R.D.&S.H NATIONAL COLLEGE & S.WA. SCIENCE COLLEGE

Bandra, Mumbai - 400050

DEPARTMENT OF COMPUTER SCIENCE

M.Sc. Computer Science – Semester I

Software Defined Networking (SDN)

JOURNAL 2023-2024

Seat No. 1290030

R.D. & S.H. NATIONAL COLLEGE & S. W.A. SCIENCE COLLEGE,

Bandra, Mumbai – 400050.

Department of Computer Science

CERTIFICATE

This is to certify that **Mr/Ms. Jishan Farid Ahmed Shaikh** of **M.Sc Part I (Sem I)** class has satisfactorily completed **9** Practicals in the subject of **Software Defined Networking (SDN)** as a part of M.Sc. Degree Course in Computer Science during the academic year 2023 – 2024.

Date of Submission:

Faculty Incharge

Co-ordinator,

Department of Computer Science

Signature of External Examiner

Sr No.	Date	Aim	PAGE NO	
1		Implement IP SLA (IP Service Level Agreement)		
2		Implement IPv4 ACLs a) Standard ACL b) Extended ACL		
3		Implement SPAN Technologies (Switch Port Analyzer)		
4		a) Implement a GRE Tunnel b) Implement VTP c) Implement NAT		
5		Implement Inter-VLAN Routing		
6		Observe STP Topology Changes and Implement RSTP		
7		a) Implement Ether Channel b) Tune and Optimize Ether Channel Operations		
8		Implement Single-Area OSPFv2		
9		a) Implement BGP Communities b) Implement MP-BGP c) Implement eBGP for IPv4 d) Implement BGP Path Manipulation		

Practical 1

Aim : Implement IP SLA (IP Service Level Agreement)

What is IP SLA?

→ IP SLA, which stands for Internet Protocol Service Level Agreement, is a feature of Cisco networking devices that allows them to measure and monitor network performance. It enables network administrators to assess the health and reliability of a network by simulating network data and monitoring various parameters.

IP SLA works by generating synthetic traffic, such as pings or other types of probes, and then measuring the performance of the network based on the responses. This can help identify issues such as latency, packet loss, and jitter. The collected data can be used for network troubleshooting, performance monitoring, and reporting.

Some common uses of IP SLA include:

1. Network Performance Monitoring: IP SLA can be configured to continuously monitor the performance of specific network paths or devices. This allows administrators to proactively identify and address potential issues before they impact users.

2. Quality of Service (QoS) Monitoring: IP SLA can be used to monitor the performance of different classes of service within a network, helping to ensure that QoS policies are being enforced effectively.

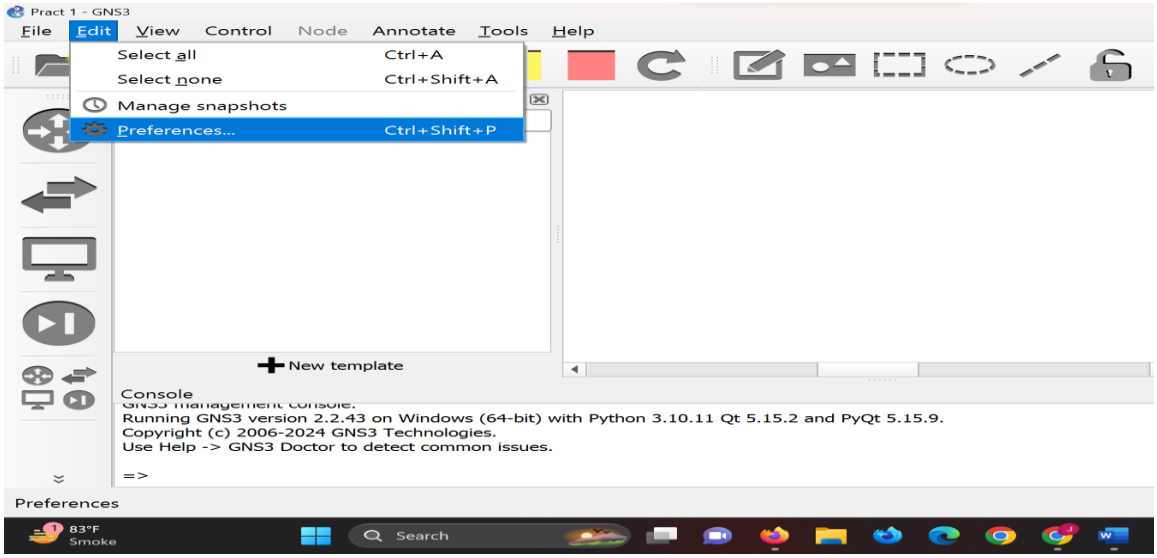
3. WAN Link Utilization: IP SLA can be employed to measure the performance of WAN (Wide Area Network) links, helping administrators to optimize bandwidth usage and detect any degradation in link quality.

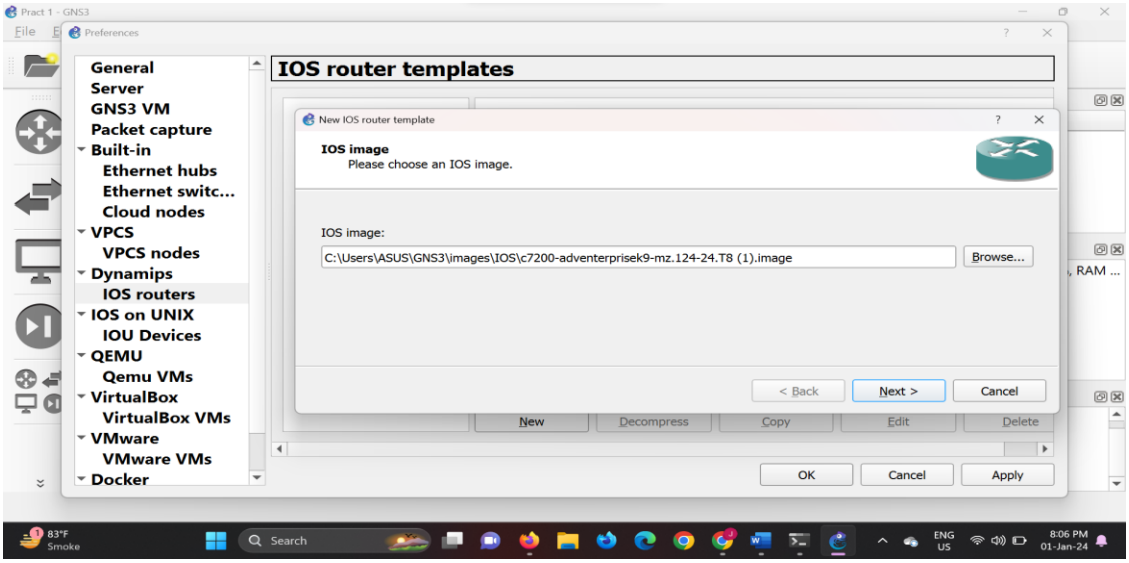
4. Failover and Redundancy Testing: IP SLA can be utilized to test the failover capabilities of redundant paths in a network. This is particularly important for ensuring high availability and minimizing downtime.

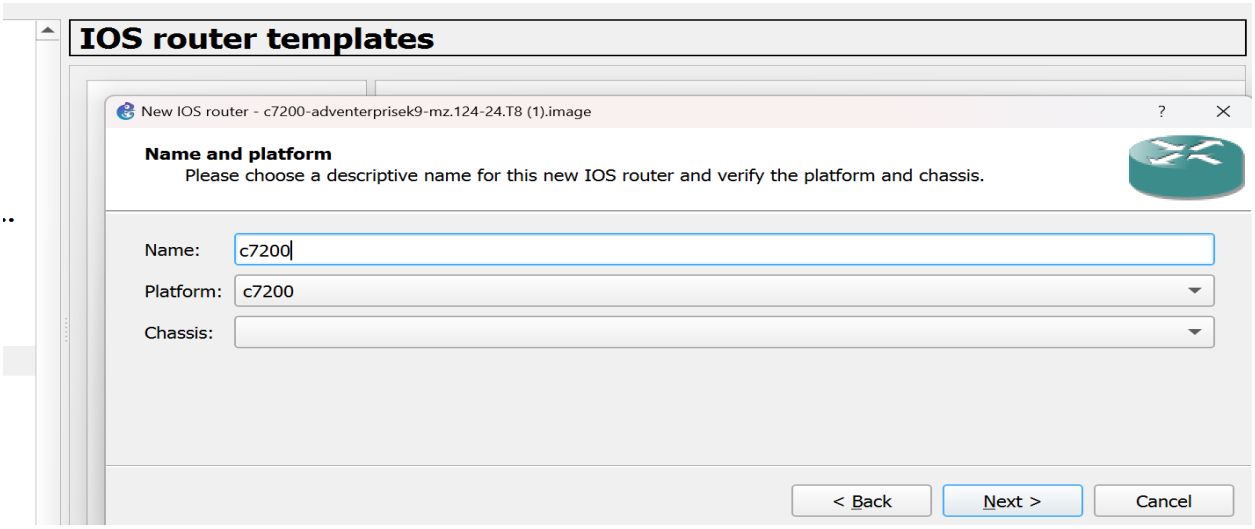
5. VoIP (Voice over IP) Monitoring: In VoIP networks, IP SLA can be used to monitor key performance metrics, such as jitter and latency, to ensure high-quality voice communications.

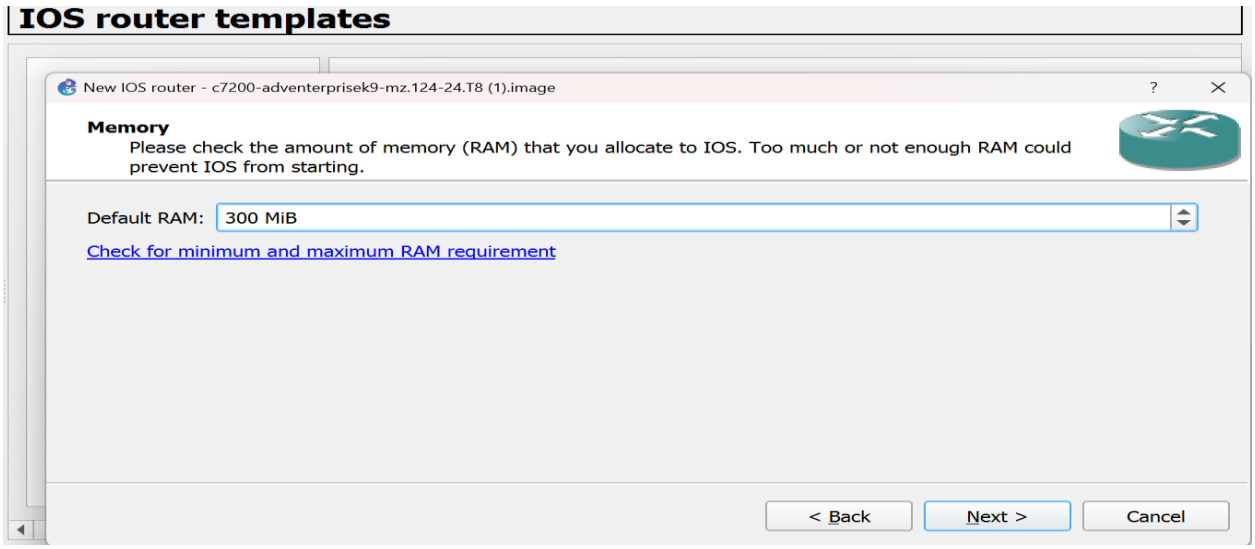
It's important to note that while IP SLA is a Cisco-specific feature, similar functionality may be available in other network equipment from different vendors, often under different names or acronyms.

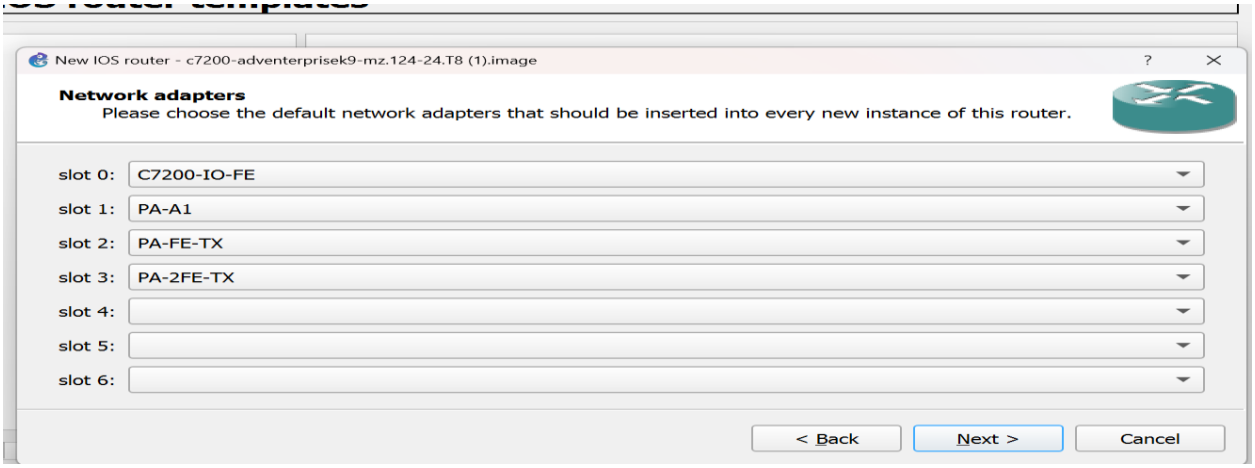
Step 1 : Download and Load Image File :



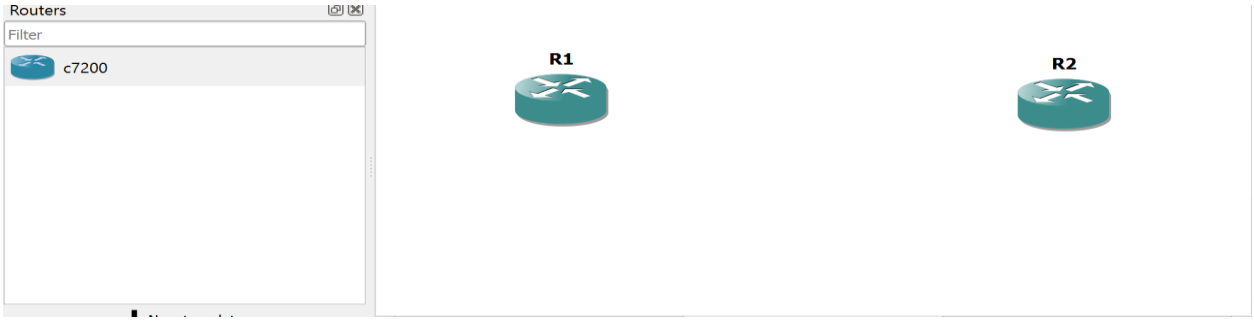


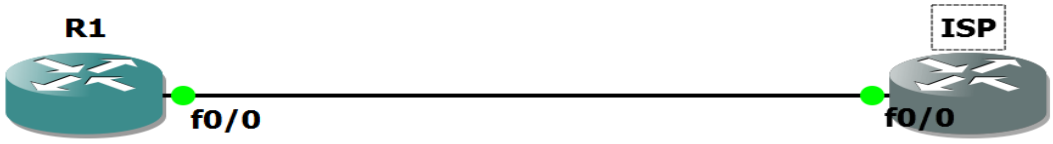






Step 2 : Build the Network





Step 3 : Configure Router

Router 1 :

```
R1#
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int
R1(config)#interface fast
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 209.165.200.9 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
*Jan  1 20:24:22.135: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan  1 20:24:23.135: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/
0, changed state to up
R1(config-if)#exit
R1(config)#
```

```
R1(config)#do sh ip int bri
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            209.165.200.9   YES manual up                    up
ATM1/0                     unassigned      YES unset  administratively down down
FastEthernet2/0            unassigned      YES unset  administratively down down
FastEthernet3/0            unassigned      YES unset  administratively down down
FastEthernet3/1            unassigned      YES unset  administratively down down
R1(config)#
```

Router 2 :

```
ISP#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#interface fas
ISP(config)#interface fastEthernet 0/0
ISP(config-if)#ip addr
ISP(config-if)#ip address 209.165.200.10 255.255.255.252
ISP(config-if)#mp shu
ISP(config-if)#no shut
ISP(config-if)#no shutdown
ISP(config-if)#
*Jan  1 20:28:22.423: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan  1 20:28:23.423: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0
, changed state to up
ISP(config-if)#exit
ISP(config)#
```

```
ISP(config)#inter
ISP(config)#interface loop
ISP(config)#interface loopback 0
ISP(config-if)#
*Jan  1 20:29:09.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, cha
nged state to up
ISP(config-if)#ip add
ISP(config-if)#ip address 198.133.219.1 255.255.255.255
ISP(config-if)#no shu
ISP(config-if)#no shutdown
ISP(config-if)#exit
ISP(config)#
```

```
R1#
*Jan  1 20:30:58.807: %SYS-5-CONFIG_I: Configured from console by console
R1#
R1#ping 209.165.200.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.10, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 20/31/40 ms
```

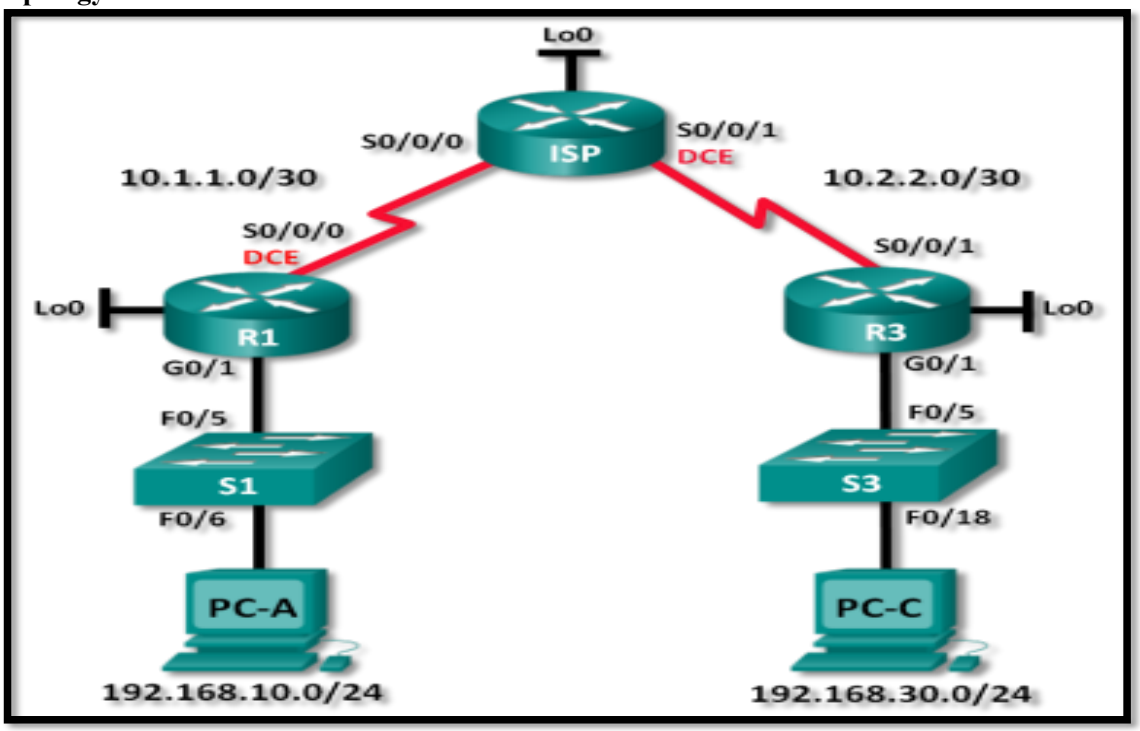
```
ISP#wr
Building configuration...
[OK]
ISP#
ISP#copy run
ISP#copy running-config star
ISP#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
ISP#
```

```
#show ip sla statistics
```

PRACTICAL 2

Lab – Configuring and Verifying Standard IPv4 ACLs :-

Topology



Lab – Configuring and Verifying Standard IPv4 ACLs

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.10.1	255.255.255.0	N/A
	Lo0	192.168.20.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A
ISP	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A

	Lo0	209.165.200.225	255.255.255.224	N/A
R3	G0/1	192.168.30.1	255.255.255.0	N/A
	Lo0	192.168.40.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
S1	VLAN 1	192.168.10.11	255.255.255.0	192.168.10.1
S3	VLAN 1	192.168.30.11	255.255.255.0	192.168.30.1

PC-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
PC-C	NIC	192.168.30.3	255.255.255.0	192.168.30.1

Objectives

Part 1: Set Up the Topology and Initialize Devices · Set up equipment to match the network topology. Initialize and reload the routers and switches.

Part 2: Configure Devices and Verify Connectivity. Assign a static IP address to PCs. Configure basic settings on routers.

Configure basic settings on switches. Configure OSPF routing on R1, ISP, and R3. Verify connectivity between devices.

Part 3: Configure and Verify Standard Numbered and Named ACLs. Configure, apply, and verify a numbered standard ACL. · Configure, apply, and verify a named ACL.

Part 4: Modify a Standard ACL. Modify and verify a named standard ACL.Test the ACL.

Background / Scenario

Network security is an important issue when designing and managing IP networks. The ability to configure proper rules to filter packets, based on established security policies, is a valuable skill.

Lab – Configuring and Verifying Standard IPv4 ACLs

In this lab, you will set up filtering rules for two offices represented by R1 and R3. Management has established some access policies between the LANs located at R1 and R3, which you must implement. The ISP router sitting between R1 and R3 will not have any ACLs placed on it. You would not be allowed any administrative access to an ISP router because you can only control and manage your own equipment.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4) M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

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

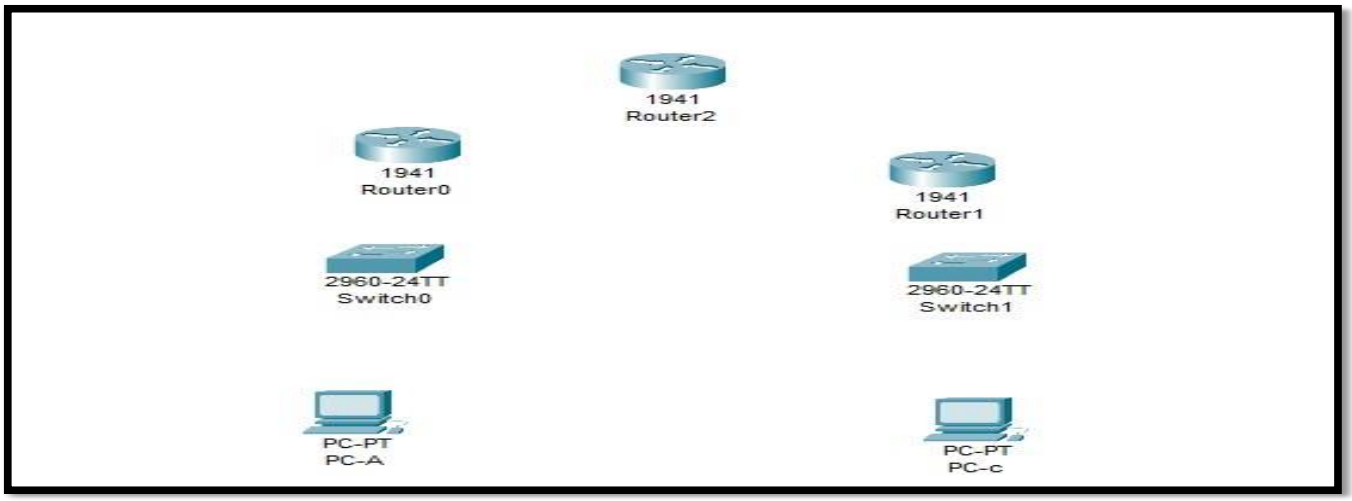
Required Resources

· 3 Routers (Cisco 1941 with Cisco IOS Release 15.2(4) M3 universal image or comparable) · 2 Switches (Cisco 2960 with Cisco IOS Release

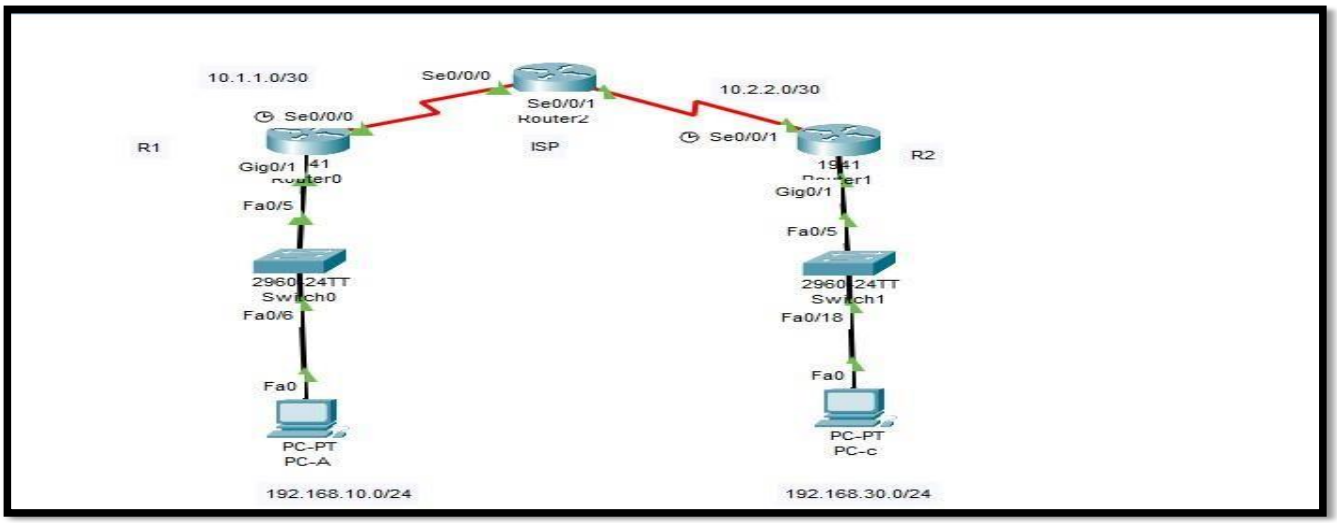
15.0(2) lanbasek9 image or comparable) · 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term) · Console cables to configure the Cisco IOS devices via the console ports · Ethernet and serial cables as shown in the topology

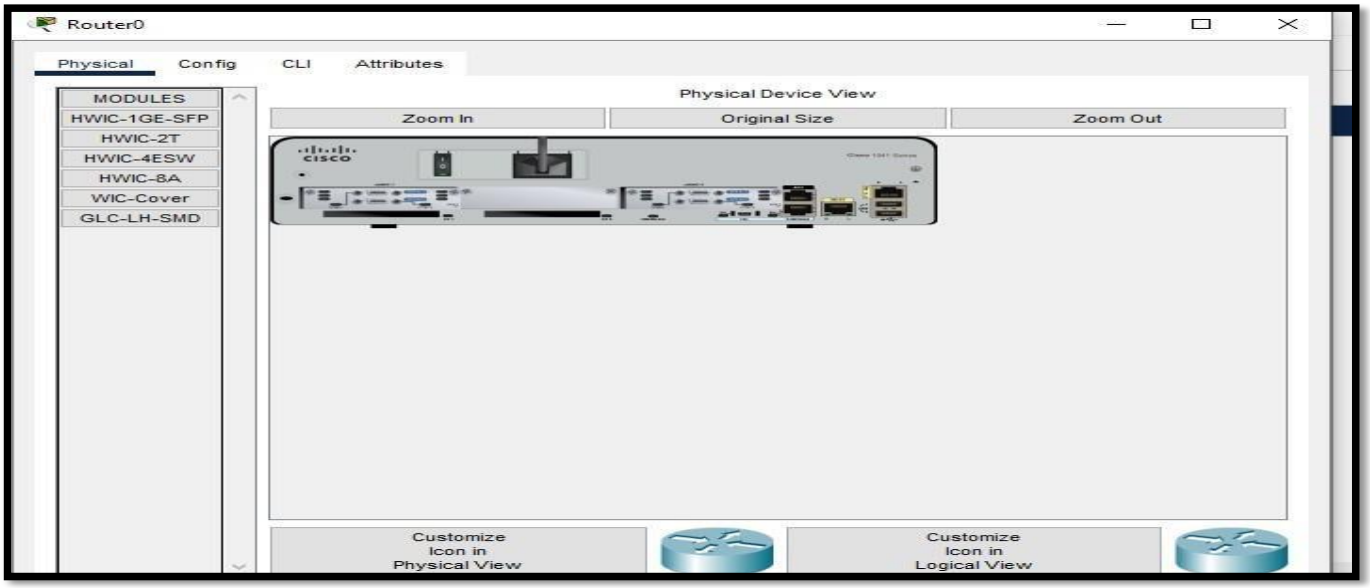
Part 1: Set Up the Topology and Initialize Devices

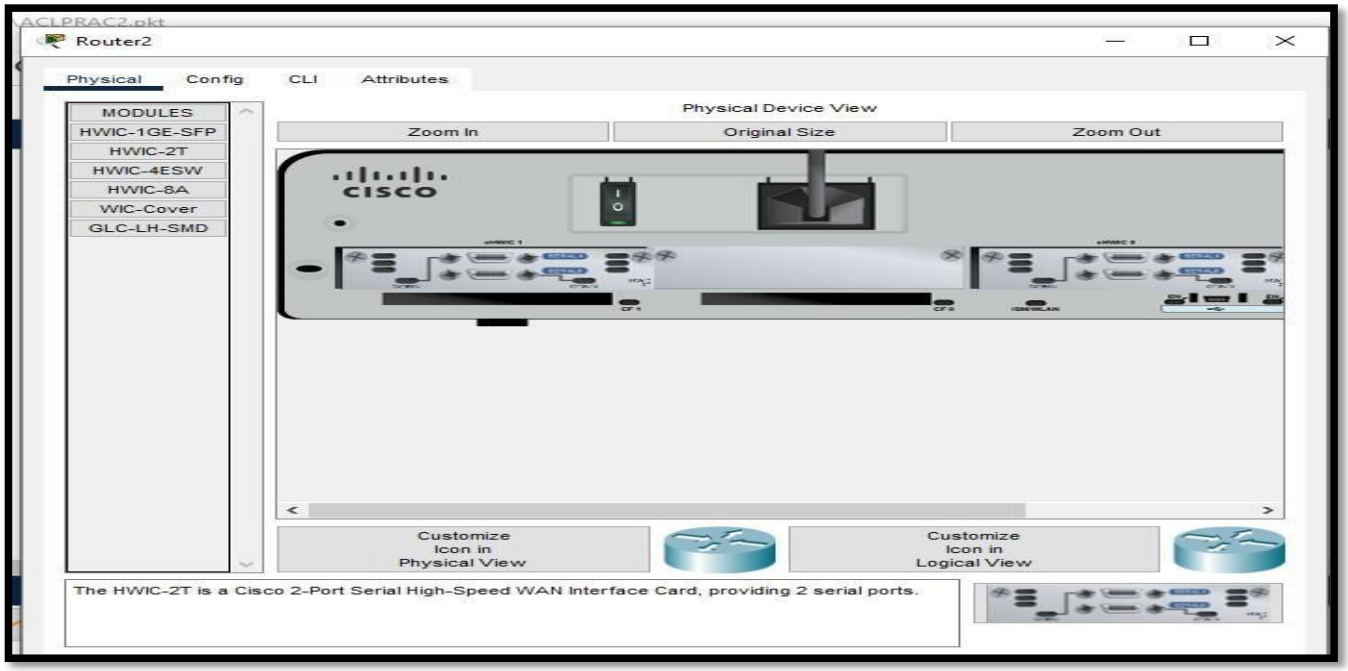
Step 1 : Cable the network as shown in the topology.

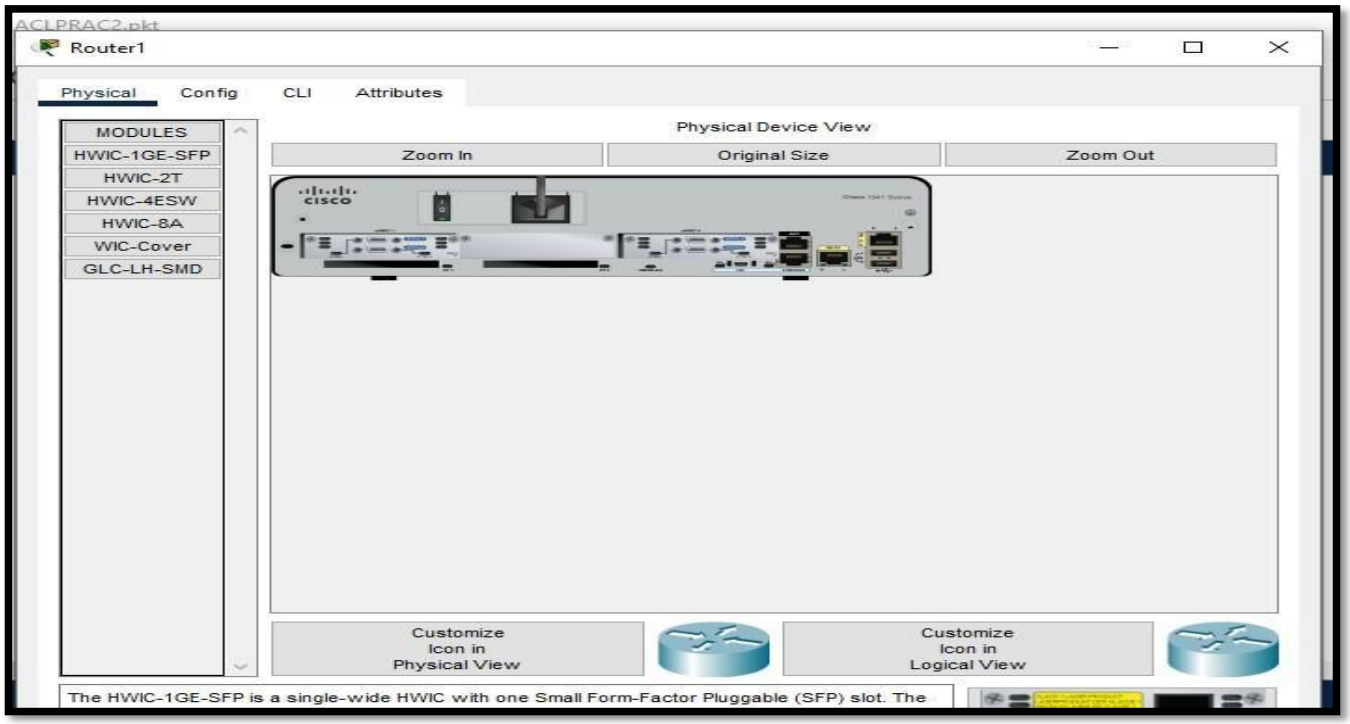


Step 2:- Initialize and reload the routers and switches.



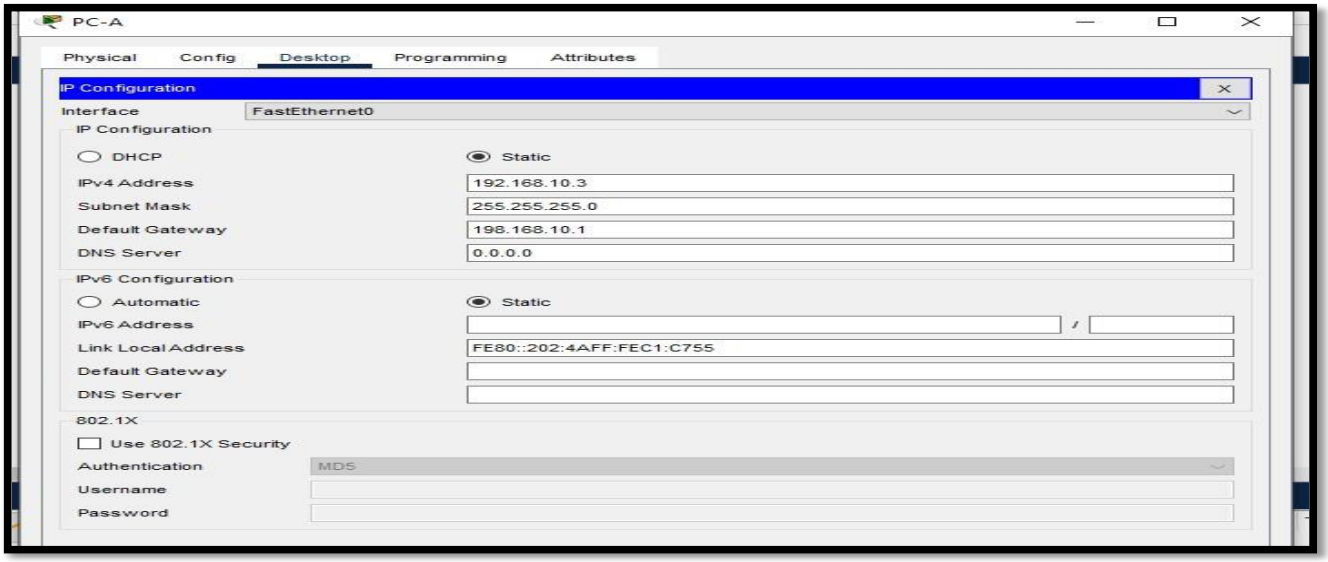


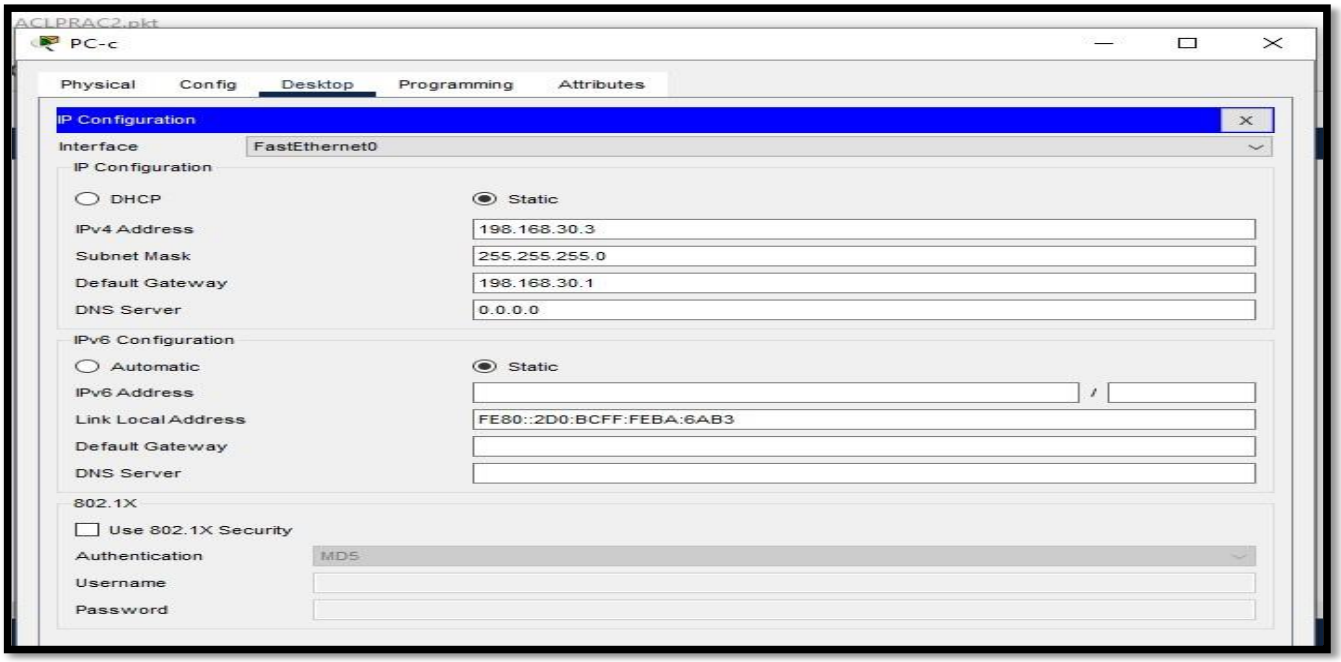




Part 2: Configure Devices and Verify Connectivity

Step 1:- Configure IP addresses on PC-A and PC-C.





Configure basic settings for the routers.

a. Console into the router and enter global configuration mode.

b. Create loopback interfaces on each router as shown in the Addressing Table.

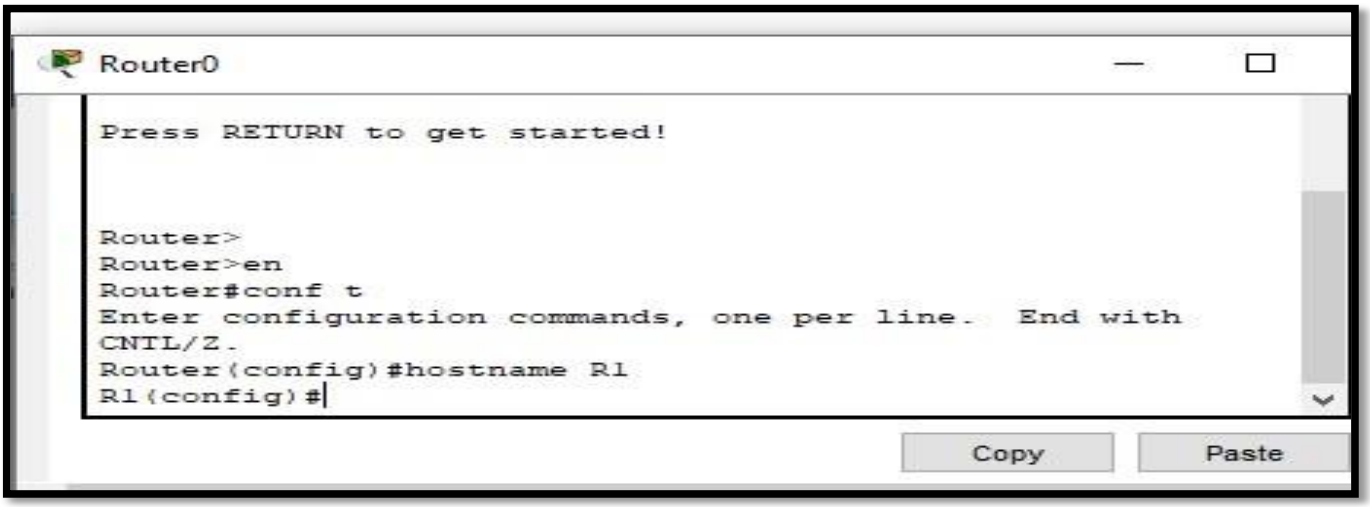
c. Configure interface IP addresses as shown in the Topology and Addressing Table.

d. Assign a clock rate of 128000 to the DCE serial interfaces.

a. Router > en

Router > conf t

Router > hostname R1


Router0
Press RETURN to get started!
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with
CNTL/Z.
Router(config)#hostname R1
R1(config)#
Copy
Paste

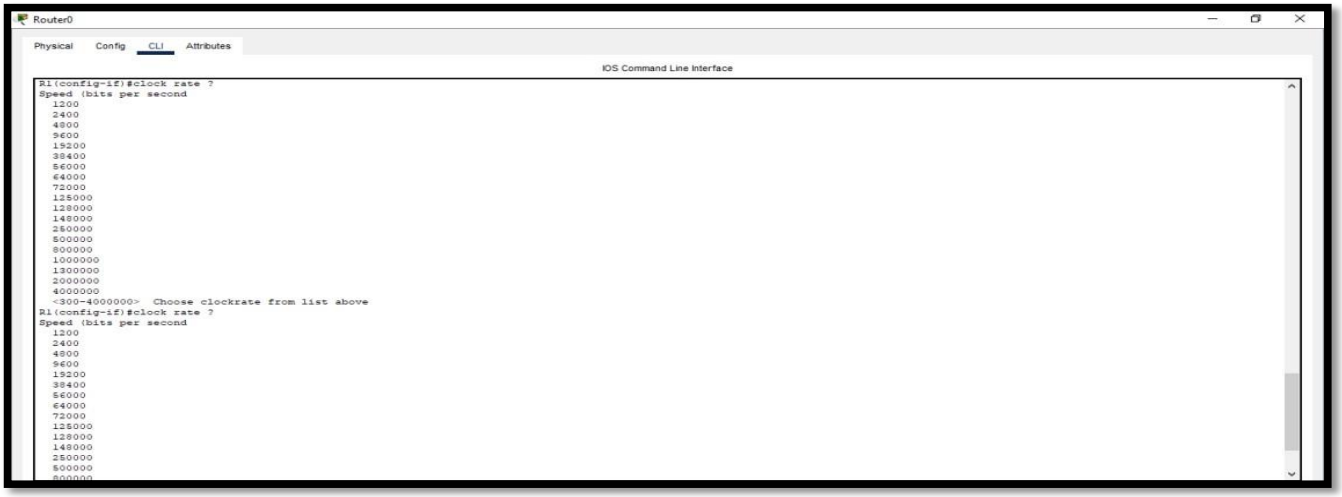
Router0

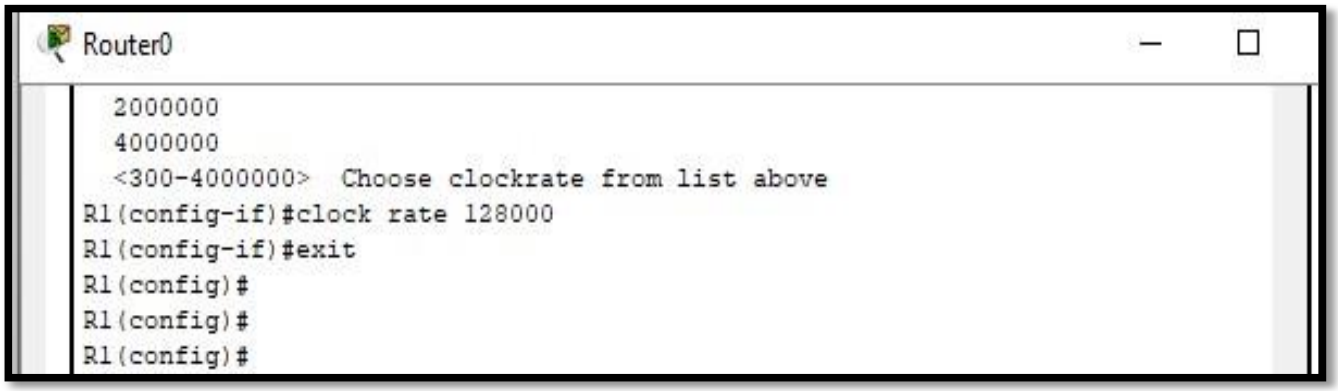
```
R1(config)#interface loopback 0

R1(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up

R1(config-if)#ip address 192.168.20.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

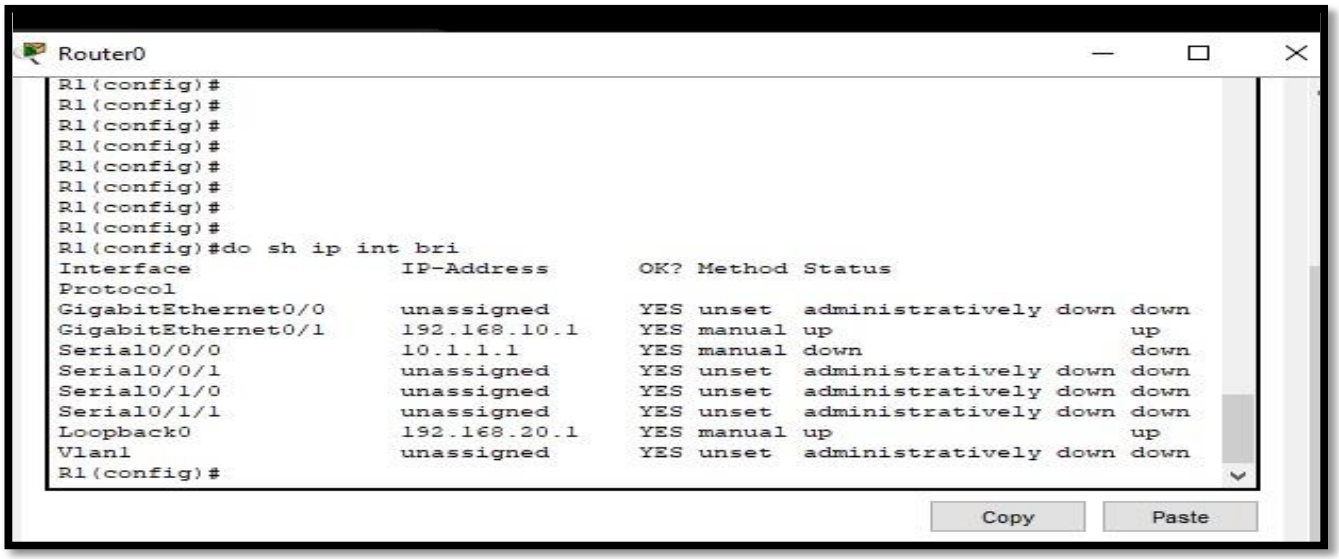
```
R1(config)#interface gigabitEthernet
R1(config)#interface gigabitEthernet 0/1
R1(config-if)#ip address 192.168.10.1 255.255.255.0
R1(config-if)#no shutdown

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to
up

R1(config-if)#exit
```

Router0

```
R1(config)#
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.252
R1(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
R1(config)#
R1(config)#
```





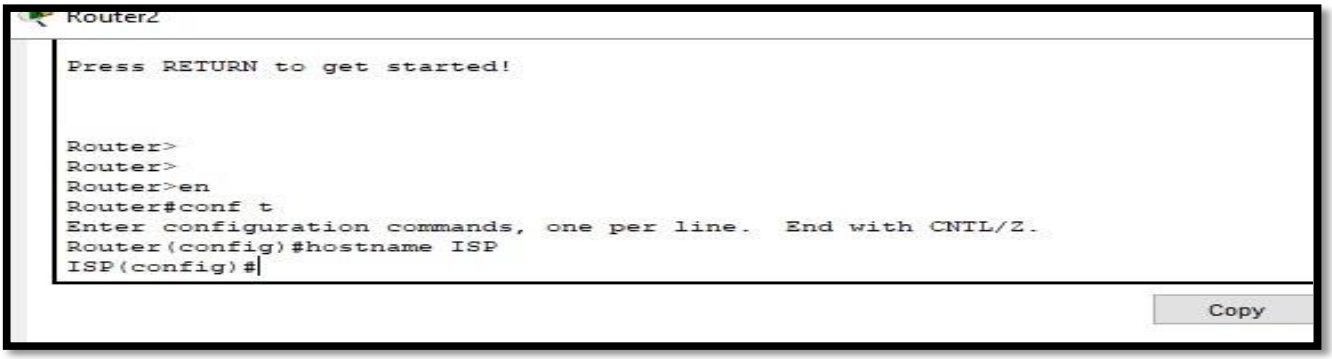


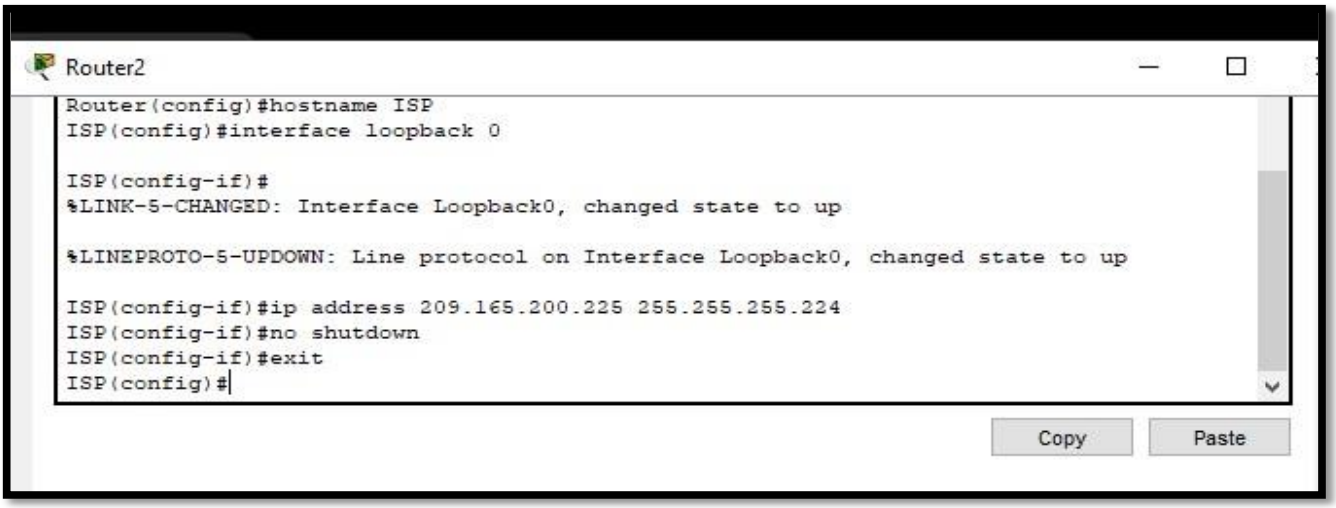
ISP

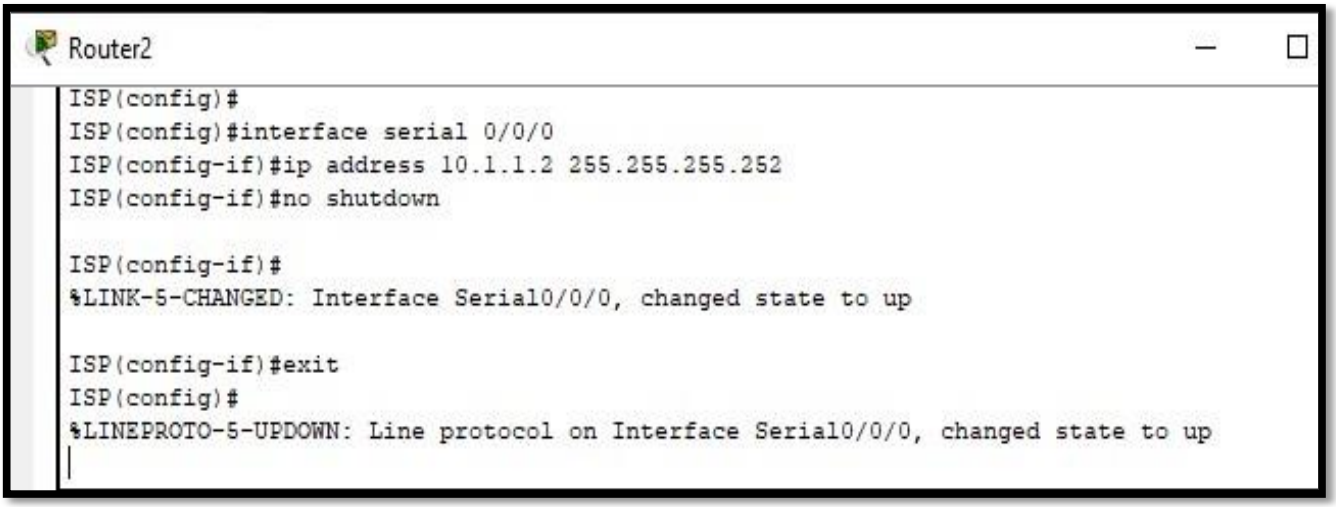
Router > en

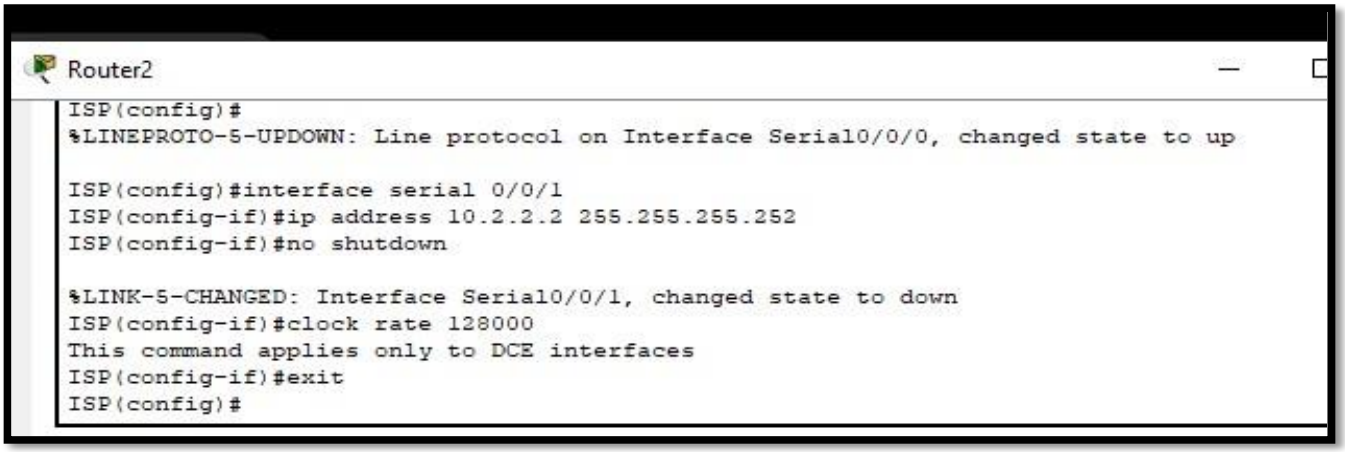
Router > conf t

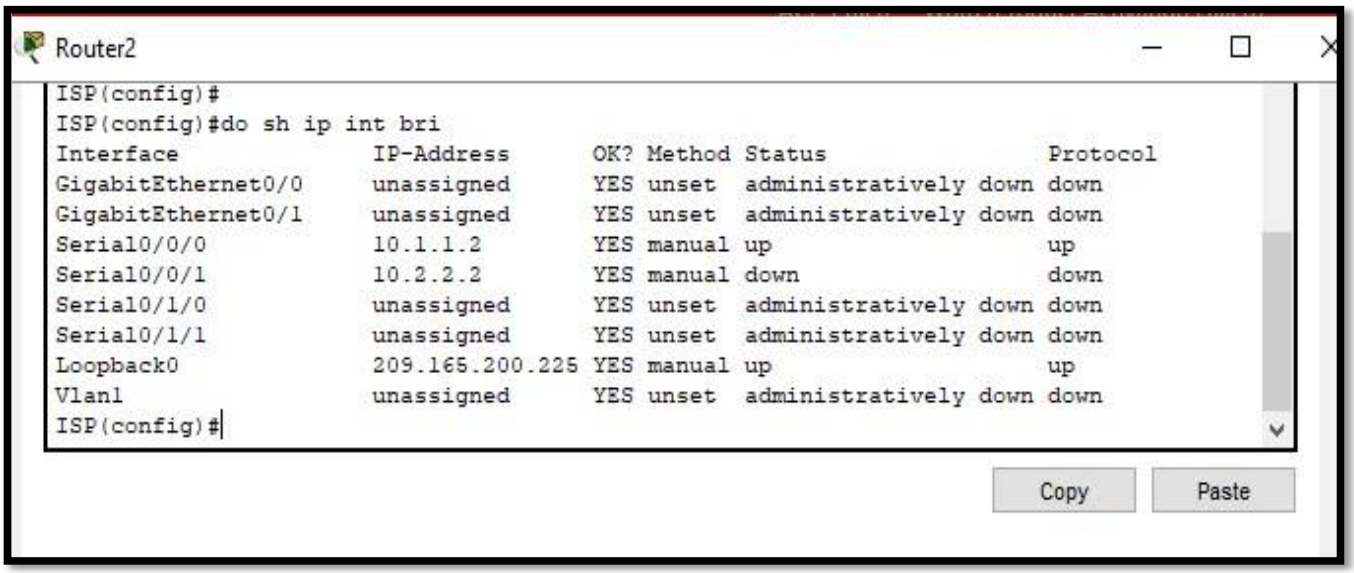
Router > hostname ISP


Router2
Press RETURN to get started!
Router>
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ISP
ISP(config)#
Copy


Router2
Router(config)#hostname ISP
ISP(config)#interface loopback 0
ISP(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
ISP(config-if)#ip address 209.165.200.225 255.255.255.224
ISP(config-if)#no shutdown
ISP(config-if)#exit
ISP(config)#
Copy
Paste


Router2
ISP(config)#
ISP(config)#interface serial 0/0/0
ISP(config-if)#ip address 10.1.1.2 255.255.255.252
ISP(config-if)#no shutdown
ISP(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
ISP(config-if)#exit
ISP(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up



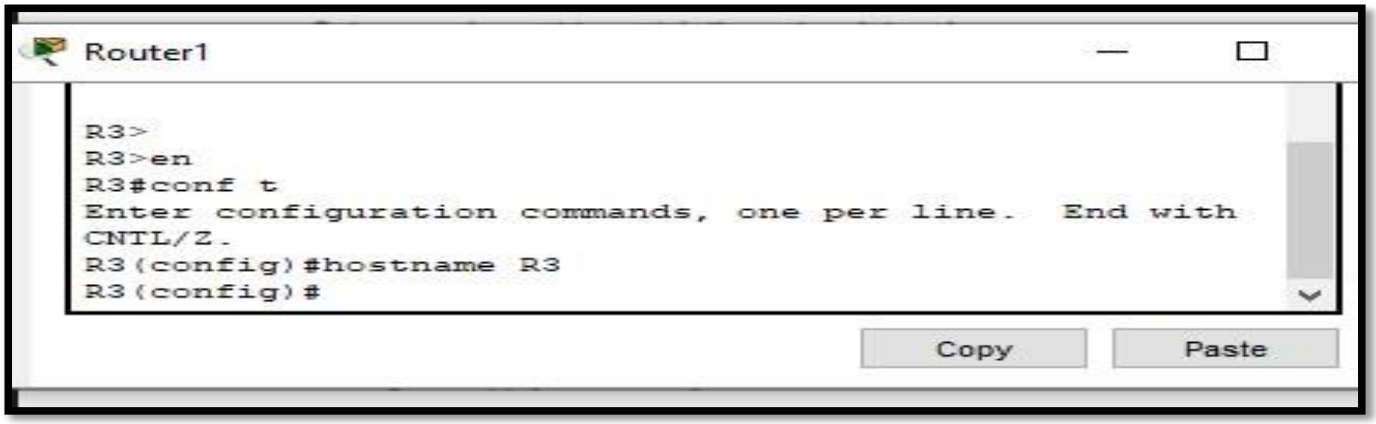


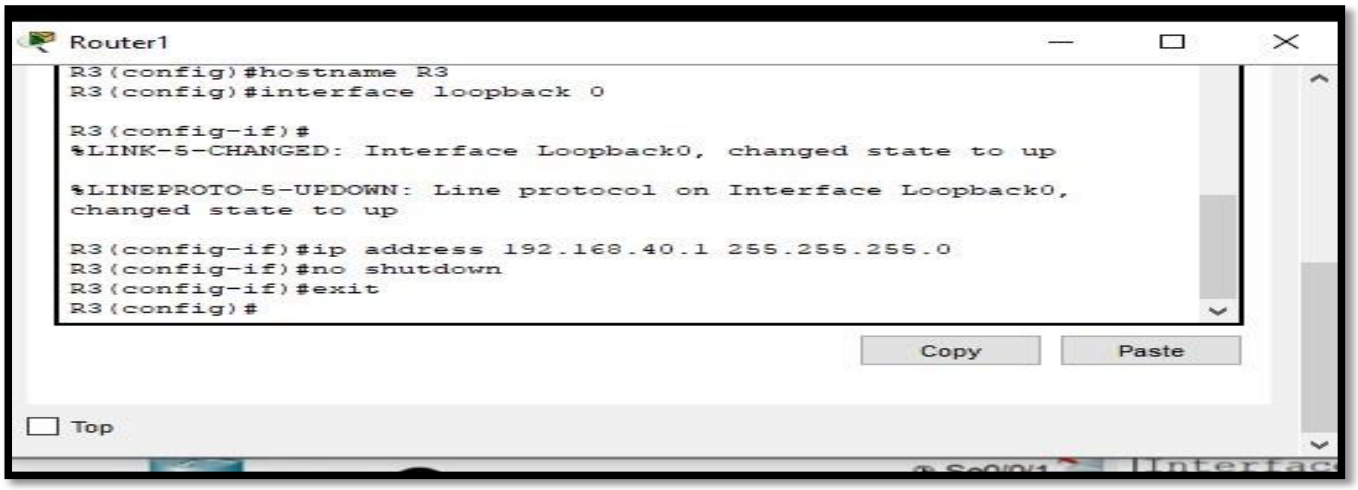
R3

Router > en

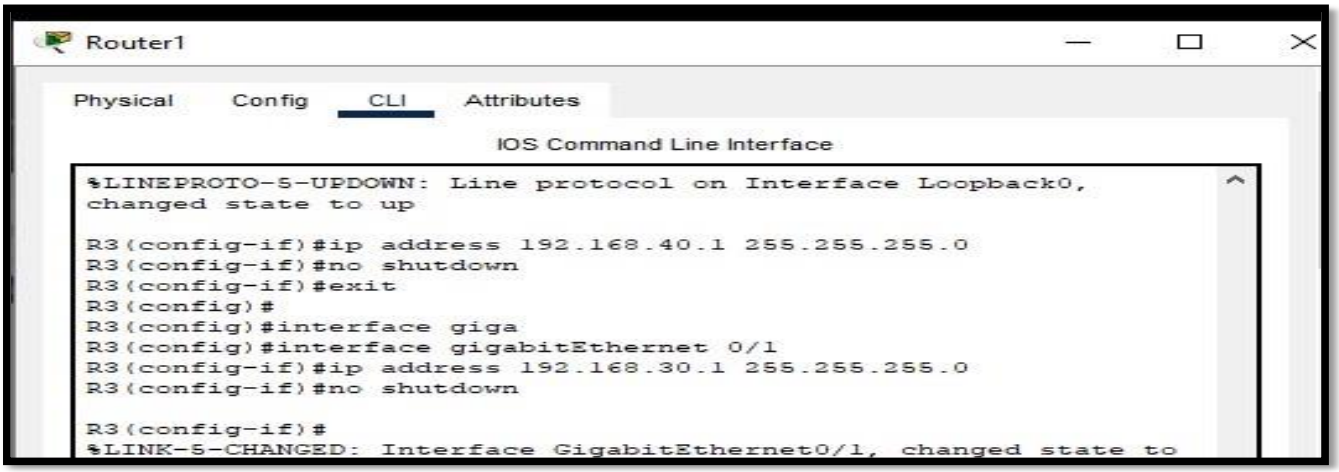
Router > conf t

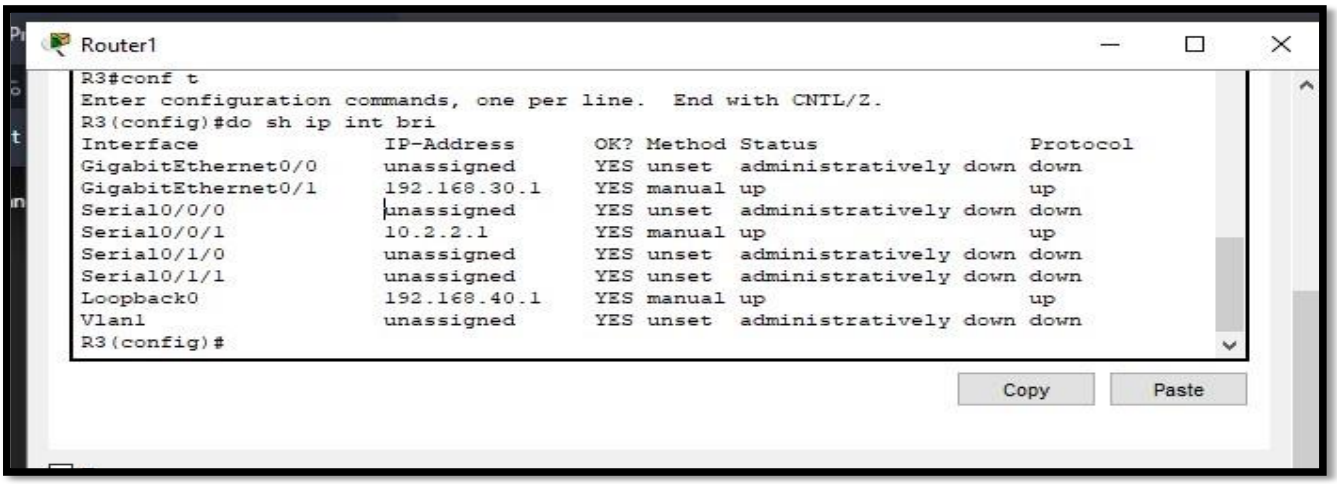
Router > hostname R3


Router1
R3>
R3>en
R3#conf t
Enter configuration commands, one per line. End with
CNTL/Z.
R3(config)#hostname R3
R3(config)#
Copy
Paste


Router1
R3(config)#hostname R3
R3(config)#interface loopback 0
R3(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
R3(config-if)#ip address 192.168.40.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
Copy
Paste
Top


Router1
R3(config)#interface gigabitEthernet 0/1
R3(config-if)#ip address 192.168.30.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to
up
%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to up
R3(config-if)#exit
R3(config)#
Copy
Paste
Top

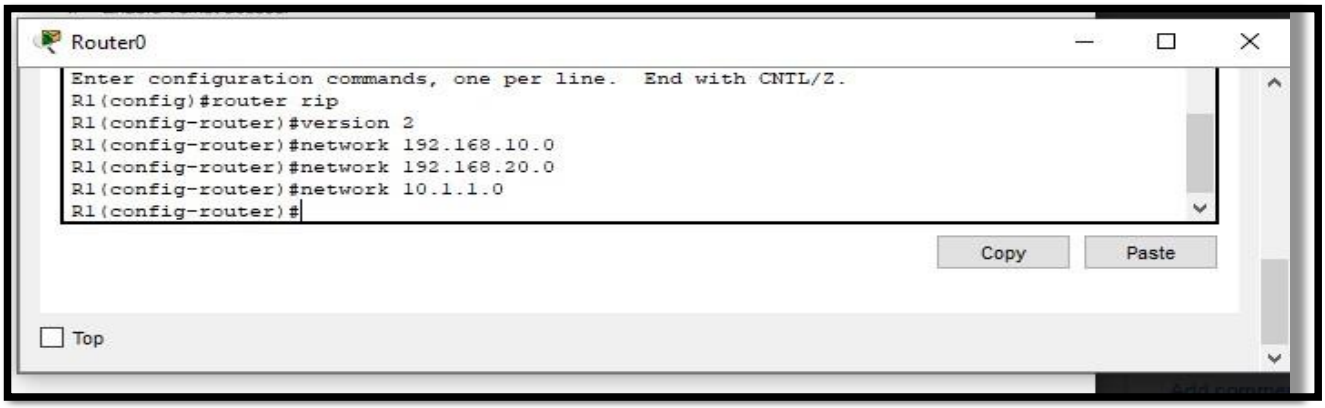




Configure Rip routing on R1, ISP, and R3.

a. Configure RIP version 2 and advertise all networks on R1, ISP, and R3. The OSPF configuration for R1 and ISP is included for reference.

R1(config)# **router rip**
R1(config-router)# **version 2**
R1(config-router)# **network 192.168.10.0**
R1(config-router)# **network 192.168.20.0**
R1(config-router)# **network 10.1.1.0**



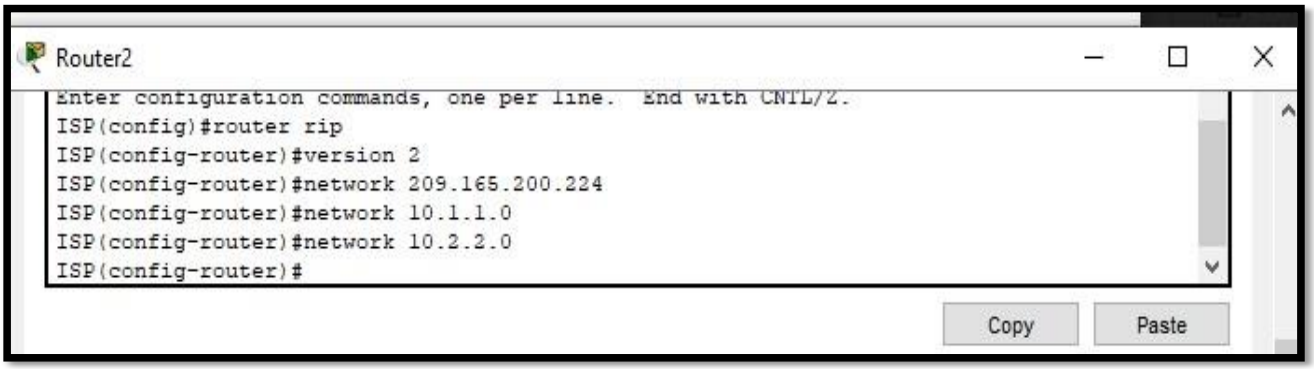
ISP(config)# **router rip**

ISP(config-router)# **version 2**

ISP(config-router)# **network 209.165.200.224**

ISP(config-router)# **network 10.1.1.0**

ISP(config -router)# **network 10.2.2.0**



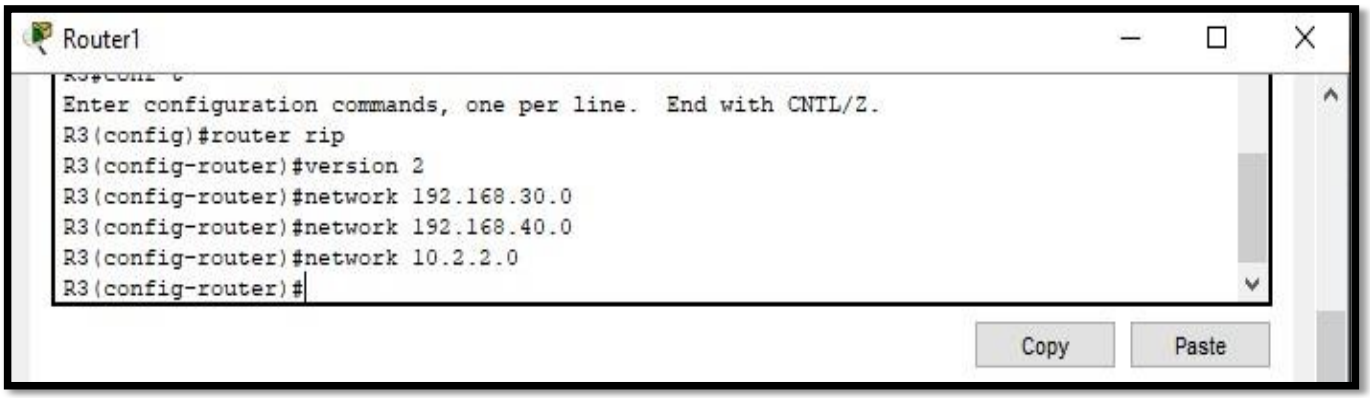
R3(config)# **router RIP**

R3(config-router)# **version 2**

R3(config-router)# **network 192.168.30.0**

R3(config-router)# **network 192.168.40.0**

R3(config-router)# **network 10.2.2.0**



Verify connectivity between devices.

Note: It is very important to test whether connectivity is working **before** you configure and apply access lists! You want to ensure that your network is properly functioning before you start to filter traffic.

a. From PC-A, ping PC-C and the loopback interface on R3. Were your pings successful? Yes

b. From R1, ping PC-C and the loopback interface on R3. Were your pings successful? Yes

c. From PC-C, ping PC-A and the loopback interface on R1. Were your pings successful? Yes

d. From R3, ping PC-A and the loopback interface on R1. Were your pings successful? Yes

Part 3: Configure and Verify Standard Numbered and Named ACLs

Step 1:- Configure a numbered standard ACL. Standard ACLs filter traffic based on the source IP address only. A typical best practice for standard ACLs is to configure and apply it as close to the destination as possible. For the first access list, create a standard numbered ACL that allows traffic from all hosts on the 192.168.10.0/24 network and all hosts on the 192.168.20.0/24 network to access all hosts on the 192.168.30.0/24 network. The security policy also states that a deny any access control entry (ACE), also referred to as an ACL statement, should be present at the end of all ACLs.

G0/1. The ACL should be applied going out. Students may answer with placing the ACL on the S0/0/1 interface on R3 going in. Emphasize to them that this would effectively block the LANs on R1 from getting to the 192.168.40.0/24 network as well!

a. **Configure the ACL on R3. Use 1 for the access list number.**

R3(config)# access-list 1 remark Allow R1 LANs Access

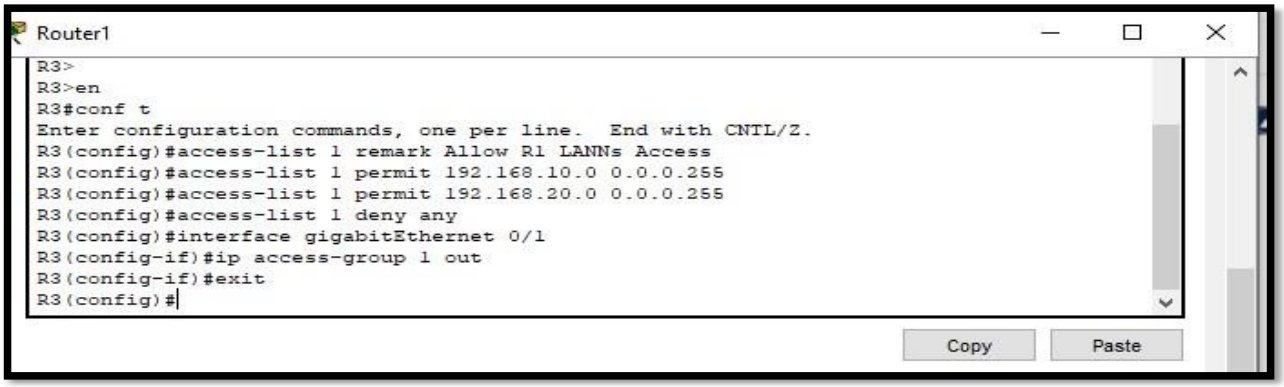
R3(config)# access-list 1 permit 192.168.10.0 0.0.0.255

R3(config)# access-list 1 permit 192.168.20.0 0.0.0.255

R3(config)# access-list 1 deny any

b. **Apply the ACL to the appropriate interface in the proper direction. R3(config)# interface g0/1**

R3(config -if)# ip access -group 1 out



c. **Verify a numbered ACL.**

The use of various show commands can aid you in verifying both the syntax and placement of your ACLs in your router.

1) On R3, issue the show access-lists 1 command.

R3# show access-lists 1

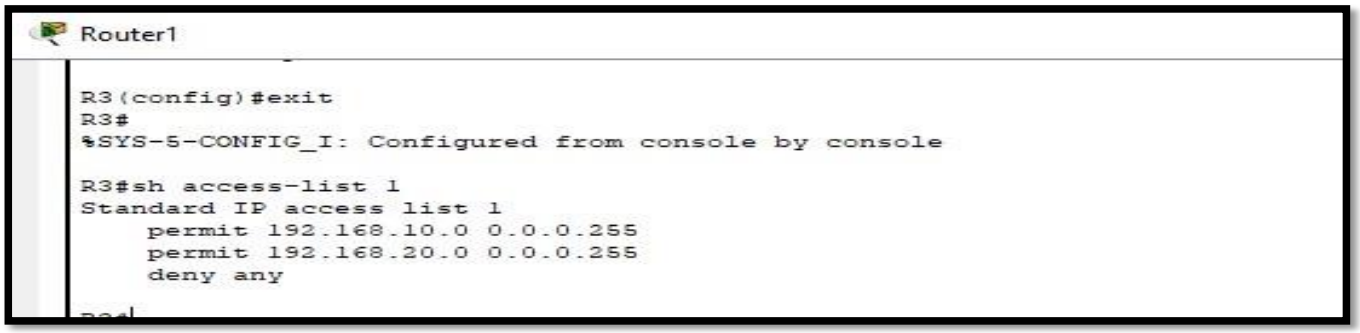
2) On R3, issue the show ip interface g0/1 command.

R3# show ip interface g0/1

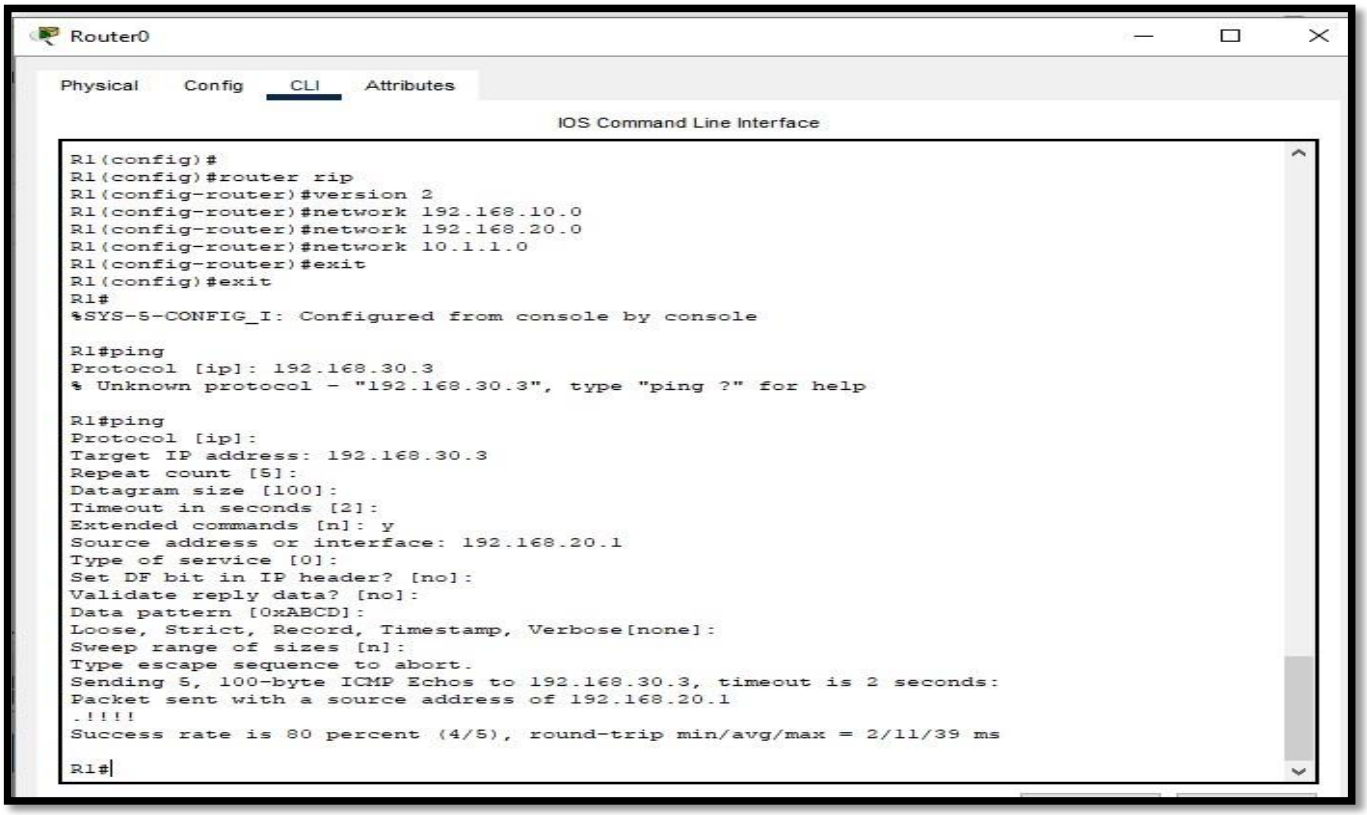
3) Test the ACL to see if it allows traffic from the 192.168.10.0/24 network access to the 192.168.30.0/24 network. From the PC-A command prompt, ping the PC-C IP address. Were the pings successful? Yes

4) Test the ACL to see if it allows traffic from the 192.168.20.0/24 network access to the 192.168.30.0/24 network. You must do an extended ping and use the loopback 0 address on

R1 as your source. Ping PC-C's IP address. Were the pings successful?_____Yes

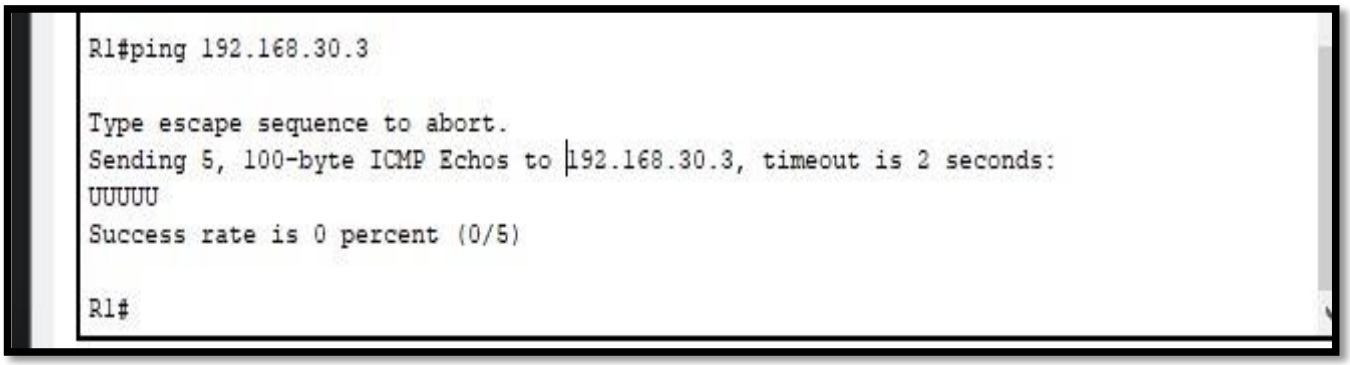


Ping :



d. **From the R1 prompt, ping PC-C's IP address again.**

R1# ping 192.168.30.3



No, the pings failed. When you ping from the router, it uses the closest interface to the destination as its source address. The pings had a source address of 10.1.1.1. The access list on R3 only allows the 192.168.10.0/24 and the 192.168.20.0/24 networks access.

Configure a named standard ACL.

Create a named standard ACL that conforms to the following policy: allow traffic from all hosts on the 192.168.40.0/24 network access to all hosts on the 192.168.10.0/24 network. Also, only allow host PC-C access to the 192.168.10.0/24 network. The name of this access list should be called BRANCH-OFFICE- POLICY.

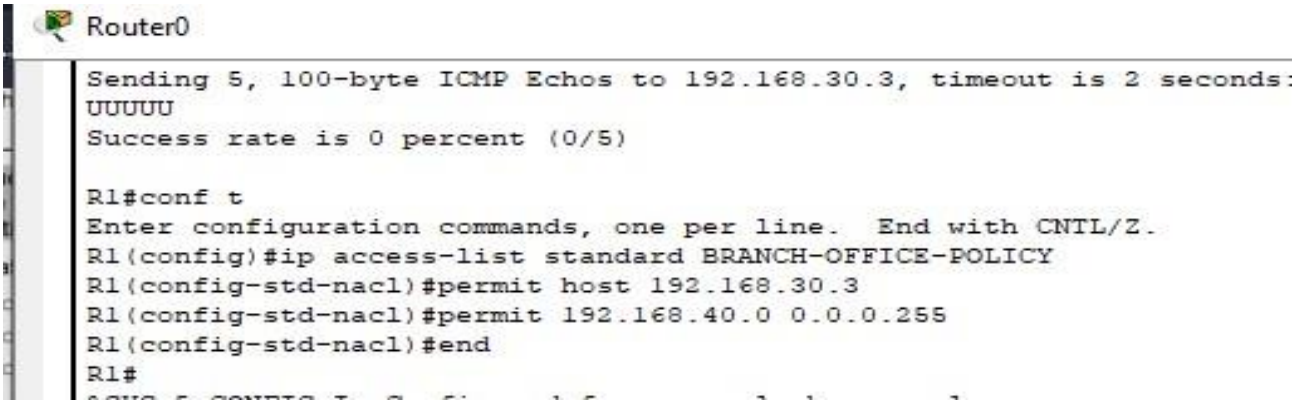
G0/1. The ACL should be applied going out. Students may answer with placing the ACL on the S0/0/0 interface on R1 going in. Emphasize to them that this would effectively block all traffic from the LANs on R3 from getting to the 192.168.20.0/24 network. a. Create the standard named ACL BRANCH-OFFICE-POLICY on R1.

R1(config)# ip access-list standard BRANCH-OFFICE-POLICY

R1(config-std-nacl)# permit host 192.168.30.3

R1(config-std-nacl)# permit 192.168.40.0 0.0.0.255

R1(config-std-nacl)# end



b. Apply the ACL to the appropriate interface in the proper direction.

R1# config t

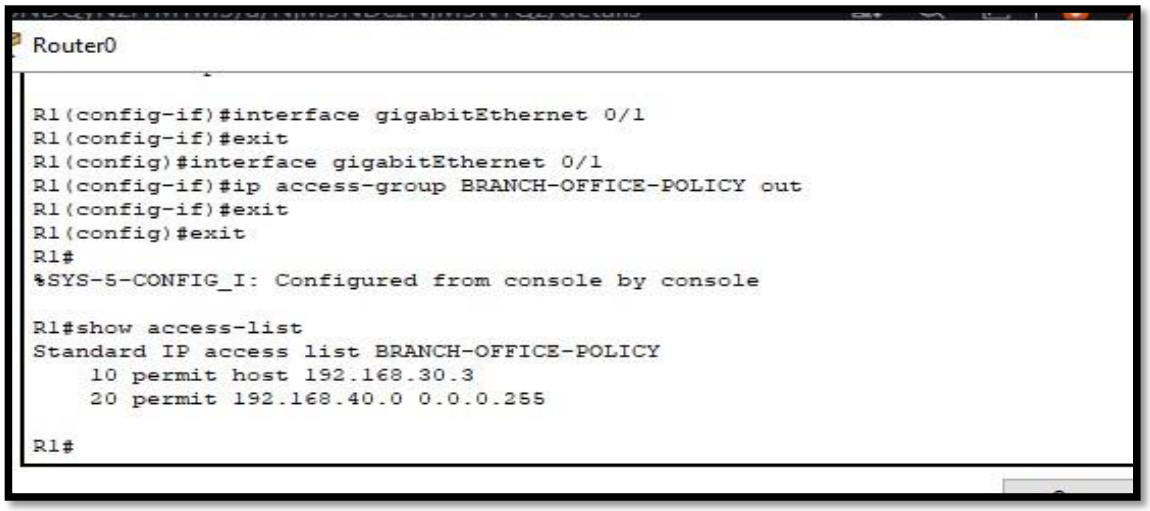
R1(config)# interface g0/1

R1(config-if)# ip access-group BRANCH-OFFICE-POLICY out c. Verify a

named ACL.

1) On R1, issue the show access-lists command.

R1# show access-lists



Although there is no line 30 with a deny any on R1, it is implied. You may wish to emphasize this to your students. Having them actually configure the deny any ACE is a good practice and reinforces the concept as it shows up in the ACL when issuing a show access-lists command. It is easy to forget the implicit deny any when troubleshooting ACLs. This could easily result in traffic being denied that should have been allowed.

2) **On R1, issue the show ip interface g0/1 command.**

R1# show ip interface g0/1

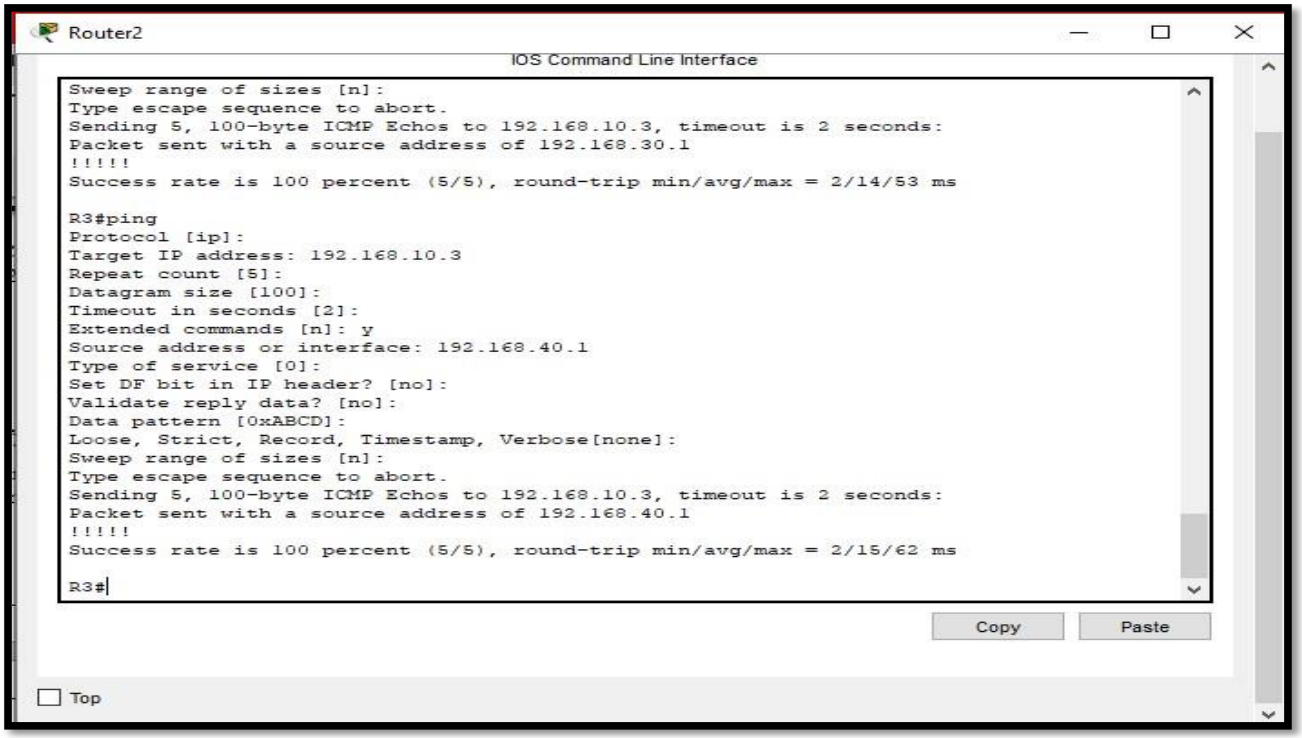
3) Test the ACL. From the command prompt on PC-C, ping PC-A's IP address. Were the pings successful? _____Yes

4) Test the ACL to ensure that only the PC-C host is allowed access to the 192.168.10.0/24 network. You must do an extended ping and use the G0/1 address on R3 as your source. Ping PC-A's IP address. Were the pings successful? No

R3# ping

5) Test the ACL to see if it allows traffic from the 192.168.40.0/24 network access to the 192.168.10.0/24 network. You must perform an extended ping and use the loopback 0 address on R3 as your source. Ping PC-A's IP address. Were the pings successful? _ Yes

R3# ping



Part 4: Modify a Standard ACL

It is common in business for security policies to change. For this reason, ACLs may need to be modified. In Part 4, you will change one of the previous ACLs you configured to match a new management policy being put in place. Management has decided that users from the 209.165.200.224/27 network should be allowed full access to the 192.168.10.0/24 network.

Management also wants ACLs on all of their routers to follow consistent rules. A deny any

ACE should be placed at the end of all ACLs. You must modify the BRANCH-OFFICE- POLICY ACL.

Step 1 :- Modify a named standard ACL.

a. From R1 privileged EXEC mode, issue a show access-lists command.

R1# show access-lists

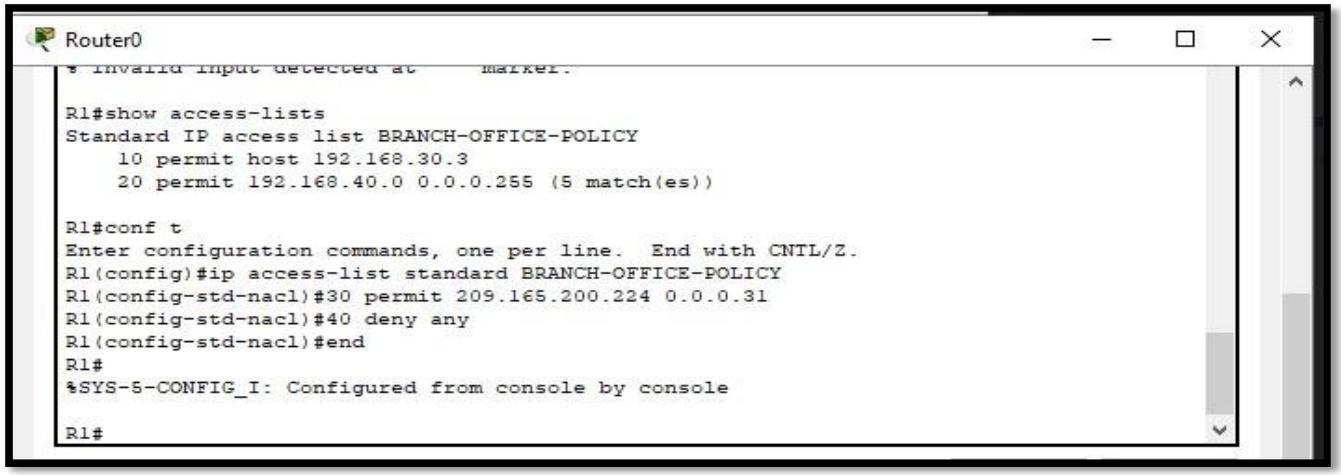
b. Add two additional lines at the end of the ACL. From global config mode, modify the ACL, BRANCH- OFFICE-POLICY.

R1#(config)# ip access-list standard BRANCH-OFFICE-POLICY

R1(config-std-nacl)# 30 permit 209.165.200.224 0.0.0.31

R1(config-std-nacl)# 40 deny any

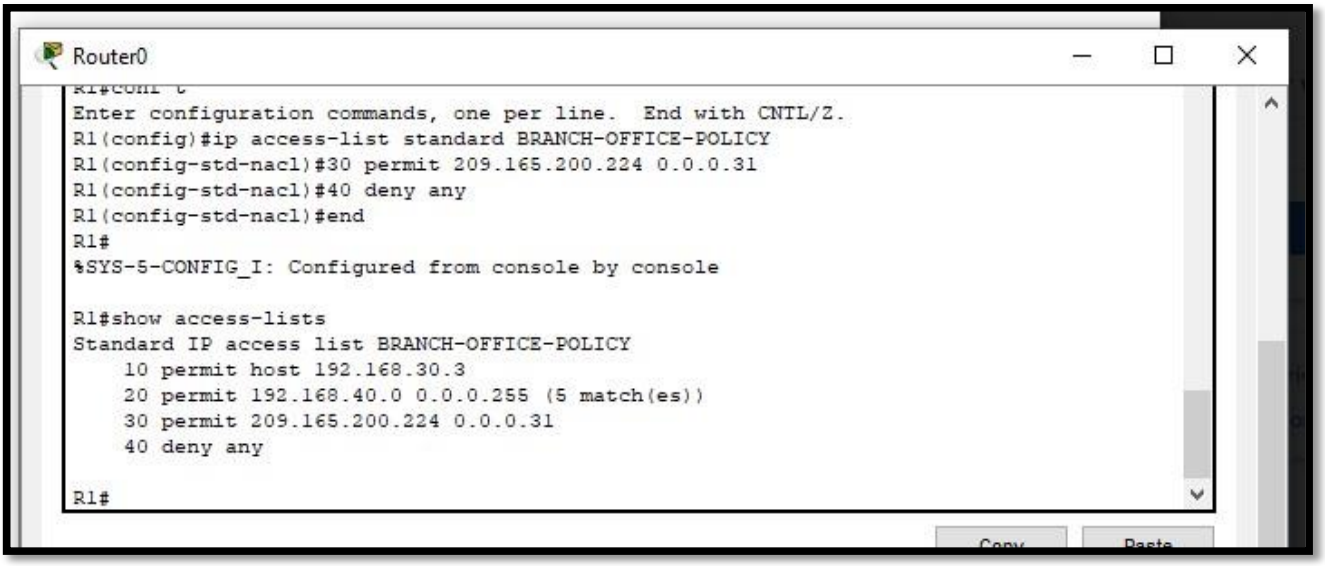
R1(config-std-nacl)# end



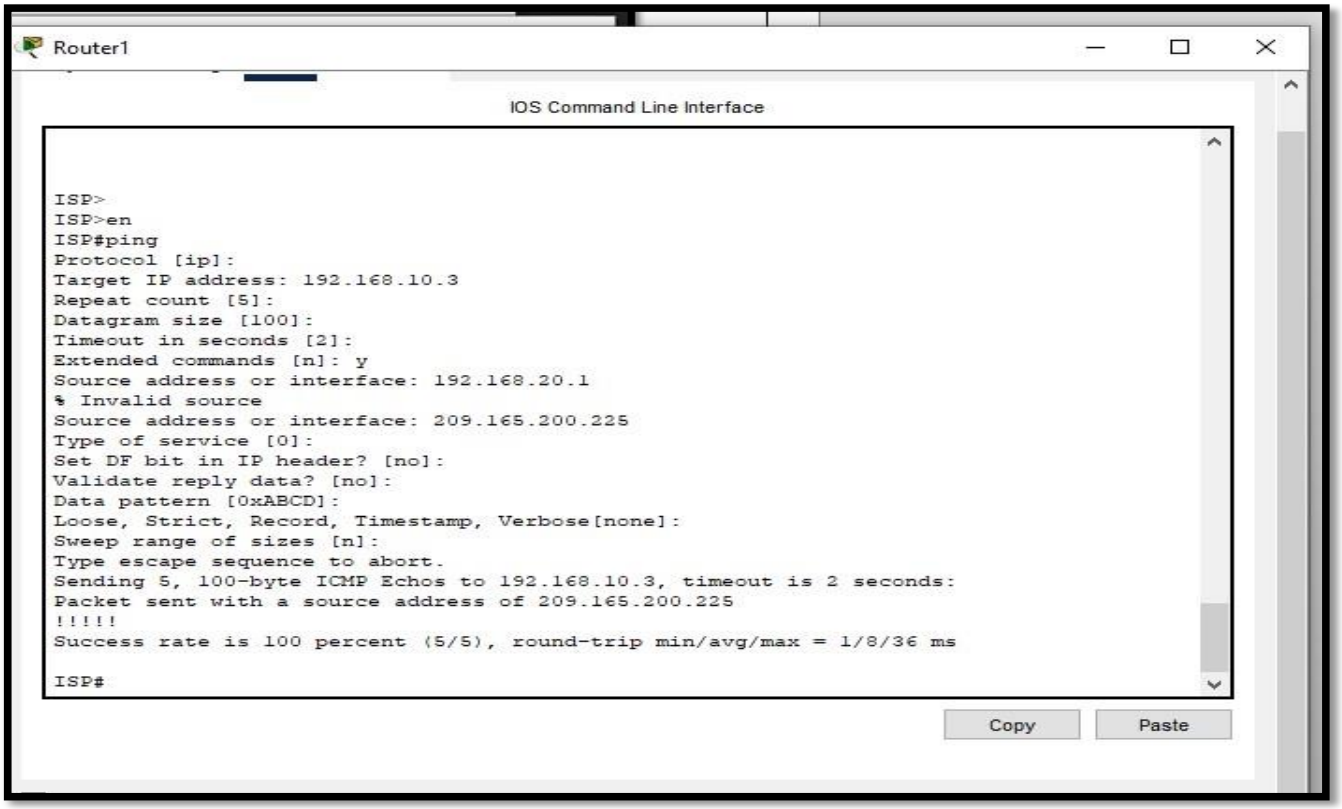
c. Verify the ACL.

1) On R1, issue the show access-lists command.

R1# show access -lists



2) From the ISP command prompt, issue an extended ping. Test the ACL to see if it allows traffic from the 209.165.200.224/27 network access to the 192.168.10.0/24 network. You must do an extended ping and use the loopback 0 address on ISP as your source. Ping PC-A's IP address. Were the pings successful? _____Yes



Router Interface Summary Table

Router Interface Summary				
Router Model	**Ethernet Interface #1**	**Ethernet Interface #2**	**Serial Interface #1**	**Serial Interface #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

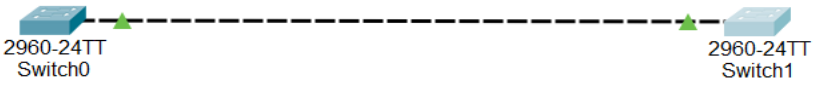
Practical 3

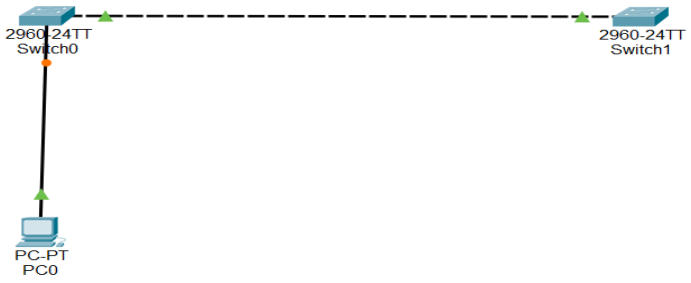
Aim : Implement SPAN Technologies (Switch Port Analyzer)

What is SPAN?

→ Cisco developed the Switched Port Analyzer (SPAN) feature to facilitate the capturing of packets. SPAN is supported on most Cisco switch platforms. SPAN works by copying the traffic from one or more source ports. The copy is then sent out a SPAN destination port. The destination port will often be connected to a host running packet analyzing software, such as Wireshark. Because SPAN only makes a copy of traffic, the source traffic is never affected. SPAN is an out-of-band process. In addition to troubleshooting network issues and performance, SPAN is useful for intrusion detection systems (IDS) and application monitoring platforms. SPAN is often referred to as port mirroring

Steps 1 : Build and Configure network





Step 2 : Configure Switch and vlan and add vlan port on s1

```
Switch>
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname S1
S1(config)#vlan 10
S1(config-vlan)#name HR
S1(config-vlan)#vlan 20
S1(config-vlan)#name Accounts
S1(config-vlan)#vlan 30
S1(config-vlan)#name Developer
S1(config-vlan)#end
S1#
%SYS-5-CONFIG_I: Configured from console by console

S1#vlan brief
     ^
% Invalid input detected at '^' marker.

S1#sh vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                 Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                 Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                 Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                 Gig0/1, Gig0/2
10   HR                                active
20   Accounts                          active
30   Developer                         active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
S1#
```

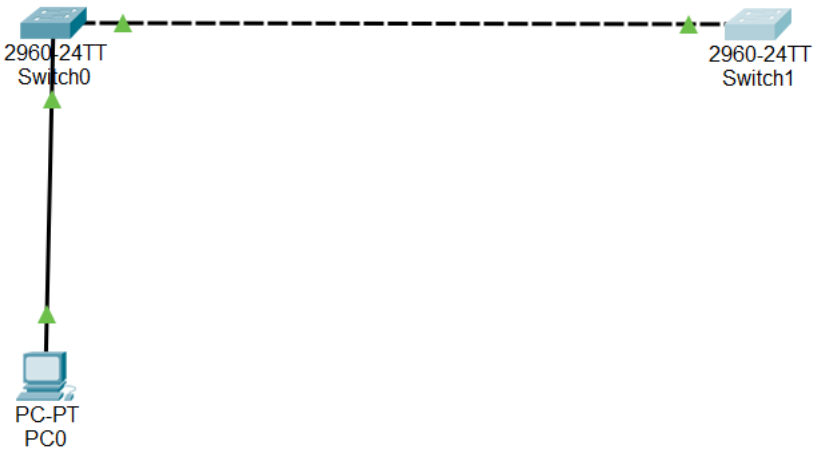
```
S1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#interfa
S1(config)#interface range f0/1-4
S1(config-if-range)#switch
S1(config-if-range)#switchport mode
S1(config-if-range)#switchport mode ?
  access    Set trunking mode to ACCESS unconditionally
  dynamic   Set trunking mode to dynamically negotiate access or trunk mode
  trunk     Set trunking mode to TRUNK unconditionally
S1(config-if-range)#switchport mode access ?
  <cr>
S1(config-if-range)#switchport mode access vlan 10
                                           ^
% Invalid input detected at '^' marker.

S1(config-if-range)#switchport mode access
S1(config-if-range)#switchport mode access vlan 10
                                           ^
% Invalid input detected at '^' marker.

S1(config-if-range)#switchport access vlan 10
S1(config-if-range)#inter
S1(config-if-range)#interfac
S1(config-if-range)#interface range f0/11-24
S1(config-if-range)#switchport mode access
S1(config-if-range)#switchport access vlan 20
S1(config-if-range)#interface range f0/5-0
                   ^
% Invalid input detected at '^' marker.

S1(config-if-range)#exit
```

```
% Invalid input detected at     marker.

S1(config)#interface range f0/5-6
S1(config-if-range)#switchport mode access
S1(config-if-range)#switchport access vlan 30
S1(config-if-range)#exit
S1(config)#exit
S1#
%SYS-5-CONFIG_I: Configured from console by console

S1#sh vlan b

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                 Gig0/1, Gig0/2
10   HR                                active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
20   Accounts                          active    Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                 Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                 Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                 Fa0/23, Fa0/24
30   Developer                         active    Fa0/5, Fa0/6
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
S1#
```

Step 3 : Configure SPAN :

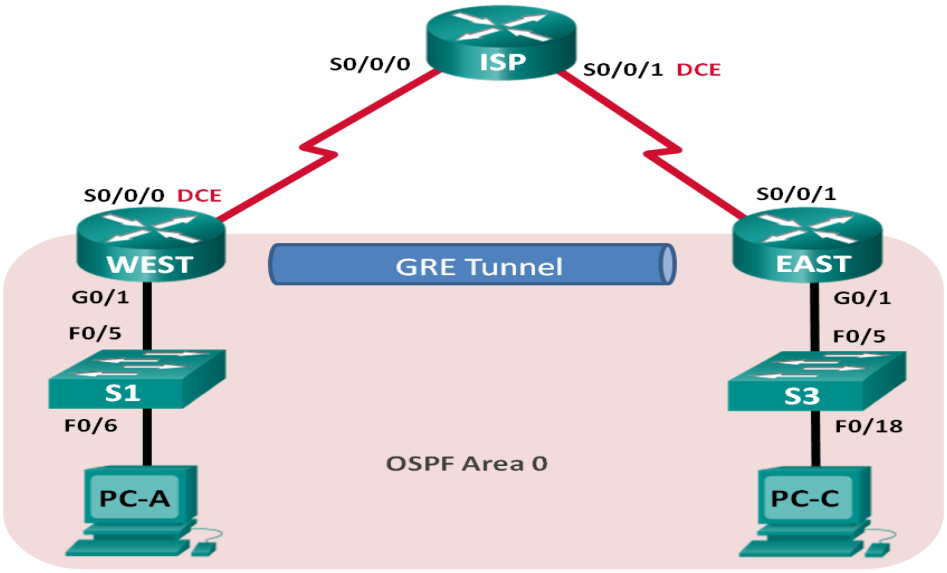
```
Switch#sh run
Building configuration...

Current configuration : 1158 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Switch
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
```

```
Switch#sh mon
Session 1
---------
Type                      : Local Session
Description               : -
Source VLANs              :
     Both                 : 20
Destination Ports         : Gig0/2
     Encapsulation        : Native
           Ingress        : Disabled


Switch#
Switch#sh mon det
Session 1
---------
Type                      : Local Session
Description               : -
Source Ports              :
     RX Only              : None
     TX Only              : None
     Both                 : None
Source VLANs              :
     RX Only              : None
     TX Only              : None
     Both                 : 20
Source RSPAN VLAN         : None
Destination Ports         : Gig0/2
     Encapsulation        : Native
           Ingress        : Disabled
Filter VLANs              : None
Dest RSPAN VLAN           : None
```



Practical 4

1. Aim : Configuring a Point-to-Point GRE VPN Tunnel



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
WEST	G0/1	172.16.1.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A
	Tunnel0	172.16.12.1	255.255.255.252	N/A
ISP	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A
EAST	G0/1	172.16.2.1	255.255.255.0	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
	Tunnel0	172.16.12.2	255.255.255.252	N/A
PC-A	NIC	172.16.1.3	255.255.255.0	172.16.1.1
PC-C	NIC	172.16.2.3	255.255.255.0	172.16.2.1

What is GRE?

→ Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco that can encapsulate a wide variety of network layer protocols into point-to-point connections. It allows the creation of a virtual point-to-point link between two network nodes over an existing network, such as the Internet. GRE is often used in conjunction with other protocols, such as IPsec, to provide a secure and encrypted communication channel.

Here are some key features and aspects of GRE:

1. Tunneling: GRE encapsulates a payload (which could be of any network layer protocol) within a GRE header. This encapsulated packet is then transmitted over an existing network.
2. Compatibility: GRE is protocol-independent, which means it can carry a variety of network layer protocols, including IP, IPX, and even non-IP protocols.
3. Routing: GRE does not have inherent encryption or security features. However, it is often used with IPsec to secure the data being transmitted through the tunnel. GRE itself focuses on providing a virtual point-to-point connection and encapsulation.
4. Dynamic Routing: GRE can carry multicast traffic and supports dynamic routing protocols, allowing it to participate in routing decisions.
5. Point-to-Point Connections: GRE is typically used to create point-to-point connections between two network nodes. It is not designed for point-to-multipoint or multipoint-to-multipoint connections.
6. Header Format: The GRE header includes information such as the key, which can be used for simple packet filtering, and the protocol type of the payload being carried.

While GRE has been widely used in various networking scenarios, it's worth noting that newer technologies, such as IPsec, have gained popularity for secure tunneling due to their built-in encryption features. Nonetheless, GRE remains relevant in certain contexts, especially when used in conjunction with other protocols to fulfill specific networking requirements.

Objectives

Part 1: Configure Basic Device Settings

Part 2: Configure a GRE Tunnel

Part 3: Enable Routing over the GRE Tunnel

Background / Scenario

Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a variety of network layer protocols between two locations over a public network, such as the Internet.

GRE can be used with:

- Connecting IPv6 networks over IPv4 networks
- Multicast packets, such as OSPF, EIGRP, and streaming applications

In this lab, you will configure an unencrypted point-to-point GRE VPN tunnel and verify that network traffic is using the tunnel. You will also configure the OSPF routing protocol inside the GRE VPN tunnel. The GRE tunnel is between the WEST and EAST routers in OSPF area 0. The ISP has no knowledge of the GRE tunnel. Communication between the WEST and EAST routers and the ISP is accomplished using default static routes.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

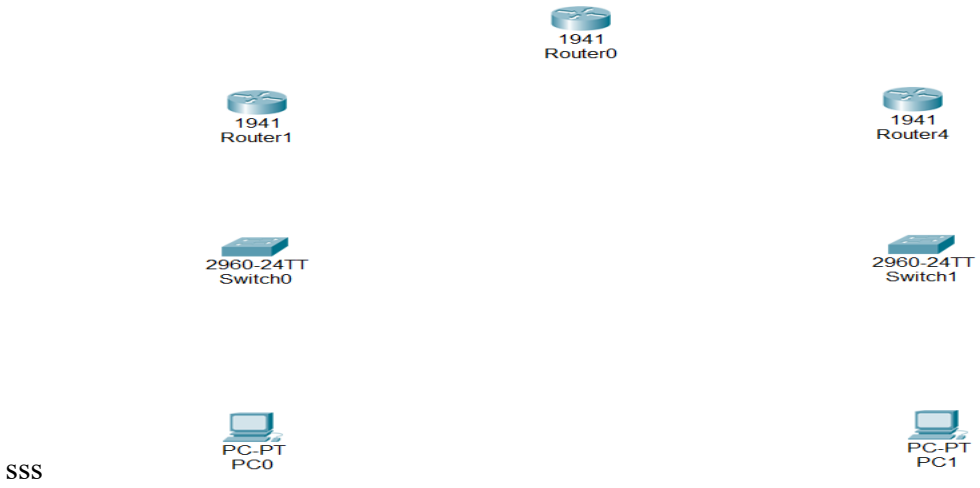
Required Resources

- 3 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)

- 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

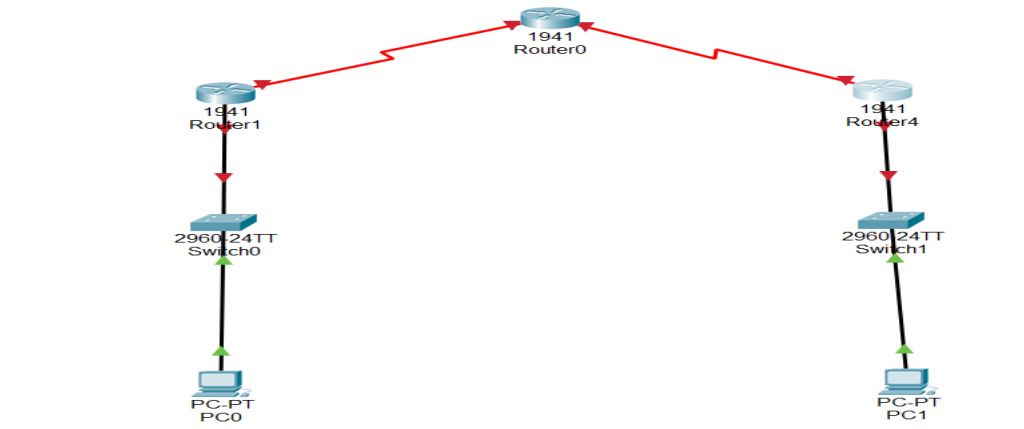
Part 1: Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic router settings, such as the interface IP addresses, routing, device access, and passwords.



sss

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



Step 2: Initialize and reload the routers and switches.

Step 3: Configure basic settings for each router.

a. Disable DNS lookup.

b. Configure the device names.

c. Apply IP addresses to Serial and Gigabit Ethernet interfaces according to the Addressing Table and activate the physical interfaces. Do NOT configure the Tunnel0 interfaces at this time.

d. Set the clock rate to **128000** for DCE serial interfaces.

West Router :

```
Router#
Router#
Router#
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface GigabitEthernet 0/1
Router(config-if)#ip address 172.16.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Ser
Router(config)#interface Serial 0/0/0
Router(config-if)#ip address 10.1.1.1 255.255.255.252
Router(config-if)#no sh

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
Router(config-if)#
Router(config-if)#clock
Router(config-if)#clock rate 128000
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

East Router :

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter
Router(config)#interface Giga
Router(config)#interface GigabitEthernet 0/1
Router(config-if)#ip
Router(config-if)#ip ad
Router(config-if)#ip address 172.16.2.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Router(config-if)#exit
Router(config)#interface se
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 10.2.2.1 255.255.255.252
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/1, changed state to down
Router(config-if)#
Router(config-if)#sexit
                  ^
% Invalid input detected at '^' marker.

Router(config-if)#
```

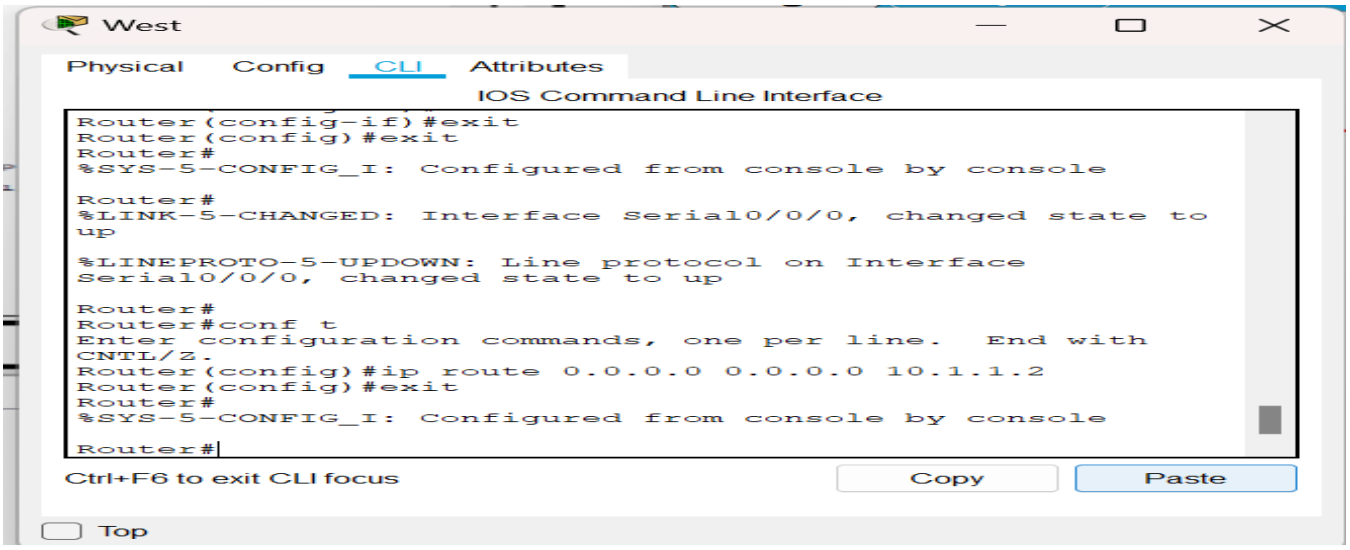
ISP Router :

```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inte
Router(config)#interface Ser
Router(config)#interface Serial 0/0/0
Router(config-if)#ip a
Router(config-if)#ip add
Router(config-if)#ip address 10.1.1.2 255.255.255.252
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial 0/0/1
Router(config-if)#interface Serial 0/0/1
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0
Router(config-if)#ip address 10.2.2.2 255.255.255.252
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/1, changed state to up

Router(config-if)#exit
```

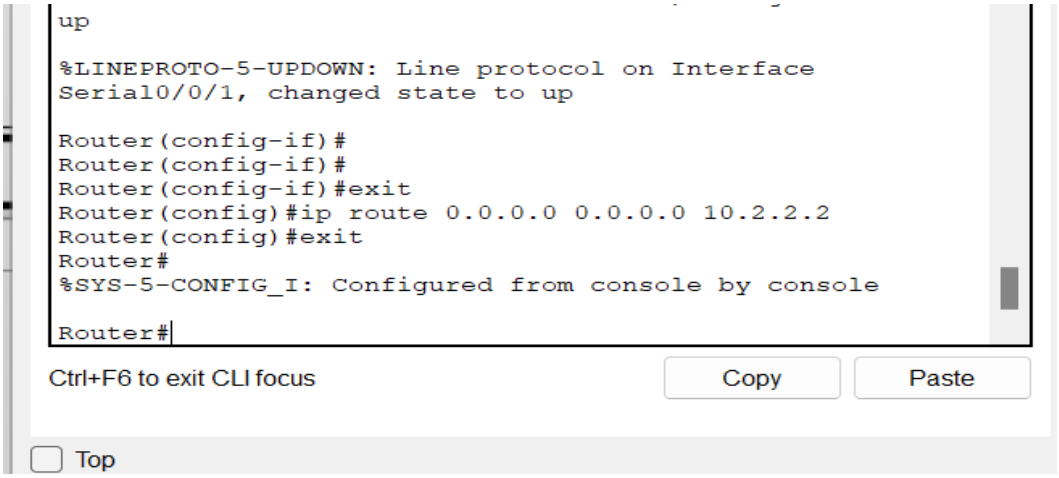
```
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial 0/0/1
Router(config-if)#interface Serial 0/0/1
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0
Router(config-if)#ip address 10.2.2.2 255.255.255.252
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/1, changed state to up

Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/1, changed state to up

Router(config)#interface Serial 0/0/1
Router(config-if)#clock rate 128000
Router(config-if)#exit
Router(config)#
```

Step 4: Configure default routes to the ISP router.

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

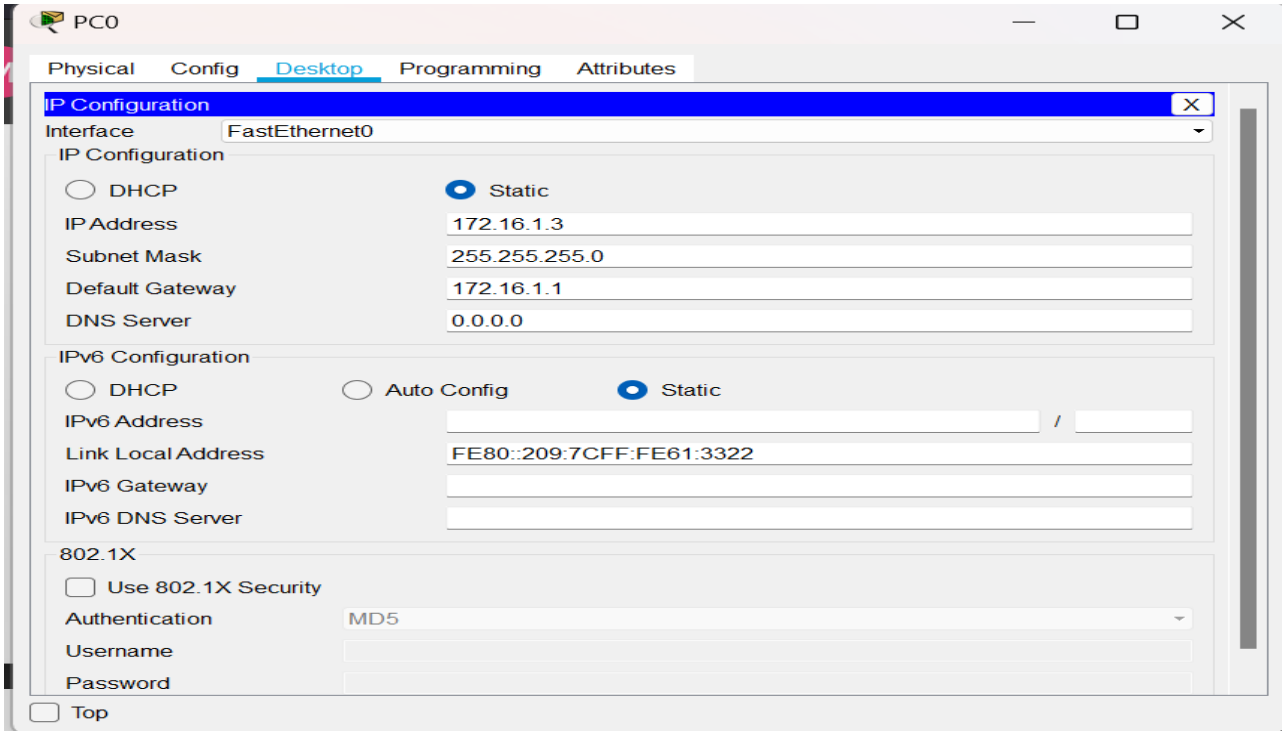


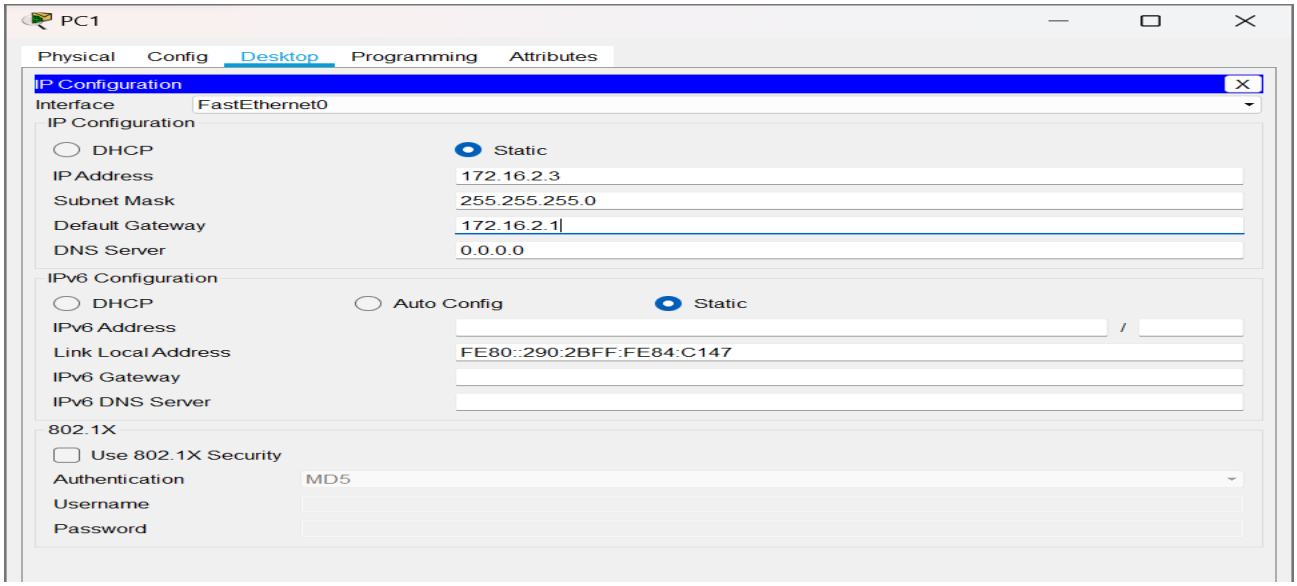
EAST(config)# **ip route 0.0.0.0 0.0.0.0 10.2.2.2**



Step 5: Configure the PCs.

Assign IP addresses and default gateways to the PCs according to the Addressing Table.





Step 6: Verify connectivity.

Step 7: Save your running configuration.

Part 2: Configure a GRE Tunnel

In Part 2, you will configure a GRE tunnel between the WEST and EAST routers.

Step 1: Configure the GRE tunnel interface.

a. Configure the tunnel interface on the WEST router. Use S0/0/0 on WEST as the tunnel source interface and 10.2.2.1 as the tunnel destination on the EAST router.

WEST(config)# **interface tunnel 0**
WEST(config-if)# **ip address 172.16.12.1 255.255.255.252**
WEST(config-if)# **tunnel source s0/0/0**
WEST(config-if)# **tunnel destination 10.2.2.1**

```
Router#
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface tunnel 0

Router(config-if)#
%LINK-5-CHANGED: Interface Tunnel0, changed state to up

Router(config-if)#ip address 172.16.12.1 255.255.255.252
Router(config-if)#tun
Router(config-if)#tunnel so
Router(config-if)#tunnel source s0/0/0
Router(config-if)#tun
Router(config-if)#tunnel dest
Router(config-if)#tunnel destination 10.2.2.1
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

Router(config-if)#
```

b. Configure the tunnel interface on the EAST router. Use S0/0/1 on EAST as the tunnel source interface and 10.1.1.1 as the tunnel destination on the WEST router.

EAST(config)# **interface tunnel 0**

EAST(config-if)# **ip address 172.16.12.2 255.255.255.252**

EAST(config-if)# **tunnel source s0/0/1**

EAST(config-if)# **tunnel destination 10.1.1.1**

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface tunnel 0

Router(config-if)#
%LINK-5-CHANGED: Interface Tunnel0, changed state to up

Router(config-if)#ip address 172.16.12.2 255.255.255.252
Router(config-if)#tunnel source 10.2.2.1
                                ^
% Invalid input detected at '^' marker.

Router(config-if)#tunnel sou
Router(config-if)#tunnel source 10.2.2.1
                                ^
% Invalid input detected at '^' marker.

Router(config-if)#tunnel source s0/0/1
Router(config-if)#tunnel destination 10.1.1.1
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

Router(config-if)#exit
```

Part 3: Enable Routing over the GRE Tunnel

Step 1: Configure OSPF routing for area 0 over the tunnel.

a. Configure OSPF process ID 1 using area 0 on the WEST router for the 172.16.1.0/24 and 172.16.12.0/24 networks.

WEST(config)# **router ospf 1**

WEST(config-router)# **network 172.16.1.0 0.0.0.255 area 0**

WEST(config-router)# **network 172.16.12.0 0.0.0.3 area 0**

```
Router(config)#
Router(config)#router ospf 1
Router(config-router)#network 172.16.1.0 0.0.0.255 area 0
Router(config-router)#network 172.16.12.0 0.0.0.3 area 0
Router(config-router)#
```

b. Configure OSPF process ID 1 using area 0 on the EAST router for the 172.16.2.0/24 and 172.16.12.0/24 networks.

EAST(config)# **router ospf 1**

EAST(config-router)# **network 172.16.2.0 0.0.0.255 area 0**

EAST(config-router)# **network 172.16.12.0 0.0.0.3 area 0**

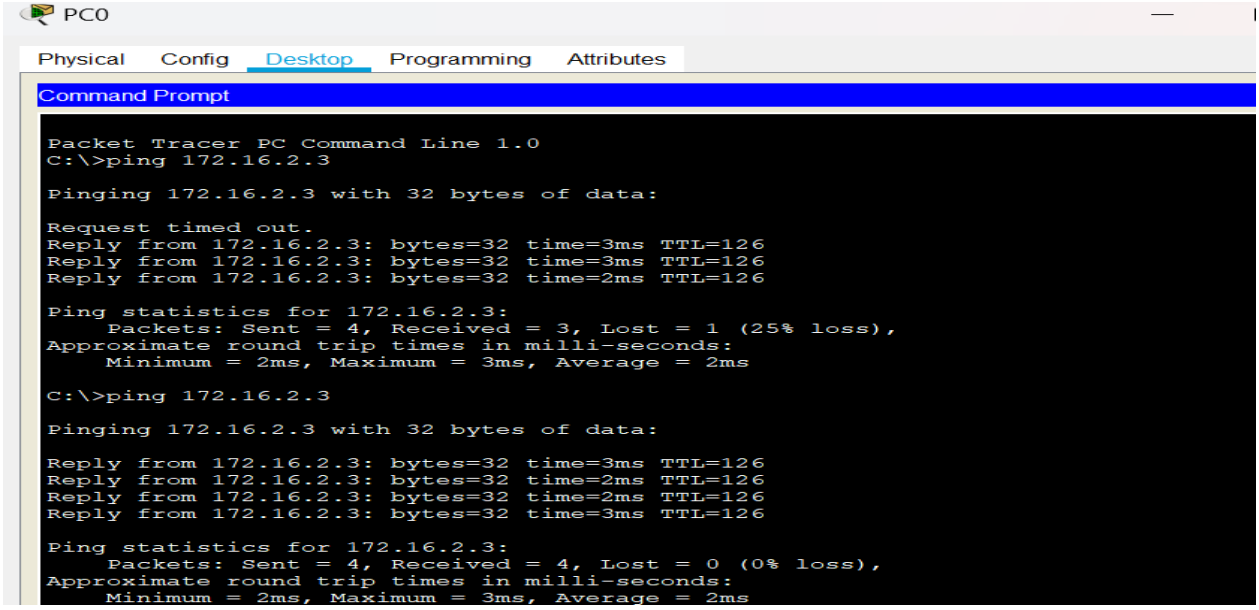
```
Router(config)#
Router(config)#router ospf 1
Router(config-router)#network 172.16.2.0 0.0.0.255 area 0
Router(config-router)#network 172.16.12.0 0.0.0.3 area 0
Router(config-router)#
```

Check Route :

```
Router#sh 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, E - EGP
       i - IS-IS, 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 10.1.1.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/30 is directly connected, Serial0/0/0
L       10.1.1.1/32 is directly connected, Serial0/0/0
     172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
C       172.16.1.0/24 is directly connected, GigabitEthernet0/1
L       172.16.1.1/32 is directly connected, GigabitEthernet0/1
O       172.16.2.0/24 [110/1001] via 172.16.12.2, 00:01:15, Tunnel0
C       172.16.12.0/30 is directly connected, Tunnel0
L       172.16.12.1/32 is directly connected, Tunnel0
S*   0.0.0.0/0 [1/0] via 10.1.1.2
```

Step 2: Verify end-to-end connectivity.

a. Ping from PC-A to PC-C. It should be successful. If not, troubleshoot until you have end-to-end connectivity.



```
Packet Tracer PC Command Line 1.0
C:\>ping 172.16.1.3

Pinging 172.16.1.3 with 32 bytes of data:

Reply from 172.16.1.3: bytes=32 time=2ms TTL=126
Reply from 172.16.1.3: bytes=32 time=3ms TTL=126
Reply from 172.16.1.3: bytes=32 time=4ms TTL=126
Reply from 172.16.1.3: bytes=32 time=2ms TTL=126

Ping statistics for 172.16.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 2ms
```

b. Traceroute from PC-A to PC-C. What is the path from PC-A to PC-C?

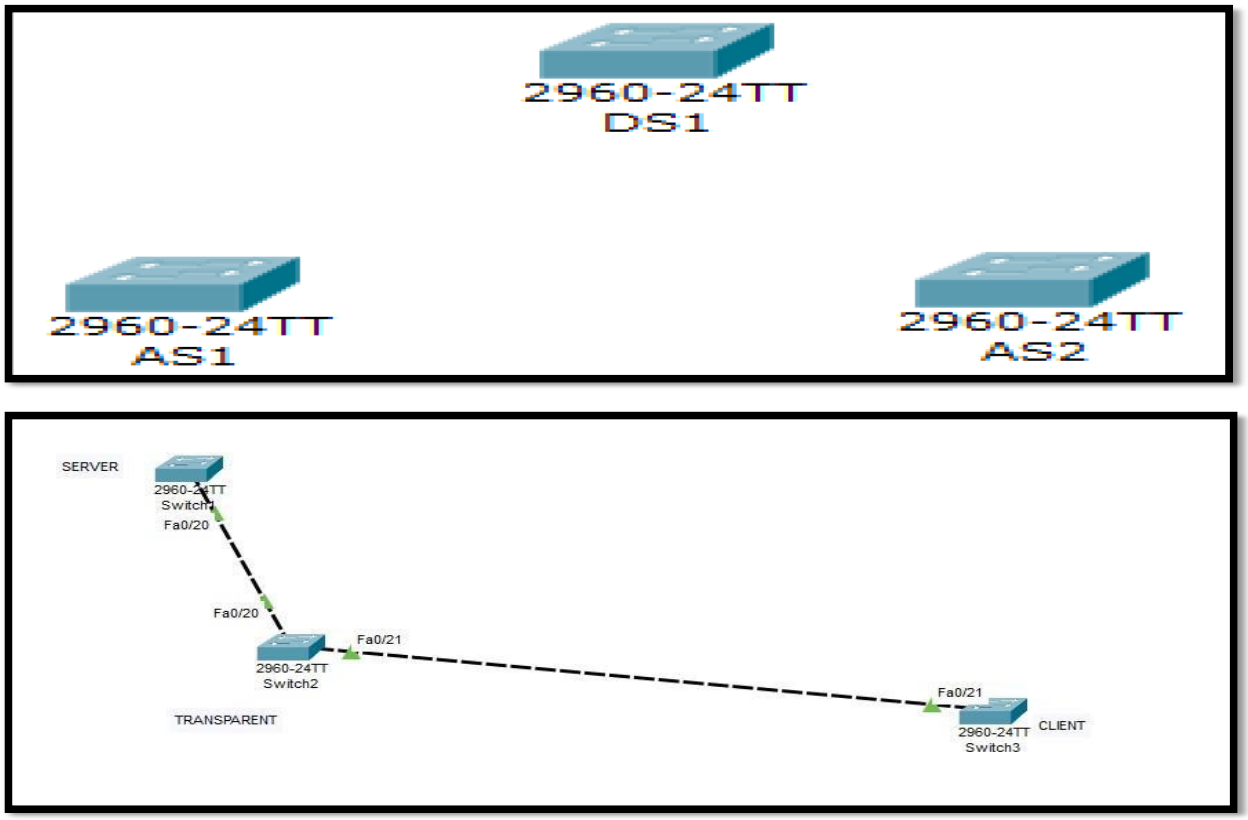
```
C:\>tracert 172.16.1.3

Tracing route to 172.16.1.3 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      172.16.2.1
  2    2 ms      2 ms      3 ms      172.16.12.1
  3    2 ms      2 ms      2 ms      172.16.1.3

Trace complete.

C:\>
```

Router Interface Summary Table

Router Interface Summary				
Router Model	**Ethernet Interface #1**	**Ethernet Interface #2**	**Serial Interface #1**	**Serial Interface #2**
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

2. **Aim : Implement VTP**

Topology



What is VTP?

VTP is a Layer 2 messaging protocol that was designed to manage the creation and deletion of VLANs and maintain network-wide VLAN database consistency. Using this protocol, a network administrator can add or delete VLANs and have those changes automatically propagated to all other switches in the network. Without VTP, switches do not exchange information about VLANs. The protocol has been designed around the centralized management idea. One or more switches are assigned the role of VTP Server. Any updates made on these switches are sent through VTP to the other switches, which are VTP Clients

What are the modes of VTP in networking?

There are three modes of VTP in networking, namely-

- VTP Server Mode
- VTP Client Mode
- VTP Transparent Mode

Objectives

In this activity, you will configure VTP

Background / Scenario

Scalability and management are two crucial considerations in large networks. VTP and DTP are technologies that improve management and scalability. VLAN Trunking Protocol (VTP) allows the switches to communicate over VLANs automatically, improving management and scalability. Dynamic Trunking Protocol (DTP) allows the switches to automatically negotiate and establish trunk links. DTP also improves scalability.

Step 1: Using Dynamic Trunk Protocol (DTP) to form trunk links

Access links transport single VLAN frames and trunk links carry frames belonging to multiple VLANs. While trunk links can be manually configured, DTP can be used to allow the switches to negotiate and establish trunk links automatically. DTP is very helpful in large networks

Configure Switches:- show vtp status en conf t hostname s1 int f0/20 switchport mode trunk

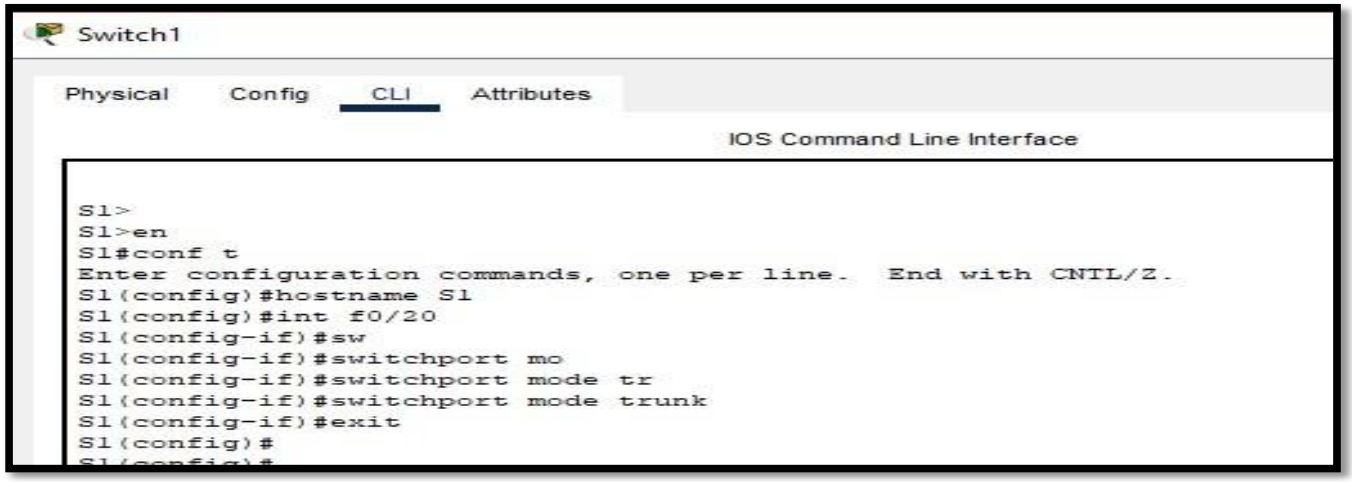
exit

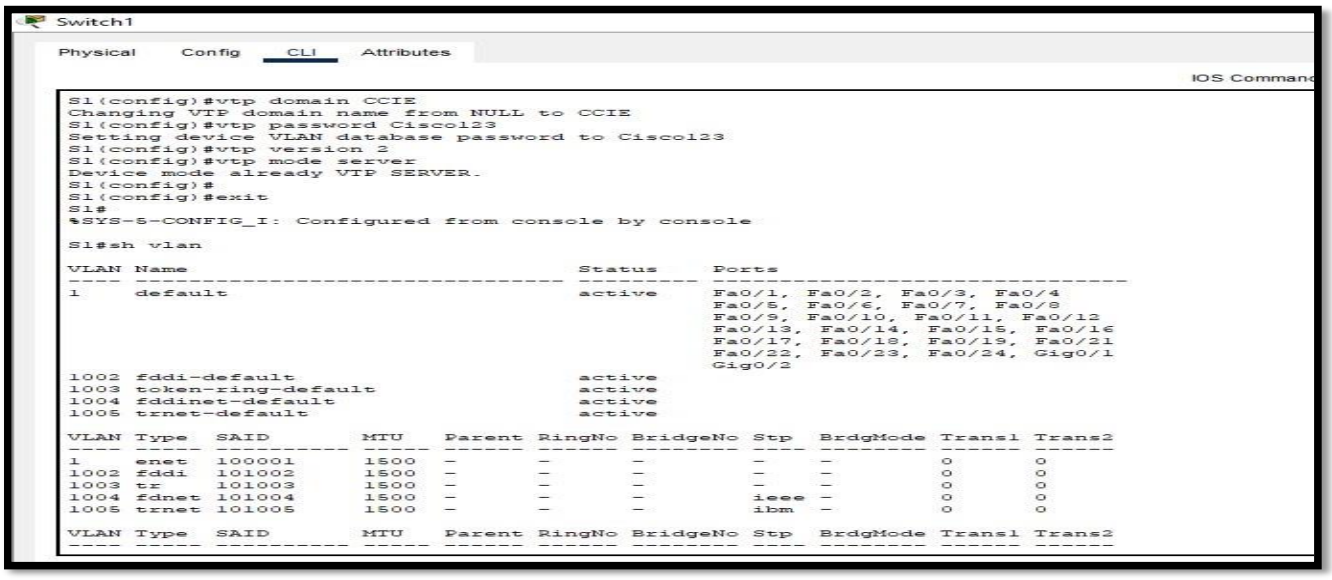
vtp domain CCIE vtp password

Cisco123 vtp version 2 vtp mode

server

exit





en conf t hostname s2 int f0/20-21

switchport mode trunk exit

vtp domain CCIE vtp password

Cisco123 vtp version 2 vtp mode server

exit

en

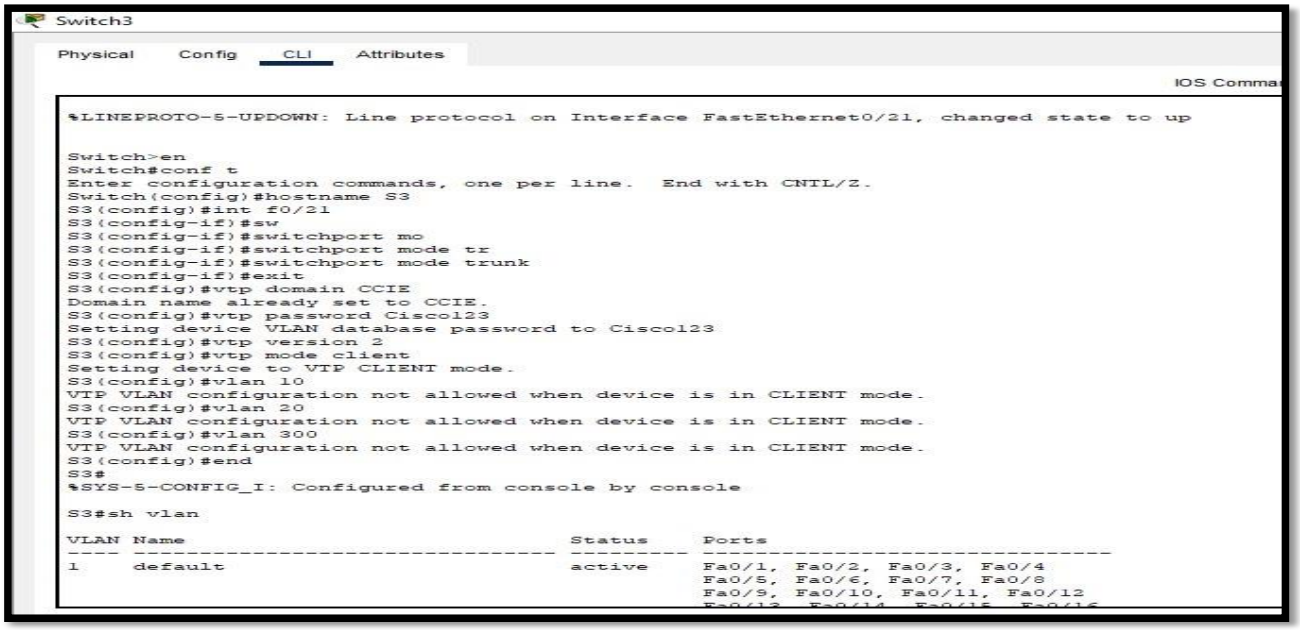
conf t hostname s3 int f0/21 switchport

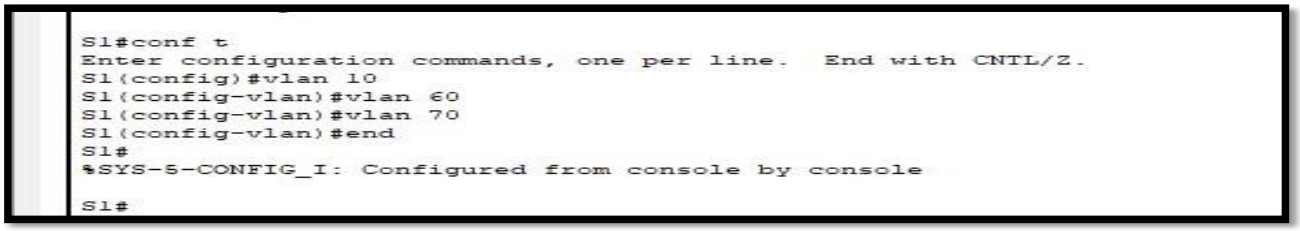
mode trunk exit

vtp domain CCIE vtp password

Cisco123 vtp version 2 vtp mode server

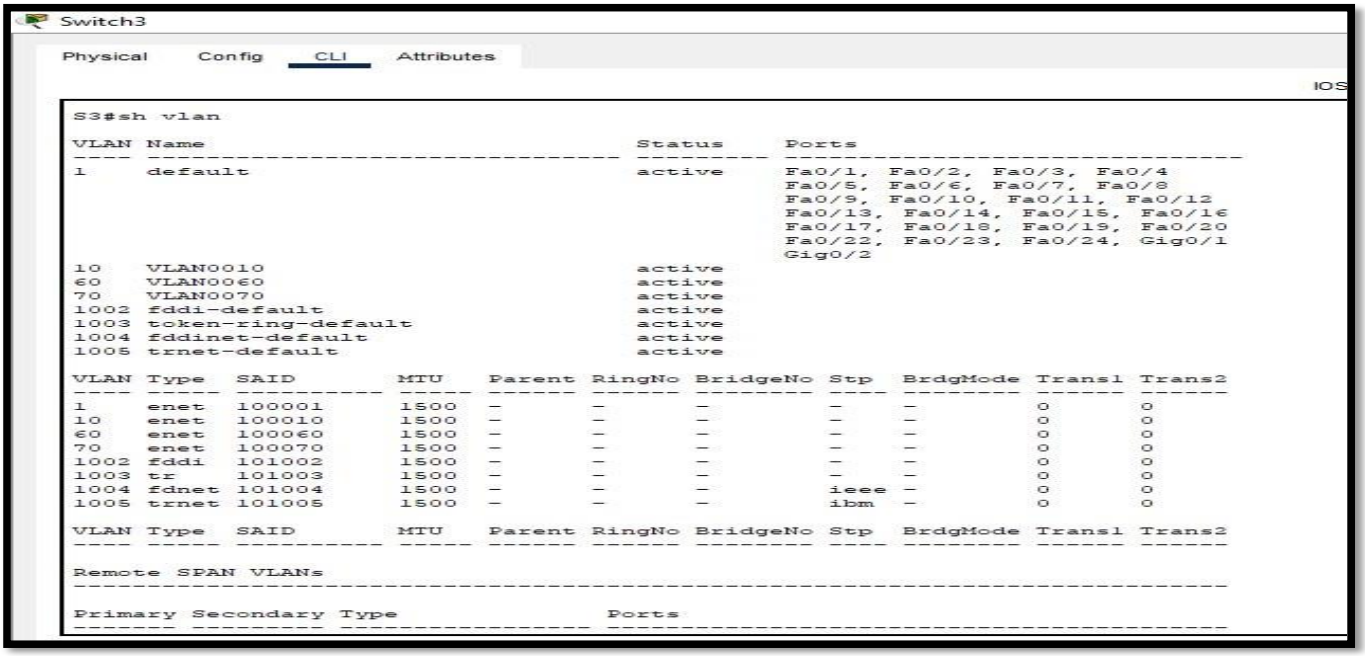
exit





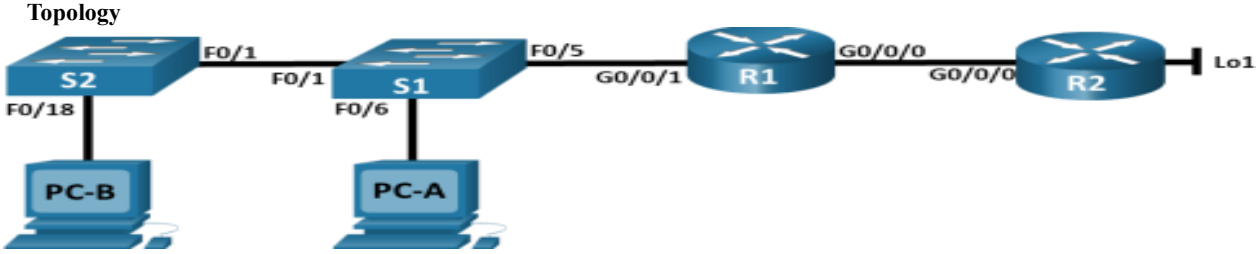
Check VLAN'S are created or not

Sh vlan



3.Aim : Implement NAT

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask
R1	G0/0/0	209.165.200.230	255.255.255.248
	G0/0/1	192.168.1.1	255.255.255.0
R2	G0/0/0	209.165.200.225	255.255.255.248
	Lo1	209.165.200.1	255.255.255.224
S1	VLAN 1	192.168.1.11	255.255.255.0
S2	VLAN 1	192.168.1.12	255.255.255.0
PC-A	NIC	192.168.1.2	255.255.255.0
PC-B	NIC	192.168.1.3	255.255.255.0

What is NAT?

Network Address Translation (NAT) is a service that enables private IP networks to use the internet and cloud. NAT translates private IP addresses in an internal network to a public IP address before packets are sent to an external network.

What is Static NAT?

Static NAT (Network Address Translation) - Static NAT (Network Address Translation) is one-to-one mapping of a private IP address to a public IP address.

What is Dynamic NAT?

Dynamic NAT (Network Address Translation) - Dynamic NAT can be defined as mapping of a private IP address to a public IP address from a group of public IP addresses called as NAT pool. The public to private mapping may vary based on the available public IP address in NAT pool.

What is PAT?

PAT (Port Address Translation) - Port Address Translation (PAT) is another type of dynamic NAT which can map multiple private IP addresses to a single public IP address by using a technology known as Port Address Translation.

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure and verify NAT for IPv4

Part 3: Configure and verify PAT for IPv4

Part 4: Configure and verify Static NAT for IPv4

Background / Scenario

Network Address Translation (NAT) is the process where a network device, such as a Cisco router, assigns a public address to host devices inside a private network . An ISP has allocated the public IP address space of 209.165.200.224/29 to a company. This network is used to address the link between the ISP router (R2) and the company gateway (R1). The first address (209.165.200.225) is assigned to the g0/0/0 interface on R2 and the last address (209.165.200.230) is assigned to the g0/0/0

interface on R1. The remaining addresses (209.165.200.226-209.165.200.229) will be used to provide internet access to the company hosts. A default route is used from R1 to R2. The internet is simulated by a loopback address on R2.

Required Resources

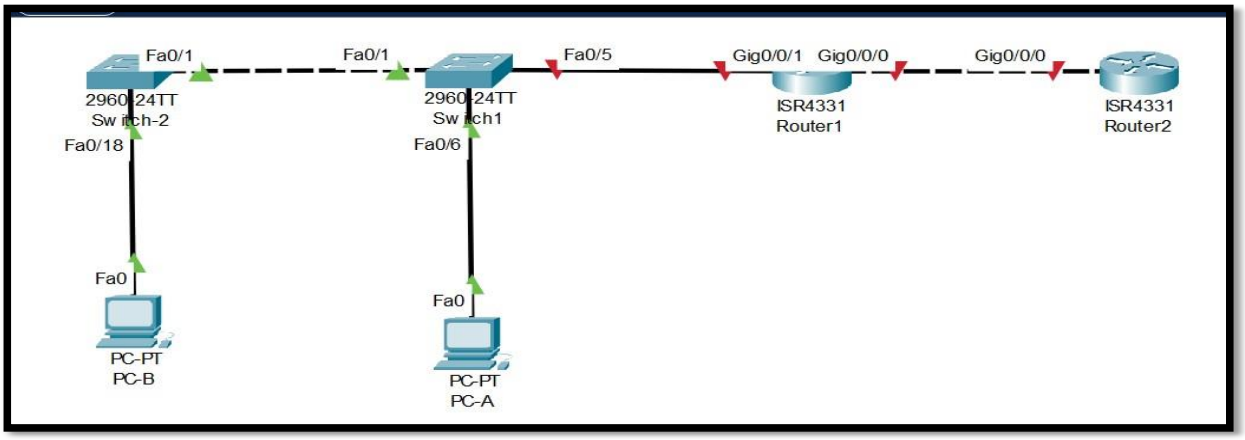
- 2 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Cisco 2960 with Cisco IOS Release 15.2(2) lanbasek9 image or comparable)
- 2 PCs (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

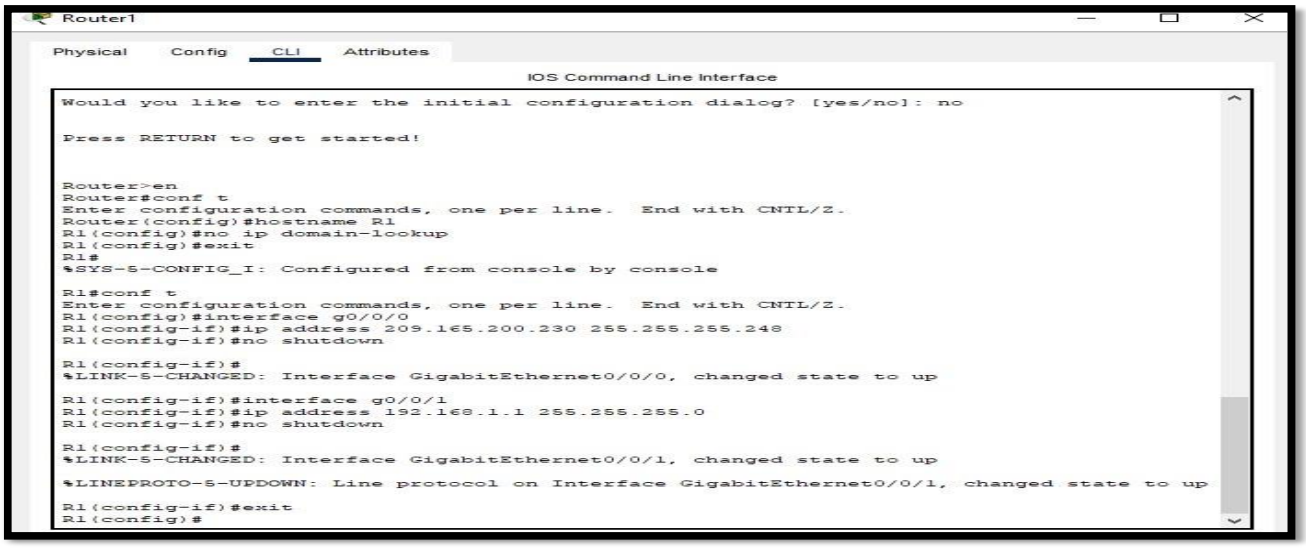
Part 1: Build the Network and Configure Basic Device Settings

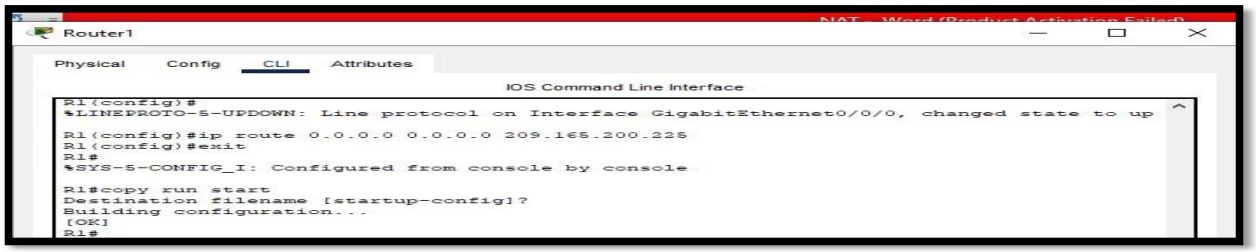
Step 1: Cable the network as shown in the topology.

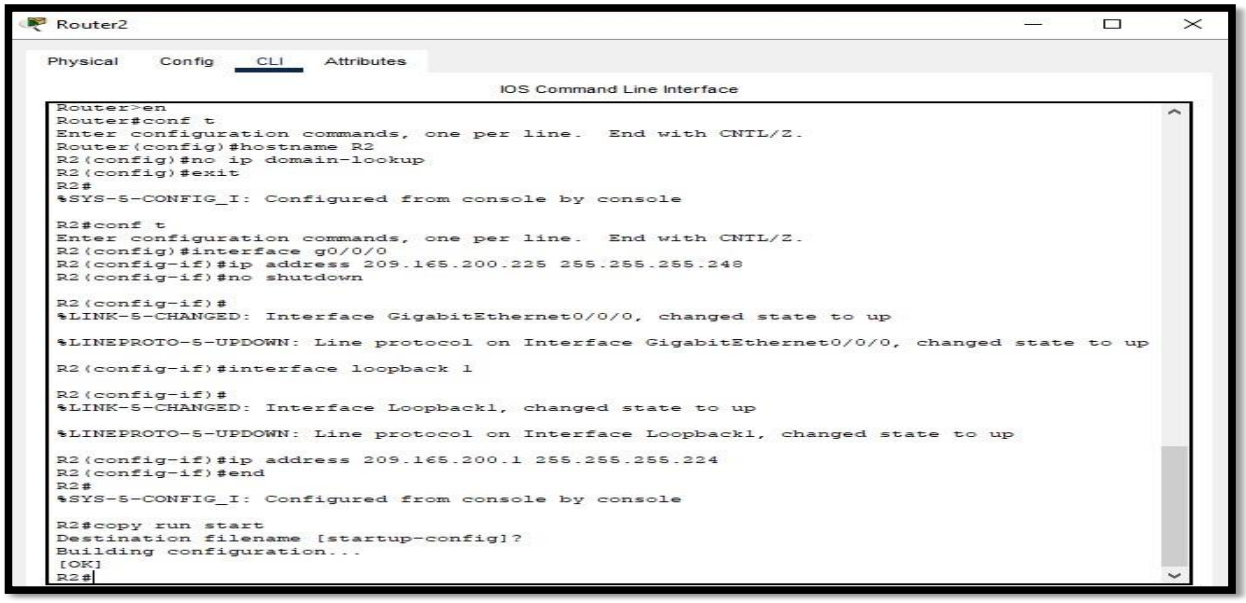
Attach the devices as shown in the topology diagram and cable as necessary.



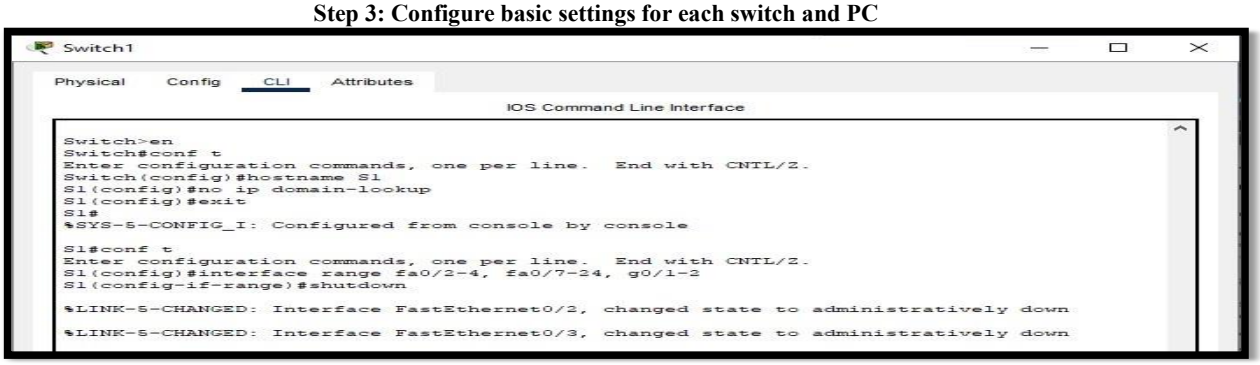
Step 2: Configure basic settings for each router.

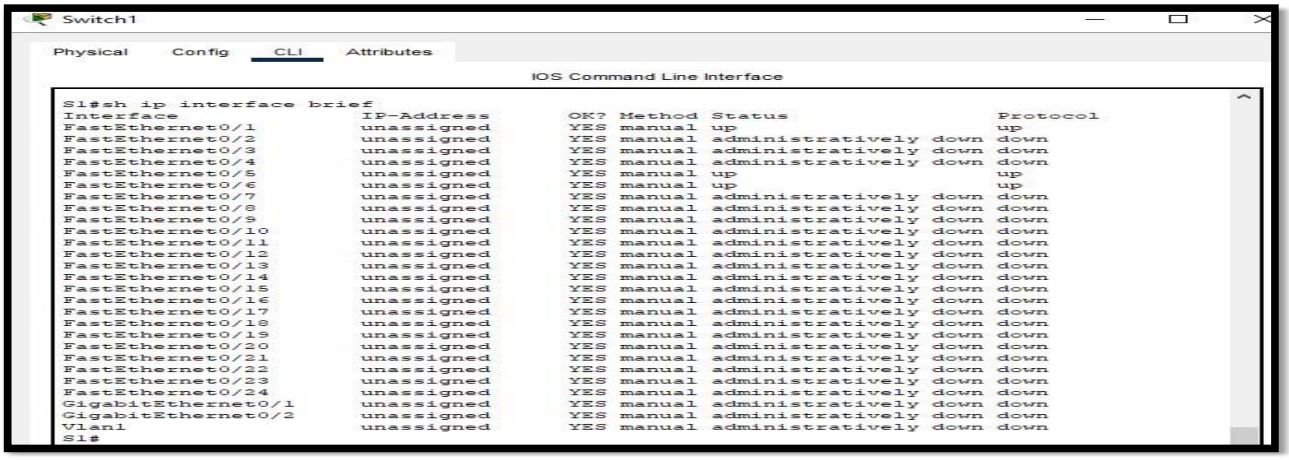


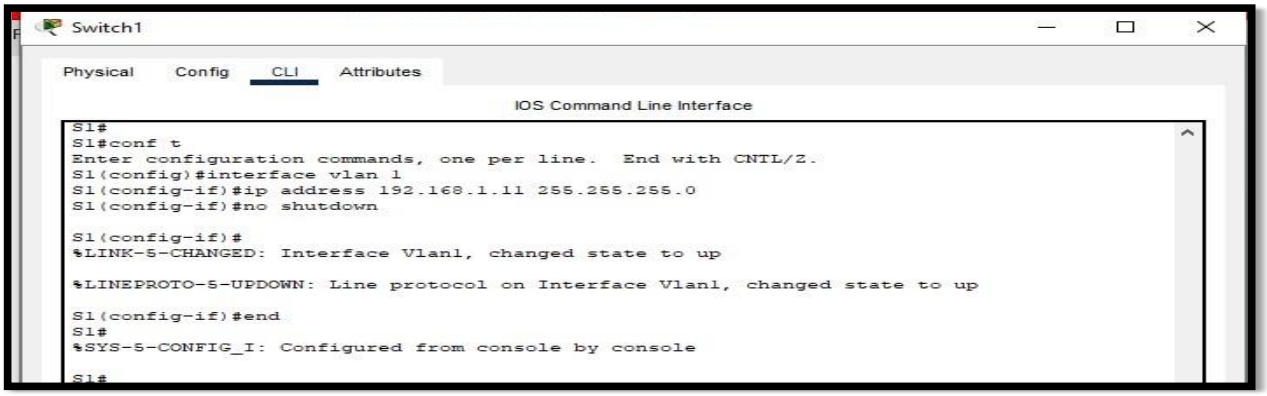


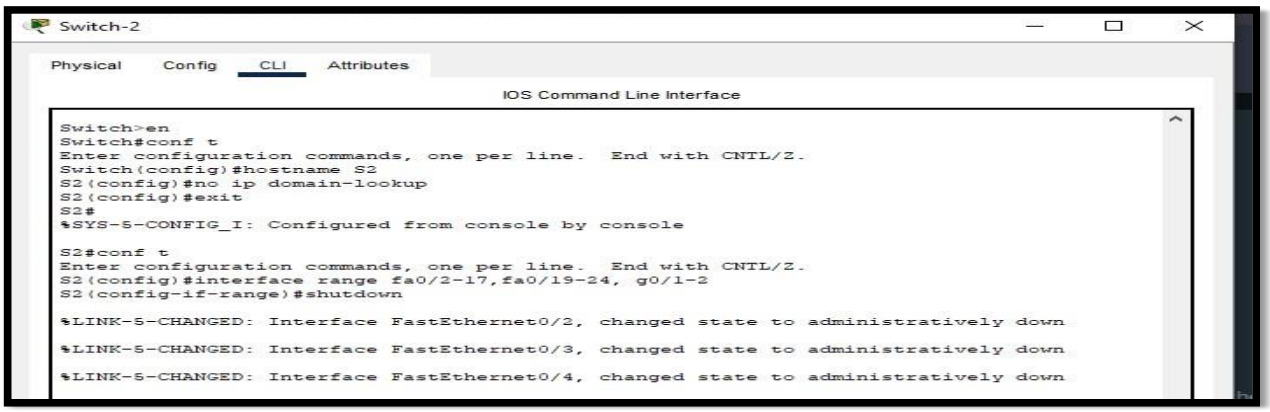


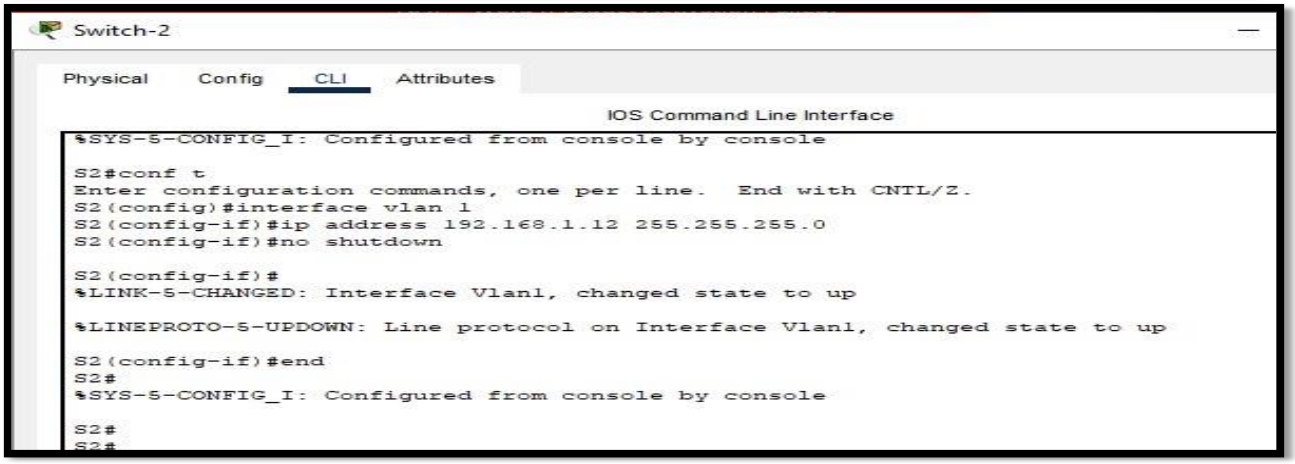
Step 3: Configure basic settings for each switch and PC

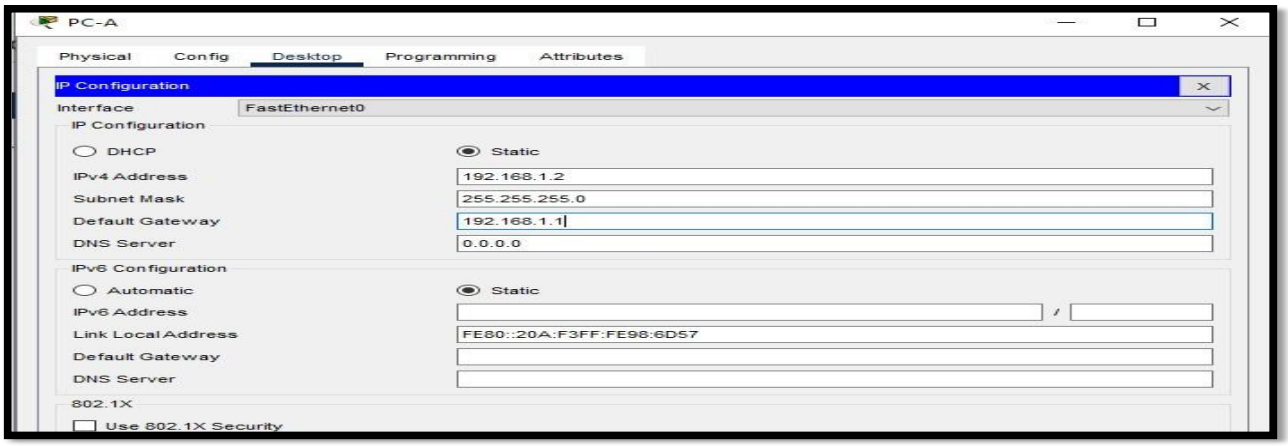


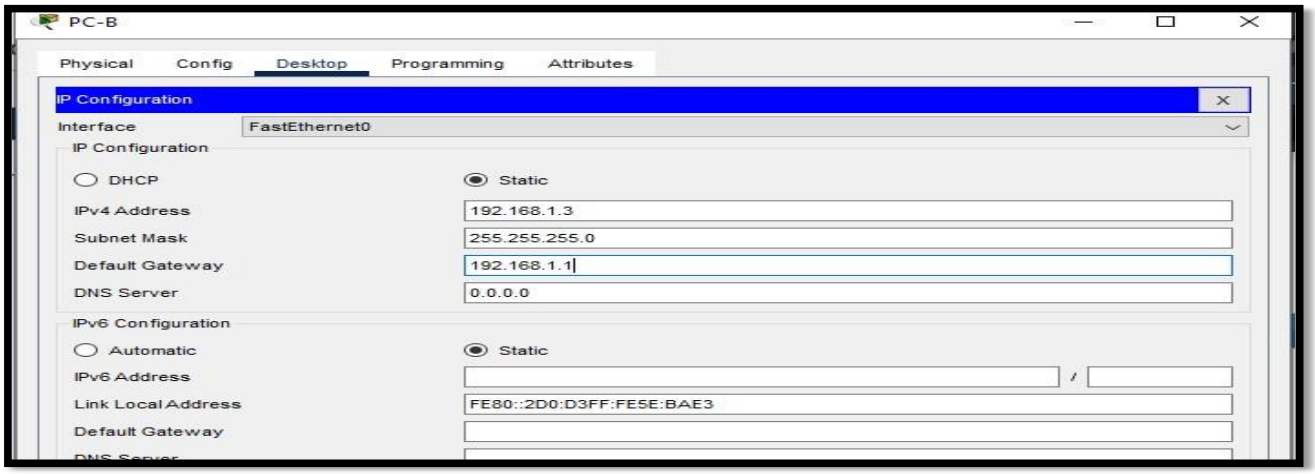



Switch1
Physical
Config
CLI
Attributes
IOS Command Line Interface
S1#
S1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
S1(config)#interface vlan 1
S1(config-if)#ip address 192.168.1.11 255.255.255.0
S1(config-if)#no shutdown
S1(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
S1(config-if)#end
S1#
%SYS-5-CONFIG_I: Configured from console by console
S1#


Switch-2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname S2
S2(config)#no ip domain-lookup
S2(config)#exit
S2#
%SYS-5-CONFIG_I: Configured from console by console
S2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
S2(config)#interface range fa0/2-17,fa0/19-24, g0/1-2
S2(config-if-range)#shutdown
%LINK-5-CHANGED: Interface FastEthernet0/2, changed state to administratively down
%LINK-5-CHANGED: Interface FastEthernet0/3, changed state to administratively down
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to administratively down


Switch-2
Physical
Config
CLI
Attributes
IOS Command Line Interface
%SYS-5-CONFIG_I: Configured from console by console
S2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
S2(config)#interface vlan 1
S2(config-if)#ip address 192.168.1.12 255.255.255.0
S2(config-if)#no shutdown
S2(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
S2(config-if)#end
S2#
%SYS-5-CONFIG_I: Configured from console by console
S2#
S2#


PC-A
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
192.168.1.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::20A:F3FF:FE98:6D57
Default Gateway
DNS Server
802.1X
Use 802.1X Security



Part 2: Configure and verify NAT for IPv4

Step 1: Configure NAT on R1 using a pool of three addresses, 209.165.200.226- 209.165.200.228.

a. Configure a simple access list that defines what hosts are going to be allowed for translation. In this case,all devices on the R1 LAN are eligible for translation.

R1(config)# **access-list 1 permit 192.168.1.0 0.0.0.255**

b. Create the NAT pool, and give it a name and a range of addresses to use.

R1(config)# **ip nat pool PUBLIC_ACCESS 209.165.200.226 209.165.200.228 netmask 255.255.255.248**

c. Configure the translation, associating the ACL and Pool to the translation process.
R1(config)# **ip nat inside source list 1 pool PUBLIC_ACCESS**

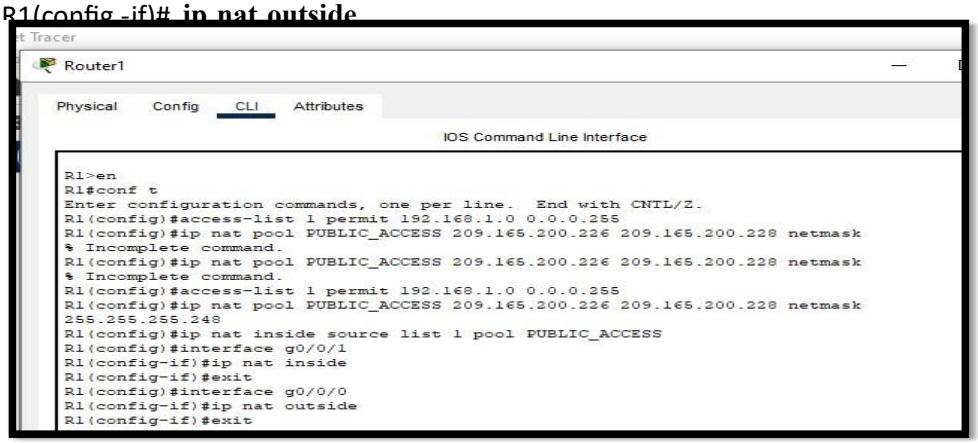
d. Define the inside interface.

R1(config)# **interface g0/0/1**

R1(config-if)# **ip nat inside**

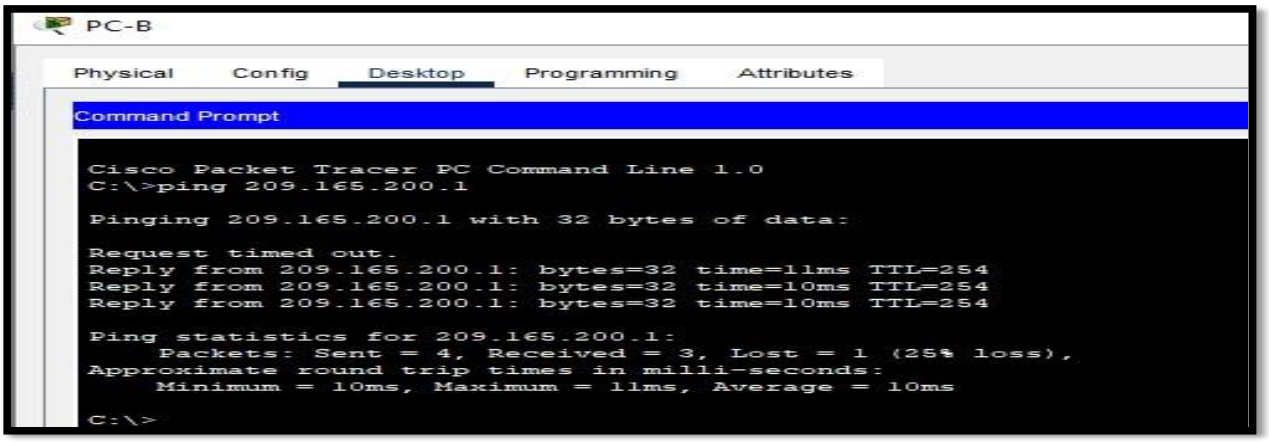
e. Define the outside interface.

R1(config)# **interface g0/0/0**
R1(config-if)# **ip nat outside**

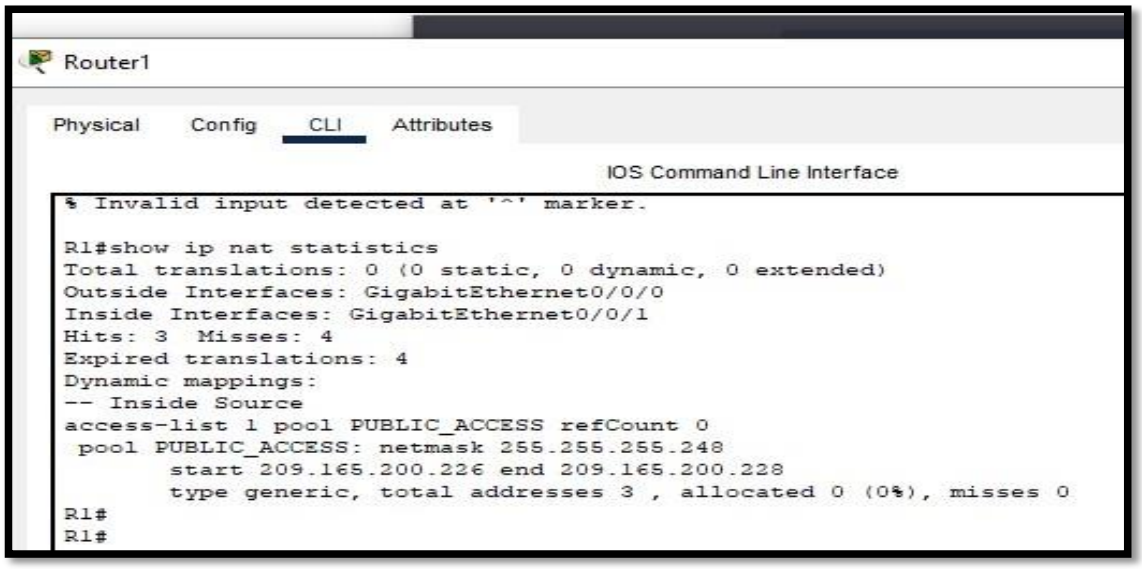


Step 2: Test and Verify the configuration.

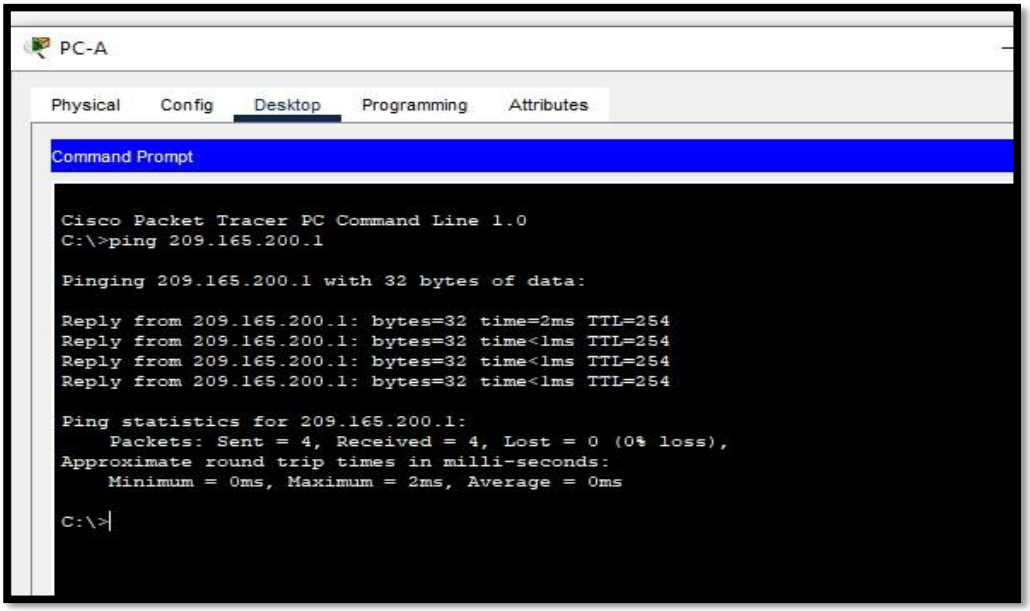
a. From PC-B, ping the Lo1 interface (209.165.200.1) on R2. If the ping was unsuccessful, troubleshoot and correct the issues. On R1, display the NAT table on
R1 with the command **show ip nat translations.**



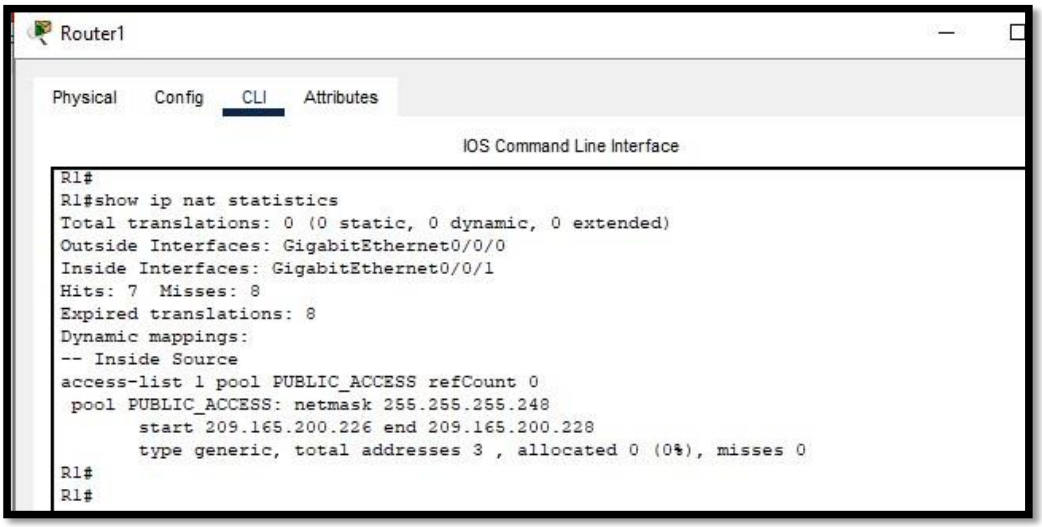
R1# **show ip nat translations**



b. From PC-A, ping the Lo1 interface (**209.165.200.1**) on R2. If the ping was unsuccessful, troubleshoot and correct the issues. On R1, display the NAT table on R1 with the command **show ip nat translations.**

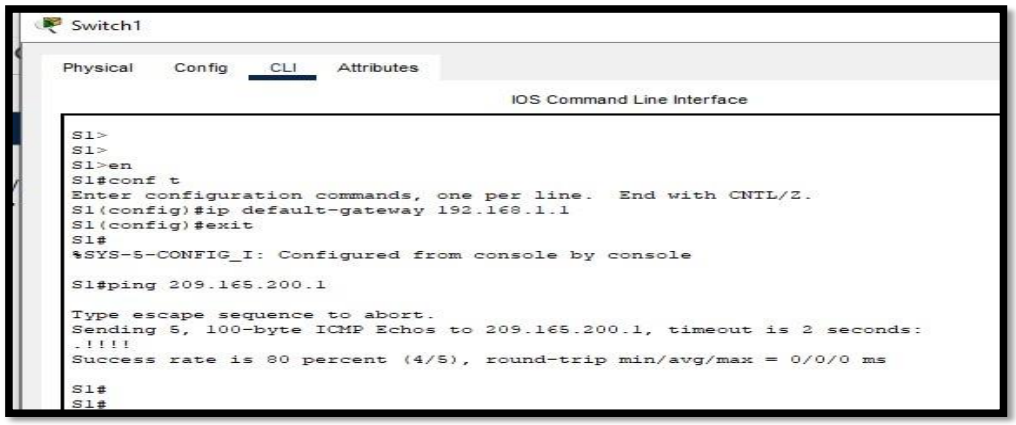


R1# **show ip nat translations**

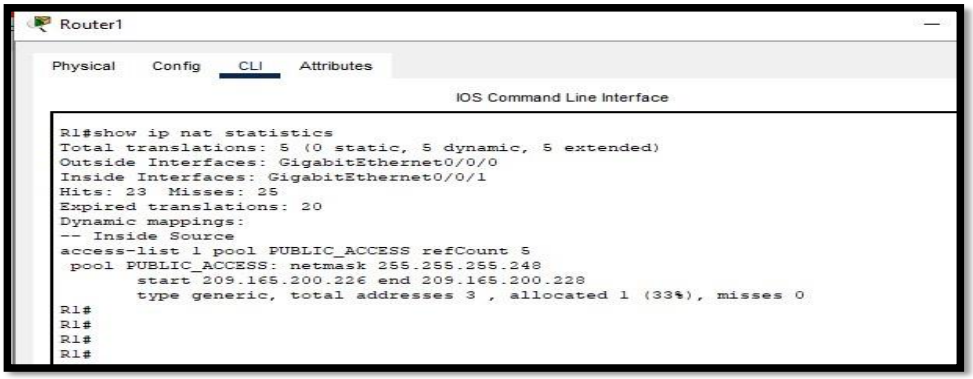


c. Notice that the previous translation for PC-B is still in the table. From S1, ping the Lo1 interface (**209.165.200.1**) on R2.

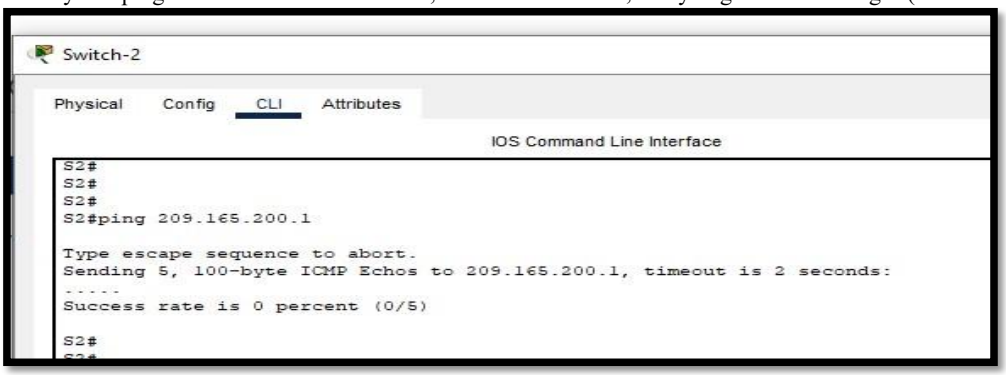
If the ping was unsuccessful, troubleshoot and correct the issues. On R1, display the NAT table on R1 with the command **show ip nat translations**.



R1# **show ip nat translations**



d. Now try and ping R2 Lo1 from S2. This time, the translations fail, and you get these messages (or similar) on the R1 console:

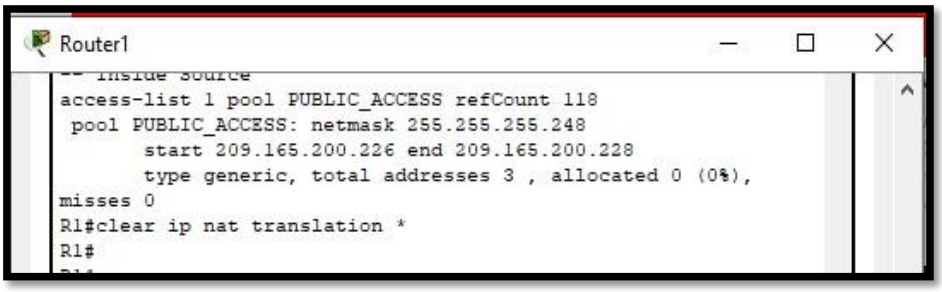


e. This is an expected result, because only 3 addresses are allocated, and we tried to ping Lo1 from four devices. Recall that NAT is a one-to-one translation. So how long are the translations allocated? Issue the command **show ip nat translations verbose** and you will see that the answer is for 24 hours.

R1# **show ip nat translations verbose**

f. Given that the pool is limited to three addresses, NAT to a pool of addresses is not adequate for our application. Clear the NAT translations and statistics and we will move on to PAT.

R1# clear ip nat translations



Part 3: Configure and verify PAT for IPv4

In Part 3, you will configure replace NAT with PAT to a pool of addresses, and then with PAT using an interface.

Step 1: Remove the translation command on R1.

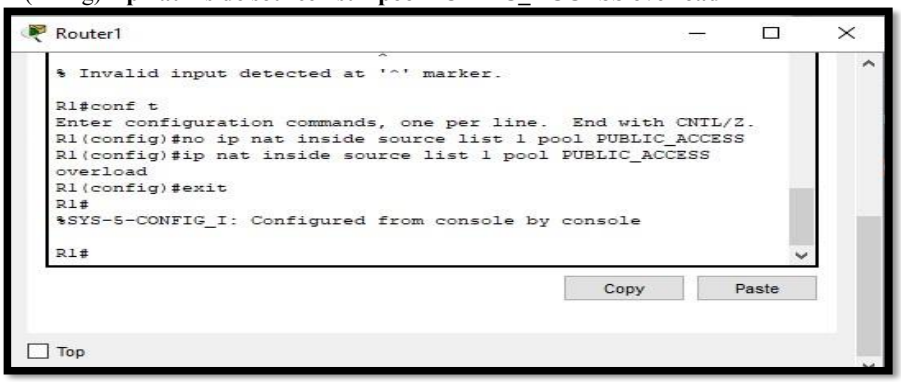
The components of an Address Translation configuration are basically the same; something (an access-list) to identify addresses eligible to be translated, an optionally configured pool of addresses to translate them to, and the commands necessary to identify the inside and outside interfaces. From Part 1, our access-list (access-list 1) is still correct for the network scenario, so there is no need to recreate it. We are going to use the same pool of addresses, so there is no need to recreate that configuration either. Also, the inside and outside interfaces are not changing. To get started in Part 3, remove the command that ties the ACL and pool together.

R1(config)# **no ip nat inside source list 1 pool PUBLIC_ACCESS**

Step 2: Add the PAT command on R1.

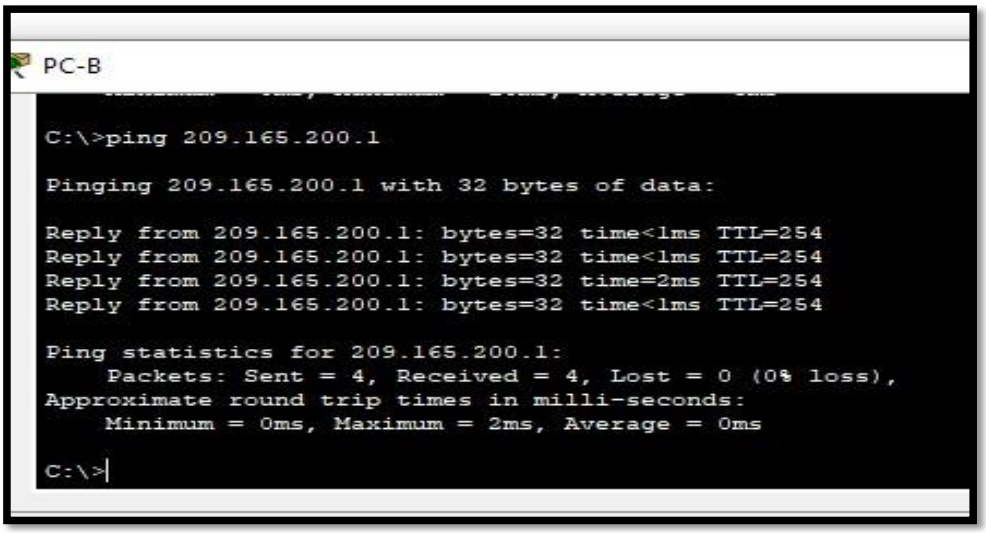
Now, configure for PAT translation to a pool of addresses (remember, the ACL and Pool are already configured, so this is the only command we need to change from NAT to PAT).

R1(config)# **ip nat inside source list 1 pool PUBLIC_ACCESS overload**

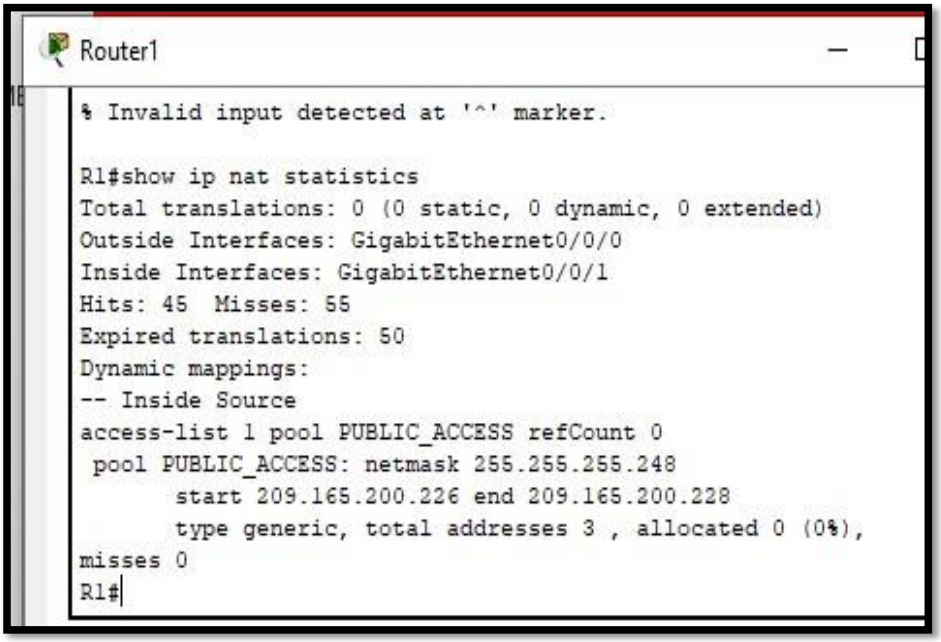


Step 3: Test and Verify the configuration.

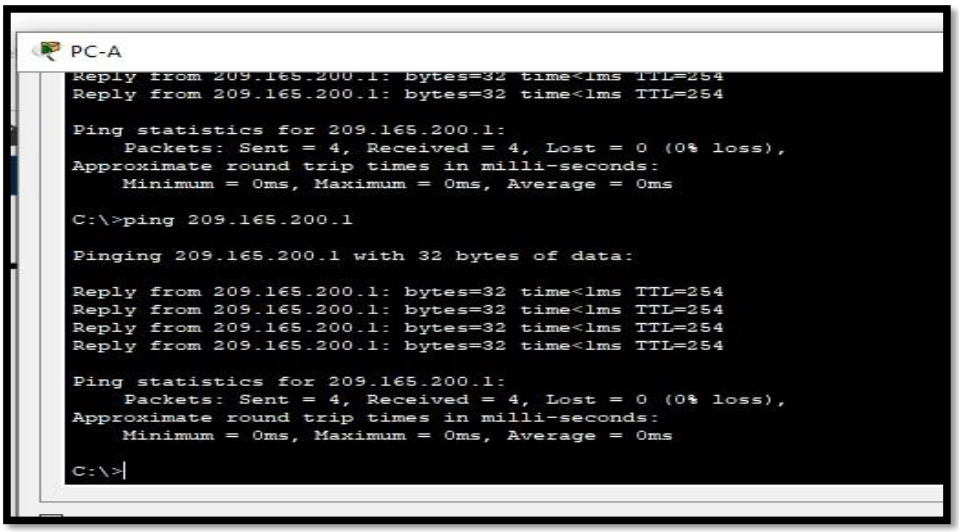
a. Let's verify PAT is working. From PC-B, ping the Lo1 interface (209.165.200.1) on R2. If the ping was unsuccessful, troubleshoot and correct the issues. On R1, display the NAT table on R1 with the command **show ip nat translations.**



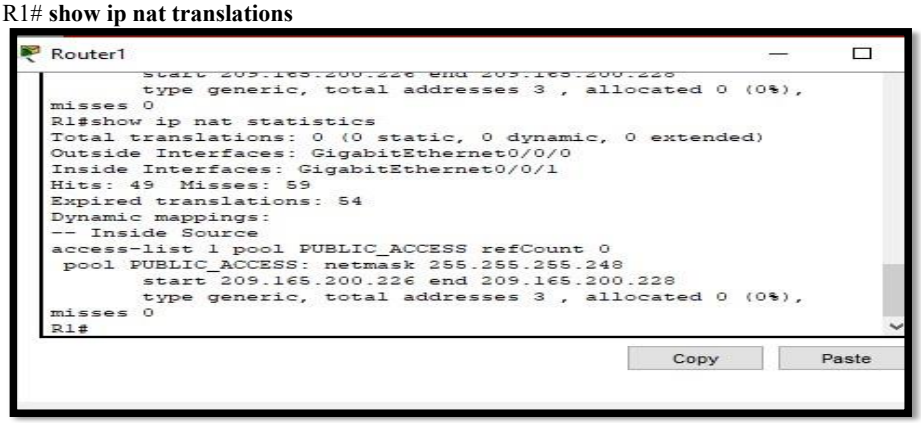
R1# **show ip nat translations**



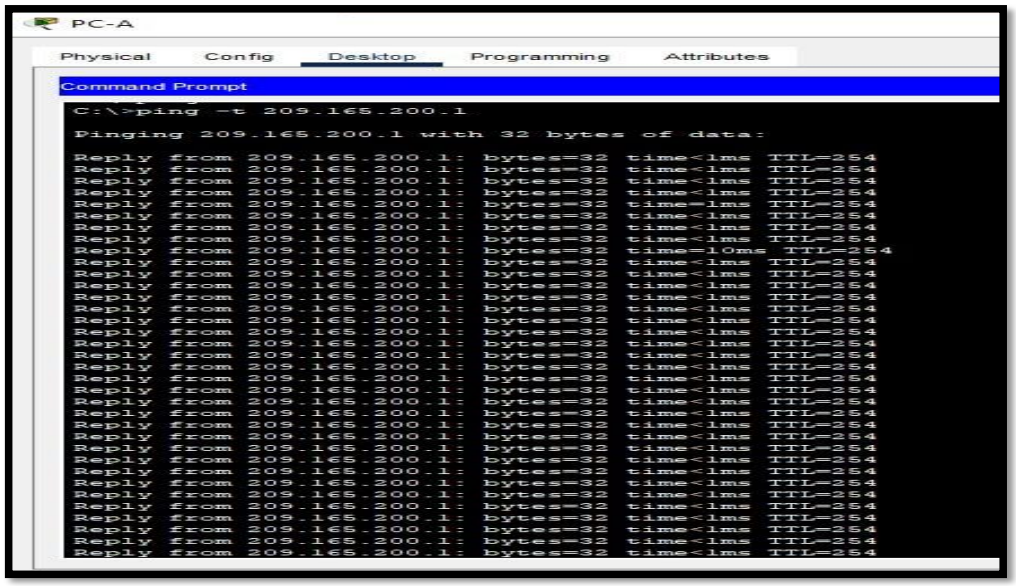
b. From PC-A, ping the Lo1 interface (209.165.200.1) on R2. If the ping was unsuccessful, troubleshoot and correct the issues. On R1, display the NAT table on R1 with the command **show ip nat translations.**

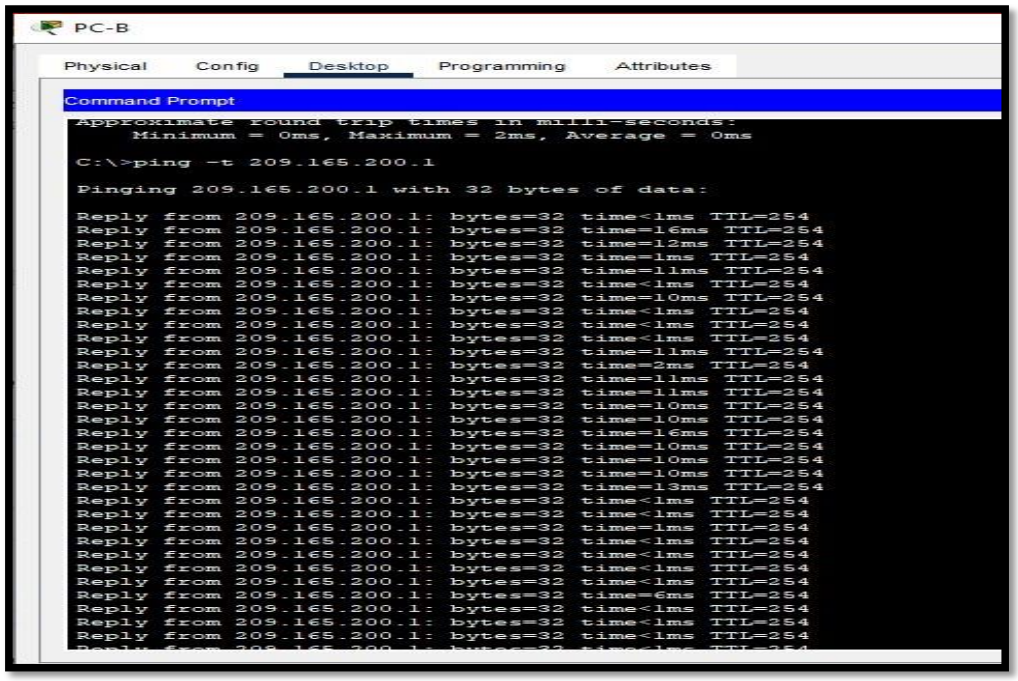


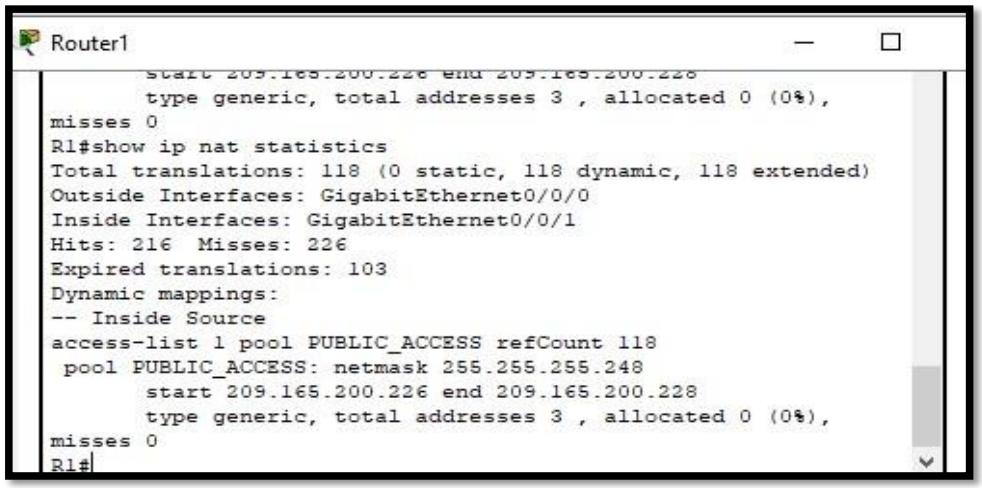
R1# **show ip nat translations**



c. Generate traffic from multiple devices to observe PAT. On PC-A and PC-B, use the -t parameter with the ping command to send a non-stop ping to R2's Lo1 interface (**ping -t 209.165.200.1**), then go back to R1 and issue the **show ip nat translations** command:



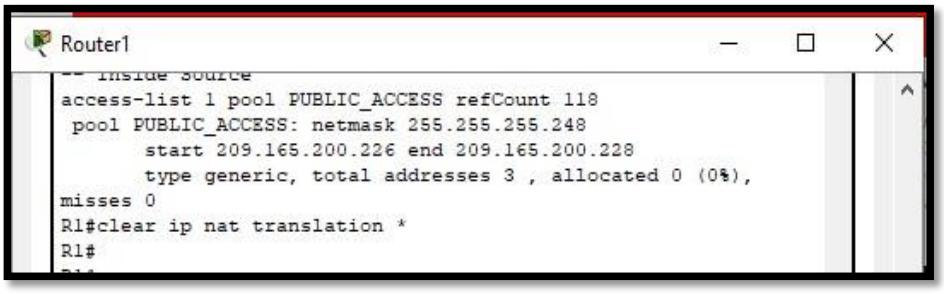




d. PAT to a pool is a very effective solution for small-to-midsize organizations. However, there are unused IPv4 addresses involved in this scenario. We will move to PAT with interface overload to eliminate this waste of IPv4 addresses. Stop the pings on PC-A and PC-B with the Control-C key combination, then clear translations and translation statistics:

R1# **clear ip nat translations ***

R1# **clear ip nat statistics**



Step 4: On R1, remove the nat pool translation commands.

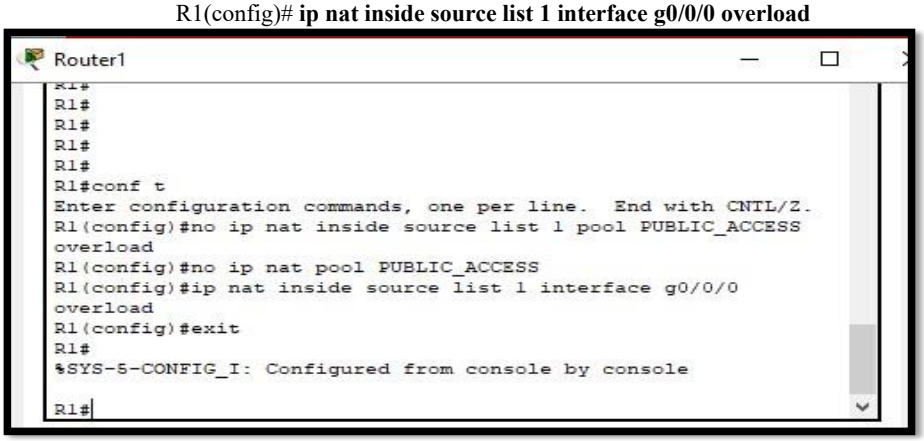
Once again, our access-list (access-list 1) is still correct for the network scenario, so there is no need to recreate it. Also, the inside and outside interfaces are not changing.

R1(config)# **no ip nat inside source list 1 pool PUBLIC_ACCESS overload**

R1(config)# **no ip nat pool PUBLIC_ACCESS**

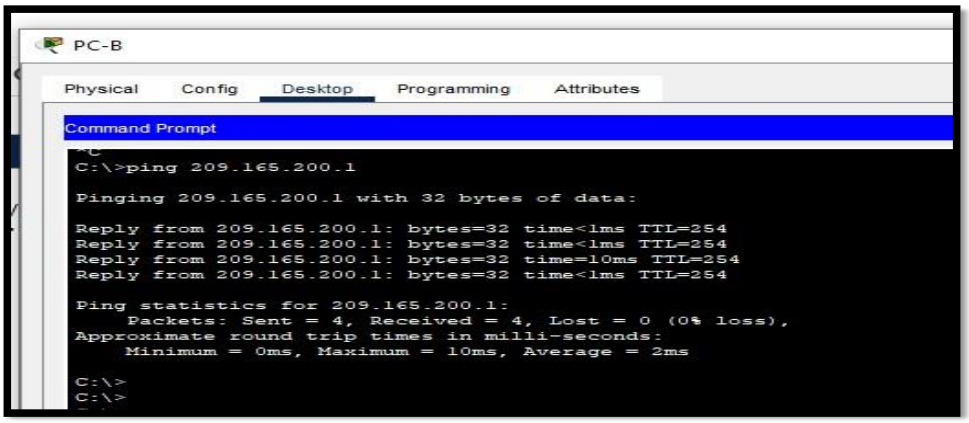
Step 5: Add the PAT overload command by specifying the outside interface.

R1(config)# **ip nat inside source list 1 interface g0/0/0 overload**

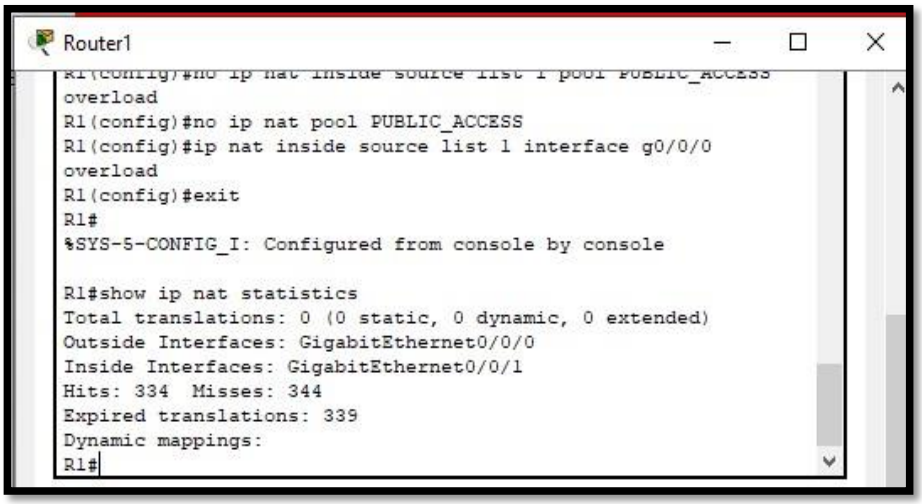


Step 6: Test and Verify the configuration.

a. Let's verify PAT to the interface is working. From PC-B, ping the Lo1 interface (209.165.200.1) on R2. If the ping was unsuccessful, troubleshoot and correct the issues

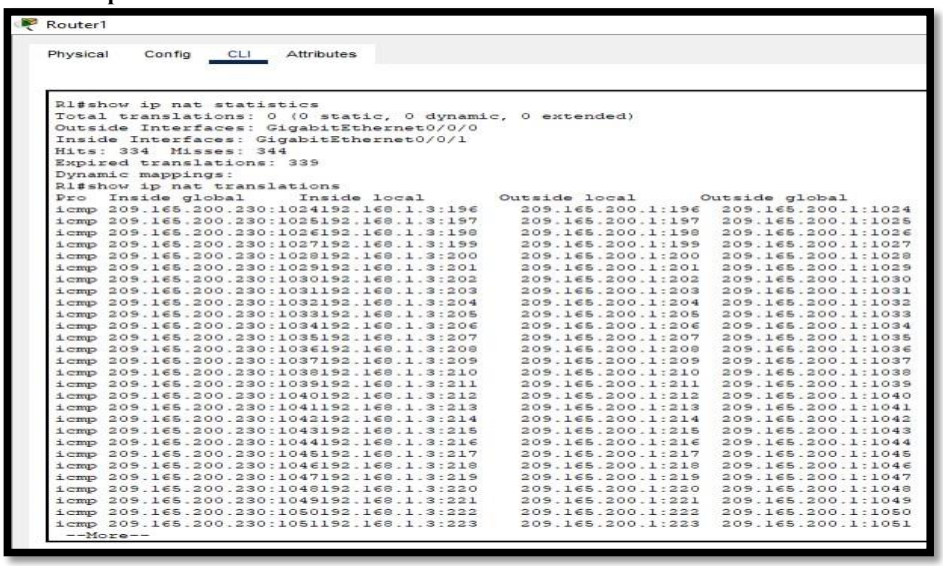


R1# **show ip nat translations**



b. Generate traffic from multiple devices to observe PAT. On PC-A and PC-B, use the -t parameter with the ping command to send a non-stop ping to R2's Lo1 interface **(ping -t 209.165.200.1)**. On S1 and S2, issue the privileged exec command ping 209.165.200.1 repeat 2000. Then go back to R1 and issue the **show ip nat translations** command.

R1# **show ip nat translations**

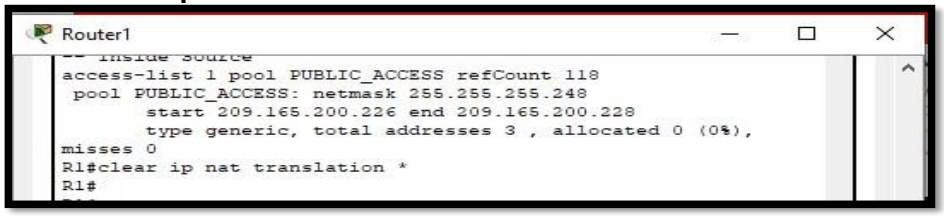


Part 4: Configure and verify Static NAT for IPv4

In Part 4, you will configure static NAT so that PC-A is directly reachable from the internet. PC-A will be reachable from R2 via the address 209.165.200.229.

Step 1: On R1, clear current translations and statistics. R1# **clear ip nat translations ***

R1# **clear ip nat statistics**



Step 2: On R1, configure the NAT command required to statically map an inside address to an outside address.

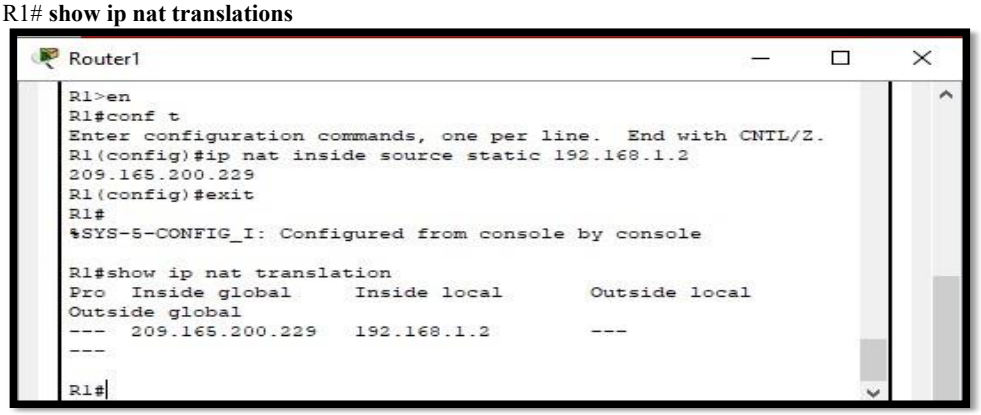
For this step, configure a static mapping between 192.168.1.11 and 209.165.200.1 using the following command:

R1(config)# **ip nat inside source static 192.168.1.2 209.165.200.229**

Step 3: Test and Verify the configuration.

a. Let's verify the Static NAT is working. On R1, display the NAT table on R1 with the command **show ip nat translations**, and you should see the static mapping.

R1# **show ip nat translations**

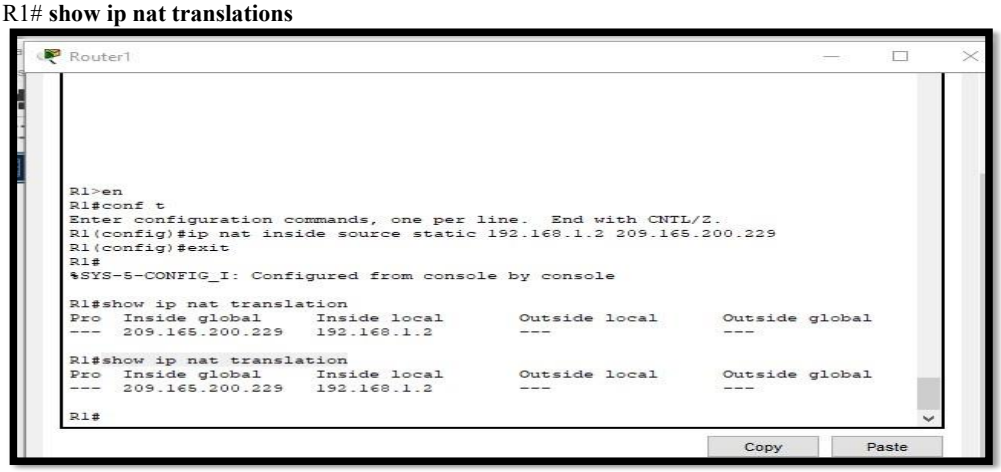


b. The translation table shows the static translation is in effect. Verify this by pinging from R2 to 209.165.200.229. The pings should work.

Note: you may have to disable the PC firewall for the pings to work.

c. On R1, display the NAT table on R1 with the command **show ip nat translations**, and you should see the static mapping and the port-level translation for the inbound pings.

R1# **show ip nat translations**



This validates that the Static NAT is working.

Practical 5

Aim : Implement Inter-VLAN Networking

❖ What is Inter-Vlan Routing ?

- Inter-VLAN (Virtual Local Area Network) routing is a technique used to enable communication between different VLANs in a network. VLANs are used to logically segment a network into multiple broadcast domains, and each VLAN functions as a separate virtual network
- Inter-VLAN routing is necessary when you want devices in different VLANs to communicate with each other. This is typically achieved by using a router or a Layer 3 switch to route traffic between the VLANs. Here's a brief overview of how inter-VLAN routing works:
- Physical or Logical Separation: VLANs are created to logically or physically separate devices in a network. Each VLAN has its own broadcast domain.
- Router or Layer 3 Switch: A router or a Layer 3 switch is used to perform inter-VLAN routing. This device has interfaces configured for each VLAN, effectively acting as a gateway for the devices in those VLANs.
- Subinterfaces: On a router, subinterfaces are configured on the router interface connected to the switch, with each subinterface corresponding to a specific VLAN. Each subinterface is assigned an IP address in the respective VLAN's subnet.
- segmentation.

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0/1.10	192.168.10.1	255.255.255.0	N/A
	G0/0/1.20	192.168.20.1	255.255.255.0	
	G0/0/1.30	192.168.30.1	255.255.255.0	
	G0/0/1.1000	N/A	N/A	
S1	VLAN 10	192.168.10.11	255.255.255.0	192.168.10.1
S2	VLAN 10	192.168.10.12	255.255.255.0	192.168.10.1
PC-A	NIC	192.168.20.3	255.255.255.0	192.168.20.1
PC-B	NIC	192.168.30.3	255.255.255.0	192.168.30.1

VLAN Table

VLAN	Name	Interface Assigned
10	Management	S1: VLAN 10 S2: VLAN 10
20	Sales	S1: F0/6
30	Operations	S2: F0/18
999	Parking_Lot	S1: F0/2-4, F0/7-24, G0/1-2 S2: F0/2-17, F0/19-24, G0/1-2
1000	Native	N/A

❖ Objectives :

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Create VLANs and Assign Switch Ports

Part 3: Configure an 802.1Q Trunk between the Switches

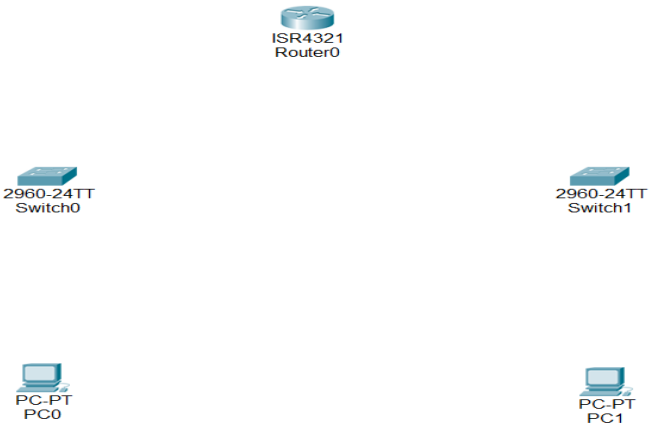
Part 4: Configure Inter-VLAN Routing on the Router

Part 5: Verify Inter-VLAN Routing is working

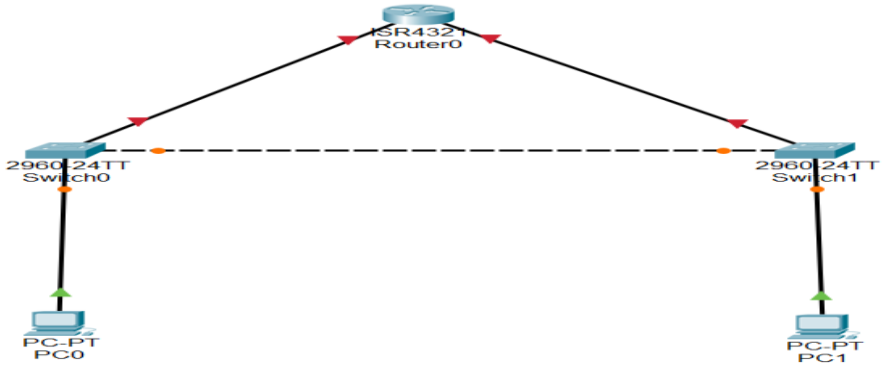
❖ Required Resources :

1. **Router : ISR4321 Router**
2. **Switches : 2960-24TT Switch**
3. **PCs**

❖ Part 1 : Build the Network and Configure Basic Device Settings



Cable the network as shown in the topology.



Step 1 : Configure basic settings for the switch.

S1 :

- en
- conf t
- hostname S1
- vlan 20
- name Sales
- vlan 30
  name Operations

S2 :

- en
- conf t
- hostname S2
- vlan 20
  name Sales
- vlan 30
  name Operations

```
FastEthernet0/2, changed state to up


Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
Switch(config)#hostname S1
S1(config)#vlan 20
S1(config-vlan)#name Sales
S1(config-vlan)#vlan 30
S1(config-vlan)#name Operations
```

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
Switch(config)#hostname S2
S2(config)#vlan 20
S2(config-vlan)#name Sales
S2(config-vlan)#vlan 30
S2(config-vlan)#name Operations
S2(config-vlan)#
```

- **do sh vlan (**To check if vlan is created**)**
  exit

```
S1(config-vlan)#vlan 30
S1(config-vlan)#name Operations
S1(config-vlan)#do sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
20   Sales                            active
30   Operations                       active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
30   enet  100030     1500  -      -      -        -    -        0      0
 --More--
```

Step 2 : Assign Vlans to correct switch interfaces

S1 :

- interface fastEthernet 0/1
- switchport mode access
- switchport access vlan 20

S2 :

- interface fastEthernet 0/1
- switchport mode access
- switchport access vlan 30

```
S1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#inter
S1(config)#interface fast
S1(config)#interface fastEthernet 0/1
S1(config-if)#switch
S1(config-if)#switchport mo
S1(config-if)#switchport mode ?
  access    Set trunking mode to ACCESS unconditionally
  dynamic   Set trunking mode to dynamically negotiate access or trunk mode
  trunk     Set trunking mode to TRUNK unconditionally
S1(config-if)#switchport mode acc
S1(config-if)#switchport mode access
S1(config-if)#switchport access vlan 20
```

- show vlan brief

```
S2#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                Gig0/2
20   Sales                            active
30   Operations                       active    Fa0/1
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
S2#
```

➢ Manually configure trunk interface F0/1 on switch S1 and S2.

S1 :

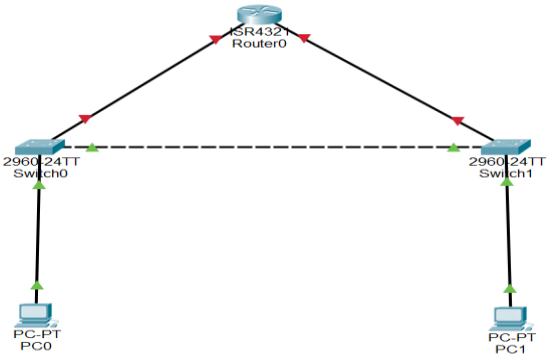
- interface f0/1
  switchport mode trunk

S2 :

- interface f0/1
  switchport mode trunk

```
S1(config)#
S1(config)#inter
S1(config)#interface f0/1
S1(config-if)#swi
S1(config-if)#switchport mo
S1(config-if)#switchport mode tr
S1(config-if)#switchport mode trunk

S1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

S1(config-if)#
```

```
S2(config)#interface f0/1
S2(config-if)#sw
S2(config-if)#switchport mo
S2(config-if)#switchport mode ?
  access   Set trunking mode to ACCESS unconditionally
  dynamic  Set trunking mode to dynamically negotiate access or trunk mode
  trunk    Set trunking mode to TRUNK unconditionally
S2(config-if)#switchport mode tru
S2(config-if)#switchport mode trunk

S2(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

S2(config-if)#
```



Step 2 : Basic config of router

- en
  conf t
  hostname R1

- interface gigabitEthernet 0/0/0
  no shutdown
  exit

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R!
R!(config)#hostname R1
R1(config)#inter
R1(config)#interface gig
R1(config)#interface gigabitEthernet 0/0/0
R1(config-if)#no shutdown

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

R1(config-if)#exit
R1(config)#
```

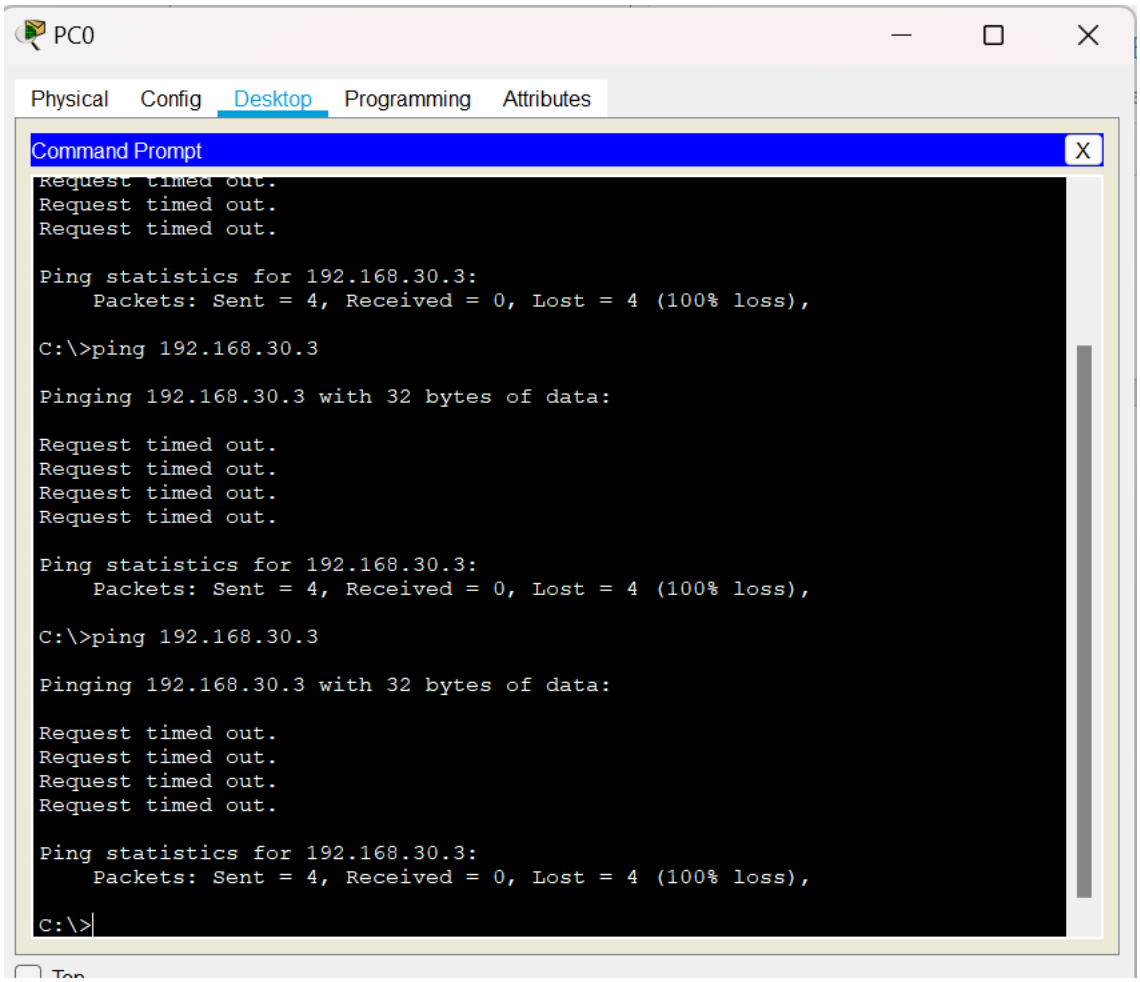
- interface gigaEthernet 0/0/0.20
  encapsulation dot1Q 20
  description Sales
  ip address 192.168.20.1 255.255.255.0

```
R1(config-if)#exit
R1(config)#interface gigabitEthernet 0/0/0.20
R1(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0.20, changed state t

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/

R1(config-subif)#en
R1(config-subif)#encapsulation do
R1(config-subif)#encapsulation dot1Q
% Incomplete command.
R1(config-subif)#encapsulation dot1Q 20
R1(config-subif)#description Sales
R1(config-subif)#ip address 192.168.20.1 255.255.255.0
R1(config-subif)#exit
R1(config)#
```

- interface gigaEthernet 0/0/0.30
  encapsulation dot1Q 30
  description Sales
  ip address 192.168.30.1 255.255.255.0

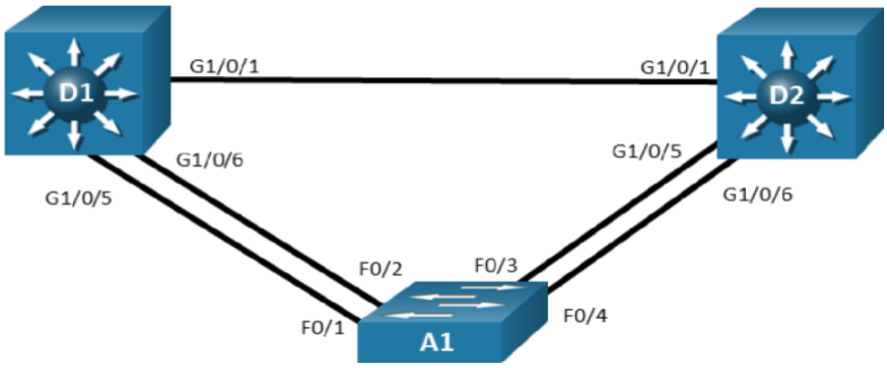
```
R1(config-subif)#exit
R1(config)#interface gigabitEthernet 0/0/0.30
R1(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0.30, changed state to up

R1(config-subif)#encapsulation dot1Q 30
R1(config-subif)#description Operations
R1(config-subif)#ip address 192.168.30.1 255.255.255.0
R1(config-subif)#exit
R1(config)#
```

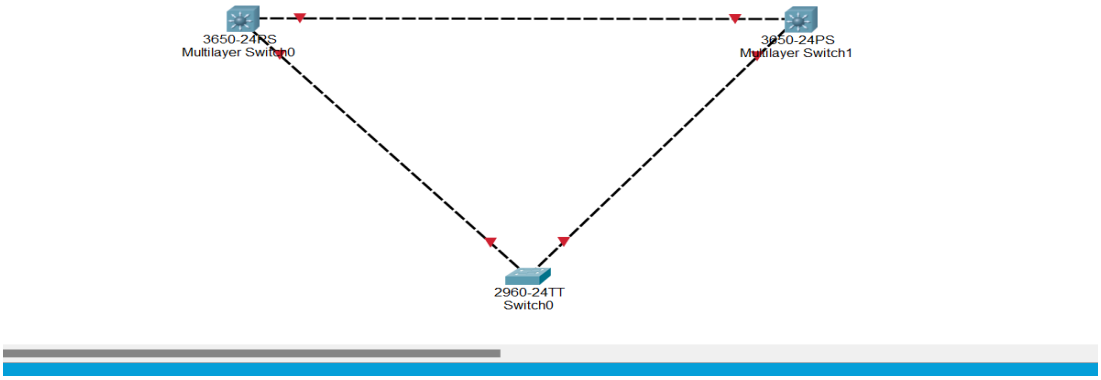
Step 3: Verify Inter-VLAN Routing is Working



Practical 6

Aim : Observe STP topology changes and implement RSTP





Addressing Table

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

Objectives:-

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Observe STP Convergence and Topology Change Part 3: Configure and

Required Resources

- 2 Switches (Cisco 3650 with Cisco IOS XE release 16.9.4 universal image or comparable)

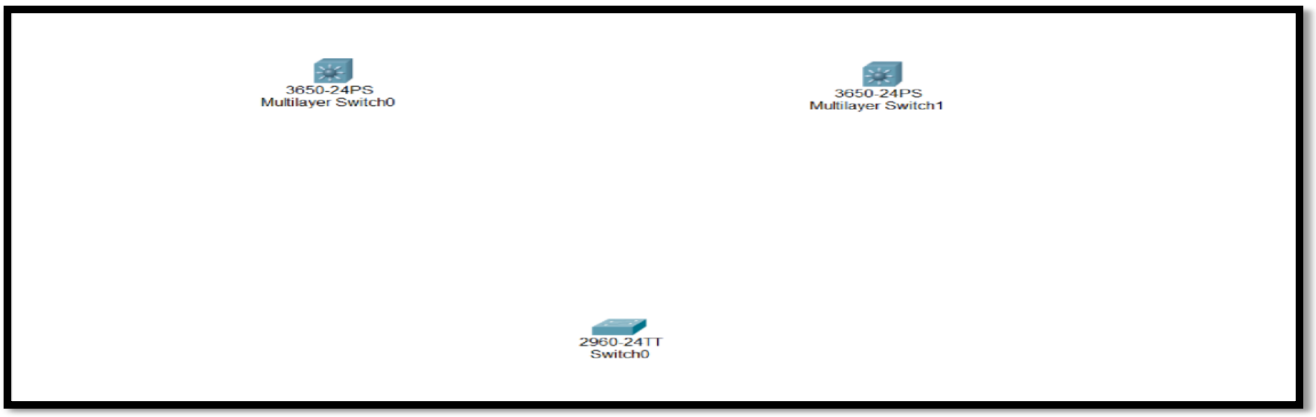
- 1 Switch (Cisco 2960+ with Cisco IOS release 15.2 lanbase image or comparable)
- 1 PC (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

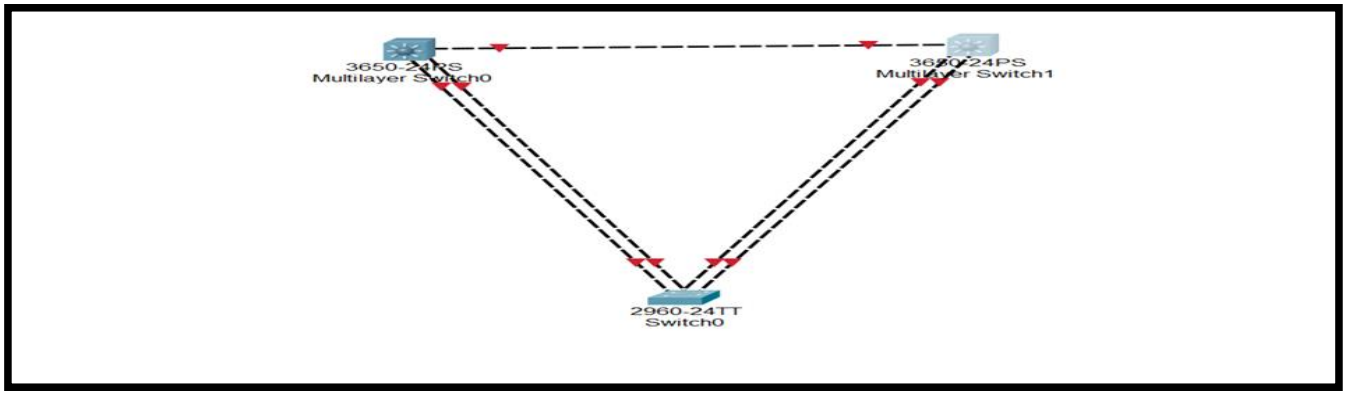
What is STP?

- Spanning Tree Protocol (STP) is a Layer 2 network protocol used to prevent looping within a network topology.
- STP was created to avoid the problems that arise when computers exchange data on a local area network (LAN) that contains redundant paths.
- If the flow of traffic is not carefully monitored and controlled, the data can be caught in a loop that circles around network segments, affecting performance and bringing traffic to a near halt. Networks are often configured with redundant paths when connecting network segments.
- STP can help prevent bridge looping on LANs that include redundant links.
- STP monitors all network links, identifies redundant connections, and disables the ports that can lead to looping.

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

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





Step 2: Configure basic settings for each switch.

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

Switch D1 hostname D1 spanning-tree mode

pvst

interface range g1/0/1-24, g1/1/1-4, g0/0

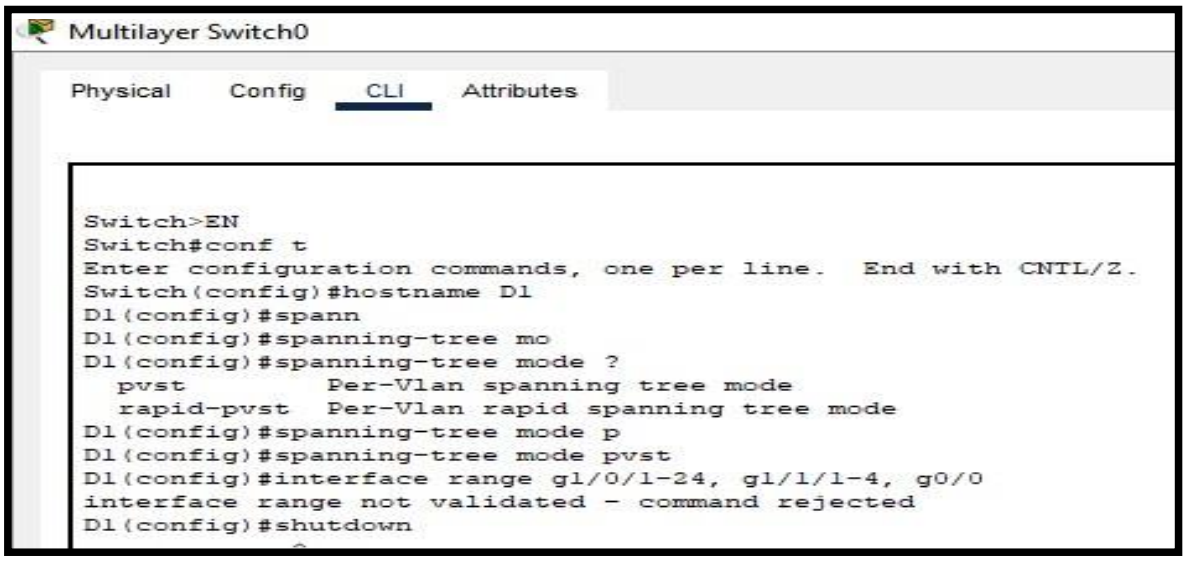
shutdown

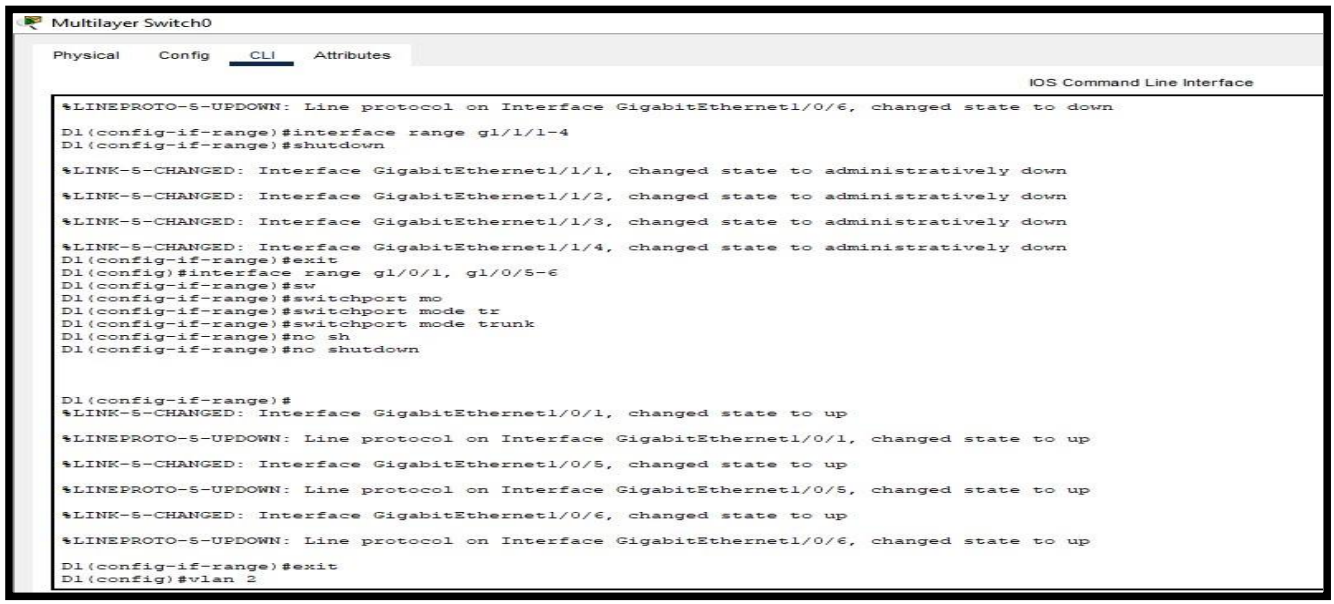
exit

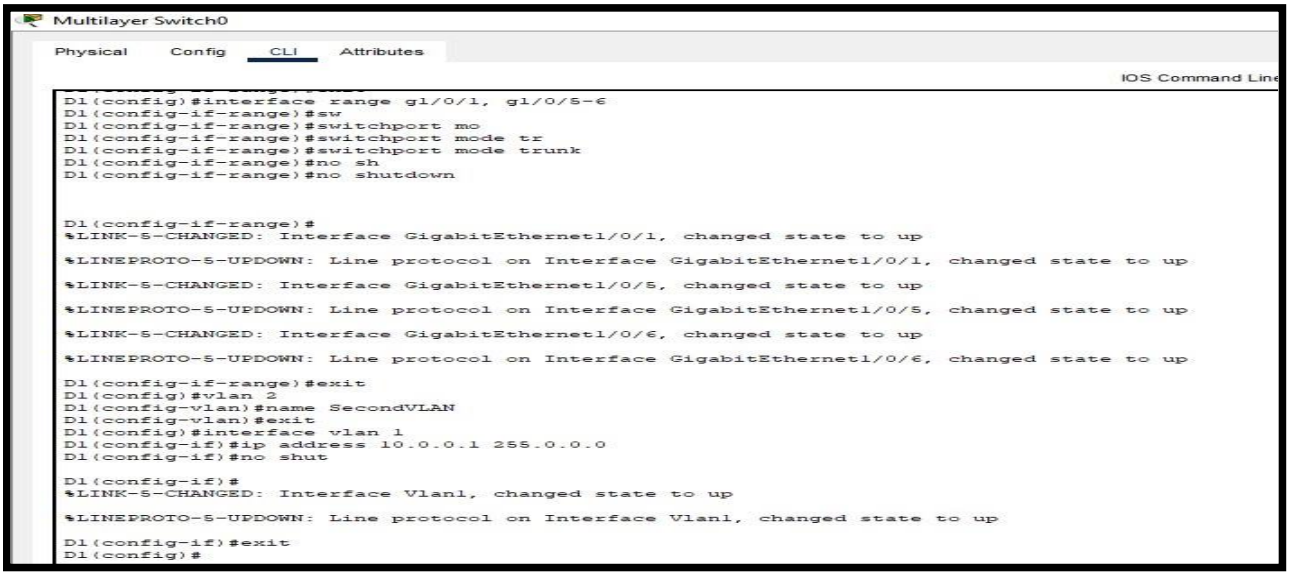
interface range g1/0/1, g1/0/5-6 switchport mode
trunk no shutdown exit vlan 2

name SecondVLAN

exit interface vlan 1 ip address 10.0.0.1 255.0.0.0
no shut exit







Switch D2 hostname D2

banner motd # D2, STP Topology Change and RSTP Lab # spanning-tree mode pvst

interface range g1/0/1-24, g1/1/1-4, g0/0 shutdown

exit interface range g1/0/1, g1/0/5-6 switchport

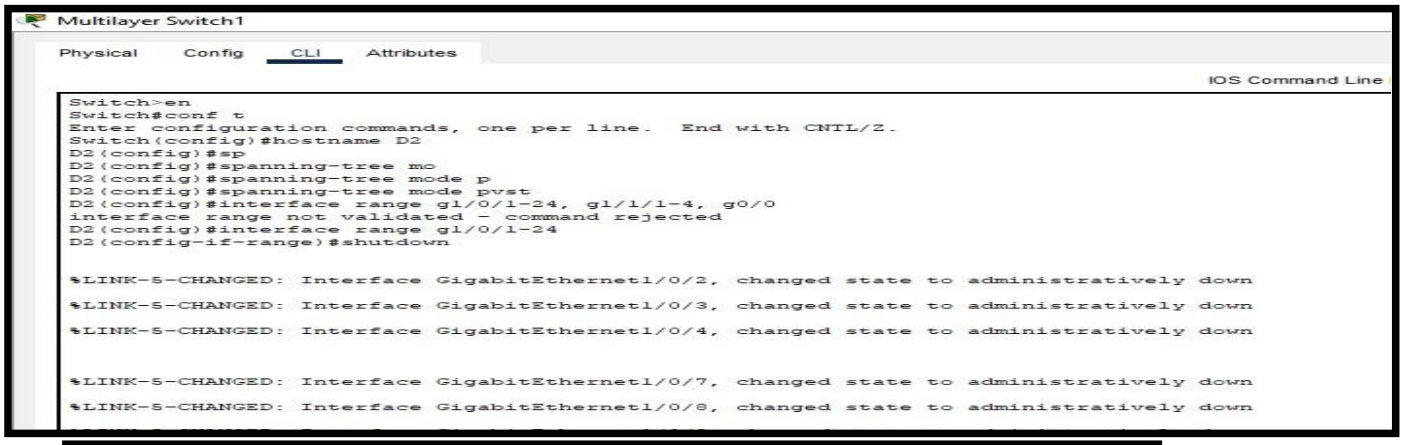
mode trunk no shutdown exit vlan 2

name SecondVLAN

exit interface vlan 1 ip address 10.0.0.2

255.0.0.0 no shut

exit



Switch A1

hostname A1

banner motd # A1, STP Topology Change and RSTP Lab # spanning-tree mode

pvst line con 0 exec-timeout 0 0 logging synchronous exit

interface range f0/1-24, g0/1-2 shutdown

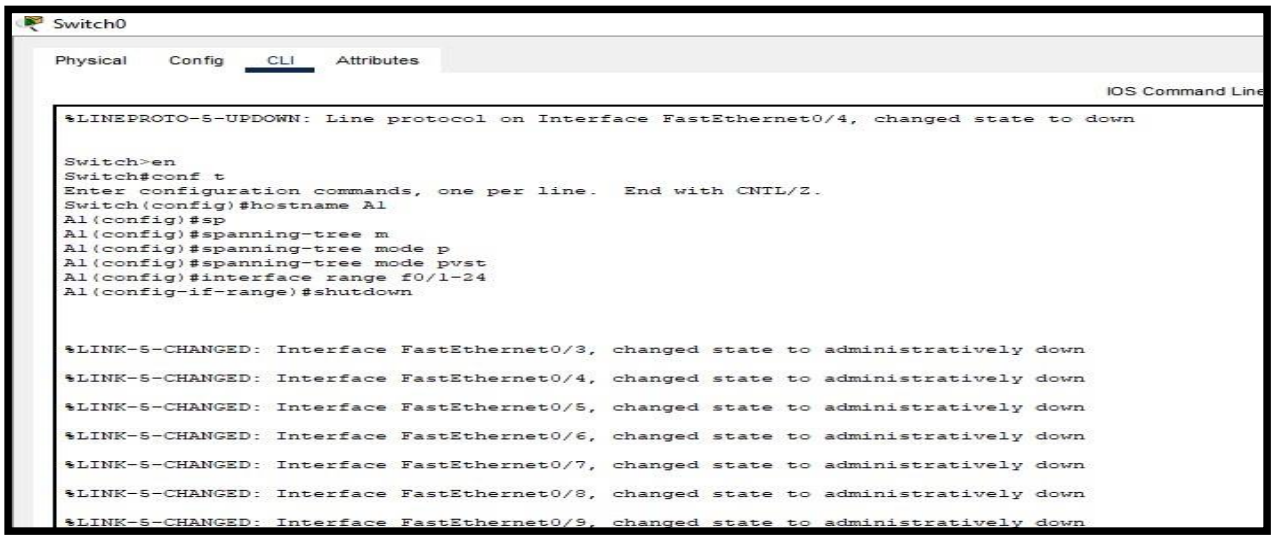
exit interface range f0/1-4 switchport

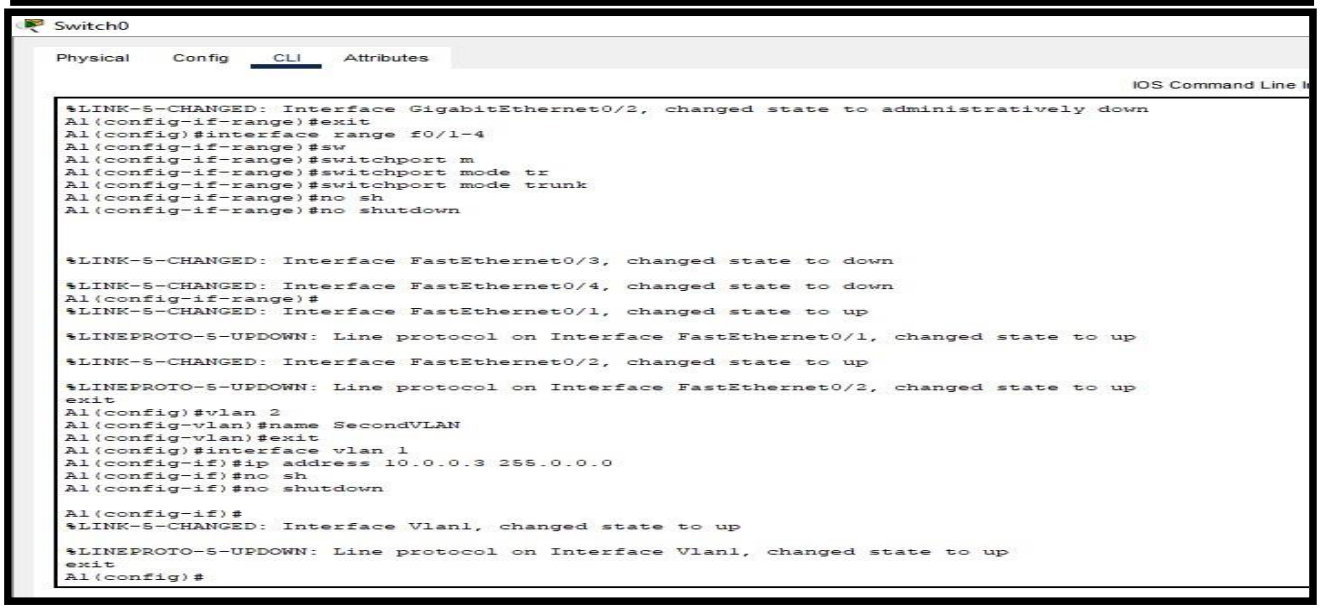
mode trunk no shutdown

exit vlan 2

name SecondVLAN

exit





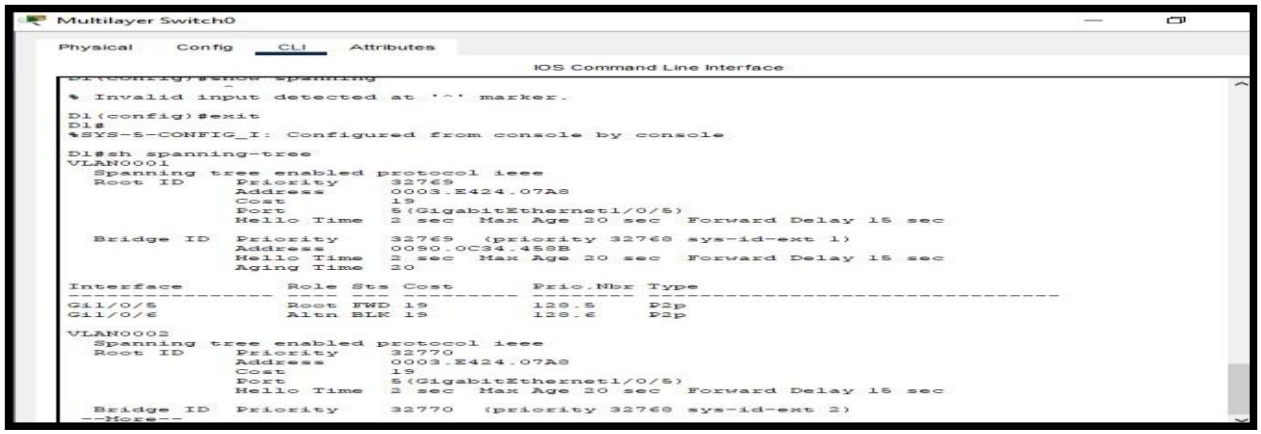
b. Set the clock on each switch to UTC time.

c. Save the running configuration to startup-config.

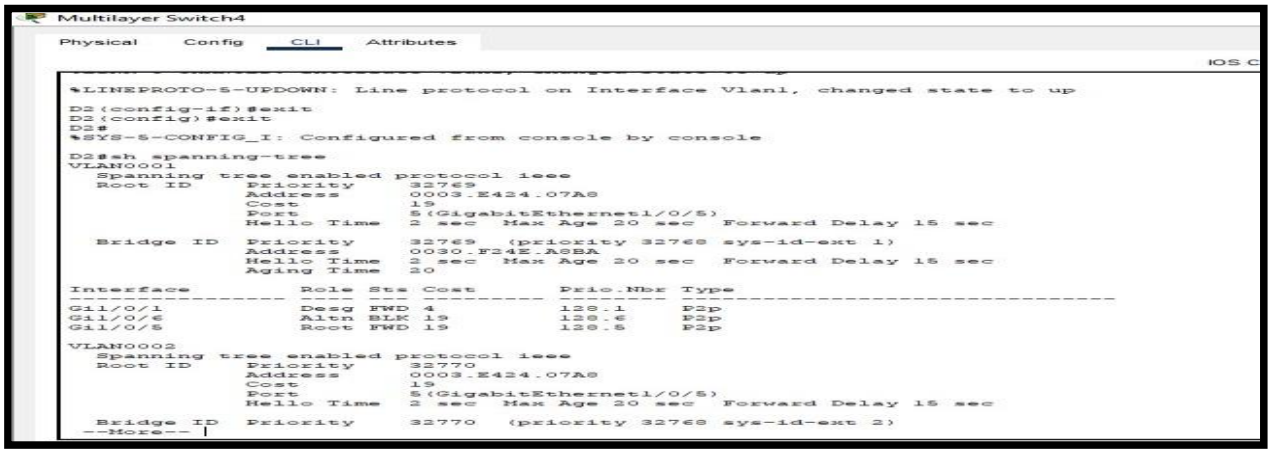
Part 2: Discover the Default Spanning Tree

Your switches have been configured and interfaces have been enabled, and the Spanning Tree Protocol, operational by default, has already converged onto a loop-free logical network. In this part of the lab, we will discover what that default spanning tree looks like and evaluate why it converged the way it did

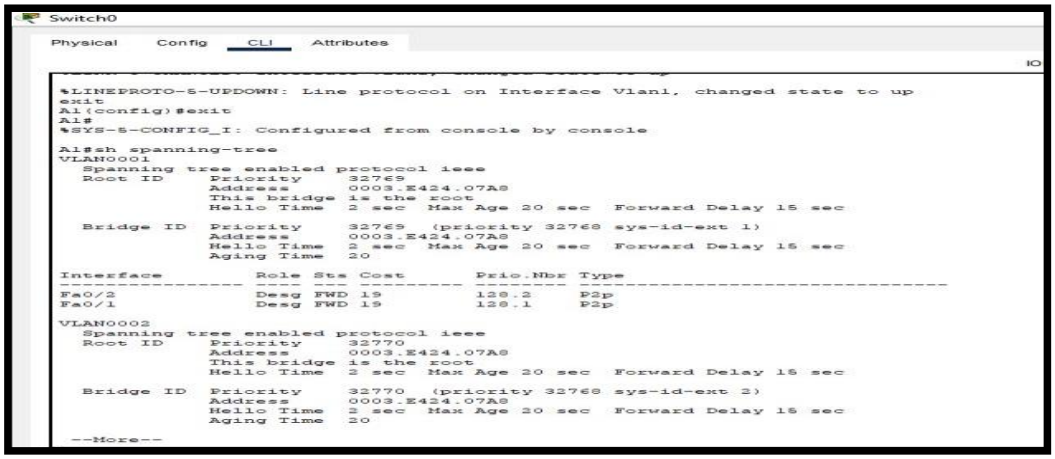
 # show spanning-tree Switch D1



Switch D2

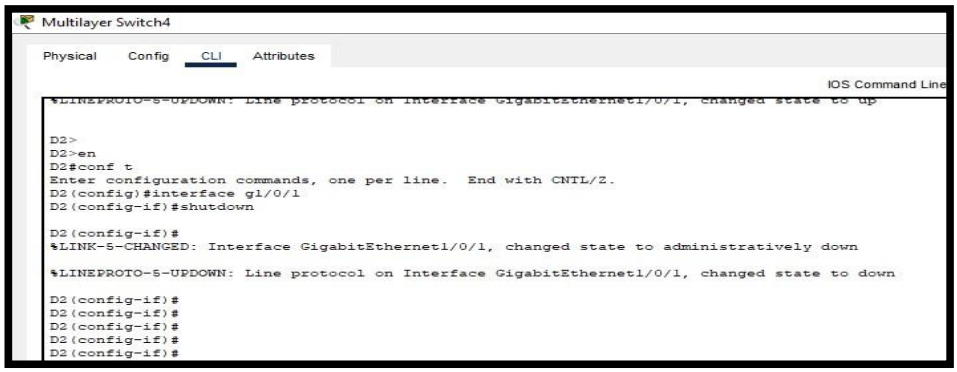


Switch A1

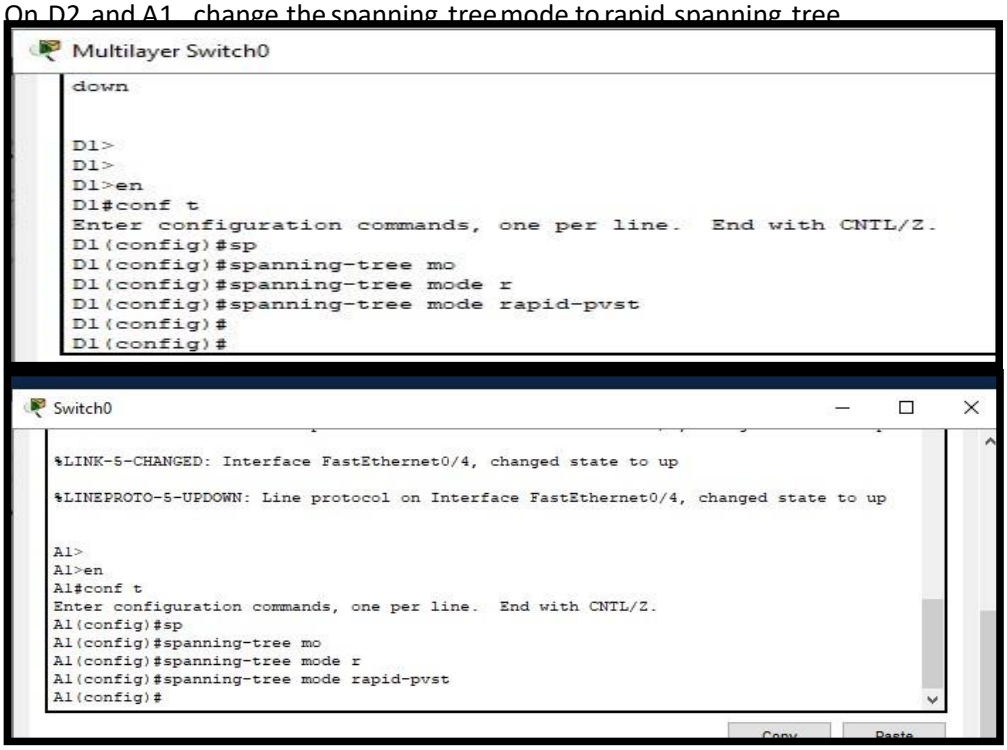


Part 3: Implement and Observe Rapid Spanning Tree Protocol

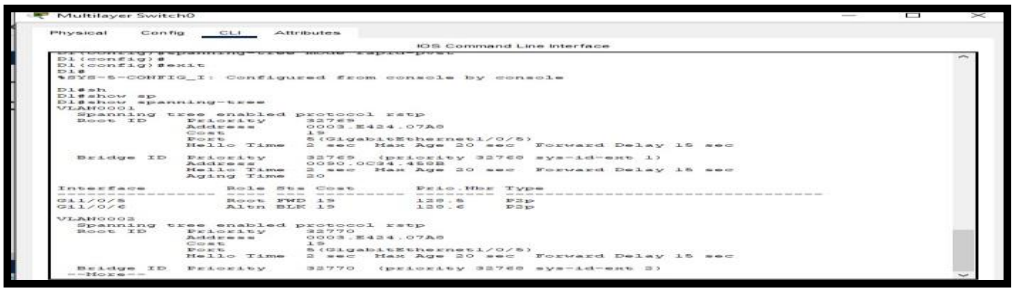
a. On D2, issue the debug spanning-tree events command, and then issue the shutdown command for interface g1/0/1 and observe the output.



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



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



A1 was the last switch that was configured for RSTP. As you can see, interface

VLAN1 was only down for 0.048 seconds. This is the “rapid” in rapid spanning tree.

PRACTICAL 7

Aim : Packet Tracer – Configure EtherChannel

❖ What is EtherChannel ?

→

1. EtherChannel, also known as port channel or link aggregation, is a technology used in computer networking to combine multiple physical Ethernet links into a single logical link. This logical link provides increased bandwidth, fault tolerance, and load balancing.
2. In an EtherChannel configuration, multiple physical links between two network devices, such as switches or routers, are bundled together to form a single logical link.
3. The main purposes of EtherChannel include:
   - Increased Bandwidth
   - Fault Tolerance
   - Load Balancing

❖ Objectives :

Part 1: Configure Basic Switch Settings

Part 2: Configure an EtherChannel with Cisco PAgP

Part 3: Configure an 802.3ad LACP EtherChannel

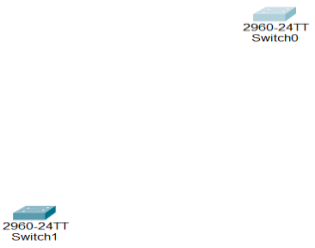
Part 4: Configure a Redundant EtherChannel Link

❖ Port Channel Table:

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

❖ Configure Basic Switch Settings :

Switch Model : 2960-24TT



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

```
%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22, changed state to up


Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname S1
S1(config)#
```

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

Open configuration window

S1# show interfaces | include Ethernet

S1# show interface status

S1# show interfaces trunk

```
S1#show inte
S1#show interfaces | in
S1#show interfaces | include Eth
S1#show interfaces | include Ether
S1#show interfaces | include Ethernet
FastEthernet0/1 is down, line protocol is down (disabled)
FastEthernet0/2 is down, line protocol is down (disabled)
FastEthernet0/3 is down, line protocol is down (disabled)
FastEthernet0/4 is down, line protocol is down (disabled)
FastEthernet0/5 is down, line protocol is down (disabled)
FastEthernet0/6 is down, line protocol is down (disabled)
FastEthernet0/7 is down, line protocol is down (disabled)
FastEthernet0/8 is down, line protocol is down (disabled)
FastEthernet0/9 is down, line protocol is down (disabled)
FastEthernet0/10 is down, line protocol is down (disabled)
FastEthernet0/11 is down, line protocol is down (disabled)
FastEthernet0/12 is down, line protocol is down (disabled)
FastEthernet0/13 is down, line protocol is down (disabled)
FastEthernet0/14 is down, line protocol is down (disabled)
FastEthernet0/15 is down, line protocol is down (disabled)
FastEthernet0/16 is down, line protocol is down (disabled)
FastEthernet0/17 is down, line protocol is down (disabled)
FastEthernet0/18 is down, line protocol is down (disabled)
FastEthernet0/19 is down, line protocol is down (disabled)
FastEthernet0/20 is down, line protocol is down (disabled)
FastEthernet0/21 is up, line protocol is up (connected)
FastEthernet0/22 is up, line protocol is up (connected)
FastEthernet0/23 is down, line protocol is down (disabled)
FastEthernet0/24 is down, line protocol is down (disabled)
GigabitEthernet0/1 is up, line protocol is up (connected)
GigabitEthernet0/2 is up, line protocol is up (connected)
S1#
```

```
GigabitEthernet0/2 is up, line protocol is up (connected)
S1#show inter
S1#show interfaces stat
S1#show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1                        notconnect   1          auto    auto  10/100BaseTX
Fa0/2                        notconnect   1          auto    auto  10/100BaseTX
Fa0/3                        notconnect   1          auto    auto  10/100BaseTX
Fa0/4                        notconnect   1          auto    auto  10/100BaseTX
Fa0/5                        notconnect   1          auto    auto  10/100BaseTX
Fa0/6                        notconnect   1          auto    auto  10/100BaseTX
Fa0/7                        notconnect   1          auto    auto  10/100BaseTX
Fa0/8                        notconnect   1          auto    auto  10/100BaseTX
Fa0/9                        notconnect   1          auto    auto  10/100BaseTX
Fa0/10                       notconnect   1          auto    auto  10/100BaseTX
Fa0/11                       notconnect   1          auto    auto  10/100BaseTX
Fa0/12                       notconnect   1          auto    auto  10/100BaseTX
Fa0/13                       notconnect   1          auto    auto  10/100BaseTX
Fa0/14                       notconnect   1          auto    auto  10/100BaseTX
Fa0/15                       notconnect   1          auto    auto  10/100BaseTX
Fa0/16                       notconnect   1          auto    auto  10/100BaseTX
Fa0/17                       notconnect   1          auto    auto  10/100BaseTX
Fa0/18                       notconnect   1          auto    auto  10/100BaseTX
Fa0/19                       notconnect   1          auto    auto  10/100BaseTX
Fa0/20                       notconnect   1          auto    auto  10/100BaseTX
Fa0/21                       connected    1          auto    auto  10/100BaseTX
Fa0/22                       connected    1          auto    auto  10/100BaseTX
Fa0/23                       notconnect   1          auto    auto  10/100BaseTX
Fa0/24                       notconnect   1          auto    auto  10/100BaseTX
Gig0/1                       connected    1          auto    auto  10/100BaseTX
Gig0/2                       connected    1          auto    auto  10/100BaseTX

S1#
```

```
Gig0/2                       connected    1          auto    auto  10/100BaseTX

S1#show interfaces trunk


S1#show interfaces trunk
```

❖ Configure an EtherChannel with Cisco PAgP

Step 1: Configure Port Channel 1.

S1 :

interface range f0/21 – 22
shutdown
channel-group 1 mode desirable
no shutdown

S2 :

interface range f0/21 – 22
shutdown
channel-group 1 mode desirable
no shutdown

IOS Command Li

```
                                    ^
% Invalid input detected at '^' marker.

S1(config)#interface range f0/21 - 22
S1(config-if-range)#shutdown

S1(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21, changed state to down

%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22, changed state to down

S1(config-if-range)#channel
S1(config-if-range)#channel-gro
S1(config-if-range)#channel-group 1 ?
  mode  Etherchannel Mode of the interface
S1(config-if-range)#channel-group 1 mo
S1(config-if-range)#channel-group 1 mode ?
  active      Enable LACP unconditionally
  auto        Enable PAgP only if a PAgP device is detected
  desirable   Enable PAgP unconditionally
  on          Enable Etherchannel only
  passive     Enable LACP only if a LACP device is detected
S1(config-if-range)#channel-group 1 mode de
S1(config-if-range)#channel-group 1 mode desirable
S1(config-if-range)#
Creating a port-channel interface Port-channel 1

S1(config-if-range)#no shutdown

S1(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22, changed state to up
```

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname S3
S3(config)#interface range f0/21 - 22
S3(config-if-range)#shutdown

S3(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21, changed state to down

%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22, changed state to down

S3(config-if-range)#channel-group 1 mode desirable
S3(config-if-range)#
Creating a port-channel interface Port-channel 1

S3(config-if-range)#no shutdown

S3(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22, changed state to up

%LINK-5-CHANGED: Interface Port-channel1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed state to up

S3(config-if-range)#
```

S1 :
interface port-channel 1
switchport mode trunk

S3:
interface port-channel 1
switchport mode trunk

```
%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22, changed state to down

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

S3(config-if)#
```

```
S1>en
S1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#interface port-channel 1
S1(config-if)#switchport mode trunk

S1(config-if)#
```

Step 2: Verify Port Channel 1 status.

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

show etherchannel summary

```
S1#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

1      Po1(SD)           PAgP   Fa0/21(D) Fa0/22(D)
S1#
```

Ctrl+F6 to exit CLI focus

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

Part 3: Configure an 802.3ad LACP EtherChannel :

Step 1: Configure Port Channel 2.

S1 :
interface range g0/1 – 2
shutdown
channel-group 2 mode active

no shutdown
interface port-channel 2
switchport mode trunk

S2 :
interface range g0/1 – 2
shutdown
channel-group 2 mode active
no shutdown
interface port-channel 2
switchport mode trunk

```
S1(config)#interface range g0/1 - 2
S1(config-if-range)#shutdown


S1(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to down

S1(config-if-range)#channel-group 2 mode active
S1(config-if-range)#
Creating a port-channel interface Port-channel 2

S1(config-if-range)#no shutdown


S1(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up

S1(config-if-range)#interface port-channel 2
```

```
S2#en
S2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S2(config)#interface range g0/1 - 2
S2(config-if-range)#shutdown
S2(config-if-range)#channel-group 2 mode active
S2(config-if-range)#no shutdown


S2(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up

%LINK-5-CHANGED: Interface Port-channel2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel2, changed state to up

S2(config-if-range)#interface port-channel 2
S2(config-if)#interface port-channel 2
S2(config-if)#
```

Ctrl+F6 to exit CLI focus

Step 2: Verify Port Channel 2 status.

S1 : show etherchannel summary

```
S1#
%SYS-5-CONFIG_I: Configured from console by console

S1#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

1      Po1(SD)           PAgP    Fa0/21(D)  Fa0/22(D)
2      Po2(SU)           LACP    Gig0/1(P)  Gig0/2(P)
S1#
```

Part 4: Configure a Redundant EtherChannel Link

Step 1: Configure Port Channel 3.

S2 :
interface range f0/23 – 24
shutdown
channel-group 3 mode passive
no shutdown
interface port-channel 3
switchport mode trunk

```
S2(config-if)#interface port-channel 2
S2(config-if)#exit
S2(config)#interface range f0/23 - 24
S2(config-if-range)#channel-group 3 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected
S2(config-if-range)#shutdown

S2(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/23, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23, changed state to down

%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to down

S2(config-if-range)#channel-group 3 mode passive
S2(config-if-range)#
Creating a port-channel interface Port-channel 3

S2(config-if-range)#no shutdown

S2(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/23, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to up

S2(config-if-range)#interface port-channel 3
S2(config-if)#switchport mode trunk
S2(config-if)#
```

Step 2: Verify Port Channel 3 status.

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

S2 : show etherchannel summary

```
S2#
%SYS-5-CONFIG_I: Configured from console by console

S2#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

2      Po2(SU)         LACP      Gig0/1(P) Gig0/2(P)
3      Po3(SD)         LACP      Fa0/23(I) Fa0/24(I)
```

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

S1 : show spanning-tree active

```
S1#show spanning-tree active
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     0000.0C78.87D9
              Cost        22
              Port        28(Port-channel2)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     000C.850B.D25D
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po2              Root FWD 3         128.28   Shr
```

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

S1 : spanning-tree vlan 1 root primary
or
S1 : spanning-tree vlan 1 priority 24576

```
S1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#spanning-tree vlan 1 root primary
```

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

❖ **EtherChannel Configuration Guidelines and Restrictions**

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

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

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

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

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

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

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

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

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

Valid values are:
active LACP is enabled unconditionally
passive LACP is enabled only if another LACP-capable device is connected.
desirable PAgP is enabled unconditionally
auto PAgP is enabled only if another PAgP-capable device is connected.
on EtherChannel is enabled, but without either LACP or PAgP.

9) LAN ports can form an EtherChannel using PAgP if the modes are compatible. Compatible PAgP modes are:
desirable => desirable
desirable => auto
If both interfaces are in auto mode, an Etherchannel cannot form.

10) LAN ports can form an EtherChannel using LACP if the modes are compatible. Compatible LACP modes are:
active => active
active => passive
If both interfaces are in passive mode, an EtherChannel cannot form using LACP.

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

PRACTICAL 8

Aim: Implement Single-Area OSPFv2

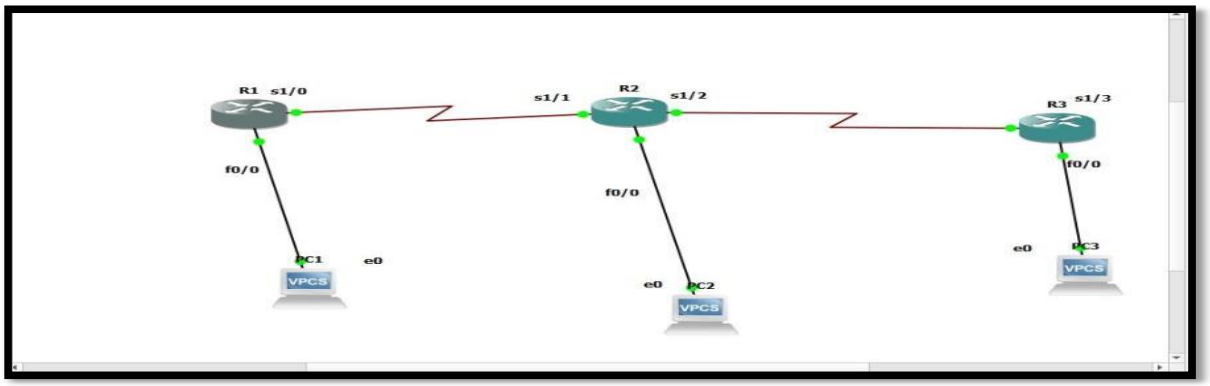
What is Open Shortest Path First (OSPF)?

- The OSPF stands for **Open Shortest Path First**. It is a widely used and supported routing protocol. It is an intra-domain protocol, which means that it is used within an area or a network.
- It is an interior gateway protocol that has been designed within a single autonomous system. It is based on a link-state routing algorithm in which each router contains the information of every domain, and based on this information, it determines the shortest path.
- The goal of routing is to learn routes.
- The OSPF achieves by learning about every router and subnet within the entire network. Every router contains the same information about the network.

1. Implement Single-Area OSPFv2:

Step 1: Assess the requirements for the network.

- For the network, I will be using 3 Routers and 3 PCs. **Step 2: Deign the topology of the network.**



Step 3: Build and Configure the network.

- Configuring Router 1:

```
R1#confi t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 192.168.1.50 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:05:59.915: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:06:00.915: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit

R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface serial  0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:08:34.851: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:08:35.851: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R1(config-if)#exit
```

- Configuring Router 2:

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface serial 0/0
R2(config-if)#ip address 10.1.1.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
```

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface serial 0/1
R2(config-if)#ip address 11.1.1.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#no shutdown
*Mar  1 00:19:45.619: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:19:46.619: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
R2(config-if)#exit
R2(config)#end
R2#
```

- Configuring Router 3:

```
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 192.168.3.50 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#
*Mar  1 00:09:02.035: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:09:03.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config-if)#exit
R3(config)#end
```

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface serial  0/0
R3(config-if)#exit
R3(config)#interface serial  0/1
R3(config-if)#no shutdown
R3(config-if)#
*Mar  1 00:26:38.571: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:26:39.571: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
R3(config-if)#exit
R3(config)#end
```

```
R3#sh ip int br
*Mar  1 00:26:48.799: %SYS-5-CONFIG_I: Configured from console by console
R3#sh ip int bri
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.3.50    YES manual up                    up
Serial0/0                  unassigned      YES unset  administratively down down
FastEthernet0/1            unassigned      YES unset  administratively down down
Serial0/1                  unassigned      YES unset  up                    down
Serial0/2                  unassigned      YES unset  administratively down down
Serial0/3                  unassigned      YES unset  administratively down down
Serial0/4                  unassigned      YES unset  administratively down down
Serial0/5                  unassigned      YES unset  administratively down down
```

- Configure PC 1:

```
R2    R1    R3    PC1

Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file


PC1> ip 192.168.1.1/24 192.168.1.50
Checking for duplicate address...
PC1 : 192.168.1.1 255.255.255.0 gateway 192.168.1.50

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.1/24
GATEWAY     : 192.168.1.50
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10026
RHOST:PORT  : 127.0.0.1:10027
MTU:        : 1500
```

- Configure PC 2:

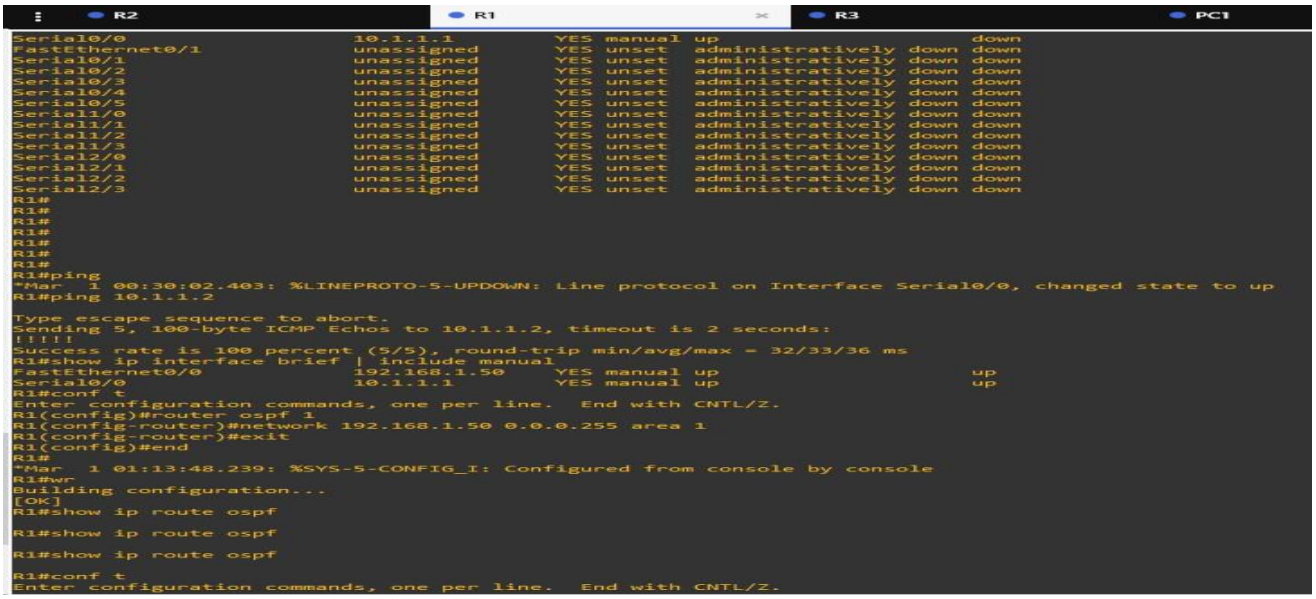
- Configure PC 3:

- Check whether all the interfaces are up.

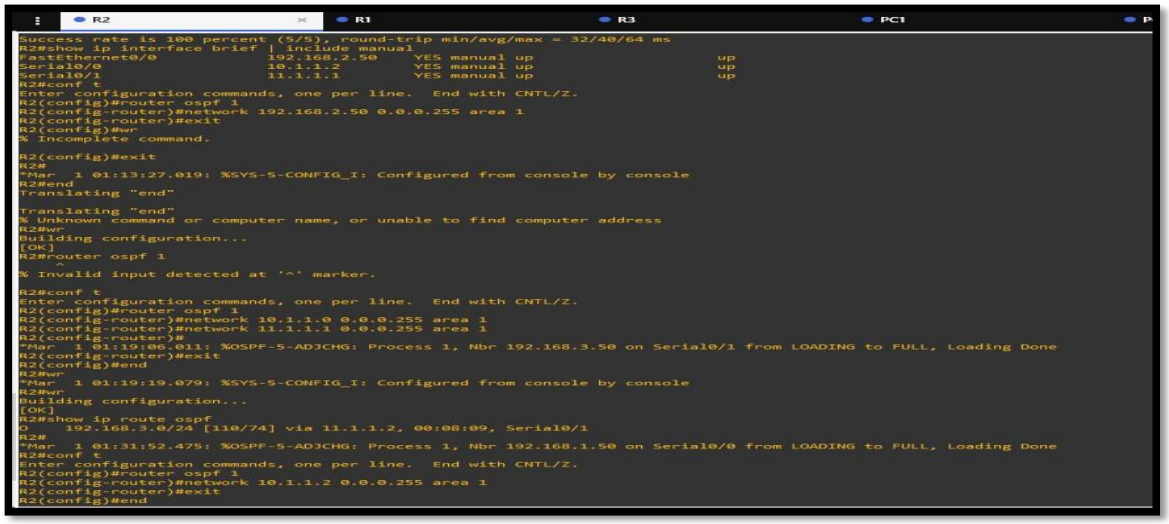
Step 4: Configure and test the Single – Area OSPFv2 for IPv4 on R1, R2 and R3.

- Configure Single – Area OSPFv2 on R1:
- Test the Single – Area OSPFv2 on R1:
- This shows the neighbour of Router 1 (which is only possible via OSPF)

- This will show the various routes used by OSPF. The ‘O’ at the start indicates a single area network.

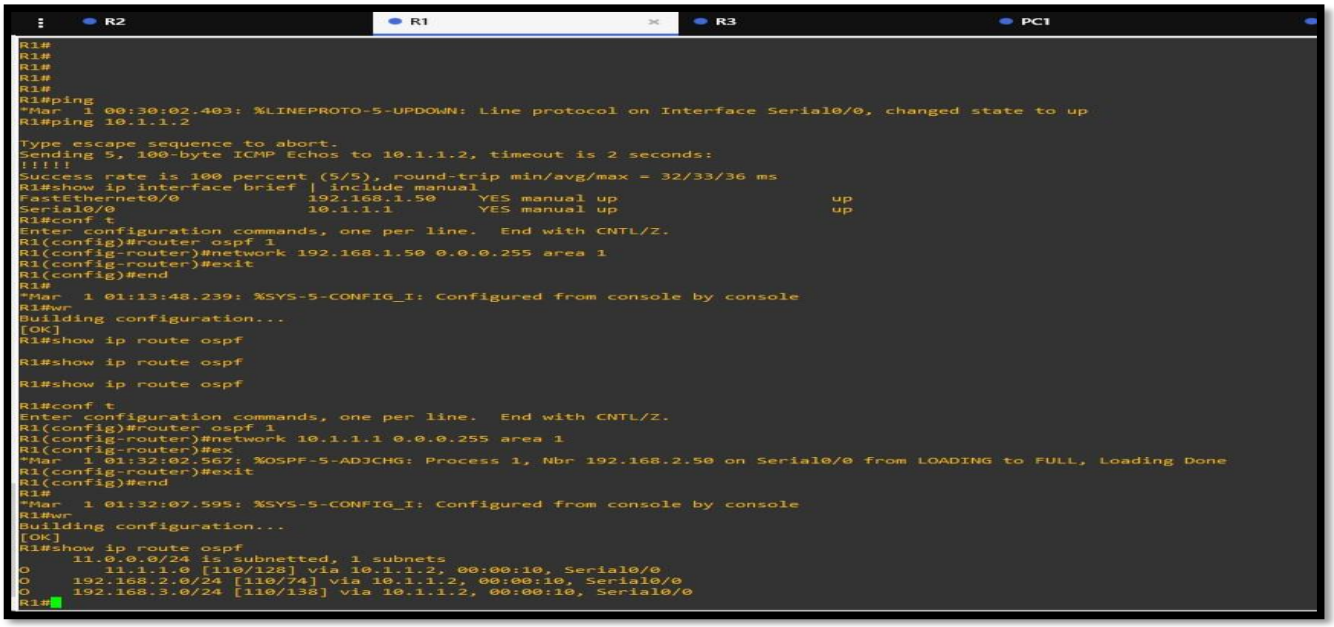


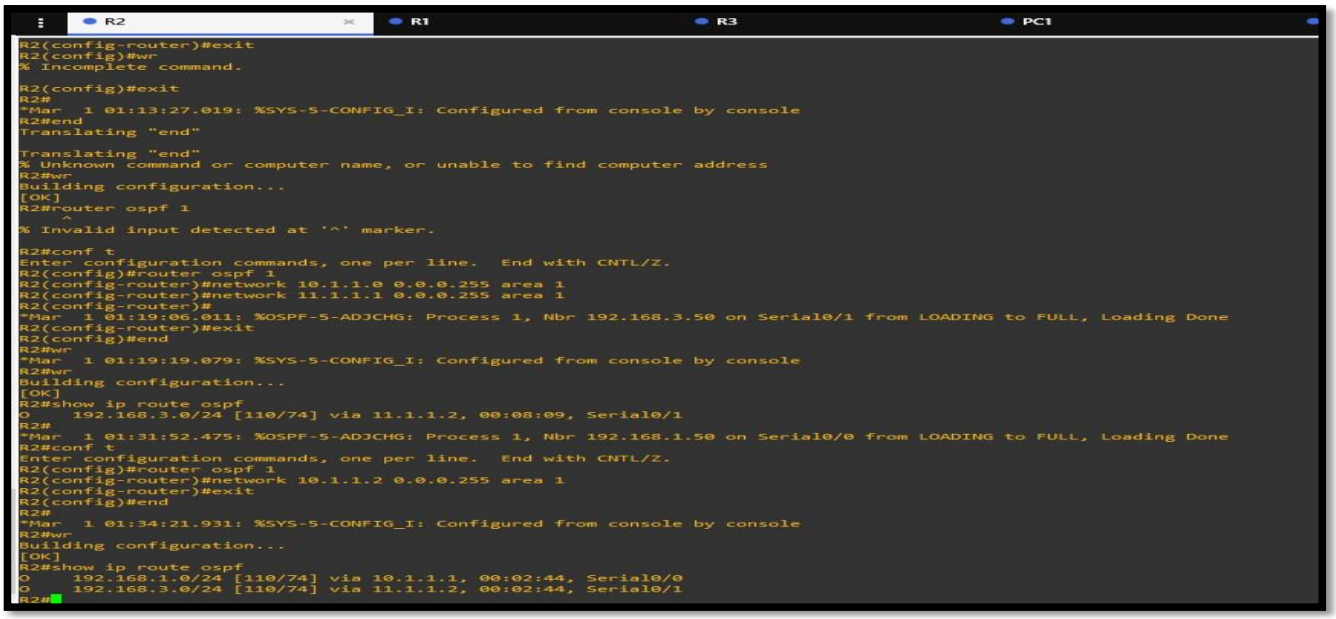
- Configure Single – Area OSPFv2 on R2:
- Test the Single – Area OSPFv2 on R1:
- This will show the various routes used by OSPF. The ‘O’ at the start indicates a single area network.
- This shows the neighbour of Router 2 (which is only possible via OSPF)

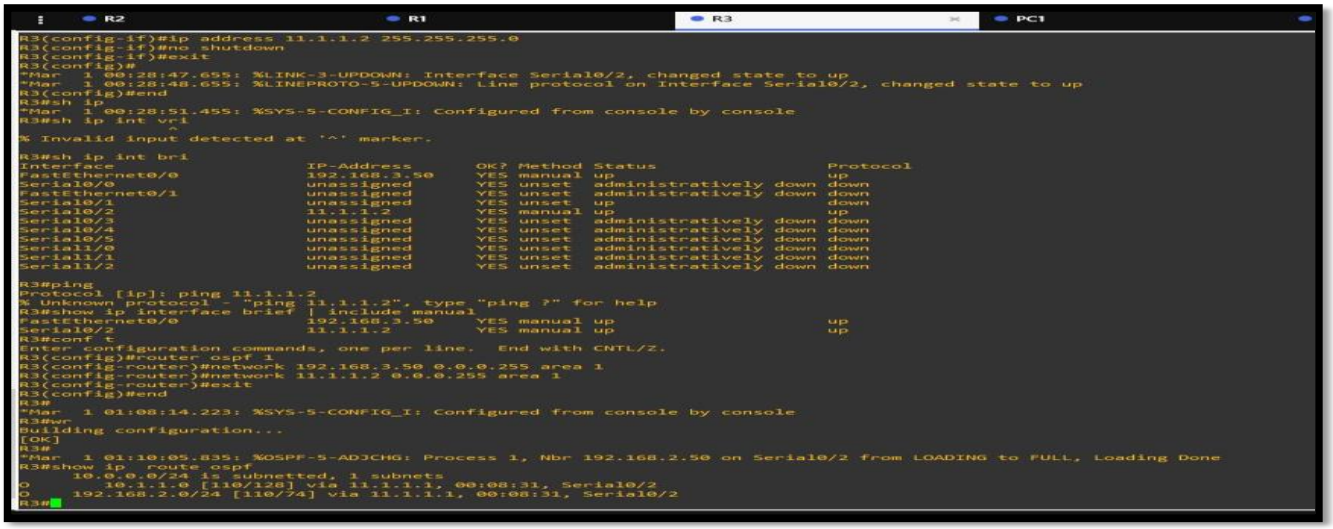


- Configure Single – Area OSPFv2 on R3:
- This shows the neighbour of Router 3 (which is only possible via OSPF).
- This will show the various routes used by OSPF. The ‘O’ at the start indicates a single area network.


R2
R1
R3
PC1
Serial1/0 unassigned YES unset administratively down down
Serial1/3 unassigned YES unset administratively down down
Serial1/2 unassigned YES unset administratively down down
--More--
*Mar 1 00:27:02.231: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to do
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface serial 0/2
R3(config-if)#ip address 11.1.1.2 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*Mar 1 00:28:47.655: %LINK-3-UPDOWN: Interface Serial0/2, changed state to up
*Mar 1 00:28:48.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/2, changed state to up
R3(config)#end
R3#sh ip
*Mar 1 00:28:51.455: %SYS-5-CONFIG_I: Configured from console by console
R3#sh ip int vri
^
% Invalid input detected at '^' marker.
R3#sh ip int bri
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.3.50 YES manual up up
Serial0/0 unassigned YES unset administratively down down
FastEthernet0/1 unassigned YES unset administratively down down
Serial0/1 unassigned YES unset up down
Serial0/2 11.1.1.2 YES manual up up
Serial0/3 unassigned YES unset administratively down down
Serial0/4 unassigned YES unset administratively down down
Serial0/5 unassigned YES unset administratively down down
Serial1/0 unassigned YES unset administratively down down
Serial1/1 unassigned YES unset administratively down down
Serial1/2 unassigned YES unset administratively down down
R3#ping
Protocol [ip]: ping 11.1.1.2
% Unknown protocol. "ping 11.1.1.2", type "ping ?" for help
R3#show ip interface brief | include manual
FastEthernet0/0 192.168.3.50 YES manual up up
Serial0/2 11.1.1.2 YES manual up up
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#network 192.168.3.50 0.0.0.255 area 1
R3(config-router)#network 11.1.1.2 0.0.0.255 area 1
R3(config-router)#exit
R3(config)#end
R3#
*Mar 1 01:08:14.223: %SYS-5-CONFIG_I: Configured from console by console
R3#wr


R2
R1
R3
PC1
R1#
R1#
R1#
R1#
R1#
R1#ping
*Mar 1 00:30:02.403: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R1#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
R1#show ip interface brief | include manual
FastEthernet0/0 192.168.1.50 YES manual up up
Serial0/0 10.1.1.1 YES manual up up
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.50 0.0.0.255 area 1
R1(config-router)#exit
R1(config)#end
R1#
*Mar 1 01:13:48.239: %SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#show ip route ospf
R1#show ip route ospf
R1#show ip route ospf
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 10.1.1.1 0.0.0.255 area 1
R1(config-router)#ex
*Mar 1 01:32:02.567: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.50 on Serial0/0 from LOADING to FULL, Loading Done
R1(config-router)#exit
R1(config)#end
R1#
*Mar 1 01:32:07.595: %SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#show ip route ospf
11.0.0.0/24 is subnetted, 1 subnets
O 11.1.1.0 [110/128] via 10.1.1.2, 00:00:10, Serial0/0
O 192.168.2.0/24 [110/74] via 10.1.1.2, 00:00:10, Serial0/0
O 192.168.3.0/24 [110/138] via 10.1.1.2, 00:00:10, Serial0/0
R1#

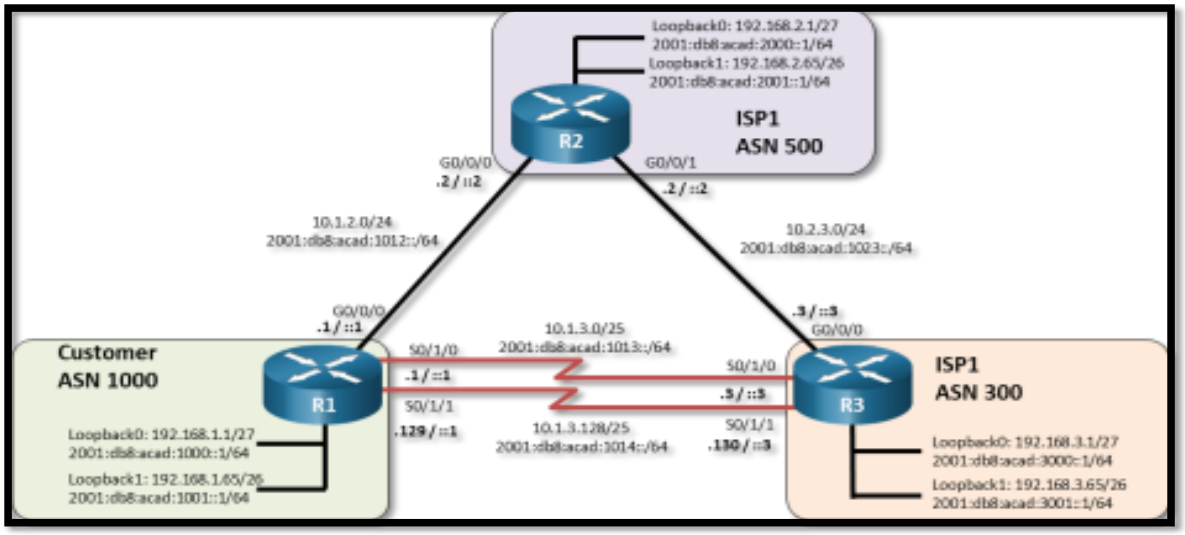

R2
R1
R3
PC1
R2(config-router)#exit
R2(config)#wr
% Incomplete command.
R2(config)#exit
R2#
*Mar 1 01:13:27.019: %SYS-5-CONFIG_I: Configured from console by console
R2#end
Translating "end"
Translating "end"
% Unknown command or computer name, or unable to find computer address
R2#wr
Building configuration...
[OK]
R2#router ospf 1
^
% Invalid input detected at '^' marker.
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#network 10.1.1.0 0.0.0.255 area 1
R2(config-router)#network 11.1.1.1 0.0.0.255 area 1
R2(config-router)#
*Mar 1 01:19:06.011: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.3.50 on Serial0/1 from LOADING to FULL, Loading Done
R2(config-router)#exit
R2(config)#end
R2#wr
*Mar 1 01:19:19.079: %SYS-5-CONFIG_I: Configured from console by console
R2#wr
Building configuration...
[OK]
R2#show ip route ospf
O 192.168.3.0/24 [110/74] via 11.1.1.2, 00:08:09, Serial0/1
R2#
*Mar 1 01:31:52.475: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.50 on Serial0/0 from LOADING to FULL, Loading Done
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#network 10.1.1.2 0.0.0.255 area 1
R2(config-router)#exit
R2(config)#end
R2#
*Mar 1 01:34:21.931: %SYS-5-CONFIG_I: Configured from console by console
R2#wr
Building configuration...
[OK]
R2#show ip route ospf
O 192.168.1.0/24 [110/74] via 10.1.1.1, 00:02:44, Serial0/0
O 192.168.3.0/24 [110/74] via 11.1.1.2, 00:02:44, Serial0/1
R2#


R2
R1
R3
PC1
R3(config-if)#ip address 11.1.1.2 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*Mar 1 00:28:47.655: %LINK-3-UPDOWN: Interface Serial0/2, changed state to up
*Mar 1 00:28:48.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/2, changed state to up
R3(config)#end
R3#sh ip
*Mar 1 00:28:51.455: %SYS-5-CONFIG_I: Configured from console by console
R3#sh ip int vri
                ^
% Invalid input detected at '^' marker.
R3#sh ip int bri
Interface   IP-Address   OK? Method Status   Protocol
FastEthernet0/0   192.168.3.50   YES manual up   up
Serial0/0   unassigned   YES unset administratively down down
FastEthernet0/1   unassigned   YES unset administratively down down
Serial0/1   unassigned   YES unset up   down
Serial0/2   11.1.1.2   YES manual up   up
Serial0/3   unassigned   YES unset administratively down down
Serial0/4   unassigned   YES unset administratively down down
Serial0/5   unassigned   YES unset administratively down down
Serial1/0   unassigned   YES unset administratively down down
Serial1/1   unassigned   YES unset administratively down down
Serial1/2   unassigned   YES unset administratively down down
R3#ping
Protocol [ip]: ping 11.1.1.2
% Unknown protocol - "ping 11.1.1.2", type "ping ?" for help
R3#show ip interface brief | include manual
FastEthernet0/0   192.168.3.50   YES manual up   up
Serial0/2   11.1.1.2   YES manual up   up
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#network 192.168.3.50 0.0.0.255 area 1
R3(config-router)#network 11.1.1.2 0.0.0.255 area 1
R3(config-router)#exit
R3(config)#end
R3#
*Mar 1 01:08:14.223: %SYS-5-CONFIG_I: Configured from console by console
R3#wr
Building configuration...
[OK]
R3#
*Mar 1 01:10:05.835: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.50 on Serial0/2 from LOADING to FULL, Loading Done
R3#show ip route ospf
      10.0.0.0/24 is subnetted, 1 subnets
O   10.1.1.0 [110/128] via 11.1.1.1, 00:08:31, Serial0/2
O   192.168.2.0/24 [110/74] via 11.1.1.1, 00:08:31, Serial0/2
R3#

PRACTICAL 9

11.1.3 Lab - Implement MP-BGP

Topology



Addressing Table

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link- Local
R1	G0/0/0	10.1.2.1/24	2001:db8:acad:1012::1/64	fe80::1:1
	S0/1/0	10.1.3.1/25	2001:db8:acad:1013::1/64	fe80::1:2
	S0/1/1	10.1.3.129/25	2001:db8:acad:1014::1/64	fe80::1:3
	Loopback0	192.168.1.1/27	2001:db8:acad:1000::1/64	fe80::1:4
	Loopback1	192.168.1.65/26	2001:db8:acad:1001::1/64	fe80::1:5
R2	G0/0/0	10.1.2.2/24	2001:db8:acad:1012::2/64	fe80::2:1
	G0/0/1	10.2.3.2/24	2001:db8:acad:1023::2/64	fe80::2:2
	Loopback0	192.168.2.1/27	2001:db8:acad:2000::1/64	fe80::2:3
	Loopback1	192.168.2.65/26	2001:db8:acad:2001::1/64	fe80::2:4
R3	G0/0/0	10.2.3.3/24	2001:db8:acad:1023::3/64	fe80::3:1
	S0/1/0	10.1.3.3/25	2001:db8:acad:1013::3/64	fe80::3:2
	S0/1/1	10.1.3.130/25	2001:db8:acad:1014::3/64	fe80::3:3
	Loopback0	192.168.3.1/27	2001:db8:acad:3000::1/64	fe80::3:4
	Loopback1	192.168.3.65/26	2001:db8:acad:3001::1/64	fe80::3:5

Objectives

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

Part 2: Configure MP-BGP on all Routers
Part 3: Verify MP-BGP

Part 4: Configure and Verify IPv6 Summarization

What is BGP?

Border Gateway Protocol (BGP) refers to a gateway protocol that enables the internet to exchange routing information between autonomous systems (AS). As networks interact with each other, they need a way to communicate. This is accomplished through peering. BGP makes peering possible. Without it, networks would not be able to send and receive information with each other.

How Does BGP Work?

When you have a network router that connects to other networks, it does not know which network is the best one to send its data to. BGP takes into consideration all the different peering options a router has and chooses the one closest to where the router is. Each potential peer communicates the routing information it has and that gets stored within a routing information base (RIB). BGP can access this information and use it to choose the best peering option.

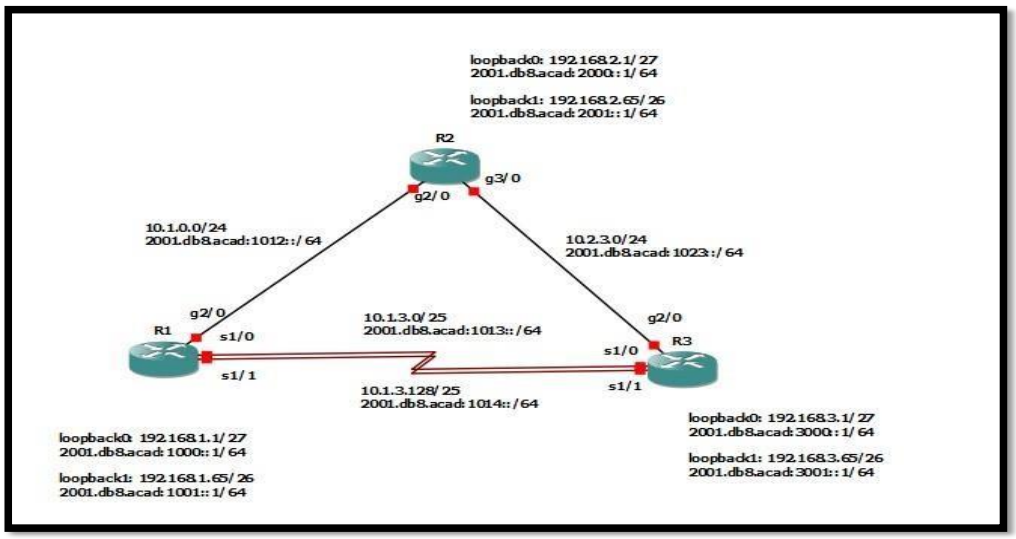
Instructions

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

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

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

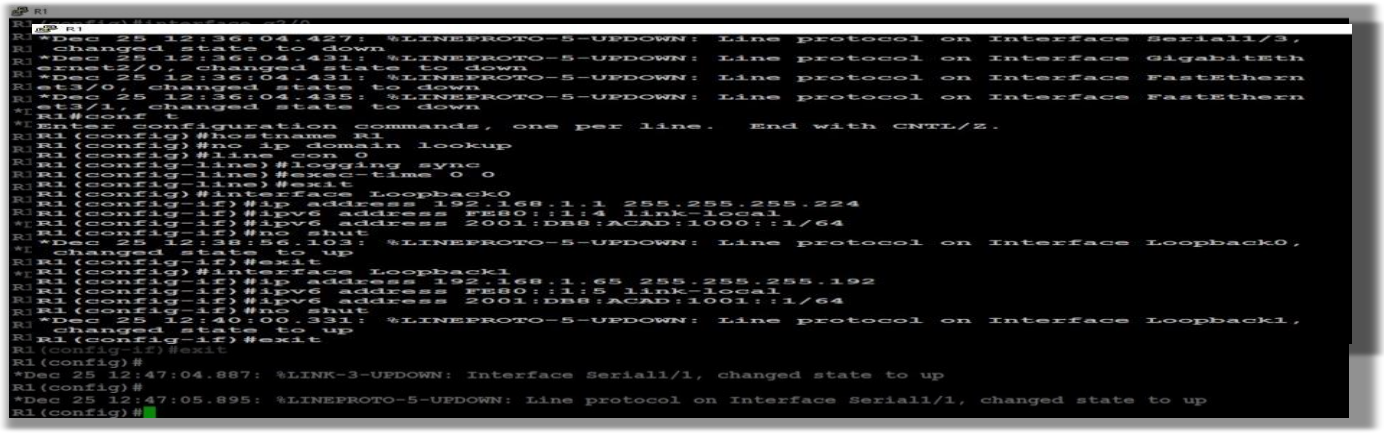
Attach the devices as shown in the topology diagram, and cable as necessary.



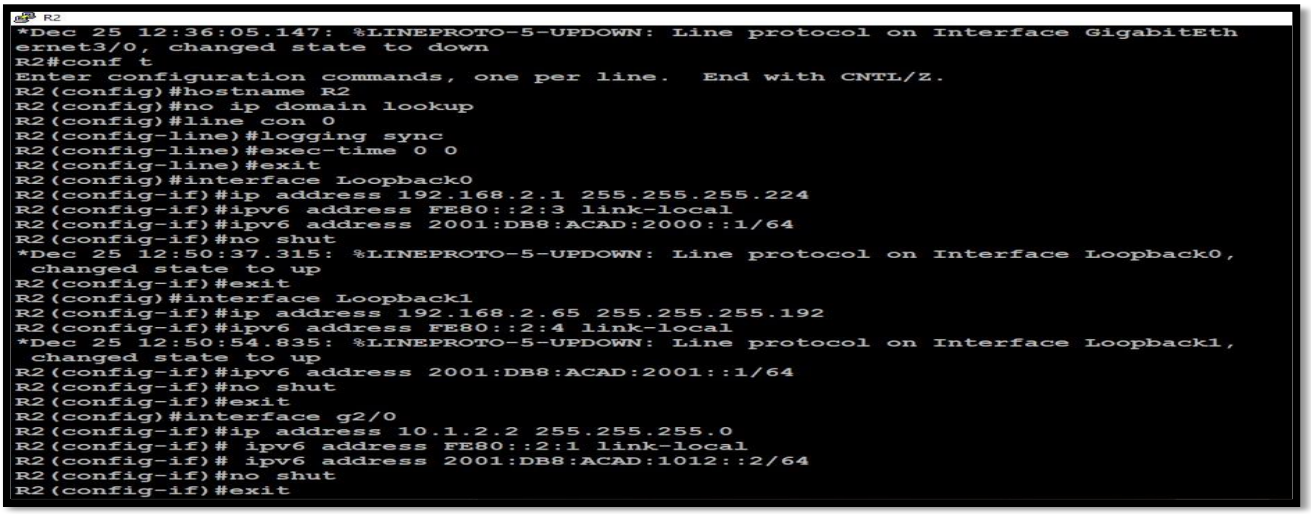
Step 2: Configure basic settings for each router.

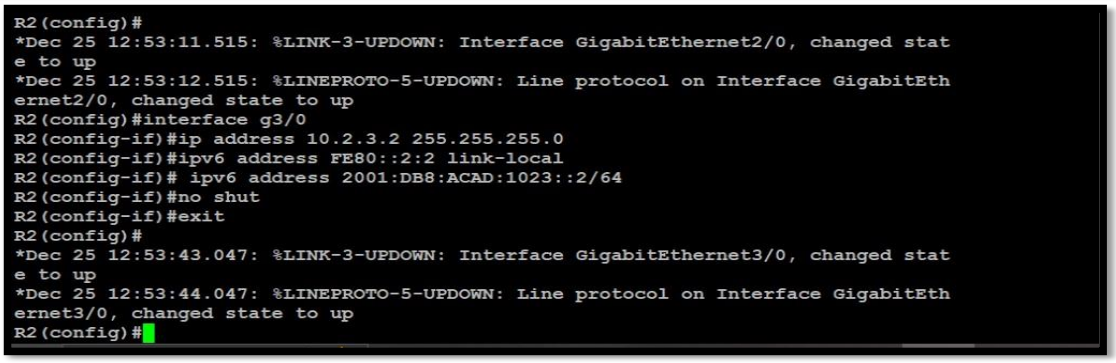
a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing. A command list for each router is listed below to perform initial configuration.

Router R1



Router R2





Router R3

```
R3
ernet2/0, changed state to down
*Dec 25 12:36:06.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et3/0, changed state to down
*Dec 25 12:36:06.287: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et3/1, changed state to down
R3#conf t
Enter configuration commands, one per line.  End with CNTL/z.
R3(config)#hostname R3
R3(config)#no ip domain lookup
R3(config)#line con 0
R3(config-line)#logging sync
R3(config-line)#exec-time 0 0
R3(config-line)#exit
R3(config)#interface Loopback0
R3(config-if)#ip address 192.168.3.1 255.255.255.224
R3(config-if)#ipv6 address FE80::3:4 link-local
R3(config-if)#ipv6 address 2001:DB8:ACAD:3000::1/64
R3(config-if)#no shut
*Dec 25 12:57:40.519: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
 changed state to up
R3(config-if)#exit
R3(config)#interface Loopback1
R3(config-if)#ip address 192.168.3.65 255.255.255.192
R3(config-if)#ipv6 address FE80::3:5 link-local
R3(config-if)#ipv6 address 2001:DB8:ACAD:3001::1/64
R3(config-if)#no shut
*Dec 25 12:57:57.135: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
 changed state to up
R3(config-if)#exit
R3(config)#interface g2/0
R3(config-if)#ip address 10.2.3.3 255.255.255.0
R3(config-if)#negotiation auto
```

```
R3
R3(config-if)#ip address 10.2.3.3 255.255.255.0
R3(config-if)#negotiation auto
R3(config-if)#ipv6 address FE80::3:1 link-local
R3(config-if)#ipv6 address 2001:DB8:ACAD:1023::3/64
R3(config-if)#no shut
R3(config-if)#exi
*Dec 25 13:01:37.139: %LINK-3-UPDOWN: Interface GigabitEthernet2/0, changed stat
e to up
*Dec 25 13:01:38.139: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEth
ernet2/0, changed state to up
R3(config-if)#exit
R3(config)#interface Serial1/0
R3(config-if)#ip address 10.1.3.3 255.255.255.128
R3(config-if)#ipv6 address FE80::3:2 link-local
R3(config-if)# ipv6 address 2001:DB8:ACAD:1013::3/64
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Dec 25 13:01:59.195: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R3(config)#
*Dec 25 13:02:00.203: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
 changed state to up
R3(config)#interface Serial1/1
R3(config-if)#ip address 10.1.3.130 255.255.255.128
R3(config-if)#ipv6 address FE80::3:3 link-local
R3(config-if)#ipv6 address 2001:DB8:ACAD:1014::3/64
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Dec 25 13:02:20.439: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R3(config)#
*Dec 25 13:02:21.447: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
```

```
R3
R3#sh ip int bri
Interface                  IP-Address      OK? Method Stat
FastEthernet0/0            unassigned      YES unset  admi
Serial1/0                  10.1.3.3        YES manual up
Serial1/1                  10.1.3.130      YES manual up
Serial1/2                  unassigned      YES unset  admi
Serial1/3                  unassigned      YES unset  admi
GigabitEthernet2/0         10.2.3.3        YES manual up
FastEthernet3/0            unassigned      YES unset  admi
FastEthernet3/1            unassigned      YES unset  admi
Loopback0                  192.168.3.1     YES manual up
Loopback1                  192.168.3.65    YES manual up
R3#
```

b. Save the running configuration to startup-config.

Part 2: Configure MP-BGP on all Routers

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

a. Enable IPv6 routing.

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

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

R1(config)# **router bgp 1000**

R1(config-router)# **bgp router-id 1.1.1.1**

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

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

R1(config-router)# **neighbor 10.1.3.130 remote-as 300**

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

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

R1(config-router)# **neighbor 2001:db8:acad:1014::3 remote-as 300**

e. Enter address family configuration mode for IPv4 and activate each of the IPv4 neighbors. R1(config-router)# **address-family ipv4 unicast**

R1(config-router-af)# **neighbor 10.1.2.2 activate**

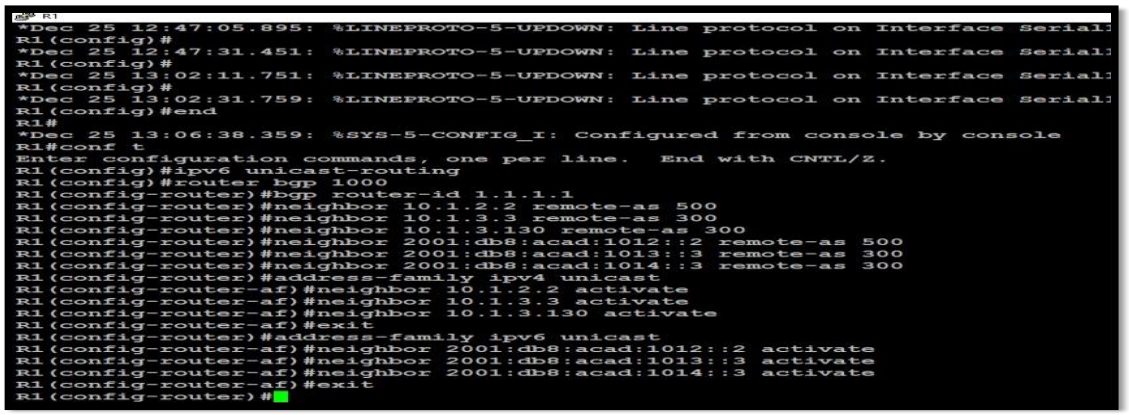
R1(config-router-af)# **neighbor 10.1.3.3 activate**

R1(config-router-af)# **neighbor 10.1.3.130 activate**

R1(config-router-af)# **exit**

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

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



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

a. Enable IPv6 routing.

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

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

R2(config)# **router bgp 500**

R2(config-router)# **bgp router-id 2.2.2.2**

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

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

d. Based on the topology diagram, configure all the designated IPv6 neighbors for R1.
R2(config-router)# **neighbor 2001:db8:acad:1012::1 remote-as 1000**
R2(config-router)# **neighbor 2001:db8:acad:1023::3 remote-as 300**

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

R2(config-router)# **address-family ipv4 unicast** R2(config-router-af)# **neighbor 10.1.2.1 activate**

R2(config-router-af)# **neighbor 10.2.3.3 activate**

R2(config-router-af)# **exit**

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

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

```
R2
R2(config-if)#ipv6 address FE80::2:2 link-local
R2(config-if)# ipv6 address 2001:DB8:ACAD:1023::2/64
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Dec 25 12:53:43.047: %LINK-3-UPDOWN: Interface GigabitEthernet3/0, changed stat
e to up
*Dec 25 12:53:44.047: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEth
ernet3/0, changed state to up
R2(config)#end
R2#
*Dec 25 13:06:42.779: %SYS-5-CONFIG_I: Configured from console by console
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#ipv6 unicast-routing
R2(config)#router bgp 500
R2(config-router)#bgp router-id 2.2.2.2
R2(config-router)#neighbor 10.1.2.1 remote-as 1000
R2(config-router)#neighbor 10.2.3.3 remote-as 300
R2(config-router)#neighbor 2001:db8:acad:1012::1 remote-as 1000
R2(config-router)#neighbor 2001:db8:acad:1023::3 remote-as 300
R2(config-router)#address-family ipv4 unicast
R2(config-router-af)#neighbor 10.1.2.1 activate
R2(config-router-af)#neighbor 10.2.3.3 activate
R2(config-router-af)#exit
R2(config-router)#address-family ipv6 unicast
R2(config-router-af)#neighbor 2001:db8:acad:1012::1 activate
R2(config-router-af)#neighbor 2001:db8:acad:1023::3 activate
R2(config-router-af)#exit
R2(config-router)#
*Dec 25 13:21:45.115: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1012::1 Up
R2(config-router)#
```

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

a. Enable IPv6 routing.

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

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

R3(config)# **router bgp 300**

R3(config-router)# **bgp router-id 3.3.3.3**

c. Based on the topology diagram, configure all the designated IPv4 neighbors for R1. R3(config-router)# **neighbor 10.2.3.2 remote-as 500**

R3(config-router)# **neighbor 10.1.3.1 remote-as 1000**

R3(config-router)# **neighbor 10.1.3.129 remote-as 1000**

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

R3(config-router)# **neighbor 2001:db8:acad:1023::2 remote-as 500**

R3(config-router)# **neighbor 2001:db8:acad:1013::1 remote-as 1000**

R3(config-router)# **neighbor 2001:db8:acad:1014::1 remote-as 1000**

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

R3(config-router)# **address-family ipv4 unicast**
R3(config-router-af)# **neighbor 10.1.3.1 activate**

R3(config-router-af)# **neighbor 10.1.3.129 activate**

R3(config-router-af)# **neighbor 10.2.3.2 activate**

R3(config-router-af)# **exit**

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

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

```
R3
FastEthernet3/0            unassigned      YES unset  administratively down down
FastEthernet3/1            unassigned      YES unset  administratively down down
Loopback0                  192.168.3.1     YES manual up                    up
Loopback1                  192.168.3.65    YES manual up                    up
R3#
R3#
R3#
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#ipv6 unicast-routing
R3(config)#router bgp 300
R3(config-router)#bgp router-id 3.3.3.3
R3(config-router)#neighbor 10.1.3.1 remote-as 1000
R3(config-router)#neighbor 10.1.3.129 remote-as 1000
R3(config-router)#neighbor 2001:db8:acad:1023::2 remote-as 500
R3(config-router)#neighbor 2001:db8:acad:1013::1 remote-as 1000
R3(config-router)#neighbor 2001:db8:acad:1014::1 remote-as 1000
R3(config-router)#address-family ipv4 unicast
R3(config-router-af)#neighbor 10.1.3.1 activate
R3(config-router-af)#neighbor 10.1.3.129 activate
R3(config-router-af)#neighbor 10.2.3.2 activate
% Specify remote-as or peer-group commands first
R3(config-router-af)#exit
R3(config-router)#address-family ipv6 unicast
R3(config-router-af)#neighbor 2001:db8:acad:1023::2 activate
R3(config-router-af)#neighbor 2001:db8:acad:1013::1 activate
R3(config-router-af)#neighbor 2001:db8:acad:1014::1 activate
R3(config-router-af)#exit
R3(config-router)#
R3(config-router)#
*Dec 25 13:25:50.155: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1023::2 Up
R3(config-router)#
```

Step 4: Advertise IPv4 and IPv6 prefixes on R1.

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

R1(config-router)# **address-family ipv4 unicast**

R1(config-router-af)# **network 192.168.1.0 mask 255.255.255.224**

R1(config-router-af)# **network 192.168.1.64 mask 255.255.255.192**

R1(config -router-af)# **exit**

```
R1
R1(config-router)#
*Dec 25 13:26:17.195: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1013::3 Up
R1(config-router)#address-family ipv4 unicast
R1(config-router-af)#network 192.168.1.0 mask 255.255.255.224
R1(config-router-af)#network 192.168.1.64 mask 255.255.255.192
R1(config-router-af)#exit
```

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

R1(config-router)# **address-family ipv6 unicast**

R1(config-router-af)# **network 2001:db8:acad:1000::/64**

R1(config-router-af)# **network 2001:db8:acad:1001::/64**

R1(config-router-af)# **exit**

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

Step 5: Advertise IPv4 and IPv6 prefixes on R2.

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

R2(config-router)# **address-family ipv4 unicast**

R2(config-router-af)# **network 192.168.2.0 mask 255.255.255.224**

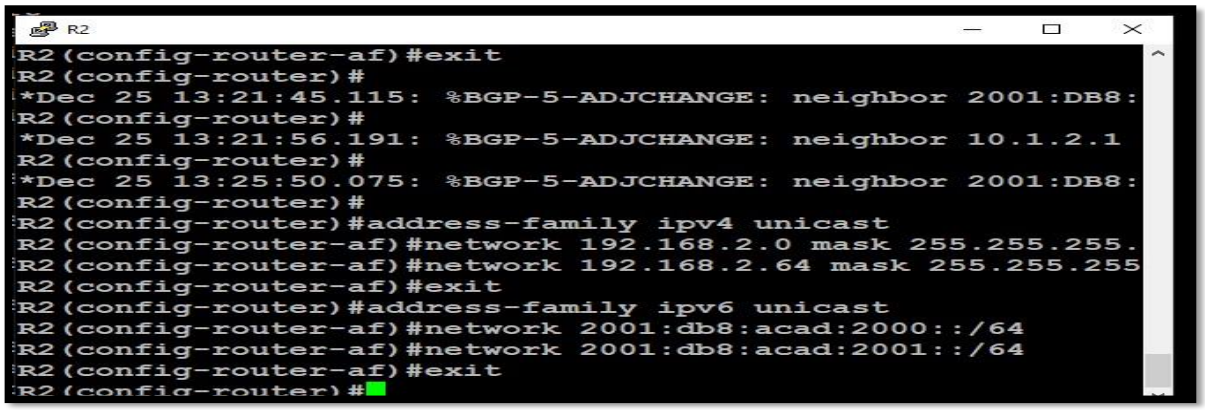
R2(config-router-af)# **network 192.168.2.64 mask**
255.255.255.192

R2(config-router-af)# **exit**

b. Enter address family configuration mode for IPv6 and advertise the IPv6 prefixes. R2(config-router)# **address-family ipv6 unicast**

R2(config-router-af)# **network 2001:db8:acad:2000::/64**

R2(config-router-af)# **network 2001:db8:acad:2001::/64** R2(config-router-af)# **exit**



Step 6: Advertise IPv4 and IPv6 prefixes on R3.

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

R3(config-router)# **address-family ipv4 unicast**

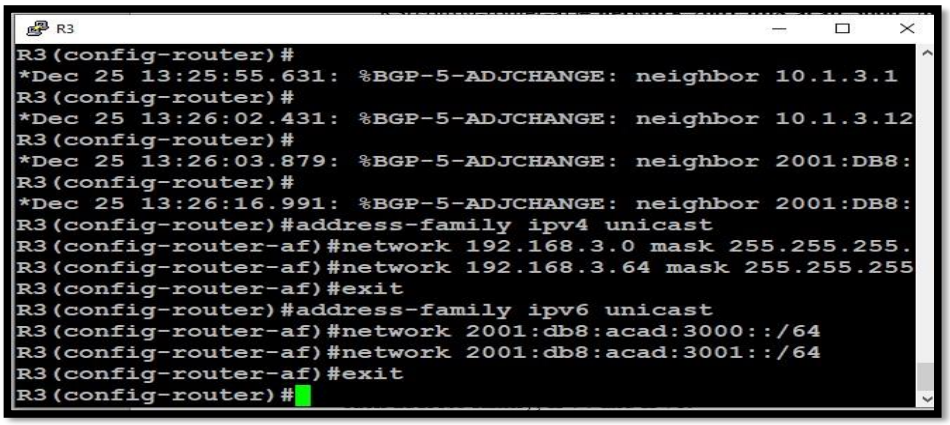
R3(config-router-af)# **network 192.168.3.0 mask 255.255.255.224** R3(config-router-af)# **network 192.168.3.64 mask 255.255.255.192**

R3(config-router-af)# **exit**

b. Enter address family configuration mode for IPv6 and advertise the IPv6 prefixes. R3(config-router)# **address-family ipv6 unicast**

R3(config-router-af)# **network 2001:db8:acad:3000::/64**

R3(config-router-af)# **network 2001:db8:acad:3001::/64** R3(config-router-af)# **exit**



Part 3: Verify MP-BGP

Step 1: Display detailed neighbor adjacency information.

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

R2# **show bgp all neighbors**

```
R2
R2(config-router-af)#exit
R2(config-router)#exit
R2(config)#exit
R2#
*Dec 25 13:40:18.339: %SYS-5-CONFIG_I: Configured from con
sole by console
R2#show bgp all neighbors
For address family: IPv4 Unicast
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:18:26
  Last read 00:00:52, last write 00:00:52, hold time is 18
0, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    New ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                5          5
    Keepalives:            16         16
    Route Refresh:          0          0
    Total:                 22         22
  Default minimum time between advertisement runs is 30 se
conds

 --More--
```

Step 2: Display summary neighbor adjacency information.

R2# **show bgp ipv4 unicast summary**

```
R2#show bgp ipv4 unicast summary
BGP router identifier 2.2.2.2, local AS number 500
BGP table version is 7, main routing table version 7
6 network entries using 792 bytes of memory
14 path entries using 728 bytes of memory
6/3 BGP path/bestpath attribute entries using 1008 bytes of memory
4 BGP AS-PATH entries using 96 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
Bitfield cache entries: current 1 (at peak 1) using 32 bytes of memory
BGP using 2656 total bytes of memory
BGP activity 12/0 prefixes, 24/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.2.1        4         1000      25      25        7    0    0 00:20:36        4
10.2.3.3        4          300       0       0        0    0    0 never    Active
2001:DB8:ACAD:1012::1
                4         1000      28      28        7    0    0 00:20:47        4
2001:DB8:ACAD:1023::3
                4          300      25      25        7    0    0 00:16:42        4
R2#
```

R2# **show bgp ipv6 unicast summary**

```
R2#show bgp ipv6 unicast summary
BGP router identifier 2.2.2.2, local AS number 500
BGP table version is 7, main routing table version 7
6 network entries using 936 bytes of memory
10 path entries using 760 bytes of memory
6/3 BGP path/bestpath attribute entries using 1008 bytes of memory
4 BGP AS-PATH entries using 96 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
Bitfield cache entries: current 1 (at peak 1) using 32 bytes of memory
BGP using 2832 total bytes of memory
BGP activity 12/0 prefixes, 24/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:ACAD:1012::1
                4         1000      28      28        7    0    0 00:21:43        4
2001:DB8:ACAD:1023::3
                4          300      26      25        7    0    0 00:17:38        4
R2#
```

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

Step 3: Verify BGP tables for IPv4 and IPv6.

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

R2# **show bgp ipv4 unicast**

```
R2#show bgp ipv4 unicast
BGP table version is 7, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  192.168.1.0/27   32.1.13.184                             0 300 1000 i
*                   32.1.13.184                0            0 1000 i
*>                  10.1.2.1                   0            0 1000 i
*  192.168.1.64/26  32.1.13.184                             0 300 1000 i
*                   32.1.13.184                0            0 1000 i
*>                  10.1.2.1                   0            0 1000 i
*> 192.168.2.0/27   0.0.0.0                    0        32768 i
*> 192.168.2.64/26  0.0.0.0                    0        32768 i
*  192.168.3.0/27   32.1.13.184                             0 1000 300 i
*>                  10.1.2.1                                 0 1000 300 i
*                   32.1.13.184                0            0 300 i
*  192.168.3.64/26  32.1.13.184                             0 1000 300 i
*>                  10.1.2.1                                 0 1000 300 i
*                   32.1.13.184                0            0 300 i
R2#
```

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

R2# **show bgp ipv6 unicast**

```
R2#show bgp ipv6 unicast
BGP table version is 7, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

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

Step 4: Viewing explicit routes and path attributes.

a. Use the **show bgp ipv4 unicast** *ipv4-prefix subnet-mask* command on R2 to display all the paths for a specific route and BGP path attributes for that route. R2# **show bgp ipv4 unicast 192.168.1.0 255.255.255.224**

```
R2#show bgp ipv4 unicast 192.168.1.0 255.255.255.224
BGP routing table entry for 192.168.1.0/27, version 2
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Advertised to update-groups:
        2
  300 1000
    32.1.13.184 (inaccessible) from 2001:DB8:ACAD:1023::3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
  1000
    32.1.13.184 (inaccessible) from 2001:DB8:ACAD:1012::1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external
  1000
    10.1.2.1 from 10.1.2.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
R2#
```

R2# **show bgp ipv6 unicast 2001:db8:acad:1000::/64**

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

R2#show bgp ipv4 unicast 192.168.1.0 255.255.255.224
BGP routing table entry for 192.168.1.0/27, version 2
Paths: (3 available, best #3, table Default-IP-Routing-Table)
  Advertised to update-groups:
        2
  300 1000
    32.1.13.184 (inaccessible) from 2001:DB8:ACAD:1023::3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
  1000
    32.1.13.184 (inaccessible) from 2001:DB8:ACAD:1012::1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external
  1000
    10.1.2.1 from 10.1.2.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
R2#show bgp ipv6 unicast 2001:db8:acad:1000::/64
BGP routing table entry for 2001:DB8:ACAD:1000::/64, version 2
Paths: (2 available, best #2, table Default)
  Advertised to update-groups:
        2
  300 1000
    2001:DB8:ACAD:1023::3 (FE80::3:1) from 2001:DB8:ACAD:1023::3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
  1000
    2001:DB8:ACAD:1012::1 (FE80::1:1) from 2001:DB8:ACAD:1012::1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
R2#
```

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

R2# **show bgp ipv4 unicast neighbors 10.1.2.1 advertised-routes**

```
R2#show bgp ipv4 unicast neighbors 10.1.2.1 advertised-routes
BGP table version is 7, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

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

Total number of prefixes 6
R2#
```

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

R2# **show bgp ipv6 unicast neighbors 2001:db8:acad:1012::1 advertised- routes**

```
R2#show bgp ipv6 unicast neighbors 2001:db8:acad:1012::1 advertised-routes
BGP table version is 7, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

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

 --More--
```

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

a. By examining the IPv4 and IPv6 routing tables on R2, you can verify that BGP is receiving the IPv4 and IPv6 prefixes from R1 and R3. R2# **show ip route bgp | begin Gateway**

Part 4: Configure and Verify IPv6 Route Summarization

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

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

R2# **show ipv6 route bgp | section 2001**

```
R2#
R2#show ip route bgp | begin Gateway
R2#
R2#show ipv6 route bgp | section 2001
B    2001:DB8:ACAD:1000::/64 [20/0]
     via FE80::1:1, GigabitEthernet2/0
B    2001:DB8:ACAD:1001::/64 [20/0]
     via FE80::1:1, GigabitEthernet2/0
B    2001:DB8:ACAD:3000::/64 [20/0]
     via FE80::3:1, GigabitEthernet3/0
B    2001:DB8:ACAD:3001::/64 [20/0]
     via FE80::3:1, GigabitEthernet3/0
R2#
```

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

R1(config)# **router bgp 1000**

R1(config-router)# **address-family ipv6 unicast**

R1(config-router-af)# **aggregate-address 2001:db8:acad:1000::/52 summary- only**

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

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

R2# **show bgp ipv6 unicast | begin Network**

```
R2#show  bgp ipv6 unicast | begin Network
   Network          Next Hop            Metric LocPrf Weight Path
*  2001:DB8:ACAD:1000::/52
                    2001:DB8:ACAD:1023::3
                                                   0 300 1000 i
*>                  2001:DB8:ACAD:1012::1
                                         0         0 1000 i
*> 2001:DB8:ACAD:2000::/64
                    ::                   0      32768 i
*> 2001:DB8:ACAD:2001::/64
                    ::                   0      32768 i
*  2001:DB8:ACAD:3000::/64
                    2001:DB8:ACAD:1012::1
                                                   0 1000 300 i
*>                  2001:DB8:ACAD:1023::3
                                         0         0 300 i
*  2001:DB8:ACAD:3001::/64
                    2001:DB8:ACAD:1012::1
                                                   0 1000 300 i
*>                  2001:DB8:ACAD:1023::3
                                         0         0 300 i
R2#
```

R3# **show bgp ipv6 unicast | begin Network**

```
R3#show bgp ipv6 unicast | begin Network
*Dec 25 13:49:42.123: %SYS-5-CONFIG_I: Configured from console by console
R3#show bgp ipv6 unicast | begin Network
   Network          Next Hop            Metric LocPrf Weight Path
*  2001:DB8:ACAD:1000::/52
                    2001:DB8:ACAD:1023::2
                                                   0 500 1000 i
*                   2001:DB8:ACAD:1014::1
                                         0         0 1000 i
*>                  2001:DB8:ACAD:1013::1
                                         0         0 1000 i
*  2001:DB8:ACAD:2000::/64
                    2001:DB8:ACAD:1014::1
                                                   0 1000 500 i
*                   2001:DB8:ACAD:1013::1
                                                   0 1000 500 i
*>                  2001:DB8:ACAD:1023::2
                                         0         0 500 i
*  2001:DB8:ACAD:2001::/64
                    2001:DB8:ACAD:1014::1
                                                   0 1000 500 i
*                   2001:DB8:ACAD:1013::1
                                                   0 1000 500 i
*>                  2001:DB8:ACAD:1023::2
                                         0         0 500 i
*> 2001:DB8:ACAD:3000::/64
 --More--
```

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

R2# **show ipv6 route bgp | section 2001**

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)