Modern Networking

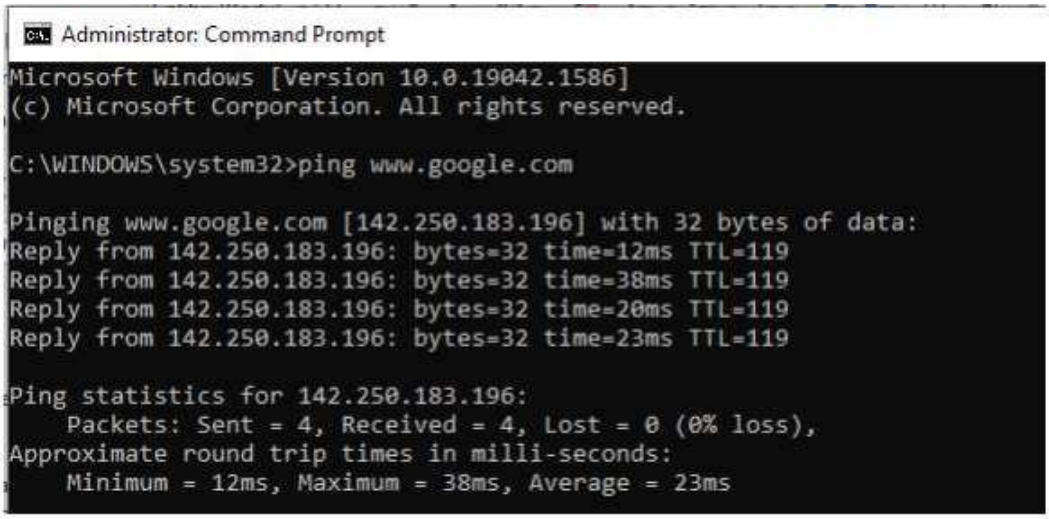
INDEX

Sr. No	Practical Name	Sign
1.	A. Networking Commands. B. Wi-Fi Hacking Password. C. Simple Packet Tracer.	
2.	Configure IP SLA Tracking and Path Control Topology	
3.	Using the AS_PATH Attribute.	
4.	Configuring IBGP and EBGP Sessions, Local Preference, and MED	
5.	Secure the Management Plane	
6.	Configure and Verify Path Control Using PBR	
7.	IP Service Level Agreements and Remote SPAN in a Campus Environment	
8.	Inter-VLAN Routing	
9.	Simulating MPLS environment	
10.	Simulating SDN with • Open Day light SDN Controller with the Mininet NetworkEmulator • OF Net SDN network emulator	

Practical No: 01

1. Command line commands for modern networking

Command	Descriptions
Ping www.google.com	To check the connection of internet.
netstat arp –a aep -g	Show the statistic, active connection.
Nbtstat	Show physical statistics.
hostname	Shows the physical devices.
trancert	Shows IP and MAC (Server side command).
ipconfig	Show configuration.
route	Show the root.
pathping	192.168.1.1 Check IP address from PC.
netdiag	Connect to network (server side command).
getmac	Show the addresses which PC's are connected.
nslookup	Show the mac address. IP (To back press ctrl + c).
taskkill	Shows not working IP address.
systeminfo	Show system information.
netview	Find the server side command.



Administrator: Command Prompt

```
C:\WINDOWS\system32>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49689        kubernetes:49690       ESTABLISHED
  TCP    127.0.0.1:49690        kubernetes:49689       ESTABLISHED
  TCP    127.0.0.1:49691        kubernetes:49692       ESTABLISHED
  TCP    127.0.0.1:49692        kubernetes:49691       ESTABLISHED
  TCP    192.168.1.206:56844    129.227.216.7:https    CLOSE_WAIT
  TCP    192.168.1.206:56845    129.227.216.7:https    CLOSE_WAIT
  TCP    192.168.1.206:59024    129.227.216.7:https    CLOSE_WAIT
  TCP    192.168.1.206:60333    20.198.162.76:https    ESTABLISHED
  TCP    192.168.1.206:60345    relay-ced30f61:http    ESTABLISHED
  TCP    192.168.1.206:60349    se-in-f188:5228        ESTABLISHED
  TCP    192.168.1.206:60390    a23-10-42-207:https    CLOSE_WAIT
  TCP    192.168.1.206:60397    104.18.107.46:https    CLOSE_WAIT
  TCP    192.168.1.206:60406    104.18.107.46:https    CLOSE_WAIT
  TCP    192.168.1.206:60460    104.18.108.46:https    CLOSE_WAIT
  TCP    192.168.1.206:60500    13.107.21.200:https    ESTABLISHED
  TCP    192.168.1.206:60502    52.113.196.254:https   ESTABLISHED
  TCP    192.168.1.206:60503    23.50.252.69:https     CLOSE_WAIT
  TCP    192.168.1.206:60504    13.107.213.48:https    ESTABLISHED
  TCP    192.168.1.206:60505    204.79.197.222:https   ESTABLISHED
  TCP    192.168.1.206:60506    bom12s09-in-f5:https   TIME_WAIT
  TCP    192.168.1.206:60507    a118-214-128-52:http   TIME_WAIT
```

```
C:\WINDOWS\system32>arp -a

Interface: 192.168.56.1 --- 0x8
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.1.206 --- 0xa
  Internet Address      Physical Address      Type
  192.168.1.1           a0-47-d7-31-c0-58     dynamic
  192.168.1.201         e0-37-bf-0c-11-65     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

```
C:\WINDOWS\system32>arp -g

Interface: 192.168.56.1 --- 0x8
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.1.206 --- 0xa
  Internet Address      Physical Address      Type
  192.168.1.1           a0-47-d7-31-c0-58     dynamic
  192.168.1.201         e0-37-bf-0c-11-65     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

```
C:\WINDOWS\system32>nbtstat

Displays protocol statistics and current TCP/IP connections using NBT
(NetBIOS over TCP/IP).

NBTSTAT [ [-a RemoteName] [-A IP address] [-c] [-n]
        [-r] [-R] [-RR] [-s] [-S] [interval] ]

  -a   (adapter status) Lists the remote machine's name table given its name
  -A   (Adapter status) Lists the remote machine's name table given its
                        IP address.
  -c   (cache)          Lists NBT's cache of remote [machine] names and their IP addresses
  -n   (names)          Lists local NetBIOS names.
  -r   (resolved)       Lists names resolved by broadcast and via WINS
  -R   (Reload)         Purges and reloads the remote cache name table
  -S   (Sessions)       Lists sessions table with the destination IP addresses
  -s   (sessions)       Lists sessions table converting destination IP
                        addresses to computer NETBIOS names.
  -RR  (ReleaseRefresh) Sends Name Release packets to WINS and then, starts Refresh

  RemoteName   Remote host machine name.
  IP address   Dotted decimal representation of the IP address.
  interval     Redisplays selected statistics, pausing interval seconds
               between each display. Press Ctrl+C to stop redisplaying
               statistics.
```

```
C:\WINDOWS\system32>hostname
DESKTOP-DRDLPDC
```

```
C:\WINDOWS\system32>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
```

```
C:\WINDOWS\system32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::ddc0:eb13:7b19:7ce0%8
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:
```

```
   Connection-specific DNS Suffix  . : iball.wifi.net
   Link-local IPv6 Address . . . . . : fe80::31d5:de51:e9e0:cd13%10
   IPv4 Address. . . . . . . . . . . : 192.168.1.206
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

```
C:\WINDOWS\system32>route

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
                  [MASK netmask]  [gateway] [METRIC metric]  [IF interface]

  -f           Clears the routing tables of all gateway entries.  If this is
               used in conjunction with one of the commands, the tables are
               cleared prior to running the command.

  -p           When used with the ADD command, makes a route persistent across
               boots of the system. By default, routes are not preserved
               when the system is restarted. Ignored for all other commands,
               which always affect the appropriate persistent routes.

  -4           Force using IPv4.

  -6           Force using IPv6.

  command      One of these:
                 PRINT     Prints  a route
                 ADD       Adds    a route
                 DELETE    Deletes a route
                 CHANGE    Modifies an existing route
  destination  Specifies the host.
  MASK         Specifies that the next parameter is the 'netmask' value.
  netmask      Specifies a subnet mask value for this route entry.
```

```
               If not specified, it defaults to 255.255.255.255.
  gateway      Specifies gateway.
  interface    the interface number for the specified route.
  METRIC       specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
and '?' matches any one char. Examples: 157.*.1, 157.*, 127.*, *224*.

Pattern match is only allowed in PRINT command.
Diagnostic Notes:
    Invalid MASK generates an error, that is when (DEST & MASK) != DEST.
    Example> route ADD 157.0.0.0 MASK 155.0.0.0 157.55.80.1 IF 1
             The route addition failed: The specified mask parameter is invalid. (Destination & Mask) != Destination.

Examples:

    > route PRINT
    > route PRINT -4
    > route PRINT -6
    > route PRINT 157*                .... Only prints those matching 157*
```

```
    > route ADD 157.0.0.0 MASK 255.0.0.0  157.55.80.1 METRIC 3 IF 2
              destination^      ^mask      ^gateway     metric^    ^
                                                        Interface^
      If IF is not given, it tries to find the best interface for a given
      gateway.
    > route ADD 3ffe::/32 3ffe::1

    > route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2

      CHANGE is used to modify gateway and/or metric only.

    > route DELETE 157.0.0.0
    > route DELETE 3ffe::/32
```

```
C:\WINDOWS\system32>pathping

Usage: pathping [-g host-list] [-h maximum_hops] [-i address] [-n]
                [-p period] [-q num_queries] [-w timeout]
                [-4] [-6] target_name

Options:
    -g host-list     Loose source route along host-list.
    -h maximum_hops  Maximum number of hops to search for target.
    -i address       Use the specified source address.
    -n               Do not resolve addresses to hostnames.
    -p period        Wait period milliseconds between pings.
    -q num_queries   Number of queries per hop.
    -w timeout       Wait timeout milliseconds for each reply.
    -4               Force using IPv4.
    -6               Force using IPv6.
```

```
C:\WINDOWS\system32>net diag
The syntax of this command is:

NET
    [ ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |
      HELPMSG | LOCALGROUP | PAUSE | SESSION | SHARE | START |
      STATISTICS | STOP | TIME | USE | USER | VIEW ]
```

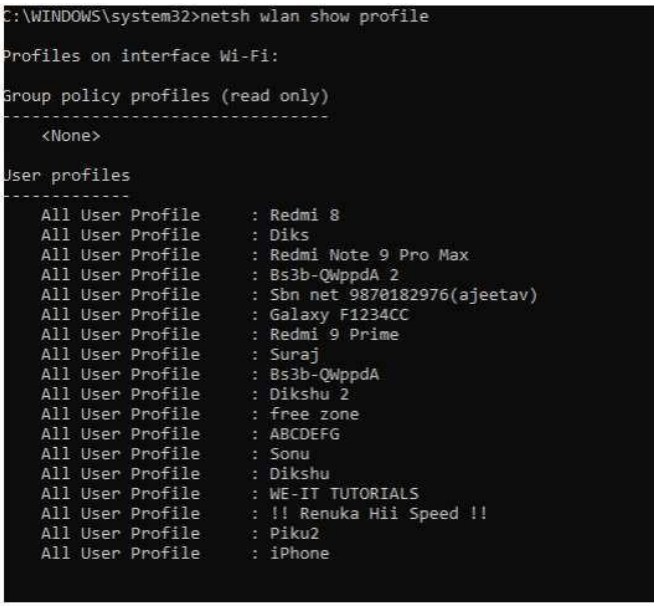
```
C:\WINDOWS\system32>getmac

Physical Address    Transport Name
=================== ==========================================================
44-1C-A8-A8-2D-1B   \Device\Tcpip_{6840A5C8-9AE0-413E-9D87-FE8E51D40F31}
A0-8C-FD-77-C3-F7   Media disconnected
44-1C-A8-A8-2D-1C   Media disconnected
0A-00-27-00-00-08   \Device\Tcpip_{32D57D00-56F0-4F56-BDB9-7261FABDE1C1}
```

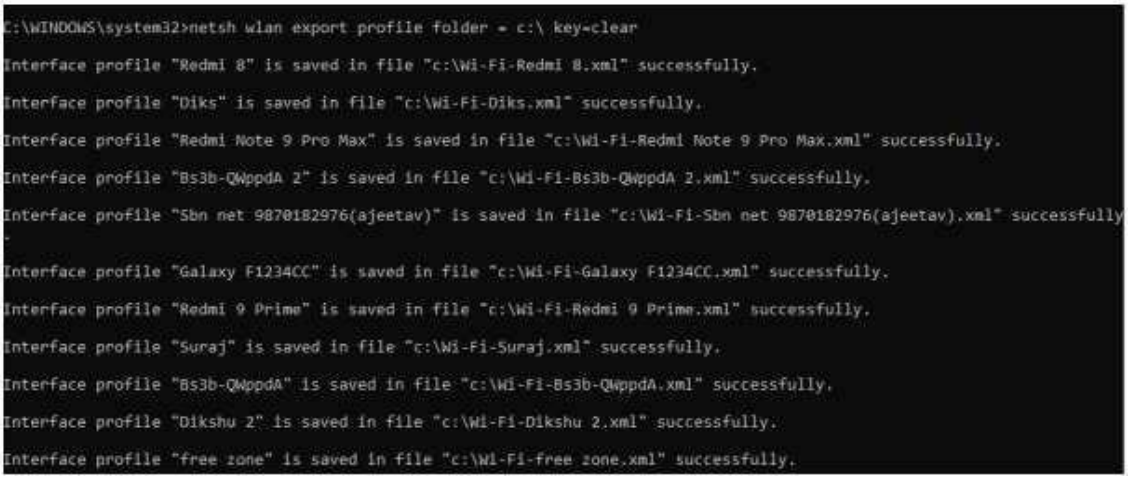
```
C:\WINDOWS\system32>nslookup
Default Server:  UnKnown
Address:  192.168.1.1
```

```
C:\WINDOWS\system32>systeminfo

Host Name:                 DESKTOP-DRDLPDC
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.19042 N/A Build 19042
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          hp
Registered Organization:
Product ID:                00331-10000-00001-AA379
Original Install Date:     24-04-2021, 19:59:17
System Boot Time:          01-04-2022, 21:15:20
System Manufacturer:       HP
System Model:              HP Notebook
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 61 Stepping 4 GenuineIntel ~1400 Mhz
BIOS Version:              Insyde F.09, 19-05-2016
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              00004009
Time Zone:                 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory:     4,008 MB
Available Physical Memory: 899 MB
Virtual Memory: Max Size:  8,519 MB
Virtual Memory: Available: 2,921 MB
Virtual Memory: In Use:    5,598 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\DESKTOP-DRDLPDC
Hotfix(s):                 9 Hotfix(s) Installed.
                           [01]: KB5009467
                           [02]: KB4562830
                           [03]: KB4577586
                           [04]: KB4580325
                           [05]: KB5011487
                           [06]: KB5006753
                           [07]: KB5007273
                           [08]: KB5011352
                           [09]: KB5005699
Network Card(s):           4 NIC(s) Installed.
                           [01]: Broadcom 8CM43142 802.11 bgn Wi-Fi M.2 Adapter
                                 Connection Name: Wi-Fi
                                 DHCP Enabled:    Yes
                                 DHCP Server:     192.168.1.1
                                 IP address(es)
                                 [01]: 192.168.1.206
                                 [02]: fe80::31d5:de51:e9e0:cd13
                           [02]: Realtek PCIe FE Family Controller
                                 Connection Name: Ethernet
                                 Status:          Media disconnected
                           [03]: Bluetooth Device (Personal Area Network)
                                 Connection Name: Bluetooth Network Connection
                                 Status:          Media disconnected
                           [04]: VirtualBox Host-Only Ethernet Adapter
                                 Connection Name: VirtualBox Host-Only Network
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 192.168.56.1
                                 [02]: fe80::ddc0:eb13:7b19:7ce0
Hyper-V Requirements:      VM Monitor Mode Extensions: Yes
                           Virtualization Enabled In Firmware: No
                           Second Level Address Translation: Yes
                           Data Execution Prevention Available: Yes
```

2. Wi-Fi Hacking Password.

1. Open cmd with administrator mode.
2. Type the command netsh wlan show profile → Shows nearby list of Wi-Fi connection.
3. To check the password use the command → netsh wlan export profilefolder = c:\ key=clear.
4. The above command provides all the profile text file under c:\ drive.
5. To verify/check the password → go to the c:\ drive → select any profile → under key Material XML tag → actual password.





```
Interface profile "free zone" is saved in file "c:\Wi-Fi-free zone.xml" successfully.

Interface profile "ABCDEFG" is saved in file "c:\Wi-Fi-ABCDEFG.xml" successfully.

Interface profile "Sonu" is saved in file "c:\Wi-Fi-Sonu.xml" successfully.

Interface profile "Dikshu" is saved in file "c:\Wi-Fi-Dikshu.xml" successfully.

Interface profile "WE-IT TUTORIALS" is saved in file "c:\Wi-Fi-WE-IT TUTORIALS.xml" successfully.

Interface profile "!! Renuka Hii Speed !!" is saved in file "c:\Wi-Fi-!! Renuka Hii Speed !!.xml" successfully.

Interface profile "Piku2" is saved in file "c:\Wi-Fi-Piku2.xml" successfully.

Interface profile "iPhone" is saved in file "c:\Wi-Fi-iPhone.xml" successfully.
```

Wi-Fi-Redmi 8 - Notepad

File Edit Format View Help

```
<?xml version="1.0"?>
<WLANProfile xmlns="http://www.microsoft.com/networking/WLAN/profile/v1">
        <name>Redmi 8</name>
        <SSIDConfig>
                <SSID>
                        <hex>5265646D692038</hex>
                        <name>Redmi 8</name>
                </SSID>
        </SSIDConfig>
        <connectionType>ESS</connectionType>
        <connectionMode>auto</connectionMode>
        <MSM>
                <security>
                        <authEncryption>
                                <authentication>WPA2PSK</authentication>
                                <encryption>AES</encryption>
                                <useOneX>false</useOneX>
                        </authEncryption>
                        <sharedKey>
                                <keyType>passPhrase</keyType>
                                <protected>false</protected>
                                <keyMaterial>123456789</keyMaterial>
                        </sharedKey>
                </security>
        </MSM>
```

3. Simple Packet Tracer.

Two PC Connection Establishment:

1. Connects two or more different LAN's.
2. It is a layer 3(Network layer) device.
3. Store routing table.
4. Router - Inevitable device in the internet.

Process:

1. Drag two switches (2960) for two local area networks.
2. Put 3 pc's in each network.
3. Use Ethernet cable to connect the pc's with the switches.

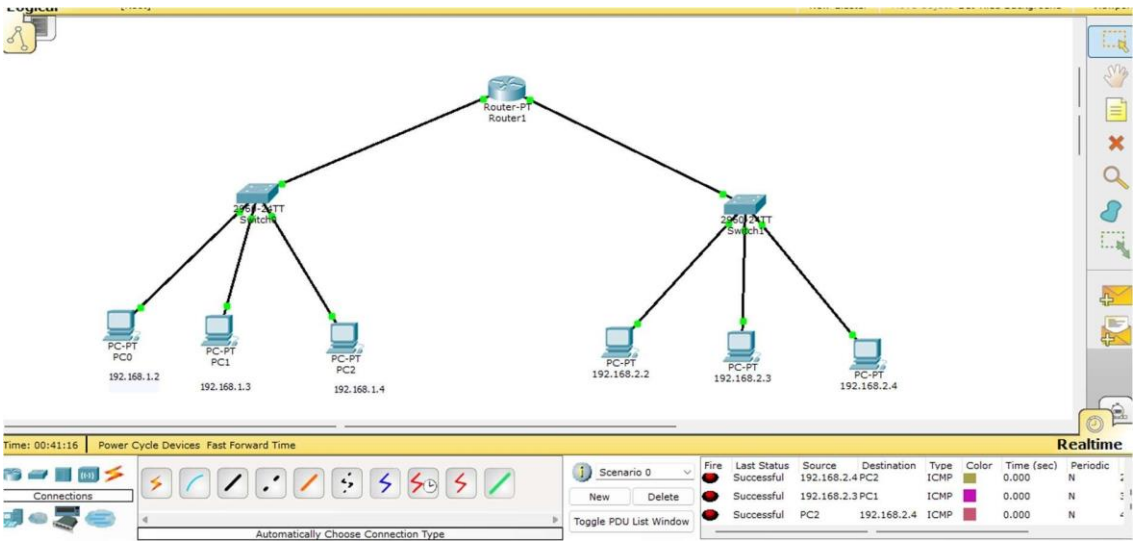
Cisco IOS Console 1:

```
Router>en
Router#
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int f0/1
%Invalid interface type and number
Router(config)#int f0/0
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#no sh

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

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

Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

FastEthernet0/0        192.168.1.1     YES manual up                    up

FastEthernet1/0        unassigned      YES unset  administratively down down

Serial2/0              unassigned      YES unset  administratively down down

Serial3/0              unassigned      YES unset  administratively down down

FastEthernet4/0        unassigned      YES unset  administratively down down

FastEthernet5/0        unassigned      YES unset  administratively down down
```

Cisco IOS Console 2:

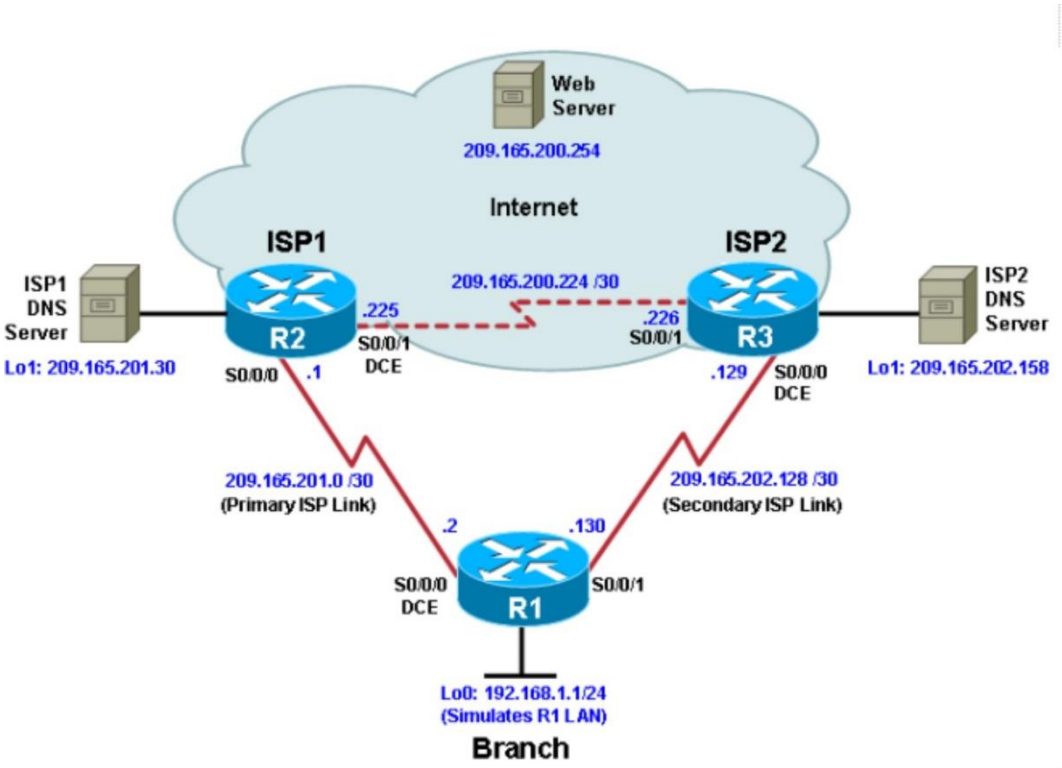
```
Serial3/0              unassigned      YES unset  administratively down down

FastEthernet4/0        unassigned      YES unset  administratively down down

FastEthernet5/0        unassigned      YES unset  administratively down down
Router#
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int f0/1
%Invalid interface type and number
Router(config)#int f1/0
Router(config-if)#ip add 192.168.2.1 255.255.255.0
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state t
o up

Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

FastEthernet0/0        192.168.1.1     YES manual up                    up

FastEthernet1/0        192.168.2.1     YES manual up                    up

Serial2/0              unassigned      YES unset  administratively down down

Serial3/0              unassigned      YES unset  administratively down down

FastEthernet4/0        unassigned      YES unset  administratively down down

FastEthernet5/0        unassigned      YES unset  administratively down down
Router#
```

OUTPUT:-



Practical No.: 02
Configure IP SLA Tracking and Path Control

Topology:-



Objectives:-

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

Required Resources:-

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

Steps:-

Step 01:- Configure loopbacks and assign address

Step 02:- Configure static routing

Step 03:- Configure IP SLA probes

Step 04:- Configure tracking option

Step 05:- Verify IP SLA operation

Step 01:- Configure loopbacks and assign address

a. 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.

Router R1:-

```
R1#conf term
R1(config)#hostname R1

R1(config)#interface Loopback 0
R1(config-if)#description R1 LAN
R1(config-if)#ip address 192.168.1.1 255.255.255.0

R1(config-if)#interface Serial4/0
R1(config-if)#description R1 --> ISP1
R1(config-if)#ip address 209.165.201.2 255.255.255.252
R1(config-if)#clock rate 128000
R1(config-if)#bandwidth 128
R1(config-if)#no shutdown

R1(config-if)#interface Serial4/2
R1(config-if)#description R1 --> ISP2
R1(config-if)#ip address 209.165.202.130 255.255.255.252
R1(config-if)#bandwidth 128
R1(config-if)#no shutdown
```

Router R2:-

```
R2#conf term
R2(config)#hostname ISP1

ISP1(config)#interface Loopback0
ISP1(config-if)#description Simulated Internet Web Server
ISP1(config-if)#ip address 209.165.200.254 255.255.255.255

ISP1(config-if)#interface Loopback1
ISP1(config-if)#description ISP1 DNS Server
ISP1(config-if)#ip address 209.165.201.30 255.255.255.255

ISP1(config-if)#interface Serial4/0
ISP1(config-if)#description ISP1 --> R1
ISP1(config-if)#ip address 209.165.201.1 255.255.255.252
ISP1(config-if)#bandwidth 128
ISP1(config-if)#no shutdown

ISP1(config-if)#interface Serial4/1
ISP1(config-if)#description ISP1 --> ISP2
ISP1(config-if)#ip address 209.165.200.225 255.255.255.252
ISP1(config-if)#clock rate 128000
ISP1(config-if)#bandwidth 128
ISP1(config-if)#no shutdown
```

Router R3:-

```
R3#conf term
R3(config)#hostname ISP2

ISP2(config)#interface Loopback0
ISP2(config-if)#description Simulated Internet Web Server
ISP2(config-if)#ip address 209.165.200.254 255.255.255.255

ISP2(config-if)#interface Loopback1
ISP2(config-if)#description ISP2 DNS Server
ISP2(config-if)#ip address 209.165.202.158 255.255.255.255

ISP2(config-if)#interface Serial4/2
ISP2(config-if)#description ISP2 --> R1
ISP2(config-if)#ip address 209.165.202.129 255.255.255.252
ISP2(config-if)#clock rate 128000
ISP2(config-if)#bandwidth 128
ISP2(config-if)#no shutdown

ISP2(config-if)#interface Serial4/1
ISP2(config-if)#description ISP2 --> ISP1
ISP2(config-if)#ip address 209.165.200.226 255.255.255.252
ISP2(config-if)#bandwidth 128
ISP2(config-if)#no shutdown
```

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

```
R1# show interfaces description
```

Interface	Status	Protocol	Description
Se4/0	up	up	R1 --> ISP1
Se4/2	up	up	R1 --> ISP2
Lo0	up	up	R1 LAN

All three interfaces should be active. Troubleshoot if necessary.

Step 02:- 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
```

Router 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)#ip route 192.168.1.0 255.255.255.0 209.165.201.2
```

Router 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

ISP1(config-router)#exit

ISP2(config-router)#ip route 192.168.1.0 255.255.255.0 209.165.202.130
ISP2(config)#end
```

b. 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.

Router R1:-

```
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#209.165.200.254
+>(tcl)#209.165.201.30
+>(tcl)#209.165.202.158
+>(tcl)#} {
+>(tcl)#ping $address source 192.168.1.1
+>(tcl)#}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.254, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/40 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.30, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/36 ms
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 209.165.202.158, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/45/52 ms
```

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

```
R1(tcl)#foreach address {
+>(tcl)#209.165.200.254
+>(tcl)#209.165.201.30
+>(tcl)#209.165.202.158
+>(tcl)#} {
+>(tcl)#trace $address source 192.168.1.1
+>(tcl)#}
Type escape sequence to abort.
Tracing the route to 209.165.202.158
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.201.1 20 msec 28 msec 28 msec
  2 209.165.200.226 44 msec 48 msec 44 msec
R1(tcl)#end
invalid command name "end"          ^
% Invalid input detected at '^' marker.

R1(tcl)#exit
```

Step 03:- 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 11
R1(config-ip-sla)#icmp-echo 209.165.201.30
R1(config-ip-sla-echo)#frequency 10
R1(config-ip-sla-echo)#exit
R1(config)#ip sla schedule 11 life forever start-time now
R1(config)#end
```

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

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

Operation frequency (seconds): 10 (not considered if randomly scheduled)
Next Scheduled Start Time: Start Time already passed
Group Scheduled : FALSE
Randomly Scheduled : FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
R1#

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

R1#**show ip sla statistics**

IPSLAs Latest Operation Statistics

IPSLA operation id: 11
Latest RTT: 8 milliseconds
Latest operation start time: 10:33:18 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 51
Number of failures: 0
Operation time to live: Forever

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

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

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

R1#**show ip sla configuration 22**
IP SLAs Infrastructure Engine-III
Entry number: 22
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.202.158/0.0.0.0

Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Verify data: No
Vrf Name:
Schedule:
Operation frequency (seconds): 10 (not considered if randomly scheduled)
Next Scheduled Start Time: Start Time already passed
Group Scheduled : FALSE
Randomly Scheduled : FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept: 2

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

IPSLA operation id: 22
Latest RTT: 60 milliseconds
Latest operation start time: 11:30:44 UTC Tue Mar 14 2023
Latest operation return code: OK
Number of successes: 7
Number of failures: 0
Operation time to live: Forever

Step 04:- Configure tracking option

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**
Codes: L - local, 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, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 209.165.201.1
192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.1.0/24 is directly connected, Loopback0
L 192.168.1.1/32 is directly connected, Loopback0
209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C 209.165.201.0/30 is directly connected, Serial4/0
L 209.165.201.2/32 is directly connected, Serial4/0
209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C 209.165.202.128/30 is directly connected, Serial4/2
L 209.165.202.130/32 is directly connected, Serial4/2

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**

d. Specify the level of sensitivity to changes of tracked objects to 10 seconds of down delay and 1 second of up delay using the **delay down 10 up 1** command.

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

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

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

f. Configure the floating static route that will be implemented when tracking object 1 is active. Use the **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1** command to create a floating static default route via 209.165.201.1 (ISP1).

R1(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1**
Mar 14 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :
via 209.165.201.1 0 1048578

Mar 14 10:45:39.119: RT: closer admin distance for 0.0.0.0, flushing 1 routes
Mar 1410:45:39.119: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]
Mar 14 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :
via 209.165.201.1 0 1048578

Mar 14 10:45:39.119: RT: rib update return code: 17
Mar 1410:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :
via 209.165.201.1 0 1048578

Mar 14 10:45:39.119: RT: rib update return code: 17

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

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

h. Verify the routing table again.

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

S*    0.0.0.0/0 [2/0] via 209.165.201.1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30 is directly connected, Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
```

Step 05:- Verify IP SLA operation.

a. On ISP1, disable the loopback interface 1.

```
ISP1(config-if)# int lo1
ISP1(config-if)# shutdown
ISP1(config-if)#
Mar 14 10:53:25.091: %LINK-5-CHANGED: Interface Loopback1, changed state to
administratively down
Mar 14 10:53:26.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to down
ISP1(config-if)#
```

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

```
R1#
Jan 10 10:53:59.551: %TRACK-6-STATE: 1 ip sla 11 reachability Up -> Down
Jan 10 10:53:59.551: RT: del 0.0.0.0 via 209.165.201.1, static metric [2/0]
Jan 10 10:53:59.551: RT: delete network route to 0.0.0.0/0
Jan 10 10:53:59.551: RT: default path has been cleared
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)  :
   via 209.165.202.129  0 1048578

Jan 10 10:53:59.551: RT: add 0.0.0.0/0 via 209.165.202.129, static metric [3/0]
Jan 10 10:53:59.551: RT: default path is now 0.0.0.0 via 209.165.202.129
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)  :
   via 209.165.201.1 0 1048578
```

Jan 10 10:53:59.551: RT: rib update return code: 17
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0) :
via 209.165.202.129 0 1048578

Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0) :
via 209.165.201.1 0 1048578

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

c. On R1, verify the routing table.

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

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

d. Verify the IP SLA statistics.

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

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

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

R1#

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

```
R1# trace 209.165.200.254 source 192.168.1.1
Type escape sequence to abort.
Tracing the route to 209.165.200.254
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.202.129 4 msec *  *
R1#
```

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

```
ISP1(config-if)# no shutdown
Jan 10 11:05:45.847: %LINK-3-UPDOWN: Interface Loopback1, changed state to up
Jan 10 11:05:46.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to up
ISP1(config-if)#
```

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

```
R1#
Jan 10 11:06:20.551: %TRACK-6-STATE: 1 ip sla 11 reachability Down -> Up
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)  :
   via 209.165.201.1  0 1048578

Jan 10 11:06:20.551: RT: closer admin distance for 0.0.0.0, flushing 1 routes
Jan 10 11:06:20.551: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)  :
   via 209.165.202.129  0 1048578

Jan 10 11:06:20.551: RT: rib update return code: 17
Jan 10 11:06:20.551: RT: u
R1#pdating static 0.0.0.0/0 (0x0) :
   via 209.165.202.129  0 1048578

Jan 10 11:06:20.551: RT: rib update return code: 17
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)  :
   via 209.165.201.1 0 1048578

Jan 10 11:06:20.551: RT: rib update return code: 17
R1#
```

g. Again examine the IP SLA statistics.

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

IPSLA operation id: 11
        Latest RTT: 8 milliseconds
Latest operation start time: 11:07:38 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 182
Number of failures: 75
Operation time to live: Forever
```

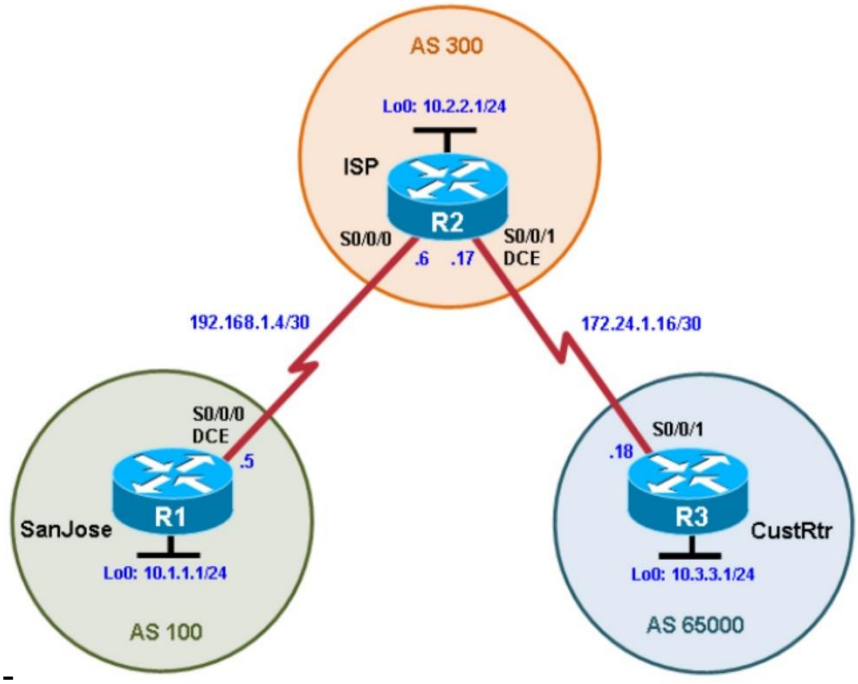
```
IPSLA operation id: 22
        Latest RTT: 16 milliseconds
Latest operation start time: 11:07:39 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 257
Number of failures: 0
Operation time to live: Forever


R1#
```

h. Verify the routing table.

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

S*    0.0.0.0/0 [2/0] via 209.165.201.1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30 is directly connected, Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
R1#
```

Practical No.: 03
Using the AS_PATH Attribute

Topology:-



Objectives:-

- Use BGP commands to prevent private AS numbers from being advertised to the outside world.
- Use the AS_PATH attribute to filter BGP routes based on their source AS numbers.

Required Resources:-

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Steps:-

Step 1: Prepare the routers for the lab.

Step 2: Configure the hostname and interface addresses.

Step 3: Configure BGP

Step 4: Remove the private AS

Step 5: Use the AS_PATH attribute to filter routes.

Step 1:-Prepare the routers for the lab.

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

Step 2: Configure the hostname and interface addresses.

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

Router R1 (hostname SanJose)

```
R1(config)#hostname SanJose
SanJose(config)#interface Loopback0
SanJose(config-if)#ip address 10.1.1.1 255.255.255.0
SanJose(config-if)#interface Serial4/0
SanJose(config-if)#ip address 192.168.1.5 255.255.255.252
SanJose(config-if)#clock rate 128000
SanJose(config-if)#no shutdown
```

Router R2 (hostname ISP)

```
R2(config)#hostname ISP
ISP(config)#interface Loopback0
ISP(config-if)#ip address 10.2.2.1 255.255.255.0
ISP(config)#interface Serial4/0
ISP(config-if)#ip address 192.168.1.6 255.255.255.252
ISP(config-if)#no shutdown
```

Router R3 (hostname CustRtr)

```
R3(config)#hostname CustRtr
CustRtr(config)#interface Loopback0
CustRtr(config-if)#ip address 10.3.3.1 255.255.255.0
CustRtr(config-if)#inerface Serial4/1
CustRtr(config-if)#ip address 172.24.1.18 255.255.255.252
CustRtr(config-if)#no shutdown
```

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

Step 3: Configure BGP.

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

```
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

ISP(config)# router bgp 300
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)# network 10.2.2.0 mask 255.255.255.0
```

```
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
```

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
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 00:02:05
<output omitted >

BGP neighbor is 192.168.1.5, remote AS 100, external link
   BGP version 4, remote router ID 10.1.1.1
   BGP state = Established, up for 00:04:19
<output omitted >
```

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#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.1.1.0/24 is directly connected, Loopback0
L        10.1.1.1/32 is directly connected, Loopback0
B        10.2.2.0/24 [20/0] via 192.168.1.6, 00:02:00
B        10.3.3.0/24 [20/0] via 192.168.1.6, 00:00:47
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.4/30 is directly connected, Serial4/0
L        192.168.1.5/32 is directly connected, Serial4/0
```

b. Ping the 10.3.3.1 address from SanJose

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

```
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:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/68 ms
SanJose# ping 10.3.3.1 source 10.1.1.1
or
SanJose# ping 10.3.3.1 source Lo0
```

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.

```
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, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     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
```

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

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

f. SanJose should be able to ping 10.3.3.1 using its loopback 0 interface as the source of the ping.

```
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 = 56/58/60 ms
```

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.

```
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, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     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
```

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)# ip as-path access-list 1 deny ^100$
ISP(config)# ip as-path access-list 1 permit .*
```

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

```
ISP(config)# router bgp 300
ISP(config-router)# neighbor 172.24.1.18 filter-list 1 out
```

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#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B        10.1.1.0/24 [20/0] via 192.168.1.5, 00:08:47
C        10.2.2.0/24 is directly connected, Loopback0
L      10.2.2.1/32 is directly connected, Loopback0B
10.3.3.0/24 [20/0] via 172.24.1.18, 00:06:54
      172.24.0.0/16 is variably subnetted, 2 subnets, 2 masks
C  172.24.1.16/30 is directly connected, Serial4/1
L    172.24.1.17/32    is    directly    connected,    Serial4/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.4/30 is directly connected, Serial4/0
L  192.168.1.6/32 is directly connected, Serial4/0
```

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

```
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, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

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

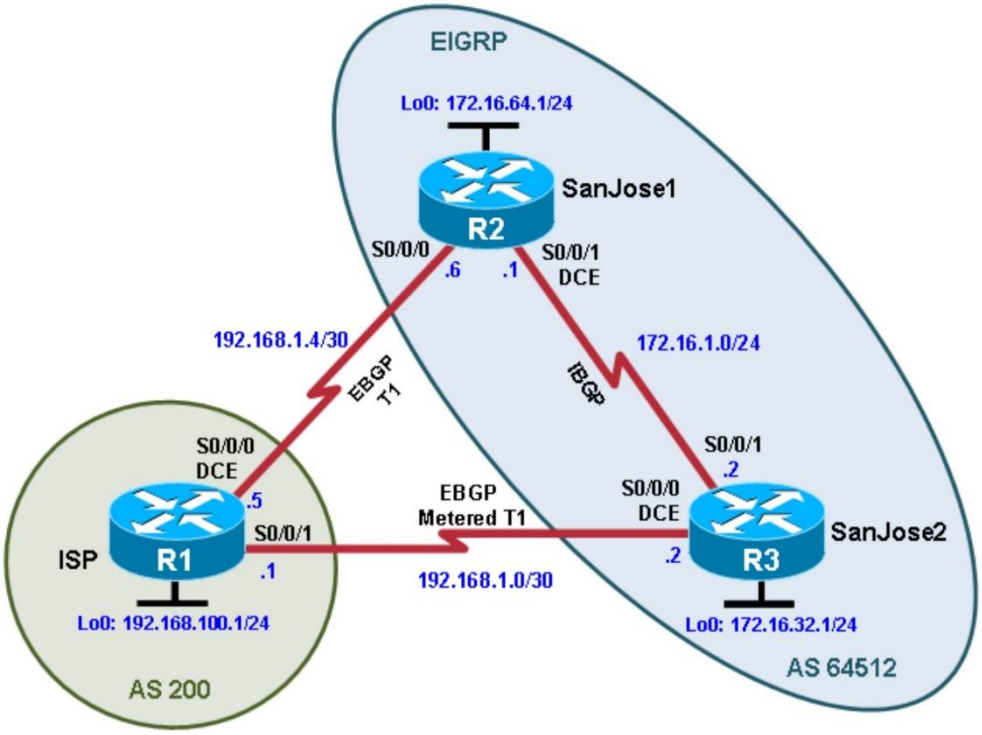
e. Run the following Tcl script on all routers to verify whether there is connectivity. All pings from ISP should be successful. SanJose should not be able to ping the CustRtr loopback 10.3.3.1 or the WAN link 172.24.1.16/30. CustRtr should not be able to ping the SanJose loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

```
ISP#tclsh
ISP(tcl)#foreach address {
+>10.1.1.1
+>10.2.2.1
+>10.3.3.1
+>192.168.1.5
+>192.168.1.6
+>172.24.1.17
+>172.24.1.18
+>} {
+>ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/31/40 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/27/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
ISP(tcl)#
```

Practical No.: 04

Configuring IBGP and EBGP Sessions, Local Preference, and MED

Topology:-



Objectives:-

- For IBGP peers to correctly exchange routing information, use the **next-hop-self** command with the **Local-Preference** and **MED** attributes.
- Ensure that the flat-rate, unlimited-use T1 link is used for sending and receiving data to and from the AS 200 on ISP and that the metered T1 only be used in the event that the primary T1 link has failed.

Required Resources

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

Steps:-

Step 0: Suggested starting configurations.

Step 1: Configure interface addresses.

Step 2: Configure EIGRP.

Step 3: Configure IBGP and verify BGP neighbors.

Step 4: Configure EBGP and verify BGP neighbors.

Step 5: View BGP summary output.

Step 6: Verify which path the traffic takes.

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

Step 8: Set BGP local preference.

Step 9: Set BGP MED.

Step 10: Establish a default route.

Step 0: Suggested starting configurations.

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

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

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 Serial4/0
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 Serial4/2
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
SanJose1#
```

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.

```
SanJose2# show ip bgp neighbors
BGP neighbor is 172.16.64.1, remote AS 64512, internal link
    BGP version 4, remote router ID 172.16.64.1
    BGP state = Established, up for 00:00:22
    Last read 00:00:22, last write 00:00:22, hold time is 180, keepalive interval is 60 seconds
<output omitted>
```

Step 4: Configure EBGP and verify BGP neighbors.

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

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

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.

```
SanJose1(config)# ip route 172.16.0.0 255.255.0.0 null0
```

c. Configure SanJose1 as an EBGP peer to ISP.

```
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 192.168.1.5 remote-as 200
SanJose1(config-router)# network 172.16.0.0
```

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

```
SanJose1# show ip bgp neighbors
BGP neighbor is 172.16.32.1, remote AS 64512, internal link
    BGP version 4, remote router ID 172.16.32.1
    BGP state = Established, up for 00:12:43
<output omitted>

BGP neighbor is 192.168.1.5, remote AS 200, external link
    BGP version 4, remote router ID 192.168.100.1
    BGP state = Established, up for 00:06:49
    Last read 00:00:42, last write 00:00:45, hold time is 180, keepalive interval is 60 seconds
<output omitted>
```

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 following.

```
SanJose2# show ip bgp summary
BGP router identifier 172.16.32.1, local AS number 64512
BGP table version is 6, main routing table version 6
2 network entries using 288 bytes of memory
4 path entries using 320 bytes of memory
4/2 BGP path/bestpath attribute entries using 640 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1272 total bytes of memory
BGP activity 2/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.64.1     4        64512      27      26        6    0    0 00:18:15        2
192.168.1.1     4          200      10       7        6    0    0 00:01:42        1
SanJose2#
```

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.

```
ISP# clear ip bgp *
ISP#
*Mar 23 22:05:32.427: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Down User reset
*Mar 23 22:05:32.427: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.2 IPv4 Unicast
topology base removed from session  User reset
*Mar 23 22:05:32.427: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Down User reset
*Mar 23 22:05:32.427: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.6 IPv4 Unicast
topology base removed from session  User reset
*Mar 23 22:05:32.851: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Up
*Mar 23 22:05:32.851: %BGP-
ISP#5-ADJCHANGE: neighbor 192.168.1.6 Up
ISP#
```

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.

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

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.

```
ISP# ping 172.16.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
ISP# ping 172.16.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/13/16 ms
ISP#
```

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

```
ISP# show ip bgp
BGP table version is 3, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

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

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**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms

ISP# **ping 172.16.32.1 source 192.168.100.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

ISP# **ping 172.16.1.2 source 192.168.100.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
ISP#

ISP# **ping 172.16.64.1 source 192.168.100.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms

You can also use the extended ping dialogue to specify the source address, as shown in this example.

ISP# **ping**
Protocol [ip]:
Target IP address: **172.16.64.1**
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: **y**
Source address or interface: **192.168.100.1**
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/24 ms
ISP#

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.

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.

```
ISP# show ip bgp
BGP table version is 5, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

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

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
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:52:03, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L     192.168.1.2/32 is directly connected, Serial0/0/0 B
192.168.1.4/30   [20/0]   via   192.168.1.1,   00:01:03   B
192.168.100.0/24 [20/0] via 192.168.1.1, 00:25:20
SanJose2#
```

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.

```
ISP(config)# router bgp 200
ISP(config-router)# no network 192.168.1.0 mask 255.255.255.252
ISP(config-router)# no network 192.168.1.4 mask 255.255.255.252
ISP(config-router)# exit
ISP(config)# interface serial 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
BGP table version is 1, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 * i 192.168.100.0    192.168.1.5              0    100      0 200 i
SanJose2#
```

```
SanJose2# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 02:41:46, Serial0/0/1
SanJose2#
```

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
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  172.16.0.0      0.0.0.0                  0         32768 i
 * i                 172.16.64.1              0    100      0 i
 *>i 192.168.100.0    172.16.64.1              0    100      0 200 i
SanJose2#
```

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
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 04:27:19, Serial0/0/1
B     192.168.100.0/24 [200/0] via 172.16.64.1, 00:00:46
SanJose2#
```

j. Before configuring the next BGP attribute, restore the WAN link between ISP and SanJose3.

```
ISP(config)# interface serial 0/0/1
ISP(config-if)# no shutdown
ISP(config-if)#

SanJose2# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 04:37:34, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
B     192.168.100.0/24 [20/0] via 192.168.1.1, 00:01:35
SanJose2#
```

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
BGP table version is 3, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.32.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>  192.168.100.0    192.168.1.5              0    150      0 200 i
SanJose1#

SanJose2# show ip bgp
BGP table version is 7, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 192.168.100.0    172.16.64.1              0    150      0 200 i
 *                    192.168.1.1              0    125      0 200 i
SanJose2#
```

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.

```
ISP# show ip bgp
BGP table version is 22, 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, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

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

```
ISP# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

B    172.16.0.0/16 [20/0] via 192.168.1.2, 00:12:45
     192.168.1.0/24 is variably subnetted, 4 subnets, 2 masks
C  192.168.1.0/30 is directly connected, Serial0/0/1
L       192.168.1.1/32 is directly connected, Serial0/0/1
C       192.168.1.4/30 is directly connected, Serial0/0/0
L       192.168.1.5/32 is directly connected, Serial0/0/0
     192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.100.0/24 is directly connected, Loopback0
L  192.168.100.1/32 is directly connected, Loopback0
ISP#
```

b. Use an extended **ping** command to verify this situation. Specify the **record** option and compare your output to the following. Notice the return path using the exit interface 192.168.1.1 to 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 (20 ms). Received packet has options
        Total option bytes= 40, padded length=40
        Record route:
          (172.16.1.2)
          (192.168.1.6)
          (192.168.100.1)
          (192.168.1.1)
          (172.16.32.1) <*>
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
        End of list
       Reply to request 1 (20 ms). Received packet has options
        Total option bytes= 40, padded length=40
        Record route:
          (172.16.1.2)
          (192.168.1.6)
          (192.168.100.1)
          (192.168.1.1)
          (172.16.32.1) <*>
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
        End of list

       Reply to request 2 (20 ms). Received packet has options
        Total option bytes= 40, padded length=40
        Record route:
          (172.16.1.2)
          (192.168.1.6)
          (192.168.100.1)
          (192.168.1.1)
          (172.16.32.1) <*>
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
        End of list

       Reply to request 3 (24 ms). Received packet has options
        Total option bytes= 40, padded length=40
        Record route:
          (172.16.1.2)
          (192.168.1.6)
          (192.168.100.1)
          (192.168.1.1)
          (172.16.32.1) <*>
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
          (0.0.0.0)
        End of list
```

```
Reply to request 4 (20 ms). Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list
```

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

```
SanJose1(config)#route-map PRIMARY_T1_MED_OUT permit 10
SanJose1(config-route-map)#set Metric 50
SanJose1(config-route-map)#exit
SanJose1(config)#router bgp 64512
SanJose1(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out

SanJose2(config)#route-map SECONDARY_T1_MED_OUT permit 10
SanJose2(config-route-map)#set Metric 75
SanJose2(config-route-map)#exit
SanJose2(config)#router bgp 64512
SanJose2(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
```

d. Use the **clear ip bgp * soft** command after issuing this new policy. Issuing the **show ip bgp** command as follows on SanJose1 or SanJose2 does not indicate anything about this newly defined policy.

```
SanJose1# clear ip bgp * soft
SanJose2# clear ip bgp * soft

SanJose1# show ip bgp
BGP table version is 4, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.32.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>  192.168.100.0    192.168.1.5              0    150      0 200 i
SanJose1#
```

e. Reissue an extended **ping** command with the **record** command. Notice the change in return path using the exit interface 192.168.1.5 to SanJose1.

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

Reply to request 1 (28 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.100.1)
(192.168.1.5)
(172.16.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list

Reply to request 2 (28 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.100.1)
(192.168.1.5)
(172.16.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list

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**
Codes: L - local, 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, H - NHRP, l - LISP
a - application route
+ - replicated route, % - next hop override

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

```
B*    0.0.0.0/0 [20/0] via 192.168.1.5, 00:00:36
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.1/32 is directly connected, Serial0/0/1
D        172.16.32.0/24 [90/2297856] via 172.16.1.2, 05:47:24, Serial0/0/1
C        172.16.64.0/24 is directly connected, Loopback0
L        172.16.64.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.4/30 is directly connected, Serial0/0/0
L        192.168.1.6/32 is directly connected, Serial0/0/0
SanJose1#
```

```
SanJose2# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override Gateway

of last resort is 172.16.64.1 to network 0.0.0.0
```

c. The preferred default route is by way of SanJose1 because of the higher local preference attribute configured on SanJose1 earlier.

```
SanJose2# show ip bgp
BGP table version is 38, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 0.0.0.0          172.16.64.1              0    150      0 200 i
 *                    192.168.1.1                   125      0 200 i
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 192.168.100.0    172.16.64.1              0    150      0 200 i
 *                    192.168.1.1              0    125      0 200 i
SanJose2#
```

d. 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
Type escape sequence to abort.
Tracing the route to 10.0.0.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.1.1 8 msec 4 msec 8 msec
  2 192.168.1.5 [AS 200] 12 msec *  12 msec
SanJose2#
```

e. 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
Type escape sequence to abort.
Tracing the route to 10.0.0.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.1.1 8 msec 4 msec 8 msec
  2 192.168.1.5 [AS 200] 12 msec *  12 msec
SanJose2#
```

f. 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)#
```

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

```
SanJose1# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override Gateway

of last resort is 172.16.32.1 to network 0.0.0.0

B*    0.0.0.0/0 [200/0] via 172.16.32.1, 00:00:06
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.1/32 is directly connected, Serial0/0/1
D        172.16.32.0/24 [90/2297856] via 172.16.1.2, 05:49:25, Serial0/0/1
C        172.16.64.0/24 is directly connected, Loopback0
L        172.16.64.1/32 is directly connected, Loopback0
B  192.168.100.0/24 [200/0] via 172.16.32.1, 00:00:06
SanJose1#

SanJose2# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override
```

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

```
B*    0.0.0.0/0 [20/0] via 192.168.1.1, 00:00:30
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 05:49:49, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
B     192.168.100.0/24 [20/0] via 192.168.1.1, 00:00:30
SanJose2#
```

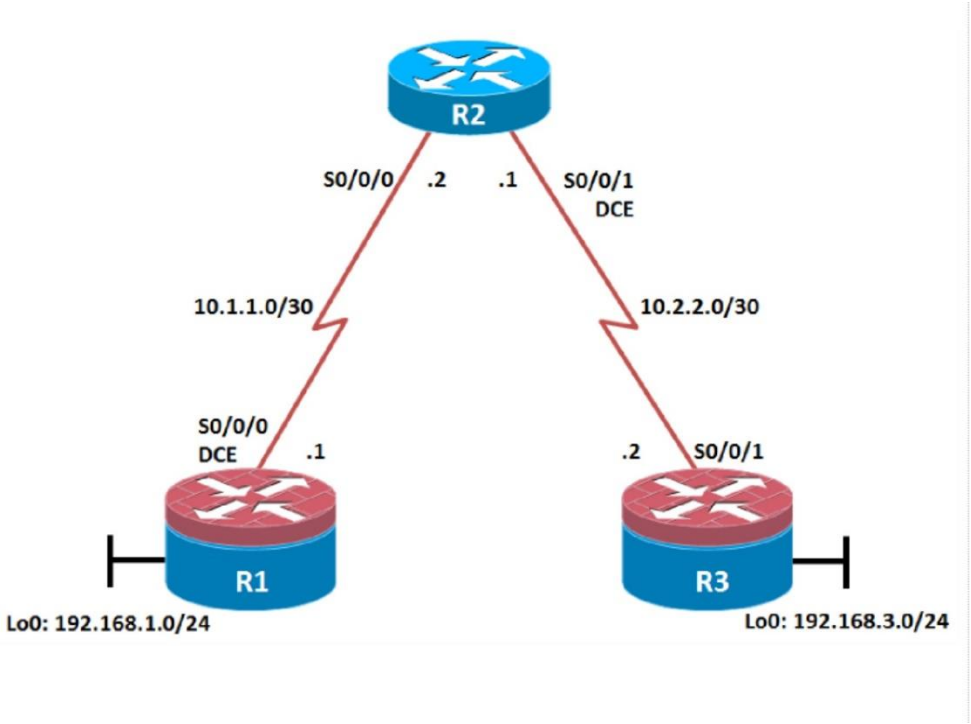
h. Verify the new path using the traceroute command to 10.0.0.1 from SanJose1. Notice the default route is now through SanJose2.

```
SanJose1# trace 10.0.0.1
Type escape sequence to abort.
Tracing the route to 10.0.0.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.1.2 8 msec 8 msec 8 msec
  2 192.168.1.1 [AS 200] 12 msec *  12 msec
SanJose1#
```

Practical No.: 05

Secure the Management Plane

Topology:-



Objectives:-

a. Secure management access.
b. Configure enhanced username password security.
c. Enable AAA RADIUS authentication.
d. Enable secure remote management.

Required Resources

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

Steps:-

Step 1: Configure loopbacks and assign addresses.

Step 2: Configure static routes

Step 3: Secure management access.

Step 4: Configure enhanced username password security.

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

Step 6: Enabling secure remote management using SSH.

Step 1: Configure loopbacks and assign addresses.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations. Using the addressing scheme in the diagram, apply the IP addresses to the interfaces on the R1, R2, and R3 routers.

R1

```
hostname R1

interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0
exit
!
interface Serial0/0/0
 description R1 --> R2
 ip address 10.1.1.1 255.255.255.252
 clock rate 128000
 no shutdown
exit
!
end
```

R2

```
hostname R2
!
interface Serial0/0/0
 description R2 --> R1
 ip address 10.1.1.2 255.255.255.252
 no shutdown
exit
interface Serial0/0/1
 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 Serial0/0/1
 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 }

R1# **tclsh**

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

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

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.

Note: To avoid repetitive logins during this lab, the **exec-timeout** command can be set to 0 0, which prevents it from expiring. However, this is not considered a good security practice.

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. Enter privileged EXEC mode and issue the **show run** command. Can you read the enable secret password? Why or why not?

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

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

h. Issue the **show run** command. Can you read the console, aux, and vty passwords? Why or why not?

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

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

j. Issue the **show run** command. What does the $ convert to in the output?

k. Exit privileged EXEC mode using the **disable** or **exit** command and press **Enter** to get started. Does the MOTD banner look like what you created with the **banner motd** command? If the MOTD banner is not as you wanted it, recreate it using the **banner motd** command.

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

Step 4: Configure enhanced username password security.

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

a. To create 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**

Note: An older method for creating local database entries is to use the **username** *name* **password** *password* command.

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)#

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

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

R1# **telnet 10.2.2.2**
Trying 10.2.2.2 ... Open
Unauthorized access strictly prohibited!
User Access Verification

Username: **ADMIN**
Password:
R3>

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

Authentication, authorization, and accounting (AAA) is a standards-based framework that can be implemented to control who is permitted to access a network (authenticate), what they can do on that network (authorize), and audit what they did while accessing the network (accounting).

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**

b.]Configure the specifics for the first RADIUS server located at 192.168.1.101. Use **RADIUS-1-pa55w0rd** as the server password.

R1(config)# **radius server RADIUS-1**
R1(config-radius-server)# **address ipv4 192.168.1.101**
R1(config-radius-server)# **key RADIUS-1-pa55w0rd**
R1(config-radius-server)# **exit**
R1(config)#

c. Configure the specifics for the second RADIUS server located at 192.168.1.102. Use **RADIUS-2-pa55w0rd** as the server password.

R1(config)# **radius server RADIUS-2**
R1(config-radius-server)# **address ipv4 192.168.1.102**
R1(config-radius-server)# **key RADIUS-2-pa55w0rd**
R1(config-radius-server)# **exit**
R1(config)#

d. Assign both RADIUS servers to a server group.

R1(config)# **aaa group server radius RADIUS-GROUP**
R1(config-sg-radius)# **server name RADIUS-1**
R1(config-sg-radius)# **server name RADIUS-2**
R1(config-sg-radius)# **exit**
R1(config)#

e. Enable the default AAA authentication login to attempt to validate against the server group. If they are not available, then authentication should be validated against the local database..

R1(config)# **aaa authentication login default group RADIUS-GROUP local**
R1(config)#

Note: Once this command is configured, all line access methods default to the default authentication method. The **local** option enables AAA to refer to the local database. Only the password is case sensitive.

f. Enable the default AAA authentication Telnet login to attempt to validate against the server group. If they are not available, then authentication should be validated against a case sensitive local database.

R1(config)# **aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case**
R1(config)#

Note: Unlike the **local** option that makes the password is case sensitive, local-case makes the username and password case sensitive.

g. Alter the VTY lines to use the TELNET-LOGIN AAA authentiaito0n method.

R1(config)# **line vty 0 4**
R1(config-line)# **login authentication TELNET-LOGIN**
R1(config-line)# **exit**
R1(config)#

h. Repeat the steps 5a to 5g on R3.

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

R1# **telnet 10.2.2.2**
Trying 10.2.2.2 ... Open
Unauthorized access strictly prohibited!

User Access Verification

Username: **admin**
Password:

% Authentication failed

Username: **ADMIN**
Password:

R3

Step 6: Enabling secure remote management using SSH.

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

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

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

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

R1(config)# **crypto key zeroize rsa**

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

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

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

R1(config)#
Jan 10 13:44:44.711: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#

d. Cisco routers support two versions of SSH:

- **SSH version 1 (SSHv1)**: Original version but has known vulnerabilities.
- **SSH version 2 (SSHv2)**: Provides better security using the Diffie-Hellman key exchange and the strong integrity-checking message authentication code (MAC).

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

Configure SSH version 2 on R1.

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

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

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

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

R1# **show ip ssh**
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQC3Lehh7ReYlgyDzls6wq+mFzxqzoaZFr9XGx+Q/yio
dFYw00hQo80tZy1W1Ff3Pz6q7Qi0y00urwddHZ0kBZceZK9EzJ6wZ+9a87KKDETCWrGSLi6c8lE/y4
K+
Z/oVrMMZk7bpTM1MFdP41YgkTf35utYv+TcqbsYo++KJiYk+xw==
R1#

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

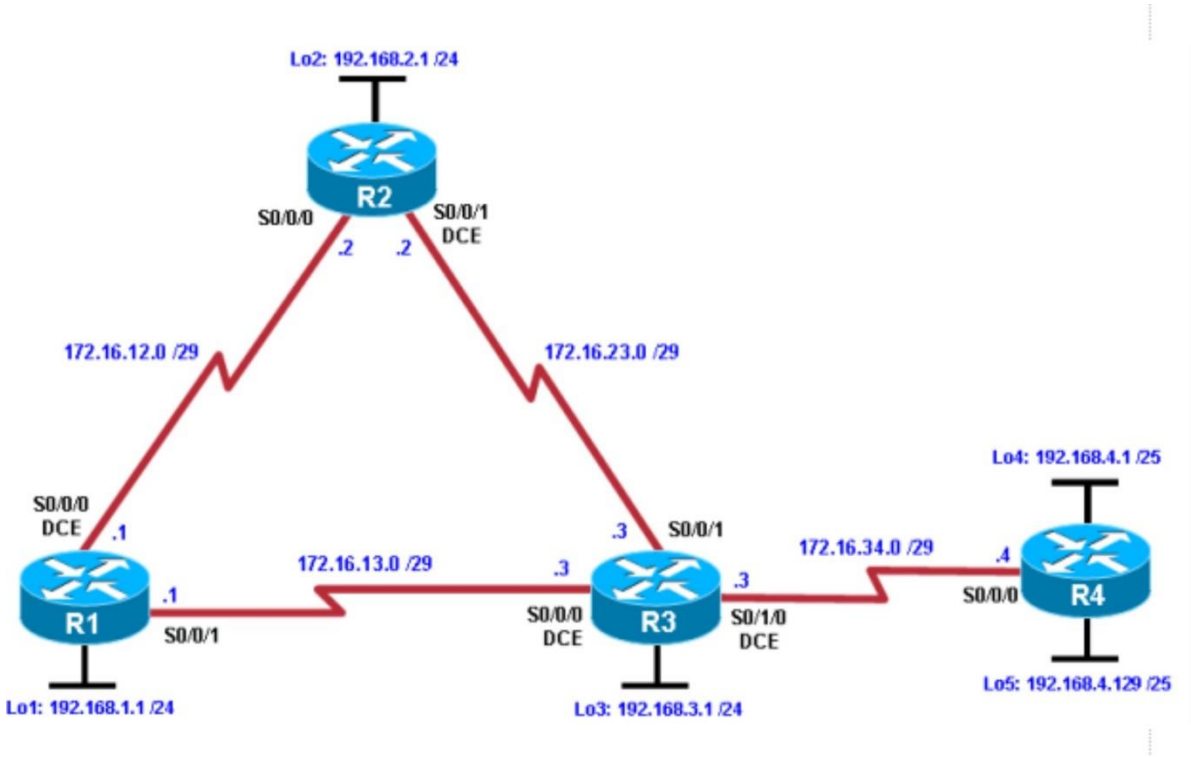
h. Although a user can SSH from a host using the SSH option of TeraTerm of PuTTY, a router can also SSH to another SSH enabled device. SSH to R3 from R1.

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

Practical No.: 06

Configure and Verify Path Control Using PBR

Topology:-



Objectives:-

- Configure and verify policy-based routing.
- Select the required tools and commands to configure policy-based routing operations.
- Verify the configuration and operation by using the proper show and debug commands.

Required Resources

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

Steps:-

Step 1: Configure loopbacks and assign addresses.

Step 2: Configure basic EIGRP.

Step 3: Verify EIGRP connectivity.

Step 4: Verify the current path.

Step 5: Configure PBR to provide path control.

Step 6: Test the policy.

Step 1: Configure loopbacks and assign addresses.

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

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

Router R1

```
hostname R1
!
interface Lo1
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/0
 description R1 --> R2
 ip address 172.16.12.1 255.255.255.248
 clock rate 128000
 bandwidth 128
 no shutdown
!
interface Serial0/0/1
 description R1 --> R3
 ip address 172.16.13.1 255.255.255.248
 bandwidth 64
 no shutdown
!
end
```

Router R2

```
hostname R2
!
interface Lo2
 description R2 LAN
 ip address 192.168.2.1 255.255.255.0
!
interface Serial0/0/0
 description R2 --> R1
 ip address 172.16.12.2 255.255.255.248
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description R2 --> R3
 ip address 172.16.23.2 255.255.255.248
 clock rate 128000
 bandwidth 128
 no shutdown
!
end
```

Router R3

```
hostname R3
!
interface Lo3
 description R3 LAN
 ip address 192.168.3.1 255.255.255.0
!
interface Serial0/0/0
 description R3 --> R1
 ip address 172.16.13.3 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 description R3 --> R2
 ip address 172.16.23.3 255.255.255.248
 bandwidth 128
 no shutdown
!
interface Serial0/1/0
 description R3 --> R4
 ip address 172.16.34.3 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
end
```

Router R4

```
hostname R4
!
interface Lo4
 description R4 LAN A
 ip address 192.168.4.1 255.255.255.128
!
interface Lo5
 description R4 LAN B
 ip address 192.168.4.129 255.255.255.128
!
interface Serial0/0/0
 description R4 --> R3
 ip address 172.16.34.4 255.255.255.248
 bandwidth 64
 no shutdown
!
end
```

c. 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
Serial0/0/0            172.16.13.3     YES manual up                    up
Serial0/0/1            172.16.23.3     YES manual up                    up
Serial0/1/0            172.16.34.3     YES manual up                    up
Loopback3              192.168.3.1     YES manual up                    up
R3#
R3# show protocols
Global values:
  Internet Protocol routing is enabled
Embedded-Service-Engine0/0 is administratively down, line protocol is down
GigabitEthernet0/0 is administratively down, line protocol is down
GigabitEthernet0/1 is administratively down, line protocol is down
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.13.3/29
Serial0/0/1 is up, line protocol is up
Internet address is 172.16.23.3/29
Serial0/1/0 is up, line protocol is up
Internet address is 172.16.34.3/29
Serial0/1/1 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
Se0/0/0                        up             up       R3 --> R1
Se0/0/1                        up             up       R3 --> R2
Se0/1/0                        up             up       R3 --> R4
Lo3                            up             up       R3 LAN
R3#
```

Step 2: Configure basic EIGRP.

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

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

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

Router R1

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

Router R2

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

Router R3

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

Router R4

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

You should see EIGRP neighbor relationship messages being generated.

Step 3: Verify EIGRP connectivity.

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

```
R1# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address          Interface      Hold Uptime  SRTT   RTO  Q  Seq
                                    (sec)        (ms)       Cnt Num
1   172.16.13.3      Se0/0/1         10 00:01:55  27  2340  0  9
0   172.16.12.2      Se0/0/0         13 00:02:07   8  1170  0  11
R1#
```

```
R2# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H  Address          Interface      Hold Uptime  SRTT   RTO  Q  Seq
                                    (sec)        (ms)       Cnt Num
1  172.16.23.3      Se0/0/1         12 00:02:15  12  1170  0  10
0  172.16.12.1      Se0/0/0         11 00:02:27   9  1170  0  13
```

```
R2#

R3# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address          Interface      Hold Uptime  SRTT   RTO  Q  Seq
                                    (sec)        (ms)       Cnt Num
2   172.16.34.4      Se0/1/0         12 00:02:14  44  2340  0  3
1   172.16.23.2      Se0/0/1         11 00:02:23  10  1170  0  10
0   172.16.13.1      Se0/0/0         10 00:02:23 1031  5000  0  12
R3#
R4# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H  Address          Interface      Hold Uptime  SRTT   RTO  Q  Seq
                                    (sec)        (ms)       Cnt Num
0  172.16.34.3      Se0/0/0         10 00:02:22  37  2340  0  11
R4#
```

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

```
R1# tclsh

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

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

Step 4: Verify the current path.

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

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

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

     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
C    172.16.12.0/29 is directly connected, Serial0/0/0
L       172.16.12.1/32 is directly connected, Serial0/0/0
C       172.16.13.0/29 is directly connected, Serial0/0/1
L       172.16.13.1/32 is directly connected, Serial0/0/1
D    172.16.23.0/29 [90/21024000] via 172.16.12.2, 00:07:22, Serial0/0/0D
172.16.34.0/29 [90/41024000] via 172.16.13.3, 00:07:22, Serial0/0/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
```

```
C       192.168.1.0/24 is directly connected, Loopback1
L       192.168.1.1/32 is directly connected, Loopback1
D    192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:07:22, Serial0/0/0
D    192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:07:22, Serial0/0/0
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:07:14, Serial0/0/1
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:07:14, Serial0/0/1
R1#
```

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

Note: You can specify the source as the interface address (for example 192.168.4.1) or the interface designator (for example, Fa0/0).

```
R4# traceroute 192.168.1.1 source 192.168.4.1
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 12 msec 12 msec 16 msec
  2 172.16.23.2 20 msec 20 msec 20 msec
  3 172.16.12.1 24 msec * 24 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 12 msec 16 msec 12 msec
  2 172.16.23.2 28 msec 20 msec 16 msec
  3 172.16.12.1 24 msec * 24 msec
R4#
```

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

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

     172.16.0.0/16 is variably subnetted, 7 subnets, 2 masks
D       172.16.12.0/29 [90/21024000] via 172.16.23.2, 00:10:54, Serial0/0/1
C       172.16.13.0/29 is directly connected, Serial0/0/0
L       172.16.13.3/32 is directly connected, Serial0/0/0
C       172.16.23.0/29 is directly connected, Serial0/0/1
L       172.16.23.3/32 is directly connected, Serial0/0/1
C       172.16.34.0/29 is directly connected, Serial0/1/0
L  172.16.34.3/32 is directly connected, Serial0/1/0
D   192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:10:54, Serial0/0/1D
192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:10:54, Serial0/0/1
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.3.0/24 is directly connected, Loopback3
L       192.168.3.1/32 is directly connected, Loopback3
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/40640000] via 172.16.34.4, 00:10:47, Serial0/1/0
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:10:47, Serial0/1/0
R3#
```

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

```
R3# show interfaces serial0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is WIC MBRD 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)
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     399 packets input, 29561 bytes, 0 no buffer
     Received 186 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     393 packets output, 29567 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 unknown protocol drops
     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 serial0/0/0 | include BW
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
R3# show interfaces serial0/0/1 | include BW
  MTU 1500 bytes, BW 128 Kbit/sec, DLY 20000 usec,
R3#
```

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

```
R3# show ip eigrp topology 192.168.1.0
EIGRP-IPv4 Topology Entry for AS(1)/ID(192.168.3.1) for 192.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000
  Descriptor Blocks:
  172.16.23.2 (Serial0/0/1), 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
        Originating router is 192.168.1.1
  172.16.13.1 (Serial0/0/0), 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
```

Step 5: Configure PBR to provide path control.

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

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

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

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

c. Apply the R3-to-R1 route map to the serial interface on R3 that receives the traffic from R4. Use the **ip policy route-map** command on interface S0/1/0.

```
R3(config)# interface s0/1/0
R3(config-if)# ip policy route-map R3-to-R1
R3(config-if)# end
R3#
```

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

```
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.

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

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

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

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

R3# debug ip policy 1
```

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

R4# **traceroute 192.168.1.1 source 192.168.4.1**

Type escape sequence to abort.
Tracing the route to 192.168.1.1

1 172.16.34.3 0 msec 0 msec 4 msec
2 172.16.23.2 0 msec 0 msec 4 msec
3 172.16.12.1 4 msec 0 msec *

Notice the path taken for the packet sourced from R4 LAN A is still going through R3 --> R2 --> R1.

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

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

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

R4# **traceroute 192.168.1.1 source 192.168.4.129**

Type escape sequence to abort.
Tracing the route to 192.168.1.1

1 172.16.34.3 12 msec 12 msec 16 msec
2 172.16.13.1 28 msec 28 msec *

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

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

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

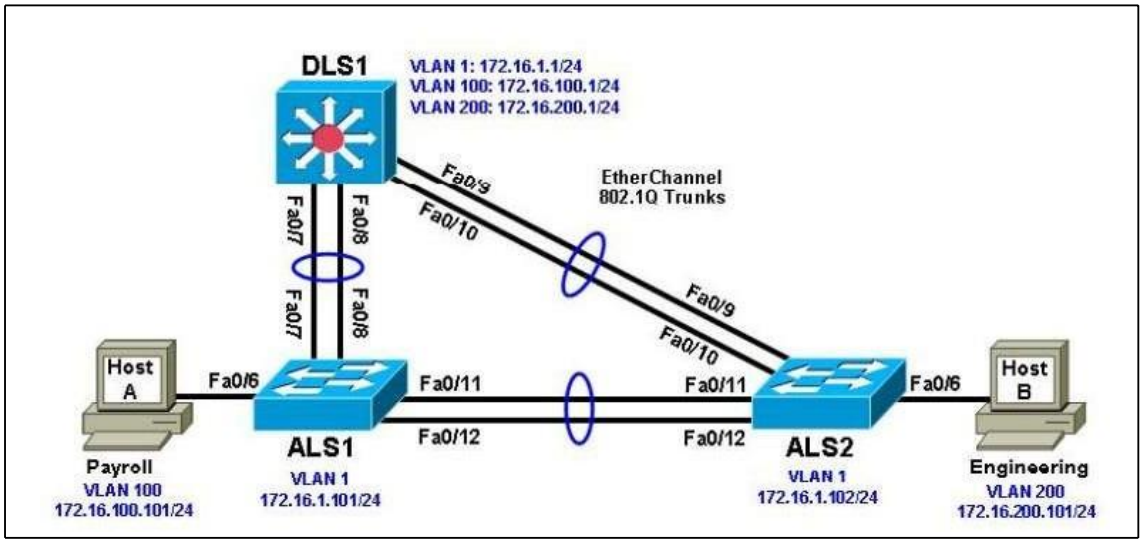
e. On R3, display the policy and matches using the **show route-map** command.

```
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
```

Practical No.: 07

IP Service Level Agreements and Remote SPAN in a CampusEnvironment.

Topology:-



Objectives:-

- Configure trunking, VTP, and SVIs.
- Implement IP SLAs to monitor various network performance characteristics.

Required Resources

- 2 switches (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- 1 switch (Cisco 3560 with the Cisco IOS Release 12.2(46)SE C3560-ADVIPSERVICESK9-mz image or comparable)
- Ethernet and console cables

Steps:-

Step 1: Prepare the switches for the lab
Step 2: Configure host PCs.
Step 3: Configure basic switch parameters
Step 4: Configure trunks and EtherChannels between switches.
Step 5: Configure VTP on ALS1 and ALS2.
Step 6: Configure VTP on DLS1.
Step 7: Configure access ports.
Step 8: Configure VLAN interfaces and enable routing.

Step 9: Configure Cisco IOS IP SLA responders.
Step 10: Configure the Cisco IOS IP SLA source to measure network performance
Step 11 : Monitor IP SLAs operations.

Step 1: Prepare the switches for the lab.

Erase the startup configuration, delete the vlan.dat file, and reload the switches. Refer to Lab 1 -1 "Clearing a Switch" and Lab 1 -2 "Clearing a Switch Connected to a Larger Network" to prepare the switches for this lab. Cable the equipment as shown.

Step 2: Configure host PCs.

Configure PCs Host A and Host B with the IP address and subnet mask shown in the topology. Host A is in VLAN 100 with a default gateway of 172.16.100.1 . Host B is in VLAN 200 with a default gateway of 172.16.200.1 .

Step 3: Configure basic switch parameters.

Configure the hostname, password, and, optionally, remote access on each switch.

Switch(config)# **hostname ALS1**
ALS1(config)# **enable secret cisco**
ALS1(config)# **line vty 0 15**
ALS1(config-line)# **password cisco**
ALS1(config-line)# **login**
Switch(config)# **hostname ALS2**
ALS2(config)# **enable secret cisco**
ALS2(config)# **line vty 0 15**
ALS2(config-line)# **password cisco**
ALS2(config-line)# **login**
Switch(config)# **hostname DLS1**
DLS1(config)# **enable secret cisco**
DLS1(config)# **line vty 0 15**
DLS1(config-line)#**password cisco**
DLS1(config-line)# **login**

ALS1(config)# **interface vlan 1**
ALS1(config-if)# **ip address 172. 16.1.101 255.255.255.0**
ALS1(config-if)# **no shutdown**
ALS2(config)# **interface vlan 1**
ALS2(config-if)# **ip address 172.16.1.102 255.255.255.0**
ALS2(config-if)# **no shutdown**
DLS1(config)# **interface vlan 1**
DLS1(config-if)# **ip address 172.16.1.1 255.255.255.0**
DLS1(config-if)# **no shutdown**

ALS1(config)# **ip default-gateway 172.16.1.1**
ALS2(config)# **ip default-gateway 172.16.1.1**

Step 4: Configure trunks and EtherChannels between switches.

To distribute VLAN and VTP information, trunks are needed between the three switches. Configure these trunks according to the diagram. EtherChannel is used for these trunks.

Configure the trunks and EtherChannel from DLS1 to ALS1.

DLS1(config)# **interface range fastEthernet 0/7 - 8**
DLS1(config-if-range)# **switchport trunk encapsulation dot1q**
DLS1(config-if-range)# **switchport mode trunk**
DLS1(config-if-range)# **channel-group 1 mode desirable**

Creating a port-channel interface Port-channel 1

Configure the trunks and EtherChannel from DLS1 to ALS2.

DLS1(config)# **interface range fastEthernet 0/9 - 10**
DLS1(config-if-range)# **switchport trunk encapsulation dot1q**
DLS1(config-if-range)# **switchport mode trunk**
DLS1(config-if-range)# **channel-group 2 mode desirable**

Creating a port-channel interface Port-channel 2

Configure the trunks and EtherChannel between ALS1 and DLS1 and between ALS1 and ALS2.

ALS1(config)# **interface range fastEthernet 0/11 - 12**
ALS1(config-if-range)# **switchport mode trunk**
ALS1(config-if-range)# **channel-group 1 mode desirable**

Creating a port-channel interface Port-channel 1

ALS1(config-if-range)# **exit**
ALS1(config)# **interface range fastEthernet 0/7 - 8**
ALS1(config-if-range)# **switchport mode trunk**
ALS1(config-if-range)# **channel-group 2 mode desirable**
Creating a port-channel interface Port-channel 2

Configure the trunks and EtherChannel between ALS2 and DLS1 and between ALS2 and ALS1.

ALS2(config)# **interface range fastEthernet 0/11 - 12**
ALS2(config-if-range)# **switchport mode trunk**
ALS2(config-if-range)# **channel-group 1 mode desirable**

Creating a port-channel interface Port-channel 1

ALS2(config-if-range)# **exit**
ALS2(config)# **interface range fastEthernet 0/9 - 10**
ALS2(config-if-range)# **switchport mode trunk**
ALS2(config-if-range)# **channel-group 2 mode desirable**
Creating a port-channel interface Port-channel 2

Step 5: Configure VTP on ALS1 and ALS2.

Change the VTP mode of ALS1 and ALS2 to client.

ALS1(config)# **vtp mode client**
Setting device to VTP CLIENT mode.
ALS2(config)# **vtp mode client**
Setting device to VTP CLIENT mode.

Step 6: Configure VTP on DLS1.

Create the VTP domain on DLS1, and create VLANs 100 and 200 for the domain.

DLS1(config)# **vtp domain SWPOD**
DLS1(config)# **vtp version 2**
DLS1(config)# **vlan 100**
DLS1(config-vlan)# **name Finance**
DLS1(config-vlan)# **vlan 200**
DLS1(config-vlan)# **name Engineering**

Step 7: Configure access ports.

Configure the host ports for the appropriate VLANs according to the diagram.

ALS1(config)# **interface fastEthernet 0/6**
ALS1(config-if)# **switchport mode access**
ALS1(config-if)# **switchport access vlan 100**
ALS2(config)# **interface fastEthernet 0/6**
ALS2(config-if)# **switchport mode access**
ALS2(config-if)# **switchport access vlan 200**

Step 8: Configure VLAN interfaces and enable routing.

On DLS1, create the SVIs for VLANs 100 and 200. Note that the corresponding Layer 2 VLANs must be
configured for the Layer 3 SVIs to activate. This was done in Step 6.

DLS1(config)# **interface vlan 100**
DLS1(config-if)# **ip address 172.16.100.1 255.255.255.0**
DLS1(config-if)# **interface vlan 200**
DLS1(config-if)# **ip address 172.16.200.1 255.255.255.0**
DLS1(config)# **ip routing**

Verify the configuration using the **show ip route** command on DLS1.

DLS1# **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, E - EGP
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static
route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
172.16.0.0/24 is subnetted, 3 subnets
C 172.16.200.0 is directly connected, Vlan200
C 172.16.1.0 is directly connected, Vlan1
C 172.16.100.0 is directly connected, Vlan100

Run the following Tcl script on DLS1 to verify full connectivity. If these pings are not successful, troubleshoot.

Tcl is only supported on DLS1.

DLS1# **tclsh**
foreach address {
172.16. 1.1
172.16. 1.101
172.16. 1.102
172.16. 100.1
172.16.200.1
172.16. 100.101
172.16.200.101
} {
ping $address }

Step 9: Configure Cisco IOS IP SLA responders.

IP SLA responders are Cisco IOS devices that support the IP SLA control protocol. An IP SLA responder uses the Cisco IOS IP SLA Control Protocol for notification configuration and on which port to listen and respond. Some operations

ALS1(config)# **ip sla responder**
ALS2(config)# **ip sla responder**
Configure ALS1 and ALS2 as IP SLA responders for UDP jitter using the **ip sla responder udp- echo ipaddress** command. Specify the IP address of DLS1 VLAN 1 to act as the destination IP address for the reflected UDP traffic on both ALS1 and ALS2.

ALS1(config)# **ip sla responder udp-echo ipaddress 172.16.1.1 port 5000**
ALS2(config)# **ip sla responder udp-echo ipaddress 172.16.1.1 port 5000**

Step 10: Configure the Cisco IOS IP SLA source to measure network performance.

IP SLA uses generated traffic to measure network performance between two networking devices. On DLS1, create an IP SLA operation and enter IP SLA configuration mode with the **ip sla** operationnumber command.

DLS1(config)# **ip sla 1**
DLS1(config-ip-sla)#

DLS1(config-ip-sla)# **icmp-echo 172.16.100.101**
DLS1(config-ip-sla-echo)# **exit**
DLS1(config)# **ip sla 2**
DLS1(config-ip-sla)# **icmp-echo 172.16.200.101**
DLS1(config-ip-sla-echo)# **exit**

DLS1(config)# **ip sla 3**
DLS1(config-ip-sla)# **udp-jitter 172.16.1.101 5000**
DLS1(config-ip-sla-j itter)# **exit**
DLS1(config)# **ip sla 4**
DLS1(config-ip-sla)# **udp-jitter 172.16.1.102 5000**
DLS1(config-ip-sla-jitter)# **exit**

DLS1(config)# **ip sla schedule 1 life forever start-time now**
DLS1(config)# **ip sla schedule 2 life forever start-time now**
DLS1(config)# **ip sla schedule 3 life forever start-time now**
DLS1(config)# **ip sla schedule 4 life forever start-time now**

Step 11 : Monitor IP SLAs operations.

View the IP SLA configuration for IP SLA 1 on DLS1 . The output for IP SLA 2 is similar.

```
DLS1# show ip sla configuration 1
IP SLAs, Infrastructure Engine-II.
Entry number: 1
Owner:
Tag:
Type of operation to perform: echo
Target address/Source address: 172.16.100.101/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
Operation frequency (seconds): 60Next Scheduled Start Time: Start Time
already passed
Group Scheduled : FALSE
```

Randomly Scheduled : FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
History Statistics:
Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: None
Enhanced History:

View the IP SLA configuration for IP SLA 3 on DLS1 . The output for IP SLA 4 is similar.

DLS1# **show ip sla configuration 3**
IP SLAs, Infrastructure Engine-II.
Entry number: 3
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 172.16.1.101/0.0.0.0
Target port/Source port: 5000/0
Type Of Service parameter: 0x0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
Operation frequency (seconds): 60
Next Scheduled Start Time: Start Time already passed
Group Scheduled : FALSE
Randomly Scheduled : FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
Enhanced History:

Display global information about Cisco IOS IP SLAs on DLS1 .

DLS1# **show ip sla application**
Version: 2.2.0 Round Trip Time MIB, Infrastructure Engine-II
Time of last change in whole IP SLAs: *13:16:30.493 UTC Fri Mar 5 2010
Estimated system max number of entries: 11928
Estimated number of configurable operations: 11924
Number of Entries configured : 4
Number of active Entries : 4
Number of pending Entries : 0
Number of inactive Entries : 0
Type of Operation to Perform: dhcp
Type of Operation to Perform: dns
Type of Operation to Perform: echo
Type of Operation to Perform: ftp
Type of Operation to Perform: http
Type of Operation to Perform: jitter
Type of Operation to Perform: pathEcho
Type of Operation to Perform: pathJitter
Type of Operation to Perform: tcpConnect
Type of Operation to Perform: udpEcho
IP SLAs low memory water mark: 16273927

Display information about Cisco IOS IP SLA responders on ALS1 . The ALS2 output is similar.

ALS1# **show ip sla responder**
IP SLAs Responder is: Enabled
Number of control message received: 38 Number of errors: 0
Recent sources:
Recent error sources:
udpEcho Responder:
IPv6/IP Address Port
172.16.1.1 5000

Display IP SLA statistics on DLS1 for IP SLA 1 . The IP SLA 2 output is similar.

DLS1# **show ip sla statistics 1**
Round Trip Time (RTT) for Index 1
Latest RTT: 1 ms
Latest operation start time: *13:17:21.231 UTC Fri Mar 5 2010
Latest operation return code: OK
Number of successes: 15
Number of failures: 1
Operation time to live: Forever

From this output, you can see that the latest round-trip time (RTT) for SLA operation Index 1 (icmp-echo) is 1 millisecond (ms). The number of packets sent successfully from DLS1 to PC Host A was 15, and there was one failure. Display IP SLA statistics on DLS1 for IP SLA 3. The IP SLA 4 output is similar.

DLS1# show ip sla statistics 3
Round Trip Time (RTT) for Index 3
Latest RTT: 3 ms
Latest operation start time: *13:19:45.322 UTC Fri Mar 5 2010
Latest operation return code: OK
RTT Values
Number Of RTT: 10
RTT Min/Avg/Max: 2/3/5 m

Latency one-way time milliseconds
Number of Latency one-way Samples: 0
Source to Destination Latency one way Min/Avg/Max: 0/0/0 ms
Destination to Source Latency one way Min/Avg/Max: 0/0/0 ms
Jitter time milliseconds
Number of SD Jitter Samples: 9
Number of DS Jitter Samples: 9
Source to Destination Jitter Min/Avg/Max: 0/1/2 ms
Destination to Source Jitter Min/Avg/Max: 0/1/1 ms
Packet Loss Values
Loss Source to Destination: 0 Loss Destination to Source: 0
Out Of Sequence: 0 Tail Drop: 0 Packet Late Arrival: 0
Voice Score Values
Calculated Planning Impairment Factor (ICPIF): 0
Mean Opinion Score (MOS): 0
Number of successes: 14
Number of failures: 0
Operation time to live: Forever

From this output, you can see that the latest RTT for SLA operation Index 3 (udp-jitter) is 3 ms. Jitter time from source to destination and from destination to source is averaging 1 ms, which is acceptable for voice applications. The number of packets sent successfully from DLS1 to ALS1 was 14, and there were no failures. Disable interface VLAN 1 on ALS1 using the **shutdown** command.

ALS1(config)# **interface vlan 1**
ALS1(config-if)# **shutdown**

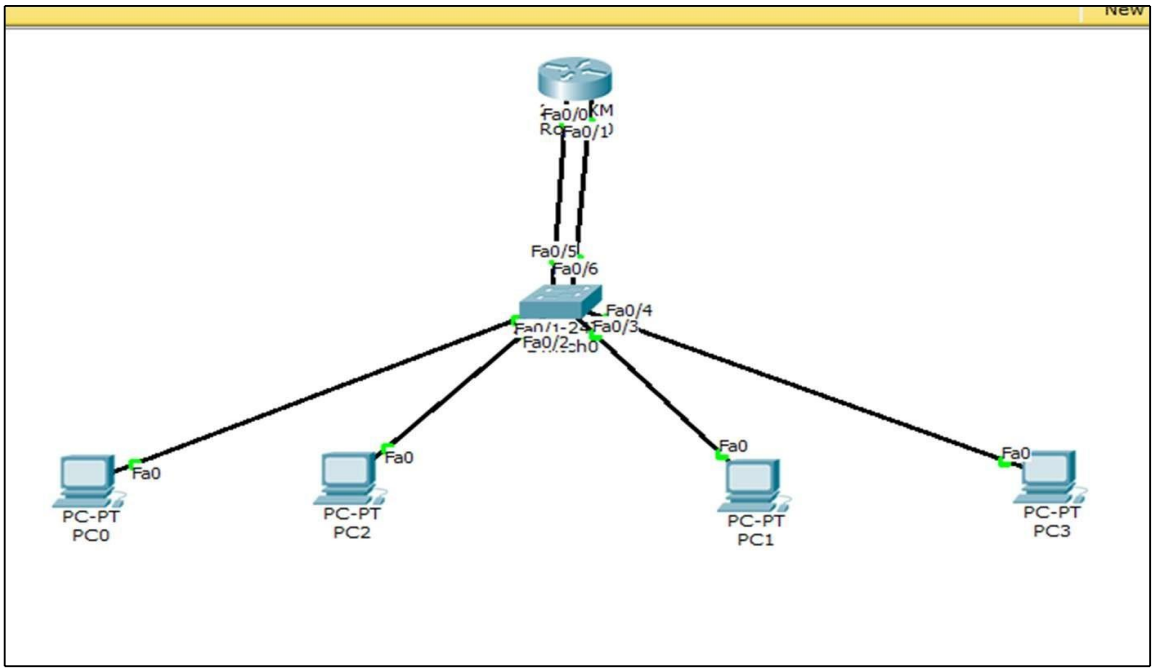
Allow a few minutes to pass and then issue the **show ip sla statistics 3** command on DLS1 . The output should look similar to the following.

DLS1# **show ip sla statistics 3**
Round Trip Time (RTT) for Index 3
Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *13:19:45.322 UTC Fri Mar 5 2010
Latest operation return code: Timeout
RTT Values
Number Of RTT: 0
RTT Min/Avg/Max: 0/0/0 ms
Latency one-way time milliseconds
Number of Latency one-way Samples: 0
Source to Destination Latency one way Min/Avg/Max: 0/0/0 ms
Destination to Source Latency one way Min/Avg/Max: 0/0/0 ms
Jitter time milliseconds
Number of SD Jitter Samples: 0
Number of DS Jitter Samples: 0
Source to Destination Jitter Min/Avg/Max: 0/0/0 ms
Destination to Source Jitter Min/Avg/Max: 0/0/0 ms
Packet Loss Values
Loss Source to Destination: 0 Loss Destination to Source: 0
Out Of Sequence: 0 Tail Drop: 0 Packet Late Arrival: 0
Voice Score Values
Calculated Planning Impairment Factor (ICPIF): 0
Mean Opinion Score (MOS): 0
Number of successes: 14
Number of failures: 2
Operation time to live: Forever

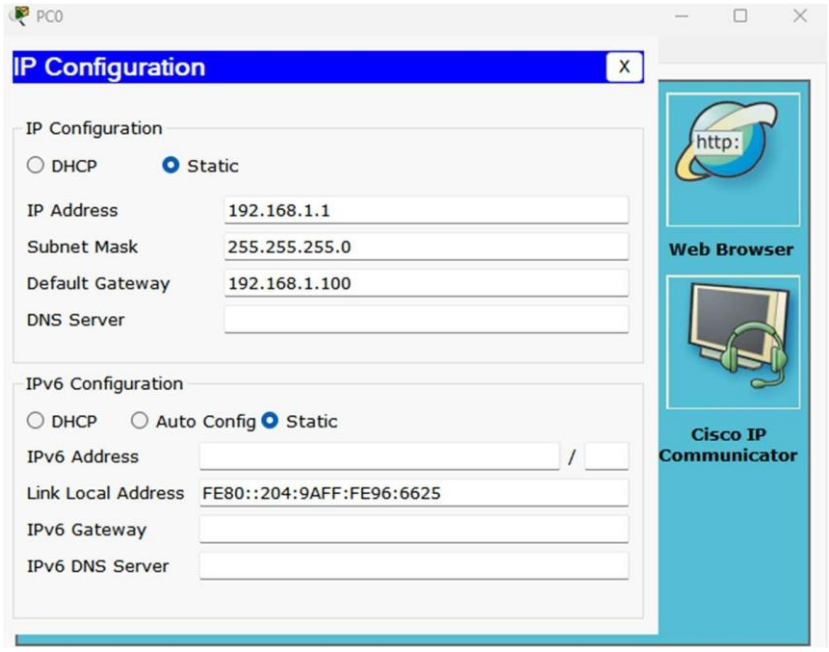
Practical No.: 08

Inter- Vlan Routing.

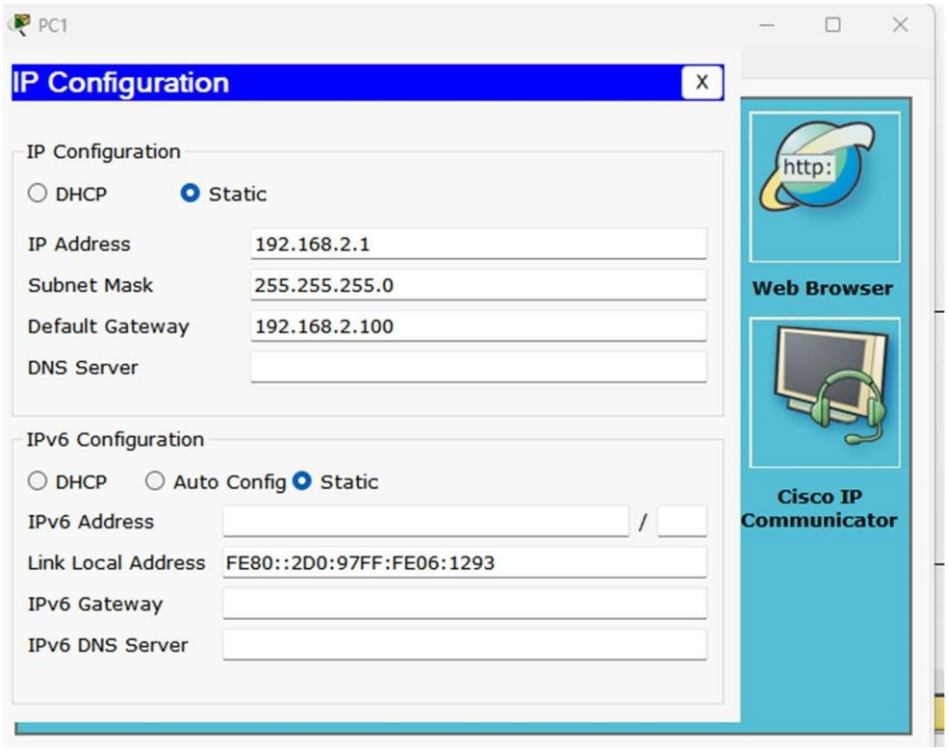
Design:



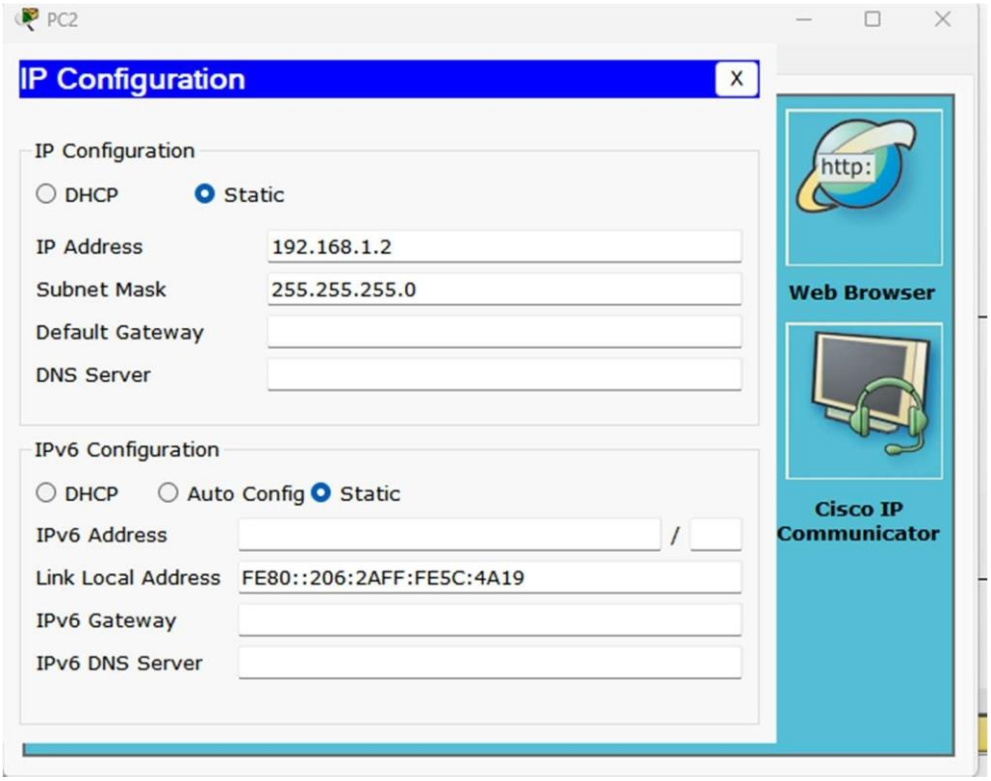
PC0:



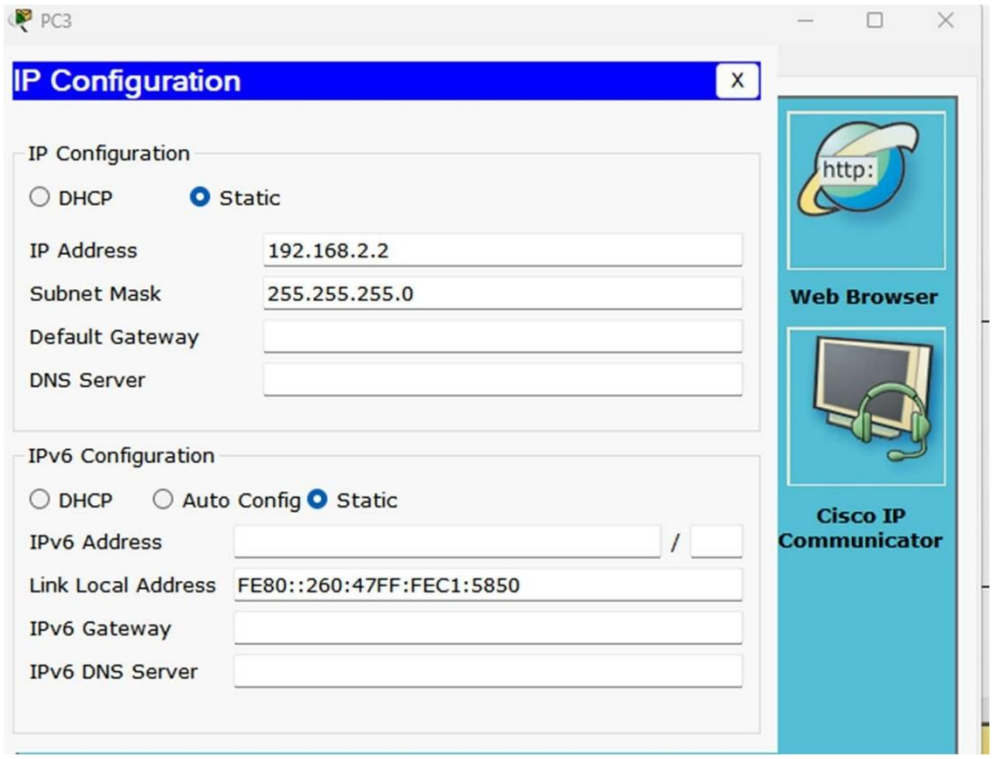
PC1:


PC1
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.1
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.100
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2D0:97FF:FE06:1293
IPv6 Gateway
IPv6 DNS Server
http:
Web Browser
Cisco IP Communicator

PC2:


PC2
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.1.2
Subnet Mask
255.255.255.0
Default Gateway
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::206:2AFF:FE5C:4A19
IPv6 Gateway
IPv6 DNS Server
http:
Web Browser
Cisco IP Communicator

PC3:



CMD PC1:

```
PC>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.1: bytes=32 time=0ms TTL=127
Reply from 192.168.2.1: bytes=32 time=0ms TTL=127
Reply from 192.168.2.1: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

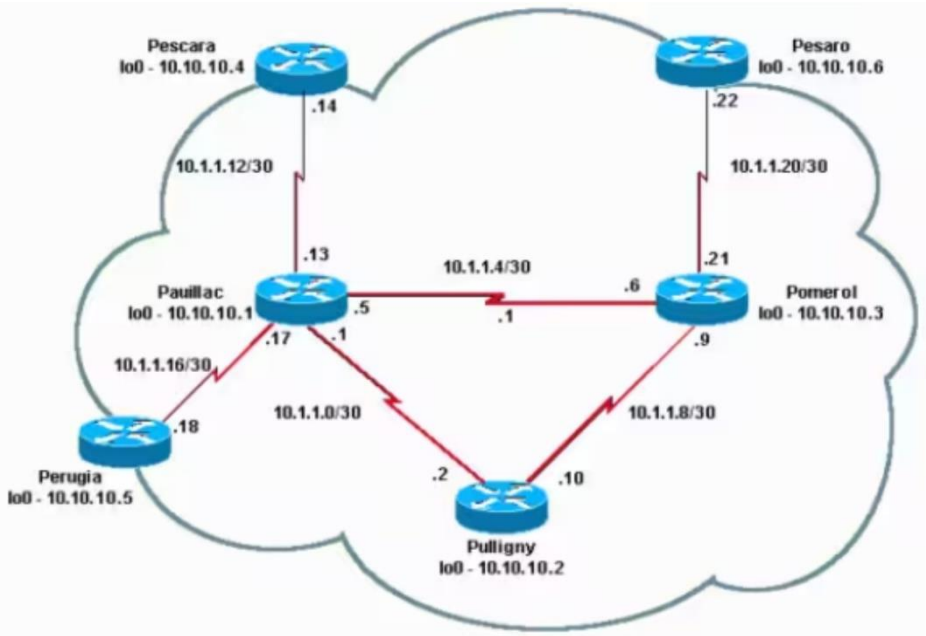
CMD PC0:

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=6ms TTL=128
Reply from 192.168.1.1: bytes=32 time=6ms TTL=128
Reply from 192.168.1.1: bytes=32 time=8ms TTL=128
Reply from 192.168.1.1: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 6ms

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=35ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 35ms, Average = 9ms

PC>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=24ms TTL=255
Reply from 192.168.1.100: bytes=32 time=0ms TTL=255
Reply from 192.168.1.100: bytes=32 time=0ms TTL=255
Reply from 192.168.1.100: bytes=32 time=0ms TTL=255

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

Practical No.: 09

Simulating MPLS environmental

Topology:-



Object82

Required Resources

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

Steps:-

Step 1:- Pomerol

Step 2:- Pulligny

Step 3:- Pauillac

Step 4:- Pescara

Step 5:- Pesaro

Step 6:- Perugia

Step 1:- Pomerol

```
version 12.2
hostname Pomerol
ip subnet-zero
ip cef

interface Loopback0
 ip address 10.10.10.3 255.255.255.255

interface Serial4/0
 ip address 10.1.1.21 255.255.255.252
 tag-switching ip

interface Serial4/2
 ip address 10.1.1.6 255.255.255.252
 tag-switching ip

interface Serial4/4
 ip address 10.1.1.9 255.255.255.252
 tag-switching ip

router ospf 10
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 9

ip classless

end
```

Step 2:- Pulligny

```
version 12.2
hostname Pulligny
ip subnet-zero
ip cef

interface Loopback0
 ip address 10.10.10.2 255.255.255.255

interface Serial4/0
 ip address 10.1.1.2 255.255.255.252
 tag-switching ip

interface Serial4/1
 ip address 10.1.1.10 255.255.255.252
 tag-switching ip

router ospf 10
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 9

ip classless

end
```

Step 3:- Pauillac

```
version 12.2
hostname Pauillac
ip subnet-zero
ip cef

interface Loopback0
 ip address 10.10.10.1 255.255.255.255

interface Serial4/1
 ip address 10.1.1.13 255.255.255.252
 tag-switching ip

interface Serial4/2
 ip address 10.1.1.17 255.255.255.252
 tag-switching ip

interface Serial4/3
 ip address 10.1.1.1 255.255.255.252
 tag-switching ip

interface Serial4/5
 ip address 10.1.1.5 255.255.255.252
 tag-switching ip

router ospf 10
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 9

ip classless

end
```

Step 4:- Pescara

```
version 12.2
hostname Pescara
ip subnet-zero
ip cef

interface Loopback0
 ip address 10.10.10.4 255.255.255.255

interface Serial4/3
 ip address 10.1.1.14 255.255.255.252
 tag-switching ip

router ospf 10
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 9

ip classless

end
```

Step 5:- Pesaro

```
version 12.2
hostname Pesaro
ip subnet-zero
ip cef

interface Loopback0
 ip address 10.10.10.6 255.255.255.255

interface Serial4/4
 ip address 10.1.1.22 255.255.255.252
 tag-switching ip

router ospf 10
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 9

ip classless

end
```

Step 6:- Perugia

```
version 12.2
hostname Perugia
ip subnet-zero
ip cef

interface Loopback0
 ip address 10.10.10.5 255.255.255.255

interface Serial4/5
 ip address 10.1.1.18 255.255.255.252
 tag-switching ip

router ospf 10
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 9

ip classless

end
```

Verify

This section provides information you can use to confirm your configuration works properly.

Commands used in the Configuring Basic MPLS Using IS-IS sample configuration are also applicable.

In order to illustrate this sample configuration, look at a particular destination, for example **10.10.10.4**, on the **Pomerol** LSR.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) , which allows you to view an analysis of **show** command output.

- show ip route —Used to check the IP route for this destination in the IP routing table:

```
Pomerol#show ip route 10.10.10.4
Routing entry for 10.10.10.4/32
  Known via "ospf 10", distance 110, metric 129, type intra area
  Last update from 10.1.1.5 on Serial3/0, 17:29:23 ago
  Routing Descriptor Blocks:
  * 10.1.1.5, from 10.10.10.4, 17:29:23 ago, via Serial3/0
      Route metric is 129, traffic share count is 1
```

- show mpls forwarding-table —Used to check the MPLS forwarding table, which is the label switching equivalent of the IP routing table for standard IP routing. It contains inbound and outbound labels and descriptions of the packets.

```
Pomerol#show mpls forwarding-table
Local  Outgoing    Prefix          Bytes tag   Outgoing    Next Hop
tag    tag or VC   or Tunnel Id    switched    interface
16     Pop tag     10.1.1.12/30    636         Se3/0       point2point
17     Pop tag     10.10.10.1/32   0           Se3/0       point2point
18     21          10.10.10.4/32   0           Se3/0       point2point
19     Pop tag     10.1.1.0/30     0           Se4/0       point2point
       Pop tag     10.1.1.0/30     0           Se3/0       point2point
20     Pop tag     10.10.10.6/32   612         Se2/0       point2point
21     Pop tag     10.1.1.16/30    0           Se3/0       point2point
22     16          10.10.10.5/32   0           Se3/0       point2point
23     Pop tag     10.10.10.2/32   0           Se4/0       point2point
```

- show mpls forwarding-table detail —Used to see MPLS forwarding table details:

```
Pomerol#show mpls forwarding-table 10.10.10.4 32 detail
Local  Outgoing    Prefix          Bytes tag   Outgoing    Next Hop
tag    tag or VC   or Tunnel Id    switched    interface
18     21          10.10.10.4/32   0           Se3/0       point2point
       MAC/Encaps=4/8, MRU=1500, Tag Stack{21}
       0F008847 00015000
       No output feature configured
     Per-packet load-sharing
```

- show mpls ldp bindings or show tag-switching tdp bindings (based on which Cisco IOS software release you use) —Used to see the label bindings associated with a particular destination. Both the local as well as the remote bindings can be seen.

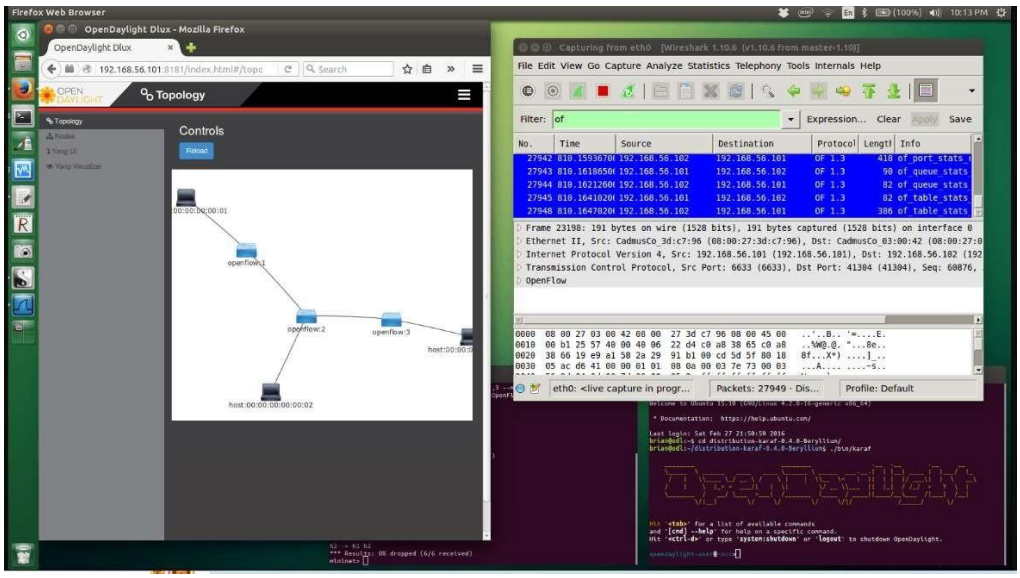
```
Pomerol#show tag-switching tdp bindings 10.10.10.4 32
  tib entry: 10.10.10.4/32, rev 14
      local binding:  tag: 18
      remote binding: tsr: 10.10.10.1:0, tag: 21
      remote binding: tsr: 10.10.10.2:0, tag: 23
      remote binding: tsr: 10.10.10.6:612, tag: 20
```

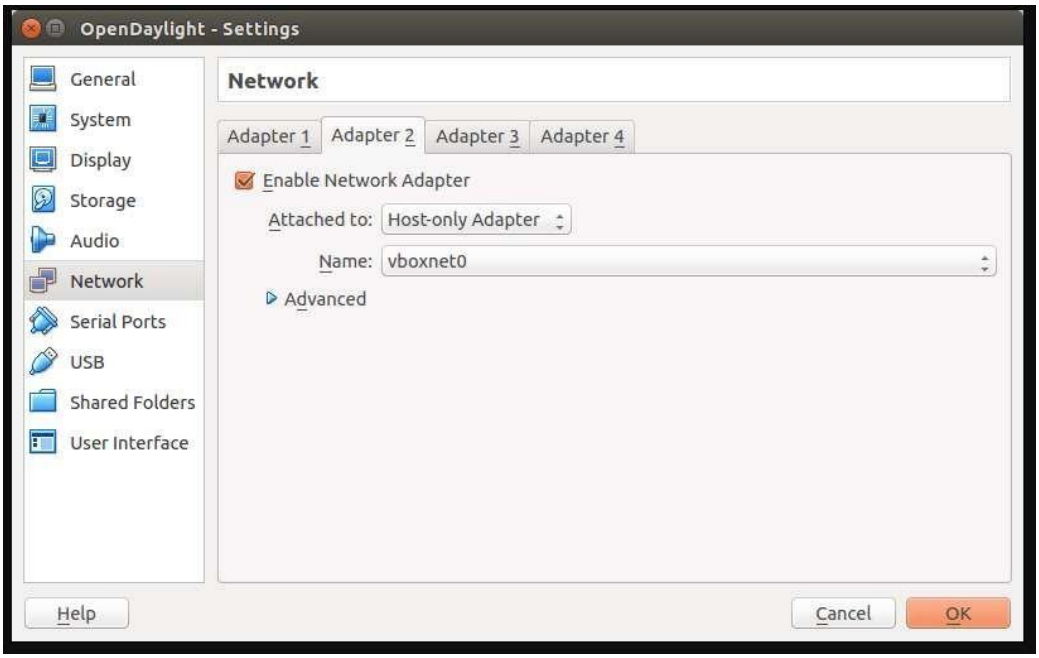
Practical No: 10

Simulating SDN

A) Open Daylight SDN Controller with the Mininet Network Emulator.

Design:





```
brian@odl:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:ec:a9:f1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feec:a9f1/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
qlen 1000
    link/ether 08:00:27:b0:f6:70 brd ff:ff:ff:ff:ff:ff
brian@odl:~$
```

```
brian@odl:~$ sudo dhclient enp0s8
```

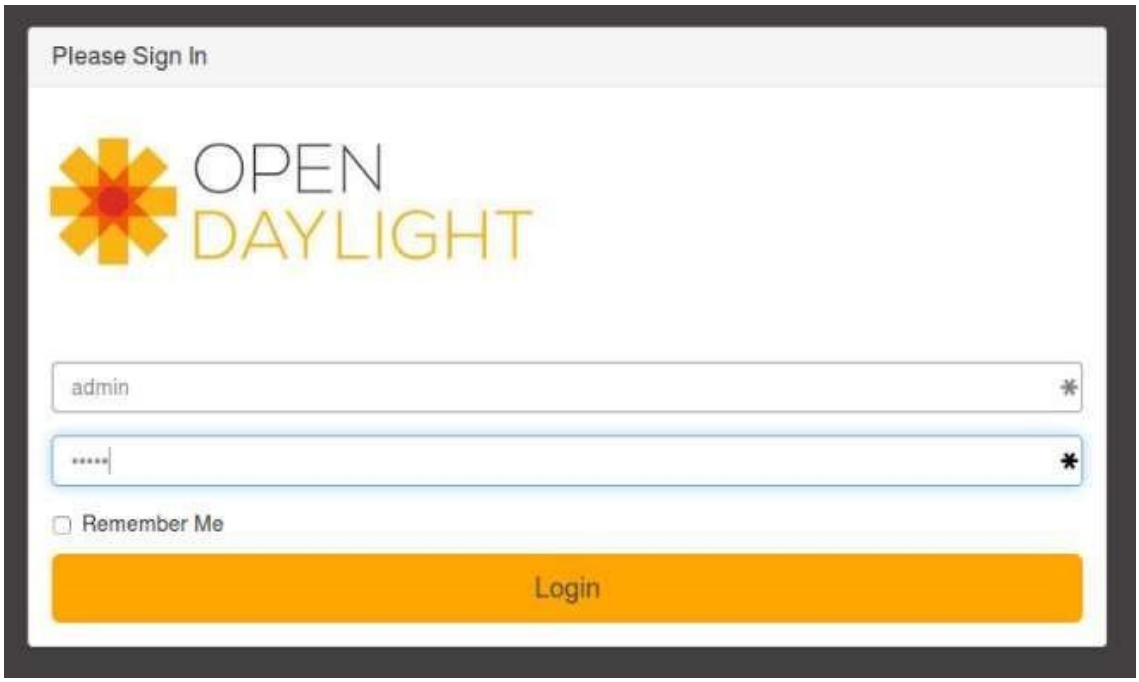
Now check the IP address assigned to *enp0s8*:

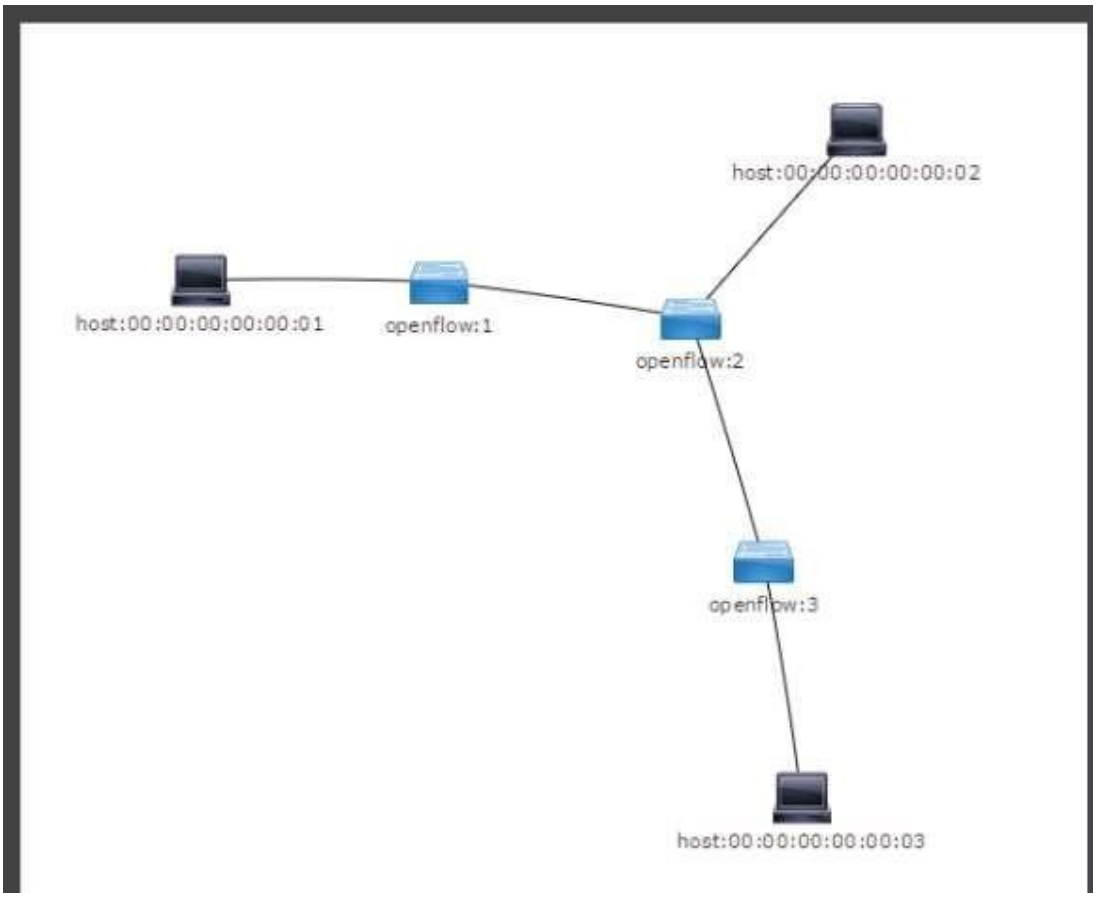
```
brian@odl:~$ ip addr show enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:b0:f6:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb0:f670/64 scope link
       valid_lft forever preferred_lft forever
brian@odl:~$
```

```
brian@odl:~$ sudo nano /etc/network/interfaces
```

Add the following lines to the end of the file */etc/network/interfaces*:

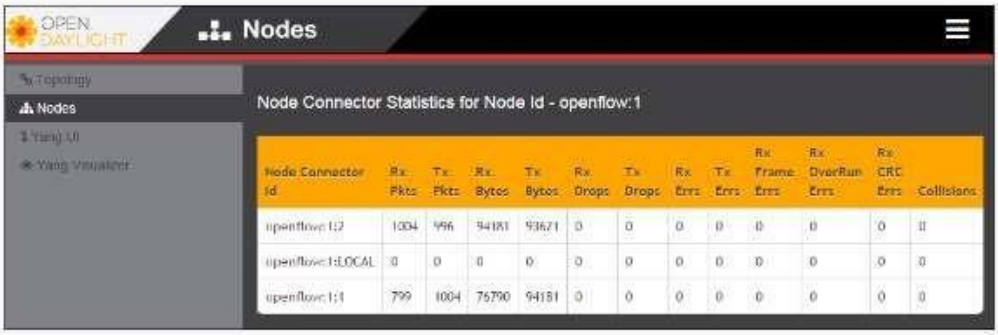
```
# the host-only network interface
auto enp0s8
iface enp0s8 inet dhcp
```


Please Sign In
OPEN
DAYLIGHT
admin
Remember Me
Login

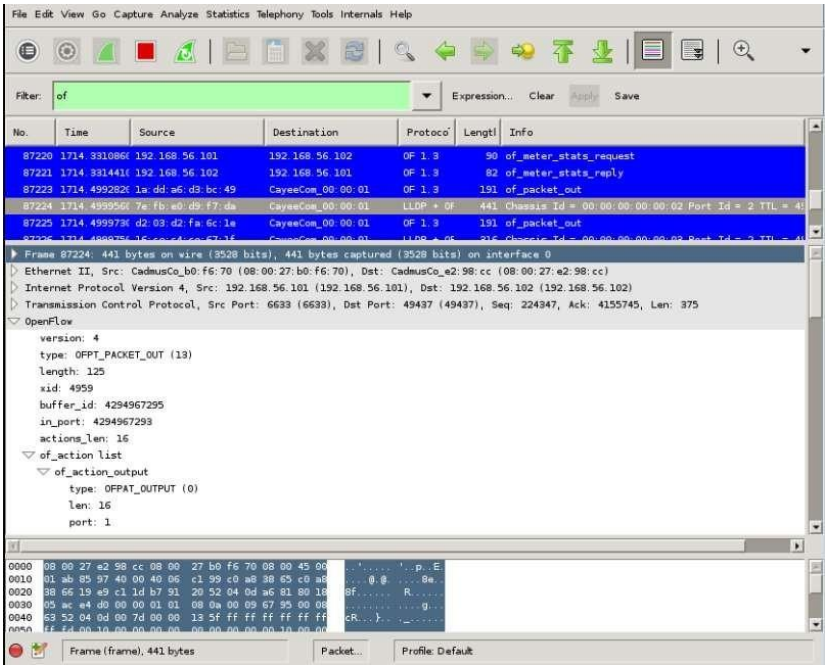

host:00:00:00:00:00:02
host:00:00:00:00:00:01
openflow:1
openflow:2
openflow:3
host:00:00:00:00:00:03

List of nodes

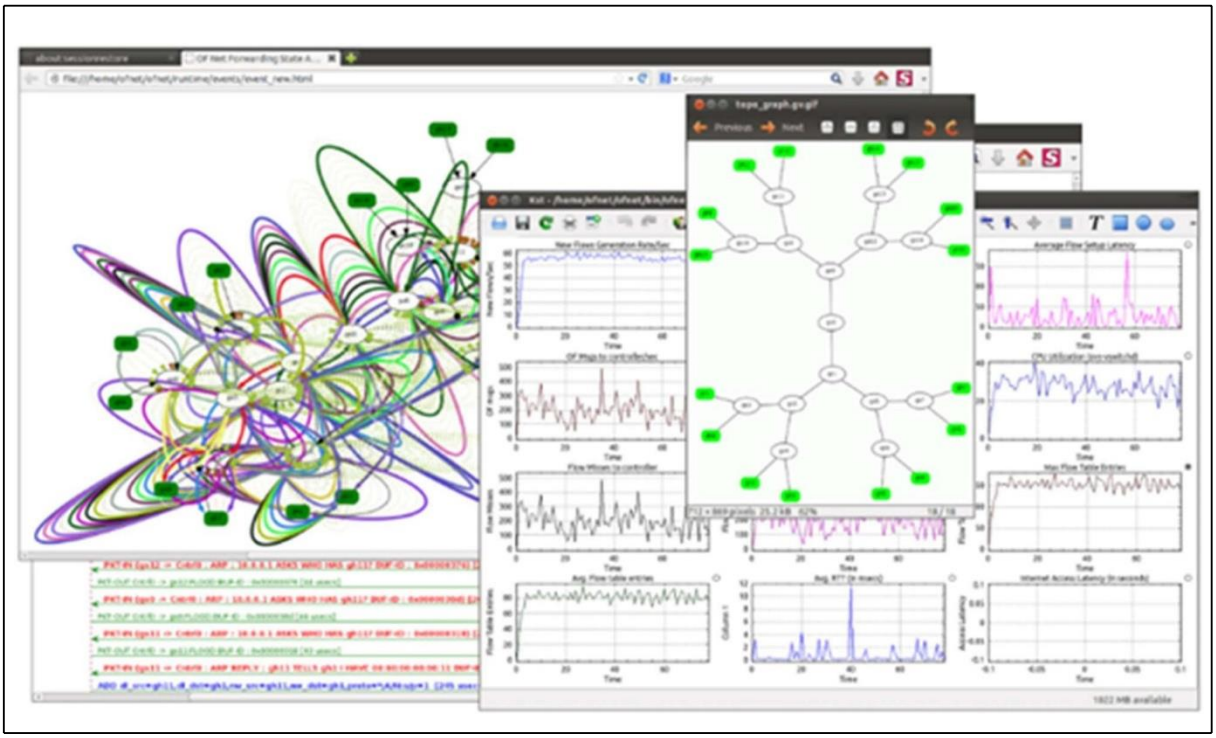
Click on the *Node Connectors* link in each row to see information about each port on the switch:



Interfaces

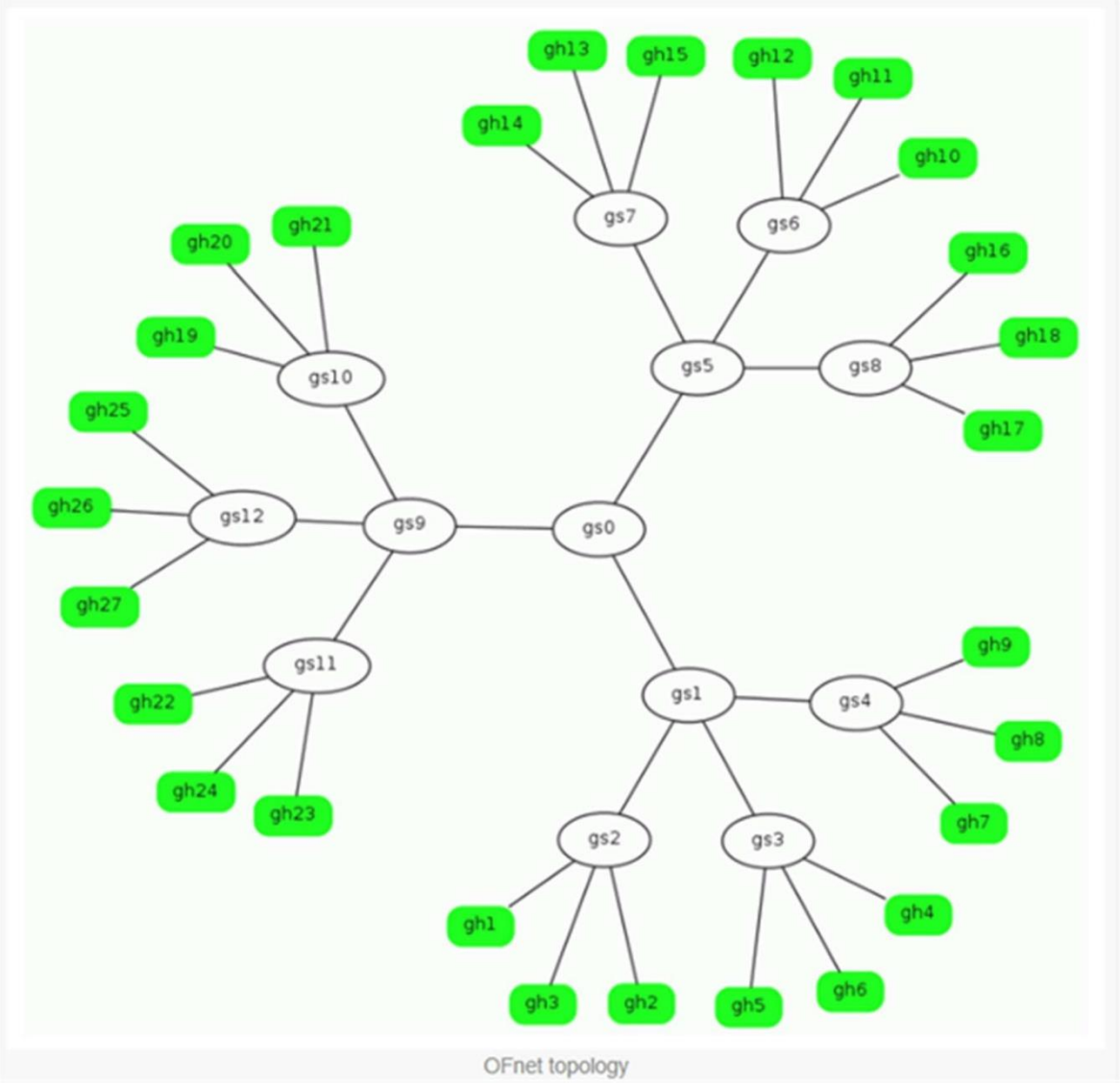


B) OF-Net SDN network emulator.



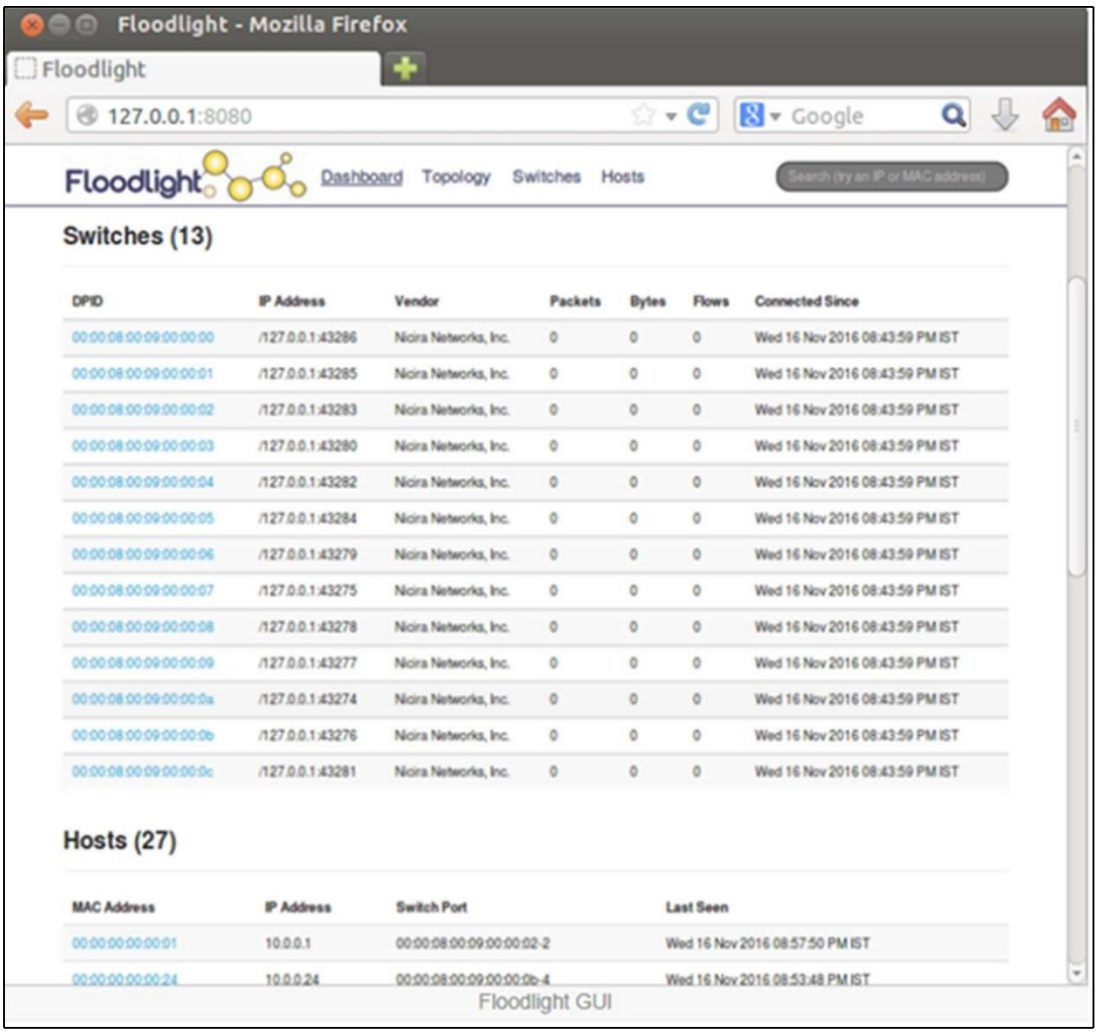
```
$ topo_create -help

Help :
   topo_create -f topo_file_name -t type p1 p2 .. [-c controller_ip]

   Syntax :

   tree    :   topo_create -t tree levels children leaves [-c controller_ip]
   ring    :   topo_create -t ring nodes hosts [-c controller_ip]
   overlay :   topo_create -t overlay vswitches vms [-c controller_ip]
   matrix  :   topo_create -t matrix rows columns hosts [-c controller_ip]
```

```
$ cd ~/ofnet/demo
$ topoc test.topo test.net
```



OFnet topology

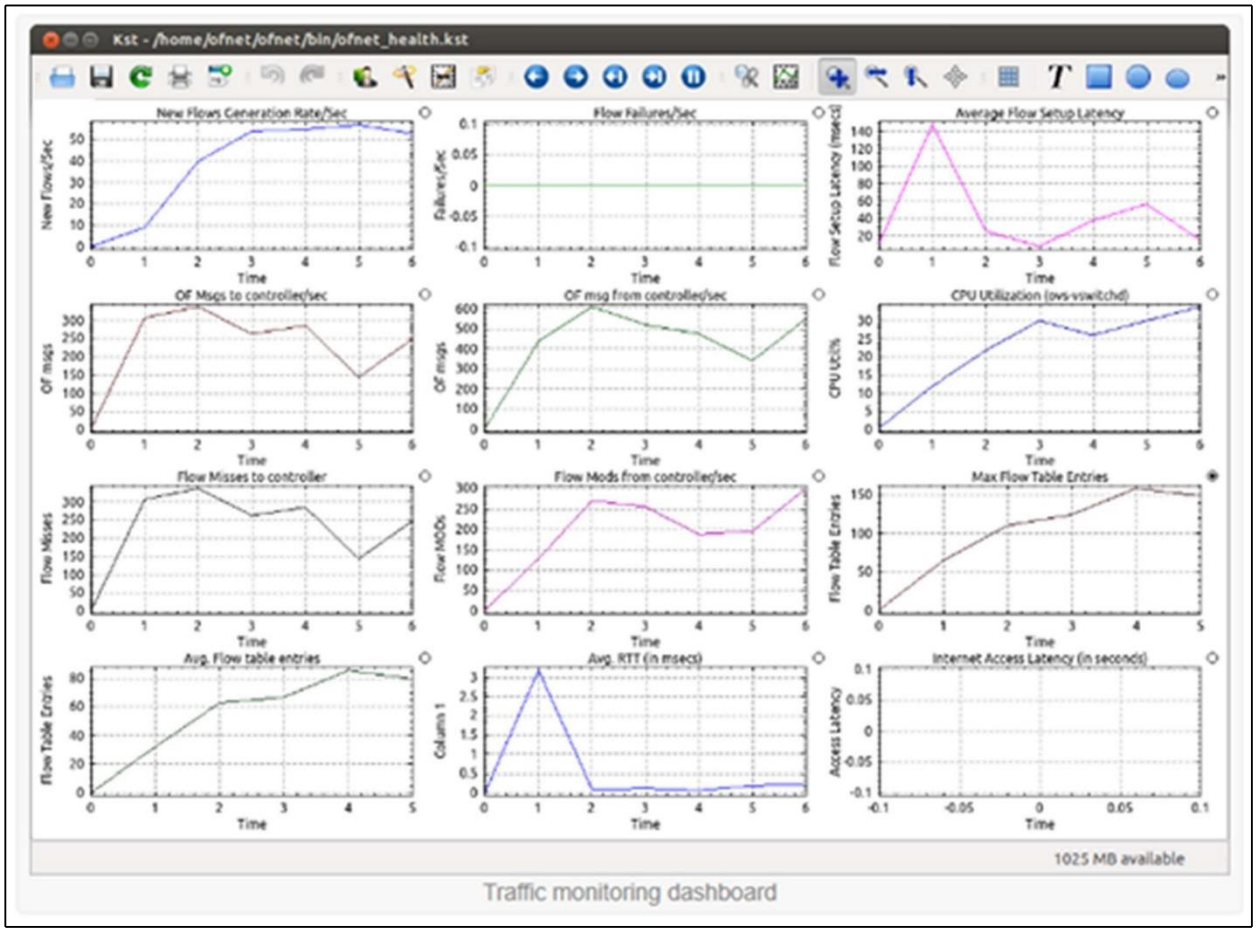
```
CREATING YOUR NETWORK TOPOLOGY. PLEASE WAIT.. THIS MIGHT TAKE SOME TIME

Total Switches : 13, Total Hosts : 27

Test : hsh gh1 ping -c 3 10.0.0.10

PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_req=1 ttl=64 time=1103 ms
64 bytes from 10.0.0.10: icmp_req=2 ttl=64 time=100 ms
64 bytes from 10.0.0.10: icmp_req=3 ttl=64 time=0.138 ms

--- 10.0.0.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.138/401.342/1103.551/498.219 ms, pipe 2
$
```



Floodlight GUI

```
$ tctrl help
     tctrl history last_n#
     tctrl failure_history last_n#
     tctrl fail_list
     tctrl start
     tctrl restart
     tctrl stop
     tctrl exit
     tctrl (all|web|dns|nfs|lsend|ftp|telnet|ping|multicast)(on|off)
     tctrl log off|on
     tctrl set_fps fps#
     tctrl get_fps
     tctrl multicast_server gh#
     tctrl dns_server gh#
     tctrl nfs_server gh#
```

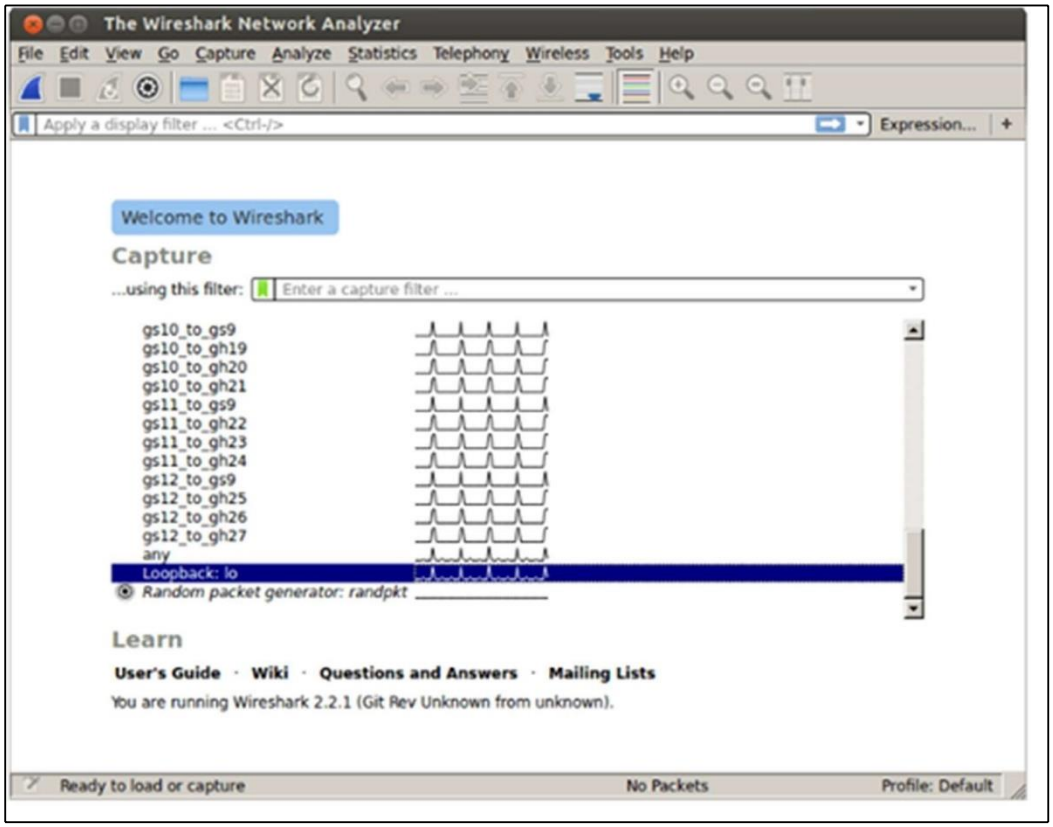
```
$ trafficup
Starting traffic generator..
Erase Previous Failed Events ? (y/n)
y
Traffic Generators Spawned

Traffic configuration file /tmp/traffic.conf does not exist.
Using defaults
Current Traffic Rate (Desired) : 100 Flows/Second
Current Traffic Rate (Actual) : 0 Flows/Second
Current Failure Rate : 0 Flows/Second
Total Flows Generated : 0
Total Failures : 0

Individual Interarrival Times (per host)
    DNS : 480 msecs
    Web : 1200 msecs
    Ping : 1600 msecs
    NFS : 4800 msecs
    Multi-cast : 4800 msecs
    Large-send : 12000 msecs
    FTP : 12000 msecs
    Telnet : 24000 msecs

EVENT_CLOCK_TICK_MS : 300 msecs Total hosts : 24

DNS Server (simulated) : gh2
NFS Server (simulated) : gh22
Multicast Server : gh12

Traffic generator started
All web traffic turned off
You may exit by running `trafficdown` from another window
```

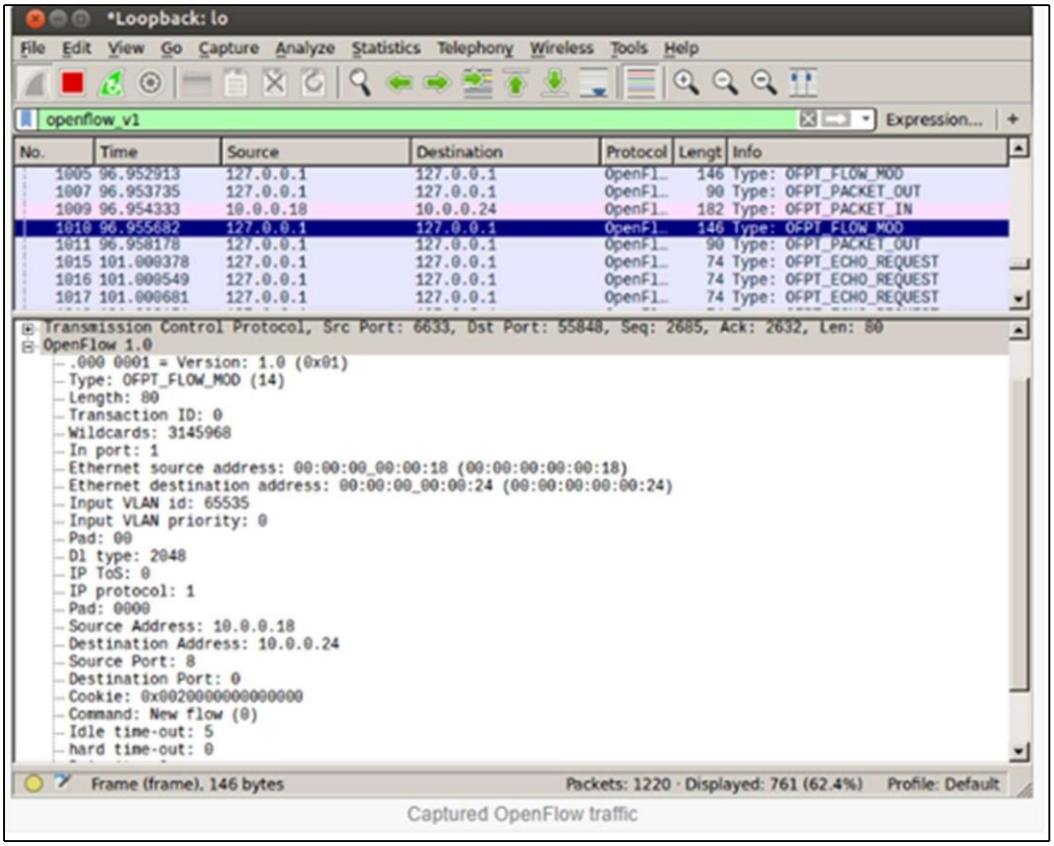


Traffic monitoring dashboard

```
$ tctrl history 4

279. On Host : gh2 Command : hsh gh2 netperf -H 10.0.0.2 -t UDP_RR -l -1 -- -
r 128,128 > /dev/null 2>> error_traffic.log Return Status : 0

278. On Host : gh4 Command : hsh gh4 netperf -H 10.0.0.2 -t UDP_RR -l -1 -- -
r 128,128 > /dev/null 2>> error_traffic.log Return Status : 0

277. On Host : gh3 Command : hsh gh3 ping -c 1 10.0.0.10 > /dev/null 2>> erro
r_traffic.log Return Status : 0

276. On Host : gh3 Command : hsh gh3 netperf -H 10.0.0.2 -t UDP_RR -l -1 -- -
r 128,128 > /dev/null 2>> error_traffic.log Return Status : 0
```

```
$ ofnet_cmds

----------------------------
OFNet Quick command reference
----------------------------

You have enter each of the command with -help option to view the detailed hel
p.

topo_create -> Creates standard topologies - tree, ring, overlay

topoc -> Compiles a custom topology in to a n/w image

ctopo -> Start/stop network, add/delete links, bring down/up links, view curr
ent topology

hsh -> Run a command on a host

pingall -> Every host pings every other host

npings -> Generate n random pings

fstate -> Show current forwarding state

ofevent -> Run a command as an event and show what happened

event_animate -> Show the previously ofevent as an animation

trafficup -> Start traffic generator

trafficdown -> Stop traffic generator

tctrl -> View/control traffic generation

save_ofevent -> Saves a the output of last ofevent command as a zip file

view_ofevent -> To view saved ofevent as a zip file

ofclean -> Clean directories in case of errors

ofstat -> Display various statistics
```


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter: Enter a capture filter ...
gs10_to_gs9
gs10_to_gh19
gs10_to_gh20
gs10_to_gh21
gs11_to_gs9
gs11_to_gh22
gs11_to_gh23
gs11_to_gh24
gs12_to_gs9
gs12_to_gh25
gs12_to_gh26
gs12_to_gh27
any
Loopback: lo
Random packet generator: randpkt
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.2.1 (Git Rev Unknown from unknown).
Ready to load or capture
No Packets
Profile: Default


*Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
openflow_v1
Expression...
No. Time Source Destination Protocol Lengt Info
1005 96.952913 127.0.0.1 127.0.0.1 OpenFl... 146 Type: OFPT_FLOW_MOD
1007 96.953735 127.0.0.1 127.0.0.1 OpenFl... 90 Type: OFPT_PACKET_OUT
1009 96.954333 10.0.0.18 10.0.0.24 OpenFl... 182 Type: OFPT_PACKET_IN
1010 96.955682 127.0.0.1 127.0.0.1 OpenFl... 146 Type: OFPT_FLOW_MOD
1011 96.958178 127.0.0.1 127.0.0.1 OpenFl... 90 Type: OFPT_PACKET_OUT
1015 101.000378 127.0.0.1 127.0.0.1 OpenFl... 74 Type: OFPT_ECHO_REQUEST
1016 101.000549 127.0.0.1 127.0.0.1 OpenFl... 74 Type: OFPT_ECHO_REQUEST
1017 101.000681 127.0.0.1 127.0.0.1 OpenFl... 74 Type: OFPT_ECHO_REQUEST
Transmission Control Protocol, Src Port: 6633, Dst Port: 55848, Seq: 2685, Ack: 2632, Len: 80
OpenFlow 1.0
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_FLOW_MOD (14)
Length: 80
Transaction ID: 0
Wildcards: 3145968
In port: 1
Ethernet source address: 00:00:00_00:00:18 (00:00:00:00:00:18)
Ethernet destination address: 00:00:00_00:00:24 (00:00:00:00:00:24)
Input VLAN id: 65535
Input VLAN priority: 0
Pad: 00
Dl type: 2048
IP ToS: 0
IP protocol: 1
Pad: 0000
Source Address: 10.0.0.18
Destination Address: 10.0.0.24
Source Port: 8
Destination Port: 0
Cookie: 0x0020000000000000
Command: New flow (0)
Idle time-out: 5
hard time-out: 0
Frame (frame), 146 bytes
Packets: 1220 · Displayed: 761 (62.4%)
Profile: Default
Captured OpenFlow traffic