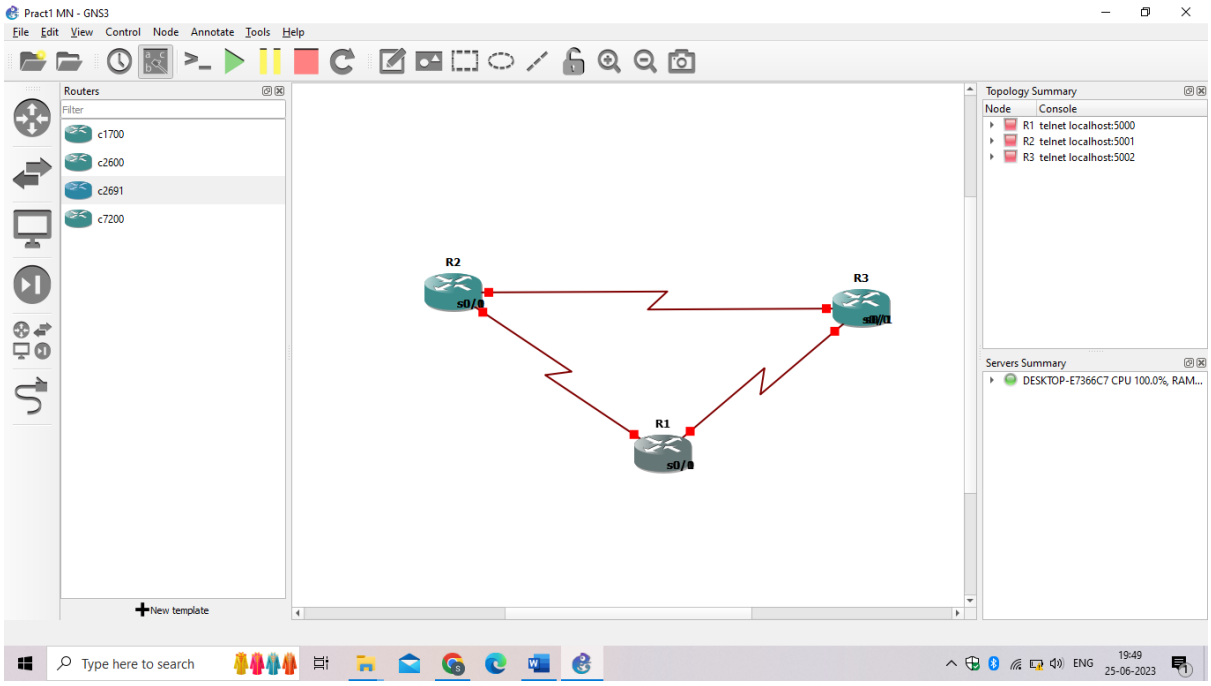
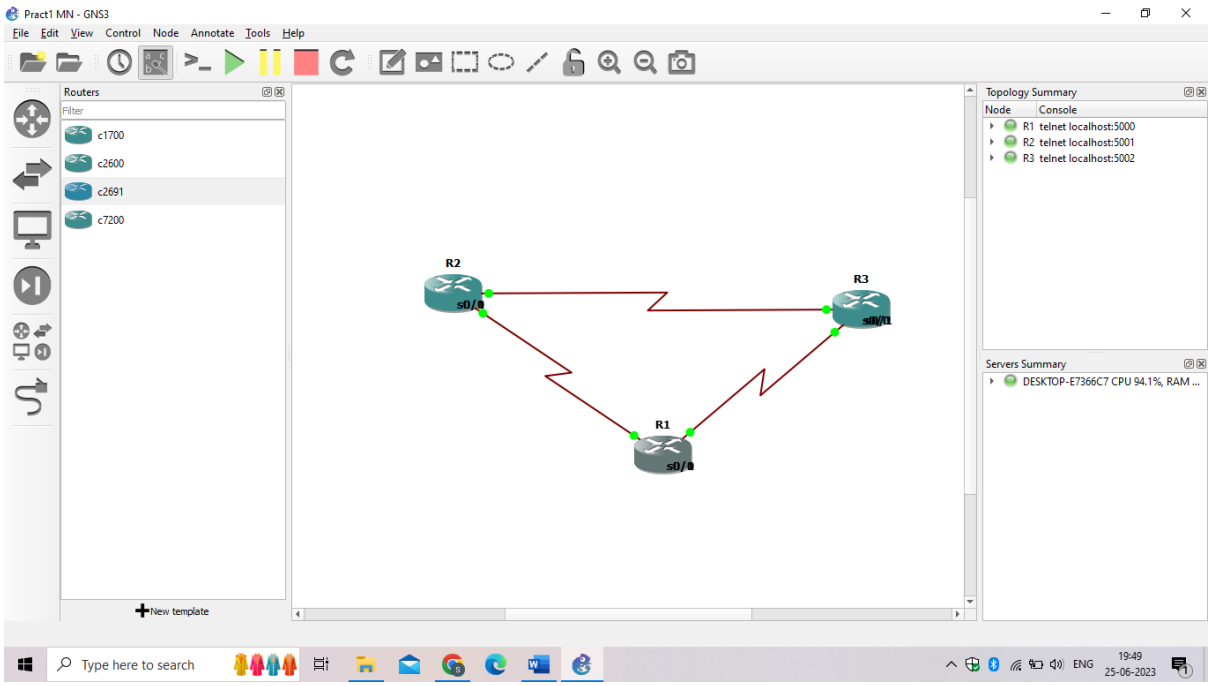
PRACTICAL NO :- 1

Aim:-Configure IP SLA Tracking and Path Control Topology





Step 1: Configure loopbacks and assign addresses.

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

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

R1# **show interfaces description | include up**

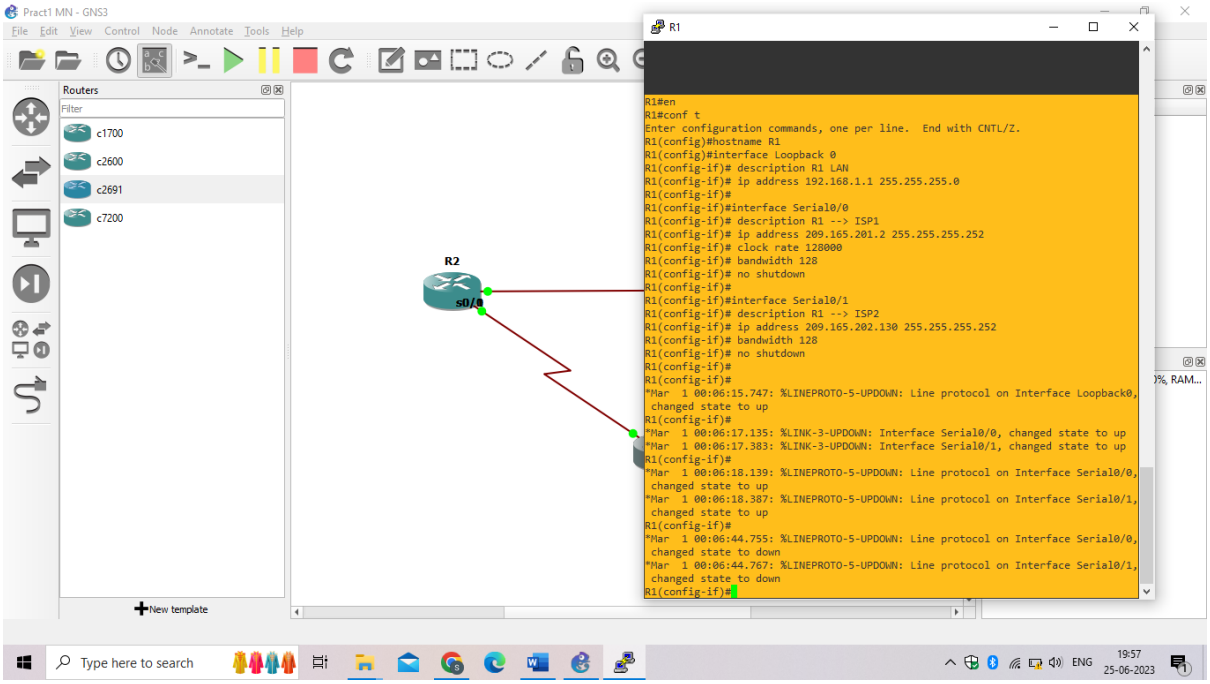
Router R1

hostname R1

interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0

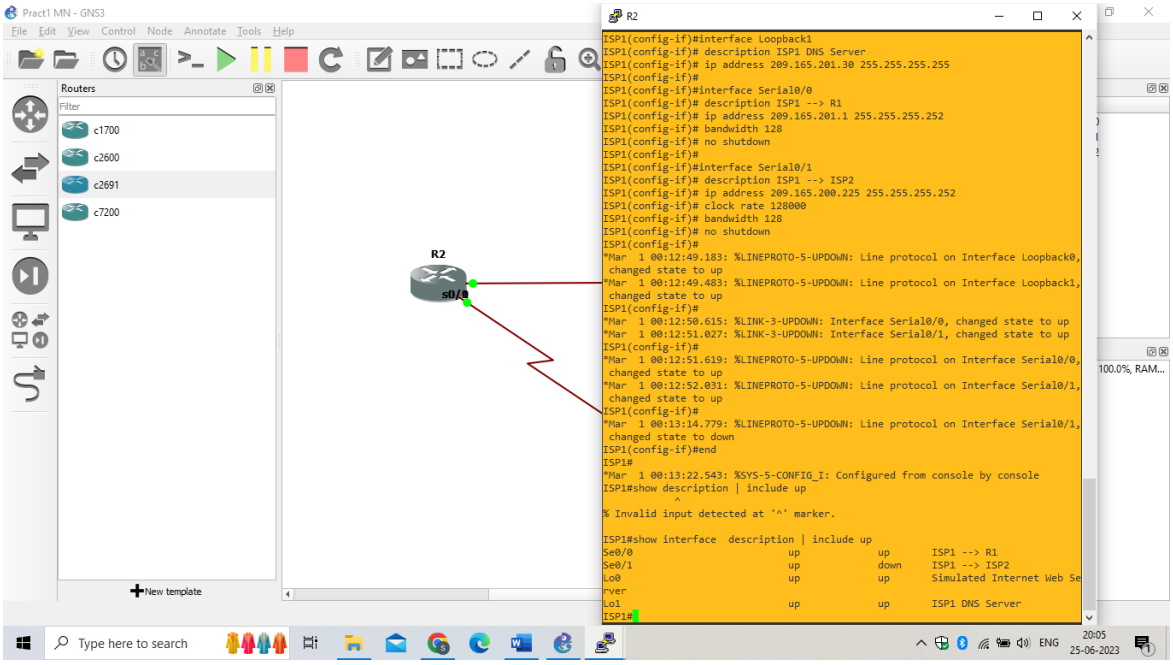
interface Serial0/0
 description R1 --> ISP1
 ip address 209.165.201.2 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

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



Router ISP1 (R2)

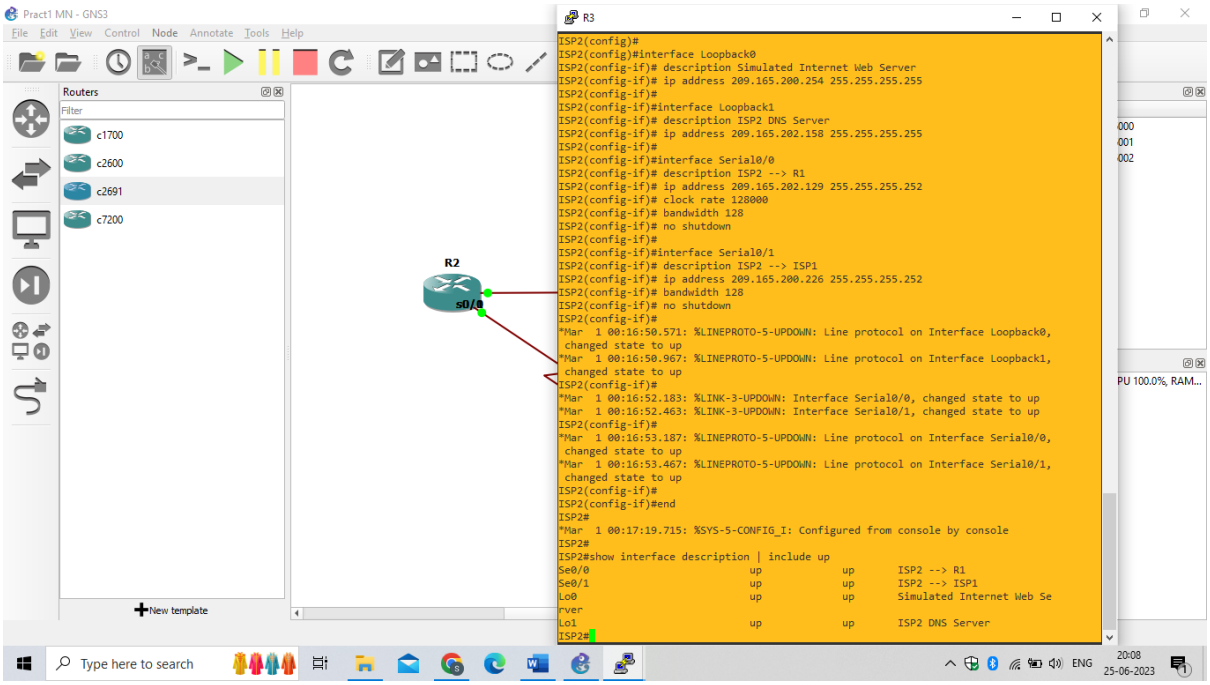
```
hostname ISP1

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255

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

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



Router ISP2 (R3)

```
hostname ISP2

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255
```

```
interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255

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

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

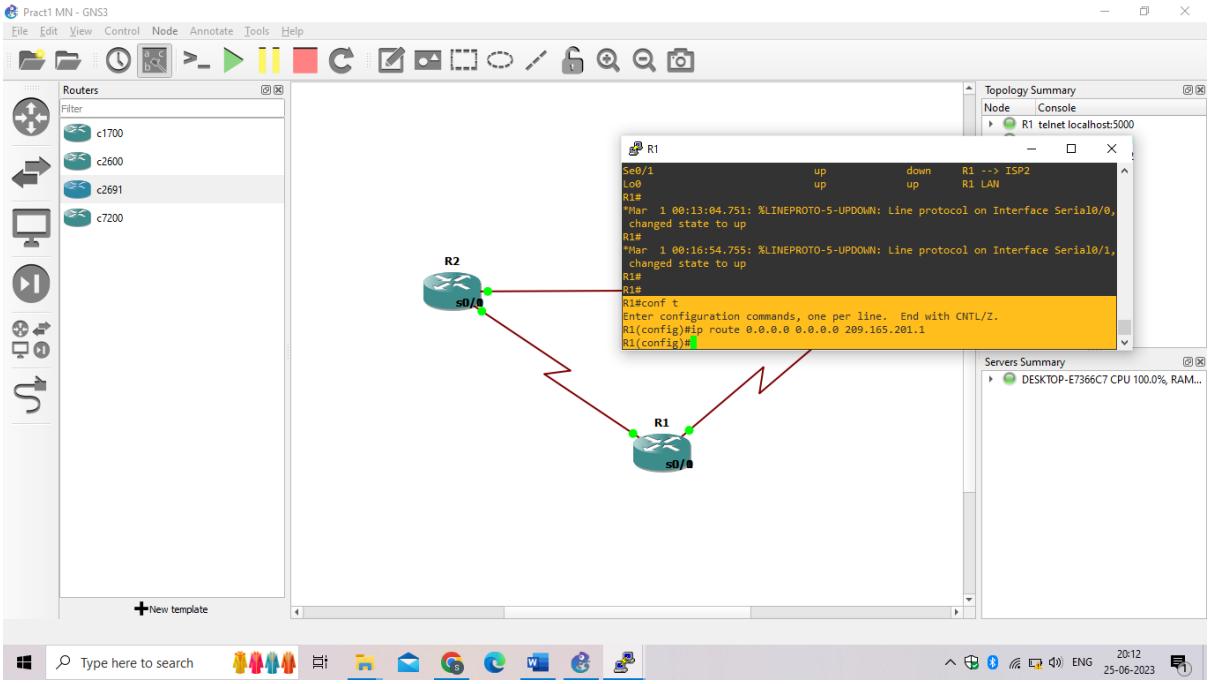


Step 2: Configure static routing.

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

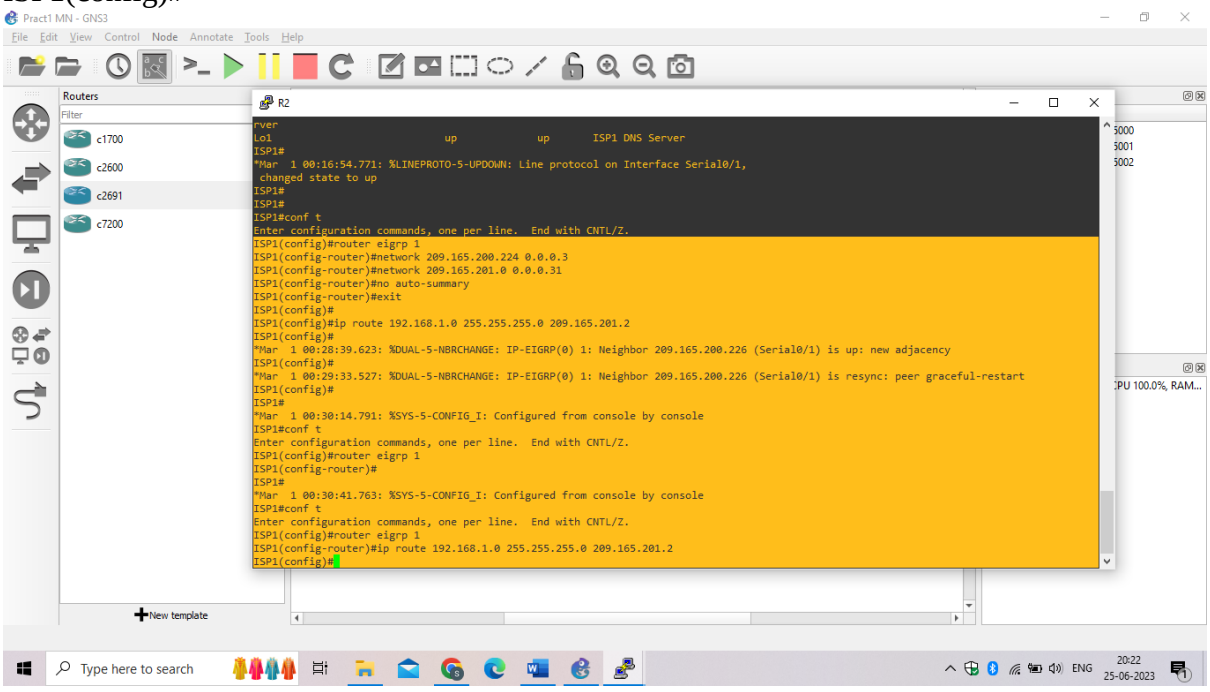
Router R1

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



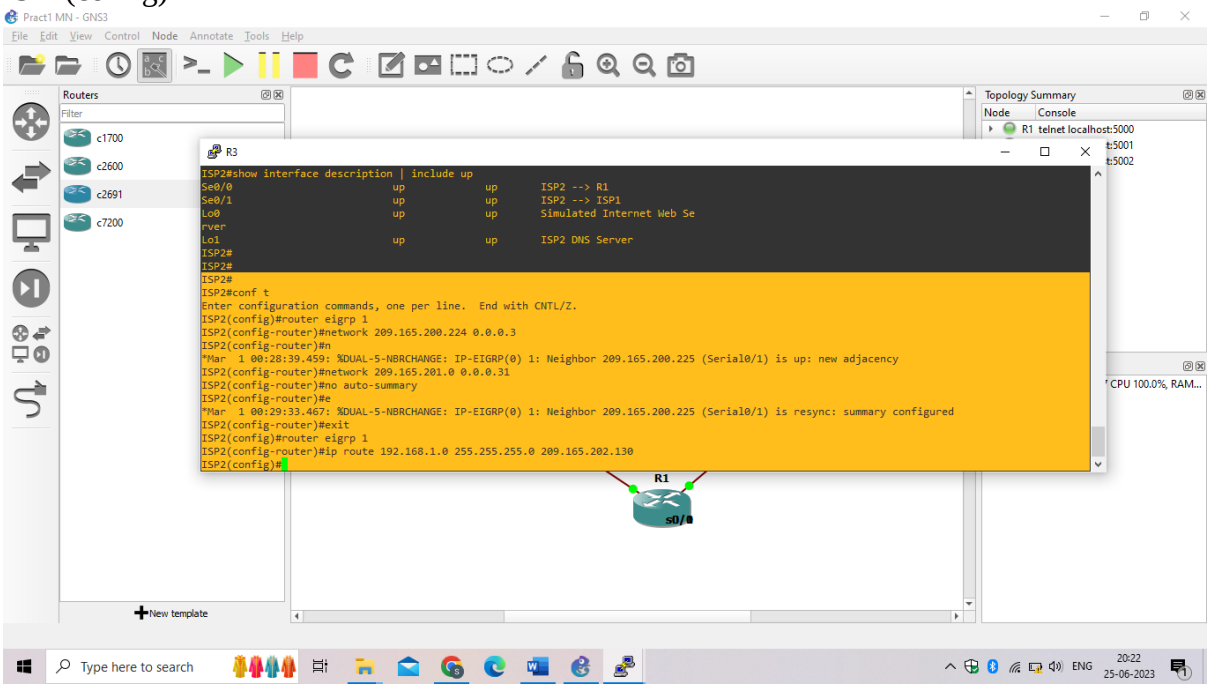
Router ISP1 (R2)

ISP1(config)# **router eigrp 1**
ISP1(config-router)# **network 209.165.200.224 0.0.0.3**
ISP1(config-router)# **network 209.165.201.0 0.0.0.31**
ISP1(config-router)# **no auto-summary**
ISP1(config-router)# **exit**
ISP1(config)#
ISP1(config-router)# **ip route 192.168.1.0 255.255.255.0 209.165.201.2**
ISP1(config)#



Router ISP2 (R3)

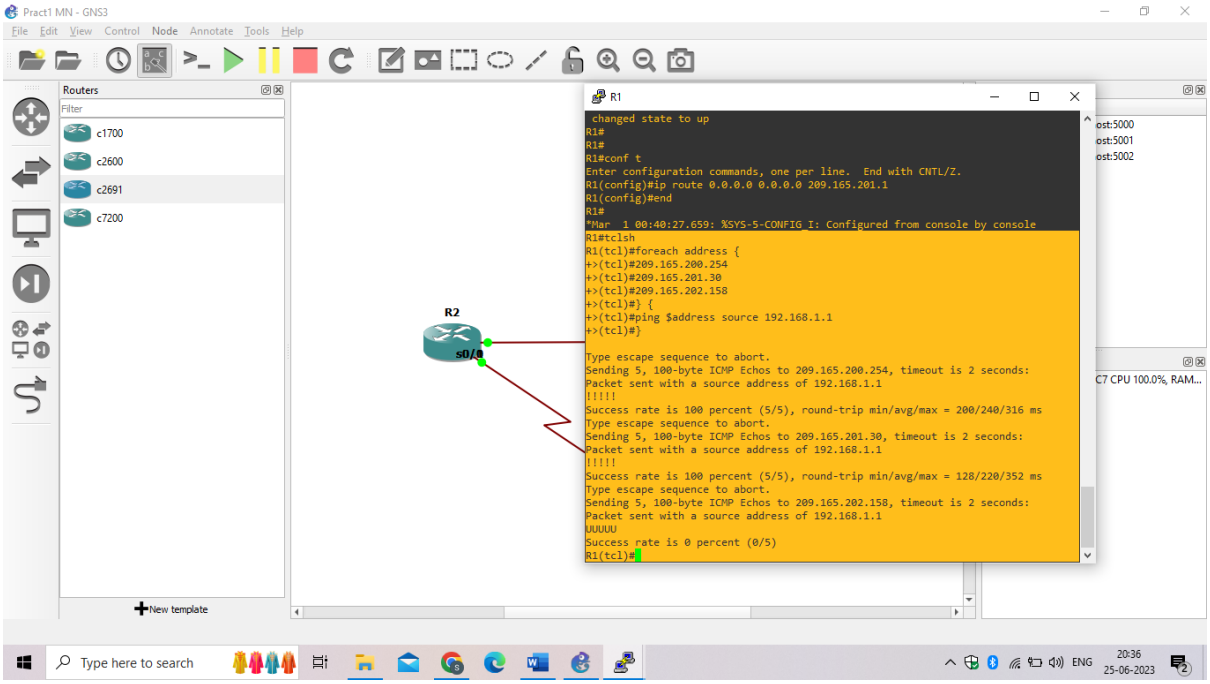
ISP2(config)# **router eigrp 1**
ISP2(config-router)# **network 209.165.200.224 0.0.0.3**
ISP2(config-router)# **network 209.165.202.128 0.0.0.31**
ISP2(config-router)# **no auto-summary**
ISP2(config-router)# **exit**
ISP2(config)#
ISP2(config-router)# **ip route 192.168.1.0 255.255.255.0 209.165.202.130**
ISP2(config)#



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

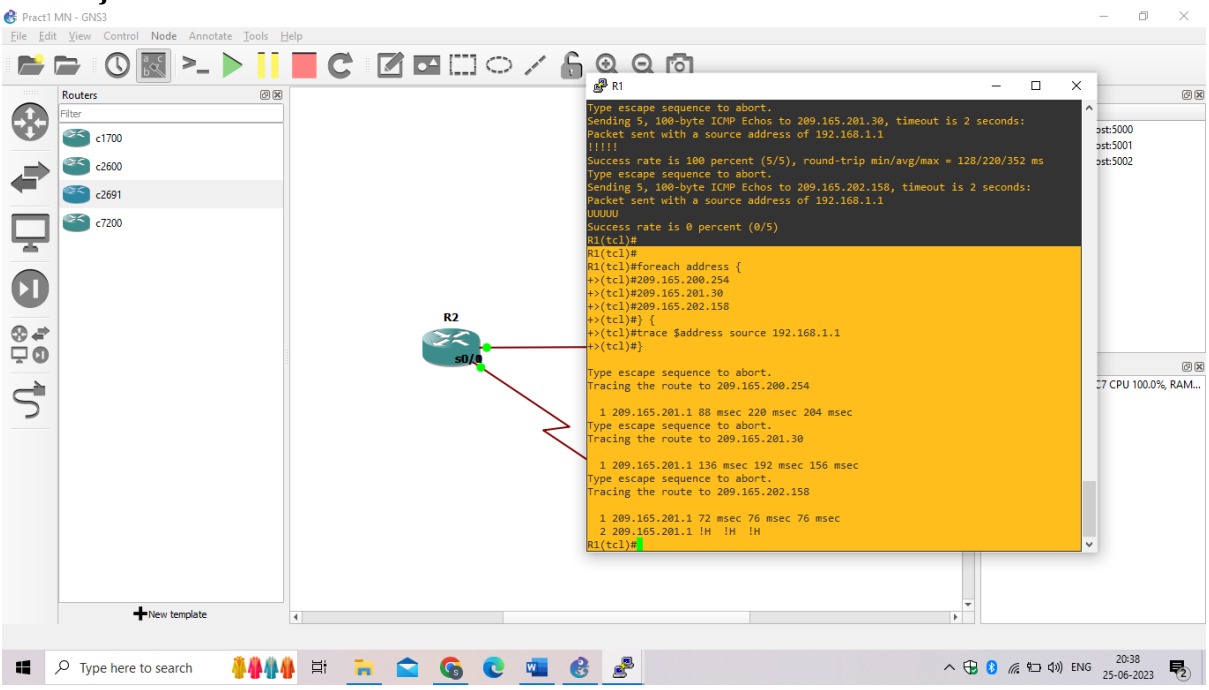
Before implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2 DNS server to verify connectivity. You can copy the following Tcl script and paste it into R1.

foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
ping $address source 192.168.1.1
}



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

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
trace $address source 192.168.1.1
}
```



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.

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

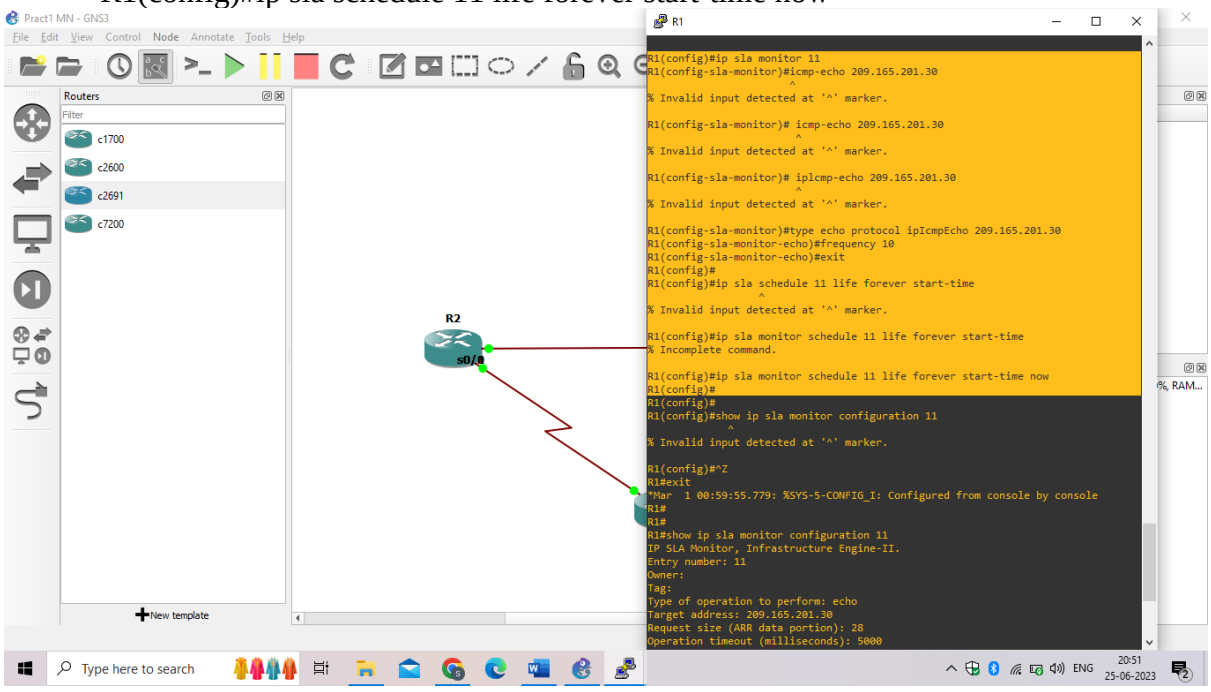
R1(config-sla-monitor)#type echo protocol ipIcmpEcho 209.165.201.30

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

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

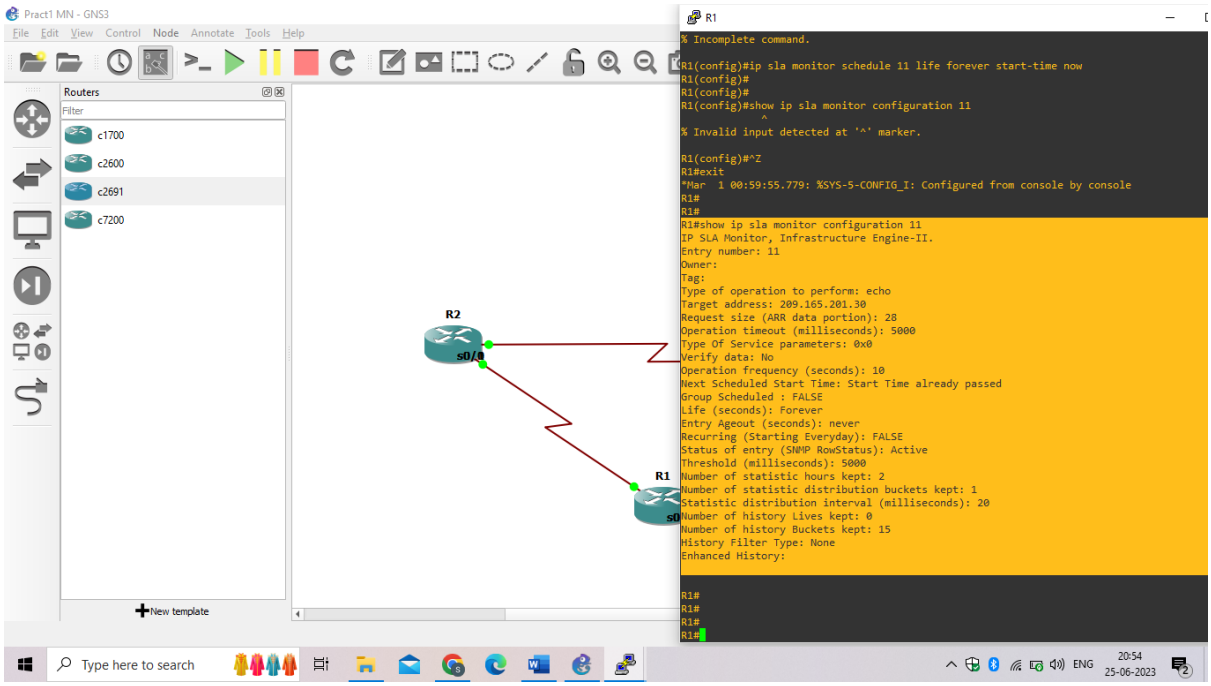
R1(config)#

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



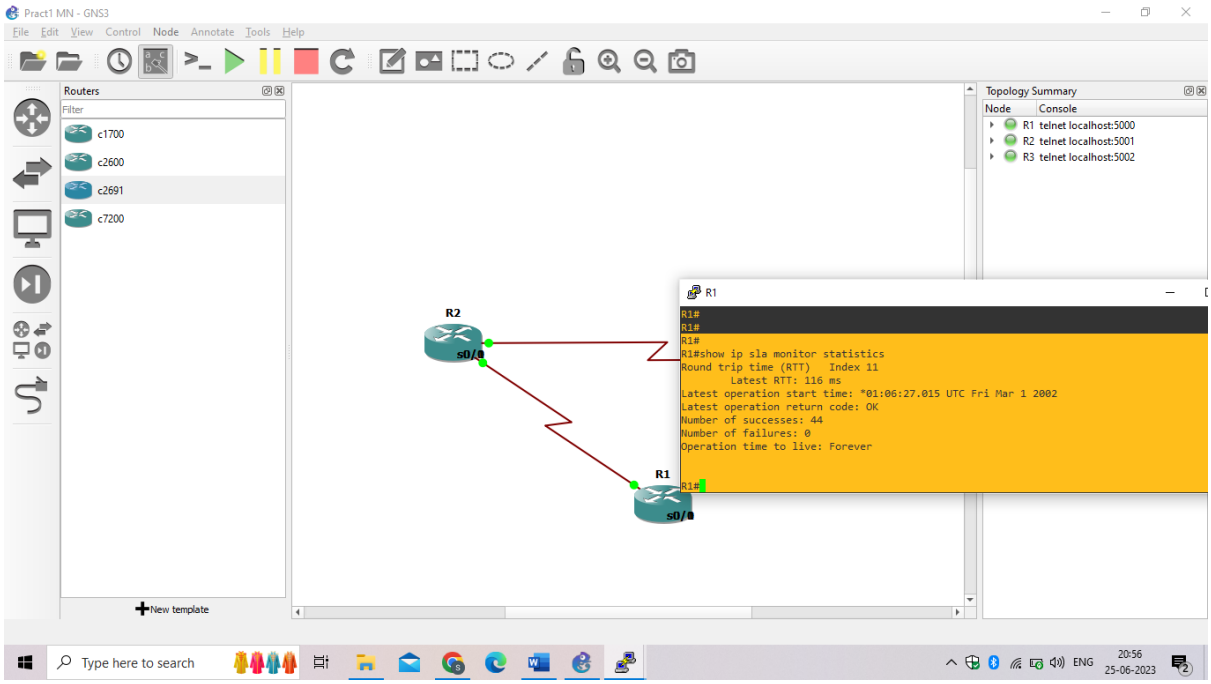
b. Verify the IP SLAs configuration of operation 11 using the **show ip sla configuration 11** command.

R1# **show ip sla monitor configuration 11**

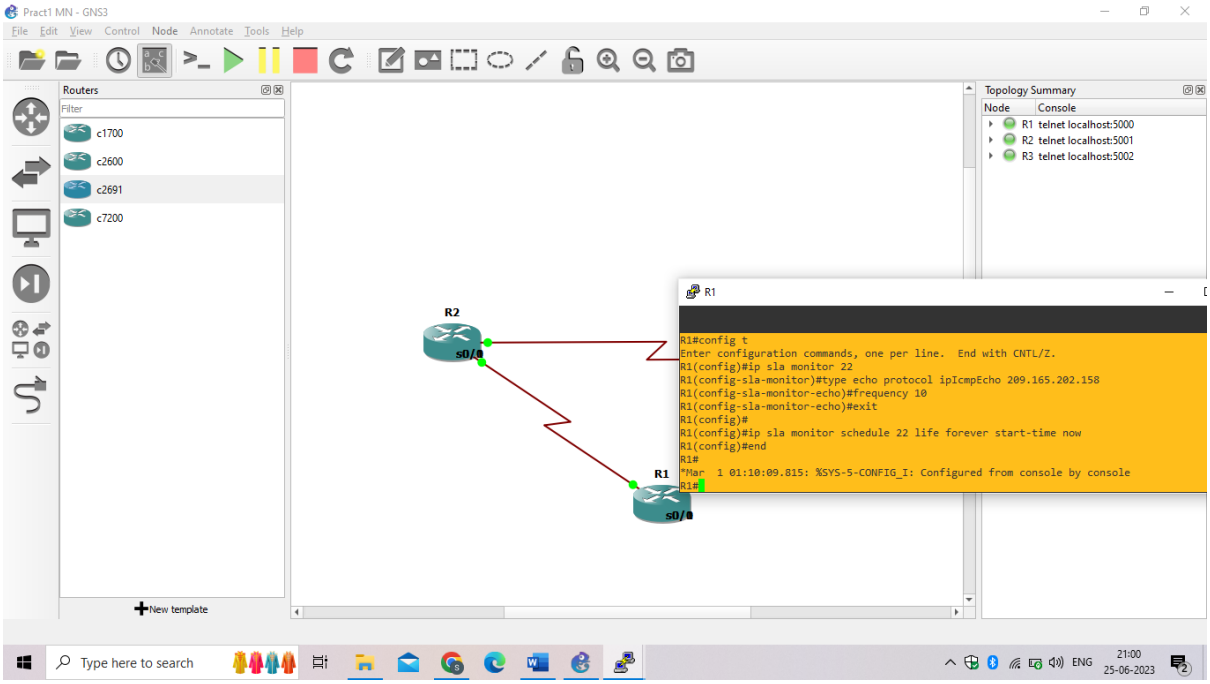


c. Issue the **show ip sla statistics** command to display the number of successes, failures, and results of the latest operations.

R1# **show ip sla statistics**

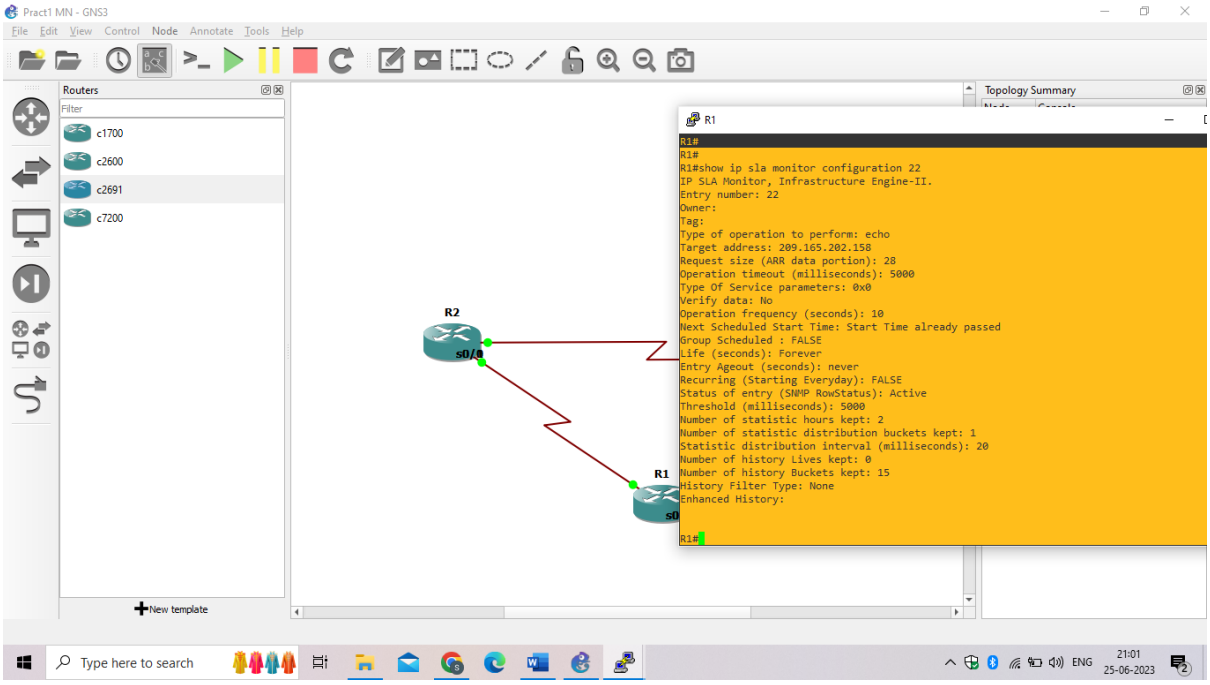


d.

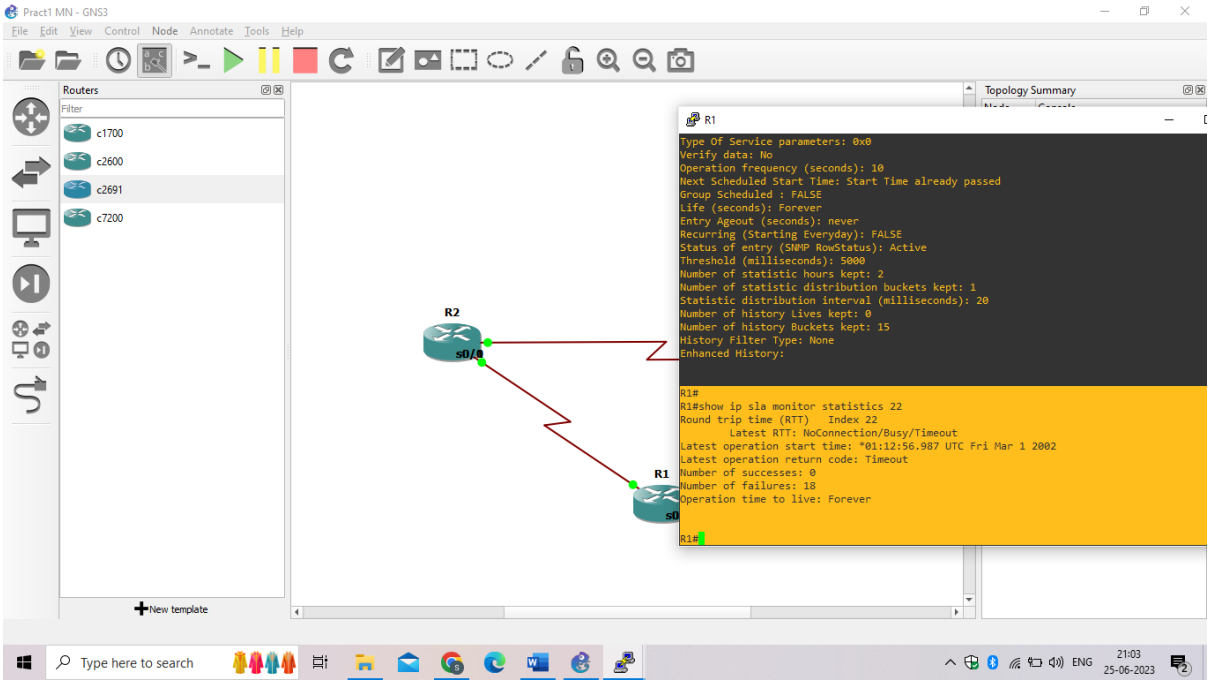


e. Verify the new probe using the **show ip sla configuration** and **show ip sla statistics** commands.

R1# **show ip sla configuration 22**



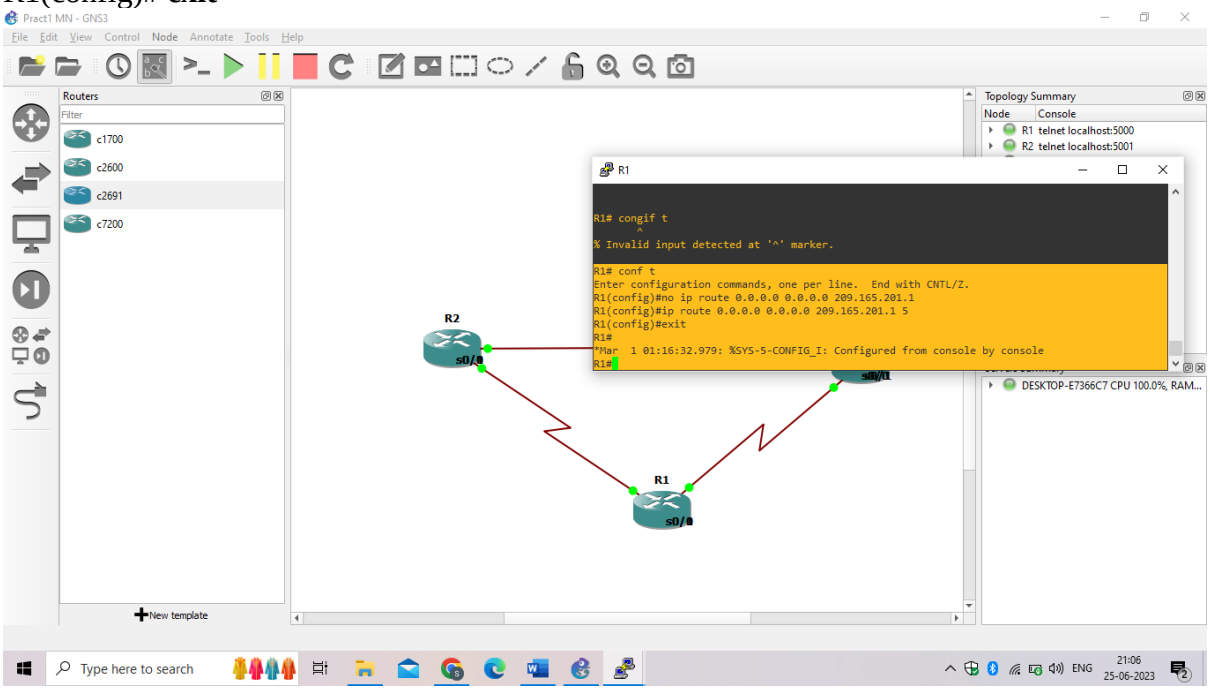
R1# **show ip sla statistics 22**



Step 4: Configure tracking options.

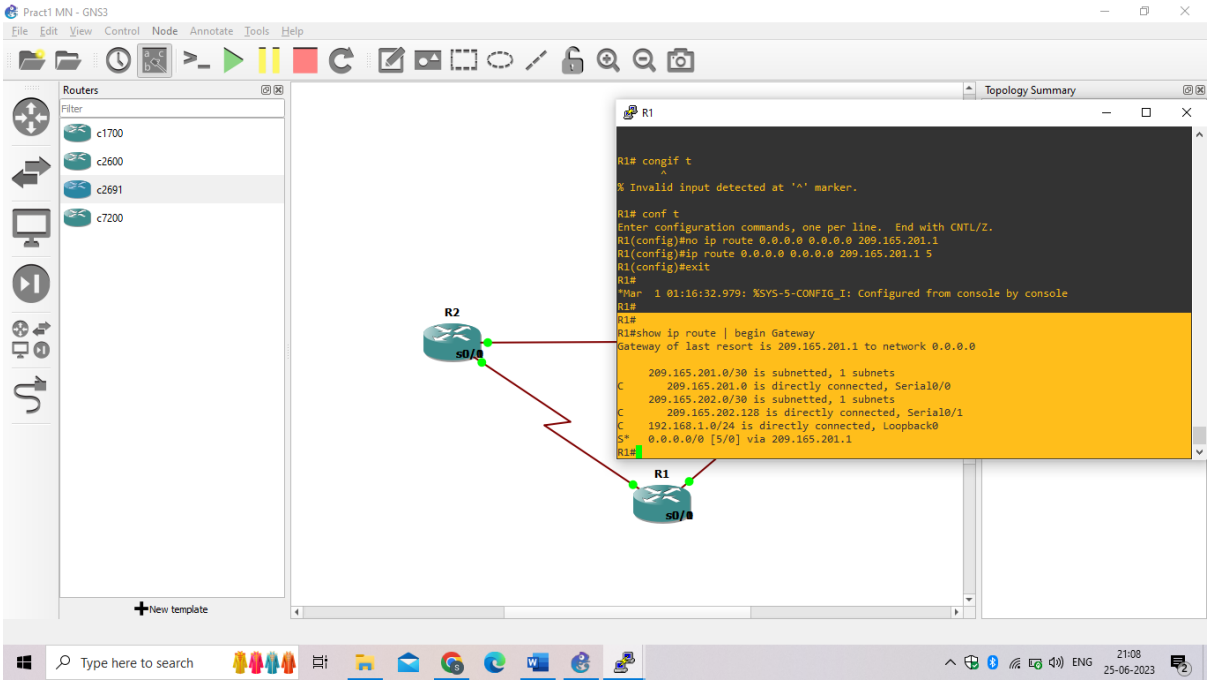
a. On R1, remove the current default route and replace it with a floating static route having an administrative distance of 5.

R1(config)# **no ip route 0.0.0.0 0.0.0.0 209.165.201.1**
R1(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.1 5**
R1(config)# **exit**



b. Verify the routing table.

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



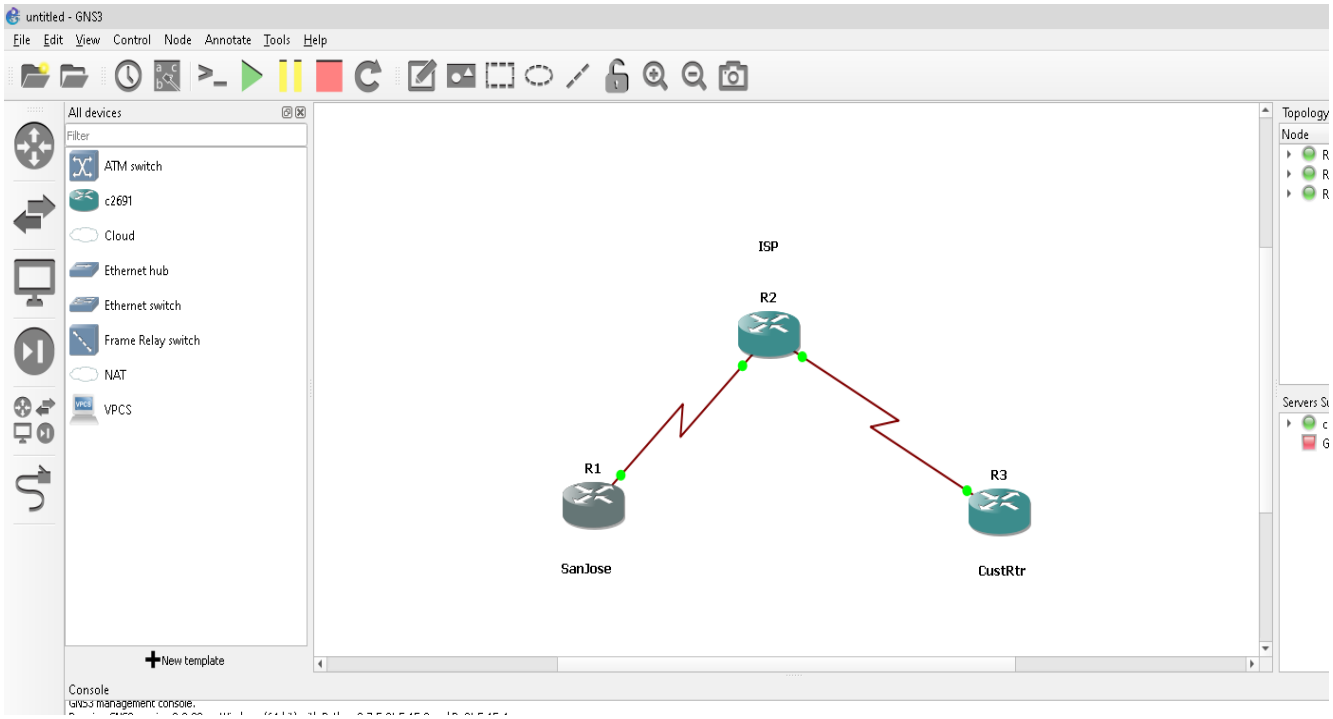
c. From global configuration mode on R1, use the **track 1 ip sla 11 reachability** command to enter the config-track subconfiguration mode.

R1(config)# **track 1 ip sla 11 reachability**
R1(config-track)#

PRACTICAL NO:-2

Aim:- Using the AS_PATH Attribute

Step 1: Prepare the routers for the lab.



Step 2: Configure the hostname and interface addresses.

Router R1 (hostname SanJose)

hostname SanJose

interface Loopback0

ip address 10.1.1.1 255.255.255.0

interface Serial1/1

ip address 192.168.1.5 255.255.255.252

clock rate 128000

no shutdown

Router R2 (hostname ISP)

hostname ISP

interface Loopback0

ip address 10.2.2.1 255.255.255.0

interface Serial1/1

ip address 192.168.1.6 255.255.255.252

no shutdown

interface Serial1/0

ip address 172.24.1.17 255.255.255.252

clock rate 128000

no shutdown

Router R3 (hostname CustRtr)

hostname CustRtr

interface Loopback0

ip address 10.3.3.1 255.255.255.0

interface Serial1/0

ip address 172.24.1.18 255.255.255.252

no shutdown

<u>Step 3</u>: 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.

```
SanJose(config-if)#exit
SanJose(config)#router bgp 100
SanJose(config-router)#neighbor 192.168.1.6 remote-as 300
SanJose(config-router)#network 10.1.1.0 mask 255.255.255.0
SanJose(config-router)#
*Mar  1 00:36:40.711: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Up
```

```
ISP(config-if)#exit
ISP(config)# router bgp 300
ISP(config-router)#
*Mar  1 00:36:21.743: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed sta
te to down
ISP(config-router)#neighbor 192.168.1.5 remote-as 100
ISP(config-router)#neighbor 172.24.1.18 remote-as 65000
ISP(config-router)#
*Mar  1 00:36:40.827: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
ISP(config-router)#network 10.2.2.0 mask 255.255.255.0
ISP(config-router)#
*Mar  1 00:40:01.735: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed sta
```

```
*Mar  1 00:39:47.487: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed sta
te to up
CustRtr(config-if)#exit
CustRtr(config)#router bgp 65000
CustRtr(config-router)# neighbor 172.24.1.17 remote-as 300
CustRtr(config-router)#network 10.3.3.0 mask 255.255.255.0
CustRtr(config-router)#
*Mar  1 00:40:56.447: %BGP-5-ADJCHANGE: neighbor 172.24.1.17 Up
```

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

```
ISP#show ip bgp neighbors
*Mar  1 00:41:13.779: %SYS-5-CONFIG_I: Configured from console by console
ISP#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 00:01:21
  Last read 00:00:21, last write 00:00:21, 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
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                3          1
    Keepalives:             4          4
    Route Refresh:          0          0
    Total:                  8          6
  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

ISP#show ip route
```

Step 4: Remove the private AS.

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

```
SanJose#
*Mar  1 00:46:32.859: %SYS-5-CONFIG_I: Configured from console by console
SanJose#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, 00:05:14
B       10.2.2.0 [20/0] via 192.168.1.6, 00:08:00
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, Serial1/1
SanJose#
```

B & C. Ping the 10.3.3.1 address from SanJose.

```
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, Serial1/1
SanJose#ping
Protocol [ip]:
Target IP address: 10.3.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
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 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 = 80/101/128 ms
SanJose#
```

d. Check the BGP table from SanJose by using the show ip bgp command. Note the AS path for the 10.3.3.0 network. The AS 65000 should be listed in the path to 10.3.3.0.

```
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 = 88/99/112 ms
SanJose#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
SanJose#
```

e. Configure ISP to strip the private AS numbers from BGP routes exchanged with SanJose using the following commands.

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

ISP#
ISP#
ISP#Config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 300
ISP(config-router)# neighbor 192.168.1.5 remove-private-as
ISP(config-router)#
```

f. After issuing these commands, use the clear ip bgp * command on ISP to reestablish the BGP relationship between the three routers. Wait several seconds and then return to SanJose to check its routing table.

Note: The clear ip bgp * soft command can also be used to force each router to resend its BGP table.

```
   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
SanJose#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 = 68/93/128 ms
SanJose#
```

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

```
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 = 68/93/128 ms
SanJose#show ip bgp
BGP table version is 5, 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 i
SanJose#
```

Step 5: Use the AS_PATH attribute to filter routes.

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 ISP.

```
ISP#Config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 300
ISP(config-router)# neighbor 192.168.1.5 remove-private-as
ISP(config-router)#exit
ISP(config)#ip as-path access-list 1 deny ^100$
ISP(config)#ip as-path access-list 1 permit .*
ISP(config)#
```

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

```
ISP(config-router)# neighbor 192.168.1.5 remove-private-as
ISP(config-router)#exit
ISP(config)#ip as-path access-list 1 deny ^100$
ISP(config)#ip as-path access-list 1 permit .*
ISP(config)#router bgp 300
ISP(config-router)#neighbor 172.24.1.18 filter-list 1 out
ISP(config-router)#
```

c. Use the clear ip bgp * command to reset the routing information. Wait several seconds and then check the routing table for ISP. The route to 10.1.1.0 should be in the routing table.

```
ISP(config-router)#neighbor 172.24.1.18 filter-list 1 out
ISP(config-router)#exit
ISP(config)#exit
ISP#show ip route
*Mar  1 01:08:55.247: %SYS-5-CONFIG_I: Configured from console by console
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

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected, Serial1/0
     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 172.24.1.18, 00:28:02
C       10.2.2.0 is directly connected, Loopback0
B       10.1.1.0 [20/0] via 192.168.1.5, 00:30:18
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial1/1
ISP#
```

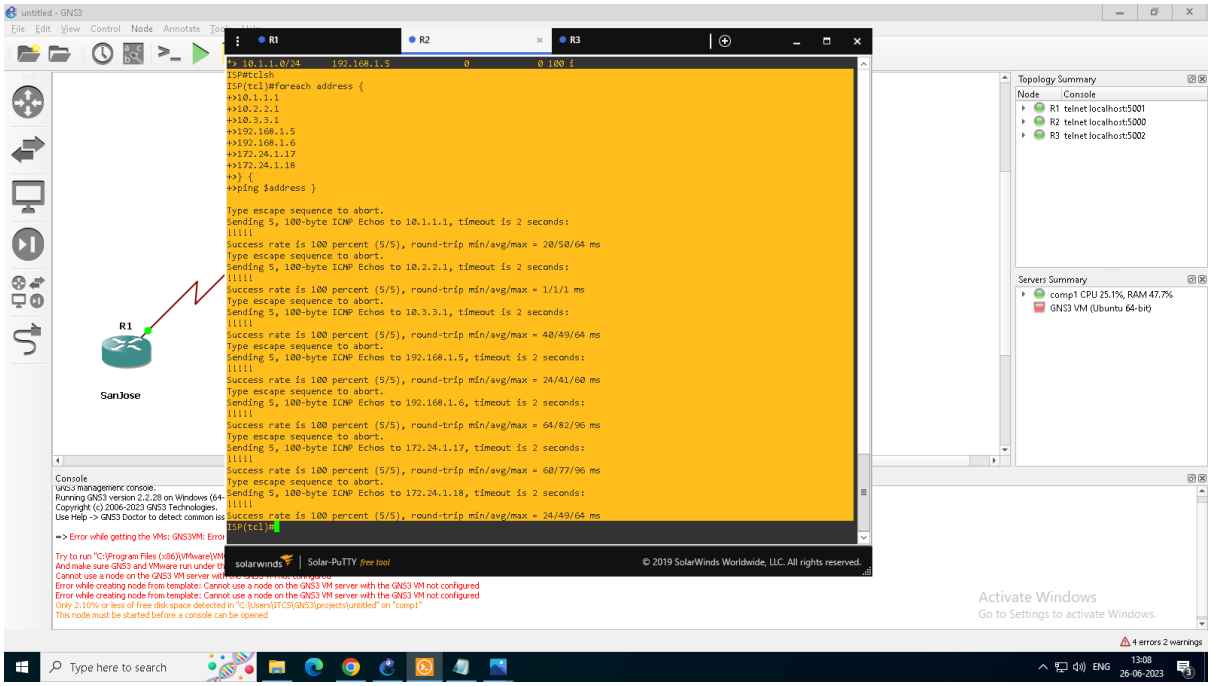
d. Check the routing table for CustRtr. It should not have a route to 10.1.1.0 in its routing table.

```
CustRtr(config-router)#exit
CustRtr(config)#exit
CustRtr#
*Mar  1 01:09:47.547: %SYS-5-CONFIG_I: Configured from console by console
CustRtr#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

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected, Serial1/0
     10.0.0.0/24 is subnetted, 2 subnets
C       10.3.3.0 is directly connected, Loopback0
B       10.2.2.0 [20/0] via 172.24.1.17, 00:28:57
CustRtr#
```

e. Return to ISP and verify that the filter is working as intended. Issue the show ip bgp regexp ^100$ command.

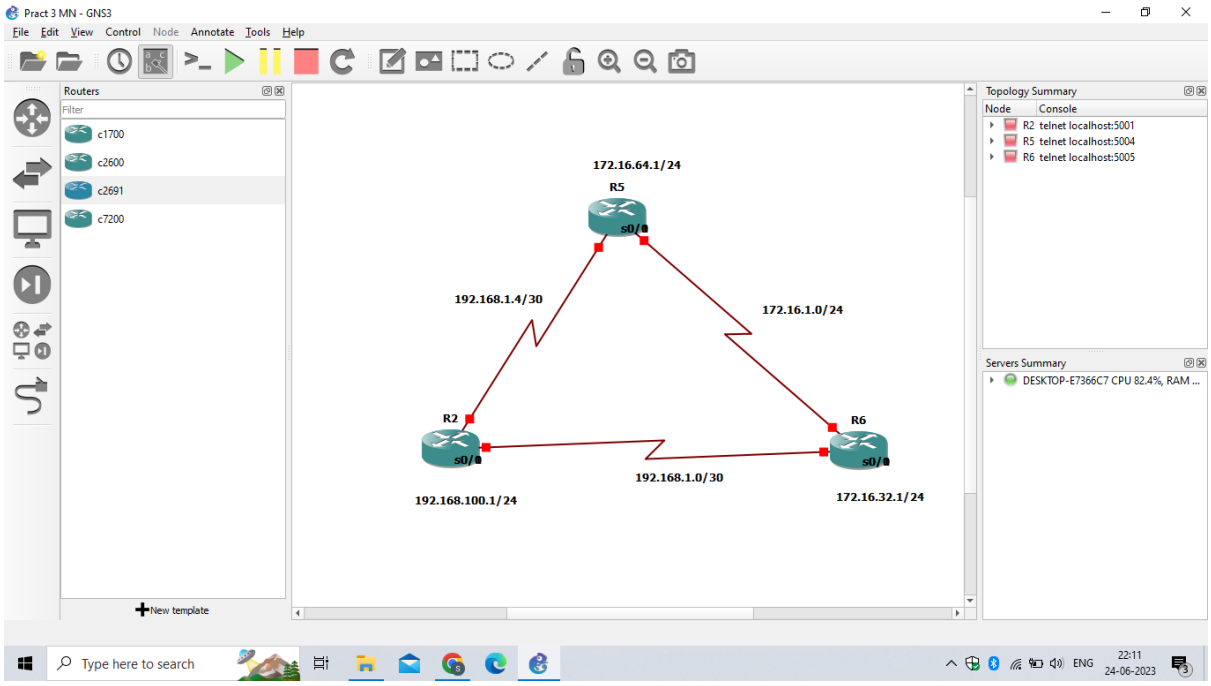
```
C       10.2.2.0 is directly connected, Loopback0
B       10.1.1.0 [20/0] via 192.168.1.5, 00:30:18
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial1/1
ISP#show ip bgp regexp ^100$
BGP table version is 4, local router ID is 10.2.2.1
Status codes: s suppressed, 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
ISP#
```

f. Run the following Tcl script on all routers to verify whether there is connectivity. All pings from ISP should be successful. SanJose should not be able to ping the CustRtr loopback 10.3.3.1 or the WAN link

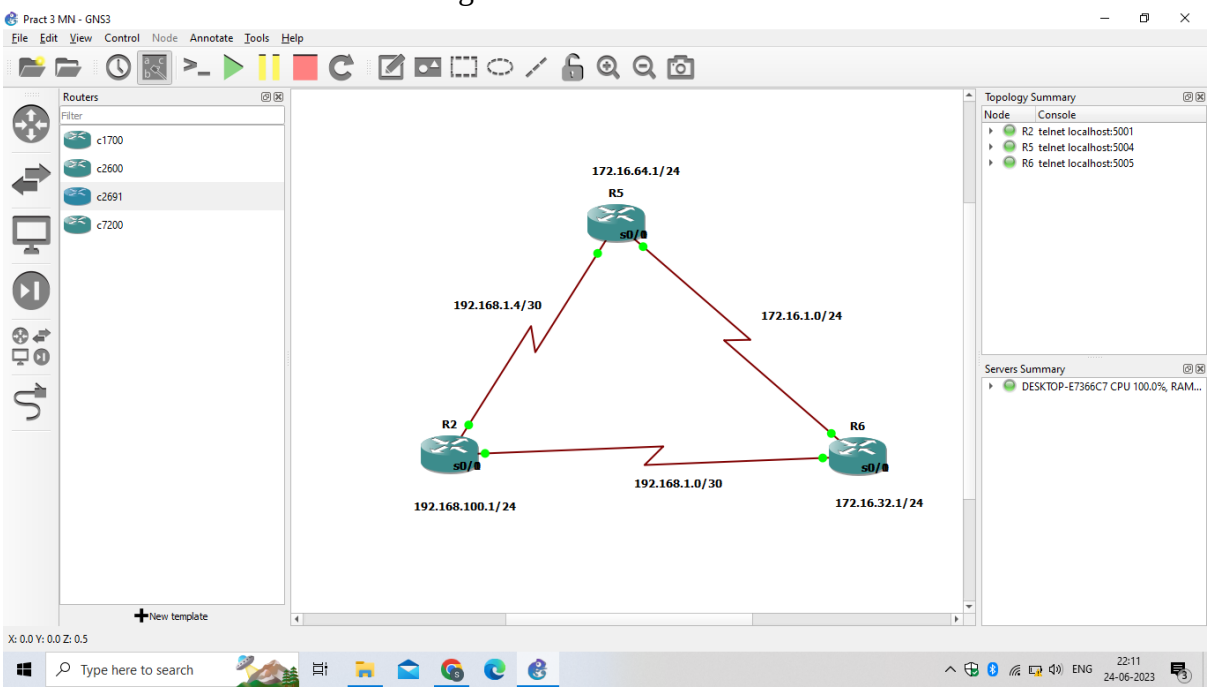


PRACTICAL NO:- 3

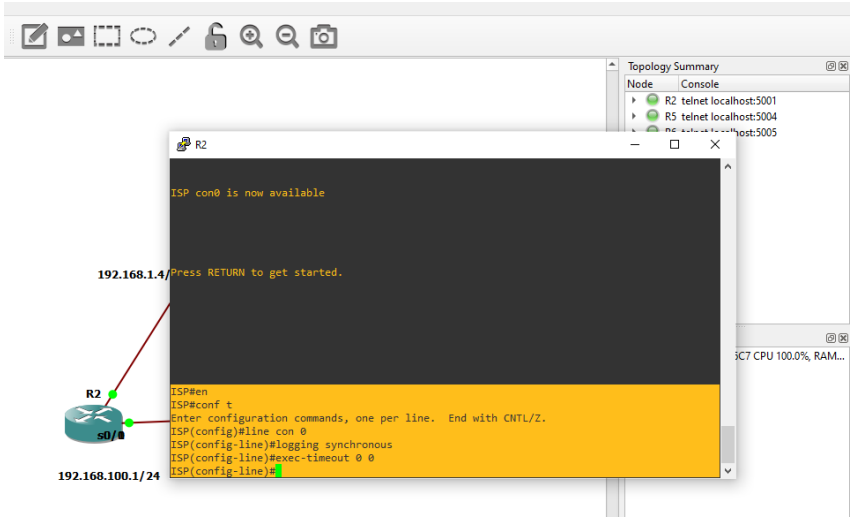
Aim:- Configuring IBGP and EBGP Sessions, Local Preference, and MED



- Click on start button for start routing connection



Step 0: Suggested starting configurations



Step 1: Configure interface addresses.

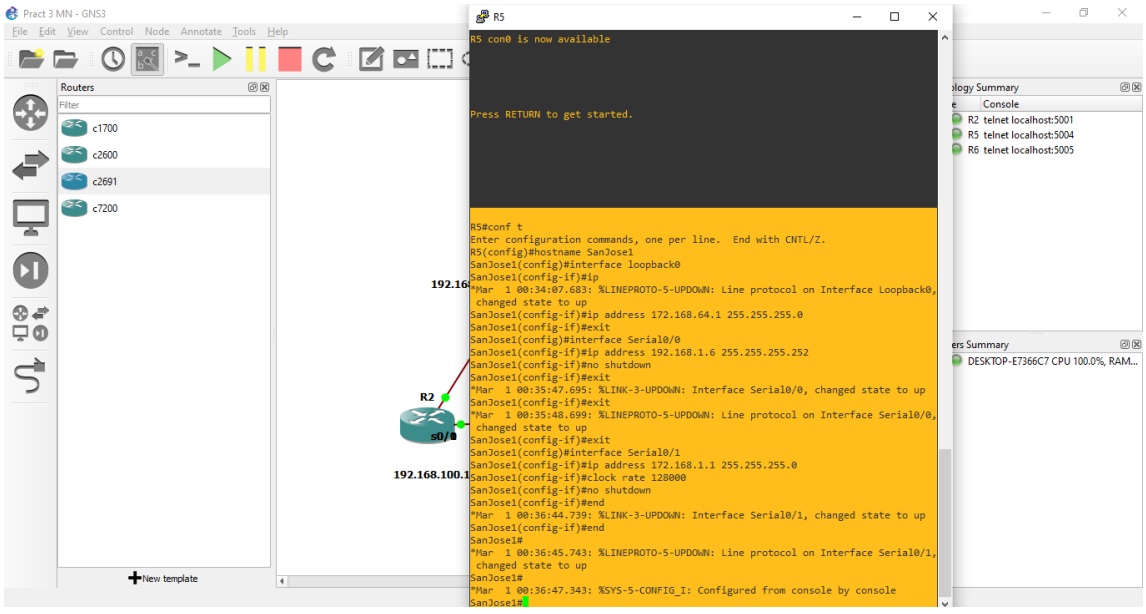
a. Using the addressing scheme in the diagram, create the loopback interfaces and apply IPv4 addresses to these and the serial interfaces on ISP (R1), SanJose1 (R2), and SanJose2 (R3).

Router R1 (hostname ISP)

ISP(config)# **interface Loopback0**
ISP(config-if)# **ip address 192.168.100.1 255.255.255.0**
ISP(config-if)# **exit**
ISP(config)# **interface Serial0/0/0**
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 Serial0/0/1**
ISP(config-if)# **ip address 192.168.1.1 255.255.255.252**
ISP(config-if)# **no shutdown**
ISP(config-if)# **end**
ISP#

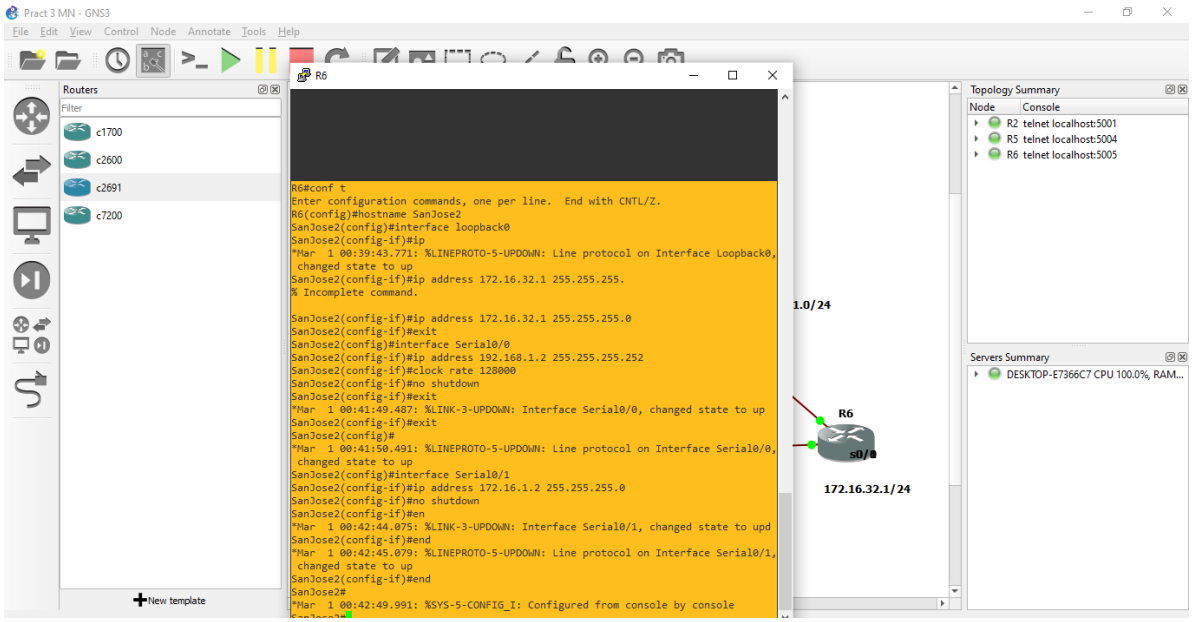
Router R2 (hostname SanJose1)

SanJose1(config)# **interface Loopback0**
SanJose1(config-if)# **ip address 172.16.64.1 255.255.255.0**
SanJose1(config-if)# **exit**
SanJose1(config)# **interface Serial0/0/0**
SanJose1(config-if)# **ip address 192.168.1.6 255.255.255.252**
SanJose1(config-if)# **no shutdown**
SanJose1(config-if)# **exit**
SanJose1(config)# **interface Serial0/0/1**
SanJose1(config-if)# **ip address 172.16.1.1 255.255.255.0**
SanJose1(config-if)# **clock rate 128000**
SanJose1(config-if)# **no shutdown**
SanJose1(config-if)# **end**



Router R3 (hostname SanJose2)

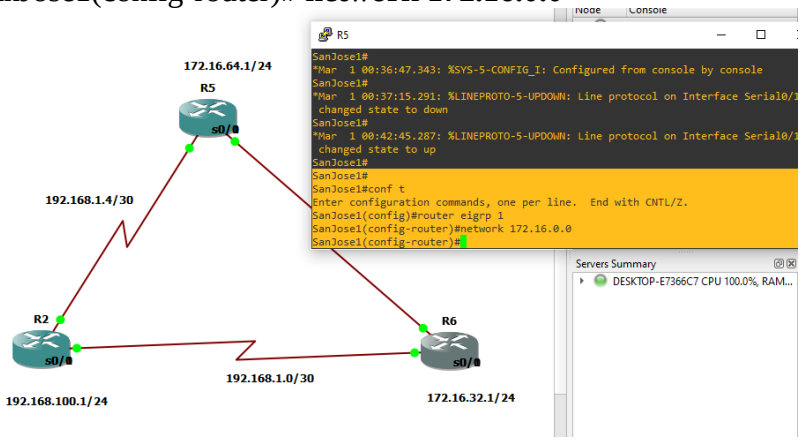
SanJose2(config)# **interface Loopback0**
SanJose2(config-if)# **ip address 172.16.32.1 255.255.255.0**
SanJose2(config-if)# **exit**
SanJose2(config)# **interface Serial0/0/0**
SanJose2(config-if)# **ip address 192.168.1.2 255.255.255.252**
SanJose2(config-if)# **clock rate 128000**
SanJose2(config-if)# **no shutdown**
SanJose2(config-if)# **exit**
SanJose2(config)# **interface Serial0/0/1**
SanJose2(config-if)# **ip address 172.16.1.2 255.255.255.0**
SanJose2(config-if)# **no shutdown**
SanJose2(config-if)# **end**
SanJose2#



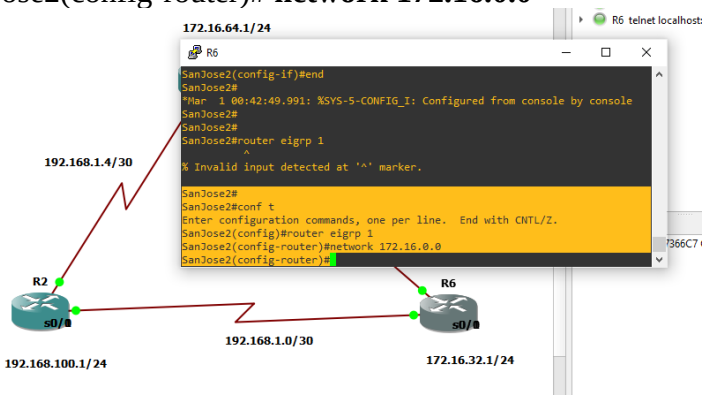
Step 2: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 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.)

SanJose1(config)# **router eigrp 1**
SanJose1(config-router)# **network 172.16.0.0**



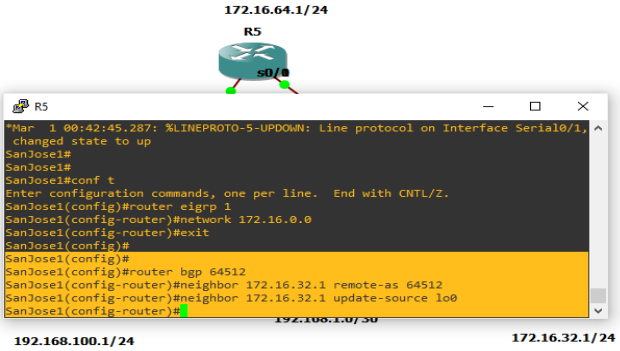
SanJose2(config)# **router eigrp 1**
SanJose2(config-router)# **network 172.16.0.0**



Step 3: Configure IBGP and verify BGP neighbors.

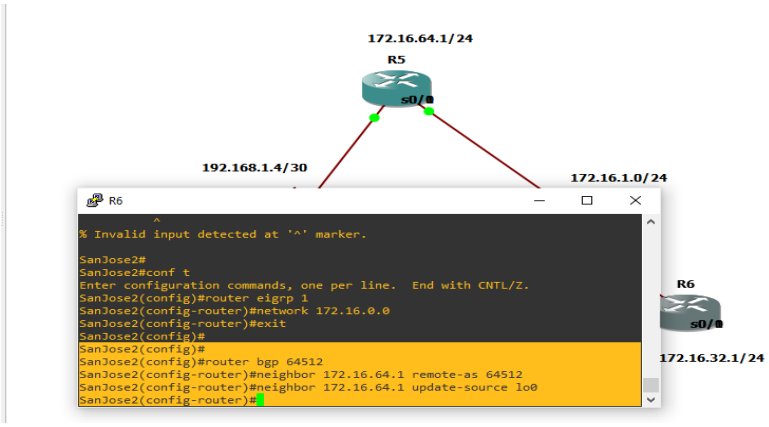
a. Configure IBGP between the SanJose1 and SanJose2 routers. On the SanJose1 router, enter the following configuration.

SanJose1(config)# **router bgp 64512**
SanJose1(config-router)# **neighbor 172.16.32.1 remote-as 64512**
SanJose1(config-router)# **neighbor 172.16.32.1 update-source lo0**

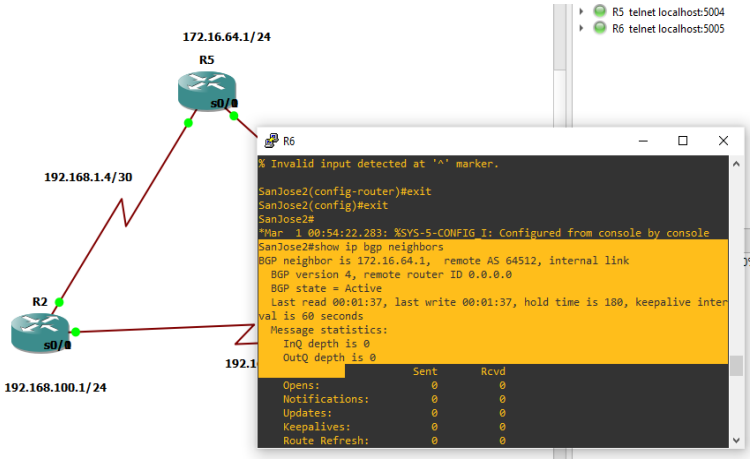


b. Complete the IBGP configuration on SanJose2 using the following commands.

SanJose2(config)# **router bgp 64512**
SanJose2(config-router)# **neighbor 172.16.64.1 remote-as 64512**
SanJose2(config-router)# **neighbor 172.16.64.1 update-source lo0**



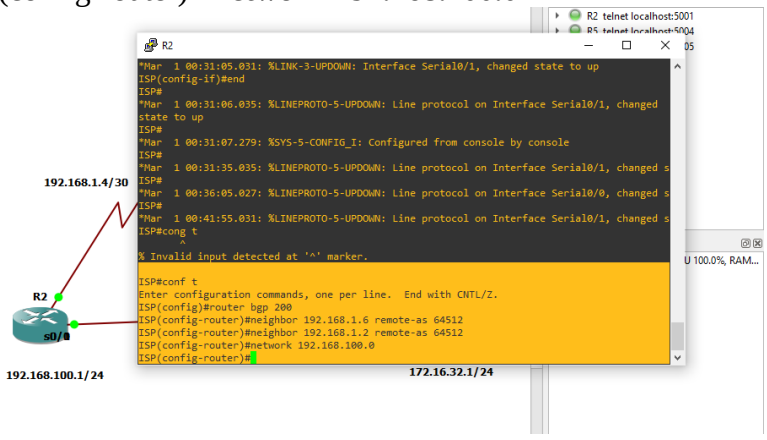
c. Verify that SanJose1 and SanJose2 become BGP neighbors by issuing the **show ip bgp neighbors**command on SanJose1. View the following partial output. If the BGP state is not established, troubleshoot the connection.



Step 4: Configure EBGP and verify BGP neighbors.

a. Configure ISP to run EBGP with SanJose1 and SanJose2. Enter the following commands on ISP.

a. ISP(config)# **router bgp 200**
b. ISP(config-router)# **neighbor 192.168.1.6 remote-as 64512**
c. ISP(config-router)# **neighbor 192.168.1.2 remote-as 64512**
d. ISP(config-router)# **network 192.168.100.0**

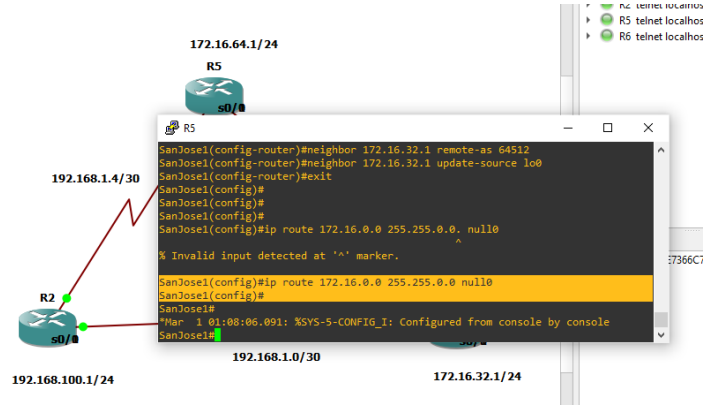


e.

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.

a. SanJose1(config)# **ip route 172.16.0.0 255.255.0.0 null0**

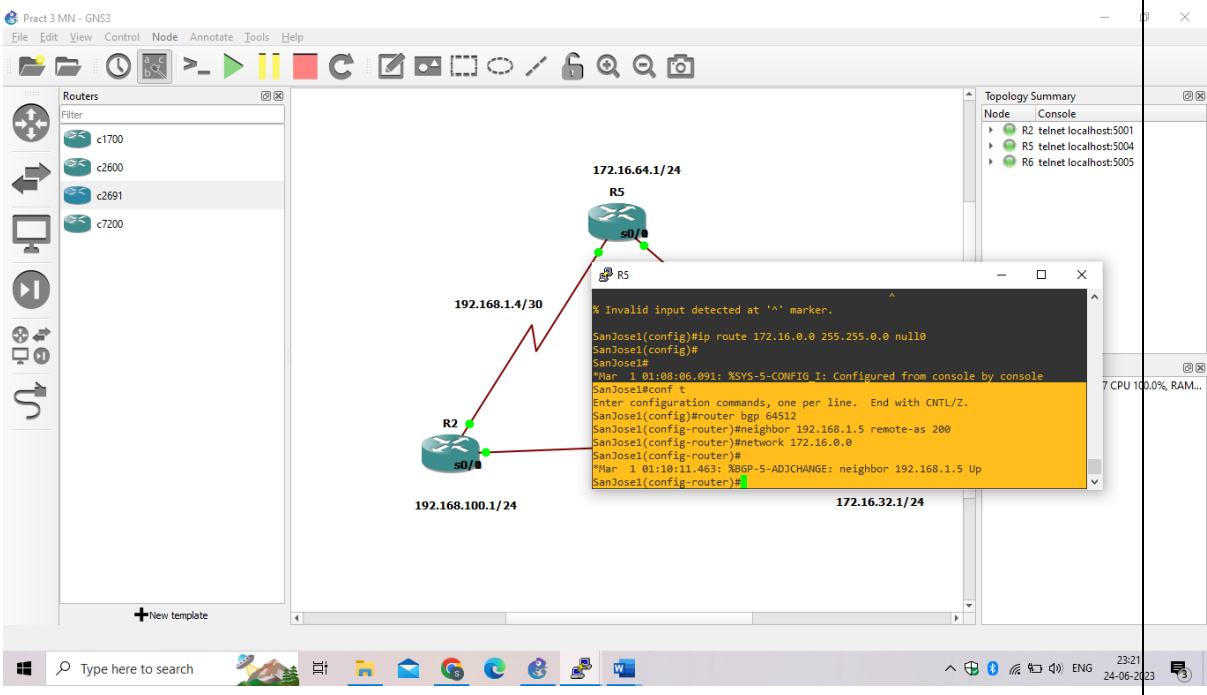
b.



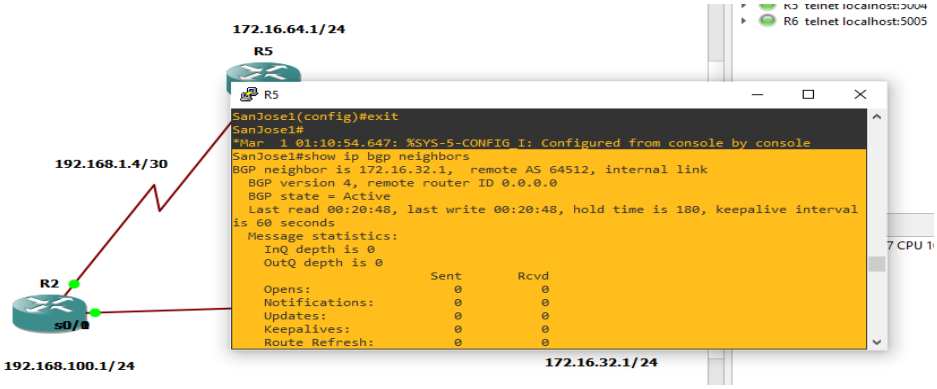
c. Configure SanJose1 as an EBGP peer to ISP.

a. SanJose1(config)# **router bgp 64512**
b. SanJose1(config-router)# **neighbor 192.168.1.5 remote-as 200**
c. SanJose1(config-router)# **network 172.16.0.0**

d.

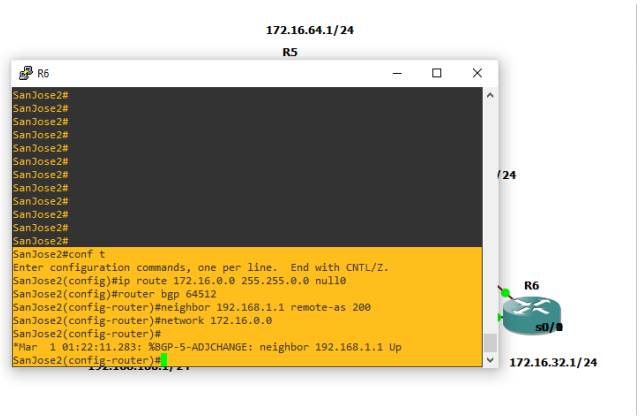


d. Use the **show ip bgp neighbors** command to verify that SanJose1 and ISP have reached the established state. Troubleshoot if necessary.



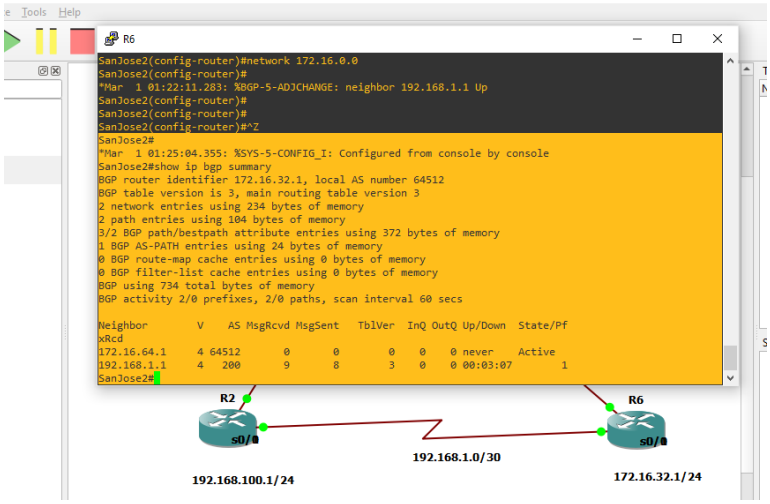
e. Configure a discard static route for 172.16.0.0/16 on SanJose2 and as an EBGP peer to ISP.

SanJose2(config)# **ip route 172.16.0.0 255.255.0.0 null0**
SanJose2(config)# **router bgp 64512**
SanJose2(config-router)# **neighbor 192.168.1.1 remote-as 200**
SanJose2(config-router)# **network 172.16.0.0**



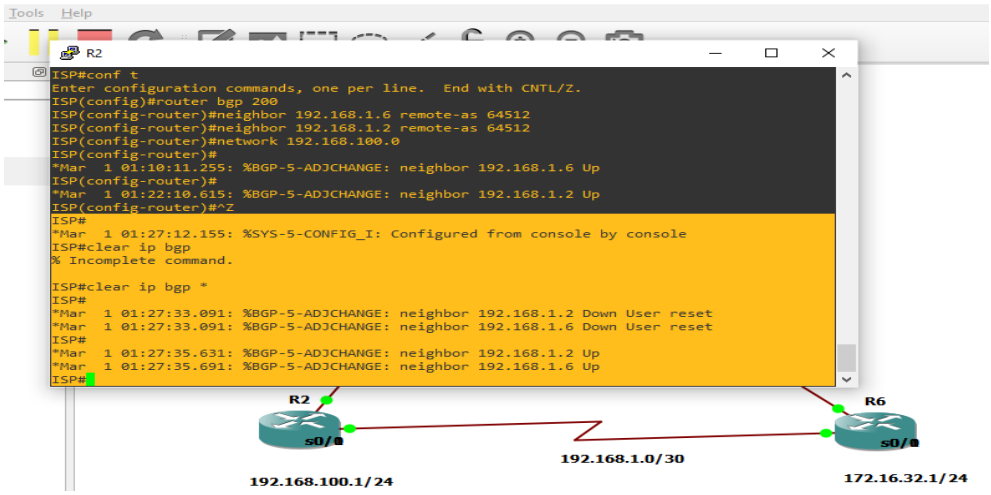
Step 5: View BGP summary output.

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

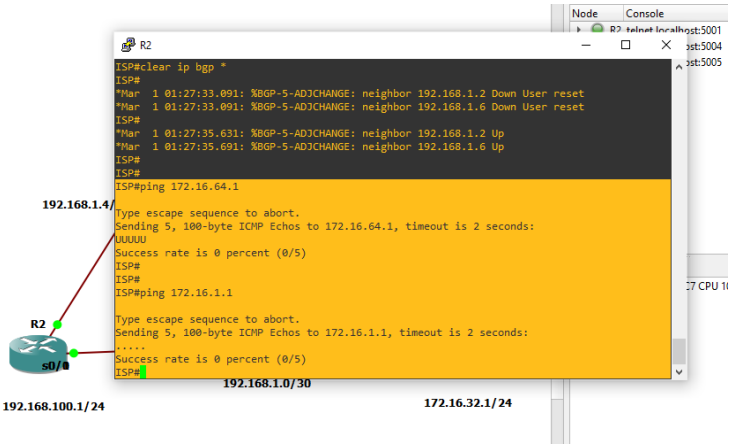


Step 6: Verify which path the traffic takes.

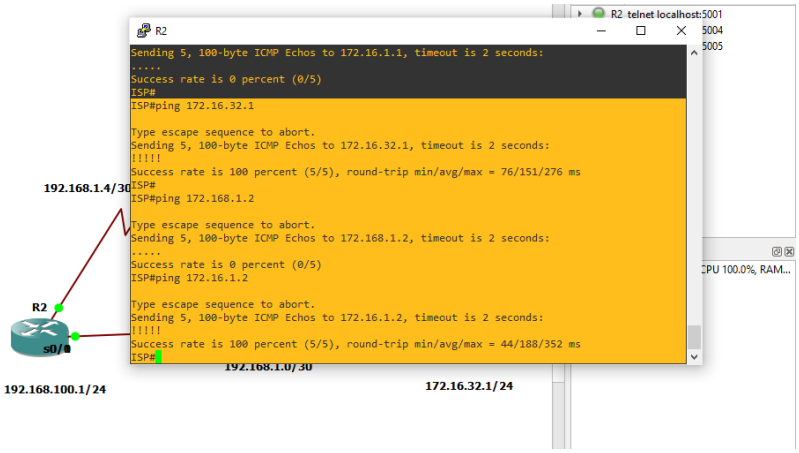
a. Clear the IP BGP conversation with the **clear ip bgp *** command on ISP. Wait for the conversations to reestablish with each SanJose router.



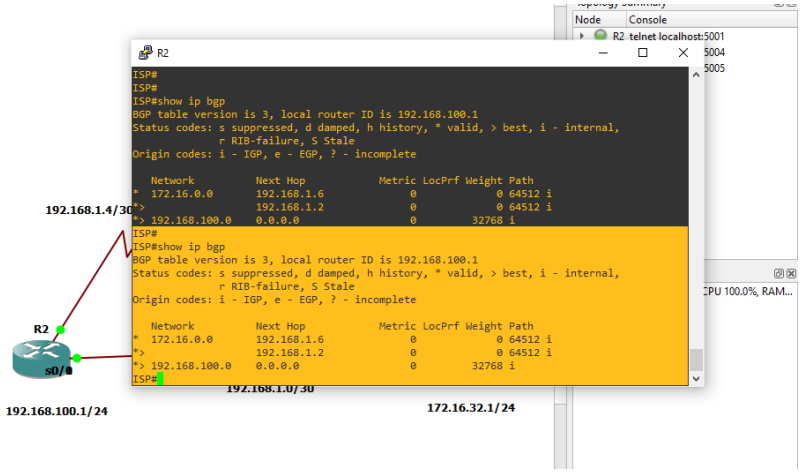
b. Test whether ISP can ping the loopback 0 address of 172.16.64.1 on SanJose1 and the serial link between SanJose1 and SanJose2, 172.16.1.1.



c. Now ping from ISP to the loopback 0 address of 172.16.32.1 on SanJose2 and the serial link between SanJose1 and SanJose2, 172.16.1.2.

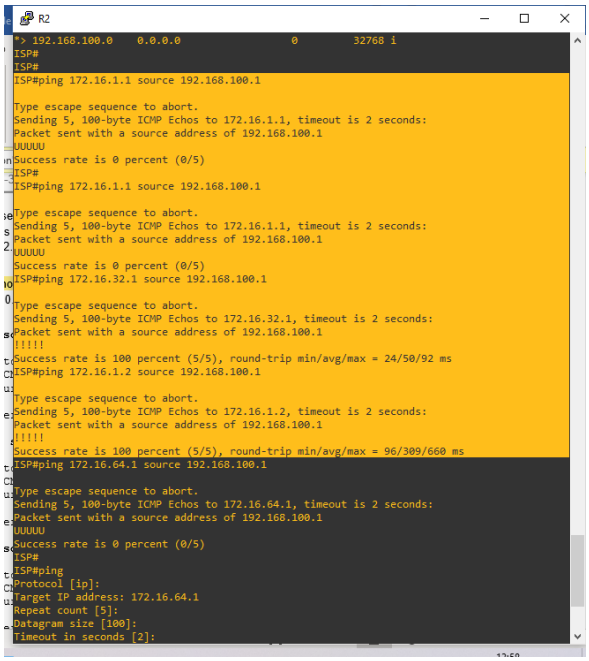


d. Issue the **show ip bgp**command on ISP to verify BGP routes and metrics.

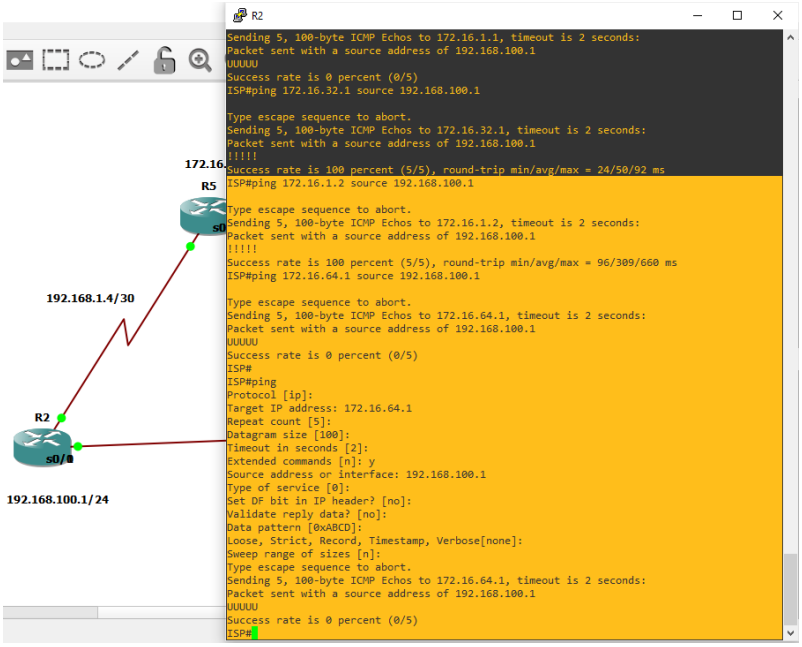


e. At this point, the ISP router should be able to get to each network connected to SanJose1 and SanJose2 from the loopback address 192.168.100.1. Use the extended **ping** command and specify the source address of ISP Lo0 to test.

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**



f. ISP# **ping**

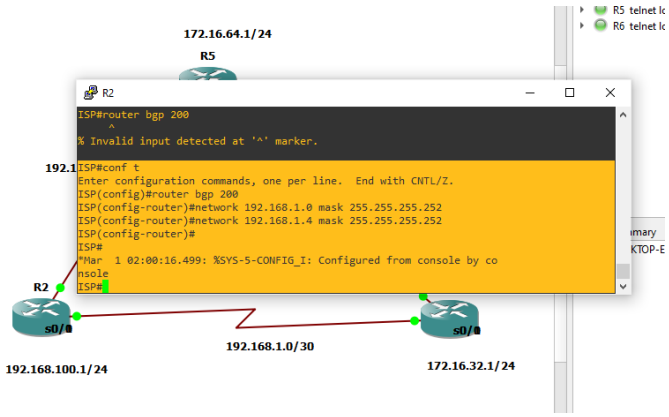


Step 7: Configure the BGP next-hop-self feature.

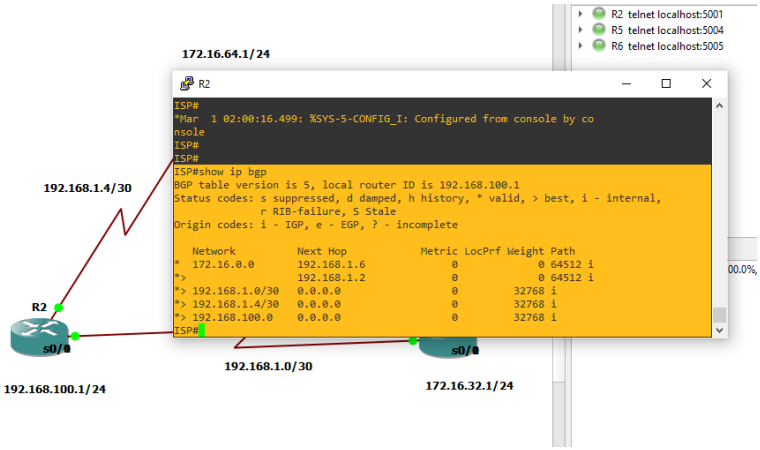
SanJose1 is unaware of the link between ISP and SanJose2, and SanJose2 is unaware of the link between ISP and SanJose1. Before ISP can successfully ping all the internal serial interfaces of AS 64512, these serial links should be advertised via BGP on the ISP router. This can also be resolved via EIGRP on each SanJose router. One method is for ISP to advertise these links.

a. Issue the following commands on the ISP router.

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**

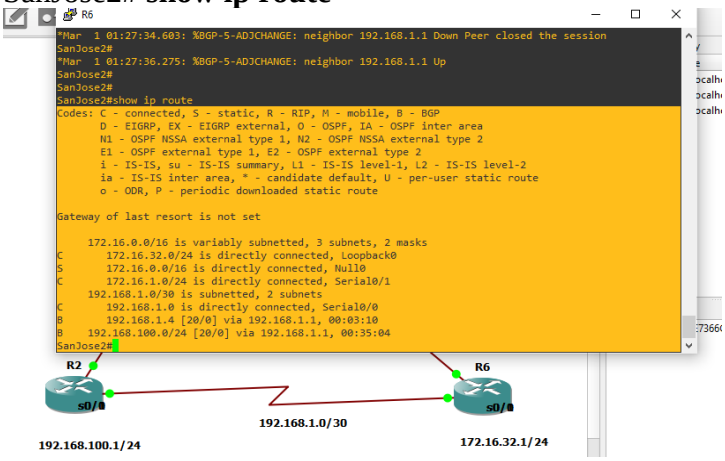


b. Issue the **show ip bgp** command to verify that the ISP is correctly injecting its own WAN links into BGP.



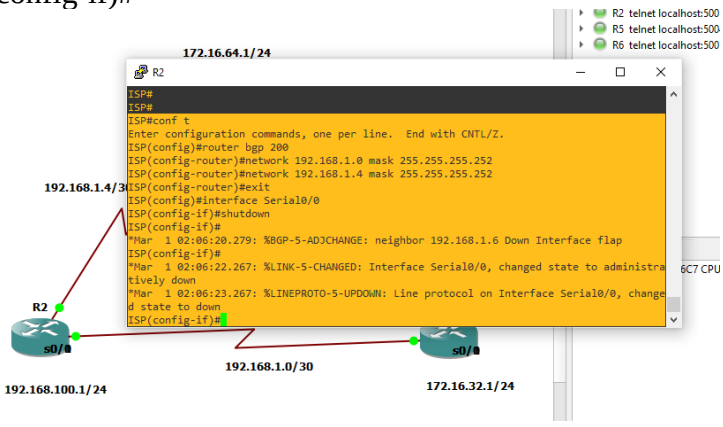
c. Verify on SanJose1 and SanJose2 that the opposite WAN link is included in the routing table. The output from SanJose2 is as follows.

SanJose2# **show ip route**



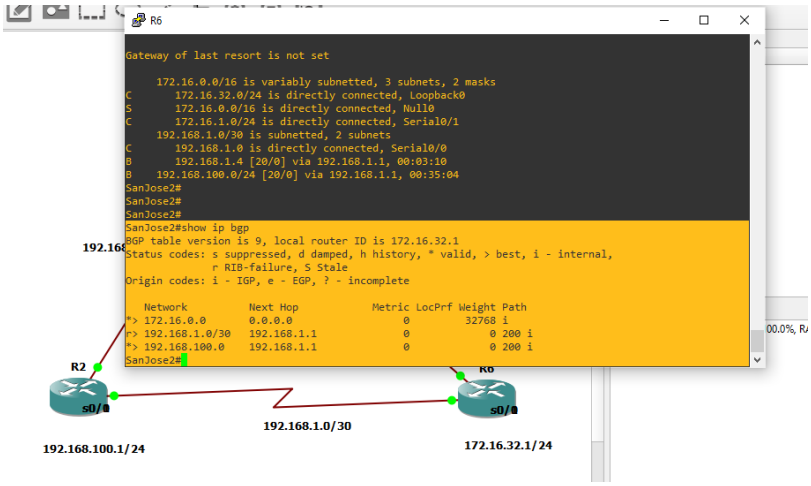
d. To better understand the **next-hop-self** command we will remove ISP advertising its two WAN links and shutdown the WAN link between ISP and SanJose2. The only possible path from SanJose2 to ISP's 192.168.100.0/24 is through SanJose1.

ISP(config)# **router bgp 200**
ISP(config-router)# **nonetwork 192.168.1.0 mask 255.255.255.252**
ISP(config-router)# **nonetwork 192.168.1.4 mask 255.255.255.252**
ISP(config-router)# **exit**
ISP(config)# **interface serial 0/0/1**
ISP(config-if)# **shutdown**
ISP(config-if)#

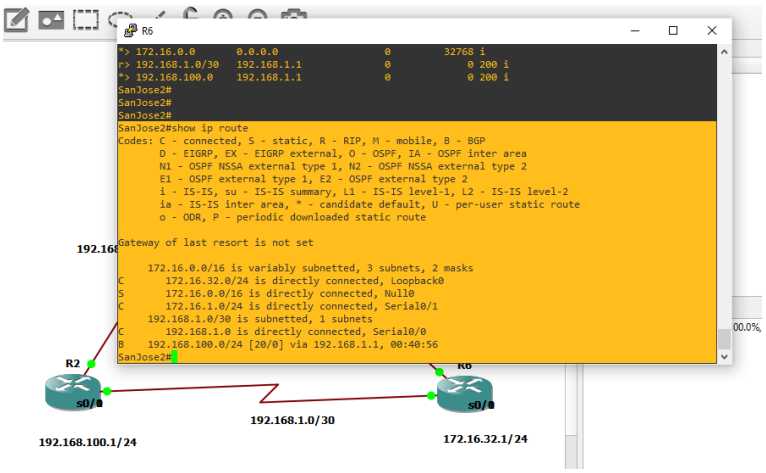


e. Display SanJose2's BGP table using the **show ip bgp** command and the IPv4 routing table with **show ip route**.

SanJose2# **show ip bgp**

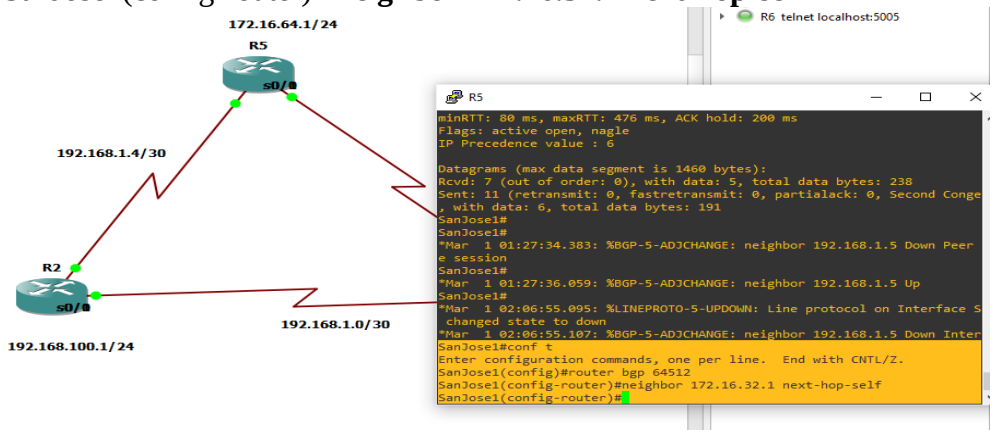


SanJose2# **show ip route**

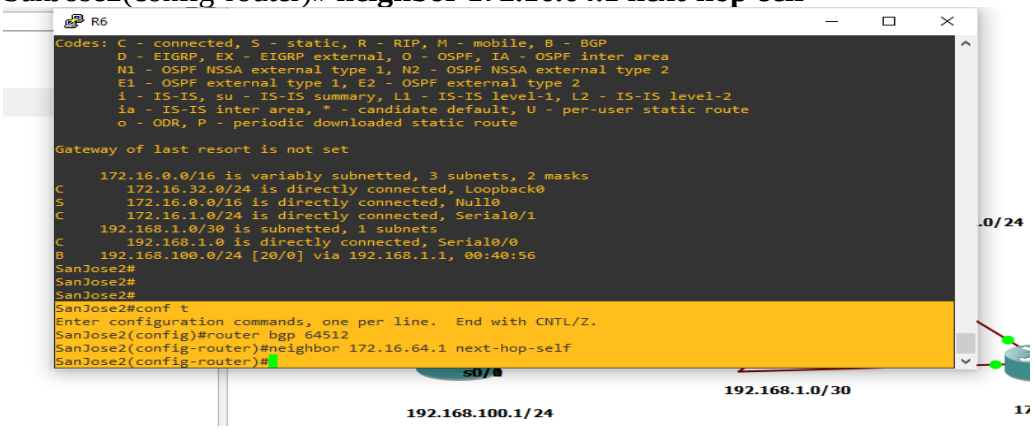


f. Issue the **next-hop-self** command on SanJose1 and SanJose2 to advertise themselves as the next hop to their IBGP peer.

SanJose1(config)# **router bgp 64512**
SanJose1(config-router)# **neighbor 172.16.32.1 next-hop-self**

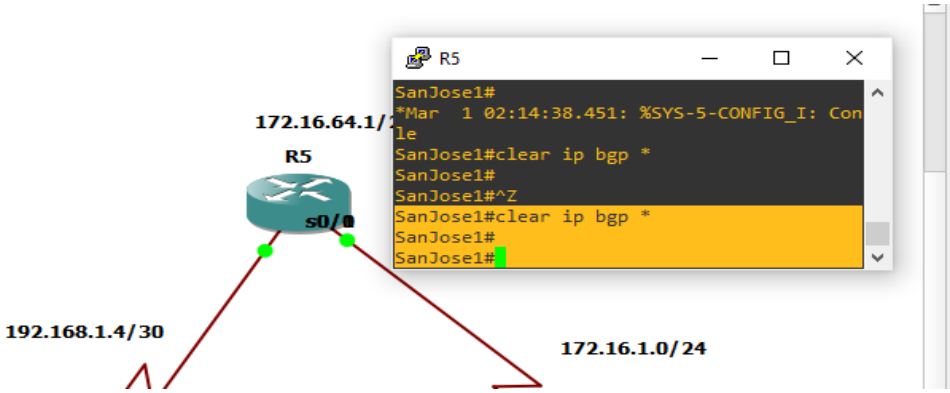


SanJose2(config)# **router bgp 64512**
SanJose2(config-router)# **neighbor 172.16.64.1 next-hop-self**

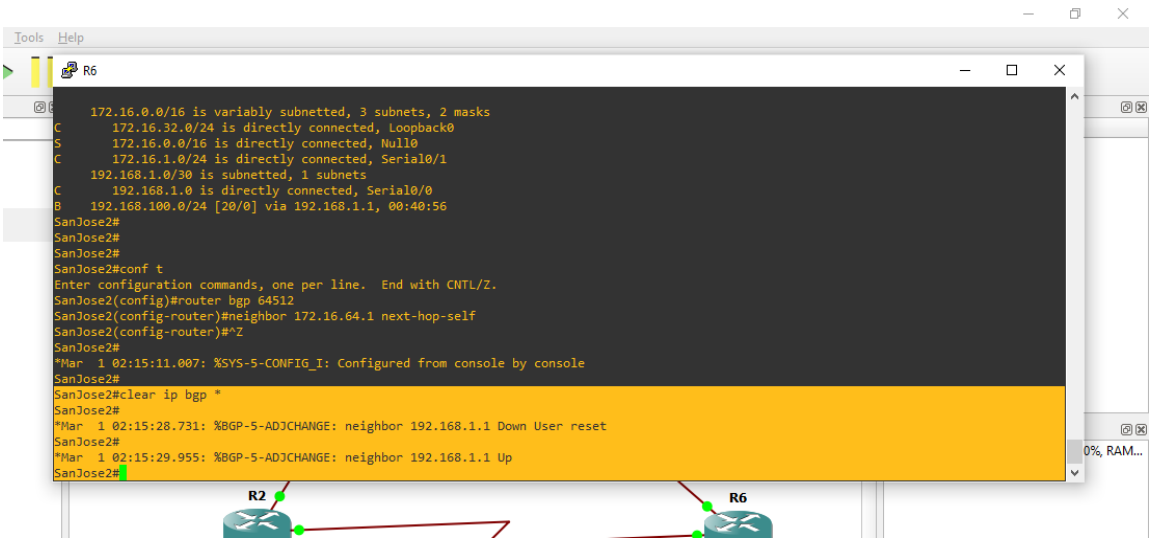


g. Reset BGP operation on either router with the **clear ip bgp *** command.

SanJose1# **clear ip bgp ***
SanJose1#

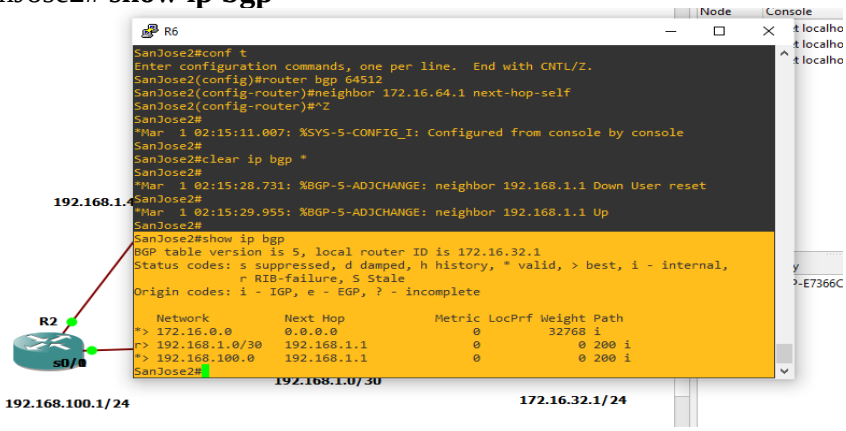


SanJose2# **clear ip bgp ***
SanJose2#



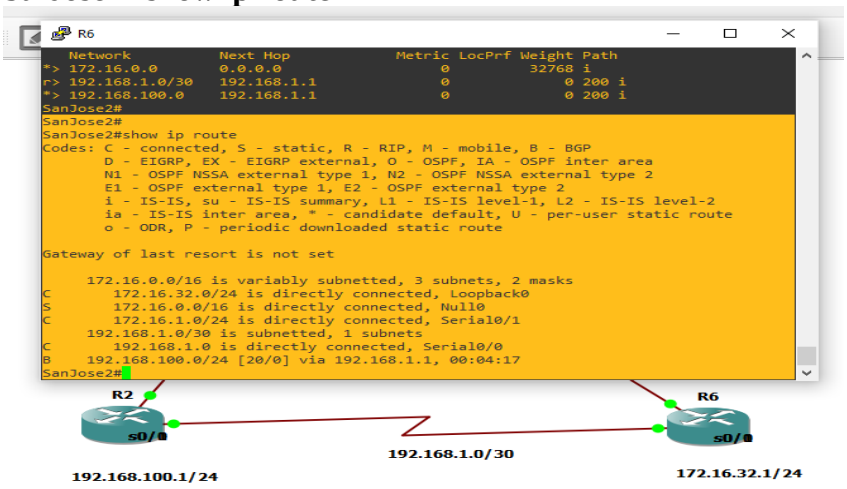
h. After the routers have returned to established BGP speakers, issue the **show ip bgp** command on SanJose2 and notice that the next hop is now SanJose1 instead of ISP.

SanJose2# **show ip bgp**



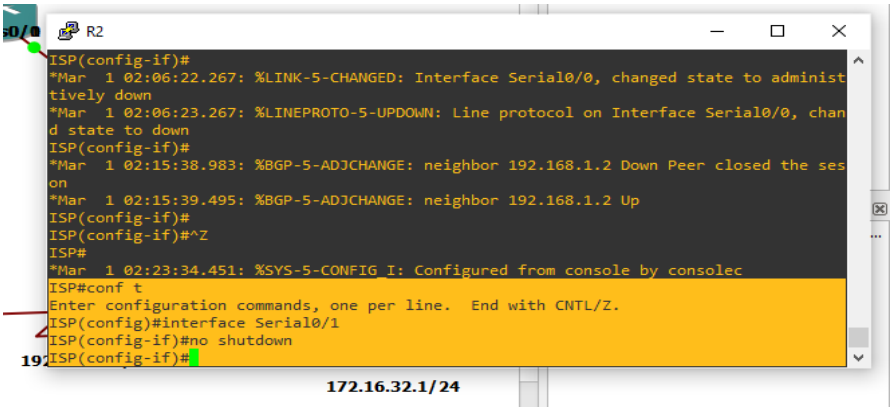
i. The **show ip route** command on SanJose2 now displays the 192.168.100.0/24 network because SanJose1 is the next hop, 172.16.64.1, which is reachable from SanJose2.

SanJose2# **show ip route**

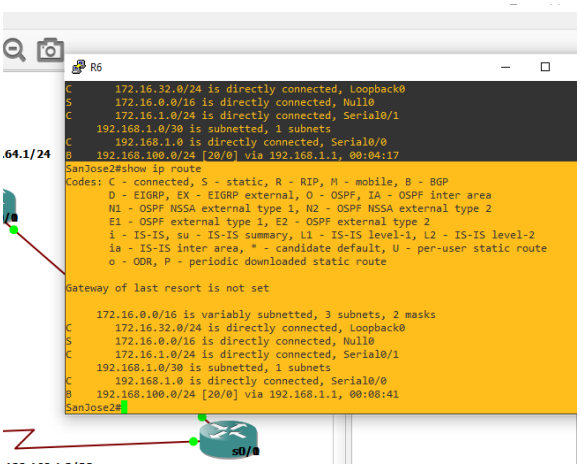


j. Before configuring the next BGP attribute, restore the WAN link between ISP and SanJose3. This will change the BGP table and routing table on both routers. For example, SanJose2's routing table shows 192.168.100.0/24 will now have a better path through ISP.

ISP(config)# **interface serial 0/0/1**
ISP(config-if)# **noshutdown**
ISP(config-if)#



SanJose2# **show ip route**

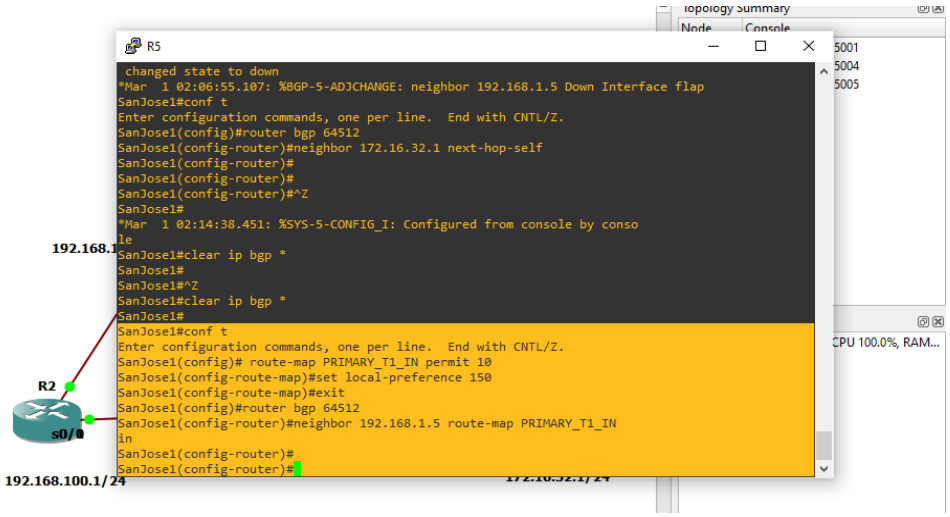


Step 8: Set BGP local preference.

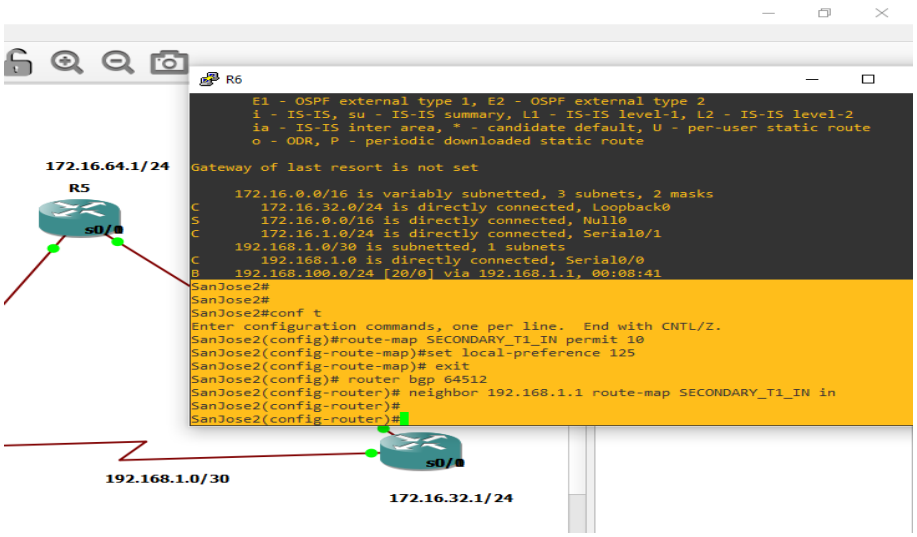
At this point, everything looks good, with the exception of default routes, the outbound flow of data, and inbound packet flow.

a. Because the local preference value is shared between IBGP neighbors, configure a simple route map that references the local preference value on SanJose1 and SanJose2. This policy adjusts outbound traffic to prefer the link off the SanJose1 router instead of the metered T1 off SanJose2.

SanJose1(config)# **route-map PRIMARY_T1_IN permit 10**
SanJose1(config-route-map)# **set local-preference 150**
SanJose1(config-route-map)#**exit**
SanJose1(config)# **router bgp 64512**
SanJose1(config-router)# **neighbor 192.168.1.5 route-map PRIMARY_T1_IN in**

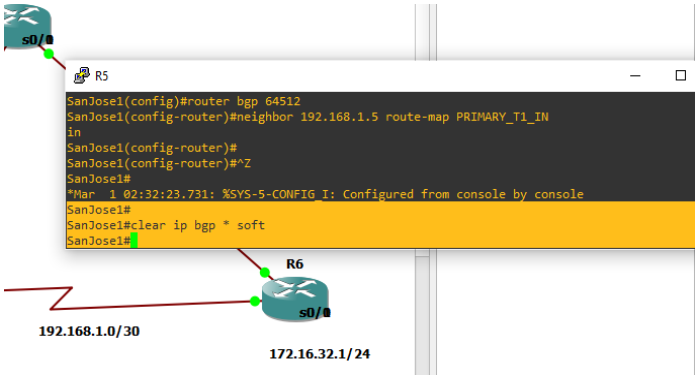


SanJose2(config)# **route-map SECONDARY_T1_IN permit 10**
SanJose2(config-route-map)# **set local-preference 125**
SanJose1(config-route-map)#**exit**
SanJose2(config)# **router bgp 64512**
SanJose2(config-router)# **neighbor 192.168.1.1 route-map SECONDARY_T1_IN in**

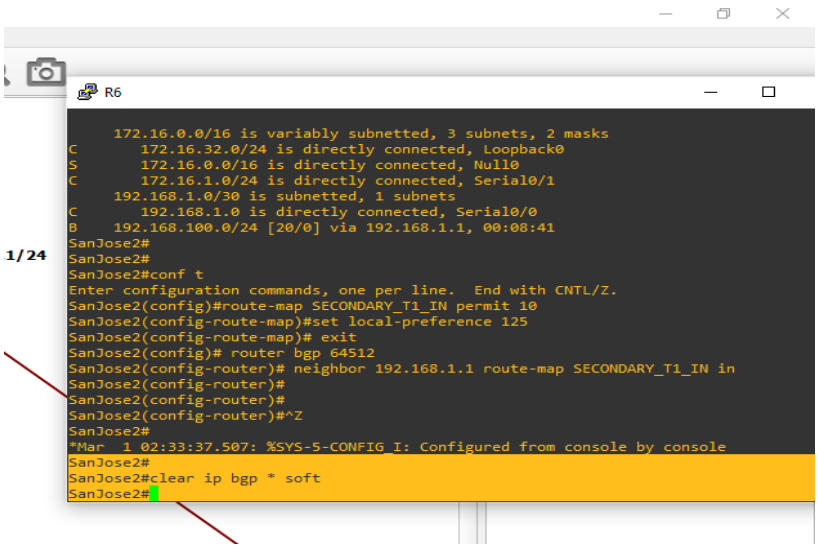


b. Use the **clear ip bgp * soft** command after configuring this new policy. When the conversations have been reestablished, issue the **show ip bgp** command on SanJose1 and SanJose2.

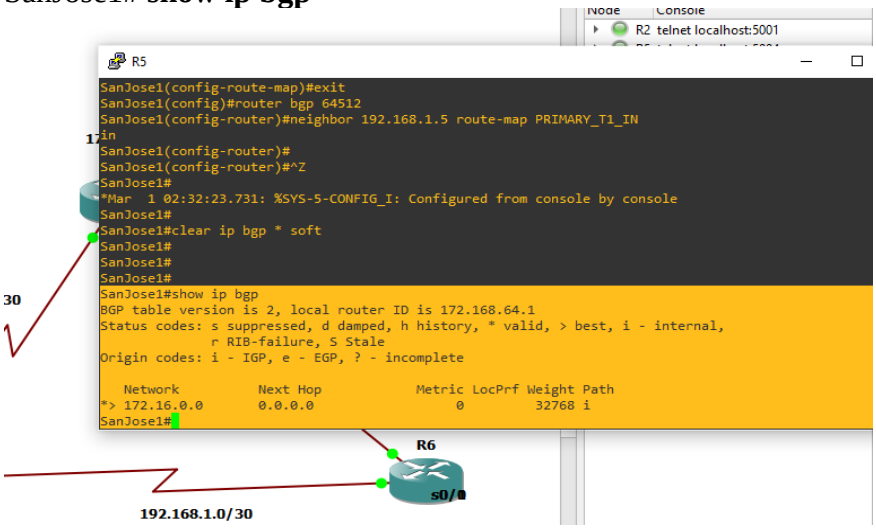
SanJose1# **clear ip bgp * soft**



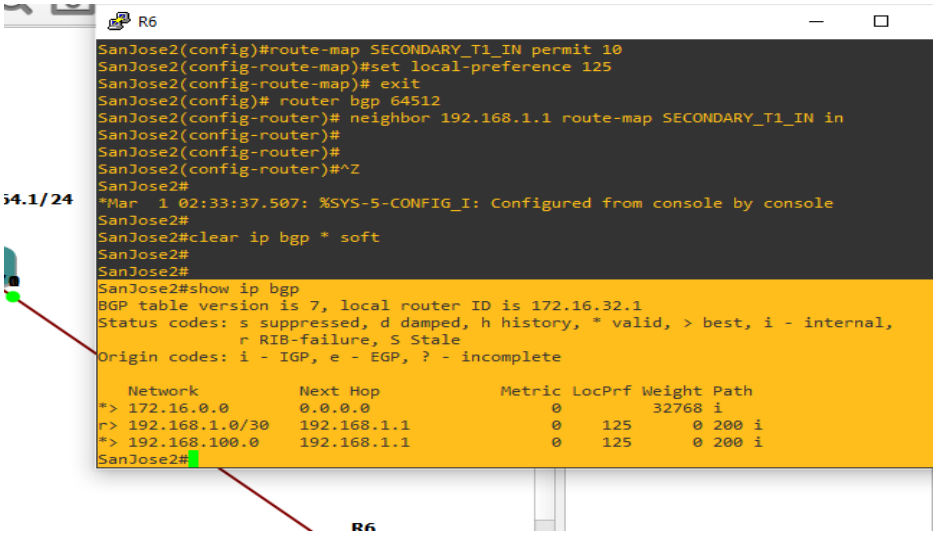
SanJose2# **clear ip bgp * soft**



SanJose1# **show ip bgp**



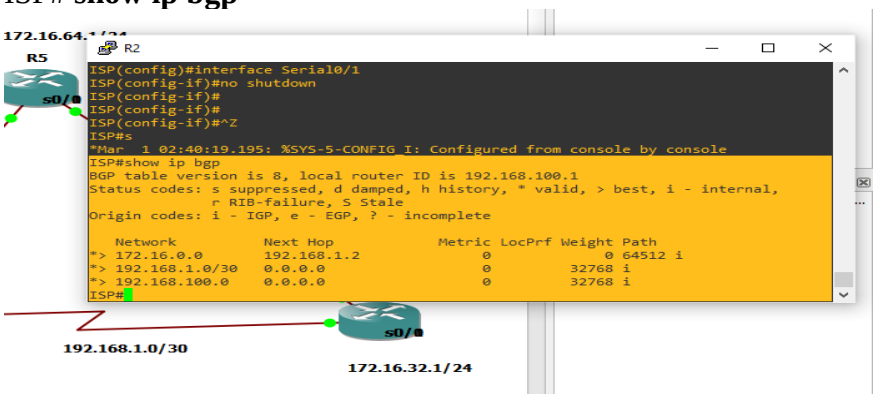
SanJose2# **show ip bgp**



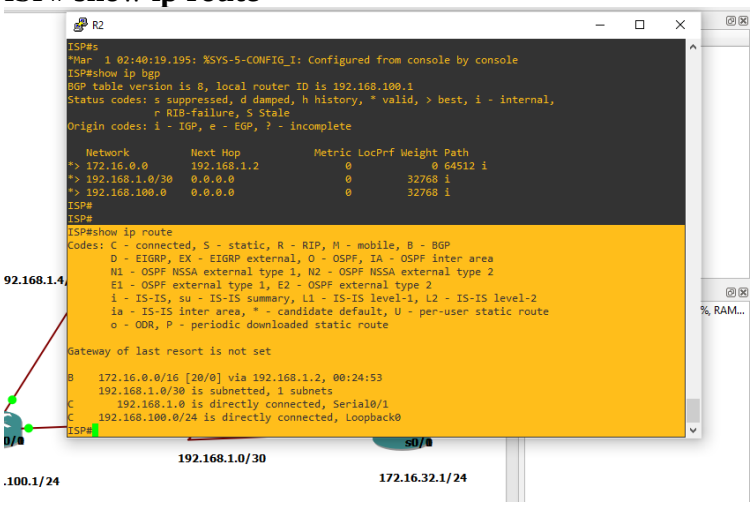
Step 9: Set BGP MED.

a. In the previous step we saw that SanJose1 and SanJose2 will route traffic for 192.168.100.0/24 using the link between SanJose1 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# **show ip bgp**



ISP# **show ip route**



SanJose2# **ping**

R6

```
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
r> 192.168.1.0/30   192.168.1.1              0    125      0 200 i
*> 192.168.100.0    192.168.1.1              0    125      0 200 i
SanJose2#
SanJose2#
SanJose2#
SanJose2#
SanJose2#
SanJose2#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 0 (196 ms).  Received packet has options
 Total option bytes= 40, padded length=40
```

R6

```
Reply to request 1 (128 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 2 (56 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 3 (152 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 4 (180 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
```

R6

```
 End of list

Reply to request 2 (56 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 3 (152 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 4 (180 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (192.168.1.2)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (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 = 56/142/196 ms
SanJose2#y
SanJose2#
```

ISP# **show ip bgp**

Topology Summary

R2

```
ISP#
ISP#
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

B    172.16.0.0/16 [20/0] via 192.168.1.2, 00:24:53
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/1
C    192.168.100.0/24 is directly connected, Loopback0
ISP#
ISP#
ISP#show ip bgp
BGP table version is 8, 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
*> 192.168.1.0/30   0.0.0.0                  0         32768 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
ISP#
```

.168.1.0/30

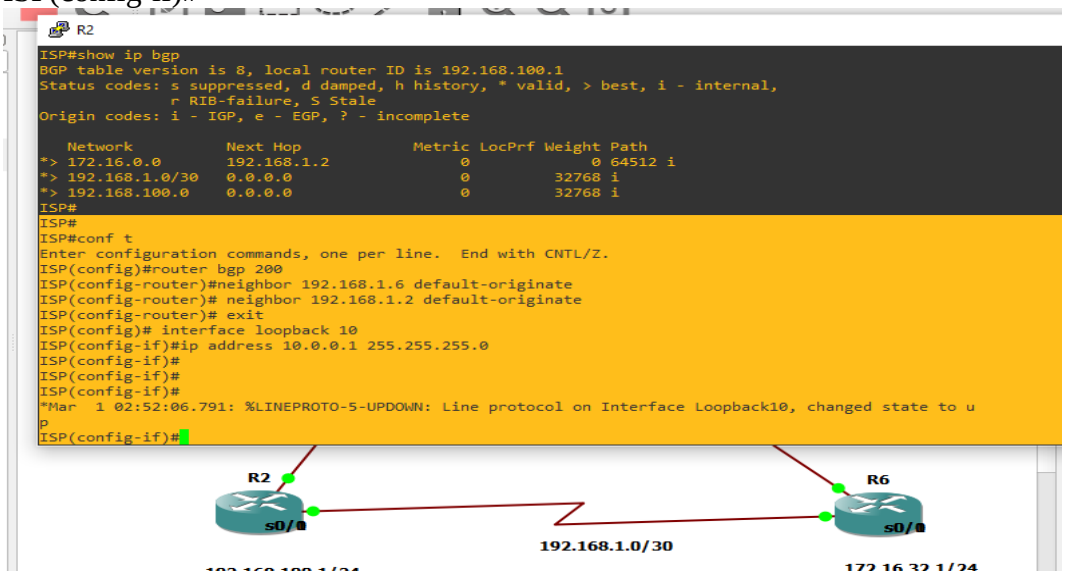
172.16.32.1/24

Step 10: 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.

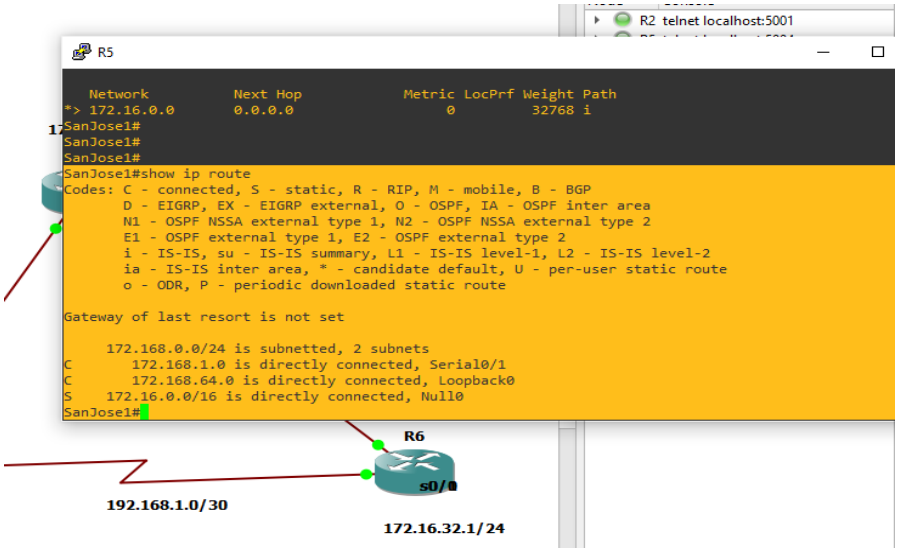
a. Configure ISP to inject a default route to both SanJose1 and SanJose2 using BGP using the **default-originate** command. This command does not require the presence of 0.0.0.0 in the ISP router. Configure the 10.0.0.0/8 network which will not be advertised using BGP. This network will be used to test the default route on SanJose1 and SanJose2.

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)#

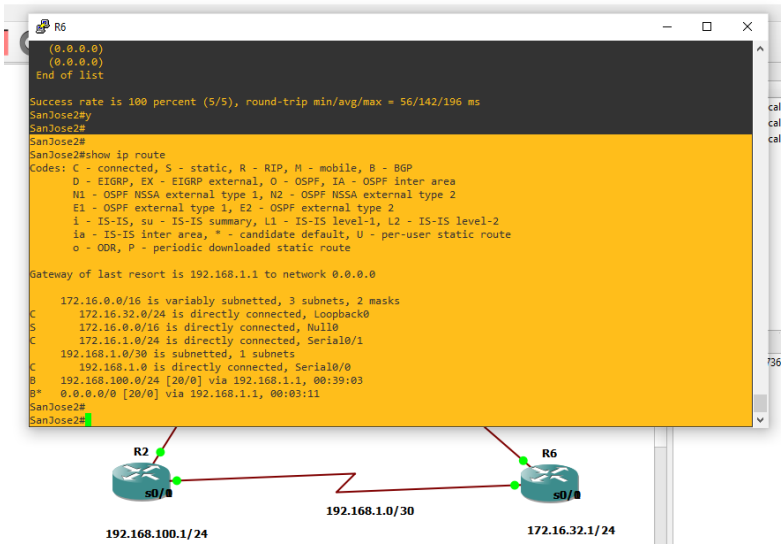


b. Verify that both routers have received the default route by examining the routing tables on SanJose1 and SanJose2. Notice that both routers prefer the route between SanJose1 and ISP

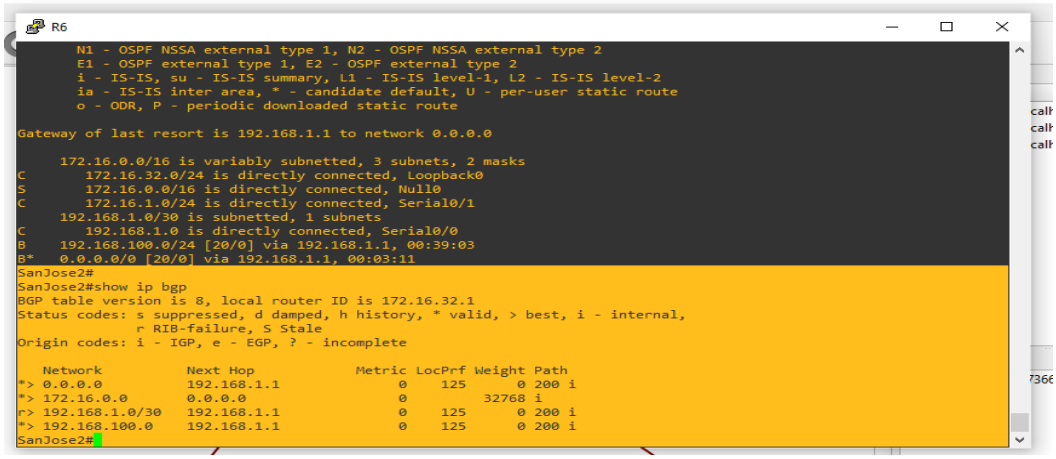
SanJose1# **show ip route**



SanJose2# **show ip route**

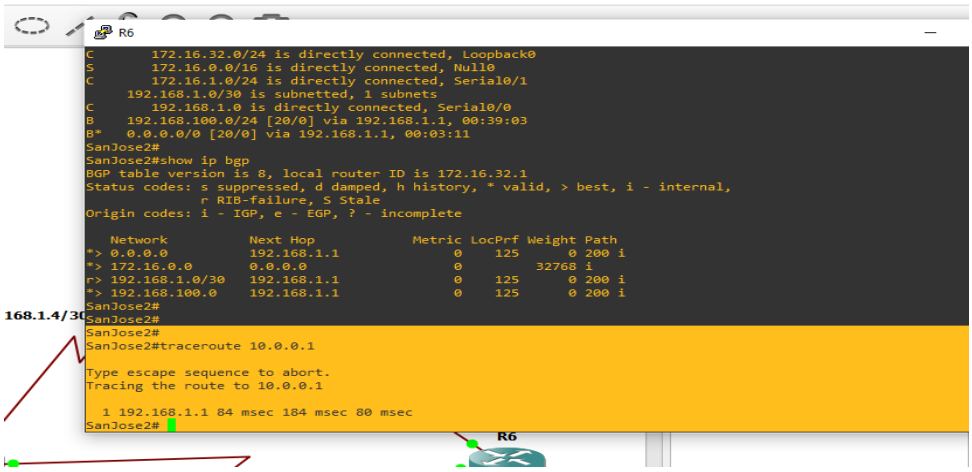


SanJose2# **show ip bgp**



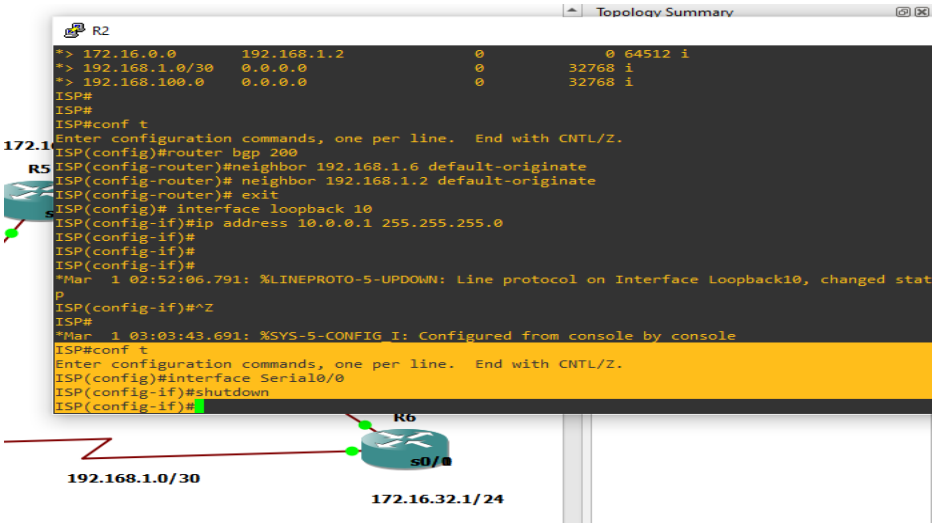
c. Using the traceroute command verify that packets to 10.0.0.1 is using the default route through SanJose1.

SanJose2# **traceroute 10.0.0.1**



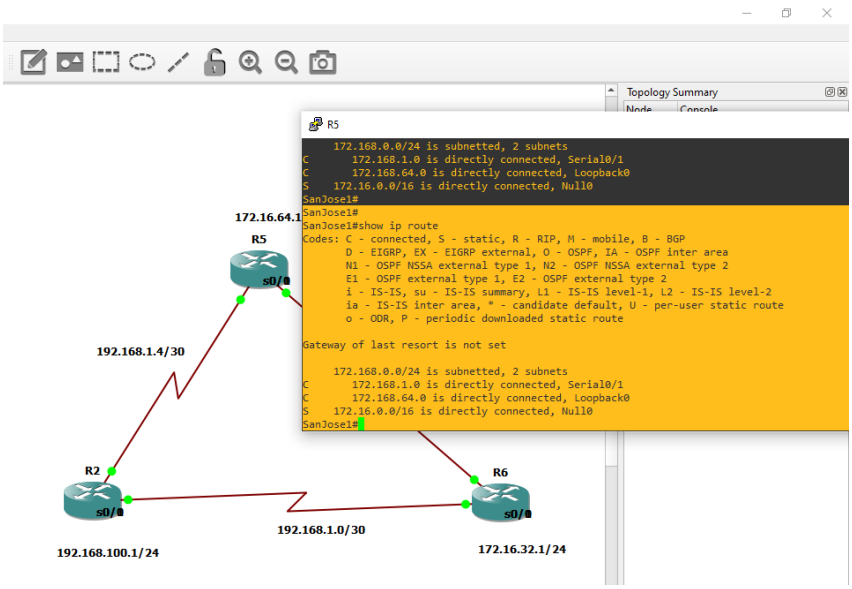
d. Next, test how BGP adapts to using a different default route when the path between SanJose1 and ISP goes down.

ISP(config)# **interface serial 0/0/0**
ISP(config-if)# **shutdown**
ISP(config-if)#

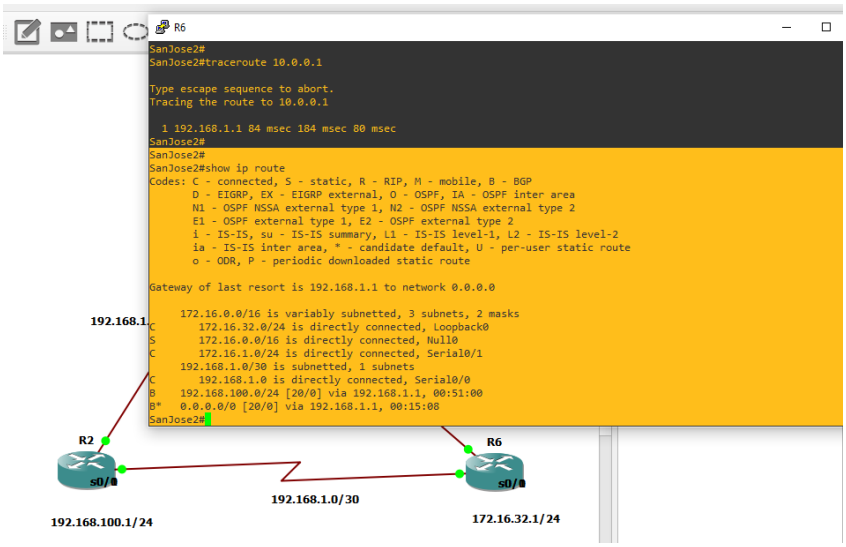


e. Verify that both routers are modified their routing tables with the default route using the path between SanJose2 and ISP.

SanJose1# **show ip route**

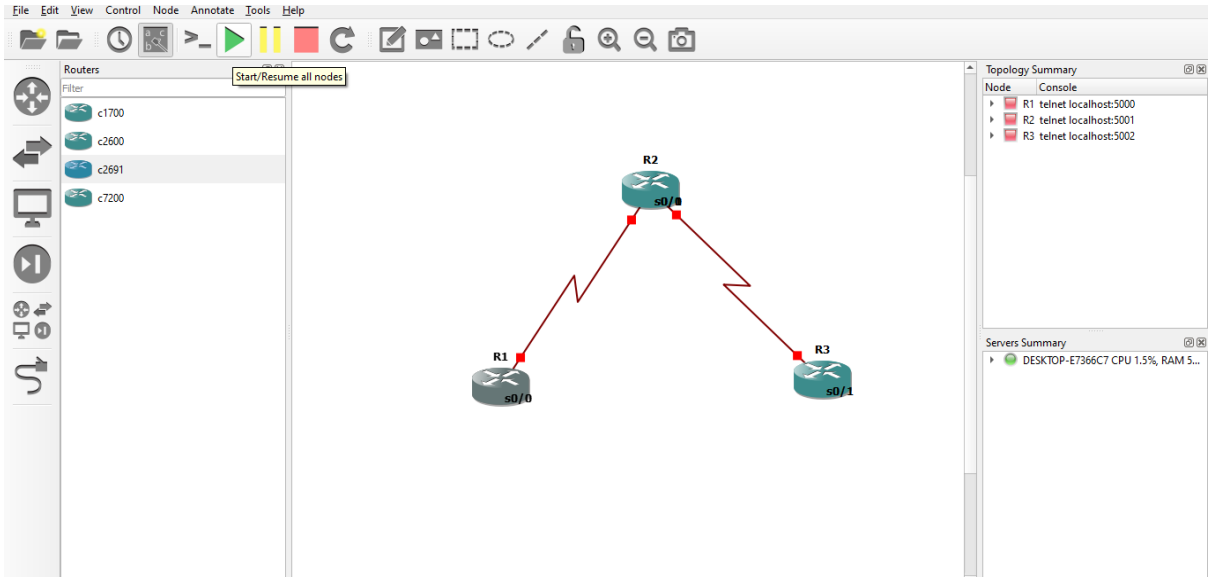


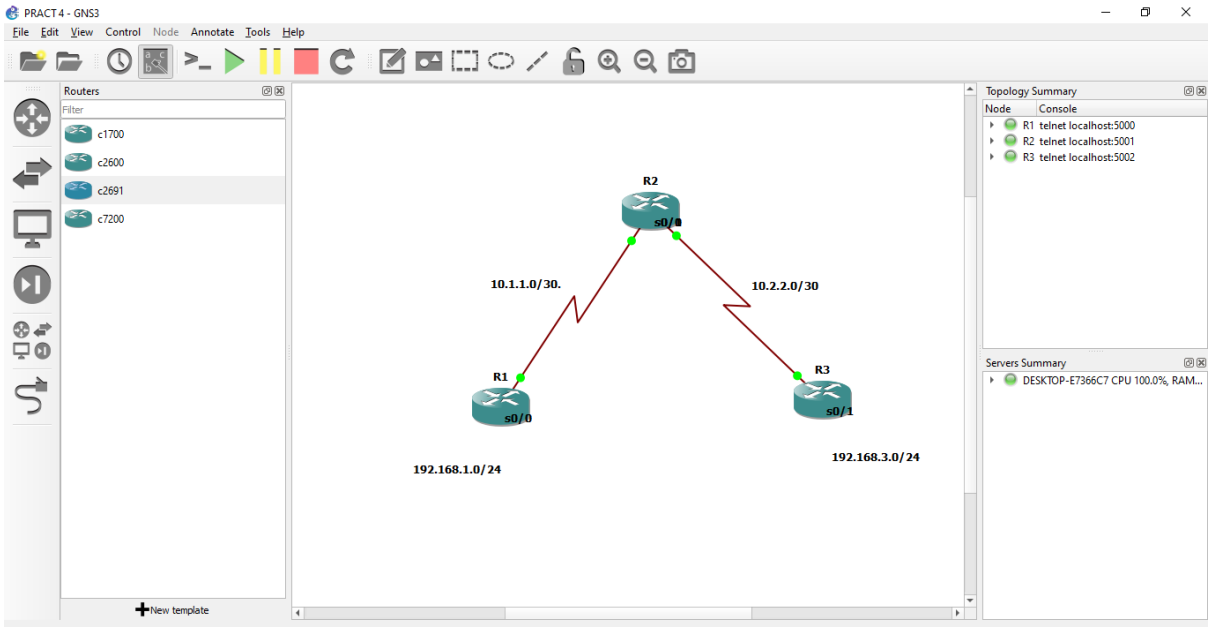
SanJose2# **show ip route**



PRACTICAL NO:- 4

Aim:- Secure the Management Plane

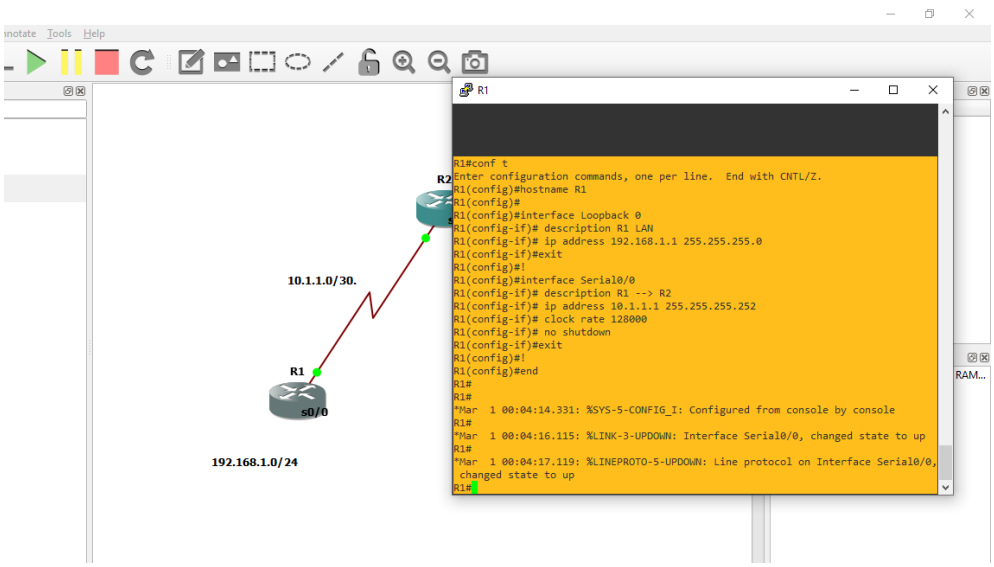




Step 1: Configure loopbacks and assign addresses.

R1

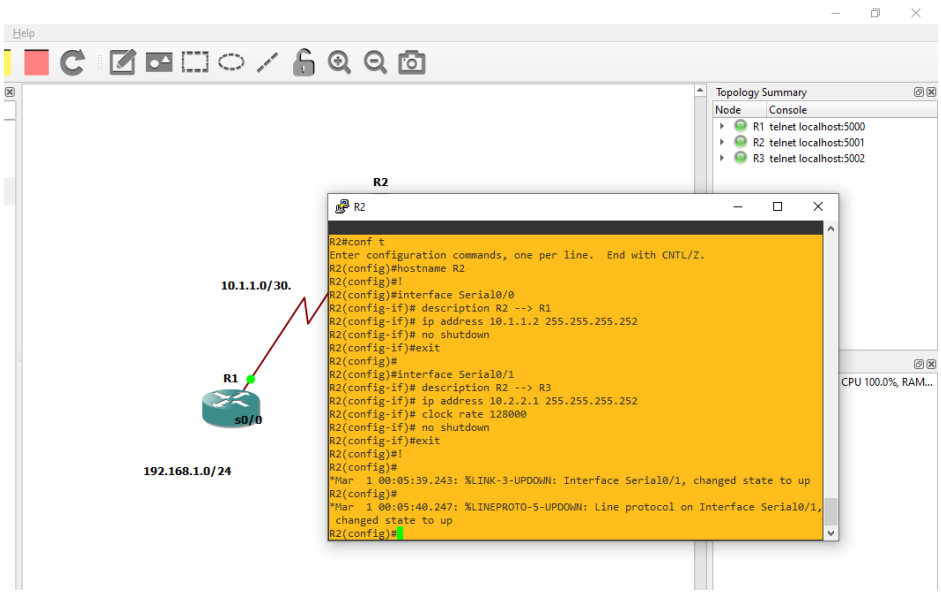
```
hostname R1

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
```



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
```

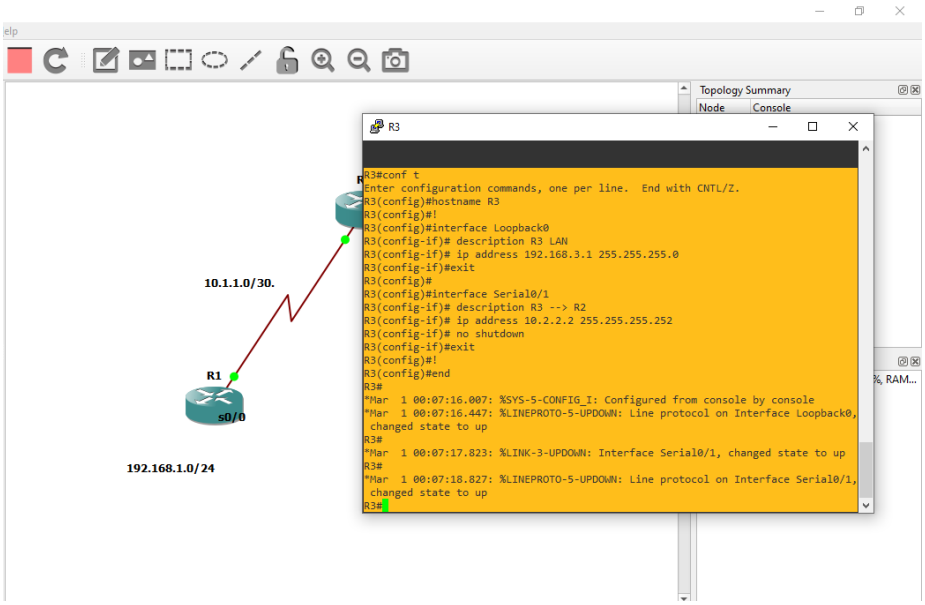
exit
!
End



R3

hostname R3
!
interface Loopback0
 description R3 LAN
 ip address 192.168.3.1 255.255.255.0
exit

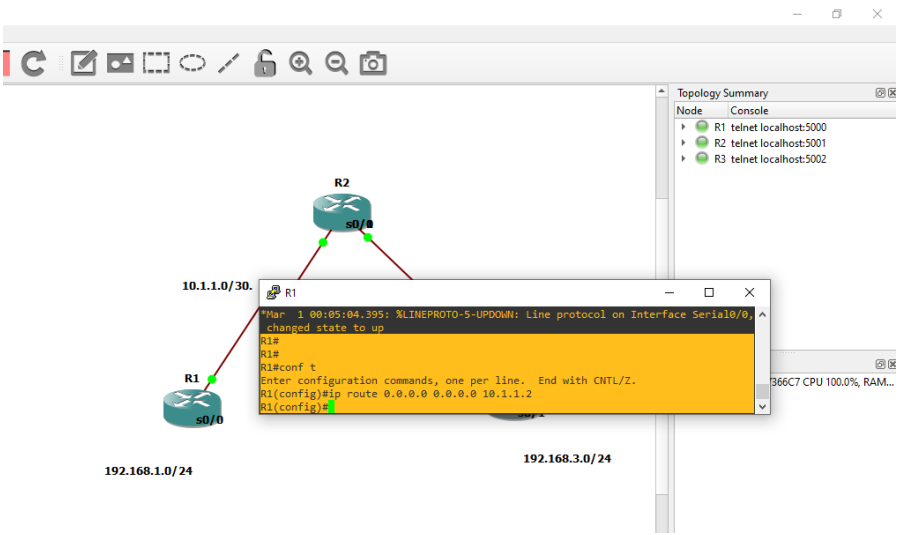
interface Serial0/1
 description R3 --> R2
 ip address 10.2.2.2 255.255.255.252
 no shutdown
exit
!
end



Step 2: Configure static routes.

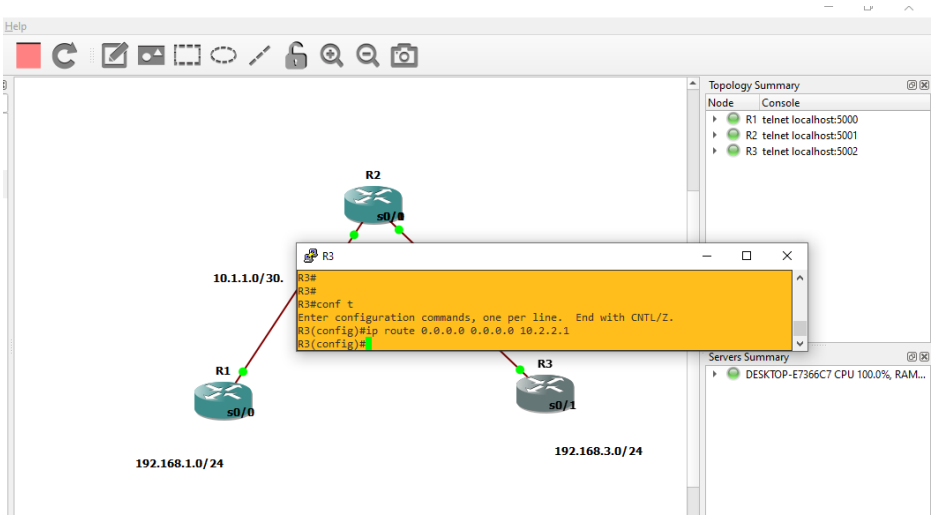
a. On R1, configure a default static route to ISP.

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



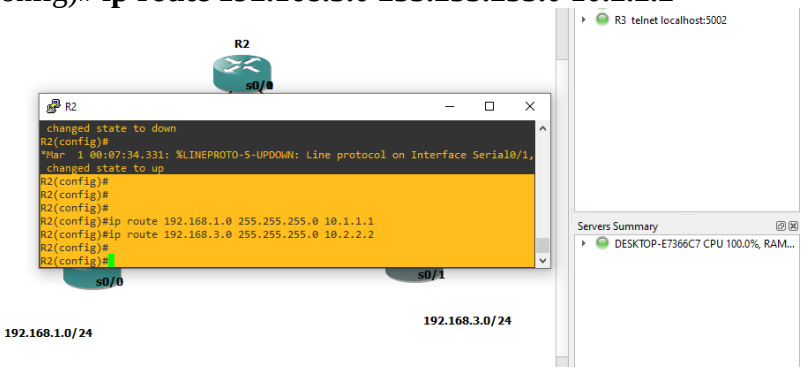
b. On R3, configure a default static route to ISP.

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



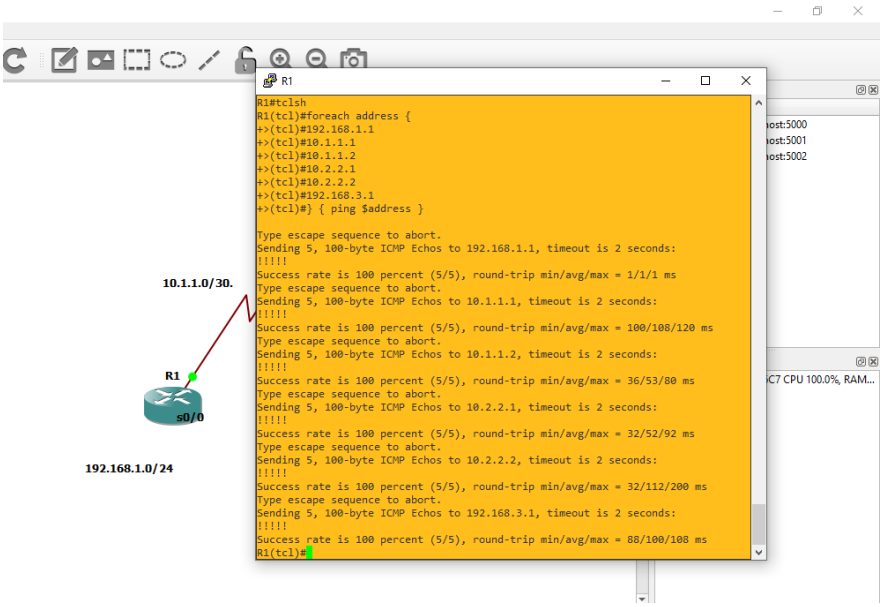
c. On R2, configure two static routes.

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**



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

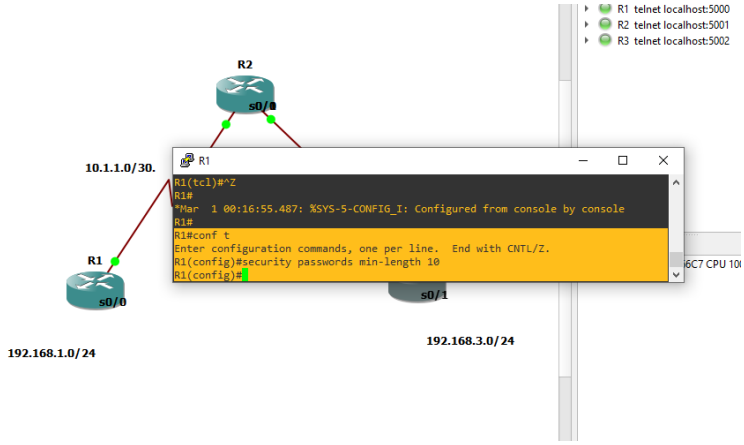
foreach address {
192.168.1.1
10.1.1.1
10.1.1.2
10.2.2.1
10.2.2.2
192.168.3.1
} { ping $address }



Step 3: Secure management access.

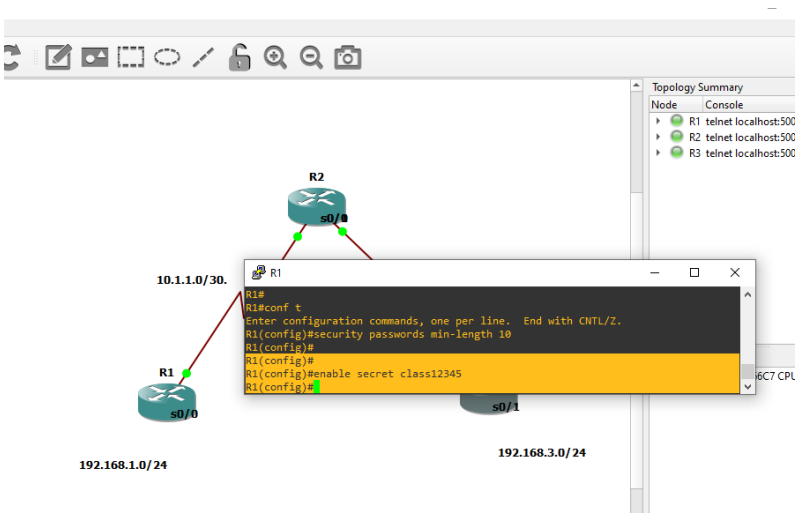
a. On R1, use the **security passwords** command to set a minimum password length of 10 characters.

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



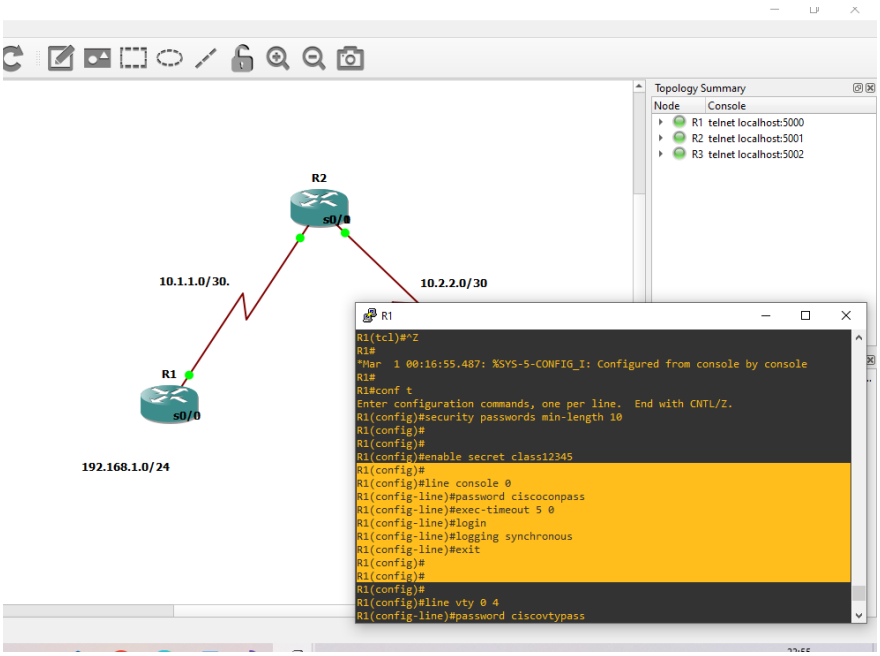
b. Configure the enable secret encrypted password on both routers.

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.

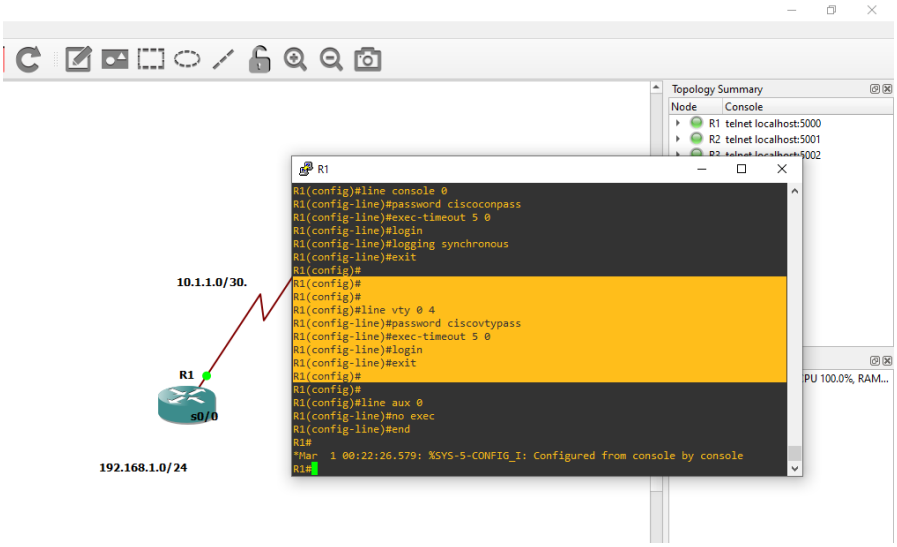
R1(config)# **line console 0**
R1(config-line)# **password ciscoconpass**
R1(config-line)# **exec-timeout 5 0**
R1(config-line)# **login**
R1(config-line)# **logging synchronous**
R1(config-line)# **exit**
R1(config)#



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

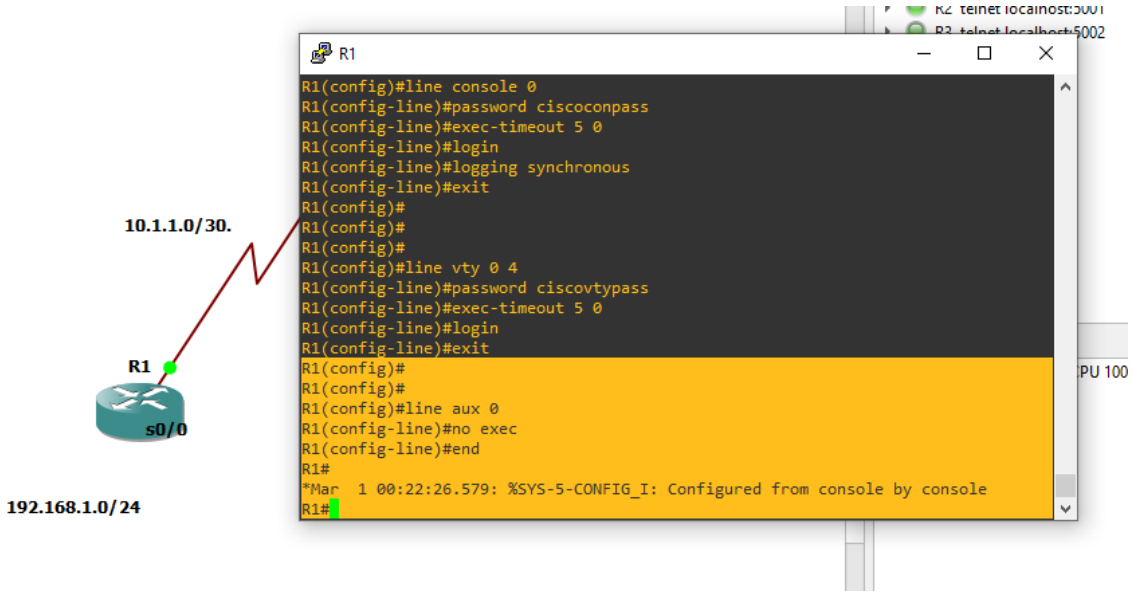
R1(config)# **line vty 0 4**

R1(config-line)# **password ciscovtypass**
R1(config-line)# **exec-timeout 5 0**
R1(config-line)# **login**
R1(config-line)# **exit**
R1(config)#



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.

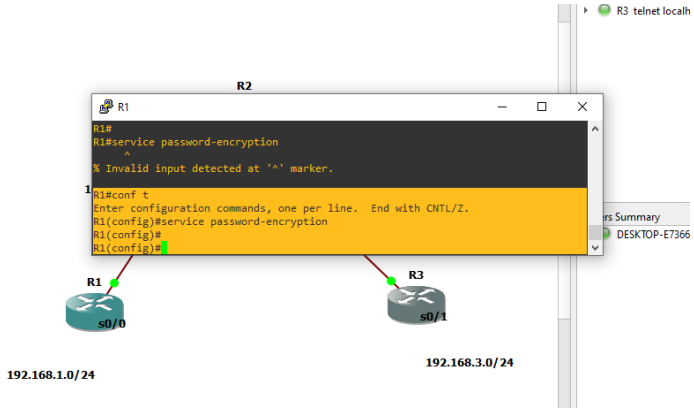
R1(config)# **line aux 0**
R1(config-line)# **no exec**
R1(config-line)# **end**
R1#



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

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

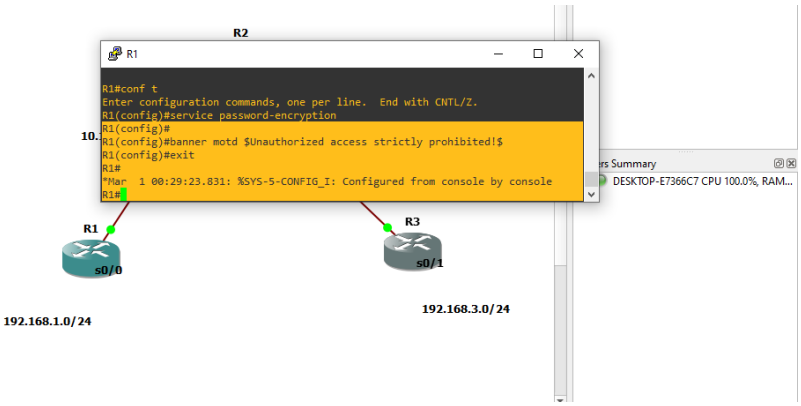
R1(config)#



g. 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**

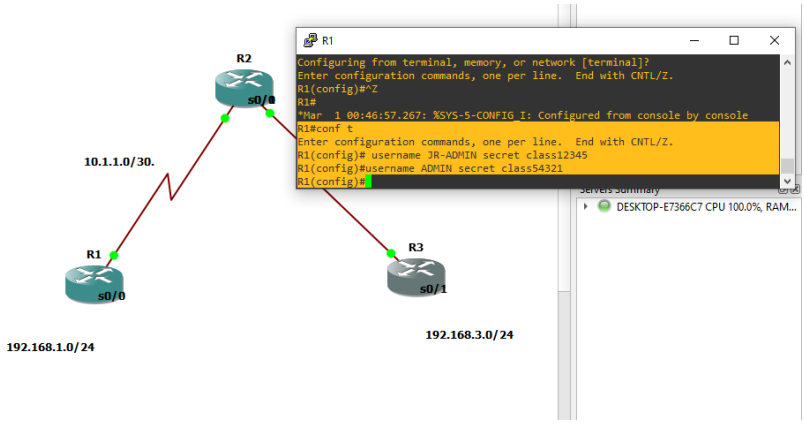


Step 4: Configure enhanced username password security.

a. To create 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:

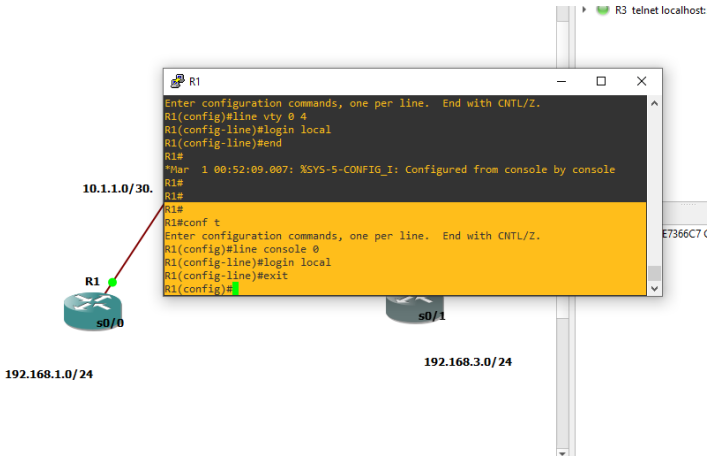
R1(config)# **username JR-ADMIN secret class12345**

R1(config)# **username ADMIN secret class54321**



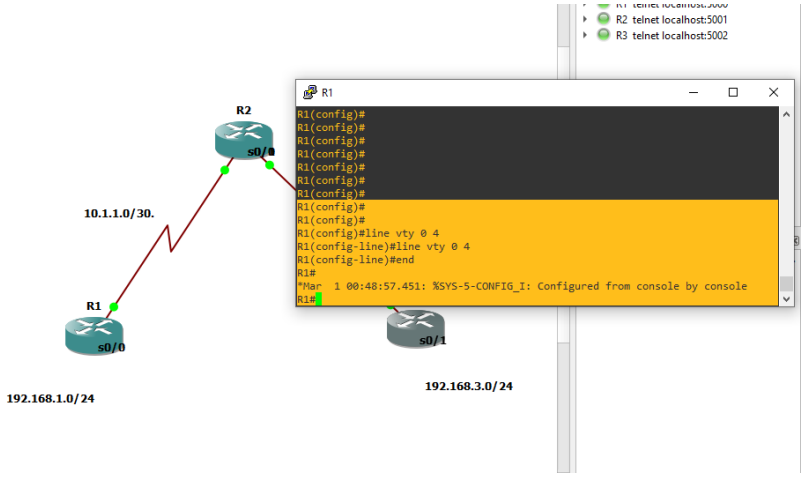
b. Set the console line to use the locally defined login accounts.

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

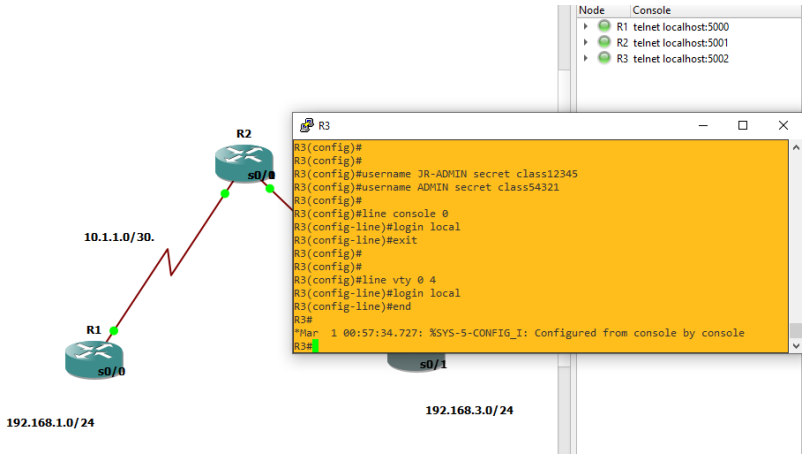


c. Set the vty lines to use the locally defined login accounts.

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

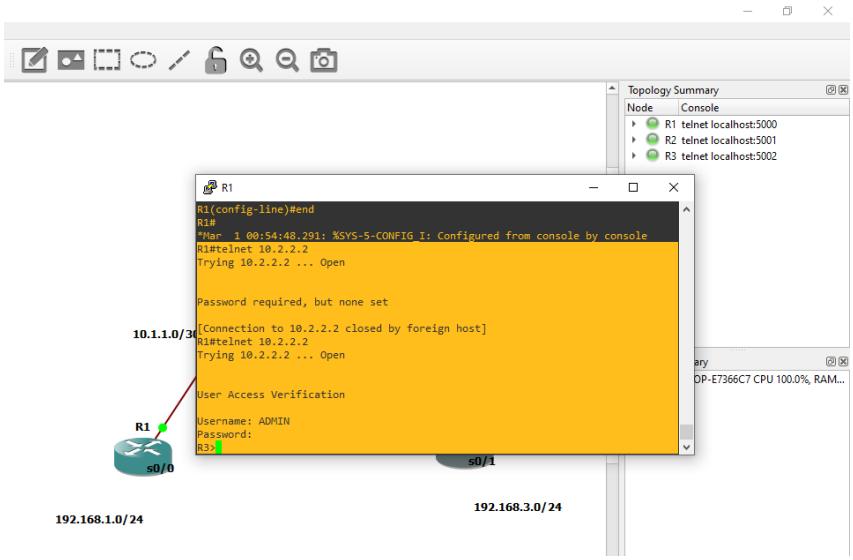


- Repeat the steps 4a to 4c on R3.



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

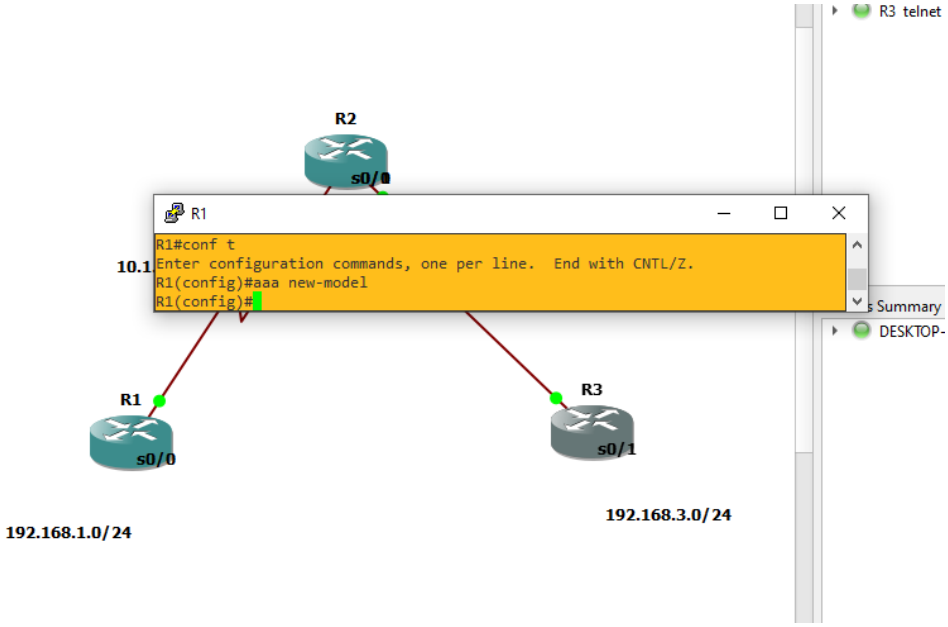
R1# **telnet 10.2.2.2**



Step 5: Enabling AAA RADIUS Authentication with Local User for Backup.

a. Always have local database accounts created before enabling AAA. Since we created two local database accounts in the previous step, then we can proceed and enable AAA on R1.

R1(config)# **aaa new-model**



PRACTICAL NO: 5

Aim: Configure and Verify Path Control Using PBR

Step 1: Configure loopbacks and assign addresses

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#interface Lo1
R1(config-if)#de
*Mar  1 00:02:46.079: %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)#interface Serial0/0
R1(config-if)#description R1 --> R2
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)#clock rate 128000
```

```
R1(config-if)#bandwidth 128
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:05:33.483: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R1(config-if)#
*Mar  1 00:05:34.487: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R1(config-if)#interface Serial0/1
R1(config-if)#description R1 --> R3
R1(config-if)#
*Mar  1 00:06:01.695: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to down
R1(config-if)#ip address 172.16.13.1 255.255.255.248
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:06:24.671: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
R1(config-if)#
*Mar  1 00:06:25.675: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
R1(config-if)#end
R1#
*Mar  1 00:06:33.207: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Mar  1 00:06:51.719: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to down
R1#
*Mar  1 00:08:51.695: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
```

Router R2

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname R2
R2(config)#interface Lo2
R2(config-if)#
*Mar  1 00:07:19.155: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback2, changed state to up
R2(config-if)#description R2 LAN
R2(config-if)#ip address 192.168.2.1 255.255.255.0
```

```
R2(config)#interface Serial0/0
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)#
*Mar  1 00:08:36.975: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:08:37.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed s
R2(config-if)#interface Serial0/1
R2(config-if)#description R2 --> R3
R2(config-if)#ip address 172.16.23.2 255.255.255.248
R2(config-if)#clock rate 128000
R2(config-if)#bandwidth 128
R2(config-if)#no shutdown
R2(config-if)#
*Mar  1 00:09:38.739: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:09:39.739: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed s
R2(config-if)#end
R2#
*Mar  1 00:09:46.371: %SYS-5-CONFIG_I: Configured from console by console
R2#
*Mar  1 00:10:01.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed s
R2#
*Mar  1 00:13:51.867: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed s
R2#
```

Router 3

```
R3#
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#interface Serial0/0
R3(config-if)#description R3 --> R1
R3(config-if)#ip address 172.16.13.3 255.255.255.248
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
R3(config-if)#interface Serial0/1
R3(config-if)#description R3 --> R2
R3(config-if)#ip address 172.16.23.3 255.255.255.248
R3(config-if)#bandwidth 128
R3(config-if)#no shutdown
```

```
R3(config)#interface Serial 1/0
R3(config-if)#description R3 --> R4
R3(config-if)#ip address 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)#
*Mar  1 00:27:00.415: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:27:01.415: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed s
R3(config-if)#end
```

Router 4

```
R4#
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#hostname R4
R4(config)#interface Lo4
R4(config-if)#
*Mar  1 00:15:28.287: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback4, changed state to up
R4(config-if)#description R4 LAN A
R4(config-if)#ip address 192.168.4.1 255.255.255.128
R4(config-if)#interface Lo5
R4(config-if)#
*Mar  1 00:15:47.139: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback5, changed state to up
R4(config-if)#description R4 LAN B
R4(config-if)#ip address 192.168.4.129 255.255.255.128
```

```
R4(config)#interface Serial0/0
R4(config-if)#description R4 --> R3
R4(config-if)#ip address 172.16.34.4 255.255.255.248
R4(config-if)#bandwidth 64
R4(config-if)#no shutdown
R4(config-if)#
*Mar  1 00:16:45.511: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R4(config-if)#
*Mar  1 00:16:46.515: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed s
R4(config-if)#end
R4#
*Mar  1 00:16:51.843: %SYS-5-CONFIG_I: Configured from console by console
R4#
*Mar  1 00:17:11.711: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed s
R4#
*Mar  1 00:27:01.703: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed s
R4#
```

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
Serial0/0                  172.16.13.3     YES manual up                    up
Serial0/1                  172.16.34.3     YES manual up                    up
Loopback3                  192.168.3.1     YES manual up                    up
R3#show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
Serial0/0 is up, line protocol is up
  Internet address is 172.16.13.3/29
FastEthernet0/1 is administratively down, line protocol is down
Serial0/1 is up, line protocol is up
  Internet address is 172.16.34.3/29
Serial1/0 is administratively down, line protocol is down
Serial1/1 is administratively down, line protocol is down
Serial1/2 is administratively down, line protocol is down
Serial1/3 is administratively down, line protocol is down
Loopback3 is up, line protocol is up
  Internet address is 192.168.3.1/24
R3#
```

```
R3#show interfaces description | include up
Se0/0                          up             up       R3 --> R1
Se0/1                          up             up       R3 --> R2
Se1/0                          up             up       R3 --> R4
Lo3                            up             up       R3 LAN
R3#
```

Step 2 : Configure basic EIGRP.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#
*Mar  1 00:41:10.835: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.2 (Serial0/0) is up: new ad
jacency
R1(config-router)#
*Mar  1 00:41:25.211: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.2 (Serial0/0) is resync: pe
er graceful-restart
R1(config-router)#
*Mar  1 00:42:05.051: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.3 (Serial0/1) is up: new ad
jacency
R1(config-router)#
*Mar  1 00:42:20.351: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.3 (Serial0/1) is resync: pe
er graceful-restart
R1(config-router)#
```

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#
*Mar  1 00:41:10.563: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.1 (Serial0/0) is up: new ad
jacency
R2(config-router)#network 172.16.23.0 0.0.0.7
R2(config-router)#no auto-summary
R2(config-router)#
*Mar  1 00:41:24.883: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.1 (Serial0/0) is resync: su
mmary configured
R2(config-router)#
*Mar  1 00:42:09.463: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.23.3 (Serial0/1) is up: new ad
jacency
R2(config-router)#
*Mar  1 00:42:20.055: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.23.3 (Serial0/1) is resync: pe
er graceful-restart
R2(config-router)#
```

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#
*Mar  1 00:42:05.071: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.1 (Serial0/0) is up: new ad
jacency
R3(config-router)#network 172.16.23.0 0.0.0.7
R3(config-router)#
*Mar  1 00:42:09.779: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.23.2 (Serial0/1) is up: new ad
jacency
R3(config-router)#network 172.16.34.0 0.0.0.7
R3(config-router)#no auto-summary
R3(config-router)#
*Mar  1 00:42:20.335: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.1 (Serial0/0) is resync: su
mmary configured
*Mar  1 00:42:20.335: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.23.2 (Serial0/1) is resync: su
mmary configured
R3(config-router)#
*Mar  1 00:42:58.235: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.4 (Serial1/0) is up: new ad
jacency
R3(config-router)#
*Mar  1 00:43:04.723: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.4 (Serial1/0) is resync: pe
er graceful-restart
R3(config-router)#
```

```
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#
*Mar  1 00:42:58.007: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Serial0/0) is up: new ad
jacency
R4(config-router)#no auto-summary
R4(config-router)#
*Mar  1 00:43:04.455: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Serial0/0) is resync: su
mmary configured
R4(config-router)#
```

Step3 : Verify EIGRP connectivity

zz. Verify the configuration by using the show ipeigrpneighbors command to check which routers have EIGRP adjacencies.

```
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             Se0/1            11 00:04:33   16  2280  0  27
0   172.16.12.2             Se0/0            13 00:05:28   13  1140  0  24
R1#
```

```
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             Se0/1            13 00:05:03   17  1140  0  28
0   172.16.12.1             Se0/0            14 00:06:01   14  1140  0  26
R2#
```

```
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             Se1/0            11 00:04:35   25  2280  0  7
1   172.16.23.2             Se0/1            13 00:05:23   16  1140  0  25
0   172.16.13.1             Se0/0            10 00:05:28   17  2280  0  27
R3#
```

```
R4#
*Mar  1 00:47:45.679: %SYS-5-CONFIG_I: Configured from console by console
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             Se0/0            11 00:04:50   27  2280  0  32
R4#
```

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

```
R1# tclsh
```

```
+>(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 }
+>(tcl)#
```

Step 4 : Verify the current path.

```
R1#
*Mar  1 00:52:54.831: %SYS-5-CONFIG_I: Configured from console by console
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, 00:10:47, Serial0/1
D       172.16.23.0 [90/21024000] via 172.16.12.2, 00:10:53, Serial0/0
C       172.16.12.0 is directly connected, Serial0/0
C       172.16.13.0 is directly connected, Serial0/1
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:09:58, Serial0/1
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:09:58, Serial0/1
C    192.168.1.0/24 is directly connected, Loopback1
D    192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:10:53, Serial0/0
D    192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:10:53, Serial0/0
R1#
```

ccc.On R4, use the traceroute command to the R1 LAN address and source the ICMP packet from R4 LAN A and LAN B.

```
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 4 msec 16 msec 16 msec
  2 172.16.23.2 28 msec 0 msec 20 msec
  3 172.16.12.1 16 msec 20 msec 20 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 8 msec 8 msec 32 msec
  2 172.16.23.2 0 msec 28 msec 0 msec
  3 172.16.12.1 16 msec 16 msec 24 msec
```

```
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, Serial1/0
C       172.16.23.0 is directly connected, Serial0/1
D       172.16.12.0 [90/21024000] via 172.16.23.2, 00:13:41, Serial0/1
C       172.16.13.0 is directly connected, Serial0/0
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/40640000] via 172.16.34.4, 00:12:46, Serial1/0
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:12:46, Serial1/0
D    192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:13:41, Serial0/1
D    192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:13:41, Serial0/1
C    192.168.3.0/24 is directly connected, Loopback3
R3#
```

On R3, use the show interfaces serial 0/0/0 and show interfaces s0/0/1 commands.

```
R3# show interfaces serial0/0
Serial0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Description: R3 --> R1
  Internet address is 172.16.13.3/29
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  CRC checking enabled
  Last input 00:00:04, output 00:00:01, 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
     596 packets input, 39215 bytes, 0 no buffer
     Received 353 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     532 packets output, 36064 bytes, 0 underruns
     0 output errors, 0 collisions, 7 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

```
R3#show interfaces serial0/0 | include BW
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
R3# show interfaces serial0/1 | include BW
  MTU 1500 bytes, BW 128 Kbit/sec, DLY 20000 usec,
R3#
```

```
 MTU 1500 bytes, BW 128 Kbit/sec, DLY 20000 usec,
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 21152000
  Routing Descriptor Blocks:
  172.16.23.2 (Serial0/1), from 172.16.23.2, Send flag is 0x0
      Composite metric is (21152000/20640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
  172.16.13.1 (Serial0/0), from 172.16.13.1, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
R3#
```

Step 5 : Configure PBR to provide path control.

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

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
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)#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)#interface s0/1/0
                          ^
% Invalid input detected at '^' marker.

R3(config)#interface s0/1
R3(config-if)#ip policy route-map R3-to-R1
R3(config-if)#
*Mar  1 01:07:01.787: PR-RP: Set Serial0/1 policy_routemap=R3-to-R1
*Mar  1 01:07:01.787: PR-RP: Set Serial0/1 policy_routemap=R3-to-R1
R3(config-if)#end
R3#
*Mar  1 01:07:11.087: %SYS-5-CONFIG_I: Configured from console by console
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#
```

Step 6: Test the policy.

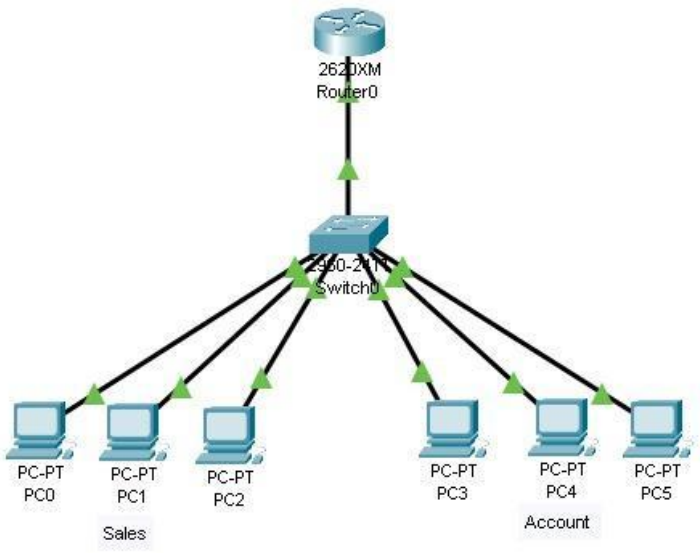
```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#
R3(config)#access-list 1 permit 192.168.4.0 0.0.0.255
R3(config)#exit
R3#
R3#
*Jul  1 10:12:15.903: %SYS-5-CONFIG_I: Configured from console by console
R3#debug ip policy ?
  <1-199>  Access list
  dynamic  dynamic PBR
  early    Early PBR
  <cr>

R3#debug ip policy
Policy routing debugging is on
R3#
R3#debug ip policy 1
Policy routing debugging is on for access list 1
R3#
R3#traceroute 192.168.1.1 source 192.168.4.1
% Invalid source address- IP address not on any of our up interfaces
R3#
*Jul  1 10:14:02.939: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, policy rejected -- normal forwarding
*Jul  1 10:14:02.967: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, policy rejected -- normal forwarding
*Jul  1 10:14:02.999: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, policy rejected -- normal forwarding
*Jul  1 10:14:03.035: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwardin
g
*Jul  1 10:14:03.099: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwardin
g
*Jul  1 10:14:03.171: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1
R3#, len 28, FIB policy rejected(no match) - normal forwarding
*Jul  1 10:14:03.223: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwardin
g
*Jul  1 10:14:03.303: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwardin
g
*Jul  1 10:14:03.387: IP: s=192.168.4.1 (Serial3/2), d=192.168.1.1, len 28, FIB policy rejected(no match) - normal forwardin
g
R3#
*Jul  1 10:14:10.975: IP: s=192.168.4.129 (Serial3/2), d=192.168.1.1, len 28, policy match
*Jul  1 10:14:10.979: IP: route map R3-to-R1, item 10, permit
*Jul  1 10:14:10.979: IP: s=192.168.4.129 (Serial3/2), d=192.168.1.1 (Serial3/0), len 28, policy routed
*Jul  1 10:14:10.983: IP: Serial3/2 to Serial3/0 172.16.13.1
*Jul  1 10:14:11.007: IP: s=192.168.4.129 (Serial3/2), d=192.168.1.1, len 28, policy match
*Jul  1 10:14:11.011: IP: route map R3-to-R1, item 10, permit
*Jul  1 10:14:11.011: IP: s=192.168.4.129 (Serial3/2), d=192.168.1.1 (Serial3/0), len 28, policy routed
*Jul  1 10:14:11.015: IP: Serial3/2 to Serial3/0 172.16.13.1
*Jul  1 10:14:11.039: IP: s=192.168.4.129 (Serial3/2), d=192.168.1.1, len 28, policy match
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
R4(tcl)#
R4(tcl)#
R4(tcl)#exit
R4#
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
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.34.3 24 msec 16 msec 32 msec
  2 172.16.23.2 64 msec 52 msec 60 msec
  3 172.16.12.1 80 msec 80 msec 76 msec
R4#
R4#traceroute 192.168.1.1 source 192.168.4.129
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.34.3 32 msec 32 msec 24 msec
  2 172.16.23.2 64 msec 60 msec 56 msec
  3 172.16.12.1 80 msec 72 msec 72 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
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.34.3 24 msec 28 msec 28 msec
  2 172.16.23.2 68 msec 60 msec 64 msec
  3 172.16.12.1 72 msec 80 msec 72 msec
R4#
R4#traceroute 192.168.1.1 source 192.168.4.129
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.34.3 24 msec 24 msec 28 msec
  2 172.16.13.1 60 msec 60 msec 72 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
VRF info: (vrf in name/id, vrf out name/id)
```

```
R3#
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
Nexthop tracking current: 0.0.0.0
172.16.13.1, fib_nh:0,oce:0,status:0

  Policy routing matches: 18 packets, 576 bytes
R3#
```

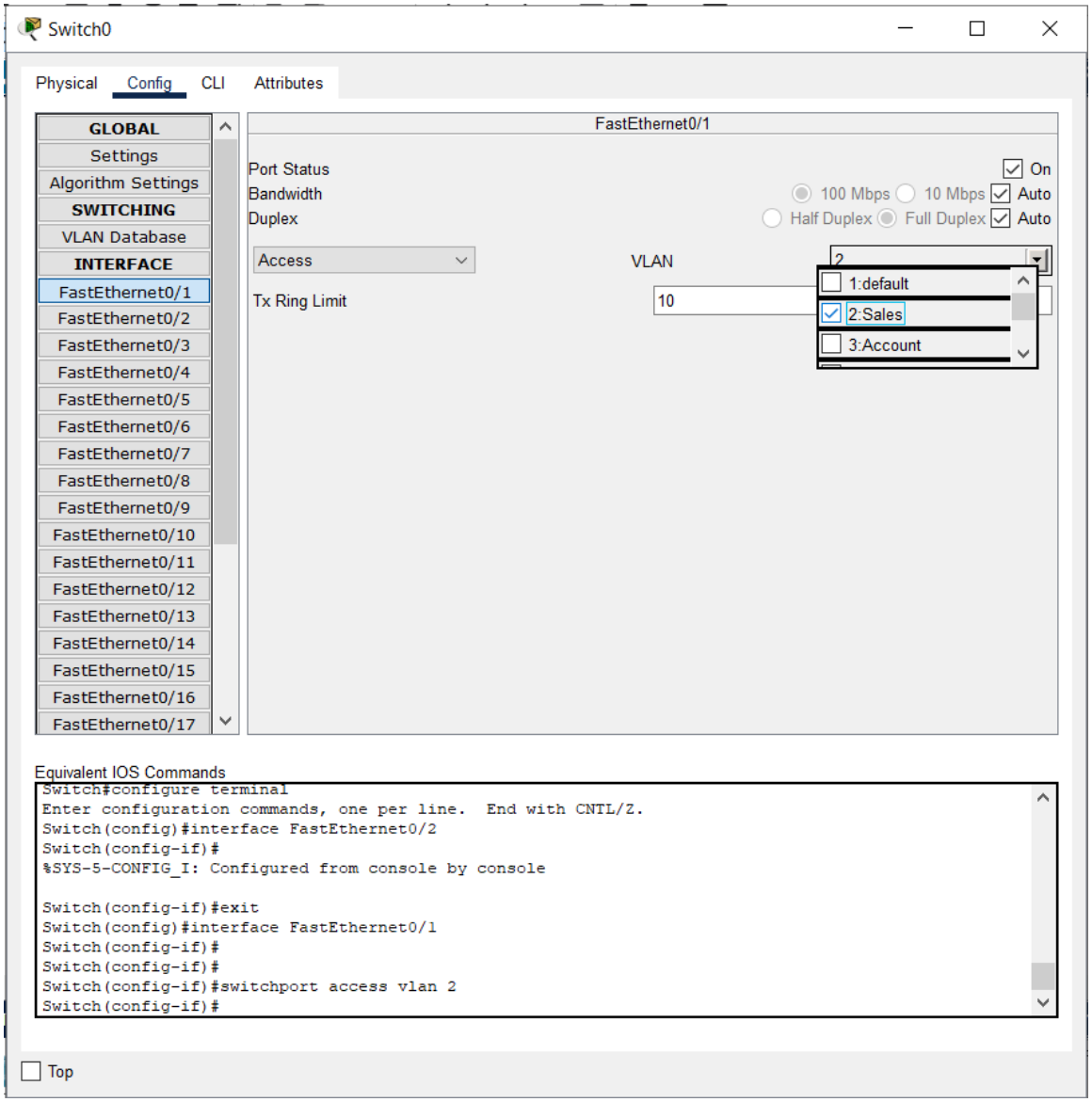
PRACTICAL NO: 6

Aim:- IP Service Level Agreements and Remote SPAN in a Campus Environment



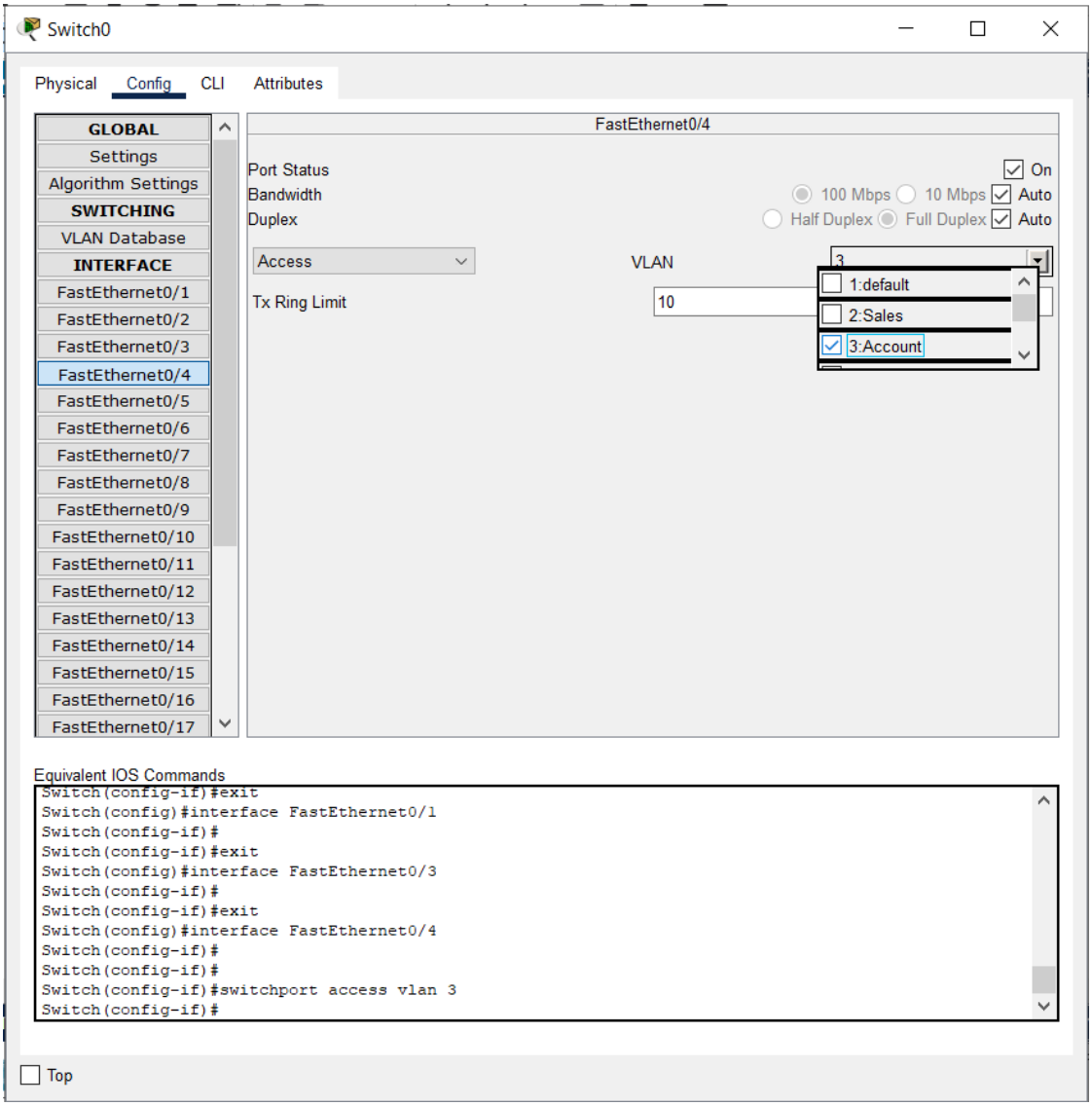
- Prepare the network as shown in the topology diagram above using the following components:
    1. **Switch (2960-24TT)**
    2. **PCs (PC0 to PC5)**
    3. **Router (2620XM)**
- Connect PC0, PC1 and PC2 with the FastEthernet 0/1, 0/2 and 0/3 respectively of switch.
- Connect PC3, PC4 and PC5 with the FastEthernet 0/4, 0/5 and 0/6 respectively of switch.
- Assign IP address to PCs as:

    PC0: 10.0.0.1
    PC1: 10.0.0.2
    PC3: 10.0.0.3
    PC4: 20.0.0.1
    PC5: 20.0.0.2
    PC6: 20.0.0.3



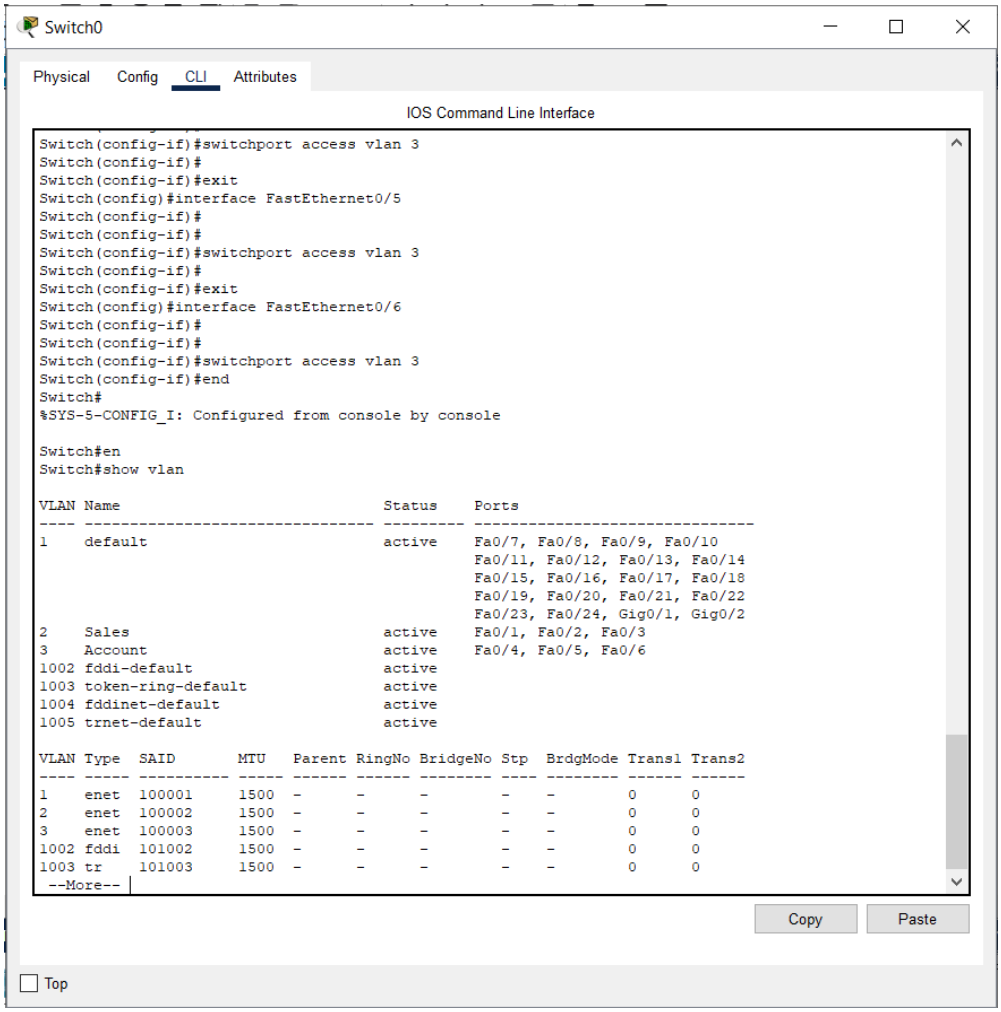
➢ PC 0,1,and 2 are added in the **SALES VLAN** as shown above.

Switch#**conf t**
Switch(config)#**interface FastEthernet0/0**
Switch(config-if)#**switchport access vlan 2**
Switch(config-if)#**exit**
Switch(config)#**interface FastEthernet0/1**
Switch(config-if)#**switchport access vlan 2**
Switch(config-if)#**exit**
Switch(config)#**interface FastEthernet0/2**
Switch(config-if)#**switchport access vlan 2**
Switch(config-if)#**exit**



➢ PC 3,4,and 5 are added in the **ACCOUNTS VLAN** as shown above.

Switch#**conf t**
Switch(config)#**interface FastEthernet0/3**
Switch(config-if)#**switchport access vlan 3**
Switch(config-if)#**exit**
Switch(config)#**interface FastEthernet0/4**
Switch(config-if)#**switchport access vlan 3**
Switch(config-if)#**exit**
Switch(config)#**interface FastEthernet0/5**
Switch(config-if)#**switchport access vlan 3**
Switch(config-if)#**exit**

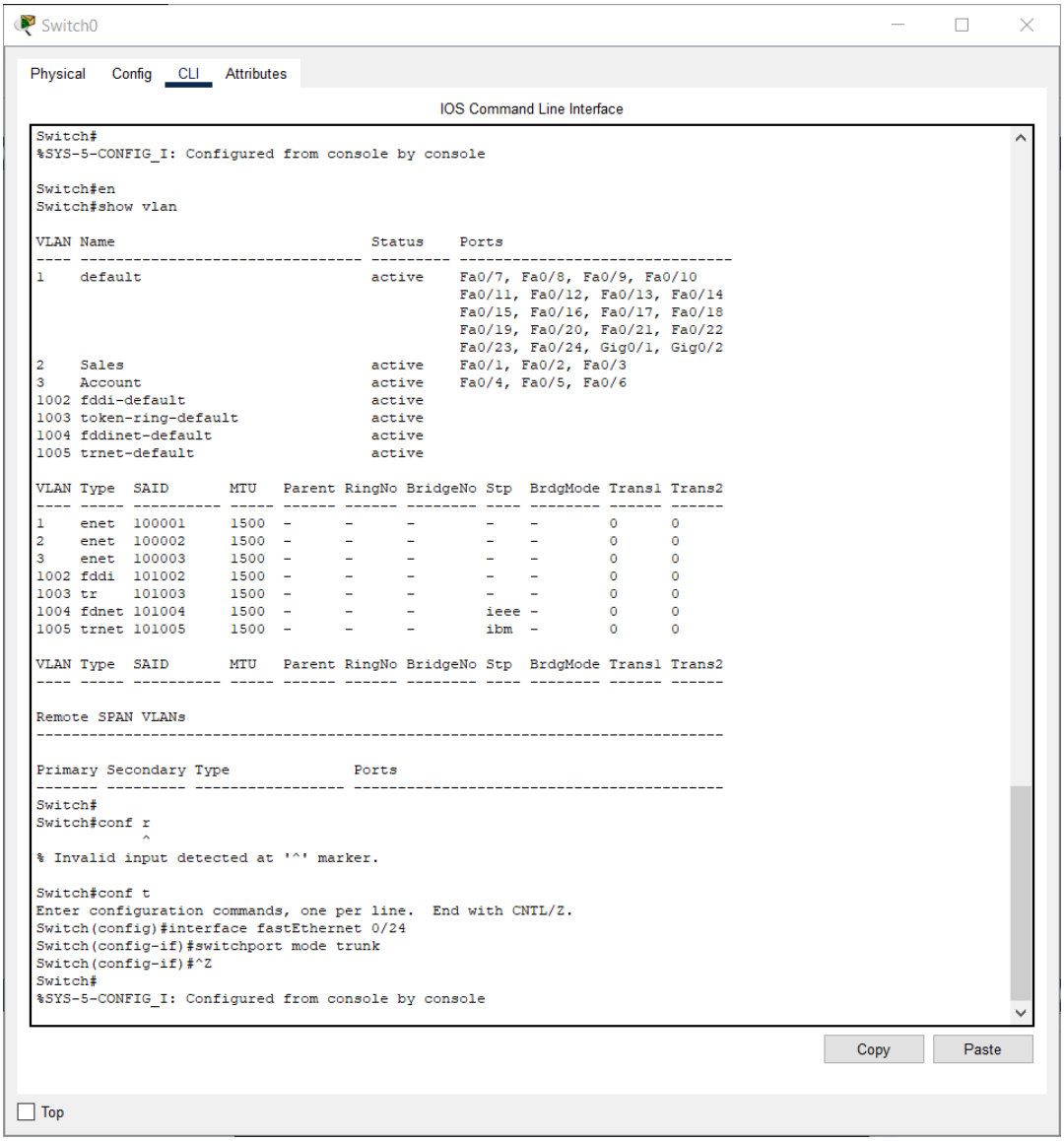


➢ Open the config of Switch

Switch#**en**

Switch#**show vlan**

➢ Connect a router to switch at FastEthernet 0/24 as shown in the topology diagram.



➢ Configure trunking ports

Switch#**conf t**

Switch(config)#**interface fastEthernet 0/24**

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

```
Router>en
Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES unset  administratively down down
```

➢ Open Router CLI to check the status of the fastEthernet 0/0 port

Router>**en**

Router#**show ip interface brief**

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

- To UP the status of fastEthernet 0/0

  Router#**conf t**

  Router(config)#**interface fastEthernet 0/0**

  Router(config-if)#**no shutdown**

```
Router(config)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES unset  up                    up
Router#
```

- To check if the status of ports are UP

  Router#**show ip interface brief**

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastEthernet 0/0.1
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed state to up

Router(config-subif)#encapsulation dot1Q 2
Router(config-subif)#ip address 10.0.0.100
% Incomplete command.
Router(config-subif)#ip address 10.0.0.100 255.0.0.0
Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.2
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.2, changed state to up

Router(config-subif)#encapsulation dot1Q 3
Router(config-subif)#ip address 20.0.0.100 255.0.0.0
Router(config-subif)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES unset  up                    up
FastEthernet0/0.1      10.0.0.100      YES manual up                    up
FastEthernet0/0.2      20.0.0.100      YES manual up                    up
Router#
```

- To create sub interfaces on Router

  Router#**conf t**

  Router(config)#**interface fastEthernet 0/0.1**

  Router(config-subif)#**encapsulation dot1Q 2**

  Router(config-subif)#**ip address 10.0.0.100 255.0.0.0**

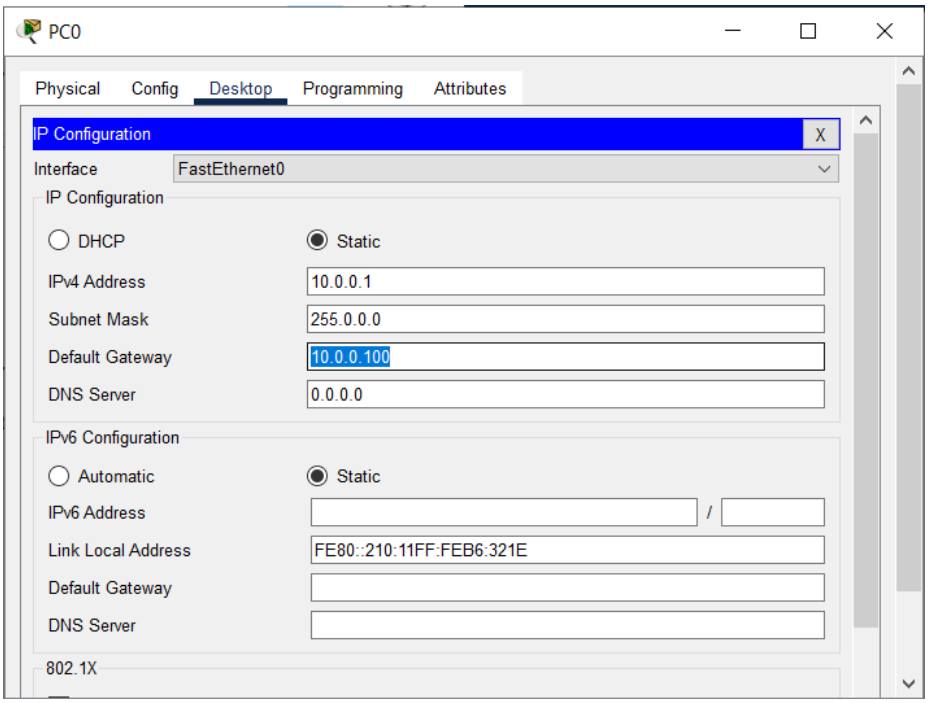
  Router(config-subif)#**exit**

Router(config)#**interface fastEthernet 0/0.2**
Router(config-subif)#**encapsulation dot1Q 3**
Router(config-subif)#**ip address 20.0.0.100 255.0.0.0**
Router(config-subif)#**^z**
Router#**show ip interface brief**

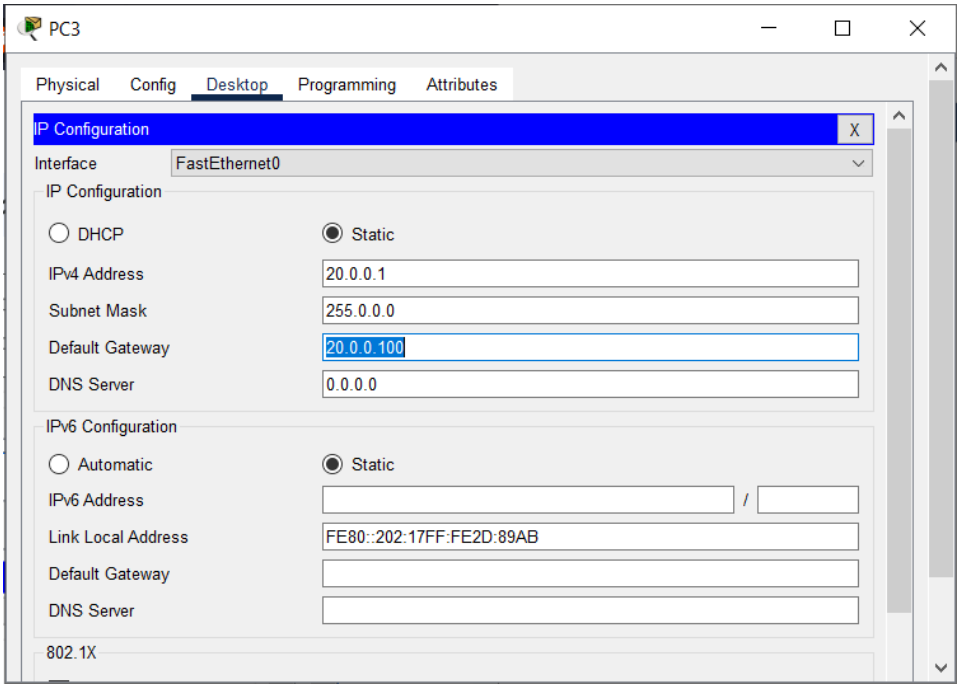
```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#
```

➢ To provide route for communicating between PCs and Router.

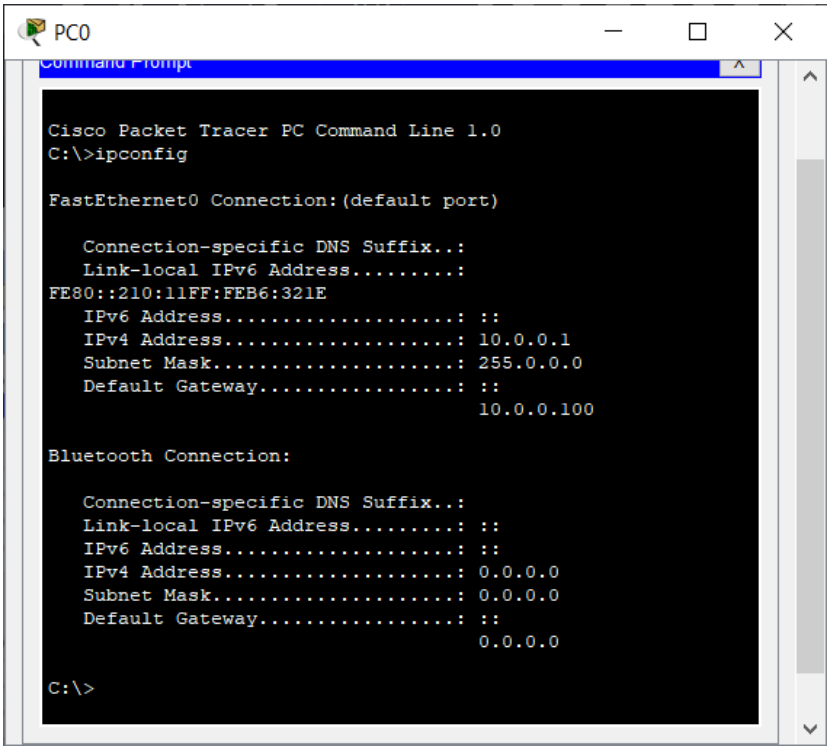
Router#**conf t**
Router(config)#**router rip**
Router(config-router)#**network 10.0.0.0**
Router(config-router)#**network 20.0.0.0**



➢ Assign Default Gateway to PC 0, 1 and 2 as **10.0.0.100.**



➢ Assign Default Gateway to PC 3, 4 and 5 as **20.0.0.100.**

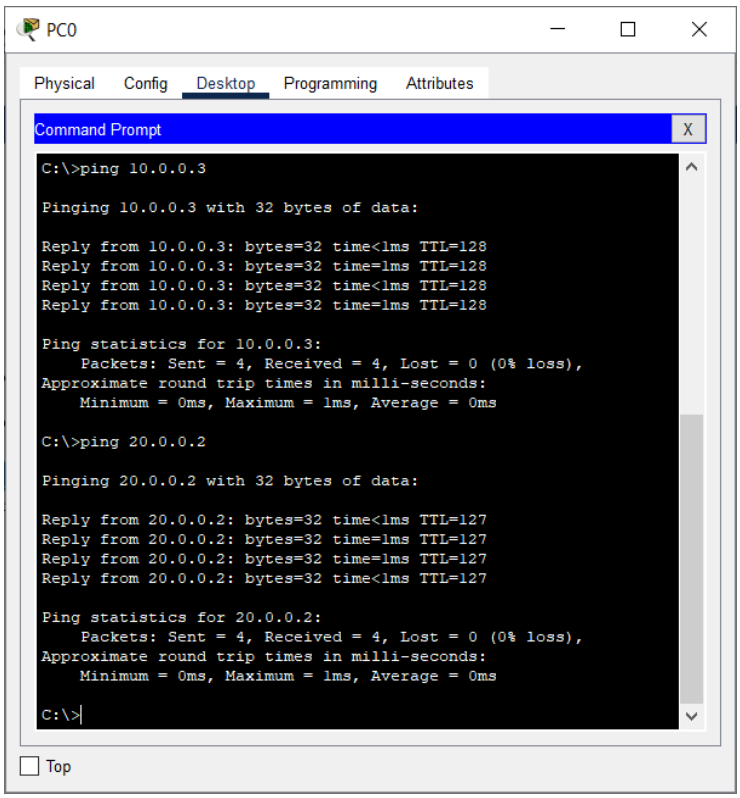


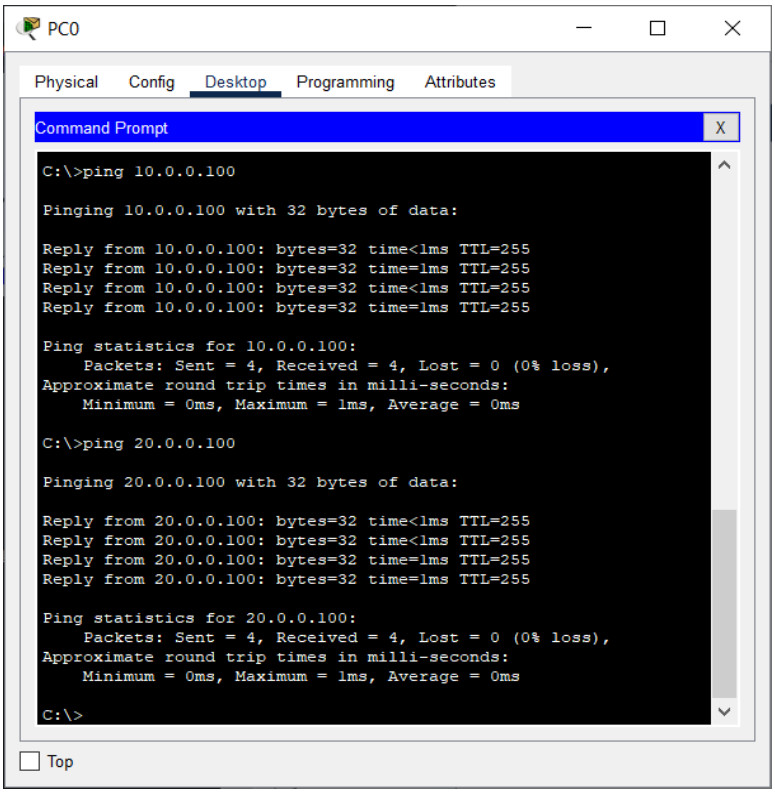
➢ Open Command Prompt of PC0 and check the ip addresses by the command **ipconfig.**

➢ Repeat the same step for other PCs also.

➢ Now using the **ping** command check if the connection is correct and the messages are sent properly without any error.

➢ Use the **ping** command as shown below to check the connection between PCs themselves and with the Router.


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
C:\>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time<1ms TTL=127
Reply from 20.0.0.2: bytes=32 time=1ms TTL=127
Reply from 20.0.0.2: bytes=32 time=1ms TTL=127
Reply from 20.0.0.2: bytes=32 time<1ms TTL=127
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
Top

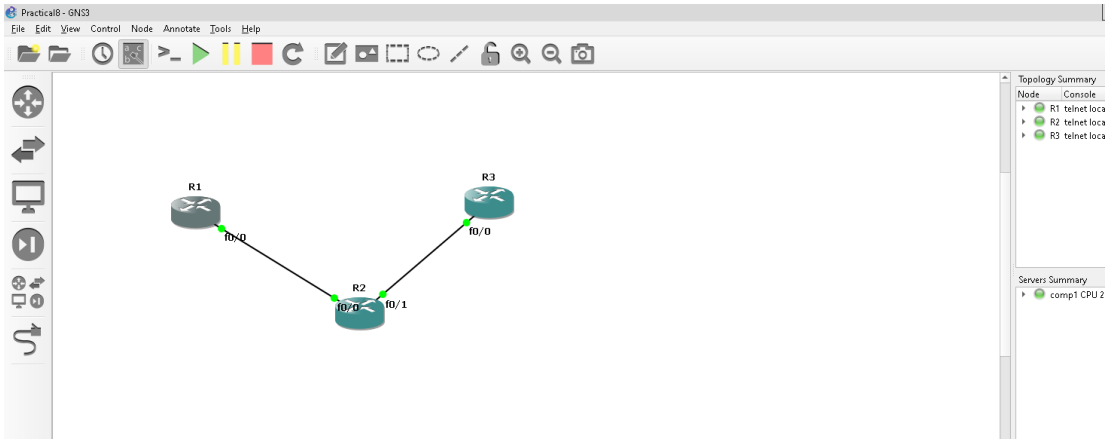

PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
C:\>ping 10.0.0.100
Pinging 10.0.0.100 with 32 bytes of data:
Reply from 10.0.0.100: bytes=32 time<1ms TTL=255
Reply from 10.0.0.100: bytes=32 time=1ms TTL=255
Reply from 10.0.0.100: bytes=32 time<1ms TTL=255
Reply from 10.0.0.100: bytes=32 time=1ms TTL=255
Ping statistics for 10.0.0.100:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>ping 20.0.0.100
Pinging 20.0.0.100 with 32 bytes of data:
Reply from 20.0.0.100: bytes=32 time<1ms TTL=255
Reply from 20.0.0.100: bytes=32 time<1ms TTL=255
Reply from 20.0.0.100: bytes=32 time=1ms TTL=255
Reply from 20.0.0.100: bytes=32 time=1ms TTL=255
Ping statistics for 20.0.0.100:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
Top

PRACTICAL NO:- 7

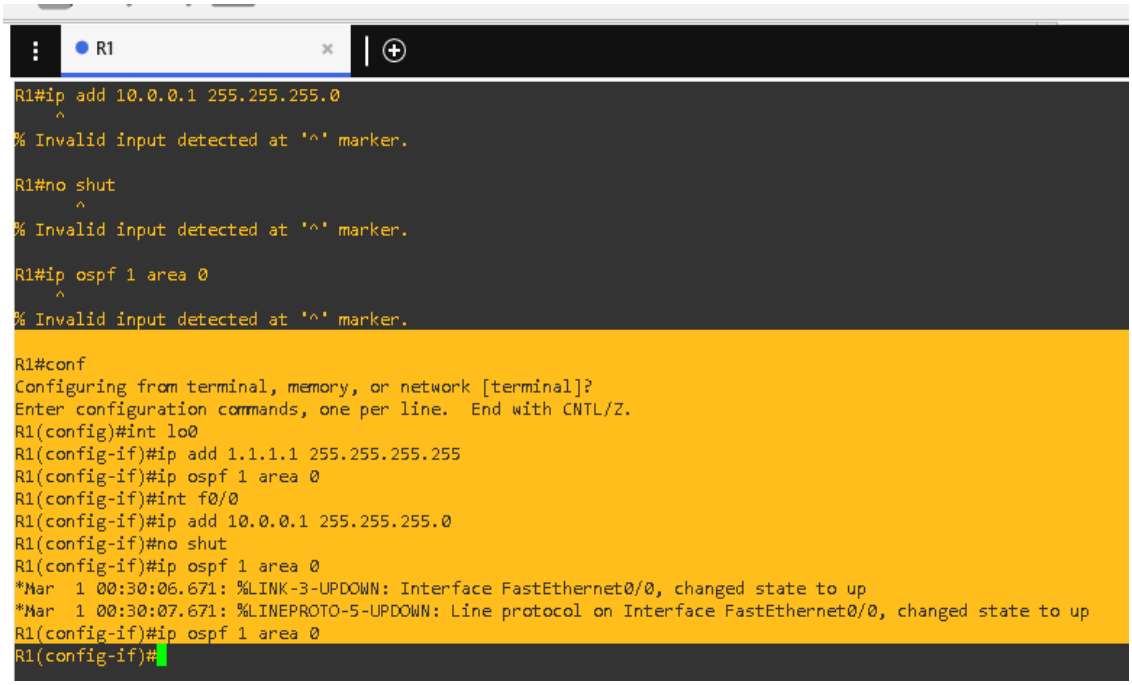
Aim:- Cisco MPLS Configuration

<u>Step 1</u> : IP addressing of MPLS Core and OSPF

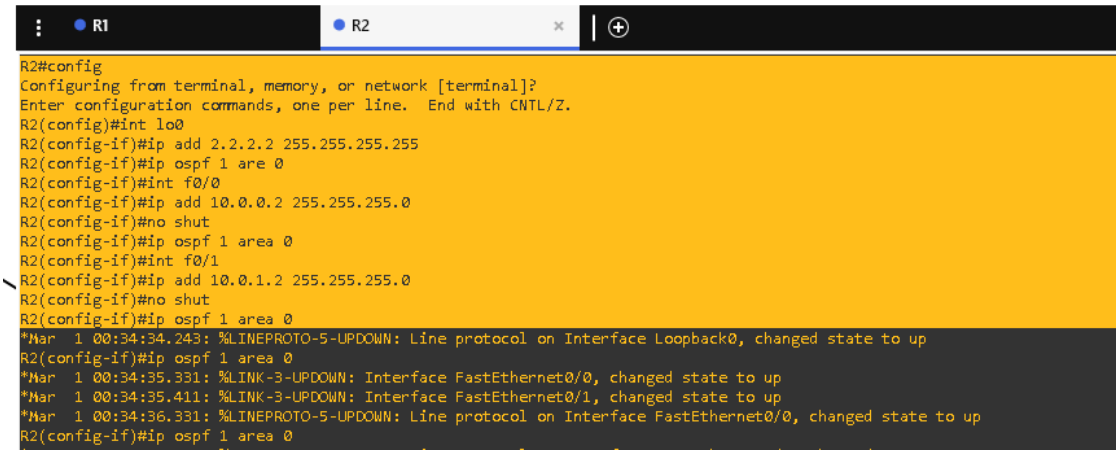
First bring 3 routers into your topology R1, R2, R3 position them as below. We are going to address the routers and configure ospf to ensure loopback to loopback connectivity between R1 and R3



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

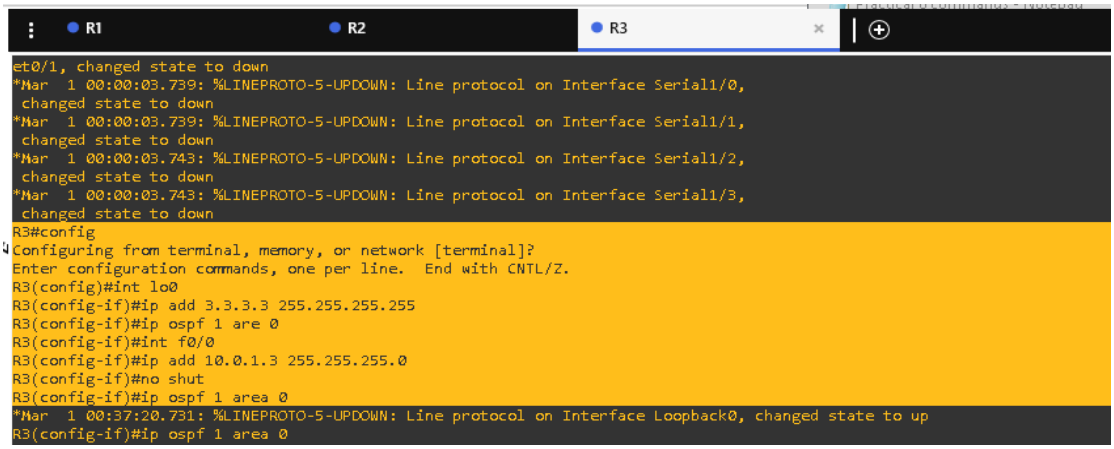


R2
hostname R2
int lo0
ip add 2.2.2.2 255.255.255.255
ipospf 1 are 0
int f0/0
ip add 10.0.0.2 255.255.255.0
no shut
ipospf 1 area 0
int f0/1
ip add 10.0.1.2 255.255.255.0
no shut
ipospf 1 area 0

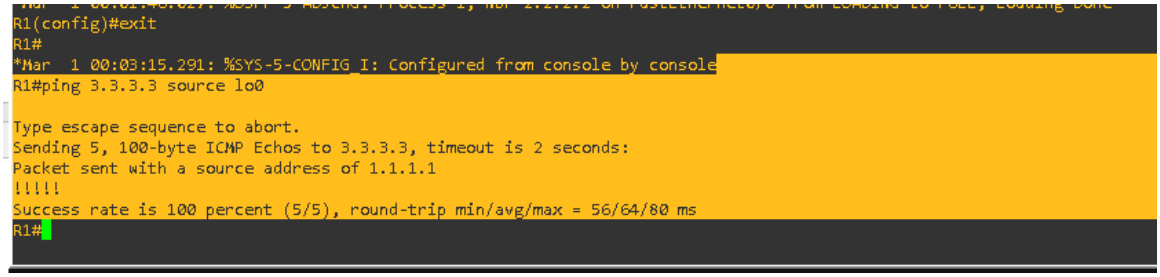


R3
hostname R3
int lo0

ip add 3.3.3.3 255.255.255.255
ipospf 1 are 0
int f0/0
ip add 10.0.1.3 255.255.255.0
no shut
ipospf 1 area 0



You should now have full ip connectivity between R1, R2, R3 to verify this we need to see if we can ping between the loopbacks of R1 and R3



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

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

- At each interface enter the mplsip command
- Under the ospf process use the mplsldpautoconfig command

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

```
*Mar  1 00:01:46.007: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0 from LOADING
R2(config-if)#ip ospf 1 area 0
R2(config-if)#
*Mar  1 00:02:54.695: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on FastEthernet0/1 from LOADING
R2(config-if)#exit
R2(config)#router ospf 1
R2(config-router)#mpls ldp autoconfig
R2(config-router)#
*Mar  1 00:07:57.835: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
R2(config-router)#
```

```
*Mar  1 00:02:49.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config-if)#ip ospf 1 area 0
R3(config-if)#
*Mar  1 00:02:54.735: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0 from LOADING to FULL, Loading Done
R3(config-if)#exit
R3(config)#router ospf 1
R3(config-router)#mpls ldp autoconfig
R3(config-router)#
*Mar  1 00:14:34.175: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is UP
R3(config-router)#
```

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

```
R2#sh mpls interface
Interface              IP             Tunnel   Operational
FastEthernet0/0        Yes (ldp)      No       Yes
FastEthernet0/1        Yes (ldp)      No       Yes
R2#sh mpls ldp neigh
*Mar  1 00:16:06.407: %SYS-5-CONFIG_I: Configured from console by console
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 - 2.2.2.2.59274
        State: Oper; Msgs sent/rcvd: 17/17; Downstream
        Up time: 00:08:34
        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: 3.3.3.3.16377 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:01:58
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 10.0.1.3
        Addresses bound to peer LDP Ident:
          10.0.1.3        3.3.3.3
R2#
```

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

Type escape sequence to abort.
Tracing the route to 3.3.3.3

  1 10.0.0.2 [MPLS: Label 17 Exp 0] 64 msec 64 msec 64 msec
  2 10.0.1.3 56 msec 64 msec 64 msec
R1#
```

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 ldpneighbors 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 Multi Protocol BGP session between R1 and R3 this is done by configuring the vpnv4 address family as below

```
R1#config
Configuring from terminal, memory, or network [terminal]?
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
```

```
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
R3(config-router-af)#
```

You should see log messages showing the BGP sessions coming up.
To verify the BGP session between R1 and R3 issue the command shbgp vpnv4 unicast all summary

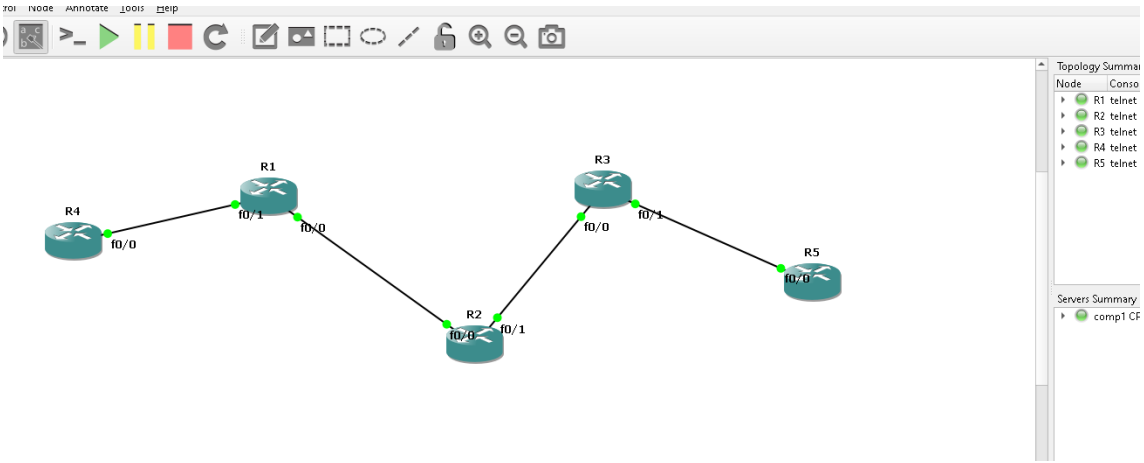
```
R1#sh bgp vpnv4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4     1       3       3        1    0    0 00:00:35        0
R1#
```

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.

PRACTICAL NO :- 8

Aim:- Simulating VRF

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.



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

```
R4#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
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
*Mar  1 00:02:49.019: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loo
```

As an example if R1 was a PE Provider Edge router of an ISP and it had two customers that were both addressed locally with the 192.168.1.0/24 address space it could accommodate

both their routing tables in different VRFs – it distinguishes between the two of them using a Route Distinguisher.

So back to the topology – we now need to create a VRF on R1
For this mpls tutorial we will be using VRF RED

The RD and route-target do not need to be the same – and for a full explanation please read this post on Route Distinguishers Route Distinguisher vs Route Target before proceeding.

```
R1(config)#int f0/1
R1(config-if)#no shut
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#
*Mar  1 00:32:43.407: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:32:44.407: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1(config-if)#exit
R1(config)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
R1(config-vrf)#int f0/1
R1(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.1.1 removed due to enabling VRF RED
```

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

```
R1(config-vrf)#exit
R1(config)#ip vrf fo
R1(config-vrf)#exit
R1(config)#int f0/1
R1(config-if)#ip vrf forwarding RED
R1(config-if)#exit
R1(config)#int f0/1
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#exit
R1#
*Mar  1 00:47:30.591: %SYS-5-CONFIG_I: Configured from console by console
```

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

If you now issue the command ship route vrf red – this will show the routes in the routing table for VRF RED
Now we can start to look int VRF's 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

NOTE: The VRF name is case sensitive!

```
R1#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
 ip vrf forwarding RED
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
end

R1#sh ip rout
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 [110/11] via 10.0.0.2, 00:49:31, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/21] via 10.0.0.2, 00:48:22, 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 [110/20] via 10.0.0.2, 00:49:21, FastEthernet0/0
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, FastEthernet0/1
R1#config
```

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

```
R1(config)#int f0/1
R1(config-if)#ip ospf 2 area 2
R1(config-if)#exit
R1(config)#exit
R1#
*Mar  1 00:26:53.439: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Mar  1 00:26:56.287: %OSPF-5-ADJCHG: Process 2, Nbr 4.4.4.4 on FastEthernet0/1 from LOADING to FULL, Loading Done
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 [110/11] via 192.168.1.4, 00:00:06, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/1
```

We now need to repeat this process for R3 & R5
Router 5 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.

```
R5#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#int lo0
R5(config-if)#ip add 6.6.6.6 255.255.255.255
R5(config-if)#ip ospf 2 area 2
R5(config-if)#int f0/0
R5(config-if)#ip add 192.168.2.6 255.255.255.0
R5(config-if)#ip ospf 2 area 2
R5(config-if)#no shut
R5(config-if)#
*Mar  1 00:27:37.383: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R5(config-if)#
*Mar  1 00:27:40.851: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:27:41.851: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

We also need to configure a VRF onto R3 as well

```
R3(config)#int f0/1
R3(config-if)#no shut
R3(config-if)#ip add 192.168.2.3 255.255.255.0
R3(config-if)#
*Mar  1 00:46:07.107: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:46:08.107: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R3(config-if)#exit
R3(config)#ip vrf RED
R3(config-vrf)#rd 4:4
R3(config-vrf)#route-target both 4:4
R3(config-vrf)#int f0/1
R3(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.2.3 removed due to enabling VRF RED
R3(config-if)#exit
R3(config)#int f0/1
R3(config-if)#ip address 192.168.2.1 255.255.255.0
R3(config-if)#
```

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

```
R3#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
 ip vrf forwarding RED
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
end
```

```
R3(config)#int f0/1
R3(config-if)#ip ospf 2 area 2
R3(config-if)#
*Mar  1 01:18:43.643: %OSPF-5-ADJCHG: Process 2, Nbr 6.6.6.6 on FastEthernet0/1 from LOADING to FULL, Loading Done
R3(config-if)#exit
R3(config)#exit
R3#
*Mar  1 01:19:09.079: %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 [110/11] via 192.168.2.6, 00:00:14, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/1
```

R1,R2,R3 form the MPLS Core and are running OSPF with all loopbacks running a /32 address and all have full connectivity. R1 and R3 are peering with MP-BGP. LDP is enabled on all the internal interfaces. The external interfaces of the MPLS core have been placed into a VRF called RED and then a site router has been joined to that VRF on each side of the MPLS core – (These represent a small office)

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

```
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
```

Check the routes on R1

```
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 [110/11] via 10.0.0.2, 01:26:06, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/21] via 10.0.0.2, 01:25:20, 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 [110/20] via 10.0.0.2, 01:26:06, FastEthernet0/0
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 [110/11] via 192.168.1.4, 01:02:18, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/1
```

Here you can see 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

```
R1(config)#router bgp 1
R1(config-router)#address-family ipv4 vrf RED
R1(config-router-af)#redistribute ospf 2
R1(config-router-af)#
```

Redistribute OSPF into MP-BGP on R3

```
R3(config)#router bgp 1
R3(config-router)#address-family ipv4 vrf RED
R3(config-router-af)#redistribute ospf 2
R3(config-router-af)#
```

This has enabled redistribution of the OSPF routes into BGP.

We can check the routes from R4 and R5 are now showing in the BGP table for their VRF with this command.

```
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             11         32768 ?
*>i6.6.6.6/32       3.3.3.3                 11    100      0 ?
*> 192.168.1.0      0.0.0.0                  0         32768 ?
*>i192.168.2.0      3.3.3.3                  0    100      0 ?
R1#
```

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
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                 11    100      0 ?
*> 6.6.6.6/32       192.168.2.6             11         32768 ?
*>i192.168.1.0      1.1.1.1                  0    100      0 ?
*> 192.168.2.0      0.0.0.0                  0         32768 ?
R3#
```

Which it is! 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

```
R1(config)#router ospf 2
R1(config-router)#redistribute bgp 1 subnets
R1(config-router)#
```

```
R3#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 2
R3(config-router)#redistribute bgp 1 subnets
R3(config-router)#
```

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
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 [110/21] via 192.168.1.1, 00:10:19, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O IA 192.168.2.0/24 [110/11] via 192.168.1.1, 00:10:19, FastEthernet0/0
R4#
```

Great we have 6.6.6.6 in there.

Also check the routing table on R5

```
R5#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 [110/21] via 192.168.2.1, 00:04:54, 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 [110/11] via 192.168.2.1, 00:04:54, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
R5#
```

Brilliant we have 4.4.4.4 in there so we should be able to ping across the MPLS. Which we can – to prove this is going over the MPLS and be label switched and not routed, lets do a trace

```
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 = 292/331/396 ms
R4#trace 6.6.6.6

Type escape sequence to abort.
Tracing the route to 6.6.6.6

  1 192.168.1.1 56 msec 112 msec 4 msec
  2 10.0.0.2 [MPLS: Labels 17/19 Exp 0] 308 msec 384 msec 260 msec
  3 192.168.2.1 [MPLS: Label 19 Exp 0] 224 msec 272 msec 380 msec
  4 192.168.2.6 392 msec 396 msec 368 msec
R4#
```