PRACTICAL NO: 1

Aim: Configure IP SLA Feature.

Introduction :

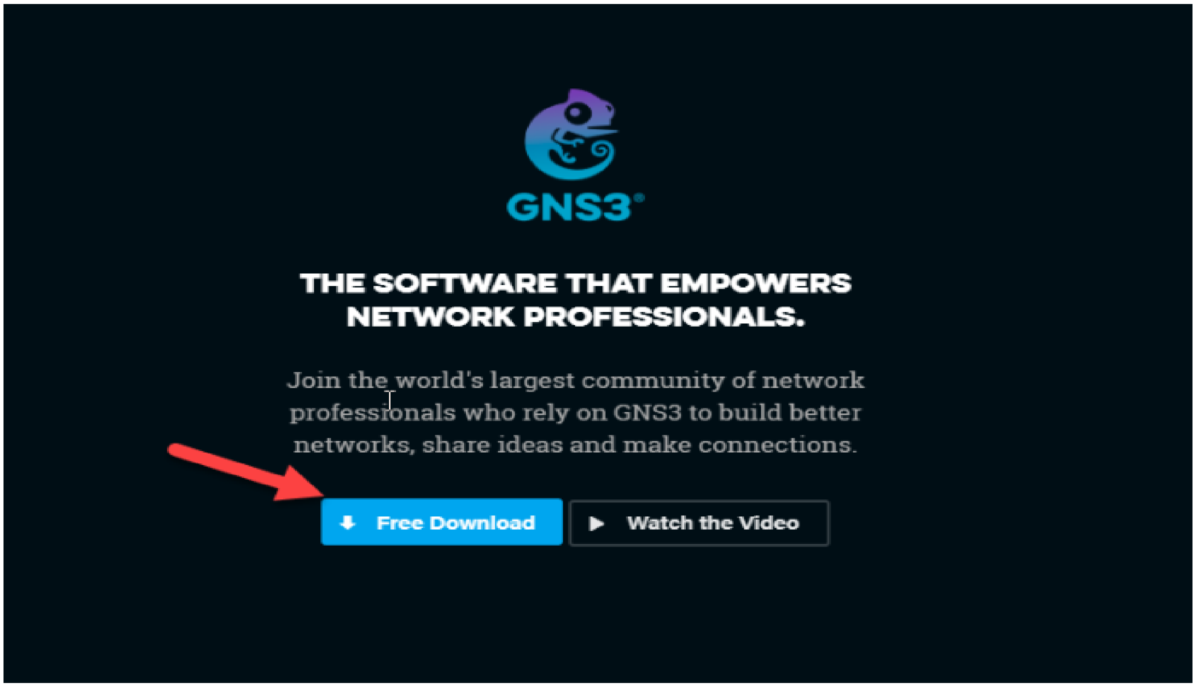
GNS3, short for Graphical Network Simulator-3, is a powerful and popular open-source network emulation software. Network engineers, students, and IT specialists can create, test, and troubleshoot network setups without requiring real hardware by using it to visually simulate complicated networks. GNS3 offers a platform to install virtual machines, routers, switches, and other network equipment, simulating actual network scenarios with its user-friendly graphical interface. GNS3 provides a versatile and effective environment for network simulation and experimentation, regardless of whether you're preparing for real-world deployments, experimenting with network configurations, or studying for certifications.

Installation :

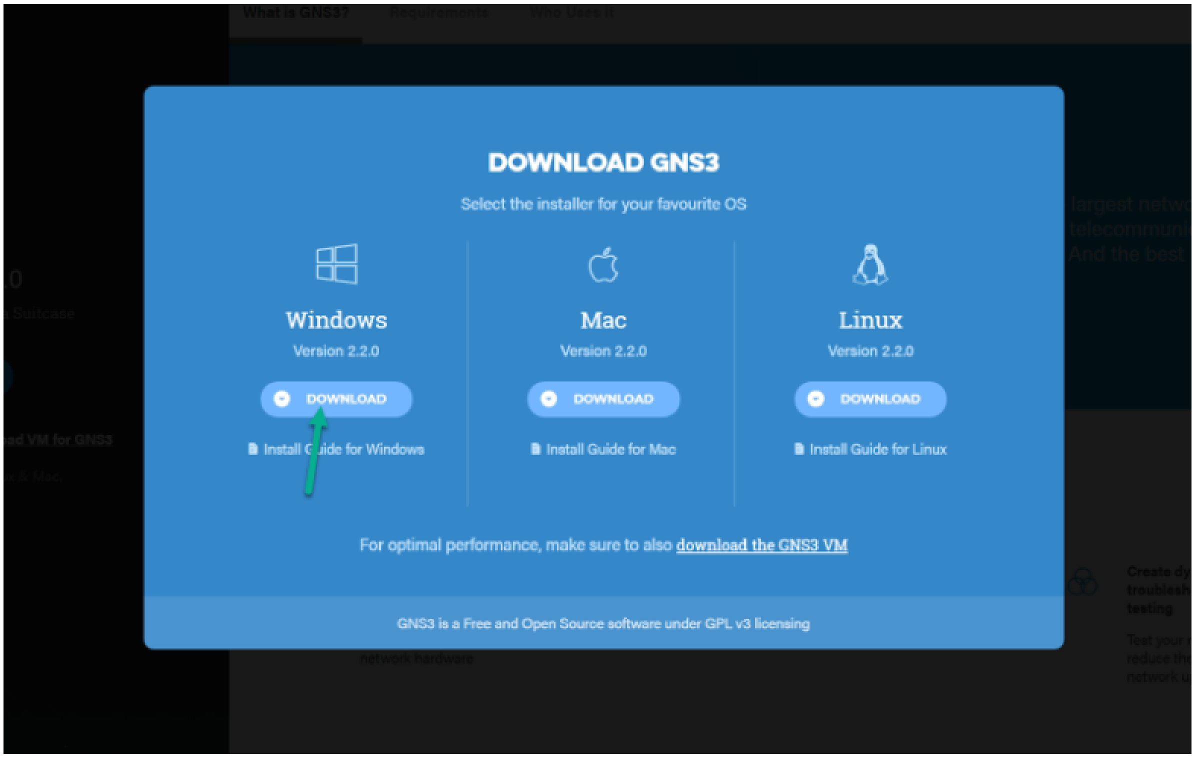
The following are the minimum requirements for a Windows GNS3 environment

Item	Requirement
Operating System	Windows 7 (64 bit) or later
Processor	2 or more Logical cores
Virtualization	Virtualization extensions required. You may need to enable this via your computer's BIOS.
Memory	4 GB RAM
Storage	1 GB available space (Windows Installation is < 200 MB).
Additional Notes	You may need additional storage for your operating system and device images.

Follow these steps to download GNS3 to your device. Using a web browser, browse to https://gns3.com and click the **Free Download** link



You will be prompted to select the version of GNS3 to download. In this guide, we will select the Windows installation. Click the Download button to download the GNS3-all-in-one package.

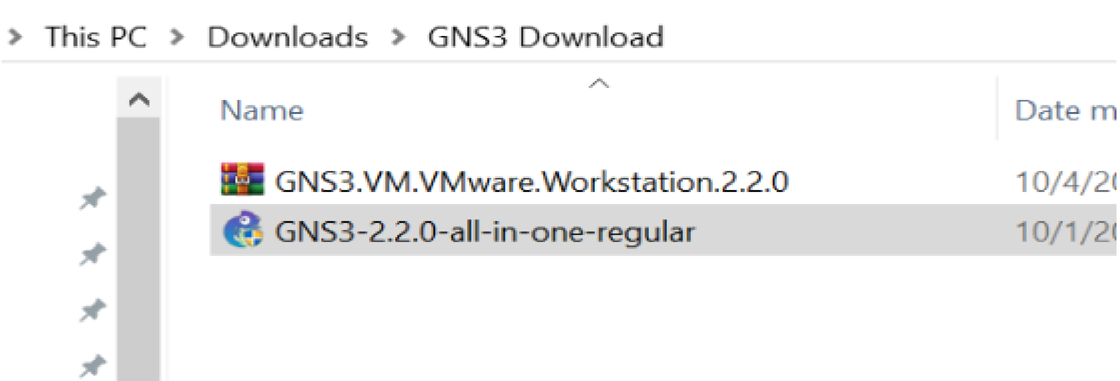


The GNS3-all-in-one package will automatically be downloaded to your PC.

Install GNS3

Follow these steps to download GNS3 on a local Windows PC.

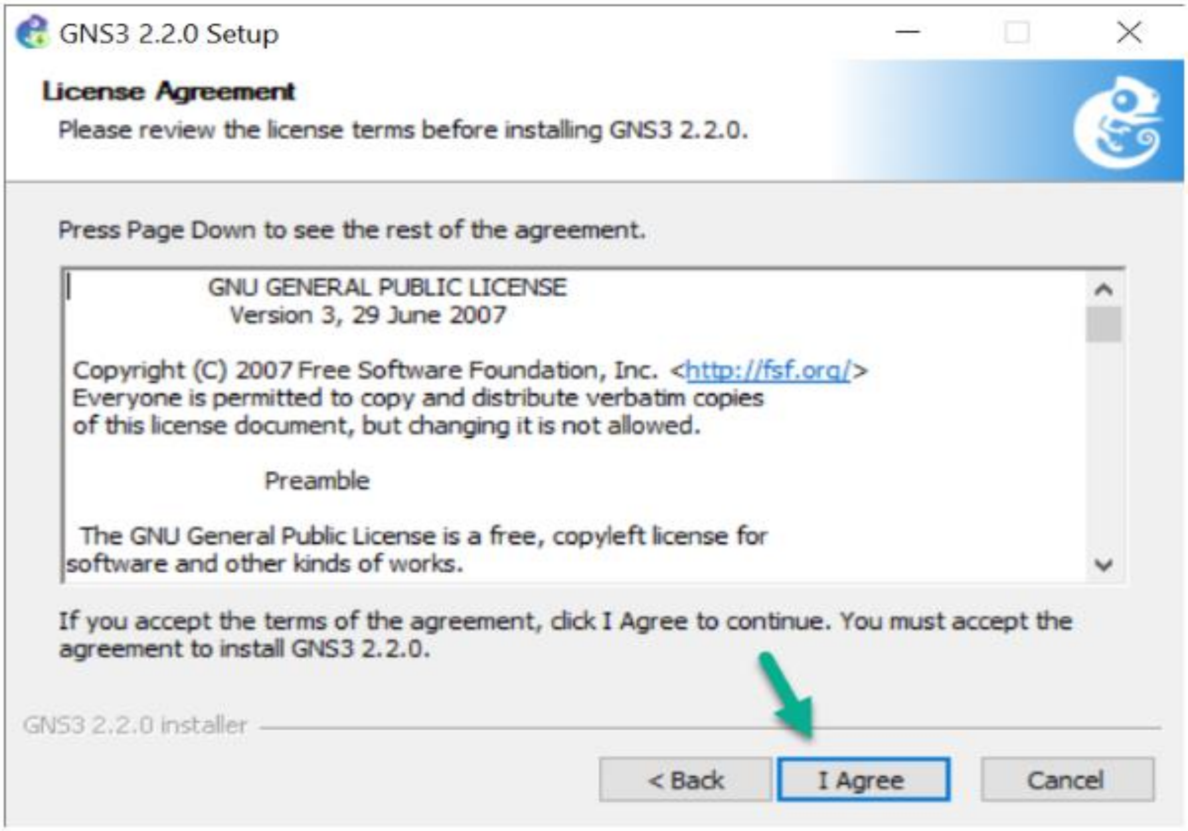
Navigate in Windows Explorer to your Downloads folder and then double-click the GNS3-2.2.0-all-in-one-regular.exe file



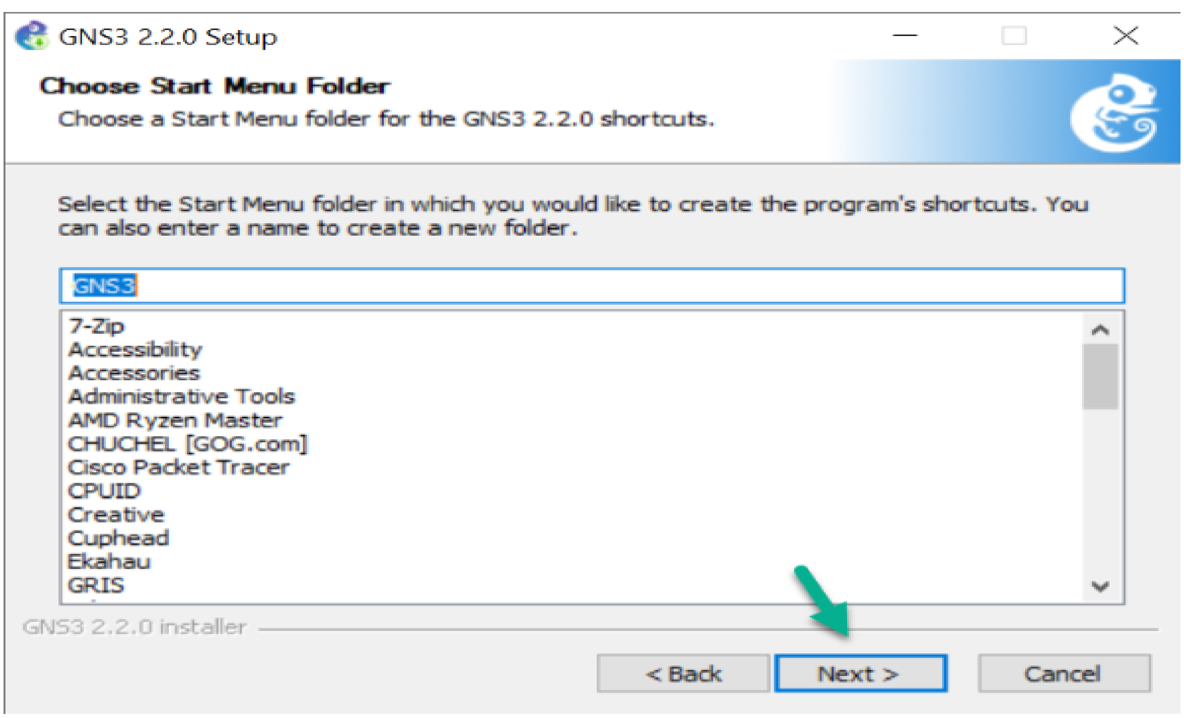
If displayed, click the **Run** button to start the GNS3 installation (it's not possible to take a screenshot of the UAC prompt for this, but allow setup to continue)

The GNS3 Setup wizard displays. Click **Next >** to start the installation

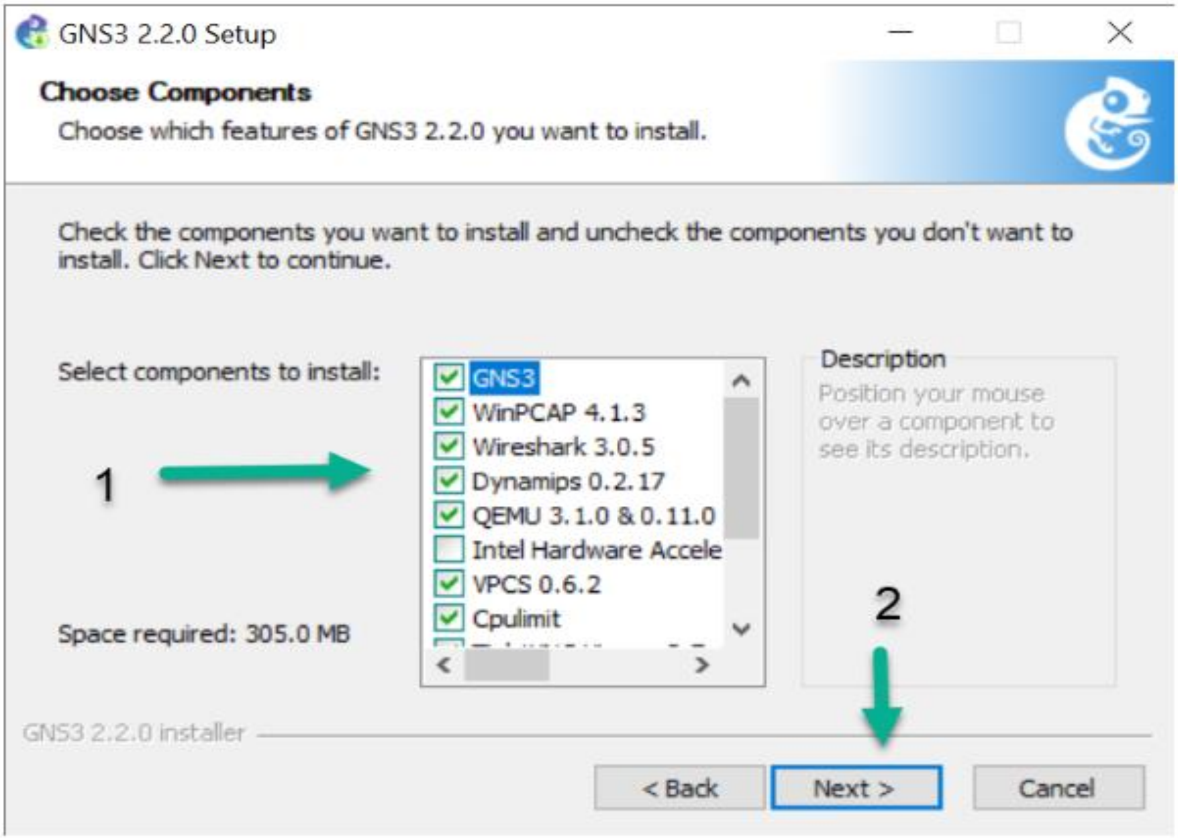
GNS3 is free open source software distributed under the GNU General Public License Version 3. Read the license agreement, and if you agree with the contents, click the **I Agree** button to continue the installation

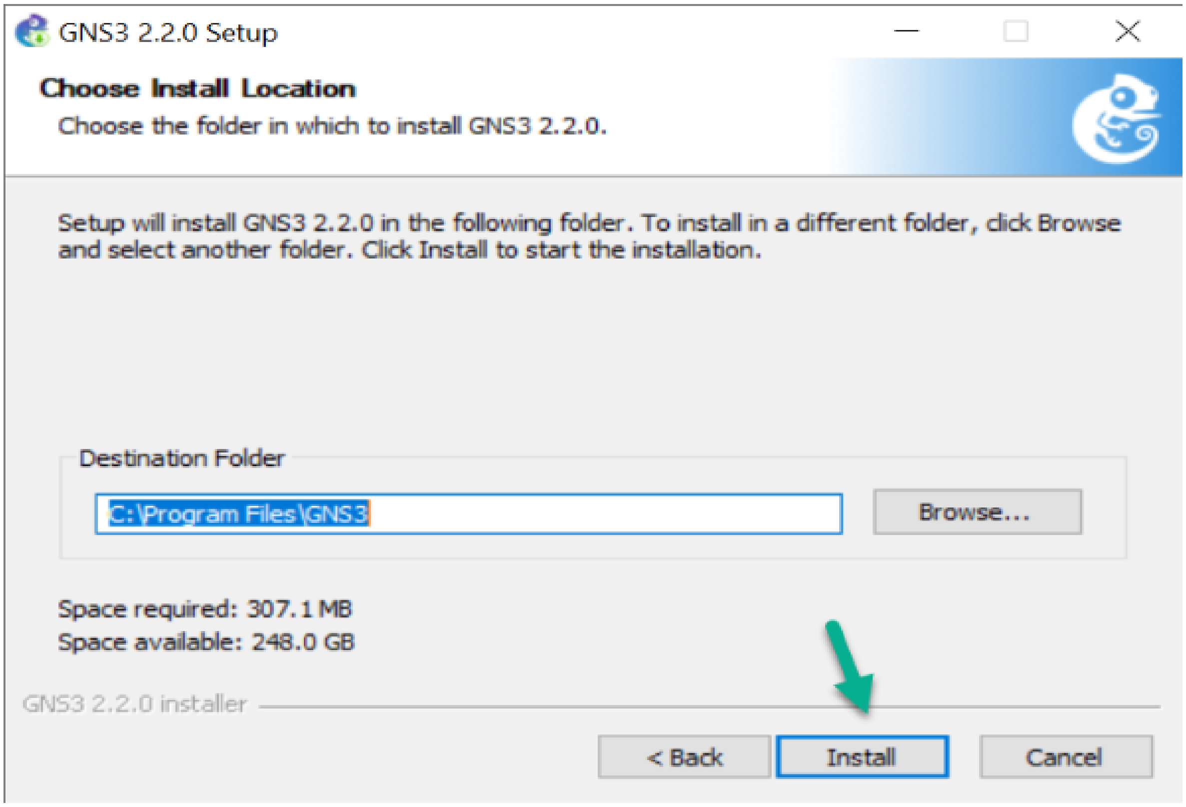


Select the Start Menu folder for the GNS3 shortcut. The default is the GNS3 folder. Click **Next >** to continue the installation



Leave all software selections at their default selection and click **Next >** to continue the installation



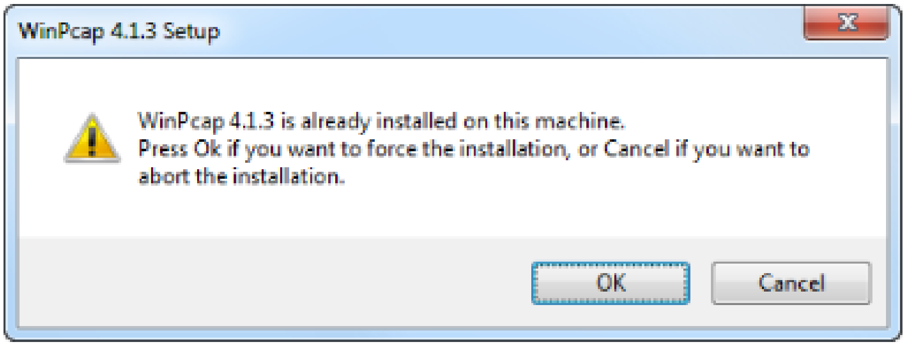


Choose an install location. The default location is C:\Program Files\GNS3. Then click **Install**

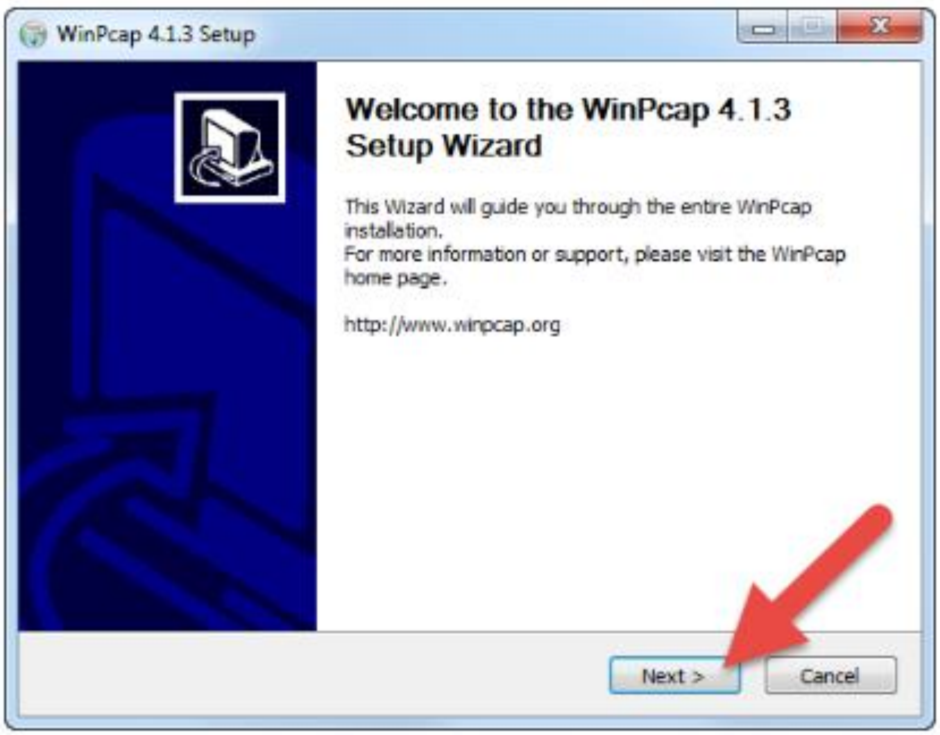
The output displayed will depend on what you have selected to install.

- If you are new to GNS3 and this is a new install, go to the next step.
- For WinPCAP install instructions go to the next step.
- For Wireshark install instructions
- For SolarWinds Response Time Viewer install instructions
- For GNS3 install instructions

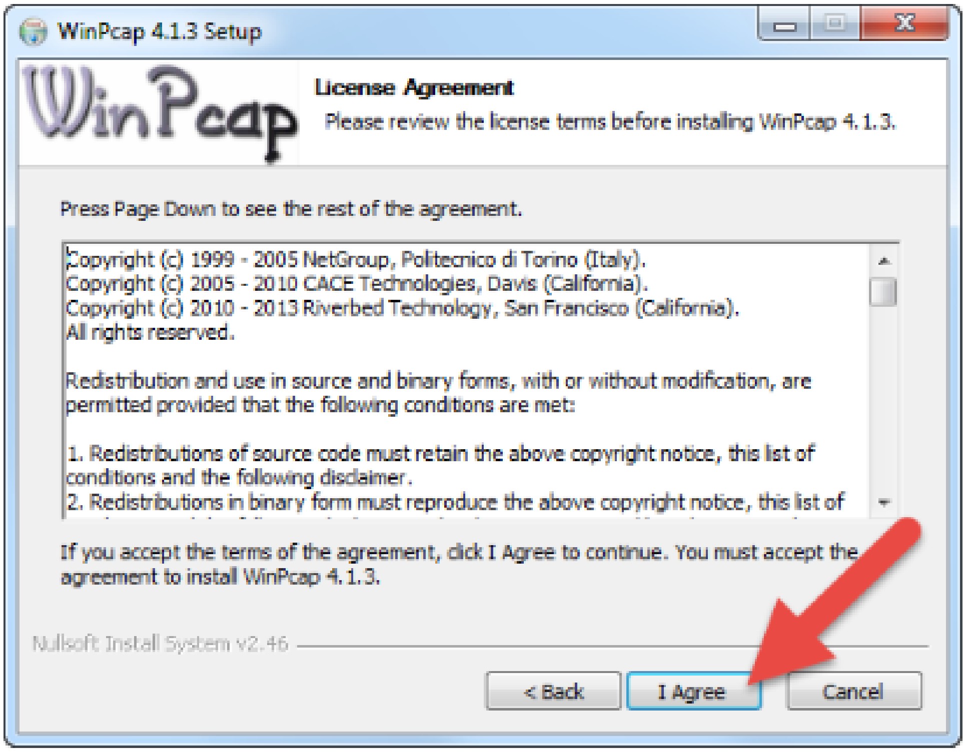
If WinPcap is already installed, a warning message is displayed. It is not necessary to reinstall WinPcap. Click **Cancel** if you do not want to reinstall WinPcap and go directly to the next step. Click **OK** to continue re-installing WinPcap:



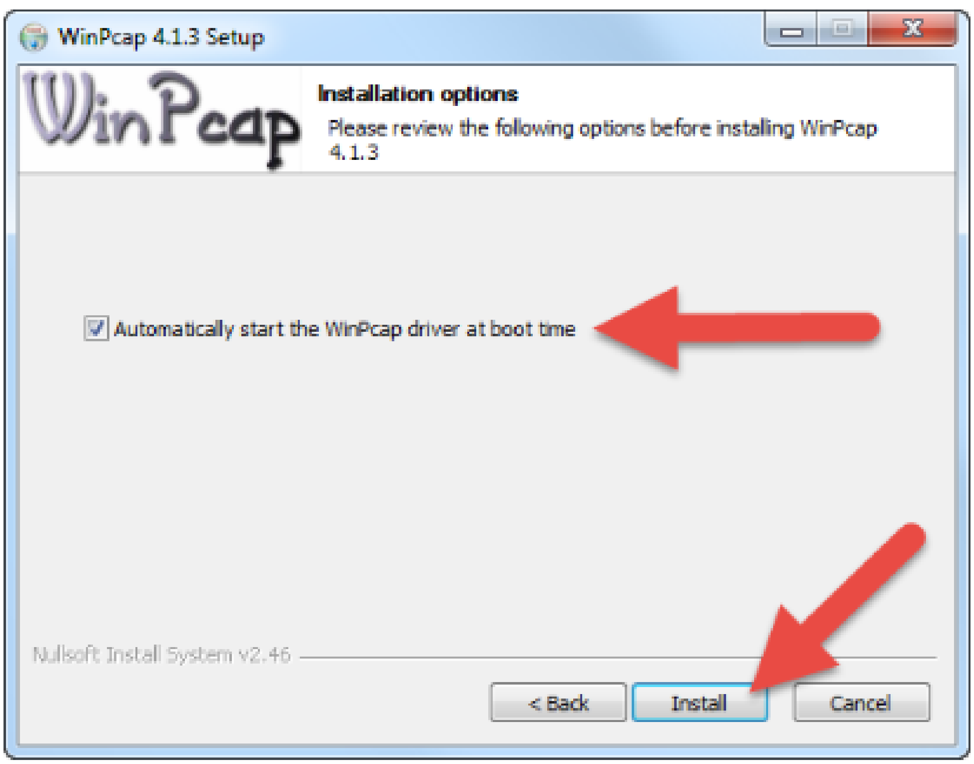
The WinPcap installation wizard displays. Click **Next >** to continue the installation



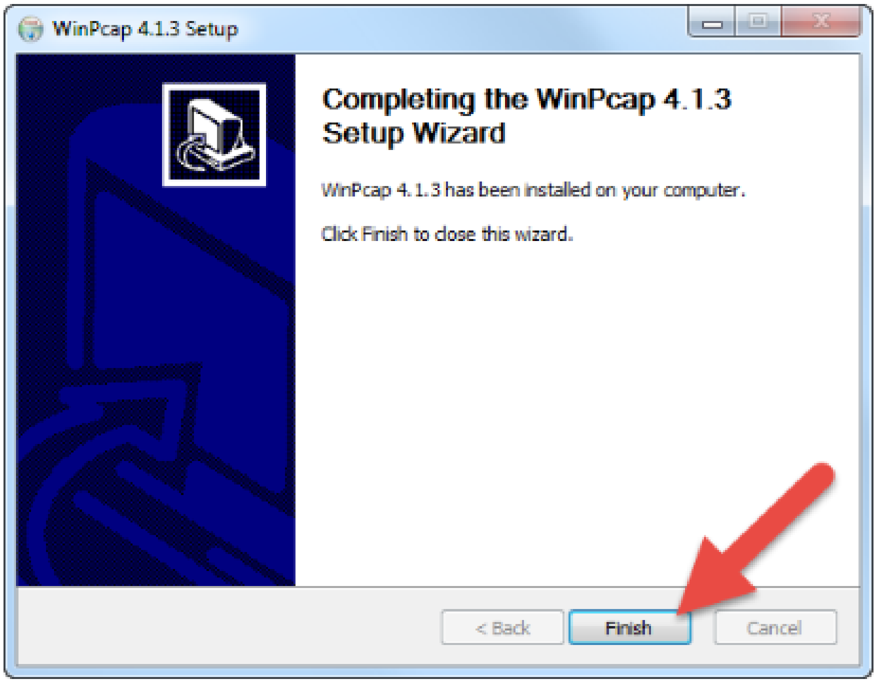
The WinPcap License Agreement displays. Read the agreement and if you agree, click the **I Agree** button to continue with the installation



Leave the checkbox **Automatically start the WinPcap driver at boot time** checked and click **Install**

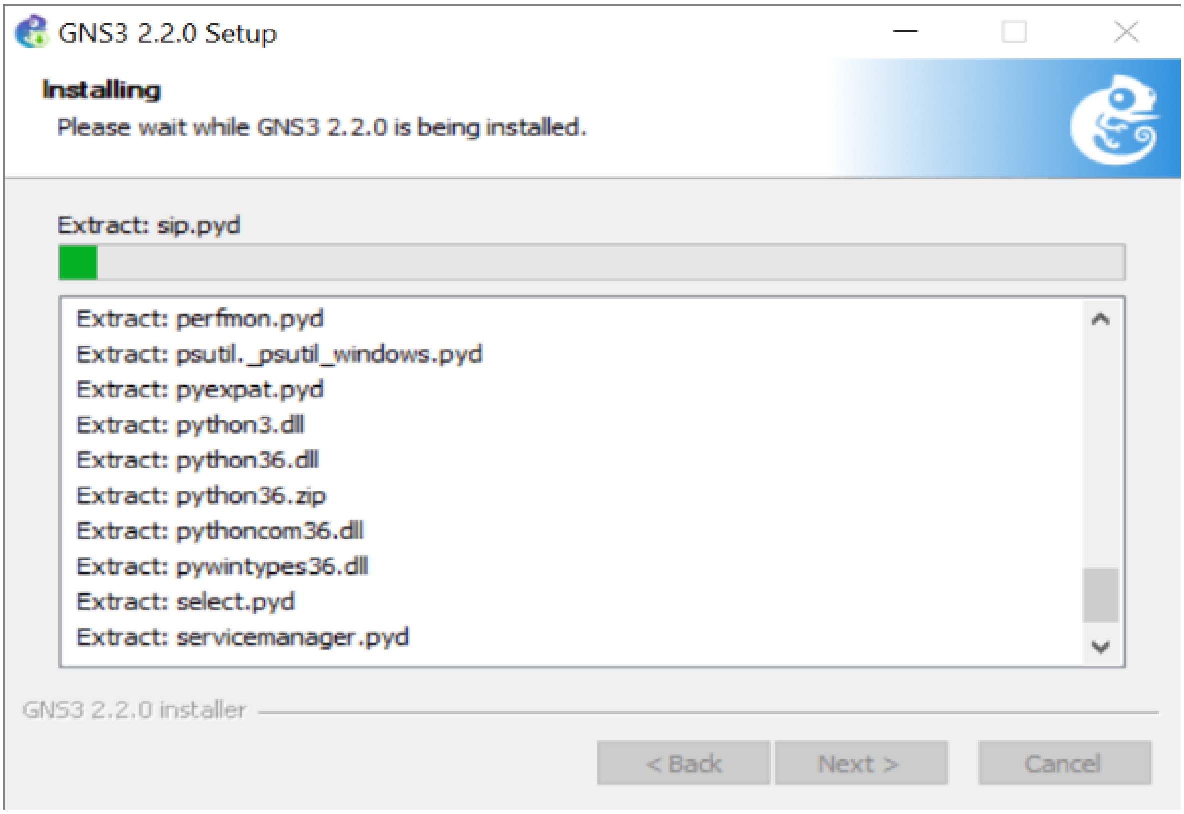


The WinPcap installation continues. Click **Finish** to complete the installation

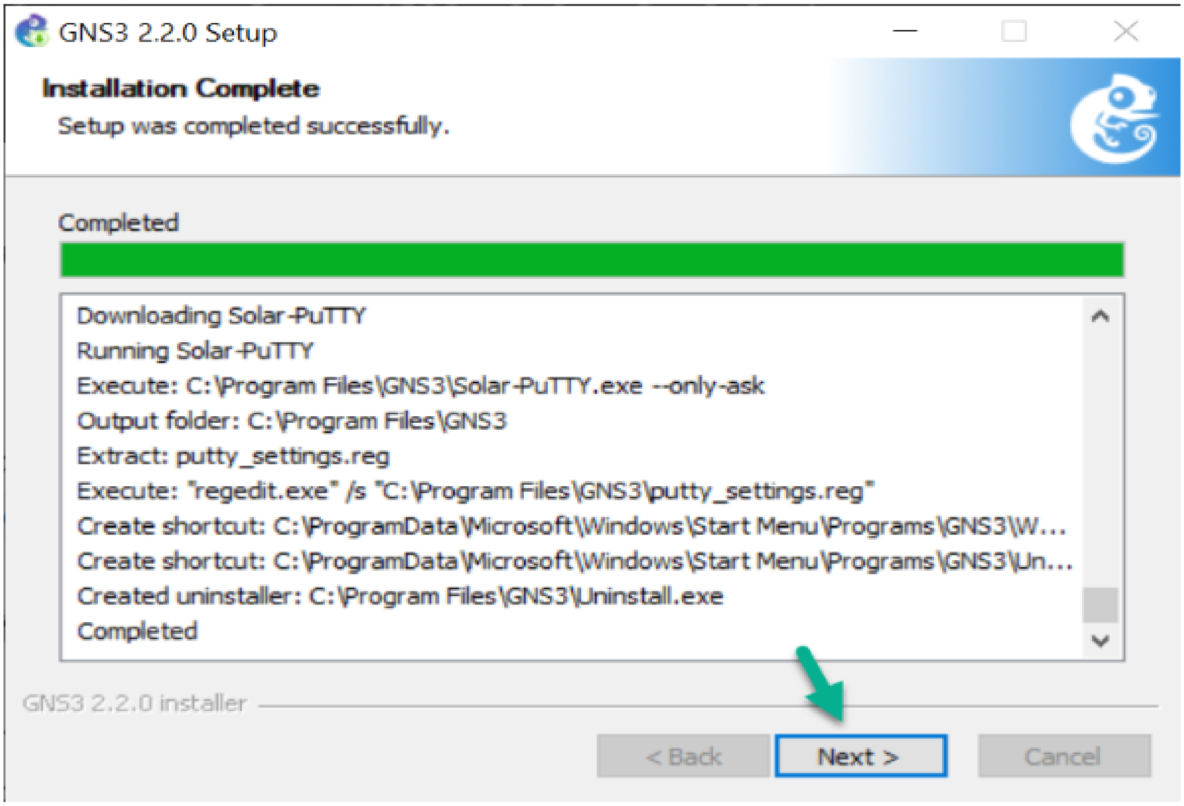


If you selected that Wireshark should be installed, the GNS3 setup software will download the Wireshark install files. Wait for the process to complete.

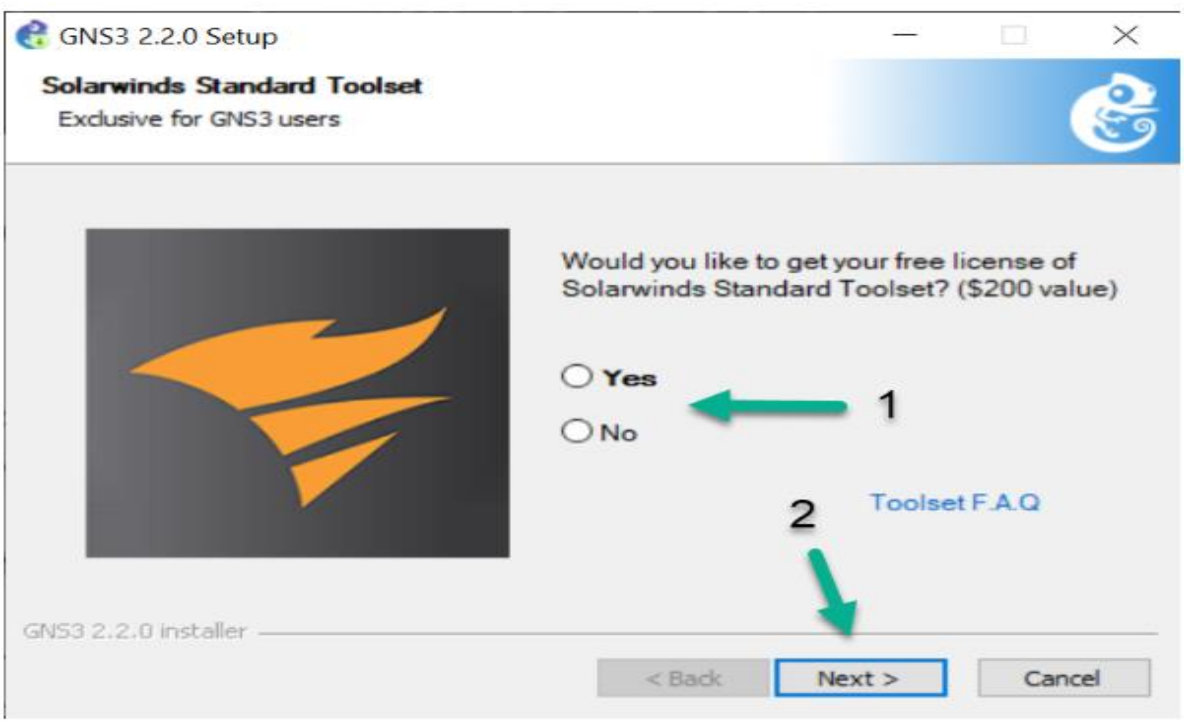
GNS3 will then perform a silent install of Wireshark. Wait for the process to complete

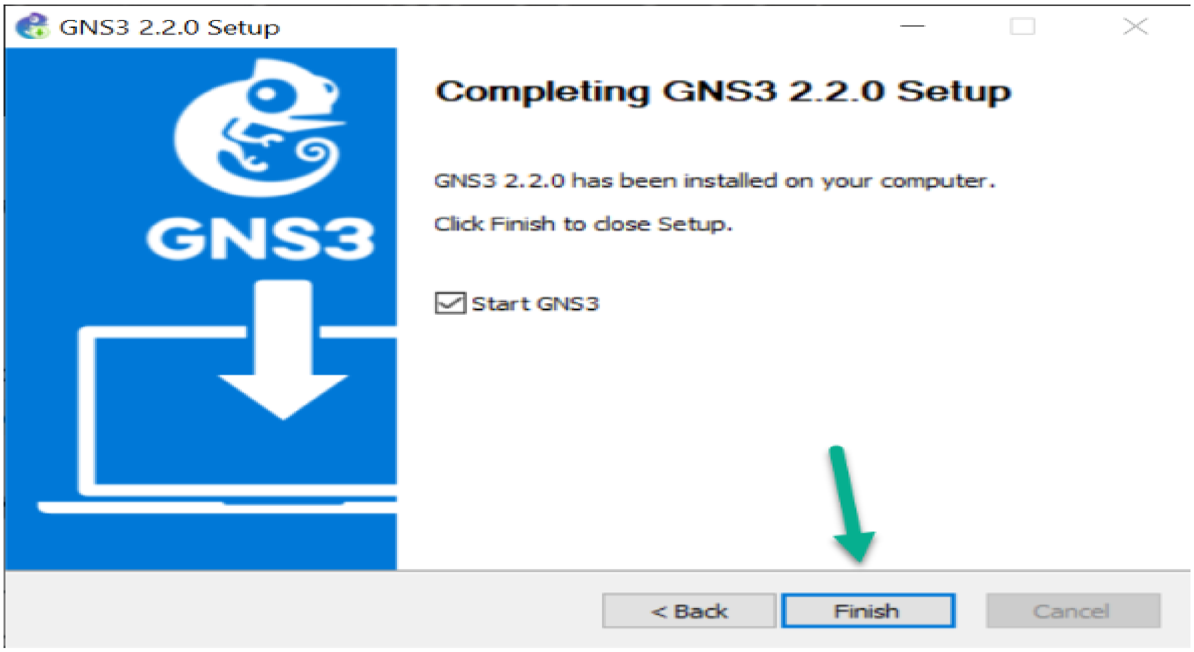


Once the core GNS3 software (and any optional selected items) is installed, Click **Next >**



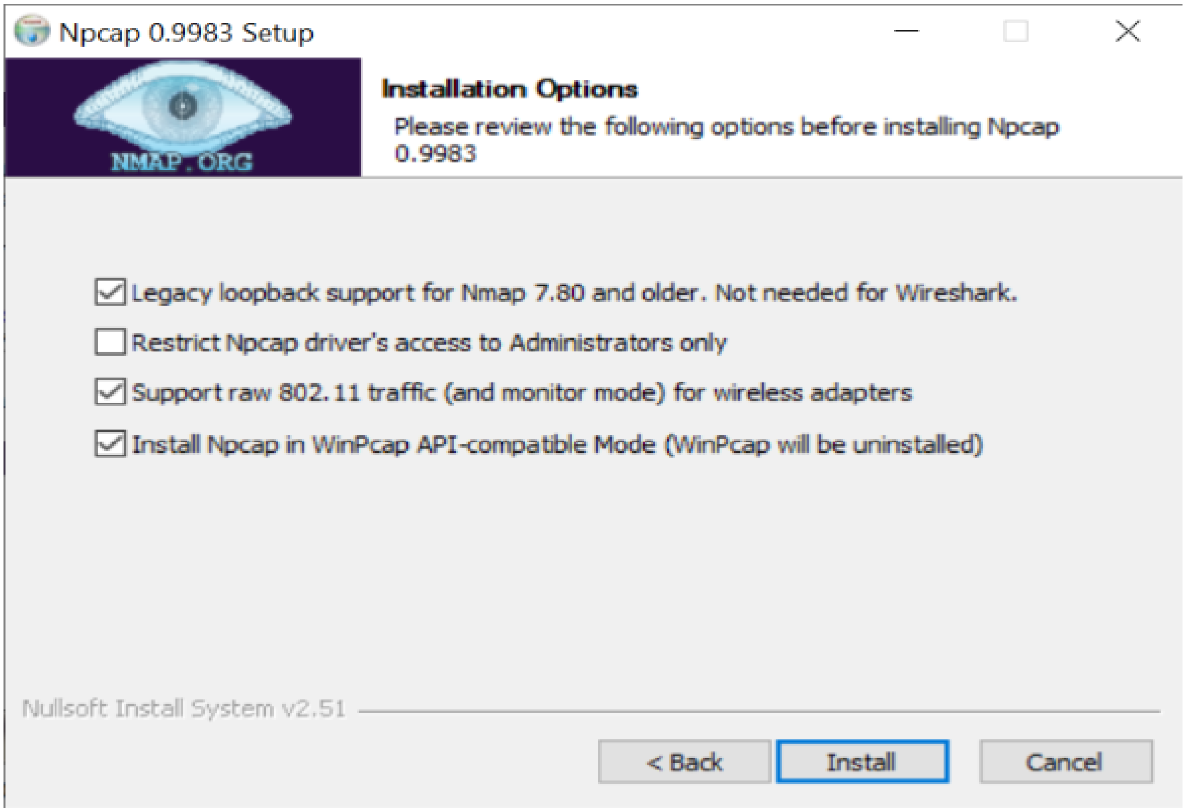
Optional: If you are interested, install the Solarwinds Standard Toolset. This is free evaluation (a $200 value). Otherwise Select **No** and click **Next** > to continue



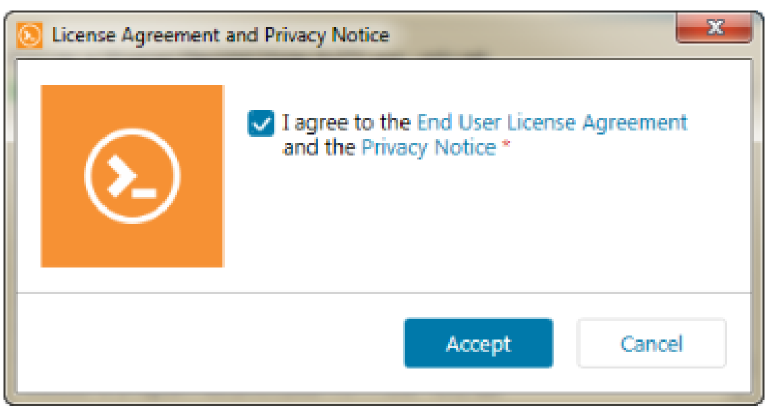


Congratulations! You have successfully installed GNS3. A browser window will open showing additional help and options. Leave the **Start GNS3** checkbox enabled and click **Finish** to complete the GNS3 installation. Click here to for a walkthrough of using the Setup Wizard to configure and use the Local Server

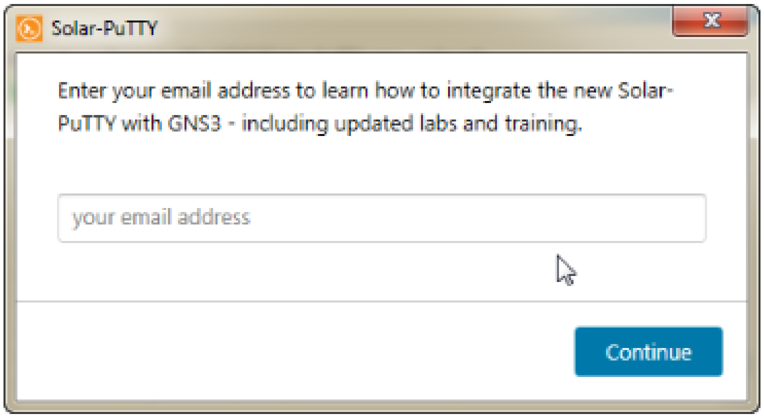
When installing Npcap with WinPcap, make sure to enable the "WinPcap API-compatible Mode" option before completing the installation



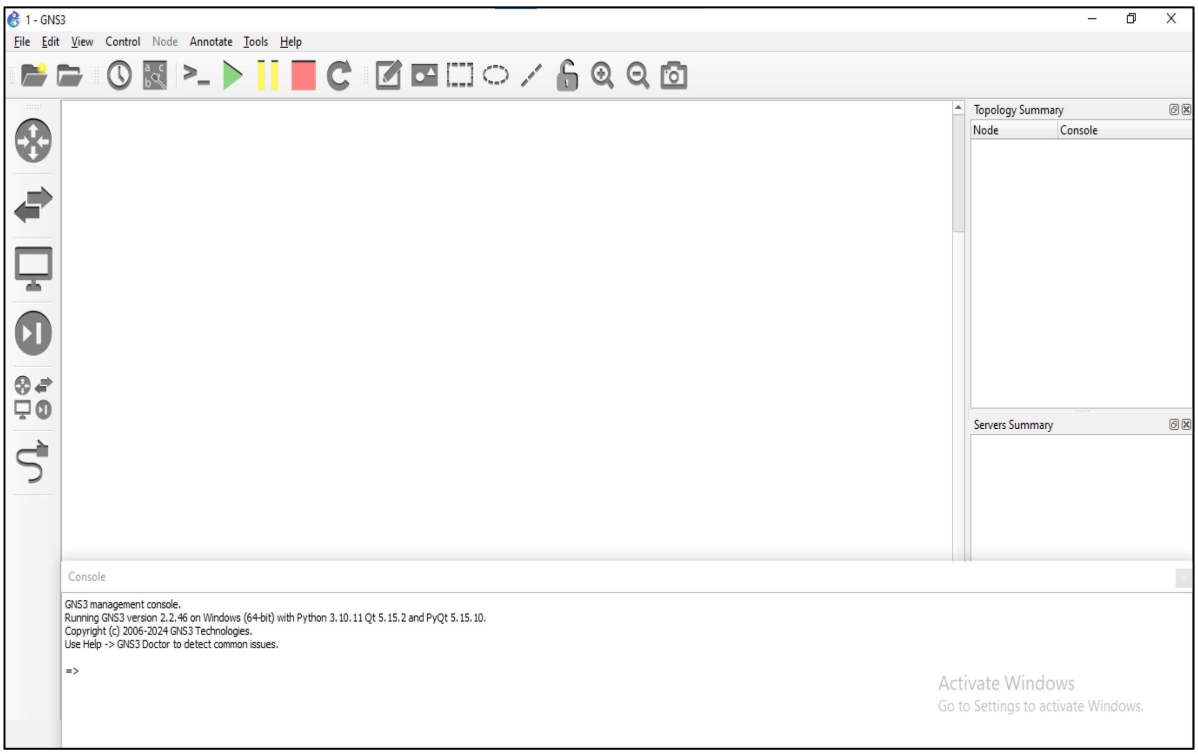
Click **Install**, and then reboot Windows a final time.



After clicking ‘Accept’, you’ll be prompted to enter a valid email address

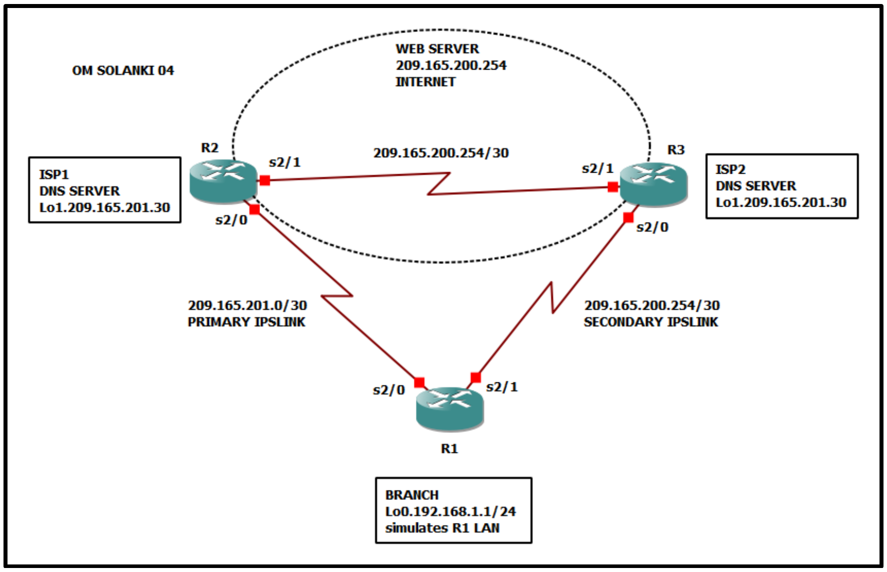


Clicking Continue will finish this part of the installation, and takes you back to the main GNS3 installation.



So, this is how the interface of GNS3 looks as you start a new project.

Topology :



<u>Objectives:</u>

→ Configure and verify the IP SLA feature.

<u>Step 1: Prepare the routers and configure the router hostname and interface addresses.</u>

a. Configure routers.

Router R1:

```
en
conf t
hostname R1
int Loopback0
description R1 LAN
ip add 192.168.1.1 255.255.255.0
int se2/0
description R1-->ISP1
ip add 209.165.201.2 255.255.255.252
clock rate 128000
bandwidth 128
no shut
int se2/1
description R1-->ISP2
ip add 209.165.202.130 255.255.255.252
bandwidth 128
no shut
end
```

```
R1#en
R1#conf t
Enter configuration commands, one per line.  End with CNT
R1(config)#hostname R1
R1(config)#int Loopback0
R1(config-if)#description R1 LAN
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#int se2/0
R1(config-if)#description R1-->ISP1
R1(config-if)#ip add 209.165.201.2 255.255.255.252
R1(config-if)#clock rate 128000
R1(config-if)#bandwidth 128
R1(config-if)#no shut
R1(config-if)#int se2/1
R1(config-if)#description R1-->ISP2
R1(config-if)#ip add 209.165.202.130 255.255.255.252
R1(config-if)#bandwidth 128
R1(config-if)#no shut
```

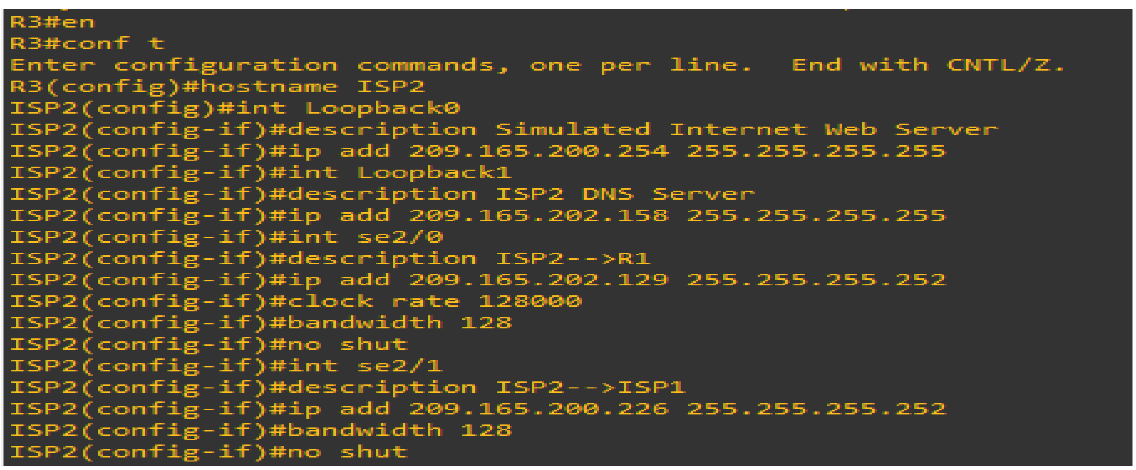
Router R2:

```
en
conf t
hostname ISP1
int Loopback0
description Simulated Internet Web Server
ip add 209.165.200.254 255.255.255.255
int Loopback1
description ISP1 DNS Server
ip add 209.165.201.30 255.255.255.255
int se2/0
description ISP1-->R1
ip add 209.165.201.1 255.255.255.252
bandwidth 128
no shut
int se2/1
description ISP1-->ISP2
ip add 209.165.200.225 255.255.255.252
clock rate 128000
bandwidth 128
no shut
end
```

```
R2#en
R2#conf t
Enter configuration commands, one per line.  End with CNTL
R2(config)#hostname ISP1
ISP1(config)#int Loopback0
ISP1(config-if)#description Simulated Internet Web Server
ISP1(config-if)#ip add 209.165.200.254 255.255.255.255
ISP1(config-if)#int Loopback1
ISP1(config-if)#description ISP1 DNS Server
ISP1(config-if)#ip add 209.165.201.30 255.255.255.255
ISP1(config-if)#int se2/0
ISP1(config-if)#description ISP1-->R1
ISP1(config-if)#ip add 209.165.201.1 255.255.255.252
ISP1(config-if)#bandwidth 128
ISP1(config-if)#no shut
ISP1(config-if)#int se2/1
ISP1(config-if)#description ISP1-->ISP2
ISP1(config-if)#ip add 209.165.200.225 255.255.255.252
ISP1(config-if)#clock rate 128000
ISP1(config-if)#bandwidth 128
ISP1(config-if)#no shut
```

Router R3:

```
en
conf t
hostname ISP2
int Loopback0
description Simulated Internet Web Server
ip add 209.165.200.254 255.255.255.255
int Loopback1
description ISP2 DNS Server
ip add 209.165.202.158 255.255.255.255
int se2/0
description ISP2-->R1
ip add 209.165.202.129 255.255.255.252
clock rate 128000
bandwidth 128
no shut
int se2/1
description ISP2-->ISP1
ip add 209.165.200.226 255.255.255.252
bandwidth 128
no shut
end
```

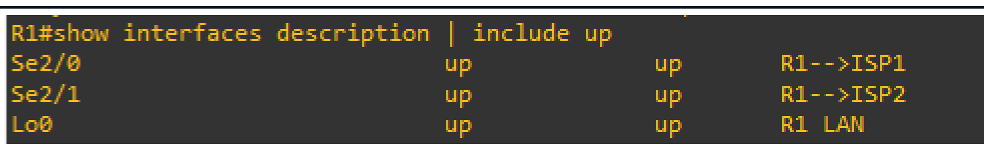


```
R3#en
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname ISP2
ISP2(config)#int Loopback0
ISP2(config-if)#description Simulated Internet Web Server
ISP2(config-if)#ip add 209.165.200.254 255.255.255.255
ISP2(config-if)#int Loopback1
ISP2(config-if)#description ISP2 DNS Server
ISP2(config-if)#ip add 209.165.202.158 255.255.255.255
ISP2(config-if)#int se2/0
ISP2(config-if)#description ISP2-->R1
ISP2(config-if)#ip add 209.165.202.129 255.255.255.252
ISP2(config-if)#clock rate 128000
ISP2(config-if)#bandwidth 128
ISP2(config-if)#no shut
ISP2(config-if)#int se2/1
ISP2(config-if)#description ISP2-->ISP1
ISP2(config-if)#ip add 209.165.200.226 255.255.255.252
ISP2(config-if)#bandwidth 128
ISP2(config-if)#no shut
```

b. Verify the configuration by using the show interfaces description command.

Router R1:

```
Show interfaces description | include up
```



```
R1#show interfaces description | include up
Se2/0                          up             up       R1-->ISP1
Se2/1                          up             up       R1-->ISP2
Lo0                            up             up       R1 LAN
```

Router ISP1 (R2):

```
Show interfaces description | include up
```

```
ISP1#show interfaces description | include up
Se2/0                          up             up       ISP1-->R1
Se2/1                          up             up       ISP1-->ISP2
Lo0                            up             up       Simulated Internet Web Server
Lo1                            up             up       ISP1 DNS Server
```

Router ISP2 (R3):

```
Show interfaces description | include up
```

```
ISP2#show interfaces description | include up
Se2/0                          up             up       ISP2-->R1
Se2/1                          up             up       ISP2-->ISP1
Lo0                            up             up       Simulated Internet Web Server
Lo1                            up             up       ISP2 DNS Server
```

c. 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.

Router R1:

```
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
```

Router ISP1 (R2):

```
config t
router eigrp 1
network 209.165.200.224 0.0.0.3
network 209.165.201.0 0.0.0.31
no auto-summary
exit
ip route 192.168.1.0 255.255.255.0 209.165.201.2
```

```
ISP1(config)#router eigrp 1
ISP1(config-router)#network 209.165.200.224 0.0.0.3
ISP1(config-router)#network 209.165.202.128 0.0.0.31
ISP1(config-router)#no auto-summary
ISP1(config-router)#exit
ISP1(config)#ip route 192.168.1.0 255.255.255.0 209.165.202.130
```

Router ISP2 (R3):

```
config t
router eigrp 1
```

```
network 209.165.200.224 0.0.0.3
network 209.165.202.128 0.0.0.31
no auto-summary
exit
ip route 192.168.1.0 255.255.255.0 209.165.202.130
```

```
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)#ip route 192.168.1.0 255.255.255.0 209.165.202.130
```

Step 2: Verify Server Reachability.

a. 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.

```
1# tclsh
R1(tcl)# foreach address {
+>(tcl)# 209.165.200.254
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } {
+>(tcl)# ping $address source 192.168.1.1
+>(tcl)# }
```

```
R1(tcl)#foreach address {
+>(tcl)#209.165.200.254
+>(tcl)#209.165.201.30
+>(tcl)#209.165.202.158
+>(tcl)#} {
+>(tcl)#ping $address source 192.168.1.1
+>(tcl)#}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.254, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/50/68 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.30, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/44/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.158, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/52/68 ms
```

b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server.

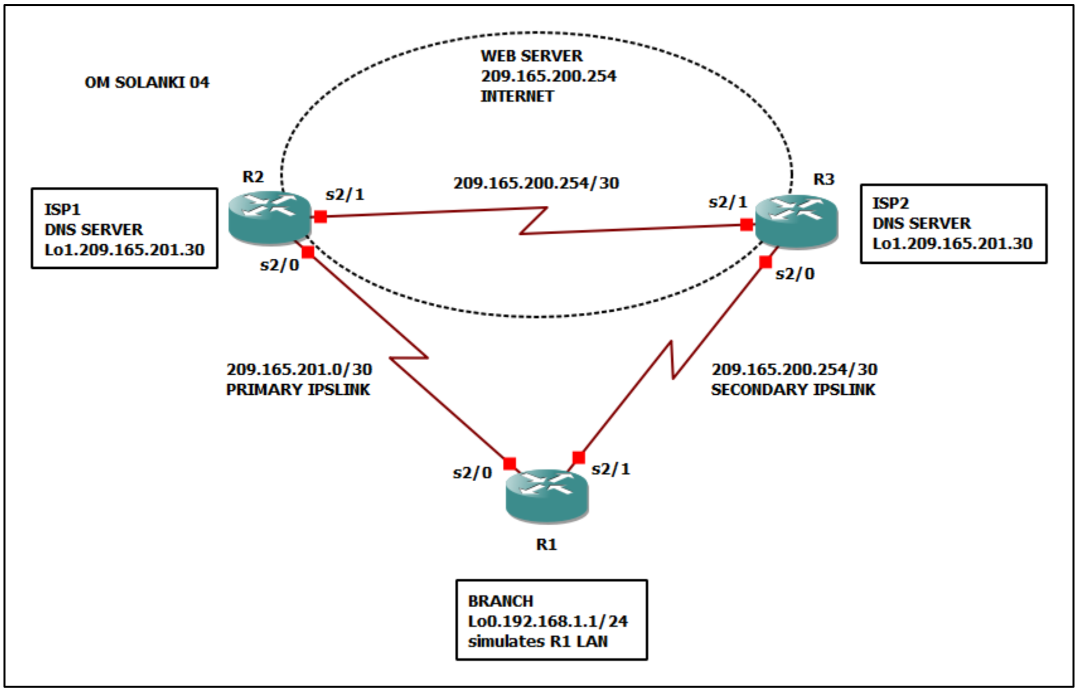
```
1(tcl)# foreach address {
+>(tcl)# 209.165.200.254
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } {
+>(tcl)# ping $address source 192.168.1.1
+>(tcl)# }
```

```
R1(tcl)#foreach address {
+>(tcl)#209.165.200.254
+>(tcl)#209.165.201.30
+>(tcl)#209.165.202.158
+>(tcl)#} {
+>(tcl)#trace $address source 192.168.1.1
+>(tcl)#}

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.201.1 20 msec 32 msec 32 msec
Type escape sequence to abort.
Tracing the route to 209.165.201.30

  1 209.165.201.1 20 msec 36 msec 44 msec
Type escape sequence to abort.
Tracing the route to 209.165.202.158

  1 209.165.201.1 20 msec 32 msec 32 msec
  2 209.165.200.226 44 msec 52 msec 52 msec
```

PRACTICAL NO: 2

Aim : Configure IP SLA Tracking and Path Control Topology.

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.

Step 1: Prepare the routers and configure the router hostname and interface addresses.

Step 2: Verify Server Reachability.

Step 3: Configure IP SLA probes.

a. Create an ICMP echo probe on R1 to the primary DNS server on ISP1 using the ip sla command. the previous ip sla monitor command. In addition, the icmp-echo command has replaced the type echo protocol ipIcmpEcho command.

```
R1 # en
R1 # config t
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)#  ip sla schedule 11 life forever start-time now
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#ip sla schedule 11 life forever start-time now
R1(config)#exit
```

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

```
R1# show ip sla configuration 11
```

```
R1#show ip sla configuration 11
IP SLAs, Infrastructure Engine-II.
Entry number: 11
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.30/0.0.0.0
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Vrf Name:
Request size (ARR data portion): 28
Verify data: No
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
```

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
```

```
R1#show ip sla statistics 11

Round Trip Time (RTT) for       Index 11
        Latest RTT: 56 milliseconds
Latest operation start time: *00:32:30.539 UTC Wed May 15 2024
Latest operation return code: OK
Number of successes: 13
Number of failures: 0
Operation time to live: Forever
```

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 t
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)#  ip sla schedule 22 life forever start-time now
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#ip sla schedule 22 life forever start-time now
```

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

```
R1# show ip sla configuration 22
```

```
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: 209.165.202.158/0.0.0.0
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Vrf Name:
Request size (ARR data portion): 28
Verify data: No
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
```

```
R1# show ip sla statistics 22
```

```
R1#show ip sla statistics 22

Round Trip Time (RTT) for       Index 22
        Latest RTT: 132 milliseconds
Latest operation start time: *00:33:44.519 UTC Wed May 15 2024
Latest operation return code: OK
Number of successes: 5
Number of failures: 0
Operation time to live: Forever
```

Step 4: Configure tracking options.

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

```
R1(config)# config t
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
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
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
```

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial2/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial2/1
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [5/0] via 209.165.201.1
```

c. Use the track 1 ip sla 11 reachability command to enter the config-track subconfiguration mode.

```
R1(config)# config t
R1(config)# track 1 rtr 11 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)#exit
R1(config)# exit
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#track 1 rtr 11 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)#exit
R1(config)#exit
```

d. Configure the floating static route that will be implemented when tracking object 1 is active. To view routing table changes as they happen, first enable the debug ip routing command. Next, 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# debug ip routing
```

```
R1#debug ip routing
IP routing debugging is on
```

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
R1(config)#
*May 15 01:37:35.307: RT: closer admin distance for 0.0.0.0, flushing 1 routes
*May 15 01:37:35.307: RT: NET-RED 0.0.0.0/0
*May 15 01:37:35.307: RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 209.165.201.1

*May 15 01:37:35.307: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]
*May 15 01:37:35.307: RT: NET-RED 0.0.0.0/0
*May 15 01:37:35.307: RT: default path is now 0.0.0.0 via 209.165.201.1
*May 15 01:37:35.307: RT: new default network 0.0.0.0
*May 15 01:37:35.307: RT: NET-RED 0.0.0.0/0
```

e. 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. track 2 ip sla 22 reachability delay down 10 up 1 exit

```
R1(config)#track 2 rtr 22 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)#exit
R1(config)# exit
R1# debug ip routing
R1# config t
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
```

```
R1(config)#track 2 rtr 22 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)#exit
R1(config)#exit
R1#debug ip routing
*May 15 01:38:01.807: %SYS-5-CONFIG_I: Configured from console by console
R1#debug ip routing
*May 15 01:38:05.015: RT: NET-RED 0.0.0.0/0
R1#debug ip routing
IP routing debugging is on
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
```

f. Verify the routing table again

```
R1# show ip route | begin Gateway
```

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial2/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial2/1
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

Step 5: Verify IP SLA operation.

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.

```
ISP1#config t
ISP1(config)# interface loopback 1
ISP1(config-if)# shutdown
```

```
ISP1#config t
Enter configuration commands, one per line.  End with C
ISP1(config)#int Lo1
ISP1(config-if)#shutdown
```

Verify the routing table.

```
R1# show ip route
```

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.202.129 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial2/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial2/1
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [3/0] via 209.165.202.129
```

Verify the IP SLA statistics.

```
R1# show ip sla statistics
```

```
R1#show ip sla statistics

Round Trip Time (RTT) for       Index 11
        Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *02:32:10.539 UTC Wed May 15 2024
Latest operation return code: Timeout
Number of successes: 0
Number of failures: 11
Operation time to live: Forever



Round Trip Time (RTT) for       Index 22
        Latest RTT: 60 milliseconds
Latest operation start time: *02:32:04.519 UTC Wed May 15 2024
Latest operation return code: OK
Number of successes: 354
Number of failures: 0
Operation time to live: Forever
```

Initiate a trace to the web server from the internal LAN IP address.

R1# trace 209.165.200.254 source 192.168.1.1

```
R1#trace 209.165.200.254 source 192.168.1.1

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.202.129 48 msec 52 msec 20 msec
```

Again examine the IP SLA statistics.

ISP1#no shutdown

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

R1# show ip sla statistics

```
R1#show ip sla statistics

Round Trip Time (RTT) for       Index 11
        Latest RTT: 60 milliseconds
Latest operation start time: *02:38:20.539 UTC Wed May 15 2024
Latest operation return code: OK
Number of successes: 8
Number of failures: 40
Operation time to live: Forever



Round Trip Time (RTT) for       Index 22
        Latest RTT: 88 milliseconds
Latest operation start time: *02:38:24.519 UTC Wed May 15 2024
Latest operation return code: OK
Number of successes: 32
Number of failures: 0
Operation time to live: Forever
```

Verify the routing table.

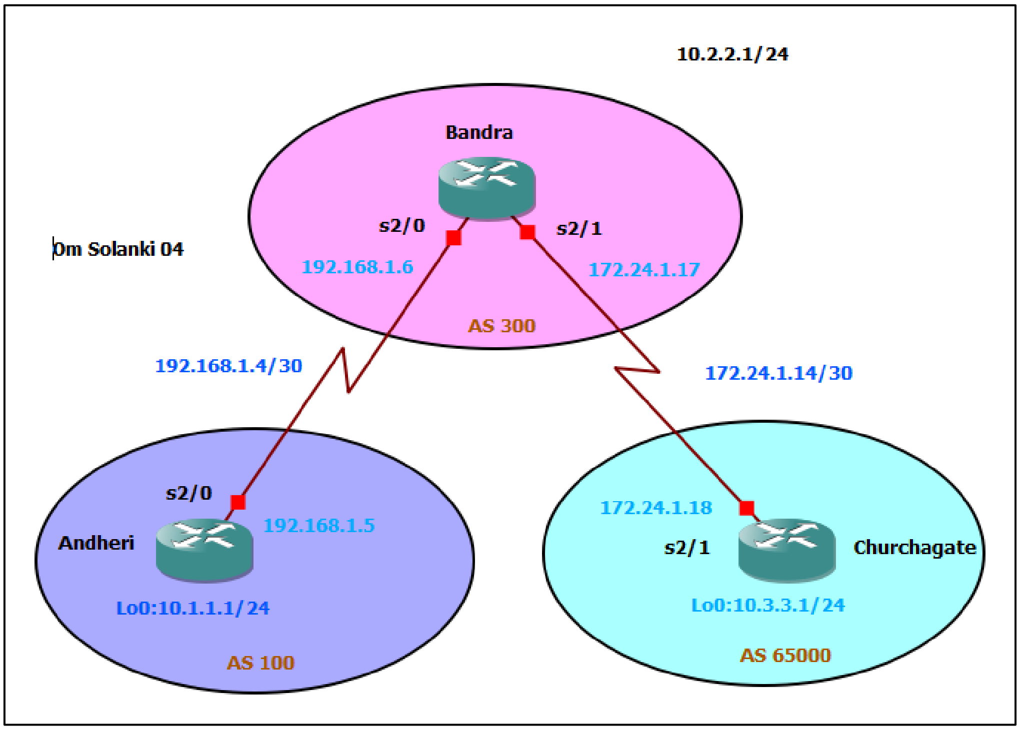
R1# show ip route | begin Gateway

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial2/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial2/1
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

PRACTICAL NO: 3

Aim: Configure using the AS_PATH Attribute

Topology:



Objectives:

→ Use BGP commands to prevent private AS numbers from being advertised to the outside world.

→ Use the AS_PATH attribute to filter BGP routes based on their source AS number.

Step 1 : Prepare the routers for the lab.

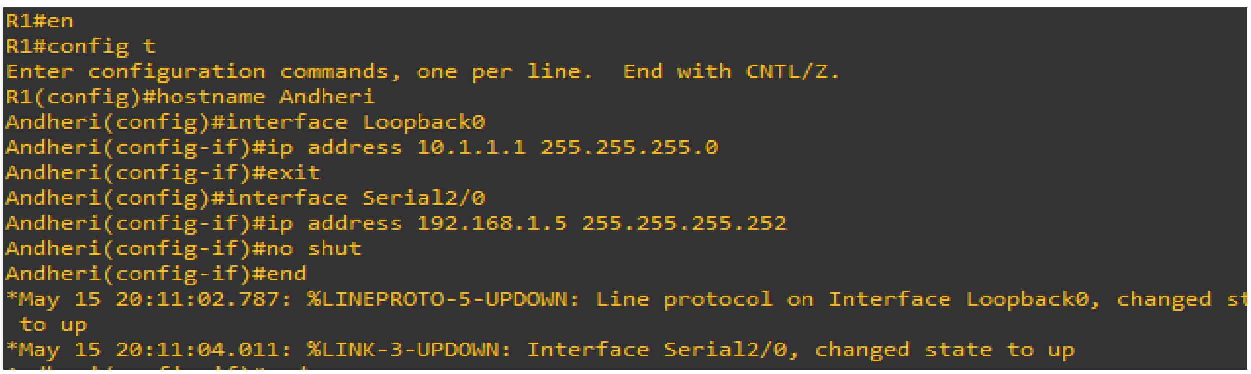
Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations.

Step 2 : Configure the hostname and interface addresses.

a. Configure Routers

Router R1 (hostname Andheri)

Andheri(config)#interface Loopback0
Andheri(config-if)#ip address 10.1.1.1 255.255.255.0
Andheri(config-if)#exit
Andheri(config)#interface Serial0/0
Andheri(config-if)#ip address 192.168.1.5 255.255.255.252
Andheri(config-if)#no shutdown
Andheri(config-if)#end



```
R1#en
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname Andheri
Andheri(config)#interface Loopback0
Andheri(config-if)#ip address 10.1.1.1 255.255.255.0
Andheri(config-if)#exit
Andheri(config)#interface Serial2/0
Andheri(config-if)#ip address 192.168.1.5 255.255.255.252
Andheri(config-if)#no shut
Andheri(config-if)#end
*May 15 20:11:02.787: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed st
 to up
*May 15 20:11:04.011: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
```

Router R2 (hostname Bandra)

Bandra(config)#interface Loopback0
Bandra(config-if)#ip address 10.2.2.1 255.255.255.0
Bandra(config-if)#interface Serial0/0
Router R2 (hostname Bandra)
Bandra(config-if)#ip address 192.168.1.6
Bandra(config-if)#no shutdown
Bandra(config-if)#exit
Bandra(config)#interface Serial0/1
255.255.255.252
Bandra(config-if)#ip address 172.24.1.17 255.255.255.252
Bandra(config-if)#no shutdown
Bandra(config-if)#end

```
R2#en
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname Bandra
Bandra(config)#interface Loopback0
Bandra(config-if)#ip address 10.2.2.1 255.255.255.0
Bandra(config-if)#exit
Bandra(config)#interface Serial2/0
Bandra(config-if)#ip address 192.168.1.6 255.255.255.252
Bandra(config-if)#no shut
Bandra(config-if)#interface Serial2/1
Bandra(config-if)#ip address 172.24.1.17 255.255.255.252
Bandra(config-if)#no shut
Bandra(config-if)#end
Bandra#
*May 15 20:19:07.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed st
 to up
*May 15 20:19:07.527: %SYS-5-CONFIG_I: Configured from console by console
*May 15 20:19:08.519: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
```

Router R3 (hostname Churchgate)

Churchgate(config)#interface Loopback0

Churchgate(config-if)#ip address 10.3.3.1 255.255.255.0

Churchgate(config-if)#exit

Churchgate(config)#interface Serial0/1

Churchgate(config-if)#ip address 172.24.1.18 255.255.255.252

Churchgate(config-if)#no shutdown

Churchgate(config-if)#end

```
R3#en
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname Churchgate
Churchgate(config)#interface Loopback0
Churchgate(config-if)#ip address 10.3.3.1 255.255.255.0
Churchgate(config-if)#exit
Churchgate(config)#interface Serial2/1
Churchgate(config-if)#ip address 172.24.1.18 255.255.255.252
Churchgate(config-if)#no shut
Churchgate(config-if)#end
Churchgate#
*May 15 20:32:52.815: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed st
 to up
*May 15 20:32:53.975: %SYS-5-CONFIG_I: Configured from console by console
*May 15 20:32:54.143: %LINK-3-UPDOWN: Interface Serial2/1, changed state to up
Churchgate#
*May 15 20:32:54.143: %ENTITY_ALARM-6-INFO: CLEAR INFO Se2/1 Physical Port Administrative S
e Down
Churchgate#
*May 15 20:32:55.147: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/1, changed st
 to up
```

Andheri

```
Andheri#show interfaces description | include up
Se2/0                          up             up
Lo0                            up             up
```

Bandra

```
Bandra#show interfaces description | include up
Se2/0                          up             up
Se2/1                          up             up
Lo0                            up             up
```

Churchgate

```
Churchgate#show interfaces description | include up
Se2/1                          up             up
Lo0                            up             up
```

b. Use ping to test the connectivity between the directly connected routers

```
Bandra#ping 192.168.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/172/504 ms
Bandra#ping 172.24.1.18

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/126/412 ms
```

Step 3 : Configure BGP.

a. Configure BGP for normal operation. Enter the appropriate BGP commands on each router so that they identify their BGP neighbors and advertise their loopback networks.

Router R1 (hostname Andheri)

```
Andheri(config)#router bgp 100
Andheri(config-router)#neighbor 192.168.1.6 remote-as 300
Andheri(config-router)#network 10.1.1.0 mask 255.255.255.0
```

```
Andheri#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Andheri(config)#router bgp 100
Andheri(config-router)#neighbor 192.168.1.6 remote-as 300
Andheri(config-router)#network 10.1.1.0 mask 255.255.255.0
```

Router R2 (hostname Bandra)

```
Bandra(config)#router bgp 300
Bandra(config-router)#neighbor 192.168.1.5 remote-as 100
Bandra(config-router)#neighbor 172.24.1.18 remote-as 65000
Bandra(config-router)#network 10.2.2.0 mask 255.255.255.0
```

```
Bandra#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Bandra(config)#router bgp 300
Bandra(config-router)#neighbor 192.168.1.5 remote-as 100
Bandra(config-router)#neighbor 172.24.1.18 remote-as 65000
Bandra(config-router)#network 10.2.2.0 mask 255.255.255.0
```

Router R3 (hostname Churchgate)

```
Churchgate(config)#router bgp 65000

Churchgate(config-router)#neighbor 172.24.1.17 remote-as 300

Churchgate(config-router)#network 10.3.3.0 mask 255.255.255.0
```

```
Churchgate#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Churchgate(config)#router bgp 65000
Churchgate(config-router)#neighbor 172.24.1.17 remote-as 300
Churchgate(config-router)#network 10.3.3.0 mask 255.255.255.0
```

b. Verify that these routers have established the appropriate neighbor relationships by issuing the show ip bgrp neighbors command on each router.

Router R2 (hostname Bandra)

```
Bandra# show ip bgp neighbors
```

```
Andheri#show ip bgp neighbors
BGP neighbor is 192.168.1.6,  remote AS 300, external link
  BGP version 4, remote router ID 10.2.2.1
  BGP state = Established, up for 00:05:13
  Last read 00:00:13, last write 00:00:13, hold time is 180, keepalive interval is 60 secor
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          3
    Keepalives:             8          8
    Route Refresh:          0          0
    Total:                 10         12
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
 Output queue size : 0
```

```
Bandra#show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 65000, external link
  BGP version 4, remote router ID 10.3.3.1
  BGP state = Established, up for 00:04:51
  Last read 00:00:51, last write 00:00:51, hold time is 180, keepalive interval is 60 seco
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                3          1
    Keepalives:             7          7
    Route Refresh:          0          0
    Total:                 11          9
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
 Output queue size : 0
```

Step 4 : Remove the private AS.

a. show the Andheri routing table using the show ip route command. Andheri should have a route to both 10.2.2.0 and 10.3.3.0. Troubleshoot if necessary.

Router R1 (hostname Andheri)

Andheri#show ip route

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 192.168.1.6, 00:06:51
B       10.2.2.0 [20/0] via 192.168.1.6, 00:09:12
C       10.1.1.0 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial2/0
```

b. Ping again, this time as an extended ping, sourcing from the Loopback0 interface address.

Router R1 (hostname Andheri)

ping 10.3.3.1 source 10.1.1.1 or ping 10.3.3.1 source Lo0

```
Andheri#ping 10.3.3.1 source 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/124/232 ms
Andheri#ping 10.3.3.1 source Lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/84/108 ms
```

c. Now check the BGP table on Andheri. The AS_ PATH to the 10.3.3.0 network should be AS 300. It no longer has the private AS in the path.

Router R1 (hostname Andheri)

Andheri# show ip bgp

```
Andheri#show ip bgp
BGP table version is 4, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*> 10.2.2.0/24      192.168.1.6              0             0 300 i
*> 10.3.3.0/24      192.168.1.6                            0 300 65000 i
```

<u>Step 5 : Use the AS_PATH attribute to filter routes.</u>

a. Configure a special kind of access list to match BGP routes with an AS_PATH attribute that both begins and ends with the number 100. Enter the following commands on Bandra.

Router R2 (hostname Bandra)

Bandra(config)# ip as-path access-list 1 deny ^100$

Bandra(config)# ip as-path access-list 1 permit .*

```
Bandra#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Bandra(config)#ip as-path access-list 1 deny ^100$
Bandra(config)#ip as-path access-list 1 permit .*
```

b. Apply the configured access list using the neighbor command with the filter-list option.

Router R2 (hostname Bandra)

Bandra(config)# router bgp 300
Bandra (config-router)# neighbor 192.168.1.5 remove-private-as

```
Bandra(config)#router bgp 300
Bandra(config-router)#neighbor 192.168.1.5 remove-private-as
Bandra(config-router)#exit
```

c. Use the clear ip bgp * command to reset the routing information. Wait several seconds

and then check the routing table for BANDRA. The route to 10.1.1.0 should be in the routing table.

Router R1 (hostname Andheri)

Andheri# show ip route

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 192.168.1.6, 00:00:49
B       10.2.2.0 [20/0] via 192.168.1.6, 00:21:16
C       10.1.1.0 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial2/0
```

d. Return to BANDRA and verify that the filter is working as intended.

Router R2 (hostname Bandra)

Bandra#show ip bgp regexp ^100$

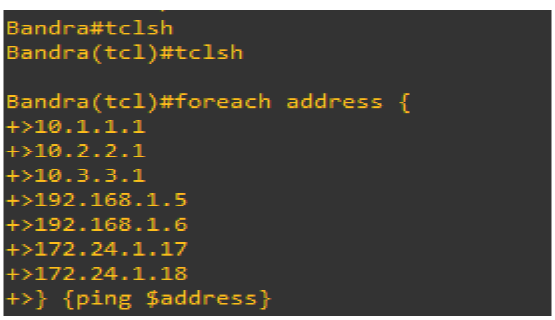
```
Bandra#show ip bgp regexp ^100$
BGP table version is 4, local router ID is 10.2.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.5              0             0 100 i
```

e. Run the following Tcl script on all routers to verify whether there is connectivity.

All pings from BANDRA should be successful. Andheri should not be able to ping the Churchgate loopback 10.3.3.1 or the WAN link 172.24.1.16/30. Churchgate should not be able to ping the Andheri loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

Router R2 (hostname Bandra)

```
Bandra#tclsh
Bandra(tcl)#foreach address {
+>10.1.1.1
+>10.2.2.1
+>10.3.3.1
+>192.168.1.5
+>192.168.1.6
+>172.24.1.17
+>172.24.1.18
+>} {ping $address}
```

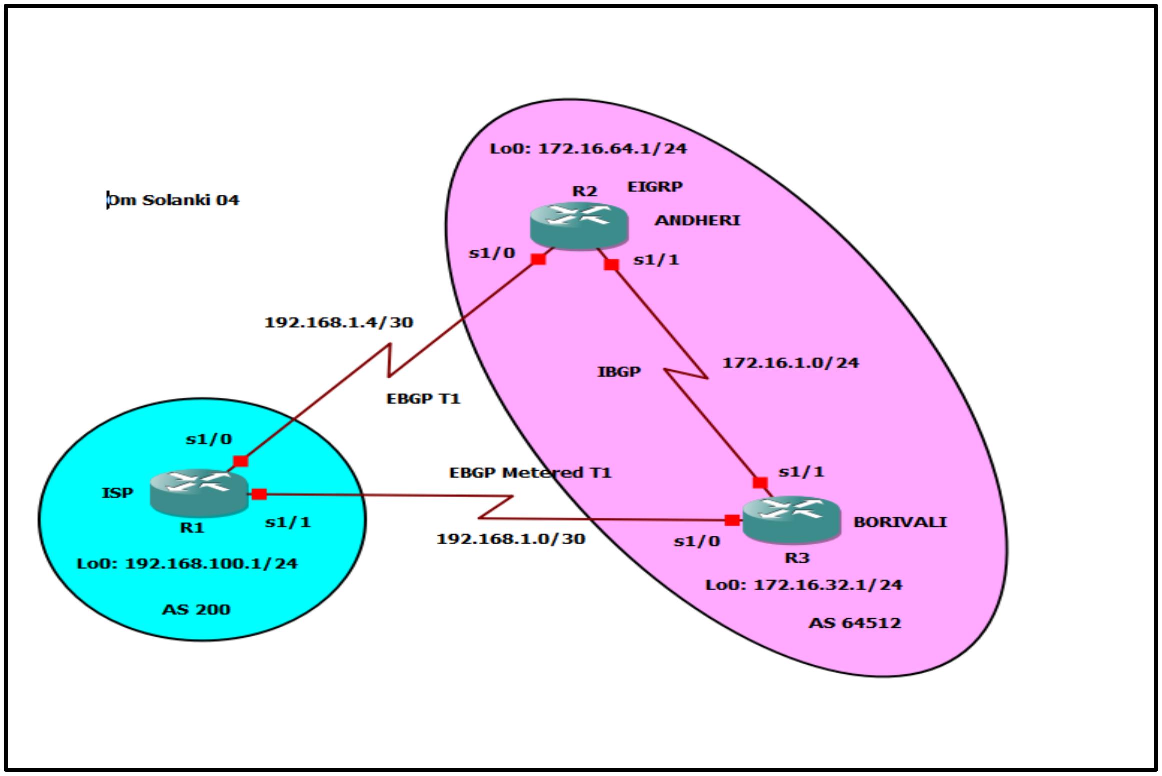


```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/70/116 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/38/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/32/48 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/76/104 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/70/80 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/35/48 ms
```

PRACTICAL NO: 4

Aim : Configuring IBGP and EBGP Sessions, Local Preference, and MED.

Topology:



Objectives:

→ For IBGP peers to correctly exchange routing information, use the next-hop-self command with the Local-Preference and MED attributes.

→ Ensure that the flat-rate, unlimited-use T1 link is used for sending and receiving data to and from the AS 200 on ISP and that the metered T1 only be used in the event that the primary T1 link has failed.

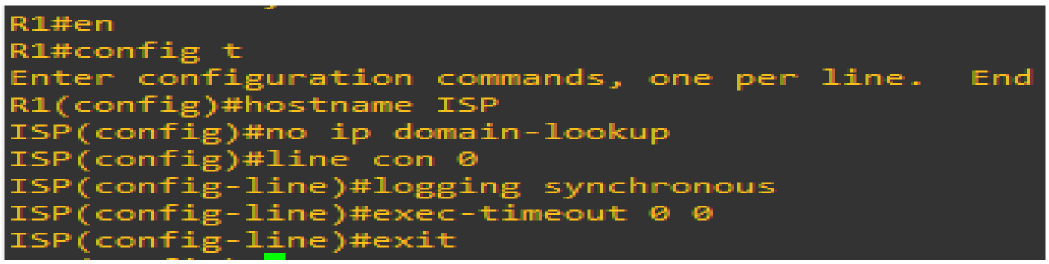
Step 1 : Suggested starting configurations.

Apply the following configuration to each router along with the appropriate hostname. The exec-timeout 0 0 command should only be used in a lab environment.

Router R1:

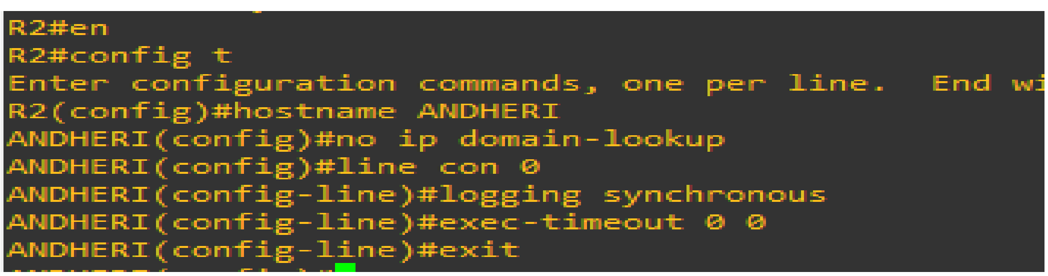
```
R1# en
R1# config t
R1(config)# hostname ISP
```

```
ISP(config)# no ip domain-lookup
ISP(config)# line con 0
ISP(config-line)# logging synchronous
ISP(config-line)# exec-timeout 0 0
ISP(config-line)# exit
```



Router R2:

```
R2# en
R2# config t
R2(config)# hostname Andheri
Andheri(config)# no ip domain-lookup
Andheri(config)# line con 0
Andheri(config-line)# logging synchronous
Andheri(config-line)# exec-timeout 0 0
Andheri(config-line)# exit
```



Router R3:

```
R3# en
R3# config t
R3(config)# hostname Borivali
Borivali(config)# no ip domain-lookup
Borivali(config)# line con 0
Borivali(config-line)# logging synchronous
Borivali(config-line)# exec-timeout 0 0
Borivali(config-line)# exit
```

```
R3#en
R3#config t
Enter configuration commands, one per line.  End wit
R3(config)#hostname BORIVALI
BORIVALI(config)#no ip domain-lookup
BORIVALI(config)#line con 0
BORIVALI(config-line)#logging synchronous
BORIVALI(config-line)#exec-timeout 0 0
BORIVALI(config-line)#exit
```

Step 2 : Configure interface addresses.

a. Using the addressing scheme in the diagram, create the loopback interfaces and apply IPv4 addresses to these and the serial interfaces on ISP (R1), Andheri (R2), and Borivali (R3).

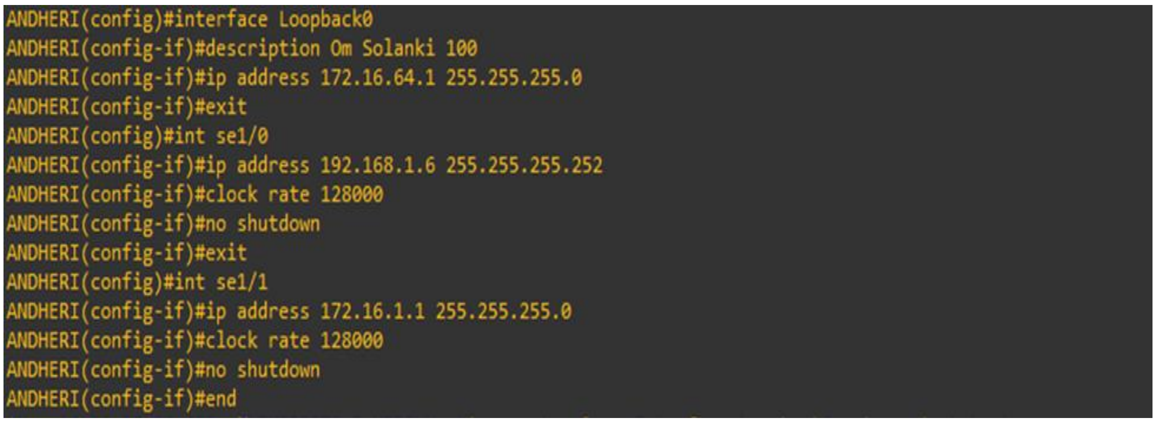
Router R1 (hostname ISP):

```
ISP(config)# interface Loopback0
ISP(config-if)# description OM SOLANKI 100
ISP(config-if)# ip address 192.168.100.1 255.255.255.0
ISP(config-if)# exit
ISP(config)# interface se1/0
ISP(config-if)# ip address 192.168.1.5 255.255.255.252
ISP(config-if)# clock rate 128000
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface se1/1
ISP(config-if)# ip address 192.168.1.1 255.255.255.252
ISP(config-if)# no shutdown
ISP(config-if)# end
```

```
ISP(config)#interface Loopback0
ISP(config-if)#description Om Solanki 100
ISP(config-if)#ip add 192.168.100.1 255.255.255.0
ISP(config-if)#exit
ISP(config)#int se1/0
ISP(config-if)#ip address 192.168.1.5 255.255.255.252
ISP(config-if)#clock rate 128000
ISP(config-if)#no shutdown
ISP(config-if)#exit
ISP(config)#int se1/1
ISP(config-if)#ip address 192.168.1.1 255.255.255.252
ISP(config-if)#clock rate 128000
ISP(config-if)#no shutdown
ISP(config-if)#end
ISP#
*May 16 08:34:06.195: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state
```

Router R2 (hostname Andheri):

```
Andheri(config)# interface Loopback0
Andheri(config-if)# ip address 172.16.64.1 255.255.255.0
Andheri(config-if)# exit
Andheri(config)# interface se1/0
Andheri(config-if)# ip address 192.168.1.6 255.255.255.252
Andheri(config-if)# no shutdown
Andheri(config-if)# exit
Andheri(config)# interface se1/1
Andheri(config-if)# ip address 172.16.1.1 255.255.255.0
Andheri(config-if)# clock rate 128000
Andheri(config-if)# no shutdown
Andheri(config-if)# end
```



```
ANDHERI(config)#interface Loopback0
ANDHERI(config-if)#description Om Solanki 100
ANDHERI(config-if)#ip address 172.16.64.1 255.255.255.0
ANDHERI(config-if)#exit
ANDHERI(config)#int se1/0
ANDHERI(config-if)#ip address 192.168.1.6 255.255.255.252
ANDHERI(config-if)#clock rate 128000
ANDHERI(config-if)#no shutdown
ANDHERI(config-if)#exit
ANDHERI(config)#int se1/1
ANDHERI(config-if)#ip address 172.16.1.1 255.255.255.0
ANDHERI(config-if)#clock rate 128000
ANDHERI(config-if)#no shutdown
ANDHERI(config-if)#end
```

Router R3 (hostname Borivali):

```
Borivali(config)# interface Loopback0
Borivali (config-if)# ip address 172.16.32.1 255.255.255.0
Borivali (config-if)# exit
Borivali (config)# interface se1/0
Borivali (config-if)# ip address 192.168.1.2 255.255.255.252
Borivali (config-if)# clock rate 128000
Borivali (config-if)# no shutdown
Borivali (config-if)# exit
Borivali (config)# interface se1/1
Borivali (config-if)# ip address 172.16.1.2 255.255.255.0
Borivali (config-if)# no shutdown
Borivali (config-if)# end
```

```
BORIVALI(config)#interface Loopback0
BORIVALI(config-if)#description Om Solanki 100
BORIVALI(config-if)#ip address 172.16.32.1 255.255.255.0
BORIVALI(config-if)#exit
*May 16 08:38:32.711: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
BORIVALI(config-if)#exit
BORIVALI(config)#int se1/0
BORIVALI(config-if)#no ip add
BORIVALI(config-if)#ip address 192.168.1.2 255.255.255.252
BORIVALI(config-if)#clock rate 128000
BORIVALI(config-if)#no shutdown
BORIVALI(config-if)#exit
BORIVALI(config)#
BORIVALI(config)#
*May 16 08:38:54.367: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
BORIVALI(config)#
*May 16 08:38:54.367: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/0 Physical Port Administrative State Down
*May 16 08:38:55.367: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
BORIVALI(config)#int se1/1
BORIVALI(config-if)#ip address 172.16.1.2 255.255.255.0
BORIVALI(config-if)#clock rate 128000
BORIVALI(config-if)#no shutdown
BORIVALI(config-if)#end
```

b. Use ping to test the connectivity between the directly connected routers. Both Andheri and Borivali routers should be able to ping each other and their local ISP serial link IP address. The ISP router cannot reach the segment between Andheri and Borivali

ISP# ping 172.16.1.1

```
ISP#ping 172.16.1.1

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

Andheri# ping 172.16.1.2

```
ANDHERI#ping 172.16.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/42/60 ms
```

Andheri# ping 192.168.1.1

```
ANDHERI#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Step 3 : Configure EIGRP.

Configure EIGRP between the Andheri and Borivali routers. (Note: If using an IOS prior to 15.0, use the no auto-summary router configuration command to disable automatic summarization. This command is the default beginning with IOS 15.)

```
config t
router eigrp 1
network 172.16.0.0
```

```
ANDHERI#config t
Enter configuration commands, one per line.  End with CN
ANDHERI(config)#router eigrp 1
ANDHERI(config-router)#network 172.16.0.0
```

```
config t
router eigrp 1
network 172.16.0.0
```

```
BORIVALI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BORIVALI(config)#router eigrp 1
BORIVALI(config-router)#network 172.16.0.0
BORIVALI(config-router)#
*May 18 10:31:39.791: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.1.1 (Serial1/1) is up:
 new adjacency
```

Step 4 : Configure IBGP and verify BGP neighbors.

a. Configure IBGP between the Andheri and Borivali routers. On the Andheri router, enter the following configuration.

```
exit
router bgp 64512
neighbor 172.16.32.1 remote-as 64512
neighbor 172.16.32.1 update-source lo0
```

```
ANDHERI(config-router)#exit
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 172.16.32.1 remote-as 64512
ANDHERI(config-router)#neighbor 172.16.32.1 update-source lo0
```

If multiple pathways to the BGP neighbor exist, the router can use multiple IP interfaces to communicate with the neighbor. The source IP address therefore depends on the outgoing interface. The update-source lo0 command instructs the router to use the IP address of the interface Loopback0 as the source IP address for all BGP messages sent to that neighbor.

b. Complete the IBGP configuration on Borivali using the following command exit.

```
router bgp 64512
neighbor 172.16.64.1 remote-as 64512
neighbor 172.16.64.1 update-source lo0
end
```

```
BORIVALI(config-router)#exit
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#neighbor 172.16.64.1 remote-as 64512
BORIVALI(config-router)#neighbor 172.16.64.1 update-source lo0
BORIVALI(config-router)#end
BORIVALI#
*May 18 10:37:10.459: %SYS-5-CONFIG_I: Configured from console by console
```

c. Verify that SanJose1 and SanJose2 become BGP neighbors by issuing the show ip bgp neighbors command on SanJose1. View the following partial output. If the BGP state is not established, troubleshoot the connection.

Borivali# show ip bgp neighbors

```
BORIVALI#show ip bgp neighbors
BGP neighbor is 172.16.64.1,  remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.64.1
  BGP state = Established, up for 00:01:48
  Last read 00:00:47, last write 00:00:47, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
```

Step 5 : Configure EBGP and verify BGP neighbors.

a. Configure ISP to run EBGP with Andheri and Borivali. Enter the following commands on ISP.

```
ISP(config)# router bgp 200
ISP(config-router)# neighbor 192.168.1.6 remote-as 64512
ISP(config-router)# neighbor 192.168.1.2 remote-as 64512
ISP(config-router)# network 192.168.100.0
```

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#neighbor 192.168.1.6 remote-as 64512
ISP(config-router)#neighbor 192.168.1.2 remote-as 64512
ISP(config-router)#network 192.168.100.0
```

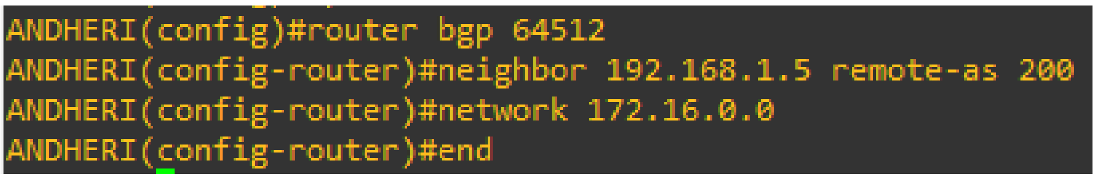
b. Configure a discard static route for the 172.16.0.0/16 network. Any packets that do not have a more specific match (longer match) for a 172.16.0.0 subnet will be dropped instead of sent to the ISP. Later in this lab we will configure a default route to the ISP.

Andheri(config-router)# exit
Andheri(config)# ip route 172.16.0.0 255.255.0.0 null0

```
ANDHERI(config-router)#exit
ANDHERI(config)#ip route 172.16.0.0 255.255.0.0 null0
```

c. Configure SanJose1 as an EBGP peer to ISP.

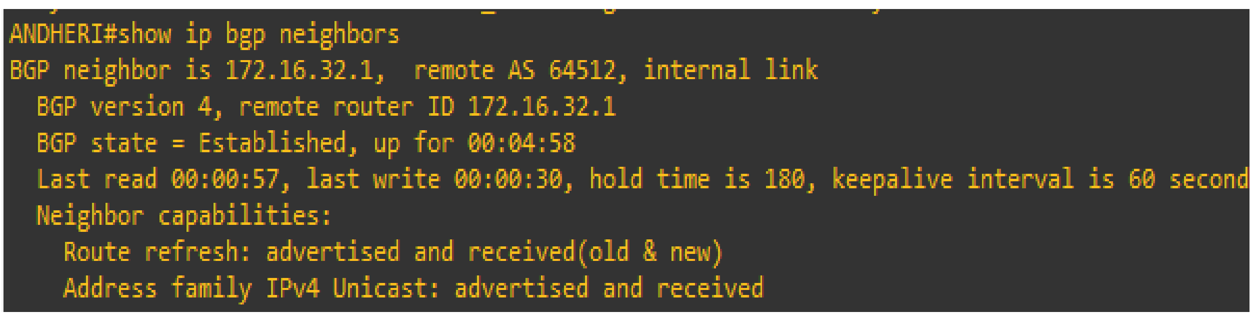
Andheri(config)# router bgp 64512
Andheri(config-router)# neighbor 192.168.1.5 remote-as 200
Andheri(config-router)# network 172.16.0.0



```
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 192.168.1.5 remote-as 200
ANDHERI(config-router)#network 172.16.0.0
ANDHERI(config-router)#end
```

d. Use the show ip bgp neighbors command to verify that Andheri and ISP have reached the established state. Troubleshoot if necessary.

Andheri# show ip bgp neighbors



```
ANDHERI#show ip bgp neighbors
BGP neighbor is 172.16.32.1,  remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.32.1
  BGP state = Established, up for 00:04:58
  Last read 00:00:57, last write 00:00:30, hold time is 180, keepalive interval is 60 second
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
```

e. Configure a discard static route for 172.16.0.0/16 on Borivali and as an EBGP peer to ISP.

Borivali# config t
Borivali(config)# ip route 172.16.0.0 255.255.0.0 null0
Borivali(config)# router bgp 64512
Borivali(config-router)# neighbor 192.168.1.1 remote-as 200
Borivali(config-router)# network 172.16.0.0

```
BORIVALI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BORIVALI(config)#ip route 172.16.0.0 255.255.0.0 null0
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#neighbor 192.168.1.1 remote-as 200
BORIVALI(config-router)#network 172.16.0.0
```

Step 6 : View BGP summary output.

In Step 4, the show ip bgp neighbors command was used to verify that SanJose1 and ISP had reached the established state. A useful alternative command is show ip bgp summary. The output should be similar to the following.

```
Borivali# show ip bgp summary
```

```
BORIVALI(config-router)#end
BORIVALI#show ip bgp neighbors
*May 18 10:46:18.907: %SYS-5-CONFIG_I: Configured from console by console
BORIVALI#show ip bgp neighbors
BGP neighbor is 172.16.64.1,  remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.64.1
  BGP state = Established, up for 00:08:47
  Last read 00:00:46, last write 00:00:30, hold time is 180, keepalive interval is 60 second
s
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
```

Step 7 : Verify which path the traffic takes.

a. Clear the IP BGP conversation with the clear ip bgp * command on ISP. Wait for the conversations to reestablish with both Andheri and Borivali router.

```
ISP(config-router)# end
ISP# clear ip bgp *
```

```
ISP(config-router)#end
ISP#clear ip bgp *
*May 18 10:47:47.791: %SYS-5-CONFIG_I: Configured from console by console
ISP#clear ip bgp *
ISP#
*May 18 10:47:49.967: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Down User reset
*May 18 10:47:49.967: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Down User reset
ISP#
*May 18 10:47:51.095: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Up
*May 18 10:47:51.127: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Up
```

b. Test whether ISP can ping the loopback 0 address of 172.16.64.1 on Andheri and the serial link between Andheri and Borivali, 172.16.1.1.

```
ISP# ping 172.16.64.1
ISP# ping 172.16.1.1
```

```
ISP#ping 172.16.64.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
ISP#ping 172.16.1.1

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

c. Now ping from ISP to the loopback 0 address of 172.16.32.1 on Borivali and the serial link between Andheri and Borivali, 172.16.1.2.

```
ISP# ping 172.16.32.1
ISP# ping 172.16.1.2
```

```
ISP#ping 172.16.32.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/45/60 ms
ISP#ping 172.16.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/49/60 ms
```

d. Issue the show ip bgp command on ISP to verify BGP routes and metrics.

```
ISP# show ip bgp
```

```
ISP#show ip bgp
BGP table version is 3, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.6              0             0 64512 i
*>                  192.168.1.2              0             0 64512 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
```

BGP operates differently than all other protocols. Unlike other routing protocols that use complex algorithms involving factors such as bandwidth, delay, reliability, and load to formulate a metric, BGP is policy-based. BGP determines the best path based on variables, such as AS path, weight, local preference, MED, and so on. If all things are equal, BGP prefers the route leading to the BGP speaker with the lowest BGP router ID. The Borivali router with BGP router ID 172.16.32.1 was preferred to the higher BGP router ID of the Andheri router (172.16.64.1).

e. At this point, the ISP router should be able to get to each network connected to Andheri and Borivali from the loopback address 192.168.100.1. Use the extended ping command and specify the source address of ISP Lo0 to test.

```
ISP# ping 172.16.1.1 source 192.168.100.1
ISP# ping 172.16.32.1 source 192.168.100.1
ISP# ping 172.16.1.2 source 192.168.100.1
ISP# ping 172.16.64.1 source 192.168.100.1
ISP# ping
```

```
ISP#ping 172.16.1.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/80/96 ms
ISP#ping 172.16.32.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/55/64 ms
ISP#ping 172.16.1.2 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/54/60 ms
ISP#ping 172.16.64.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/76/96 ms
```

```
ISP#ping
Protocol [ip]:
Target IP address: 172.16.64.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.100.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/84/96 ms
```

Step 8 : Configure the BGP next-hop-self feature.

Andheri is unaware of the link between ISP and Borivali, and Borivali is unaware of the link between ISP and Andheri. Before ISP can successfully ping all the internal serial interfaces of AS 64512, these serial links should be advertised via BGP on the ISP router. This can also be resolved via EIGRP on both the Andheri and Borivali router. One method is for ISP to advertise these links.

a. Issue the following commands on the ISP router.

```
ISP# config t
ISP(config)# router bgp 200
ISP(config-router)# network 192.168.1.0 mask 255.255.255.252
ISP(config-router)# network 192.168.1.4 mask 255.255.255.252
```

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#network 192.168.1.0 mask 255.255.255.252
ISP(config-router)#network 192.168.1.4 mask 255.255.255.252
```

b. Issue the show ip bgp command to verify that the ISP is correctly injecting its own WAN links into BGP.

```
ISP#(config-router)# end
ISP# show ip bgp
```

```
ISP(config-router)#end
ISP#show ip bgp
*May 18 11:04:48.399: %SYS-5-CONFIG_I: Configured from console by console
ISP#show ip bgp
BGP table version is 5, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.6              0             0 64512 i
*>                  192.168.1.2              0             0 64512 i
*> 192.168.1.0/30   0.0.0.0                  0         32768 i
*> 192.168.1.4/30   0.0.0.0                  0         32768 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
```

c. Verify on Andheri and Borivali that the opposite WAN link is included in the routing table. The output from Borivali is as follows.

```
Borivali# show ip route
```

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:34:17, Serial1/1
     192.168.1.0/30 is subnetted, 2 subnets
C       192.168.1.0 is directly connected, Serial1/0
B       192.168.1.4 [20/0] via 192.168.1.1, 00:02:55
B    192.168.100.0/24 [20/0] via 192.168.1.1, 00:17:16
```

The next issue to consider is BGP policy routing between autonomous systems. The next-hop attribute of a route in a different AS is set to the IP address of the border router in the next AS toward the destination, and this attribute is not modified by default when advertising this route through IBGP. Therefore, for all IBGP peers, it is either necessary to know the route to that border router (in a different neighboring AS), or our own border router needs to advertise the foreign routes using the next-hop-self feature, overriding the next-hop address with its own IP address. The Borivali router is passing a policy to Andheri and vice versa. The policy for routing from AS 64512 to AS 200 is to forward packets to the 192.168.1.1 interface. Andheri has a similar yet opposite policy: it forwards requests to the 192.168.1.5 interface. If either WAN link fails, it is critical that the opposite router become a valid gateway. This is achieved if the next-hop-self command is configured on Andheri and Borivali.

d. To better understand the next-hop-self command we will remove ISP advertising its two WAN links and shutdown the WAN link between ISP and Borivali. The only possible path from Borivali to ISP's 192.168.100.0/24 is through Andheri.

```
ISP# config t
ISP(config)# router bgp 200
ISP(config-router)# no network 192.168.1.0 mask 255.255.255.252
ISP(config-router)# no network 192.168.1.4 mask 255.255.255.252
ISP(config-router)# exit
ISP(config)# interface serial 1/1
ISP(config-if)# shutdown
```

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#no network 192.168.1.0 mask 255.255.255.252
ISP(config-router)#no network 192.168.1.4 mask 255.255.255.252
ISP(config-router)#exit
ISP(config)#interface serial 1/1
ISP(config-if)#shutdown
ISP(config-if)#
*May 18 11:06:57.967: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Down Interface flap
ISP(config-if)#
*May 18 11:06:59.951: %LINK-5-CHANGED: Interface Serial1/1, changed state to administrativel
y down
ISP(config-if)#
*May 18 11:06:59.951: %ENTITY_ALARM-6-INFO: ASSERT INFO Se1/1 Physical Port Administrative S
tate Down
*May 18 11:07:00.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed sta
te to down
```

e. Display SanJose2's BGP table using the show ip bgp command and the IPv4 routing table with show ip route.

Borivali# show ip bgp
Borivali# show ip route

```
BORIVALI#show ip bgp
BGP table version is 13, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i                 172.16.64.1              0    100      0 i
* i192.168.100.0    192.168.1.5              0    100      0 200 i
BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:36:19, Serial1/1
```

Notice that Borivali has 192.168.100.0 in it's BGP table but not in its routing table. The BGP table shows the next hop to 192.168.100.0 as 192.168.1.5. Because Borivali does not have a route to this next hop address of 192.168.1.5 in its routing table, it will not install the 192.168.100.0 network into the routing table. It won't install a route if it doesn't know how to get to the next hop.

EBGP next hop addresses are carried into IBGP unchanged. As we saw previously, we could advertise the WAN link using BGP, but this is not always desirable. It means advertising additional routes when we are usually trying to minimize the size of the routing table. Another option is to have the routers within the IGP domain advertise themselves as the next hop router using the next-hop-self command.

f. Issue the next-hop-self command on Andheri and Borivali to advertise themselves as the next hop to their IBGP peer

```
Andheri# config t
Andheri(config)# router bgp 64512
Andheri(config-router)# neighbor 172.16.32.1 next-hop-self
```

```
ANDHERI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 172.16.32.1 next-hop-self
```

Perform the same step on Borivali router

g. Reset BGP operation on either router with the clear ip bgp * command.

```
Andheri# clear ip bgp *
Andheri#
```

```
ANDHERI(config-router)#end
ANDHERI#clear ip bgp *
ANDHERI#
*May 18 11:10:40.851: %SYS-5-CONFIG_I: Configured from console by console
ANDHERI#
*May 18 11:10:42.879: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Down User reset
*May 18 11:10:42.879: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down User reset
*May 18 11:10:43.563: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Up
*May 18 11:10:43.803: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
```

Perform the same step for Borivali router

h. After the routers have returned to established BGP speakers, issue the show ip bgp command on Borivali and notice that the next hop is now SanJose1 instead of ISP.

```
Borivali# show ip bgp
```

```
BORIVALI#show ip bgp
BGP table version is 18, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.64.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i192.168.100.0    172.16.64.1              0    100      0 200 i
```

i. The show ip route command on Borivali now displays the 192.168.100.0/24 network because Andheri is the next hop, 172.16.64.1, which is reachable from Borivali.

Borivali# show ip route

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:42:48, Serial1/1
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial1/0
```

j. Before configuring the next BGP attribute, restore the WAN link between ISP and Andheri. This will change the BGP table and routing table on both routers. For example, Borivali's routing table shows 192.168.100.0/24 will now have a better path through ISP.

Borivali# show ip route

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:42:48, Serial1/1
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial1/0
```

Step 9 : Set BGP local preference.

At this point, everything looks good, with the exception of default routes, the outbound flow of data, and inbound packet flow.

a. Because the local preference value is shared between IBGP neighbors, configure a simple route map that references the local preference value on Andheri and Borivali. This policy adjusts outbound traffic to prefer the link off the Andheri router instead of the metered T1 off Borivali.

```
Andheri# config t
Andheri(config)# route-map PRIMARY_T1_IN permit 10
Andheri(config-route-map)# set local-preference 150
Andheri(config-route-map)# exit
Andheri(config)# router bgp 64512
Andheri(config-router)# neighbor 192.168.1.5 route-map PRIMARY_T1_IN in
```

```
ANDHERI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ANDHERI(config)#route-map PRIMARY_T1_IN permit 10
ANDHERI(config-route-map)#set local-preference 150
ANDHERI(config-route-map)#exit
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_IN in
```

```
Borivali# config  t
Borivali(config)# route-map SECONDARY_T1_IN permit 10
Borivali(config-route-map)# set local-preference 125
Borivali(config-route-map)# exit
Borivali(config)# router bgp 64512
Borivali(config-router)# neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
```

```
BORIVALI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BORIVALI(config)#route-map SECONDARY_T1_IN permit 10
BORIVALI(config-route-map)#set local-preference 125
BORIVALI(config-route-map)#exit
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
```

b. Use the clear ip bgp * soft command after configuring this new policy. When the conversations have been reestablished, issue the show ip bgp command on Andheri and Borivali.

```
Andheri(config-router)# end
Andheri# clear ip bgp * soft
```

```
ANDHERI(config-router)#end
ANDHERI#clear ip bgp * soft
*May 18 11:16:55.595: %SYS-5-CONFIG_I: Configured from console by console
ANDHERI#clear ip bgp * soft
```

```
Borivali(config-router)# end
Borivali# clear ip bgp * soft
```

```
BORIVALI(config-router)#end
BORIVALI#clear ip bgp * soft
```

```
Andheri# show ip bgp
```

```
ANDHERI#show ip bgp
BGP table version is 6, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i                 172.16.32.1              0    100      0 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
```

```
Borivali# show ip bgp
```

```
BORIVALI#show ip bgp
BGP table version is 20, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.64.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*  192.168.100.0    192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
```

This now indicates that routing to the loopback segment for ISP 192.168.100.0 /24 can be reached only through the link common to SanJose1 and ISP. SanJose2's next hop to 192.168.100.0/24 is SanJose1 because both routers have been configured using the next-hop-self command.

Step 10 : Set BGP MED.

a. In the previous step we saw that Andheri and Borivali will route traffic for 192.168.100.0/24 using the link between Andheri and ISP. Examine what the return path ISP takes to reach AS 64512. Notice that the return path is different from the original path. This is known as asymmetric routing and is not necessarily an unwanted trait.

```
ISP# (config-if)# end
ISP# show ip bgp
```

```
ISP(config-if)#end
ISP#show ip bgp
BGP table version is 14, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.2              0             0 64512 i
*>                  192.168.1.6              0             0 64512 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
ISP#
```

```
Borivali# show ip bgp
```

```
BORIVALI(config-router)#end
BORIVALI#clear ip bgp * soft
BORIVALI#show ip bgp
*May 18 11:25:02.087: %SYS-5-CONFIG_I: Configured from console by console
BORIVALI#show ip bgp
BGP table version is 20, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.64.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*  192.168.100.0    192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
```

g. Use an extended ping command to verify this situation. Specify the record option and compare your output to the following. Notice the return path using the exit interface 192.168.1.1 to Borivali

```
Borivali# ping
Target IP address: 192.168.100.1
Extended commands [n]: y
Source address or interface: 172.16.32.1
```

```
BORIVALI#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
```

```
Reply to request 3 (104 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 4 (92 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Success rate is 100 percent (5/5), round-trip min/avg/max = 92/98/104 ms
```

c. Create a new policy to force the ISP router to return all traffic via SanJose1. Create a second route map utilizing the MED (metric) that is shared between EBGP neighbors.

```
Andheri# config t
Andheri(config)#route-map PRIMARY_T1_MED_OUT permit 10
Andheri(config-route-map)#set Metric 50
Andheri(config-route-map)#exit
Andheri(config)#router bgp 64512
Andheri(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out
```

```
ANDHERI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ANDHERI(config)#route-map PRIMARY_T1_MED_OUT permit 10
ANDHERI(config-route-map)#set Metric 50
ANDHERI(config-route-map)#exit
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT
```

```
Borivali# config t
Borivali(config)# route-map SECONDARY_T1_MED_OUT permit 10
Borivali(config-route-map)# set Metric 75
Borivali(config-route-map)# exit
Borivali(config)# router bgp 64512
Borivali(config-router)# neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
```

```
BORIVALI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BORIVALI(config)#route-map SECONDARY_T1_MED_OUT permit 10
BORIVALI(config-route-map)#set Metric 75
BORIVALI(config-route-map)#exit
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#$2.168.1.1 route-map SECONDARY_T1_MED_OUT out
```

d. Use the clear ip bgp * soft command after issuing this new policy. Issuing the show ip bgp command as follows on Andheri or Borivali does not indicate anything about this newly defined policy.

```
Andheri(config-router)# end
Andheri# clear ip bgp * soft
Andheri# show ip bgp
```

```
ANDHERI(config-router)#end
ANDHERI#clear ip bgp * soft
ANDHERI#show ip bgp
BGP table version is 6, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i                 172.16.32.1              0    100      0 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
ANDHERI#
*May 18 11:24:19.435: %SYS-5-CONFIG_I: Configured from console by console
```

```
Borivali(config-router)# end
Borivali# clear ip bgp * soft
Borivali# show ip bgp
```

```
BORIVALI(config-router)#end
BORIVALI#clear ip bgp * soft
BORIVALI#show ip bgp
*May 18 11:25:02.087: %SYS-5-CONFIG_I: Configured from console by console
BORIVALI#show ip bgp
BGP table version is 20, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.64.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*  192.168.100.0    192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
```

e. Reissue an extended ping command with the record command. Notice the change in return path using the exit interface 192.168.1.5 to Andheri

```
Borivali# ping
Target IP address: 192.168.100.1
Extended commands [n]: y
Source address or interface: 172.16.32.1
```

```
BORIVALI#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
```

```
Reply to request 3 (104 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 4 (92 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Success rate is 100 percent (5/5), round-trip min/avg/max = 92/98/104 ms
```

The newly configured policy MED shows that the lower MED value is considered best. The ISP now prefers the route with the lower MED value of 50 to AS 64512. This is just opposite from the local-preference command configured earlier.

ISP# show ip bgp

```
ISP#show ip bgp
BGP table version is 16, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.2             75             0 64512 i
*>                  192.168.1.6             50             0 64512 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
```

Step 11 : Establish a default route.

The final step is to establish a default route that uses a policy statement that adjusts to changes in the network.

a. Configure ISP to inject a default route to both Andheri and Borivali using BGP using the default-originate command. This command does not require the presence of 0.0.0.0 in the ISP router. Configure the 10.0.0.0/8 network which will not be advertised using BGP. This network will be used to test the default route on Andheri and Borivali.

```
ISP# config t
ISP(config)# router bgp 200
ISP(config-router)# neighbor 192.168.1.6 default-originate
ISP(config-router)# neighbor 192.168.1.2 default-originate
ISP(config-router)# exit
ISP(config)# interface loopback 10
ISP(config-if)# ip address 10.0.0.1 255.255.255.0
```

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#neighbor 192.168.1.6 default-originate
ISP(config-router)#neighbor 192.168.1.2 default-originate
ISP(config-router)#exit
ISP(config)#interface loopback 10
ISP(config-if)#ip address 10.0.0.1 255.255.255.0
ISP(config-if)#
*May 18 11:28:34.935: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback10, changed st
ate to up
```

b. Verify that both routers have received the default route by examining the routing tables on Andheri and Borivali. Notice that both routers prefer the route between Andheri and ISP.

```
Andheri# show ip route
```

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

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.32.0/24 [90/2297856] via 172.16.1.2, 00:57:28, Serial1/1
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial1/0
B    192.168.100.0/24 [20/0] via 192.168.1.5, 00:17:23
B*   0.0.0.0/0 [20/0] via 192.168.1.5, 00:00:33
```

Borivali# show ip route

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

Gateway of last resort is 172.16.64.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:58:12, Serial1/1
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial1/0
B    192.168.100.0/24 [200/0] via 172.16.64.1, 00:12:55
B*   0.0.0.0/0 [200/0] via 172.16.64.1, 00:01:17
```

c. The preferred default route is by way of SanJose1 because of the higher local preference attribute configured on Andheri earlier. Using the traceroute command verify that packets to 10.0.0.1 is using the default route through Andheri.

Borivali# show ip bgp
Borivali# traceroute 10.0.0.1

```
BORIVALI#show ip bgp
BGP table version is 21, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  0.0.0.0          192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
* i172.16.0.0       172.16.64.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*  192.168.100.0    192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
BORIVALI#traceroute 10.0.0.1

Type escape sequence to abort.
Tracing the route to 10.0.0.1

  1 172.16.1.1 44 msec 56 msec 60 msec
  2 192.168.1.5 [AS 200] 84 msec 88 msec 112 msec
```

d. Next, test how BGP adapts to using a different default route when the path between Andheri and ISP goes down.

ISP(config-if)# exit
ISP(config)# interface Serial1/0
ISP(config-if)# shutdown

```
ISP(config-if)#exit
ISP(config)#interface Serial1/0
ISP(config-if)#shutdown
ISP(config-if)#
*May 18 11:31:36.467: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Down Interface flap
ISP(config-if)#
*May 18 11:31:38.459: %LINK-5-CHANGED: Interface Serial1/0, changed state to administratively down
ISP(config-if)#
*May 18 11:31:38.459: %ENTITY_ALARM-6-INFO: ASSERT INFO Se1/0 Physical Port Administrative State Down
*May 18 11:31:39.459: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to down
```

e. Verify that both routers are modified their routing tables with the default route using the path between Borivali and ISP.

Andheri# show ip route

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.32.0/24 [90/2297856] via 172.16.1.2, 01:00:42, Serial1/1
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
C       172.16.64.0/24 is directly connected, Loopback0
```

Borivali# show ip route

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

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 01:01:18, Serial1/1
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial1/0
B    192.168.100.0/24 [20/0] via 192.168.1.1, 00:00:59
B*   0.0.0.0/0 [20/0] via 192.168.1.1, 00:00:59
```

f. Verify the new path using the traceroute command to 10.0.0.1 from Andheri. Notice the default route is now through Borivali.

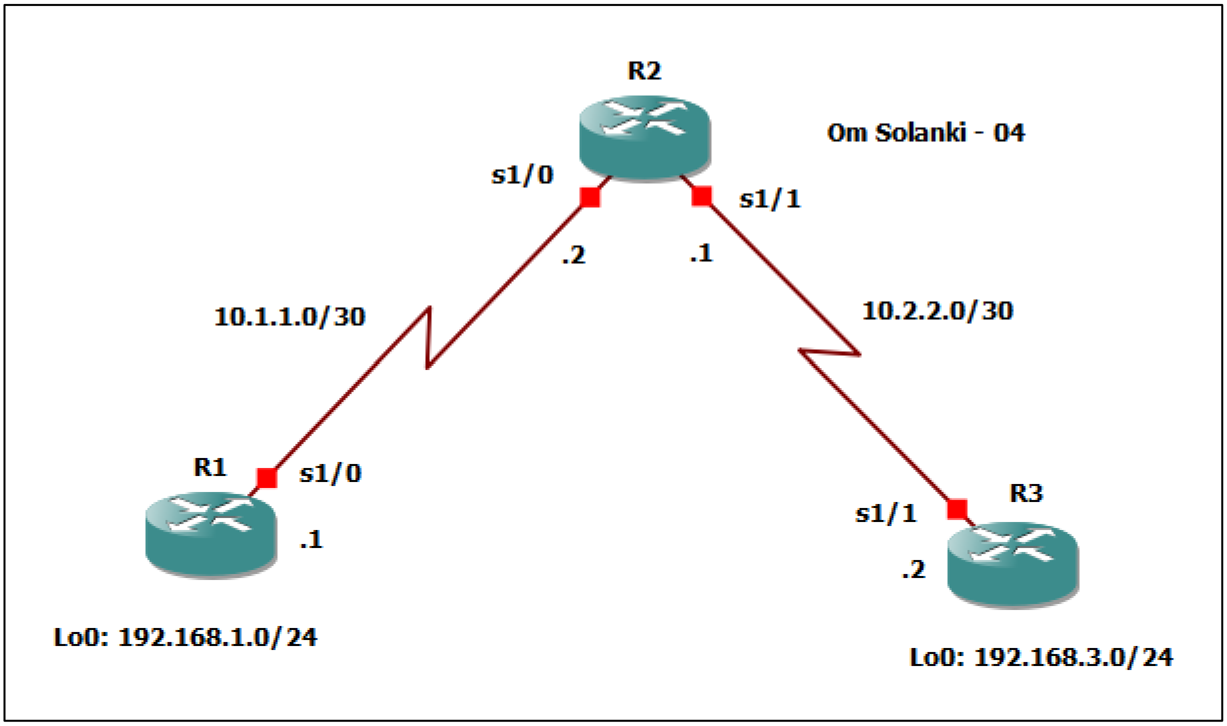
Andheri# trace 10.0.0.1

```
ANDHERI#trace 10.0.0.1

Type escape sequence to abort.
Tracing the route to 10.0.0.1
```

PRACTICAL NO: 5

Aim: Secure the Management Plane.

Topology:



Objectives:

→ Secure management access.

→ Configure enhanced username password security.

→ Enable secure remote management.

Step 1 : Configure loopbacks and assign addresses.

R1:

```
hostname R1
interface Loopback 0
description R1 LAN
ip address 192.168.1.1 255.255.255.0
exit
interface Serial0/0
description R1 --> R2
ip address 10.1.1.1 255.255.255.252
clock rate 128000
```

```
no shutdown
exit
end
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#interface loopback0
R1(config-if)#description R1 LAN
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#interface se1/0
R1(config-if)#description R1-->R2
R1(config-if)#ip add 10.1.1.1 255.255.255.252
R1(config-if)#clock rate 128000
R1(config-if)#no shut
R1(config-if)#end
R1#
*May 16 09:02:54.631: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
*May 16 09:02:55.375: %SYS-5-CONFIG_I: Configured from console by console
*May 16 09:02:55.683: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R1#
*May 16 09:02:55.683: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/0 Physical Port Administrative State Down
R1#
*May 16 09:02:56.687: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
```

R2:

```
hostname R2
interface Serial0/0
description R2 --> R1
ip address 10.1.1.2 255.255.255.252
no shutdown
exit
interface Serial0/1
description R2 --> R3
ip address 10.2.2.1 255.255.255.252
clock rate 128000
no shutdown
exit
end
```

```
R2(config)#hostname R2
R2(config)#interface se1/0
R2(config-if)#description R2-->R1
R2(config-if)#ip add 10.1.1.2 255.255.255.252
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#interface se1/1
R2(config-if)#description R2-->R3
R2(config-if)#ip add 10.2.2.1 255.255.255.252
R2(config-if)#clock rate 128000
R2(config-if)#no shut
R2(config-if)#end
R2#
*May 16 09:03:11.031: %SYS-5-CONFIG_I: Configured from console by console
*May 16 09:03:11.547: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*May 16 09:03:11.547: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/0 Physical Port Administrative State Down
*May 16 09:03:11.563: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R2#
*May 16 09:03:11.567: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/1 Physical Port Administrative State Down
R2#
*May 16 09:03:12.551: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
*May 16 09:03:12.567: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
```

R3:

```
hostname R3
interface Loopback0
description R3 LAN
ip address 192.168.3.1 255.255.255.0
Exit
interface Serial0/1
description R3 --> R2
ip address 10.2.2.2 255.255.255.252
no shutdown
exit
end
```

```
R3(config)#hostname R3
R3(config)#interface loopback0
R3(config-if)#description R3 LAN
R3(config-if)#ip add 192.168.3.1 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#interface se1/1
R3(config-if)#description R3-->R2
R3(config-if)#ip add 10.2.2.2 255.255.255.252
R3(config-if)#clock rate 128000
R3(config-if)#no shut
R3(config-if)#end
R3#
R3#
*May 16 09:03:30.587: %SYS-5-CONFIG_I: Configured from console by console
*May 16 09:03:31.499: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R3#
*May 16 09:03:32.555: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R3#
*May 16 09:03:32.555: %ENTITY_ALARM-6-INFO: CLEAR INFO Se1/1 Physical Port Administrative State Down
R3#
*May 16 09:03:33.559: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
```

Step 2 : Configure static routes.

a. On R1, configure a default static route to ISP.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.2
```

```
R1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2
```

b. On R3, configure a default static route to ISP.

```
R3(config)# ip route 0.0.0.0 0.0.0.0 10.2.2.1
```

```
R3(config)#ip route 0.0.0.0 0.0.0.0 10.2.2.1
```

c. On R2, configure two static routes.

```
R2(config)# ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)# ip route 192.168.3.0 255.255.255.0 10.2.2.2
```

```
R2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.2
R2(config)#exit
```

d. From the R1 router, run the following Tcl script to verify connectivity.

```
foreach address {
192.168.1.1
10.1.1.1
10.1.1.2
10.2.2.1
10.2.2.2
192.168.3.1
} { ping $address }
end
```

```
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#10.1.1.1
+>(tcl)#10.1.1.2
+>(tcl)#10.2.2.1
+>(tcl)#10.2.2.2
+>(tcl)#192.168.3.1
+>(tcl)#} {ping $address}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/110/120 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/51/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/56/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/108/124 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/103/124 ms
R1(tcl)#
```

Step 3 : Secure management access.

a. On R1, use the security passwords command to set a minimum password length of 10 characters.

```
R1(config)# security passwords min-length 10
```

```
R1(config)#security password min-length 10
```

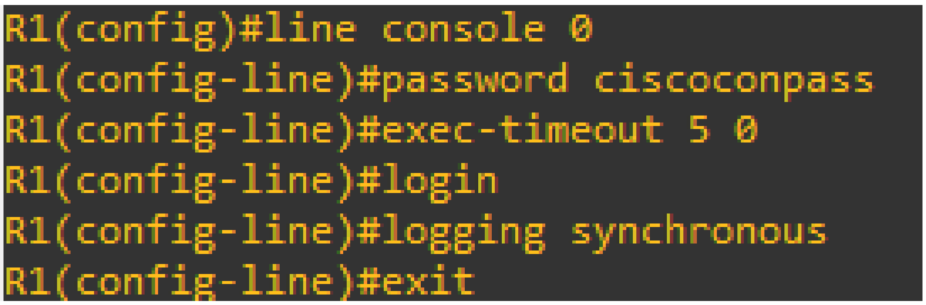
b. Configure the enable secret encrypted password on both routers.

```
R1(config)# enable secret class12345
```

```
R1(config)#enable secret class12345
```

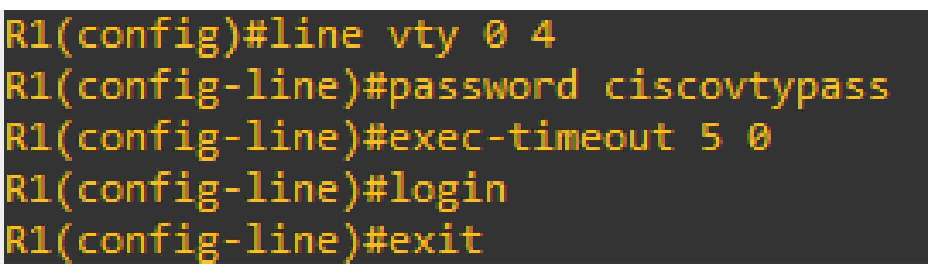
c. Configure a console password and enable login for routers. For additional security, the exec-timeout command causes the line to log out after 5 minutes of inactivity. The logging synchronous command prevents console messages from interrupting command entry.

```
R1(config)# line console 0
R1(config-line)# password ciscoconpass
R1(config-line)# exec-timeout 5 0
R1(config-line)# login
R1(config-line)# logging synchronous
R1(config-line)# exit
```



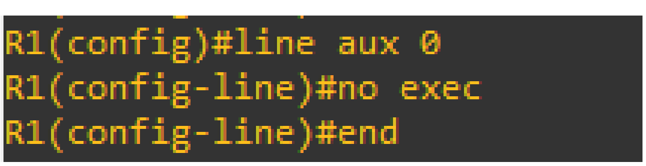
d. Configure the password on the vty lines for router R1.

```
R1(config)# line vty 0 4
R1(config-line)# password ciscovtypass
R1(config-line)# exec-timeout 5 0
R1(config-line)# login
R1(config-line)# exit
```



e. The aux port is a legacy port used to manage a router remotely using a modem and is hardly ever used. Therefore, disable the aux port.

```
R1(config)# line aux 0
R1(config-line)# no exec
R1(config-line)# end
```



f. Enter privileged EXEC mode and issue the show run command.

```
R1#show run
Building configuration...

Current configuration : 1941 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
security passwords min-length 10
enable secret 5 $1$rPJA$9OBR8777USn2ZPWN0qSk60
!
no aaa new-model
!
resource policy
!
ip subnet-zero
no ip icmp rate-limit unreachable
ip cef
ip tcp synwait-time 5
 --More--
```

```
!
line con 0
 exec-timeout 5 0
 privilege level 15
 password ciscoconpass
 logging synchronous
 login
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 no exec
 stopbits 1
line vty 0 4
 exec-timeout 5 0
 password ciscovtypass
 login
!
!
end
```

g. Use the service password-encryption command to encrypt the line console and vty passwords.

```
R1(config)# service password-encryption
```

```
R1(config)#service password-encryption
R1(config)#end
```

h. Issue the show run command.

```
R1#show run
Building configuration...

Current configuration : 1970 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
```

i. Configure a warning to unauthorized users with a message-of-the-day (MOTD) banner using the banner motd command. When a user connects to one of the routers, the MOTD banner appears before the login prompt. In this example, the dollar sign ($) is used to start and end the message.

```
R1(config)# banner motd $Unauthorized access strictly
prohibited!$
R1(config)# exit
```

```
R1(config)#banner motd $Unauthorized access strictly prohibited!$
R1(config)#exit
```

j. Issue the show run command

```
!
banner motd ^CUnauthorized access strictly prohibited!^C
!
line con 0
 exec-timeout 5 0
 privilege level 15
 password 7 060506324F410A160B0713181F
 logging synchronous
 --More--
```

k. Exit privileged EXEC mode using the disable or exit command and press Enter to get started.

```
R1 con0 is now available



Press RETURN to get started.



Unauthorized access strictly prohibited!

User Access Verification

Password:
Password:
R1#
```

1. Repeat the configuration portion of steps 3a through 3k on router R3.

```
R3(config)#security passwords min-length 10
R3(config)#enable secret class12345
R3(config)#line console 0
R3(config-line)#password ciscoconpass
R3(config-line)#exec-timeout 5 0
R3(config-line)#login
R3(config-line)#logging synchronous
R3(config-line)#exit
R3(config)#line vty 0 4
R3(config-line)#password ciscovtypass
R3(config-line)#exec-timeout 5 0
R3(config-line)#login
R3(config-line)#exit
R3(config)#line aux 0
R3(config-line)#no exec
R3(config-line)#end
R3#show run
R3(config)#service password-encryption
R3(config)# banner motd $Unauthorized access strictly
prohibited!$
R3(config)#exit
R3#show run
```

```
R3(config)#security password min-length 10
R3(config)#enable secret class12345
R3(config)#line console 0
R3(config-line)#password ciscoconpass
R3(config-line)#exec-timeout 5 0
R3(config-line)#login
R3(config-line)#logging synchronous
R3(config-line)#exit
R3(config)#line vty 0 4
R3(config-line)#password ciscovtypass
R3(config-line)#exec-timeout 5 0
R3(config-line)#login
R3(config-line)#exit
R3(config)#line aux 0
R3(config-line)#no exec
R3(config-line)#end
```

```
R3(config)#service password-encryption
```

```
R3(config)#banner motd $Unauthorized access strictly prohibited!$
R3(config)#exit
```

```
R3 con0 is now available



Press RETURN to get started.


Unauthorized access strictly prohibited!

User Access Verification

Password:
R3#
```

Step 4. Configure enhanced username password security.

To increase the encryption level of console and VTY lines, it is recommended to enable authentication using the local database. The local database consists of usernames and password combinations that are created locally on each device. The local and VTY lines are configured to refer to the local database when authenticating a user.

a. To create a local database entry encrypted to level 4 (SHA256), use the username name secret password global configuration command. In global configuration mode, enter the following command:

```
R1(config)#username JR-ADMIN secret class12345
R1(config)#username ADMIN secret class54321
```

b. Set the console line to use the locally defined login accounts.

```
R1(config)# line console 0
R1(config-line)# login local
R1(config-line)# exit
```

```
R1(config)#line console 0
R1(config-line)#login local
R1(config-line)#exit
```

c. Set the vty lines to use the locally defined login accounts.

```
R1(config)# line vty 0 4
R1(config-line)# login local
```

```
R1(config)#line vty 0 4
R1(config-line)#login local
```

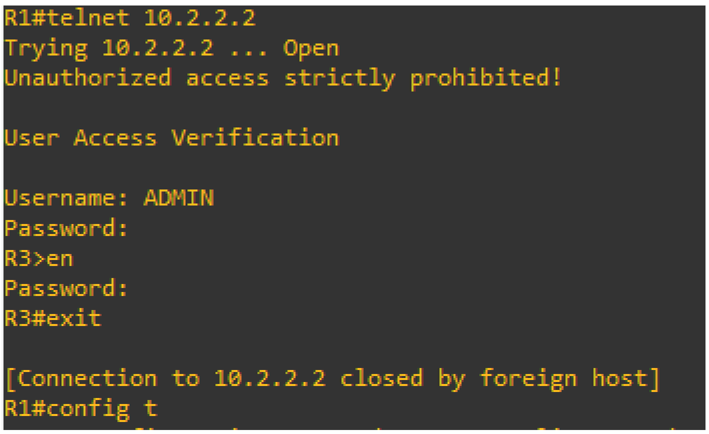
d. Repeat the steps 4a to 4c on R3.

```
R3(config)# username JR-ADMIN secret class12345
R3(config)# username ADMIN secret class54321
R3(config)# line console 0
R3(config-line)# login local
R3(config-line)# exit
R3(config)# line vty 0 4
R3(config-line)# login local
R3(config-line)# end
```

```
R3(config)#username JR-ADMIN secret class12345
R3(config)#username ADMIN secret class54321
R3(config)#line console 0
R3(config-line)#login local
R3(config-line)#exit
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)#end
```

e. To verify the configuration, telnet to R3 from R1 and login using the ADMIN local database account.

```
R1# telnet 10.2.2.2
Trying 10.2.2.2 ... Open
Unauthorized access strictly prohibited!
User Access Verification
Username: ADMIN
Password:
```



Step 5 : Enabling secure remote management using SSH.

Traditionally, remote access on routers was configured using Telnet on TCP port 23. However, Telnet was developed in the days when security was not an issue; therefore, all Telnet traffic is forwarded in plaintext.

Secure Shell (SSH) is a network protocol that establishes a secure terminal emulation connection to a router or other networking device. SSH encrypts all information that passes over the network link and provides authentication of the remote computer. SSH is rapidly replacing Telnet as the remote login tool of choice for network professionals.

Note: For a router to support SSH, it must be configured with local authentication, (AAA services, or username) or password authentication. In this task, you configure an SSH username and local authentication.

In this step, you will enable R1 and R3 to support SSH instead of Telnet.

a. SSH requires that a device name and a domain name be configured. Since the router already has a name assigned, configure the domain name

R1(config)# ip domain-name ccnasecurity.com

```
R1(config)#ip domain-name ccnasecurity.com
```

b. The router uses the RSA key pair for authentication and encryption of transmitted SSH data. Although optional it may be wise to erase any existing key pairs on the router.

R1(config)# crypto key zeroize rsa

```
R1(config)#crypto key zeroize rsa
% No Signature RSA Keys found in configuration.
```

Note: If no keys exist, you might receive this message: % No Signature RSA Keys found in configuration.

c. Generate the RSA encryption key pair for the router. Configure the RSA keys with 1024 for the number of modulus bits. The default is 512, and the range is from 360 to 2048.

R1(config)# crypto key generate rsa general-keys modulus 1024

```
R1(config)#crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.ccnasecurity.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#
*May 16 09:48:29.831: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

d. Cisco routers support two versions of SSH:

• SSH version 1 (SSHv1): Original version but has known vulnerabilities.

• SSH version 2 (SSHv2): Provides better security using the Diffie-Hellman key exchange and the strong integrity-checking message authentication code (MAC).

The default setting for SSH is SSH version 1.99. This is also known as compatibility mode and is merely an indication that the server supports both SSH version 2 and SSH version 1. However, best practices are to enable version 2 only.

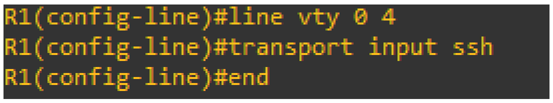
Configure SSH version 2 on R1.

R1(config)# ip ssh version 2

```
R1(config)#ip ssh version 2
```

e. Configure the vty lines to use only SSH connections.

```
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
R1(config-line)# end
```



```
R1(config-line)#line vty 0 4
R1(config-line)#transport input ssh
R1(config-line)#end
```

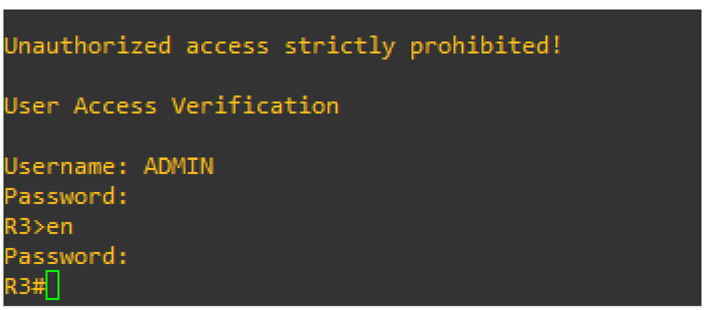
Note: SSH requires that the login local command be configured. However, in the previous step we enabled AAA authentication using the TELNET-LOGIN authentication method, therefore login local is not necessary. Note: If you add the keyword telnet to the transport input command, users can log in using Telnet as well as SSH. However, the router will be less secure. If only SSH is specified, the connecting host must have an SSH client installed.

f. Verify the SSH configuration using the show ip ssh command.

```
R1# show ip ssh
```

```
R1#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

g. Repeat the steps 6a to 6f on R3.



```
Unauthorized access strictly prohibited!

User Access Verification

Username: ADMIN
Password:
R3>en
Password:
R3#
```

Note: R3> en
Password: class12345

h. Although a user can SSH from a host using the SSH option of TeraTerm of PuTTY, a router can also SSH to another SSH enabled device. SSH to R3 from R1.

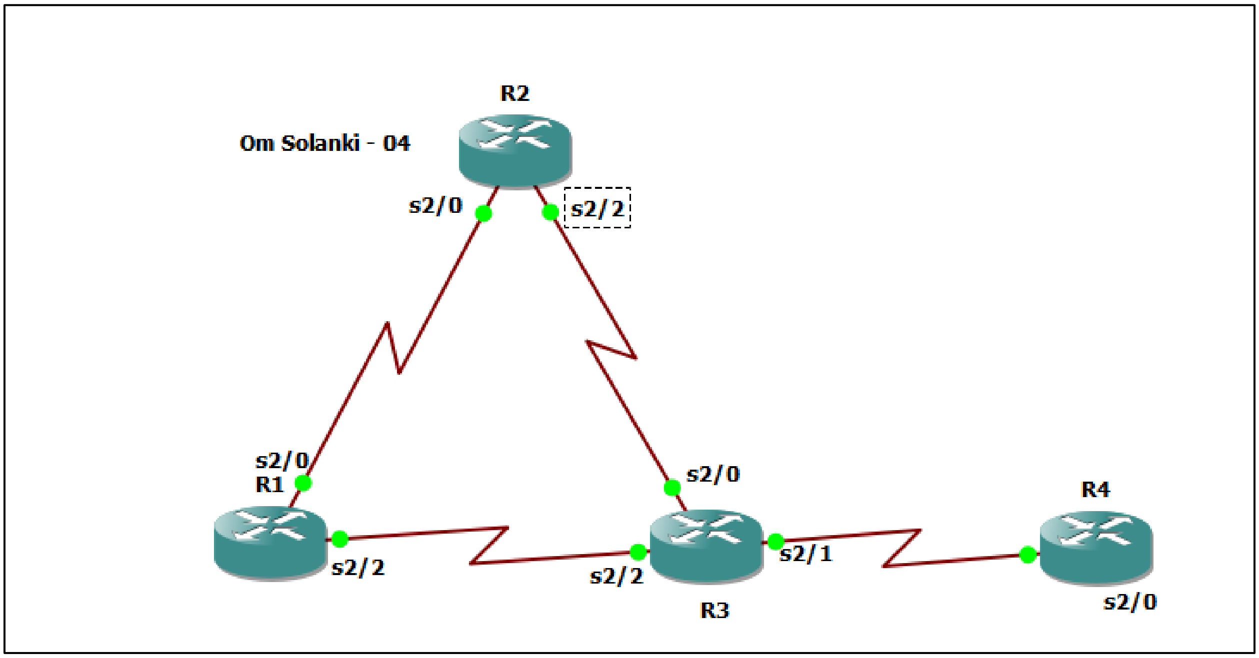
```
R1# ssh -l ADMIN 10.2.2.2
Password:
Unauthorized access strictly prohibited!
R3>
R3> en
Password:
R3#
```

```
Password:
R1#ssh -l ADMIN 10.2.2.2

Password:
Unauthorized access strictly prohibited!
R3>en
Password:
R3#
```

PRACTICAL NO:6

Aim: Configure and Verify Path Control Using PBR

Topology:



Objectives:

→ Configure and verify policy-based routing.

→ Select the required tools and commands to configure policy-based routing operations.

→ Verify the configuration and operation by using the proper show and debug commands.

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 previous configurations.

b. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to these and the serial interfaces on R1, R2, R3, and R4. On the serial interfaces connecting R1 to R3 and R3 to R4, specify the bandwidth as 64 Kb/s and set a clock rate on the DCE using the clock rate 64000 command. On the serial interfaces connecting R1 to R2 and R2 to R3, specify the bandwidth as 128 Kb/s and set a clock rate on the DCE using the clock rate 128000 command.

Router 1:

```
hostname R1
interface Lo1
description R1 LAN
ip address 192.168.1.1 255.255.255.0
interface Serial1/0
description R1 --> R2
ip address 172.16.12.1 255.255.255.248
clock rate 128000
bandwidth 128
no shutdown
interface Serial1/1
description R1 --> R3
ip address 172.16.13.1 255.255.255.248
bandwidth 64
no shutdown
end
```

```
R1(config)#hostname R1
R1(config)#interface Lo1
R1(config-if)#
*May 17 10:12:26.459: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed
 state to up
R1(config-if)#description R1 LAN
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#int se2/0
R1(config-if)#description R1-->R2
R1(config-if)#ip address 172.16.12.1 255.255.255.248
R1(config-if)#clock rate 128000
R1(config-if)#bandwidth 128
R1(config-if)#no shutdown
R1(config-if)#exit
*May 17 10:13:40.435: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
*May 17 10:13:40.435: %ENTITY_ALARM-6-INFO: CLEAR INFO Se2/0 Physical Port Administrativ
e State Down
R1(config-if)#exit
*May 17 10:13:41.439: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed
 state to up
R1(config-if)#exit
R1(config)#int se2/2
R1(config-if)#
*May 17 10:14:04.311: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed
 state to down
R1(config-if)#description R1-->R3
R1(config-if)#ip add 172.16.13.1 255.255.255.248
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#end
```

Router 2

```
hostname R2
interface Lo2
description R2 LAN
ip address 192.168.2.1 255.255.255.0
interface Serial1/0
description R2 --> R1
ip address 172.16.12.2 255.255.255.248
bandwidth 128
no shutdown
interface Serial1/1
description R2 --> R3
ip address 172.16.23.2 255.255.255.248
clock rate 128000
bandwidth 128
no shutdown
end
```

```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname R2
R2(config)#interface Lo2
R2(config-if)#description R2 LAN
R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#exit
R2(config)#
*May 17 10:16:28.475: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback2, changed
 state to up
R2(config)#int se2/0
R2(config-if)#description R2-->R1
R2(config-if)#ip address 172.16.12.2 255.255.255.248
R2(config-if)#bandwidth 128
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
*May 17 10:17:01.715: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
R2(config)#i
*May 17 10:17:01.715: %ENTITY_ALARM-6-INFO: CLEAR INFO Se2/0 Physical Port Administrativ
e State Down
R2(config)#int
*May 17 10:17:02.719: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed
 state to up
R2(config)#int se2/2
R2(config-if)#description R2-->R3
R2(config-if)#ip add 172.16.23.2 255.255.255.248
R2(config-if)#bandwidth 64
R2(config-if)#no shutdown
R2(config-if)#end
```

Router 3

```
hostname R3
interface Lo3
description R3 LAN
ip address 192.168.3.1 255.255.255.0
interface Serial1/1
description R3 --> R1
ip address 172.16.13.3 255.255.255.248
clock rate 64000
bandwidth 64
no shutdown
interface Serial1/0
description R3 --> R2
ip address 172.16.23.3 255.255.255.248
bandwidth 128
no shutdown
interface Serial1/2
description R3 --> R4
ip address 172.16.34.3 255.255.255.248
clock rate 64000
bandwidth 64
no shutdown
end
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname R3
R3(config)#interface Lo3
R3(config-if)#description R3 LAN
R3(config-if)#ip address 192.168.3.1 255.255.255.0
R3(config-if)#exit
R3(config)#
*May 17 10:19:07.191: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback3, changed
 state to up
R3(config)#int se2/2
R3(config-if)#description R3-->R1
R3(config-if)#ip address 172.16.13.3 255.255.255.248
R3(config-if)#bandwidth 128
R3(config-if)#no shutdown
R3(config-if)#exit
```

```
R3(config)#int se2/0
R3(config-if)#description R3-->R2
R3(config-if)#ip add 172.16.23.3 255.255.255.248
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*May 17 10:19:56.159: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
R3(config)#in
*May 17 10:19:56.159: %ENTITY_ALARM-6-INFO: CLEAR INFO Se2/0 Physical Port Administrativ
e State Down
R3(config)#int
*May 17 10:19:57.163: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed
 state to up
R3(config)#int se2/1
R3(config-if)#description R3-->R4
R3(config-if)#ip add 172.16.34.3 255.255.255.248
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
R3(config-if)#end
```

Router 4

```
hostname R4
interface Lo4
description R4 LAN A
ip address 192.168.4.1 255.255.255.128
interface Lo5
description R4 LAN B
ip address 192.168.4.129 255.255.255.128
interface Serial1/0
description R4 --> R3
ip address 172.16.34.4 255.255.255.248
bandwidth 64
no shutdown
end
```

```
R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#hostname R4
R4(config)#interface Lo4
R4(config-if)#description R4 LAN A
R4(config-if)#ip address 192.168.4.1 255.255.255.128
R4(config-if)#exit
R4(config)#
*May 17 10:21:12.683: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback4, changed
 state to up
R4(config)#int Lo5
R4(config-if)#description R4 LAN B
R4(config-if)#ip address 192.168.4.129 255.255.255.128
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#
*May 17 10:21:54.367: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback5, changed
 state to up
R4(config)#int se2/0
R4(config-if)#description R4-->R3
R4(config-if)#ip add 172.16.34.4 255.255.255.248
R4(config-if)#bandwidth 64
R4(config-if)#no shutdown
R4(config-if)#end
R4#
```

d. Verify the configuration with the show ip interface brief, show protocols, and show interfaces description commands. The output from router R3 is shown here as an example.

```
R3# show ip interface brief | include up
R3# show protocols
R3# show interfaces description | include up
```

```
R3#show ip interface brief | include up
Serial2/0              172.16.23.3     YES manual up                    up
Serial2/1              172.16.34.3     YES manual up                    up
Serial2/2              172.16.13.3     YES manual up                    up
Loopback3              192.168.3.1     YES manual up                    up
```

```
R3#show interfaces description | include up
Se2/0                          up             up       R3-->R2
Se2/1                          up             up       R3-->R4
Se2/2                          up             up       R3-->R1
Lo3                            up             up       R3 LAN
R3#
```

Step 3: Configure basic EIGRP.

a. Implement EIGRP AS 1 over the serial and loopback interfaces as you have configured it for the other EIGRP labs.

b. Advertise networks 172.16.12.0/29, 172.16.13.0/29, 172.16.23.0/29, 172.16.34.0/29, 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24, and 192.168.4.0/24 from their respective routers.

Router R1

```
router eigrp 1
network 192.168.1.0
network 172.16.12.0 0.0.0.7
network 172.16.13.0 0.0.0.7
no auto-summary
```

```
R1(config)#router eigrp 1
R1(config-router)#network 192.168.1.0
R1(config-router)#network 172.16.12.0 0.0.0.7
R1(config-router)#network 172.16.13.0 0.0.0.7
R1(config-router)#no auto-summary
R1(config-router)#
```

Router R2

```
router eigrp 1
network 192.168.2.0
network 172.16.12.0 0.0.0.7
network 172.16.23.0 0.0.0.7
no auto-summary
```

```
R2(config)#router eigrp 1
R2(config-router)#network 192.168.2.0
R2(config-router)#network 172.16.12.0 0.0.0.7
R2(config-router)#network 172.16.23.0 0.0.0.7
R2(config-router)#no auto-summary
```

Router R3

```
router eigrp 1
network 192.168.3.0
network 172.16.13.0 0.0.0.7
network 172.16.23.0 0.0.0.7
network 172.16.34.0 0.0.0.7
no auto-summary
```

```
R3(config)#router eigrp 1
R3(config-router)#network 192.168.3.0
R3(config-router)#network 172.16.13.0 0.0.0.7
R3(config-router)#network 172.16.23.0 0.0.0.7
R3(config-router)#network 172.16.34.0 0.0.0.7
R3(config-router)#no auto-summary
```

Router R4

```
router eigrp 1
network 192.168.4.0
network 172.16.34.0 0.0.0.7
no auto-summary
```

```
R4(config)#router eigrp 1
R4(config-router)#network 192.168.4.0
R4(config-router)#network 172.16.34.0 0.0.0.7
R4(config-router)#no auto-summary
```

You should see EIGRP neighbor relationship messages being generated.

Step 4: Verify EIGRP connectivity.

a. Verify the configuration by using the show ip eigrp neighbors command to check which routers have EIGRP adjacencies.

```
R1# show ip eigrp neighbors
```

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.13.3             Se2/2             12 00:02:07   43  2280  0  23
0   172.16.12.2             Se2/0             14 00:03:47   32  1140  0  16
R1#
```

```
R2# show ip eigrp neighbors
```

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.23.3             Se2/2             12 00:02:27   48  2280  0  28
0   172.16.12.1             Se2/0             12 00:04:08   40  1140  0  20
R2#
```

R3# show ip eigrp neighbors

```
R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
2   172.16.34.4             Se2/1             10 00:00:55   36   2280  0  6
1   172.16.13.1             Se2/2             13 00:01:11   48   1140  0  18
0   172.16.23.2             Se2/0             12 00:01:11   39   2280  0  17
R3#
```

R4# show ip eigrp neighbors

```
R4#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.34.3             Se2/0             11 00:05:02   47   2280  0  26
R4#
```

b. Run the following Tcl script on all routers to verify full connectivity.

```
R1# tclsh
foreach address {
172.16.12.1
172.16.12.2
172.16.13.1
172.16.13.3
172.16.23.2
172.16.23.3
172.16.34.3
172.16.34.4
192.168.1.1
192.168.2.1
192.168.3.1
192.168.4.1
192.168.4.129
} { ping $address }
```

```
R1#
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#192.168.4.129
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/79/92 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/29/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/99/112 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/40/52 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/70/80 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/64/84 ms
Type escape sequence to abort.
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/99/112 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/40/52 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/70/80 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/64/84 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/30/40 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/89/104 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/40/52 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/48/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/88/108 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/77/92 ms
R1(tcl)#
```

You should get ICMP echo replies for every address pinged. Make sure to run the Tcl script on each router.

Step 5: Verify the current path.

Before you configure PBR, verify the routing table on R1.

a. On R1, use the show ip route command. Notice the next-hop IP address for all networks discovered by EIGRP.

R1# show ip route | begin Gateway

```
R1#show ip route | begin Gateway
Gateway of last resort is not set

     172.16.0.0/29 is subnetted, 4 subnets
D       172.16.34.0 [90/41024000] via 172.16.13.3, 00:06:37, Serial2/2
D       172.16.23.0 [90/41024000] via 172.16.13.3, 00:06:37, Serial2/2
                    [90/41024000] via 172.16.12.2, 00:06:37, Serial2/0
C       172.16.12.0 is directly connected, Serial2/0
C       172.16.13.0 is directly connected, Serial2/2
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:06:20, Serial2/2
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:06:20, Serial2/2
C    192.168.1.0/24 is directly connected, Loopback1
D    192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:08:18, Serial2/0
D    192.168.3.0/24 [90/40640000] via 172.16.13.3, 00:06:37, Serial2/2
R1#
```

b. On R4, use the traceroute command to the R1 LAN address and source the ICMP packet from R4 LAN A and LAN B. Note: You can specify the source as the interface address (for example 192.168.4.1) or the interface designator (for example, Fa0/0).

R4# traceroute 192.168.1.1 source 192.168.4.1
R4# traceroute 192.168.1.1 source 192.168.4.129

Notice that the path taken for the packets sourced from the R4 LANs are going through R3 → R2 → R1.

```
R4#traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 44 msec 28 msec 52 msec
  2 172.16.13.1 100 msec 76 msec 60 msec
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 52 msec 40 msec 36 msec
  2 172.16.13.1 72 msec 96 msec 84 msec
R4#
R4#traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 20 msec 32 msec 32 msec
  2 172.16.13.1 92 msec 96 msec 116 msec
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 48 msec 36 msec 44 msec
  2 172.16.13.1 100 msec 104 msec 80 msec
R4#
```

c. On R3, use the show ip route command and note that the preferred route from R3 to R1 LAN 192.168.1.0/24 is via R2 using the R3 exit interface S0/0/1.

```
R3# show ip route | begin Gateway
```

```
R3#show ip route | begin Gateway
Gateway of last resort is not set

     172.16.0.0/29 is subnetted, 4 subnets
C       172.16.34.0 is directly connected, Serial2/1
C       172.16.23.0 is directly connected, Serial2/0
D       172.16.12.0 [90/21024000] via 172.16.13.1, 00:07:50, Serial2/2
C       172.16.13.0 is directly connected, Serial2/2
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/40640000] via 172.16.34.4, 00:07:33, Serial2/1
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:07:33, Serial2/1
D    192.168.1.0/24 [90/20640000] via 172.16.13.1, 00:07:50, Serial2/2
D    192.168.2.0/24 [90/21152000] via 172.16.13.1, 00:07:50, Serial2/2
C    192.168.3.0/24 is directly connected, Loopback3
```

d. On R3, use the show interfaces serial 1/1 and show interfaces s1/0 commands

```
R3# show interfaces serial1/2
```

```
R3# show interfaces serial1/1 | include BW

R3# show interfaces serial1/0 | include BW
```

```
R3#show interfaces serial2/1
Serial2/1 is up, line protocol is up
  Hardware is M8T-X.21
  Description: R3-->R4
  Internet address is 172.16.34.3/29
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 48 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     278 packets input, 19247 bytes, 0 no buffer
     Received 112 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     301 packets output, 21440 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

```
R3#show interfaces serial2/2 | include BW
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
R3#show interfaces serial2/0 | include BW
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
```

Notice that the bandwidth of the serial link between R3 and R1 (S1/1) is set to 64 Kb/s, while the bandwidth of the serial link between R3 and R2 (S1/1) is set to 128 Kb/s.

e. Confirm that R3 has a valid route to reach R1 from its serial 0/0/0 interface using the show ip eigrp topology 192.168.1.0 command.

```
R3# show ip eigrp topology 192.168.1.0
```

```
R3#show ip eigrp topology 192.168.1.0
IP-EIGRP (AS 1): Topology entry for 192.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 20640000
  Routing Descriptor Blocks:
  172.16.13.1 (Serial2/2), from 172.16.13.1, Send flag is 0x0
      Composite metric is (20640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  172.16.23.2 (Serial2/0), from 172.16.23.2, Send flag is 0x0
      Composite metric is (41152000/20640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

As indicated, R4 has two routes to reach 192.168.1.0. However, the metric for the route to R1 (172.16.13.1) is much higher (40640000) than the metric of the route to R2 (21152000), making the route through R2 the successor route.

Step 6: Configure PBR to provide path control.

Now you will deploy source-based IP routing by using PBR. You will change a default IP routing decision based on the EIGRP-acquired routing information for selected IP source-todestination flows and apply a different next-hop router.
Recall that routers normally forward packets to destination addresses based on information in their routing table. By using PBR, you can implement policies that selectively cause packets to take different paths based on source address, protocol type, or application type. Therefore, PBR overrides the router's normal routing behavior.
Configuring PBR involves configuring a route map with match and set commands and then applying the route map to the interface.
The steps required to implement path control include the following:
→ Choose the path control tool to use. Path control tools manipulate or bypass the IP routing table. For PBR, route-map commands are used.
→ Implement the traffic-matching configuration, specifying which traffic will be manipulated. The match commands are used within route maps.
→ Define the action for the matched traffic using set commands within route maps.
→ Apply the route map to incoming traffic.
As a test, you will configure the following policy on router R3:
• All traffic sourced from R4 LAN A must take the R3 → R2 → R1 path.
• All traffic sourced from R4 LAN B must take the R3 → R1 path.

a. On router R3, create a standard access list called PBR-ACL to identify the R4 LAN B network.

```
R3(config)# ip access-list standard PBR-ACL
R3(config-std-nacl)# remark ACL matches R4 LAN B traffic
R3(config-std-nacl)# permit 192.168.4.128 0.0.0.127
R3(config-std-nacl)# exit
```

```
R3(config)#ip access-list standard PBR-ACL
R3(config-std-nacl)#remark ACL matches R4 LAN B traffic
R3(config-std-nacl)#permit 192.168.4.128 0.0.0.127
R3(config-std-nacl)#exit
```

b. Create a route map called R3-to-R1 that matches PBR-ACL and sets the next-hop interface to the R1 serial 0/0/1 interface.

```
R3(config)# route-map R3-to-R1 permit
R3(config-route-map)# description RM to forward LAN B traffic
to R1
R3(config-route-map)# match ip address PBR-ACL
R3(config-route-map)# set ip next-hop 172.16.13.1
R3(config-route-map)# exit
```

```
R3(config)#route-map R3-to-R1 permit
R3(config-route-map)#description RM to forward LAN B traffic to R1
R3(config-route-map)#match ip address PBR-ACL
R3(config-route-map)#set ip next-hop 172.16.13.1
R3(config-route-map)#exit
```

c. Apply the R3-to-R1 route map to the serial interface on R3 that receives the traffic from R4. Use the ip policy route-map command on interface S1/1.

```
R3(config)# interface s1/1
R3(config-if)# ip policy route-map R3-to-R1
R3(config-if)# end
```

```
R3(config)#interface s2/2
R3(config-if)#ip policy route-map R3-to-R1
R3(config-if)#end
```

d. On R3, display the policy and matches using the show route-map command.

```
R3# show route-map
```

```
R3#show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 0 packets, 0 bytes
```

Note: There are currently no matches because no packets matching the ACL have passed through R3 S0/1/0.

Step 7: Test the policy.

Now you are ready to test the policy configured on R3. Enable the debug ip policy command on R3 so that you can observe the policy decision-making in action. To help filter the traffic, first create a standard ACL that identifies all traffic from the R4 LANs.

a. On R3, create a standard ACL which identifies all of the R4 LANs.

```
R3# conf t
R3(config)# access-list 1 permit 192.168.4.0 0.0.0.255
R3(config)# exit
```

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#access-list 1 permit 192.168.4.0 0.0.0.255
R3(config)#exit
```

b. Enable PBR debugging only for traffic that matches the R4 LANs

```
R3# debug ip policy ?
R3# debug ip policy 1
```

```
R3#debug ip policy ?
  <1-199>  Access list
  dynamic  dynamic PBR
  <cr>

R3#debug ip policy 1
Policy routing debugging is on for access list 1
```

c. Test the policy from R4 with the traceroute command, using R4 LAN A as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.1
```

```
R4#traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 44 msec 28 msec 52 msec
  2 172.16.13.1 100 msec 76 msec 60 msec
```

Notice the path taken for the packet sourced from R4 LAN A is still going through

R3 → R2 → R1.

As the traceroute was being executed, router R3 should be generating the following debug output.

R3#
Jan 10 10:49:48.411: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, policy rejected -- normal forwarding
Jan 10 10:49:48.427: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, policy rejected -- normal forwarding
Jan 10 10:49:48.439: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, policy rejected -- normal forwarding
Jan 10 10:49:48.451: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwarding
Jan 10 10:49:48.471: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwarding
Jan 10 10:49:48.491: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwarding
Jan 10 10:49:48.511: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwarding
Jan 10 10:49:48.539: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwarding
Jan 10 10:49:51.539: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwarding

d. Test the policy from R4 with the traceroute command, using R4 LAN B as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.129
```

Now the path taken for the packet sourced from R4 LAN B is R3 → R1, as expected.

```
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 52 msec 40 msec 36 msec
  2 172.16.13.1 72 msec 96 msec 84 msec
```

The debug output on R3 also confirms that the traffic meets the criteria of the R3-to-R1 policy.

R3#
Jan 10 10:50:04.283: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, policy match
Jan 10 10:50:04.283: IP: route map R3-to-R1, item 10, permit
Jan 10 10:50:04.283: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1 (Serial0/0/0), len 28, policy routed
Jan 10 10:50:04.283: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
Jan 10 10:50:04.295: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, policy match
Jan 10 10:50:04.295: IP: route map R3-to-R1, item 10, permit
Jan 10 10:50:04.295: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1 (Serial0/0/0), len 28, policy routed
Jan 10 10:50:04.295: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
Jan 10 10:50:04.311: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, policy match
Jan 10 10:50:04.311: IP: route map R3-to-R1, item 10, permit
Jan 10 10:50:04.311: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1 (Serial0/0/0), len 28, policy routed
Jan 10 10:50:04.311: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
Jan 10 10:50:04.323: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy match
Jan 10 10:50:04.323: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, PBR Counted
Jan 10 10:50:04.323: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, g=172.16.13.1, len 28, FIB policy routed
Jan 10 10:50:04.351: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy match
Jan 10 10:50:04.351: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, PBR Counted
Jan 10 10:50:04.351: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, g=172.16.13.1, len 28, FIB policy routed
Jan 10 10:50:07.347: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy match
Jan 10 10:50:07.347: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, PBR Counted

Jan 10 10:50:07.347: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, g=172.16.13.1, len 28, FIB policy routed

e. On R3, display the policy and matches using the show route-map command.

```
R3# show route-map
```

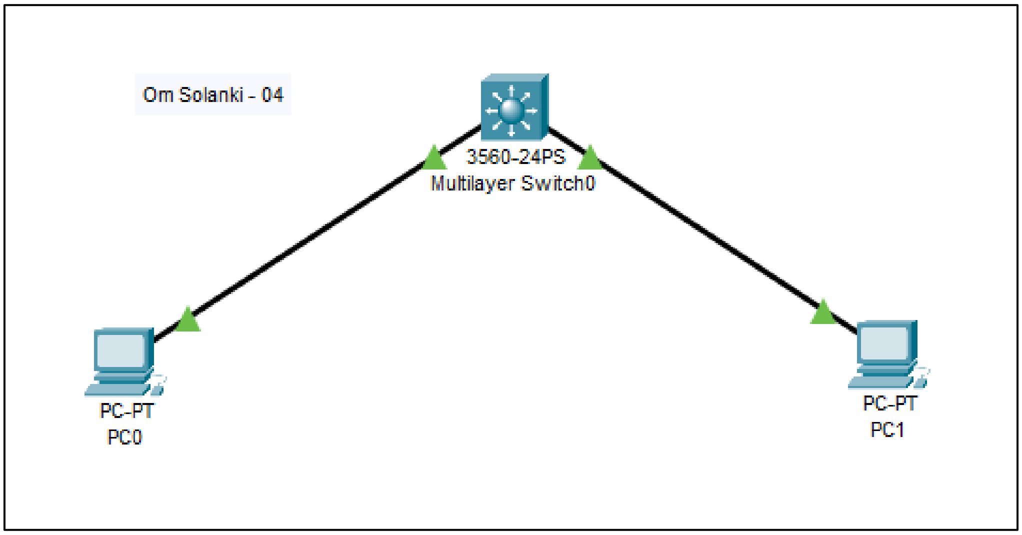
Note: There are now matches to the policy because packets matching the ACL have passed through R3 S0/1/0.

```
R3#show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 0 packets, 0 bytes
R3#
```

PRACTICAL NO: 7

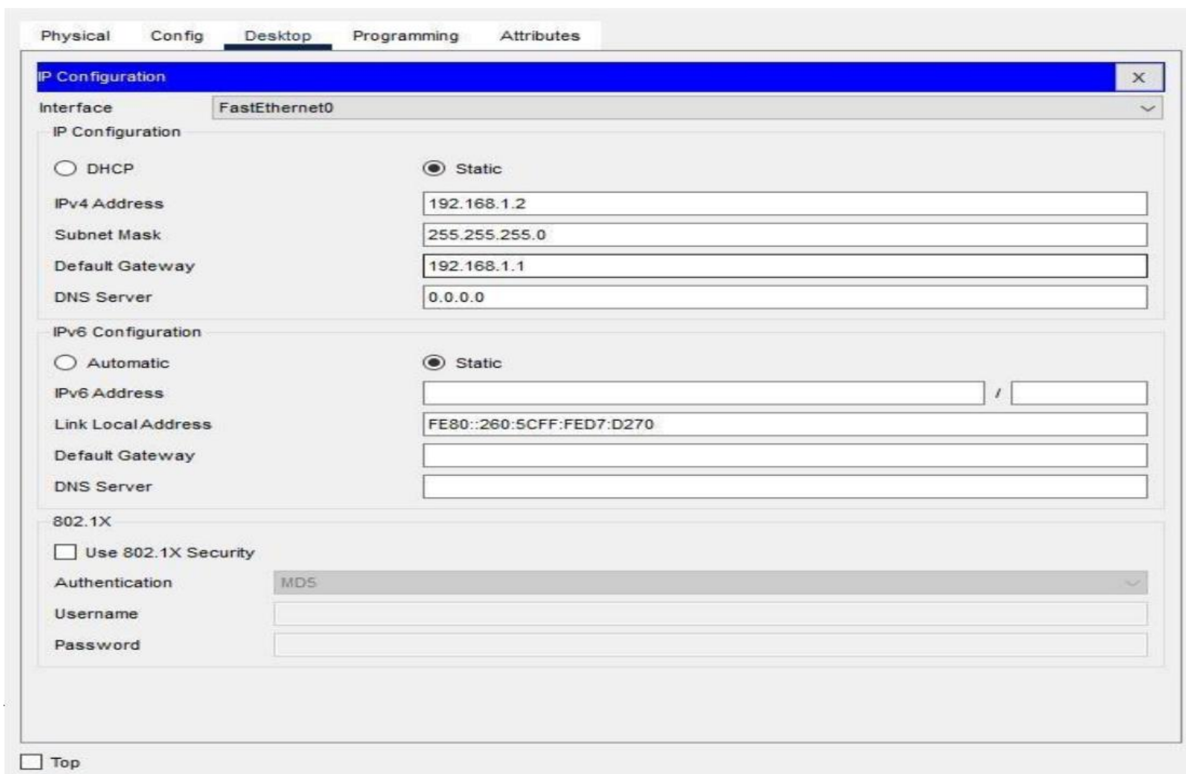
Aim: Inter VLAN Routing

Topology:

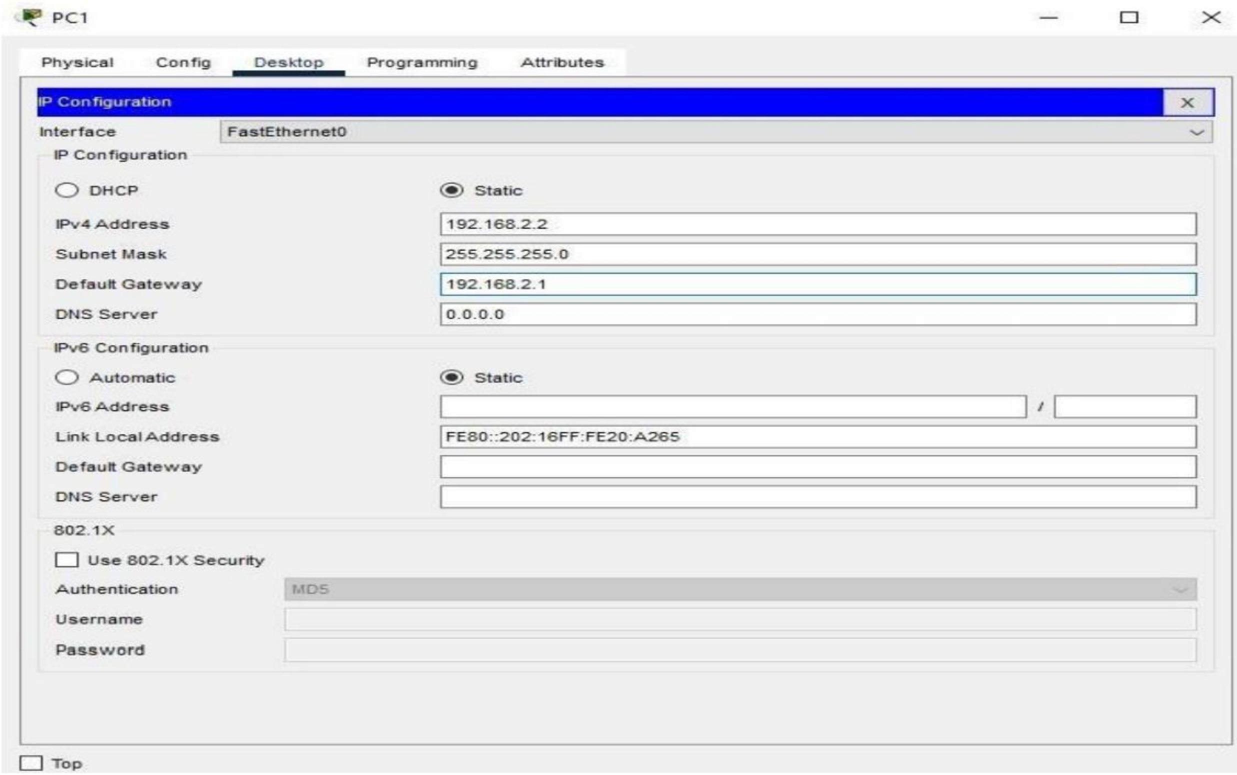


We Configure the IP addresses on PC

PC0:



PC1:



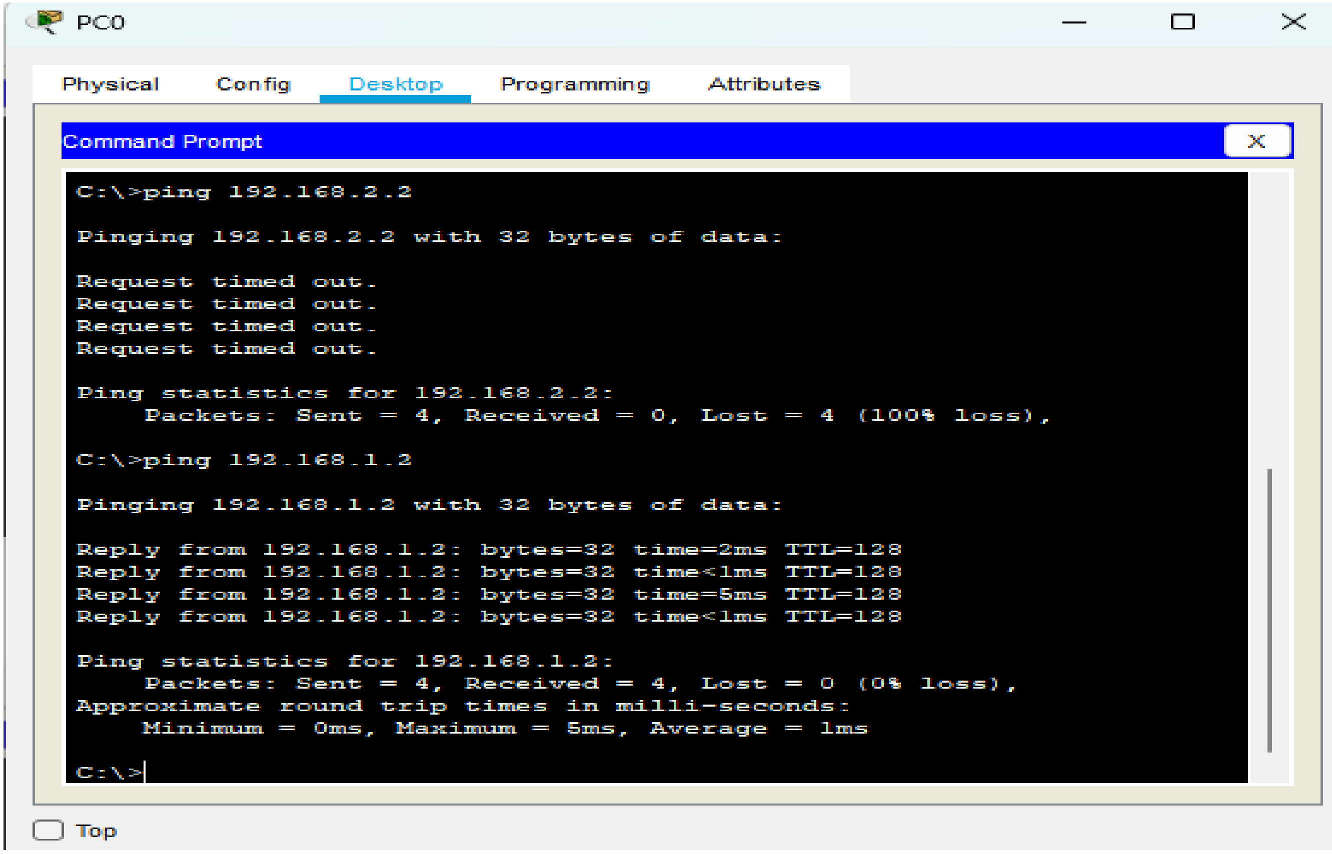
Now we configure the Multilayer switch using the following command in the CLI mode.

<u>**Code:**</u>

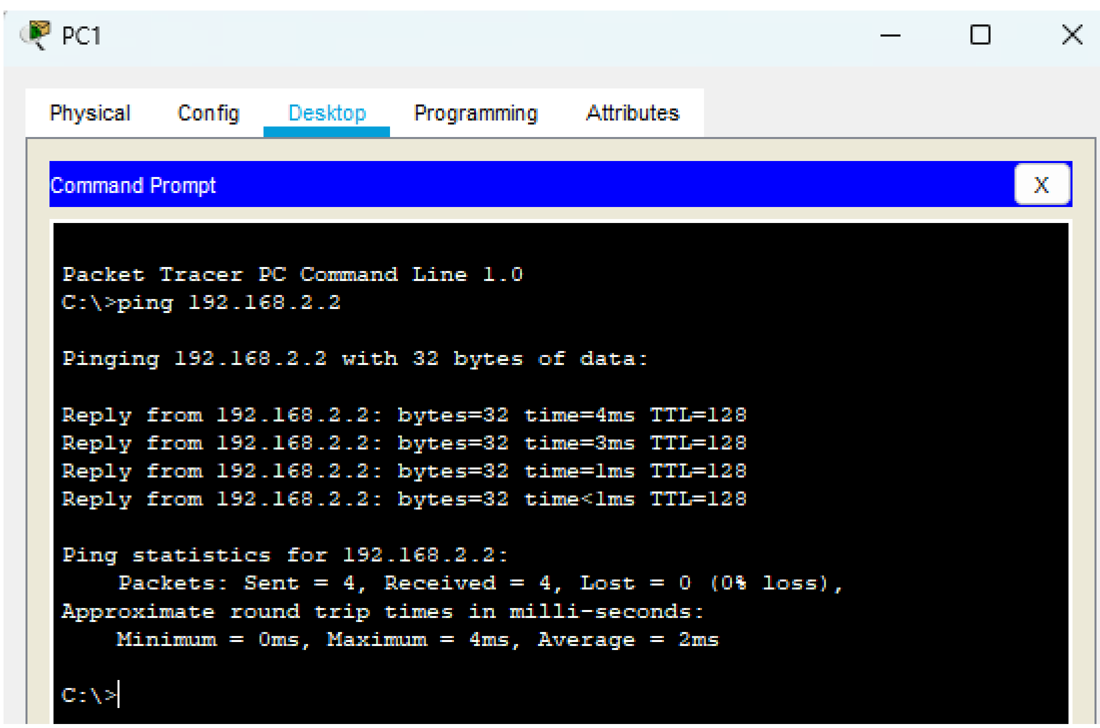
```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name Om
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name cisco
Switch(config-vlan)#exit
Switch(config)#int fastEthernet 0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#int fastEthernet 0/2
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface vlan 10
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
```

```
Switch(config-if)#ip address 192.168.1.1 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#interface vlan 20
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up

Switch(config-if)#ip address 192.168.2.1 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/1        unassigned      YES unset  up                    up
FastEthernet0/2        unassigned      YES unset  up                    up
FastEthernet0/3        unassigned      YES unset  down                  down
FastEthernet0/4        unassigned      YES unset  down                  down
FastEthernet0/5        unassigned      YES unset  down                  down
FastEthernet0/6        unassigned      YES unset  down                  down
FastEthernet0/7        unassigned      YES unset  down                  down
FastEthernet0/8        unassigned      YES unset  down                  down
FastEthernet0/9        unassigned      YES unset  down                  down
```

Output: Now we ping PC1 from PC0 to check the connectivity.



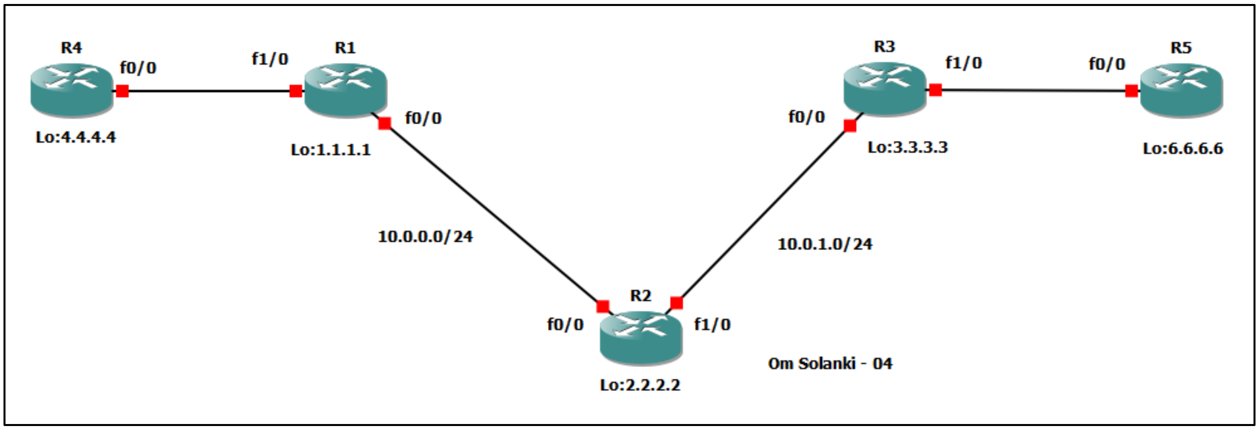
Similarly ping PC0 from PC1.



PRACTICAL 8

Aim: Simulating MPLS Environment.

Topology:



Step 1: IP addressing of MPLS Core and OSPF.

First bring 3 routers into your topology R1, R2, R3 position them as below. We are going to address the routers and configure ospf to ensure loopback to loopback connectivity between R1 and R3.

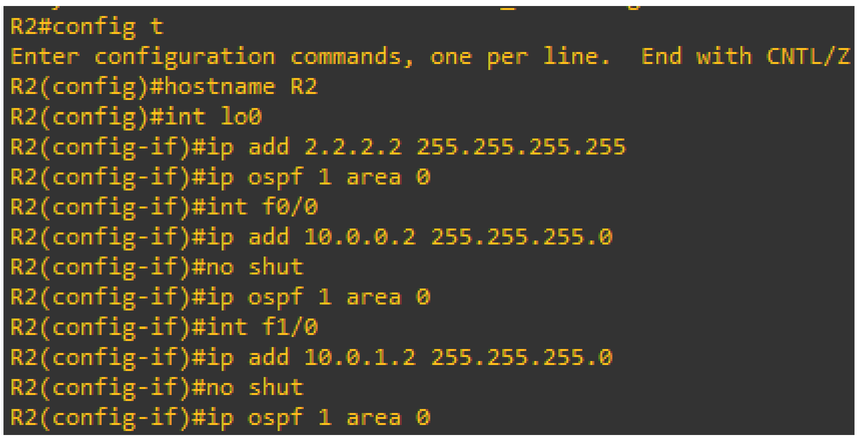
Router R1:

```
hostname R1
int lo0
ip add 1.1.1.1 255.255.255.255
ip ospf 1 area 0
int f0/0
ip add 10.0.0.1 255.255.255.0
no shut
ip ospf 1 area 0
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#int lo0
R1(config-if)#ip add 1.1.1.1 255.255.255.255
R1(config-if)#ip ospf 1 area 0
R1(config-if)#int f0/0
R1(config-if)#ip add 10.0.0.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#ip ospf 1 area 0
```

Router R2:

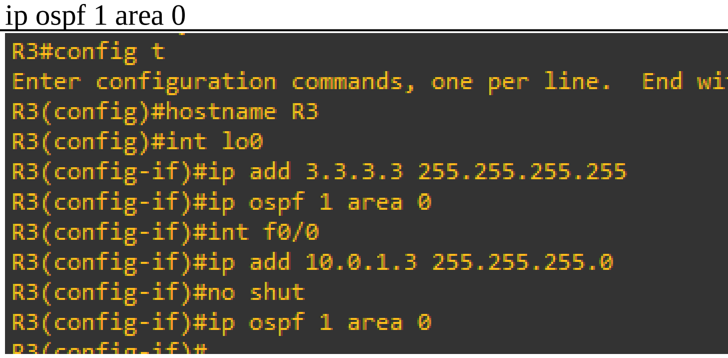
```
hostname R2
int lo0
ip add 2.2.2.2 255.255.255.255
ip ospf 1 area 0
int f0/0
ip add 10.0.0.2 255.255.255.0
no shut
ip ospf 1 area 0
int f0/1
ip add 10.0.1.2 255.255.255.0
no shut
ip ospf 1 area 0
```



```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z
R2(config)#hostname R2
R2(config)#int lo0
R2(config-if)#ip add 2.2.2.2 255.255.255.255
R2(config-if)#ip ospf 1 area 0
R2(config-if)#int f0/0
R2(config-if)#ip add 10.0.0.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#ip ospf 1 area 0
R2(config-if)#int f1/0
R2(config-if)#ip add 10.0.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#ip ospf 1 area 0
```

Router R3:

```
hostname R3
int lo0
ip add 3.3.3.3 255.255.255.255
ip ospf 1 are 0
int f0/0
ip add 10.0.1.3 255.255.255.0
no shut
ip ospf 1 area 0
```



```
R3#config t
Enter configuration commands, one per line.  End wi
R3(config)#hostname R3
R3(config)#int lo0
R3(config-if)#ip add 3.3.3.3 255.255.255.255
R3(config-if)#ip ospf 1 area 0
R3(config-if)#int f0/0
R3(config-if)#ip add 10.0.1.3 255.255.255.0
R3(config-if)#no shut
R3(config-if)#ip ospf 1 area 0
R3(config-if)#
```

You should now have full ip connectivity between R1, R2, R4 to verify this we need to see if we can ping between the loopbacks of R1 and R4.

```
R1#ping 3.3.3.3 source lo0
```

```
R1#ping 3.3.3.3 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/72/92 ms
```

You could show the routing table here, but the fact that you can ping between the loopbacks is verification enough and it is safe to move on.

<u>Step 2 – Configure LDP on all the interfaces in the MPLS Core.</u>

In order to run MPLS you need to enable it, there are two ways to do this.

• At each interface enter the mpls ip command.

• Under the ospf process use the mpls ldp autoconfig command

For this, we will be using the second option, so go into the ospf process and enter mpls ldp autoconfig – this will enable mpls label distribution protocol on every interface running ospf under that specific process.

R1

```
router ospf 1
mpls ldp autoconfig
```

```
R1#config t
Enter configuration commands, one per line.  End with CNT
R1(config)#router ospf 1
R1(config-router)#mpls ldp autoconfig
```

R2

```
router ospf 1
mpls ldp autoconfig
```

```
R2(config-if)#exit
R2(config)#router ospf 1
R2(config-router)#mpls ldp autoconfig
R2(config-router)#
*May 17 10:36:51.299: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
```

R3

```
router ospf 1
mpls ldp autoconfig
```

```
R3(config-if)#exit
R3(config)#router ospf 1
R3(config-router)#mpls ldp autoconfig
R3(config-router)#
*May 17 10:37:56.643: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is UP
```

You should see log messages coming up showing the LDP neighbors are up.

To verify the mpls interfaces the command is very simple – sh mpls interface This is done on R2 and you can see that both interfaces are running mpls and using LDP.

```
R2#sh mpls interface
```

```
R2(config-router)#end
R2#
*May 17 10:38:49.583: %SYS-5-CONFIG_I: Configured from console by console
R2#sh mpls interface
Interface              IP            Tunnel   Operational
FastEthernet0/0        Yes (ldp)     No       Yes
FastEthernet1/0        Yes (ldp)     No       Yes
```

You can also verify the LDP neighbors with the sh mpls ldp neighbors command.

```
R2#sh mpls ldp neigh
```

```
R2#sh mpls ldp neigh
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 1.1.1.1.646 - 2.2.2.2.11088
        State: Oper; Msgs sent/rcvd: 11/11; Downstream
        Up time: 00:02:49
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.0.0.1
        Addresses bound to peer LDP Ident:
          10.0.0.1        1.1.1.1
    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 3.3.3.3.53500 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 10/9; Downstream
        Up time: 00:01:44
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.0.1.3
        Addresses bound to peer LDP Ident:
          10.0.1.3        3.3.3.3
```

One more verification to confirm LDP is running ok is to do a trace between R1 and R3 and verify if you get MPLS Labels show up in the trace.

```
R1#trace 3.3.3.3
```

```
R1(config-router)#end
R1#
*May 17 10:40:40.635: %SYS-5-CONFIG_I: Configured from console by console
R1#trace 3.3.3.3

Type escape sequence to abort.
Tracing the route to 3.3.3.3

  1 10.0.0.2 [MPLS: Label 17 Exp 0] 92 msec 72 msec 92 msec
  2 10.0.1.3 72 msec 108 msec 88 msec
```

As you can see the trace to R2 used an MPLS Label in the path, as this is a very small MPLS core only one label was used as R3 was the final hop.

So to review we have now configured IP addresses on the MPLS core, enabled OSPF and full IP connectivity between all routers and finally enabled mpls on all the interfaces in the core and have established ldp neighbors between all routers.

The next step is to configure MP-BGP between R1 and R3.

This is when you start to see the layer 3 vpn configuration come to life.

Step 3 – MPLS BGP Configuration between R1 and R3.

We need to establish a MultiProtocol BGP session between R1 and R3 this is done by configuring the vpnv4 address family as below

Router R1:

```
router bgp 1
neighbor 3.3.3.3 remote-as 1
neighbor 3.3.3.3 update-source Loopback0
no auto-summary
address-family vpnv4
neighbor 3.3.3.3 activate
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router bgp 1
R1(config-router)#neighbor 3.3.3.3 remote-as 1
R1(config-router)#neighbor 3.3.3.3 update-source Loopback0
R1(config-router)#no auto-summary
R1(config-router)#address-family vpnv4
R1(config-router-af)#neighbor 3.3.3.3 activate
```

Router R3:

```
router bgp 1
neighbor 1.1.1.1 remote-as 1
neighbor 1.1.1.1 update-source Loopback0
```

```
no auto-summary
address-family vpnv4
neighbor 1.1.1.1 activate
```

```
R3(config-router)#end
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router bgp 1
R3(config-router)#neighbor 1.1.1.1 remote-as 1
R3(config-router)#neighbor 1.1.1.1 update-source Loopback0
R3(config-router)#no auto-summary
R3(config-router)#address-family vpnv4
R3(config-router-af)#neighbor 1.1.1.1 activate
```

You should see log messages showing the BGP sessions coming up.

To verify the BGP session between R1 and R3 issue the command sh bgp vpnv4 unicast all

Summary

```
R1#sh bgp vpnv4 unicast all summary
```

```
R1(config-router-af)#end
R1#sh bgp vpnv4 unicast all summary
*May 17 10:44:08.859: %SYS-5-CONFIG_I: Configured from console by console
R1#sh bgp vpnv4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4     1       3       3        1    0    0 00:00:48        0
```

You can see here that we do have a bgp vpnv4 peering to R3 – looking at the PfxRcd you can see it says 0 this is because we have not got any routes in BGP. We are now going to add two more routers to the topology. These will be the customer sites connected to R1 and R3. We will then create a VRF on each router and put the interfaces connected to each site router into that VRF.

Step 4 – Add two more routers, create VRFs We will add two more routers into the topology so it now looks like the final topology

Router 4 will peer OSPF using process number 2 to a VRF configured on R1. It will use the local site addressing of 192.168.1.0/24.

Router R4:

```
int lo0
ip add 4.4.4.4 255.255.255.255
ip ospf 2 area 2
int f0/0
ip add 192.168.1.4 255.255.255.0
ip ospf 2 area 2
no shut
```

```
R4#config t
Enter configuration commands, one per line.  End with CN
R4(config)#int lo0
R4(config-if)#ip add 4.4.4.4 255.255.255.255
R4(config-if)#ip ospf 2 area 2
R4(config-if)#int f0/0
R4(config-if)#ip add 192.168.1.4 255.255.255.0
R4(config-if)#ip ospf 2 area 2
R4(config-if)#no shut
```

Router R1:

```
int f0/1
no shut
ip add 192.168.1.1 255.255.255.0
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f1/0
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
```

Now at this point we have R4 peering to R1 but in the global routing table of R1 which is not what we want.

For this mpls tutorial I will be using VRF RED.

Router R1:

```
ip vrf RED
rd 4:4
route-target both 4:4
```

```
R1(config)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
```

The RD and route-target do not need to be the same.

So now we have configured the VRF on R1 we need to move the interface F0/1 into that VRF

Router R1:

```
int f0/1
 ip vrf forwarding RED
```

```
R1(config-vrf)#exit
R1(config)#int f1/0
R1(config-if)#ip vrf forwarding RED
% Interface FastEthernet1/0 IP address 192.168.1.1 removed due to enabling VRF RED
```

Now notice what happens when you do that – the IP address is removed.

You just need to re-apply it.

Router R1:

```
int f1/0

ip address 192.168.1.1 255.255.255.0
```

```
R1(config-if)#int f1/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
```

Now if we view the config on R1 int f0/1 you can see the VRF configured

```
R1#sh run int f0/1
```

```
R1(config-if)#end
R1#sh run int f1/0
*May 17 10:52:38.567: %SYS-5-CONFIG_I: Configured from
R1#sh run int f1/0
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet1/0
 ip vrf forwarding RED
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
end
```

Now we can start to look into VRF and how they operate – you need to understand now that

there are 2 routing tables within R1

• The Global Routing Table

• The Routing Table for VRF RED

If you issue the command sh ip route this shows the routes in the global table and you will notice that you do not see 192.168.1.0/24.

```
R1#sh ip route
```

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 10.0.0.2, 00:19:37, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/3] via 10.0.0.2, 00:19:37, FastEthernet0/0
     10.0.0.0/24 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
O       10.0.1.0 [110/2] via 10.0.0.2, 00:19:37, FastEthernet0/0
```

If you now issue the command sh ip route vrf red – this will show the routes in the routing table for VRF RED

```
R1#sh ip route vrf RED
```

NOTE: The VRF name is case sensitive!

```
R1#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet1/0
```

We just need to enable OSPF on this interface and get the loopback address for R4 in the VRF RED routing table before proceeding.

Router R1:

```
int f0/1

ip ospf 2 area 2
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f1/0
R1(config-if)#ip ospf 2 area 2
R1(config-if)#
*May 17 10:59:26.775: %OSPF-5-ADJCHG: Process 2, Nbr 4.4.4.4 on FastEthernet1/0 from LOADING to
ULL, Loading Done
```

You should see a log message showing the OSPF neighbor come up.

If we now check the routes in the VRF RED routing table you should see 4.4.4.4 in there as well.

```
R1#sh ip route vrf RED
```

```
R1#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/2] via 192.168.1.4, 00:01:51, FastEthernet1/0
C    192.168.1.0/24 is directly connected, FastEthernet1/0
```

We now need to repeat this process for R3 & R6.

Router 6 will peer OSPF using process number 2 to a VRF configured on R3. It will use the local site addressing of 192.168.2.0/24.

Router R6:

```
int lo0
ip add 6.6.6.6 255.255.255.255
ip ospf 2 area 2
int f0/0
ip add 192.168.2.6 255.255.255.0
ip ospf 2 area 2
no shut
```

```
R6(config)#int lo0
R6(config-if)#ip add 6.6.6.6 255.255.255.255
R6(config-if)#ip ospf 2 area 2
R6(config-if)#int f0/0
R6(config-if)#ip add 192.168.2.6 255.255.255.0
R6(config-if)#ip ospf 2 area 2
R6(config-if)#no shut
*May 17 11:04:02.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
 up
R6(config-if)#no shut
R6(config-if)#
*May 17 11:04:06.315: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R6(config-if)#
*May 17 11:04:06.315: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State D
own
*May 17 11:04:07.315: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed st
ate to up
```

Router R3:

```
int f0/1
no shut
ip add 192.168.2.3 255.255.255.0
```

```
R3(config)#int f1/0
R3(config-if)#ip add 192.168.2.3 255.255.255.0
R3(config-if)#no shut
```

We also need to configure a VRF onto R3 as well.

Router R3:

```
ip vrf RED
rd 4:4
route-target both 4:4
```

```
R3(config-if)#exit
R3(config)#ip vrf RED
R3(config-vrf)#rd 4:4
R3(config-vrf)#route-target both 4:4
```

So now we have configured the VRF on R3 we need to move the interface F0/1 into that VRF

Router R3:

```
int f0/1
ip vrf forwarding RED
```

```
R3(config-vrf)#exit
R3(config)#int f1/0
R3(config-if)#ip vrf forwarding RED
% Interface FastEthernet1/0 IP address 192.168.2.3 removed due to enabling VRF RED
```

Now notice what happens when you do that – the IP address is removed.

You just need to re-apply it.

Router R3:

int f1/0

ip address 192.168.2.1 255.255.255.0

```
R3(config-if)#exit
R3(config)#int f1/0
R3(config-if)#ip address 192.168.2.1 255.255.255.0
```

Now if we view the config on R3 int f0/1 you can see the VRF configured.

Router R3:

R3#sh run int f0/1

```
R3(config-if)#end
R3#sh run int f1/0
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet1/0
 ip vrf forwarding RED
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
end
```

Finally we just need to enable OSPF on that interface and verify the routes are in the RED routing table.
Router R3:

int f0/1

ip ospf 2 area 2

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int f1/0
R3(config-if)#ip ospf 2 area 2
```

Check the routes in vrf RED

R3#sh ip route vrf RED

```
R3(config-if)#end
R3#sh ip route vrf RED
*May 17 11:15:22.367: %SYS-5-CONFIG_I: Configured from console by console
R3#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     6.0.0.0/32 is subnetted, 1 subnets
O       6.6.6.6 [110/2] via 192.168.2.6, 00:00:15, FastEthernet1/0
C    192.168.2.0/24 is directly connected, FastEthernet1/0
```

The final step to get full connectivity across the MPLS core is to redistribute the routes in OSPF on R1 and R3 into MP-BGP and MP-BGP into OSPF, this is what we are going to do now.

We need to redistribute the OSPF routes from R4 into BGP in the VRF on R1, the OSPF routes from R6 into MP-BGP in the VRF on R3 and then the routes in MP-BGP in R1 and R3 back out to OSPF.

Check the routes on R4

```
R4#sh ip route
```

```
R4(config-if)#end
R4#sh ip route
*May 17 11:17:31.695: %SYS-5-CONFIG_I: Configured from console by console
R4#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

As expected we have the local interface and the loopback address.

When we are done we want to see 6.6.6.6 in there so we can ping across the MPLS

Check the routes on R1

R1# sh ip route

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 10.0.0.2, 00:44:12, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/3] via 10.0.0.2, 00:44:12, FastEthernet0/0
     10.0.0.0/24 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
O       10.0.1.0 [110/2] via 10.0.0.2, 00:44:12, FastEthernet0/0
```

Remember we have a VRF configured on this router so this command will show routes in the global routing table (the MPLS Core) and it will not show the 192.168.1.0/24 route as that is in VRF RED – to see that we run the following command

R1#sh ip route vrf RED

```
R1#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/2] via 192.168.1.4, 00:19:34, FastEthernet1/0
C    192.168.1.0/24 is directly connected, FastEthernet1/0
```

Here you can see the Routing Table: RED is shown and the routes to R4 are now visible with 4.4.4.4 being in OSPF.

So we need to do the following;

• Redistribute OSPF into MP-BGP on R1
• Redistribute MP-BGP into OSPF on R1
• Redistribute OSPF into MP-BGP on R3
• Redistribute MP-BGP into OSPF on R3

Redistribute OSPF into MP-BGP on R1

Router R1:

```
router bgp 1
address-family ipv4 vrf RED
redistribute ospf 2
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router bgp 1
R1(config-router)#address-family ipv4 vrf RED
R1(config-router-af)#redistribute ospf 2
```

Redistribute OSPF into MP-BGP on R3

Router R3:

```
router bgp 1
address-family ipv4 vrf RED
redistribute ospf 2
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router bgp 1
R3(config-router)#address-family ipv4 vrf RED
R3(config-router-af)#redistribute ospf 2
```

This has enabled redistribution of the OSPF routes into BGP. We can check the routes from R4 and R6 are now showing in the BGP table for their VRF with this command

```
R1#sh ip bgp vpnv4 vrf RED
```

```
R1(config-router-af)#end
R1#sh ip bgp vpnv4 vrf RED
*May 17 11:21:48.471: %SYS-5-CONFIG_I: Configured from console by console
R1#sh ip bgp vpnv4 vrf RED
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
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
*> 4.4.4.4/32       192.168.1.4              2         32768 ?
*>i6.6.6.6/32       3.3.3.3                  2    100      0 ?
*> 192.168.1.0      0.0.0.0                  0         32768 ?
*>i192.168.2.0      3.3.3.3                  0    100      0 ?
```

Here we can see that 4.4.4.4 is now in the BGP table in VRF RED on R1 with a next hop of 192.168.1.4 (R4) and also 6.6.6.6 is in there as well with a next hop of 3.3.3.3 (which is the loopback of R3 – showing that it is going over the MPLS and R1 is not in the picture) The same should be true on R3

```
R3#sh ip bgp vpnv4 vrf RED
```

```
R3(config-router-af)#end
R3#sh ip bgp vpnv4 vrf RED
BGP table version is 9, 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
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
*>i4.4.4.4/32       1.1.1.1                  2    100      0 ?
*> 6.6.6.6/32       192.168.2.6              2         32768 ?
*>i192.168.1.0      1.1.1.1                  0    100      0 ?
*> 192.168.2.0      0.0.0.0                  0         32768 ?
R3#
*May 17 11:22:51.851: %SYS-5-CONFIG_I: Configured from console by console
```

6.6.6.6 is now in the BGP table in VRF RED on R3 with a next hop of 192.168.2.6 (R6) and also 4.4.4 is in there as well with a next hop of 1.1.1.1 (which is the loopback of R1 – showing that it is going over the MPLS and R2 is not in the picture)

The final step is to get the routes that have come across the MPLS back into OSPF and then we can get end to end connectivity

Router R1:

```
router ospf 2
redistribute bgp 1 subnets
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 2
R1(config-router)#redistribute bgp 1 subnets
```

Router R3:

```
router ospf 2
redistribute bgp 1 subnets
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 2
R3(config-router)#redistribute bgp 1 subnets
```

If all has worked we should be now able to ping 6.6.6.6 from R4.

Before we do let’s see what the routing table looks like on R4

```
R4#sh ip route
```

```
R4#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
     6.0.0.0/32 is subnetted, 1 subnets
O IA    6.6.6.6 [110/3] via 192.168.1.1, 00:01:36, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O IA 192.168.2.0/24 [110/2] via 192.168.1.1, 00:01:36, FastEthernet0/0
```

Also check the routing table on R6

R6#sh ip route

```
R6#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/3] via 192.168.2.1, 00:01:49, FastEthernet0/0
     6.0.0.0/32 is subnetted, 1 subnets
C       6.6.6.6 is directly connected, Loopback0
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 00:01:49, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

R4#ping 6.6.6.6

```
R4#ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 132/150/168 ms
```

Which we can – to prove this is going over the MPLS and be label switched and not routed, let's do a trace

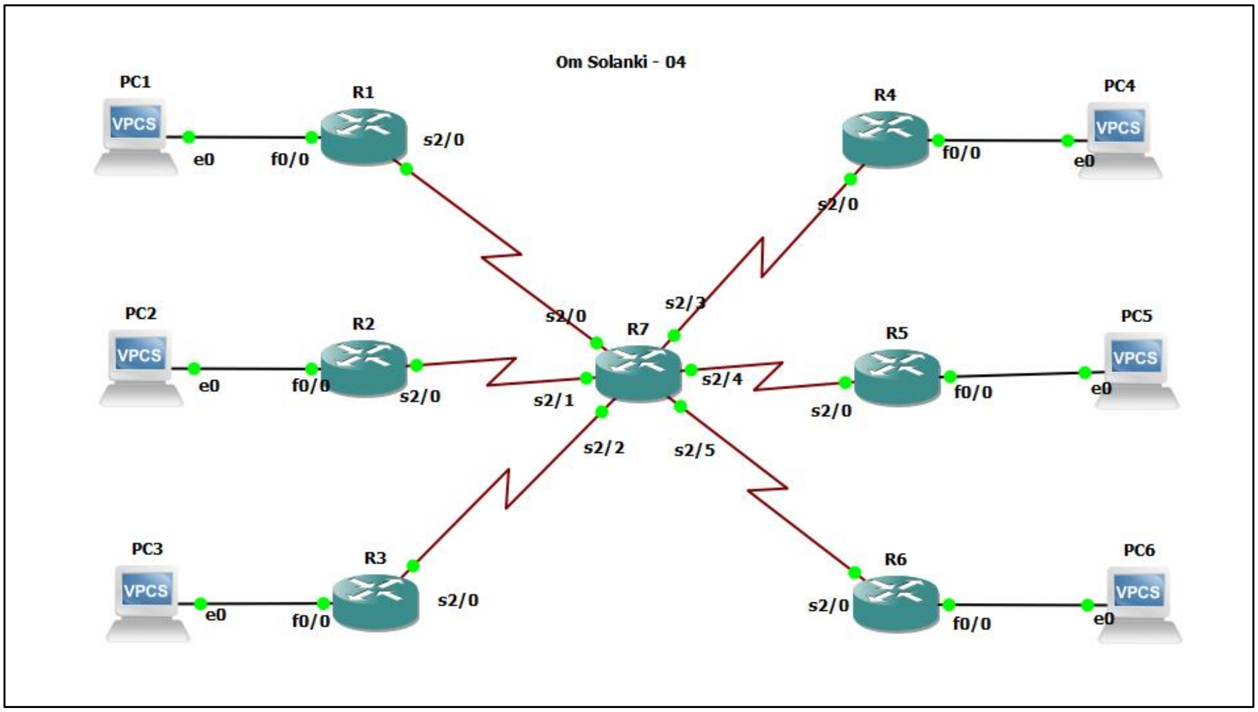
R4#trace 6.6.6.6

```
R4#trace 6.6.6.6

Type escape sequence to abort.
Tracing the route to 6.6.6.6

  1 192.168.1.1 16 msec 28 msec 28 msec
  2 10.0.0.2 [MPLS: Labels 16/19 Exp 0] 108 msec 120 msec 96 msec
  3 192.168.2.1 [MPLS: Label 19 Exp 0] 124 msec 92 msec 80 msec
  4 192.168.2.6 112 msec 132 msec 100 msec
```

PRACTICAL NO: 9

Aim: Configure VRF-lite on routers.

Topology:



In order to demonstrate how VRF-lite works let's consider the following scenario using the topology shown below. We have 3 different customers each one having 2 sites connected through the ISP router. Since each customer has sensitive information they will be placed in a separate VRF and the connectivity will be provided through the ISP router. For customers 2 and 3 we will use the same address space to demonstrate that overlapping IPs does not cause any conflicts. One thing to keep in mind is that interfaces assigned to VRFs can be physical or logical and that an interface cannot belong to more than one VRF at any time. The IP addressing scheme for each customer router and each PC will be configured according to the topology presented above. For the connection between the ISP router and each customer we will use a /30 subnet with the first IP from the subnet being allocated to the ISP router interface and the second IP to the customer router interface. We will use OSPF as the routing protocol between the ISP and each customer router. First we'll go on the ISP router and globally define the VRF for each customer. In order to create the VRFs we need to use the ip vrf command in global configuration mode.

PC1> ip 172.20.1.100 255.255.255.0 gateway 172.20.1.1
PC2> ip 172.20.2.100 255.255.255.0 gateway 172.20.2.1
PC3> ip 172.20.1.200 255.255.255.0 gateway 172.20.1.1
PC4> ip 172.20.4.100 255.255.255.0 gateway 172.20.4.1
PC5> ip 172.20.5.100 255.255.255.0 gateway 172.20.5.1
PC6> ip 172.20.4.200 255.255.255.0 gateway 172.20.4.1

```
PC1> ip 172.20.1.100 255.255.255.0 gateway 172.20.1.1
Checking for duplicate address...
PC1 : 172.20.1.100 255.255.255.0 gateway 172.20.1.1
```

```
PC2> ip 172.20.2.100 255.255.255.0 gateway 172.20.2.1
Checking for duplicate address...
PC1 : 172.20.2.100 255.255.255.0 gateway 172.20.2.1
```

```
PC3> ip 172.20.1.200 255.255.255.0 gateway 172.20.1.1
Checking for duplicate address...
PC1 : 172.20.1.200 255.255.255.0 gateway 172.20.1.1
```

```
PC4> ip 172.20.4.100 255.255.255.0 gateway 172.20.4.1
Checking for duplicate address...
PC1 : 172.20.4.100 255.255.255.0 gateway 172.20.4.1
```

```
PC5> ip 172.20.5.100 255.255.255.0 gateway 172.20.5.1
Checking for duplicate address...
PC1 : 172.20.5.100 255.255.255.0 gateway 172.20.5.1
```

```
PC6> ip 172.20.4.200 255.255.255.0 gateway 172.20.4.1
Checking for duplicate address...
PC1 : 172.20.4.200 255.255.255.0 gateway 172.20.4.1
```

R1:

```
config t
int fa0/0
ip address 172.20.1.1 255.255.255.0
no shutdown
exit
int S2/0
ip address 10.10.10.2 255.255.255.252
no shutdown
```

```
R1#en
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int fa0/0
R1(config-if)#ip address 172.20.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
*May 17 11:16:17.583: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
*May 17 11:16:17.583: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State Dow
*May 17 11:16:18.583: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed stat
R1(config-if)#exit
R1(config)#int S2/0
R1(config-if)#ip address 10.10.10.2 255.255.255.252
R1(config-if)#no shutdown
```

R2:

```
config t
int fa0/0
ip address 172.20.2.1 255.255.255.0
no shutdown
exit
int S2/0
ip address 10.10.20.2 255.255.255.252
no shutdown
```

```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int fa0/0
R2(config-if)#ip address 172.20.2.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
*May 17 11:16:47.819: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R2(config)#
*May 17 11:16:47.819: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State Dow
*May 17 11:16:48.819: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed stat
R2(config)#int S2/0
R2(config-if)#ip address 10.10.20.2 255.255.255.252
R2(config-if)#no shutdown
```

R3:

```
config t
int fa0/0
ip address 172.20.1.1 255.255.255.0
no shutdown
exit
int S2/0
ip address 10.10.30.2 255.255.255.252
no shutdown
```

```
R3#en
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int fa0/0
R3(config-if)#ip address 172.20.1.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#int S2/0
R3(config-if)#ip address 10.10.30.2 255.255.255.252
R3(config-if)#no shutdown
*May 17 11:17:29.307: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R3(config-if)#no shutdown
*May 17 11:17:29.307: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State
*May 17 11:17:30.307: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
R3(config-if)#no shutdown
```

R4:

```
config t
int fa0/0
ip address 172.20.4.1 255.255.255.0
no shutdown
exit
int S2/0
ip address 10.10.40.2 255.255.255.252
no shutdown
```

```
R4#en
R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#int fa0/0
R4(config-if)#ip address 172.20.4.1 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#
*May 17 11:17:50.355: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R4(config)#
*May 17 11:17:50.355: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State D
*May 17 11:17:51.355: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed st
R4(config)#int S2/0
R4(config-if)#ip address 10.10.40.2 255.255.255.252
R4(config-if)#no shutdown
```

R5:

```
config t
int fa0/0
ip address 172.20.5.1 255.255.255.0
no shutdown
exit
int S2/0
ip address 10.10.50.2 255.255.255.252
no shutdown
```

```
R5#en
R5#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#int fa0/0
R5(config-if)#ip address 172.20.5.1 255.255.255.0
R5(config-if)#no shutdown
R5(config-if)#
*May 17 11:18:21.823: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R5(config-if)#
*May 17 11:18:21.823: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State
*May 17 11:18:22.823: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
R5(config-if)#exit
R5(config)#int S2/0
R5(config-if)#ip address 10.10.50.2 255.255.255.252
R5(config-if)#no shutdown
```

R6:

```
config t
int fa0/0
ip address 172.20.4.1 255.255.255.0
no shutdown
exit
int S2/0
ip address 10.10.60.2 255.255.255.252
no shutdown
```

```
R6#en
R6#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R6(config)#int fa0/0
R6(config-if)#ip address 172.20.4.1 255.255.255.0
R6(config-if)#no shutdown
R6(config-if)#exit
R6(config)#
*May 17 11:19:07.639: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R6(config)#
*May 17 11:19:07.639: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State D
*May 17 11:19:08.639: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed st
R6(config)#int S2/0
R6(config-if)#ip address 10.10.60.2 255.255.255.252
R6(config-if)#no shutdown
```

R7(ISP):

```
config t
int S2/0
ip address 10.10.10.1 255.255.255.252
exit
int S2/1
ip address 10.10.20.1 255.255.255.252
exit
int S2/2
```

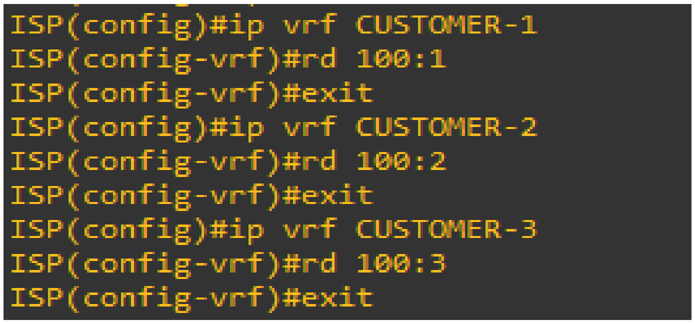
```
ip address 10.10.30.1 255.255.255.252
exit
int S2/3
ip address 10.10.40.1 255.255.255.252
exit
int S2/4
ip address 10.10.50.1 255.255.255.252
exit
int S2/5
ip address 10.10.60.1 255.255.255.252
exit
```

```
R7#confi t
Enter configuration commands, one per line.  End with CNTL/Z.
R7(config)#hostname ISP
ISP(config)#int S2/0
ISP(config-if)#ip address 10.10.10.1 255.255.255.252
ISP(config-if)#exit
ISP(config)#int S2/1
ISP(config-if)#ip address 10.10.20.1 255.255.255.252
ISP(config-if)#exit
ISP(config)#int S2/2
ISP(config-if)#ip address 10.10.30.1 255.255.255.252
ISP(config-if)#exit
ISP(config)#int S2/3
ISP(config-if)#ip address 10.10.40.1 255.255.255.252
ISP(config-if)#exit
ISP(config)#int S2/4
ISP(config-if)#ip address 10.10.50.1 255.255.255.252
ISP(config-if)#exit
ISP(config)#int S2/5
ISP(config-if)#ip address 10.10.60.1 255.255.255.252
ISP(config-if)#exit
```

Since CUSTOMER-1 and CUSTOMER-3 use the same address space (172.20.1.0/24 and 172.20.4.0/24) for their sites we need a way to identify which route belongs to which customer. A solution to this problem is to use route distinguishers (RD). A route distinguisher is a unique number of 64 bits in length which is prepended to each route within a VRF and helps us to identify to which VRF this route belongs. The RD value is usually composed from an AS number followed by a semicolon and an arbitrary number. In this scenario for the RD value I'll use 100 as the AS number and the site ID of each customer. Below is the VRF and RD configuration for each customer.

```
ip vrf CUSTOMER-1
rd 100:1
exit
ip vrf CUSTOMER-2
rd 100:2
exit
```

```
ip vrf CUSTOMER-3
rd 100:3
exit
```



```
ISP(config)#ip vrf CUSTOMER-1
ISP(config-vrf)#rd 100:1
ISP(config-vrf)#exit
ISP(config)#ip vrf CUSTOMER-2
ISP(config-vrf)#rd 100:2
ISP(config-vrf)#exit
ISP(config)#ip vrf CUSTOMER-3
ISP(config-vrf)#rd 100:3
ISP(config-vrf)#exit
```

The next step is to associate each VRF instance with at least one interface. For this we need to go into interface configuration mode and use the ip vrf forwarding command.

According to the diagram we'll associate interfaces Gi0/0 and Gi0/3 to CUSTOMER-1 vrf, interfaces Gi0/1 and Gi0/4 to CUSTOMER-2 vrf and interfaces Gi0/2 and Gi0/5 to CUSTOMER-3 vrf. Below is the configuration for each interface.

```
int s2/0
ip vrf forwarding CUSTOMER-1
int s2/1
ip vrf forwarding CUSTOMER-2
int s2/2
ip vrf forwarding CUSTOMER-3
int s2/3
ip vrf forwarding CUSTOMER-1
int s2/4
ip vrf forwarding CUSTOMER-2
int s2/5
ip vrf forwarding CUSTOMER-3
end
```

```
ISP(config)#int s2/0
ISP(config-if)#ip vrf forwarding CUSTOMER-1
ISP(config-if)#int s2/1
ISP(config-if)#ip vrf forwarding CUSTOMER-2
ISP(config-if)#int s2/2
ISP(config-if)#ip vrf forwarding CUSTOMER-3
ISP(config-if)#int s2/3
ISP(config-if)#ip vrf forwarding CUSTOMER-1
ISP(config-if)#int s2/4
ISP(config-if)#ip vrf forwarding CUSTOMER-2
ISP(config-if)#int s2/5
ISP(config-if)#ip vrf forwarding CUSTOMER-3
ISP(config-if)#end
```

Note: It is recommended to apply this command on the interface before configuring an IP address otherwise the IP address will be removed and you will need to configure it again. A message similar to the one below will be displayed on the router console.

If Ip address is removed then configure all ip addresses again.
In order to display the created VRFs and the associated interfaces we can use the show ip vrf command in EXEC mode.

```
show ip vrf
```

```
ISP#show ip vrf
  Name                             Default RD          Interfaces
  CUSTOMER-1                       100:1               Se2/0
                                                       Se2/3
  CUSTOMER-2                       100:2               Se2/1
                                                       Se2/4
  CUSTOMER-3                       100:3               Se2/2
                                                       Se2/5
```

Configuring OSPF routing

In order to keep things simple we'll enable OSPF on all interfaces for each customer router by running the following command.

R1:

```
exit
router ospf 1
network 0.0.0.0 255.255.255.255 area 0
```

```
R1(config-if)#exit
R1(config)#router ospf 1
R1(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

R2

```
exit
router ospf 2
network 0.0.0.0 255.255.255.255 area 0
```

```
R2(config-if)#exit
R2(config)#router ospf 2
R2(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

R3

```
exit
router ospf 3
network 0.0.0.0 255.255.255.255 area 0
```

```
R3(config-if)#exit
R3(config)#router ospf 3
R3(config-router)#network 0.0.0.0 255.255.255.255 area 0
*May 17 11:23:22.063: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
R3(config-router)#network 0.0.0.0 255.255.255.255 area 0
*May 17 11:23:22.063: %ENTITY_ALARM-6-INFO: CLEAR INFO Se2/0 Physical Port Administrative State Down
R3(config-router)#network 0.0.0.0 255.255.255.255 area 0
*May 17 11:23:23.067: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
R3(config-router)#network 0.0.0.0 255.255.255.255 area 0
*May 17 11:23:52.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down
R3(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

R4

```
exit
router ospf 1
network 0.0.0.0 255.255.255.255 area 0
```

```
R4(config-if)#exit
R4(config)#router ospf 1
R4(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

R5

```
exit
router ospf 2
network 0.0.0.0 255.255.255.255 area 0
```

```
R5(config-router)#router ospf 2
R5(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

R6

```
exit
router ospf 2
network 0.0.0.0 255.255.255.255 area 0
```

```
R6(config-if)#exit
R6(config)#router ospf 3
R6(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

R7

```
config t
router ospf 1 vrf CUSTOMER-1
network 0.0.0.0 255.255.255.255 area 0
router ospf 2 vrf CUSTOMER-2
network 0.0.0.0 255.255.255.255 area 0
router ospf 3 vrf CUSTOMER-3
network 0.0.0.0 255.255.255.255 area 0
end
```

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router ospf 1 vrf CUSTOMER-1
ISP(config-router)#network 0.0.0.0 255.255.255.255 area 0
ISP(config-router)#router ospf 2 vrf CUSTOMER-2
ISP(config-router)#network 0.0.0.0 255.255.255.255 area 0
ISP(config-router)#router ospf 3 vrf CUSTOMER-3
ISP(config-router)#network 0.0.0.0 255.255.255.255 area 0
ISP(config-router)#end
```

At this time the OSPF neighborship should be established. We can verify this by running the following command.

```
ISP# show ip ospf neighbor
```

```
ISP#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.60.2        0   FULL/  -        00:00:31    10.10.60.2      Serial2/2
10.10.30.2        0   FULL/  -        00:00:31    10.10.30.2      Serial1/2
10.10.50.2        0   FULL/  -        00:00:38    10.10.50.2      Serial2/1
10.10.20.2        0   FULL/  -        00:00:37    10.10.20.2      Serial1/1
10.10.40.2        0   FULL/  -        00:00:35    10.10.40.2      Serial2/0
10.10.10.2        0   FULL/  -        00:00:36    10.10.10.2      Serial1/0
```

VRF-lite verification

Now let's check the routing tables and run some connectivity tests.

First let's check the global routing table.

```
show ip route
```

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

Gateway of last resort is not set
```

show ip route vrf CUSTOMER-1

```
ISP#show ip route vrf CUSTOMER-1

Routing Table: CUSTOMER-1
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

From the traceroute output we can see the destination IP address is located on PC-4 since the third hop in the path (10.10.40.2) is the Gi0/0 interface on CUST1-R2 router.

Now let’s try to ping PC-5 from PC-1.

PC-1> ping 172.20.5.100

```
PC1> ping 172.20.4.100
*172.20.1.1 icmp_seq=1 ttl=255 time=15.193 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=2 ttl=255 time=30.133 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=3 ttl=255 time=14.985 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=4 ttl=255 time=29.993 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=5 ttl=255 time=15.208 ms (ICMP type:3, code:1, Destination host unreachab
le)
```

trace 172.20.4.100

```
PC1> trace 172.20.4.100
trace to 172.20.4.100, 8 hops max, press Ctrl+C to stop
 1   172.20.1.1   30.084 ms  15.178 ms  15.257 ms
 2   *172.20.1.1   15.267 ms (ICMP type:3, code:1, Destination host unreachable)
```

From the traceroute output we can see the destination IP address is located on PC-4 since the third hop in the path (10.10.40.2) is the Gi0/0 interface on CUST1-R2 router. Now let's try to ping PC-5 from PC-1.

```
ping 172.20.5.100
```

```
PC1> ping 172.20.5.100
*172.20.1.1 icmp_seq=1 ttl=255 time=14.956 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=2 ttl=255 time=15.117 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=3 ttl=255 time=15.215 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=4 ttl=255 time=30.032 ms (ICMP type:3, code:1, Destination host unreachab
le)
*172.20.1.1 icmp_seq=5 ttl=255 time=15.240 ms (ICMP type:3, code:1, Destination host unreachab
le)
```

PC3

Since host PC-5 is located in CUSTOMER-2 VRF is not reachable from PC-1. Router CUST1 R1 since it doesn't have a route to network 172.20.5.0/24 and a default gateway it drops the packets. As a last test let's run a traceroute from PC-3 to PC-6 (172.20.4.200).

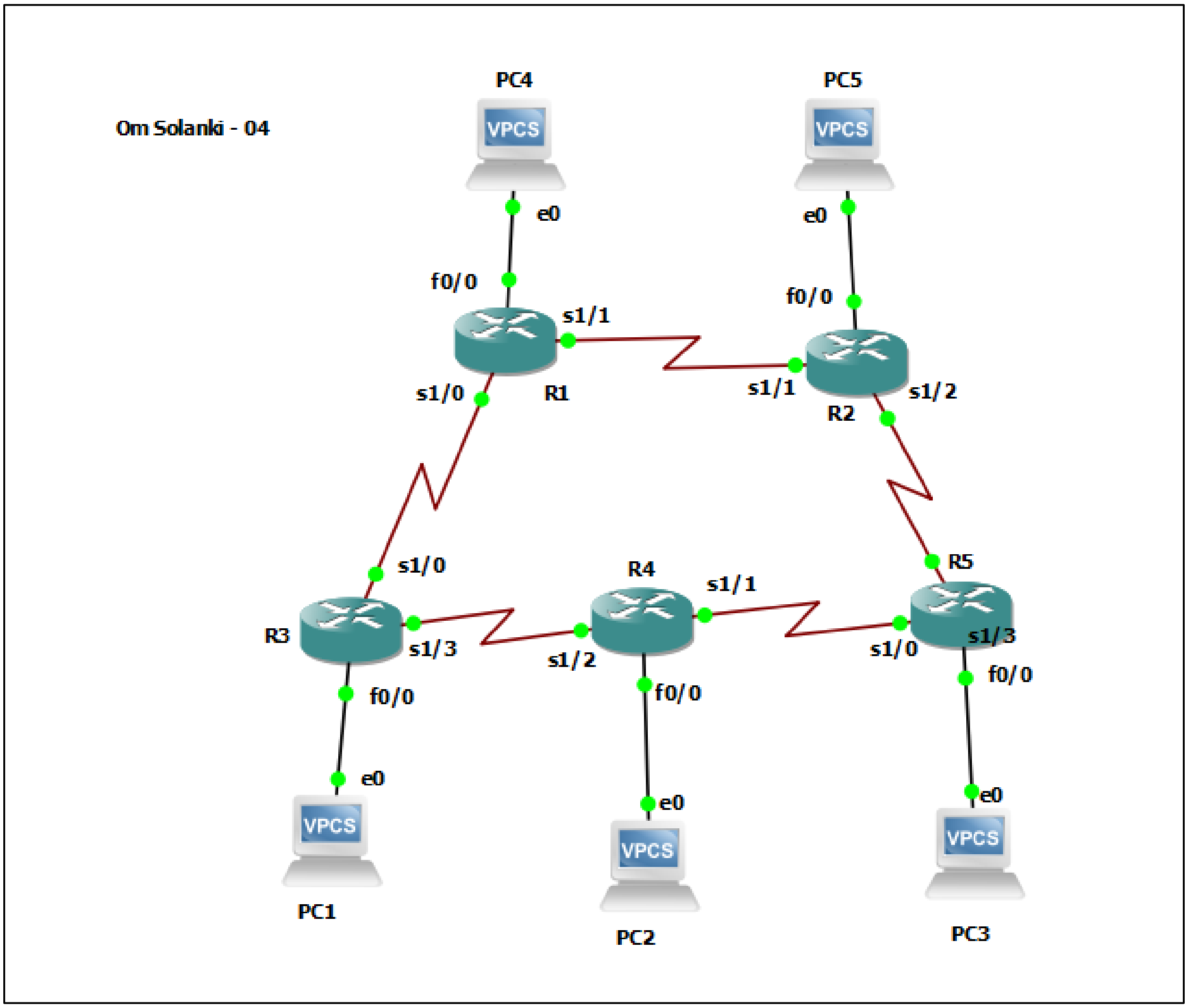
```
trace 172.20.4.200
```

```
PC3> trace 172.20.4.200
trace to 172.20.4.200, 8 hops max, press Ctrl+C to stop
 1   172.20.1.1   15.141 ms  29.916 ms  29.952 ms
 2   *172.20.1.1   14.918 ms (ICMP type:3, code:1, Destination host unreachable)
```

PRACTICAL NO: 10

Aim: Simulating Static Routing.

Topology



Step 1: Configuring interface and Assigning IP Address

R1

```
config t
int s1/0
ip add 20.0.0.1 255.0.0.0
no shut
exit
```

```
R1(config)#int s1/0
R1(config-if)#ip add 20.0.0.1 255.0.0.0
R1(config-if)#no shut
R1(config-if)#exit
```

```
int s1/1
ip add 10.0.0.2 255.0.0.0
no shut
exit
```

```
R1(config)#int s1/1
R1(config-if)#ip add 10.0.0.2 255.0.0.0
R1(config-if)#no shut
R1(config-if)#exit
```

```
int f0/0
ip add 60.0.0.1 255.0.0.0
no shut
end
```

```
R1(config)#int f0/0
R1(config-if)#ip add 60.0.0.1 255.0.0.0
R1(config-if)#no shut
R1(config-if)#end
R1#
*May 17 10:25:04.743: %SYS-5-CONFIG_I: Configured from console by console
```

```
show ip interface brief | ex un
```

```
R1#show ip interface brief | ex un
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            60.0.0.1        YES NVRAM  up                    up
Serial1/0                  20.0.0.1        YES NVRAM  up                    up
Serial1/1                  10.0.0.2        YES NVRAM  up                    up
R1#
```

R2

```
config t
int s1/1
ip add 20.0.0.2 255.0.0.0
no shut
exit
```

```
R2(config-if)#int s1/1
R2(config-if)#ip add 20.0.0.2 255.0.0.0
R2(config-if)#no shut
```

```
int s1/2
ip add 30.0.0.1 255.0.0.0
no shut
exit
```

```
R2(config)#int s1/2
R2(config-if)#ip add 30.0.0.1 255.0.0.0
R2(config-if)#no shut
R2(config-if)#exit
```

```
int f0/0
ip add 70.0.0.1 255.0.0.0
no shut
end
```

```
R2(config)#int f0/0
R2(config-if)#ip add 70.0.0.1 255.0.0.0
R2(config-if)#no shut
R2(config-if)#end
R2#
*May 17 10:26:54.359: %SYS-5-CONFIG_I: Configured from console by console
```

```
show ip interface brief | ex un
```

```
R2#show ip interface brief | ex un
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            70.0.0.1        YES NVRAM  up                    up
Serial1/1                  20.0.0.2        YES NVRAM  up                    up
Serial1/2                  30.0.0.1        YES NVRAM  up                    up
R2#
```

R3

```
config t
int s1/0
ip add 10.0.0.1 255.0.0.0
no shut
exit
```

```
R3(config)#int s1/0
R3(config-if)#ip add 10.0.0.1 255.0.0.0
R3(config-if)#no shut
R3(config-if)#exit
```

```
int s1/3
ip add 50.0.0.2 255.0.0.0
no shut
exit
```

```
R3(config)#int s1/3
R3(config-if)#ip add 50.0.0.2 255.0.0.0
R3(config-if)#no shut
R3(config-if)#exit
```

```
int f0/0
ip add 80.0.0.1 255.0.0.0
no shut
end
```

```
R3(config)#int f0/0
R3(config-if)#ip add 80.0.0.1 255.0.0.0
R3(config-if)#no shut
R3(config-if)#end
R3#
*May 17 10:28:07.579: %SYS-5-CONFIG_I: Configured from console by console
```

```
show ip interface brief | ex un
```

```
R3#show ip interface brief | ex un
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            80.0.0.1        YES NVRAM  up                    up
Serial1/0                  10.0.0.1        YES NVRAM  up                    up
Serial1/3                  50.0.0.2        YES NVRAM  up                    up
R3#
```

R4

```
config t
int s2/1
ip add 40.0.0.2 255.0.0.0
no shut
exit
```

```
R4(config)#int s1/1
R4(config-if)#ip add 40.0.0.2 255.0.0.0
R4(config-if)#no shut
R4(config-if)#exit
```

```
int s2/2
ip add 50.0.0.1 255.0.0.0
no shut
exit
```

```
R4(config)#int s1/2
R4(config-if)#ip add 50.0.0.1 255.0.0.0
R4(config-if)#no shut
R4(config-if)#exit
```

```
int f0/0
ip add 90.0.0.1 255.0.0.0
no shut
end
```

```
R4(config)#int f0/0
R4(config-if)#ip add 90.0.0.1 255.0.0.0
R4(config-if)#no shut
R4(config-if)#end
R4#
*May 17 10:28:46.619: %SYS-5-CONFIG I: Configured from console by console
```

```
show ip interface brief | ex un
```

```
R4#show ip interface brief | ex un
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            90.0.0.1        YES NVRAM  up                    up
Serial1/1                  40.0.0.2        YES NVRAM  up                    up
Serial1/2                  50.0.0.1        YES NVRAM  up                    up
R4#S
```

R5

```
config t
int s2/3
ip add 30.0.0.2 255.0.0.0
no shut
exit
```

```
R5(config)#int s1/3
R5(config-if)#ip add 30.0.0.2 255.0.0.0
R5(config-if)#no shut
R5(config-if)#exit
```

```
int s2/0
ip add 40.0.0.1 255.0.0.0
no shut
exit
```

```
R5(config)#int s1/0
R5(config-if)#ip add 40.0.0.1 255.0.0.0
R5(config-if)#no shut
R5(config-if)#exit
```

```
int f0/0
ip add 100.0.0.1 255.0.0.0
no shut
end
```

```
R5(config-if)#ip add 100.0.0.1 255.0.0.0
R5(config-if)#no shut
R5(config-if)#end
R5#
*May 17 10:29:17.571: %SYS-5-CONFIG_I: Configured from console by console
```

```
show ip interface brief | ex un
```

```
R5#show ip interface brief | ex un
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            100.0.0.1       YES NVRAM  up                    up
Serial1/0                  40.0.0.1        YES NVRAM  up                    up
Serial1/3                  30.0.0.2        YES NVRAM  up                    up
R5#
```

Configuring gateway for PC1, 2, 3, 4, 5

PC1

```
ip 80.0.0.2 255.0.0.0 gateway 80.0.0.1
```

```
PC1> ip 80.0.0.2 255.0.0.0 gateway 80.0.0.1
Checking for duplicate address...
PC1 : 80.0.0.2 255.0.0.0 gateway 80.0.0.1

PC1>
```

PC2

```
ip 90.0.0.2 255.0.0.0 gateway 90.0.0.1
```

```
PC2> ip 90.0.0.2 255.0.0.0 gateway 90.0.0.1
Checking for duplicate address...
PC1 : 90.0.0.2 255.0.0.0 gateway 90.0.0.1
```

PC3

```
ip 100.0.0.2 255.0.0.0 gateway 100.0.0.1
```

```
PC3> ip 100.0.0.2 255.0.0.0 gateway 100.0.0.1
Checking for duplicate address...
PC1 : 100.0.0.2 255.0.0.0 gateway 100.0.0.1
```

PC4

```
ip 60.0.0.2 255.0.0.0 gateway 60.0.0.1
```

```
PC4> ip 60.0.0.2 255.0.0.0 gateway 60.0.0.1
Checking for duplicate address...
PC1 : 60.0.0.2 255.0.0.0 gateway 60.0.0.1
```

PC5

```
ip 70.0.0.2 255.0.0.0 gateway 70.0.0.1
```

```
PC5> ip 70.0.0.2 255.0.0.0 gateway 70.0.0.1
Checking for duplicate address...
PC1 : 70.0.0.2 255.0.0.0 gateway 70.0.0.1
```

Step 2: Configure Static routing in Router 1

```
config t
ip forward-protocol nd
ip route 30.0.0.0 255.0.0.0 20.0.0.2
ip route 50.0.0.0 255.0.0.0 10.0.0.1
ip route 80.0.0.0 255.0.0.0 10.0.0.1
ip route 90.0.0.0 255.0.0.0 50.0.0.1
ip route 100.0.0.0 255.0.0.0 30.0.0.2
do show ip route
```

```
R1(config)#ip forward-protocol nd
R1(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
R1(config)#ip route 50.0.0.0 255.0.0.0 10.0.0.1
R1(config)#ip route 80.0.0.0 255.0.0.0 10.0.0.1
R1(config)#ip route 90.0.0.0 255.0.0.0 50.0.0.1
R1(config)#ip route 100.0.0.0 255.0.0.0 30.0.0.2
R1(config)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S    50.0.0.0/8 [1/0] via 10.0.0.1
S    100.0.0.0/8 [1/0] via 30.0.0.2
S    80.0.0.0/8 [1/0] via 10.0.0.1
C    20.0.0.0/8 is directly connected, Serial1/0
C    10.0.0.0/8 is directly connected, Serial1/1
S    90.0.0.0/8 [1/0] via 50.0.0.1
C    60.0.0.0/8 is directly connected, FastEthernet0/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
R1(config)#
```

Configure Static routing in Router 2

```
config t

ip forward-protocol nd

ip route 10.0.0.0 255.0.0.0 20.0.0.1

ip route 80.0.0.0 255.0.0.0 10.0.0.1

ip route 50.0.0.0 255.0.0.0 10.0.0.1

ip route 90.0.0.0 255.0.0.0 50.0.0.1

ip route 100.0.0.0 255.0.0.0 30.0.0.2

ip route 40.0.0.0 255.0.0.0 30.0.0.2

do show ip route
```

```
R2(config)#ip forward-protocol nd
R2(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
R2(config)#ip route 80.0.0.0 255.0.0.0 10.0.0.1
R2(config)#ip route 50.0.0.0 255.0.0.0 10.0.0.1
R2(config)#ip route 90.0.0.0 255.0.0.0 50.0.0.1
R2(config)#ip route 100.0.0.0 255.0.0.0 30.0.0.2
R2(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
R2(config)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S    50.0.0.0/8 [1/0] via 10.0.0.1
S    100.0.0.0/8 [1/0] via 30.0.0.2
C    70.0.0.0/8 is directly connected, FastEthernet0/0
S    80.0.0.0/8 [1/0] via 10.0.0.1
C    20.0.0.0/8 is directly connected, Serial1/1
S    40.0.0.0/8 [1/0] via 30.0.0.2
S    10.0.0.0/8 [1/0] via 20.0.0.1
S    90.0.0.0/8 [1/0] via 50.0.0.1
C    30.0.0.0/8 is directly connected, Serial1/2
R2(config)#
```

Configure Static routing in Router 3

```
ip forward-protocol nd
ip route 60.0.0.0 255.0.0.0 10.0.0.2
ip route 20.0.0.0 255.0.0.0 10.0.0.2
ip route 70.0.0.0 255.0.0.0 20.0.0.2
ip route 30.0.0.0 255.0.0.0 20.0.0.2
ip route 90.0.0.0 255.0.0.0 50.0.0.1
ip route 40.0.0.0 255.0.0.0 50.0.0.1
ip route 100.0.0.0 255.0.0.0 40.0.0.1
do show ip route
```

```
R3(config)#ip forward-protocol nd
R3(config)#ip route 60.0.0.0 255.0.0.0 10.0.0.2
R3(config)#ip route 20.0.0.0 255.0.0.0 10.0.0.2
R3(config)#ip route 70.0.0.0 255.0.0.0 20.0.0.2
R3(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
R3(config)#ip route 90.0.0.0 255.0.0.0 50.0.0.1
R3(config)#ip route 40.0.0.0 255.0.0.0 50.0.0.1
R3(config)#ip route 100.0.0.0 255.0.0.0 40.0.0.1
R3(config)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    50.0.0.0/8 is directly connected, Serial1/3
S    100.0.0.0/8 [1/0] via 40.0.0.1
S    70.0.0.0/8 [1/0] via 20.0.0.2
C    80.0.0.0/8 is directly connected, FastEthernet0/0
S    20.0.0.0/8 [1/0] via 10.0.0.2
S    40.0.0.0/8 [1/0] via 50.0.0.1
C    10.0.0.0/8 is directly connected, Serial1/0
S    90.0.0.0/8 [1/0] via 50.0.0.1
S    60.0.0.0/8 [1/0] via 10.0.0.2
S    30.0.0.0/8 [1/0] via 20.0.0.2
R3(config)#
```

Configure Static routing in Router 4

```
config t
ip forward-protocol nd
ip route 80.0.0.0 255.0.0.0 50.0.0.2
ip route 10.0.0.0 255.0.0.0 50.0.0.2
ip route 60.0.0.0 255.0.0.0 10.0.0.2
ip route 20.0.0.0 255.0.0.0 10.0.0.2
ip route 100.0.0.0 255.0.0.0 40.0.0.1
ip route 30.0.0.0 255.0.0.0 40.0.0.1
ip route 70.0.0.0 255.0.0.0 30.0.0.1
```

do show ip route

```
R4(config)#ip forward-protocol nd
R4(config)#ip route 80.0.0.0 255.0.0.0 50.0.0.2
R4(config)#ip route 10.0.0.0 255.0.0.0 50.0.0.2
R4(config)#ip route 60.0.0.0 255.0.0.0 10.0.0.2
R4(config)#ip route 20.0.0.0 255.0.0.0 10.0.0.2
R4(config)#ip route 100.0.0.0 255.0.0.0 40.0.0.1
R4(config)#ip route 30.0.0.0 255.0.0.0 40.0.0.1
R4(config)#ip route 70.0.0.0 255.0.0.0 30.0.0.1
R4(config)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    50.0.0.0/8 is directly connected, Serial1/2
S    100.0.0.0/8 [1/0] via 40.0.0.1
S    70.0.0.0/8 [1/0] via 30.0.0.1
S    80.0.0.0/8 [1/0] via 50.0.0.2
S    20.0.0.0/8 [1/0] via 10.0.0.2
C    40.0.0.0/8 is directly connected, Serial1/1
S    10.0.0.0/8 [1/0] via 50.0.0.2
C    90.0.0.0/8 is directly connected, FastEthernet0/0
S    60.0.0.0/8 [1/0] via 10.0.0.2
S    30.0.0.0/8 [1/0] via 40.0.0.1
R4(config)#
```

Configure Static routing in Router 5

```
config t
ip forward-protocol nd
ip route 90.0.0.0 255.0.0.0 40.0.0.2
ip route 50.0.0.0 255.0.0.0 40.0.0.2
ip route 80.0.0.0 255.0.0.0 50.0.0.2
ip route 10.0.0.0 255.0.0.0 50.0.0.2
ip route 70.0.0.0 255.0.0.0 30.0.0.1
ip route 20.0.0.0 255.0.0.0 30.0.0.1
ip route 60.0.0.0 255.0.0.0 20.0.0.1
do show ip route
```

```
R5(config)#ip forward-protocol nd
R5(config)#ip route 90.0.0.0 255.0.0.0 40.0.0.2
R5(config)#ip route 50.0.0.0 255.0.0.0 40.0.0.2
R5(config)#ip route 80.0.0.0 255.0.0.0 50.0.0.2
R5(config)#ip route 10.0.0.0 255.0.0.0 50.0.0.2
R5(config)#ip route 70.0.0.0 255.0.0.0 30.0.0.1
R5(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
R5(config)#ip route 60.0.0.0 255.0.0.0 20.0.0.1
R5(config)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S    50.0.0.0/8 [1/0] via 40.0.0.2
C    100.0.0.0/8 is directly connected, FastEthernet0/0
S    70.0.0.0/8 [1/0] via 30.0.0.1
S    80.0.0.0/8 [1/0] via 50.0.0.2
S    20.0.0.0/8 [1/0] via 30.0.0.1
C    40.0.0.0/8 is directly connected, Serial1/0
S    10.0.0.0/8 [1/0] via 50.0.0.2
S    90.0.0.0/8 [1/0] via 40.0.0.2
S    60.0.0.0/8 [1/0] via 20.0.0.1
C    30.0.0.0/8 is directly connected, Serial1/3
```

Step 4: Here we are checking the packet transfer only between PC5 and PC4

ping 70.0.0.2

```
PC5> ping 70.0.0.2
70.0.0.2 icmp_seq=1 ttl=64 time=0.001 ms
70.0.0.2 icmp_seq=2 ttl=64 time=0.001 ms
70.0.0.2 icmp_seq=3 ttl=64 time=0.001 ms
70.0.0.2 icmp_seq=4 ttl=64 time=0.001 ms
70.0.0.2 icmp_seq=5 ttl=64 time=0.001 ms
```

Here we are checking the packet transfer only between PC1 and PC3

ping 100.0.0.2

```
PC1> ping 100.0.0.2
100.0.0.2 icmp_seq=1 timeout
84 bytes from 100.0.0.2 icmp_seq=2 ttl=61 time=149.783 ms
84 bytes from 100.0.0.2 icmp_seq=3 ttl=61 time=120.066 ms
84 bytes from 100.0.0.2 icmp_seq=4 ttl=61 time=119.831 ms
84 bytes from 100.0.0.2 icmp_seq=5 ttl=61 time=119.981 ms
```