and configure the router ID.

```
R2(config)# router bgp 500
R2(config-router)# bgp router-id 2.2.2.2
```

c. Based on the topology diagram, configure all the designated IPv4 neighbors for R1.

```
R2(config-router)# neighbor 10.1.2.1 remote-as 1000
R2(config-router)# neighbor 10.2.3.3 remote-as 300
```

d. Based on the topology diagram, configure all the designated IPv6 neighbors for R1.

```
R2(config-router)# neighbor 2001:db8:acad:1012::1 remote-
as 1000
R2(config-router)# neighbor 2001:db8:acad:1023::3 remote-
as 300
```

e. Enter address family configuration mode for IPv4 and activate each of the IPv4 neighbors.

```
R2(config-router)# address-family ipv4 unicast
R2(config-router-af)# neighbor 10.1.2.1 activate
R2(config-router-af)# neighbor 10.2.3.3 activate
R2(config-router-af)# exit
```

f. Enter address family configuration mode for IPv6 and activate each of the IPv6 neighbors.

```
R2(config-router)# address-family ipv6 unicast
R2(config-router-af)# neighbor 2001:db8:acad:1012::1
activate
R2(config-router-af)# neighbor 2001:db8:acad:1023::3
activate
R2(config-router-af)# exit
```

Step 3: Implement eBGP and neighbor relationships on R3 for IPv4 and IPv6.

a. Enable IPv6 routing.

```
R3(config)# ipv6 unicast-routing
```

b. Enter BGP configuration mode from global configuration mode, specifying AS 300 and configure the router ID.

```
R3(config)# router bgp 300
R3(config-router)# bgp router-id 3.3.3.3
```

c. Based on the topology diagram, configure all the designated IPv4 neighbors for R1.

```
R3(config-router)# neighbor 10.2.3.2 remote-as 500
R3(config-router)# neighbor 10.1.3.1 remote-as 1000
R3(config-router)# neighbor 10.1.3.129 remote-as 1000
```

d. Based on the topology diagram, configure all the designated IPv6 neighbors for R1.

```
R3(config-router)# neighbor 2001:db8:acad:1023::2 remote-
as 500
R3(config-router)# neighbor 2001:db8:acad:1013::1 remote-
as 1000
R3(config-router)# neighbor 2001:db8:acad:1014::1 remote-
as 1000
```

e. Enter address family configuration mode for IPv4 and activate each of the IPv4 neighbors.

```
R3(config-router)# address-family ipv4 unicast
R3(config-router-af)# neighbor 10.1.3.1 activate
R3(config-router-af)# neighbor 10.1.3.129 activate
R3(config-router-af)# neighbor 10.2.3.2 activate
R3(config-router-af)# exit
```

f. Enter address family configuration mode for IPv6 and activate each of the IPv6 neighbors.

```
R3(config-router)# address-family ipv6 unicast
R3(config-router-af)# neighbor 2001:db8:acad:1023::2
activate
R3(config-router-af)# neighbor 2001:db8:acad:1013::1
activate
R3(config-router-af)# neighbor 2001:db8:acad:1014::1
activate
R3(config-router-af)# exit
```

Step 4: Advertise IPv4 and IPv6 prefixes on R1.

a. Enter address family configuration mode for IPv4 and advertise the IPv4 prefixes.

```
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# network 192.168.1.0 mask
255.255.255.224
R1(config-router-af)# network 192.168.1.64 mask
255.255.255.192
R1(config-router-af)# exit
```

b. Enter address family configuration mode for IPv6 and advertise the IPv6 prefixes.

```
R1(config-router)# address-family ipv6 unicast
R1(config-router-af)# network 2001:db8:acad:1000::/64
R1(config-router-af)# network 2001:db8:acad:1001::/64
R1(config-router-af)# exit
```

Step 5: Advertise IPv4 and IPv6 prefixes on R2.

a. Enter address family configuration mode for IPv4 and advertise the IPv4 prefixes.

```
R2(config-router)# address-family ipv4 unicast
R2(config-router-af)# network 192.168.2.0 mask
255.255.255.224
R2(config-router-af)# network 192.168.2.64 mask
255.255.255.192
R2(config-router-af)# exit
```

b. Enter address family configuration mode for IPv6 and advertise the IPv6 prefixes.

```
R2(config-router)# address-family ipv6 unicast
R2(config-router-af)# network 2001:db8:acad:2000::/64
R2(config-router-af)# network 2001:db8:acad:2001::/64
R2(config-router-af)# exit
```

Step 6: Advertise IPv4 and IPv6 prefixes on R3.

a. Enter address family configuration mode for IPv4 and advertise the IPv4 prefixes.

```
R3(config-router)# address-family ipv4 unicast
R3(config-router-af)# network 192.168.3.0 mask
255.255.255.224
R3(config-router-af)# network 192.168.3.64 mask
255.255.255.192
R3(config-router-af)# exit
```

b. Enter address family configuration mode for IPv6 and advertise the IPv6 prefixes.

```
R3(config-router)# address-family ipv6 unicast
R3(config-router-af)# network 2001:db8:acad:3000::/64
R3(config-router-af)# network 2001:db8:acad:3001::/64
R3(config-router-af)# exit
```

Note: Notice that the networks between the routers are not being advertised in eBGP. Typically, only the prefixes of the AS need to be advertised in eBGP. eBGP neighbors are typically directly connected and therefore will be able to form an adjacency. There is typically no need to advertise and inject the directly connected prefixes into the BGP routing table.

Part 3: Verify MP-BGP

Step 1: Display detailed neighbor adjacency information.

Use the **show bgp all neighbors** command on R2 to display detailed information about BGP connections to neighbors for all (IPv4 and IPv6) address families. Each neighbor shows that it is in the “Established” state. This indicates that the router can send and receive BGP messages. R2 has two neighbor addresses, R1 and R3, for each address family, IPv4 and IPv6.

```
R2# show bgp all neighbors
For address family: IPv4 Unicast
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 01:56:25
  Last read 00:00:48, last write 00:00:50, hold time is
180, keepalive interval is 60 seconds
<output omitted>

BGP neighbor is 10.2.3.3,  remote AS 300, external link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 01:55:47
  Last read 00:00:04, last write 00:00:41, hold time is
180, keepalive interval is 60 seconds
<output omitted>

For address family: IPv6 Unicast
BGP neighbor is 2001:DB8:ACAD:1012::1,  remote AS 1000,
external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 01:56:39
  Last read 00:00:07, last write 00:00:04, hold time is
180, keepalive interval is 60 seconds
<output omitted>

BGP neighbor is 2001:DB8:ACAD:1023::3,  remote AS 300,
external link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 01:56:09
  Last read 00:00:32, last write 00:00:48, hold time is
180, keepalive interval is 60 seconds
<output omitted>
```

Note: Most information displayed using **show bgp all neighbors** command has been omitted for brevity. The command **show bgp neighbors** is used to display only BGP for IPv4 adjacencies. To display the same information for only IPv6 neighbors, use the command **show bgp ipv6 neighbors**.

Questions:

What is the BGP state for each neighbor adjacency?

Type your answers here.

Established

How often are BGP keepalives sent?

Type your answers here.

Every 60 seconds

How many seconds will a BGP session remain open if no further keepalive messages are received?

Type your answers here.

180 seconds, the value of the hold time interval

Step 2: Display summary neighbor adjacency information.

Use the **show bgp ipv4 unicast summary** and **show bgp ipv6 unicast summary** commands on R2 to display a summary of IPv4/IPv6 peering information with R1 and R3. The information displayed using the **show bgp ipv4 unicast summary** is a subset of **show ip all bgp** command.

```
R2# show bgp ipv4 unicast summary
BGP router identifier 2.2.2.2, local AS number 500
BGP table version is 11, main routing table version 11
6 network entries using 1488 bytes of memory
10 path entries using 1360 bytes of memory
5/3 BGP path/bestpath attribute entries using 1400 bytes
of memory
4 BGP AS-PATH entries using 128 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4376 total bytes of memory
BGP activity 12/0 prefixes, 20/0 paths, scan interval 60
secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer
InQ OutQ Up/Down  State/PfxRcd
10.1.2.1        4         1000     152     151       11
0    0 02:12:36   4
10.2.3.3        4          300     150     150       11
0    0 02:11:51   4
```

```
R2# show bgp ipv6 unicast summary
BGP router identifier 2.2.2.2, local AS number 500
BGP table version is 9, main routing table version 9
6 network entries using 1632 bytes of memory
10 path entries using 1520 bytes of memory
5/3 BGP path/bestpath attribute entries using 1400 bytes
of memory
4 BGP AS-PATH entries using 128 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4680 total bytes of memory
BGP activity 12/0 prefixes, 20/0 paths, scan interval 60
secs

Neighbor        V          AS MsgRcvd MsgSent   TblVer
InQ OutQ Up/Down  State/PfxRcd
2001:DB8:ACAD:1012::1
                4        1000     150     150        9
0    0 02:12:39  4
2001:DB8:ACAD:1023::3
                4         300     151     150        9
0    0 02:11:54  4
```

Question:

What is the difference between the "local AS number" and the "AS" number displayed in the list of BGP neighbors?

Type your answers here.

The local AS is the AS that this router belongs to. The AS in the list of BGP neighbors is the AS of the remote neighbor.

Step 3: Verify BGP tables for IPv4 and IPv6.

a. Use the **show bgp ipv4 unicast** command on R2 to display its IPv4 BGP table. This command is equivalent to the **show ip bgp** command and either command can be used. Notice that R1 shows six IPv4 networks in its IPv4 BGP table. Each network is valid "*" and has one path which is the best path ">". Amongst other information, the next hop IPv4 address and the AS path are included.

```
R2# show bgp ipv4 unicast
BGP table version is 11, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *    192.168.1.0/27   10.2.3.3
0 300 1000 i
 *>                    10.1.2.1                 0
0 1000 i
 *    192.168.1.64/26  10.2.3.3
0 300 1000 i
 *>                    10.1.2.1                 0
0 1000 i
 *>   192.168.2.0/27   0.0.0.0                  0
32768 i
 *>   192.168.2.64/26  0.0.0.0                  0
32768 i
 *    192.168.3.0/27   10.1.2.1
0 1000 300 i
 *>                    10.2.3.3                 0
0 300 i
 *    192.168.3.64/26  10.1.2.1
0 1000 300 i
 *>                    10.2.3.3                 0
0 300 i
```

b. Use the **show bgp ipv6 unicast** command on R2 to display similar information for its IPv6 BGP table.

```
R2# show bgp ipv6 unicast
BGP table version is 9, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
```

```
 *    2001:DB8:ACAD:1000::/64
                      2001:DB8:ACAD:1023::3

0 300 1000 i
 *>                   2001:DB8:ACAD:1012::1
                                          0
0 1000 i
 *    2001:DB8:ACAD:1001::/64
                      2001:DB8:ACAD:1023::3

0 300 1000 i
 *>                   2001:DB8:ACAD:1012::1
                                          0
0 1000 i
 *>   2001:DB8:ACAD:2000::/64
                      ::                 0
32768 i
 *>   2001:DB8:ACAD:2001::/64
                      ::                 0
32768 i
 *    2001:DB8:ACAD:3000::/64
                      2001:DB8:ACAD:1012::1

0 1000 300 i
 *>                   2001:DB8:ACAD:1023::3
                                          0
0 300 i
 *    2001:DB8:ACAD:3001::/64
                      2001:DB8:ACAD:1012::1

0 1000 300 i
 *>                   2001:DB8:ACAD:1023::3
                                          0
0 300 i
```

Questions:

In the first output **show bgp ipv4 unicast**, why is 10.1.2.1 the preferred next hop address for 192.168.1.0 instead of 10.2.3.3?

Type your answers here.

Given that prior BGP path selection attributes are equal, this next hop is only one AS away, whereas 10.2.3.3 is two AS hops.

Why do some entries in the **show bgp ipv6 unicast** output include a next hop address of "::"?

Type your answers here.

This is unspecified address and indicates that the local router is generating the prefix for the BGP table.

Step 4: Viewing explicit routes and path attributes.

a. Use the **show bgp ipv4 unicast** *ipv4-prefix subnet-mask* command on R2 to display all the paths for a specific route and BGP path attributes for that route.

```
R2# show bgp ipv4 unicast 192.168.1.0 255.255.255.224
BGP routing table entry for 192.168.1.0/27, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  300 1000
    10.2.3.3 from 10.2.3.3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  1000
    10.1.2.1 from 10.1.2.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid,
external, best
      rx pathid: 0, tx pathid: 0x0
```

The **show bgp ipv6 unicast** *ipv6-prefix prefix-length* command displays similar information for IPv6 prefixes.

```
R2# show bgp ipv6 unicast 2001:db8:acad:1000::/64
BGP routing table entry for 2001:DB8:ACAD:1000::/64,
version 2
Paths: (2 available, best #2, table default)
  Flag: 0x100
  Advertised to update-groups:
     1
  Refresh Epoch 1
  300 1000
    2001:DB8:ACAD:1023::3 (FE80::3:1) from
2001:DB8:ACAD:1023::3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  1000
    2001:DB8:ACAD:1012::1 (FE80::1:1) from
2001:DB8:ACAD:1012::1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid,
external, best
```

```
      rx pathid: 0, tx pathid: 0x0
```

Question:

Why does the output for the **show bgp ipv6 unicast** command include the link-local address following the global unicast address?

Type your answers here.

This is the link-local address of the router that sent the BGP update. Link-local addresses are used to send BGP messages.

b. Use the **show bgp ipv4 unicast neighbors** *ipv4-prefix* **advertised-routes** command on R2 to display IPv4 routes advertised to a specific neighbor.

```
R2# show bgp ipv4 unicast neighbors 10.1.2.1 advertised-
routes
BGP table version is 11, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *>   192.168.1.0/27   10.1.2.1                 0
0 1000 i
 *>   192.168.1.64/26  10.1.2.1                 0
0 1000 i
 *>   192.168.2.0/27   0.0.0.0                  0
32768 i
 *>   192.168.2.64/26  0.0.0.0                  0
32768 i
 *>   192.168.3.0/27   10.2.3.3                 0
0 300 i
 *>   192.168.3.64/26  10.2.3.3                 0
0 300 i

Total number of prefixes 6
```

c. Use the **show bgp ipv6 unicast** *ipv5-prefix prefix-length* command to display similar information for IPv6 advertised routes.

```
R2# show bgp ipv6 unicast neighbors 2001:db8:acad:1012::1
advertised-routes
BGP table version is 9, local router ID is 2.2.2.2
```

```
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *>   2001:DB8:ACAD:1000::/64
                      2001:DB8:ACAD:1012::1
                                                0
0 1000 i
 *>   2001:DB8:ACAD:1001::/64
                      2001:DB8:ACAD:1012::1
                                                0
0 1000 i
 *>   2001:DB8:ACAD:2000::/64
                      ::                       0
32768 i
 *>   2001:DB8:ACAD:2001::/64
                      ::                       0
32768 i
 *>   2001:DB8:ACAD:3000::/64
                      2001:DB8:ACAD:1023::3
                                                0
0 300 i
 *>   2001:DB8:ACAD:3001::/64
                      2001:DB8:ACAD:1023::3
                                                0
0 300 i

Total number of prefixes 6
```

Question:

Why do some entries in the **show bgp ipv4 unicast neighbors** output include a next hop address of 0.0.0.0 and the **show bgp ipv6 unicast neighbors** output includes a next hop address of "::"?

Type your answers here.

This indicates that the local router is generating the prefix.

Step 5: Verifying the IP routing tables for IPv4 and IPv6.

a. By examining the IPv4 and IPv6 routing tables on R2, you can verify that BGP is receiving the IPv4 and IPv6 prefixes from R1 and R3.

```
R2# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 
masks
B        192.168.1.0/27 [20/0] via 10.1.2.1, 04:29:03
B        192.168.1.64/26 [20/0] via 10.1.2.1, 04:28:32
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 
masks
B        192.168.3.0/27 [20/0] via 10.2.3.3, 04:17:14
B        192.168.3.64/26 [20/0] via 10.2.3.3, 04:16:44

R2# show ipv6 route bgp | section 2001
B   2001:DB8:ACAD:1000::/64 [20/0]
     via FE80::1:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:1001::/64 [20/0]
     via FE80::1:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:3000::/64 [20/0]
     via FE80::3:1, GigabitEthernet0/0/1
B   2001:DB8:ACAD:3001::/64 [20/0]
     via FE80::3:1, GigabitEthernet0/0/1
```

Part 4: Configure and Verify IPv6 Route Summarization

Summarizing prefixes conserves router resources and accelerates best-path calculation by reducing the size of the table. Summarization can be configured either for prefixes originated by the AS or prefixes received from downstream providers. Summarization also provides the benefits of stability by hiding flapping routes or having to install new prefixes when they are contained within a summary.

a. Verify R2 and R3 are receiving 2001:db8:acad:1000::/64 and 2001:db8:acad:1001::/64 from R1.

```
R2# show ipv6 route bgp | section 2001
B   2001:DB8:ACAD:1000::/64 [20/0]
     via FE80::1:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:1001::/64 [20/0]
     via FE80::1:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:3000::/64 [20/0]
     via FE80::3:1, GigabitEthernet0/0/1
```

```
B   2001:DB8:ACAD:3001::/64 [20/0]
     via FE80::3:1, GigabitEthernet0/0/1

R3# show ipv6 route bgp | section 2001
B   2001:DB8:ACAD:1000::/64 [20/0]
     via FE80::1:2, Serial0/1/0
B   2001:DB8:ACAD:1001::/64 [20/0]
     via FE80::1:2, Serial0/1/0
B   2001:DB8:ACAD:2000::/64 [20/0]
     via FE80::2:2, GigabitEthernet0/0/0
B   2001:DB8:ACAD:2001::/64 [20/0]
     via FE80::2:2, GigabitEthernet0/0/0
```

b. Although AS 1000 only has two IPv6 prefixes - 2001:db8:acad:1000::/64 and 2001:db8:acad:1001::/64, this customer has been allocated the entire 2001:db8:acad:1000::/52 prefix (2001:db8:acad:1xxx).

R1 is configured using the **aggregate-address** command in IPv6 AF mode to summarize its IPv6 prefixes. This is known as a summary route or aggregate route. The **summary-only** option suppresses the more specific prefixes from also being advertised.

```
R1(config)# router bgp 1000
R1(config-router)# address-family ipv6 unicast
R1(config-router-af)# aggregate-address
2001:db8:acad:1000::/52 summary-only
```

c. Verify that R2 and R3 are now receiving the aggregate route and installing it in the IPv6 BGP table.

```
R2# show bgp ipv6 unicast | begin Network
     Network          Next Hop            Metric LocPrf
Weight Path
 *    2001:DB8:ACAD:1000::/52
                      2001:DB8:ACAD:1023::3

0 300 1000 i
 *>                   2001:DB8:ACAD:1012::1
                                                0
0 1000 i
<output omitted>

R3# show bgp ipv6 unicast | begin Network
     Network          Next Hop            Metric LocPrf
Weight Path
 *    2001:DB8:ACAD:1000::/52
                      2001:DB8:ACAD:1023::2
```

```
0 500 1000 i
 *                      2001:DB8:ACAD:1014::1
                                                          0
0 1000 i
<output omitted>
```

d. Verify that R2 and R3 are now receiving the aggregate route and it is installed in the IPv6 routing table.

```
R2# show ipv6 route bgp | section 2001
B   2001:DB8:ACAD:1000::/52 [20/0]
     via FE80::1:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:3000::/64 [20/0]
     via FE80::3:1, GigabitEthernet0/0/1
B   2001:DB8:ACAD:3001::/64 [20/0]
     via FE80::3:1, GigabitEthernet0/0/1

R3# show ipv6 route bgp | section 2001
B   2001:DB8:ACAD:1000::/52 [20/0]
     via FE80::1:2, Serial0/1/0
B   2001:DB8:ACAD:2000::/64 [20/0]
     via FE80::2:2, GigabitEthernet0/0/0
B   2001:DB8:ACAD:2001::/64 [20/0]
     via FE80::2:2, GigabitEthernet0/0/0
```

Question:

If R1's 2001:db8:acad:1000::/64 network went down, what would be the effect, if any, on the routing tables of R2 and R3? Explain.

Type your answers here.

There would be no effect because the summary route will still be valid and advertised as long as at least one subnet within the summary route is still reachable. In this case, as long as 2001:db8:acad:1001::/64 is still reachable, this aggregated route will still be advertised.

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs - Final

Router R1

```
R1# show running-config
Building configuration...


Current configuration : 2651 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R1
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.224
 ipv6 address FE80::1:4 link-local
 ipv6 address 2001:DB8:ACAD:1000::1/64
!
interface Loopback1
 ip address 192.168.1.65 255.255.255.192
 ipv6 address FE80::1:5 link-local
 ipv6 address 2001:DB8:ACAD:1001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.1.2.1 255.255.255.0
 negotiation auto
 ipv6 address FE80::1:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::1/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface Serial0/1/0
 ip address 10.1.3.1 255.255.255.128
 ipv6 address FE80::1:2 link-local
 ipv6 address 2001:DB8:ACAD:1013::1/64
!
interface Serial0/1/1
 ip address 10.1.3.129 255.255.255.128
 ipv6 address FE80::1:3 link-local
 ipv6 address 2001:DB8:ACAD:1014::1/64
!
router bgp 1000
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor 10.1.2.2 remote-as 500
 neighbor 10.1.3.3 remote-as 300
 neighbor 10.1.3.130 remote-as 300
 neighbor 2001:DB8:ACAD:1012::2 remote-as 500
 neighbor 2001:DB8:ACAD:1013::3 remote-as 300
 neighbor 2001:DB8:ACAD:1014::3 remote-as 300
 !
 address-family ipv4
  network 192.168.1.0 mask 255.255.255.224
  network 192.168.1.64 mask 255.255.255.192
  neighbor 10.1.2.2 activate
  neighbor 10.1.3.3 activate
  neighbor 10.1.3.130 activate
  no neighbor 2001:DB8:ACAD:1012::2 activate
  no neighbor 2001:DB8:ACAD:1013::3 activate
  no neighbor 2001:DB8:ACAD:1014::3 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:DB8:ACAD:1000::/64
  network 2001:DB8:ACAD:1001::/64
  aggregate-address 2001:DB8:ACAD:1000::/52 summary-only
  neighbor 2001:DB8:ACAD:1012::2 activate
  neighbor 2001:DB8:ACAD:1013::3 activate
  neighbor 2001:DB8:ACAD:1014::3 activate
 exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
```

```
 stopbits 1
line vty 0 4
 login
!
end
```

Router R2

```
R2# show running-config
Building configuration...


Current configuration : 2218 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.2.1 255.255.255.224
 ipv6 address FE80::2:3 link-local
 ipv6 address 2001:DB8:ACAD:2000::1/64
!
interface Loopback1
 ip address 192.168.2.65 255.255.255.192
 ipv6 address FE80::2:4 link-local
 ipv6 address 2001:DB8:ACAD:2001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.1.2.2 255.255.255.0
 negotiation auto
 ipv6 address FE80::2:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::2/64
!
interface GigabitEthernet0/0/1
 ip address 10.2.3.2 255.255.255.0
 negotiation auto
 ipv6 address FE80::2:2 link-local
 ipv6 address 2001:DB8:ACAD:1023::2/64
!
router bgp 500
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 neighbor 10.1.2.1 remote-as 1000
 neighbor 10.2.3.3 remote-as 300
 neighbor 2001:DB8:ACAD:1012::1 remote-as 1000
 neighbor 2001:DB8:ACAD:1023::3 remote-as 300
 !
 address-family ipv4
  network 192.168.2.0 mask 255.255.255.224
  network 192.168.2.64 mask 255.255.255.192
  neighbor 10.1.2.1 activate
  neighbor 10.2.3.3 activate
  no neighbor 2001:DB8:ACAD:1012::1 activate
  no neighbor 2001:DB8:ACAD:1023::3 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:DB8:ACAD:2000::/64
  network 2001:DB8:ACAD:2001::/64
  neighbor 2001:DB8:ACAD:1012::1 activate
  neighbor 2001:DB8:ACAD:1023::3 activate
 exit-address-family
!
ip forward-protocol nd
```

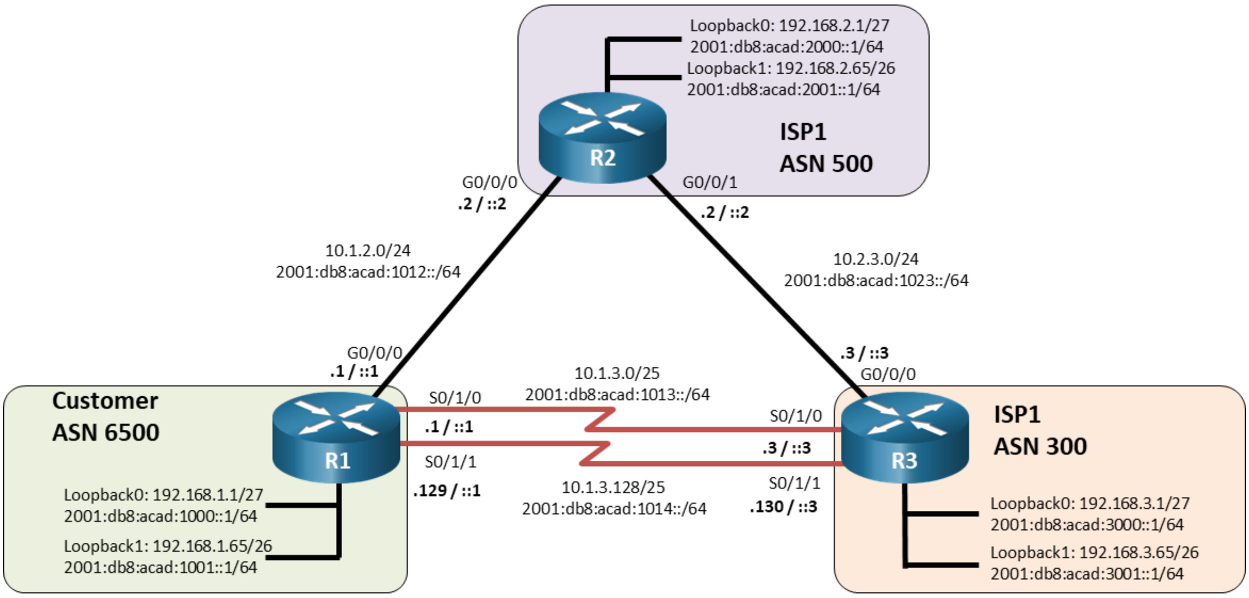
```
no ip http server
ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

Router R3

```
R3#show running-config
Building configuration...


Current configuration : 2597 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
```

```
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.3.1 255.255.255.224
 ipv6 address FE80::3:4 link-local
 ipv6 address 2001:DB8:ACAD:3000::1/64
!
interface Loopback1
 ip address 192.168.3.65 255.255.255.192
 ipv6 address FE80::3:5 link-local
 ipv6 address 2001:DB8:ACAD:3001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.2.3.3 255.255.255.0
 negotiation auto
 ipv6 address FE80::3:1 link-local
 ipv6 address 2001:DB8:ACAD:1023::3/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface Serial0/1/0
 ip address 10.1.3.3 255.255.255.128
 ipv6 address FE80::3:2 link-local
 ipv6 address 2001:DB8:ACAD:1013::3/64
!
interface Serial0/1/1
 ip address 10.1.3.130 255.255.255.128
 ipv6 address FE80::3:3 link-local
 ipv6 address 2001:DB8:ACAD:1014::3/64
!
router bgp 300
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 neighbor 10.1.3.1 remote-as 1000
 neighbor 10.1.3.129 remote-as 1000
 neighbor 10.2.3.2 remote-as 500
 neighbor 2001:DB8:ACAD:1013::1 remote-as 1000
 neighbor 2001:DB8:ACAD:1014::1 remote-as 1000
 neighbor 2001:DB8:ACAD:1023::2 remote-as 500
 !
 address-family ipv4
  network 192.168.3.0 mask 255.255.255.224
  network 192.168.3.64 mask 255.255.255.192
  neighbor 10.1.3.1 activate
  neighbor 10.1.3.129 activate
  neighbor 10.2.3.2 activate
  no neighbor 2001:DB8:ACAD:1013::1 activate
  no neighbor 2001:DB8:ACAD:1014::1 activate
  no neighbor 2001:DB8:ACAD:1023::2 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:DB8:ACAD:3000::/64
  network 2001:DB8:ACAD:3001::/64
  neighbor 2001:DB8:ACAD:1013::1 activate
  neighbor 2001:DB8:ACAD:1014::1 activate
  neighbor 2001:DB8:ACAD:1023::2 activate
 exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

[Title] Topology



Addressing Table

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
R1	G0/0/0	10.1.2.1/24	2001:db8:acad:1012::1/64	fe80::1:1
	S0/1/0	10.1.3.1/25	2001:db8:acad:1013::1/64	fe80::1:2
	S0/1/1	10.1.3.129/25	2001:db8:acad:1014::1/64	fe80::1:3
	Loopback0	192.168.1.1/27	2001:db8:acad:1000::1/64	fe80::1:4
	Loopback1	192.168.1.65/26	2001:db8:acad:1001::1/64	fe80::1:5
R2	G0/0/0	10.1.2.2/24	2001:db8:acad:1012::2/64	fe80::2:1
	G0/0/1	10.2.3.2/24	2001:db8:acad:1023::2/64	fe80::2:2
	Loopback0	192.168.2.1/27	2001:db8:acad:2000::1/64	fe80::2:4
	Loopback1	192.168.2.65/26	2001:db8:acad:2001::1/64	fe80::2:4
R3	G0/0/0	10.2.3.3/24	2001:db8:acad:1023::3/64	fe80::3:1

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
	S0/1/0	10.1.3.3/25	2001:db8:acad:1013::3/64	fe80::3:2
	S0/1/1	10.1.3.130/25	2001:db8:acad:1014::3/64	fe80::3:3
	Loopback0	192.168.3.1/27	2001:db8:acad:3000::1/64	fe80::3:4
	Loopback1	192.168.3.65/26	2001:db8:acad:3001::1/64	fe80::3:5

Objectives

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

Part 2: Configure and Verify Multi-Protocol BGP on all Routers

Part 3: Configure and Verify BGP Path Manipulation Settings on all Routers

Background / Scenario

The default settings in BGP allow for a great deal of undesired route information to pass between autonomous systems. In this lab you will configure Multi-Protocol BGP and implement various path manipulation options for both IPv4 and IPv6.

Note: This lab is an exercise in developing, deploying, and verifying various path manipulation tools for BGP, and does not reflect networking best practices.

Note: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the routers have been erased and have no startup configurations. If you are unsure contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 PC (Choice of operating system with a terminal emulation program installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each router.

a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing. A command list for each router is listed below to perform initial configuration.

Router R1

```
no ip domain lookup
hostname R1
line con 0
 exec-timeout 0 0
 logging synchronous
banner motd # This is R1, BGP Path Manipulation Lab #
ipv6 unicast-routing
interface g0/0/0
 ip address 10.1.2.1 255.255.255.0
 ipv6 address fe80::1:1 link-local
 ipv6 address 2001:db8:acad:1012::1/64
 no shutdown
interface s0/1/0
 ip address 10.1.3.1 255.255.255.128
 ipv6 address fe80::1:2 link-local
 ipv6 address 2001:db8:acad:1013::1/64
 no shutdown
interface s0/1/1
 ip address 10.1.3.129 255.255.255.128
 ipv6 address fe80::1:3 link-local
 ipv6 address 2001:db8:acad:1014::1/64
 no shutdown
interface loopback 0
 ip address 192.168.1.1 255.255.255.224
 ipv6 address fe80::1:4 link-local
 ipv6 address 2001:db8:acad:1000::1/64
 no shutdown
```

```
interface loopback 1
 ip address 192.168.1.65 255.255.255.192
 ipv6 address fe80::1:5 link-local
 ipv6 address 2001:db8:acad:1001::1/64
 no shutdown
```

Router R2

```
no ip domain lookup
hostname R2
line con 0
 exec-timeout 0 0
 logging synchronous
banner motd # This is R2, BGP Path Manipulation Lab #
ipv6 unicast-routing
interface g0/0/0
 ip address 10.1.2.2 255.255.255.0
 ipv6 address fe80::2:1 link-local
 ipv6 address 2001:db8:acad:1012::2/64
 no shutdown
interface g0/0/1
 ip address 10.2.3.2 255.255.255.0
 ipv6 address fe80::2:2 link-local
 ipv6 address 2001:db8:acad:1023::2/64
 no shutdown
interface loopback 0
 ip address 192.168.2.1 255.255.255.224
 ipv6 address fe80::2:3 link-local
 ipv6 address 2001:db8:acad:2000::1/64
 no shutdown
interface loopback 1
 ip address 192.168.2.65 255.255.255.192
 ipv6 address fe80::2:4 link-local
 ipv6 address 2001:db8:acad:2001::1/64
 no shutdown
```

Router R3

```
no ip domain lookup
hostname R3
line con 0
 exec-timeout 0 0
 logging synchronous
banner motd # This is R3, BGP Path Manipulation Lab #
ipv6 unicast-routing
```

```
interface g0/0/0
 ip address 10.2.3.3 255.255.255.0
 ipv6 address fe80::3:1 link-local
 ipv6 address 2001:db8:acad:1023::3/64
 no shutdown
interface s0/1/0
 ip address 10.1.3.3 255.255.255.128
 ipv6 address fe80::3:2 link-local
 ipv6 address 2001:db8:acad:1013::3/64
 no shutdown
interface s0/1/1
 ip address 10.1.3.130 255.255.255.128
 ipv6 address fe80::3:3 link-local
 ipv6 address 2001:db8:acad:1014::3/64
 no shutdown
interface loopback 0
 ip address 192.168.3.1 255.255.255.224
 ipv6 address fe80::3:4 link-local
 ipv6 address 2001:db8:acad:3000::1/64
 no shutdown
interface loopback 1
 ip address 192.168.3.65 255.255.255.192
 ipv6 address fe80::3:5 link-local
 ipv6 address 2001:db8:acad:3001::1/64
 no shutdown
```

b. Set the clock on each router to UTC time.

c. Save the running configuration to startup-config.

Part 2: Configure and Verify Multi-Protocol BGP on all Routers

In Part 2, you will configure and verify Multi-Protocol BGP on all routers to achieve full connectivity between the routers. The text below provides you with the complete configuration for R1. You will use this to inform your configuration of R2 and R3. The configuration being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: On R1, create the core BGP configuration.

a. Enter BGP configuration mode from global configuration mode, specifying AS 6500.

```
R1(config)# router bgp 6500
```

b. Configure the BGP router-id for R1.

```
R1(config-router)# bgp router-id 1.1.1.1
```

c. Disable the default IPv4 unicast address family behavior.

```
R1(config-router)# no bgp default ipv4-unicast
```

d. Based on the topology diagram, configure all the designated neighbors for R1.

```
R1(config-router)# neighbor 10.1.2.2 remote-as 500
R1(config-router)# neighbor 10.1.3.3 remote-as 300
R1(config-router)# neighbor 10.1.3.130 remote-as 300
R1(config-router)# neighbor 2001:db8:acad:1012::2 remote-
as 500
R1(config-router)# neighbor 2001:db8:acad:1013::3 remote-
as 300
R1(config-router)# neighbor 2001:db8:acad:1014::3 remote-
as 300
```

Step 2: On R1, configure the IPv4 unicast address family.

a. Enter the IPv4 unicast address family configuration mode.

```
R1(config-router)# address-family ipv4 unicast
```

b. Configure network statements for the IPv4 networks attached to interfaces loopback0 and loopback1. Remember that BGP does not work the same way that an IGP does, and that the network statement has no impact on neighbor adjacency; it is used solely for advertising purposes.

```
R1(config-router-af)# network 192.168.1.0 mask
255.255.255.224
R1(config-router-af)# network 192.168.1.64 mask
255.255.255.192
```

c. Deactivate the IPv6 neighbors and activate the IPv4 neighbors.

```
R1(config-router-af)# no neighbor 2001:db8:acad:1012::2
activate
R1(config-router-af)# no neighbor 2001:db8:acad:1013::3
activate
R1(config-router-af)# no neighbor 2001:db8:acad:1014::3
activate
R1(config-router-af)# neighbor 10.1.2.2 activate
R1(config-router-af)# neighbor 10.1.3.3 activate
R1(config-router-af)# neighbor 10.1.3.130 activate
```

Step 3: On R1, configure the IPv6 unicast address family.

a. Enter the IPv6 unicast address family configuration mode.

```
R1(config-router)# address-family ipv6 unicast
```

b. Configure network statements for the IPv6 networks that are attached to interfaces loopback0 and loopback1. Remember that BGP does not work the same way that an IGP does; therefore, the network statement has no impact on neighbor adjacency; it is used solely for advertising purposes.

```
R1(config-router-af)# network 2001:db8:acad:1000::/64
R1(config-router-af)# network 2001:db8:acad:1001::/64
```

c. Activate the IPv6 neighbors that are configured for BGP.

```
R1(config-router-af)# neighbor 2001:db8:acad:1012::2
activate
R1(config-router-af)# neighbor 2001:db8:acad:1013::3
activate
R1(config-router-af)# neighbor 2001:db8:acad:1014::3
activate
```

Step 4: Configure MP-BGP on R2 and R3 as you did in the previous step.

```
R2(config)# router bgp 500
R2(config-router)# bgp router-id 2.2.2.2
R2(config-router)# no bgp default ipv4-unicast
R2(config-router)# neighbor 10.1.2.1 remote-as 6500
R2(config-router)# neighbor 10.2.3.3 remote-as 300
R2(config-router)# neighbor 2001:db8:acad:1012::1 remote-
as 6500
R2(config-router)# neighbor 2001:db8:acad:1023::3 remote-
as 300
R2(config-router)# address-family ipv4
R2(config-router-af)# network 192.168.2.0 mask
255.255.255.224
R2(config-router-af)# network 192.168.2.64 mask
255.255.255.192
R2(config-router-af)# neighbor 10.1.2.1 activate
R2(config-router-af)# neighbor 10.2.3.3 activate
R2(config-router-af)# no neighbor 2001:db8:acad:1012::1
activate
R2(config-router-af)# no neighbor 2001:db8:acad:1023::3
activate
R2(config-router-af)# exit-address-family
R2(config-router)# address-family ipv6
R2(config-router-af)# network 2001:db8:acad:2000::/64
R2(config-router-af)# network 2001:db8:acad:2001::/64
R2(config-router-af)# neighbor 2001:db8:acad:1012::1
activate
R2(config-router-af)# neighbor 2001:db8:acad:1023::3
activate
R2(config-router-af)# exit-address-family
```

```
R3(config)# router bgp 300
R3(config-router)# bgp router-id 3.3.3.3
R3(config-router)# no bgp default ipv4-unicast
R3(config-router)# neighbor 10.1.3.1 remote-as 6500
R3(config-router)# neighbor 10.1.3.129 remote-as 6500
R3(config-router)# neighbor 10.2.3.2 remote-as 500
R3(config-router)# neighbor 2001:db8:acad:1013::1 remote-
as 6500
R3(config-router)# neighbor 2001:db8:acad:1014::1 remote-
as 6500
R3(config-router)# neighbor 2001:db8:acad:1023::2 remote-
as 500
R3(config-router)# address-family ipv4
R3(config-router-af)# network 192.168.3.0 mask
255.255.255.224
R3(config-router-af)# network 192.168.3.64 mask
255.255.255.192
R3(config-router-af)# neighbor 10.1.3.1 activate
R3(config-router-af)# neighbor 10.1.3.129 activate
R3(config-router-af)# neighbor 10.2.3.2 activate
R3(config-router-af)# no neighbor 2001:db8:acad:1013::1
activate
R3(config-router-af)# no neighbor 2001:db8:acad:1014::1
activate
R3(config-router-af)# no neighbor 2001:db8:acad:1023::2
activate
R3(config-router-af)# exit-address-family
R3(config-router)# address-family ipv6
R3(config-router-af)# network 2001:db8:acad:3000::/64
R3(config-router-af)# network 2001:db8:acad:3001::/64
R3(config-router-af)# neighbor 2001:db8:acad:1013::1
activate
R3(config-router-af)# neighbor 2001:db8:acad:1014::1
activate
R3(config-router-af)# neighbor 2001:db8:acad:1023::2
activate
R3(config-router-af)# exit-address-family
```

Step 5: Verify that MP-BGP is operational.

a. Use the **show bgp ipv4 unicast summary** and **show bgp ipv6 unicast summary** commands to verify that BGP has established three IPv4 and three IPv6 adjacencies and received four prefixes from each neighbor.

```
R1# show bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 6500
BGP table version is 9, main routing table version 9
6 network entries using 1488 bytes of memory
14 path entries using 1904 bytes of memory
5/3 BGP path/bestpath attribute entries using 1400 bytes
of memory
4 BGP AS-PATH entries using 128 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4920 total bytes of memory
BGP activity 12/0 prefixes, 28/0 paths, scan interval 60
secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer
InQ OutQ Up/Down  State/PfxRcd
10.1.2.2        4          500       8       8        9
0    0 00:02:42  4
10.1.3.3        4          300       8       8        9
0    0 00:02:12  4
10.1.3.130      4          300       8       8        9
0    0 00:02:11  4

R1# show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 6500
BGP table version is 9, main routing table version 9
6 network entries using 1632 bytes of memory
14 path entries using 2128 bytes of memory
5/3 BGP path/bestpath attribute entries using 1400 bytes
of memory
4 BGP AS-PATH entries using 128 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 5288 total bytes of memory
BGP activity 12/0 prefixes, 28/0 paths, scan interval 60
secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer
InQ OutQ Up/Down  State/PfxRcd
2001:DB8:ACAD:1012::2
                4          500       8       8        9
0    0 00:02:50  4
2001:DB8:ACAD:1013::3
                4          300       8       8        9
0    0 00:02:14  4
2001:DB8:ACAD:1014::3
```

```
                4         300        8       8        9
0    0 00:02:13  4
```

b. Use the **show bgp ipv4 unicast** and **show bgp ipv6 unicast** commands to view the specified BGP tables. Note that R1 has multiple paths to each destination network. Take note of the next hop address for the destination networks marked with the “>” symbol.

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf
Weight Path
 *>   192.168.1.0/27    0.0.0.0                  0
32768 i
 *>   192.168.1.64/26   0.0.0.0                  0
32768 i
 *    192.168.2.0/27    10.1.3.130
0 300 500 i
 *>                     10.1.2.2                 0
0 500 i
 *                      10.1.3.3
0 300 500 i
 *    192.168.2.64/26   10.1.3.130
0 300 500 i
 *>                     10.1.2.2                 0
0 500 i
 *                      10.1.3.3
0 300 500 i
 *    192.168.3.0/27    10.1.3.130               0
0 300 i
 *                      10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
 *    192.168.3.64/26   10.1.3.130               0
0 300 i
 *                      10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i

R1# show bgp ipv6 unicast | begin Network
     Network          Next Hop            Metric LocPrf
Weight Path
 *>   2001:DB8:ACAD:1000::/64
                       ::                       0
32768 i
 *>   2001:DB8:ACAD:1001::/64
```

```
                      ::                        0
32768 i
 *   2001:DB8:ACAD:2000::/64
                      2001:DB8:ACAD:1013::3

0 300 500 i
 *>                    2001:DB8:ACAD:1012::2
                                                 0
0 500 i
 *                     2001:DB8:ACAD:1014::3

0 300 500 i
 *   2001:DB8:ACAD:2001::/64
                      2001:DB8:ACAD:1013::3

0 300 500 i
 *>                    2001:DB8:ACAD:1012::2
                                                 0
0 500 i
 *                     2001:DB8:ACAD:1014::3

0 300 500 i
 *>  2001:DB8:ACAD:3000::/64
                      2001:DB8:ACAD:1013::3
                                                 0
0 300 i
 *                     2001:DB8:ACAD:1012::2

0 500 300 i
 *                     2001:DB8:ACAD:1014::3
                                                 0
0 300 i
 *>  2001:DB8:ACAD:3001::/64
                      2001:DB8:ACAD:1013::3
                                                 0
0 300 i
 *                     2001:DB8:ACAD:1012::2

0 500 300 i
 *                     2001:DB8:ACAD:1014::3
                                                 0
0 300 i
```

c. Use the **show ip route bgp** and **show ipv6 route bgp** commands to view the routing tables. Note that there is only one route to each destination, and that the routes included in the routing table have the same next hop as those with the “>” symbol in the BGP tables.

```
R1# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:04:10
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:04:10
      192.168.3.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.3.0/27 [20/0] via 10.1.3.3, 00:04:09
B        192.168.3.64/26 [20/0] via 10.1.3.3, 00:04:09
```

```
R1# show ipv6 route bgp
IPv6 Routing Table - default - 15 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user
Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS
summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND
Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI -
OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF
NSSA ext 1
       ON2 - OSPF NSSA ext 2, a - Application
B   2001:DB8:ACAD:2000::/64 [20/0]
     via FE80::2:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:2001::/64 [20/0]
     via FE80::2:1, GigabitEthernet0/0/0
B   2001:DB8:ACAD:3000::/64 [20/0]
     via FE80::3:2, Serial0/1/0
B   2001:DB8:ACAD:3001::/64 [20/0]
     via FE80::3:2, Serial0/1/0
```

Part 3: Configure and Verify BGP Path Manipulation Settings on all Routers

In Part 3, you will configure path manipulation tools for BGP. The way these tools are being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: Configure ACL-based route filtering.

In this step, you will configure R3 so that it only sends ASN300 networks to R1; it will not tell R1 that it knows about the networks in ASN200.

a. On R1, issue the command **show bgp ipv4 unicast | i 300** to see what prefixes ASN300 is sharing via BGP. Take note of those prefixes that do not originate in ASN300.

```
R1# show bgp ipv4 unicast | i 300
 *     192.168.2.0/27   10.1.3.3
0 300 500 i
 *                      10.1.3.130
0 300 500 i
 *     192.168.2.64/26  10.1.3.3
0 300 500 i
 *                      10.1.3.130
0 300 500 i
 *     192.168.3.0/27   10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
 *                      10.1.3.130               0
0 300 i
 *     192.168.3.64/26  10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
 *                      10.1.3.130               0
0 300 i
```

b. On R3, configure an access list designed to match the source address and mask of the networks belonging to ASN300:

```
R3(config)# ip access-list extended ALLOWED_TO_R1
R3(config-ext-nacl)# permit ip 192.168.3.0 0.0.0.0
255.255.255.224 0.0.0.0
R3(config-ext-nacl)# permit ip 192.168.3.64 0.0.0.0
255.255.255.192 0.0.0.0
R3(config-ext-nacl)# exit
```

c. On R3, apply the ALLOWED_TO_R1 ACL as a distribute list to the IPv4 neighbor adjacencies with R1.

```
R3(config)# router bgp 300
R3(config-router)# address-family ipv4 unicast
R3(config-router-af)# neighbor 10.1.3.1 distribute-list
ALLOWED_TO_R1 out
R3(config-router-af)# neighbor 10.1.3.129 distribute-list
ALLOWED_TO_R1 out
```

```
R3(config-router-af)# end
```

d. Perform a reset of the IPv4 adjacency with R1 for the outbound traffic without tearing down the session.

```
R3# clear bgp ipv4 unicast 6500 out
```

e. On R1, issue the command **show bgp ipv4 unicast | i 300** to see what prefixes routes ASN300 is now sharing via BGP. All of the prefixes should now originate in ASN300:

```
R1# show bgp ipv4 unicast | i 300
 *     192.168.3.0/27   10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
 *                      10.1.3.130               0
0 300 i
 *     192.168.3.64/26  10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
 *                      10.1.3.130               0
0 300 i
```

Step 2: Configure prefix-list-based route filtering.

In this step, you will configure R1 so that it only accepts ASN500 networks from R2; it will not accept information about ASN300 networks from R2.

a. On R1, issue the command **show bgp ipv4 unicast | begin 192.168.3** to see what prefixes ASN500 is sharing via BGP. Take note of those prefixes that do not originate in ASN500.

```
R1# show bgp ipv4 unicast | begin 192.168.3
 *     192.168.3.0/27   10.1.3.130               0
0 300 i
 *                      10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
 *     192.168.3.64/26  10.1.3.130               0
0 300 i
 *                      10.1.2.2
0 500 300 i
 *>                     10.1.3.3                 0
0 300 i
```

b. On R1, configure a prefix list designed to match the source address and mask of networks belonging to ASN500.

```
R1(config)# ip prefix-list ALLOWED_FROM_R2 seq 5 permit
192.168.2.0/24 le 27
```

c. Apply the ALLOWED_FROM_R2 prefix list to the IPv4 neighbor adjacencies for R2.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.2.2 prefix-list
ALLOWED_FROM_R2 in
R1(config-router-af)# end
```

d. Perform a reset of the IPv4 adjacency with R2 for the inbound traffic without tearing down the session.

```
R1# clear bgp ipv4 unicast 500 in
```

e. On R1, issue the command **show bgp ipv4 unicast | i 500** to see what prefixes routes ASN500 is now sharing via BGP. All of the prefixes should now originate in ASN500.

```
R1# show bgp ipv4 unicast | i 500
 *>   192.168.2.0/27   10.1.2.2                 0
0 500 i
 *>   192.168.2.64/26  10.1.2.2                 0
0 500 i
```

Step 3: Configure an AS-PATH ACL to filter routes being advertised.

In this step, you will configure R1 so that it only sends ASN100 networks to R2; it will not forward information about prefixes from any other ASN to ASN500.

a. On R2, issue the command **show bgp ipv4 unicast | begin Network** to see what prefixes ASN6500 is sharing via BGP. Take note of those prefixes that do not originate in ASN6500. Advertising these routes could set ASN6500 up as a transit AS, and that is not a desirable scenario.

```
R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf
Weight Path
 *    192.168.1.0/27   10.2.3.3
0 300 6500 i
 *>                    10.1.2.1                 0
0 6500 i
 *    192.168.1.64/26  10.2.3.3
0 300 6500 i
 *>                    10.1.2.1                 0
0 6500 i
 *>   192.168.2.0/27   0.0.0.0                  0
32768 i
```

```
 *>   192.168.2.64/26  0.0.0.0                  0
32768 i
 *    192.168.3.0/27   10.1.2.1
0 6500 300 i
 *>                    10.2.3.3                 0
0 300 i
 *    192.168.3.64/26  10.1.2.1
0 6500 300 i
 *>                    10.2.3.3                 0
0 300 i
```

b. On R1, configure AS-PATH ACL to match the routes from the local ASN.

```
R1(config)# ip as-path access-list 1 permit ^$
```

c. On R1, apply the AS-PATH ACL as a filter-list on the adjacency configured with R2.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.2.2 filter-list 1 out
R1(config-router-af)# end
```

d. On R1, perform a reset of the IPv4 adjacency with R2 for the outbound traffic without tearing down the session.

```
R1# clear bgp ipv4 unicast 500 out
```

e. On R2, issue the command **show bgp ipv4 unicast | i 6500** to see what prefixes routes ASN6500 is now sharing via BGP. All of the prefixes should now originate in ASN6500.

```
R2# show bgp ipv4 unicast | i 6500
 *    192.168.1.0/27   10.2.3.3
0 300 6500 i
 *>                    10.1.2.1                 0
0 6500 i
 *    192.168.1.64/26  10.2.3.3
0 300 6500 i
 *>                    10.1.2.1                 0
0 6500 i
```

Step 4: Configure IPv6 prefix-list-based route filtering.

In this step, you will configure R1 so that it only accepts ASN500 IPv6 networks from R2. It will not accept information about ASN300 IPv6 networks from R2.

a. On R1, issue the command **show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes** to see what IPv6 prefixes ASN500 is sharing via BGP. Take note of those IPv6 prefixes that do not originate in ASN500.

```
R1# show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2
routes
```

```
BGP table version is 9, local router ID is 1.1.1.1
Status code001s: s suppressed, d damped, h history, *
valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *>   2001:DB8:ACAD:2000::/64
                      2001:DB8:ACAD:1012::2
                                                0
0 500 i
 *>   2001:DB8:ACAD:2001::/64
                      2001:DB8:ACAD:1012::2
                                                0
0 500 i
 *    2001:DB8:ACAD:3000::/64
                      2001:DB8:ACAD:1012::2

0 500 300 i
 *    2001:DB8:ACAD:3001::/64
                      2001:DB8:ACAD:1012::2

0 500 300 i

Total number of prefixes 4
```

b. On R1, configure an IPv6 prefix list designed to match the source address and mask of networks belonging to ASN500.

```
R1(config)# ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 5
permit 2001:db8:acad:2000::/64
R1(config)# ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 10
permit 2001:db8:acad:2001::/64
```

c. Apply the IPV6_ALLOWED_FROM_R2 prefix list to the IPv6 neighbor adjacencies for R2.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv6 unicast
R1(config-router-af)# neighbor 2001:db8:acad:1012::2
prefix-list IPV6_ALLOWED_FROM_R2 in
```

```
R1(config-router-af)# end
```

d. Perform a reset of the IPv6 adjacency with R2 for the inbound traffic without tearing down the session.

```
R1# clear bgp ipv6 unicast 500 in
```

e. On R1, issue the command **show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes** to see what IPv6 prefixes routes ASN500 is now sharing via BGP. All of the IPv6 prefixes should now originate in ASN500.

```
R1# show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2
routes
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *>   2001:DB8:ACAD:2000::/64
                      2001:DB8:ACAD:1012::2
                                                0
0 500 i
 *>   2001:DB8:ACAD:2001::/64
                      2001:DB8:ACAD:1012::2
                                                0
0 500 i

Total number of prefixes 2
```

f. Configure and apply an IPv6 filter to do the same thing on the adjacency with ASN300.

Step 5: Configure BGP path attribute manipulation to effect routing.

In this step, you will configure R1 so that it prefers the next-hop address of 192.168.3.130 over 192.168.3.3, which would normally be the preferred path to ASN300 networks. You will do this by using a prefix list to identify the destination networks and then use a route map to match the prefix list and set the matched networks to have a local preference of 250.

a. On R1, issue the command **show ip route bgp** and take note of the next hop addresses for the 192.168.3.0/27 and 192.168.3.64/26 networks. Then issue the command **show bpg ipv4 unicast** and note that the 10.1.3.130 is a valid next hop (It's just not the *best* next hop, according to the BGP path selection algorithm.) Lastly, issue the command **show bgp ipv4 unicast 192.168.3.0** to see details about all the paths available and which one was selected.

```
R1# show bgp ipv4 unicast 192.168.3.0
BGP routing table entry for 192.168.3.0/27, version 8
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  300
    10.1.3.3 from 10.1.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid,
external, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  300
    10.1.3.130 from 10.1.3.130 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid,
external
      rx pathid: 0, tx pathid: 0
```

b. On R1, configure a prefix list designed to match the source address and mask of networks belonging to ASN300.

```
R1(config)# ip prefix-list PREFERRED_IPV4_PATH seq 5
permit 192.168.3.0/24 le 27
```

c. Create a route-map named USE_THIS_PATH_FOR_IPV4 that matches on the prefix list you just created and sets the local preference to 250.

```
R1(config)# route-map USE_THIS_PATH_FOR_IPV4 permit 10
R1(config)# match ip address prefix-list
PERFERRED_IPV4_PATH
R1(config)# set local-preference 250
```

d. Next, apply this route map to the BGP neighbor 10.1.3.130.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.3.130 route-map
USE_THIS_PATH_FOR_IPV4 in
R1(config-router-af)# end
```

e. Perform a reset of the IPv4 adjacency with R3 for the inbound traffic without tearing down the session.

```
R1# clear bgp ipv4 unicast 300 in
```

f. On R1, issue the command **show ip route bgp** and take note of the next hop addresses for the 192.168.3.0/27 and 192.168.3.64/26 networks; it should be 10.1.3.130 for both. Issue the command **show bgp ipv4 unicast** and you should see the local preference value in the appropriate column.

```
R1# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:35:17
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:35:17
      192.168.3.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.3.0/27 [20/0] via 10.1.3.130, 00:00:08
B        192.168.3.64/26 [20/0] via 10.1.3.130, 00:00:08

R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf
Weight Path
 *>   192.168.1.0/27   0.0.0.0                  0
32768 i
 *>   192.168.1.64/26  0.0.0.0                  0
32768 i
 *>   192.168.2.0/27   10.1.2.2                 0
0 500 i
 *>   192.168.2.64/26  10.1.2.2                 0
0 500 i
 *    192.168.3.0/27   10.1.3.3                 0
0 300 i
 *>                    10.1.3.130               0    250
0 300 i
 *    192.168.3.64/26  10.1.3.3                 0
0 300 i
 *>                    10.1.3.130               0    250
0 300 i
```

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs - Final

Router R1

```
R1# show run
Building configuration...


Current configuration : 5819 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
```

```
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.224
 ipv6 address FE80::1:4 link-local
 ipv6 address 2001:DB8:ACAD:1000::1/64
!
interface Loopback1
 ip address 192.168.1.65 255.255.255.192
 ipv6 address FE80::1:5 link-local
 ipv6 address 2001:DB8:ACAD:1001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.1.2.1 255.255.255.0
 negotiation auto
 ipv6 address FE80::1:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::1/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface Serial0/1/0
 ip address 10.1.3.1 255.255.255.128
 ipv6 address FE80::1:2 link-local
 ipv6 address 2001:DB8:ACAD:1013::1/64
!
interface Serial0/1/1
 ip address 10.1.3.129 255.255.255.128
 ipv6 address FE80::1:3 link-local
 ipv6 address 2001:DB8:ACAD:1014::1/64
!
router bgp 6500
```

```
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.1.2.2 remote-as 500
 neighbor 10.1.3.3 remote-as 300
 neighbor 10.1.3.130 remote-as 300
 neighbor 2001:DB8:ACAD:1012::2 remote-as 500
 neighbor 2001:DB8:ACAD:1013::3 remote-as 300
 neighbor 2001:DB8:ACAD:1014::3 remote-as 300
 !
 address-family ipv4
  network 192.168.1.0 mask 255.255.255.224
  network 192.168.1.64 mask 255.255.255.192
  neighbor 10.1.2.2 activate
  neighbor 10.1.2.2 prefix-list ALLOWED_FROM_R2 in
  neighbor 10.1.2.2 filter-list 1 out
  neighbor 10.1.3.3 activate
  neighbor 10.1.3.130 activate
  neighbor 10.1.3.130 route-map USE_THIS_PATH_FOR_IPV4 in
exit-address-family
 !
 address-family ipv6
  network 2001:DB8:ACAD:1000::/64
  network 2001:DB8:ACAD:1001::/64
  neighbor 2001:DB8:ACAD:1012::2 activate
  neighbor 2001:DB8:ACAD:1012::2 prefix-list
IPV6_ALLOWED_FROM_R2 in
  neighbor 2001:DB8:ACAD:1013::3 activate
  neighbor 2001:DB8:ACAD:1014::3 activate
exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
ip as-path access-list 1 permit ^$
!
ip prefix-list ALLOWED_FROM_R2 seq 5 permit 192.168.2.0/24 le
27
!
ip prefix-list PREFERRED_IPV4_PATH seq 5 permit 192.168.3.0/24
le 27
!
```

```
ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 5 permit
2001:DB8:ACAD:2000::/64
ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 10 permit
2001:DB8:ACAD:2001::/64
!
route-map USE_THIS_PATH_FOR_IPV4 permit 10
 match ip address prefix-list PERFERRED_IPV4_PATH
 set local-preference 250
!
control-plane
!
banner motd ^C This is R1, BGP Path Manipulation Lab ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

Router R2

```
R2# show run
Building configuration...


Current configuration : 4600 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R2
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
!
no ip domain lookup
!
ip dhcp pool webuidhcp
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.2.1 255.255.255.224
 ipv6 address FE80::2:3 link-local
 ipv6 address 2001:DB8:ACAD:2000::1/64
!
interface Loopback1
 ip address 192.168.2.65 255.255.255.192
 ipv6 address FE80::2:4 link-local
 ipv6 address 2001:DB8:ACAD:2001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.1.2.2 255.255.255.0
 negotiation auto
 ipv6 address FE80::2:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::2/64
!
interface GigabitEthernet0/0/1
 ip address 10.2.3.2 255.255.255.0
 negotiation auto
 ipv6 address FE80::2:2 link-local
 ipv6 address 2001:DB8:ACAD:1023::2/64
!
router bgp 500
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.1.2.1 remote-as 6500
 neighbor 10.2.3.3 remote-as 300
 neighbor 2001:DB8:ACAD:1012::1 remote-as 6500
 neighbor 2001:DB8:ACAD:1023::3 remote-as 300
 !
 address-family ipv4
  network 192.168.2.0 mask 255.255.255.224
  network 192.168.2.64 mask 255.255.255.192
  neighbor 10.1.2.1 activate
  neighbor 10.2.3.3 activate
exit-address-family
 !
 address-family ipv6
  network 2001:DB8:ACAD:2000::/64
  network 2001:DB8:ACAD:2001::/64
  neighbor 2001:DB8:ACAD:1012::1 activate
  neighbor 2001:DB8:ACAD:1023::3 activate
 exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
control-plane
!
banner motd ^C This is R2, BGP Path Manipulation Lab ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

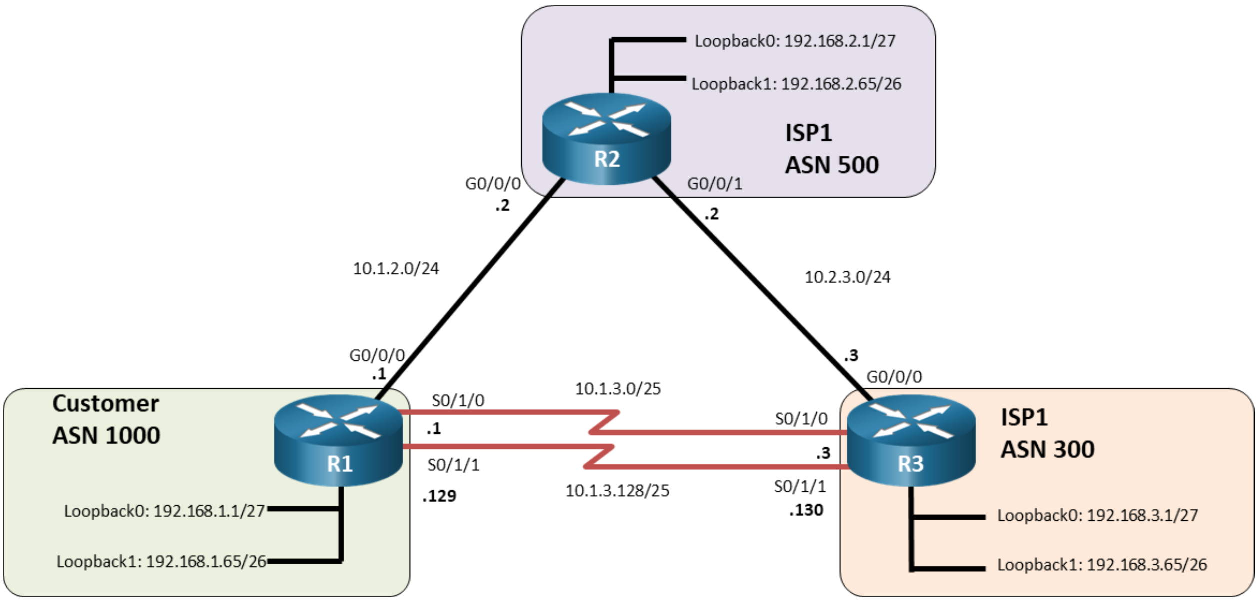
Router R3

```
R3# show run
Building configuration...
```

```
Current configuration : 5180 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
ip dhcp po ol webuidhcp
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.3.1 255.255.255.224
 ipv6 address FE80::3:4 link-local
 ipv6 address 2001:DB8:ACAD:3000::1/64
!
interface Loopback1
 ip address 192.168.3.65 255.255.255.192
 ipv6 address FE80::3:5 link-local
 ipv6 address 2001:DB8:ACAD:3001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.2.3.3 255.255.255.0
 negotiation auto
 ipv6 address FE80::3:1 link-local
 ipv6 address 2001:DB8:ACAD:1023::3/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface Serial0/1/0
 ip address 10.1.3.3 255.255.255.128
 ipv6 address FE80::3:2 link-local
 ipv6 address 2001:DB8:ACAD:1013::3/64
!
interface Serial0/1/1
 ip address 10.1.3.130 255.255.255.128
 ipv6 address FE80::3:3 link-local
 ipv6 address 2001:DB8:ACAD:1014::3/64
!
router bgp 300
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.1.3.1 remote-as 6500
 neighbor 10.1.3.129 remote-as 6500
 neighbor 10.2.3.2 remote-as 500
 neighbor 2001:DB8:ACAD:1013::1 remote-as 6500
 neighbor 2001:DB8:ACAD:1014::1 remote-as 6500
 neighbor 2001:DB8:ACAD:1023::2 remote-as 500
 !
 address-family ipv4
  network 192.168.3.0 mask 255.255.255.224
  network 192.168.3.64 mask 255.255.255.192
  neighbor 10.1.3.1 activate
  neighbor 10.1.3.1 distribute-list ALLOWED_TO_R1 out
  neighbor 10.1.3.129 activate
  neighbor 10.1.3.129 distribute-list ALLOWED_TO_R1 out
  neighbor 10.2.3.2 activate
exit-address-family
 !
 address-family ipv6
  network 2001:DB8:ACAD:3000::/64
  network 2001:DB8:ACAD:3001::/64
  neighbor 2001:DB8:ACAD:1013::1 activate
  neighbor 2001:DB8:ACAD:1014::1 activate
  neighbor 2001:DB8:ACAD:1023::2 activate
 exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
ip access-list extended ALLOWED_TO_R1
 permit ip host 192.168.3.0 host 255.255.255.224
 permit ip host 192.168.3.64 host 255.255.255.192
!
control-plane
!
banner motd ^C This is R3, BGP Path Manipulation Lab ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

[Title]

Topology



Addressing Table

Device	Interface	IPv4 Address
R1	G0/0/0	10.1.2.1/24
	S0/1/0	10.1.3.1/25
	S0/1/1	10.1.3.129/25
	Loopback0	192.168.1.1/27
	Loopback1	192.168.1.65/26
R2	G0/0/0	10.1.2.2/24
	G0/0/1	10.2.3.2/24
	Loopback0	192.168.2.1/27
	Loopback1	192.168.2.65/26
R3	G0/0/0	10.2.3.3/24
	S0/1/0	10.1.3.3/25
	S0/1/1	10.1.3.130/25
	Loopback0	192.168.3.1/27
	Loopback1	192.168.3.65/26

Objectives

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

Part 2: Configure and Verify eBGP for IPv4 on all Routers

Part 3: Configure and Verify Route Summarization and Atomic Aggregate

Part 4: Configure and Verify Route Summarization with Atomic Aggregate and AS-Set

Part 5: Configure and Verify the Advertising of a Default Route

Background / Scenario

In this lab you will configure eBGP for IPv4.

Note: This lab is an exercise in developing, deploying, and verifying various path manipulation tools for BGP, and does not reflect networking best practices.

Note: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the routers and switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 PC (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each router.

a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing. A command list for each router is provided below.

Router R1

```
hostname R1
no ip domain lookup
line con 0
logging sync
exec-time 0 0
exit
interface Loopback0
 ip address 192.168.1.1 255.255.255.224
 no shut
 exit
interface Loopback1
 ip address 192.168.1.65 255.255.255.192
 no shut
 exit
interface GigabitEthernet0/0/0
 ip address 10.1.2.1 255.255.255.0
 no shut
 exit
interface Serial0/1/0
 ip address 10.1.3.1 255.255.255.128
 no shut
 exit
interface Serial0/1/1
 ip address 10.1.3.129 255.255.255.128
 no shut
 exit
```

Router R2

```
hostname R2
no ip domain lookup
line con 0
logging sync
exec-time 0 0
exit
interface Loopback0
 ip address 192.168.2.1 255.255.255.224
```

```
  no shut
  exit
 interface Loopback1
  ip address 192.168.2.65 255.255.255.192
  no shut
  exit
 interface GigabitEthernet0/0/0
  ip address 10.1.2.2 255.255.255.0
  no shut
  exit
 interface GigabitEthernet0/0/1
  ip address 10.2.3.2 255.255.255.0
  no shut
  exit
```

Router R3

```
 hostname R3
 no ip domain lookup
 line con 0
  logging sync
  exec-time 0 0
  exit
 interface Loopback0
  ip address 192.168.3.1 255.255.255.224
  no shut
  exit
 interface Loopback1
  ip address 192.168.3.65 255.255.255.192
  no shut
  exit
 interface GigabitEthernet0/0/0
  ip address 10.2.3.3 255.255.255.0
  negotiation auto
  no shut
  exit
 interface Serial0/1/0
  ip address 10.1.3.3 255.255.255.128
  no shut
  exit
 interface Serial0/1/1
  ip address 10.1.3.130 255.255.255.128
  no shut
 exit
```

b. Save the running configuration to startup-config.

Part 2: Configure and Verify eBGP for IPv4 on all Routers

Step 1: Implement BGP and neighbor relationships on R1.

a. Enter BGP configuration mode from global configuration mode, specifying AS 1000.

```
R1(config)# router bgp 1000
```

b. Configure the BGP router-id for R1.

```
R1(config-router)# bgp router-id 1.1.1.1
```

c. Based on the topology diagram, configure all the designated neighbors for R1.

```
R1(config-router)# neighbor 10.1.2.2 remote-as 500
R1(config-router)# neighbor 10.1.3.3 remote-as 300
R1(config-router)# neighbor 10.1.3.130 remote-as 300
```

d. Configure R1 to advertise the IPv4 prefixes local to ASN 1000.

```
R1(config-router)# network 192.168.1.0 mask
255.255.255.224
R1(config-router)# network 192.168.1.64 mask
255.255.255.192
```

Step 2: Implement BGP and neighbor relationships on R2.

a. Enter BGP configuration mode from global configuration mode, specifying AS 500.

```
R2(config)# router bgp 500
```

b. Configure the BGP router-id for R2.

```
R2(config-router)# bgp router-id 2.2.2.2
```

c. Based on the topology diagram, configure all the designated neighbors for R2.

```
R2(config-router)# neighbor 10.1.2.1 remote-as 1000
R2(config-router)# neighbor 10.2.3.3 remote-as 300
```

d. Configure R2 to advertise the IPv4 prefixes local to ASN 500.

```
R2(config-router)# network 192.168.2.0 mask
255.255.255.224
R2(config-router)# network 192.168.2.64 mask
255.255.255.192
```

Step 3: Implement BGP and neighbor relationships on R3.

a. Enter BGP configuration mode from global configuration mode, specifying AS 300.

```
R3(config)# router bgp 300
```

b. Configure the BGP router-id for R3.

```
R3(config-router)# bgp router-id 3.3.3.3
```

c. Unlike the configuration on R1 and R2, disable the default IPv4 unicast behavior.

```
R3(config-router)# no bgp default ipv4-unicast
```

The default behavior in IOS is **bgp default ipv4-unicast**. Routers R1 and R2 were configured using this default behavior. The **bgp default ipv4-unicast** command enables the automatic exchange of IPv4 address family prefixes. When this command is disabled using **no bgp default ipv4-unicast**, bgp neighbors must be activated within IPv4 address family (AF) configuration mode. BGP **network** commands must also be configured within IPv4 AF mode.

d. Based on the topology diagram, configure all the designated neighbors for R3.

```
R3(config-router)# neighbor 10.2.3.2 remote-as 500
R3(config-router)# neighbor 10.1.3.1 remote-as 1000
R3(config-router)# neighbor 10.1.3.129 remote-as 1000
```

Step 4: Verifying BGP neighbor relationships.

a. Examine the routing tables on each router. Notice that R1 and R2 are receiving BGP prefixes from each other but not receiving BGP prefixes from R3. And R3 is not receiving any prefixes from R1 or R2. This is because R3 was configured using **no bgp default ipv4-unicast** and the interfaces must be activated within IPv4 address configuration mode.

```
R1# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:28:40
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:28:40

R2# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2
masks
```

```
B        192.168.1.0/27 [20/0] via 10.1.2.1, 00:29:41
B        192.168.1.64/26 [20/0] via 10.1.2.1, 00:29:41

R3# show ip route bgp | begin Gateway
Gateway of last resort is not set
```

b. This can be further verified by examining the BGP neighbor adjacencies on R2. Notice the BGP state between R2 and R1 is **established**, while the BGP state between R2 and R3 is **idle**.

```
R2# show ip bgp neighbors
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:35:34
  Last read 00:00:28, last write 00:00:35, hold time is
180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
<output omitted>

BGP neighbor is 10.2.3.3,  remote AS 300, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle, down for never
  Neighbor sessions:
    0 active, is not multisession capable (disabled)
<output omitted>
```

c. The interfaces on R3 need to be activated in IPv4 AF configuration mode. The **neighbor activate** command in IPv4 AF configuration mode is required to enable the exchange of BGP information between neighbors. This will enable R3 to form an established neighbor adjacency with both R1 and R2. Additionally, because **bgp default ipv4-unicast** is disabled, **network** commands must be configured in IPv4 AF configuration mode.

```
R3(config-router)# address-family ipv4
R3(config-router-af)# neighbor 10.1.3.1 activate
R3(config-router-af)# neighbor 10.1.3.129 activate
R3(config-router-af)# neighbor 10.2.3.2 activate
R3(config-router-af)# network 192.168.3.0 mask
255.255.255.224
R3(config-router-af)# network 192.168.3.64 mask
255.255.255.192
```

d. Verify that all BGP speakers are receiving prefixes from their neighbors. The prefixes from R3 are highlighted in the routing tables of R1 and R2.

Note: The prefixes in the lab are for example purposes only. Most service providers do not accept prefixes larger than /24 for IPv4 (/25 through /32).

```
R1# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:51:09
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:51:09
      192.168.3.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.3.0/27 [20/0] via 10.1.3.3, 00:01:43
B        192.168.3.64/26 [20/0] via 10.1.3.3, 00:01:43

R2# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.1.0/27 [20/0] via 10.1.2.1, 00:51:17
B        192.168.1.64/26 [20/0] via 10.1.2.1, 00:51:17
      192.168.3.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.3.0/27 [20/0] via 10.2.3.3, 00:01:51
B        192.168.3.64/26 [20/0] via 10.2.3.3, 00:01:51

R3# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.1.0/27 [20/0] via 10.1.3.1, 00:02:11
B        192.168.1.64/26 [20/0] via 10.1.3.1, 00:02:11
      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.2.3.2, 00:02:11
B        192.168.2.64/26 [20/0] via 10.2.3.2, 00:02:11
```

e. Verify that the BGP state between R2 and R3 has now been **established**.

```
R2# show ip bgp neighbors | begin BGP neighbor is
10.2.3.3
BGP neighbor is 10.2.3.3,  remote AS 300, external link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 00:12:16
  Last read 00:00:37, last write 00:00:52, hold time is
180, keepalive interval is 60 seconds
```

```
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
<output omitted>
```

Step 5: Examining the running-configs.

Examine the running-configs on all three routers. Because router R3 was configured using **no bgp default ipv4-unicast** command, notice that the network commands were automatically entered under the IPv4 AF. This is the same configuration mode where the neighbors were activated to exchange BGP information.

```
R1# show running-config | section bgp
router bgp 1000
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 network 192.168.1.0 mask 255.255.255.224
 network 192.168.1.64 mask 255.255.255.192
 neighbor 10.1.2.2 remote-as 500
 neighbor 10.1.3.3 remote-as 300
 neighbor 10.1.3.130 remote-as 300

R2# show running-config | section bgp
router bgp 500
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 network 192.168.2.0 mask 255.255.255.224
 network 192.168.2.64 mask 255.255.255.192
 neighbor 10.1.2.1 remote-as 1000
 neighbor 10.2.3.3 remote-as 300

R3# show running-config | section bgp
router bgp 300
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.1.3.1 remote-as 1000
 neighbor 10.1.3.129 remote-as 1000
 neighbor 10.2.3.2 remote-as 500
 !
 address-family ipv4
  network 192.168.3.0 mask 255.255.255.224
  network 192.168.3.64 mask 255.255.255.192
  neighbor 10.1.3.1 activate
  neighbor 10.1.3.129 activate
  neighbor 10.2.3.2 activate
 exit-address-family
```

Step 6: Verifying BGP operations.

a. To verify the BGP operation on R2, issue the **show ip bgp** command.

```
R2# show ip bgp
BGP table version is 11, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *    192.168.1.0/27   10.2.3.3
0 300 1000 i
 *>                    10.1.2.1                 0
0 1000 i
 *    192.168.1.64/26  10.2.3.3
0 300 1000 i
 *>                    10.1.2.1                 0
0 1000 i
 *>   192.168.2.0/27   0.0.0.0                  0
32768 i
 *>   192.168.2.64/26  0.0.0.0                  0
32768 i
 *>   192.168.3.0/27   10.2.3.3                 0
0 300 i
 *                     10.1.2.1
0 1000 300 i
 *>   192.168.3.64/26  10.2.3.3                 0
0 300 i
 *                     10.1.2.1
0 1000 300 i
```

Questions:

What does the * at the beginning of an entry indicate?

Type your answers here.

What does the angle bracket (>) in an entry indicate?

Type your answers here.

What is the address of the preferred next hop router to reach the 192.168.1.0/27 network? Explain.

Type your answers here.

How can you verify that 10.1.2.1 is the next hop router used to reach 192.168.1.0/27?

Type your answers here.

What does a next hop of 0.0.0.0 indicate?

Type your answers here.

b. Use the **show ip bgp** *ip-prefix* command to display all the paths for a specific route and the BGP path attributes for that route.

```
R2# show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/27, version 14
Paths: (2 available, best #2, table default)
 Advertised to update-groups:
    1
 Refresh Epoch 1
 300 1000
  10.2.3.3 from 10.2.3.3 (3.3.3.3)
   Origin IGP, localpref 100, valid, external
   rx pathid: 0, tx pathid: 0
 Refresh Epoch 2
 1000
  10.1.2.1 from 10.1.2.1 (1.1.1.1)
   Origin IGP, metric 0, localpref 100, valid,
external, best
   rx pathid: 0, tx pathid: 0x0
```

Question:

What is the IPv4 address of the next hop router with the best path?

Type your answers here.

c. Examine the BGP neighbor relationships on R2 using the **show ip bgp neighbors** command.

```
R2# show ip bgp neighbors
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
 BGP version 4, remote router ID 1.1.1.1
 BGP state = Established, up for 00:00:51
 Last read 00:00:00, last write 00:00:51, hold time is
180, keepalive interval is 60 seconds
```

```
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                5          5
    Keepalives:             2          3
    Route Refresh:          0          0
    Total:                 10         11
<output omitted>

BGP neighbor is 10.2.3.3,  remote AS 300, external link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 16:23:45
  Last read 00:00:29, last write 00:00:51, hold time is
180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                9          5
    Keepalives:          1082       1088
    Route Refresh:          0          0
    Total:               1096       1096
  Do log neighbor state changes (via global
configuration)
  Default minimum time between advertisement runs is 30
seconds
<output omitted>
```

Questions:

How many neighbors does R2 have and what are their router IDs?

Type your answers here.

What is the BGP state of both neighbors?

Type your answers here.

What are the keepalive and hold time value for both neighbors?

Type your answers here.

Part 3: Configure and Verify Route Summarization and Atomic Aggregate

Step 1: Configure route summarization using atomic aggregate.

Summarizing prefixes conserves router resources and accelerates best-path calculation by reducing the size of the table. Summarization can be configured either for prefixes originated by the AS or prefixes received from downstream providers. Summarization also provides the benefits of stability by hiding flapping routes or having to install new prefixes when they are contained within a summary.

Although AS 1000 only has two prefixes 192.168.1.0/27 and 192.168.1.64/26, this customer has been allocated the entire 192.168.1.0/24 prefix. R3 in AS 300 has two prefixes 192.168.3.0/27 and 192.168.3.64/26 but has been allocated the entire 192.168.3.0/24 prefix.

Configure R1 and R3 to advertise a summary or aggregate route using the **aggregate-address** command. The **summary-only** option suppresses the specific prefixes that are summarized from also being advertised. Notice that this command is configured in **address-family ipv4** configuration mode on R3.

```
R1(config)# router bgp 1000
R1(config-router)# aggregate-address 192.168.1.0
255.255.255.0 summary-only

R3(config)# router bgp 300
R3(config-router)# address-family ipv4
```

```
R3(config-router-af)# aggregate-address 192.168.3.0
255.255.255.0 summary-only
```

Step 2: Verify route summarization using atomic aggregate.

a. Examine the routing tables on each router to verify the route summarization for the two prefixes. Verify that R1 and R3 are each receiving the summary route from the other router. Verify that R2 is receiving aggregate routes from both R1 and R3.

```
R1# show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 5 subnets, 4
masks
B        192.168.1.0/24 [200/0], 00:27:47, Null0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 13:34:31
B        192.168.2.64/26 [20/0] via 10.1.2.2, 13:34:31
B     192.168.3.0/24 [20/0] via 10.1.3.3, 00:26:01

R2# show ip route bgp | begin Gateway
Gateway of last resort is not set

B     192.168.1.0/24 [20/0] via 10.1.2.1, 00:33:53
B     192.168.3.0/24 [20/0] via 10.2.3.3, 00:32:08

R3# show ip route bgp | begin Gateway
Gateway of last resort is not set

B     192.168.1.0/24 [20/0] via 10.1.3.1, 00:36:52
      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.2.3.2, 02:10:48
B        192.168.2.64/26 [20/0] via 10.2.3.2, 02:10:48
      192.168.3.0/24 is variably subnetted, 5 subnets, 4
masks
B        192.168.3.0/24 [200/0], 00:35:07, Null0
```

Question:

Why do R1 and R3 contain an entry with a next hop address of Null0? What is the result of having this Null0 route in the routing table?

Type your answers here.

b. Examine the BGP table on router R2 to verify the route summarization. When a prefix has the default classful mask, the subnet mask is not displayed. Both 192.168.1.0 and 192.168.3.0 prefixes have a /24 prefix length which would be the default mask for a Class C address.

```
R2# show ip bgp
BGP table version is 69, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
<output omitted>

     Network          Next Hop            Metric LocPrf
Weight Path
 *     192.168.1.0      10.2.3.3
0 300 1000 i
 *>                     10.1.2.1                 0
0 1000 i
 *>    192.168.2.0/27   0.0.0.0                  0
32768 i
 *>    192.168.2.64/26  0.0.0.0                  0
32768 i
 *     192.168.3.0      10.1.2.1
0 1000 300 i
 *>                     10.2.3.3                 0
0 300 i
```

c. Examine the BGP table on routers R2 and R3 and verify that each router is receiving the summary route from the other router.

```
R1# show ip bgp
BGP table version is 69, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
<output omitted>

     Network          Next Hop            Metric LocPrf
Weight Path
 s>    192.168.1.0/27   0.0.0.0                  0
32768 i
 *>    192.168.1.0      0.0.0.0
32768 i
 s>    192.168.1.64/26  0.0.0.0                  0
32768 i
 *     192.168.2.0/27   10.1.3.130
0 300 500 i
 *                      10.1.3.3
0 300 500 i
```

```
 *>                    10.1.2.2                 0
0 500 i
 *    192.168.2.64/26  10.1.3.130
0 300 500 i
 *                     10.1.3.3
0 300 500 i
 *>                    10.1.2.2                 0
0 500 i
 *    192.168.3.0      10.1.2.2
0 500 300 i
 *                     10.1.3.130               0
0 300 i
 *>                    10.1.3.3                 0
0 300 i

R3# show ip bgp
BGP table version is 22, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid,
> best, i - internal,
              r RIB-failure, S Stale, m multipath, b
backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf
Weight Path
 *    192.168.1.0      10.2.3.2
0 500 1000 i
 *>                    10.1.3.1                 0
0 1000 i
 *                     10.1.3.129               0
0 1000 i
 *    192.168.2.0/27   10.1.3.1
0 1000 500 i
 *                     10.1.3.129
0 1000 500 i
 *>                    10.2.3.2                 0
0 500 i
 *    192.168.2.64/26  10.1.3.1
0 1000 500 i
 *                     10.1.3.129
0 1000 500 i
```

```
 *>                     10.2.3.2                 0
0 500 i
 s>    192.168.3.0/27   0.0.0.0                  0
32768 i
 *>    192.168.3.0      0.0.0.0
32768 i
 s>    192.168.3.64/26  0.0.0.0                  0
32768 i
```

Question:

Why do two of the entries have the status code of "s"? Specifically, this is the result of what command or option that was configured on these two routers?

Type your answers here.

d. Examine the explicit 192.168.1.0 prefix entry in R2's BGP table. The route's NLRI information indicates that the route was aggregated in AS 1000 by router with the RID 1.1.1.1.

```
R2# show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/24, version 45
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  300 1000, (aggregated by 1000 1.1.1.1)
    10.2.3.3 from 10.2.3.3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external, atomic-
aggregate
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 2
  1000, (aggregated by 1000 1.1.1.1)
    10.1.2.1 from 10.1.2.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid,
external, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0
```

Part 4: Configure and Verify Route Summarization with Atomic Aggregate and AS-Set

Step 1: Configure route summarization using atomic aggregate and AS-Set.

a. Shut down both serial interfaces on R1. This will create a single path from R1 (AS 1000) to R2 (AS 500) to R3 (AS 300).

```
R1(config)# interface s0/1/0
R1(config-if)# shutdown
```

```
R1(config-if)# exit
R1(config)# interface s0/1/1
R1(config-if)# shutdown
```

b. Remove route aggregation previously configured on R1.

```
R1(config)# router bgp 1000
R1(config-router)# no aggregate-address 192.168.1.0
255.255.255.0 summary-only
```

c. Verify that R3 is now receiving the non-summarized prefixes 192.168.1.0/27 and 192.168.1.64/26.

```
R3# show ip route 192.168.1.0
Routing entry for 192.168.1.0/24, 2 known subnets
  Variably subnetted with 2 masks
B        192.168.1.0/27 [20/0] via 10.2.3.2, 00:01:26
B        192.168.1.64/26 [20/0] via 10.2.3.2, 00:01:26
```

d. On R2, summarize the prefixes 192.168.1.0/27 and 192.168.1.64/26 received from R1 as 192.168.1.0/24.

```
R2(config)# router bgp 500
R2(config-router)# aggregate-address 192.168.1.0
255.255.255.0 summary-only
```

Step 2: Verify route summarization using atomic aggregate and AS-Set.

a. Verify that R3 is receiving the aggregated prefix 192.168.1.0/24.

```
R3# show ip route bgp | begin Gateway
Gateway of last resort is not set

B     192.168.1.0/24 [20/0] via 10.2.3.2, 00:00:51
      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.2.3.2, 08:46:37
B        192.168.2.64/26 [20/0] via 10.2.3.2, 08:46:37
      192.168.3.0/24 is variably subnetted, 5 subnets, 4
masks
B        192.168.3.0/24 [200/0], 08:46:07, Null0
```

b. Examine R3's BGP table. Notice that the AS path only includes the AS that summarized the route, AS 500, router R2.

```
R3# show ip bgp
<output omitted>
```

```
     Network          Next Hop            Metric LocPrf
Weight Path
 *>   192.168.1.0      10.2.3.2                 0
0 500 i
 *>   192.168.2.0/27   10.2.3.2                 0
0 500 i
 *>   192.168.2.64/26  10.2.3.2                 0
0 500 i
 s>   192.168.3.0/27   0.0.0.0                  0
32768 i
 *>   192.168.3.0      0.0.0.0
32768 i
 s>   192.168.3.64/26  0.0.0.0                  0
32768 i
```

c. On R2, remove the current route aggregation for the 192.168.1.0/24 prefix and configure it again, this time using the **as-set** option.

```
R2(config)# router bgp 500
R2(config-router)# no aggregate-address 192.168.1.0
255.255.255.0 summary-only
R2(config-router)# aggregate-address 192.168.1.0
255.255.255.0 as-set summary-only
```

d. Verify that R3 is receiving the aggregated prefix 192.168.1.0/24.

```
R3# show ip route bgp | begin Gateway
Gateway of last resort is not set

B     192.168.1.0/24 [20/0] via 10.2.3.2, 00:01:35
      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.2.3.2, 08:50:02
B        192.168.2.64/26 [20/0] via 10.2.3.2, 08:50:02
      192.168.3.0/24 is variably subnetted, 5 subnets, 4
masks
B        192.168.3.0/24 [200/0], 08:49:32, Null0
```

e. Examine R3's BGP table again. Notice that the entry for 192.168.1.0 this time includes the entire AS path. The output from the **show ip bgp 192.168.1.0** command displays both AS numbers and identifies that R2 (2.2.2.2) aggregated the route.

```
R3# show ip bgp
<output omitted>
     Network          Next Hop            Metric LocPrf
Weight Path
 *>   192.168.1.0      10.2.3.2                 0
0 500 1000 i
```

```
 *>   192.168.2.0/27   10.2.3.2                 0
0 500 i
 *>   192.168.2.64/26  10.2.3.2                 0
0 500 i
 s>   192.168.3.0/27   0.0.0.0                  0
32768 i
 *>   192.168.3.0      0.0.0.0
32768 i
 s>   192.168.3.64/26  0.0.0.0                  0
32768 i
```

```
R3# show ip bgp 192.168.1.0 | begin Refresh
  Refresh Epoch 7
  500 1000, (aggregated by 500 2.2.2.2)
    10.2.3.2 from 10.2.3.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid,
external, best
      rx pathid: 0, tx pathid: 0x0
```

Part 5: Configure and Verify the Advertising of a Default Route

Step 1: Configure default route advertisement on R2.

Configure R2 to advertise a default router to R1. R2 does not necessarily have to have a default route of its own. Core internet routers that have full internet routing tables and do not require a default route are referred to as being in a default-free zone (DFZ).

```
R2(config)# router bgp 500
R2(config-router)# neighbor 10.1.2.1 default-originate
```

Step 2: Verify default route advertisement on R1.

a. Examine R1's routing table to verify that it has received a default route.

```
R1# show ip route bgp | begin Gateway
Gateway of last resort is 10.1.2.2 to network 0.0.0.0

B*    0.0.0.0/0 [20/0] via 10.1.2.2, 00:00:37
      192.168.2.0/24 is variably subnetted, 2 subnets, 2
masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 21:24:43
B        192.168.2.64/26 [20/0] via 10.1.2.2, 21:24:43
B     192.168.3.0/24 [20/0] via 10.1.2.2, 12:41:58
```

b. Examine R1's BGP table to verify that it has received a default route.

```
R1# show ip bgp
```

```
<output omitted>

     Network          Next Hop            Metric LocPrf
Weight Path
 *>   0.0.0.0          10.1.2.2
0 500 i
 *>   192.168.1.0/27   0.0.0.0                  0
32768 i
 *>   192.168.1.64/26  0.0.0.0                  0
32768 i
 *>   192.168.2.0/27   10.1.2.2                 0
0 500 i
 *>   192.168.2.64/26  10.1.2.2                 0
0 500 i
 *>   192.168.3.0      10.1.2.2
0 500 300 i
```

Router Interface Summary Table

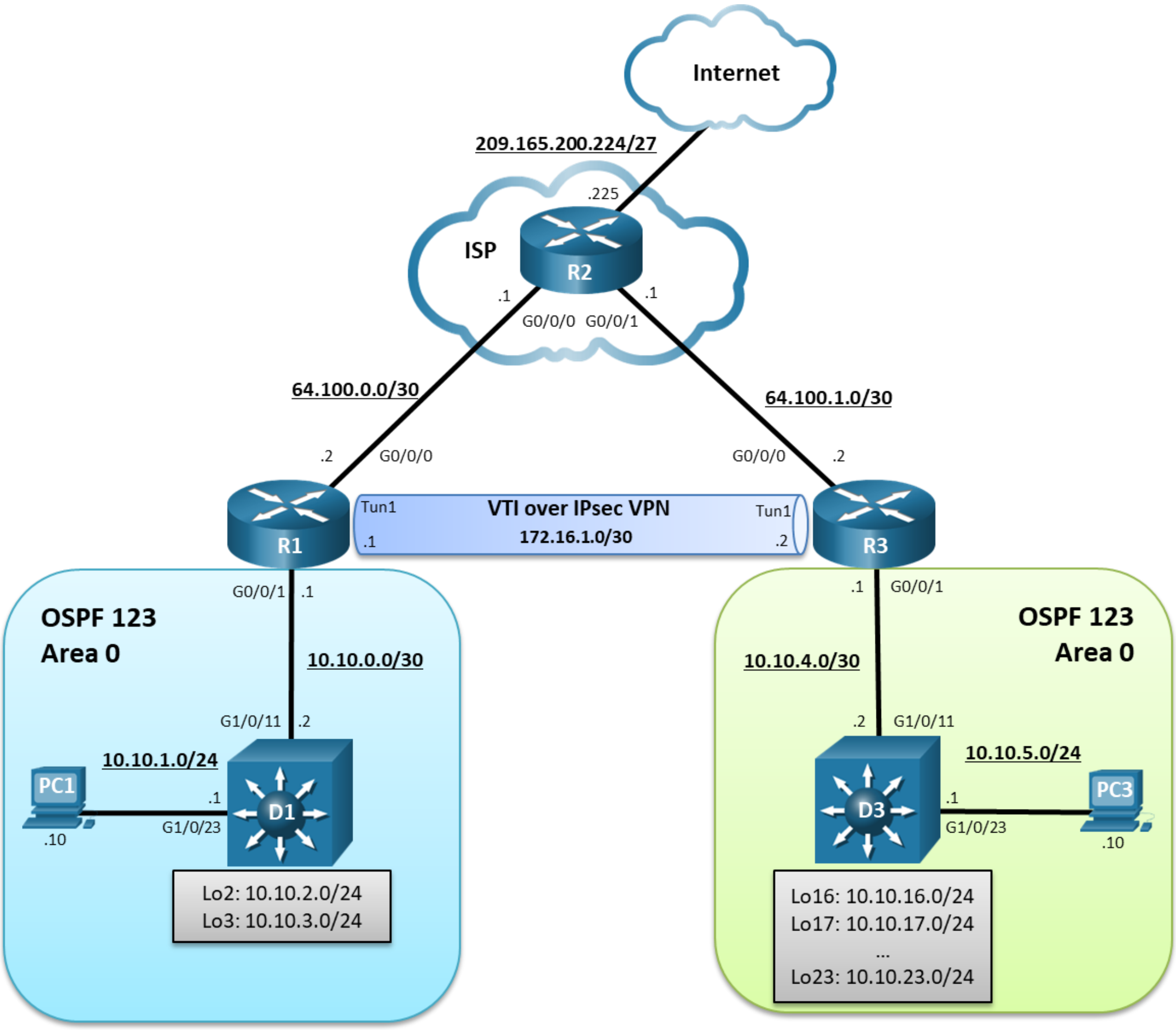
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Practical No_10

Implement IPsec Site-to-Site VPNs
1. Implement GRE over IPsec Site-to-Site VPNs
2. Implement VRF Lite

Topology



Addressing Table

Device	Interface	IPv4 Address	Default Gateway
R1	G0/0/0	64.100.0.2/30	N/A
	G0/0/1	10.10.0.1/29	
	Tunnel 1	172.16.1.1/30	
R2	G0/0/0	64.100.0.1/30	N/A
	G0/0/1	64.100.1.1/30	
	Lo0	209.165.200.225	
R3	G0/0/0	64.100.1.2/30	N/A
	G0/0/1	10.10.4.1/30	
	Tunnel 1	172.16.1.2/30	
D1	G1/0/11	10.10.0.2/29	N/A
	G1/0/23	10.10.1.1/24	
	Lo2	10.10.2.1/24	
	Lo3	10.10.3.1/24	
D3	G1/0/11	10.10.0.3/29	N/A
	G1/0/23	10.10.5.1/24	
	Lo16	10.10.16.1/24	
	Lo17	10.10.17.1/24	
	Lo18	10.10.18.1/24	
	Lo19	10.10.19.1/24	
	Lo20	10.10.20.1/24	
	Lo21	10.10.21.1/24	
	Lo22	10.10.22.1/24	
	Lo23	10.10.23.1/24	
PC1	NIC	10.10.1.10/24	10.10.1.1
PC3	NIC	10.10.5.10/24	10.10.5.1

Objectives

Part 1: Build the Network, Configure Basic Device Settings and Static Routing

Part 2: Configure Static IPsec VTI on R1 and R3

Part 3: Verify Static IPsec VTI on R1 and R3

Background / Scenario

IPsec can only send unicast IP traffic. Therefore, it does not support protocols that require multicast or broadcast communication such as routing protocols. Although GRE over IPsec can be configured to provide security and support for routing protocols, there is a newer more efficient method that can be used.

IPsec Virtual Tunnel Interface (VTI) greatly simplifies the VPN configuration process and provides a simpler alternative to using GRE tunnels for encapsulation and crypto maps with IPsec. Like GRE over IPsec, IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using the IP routing table simplifies the IPsec VPN configuration compared to the more complex process of using access control lists (ACLs) with the crypto map in native IPsec configurations. VTI over IPsec also encapsulates IPv4 or IPv6 traffic without the need for an additional GRE header. GRE adds a 4-byte header to every packet.

In this lab, you will build and configure a static VTI over IPsec with pre-shared key to enable a site-to-site VPN capable of supporting the OSPF routing protocol.

Note: This lab is an exercise in developing, deploying, and verifying how VNPs operate and does not reflect networking best practices.

Note: The routers used with this CCNP hands-on lab are Cisco 4221routers and the two Layer 3 switches are Catalyst 3650 switches. Other routers and Layer 3 switches and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the routers and switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Cisco 3650 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 PCs (Choice of operating system with a terminal emulation program installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network, Configure Basic Device Settings and Static Routing

In Part 1, you will set up the network topology, configure basic settings, interface addressing, and single-area OSPFv2 on the routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for the routers.

a. Console into each router and switch, enter global configuration mode, and apply the basic settings, and interface addressing. A command list for each device is provided for your reference.

Routing is enabled as follows:

- R2 has a static route to the networks connected to R1 (i.e., 10.10.0.0/22) and two static routes to the networks connected to R3 (i.e., 10.10.4.0/22, 10.10.16.0/21).
- R1 and R3 each have a default static route to R2.
- OSPFv2 routing is enabled between R1 and D1, and R1 is propagating the default route to D1.
- OSPFv2 routing is enabled between R3 and D3, and R3 is propagating the default route to D3.
- A command list for each device is listed below to perform initial configurations.

Router R1

```
hostname R1
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is R1, Implement IPsec VTI Site-to-Site VPNs #
interface g0/0/0
 description Connection to R2
 ip add 64.100.0.2 255.255.255.252
 no shut
 exit
interface GigabitEthernet0/0/1
 description Connection to D1
 ip address 10.10.0.1 255.255.255.252
 no shut
 exit
router ospf 123
 router-id 1.1.1.1
 auto-cost reference-bandwidth 1000
 network 10.10.0.0 0.0.0.3 area 0
 default-information originate
exit
ip route 0.0.0.0 0.0.0.0 64.100.0.1
```

Router R2

```
hostname R2
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is R2, Implement IPsec VTI Site-to-Site VPNs #
interface g0/0/0
 description Connection to R1
 ip add 64.100.0.1 255.255.255.252
 no shut
 exit
interface GigabitEthernet0/0/1
 description Connection to R3
 ip address 64.100.1.1 255.255.255.252
 no shut
 exit
int lo0
 description Internet simulated address
 ip add 209.165.200.225 255.255.255.224
 exit
ip route 0.0.0.0 0.0.0.0 Loopback0
ip route 10.10.0.0 255.255.252.0 64.100.0.2
ip route 10.10.4.0 255.255.252.0 64.100.1.2
ip route 10.10.16.0 255.255.248.0 64.100.1.2
```

Router R3

```
hostname R3
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is R3, Implement IPsec VTI Site-to-Site VPNs #
interface g0/0/0
 description Connection to R2
 ip add 64.100.1.2 255.255.255.252
 no shut
 exit
interface GigabitEthernet0/0/1
 description Connection to D3
 ip address 10.10.4.1 255.255.255.252
 no shut
 exit
ip route 0.0.0.0 0.0.0.0 64.100.1.1
router ospf 123
 router-id 3.3.3.1
 auto-cost reference-bandwidth 1000
 network 10.10.4.0 0.0.0.3 area 0
 default-information originate
exit
```

Switch D1

```
hostname D1
no ip domain lookup
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
banner motd # This is D1, Implement IPsec VTI Site-to-Site VPNs #
interface G1/0/11
 description Connection to R1
 no switchport
 ip address 10.10.0.2 255.255.255.252
 no shut
 exit
interface G1/0/23
 description Connection to PC1
 no switchport
 ip address 10.10.1.1 255.255.255.0
 no shut
 exit
int Lo2
 description Loopback to simulate an OSPF network
 ip add 10.10.2.1 255.255.255.0
 ip ospf network point-to-point
exit
int Lo3
 description Loopback to simulate an OSPF network
 ip add 10.10.3.1 255.255.255.0
 ip ospf network point-to-point
exit
ip routing
router ospf 123
 router-id 1.1.1.2
 auto-cost reference-bandwidth 1000
 network 10.10.0.0 0.0.3.255 area 0
 exit
int range G1/0/1 - 10, G1/0/12 - 22, G1/0/24
 shut
 exit
```

Switch D3

```
hostname D3
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is D3, Implement IPsec VTI Site-to-Site VPNs #
interface G1/0/11
 description Connection to R3
 no switchport
 ip address 10.10.4.2 255.255.255.252
 no shut
 exit
interface G1/0/23
 description Connection to PC3
 no switchport
 ip address 10.10.5.1 255.255.255.0
 no shut
 exit
int Lo16
 description Loopback to simulate an OSPF network
 ip add 10.10.16.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo17
 description Loopback to simulate an OSPF network
 ip add 10.10.17.1 255.255.255.0
 ip ospf network point-to-point
 exit
```

```
int Lo18
 description Loopback to simulate an OSPF network
 ip add 10.10.18.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo19
 description Loopback to simulate an OSPF network
 ip add 10.10.19.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo20
 description Loopback to simulate an OSPF network
 ip add 10.10.20.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo21
 description Loopback to simulate an OSPF network
 ip add 10.10.21.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo22
 description Loopback to simulate an OSPF network
 ip add 10.10.22.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo23
 description Loopback to simulate an OSPF network
 ip add 10.10.23.1 255.255.255.0
 ip ospf network point-to-point
 exit
ip routing
router ospf 123
 router-id 3.3.3.2
 auto-cost reference-bandwidth 1000
 network 10.10.4.0 0.0.1.255 area 0
 network 10.10.16.0 0.0.7.255 area 0
 exit
int range G1/0/1 - 10, G1/0/12 - 22, G1/0/24
 shut
```

b. Save the running configuration to startup-config.

Step 3: Configure PC1 and PC3 with IP addressing.

Configure the two PCs with the IP addresses listed in the Address Table. Also configure their respective default gateways.

Step 4: On PC1, verify end-to-end connectivity.

a. From PC1, **ping** PC3 (10.10.5.10).

```
PC1> ping 10.10.5.10

Pinging 10.10.5.10 with 32 bytes of data:
Reply from 10.10.5.10: bytes=32 time=1ms TTL=123
Reply from 10.10.5.10: bytes=32 time=1ms TTL=123
Reply from 10.10.5.10: bytes=32 time=1ms TTL=123
Reply from 10.10.5.10: bytes=32 time=1ms TTL=123

Ping statistics for 10.10.5.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

The pings should be successful. If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

b. From PC1, **ping** the first loopback on D3 (10.10.16.1).

```
PC1> ping 10.10.16.1

Pinging 10.10.16.1 with 32 bytes of data:
Reply from 10.10.16.1: bytes=32 time=2ms TTL=250
Reply from 10.10.16.1: bytes=32 time=2ms TTL=250
Reply from 10.10.16.1: bytes=32 time=2ms TTL=250
Reply from 10.10.16.1: bytes=32 time=2ms TTL=250

Ping statistics for 10.10.16.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 2ms, Average = 2ms
```

The pings should be successful. If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

c. From PC1, **ping** the default gateway loopback on R2 (209.165.200.225).

```
PC1> ping 209.165.200.225

Pinging 209.165.200.225 with 32 bytes of data:
Reply from 209.165.200.225: bytes=32 time=1ms TTL=253
Reply from 209.165.200.225: bytes=32 time=1ms TTL=253
Reply from 209.165.200.225: bytes=32 time=1ms TTL=253
Reply from 209.165.200.225: bytes=32 time=1ms TTL=253

Ping statistics for 209.165.200.225:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

The pings should be successful. If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

Step 5: Verify the routing table of R1.

a. Verify the OSPF routing table of R1.

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.0.1 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
O       10.10.1.0/24 [110/11] via 10.10.0.2, 00:29:03, GigabitEthernet0/0/1
O       10.10.2.0/24 [110/2] via 10.10.0.2, 00:29:03, GigabitEthernet0/0/1
O       10.10.3.0/24 [110/2] via 10.10.0.2, 00:29:03, GigabitEthernet0/0/1
```

The routing table confirms that R1 has knowledge of the networks connected to D1. Notice that R1 has no knowledge of the routes connected to the R3 OSPF domain. The reason why PC1 can still reach PC3 is because R1 has a default static route to R2. R1 forwarded the traffic to R2 because it did not know where the 10.10.5.0 network was. R2 has a static route to this network and therefore forwarded it to R3.

b. Verify the routing table of R3.

```
R3# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.1.1 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 11 subnets, 3 masks
O       10.10.5.0/24 [110/11] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.16.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.17.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.18.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.19.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.20.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.21.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.22.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O       10.10.23.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
```

Like R1, the routing table of R3 only contains its local routes.

Part 2: Configure Static IPsec VTI on R1 and R3

A limitation of IPsec VPNs is that it only forwards unicast traffic across the VPN tunnel. Therefore, routing protocol traffic is not propagated across the VPN tunnel.

GRE over IPsec VPN could be configured to support routing protocol traffic over the IPsec VPN. However, IP VTI is simpler and more efficient than GRE over IPsec.

IPsec VTI can be configured using:

- **Static VTIs (SVTIs)** - SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites. The advantage of using SVTIs as opposed to crypto map configurations is that users can enable dynamic routing protocols on the tunnel interface without the extra 4 bytes required for GRE headers, therefore reducing the bandwidth for sending encrypted data.
- **Dynamic VTIs (DVTIs)** - DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.

The steps to enable IPsec VTI are very similar to GRE over IPsec except:

Step 1. The tunnel interface is configured with the **tunnel mode ipsec {ipv4 | ipv6}** command.

Step 2. The transform set is configured with the mode tunnel command. An ACL is not required.

Like site-to-site VPNs using crypto maps and GRE over IPsec using crypto maps, IPsec VTI also requires the following:

- ISAKMP policy configuration and pre-shared key configured
- Transform set configured
- IPsec profile configured

In this part, you will configure a static IPsec SVTI to provide an always on site-to-site VPN as shown in the topology diagram.

Step 1: On R1 and R3, configure the ISAKMP policy and pre-shared key.

In this lab, we will use the following parameters for the ISAKMP policy 10 on R1 and R3:

- Encryption: **aes 256**
- Hash: sha256
- Authentication method: **pre-share key**
- Diffie-Hellman group: **14**
- Lifetime: **3600** seconds (60 minutes / 1 hour)

a. Configure ISAKMP policy 10 on R1 and R3.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# hash sha256
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 14
R1(config-isakmp)# lifetime 3600
R1(config-isakmp)# exit
```

```
R3(config)# crypto isakmp policy 10
R3(config-isakmp)# encryption aes 256
R3(config-isakmp)# hash sha256
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# group 14
R3(config-isakmp)# lifetime 3600
R3(config-isakmp)# exit
```

b. Configure the pre-shared key of **cisco123** on R1 and R3.

Note: Production networks should use longer and more complex keys.

```
R1(config)# crypto isakmp key cisco123 address 64.100.1.2
```

```
R3(config)# crypto isakmp key cisco123 address 64.100.0.2
```

Step 2: On R1 and R3, configure the transform set and tunnel mode.

Create a new transform set called VTI-VPN using ESP AES 256 for encryption and ESP SHA256 HMAC for authentication and set the mode to **tunnel**.

Note: The transform set would default to tunnel mode automatically but is configured in the example for emphasis.

```
R1(config)# crypto ipsec transform-set VTI-VPN esp-aes 256 esp-sha256-hmac
R1(cfg-crypto-trans)# mode tunnel
R1(cfg-crypto-trans)# exit
```

```
R3(config)# crypto ipsec transform-set VTI-VPN esp-aes 256 esp-sha256-hmac
R3(cfg-crypto-trans)# mode tunnel
R3(cfg-crypto-trans)# exit
```

Step 3: On R1 and R3, configure VTI over IPsec using IPsec profiles.

Configure an IPsec profile called **VTI-PROFILE** using the **crypto ipsec profile** *ipsec-profile-name* global configuration command and set the transform set to VTI-VPN.

```
R1(config)# crypto ipsec profile VTI-PROFILE
R1(ipsec-profile)# set transform-set VTI-VPN
R1(ipsec-profile)# exit


R3(config)# crypto ipsec profile VTI-PROFILE
R3(ipsec-profile)# set transform-set VTI-VPN
R3(ipsec-profile)# exit
```

Step 4: On R1, configure the tunnel interface.

a. Next, configure a tunnel interface on R1.

```
R1(config)# interface Tunnel1
R1(config-if)# bandwidth 4000
R1(config-if)# ip address 172.16.1.1 255.255.255.252
R1(config-if)# ip mtu 1400
R1(config-if)# tunnel source 64.100.0.2
R1(config-if)# tunnel destination 64.100.1.2
R1(config-if)#
*Jan 21 12:31:13.824: %LINEPROTO-5-UPDOWN:  Line protocol on Interface
Tunnel1, changed state to up
```

b. Tunnel interfaces default to **tunnel mode gre** mode. However, we must now change the tunnel mode from the default GRE setting to the IPsec setting. Configure Tunnel 1 using the **tunnel mode ipsec ipv4** command.

```
R1(config-if)# tunnel mode ipsec ipv4
R1(config-if)#
*Jan 21 12:32:15.047: %LINEPROTO-5-UPDOWN:  Line protocol on Interface
Tunnel1, changed state to down
```

c. Next, the IPsec profile **VTI-PROFILE** must be applied using the **tunnel protection ipsec profile** *profile-name* command.

```
R1(config-if)# tunnel protection ipsec profile VTI-PROFILE
R1(config-if)#
*Jan 21 12:32:50.103: %CRYPTO-6-ISAKMP_ON_OFF:  ISAKMP is ON
R1(config-if)# exit
```

Notice the informational message that the ISAKMP policy will be used.

Step 5: On R3, configure the tunnel interface.

Now we must mirror the configuration of R1 on R3.

a. Next, configure a GRE tunnel interface on R3.

```
R3(config)# interface Tunnel1
R3(config-if)# bandwidth 4000
R3(config-if)# ip address 172.16.1.2 255.255.255.252
R3(config-if)# ip mtu 1400
R3(config-if)# tunnel source 64.100.1.2
*Feb 20 12:53:14.367: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to down
R3(config-if)# tunnel destination 64.100.0.2
R3(config-if)#
*Feb 20 12:53:16.683: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to up
```

Notice the information messages indicating the line going down and then up.

b. Tunnel 1 must be configured using the **tunnel mode ipsec ipv4** command.

```
R3(config-if)# tunnel mode ipsec ipv4
R3(config-if)#
*Feb 20 12:53:45.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to down
```

Again, the Tunnel 1 interface goes down.

c. Finally, the IPsec profile **VTI-PROFILE** must be applied using the **tunnel protection ipsec profile** *profile-name* command.

```
R3(config-if)# tunnel protection ipsec profile VTI-PROFILE
R3(config-if)#
*Feb 20 12:54:05.111: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R3(config-if)#
*Feb 20 12:54:05.381: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to up
R3(config-if)# exit
```

Notice the informational message that the ISAKMP policy will be used and that the Tunnel 1 interface is up.

Step 6: On R1 and R3, advertise the tunnel interface in OSPF.

a. On R1, configure OSPF to advertise the tunnel interfaces.

```
R1(config)# router ospf 123
R1(config-router)# network 172.16.1.0 0.0.0.3 area 0
R1(config-router)# end
```

b. Next on R3, configure OSPF to advertise the tunnel interfaces.

```
R3(config)# router ospf 123
R3(config-router)# network 172.16.1.0 0.0.0.3 area 0
R3(config-router)# exit
R3(config)#
*Feb 20 13:09:48.456: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.1 on Tunnel1
from LOADING to FULL, Loading Done
R3(config)# exit
```

Notice the OSPF adjacency message that appears when the network command is entered.

Part 3: Verify Static IPsec VTI on R1 and R3

Now that the IPsec has been configured, we must verify that the tunnel interfaces are correctly enabled, that the crypto session is active, and then generate traffic to confirm it is traversing securely over the IPsec VTI tunnel.

Step 1: On R1 and R3, verify the tunnel interfaces.

a. Use the **show interfaces tunnel 1** command to verify the interface settings.

```
R1# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/30
  MTU 9938 bytes, BW 4000 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 64.100.0.2, destination 64.100.1.2
  Tunnel protocol/transport IPSEC/IP
  Tunnel TTL 255
  Tunnel transport MTU 1438 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "VTI-PROFILE")
  Last input 00:00:07, output 00:00:08, output hang never
  Last clearing of "show interface" counters 00:32:55
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     20 packets input, 2368 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     23 packets output, 2424 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

Notice the highlighted output identifying various aspects of the tunnel interface.

b. On R3, use the **show interfaces tunnel 1** command to verify the interface settings.

```
R3# show interface tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.2/30
  MTU 9938 bytes, BW 4000 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 64.100.1.2, destination 64.100.0.2
  Tunnel protocol/transport IPSEC/IP
  Tunnel TTL 255
  Tunnel transport MTU 1438 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "VTI-PROFILE")
  Last input 00:00:03, output 00:00:09, output hang never
  Last clearing of "show interface" counters 00:24:32
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     62 packets input, 6324 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  58 packets output, 6168 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 unknown protocol drops
  0 output buffer failures, 0 output buffers swapped out
```

Again, the highlighted output identifies various aspects of the tunnel interface.

Step 2: On R1 and R3, verify the crypto settings.

a. On R1, use the **show crypto session** command to verify the operation of the VPN tunnel.

```
R1# show crypto session
Crypto session current status

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 64.100.1.2 port 500
  Session ID: 0
  IKEv1 SA: local 64.100.0.2/500 remote 64.100.1.2/500 Active
  Session ID: 0
  IKEv1 SA: local 64.100.0.2/500 remote 64.100.1.2/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 4, origin: crypto map
```

The output confirms that Tunnel 1 is up and active with R3 (64.100.1.2). The port 500 refers to ISAKMP using UDP port 500.

b. On R3, use the **show crypto session** command to verify the operation of the VPN tunnel.

```
R3# show crypto session
Crypto session current status

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 64.100.0.2 port 500
  Session ID: 0
  IKEv1 SA: local 64.100.1.2/500 remote 64.100.0.2/500 Active
  Session ID: 0
  IKEv1 SA: local 64.100.1.2/500 remote 64.100.0.2/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 4, origin: crypto map
```

Step 3: On R1 and R3, verify the routing tables.

a. Verify the R1 routing table for OSPF routes.

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.0.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
O        10.10.1.0/24 [110/11] via 10.10.0.2, 01:28:00, GigabitEthernet0/0/1
O        10.10.2.0/24 [110/2] via 10.10.0.2, 01:28:00, GigabitEthernet0/0/1
O        10.10.3.0/24 [110/2] via 10.10.0.2, 01:28:00, GigabitEthernet0/0/1
O        10.10.4.0/30 [110/251] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.5.0/24 [110/261] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.16.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.17.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.18.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.19.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.20.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.21.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.22.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.23.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
```

Notice how R1 has learned about the R3 OSPF networks via the tunnel interface.

b. Verify the R3 routing table for OSPF routes.

```
R3# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.1.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
O        10.10.0.0/30 [110/251] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.1.0/24 [110/261] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.2.0/24 [110/252] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.3.0/24 [110/252] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.5.0/24 [110/11] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.16.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.17.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.18.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.19.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.20.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.21.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.22.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.23.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
```

Notice how R3 has learned about the R1 OSPF networks via the tunnel interface.

c. From D1, trace the path taken to the R3 10.10.5.1 interface.

```
D1# trace 10.10.5.1
Type escape sequence to abort.
Tracing the route to 10.10.5.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.0.1 2 msec 2 msec 2 msec
  2 172.16.1.2 3 msec 2 msec 3 msec
  3 10.10.4.2 3 msec *  4 msec
```

Notice how the path taken is through the VPN tunnel interface.

d. On R1, verify the IPsec SA encrypted and decrypted statistics.

```
R1# show crypto ipsec sa | include encrypt|decrypt
    #pkts encaps: 28, #pkts encrypt: 28, #pkts digest: 28
    #pkts decaps: 26, #pkts decrypt: 26, #pkts verify: 26
```

e. Verify that there is an operational logical point-to-point link between R1 and R3 using the VTI tunnel interface.

```
R1# show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 2 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
C       172.16.1.0/30 is directly connected, Tunnel1
L       172.16.1.1/32 is directly connected, Tunnel1
```

```
R3# show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 2 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
C       172.16.1.0/30 is directly connected, Tunnel1
L       172.16.1.2/32 is directly connected, Tunnel1
```

Step 4: Test the IPsec VTI tunnel.

a. From D1, trace the path taken to the R3 10.10.16.1 interface.

```
D1# trace 10.10.16.1
Type escape sequence to abort.
```

```
Tracing the route to 10.10.16.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.0.1 0 msec 0 msec 9 msec
  2 172.16.1.2 0 msec 0 msec 0 msec
  3 10.10.4.2 8 msec *  0 msec
```

Notice now that the path taken is through the VPN tunnel interface.

b. On R1, verify the IPsec SA encrypted and decrypted statistics.

```
R1# show crypto ipsec sa | include encrypt|decrypt
    #pkts encaps: 230, #pkts encrypt: 230, #pkts digest: 230
    #pkts decaps: 200, #pkts decrypt: 200, #pkts verify: 200
```

The output verifies that the IPsec VTI is properly encrypting traffic between both sites. The packets encrypted include the trace packets along with OSPF packets.

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs – Final

Router R1

```
R1# show run
Building configuration...

Current configuration : 2005 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
license udi pid ISR4221/K9 sn FGL23313183
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
redundancy
 mode none
!
crypto isakmp policy 10
 encr aes 256
 hash sha256
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco123 address 64.100.1.2
!
crypto ipsec transform-set VTI-VPN esp-aes 256 esp-sha256-hmac
 mode tunnel
!
crypto ipsec profile VTI-PROFILE
 set transform-set VTI-VPN
!
interface Tunnel1
 bandwidth 4000
 ip address 172.16.1.1 255.255.255.252
 ip mtu 1400
 tunnel source 64.100.0.2
 tunnel mode ipsec ipv4
 tunnel destination 64.100.1.2
 tunnel protection ipsec profile VTI-PROFILE
!
interface GigabitEthernet0/0/0
 description Connection to R2
 ip address 64.100.0.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 description Connection to D1
 ip address 10.10.0.1 255.255.255.252
 negotiation auto
!
interface Serial0/1/0
 no ip address
!
interface Serial0/1/1
 no ip address
!
router ospf 123
 router-id 1.1.1.1
 auto-cost reference-bandwidth 1000
 network 10.10.0.0 0.0.0.3 area 0
 network 172.16.1.0 0.0.0.3 area 0
 default-information originate
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip route 0.0.0.0 0.0.0.0 64.100.0.1
!
control-plane
!
banner motd ^C This is R1, Implement IPsec VTI Site-to-Site VPNs ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

Router R2

```
R2# show run
Building configuration...

Current configuration : 1482 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
```

```
multilink bundle-name authenticated
!
license udi pid ISR4221/K9 sn FGL23313182
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 description Internet simulated address
 ip address 209.165.200.225 255.255.255.224
!
interface GigabitEthernet0/0/0
 description Connection to R1
 ip address 64.100.0.1 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 description Connection to R3
 ip address 64.100.1.1 255.255.255.252
 negotiation auto
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip route 0.0.0.0 0.0.0.0 Loopback0
ip route 10.10.0.0 255.255.252.0 64.100.0.2
ip route 10.10.4.0 255.255.252.0 64.100.1.2
ip route 10.10.16.0 255.255.248.0 64.100.1.2
!
control-plane
!
banner motd ^C This is R2, Implement IPsec VTI Site-to-Site VPNs ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
```

```
line vty 0 4
 login
!
end
```

Router R3

```
R3# show run
Building configuration...

Current configuration : 2005 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
license udi pid ISR4221/K9 sn FGL23313186
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
redundancy
 mode none
!
crypto isakmp policy 10
 encr aes 256
 hash sha256
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco123 address 64.100.0.2
!
crypto ipsec transform-set VTI-VPN esp-aes 256 esp-sha256-hmac
 mode tunnel
!
crypto ipsec profile VTI-PROFILE
 set transform-set VTI-VPN
!
interface Tunnel1
 bandwidth 4000
 ip address 172.16.1.2 255.255.255.252
 ip mtu 1400
 tunnel source 64.100.1.2
 tunnel mode ipsec ipv4
 tunnel destination 64.100.0.2
 tunnel protection ipsec profile VTI-PROFILE
!
interface GigabitEthernet0/0/0
 description Connection to R2
 ip address 64.100.1.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 description Connection to D3
 ip address 10.10.4.1 255.255.255.252
 negotiation auto
!
interface Serial0/1/0
 no ip address
!
interface Serial0/1/1
 no ip address
!
router ospf 123
 router-id 3.3.3.1
 auto-cost reference-bandwidth 1000
 network 10.10.4.0 0.0.0.3 area 0
 network 172.16.1.0 0.0.0.3 area 0
 default-information originate
!
```

```
ip forward-protocol nd
no ip http server
ip http secure-server
ip route 0.0.0.0 0.0.0.0 64.100.1.1
!
control-plane
!
banner motd ^C This is R3, Implement IPsec VTI Site-to-Site VPNs ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

Switch D1

```
D1# show run
Building configuration...

Current configuration : 7035 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D1
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ps
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email
address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
ip routing
!
no ip domain lookup
!
login on-success log
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388
8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7
D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F
EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF
58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8
8F27D191
```

```
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368
95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1
6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146
8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2
55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49
1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69
39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD
230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
        quit
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception,
NFLSAMPLED DATA, RPF Failed
```

```
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface Loopback2
 description Loopback to simulate an OSPF network
 ip address 10.10.2.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback3
 description Loopback to simulate an OSPF network
 ip address 10.10.3.1 255.255.255.0
 ip ospf network point-to-point
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 negotiation auto
```

```
!
interface GigabitEthernet1/0/1
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 shutdown
!
interface GigabitEthernet1/0/6
 shutdown
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 description Connection to R1
 no switchport
 ip address 10.10.0.2 255.255.255.252
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
```

```
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 description Connection to PC1
 no switchport
 ip address 10.10.1.1 255.255.255.0
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
!
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
!
interface GigabitEthernet1/1/4
!
interface Vlan1
 no ip address
!
```

```
router ospf 123
 router-id 1.1.1.2
 auto-cost reference-bandwidth 1000
 network 10.10.0.0 0.0.3.255 area 0
!
ip forward-protocol nd
ip http server
ip http secure-server
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C This is D1, Implement IPsec VTI Site-to-Site VPNs ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
!
end
```

Switch D3

```
D3# show run
Building configuration...

Current configuration : 7928 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D3
!
```

```
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ps
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email
address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
ip routing
!
no ip domain lookup
!
login on-success log
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
! crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388
8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7
D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F
EA2956AC
```

```
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF
58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8
8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368
95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1
6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146
8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2
55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49
1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69
39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD
230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
        quit
!
license boot level ipservicesk9
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
```

```
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception,
NFLSAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface Loopback16
 description Loopback to simulate an OSPF network
 ip address 10.10.16.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback17
 description Loopback to simulate an OSPF network
 ip address 10.10.17.1 255.255.255.0
 ip ospf network point-to-point
```

```
!
interface Loopback18
 description Loopback to simulate an OSPF network
 ip address 10.10.18.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback19
 description Loopback to simulate an OSPF network
 ip address 10.10.19.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback20
 description Loopback to simulate an OSPF network
 ip address 10.10.20.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback21
 description Loopback to simulate an OSPF network
 ip address 10.10.21.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback22
 description Loopback to simulate an OSPF network
 ip address 10.10.22.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback23
 description Loopback to simulate an OSPF network
 ip address 10.10.23.1 255.255.255.0
 ip ospf network point-to-point
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 negotiation auto
!
interface GigabitEthernet1/0/1
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
```

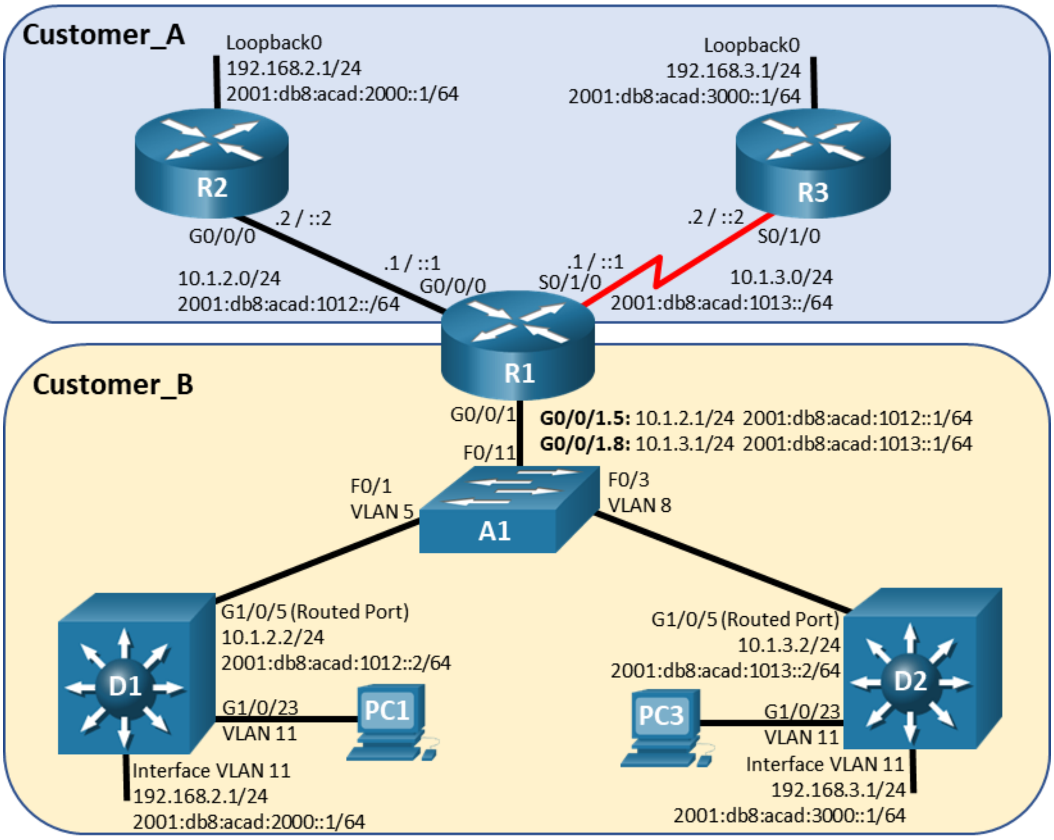
```
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 shutdown
!
interface GigabitEthernet1/0/6
 shutdown
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 description Connection to R3
 no switchport
 ip address 10.10.4.2 255.255.255.252
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
 shutdown
```

```
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 description Connection to PC3
 no switchport
 ip address 10.10.5.1 255.255.255.0
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
!
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
!
interface GigabitEthernet1/1/4
!
interface Vlan1
 no ip address
!
router ospf 123
 router-id 3.3.3.2
 auto-cost reference-bandwidth 1000
 network 10.10.4.0 0.0.1.255 area 0
 network 10.10.16.0 0.0.7.255 area 0
!
ip forward-protocol nd
ip http server
ip http secure-server
```

```
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C This is D3, Implement IPsec VTI Site-to-Site VPNs ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
!
end
```

[Title]

Topology



Addressing Table

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
R1	G0/0/0	10.1.2.1/24	2001:db8:acad:1012::1/64	fe80::1:1
	G0/0/1.5	10.1.2.1/24	2001:db8:acad:1012::1/64	fe80::1:2
	G0/0/1.8	10.1.3.1/24	2001:db8:acad:1013::1/64	fe80::1:4
	S0/1/0	10.1.3.1/25	2001:db8:acad:1013::1/64	fe80::1:2
R2	G0/0/0	10.2.3.2/24	2001:db8:acad:1023::2/64	fe80::2:1
	Loopback0	192.168.2.1/24	2001:db8:acad:2000::1/64	fe80::2:2
R3	S0/1/0	10.1.3.3/25	2001:db8:acad:1013::3/64	fe80::3:1
	Loopback0	192.168.3.1/27	2001:db8:acad:3000::1/64	fe80::3:2
D1	G1/0/5	10.1.2.2/24	2001:db8:acad:1012::2/64	fe80::d1:1
	VLAN 11	192.168.2.1/24	2001:db8:acad:2000::2/64	fe80::d1:2
D2	G1/0/5	10.1.3.2/24	2001:db8:acad:1013::2/64	fe80::d2:1

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
	VLAN 11	192.168.3.1/24	2001:db8:acad:3000::1/64	fe80::d2:2

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure and Verify VRF and Interface Addressing

Part 3: Configure and Verify Static Routing for Reachability Inside Each VRF

Background / Scenario

By default, all interfaces on a router are included in the global routing table. Service providers must be able to virtualize the router, thus creating multiple, virtual routing tables. Virtual Routing and Forwarding (VRF) can do just that. VRF-Lite is VRF without the MPLS component.

In this lab, you will work on R1, playing the part of a service provider router, as it supports two customers who have the same addressing scheme configured. Your task is to deploy VRF-Lite and static routing so that the customers have full reachability within their network.

Note: This lab is an exercise in developing, deploying, and verifying VRF-Lite, and does not reflect networking best practices.

Note: The routers and switches used with CCNP hands-on labs are Cisco 4221 and Cisco 3650, both with Cisco IOS XE Release 16.9.4 (universalk9 image), and Cisco 2960+ with IOS release 15.2 (lanbase image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs

Note: Ensure that the routers and switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Note: The PCs used in this lab do not require addressing. They are needed to bring interface VLAN 11 up.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Cisco 3650 with Cisco IOS XE release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 2960+ with Cisco IOS release 15.2 lanbase image or comparable)
- 2 PCs (Windows with a terminal emulation program, such as Tera Term)

- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings on all devices.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each device.

a. Console into each device, enter global configuration mode, and apply the basic settings. A command list for each device using the following startup configurations.

Router R1

```
enable
configure terminal
hostname R1
no ip domain lookup
ipv6 unicast-routing
banner motd # R1, Implement VRF-Lite #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
```

Router R2

```
enable
configure terminal
hostname R2
no ip domain lookup
ipv6 unicast-routing
banner motd # R2, Implement VRF-Lite #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
interface g0/0/0
 ip address 10.1.2.2 255.255.255.0
 ipv6 address fe80::2:1 link-local
 ipv6 address 2001:db8:acad:1012::2/64
 no shutdown
 exit
interface loopback 0
 ip address 192.168.2.1 255.255.255.0
 ipv6 address fe80::2:2 link-local
 ipv6 address 2001:db8:acad:2000::1/64
 no shutdown
 exit
ip route 0.0.0.0 0.0.0.0 g0/0/0 10.1.2.1
ipv6 route ::/0 g0/0/0 2001:db8:acad:1012::1
```

Router R3

```
enable
configure terminal
hostname R3
no ip domain lookup
ipv6 unicast-routing
banner motd # R3, Implement VRF-Lite #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
interface s0/1/0
```

```
 ip address 10.1.3.2 255.255.255.0
 ipv6 address fe80::3:1 link-local
 ipv6 address 2001:db8:acad:1013::2/64
 no shutdown
 exit
interface loopback 0
 ip address 192.168.3.1 255.255.255.0
 ipv6 address fe80::3:2 link-local
 ipv6 address 2001:db8:acad:3000::1/64
 no shutdown
 exit
ip route 0.0.0.0 0.0.0.0 s0/1/0 10.1.3.1
ipv6 route ::/0 s0/1/0 2001:db8:acad:1013::1
```

Switch D1

```
enable
configure terminal
hostname D1
no ip domain lookup
ip routing
ipv6 unicast-routing
banner motd # D1, Implement VRF-Lite #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface g1/0/5
 no switchport
 ip address 10.1.2.2 255.255.255.0
 ipv6 address fe80::d1:1 link-local
 ipv6 address 2001:db8:acad:1012::2/64
 no shutdown
 exit
vlan 11
 name LOCAL_VLAN
 exit
interface vlan 11
 ip address 192.168.2.1 255.255.255.0
 ipv6 address fe80::d1:2 link-local
 ipv6 address 2001:db8:acad:2000::1/64
 no shutdown
 exit
interface g1/0/23
 switchport mode access
 switchport access vlan 11
 no shutdown
 exit
ip route 0.0.0.0 0.0.0.0 g1/0/5 10.1.2.1
ipv6 route ::/0 g1/0/5 2001:db8:acad:1012::1
```

Switch D2

```
enable
configure terminal
hostname D2
no ip domain lookup
ip routing
ipv6 unicast-routing
banner motd # D2, Implement VRF-Lite #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface g1/0/5
 no switchport
 ip address 10.1.3.2 255.255.255.0
 ipv6 address fe80::d2:1 link-local
 ipv6 address 2001:db8:acad:1013::2/64
 no shutdown
 exit
vlan 11
 name LOCAL_VLAN
 exit
interface vlan 11
 ip address 192.168.3.1 255.255.255.0
 ipv6 address fe80::d2:2 link-local
 ipv6 address 2001:db8:acad:3000::1/64
 no shutdown
 exit
interface g1/0/23
 switchport mode access
 switchport access vlan 11
 no shutdown
 exit
ip route 0.0.0.0 0.0.0.0 g1/0/5 10.1.3.1
ipv6 route ::/0 g1/0/5 2001:db8:acad:1013::1
```

Switch A1

```
enable
configure terminal
hostname A1
no ip domain lookup
banner motd # A1, Implement VRF-Lite #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 password cisco123
 exec-timeout 0 0
 logging synchronous
 login
 exit
interface range f0/1-24, g0/1-2
 shutdown
 exit
```

```
vlan 5
 name D1
 exit
vlan 8
 name D2
 exit
interface f0/11
 switchport mode trunk
 switchport nonegotiate
 no shutdown
 exit
interface f0/1
 switchport mode access
 switchport access vlan 5
 no shutdown
 exit
interface f0/3
 switchport mode access
 switchport access vlan 8
 no shutdown
```

b. Set the clock on each router to UTC time.

c. Save the running configuration to startup-config.

Part 2: Configure and Verify VRF and Interface Addressing

In Part 2, you will configure and verify VRF-Lite on R1. The other devices, R2, R3, D1, D2, and A1 require no additional configuration. Once again, the configuration being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: On R1, create the required VRFs.

a. Create the Customer_A and Customer_B VRFs, and initialize them for both IPv4 and IPv6. The VRF names are case sensitive.

```
R1(config)# vrf definition Customer_A
R1(config-vrf)# address-family ipv4
R1(config-vrf-af)# address-family ipv6
R1(config-vrf-af)# exit
R1(config-vrf)# vrf definition Customer_B
R1(config-vrf)# address-family ipv4
R1(config-vrf-af)# address-family ipv6
R1(config-vrf-af)# exit
```

b. Configure interfaces G0/0/0 and S0/1/0 for the Customer_A network.

```
R1(config)# interface g0/0/0
R1(config-if)# vrf forwarding Customer_A
R1(config-if)# ip address 10.1.2.1 255.255.255.0
R1(config-if)# ipv6 address fe80::1:1 link-local
R1(config-if)# ipv6 address 2001:db8:acad:1012::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface s0/1/0
R1(config-if)# vrf forwarding Customer_A
R1(config-if)# ip address 10.1.3.1 255.255.255.0
R1(config-if)# ipv6 address fe80::1:4 link-local
R1(config-if)# ipv6 address 2001:db8:acad:1013::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
```

c. Configure R1 interface G0/0/1 to support the Customer_B networks. G0/0/1 will be performing inter-VLAN routing between VLANs 5 and 8.

```
R1(config)# interface g0/0/1
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface g0/0/1.5
R1(config-subif)# encapsulation dot1q 5
R1(config-subif)# vrf forwarding Customer_B
R1(config-subif)# ip address 10.1.2.1 255.255.255.0
R1(config-subif)# ipv6 address fe80::1:2 link-local
R1(config-subif)# ipv6 address 2001:db8:acad:1012::1/64
R1(config-subif)# exit
R1(config)# interface g0/0/1.8
R1(config-subif)# encapsulation dot1q 8
R1(config-subif)# vrf forwarding Customer_B
R1(config-subif)# ip address 10.1.3.1 255.255.255.0
R1(config-subif)# ipv6 address fe80::1:3 link-local
R1(config-subif)# ipv6 address 2001:db8:acad:1013::1/64
R1(config-subif)# end
```

Step 2: Verify the VRF-Lite configuration.

a. Verify the interface assignments using the **show ip vrf interfaces** command.

```
R1# show ip vrf interfaces
Interface        IP-Address      VRF                    Protocol
Gi0/0/0          10.1.2.1      Customer_A                 up
Se0/1/0          10.1.3.1      Customer_A                 up
```

```
Gi0/0/1.5          10.1.2.1      Customer_B              up
Gi0/0/1.8          10.1.3.1      Customer_B              up
```

b. Verify the VRF routing tables with the **show ip route vrf** *vrf_name* and **show ipv6 route vrf** *vrf_name* command.

```
R1# show ip route vrf Customer_A | begin Gateway
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.2.0/24 is directly connected, GigabitEthernet0/0/0
L       10.1.2.1/32 is directly connected, GigabitEthernet0/0/0
C       10.1.3.0/24 is directly connected, Serial0/1/0
L       10.1.3.1/32 is directly connected, Serial0/1/0

R1# show ipv6 route vrf Customer_B
IPv6 Routing Table - Customer_B - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
<output omitted>
     a - Application
C   2001:DB8:ACAD:1012::/64 [0/0]
    via GigabitEthernet0/0/1.5, directly connected
L   2001:DB8:ACAD:1012::1/128 [0/0]
    via GigabitEthernet0/0/1.5, receive
C   2001:DB8:ACAD:1013::/64 [0/0]
    via GigabitEthernet0/0/1.8, directly connected
L   2001:DB8:ACAD:1013::1/128 [0/0]
    via GigabitEthernet0/0/1.8, receive
L   FF00::/8 [0/0]
    via Null0, receive
```

c. Verify next-hop reachability within each vrf with the **ping vrf** *vrf_name* **address** command.

```
R1# ping vrf Customer_A 10.1.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1# ping vrf Customer_A 2001:db8:acad:1012::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:1012::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
R1#  ping vrf Customer_A 10.1.3.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
R1# ping vrf Customer_A 2001:db8:acad:1013::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:1013::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

Part 3: Configure and Verify Static Routing for Reachability Inside Each VRF

In Part 3, you will configure static routing so that all networks are reachable within their respective VRFs. At the end of this part, R1 should be able to successfully source a ping from interface loopback0 to R3 interface loopback0, and D1 should be able to successfully source a ping from interface VLAN 11 to D2 interface VLAN 11. Once again, the way these networks are being implemented is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: Verify that distant networks are not reachable within each VRF.

In this step, you will check to make sure that distant networks are not reachable from R1 within each VRF.

a. On R1, issue the commands **ping vrf Customer_A 192.168.2.1** and **ping vrf Customer_A 192.168.3.1**. Neither should succeed.

```
R1# ping vrf Customer_A 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1# ping vrf Customer_A 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

b. On R1, issue the commands **ping vrf Customer_A 2001:db8:acad:2000::1** and **ping vrf Customer_A 2001:db8:acad:3000::1**. Neither should succeed.

```
R1# ping vrf Customer_A 2001:db8:acad:2000::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:2000::1, timeout is 2 seconds:
```

```
% No valid route for destination
Success rate is 0 percent (0/1)
R1# ping vrf Customer_A 2001:db8:acad:3000::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:3000::1, timeout is 2 seconds:

% No valid route for destination
Success rate is 0 percent (0/1)
```

c. On R1, issue the commands **ping vrf Customer_B 192.168.2.1** and **ping vrf Customer_B 192.168.3.1**. Neither should succeed.

```
R1# ping vrf Customer_B 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1# ping vrf Customer_B 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

d. On R1, issue the commands **ping vrf Customer_B 2001:db8:acad:2000::1** and **ping vrf Customer_B 2001:db8:acad:3000::1**. Neither should succeed.

```
R1# ping vrf Customer_B 2001:db8:acad:2000::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:2000::1, timeout is 2 seconds:

% No valid route for destination
Success rate is 0 percent (0/1)

R1# ping vrf Customer_B 2001:db8:acad:3000::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:3000::1, timeout is 2 seconds:

% No valid route for destination
Success rate is 0 percent (0/1)
```

Step 2: Configure static routing at R1 for each VRF.

In this step, you will configure R1 so that it can reach distant networks in each VRF. The neighbor systems (D1, D2, R2, and R3) have static routes already configured, so as soon as you correctly install these static routes, there will be full reachability within each VRF.

a. On R1, create static routes for the distant networks in the Customer_A VRF using the **ip route vrf** *vrf_name destination_network next-hop* command.

```
R1(config)# ip route vrf Customer_A 192.168.2.0 255.255.255.0 g0/0/0 10.1.2.2
R1(config)# ip route vrf Customer_A 192.168.3.0 255.255.255.0 s0/1/0 10.1.3.2
R1(config)# ipv6 route vrf Customer_A 2001:db8:acad:2000::/64 g0/0/0 2001:db8:acad:1012::2
R1(config)# ipv6 route vrf Customer_A 2001:db8:acad:3000::/64 s0/1/0 2001:db8:acad:1013::2
```

b. Use the example above to correctly configure fully specified static routes for the Customer_B network.

```
R1(config)# ip route vrf Customer_B 192.168.2.0 255.255.255.0 GigabitEthernet0/0/1.5 10.1.2.2
R1(config)# ip route vrf Customer_B 192.168.3.0 255.255.255.0 GigabitEthernet0/0/1.8 10.1.3.2
R1(config)# ipv6 route vrf Customer_B 2001:DB8:ACAD:2000::/64 GigabitEthernet0/0/1.5 2001:DB8:ACAD:1012::2
R1(config)# ipv6 route vrf Customer_B 2001:DB8:ACAD:3000::/64 GigabitEthernet0/0/1.8 2001:DB8:ACAD:1013::2
```

Step 3: Verify full reachability within each VRF.

a. On R2, ping the IPv4 and IPv6 addresses of R3 interface Loopback0 using a source address of R2 interface Loopback0. All pings should be successful.

```
R2# ping 192.168.3.1 source loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.2.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
R2# ping 2001:db8:acad:3000::1 source loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:3000::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:ACAD:2000::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/2 ms
```

b. On D1, ping the IPv4 and IPv6 addresses of D2 interface VLAN 11 using a source address of D1 interface VLAN 11. All pings should be successful.

```
D1# ping 192.168.3.1 source vlan11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
```

```
Packet sent with a source address of 192.168.2.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms

D1# ping 2001:db8:acad:3000::1 source vlan11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:3000::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:ACAD:2000::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/5/17 ms
```

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs - Final

Router R1

```
R1# show run
Building configuration...

Current configuration : 3151 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R1
!
boot-start-marker
boot-end-marker
!
vrf definition Customer_A
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
vrf definition Customer_B
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
!
```

```
no ip domain lookup
!
login on-success log
!
subscriber templating
!
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface GigabitEthernet0/0/0
 vrf forwarding Customer_A
 ip address 10.1.2.1 255.255.255.0
 negotiation auto
 ipv6 address FE80::1:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::1/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/1.5
 encapsulation dot1Q 5
 vrf forwarding Customer_B
 ip address 10.1.2.1 255.255.255.0
 ipv6 address FE80::1:2 link-local
 ipv6 address 2001:DB8:ACAD:1012::1/64
!
interface GigabitEthernet0/0/1.8
 encapsulation dot1Q 8
 vrf forwarding Customer_B
 ip address 10.1.3.1 255.255.255.0
 ipv6 address FE80::1:3 link-local
 ipv6 address 2001:DB8:ACAD:1013::1/64
!
interface Serial0/1/0
 vrf forwarding Customer_A
 ip address 10.1.3.1 255.255.255.0
 ipv6 address FE80::1:3 link-local
 ipv6 address 2001:DB8:ACAD:1013::1/64
!
interface Serial0/1/1
 no ip address
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
ip route vrf Customer_A 192.168.2.0 255.255.255.0 GigabitEthernet0/0/0 10.1.2.2
ip route vrf Customer_A 192.168.3.0 255.255.255.0 Serial0/1/0 10.1.3.2
ip route vrf Customer_B 192.168.2.0 255.255.255.0 GigabitEthernet0/0/1.5 10.1.2.2
ip route vrf Customer_B 192.168.3.0 255.255.255.0 GigabitEthernet0/0/1.8 10.1.3.2
!
ipv6 route vrf Customer_B 2001:DB8:ACAD:2000::/64 GigabitEthernet0/0/1.5
2001:DB8:ACAD:1012::2
ipv6 route vrf Customer_A 2001:DB8:ACAD:2000::/64 GigabitEthernet0/0/0
2001:DB8:ACAD:1012::2
ipv6 route vrf Customer_B 2001:DB8:ACAD:3000::/64 GigabitEthernet0/0/1.8
2001:DB8:ACAD:1013::2
ipv6 route vrf Customer_A 2001:DB8:ACAD:3000::/64 Serial0/1/0
2001:DB8:ACAD:1013::2
!
control-plane
!
banner motd ^C R1, Implement VRF-Lite ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password cisco123
 logging synchronous
 login
!
end
```

Router R2

```
R2# show run
Building configuration...


Current configuration : 1760 bytes
!
! Last configuration change at 04:14:23 UTC Wed Jan 8 2020
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.2.1 255.255.255.0
 ipv6 address FE80::2:2 link-local
 ipv6 address 2001:DB8:ACAD:2000::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.1.2.2 255.255.255.0
 negotiation auto
 ipv6 address FE80::2:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::2/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0/0 10.1.2.1
!
ipv6 route ::/0 GigabitEthernet0/0/0 2001:DB8:ACAD:1012::1
!
control-plane
!
banner motd ^C R2, Implement VRF-Lite ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password cisco123
 logging synchronous
 login
!
end
```

Router R3

```
R3# show run
Building configuration...

Current configuration : 1821 bytes
!
version 16.9
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.3.1 255.255.255.0
 ipv6 address FE80::3:2 link-local
 ipv6 address 2001:DB8:ACAD:3000::1/64
!
interface GigabitEthernet0/0/0
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface Serial0/1/0
 ip address 10.1.3.2 255.255.255.0
 ipv6 address FE80::3:1 link-local
 ipv6 address 2001:DB8:ACAD:1013::2/64
!
```

```
interface Serial0/1/1
 no ip address
!
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
ip route 0.0.0.0 0.0.0.0 Serial0/1/0 10.1.3.1
!
ipv6 route ::/0 Serial0/1/0 2001:DB8:ACAD:1013::1
!
control-plane
!
banner motd ^C R3, Implement VRF-Lite ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password cisco123
 logging synchronous
 login
!
end
```

Switch D1

```
D1# show run
Building configuration...

Current configuration : 9267 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
```

```
no platform punt-keepalive disable-kernel-core
!
hostname D1
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
ip routing
!
no ip domain lookup
!
login on-success log
ipv6 unicast-routing
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL
SAMPLED DATA, RPF Failed
```

```
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
```

```
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 no switchport
 ip address 10.1.2.2 255.255.255.0
 ipv6 address FE80::D1:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::2/64
!
interface GigabitEthernet1/0/6
 shutdown
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 shutdown
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
```

```
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 switchport access vlan 11
 switchport mode access
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
 shutdown
!
interface GigabitEthernet1/1/2
 shutdown
!
interface GigabitEthernet1/1/3
 shutdown
!
interface GigabitEthernet1/1/4
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan11
 ip address 192.168.2.1 255.255.255.0
 ipv6 address FE80::D1:2 link-local
 ipv6 address 2001:DB8:ACAD:2000::1/64
```

```
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1/0/5 10.1.2.1
!
ipv6 route ::/0 GigabitEthernet1/0/5 2001:DB8:ACAD:1012::1
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C D1, Implement VRF-Lite ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password cisco123
 logging synchronous
 login
line vty 5 15
 login
!
end
```

Switch D2

```
D2# show run
Building configuration...

Current configuration : 9267 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
```

```
no platform punt-keepalive disable-kernel-core
!
hostname D2
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
ip routing
!
no ip domain lookup
!
login on-success log
ipv6 unicast-routing
!
license boot level ipservicesk9
!
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
```

```
 description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL
SAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
 description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
 description L2 LVX control packets
class-map match-any system-cpp-police-forus
 description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
 description MCAST END STATION
class-map match-any system-cpp-police-multicast
 description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
 description L2 control
class-map match-any system-cpp-police-dot1x-auth
 description DOT1X Auth
class-map match-any system-cpp-police-data
 description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
 description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
 description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
 description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
 description DHCP snooping
class-map match-any system-cpp-police-system-critical
 description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
```

```
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 no switchport
 ip address 10.1.3.2 255.255.255.0
 ipv6 address FE80::D2:1 link-local
 ipv6 address 2001:DB8:ACAD:1013::2/64
!
interface GigabitEthernet1/0/6
 shutdown
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 shutdown
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
```

```
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 switchport access vlan 11
 switchport mode access
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
 shutdown
!
interface GigabitEthernet1/1/2
 shutdown
!
interface GigabitEthernet1/1/3
 shutdown
!
interface GigabitEthernet1/1/4
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan11
 ip address 192.168.3.1 255.255.255.0
 ipv6 address FE80::D2:2 link-local
 ipv6 address 2001:DB8:ACAD:3000::1/64
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1/0/5 10.1.3.1
!
ipv6 route ::/0 GigabitEthernet1/0/5 2001:DB8:ACAD:1013::1
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C D2, Implement VRF-Lite ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password cisco123
 logging synchronous
 login
line vty 5 15
 login
!
end
```

Switch A1

```
A1# show run
Building configuration...

Current configuration : 1883 bytes
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname A1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
system mtu routing 1500
!
no ip domain-lookup
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface FastEthernet0/1
 switchport access vlan 5
 switchport mode access
!
interface FastEthernet0/2
 shutdown
!
interface FastEthernet0/3
 switchport access vlan 8
 switchport mode access
!
interface FastEthernet0/4
 shutdown
!
interface FastEthernet0/5
 shutdown
!
interface FastEthernet0/6
 shutdown
!
interface FastEthernet0/7
 shutdown
!
interface FastEthernet0/8
 shutdown
!
interface FastEthernet0/9
```

```
 shutdown
!
interface FastEthernet0/10
 shutdown
!
interface FastEthernet0/11
 switchport mode trunk
 switchport nonegotiate
!
interface FastEthernet0/12
 shutdown
!
interface FastEthernet0/13
 shutdown
!
interface FastEthernet0/14
 shutdown
!
interface FastEthernet0/15
 shutdown
!
interface FastEthernet0/16
 shutdown
!
interface FastEthernet0/17
 shutdown
!
interface FastEthernet0/18
 shutdown
!
interface FastEthernet0/19
 shutdown
!
interface FastEthernet0/20
 shutdown
!
interface FastEthernet0/21
 shutdown
!
interface FastEthernet0/22
 shutdown
!
interface FastEthernet0/23
 shutdown
```

```
!
interface FastEthernet0/24
 shutdown
!
interface GigabitEthernet0/1
 shutdown
!
interface GigabitEthernet0/2
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip http server
ip http secure-server
!
banner motd ^C A1, Implement VRF-Lite ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password cisco123
 logging synchronous
 login
line vty 5 15
 login
```