VISVESVARAYATECHNOLOGICALUNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

Computer Networks

Submitted by

ROHITH U (1BM21CS170)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING BENGALURU-560019 June-2023 to September-2023

(Autonomous Institution under VTU)

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **Computer Networks**" carried out by **ROHITH U (1BM21CS170),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks - (22CS4PCCON)** work prescribed for the said degree.

Dr. LOHITH JJ
Assistant Professor
Department of CSE
BMSCE,Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department Of CSE
BMSCE,Bengaluru

Index

CYCLE 1

Experiment No. 1

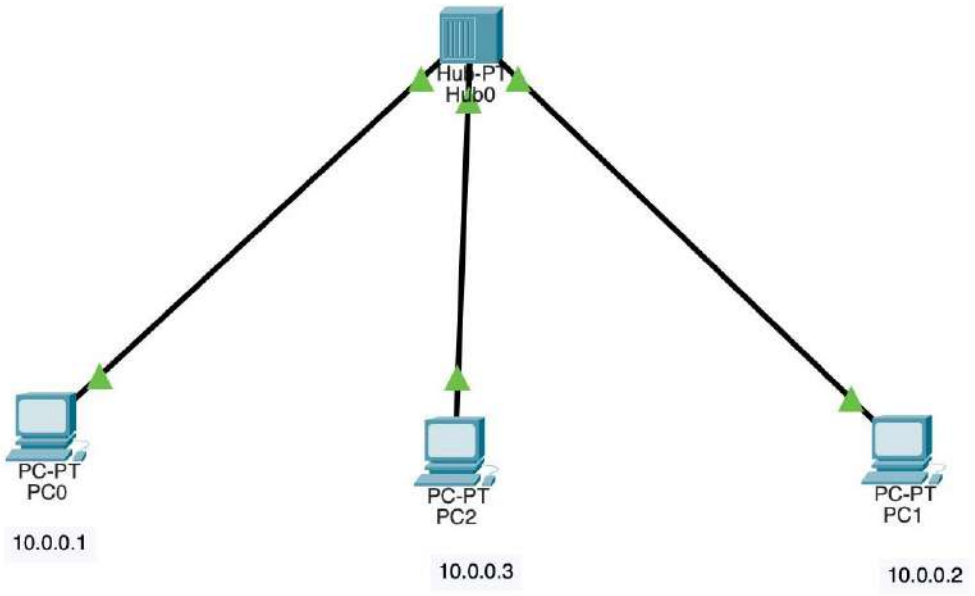
Title:
Create a topology with 3 or more end devices using

i)Hub

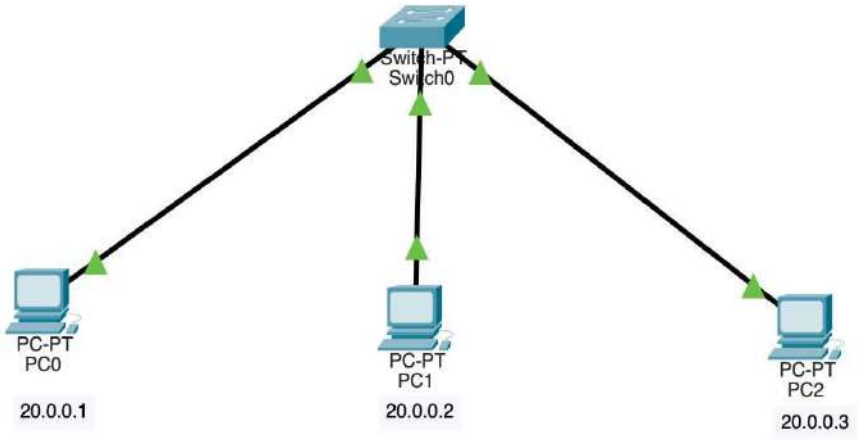
ii)Switch

iii)Hub & Switch Hybrid

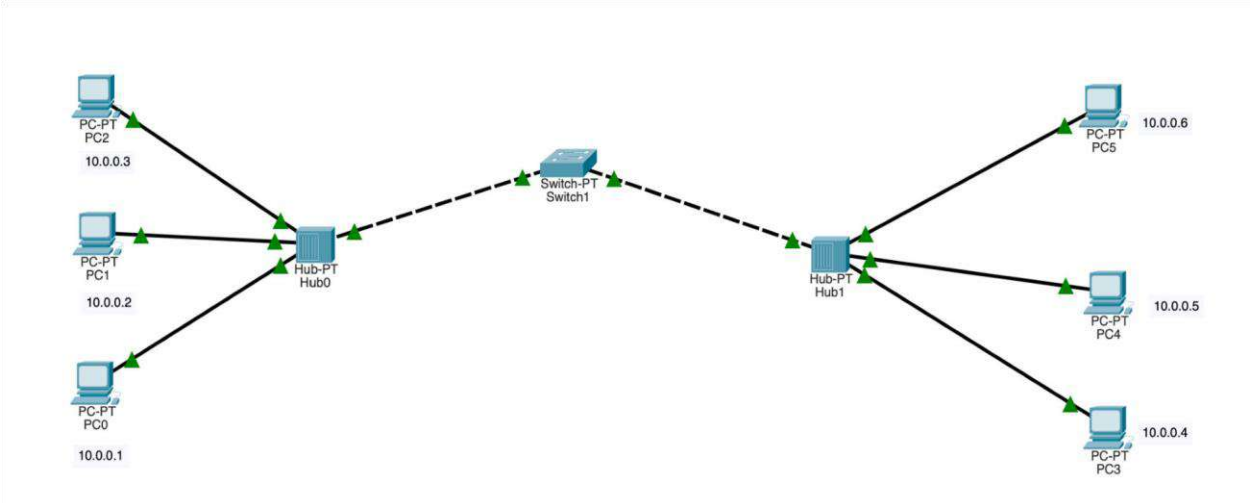
Topology:



i)Hub topology



ii)Switch topology



iii)Hub&Switch Hybrid topology

```
C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time=18ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 18ms, Average = 4ms
```

```
C:\>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time<1ms TTL=128
Reply from 20.0.0.3: bytes=32 time<1ms TTL=128
Reply from 20.0.0.3: bytes=32 time<1ms TTL=128
Reply from 20.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time<1ms TTL=128
Reply from 20.0.0.2: bytes=32 time=14ms TTL=128
Reply from 20.0.0.2: bytes=32 time<1ms TTL=128
Reply from 20.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 14ms, Average = 3ms
```

```
C:\>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time<1ms TTL=128
Reply from 20.0.0.1: bytes=32 time<1ms TTL=128
Reply from 20.0.0.1: bytes=32 time<1ms TTL=128
Reply from 20.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time<1ms TTL=128
Reply from 20.0.0.3: bytes=32 time<1ms TTL=128
Reply from 20.0.0.3: bytes=32 time<1ms TTL=128
Reply from 20.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

iii)

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.04

Pinging 10.0.0.04 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data:

Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128
Reply from 10.0.0.6: bytes=32 time=16ms TTL=128

Ping statistics for 10.0.0.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 16ms, Average = 4ms

C:\>
```

```
C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Lab 1 - Cisco Packet Tracer

URBAN EDGE 9.6.23

1] LAN - Local Area Network is a collection of devices connected together in one physical location such as a building, office or home. It can be small or large, ranging from home network.

2] WAN - wide area network, a network that connects separate machines over a wide range area, for example in different countries, using telecommunication system.

3] Ethernet: It is a traditional technology for connecting devices in wLAN (or) WAN. It enables devices to communicate with each other via a protocol, which is a set of rules (or) common network language.

4] IP - address: Internet Protocol, unique address that identifies a device on internet or a local network. They contain location information and make devices accessible for communication.

5] HUB: It is a physical layer networking device which is used to connect multiple devices in a network.

6] Switch: Switch is a device in a computer network that connects other devices together.

7] Server: It is a computer program or device that provides a service to another program and its user.

8] End Device: It is either source or destination

of data transmitted over the network.

9] Nodes: It is the connection point among network devices such as router or printer that can receive and send data from one endpoint to other.

* Creating a First Network

Step1: open the Cisco Packet Tracer student version

Step2: click on the Logical dot the left-top corner and Simulation at right-down corner

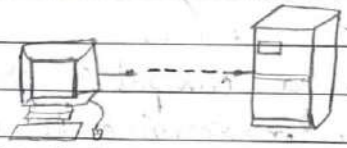
Step3: Go to the End devices and add Generic PC and a generic Server.

Step4: Under Connection, Select Copper-Cross-over cable

Step5: Now check if the lights are green, if not the connection.

Step6: Click the Server PC table, set the display name as client and the DNS server as 192.168.0.105 and IP address as 192.168.0.110.

Step7: Click on server table and change web server and IP address as 192.168.0.105. Make sure Port Status is ON. Load a background image and save the file.



PC PT
Client
192.168.0.110

Server-PT
Web server
192.168.0.105

EDGE

Steps: under connections, select copper-straight-through cable and connect it. The red lights indicate connections is not working. Hence connect copper-cross-over cable.

Step9: Go to realtime and select PC table and go to desktop and select command prompt in that type ping 192.168.0.110.

output:

command prompt:

PC> ping 192.168.0.110

pinging 192.168.0.110 with 32 bytes of data:

Reply from 192.168.0.110: bytes=32 time=0ms TTL=128.

Reply from 192.168.0.110: bytes=32 time=2ms TTL=128.

Reply from 192.168.0.110: bytes=32 time=0ms TTL=128.

Reply from 192.168.0.110: bytes=32 time=20ms TTL=128.

Ping statistic for 192.168.0.110

Packet sot=4, received=4, Lost=0(0% Loss)

Approx round trip times in milli-seconds:

Minimum=0ms, Maximum=20ms, Average=5ms.

PC> ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data:

Request timed out

Request timed out

Request timed out

Request timed out

Packet: Sent: 4, received=0, Lost: 4 (100% Loss

URBAN EDGE 10/05/23

LAB-2

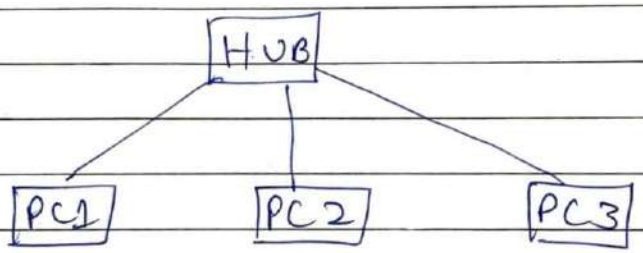
1. Create a Topology consisting of 3 or more devices connected with the help of HUB.

2. Create a topology consisting of 3 or more devices connected with the help of a Switch.

3. Create a topology and Simulate sending a Simple PDU from source to destination using HUB and switch as connecting devices and demonstrate ping message.

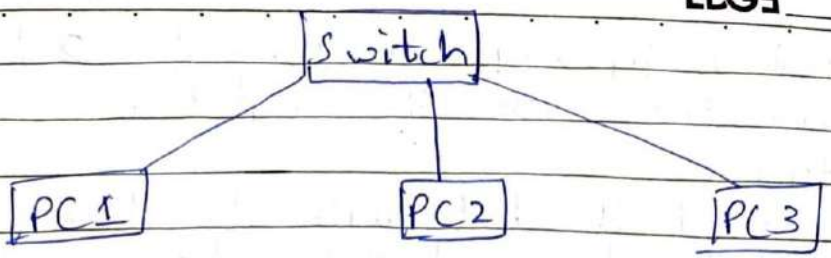
Observations:

(1)



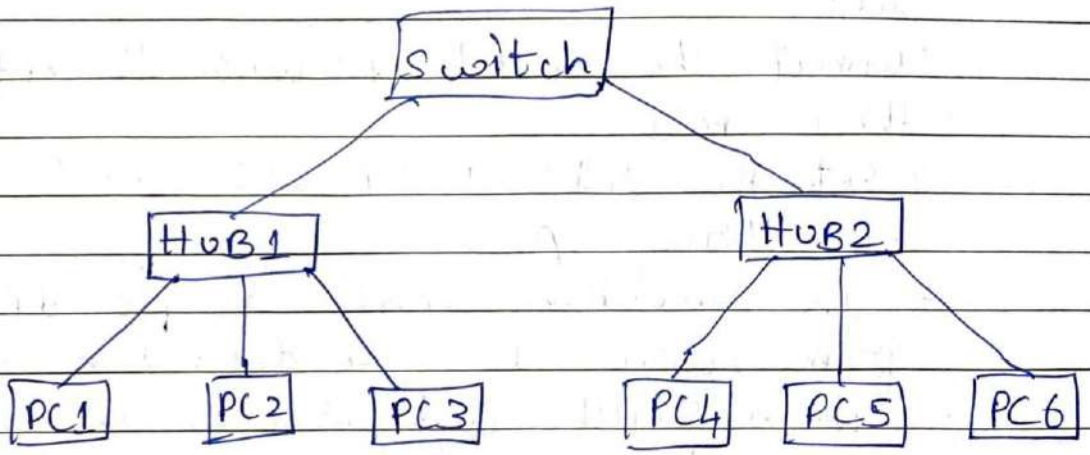
Whenever a source node sends data in a network, the hub recieves the data from the source and sends/broadcast over the network i.e, it sends data to all remaining nodes in the network and the node whose destination address matches the data will accept that data and acknowledge back and the rest of the nodes just ignores that.

(2)



The end devices are connected to single device (switch). when a source node sends data to other node, the switch recieves the data and sends only to the node whose destination address matches.

(3)



Sending message from PC1 to PC4.

⇒ PC1 sends the message, HUB1 recieves the message and transmits to PC2 and PC3 and PC2 and PC3 rejects the message as the destination address does not match.

⇒ Then switch transmits the message to HUB2, the HUB2 sends to PC4, PC5, PC6 simultaneously.

⇒ PC4 accepts the message and acknowledges back and PC5 and PC6 rejects the message.

URBAN EDGE 23.05.23

Create a topology consisting of 2 devices connected with the help of a router

step 1:

* Connect the hub with the three PC's.
* set the different IP addresses to the all three PC's.
* In simulation send simple PDU from source PC's to destination PC within the HUB and observe the simulation upto gets successfully message in the event list.
* Connect the switch - PT with the another three PC's
* set the different IP addresses to the all three PC's.
* In Simulation send Simple PDU from source PC's to destination PC within Switch and observe the simulation upto gets successfully message in the event list.
* Lastly Connect the Hub and switch and send the simple PDU to the Hub. Connected source PC and destination PC from the switch connected PC.
* finally when switch is in off the sending PDU will get stoped (failed)

Command prompt:

Hub on switch on

PC> ping 10.0.0.1

pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1 : bytes = 32 time = 1ms TTL = 128

Reply from 10.0.0.1 : bytes = 32 time = 2ms TTL = 128

0/0 1/0 4/0 5/0

10.0.0.5, 20.0.0.8, URBAN EDGE

Reply from 10.0.0.1 : bytes = 32 time = 1m
TTL = 128
Reply from 10.0.0.1 : bytes = 32 time =
TTL = 128.

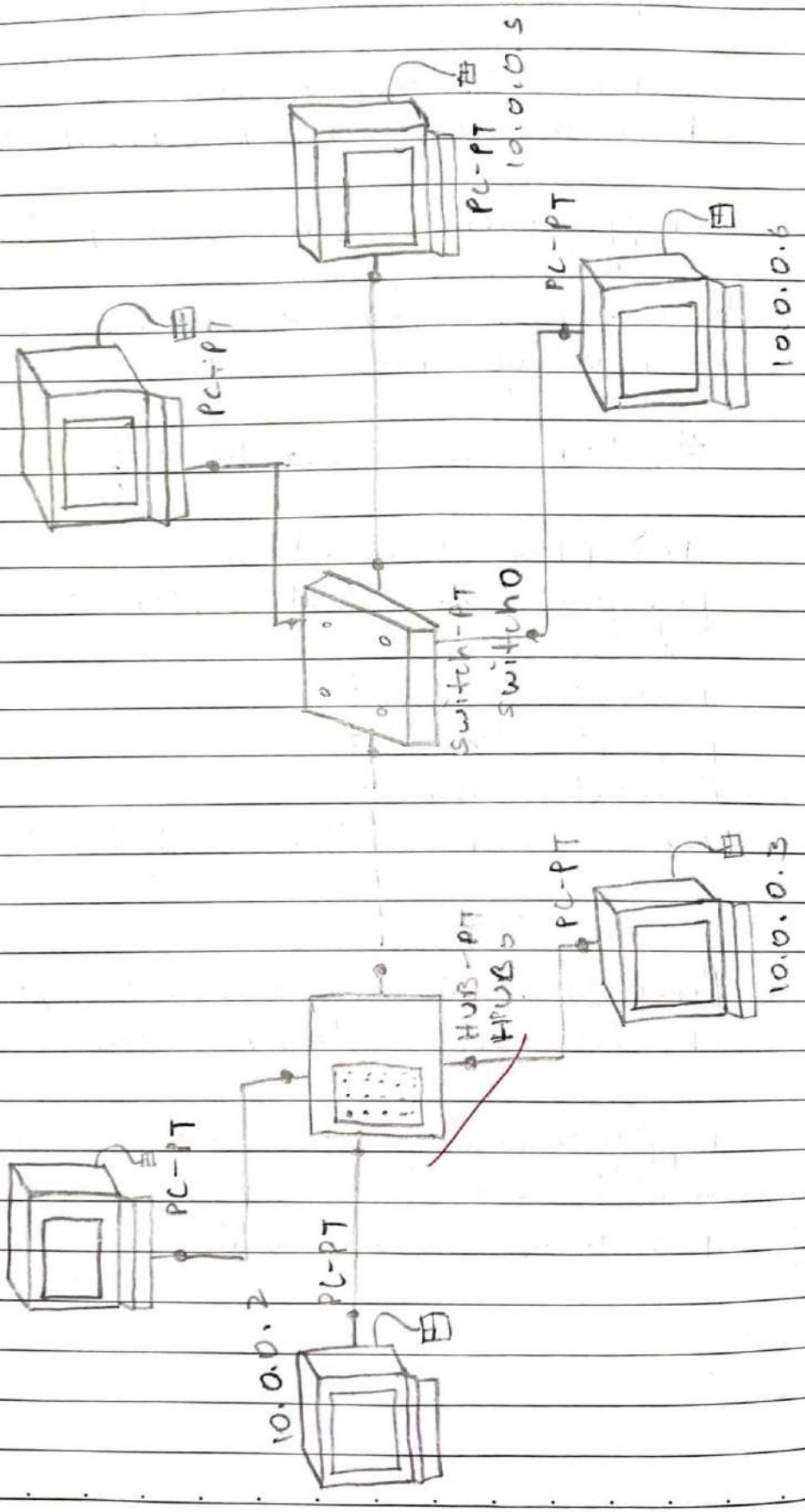
Ping statistics for 10.0.0.1:
Packets : sent = 4, Received = 4, Lost = 0
(0% loss
Approximate round trip time in
milli-seconds
minimum = 1ms, maximum = 2ms
Average = 1ms

HUB ON switch off.

PC> ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of
data:
Request time out
Request time out
Request time out
Reply from 10.0.0.4: bytes = 32, time
= 0ms
TTL = 128
Ping statistics for 10.0.0.4:
Packets Sent = 4, Received = 1, lost 3
(75% loss)
Approximate round trip times in
milli seconds:
Minimum = 0ms, maximum = 0ms,
Average = 0ms

URBAN EDGE

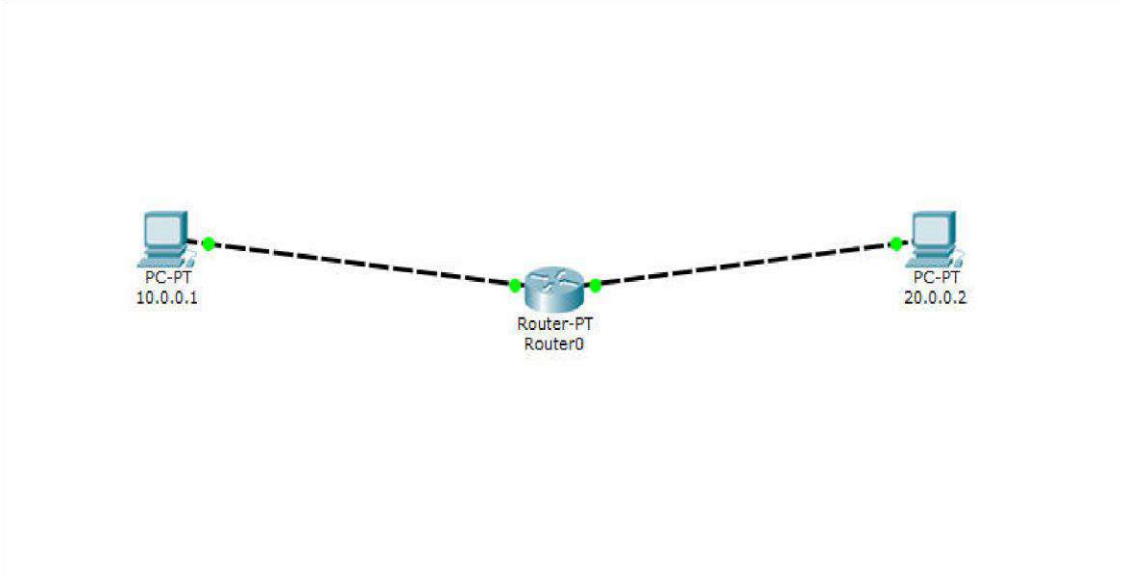
Topology:



Experiment No. 2

Title:
Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

Topology: with single router



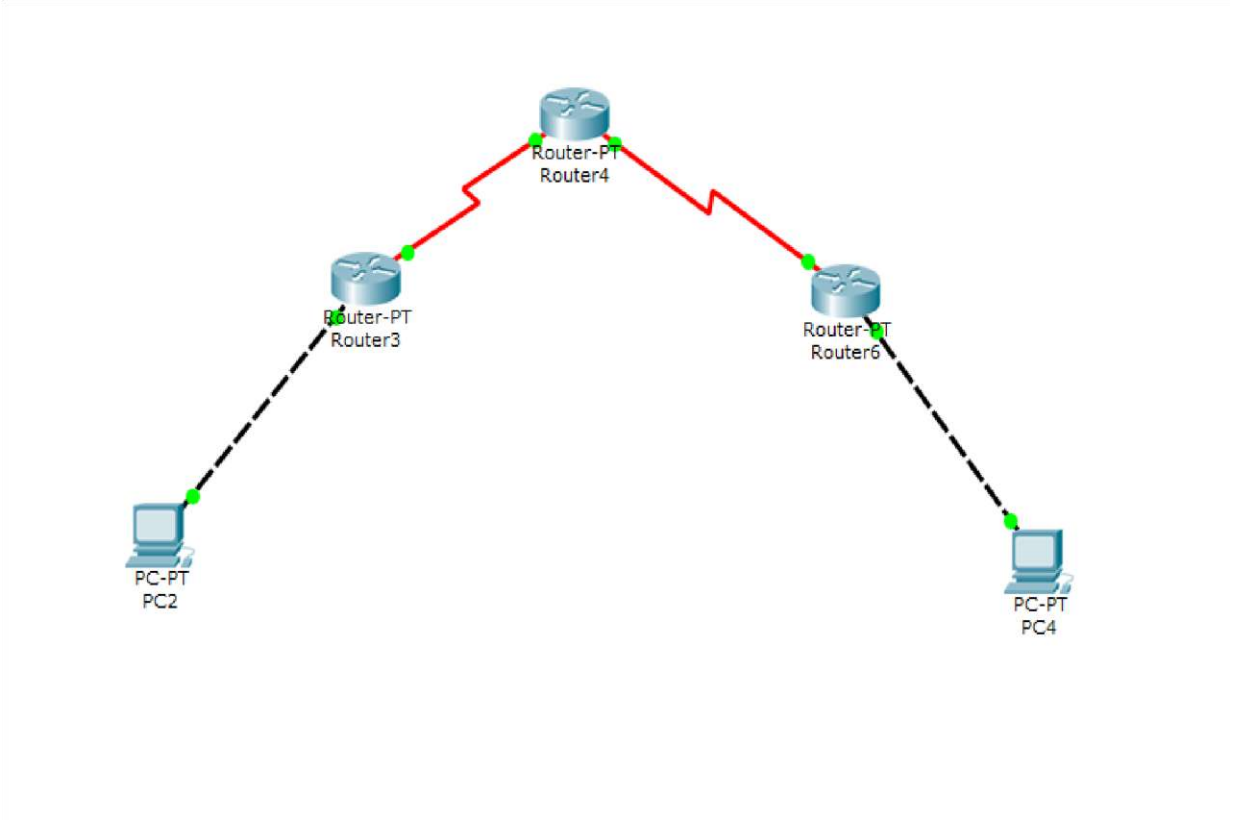
Pinging PC2 from PC1:

```
PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time=1ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time=1ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

Pinging PC1 from PC2:

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=0ms TTL=127
Reply from 10.0.0.1: bytes=32 time=0ms TTL=127
Reply from 10.0.0.1: bytes=32 time=0ms TTL=127
Reply from 10.0.0.1: bytes=32 time=0ms TTL=127

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

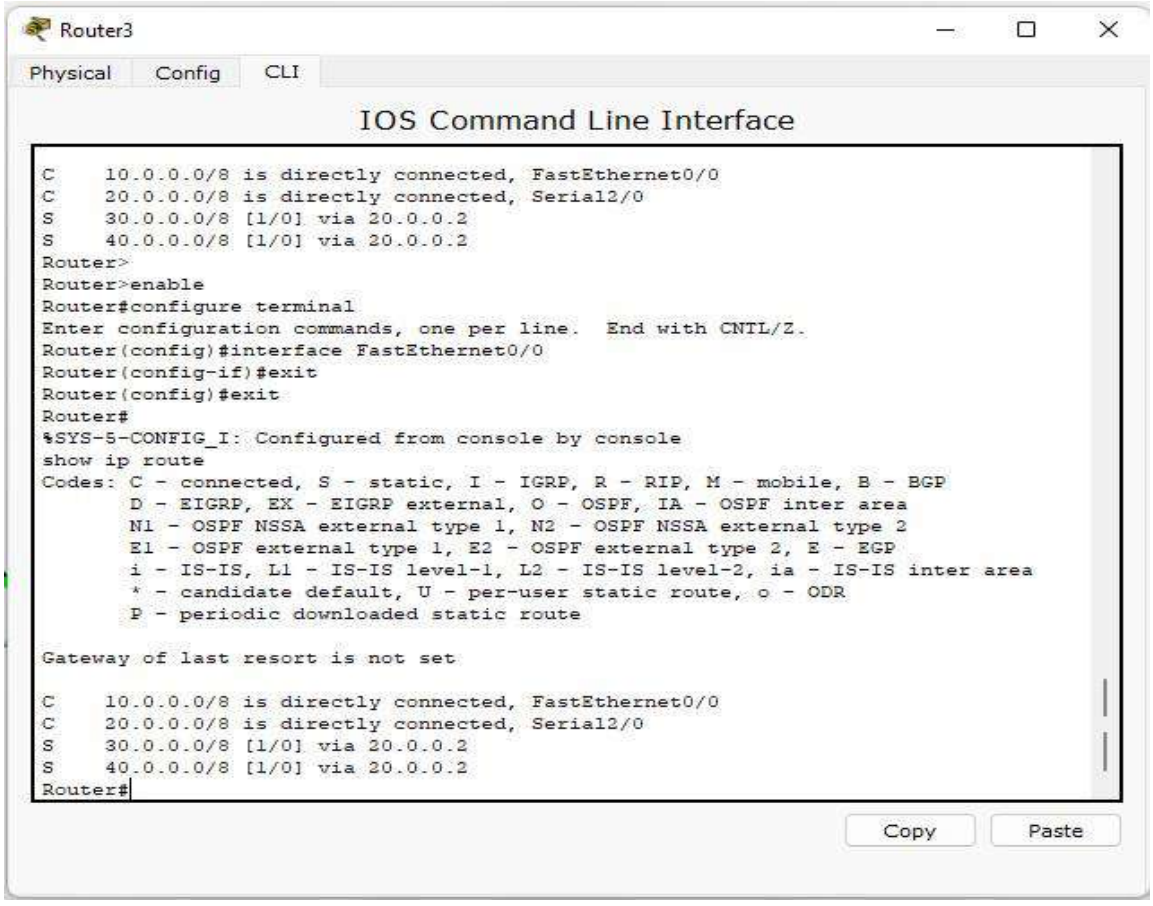
Topology: with 3 Routers

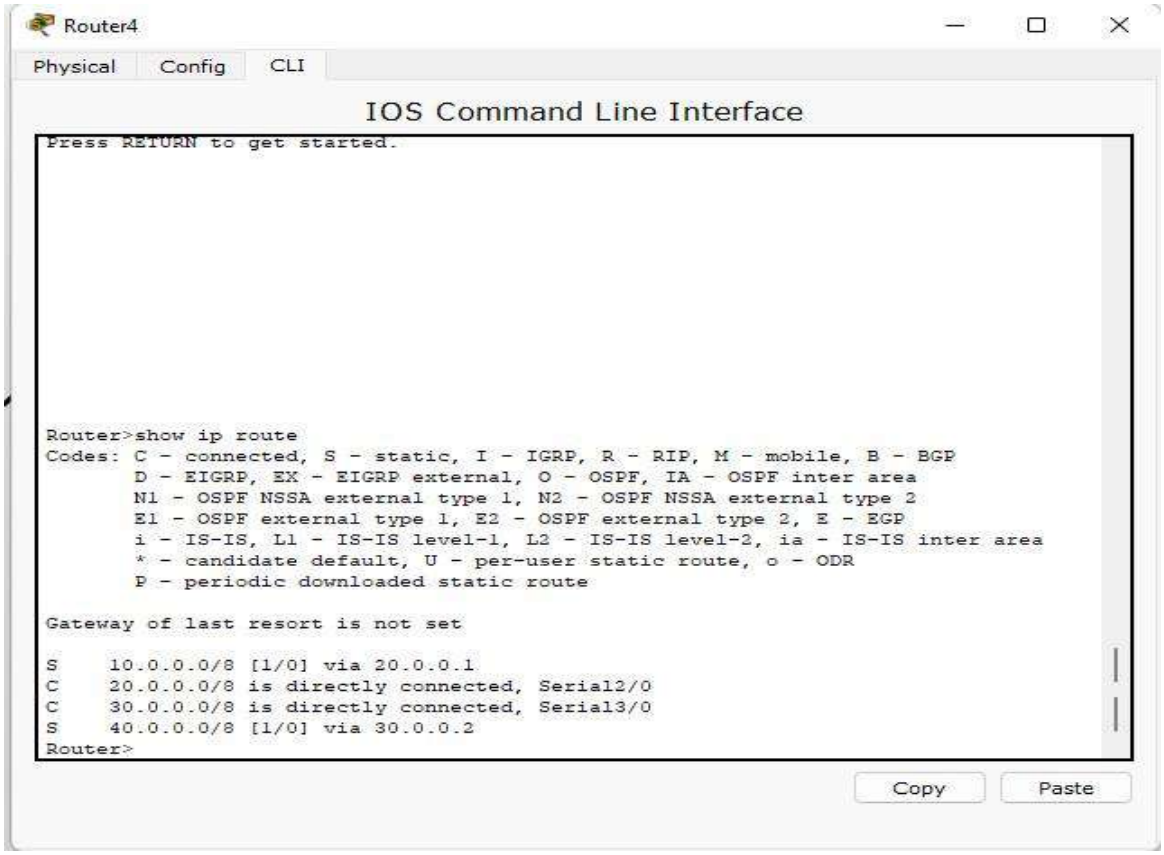


Pinging end device - in different network before setting IP route

```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

IP route - for all routers




Router4
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started.
Router>show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
Router>
Copy
Paste


Router6
Physical
Config
CLI
IOS Command Line Interface
Router>show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set
S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
Router>
Copy
Paste

After setting IP route

Pinging PC4 from PC2

```
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=5ms TTL=253
Reply from 40.0.0.1: bytes=32 time=7ms TTL=253

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 5ms
```

Pinging PC2 from PC4

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=2ms TTL=253
Reply from 10.0.0.1: bytes=32 time=7ms TTL=253
Reply from 10.0.0.1: bytes=32 time=7ms TTL=253
Reply from 10.0.0.1: bytes=32 time=6ms TTL=253

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 7ms, Average = 5ms

PC>
```

Week-3

URBAN EDGE

Create IP address to routers in packet tracker. Explore the following messages ping responses destination unreachable, request timed out reply.

Step1: place 4 PC's on workspace and 3 routers on workspace

Step2: Connect 2 PC's to one router and another 2 PC's to other router connect these 2 routers to third router

Step3: Set the IP address of 1st PC as 10.0.0.1 and gateway to 10.0.0.2

Set IP address of 2nd PC to 20.0.0.1 and gateway to 20.0.0.2.

Set IP address of 3rd PC to 30.0.0.1 and gateway to 30.0.0.2.

Set IP address of with PC to 40.0.0.1 and gateway to 40.0.0.2.

Step4: In CLI of router 1,

enable

Configterminal

interface fastethernet 0/0

IP address 10.0.0.2 255.0.0.0

no shutdown.

Repeat the same in CLI of router 2.

Step5: In CLI of router 1

To set the serial configuration

enable

config terminal

interface serial 2/0.

ip address 50.0.0.1

no shutdown

URBAN EDGE

Repeat the same in router 2 and 3.

Command output:

In PC1

ping 30.0.0.1

pinging 30.0.0.1 with 32 bytes of data.

Reply from 10.0.0.2 Destination host unreachable.

Reply from 10.0.0.2 Destination host unreachable

Reply from 10.0.0.2 Destination host unreachable.

Reply from 10.0.0.2 Destination host unreachable.

Ping Statistics for 30.0.0.1

Packet : Sent = 4 ; Recieved = 0 Lost = 4 (100% Loss)

Since, If address 30.0.0.1 is not directly connected to router 1, so, manually 30.0.0.0 needs to be connected to router.

In CLI of router

To ST the static configuration enables.

Config terminal

ip route 30.0.0.0 255.0.0.0 S0.0.0.1

ip route 40.0.0.0 255.0.0.0 S0.0.0.1

ip route 60.0.0.0 255.0.0.0 S0.0.0.1

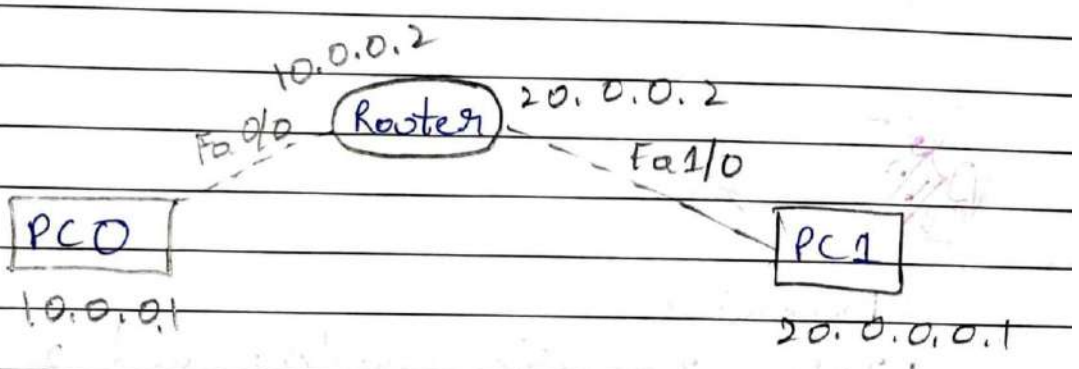
Same is repeated to router 2 and routers.

To view ip route

Show ip route

C. 10.0.0.0/8 is directly connected fast Ethernet 0/0
C 20.0.0.0/8 is directly connected fast Ethernet 1/0.
S 30.0.0.0/8 [1/0] via 50.0.0.3.
S 40.0.0.0/8 [1/0] via 50.0.0.3
C 50.0.0.0/8 is directly connected serial 2/0
S 60.0.0.0/8 [1/0] via 50.0.0.3.

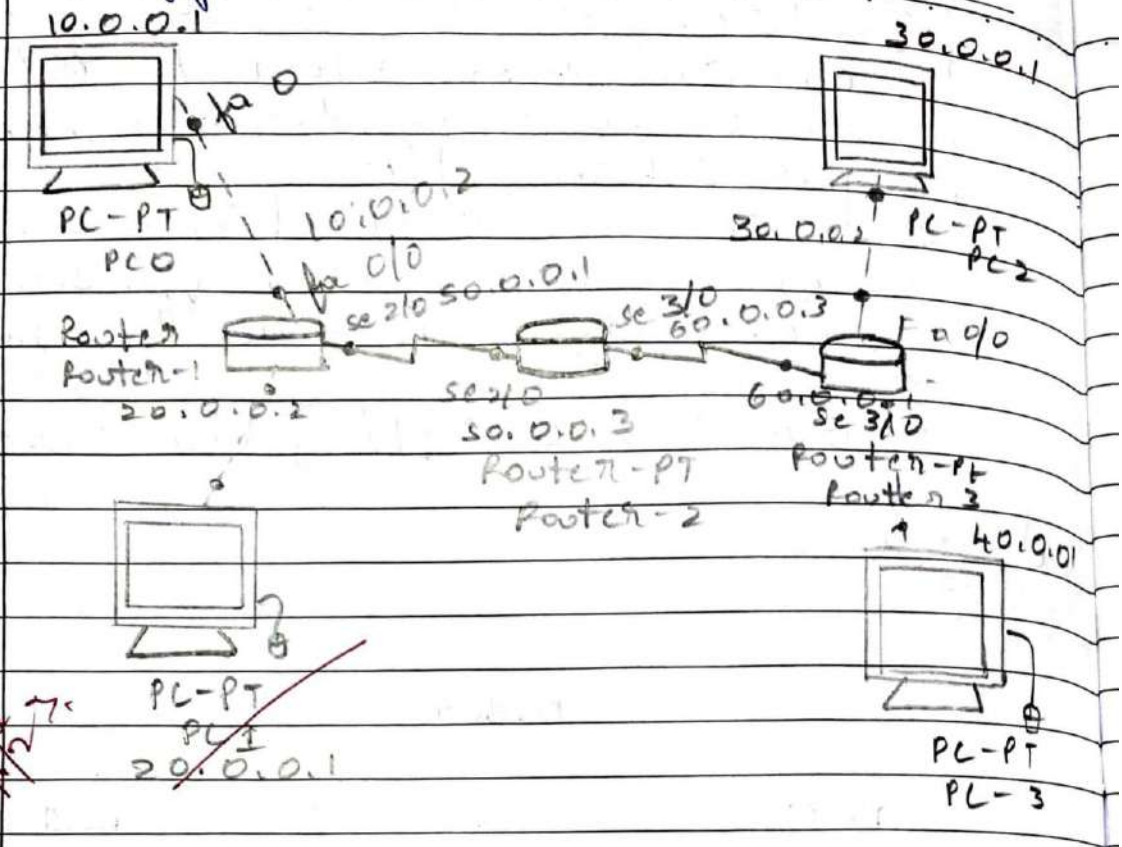
observation
Single Route Topology



CLI for router:
→ Enable
→ Configure terminal
→ Interface Fa 0/0
→ IP address 10.0.0.2 255.0.0.0
→ NO shutdown
→ exit.

Result:
The PC0 and PC1 of different networks are connected using router with their respective gateways.

Topology:



Procedure:

1. Take 3 generic routers and 2 end devices.
2. Connect end devices to router and used suitable wire between the routers.
3. Let IP address and default gateway to each end device of different network.
4. Configure the router using CLI. Put the IP addres same as respective gateway address for end devices.
5. Select PC0 and open the command prompt Ping PC1 with its IP address

Result:

The immediate router is connected with other two routers properly with green signal between the 3 router and end device.

URBAN EDGE

observation:
Pinging PC1 from PC0 works
* three routers and 2 PC's are connected as shown in topology.
* Pinging PC1 from PC0 shows destination host unreachable.
* Adding static routers to routers using
IP router < dest < subnet mask > < next hop >
in enable.
Config terminal mode.

PC 0 to Router 1
>enable
config terminal
(config)# interface Fa 0/0
(config) # IP address 10.0.0.2 255.0.0.0
(config) # NO shutdown.

Initial Ping:
> ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Reply from 10.0.0.1 destination host unreachable.
Reply from 10.0.0.1 destination host unreachable
Reply from 10.0.0.1 destination host unreachable
Request timed out

Packet: set=4, Recieved=0, lost=4
(100% Loss)

URBAN EDGE

→ setting IP Route : In Router 0
Router# Show IP route
C 10.0.0.0/8 is directly connected, Fa 0/0
C 20.0.0.0/8 is directly connected, Se 2/0
Router# configure terminal
Router(config)# ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)# ip route 40.0.0.0 255.0.0.0 20.0.0.2
Router(config)# exit
Router(config)# show ip route
C 10.0.0.0/8 is directly connected, Fa0/0
C 20.0.0.0/8 is directly connected, Se 2/0
S 30.0.0.0/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2

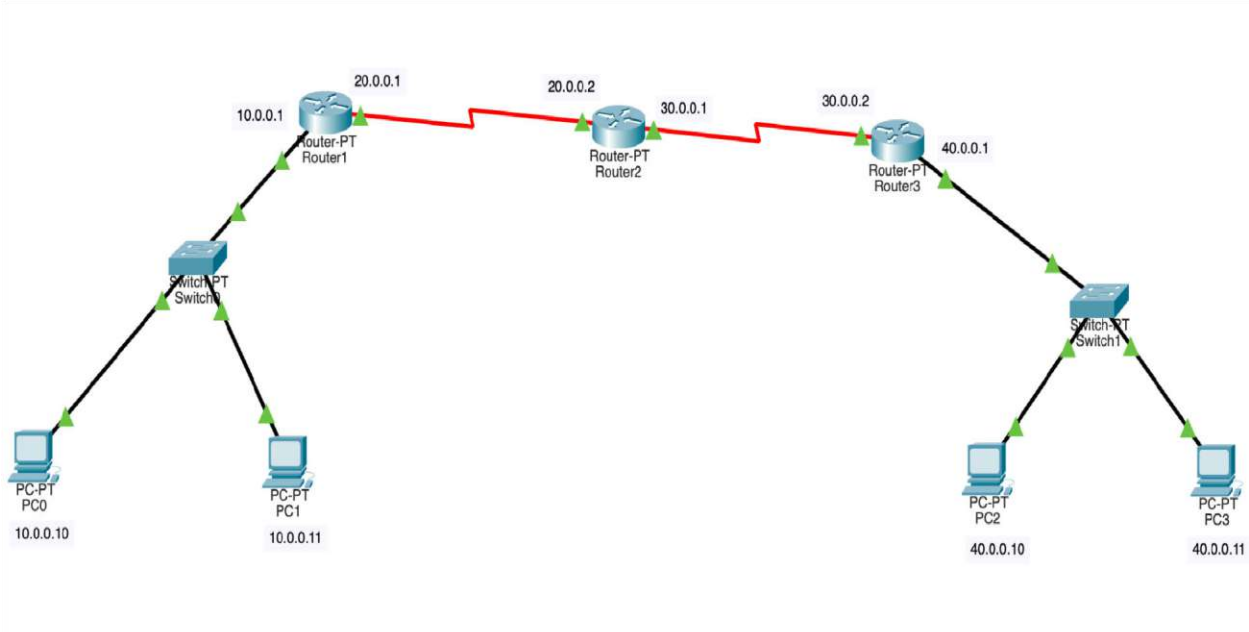
→ ping PC from PC
PC> ping 10.0.0.1
Reply from 10.0.0.1 : bytes=32 time=2ms TTL=253
Reply from 10.0.0.1 : bytes=32 time=7ms TTL=253
Reply from 10.0.0.1 : bytes=32 time=7ms TTL=253
Reply from 10.0.0.1 : bytes=32 time=6ms TTL=253
Ping statistics for 10.0.0.1 :
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

Approx round trip
Min = 2ms , Max = 7ms Average = 5ms

Experiment No. 3

Title:
Configure default route, static route to the Router

Topology:



IP Route for all routers:

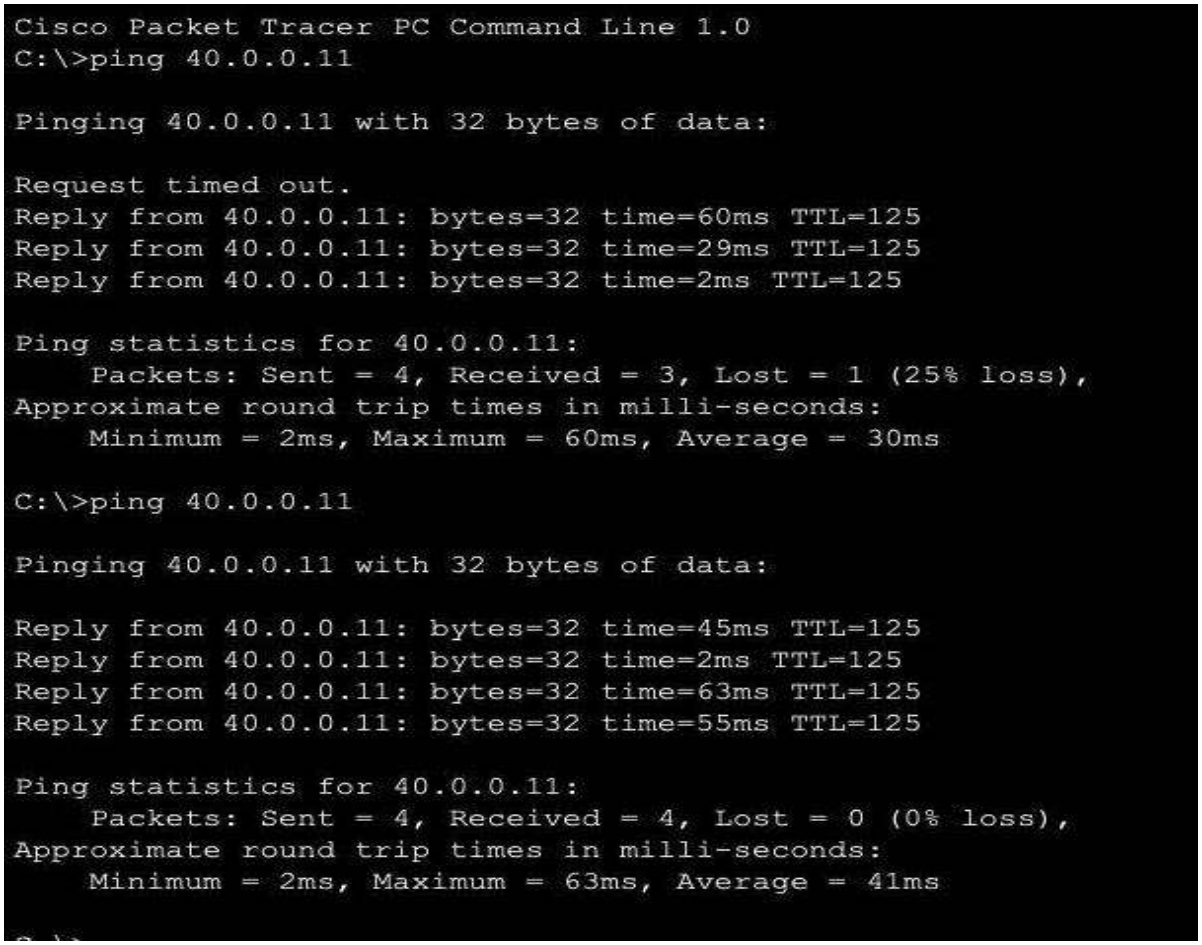
```
%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
show ip route
                ^
% Invalid input detected at '^' marker.

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0

Router#
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
S    40.0.0.0/8 [1/0] via 20.0.0.2
```

```
Router(config-if)#
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
show ip route
               ^
% Invalid input detected at '^' marker.

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set

C    20.0.0.0/8 is directly connected, Serial3/0
C    30.0.0.0/8 is directly connected, Serial2/0

Router#
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set

S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial3/0
C    30.0.0.0/8 is directly connected, Serial2/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
```

```
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
show ip route
                ^
% Invalid input detected at '^' marker.

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set

C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0

Router#
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set

S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
```

Pinging end devices in different network:

Ping PC3 from PC0



Ping PC1 from PC2

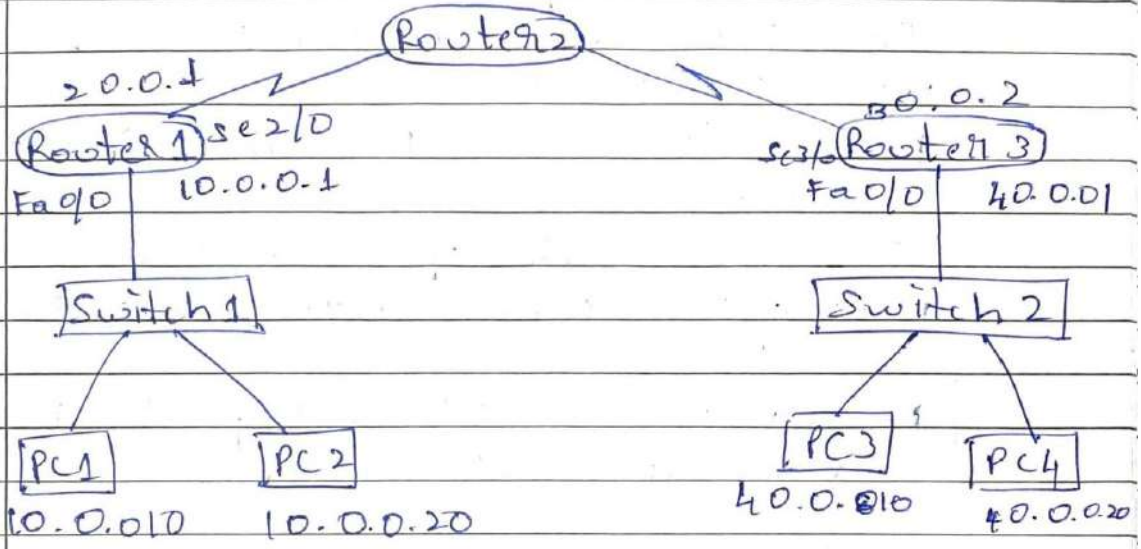
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.11: bytes=32 time=2ms TTL=125
Reply from 10.0.0.11: bytes=32 time=2ms TTL=125
Reply from 10.0.0.11: bytes=32 time=3ms TTL=125

Ping statistics for 10.0.0.11:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\>ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data:

Reply from 10.0.0.11: bytes=32 time=81ms TTL=125
Reply from 10.0.0.11: bytes=32 time=50ms TTL=125
Reply from 10.0.0.11: bytes=32 time=71ms TTL=125
Reply from 10.0.0.11: bytes=32 time=48ms TTL=125

Ping statistics for 10.0.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 48ms, Maximum = 81ms, Average = 62ms
```

URBAN EDGE

Configure default route, static route to the Router.

Aim:
To understand how to configure static ip route and default route using CLI.

Topology:



Procedure:
1. Connect to end devices to a switch which inturn is connected to a router.
2. These 2 routers are connected to Router 2.
3. The end devices PC1 & PC2 are set to IP address 10.0.0.10 and 10.0.0.20 statically. Similarly PC3 & PC4 are set to IP 40.0.0.10 and 40.0.0.20.
4. The end devices and switches are connected with copper-straight through wire. The rooters are connected using serial ports.

5. The ip is set for Router 1 interface using CLI.
Router(config)# Interface Fa 0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no Shutdown
Router(config-if)# exit

6. Step 5 is repeated for Serial interface Se 2/0 of Router 1 with IP set as 20.0.0.1
Router 2's serial interface Se 2/0 with 20.0.0.2
Se 3/0 with ip address of 30.0.0.1
Router 3's Serial interface Se 3/0 with 30.0.0.2
Router 3's Fast Ethernet port Fa 0/0 with 40.0.0.1.

7. Display ip routes in router 1 using, Router > show ip route.

8. Default route is set for Router 1
Router > enable
Router # configure termind
Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)# exit

9. Step 8 is repeated for Router 2 with Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

10. Display ip route of router 1 after setting default ip route.

11. Ip route is set for router 2 statically
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1

Result:
Ip route before setting default route, for Router 1
Gateway of last resort not set
C 10.0.0.0/8 is directly connected, Fa 0/0
C 20.0.0.0/8 is directly connected, Se 2/0

After setting default route and static route.
C 10.0.0.0/8 is directly connected, Fa 0/0
C 20.0.0.0/8 is directly connected, Se 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2

Ip route before setting ip route, for router 2

C 20.0.0.0/8 is directly connected, Se 2/0.
C 30.0.0.0/8 is directly connected, Se 3/0.

EDGE

After setting ip route,
C 20.0.0.0/8 is directly connected, Se 2/0.
C 30.0.0.0/8 is directly connected, Se 3/0.
S 10.0.0.0/8 [1/0] via 20.0.0.1
S 40.0.0.0/8 [1/0] via 30.0.0.2

IP route for Router 3 after setting IP route.
C 30.0.0.0/8 is directly connected, Se 2/0
C 40.0.0.0/8 is directly connected, Fa 0/0
S* 0.0.0.0/0 is [1/0] via 30.0.0.1

Pinging PC2 from PC4

PC4> ping 10.0.0.20
Reply from 10.0.0.20 byte=32 time=1ms TTL=128.
Reply from 10.0.0.20 byte=32 time=6ms TTL=128
Reply from 10.0.0.20 byte=32 time=8ms TTL=125
Reply from 10.0.0.20 byte=32 time=5ms TTL=125
Packet sent=4, Received=4, Lost=0 (0% Lost)
Approx Round trip time
Min=3ms Max=10ms, Average=7ms

Observation:

Static and Default ip route can be configured for a router connected to a switch, using CLI. After executing 'show ip route' command, different ip routes for that router are shown.

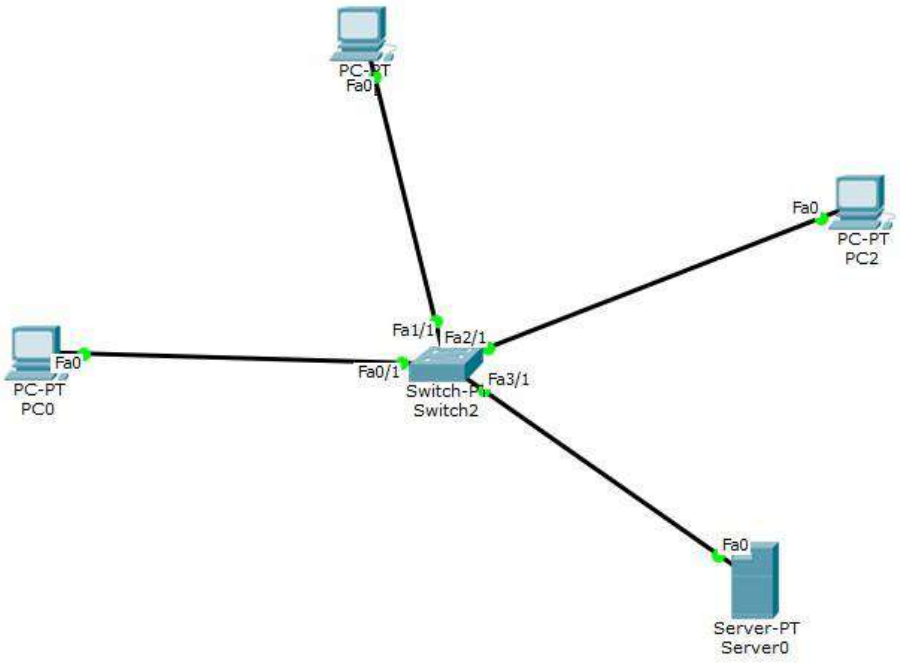
C denotes connected, S denotes static and S* denotes default route.

Experiment No. 4

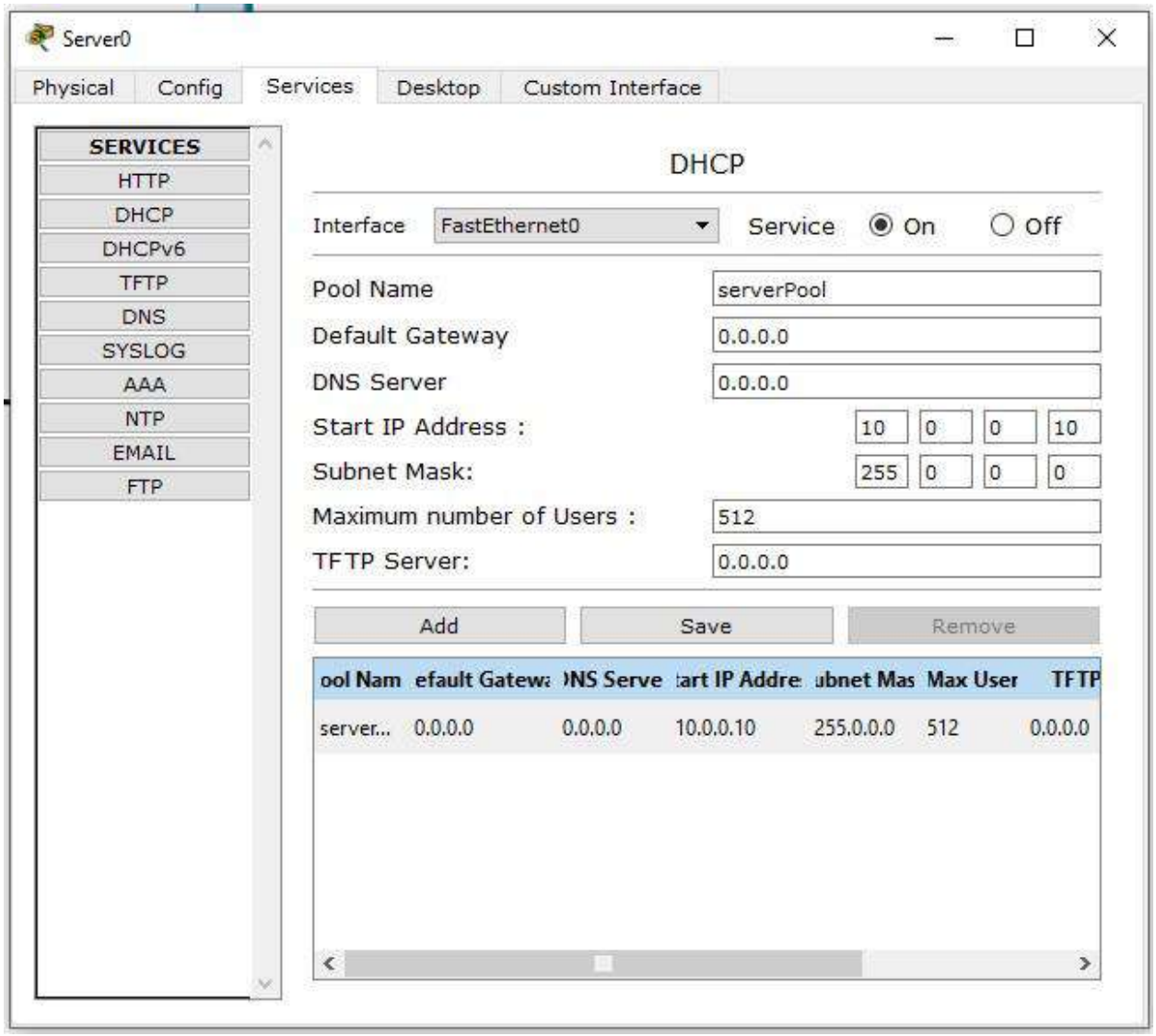
Title:
Configure DHCP within a LAN and outside LAN.

a)Within LAN

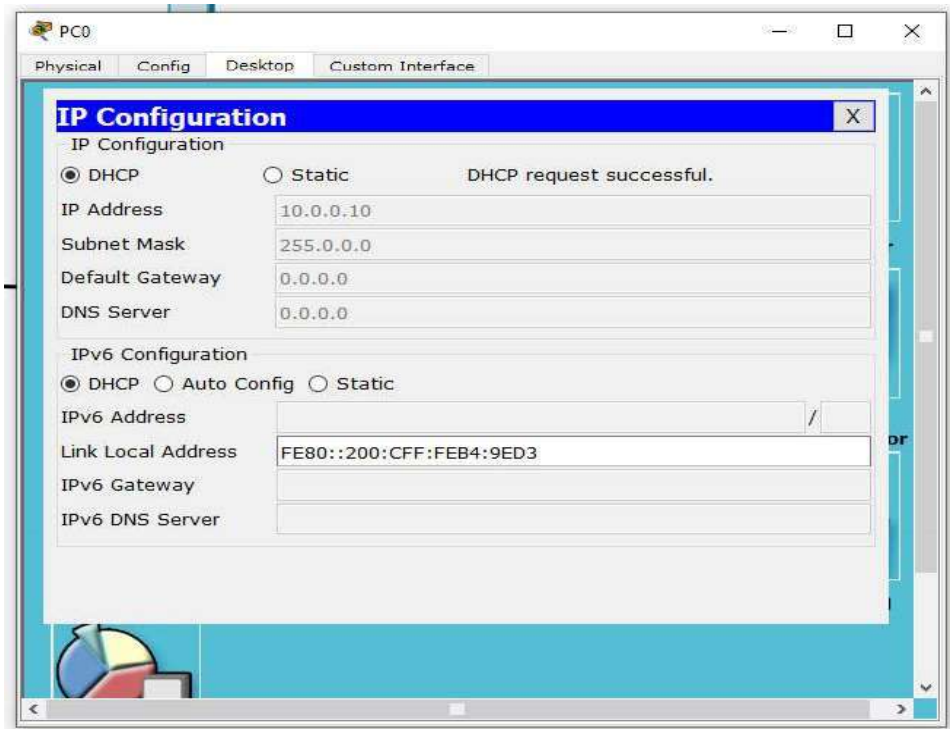
Topology:

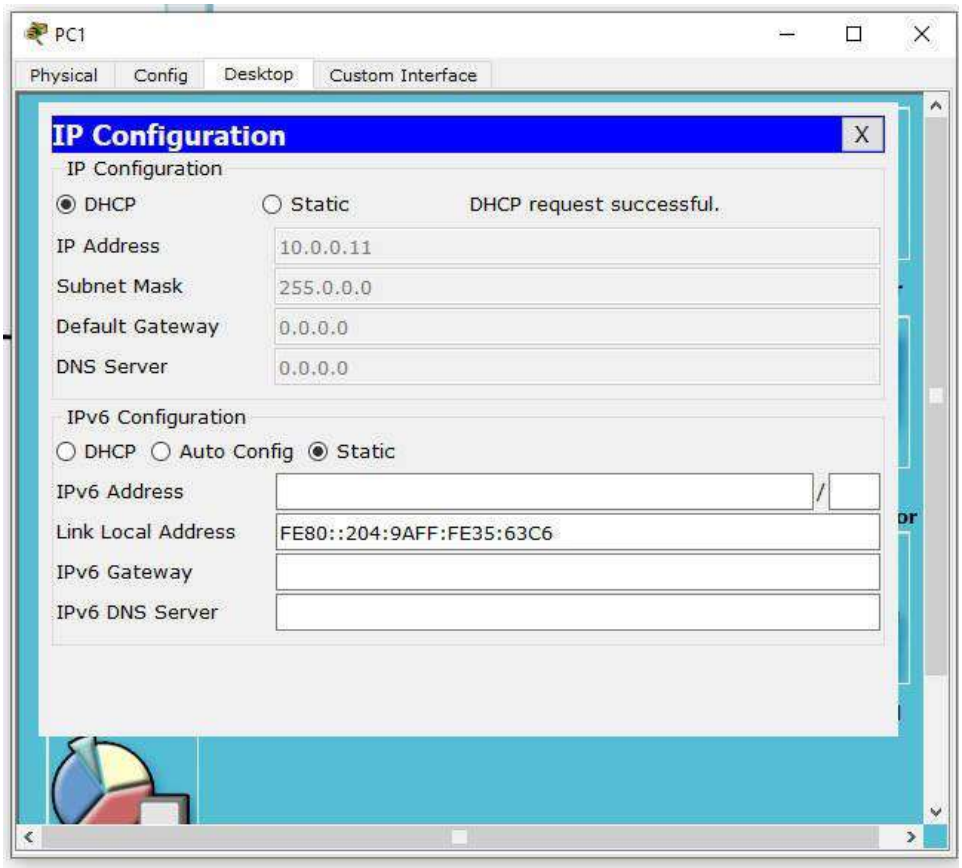


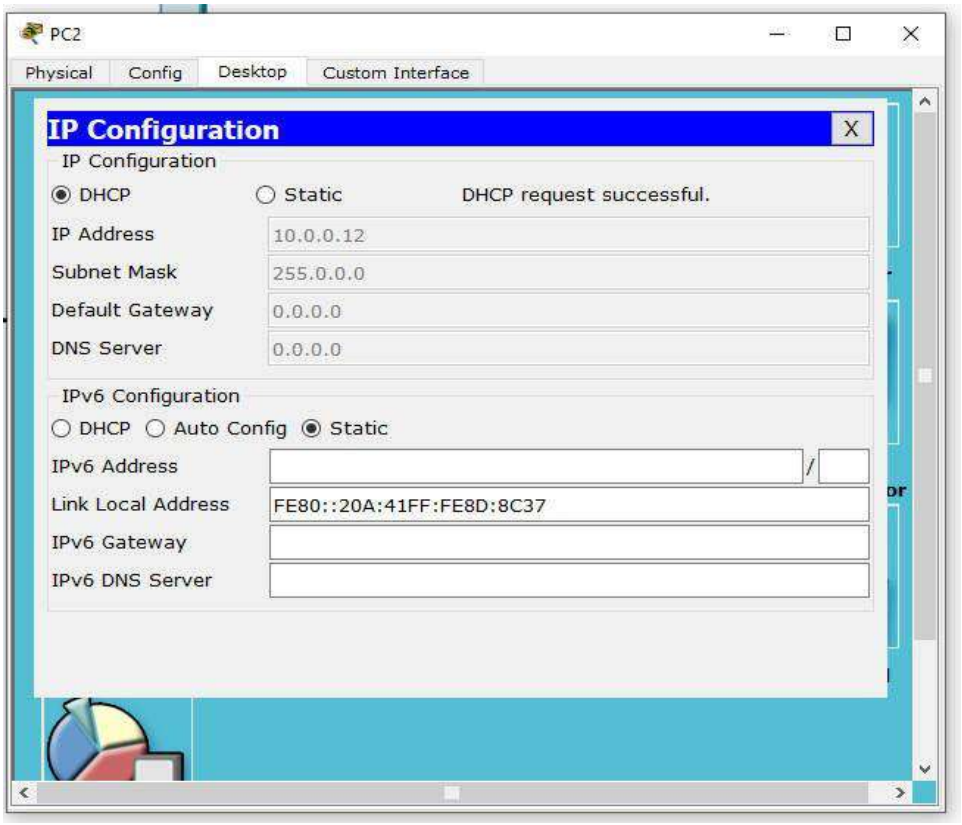
Server:



Obtaining IP:


PC0
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
10.0.0.10
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::200:CFF:FEB4:9ED3
IPv6 Gateway
IPv6 DNS Server

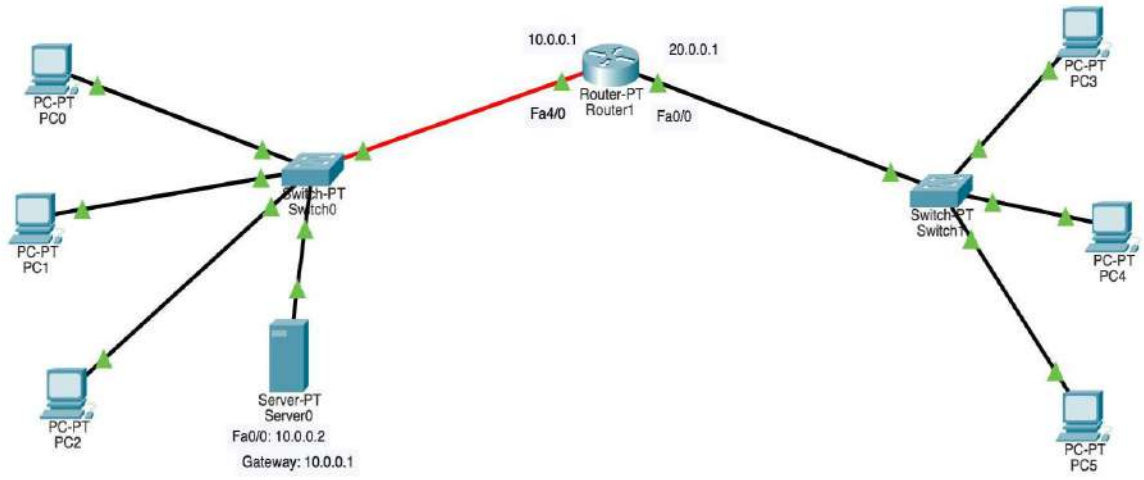

PC1
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
10.0.0.11
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::204:9AFF:FE35:63C6
IPv6 Gateway
IPv6 DNS Server



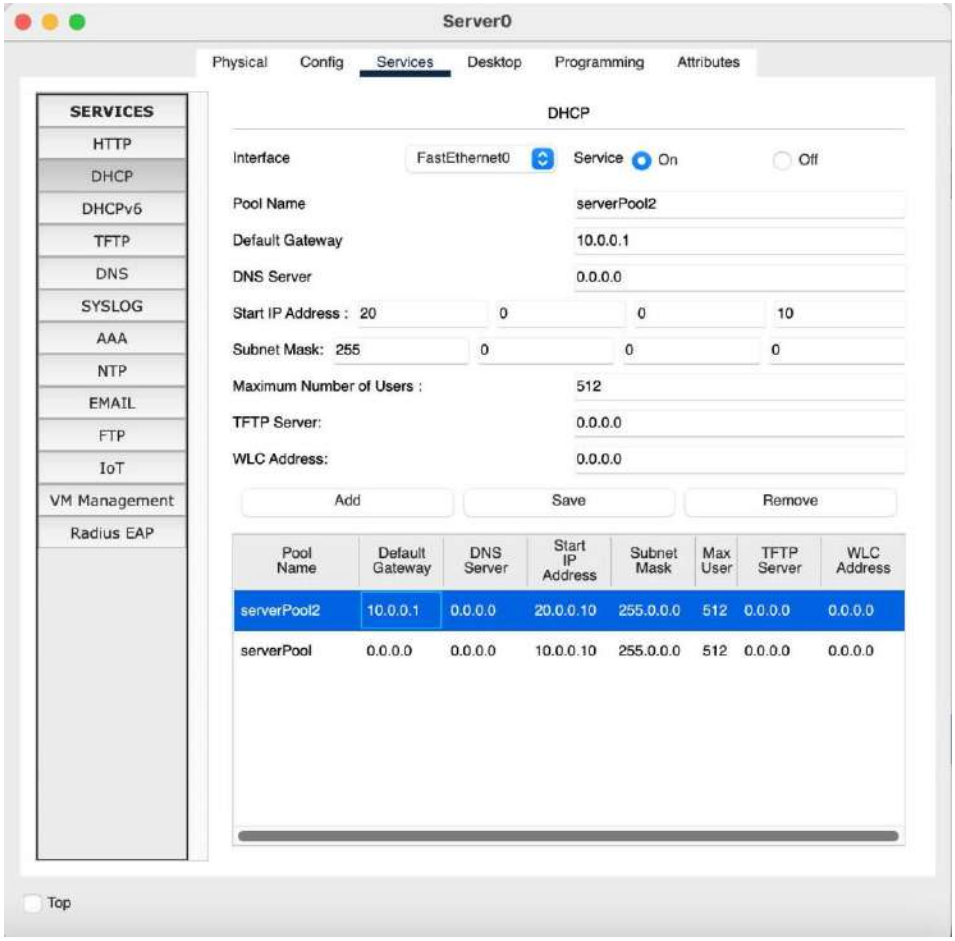
b)Outside LAN

I. Single

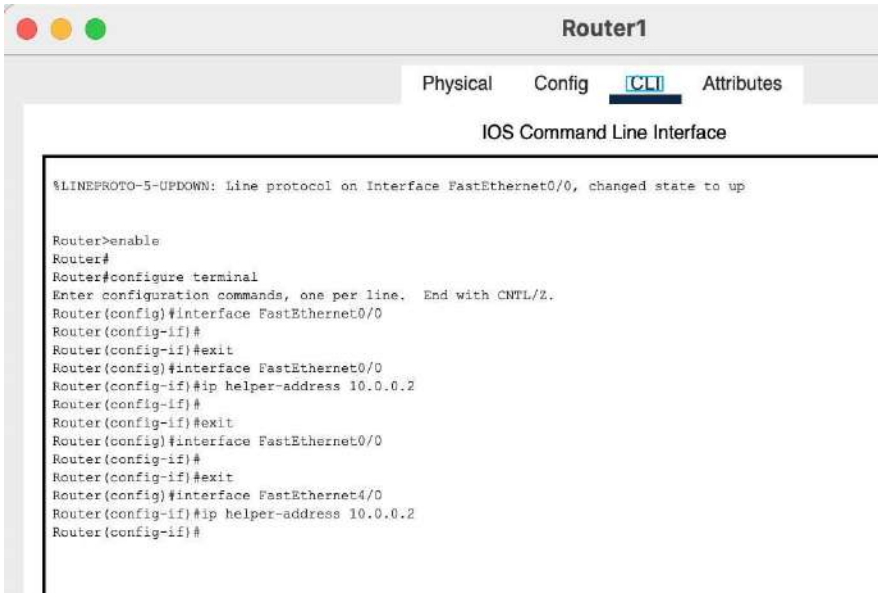
Router**Topology:**



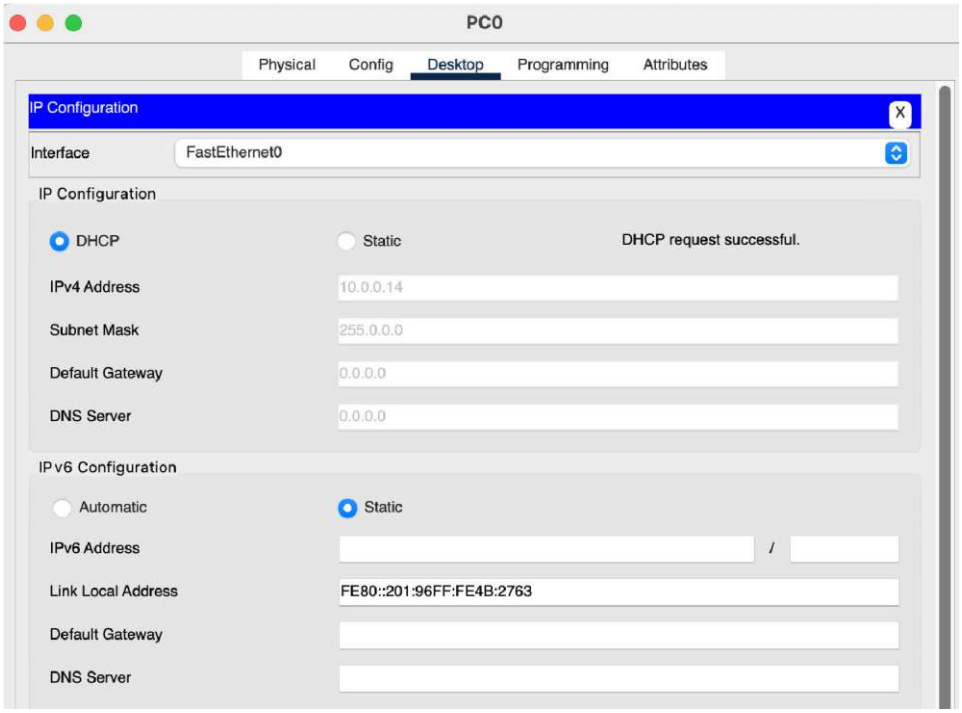
Server:

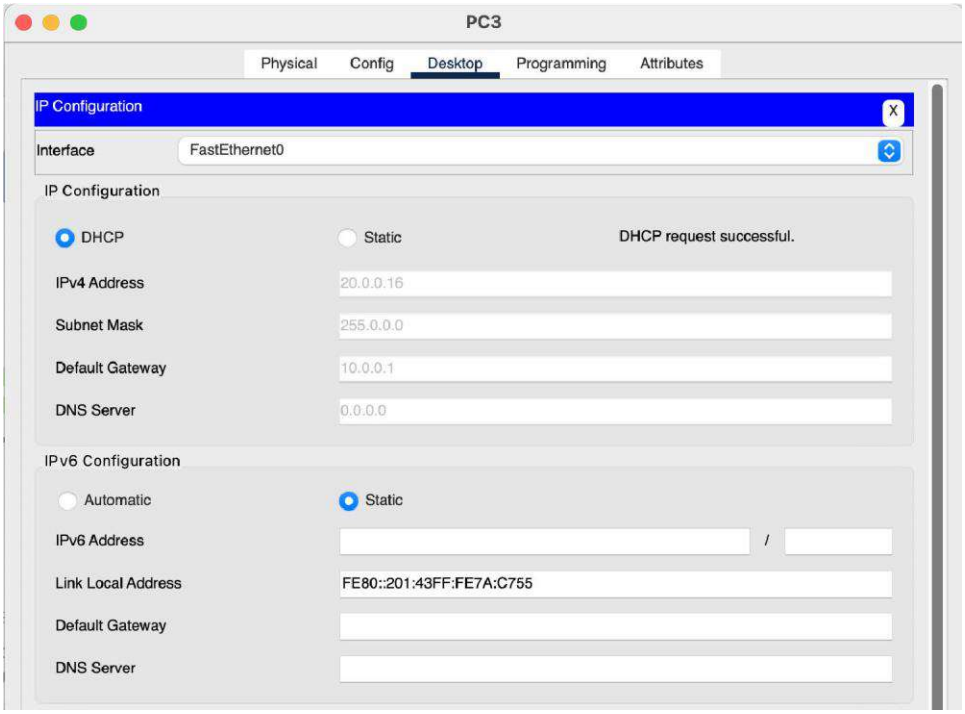


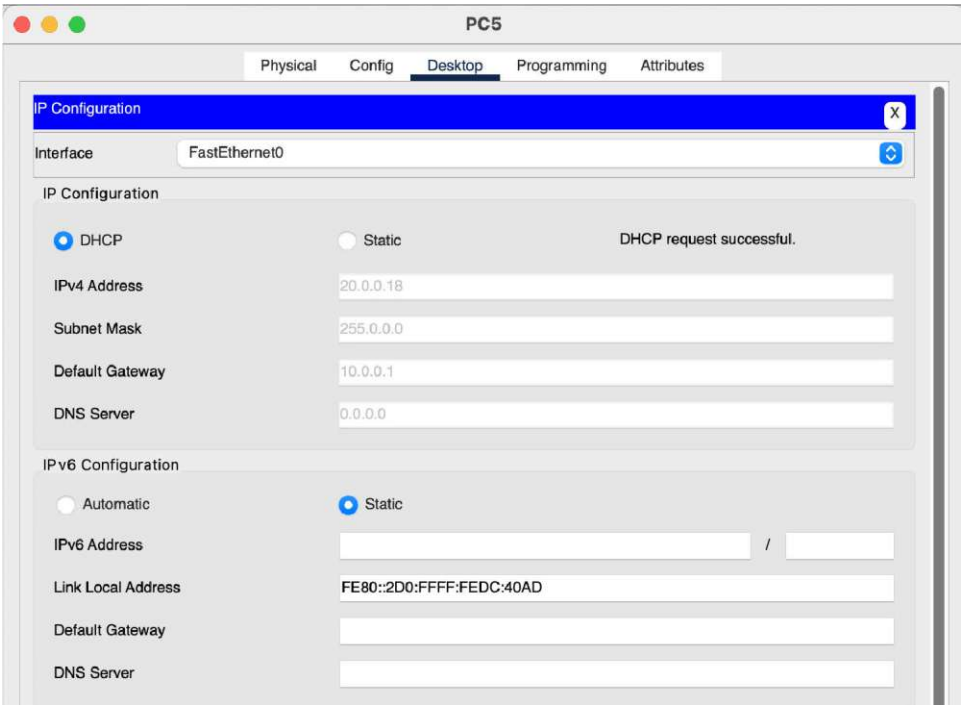
Router:



Obtaining IP:

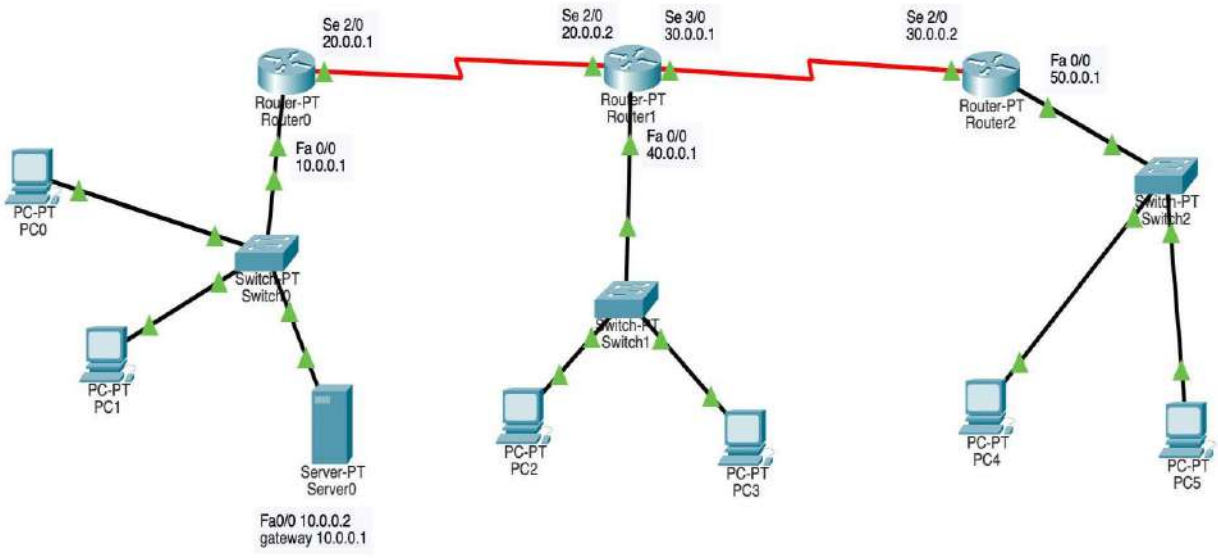

PC0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
10.0.0.14
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::201:96FF:FE4B:2763
Default Gateway
DNS Server


PC3
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
20.0.0.16
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::201:43FF:FE7A:C755
Default Gateway
DNS Server

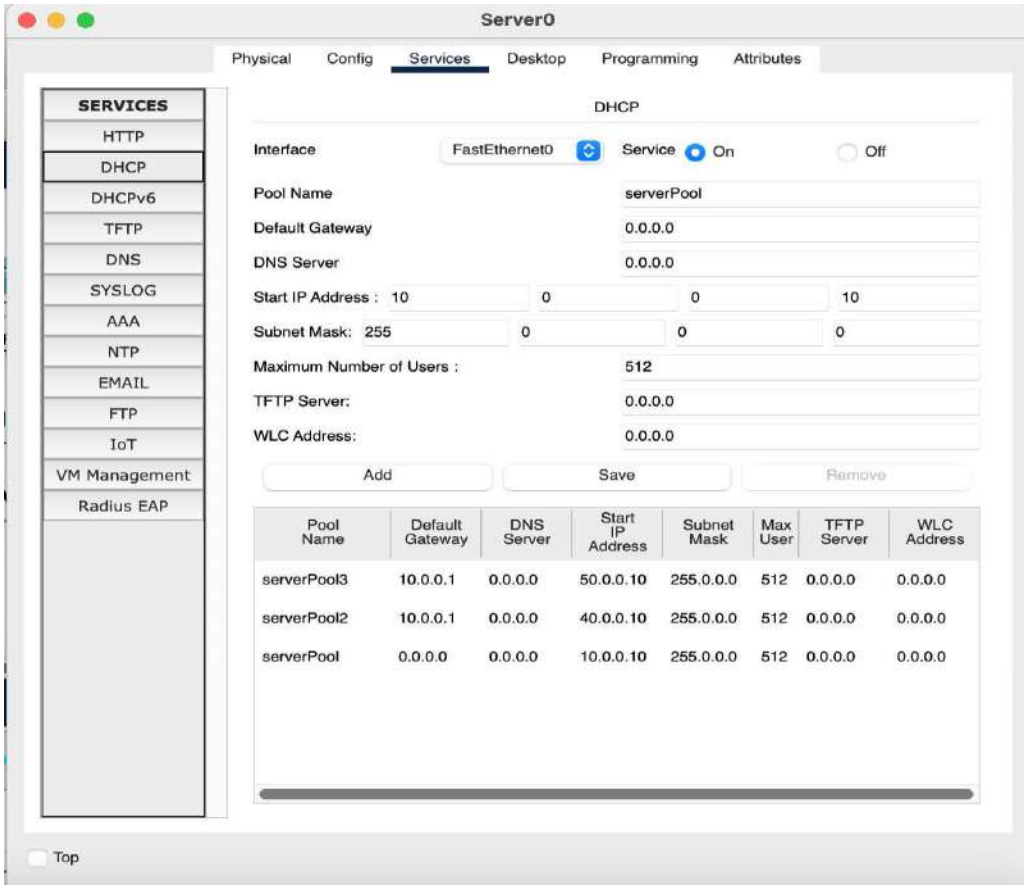


II. Multiple

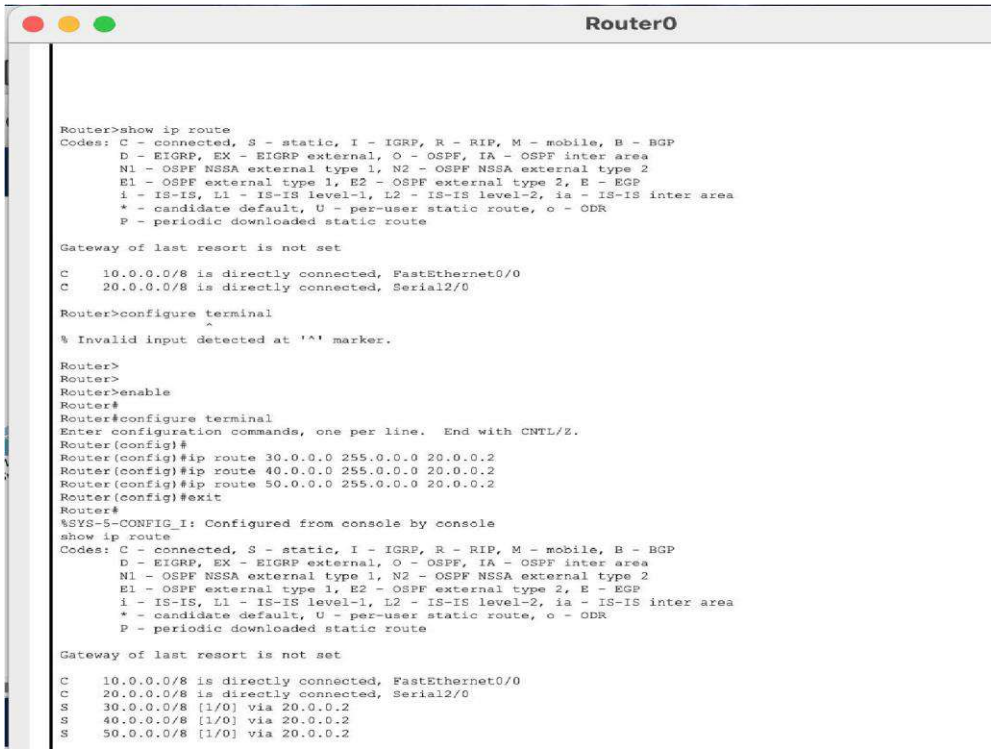
Routers**Topology:**

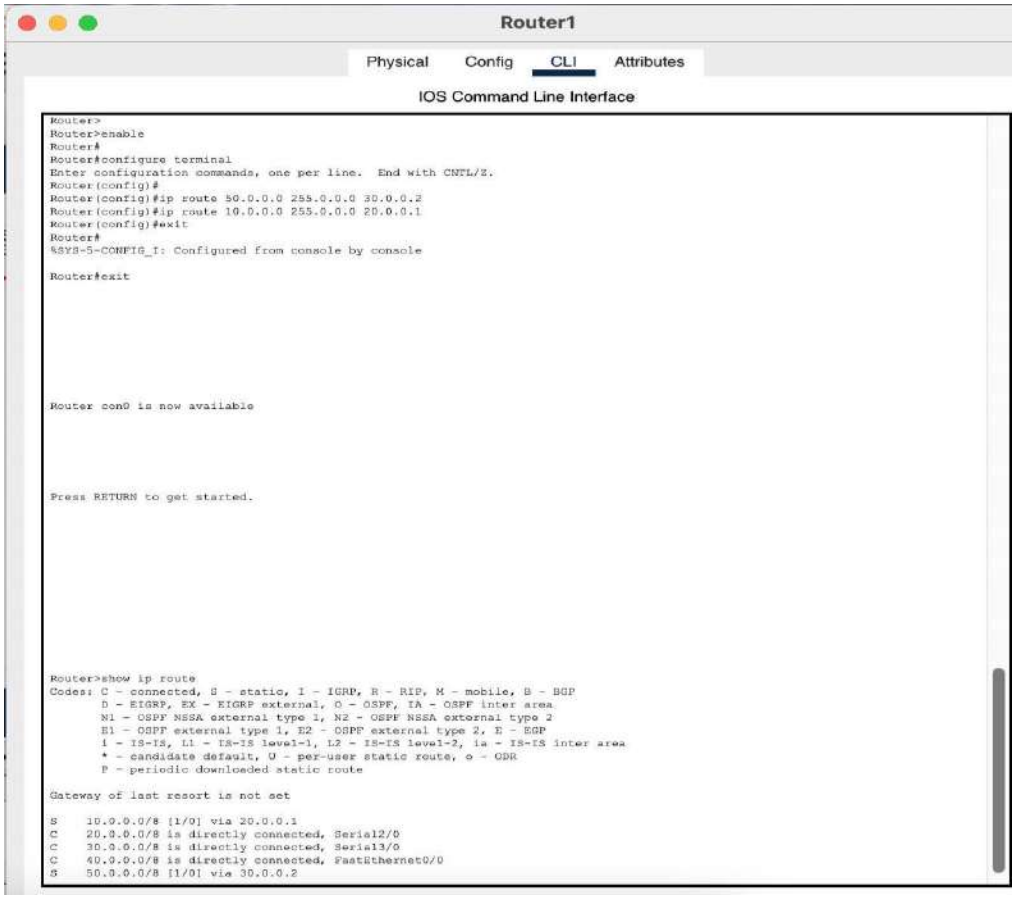


Server:

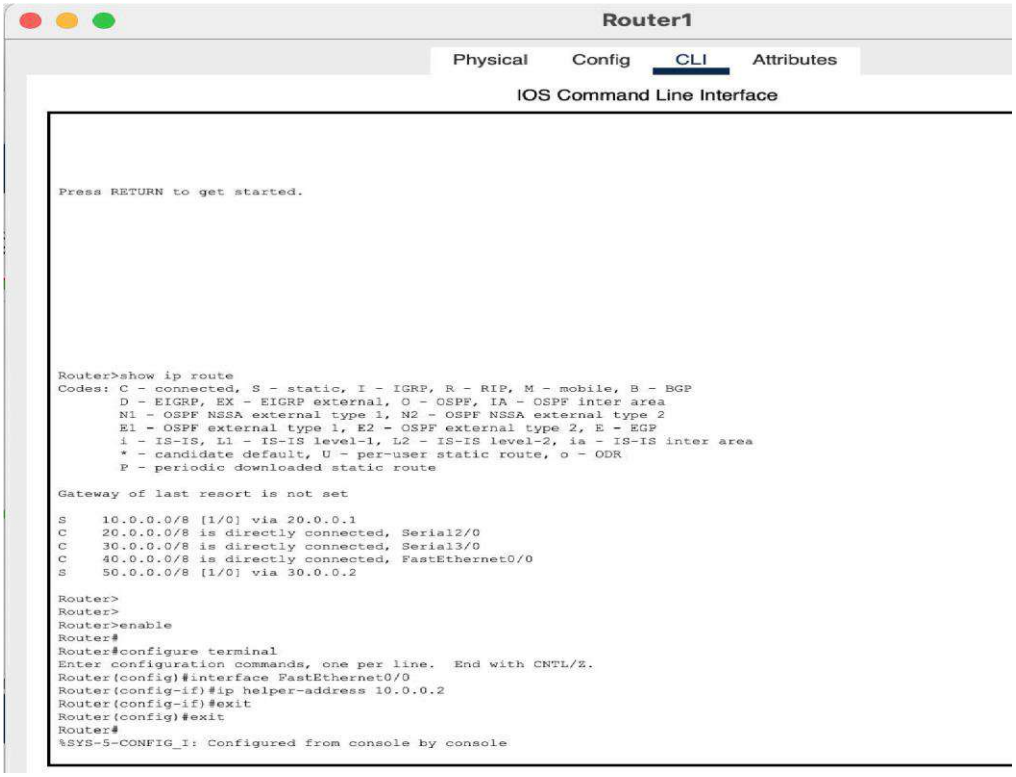


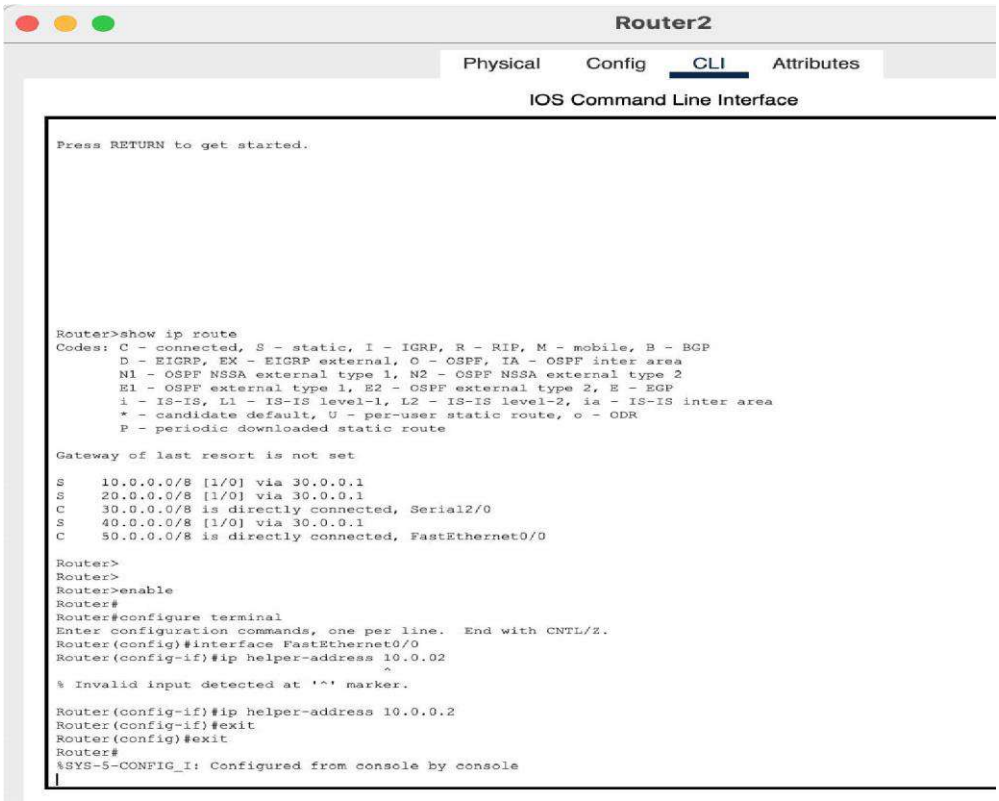
Router: setting IP route



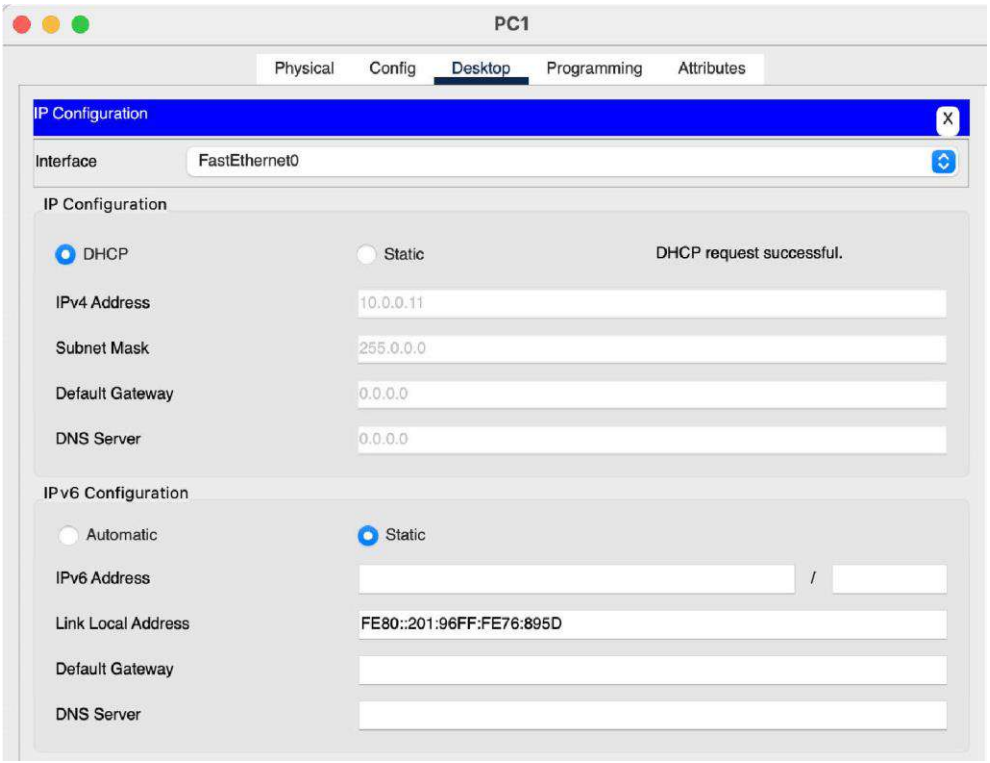


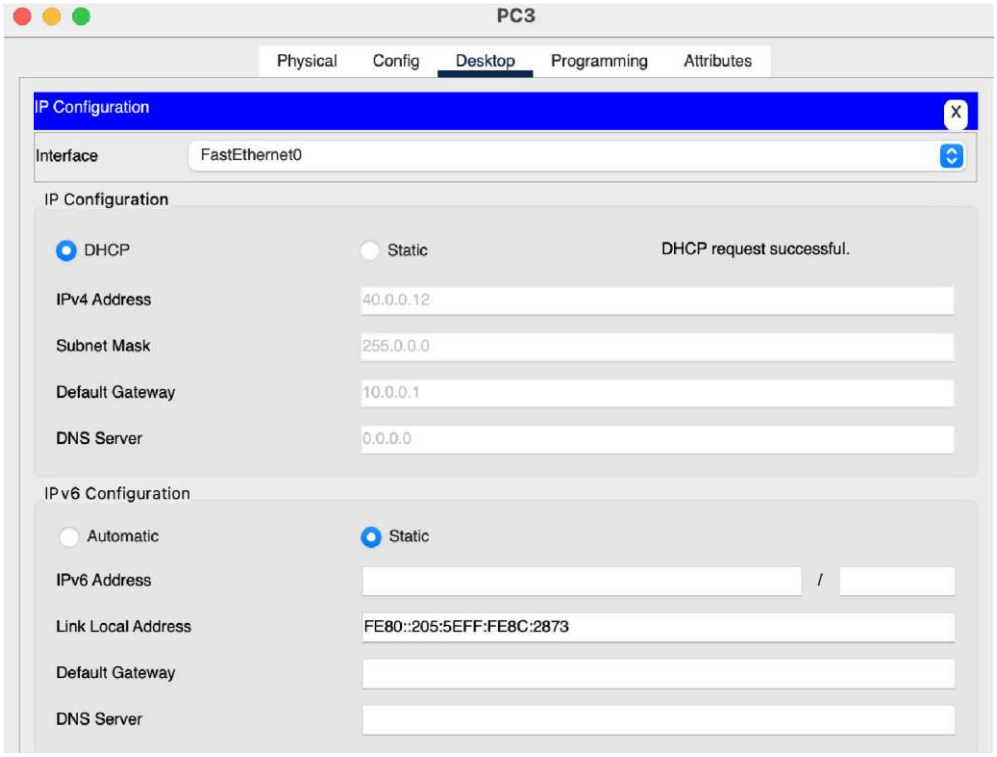
Setting IP helper address-

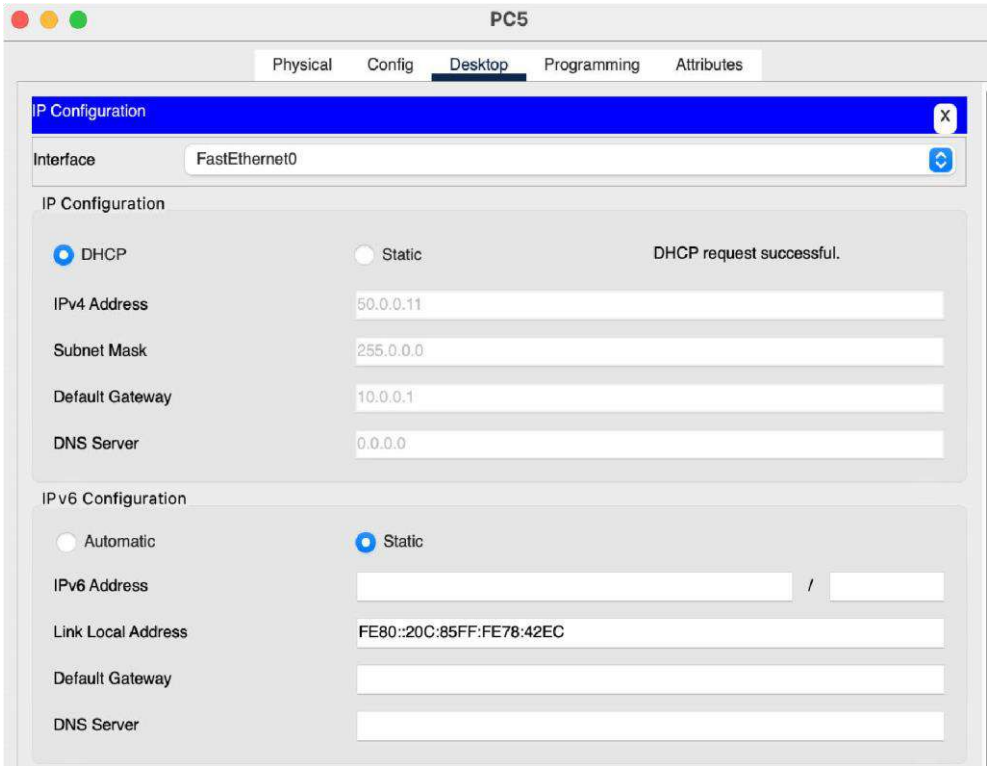




Obtaining IP:


PC1
Physical
Config
Desktop
Programming
Attributes
IP Configuration
X
Interface
FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
10.0.0.11
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::201:96FF:FE76:895D
Default Gateway
DNS Server


PC3
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
40.0.0.12
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::205:5EFF:FE8C:2873
Default Gateway
DNS Server

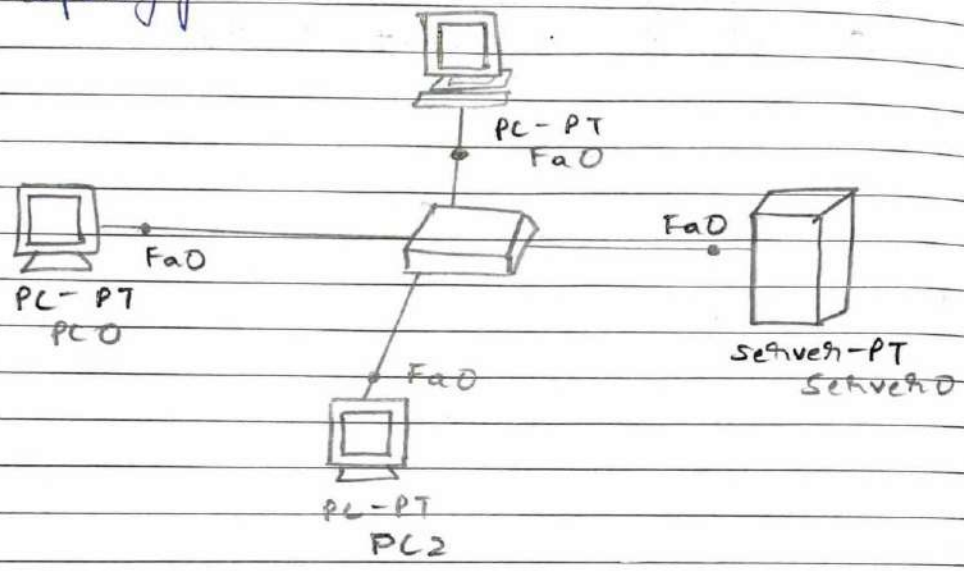

PC5
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
50.0.0.11
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::20C:85FF:FE78:42EC
Default Gateway
DNS Server

URBAN EDGE

Configure DHCP within a LAN and outside LAN.

AIM: To understand how to configure DHCP within a LAN.

Topology:



Procedure:

1. Connect the end device to a switch as shown above.
   Connect a server to the same switch.
2. Set IP address for server 0 as 10.0.0.
3. In server 0 under services click on DHCP.
4. Create a server pool with starting IP address 10.0.0.10 and subnet mask 255.0.0
   click on the Save button
5. Click on any of the PC, navigate to Desktop → IP configuration
6. Choose DHCP under IP configuration.

Result:

→ PC0

IP configuration: Ip-address - 10.0.0.10
Subnet Mask - 255.0.0.0

→ PC 1

IP configuration: IP - address - 10.0.0.11
Subnet mask - 255.0.0.0

→ PC 2

IP configuration: IP-address - 10.0.0.12
Subnet mask - 255.0.0.0

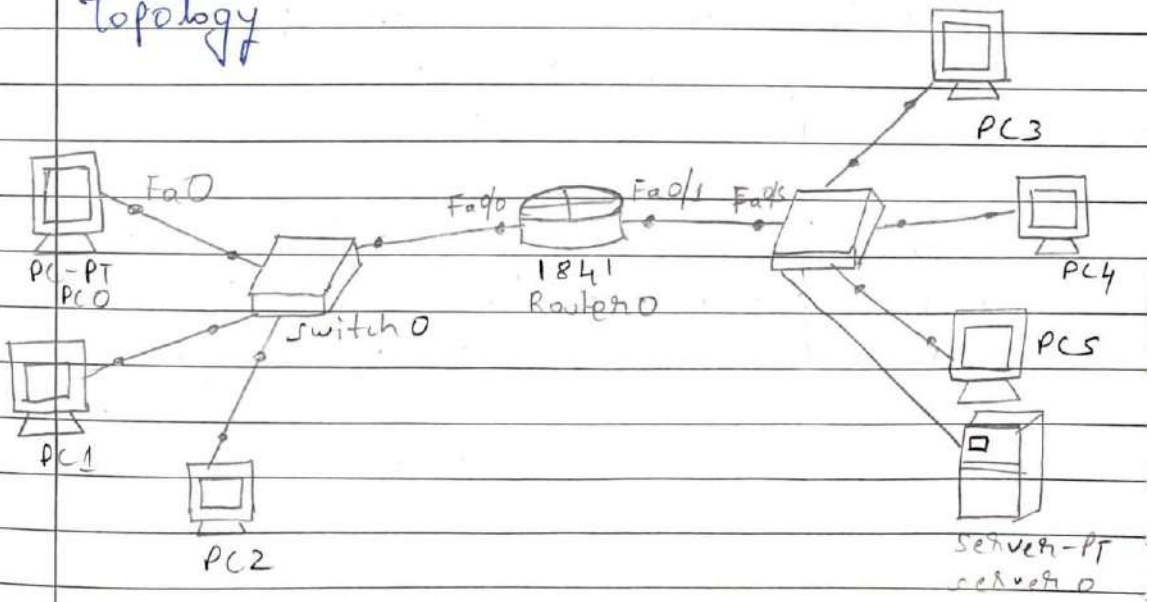
Observation:-

IP addresses for all the end devices in the network i.e PC0, PC1, PC2 is generated dynamically by the server

(b) Outside the LAN

To understand how to configure DHCP outside a LAN (with single Router)

Topology



Procedure:

1. Connect end devices and server to one switch and only end devices to other switch.
2. Connect these two switches to a router.
3. Set IP address and gateway to server0.
   IP: 10.0.0.2
   Gateway: 10.0.0.1
4. Configure the router, Router 1
   Router(config)# Interface Fa4/0
   Router(config)# ip address 10.0.0.1 255.0.0.0
   Router(Config)# no shutdown

   Similarly set Fa0/0 to 20.0.0.1
5. Under Fa4/0 add ip helper-address as server address. Create
6. Create serverPool with starting address 10.0.0.10 and serverpool 2 with starting address 20.0.0.10 gateway 10.0.0.1
7. Click on any of the PCs, navigate to Desktop → IP configuration and choose DHCP.

Adding IP helper address :-
Router(config)# interface Fa0/0
Router(config-if)# ip helper-address 10.0.0.2
Router(config-if)# exit

Result:
End devices in network 10.0.0.0

→PC0
IP address: 10.0.0.14
Subnet mask: 255.0.0.0

→PC1
IP address: 10.0.0.16
Subnet mask: 255.0.0.0

End devices in network 20.0.0.0

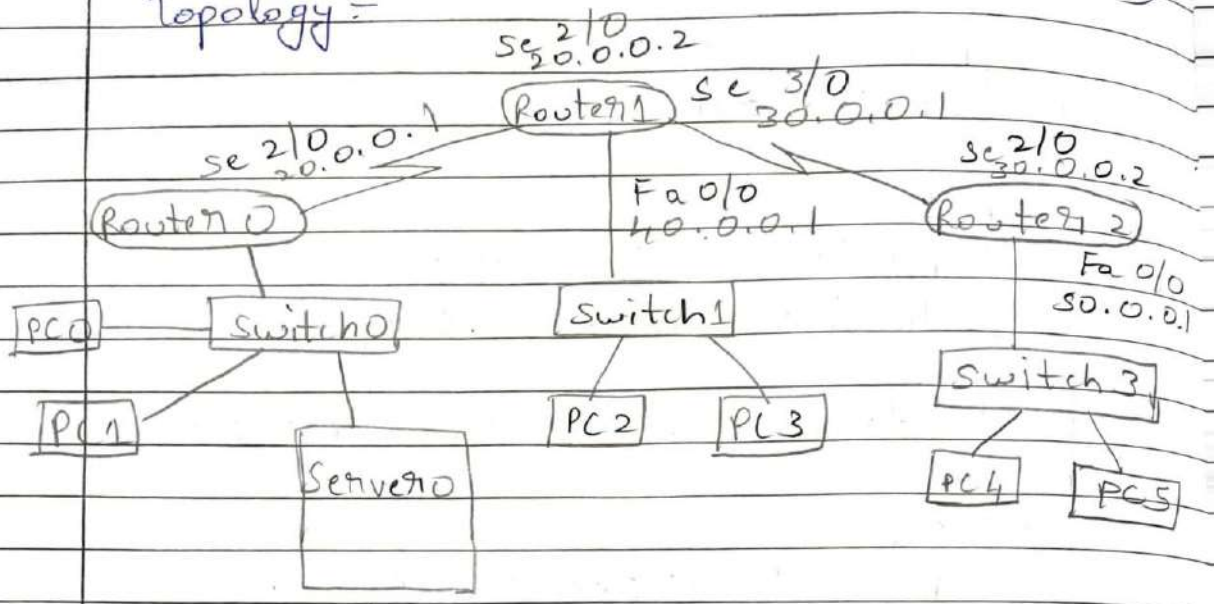
→PC3
IP address: 20.0.0.16
Subnet mask: 255.0.0.0

→PC5
IP address: 20.0.0.18
Subnet mask: 255.0.0.0

observation:
IP address for all end devices are generated dynamically by the server. For end devices in different network, ip helper address is added with that address, the ip address is generated dynamically by the server pointed by ip helper-address.

Aim: To understand how to configure DHCP outside LAN (with Multiple Router)

Topology :-



Fa 0/0 - 10.0.0.2
Gateway - 10.0.0.1

Procedure:

1. Connected the Switch, Router, end device, and server as shown in the topology
2. Set IP address and gateway to server 0.
   IP address - Fa 0/0 => 10.0.0.2
   Gateway - 10.0.0.1
3. Create 3 ServerPool. Navigate to server 0 -> services -> DHCP.
   -> Server Pool
   Starting IP address - 10.0.0.10
   -> Server Pool 2
   Starting IP address - 40.0.0.10
   Gateway - 10.0.0.1
   -> Server Pool 3
   Starting IP address - 50.0.0.10
   Gate way - 10.0.0.1

4. Configure each router with respective IP addresses.

Router 0 - Fa 0/0 - 10.0.0.1
Se 2/0 - 20.0.0.1

Router 1 -
Se 2/0 - 20.0.0.2
Se 3/0 - 30.0.0.1
Fa 0/0 - 40.0.0.1

Router 2 -
Se 2/0 - 30.0.0.2
Fa 0/0 - 50.0.0.1

5. Set up IP route for each network wherever necessary using IP route command.

6. Add IP helper address for networks that doesn't have a server.

→ For network 40.0.0.0
under Fa 0/0
Router(Config) # Interface Fa 0/0
Router(config-if) # ip helper-address 10.0.0.2
Router(config-if) # exit.

→ For Network 50.0.0.0
Router(config) # Interface Fa 0/0
Router(config-if) # ip helper-address 10.0.0.2.
Router(config-if) # exit.

7. Click on any of PC's, navigate to Desktop → IP configuration. and choose DHCP.

Result:

C 10.0.0.0/8 directly, connected Fa0/0
C 20.0.0.0/8 directly connected Se2/0
S 30.0.0.0/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2
S 50.0.0.0/8 [1/0] via 20.0.0.2

Router 1:

S 10.0.0.0/8 [1/0] via 20.0.2.1
C 20.0.0.0/8 directly connected, Se 2/0
C 30.0.0.0/8 directly connected, Se 3/0
C 40.0.0.0/8 Directly Connected, Fa 0/0
S 50.0.0.0/8 [1/0] via 30.0.0.2

Router 2:

S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 [directly connected, Se 2/0
S 40.0.0.0/8 [1/0] via 30.0.0.1
C 50.0.0.0/8 directly connected, Fa 0/0

PC1 -

IP address - 10.0.0.11

PC3 -

IP address - 40.0.0.12
Gateway - 10.0.0.1

PC5 -

IP address - 50.0.0.11
Gateway - 10.0.0.1

Observation:

IP addresses for end devices in all the networks are generated dynamically. For networks that doesn't have servi

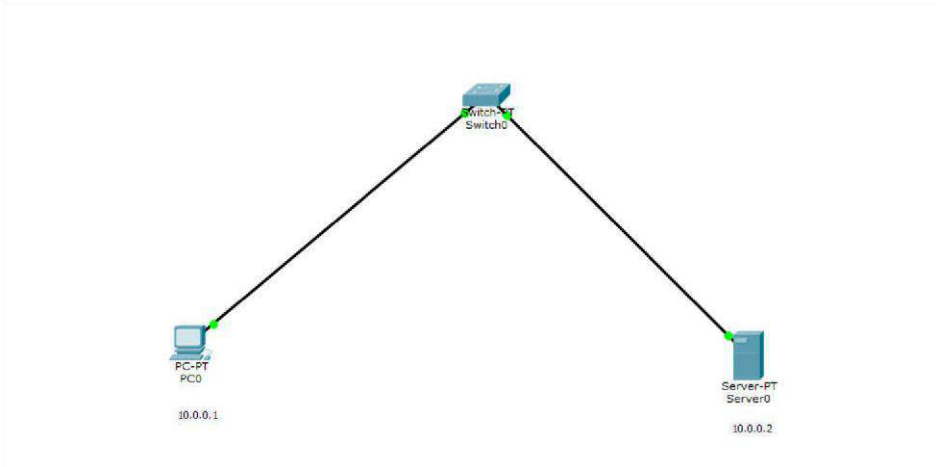
IP helper address is added, which helps in generating the IP address.

Experiment No. 5

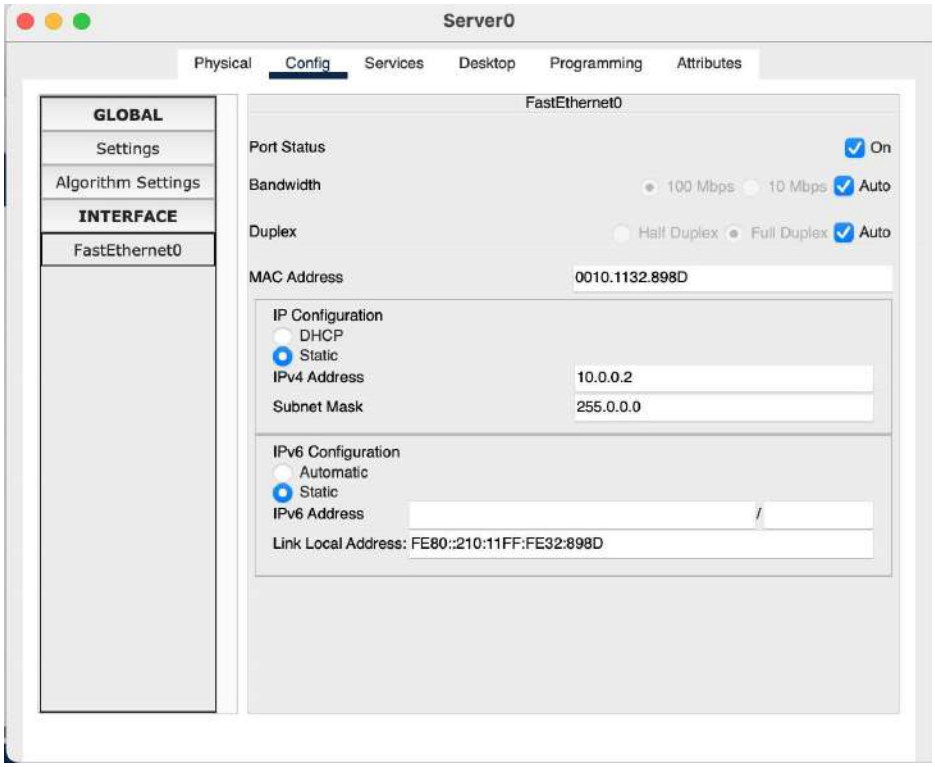
Title:

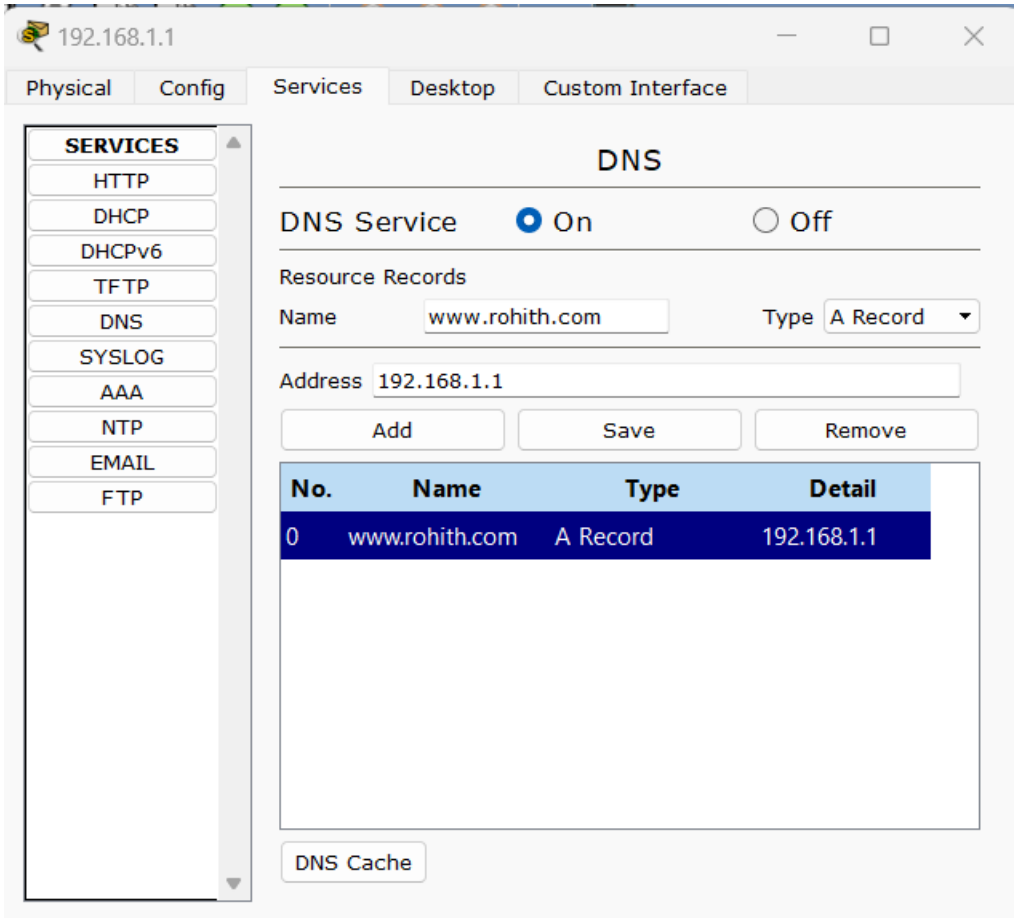
Configure Web Server, DNS within a LAN.

Topology:

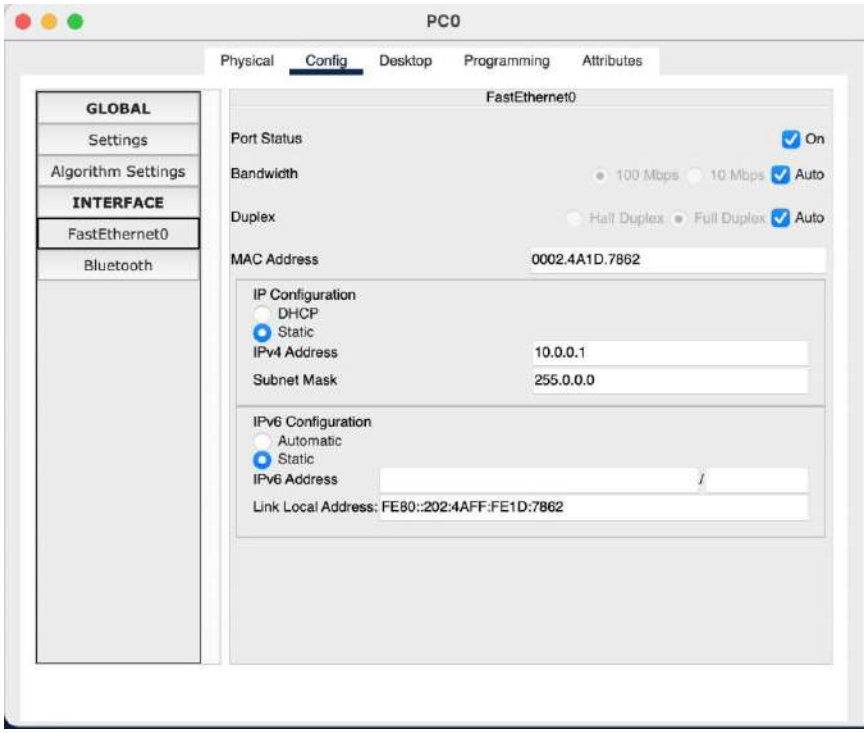


Server:

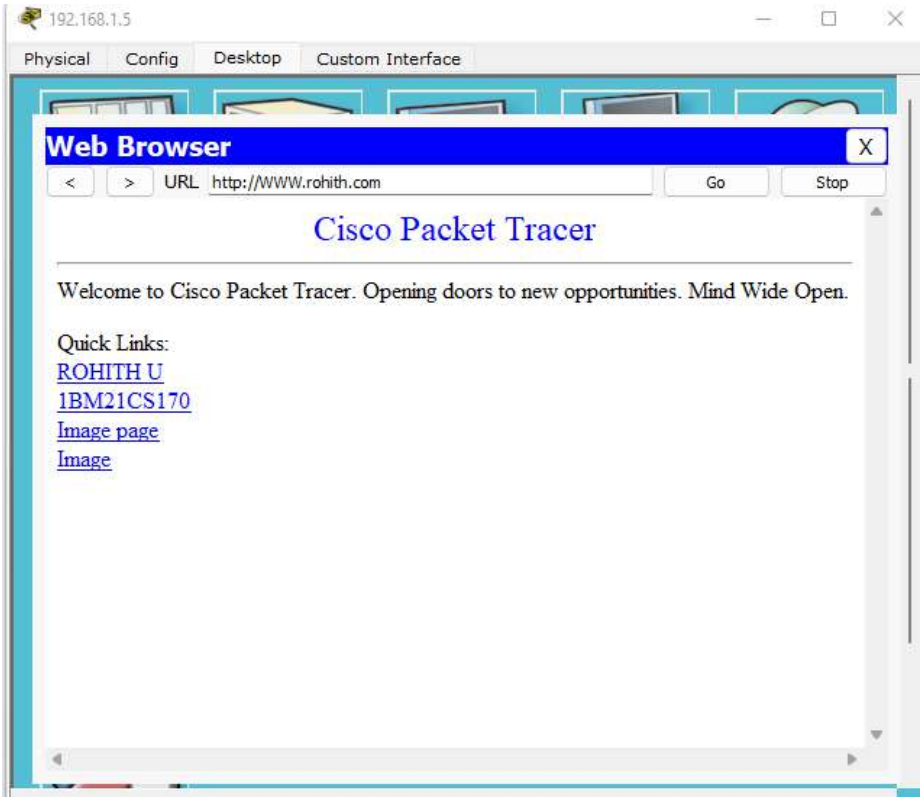



192.168.1.1
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
www.rohith.com
Type
A Record
Address
192.168.1.1
Add
Save
Remove
No.
Name
Type
Detail
0
www.rohith.com
A Record
192.168.1.1
DNS Cache

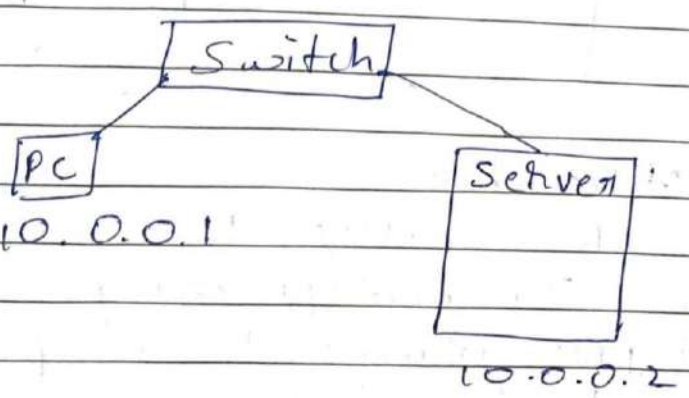
PC:


PC0
Physical
Config
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Bluetooth
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0002.4A1D.7862
IP Configuration
DHCP
Static
IPv4 Address
10.0.0.1
Subnet Mask
255.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address: FE80::202:4AFF:FE1D:7862

Output:


192.168.1.5
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL
http://WWW.rohith.com
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
ROHITH U
1BM21CS170
Image page
Image

(5) Title: Configure Web Server, DNS, within a
Aim: To understand how to configure
web server, DNS within a LAN.
Topology:



Procedure:

(1) Connect the PC, Switch & Server as shown in the above topology.
(2) Set IP address for PC (10.0.0.1) and server (10.0.0.2)
(3) In server, under services click on DNS.
(4) Turn on DNS service, Give name for the website and give the IP address Csame as server IP. Then click add.
(5) Under Services, click on HTTP. Type the HTML code in the index.html file, for the website to display.
(6) Open PC, under Desktop click on Web Browser. Enter the website name with ".com" and hit 'Go'.

URBAN EDGE

Result:

Web Browser

URL: http://name.com

Rohith.U
USN: 1BM21CS170

Observation:

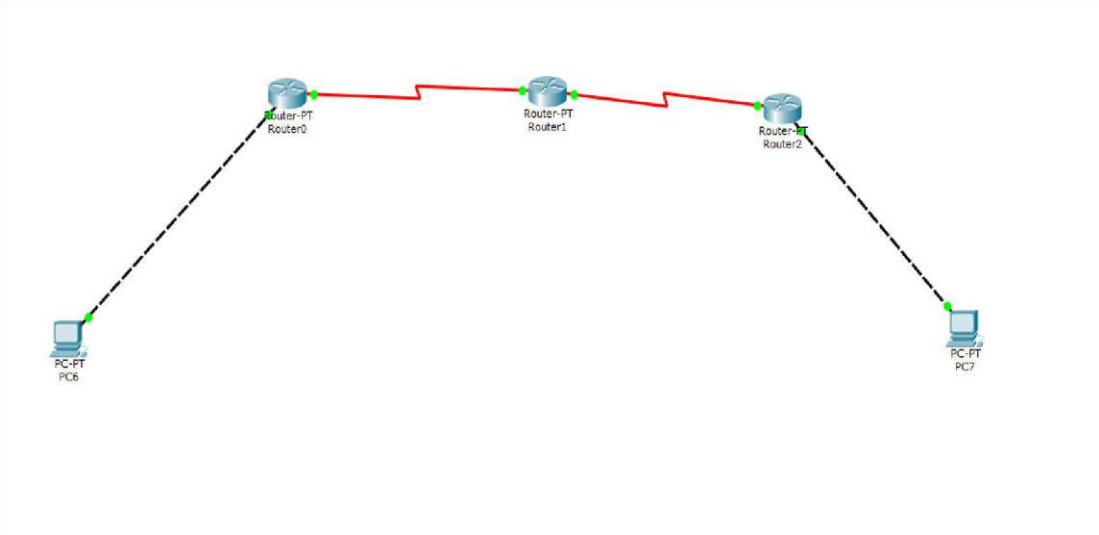
DNS enables to resolve domain names, IP address using different servers.

DNS allows to map domain name to host address.

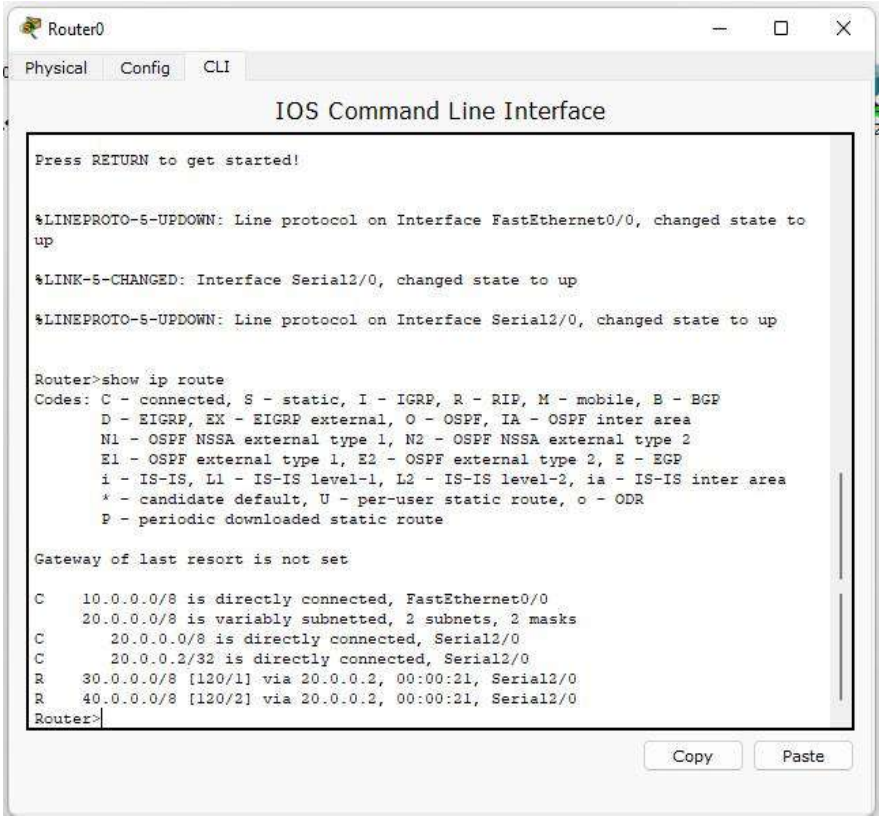
Experiment No. 6

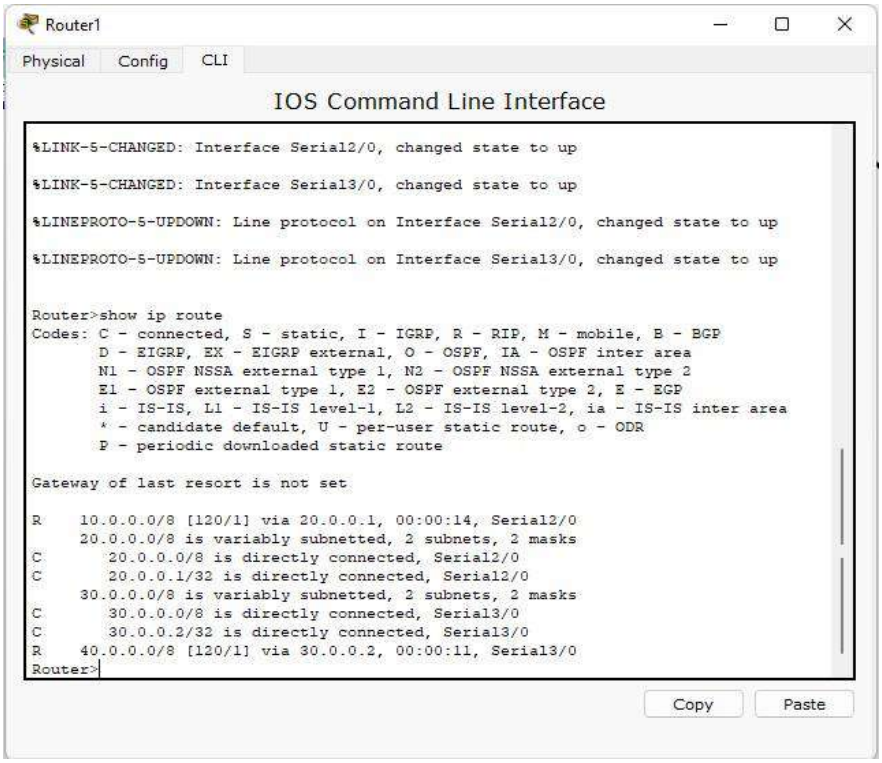
Title:
Configure RIP routing Protocol in Routers

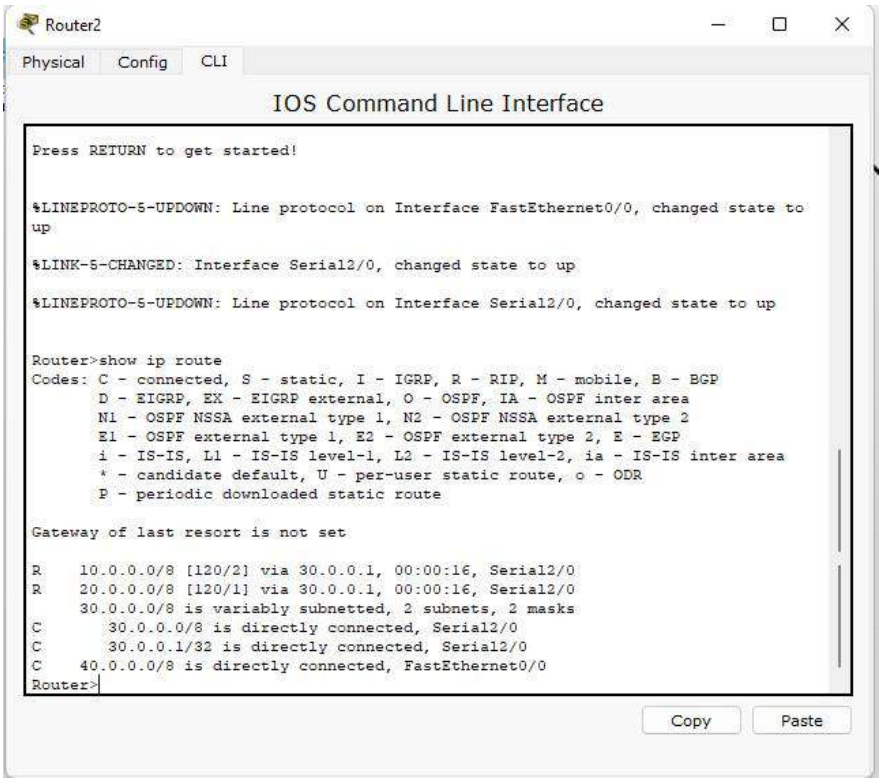
Topology:



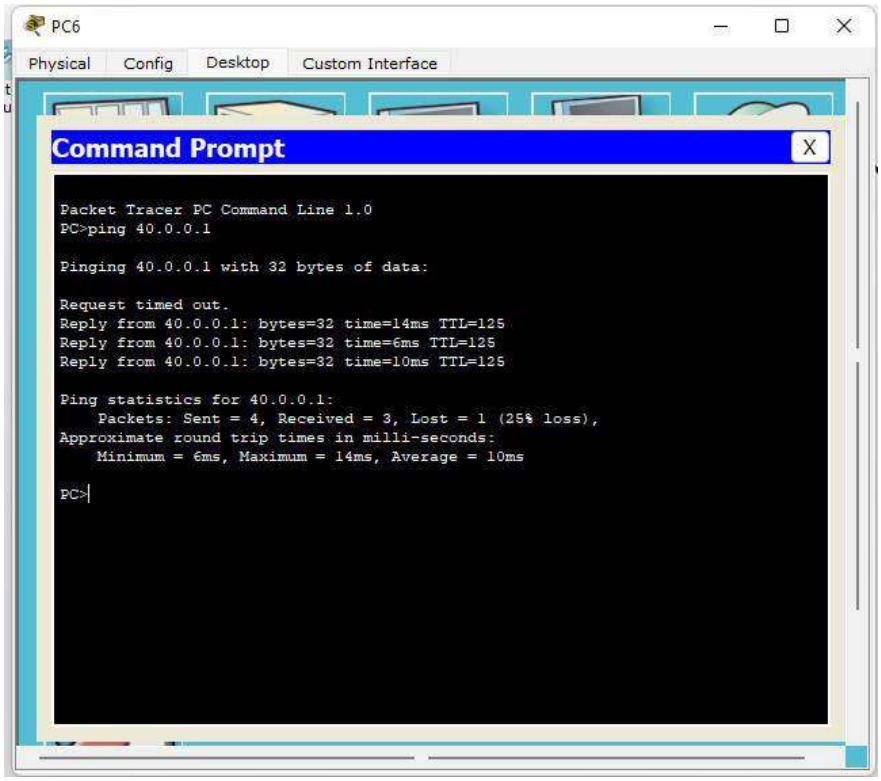
IP Route:



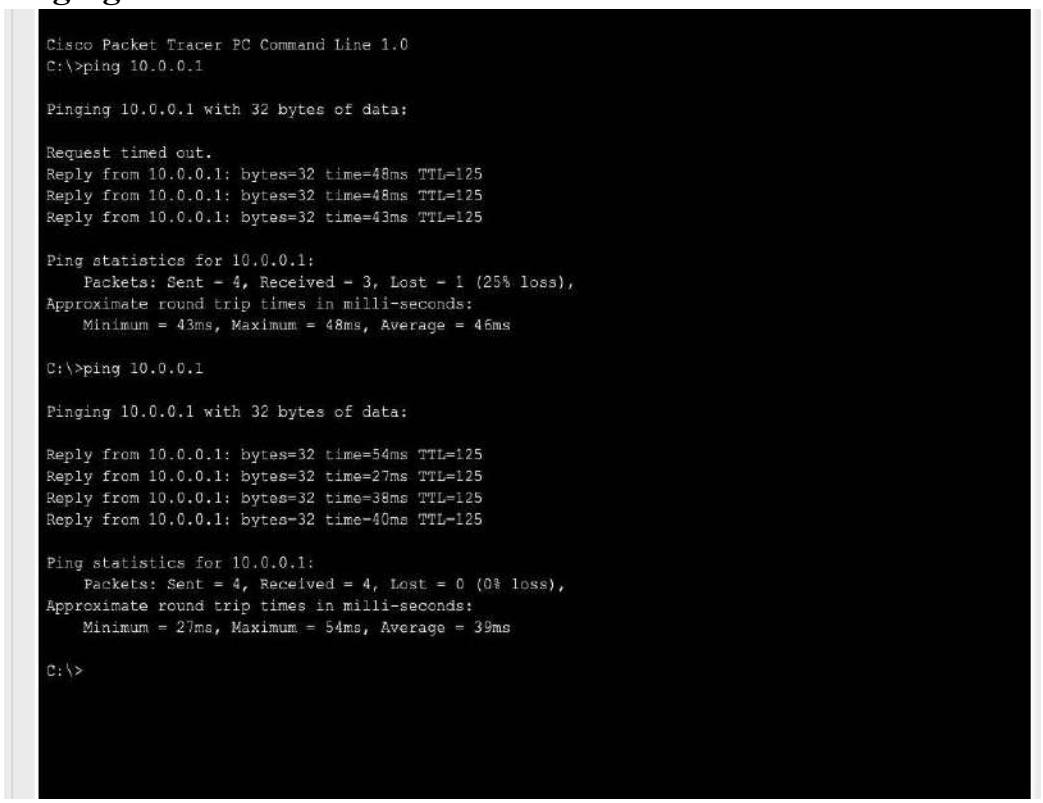




Pinging PC7 from PC6:



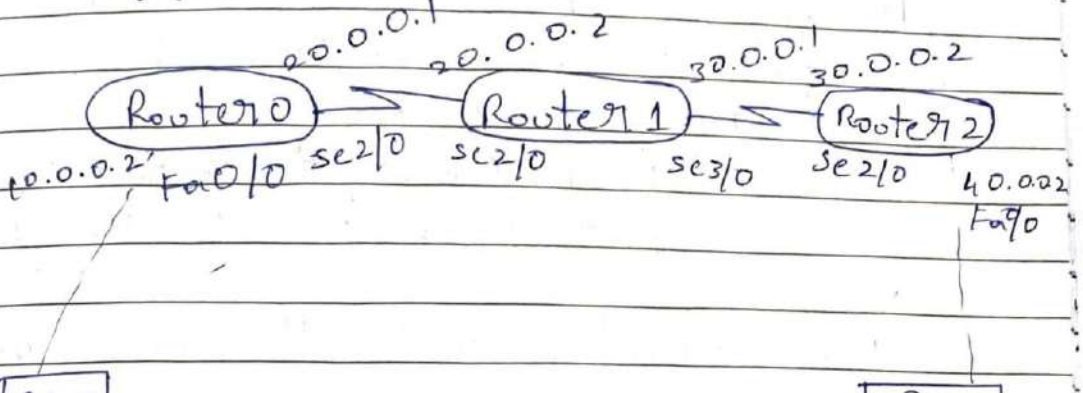
Pinging PC6 from PC7:



⑥ Title: Configure RIP routing protocol in Routers.

Aim: Understand how to configure RIP routing.

Topology:



Procedure:

① Connect the routers & PC's as shown in topology

② Configure the end devices with their IP addresses & subnet mask

PC0 - 10.0.0.1 & PC1 - 40.0.0.1

③ Configure the router with static IP addresses

Router 0 -

Fa 0/0 - 10.0.0.1

Se 2/0 - 20.0.0.1

Router 1 -

se 2/0 - 20.0.0.2

se 3/0 - 30.0.0.1

Router 2 -

Se 2/0 - 30.0.0.2

Fa 0/0 - 40.0.0.2

④ Set clock rate for the necessary interface using the following command

Router(config-if)# clock rate 64000

⑤ Configure RIP route for each router using the following command

For Router 2

Router(config)# router rip
Router(config-router)# network 30.0.0.0
Router(config-router)#network 40.0.0.0

⑥ Click on any one PC and open command prompt under Desktop. Ping the other PC in the different network.

Config using Router Example : Router 0

Router(config-if)#interface serial 2/0.
Router(config-if)# clock rate 64000
Router(config-if)# encapsulation PPP
Router(config-if)# no shutdown
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config)#router rip
Router(config)#network 20.0.0.0

*] Router 1:

Router(config)-# interface Serial 3/0
Router(config-if)# clock rate 64000
Router(config-if)#en capsulation PPP
Router(config-if)# no shutdown

Router (config)# router rip
Router (config-router)# network 20.0.0.0
Router (config)# router rip
Router (config-if)# network 30.0.0.0

* Router 2
Router (config)# router rip
Router (config-router)# network 40.0.0.0
Router (config)# router rip
Router (config-router)# network 30.0.0.0

Result:
Router > show ip route

Route 0

C 10.0.0.0/8 is directly connected Fa0/0
20.0.0.0/8 is variably connected
2 subnets, 2 masks
C 20.0.0.0/8 is directly connected
Se 2/0
20.0.0.2/32 is directly connected
Se 2/0
R 30.0.0.0/8 [120/1] via 20.00.2,
0 0:00:21, Se2/0
R 40.0.0.0/8 [120/2] via 20.0.0.2,
00:00:21, Se2/0

Router 1

R 10.0.0.0/8 [120/1] via 20.0.0.1,
00.00:14, Se 2/0
C 20.0.0.0/8 is directly connected,
Se 2/0
C 30.0.0.0/8 is directly connected,
Se3/0

R 40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:11, Se3/0

Router 2

R 10.0.0.0/8 [120/1] via 30.0.0.1, 00:00:16, Se2/0
R 20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:16, Se2/0
C 30.0.0.0/8 directly connected, Se2/0
C 40.0.0.0/8 directly connected, Fa0/0

Pinging PC1 from PC0

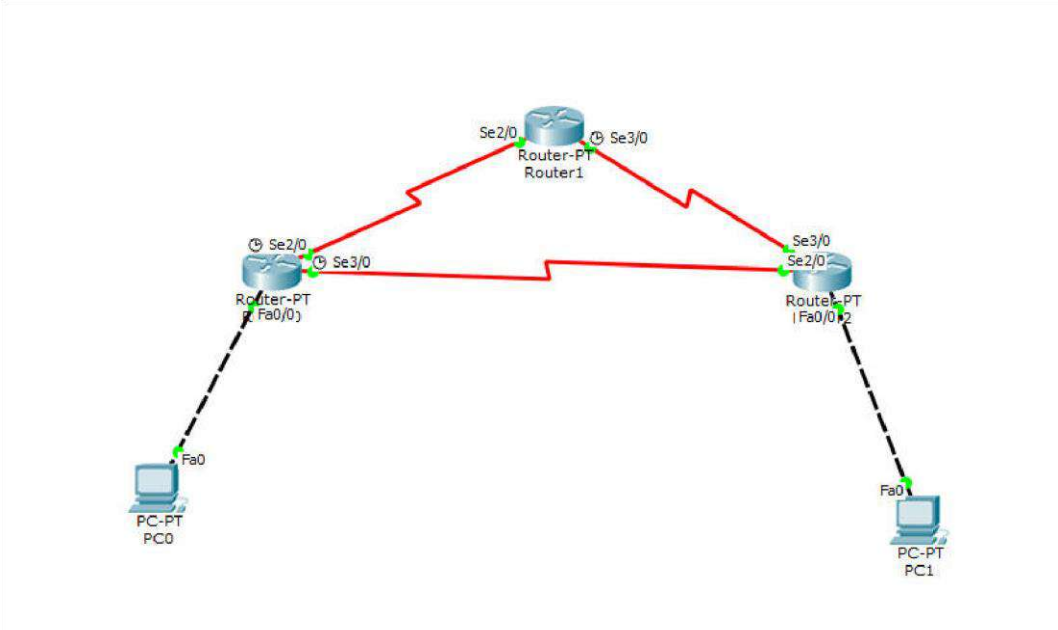
PC0
PC> ping 40.0.0.1
Pinging 40.0.0.1 bytes=32 with 32 bytes of data
Reply from 40.0.0.1 bytes = 32 time = 14ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = 6ms TTL = 125
Reply from 40.0.0.1 bytes = 32 time = 10ms TTL = 125
Packet : Sent = 4, Received = 4, Loss = 0
Min = 6ms, Max = 14ms, Avg = 10ms

Observations:
Static and RIP routing can be configured to each router using CLI, RIP routing is added using 'router rip' command. When 'show ip route' is executed in the CLI C denotes connected and R denotes the RIP route.

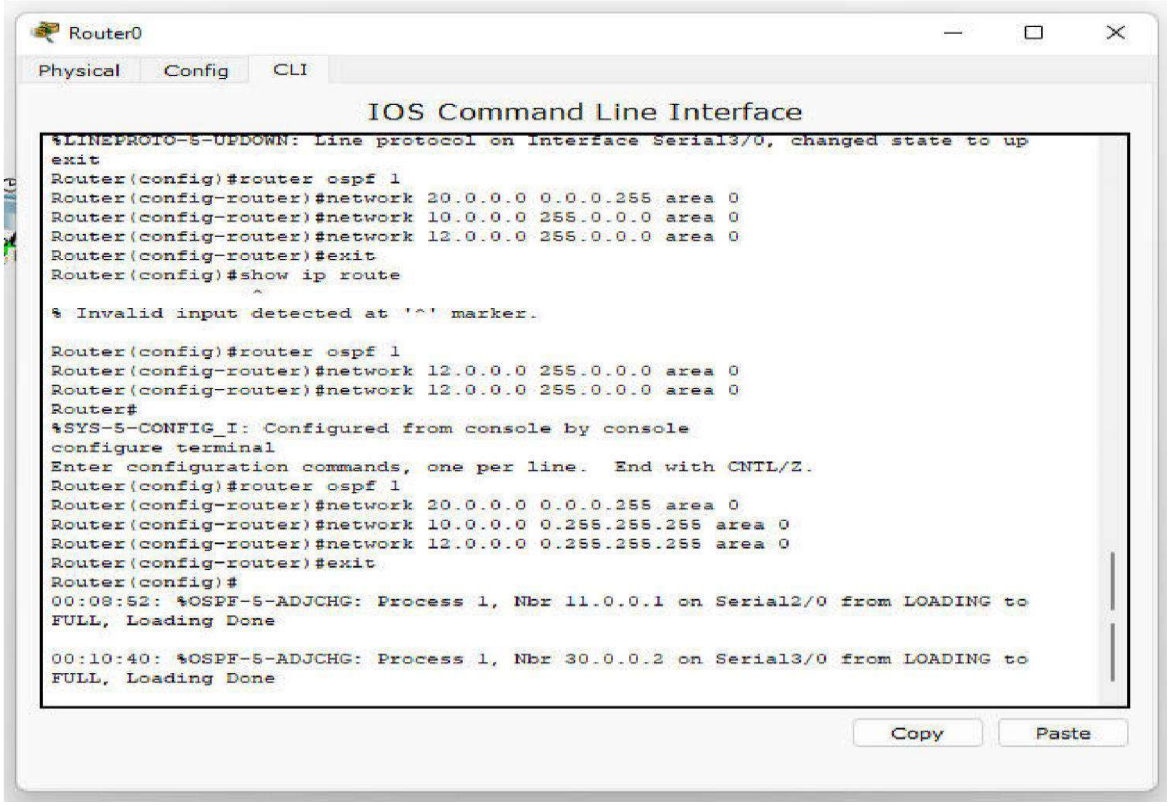
Experiment No. 7

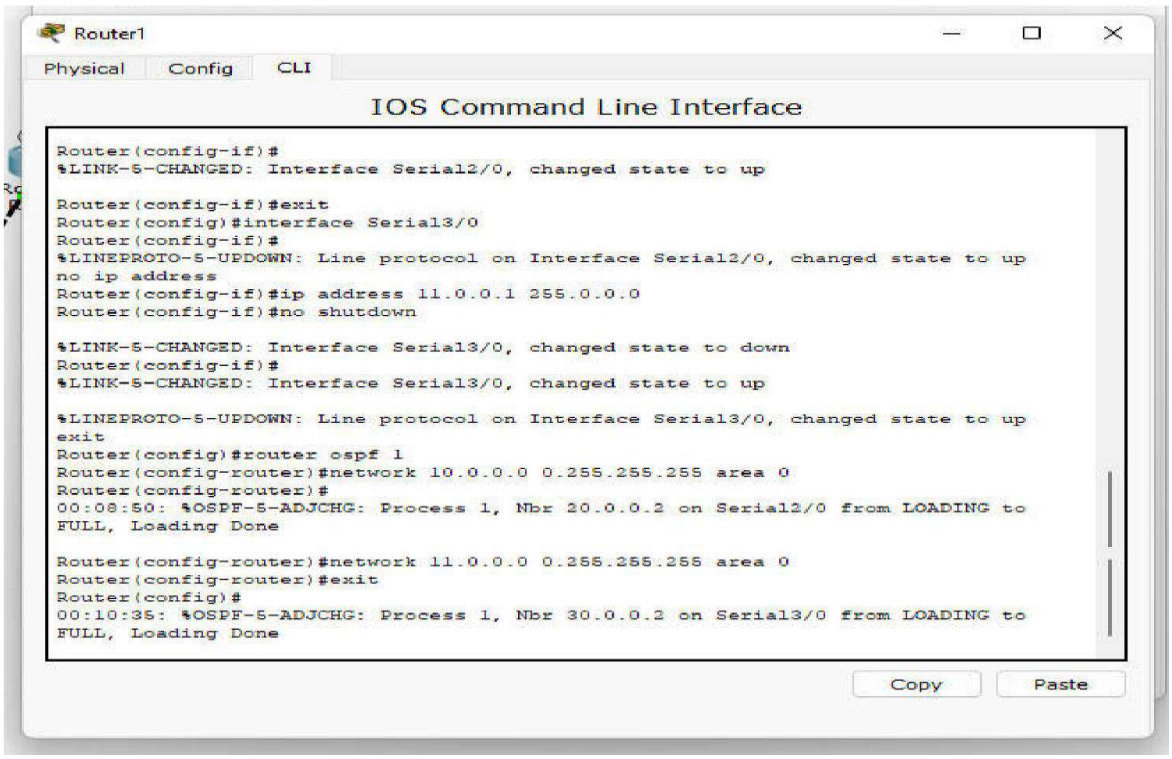
Title:
Configure OSPF routing protocol

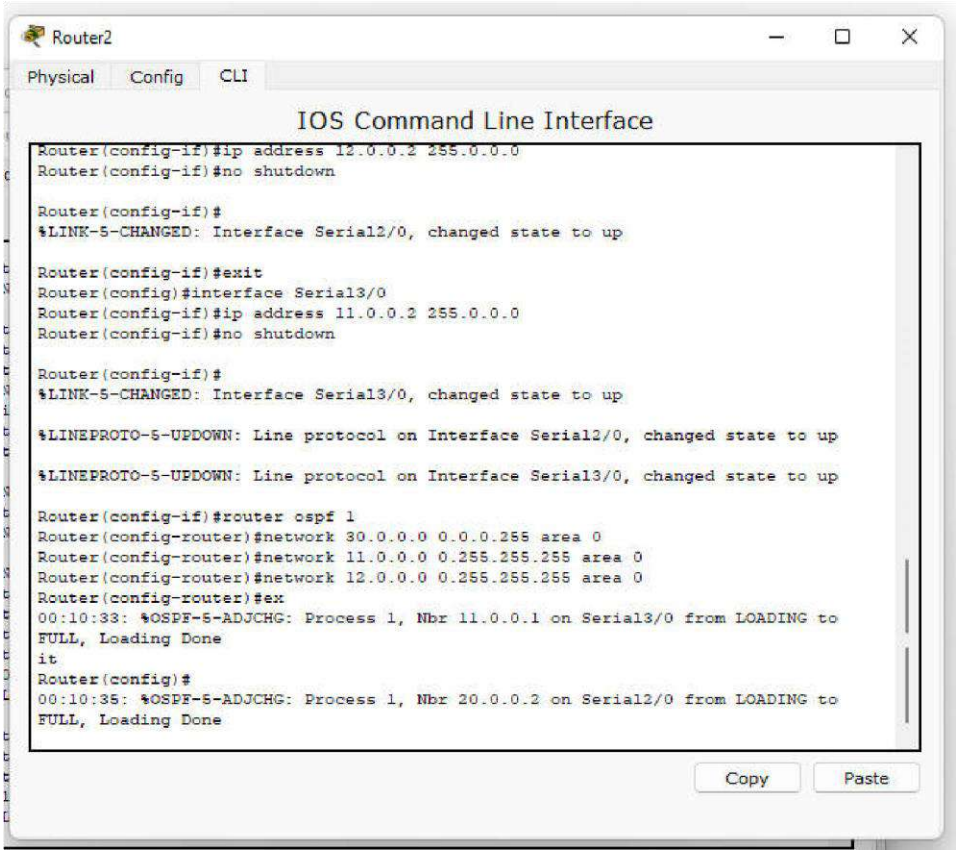
Topology:

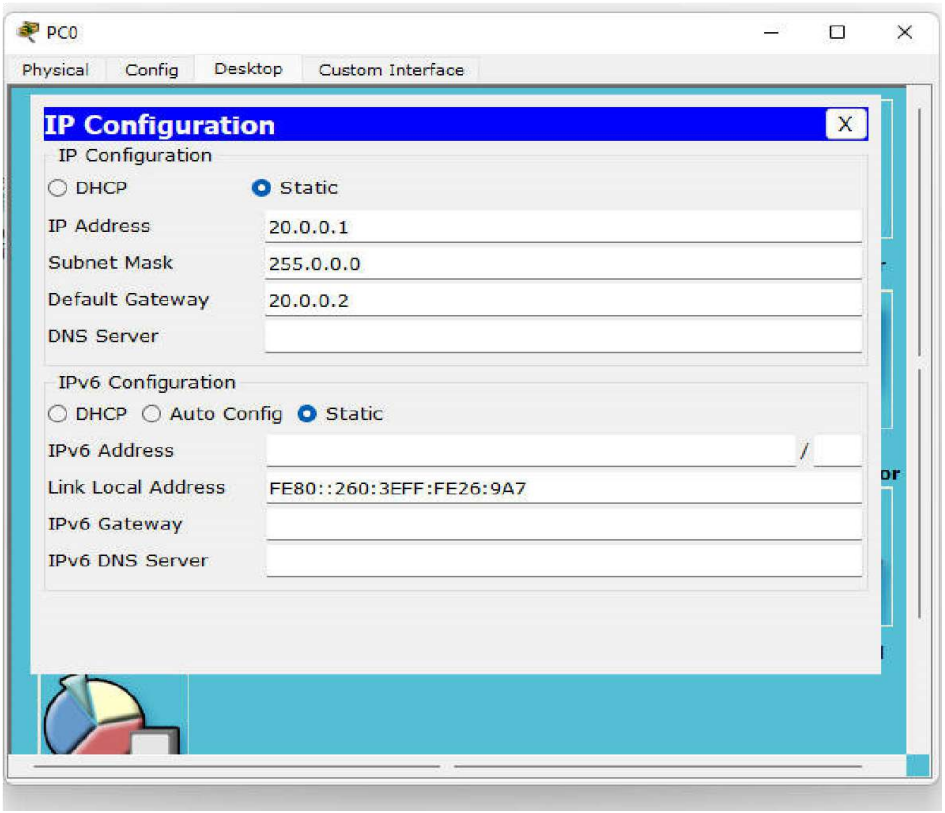


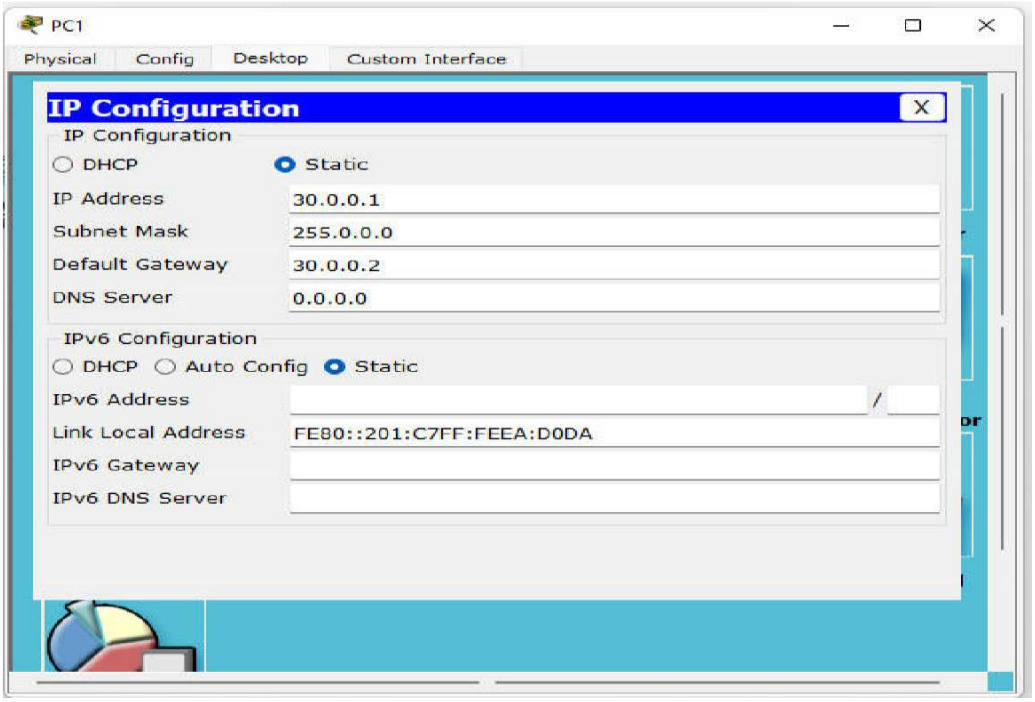
Configure Routers:

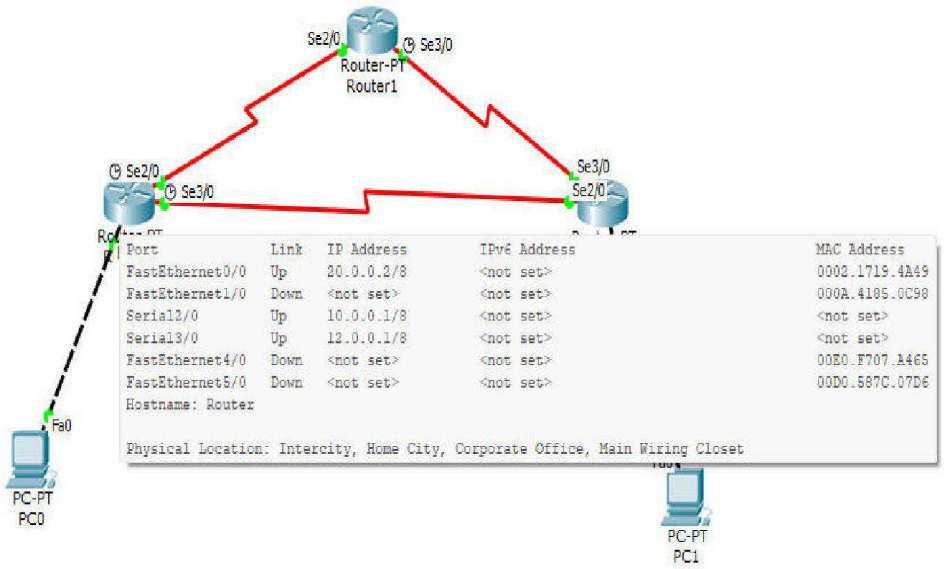

Router0
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
exit
Router(config)#router ospf 1
Router(config-router)#network 20.0.0.0 0.0.0.255 area 0
Router(config-router)#network 10.0.0.0 255.0.0.0 area 0
Router(config-router)#network 12.0.0.0 255.0.0.0 area 0
Router(config-router)#exit
Router(config)#show ip route
% Invalid input detected at '^' marker.
Router(config)#router ospf 1
Router(config-router)#network 12.0.0.0 255.0.0.0 area 0
Router(config-router)#network 12.0.0.0 255.0.0.0 area 0
Router#
%SYS-5-CONFIG_I: Configured from console by console
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 20.0.0.0 0.0.0.255 area 0
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#network 12.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#
00:08:52: %OSPF-5-ADJCHG: Process 1, Nbr 11.0.0.1 on Serial2/0 from LOADING to FULL, Loading Done
00:10:40: %OSPF-5-ADJCHG: Process 1, Nbr 30.0.0.2 on Serial3/0 from LOADING to FULL, Loading Done
Copy
Paste

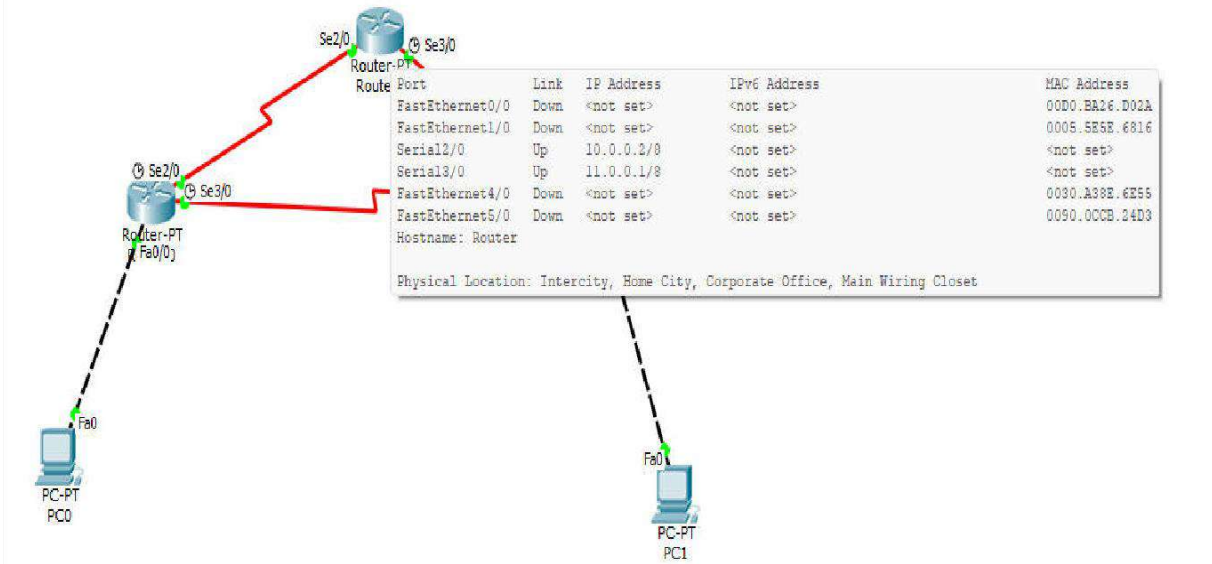

Router1
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
no ip address
Router(config-if)#ip address 11.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
exit
Router(config)#router ospf 1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#
00:08:50: %OSPF-5-ADJCHG: Process 1, Nbr 20.0.0.2 on Serial2/0 from LOADING to FULL, Loading Done
Router(config-router)#network 11.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#
00:10:35: %OSPF-5-ADJCHG: Process 1, Nbr 30.0.0.2 on Serial3/0 from LOADING to FULL, Loading Done
Copy
Paste

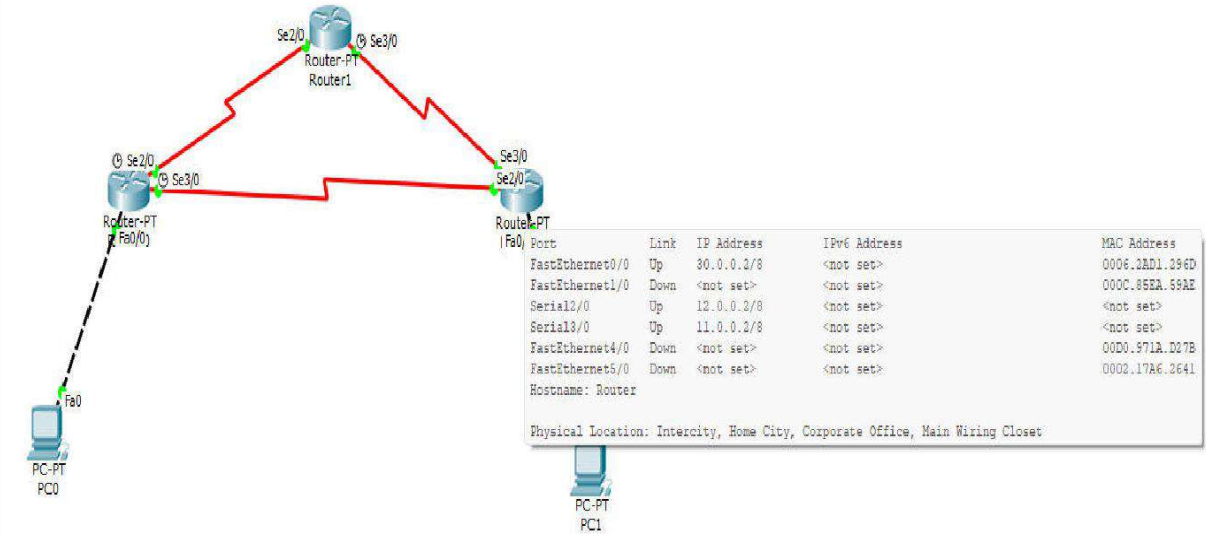

Router2
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#ip address 12.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 11.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
Router(config-if)#router ospf 1
Router(config-router)#network 30.0.0.0 0.0.0.255 area 0
Router(config-router)#network 11.0.0.0 0.255.255.255 area 0
Router(config-router)#network 12.0.0.0 0.255.255.255 area 0
Router(config-router)#ex
00:10:33: %OSPF-5-ADJCHG: Process 1, Nbr 11.0.0.1 on Serial3/0 from LOADING to FULL, Loading Done
it
Router(config)#
00:10:35: %OSPF-5-ADJCHG: Process 1, Nbr 20.0.0.2 on Serial2/0 from LOADING to FULL, Loading Done
Copy
Paste


PC0
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
20.0.0.1
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.2
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::260:3EFF:FE26:9A7
IPv6 Gateway
IPv6 DNS Server

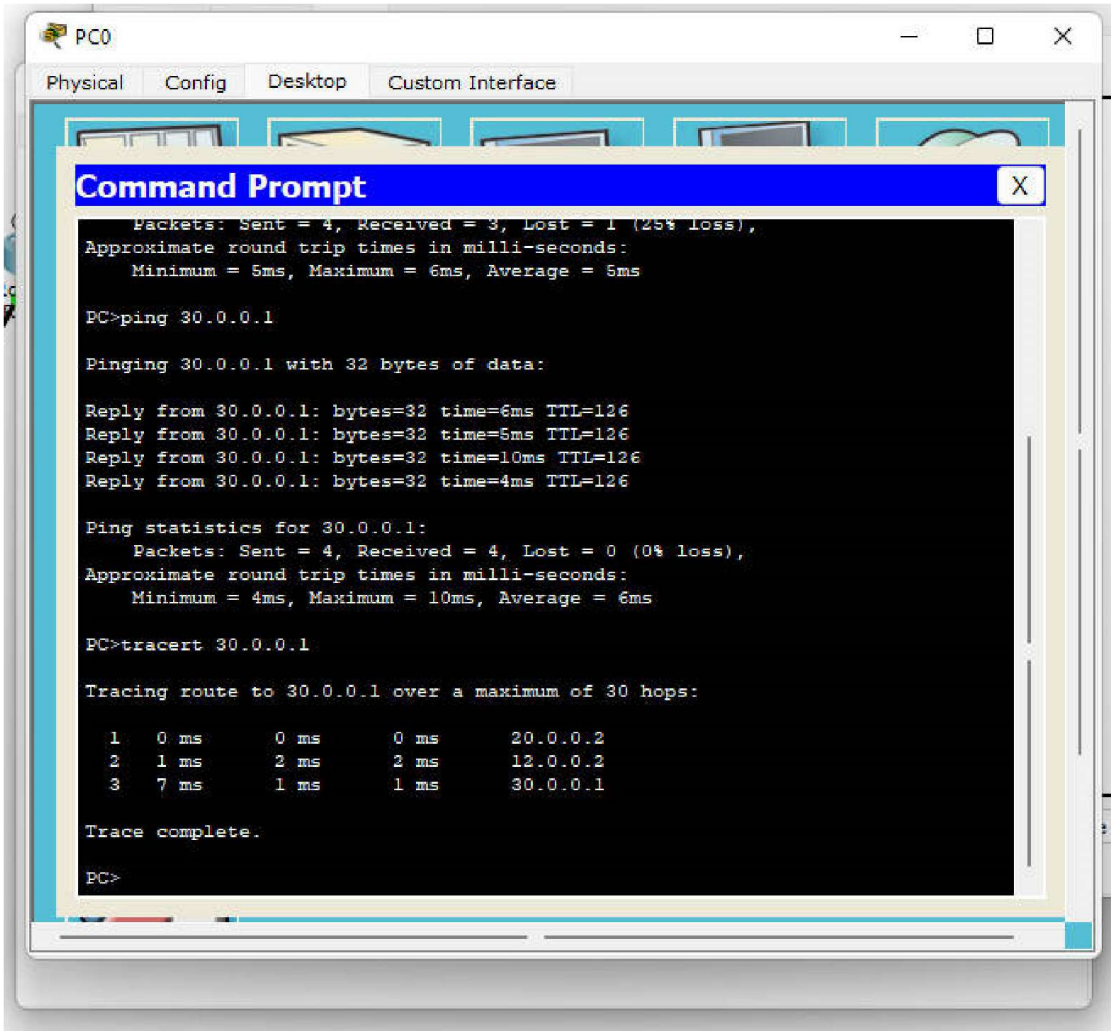

PC1
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
IP Address
30.0.0.1
Subnet Mask
255.0.0.0
Default Gateway
30.0.0.2
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::201:C7FF:FEEA:D0DA
IPv6 Gateway
IPv6 DNS Server


Se2/0
Se3/0
Router-PT
Router1
Se2/0
Se3/0
Se3/0
Se2/0
Port Link IP Address IPv6 Address MAC Address
FastEthernet0/0 Up 20.0.0.2/8 <not set> 0002.1719.4A49
FastEthernet1/0 Down <not set> <not set> 000A.4185.0C98
Serial2/0 Up 10.0.0.1/8 <not set> <not set>
Serial3/0 Up 12.0.0.1/8 <not set> <not set>
FastEthernet4/0 Down <not set> <not set> 00E0.F707.A465
FastEthernet5/0 Down <not set> <not set> 00D0.587C.07D6
Hostname: Router
Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet
Fa0
PC-PT
PC0
PC-PT
PC1


Se2/0
Se3/0
Router-PT
Route
Se2/0
Se3/0
Router-PT
Fa0/0
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Port Link IP Address IPv6 Address MAC Address
FastEthernet0/0 Down <not set> <not set> 00D0.BA26.D02A
FastEthernet1/0 Down <not set> <not set> 0005.5E5E.6816
Serial2/0 Up 10.0.0.2/8 <not set> <not set>
Serial3/0 Up 11.0.0.1/8 <not set> <not set>
FastEthernet4/0 Down <not set> <not set> 0030.A38E.6E55
FastEthernet5/0 Down <not set> <not set> 0090.0CCB.24D3
Hostname: Router
Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet

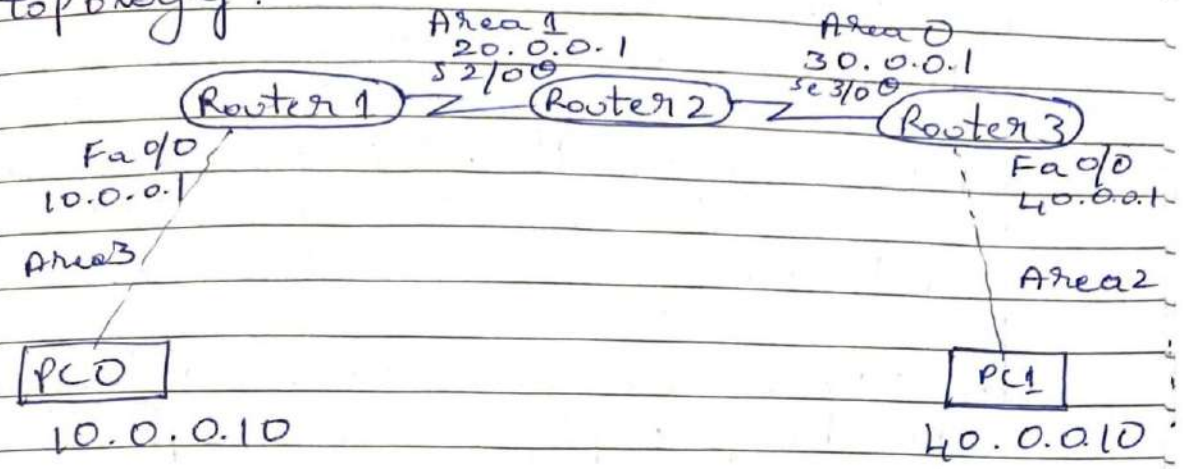

Se2/0
Se3/0
Router-PT
Router1
Se2/0
Se3/0
Router-PT
Fa0/0
Se3/0
Se2/0
Fa0
PC-PT
PC0
PC-PT
PC1
Port Link IP Address IPv6 Address MAC Address
FastEthernet0/0 Up 30.0.0.2/8 <not set> 0006.2AD1.296D
FastEthernet1/0 Down <not set> <not set> 000C.85EA.59AE
Serial2/0 Up 12.0.0.2/8 <not set> <not set>
Serial3/0 Up 11.0.0.2/8 <not set> <not set>
FastEthernet4/0 Down <not set> <not set> 00D0.971A.D27B
FastEthernet5/0 Down <not set> <not set> 0002.17A6.2641
Hostname: Router
Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet

Output:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 6ms, Average = 5ms
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=6ms TTL=126
Reply from 30.0.0.1: bytes=32 time=5ms TTL=126
Reply from 30.0.0.1: bytes=32 time=10ms TTL=126
Reply from 30.0.0.1: bytes=32 time=4ms TTL=126
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 10ms, Average = 6ms
PC>tracert 30.0.0.1
Tracing route to 30.0.0.1 over a maximum of 30 hops:
1 0 ms 0 ms 0 ms 20.0.0.2
2 1 ms 2 ms 2 ms 12.0.0.2
3 7 ms 1 ms 1 ms 30.0.0.1
Trace complete.
PC>

AIM: To find the shortest route using ospf commands

Topology :-



Procedure:

1. Create a topology as shown above
2. Configure each router with the IP addresses for required interface. Set up clock rate for interface having DCE symbol.

Router 2

```
R2(config)# interface Serial 3/0
R2(config-if)# ip address 30.0.0.1
                           255.0.0.0
R2(config-if)# encapsulation ppp
R2(config-if)# clock rate 64000
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# interface Se2/0
R2(config-if)# ip address 20.0.0.2
                           255.0.0.0
R2(config-if)# encapsulation ppp
R2(config-if)# no shutdown
R2(config-if)# exit
```

3. Configure ospf routing for each router.

```
R1(config) # router ospf 1
R1(config-router) # router-id 1.1.1.1
R1(config-router) # network 10.0.0.0
        0.255.255.255 area 3
R1(config-router) # network 20.0.0.0
        0.255.255.255 area 1
R1(config-router) # exit.
```

4. There must be one interface up to keep ospf process up. So it's better to configure loopback address to routers. It is a virtual interface that never goes down.

```
R1(config) # interface se2/0
R1(config-if) # ip interface loopback 0
R1(config-if) # ip add 172.16.1.252
        255.255.0.0
R1(config-if) # no shutdown.
```

This is done for each router.

5. Still R3 doesn't know about Area 3. This can be verified using "show ip route" command.

```
R3 # show ip route
O IA  20.0.0.0/8 via 30.0.0.1
        se1/0
C   40.0.0.0/8 directly connected
        Fa2/0
L   30.0.0.0/8 directly connected,
        se1/0.
```

so a virtual link b/w R1 & R2 must be created, to connect area 3 to area 0.

R1(config)# router ospf 1
R1(config-router)# area 1 virtual-link 2.2.2.2
R1(config-router)# exit
Do the same for R2.

6. Ping PC0 from PC1 to check connectivity.

Result:

R3# show ip route
O IA 20.0.0.0/8 via 30.0.0.1, Se1/0
O IA 10.0.0.0/8 via 30.0.0.1, Se1/0
C 40.0.0.0/8 directly connected, Fa2/0
C 30.0.0.0/8 directly connected, Se1/0

R2# show ip route
C 20.0.0.0/8 directly connected, Se2/0
C 30.0.0.0/8 directly connected, Se3/0
C 172.16.0.0/8 directly connected, loopback

R1# show ip route
O IA 40.0.0.0/8 via 20.0.0.2, Se2/0
O 30.0.0.0/8 via 20.0.0.2, Se2/0
C 10.0.0.0/8 directly connected, Fa0/0
C 20.0.0.0/8 directly connected, Se2/0

Ping PC0 from PC1

PC1 > ping 10.0.0.10
Reply from 10.0.0.10, bytes = 32
time = 13 ms TTL = 125
Reply from 10.0.0.10, bytes = 32
time = 11 ms TTL = 125
Reply from 10.0.0.10, bytes = 32 time = 6ms
TTL = 125
Reply from 10.0.0.10, bytes = 32 time = 7ms
TTL = 125

Ping Statistics:
Packets: Sent = 4, Received = 4, Lost = 0
(0% Loss) Min = 6ms, MAX = 13ms, Avg = 9ms

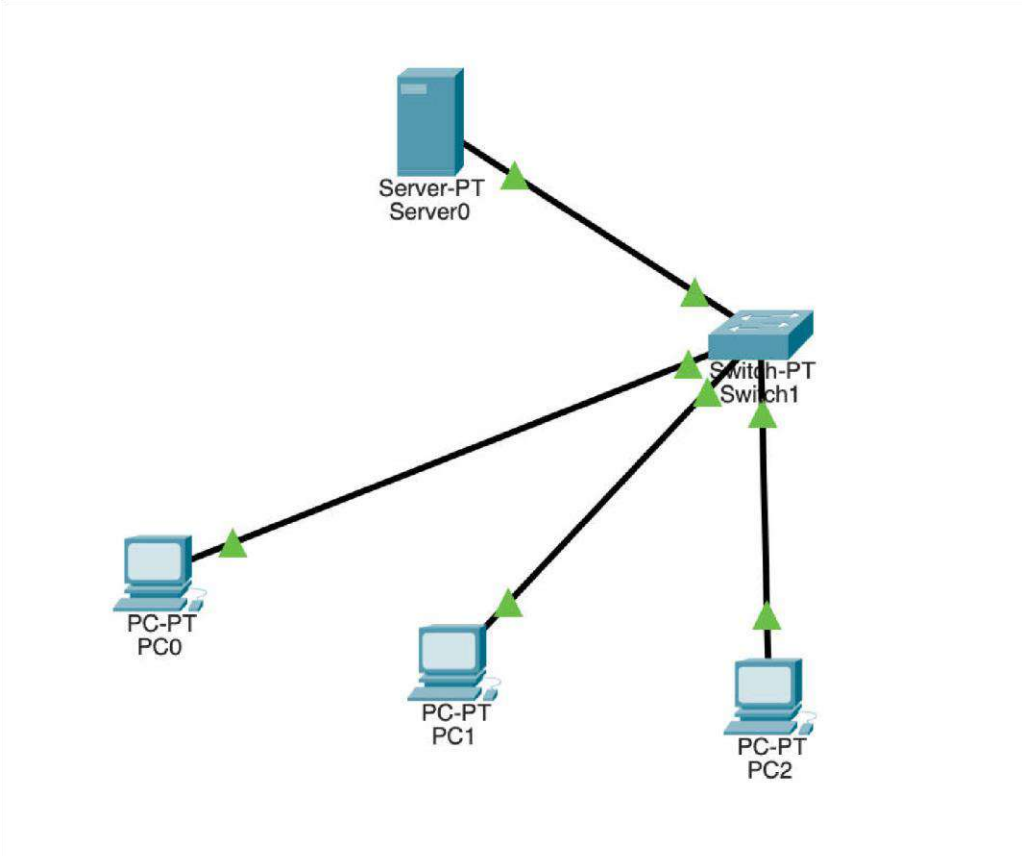
Observation:

OSPF protocol enables the routers to distribute IP routing information throughout. The network using the shortest path. A loopback address is needed for the routers to keep the ospf process up all the time. A virtual link is needed for some routers that doesn't know the others area's in the topology.

Experiment No. 8

Title:
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

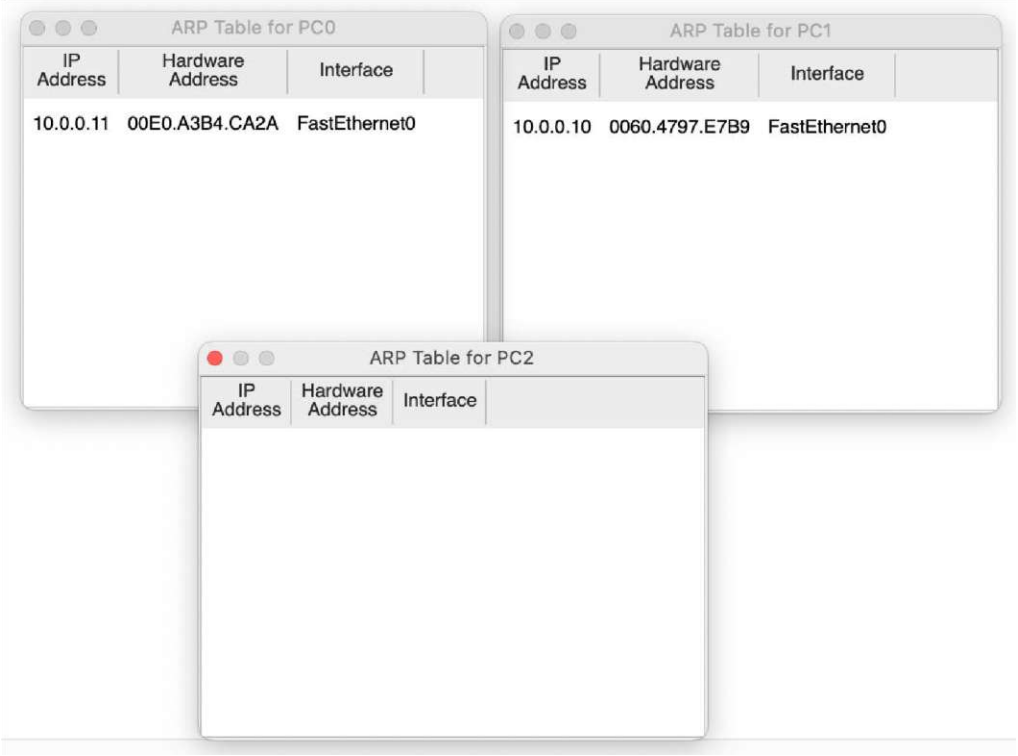
Topology:



Ping PC1 from PC0:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>arp
Cisco Packet Tracer PC ARP
Display ARP entries: arp -a
Clear ARP table: arp -d

C:\>arp -a
No ARP Entries Found
C:\>ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data:

Reply from 10.0.0.11: bytes=32 time=1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>arp -a
  Internet Address      Physical Address      Type
  10.0.0.11             00e0.a3b4.ca2a        dynamic
```

ARP table for all PCs:

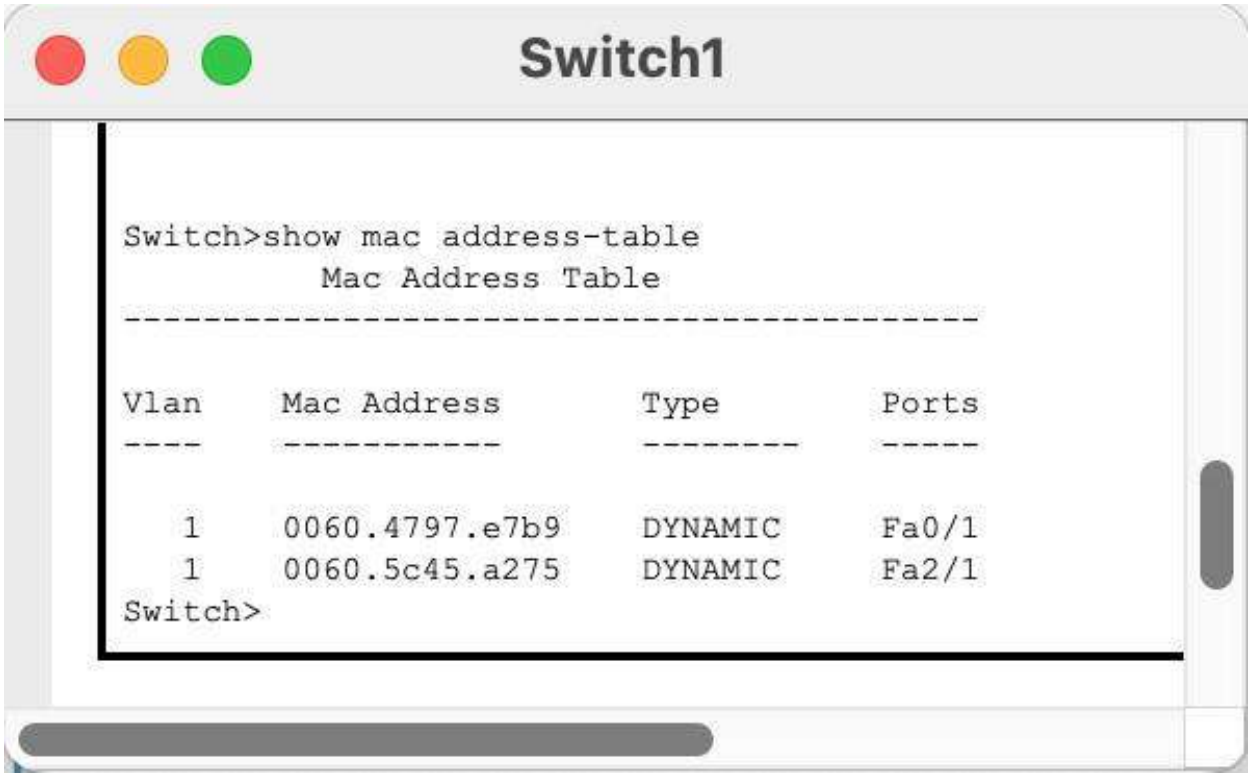


Mac Address Table:

```
Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0060.4797.e7b9    DYNAMIC     Fa0/1
   1    00e0.a3b4.ca2a    DYNAMIC     Fa1/1
Switch>
```

Ping PC2 form PC0:

```
C:\>ping 10.0.0.12

Pinging 10.0.0.12 with 32 bytes of data:

Reply from 10.0.0.12: bytes=32 time<1ms TTL=128
Reply from 10.0.0.12: bytes=32 time<1ms TTL=128
Reply from 10.0.0.12: bytes=32 time<1ms TTL=128
Reply from 10.0.0.12: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>arp -a
  Internet Address      Physical Address      Type
  10.0.0.11             00e0.a3b4.ca2a        dynamic
  10.0.0.12             0060.5c45.a275        dynamic

C:\>
```

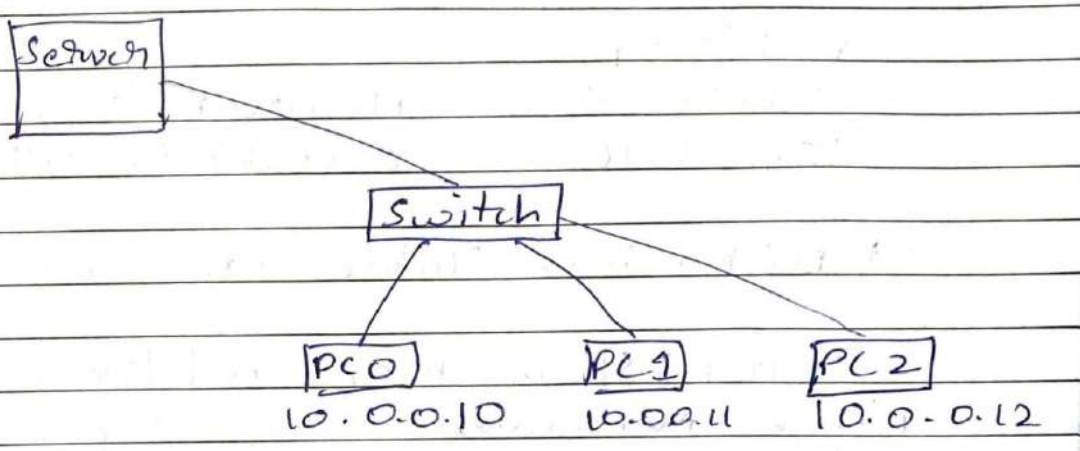


URBAN EDGE

(8) Title:
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).

Aim:
Understand the concept and operation of ARP.

Topology:



Procedure:
1. Create topology as shown above.
2. Configure IP address for each end devices.
3. Use the "inspect tool" to see ARP table and mac-address table (or) use the following commands
PC> arp -a [ARP table]
PC> show mac address-table [mac address table]
4. Go to simulation mode to send packets between end devices. Use "capture" button to go step by step & observe changes in ARP table, as and when new communication starts.

URBAN EDGE

Result:
Send packet from PC0 to PC1

Before sending packet -
PC0 > arp -a
NO ARP entries found
After sending packet -
PC0 > arp -a
Internet address Physical Address Type
10.0.0.11 00e0.a3b4.ca2a dynamic

PC1 > arp -a
Internet address Physical Address Type
10.0.0.10 0060.4797.c7b9 dynamic

MAC Address Table in switch

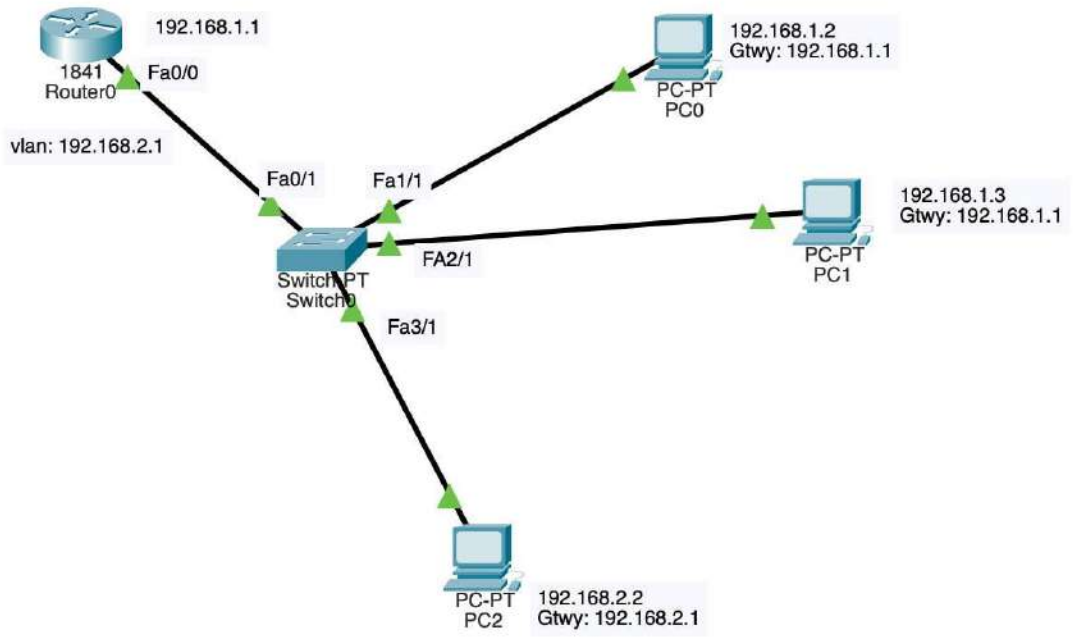
Switch>show mac address-table
Vlan Mac Address Type Ports
1 0060.4797.e7b9 Dynamic Fa0/1
1 00e0.e3b4.ca2a Dynamic Fa1/1

Observation:
Initially the ARP table is empty, after the host encapsulates a packet into a frame, it is reflected in the MAC address table to determine the mapping of IP address to MAC address

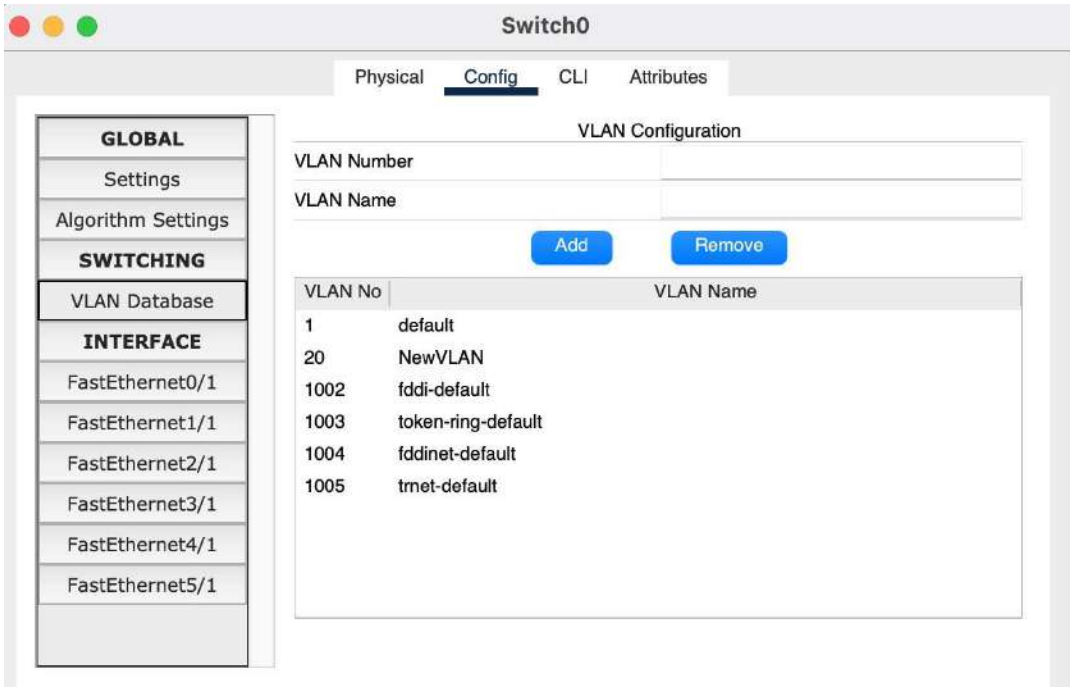
Experiment No. 9

Title:
To construct a VLAN and make the PC's communicate among a VLAN

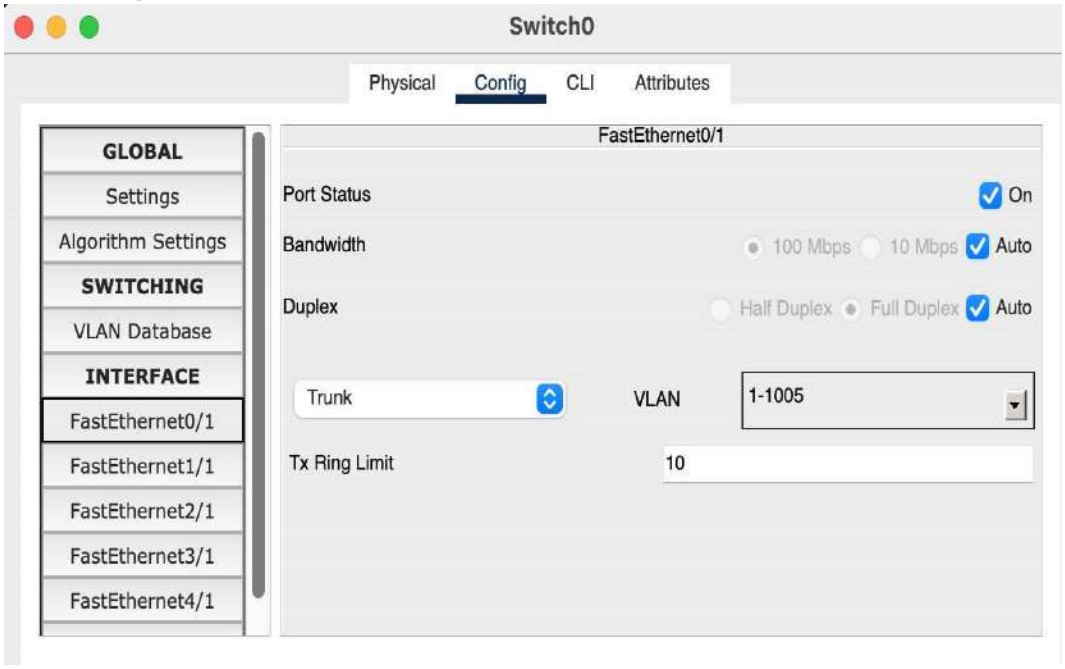
Topology:



Create VLAN:



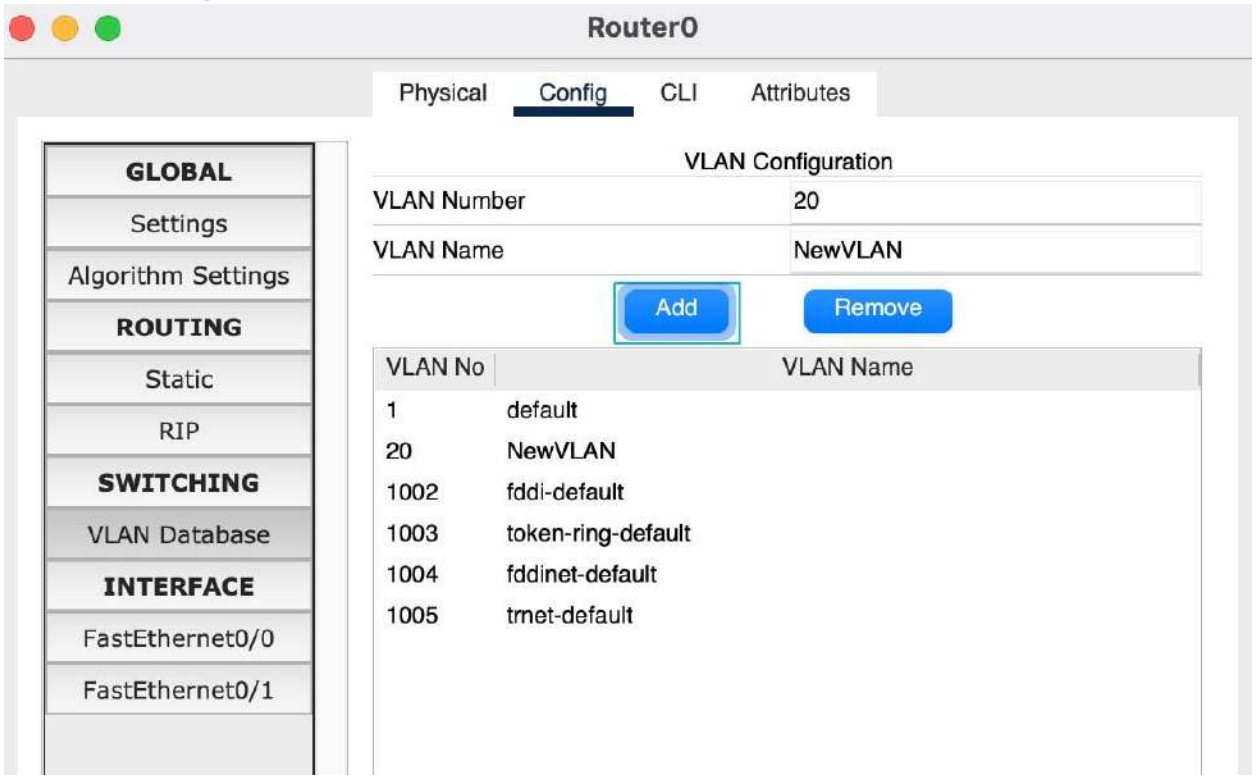
Trunking:

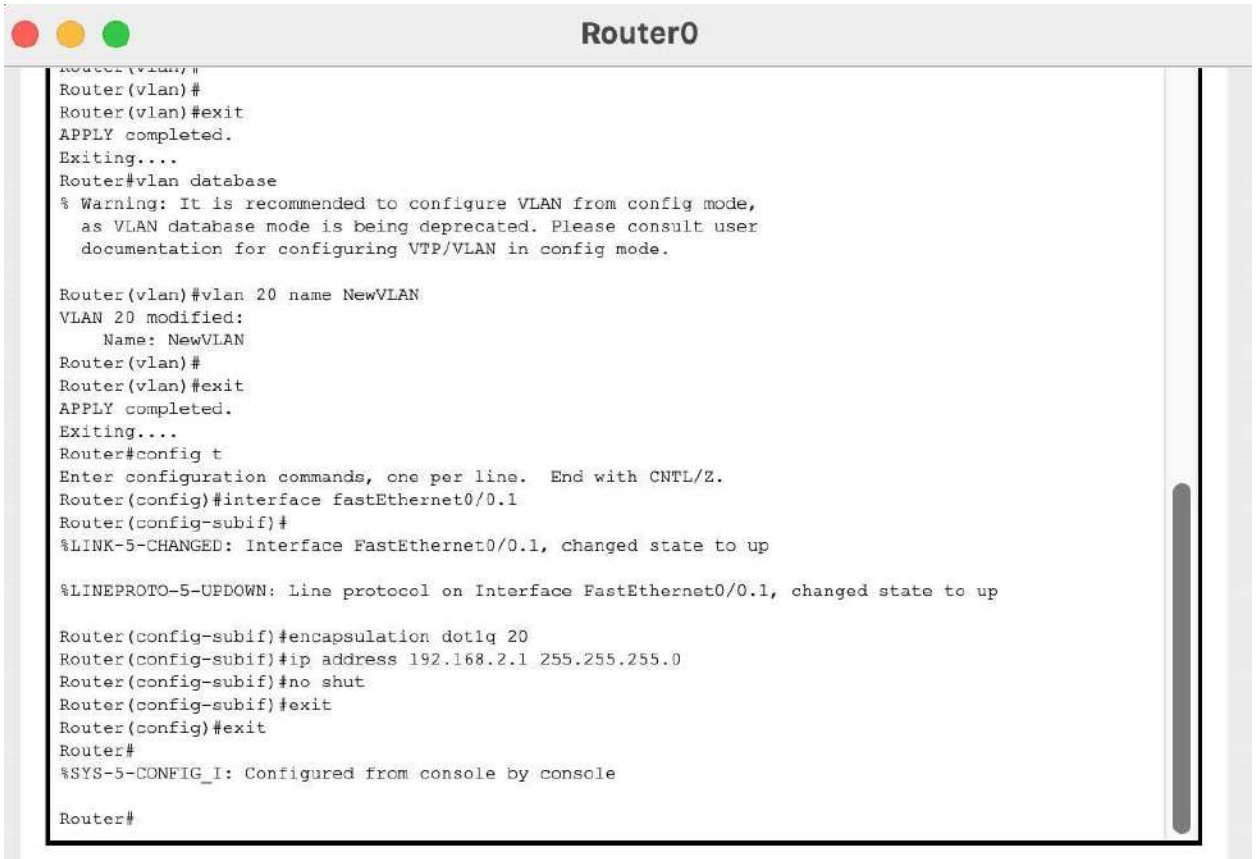


Add end devices to VLAN:

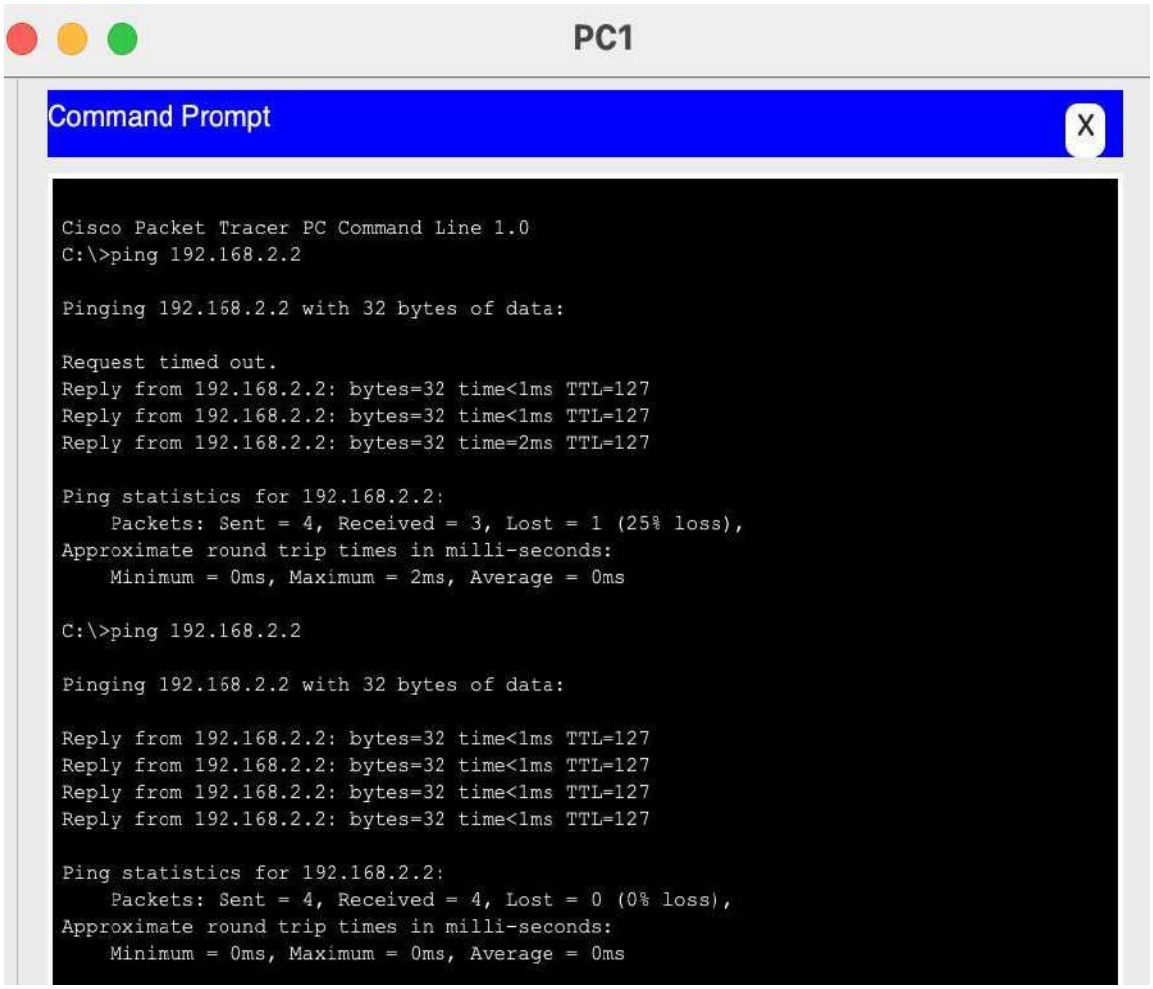

Switch0
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet1/1
FastEthernet2/1
FastEthernet3/1
FastEthernet4/1
FastEthernet3/1
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
Access
VLAN
20
Tx Ring Limit
10
1:default
20:NewVLAN
1002:fddi-default

Router config:

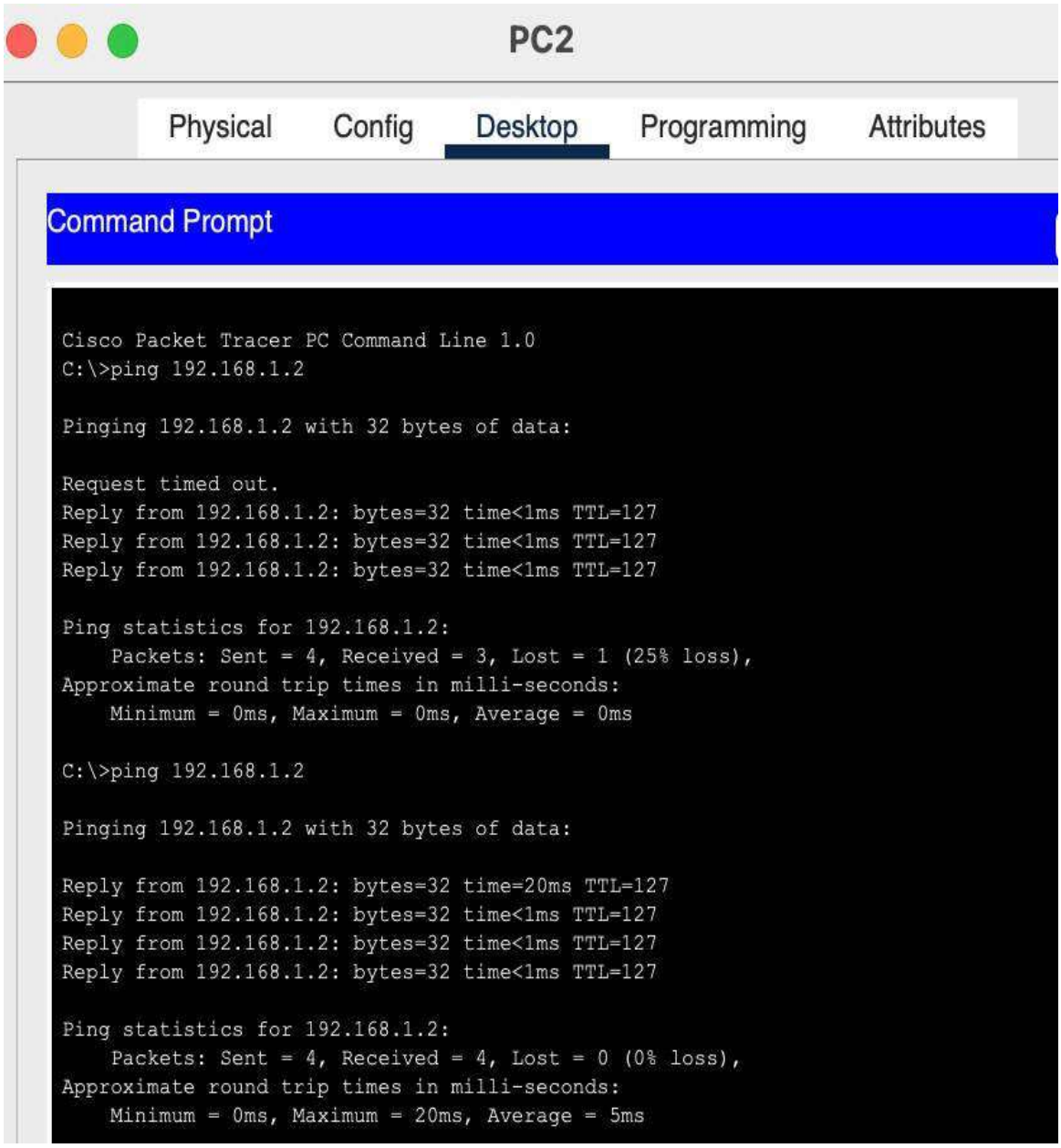




Pinging PC2(in VLAN) from PC1:



Pinging PC0 from PC2(in VLAN):

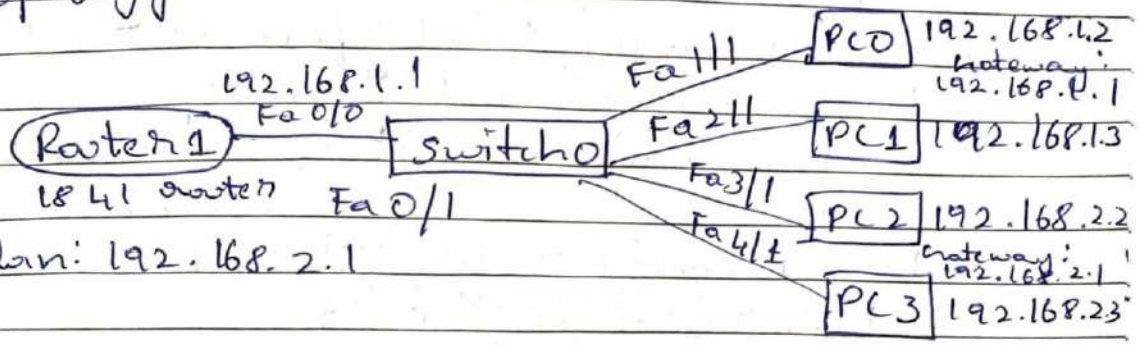

PC2
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=20ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 20ms, Average = 5ms

URBAN EDGE

(9) Title: To construct a VLAN and make the PC's communicate among VLAN.

Aim: Understand how to construct a VLAN.

Topology:



Procedure:

(1) Create a topology as shown above. use Router 1841.

(2) Click on switch 0. Under config tab chose VLAN database.

(3) Create a VLAN by typing VLAN no. and VLAN name.
Then click on add
VLAN number: 20
VLAN name: New VLAN

(4) Select the interface of the switch which connects it to Router, here Fa 0/1 and make it Trunk.

(5) In the interfaces of the switch connecting to PC2 and PC3 [end devices of New VLAN] select 20: New VLAN under VLAN. This makes the switch understand PC2 and PC3 are in New VLAN.

(6) Click on Router. Under config tab select VLAN Database. Add the newly created VLAN by entering its number and name.

(7) open CLI in the Router. Enter the following commands.

```
Router(vlan)#exit
APPLY completed
Exiting....
Router#config t
Router(config)#interface fast Ethernet 0/0.1
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.2.1
                              255.255.255.0
Router(config-subif)#exit
Router(config)#exit
```

(8) Ping the end devices using end device in VLAN and Physical LAN to check connection.

Result:

Ping PC2 (VLAN) from PC1

```
PC1> Ping 192.168.2.2
Reply from 192.168.2.2, bytes=32, time<1ms,
                                TTL=127
Reply from 192.168.2.2, bytes=32 time<1ms
                                TTL=127
Reply from 192.168.2.2, bytes=32, time<1ms
                                TTL=127
Reply from 192.168.2.2, bytes=32, time<1ms
                                TTL=127
Packets: Sent=4, Received=4, Lost=0
                                (0% loss)
Round trip time in ms
        Min=0 Max=0 Avg=0
```

URBAN EDGE

Ping PC0 from PC2 (VLAN)

PC2> Ping 192.168.1.2

Reply from 192.168.1.2 bytes=32 time=20ms TTL=127

Reply from 192.168.1.2 bytes=32 time<1ms TTL=127

Reply from 192.168.1.2 bytes=32 time<1ms TTL=127

Reply from 192.168.1.2 bytes=32 time<1ms TTL=127

Packet: Sent=4, Received=4, Loss=0(0% Loss)

Approx Round Trip Time

Min=0 Max=20 Avg=5

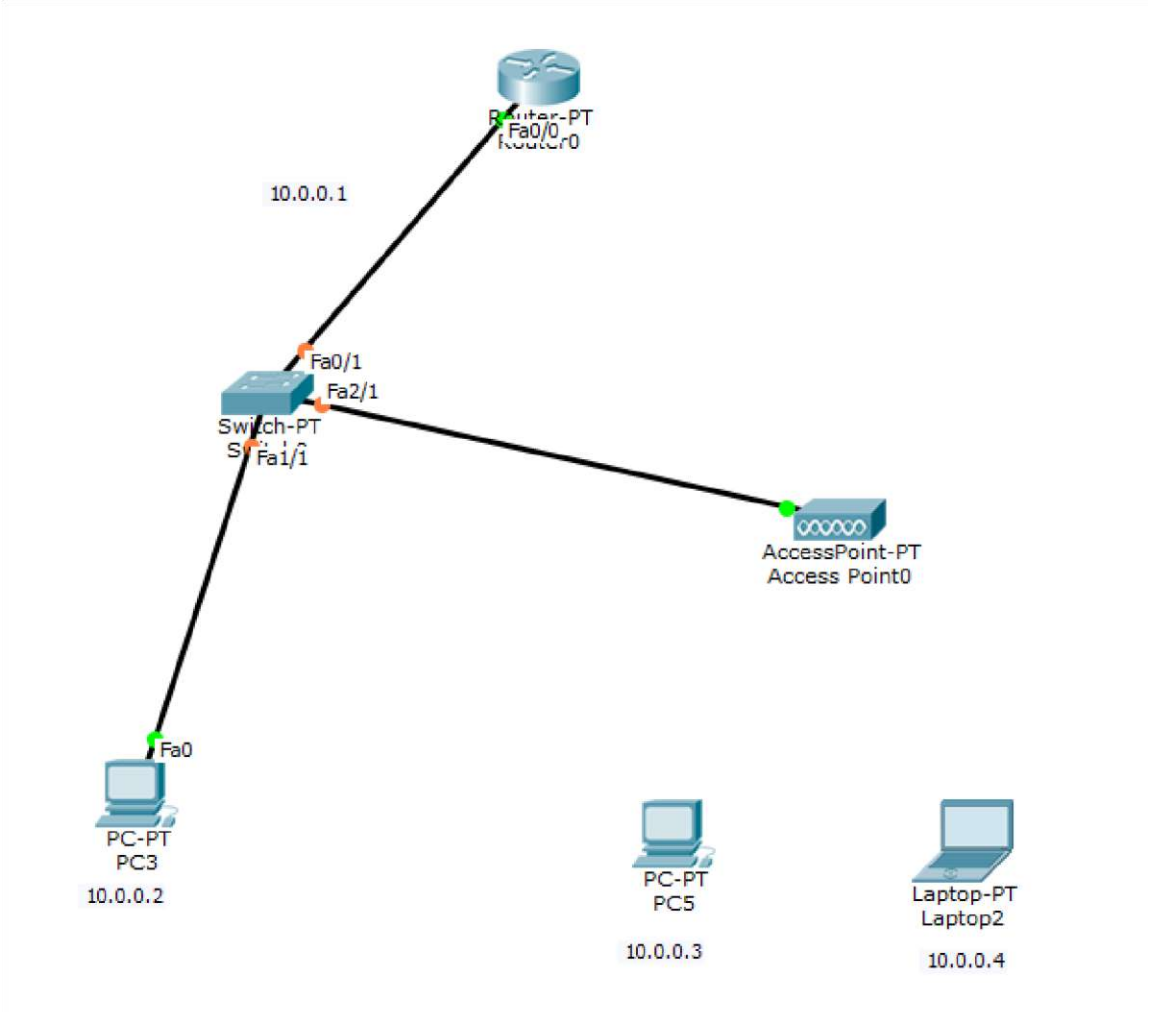
Observation:

A virtual LAN can be created by specifying the VLAN no and VLAN name in the switch and the same should be added to the router to make the router identify the newly created VLAN. The physical LAN and virtual LAN can communicate between each other even through both have different gateways.

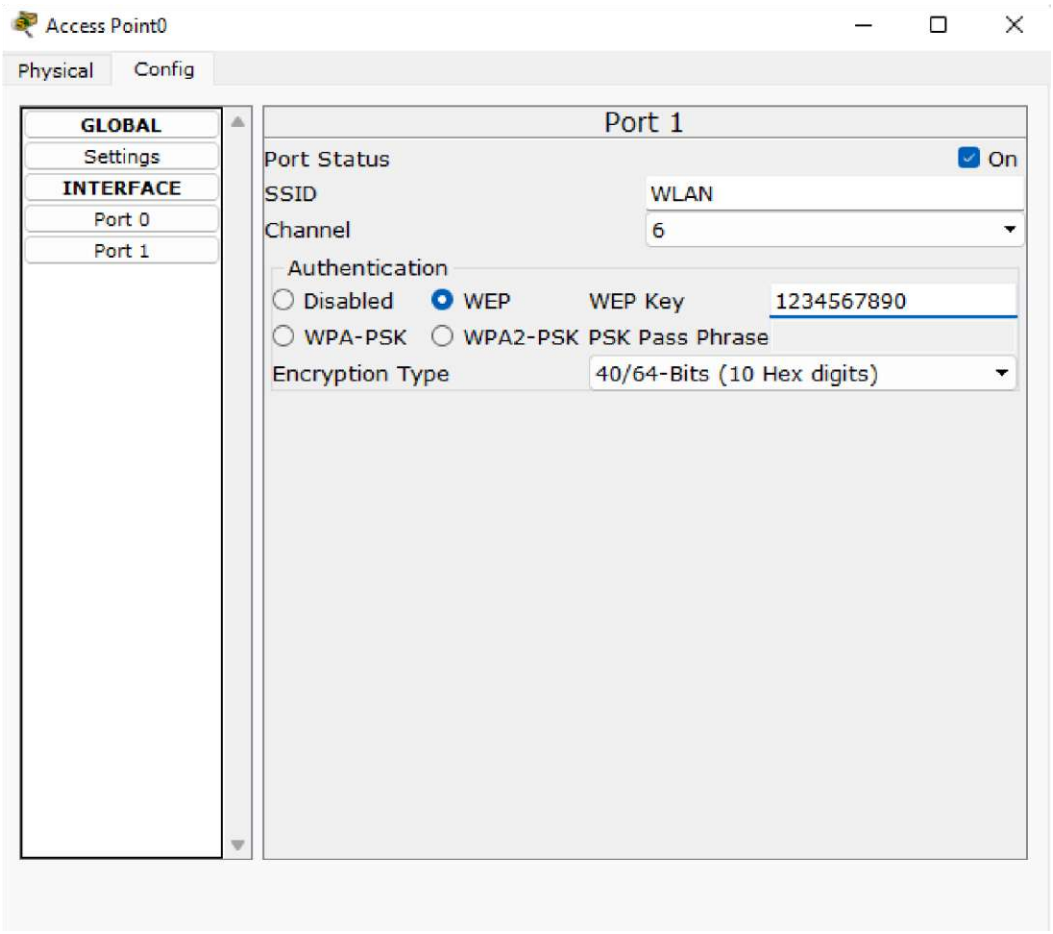
Experiment No. 10

Title:
To construct a WLAN and make the nodes communicate wirelessly

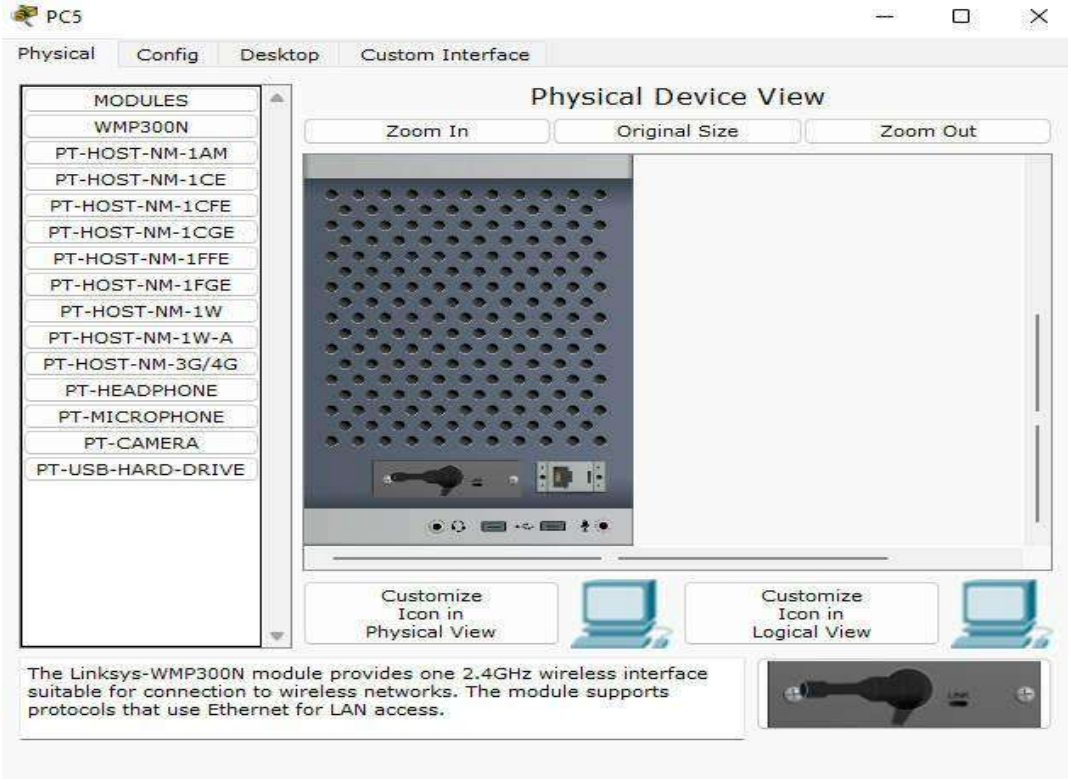
Topology:

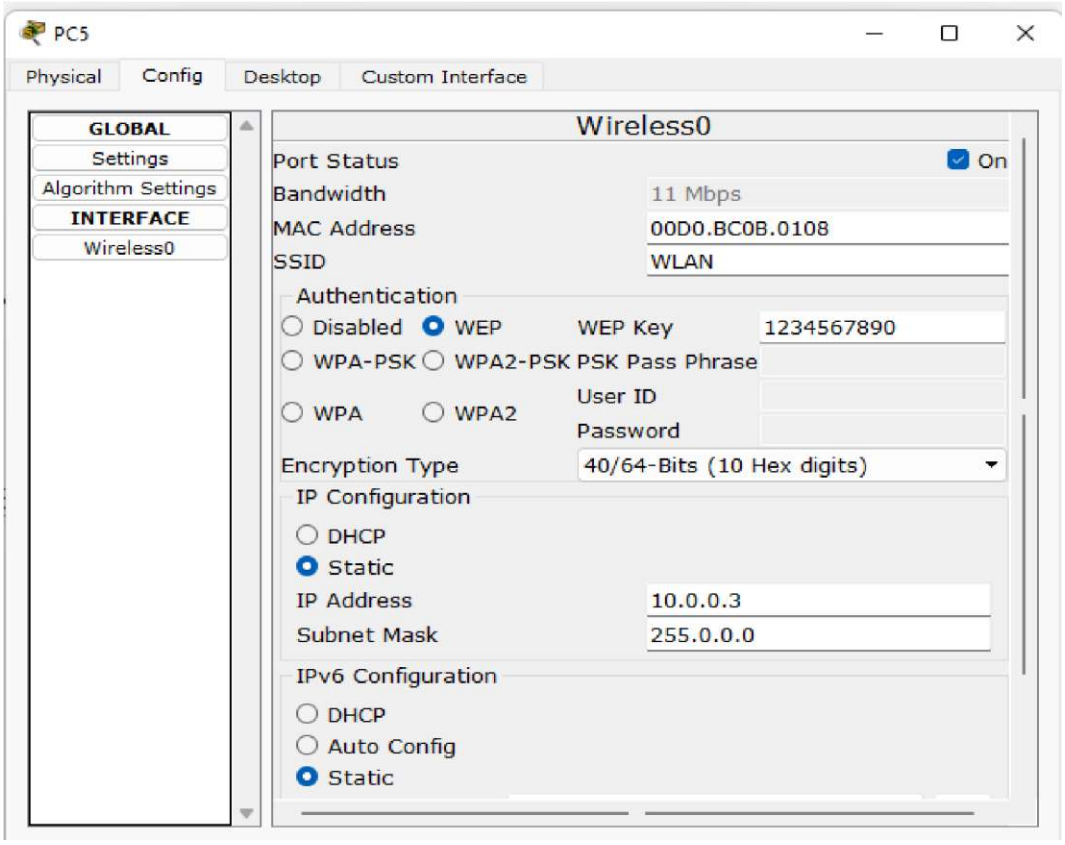


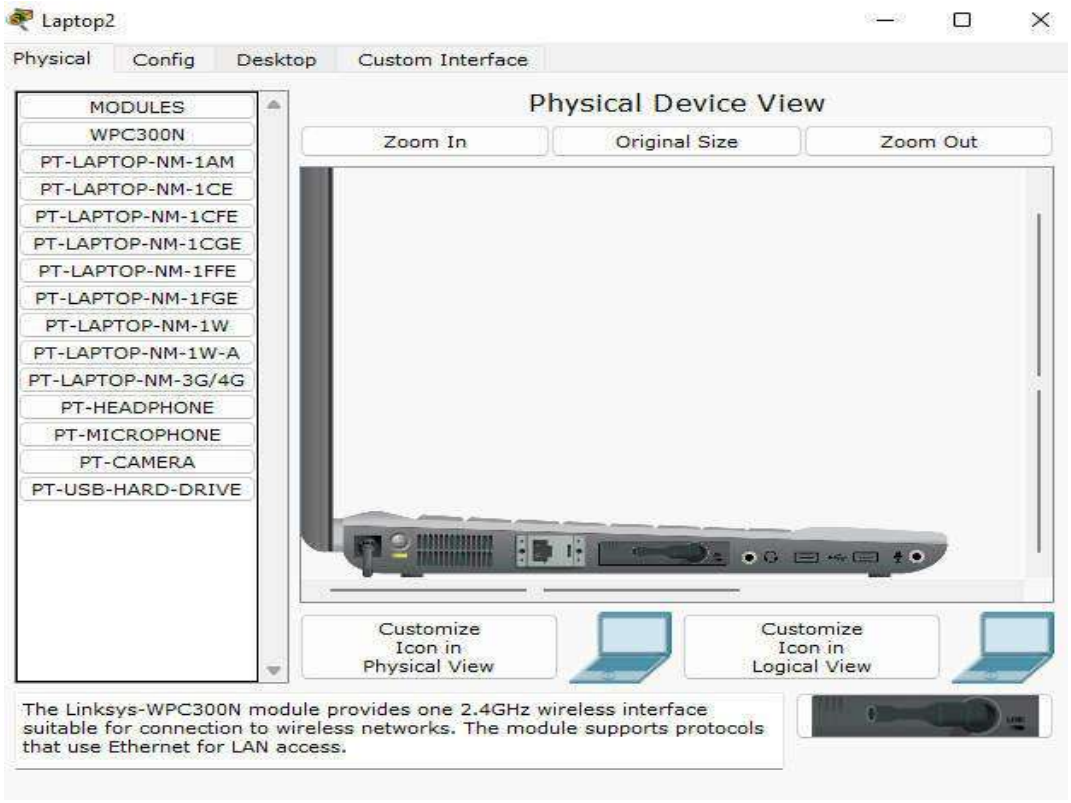
AccessPoint config:

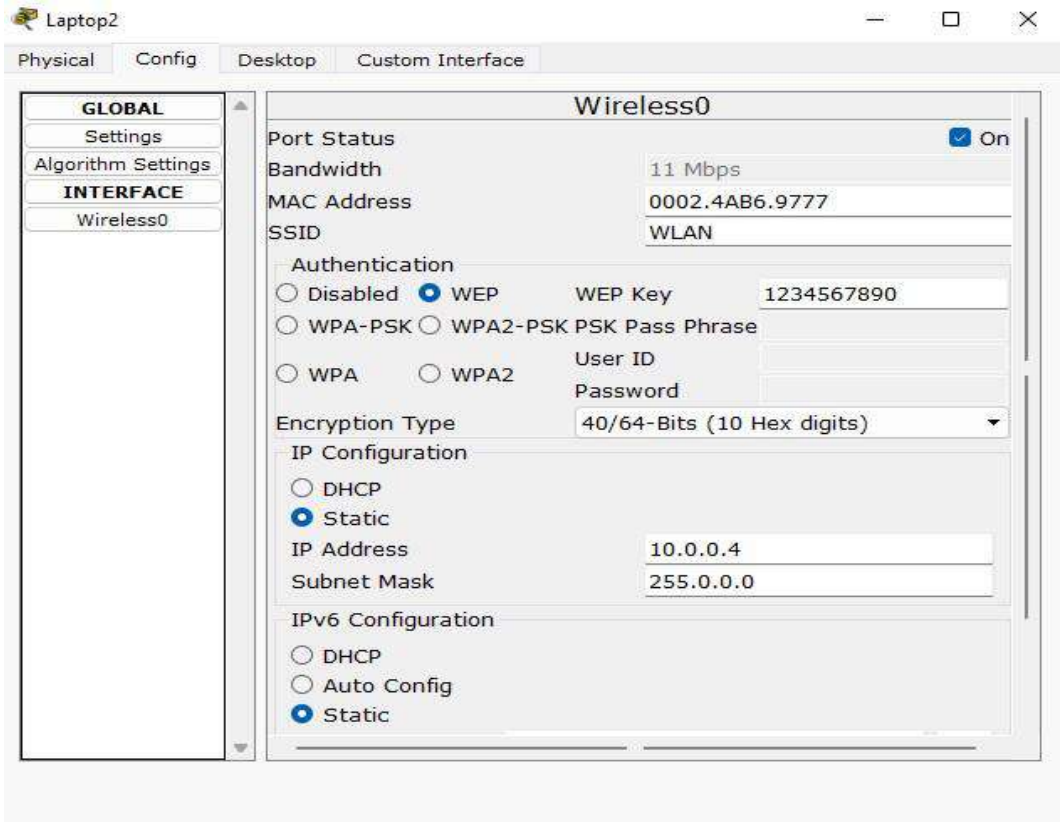


Configure wireless nodes:

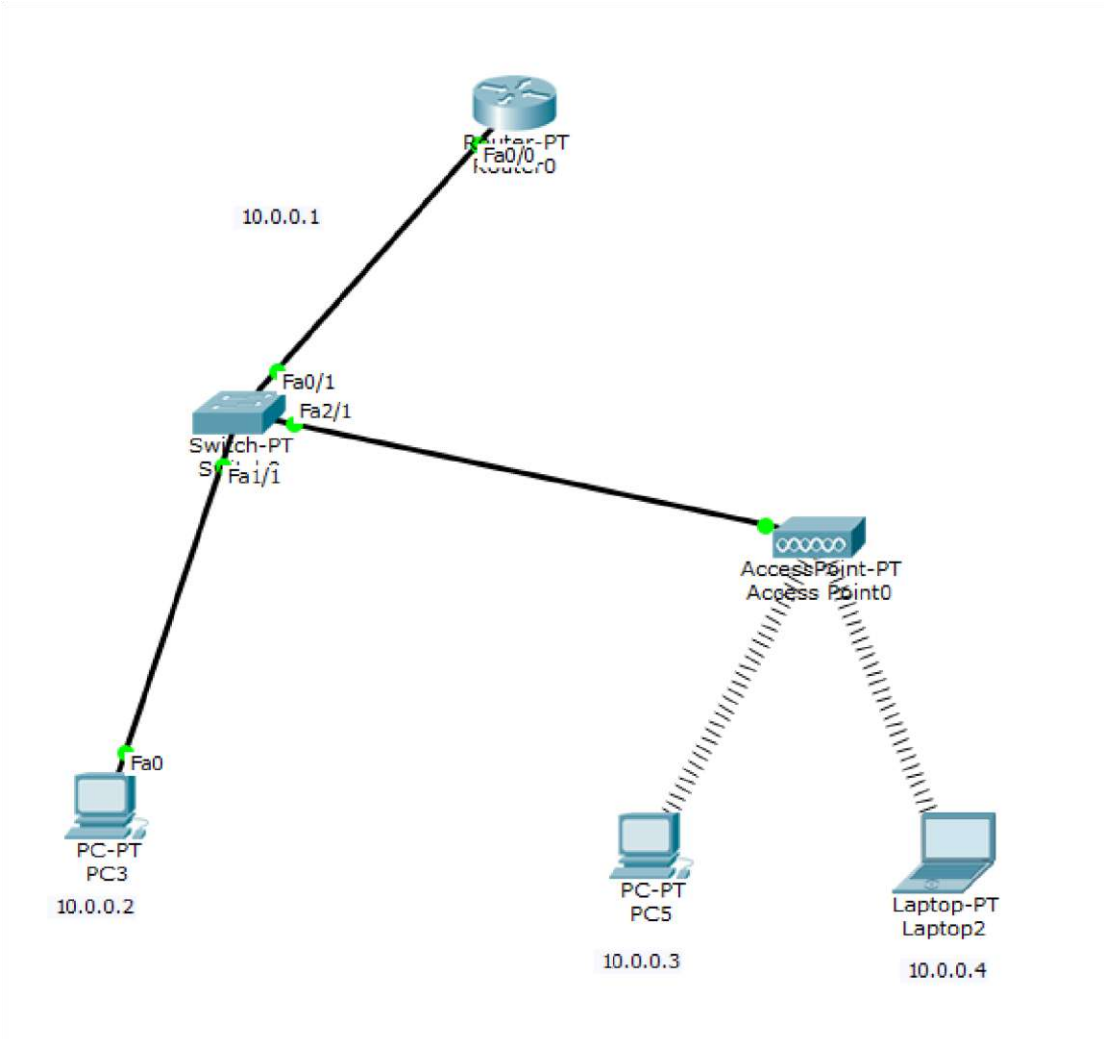



PC5
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
Wireless0
Port Status
On
Bandwidth
11 Mbps
MAC Address
00D0.BC0B.0108
SSID
WLAN
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
WPA
WPA2
User ID
Password
Encryption Type
40/64-Bits (10 Hex digits)
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static

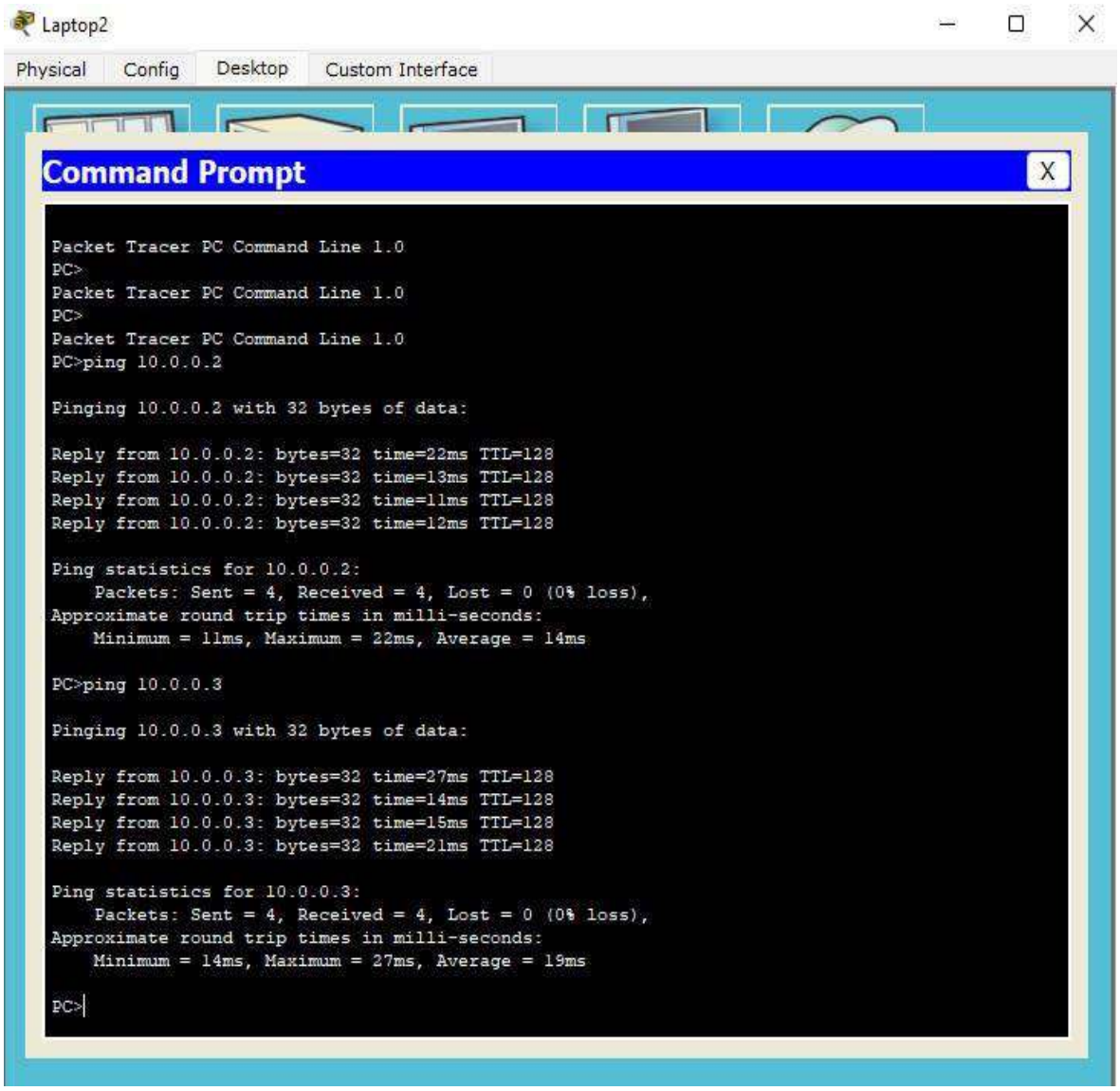


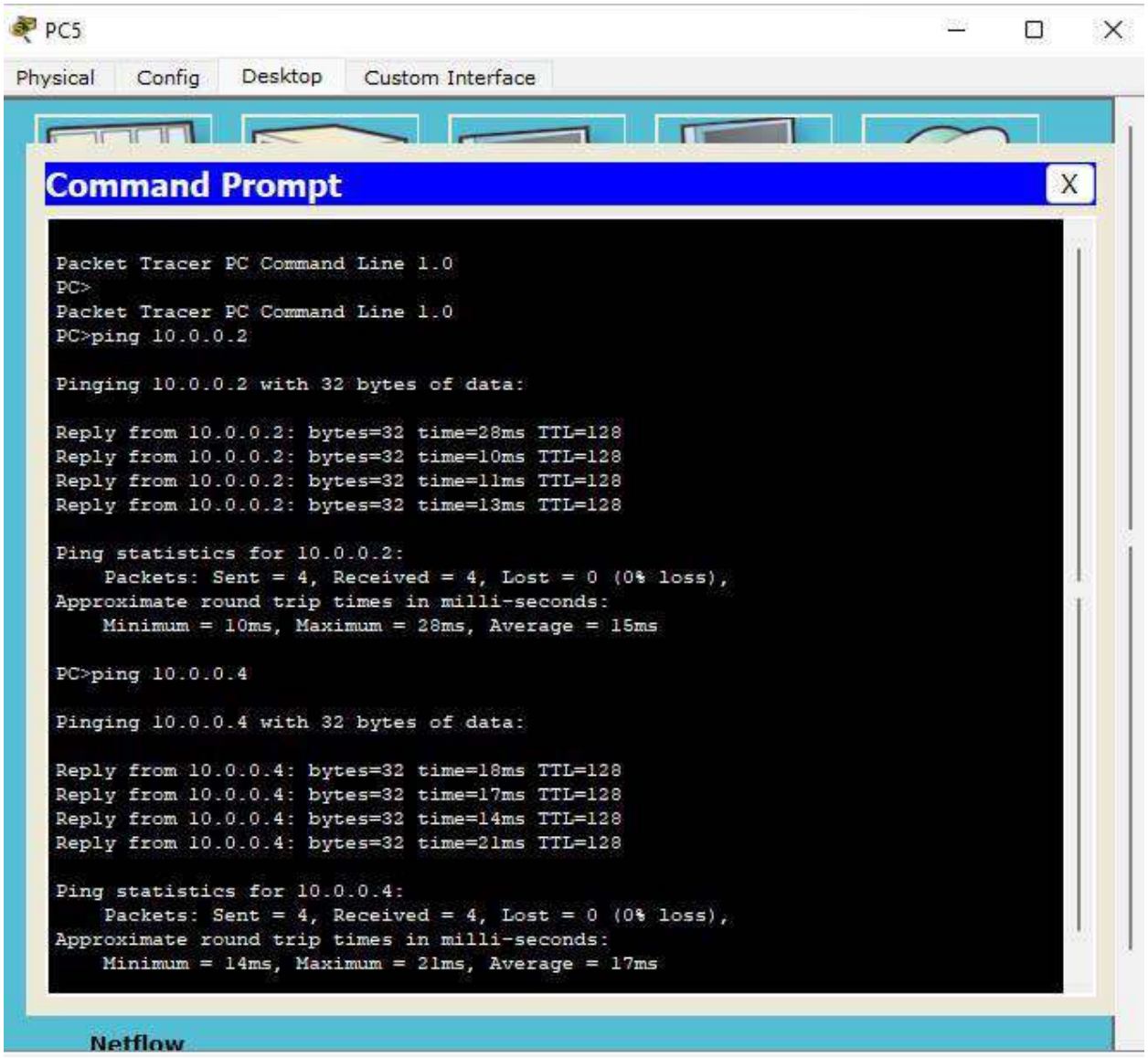


Final Topology:



Pinging end devices:


Laptop2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>
Packet Tracer PC Command Line 1.0
PC>
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=22ms TTL=128
Reply from 10.0.0.2: bytes=32 time=13ms TTL=128
Reply from 10.0.0.2: bytes=32 time=11ms TTL=128
Reply from 10.0.0.2: bytes=32 time=12ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 11ms, Maximum = 22ms, Average = 14ms
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=27ms TTL=128
Reply from 10.0.0.3: bytes=32 time=14ms TTL=128
Reply from 10.0.0.3: bytes=32 time=15ms TTL=128
Reply from 10.0.0.3: bytes=32 time=21ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 14ms, Maximum = 27ms, Average = 19ms
PC>

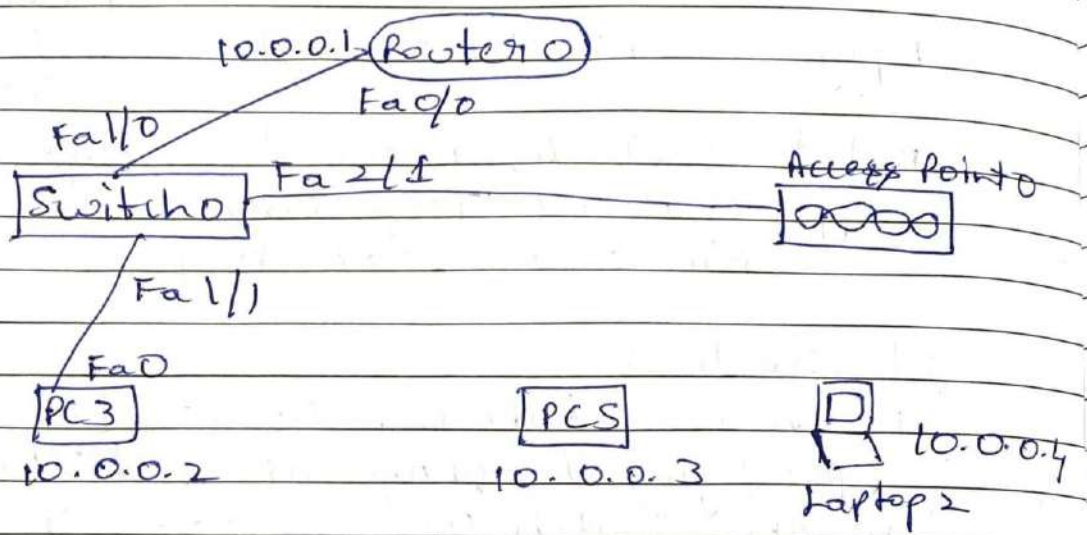

PC5
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=28ms TTL=128
Reply from 10.0.0.2: bytes=32 time=10ms TTL=128
Reply from 10.0.0.2: bytes=32 time=11ms TTL=128
Reply from 10.0.0.2: bytes=32 time=13ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 10ms, Maximum = 28ms, Average = 15ms
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=18ms TTL=128
Reply from 10.0.0.4: bytes=32 time=17ms TTL=128
Reply from 10.0.0.4: bytes=32 time=14ms TTL=128
Reply from 10.0.0.4: bytes=32 time=21ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 14ms, Maximum = 21ms, Average = 17ms
Netflow

URBAN EDGE

(10) Title: To construct a WLAN and make the nodes communicate wirelessly.

Aim: Understand how to construct a WLAN.

Topology:



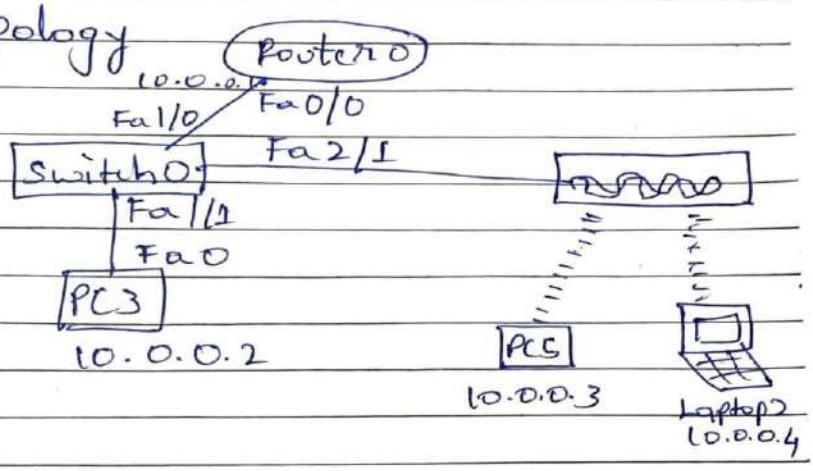
Procedure:

1. Create the topology as shown above.
2. Configure PC3 & Router 0 as normally done.
3. Configure Access Point 0, goto port 1 and give SSID name (any name), here WLAN
4. Select WEP and give any 10 digit hex key (1234567890 here). Configure PC5 & Laptop 2 with wireless standards.
5. Switch off the device. Drag existing PT-HOST-NM-1AM to component list. Drag WMP300N wireless interface into the empty ports and switch on the device.
6. In the Config tab, a new wireless interface will be added. Now configure

SSID, WEP, WEP key, IP address & gateway to the devices (PCs, Laptops)

Result :

Final Topology



Pinging End Devices :

Laptop 2 > ping 10.0.0.2

Reply from 10.0.0.2 : bytes = 32 time = 20ms TTL = 120

Reply from 10.0.0.2 : bytes = 32 time = 13ms TTL = 120

Reply from 10.0.0.2 : bytes = 32 time = 11ms TTL = 120

Reply from 10.0.0.2 : bytes = 32 time = 12ms TTL = 120

Packets : sent = 4, Recieved = 4, Loss = 0 (0% Loss)

Min = 11ms Max = 22ms, Avg = 14ms

observation :

By using the 'line vty 0 5" command virtual access to the router and the no. of users having this access can be set. The "enable secret" command is used to

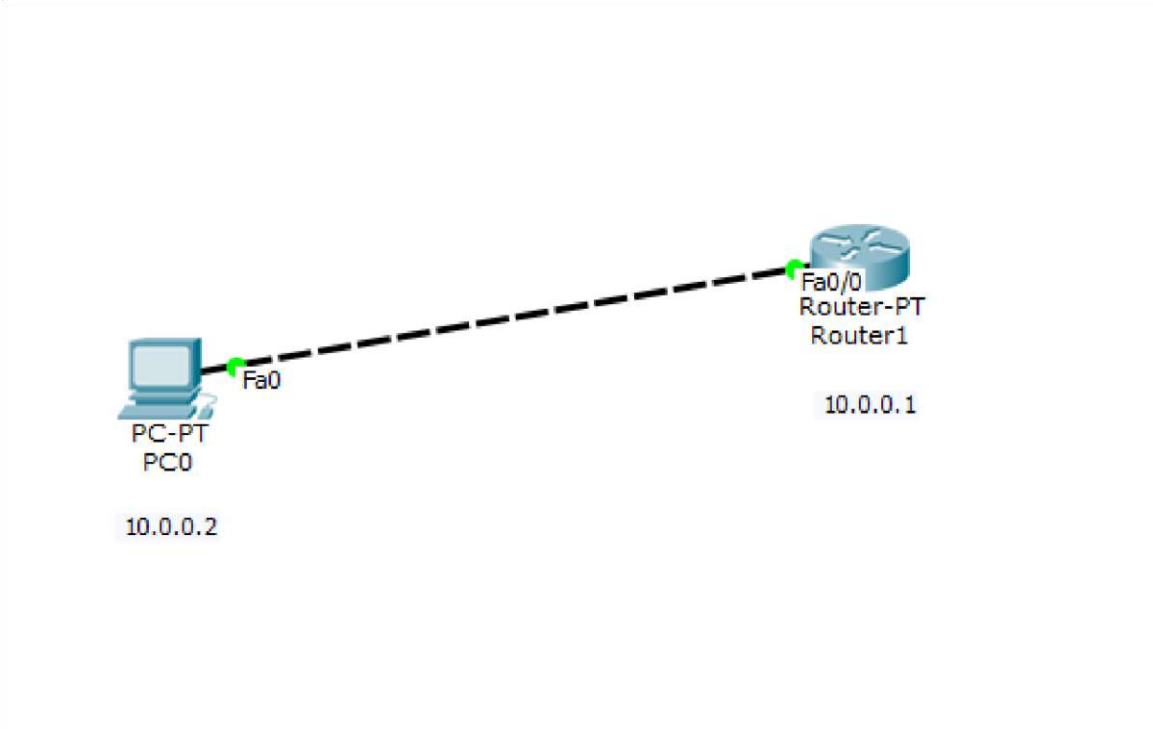
URBAN EDGE

set the password for enabling the Router and under r1 (config-lin) # "password" command is used to set login password.

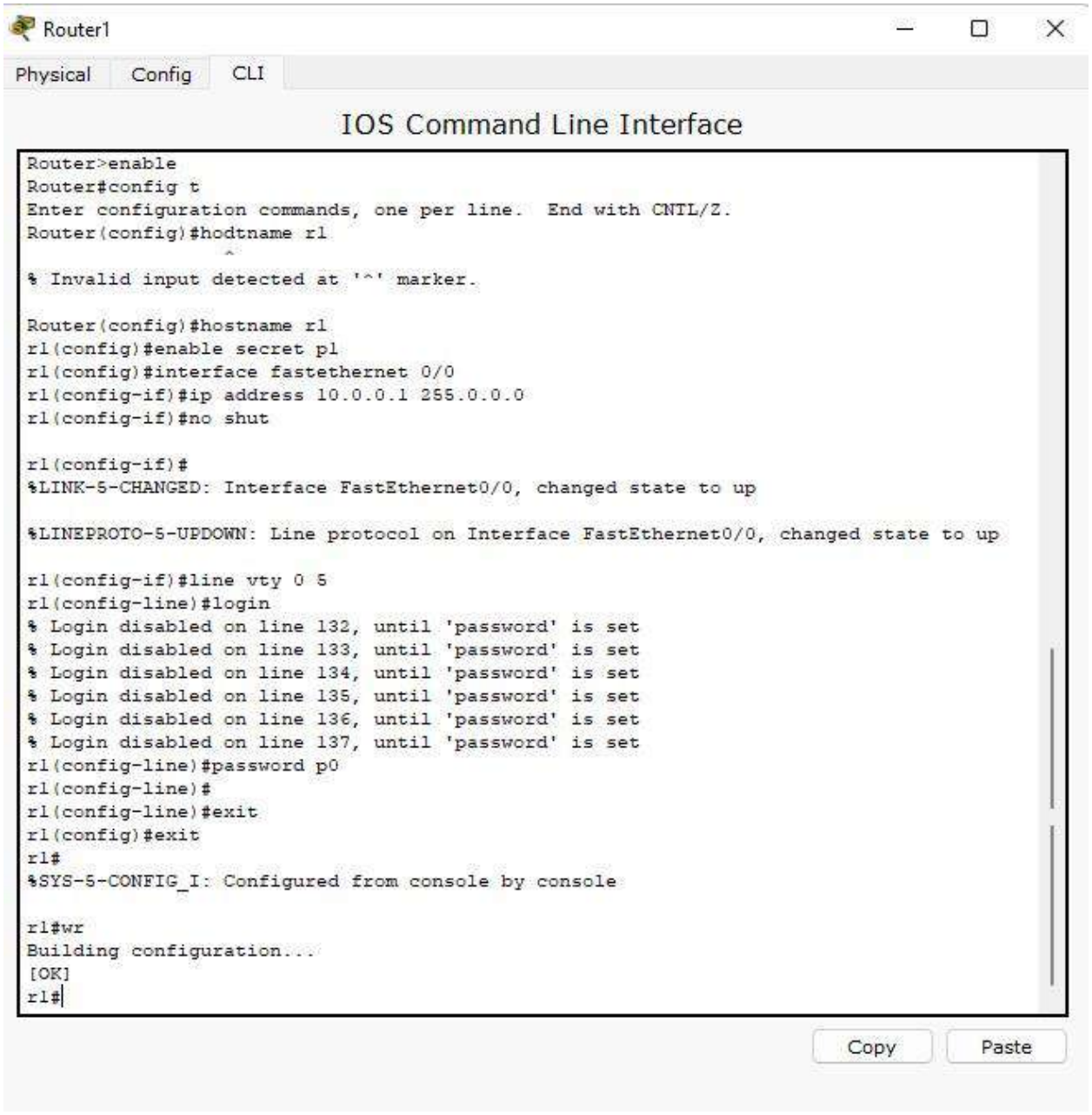
Experiment No. 11

Title:
To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.

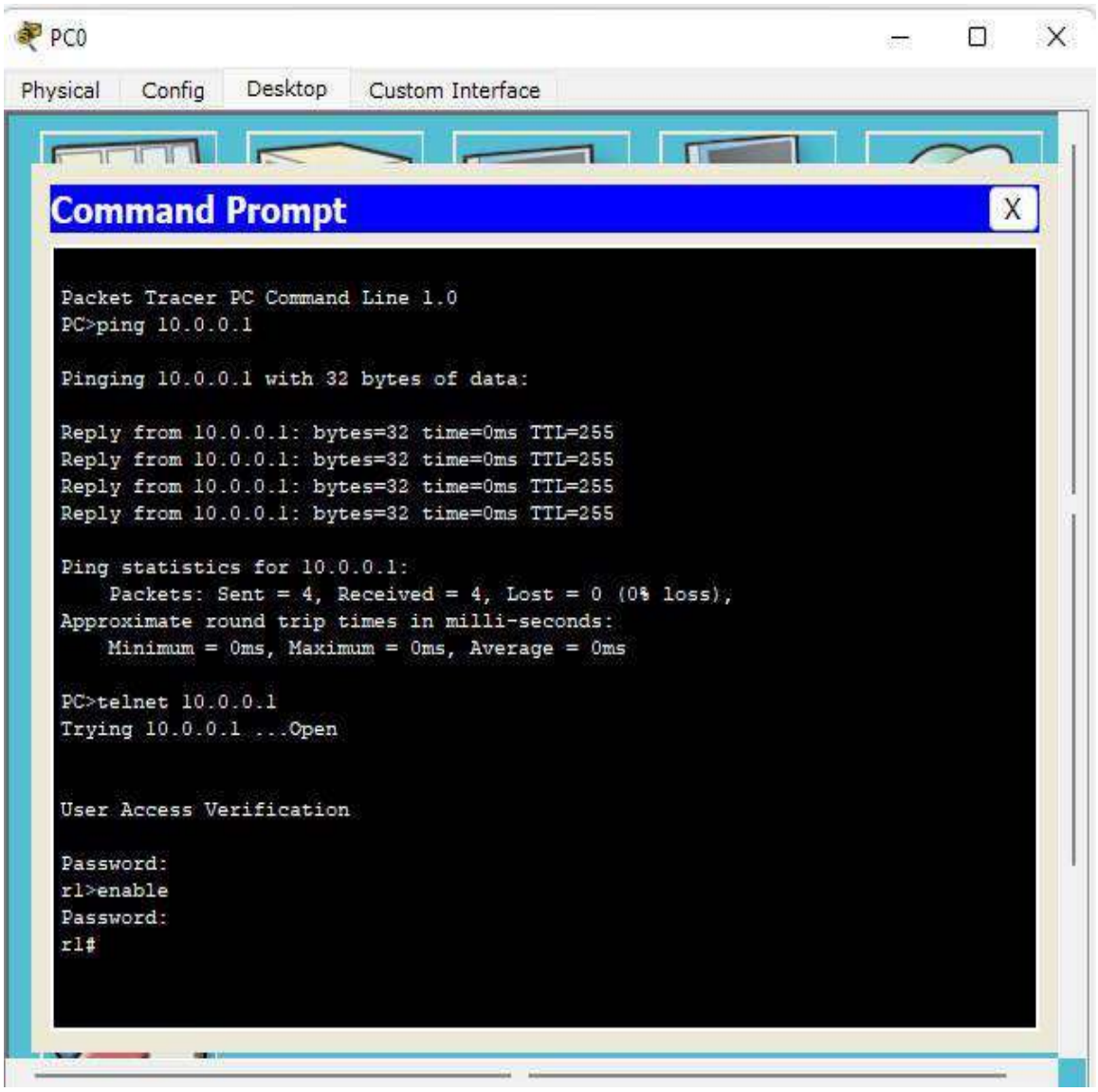
Topology:

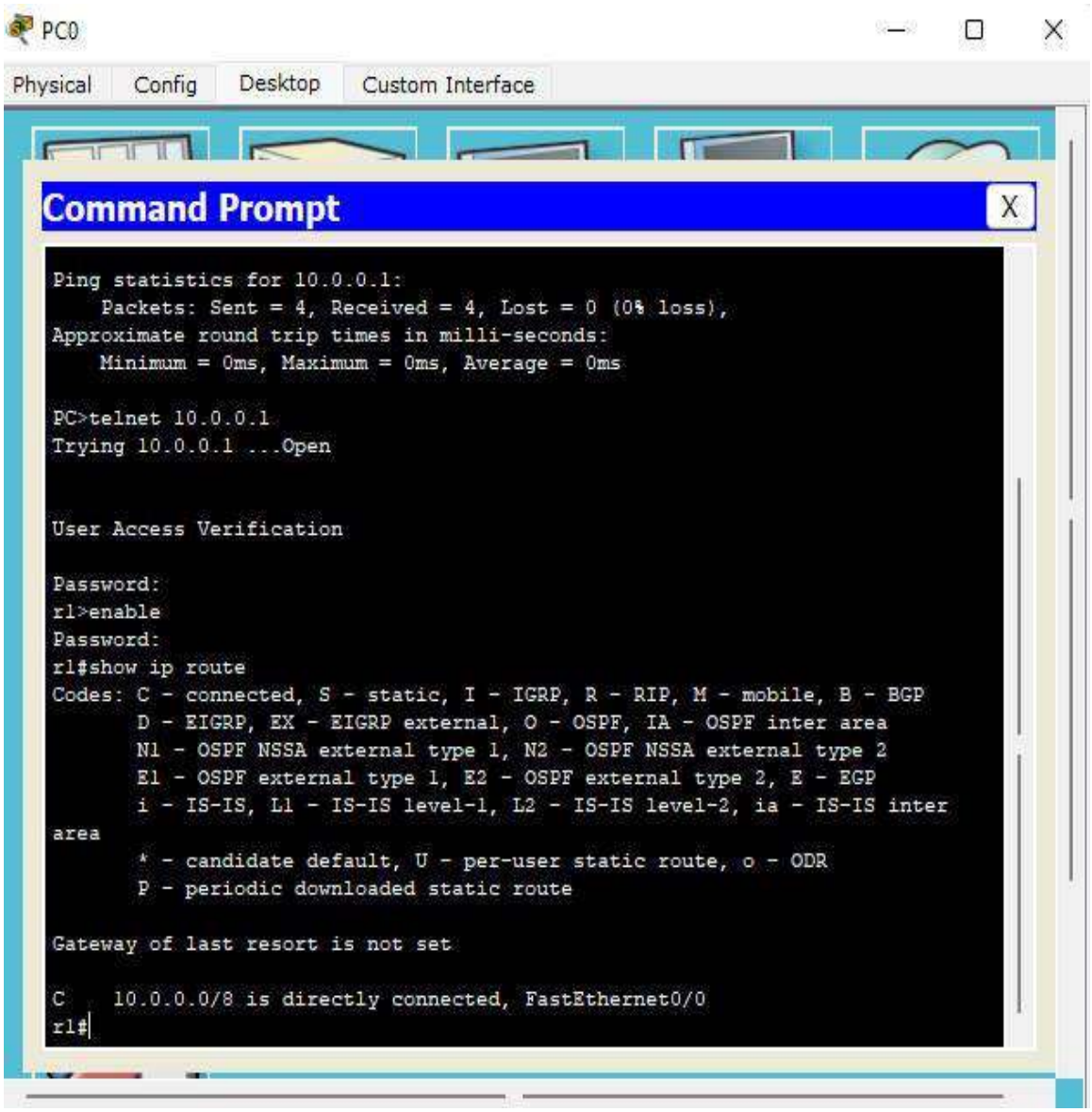


Router config:



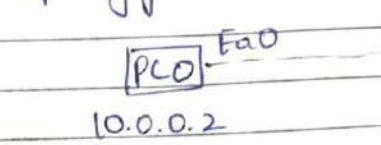
Pinging & accession Router CLI from PC:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
r1#


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
rl>enable
Password:
rl#show ip route
Codes: C - connected, S - static, I - IGRP, 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 not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
rl#

(11) Title/Aim: To understand the operations of TELNET by accessing the router in server room from a PC in IT office.

Topology:



Procedure:

1. Construct a topology as shown above.
2. Configure the PC0 with IP address 10.0.0.2.
3. In router 1 open CLI and enter the following commands.

```
Router # configt
Router (config) # hostname r1
r1(config) # enable secret p1
r1(config) # interface fastethernet 0/0
r1(config-if) # ip address 10.0.0.1 255.0.0.0
r1(config-if) # no shutdown.

r1(config-if) # line vty 0 5 [to allow access for 6 users]

r1(config-if) # login
r1(config-if) # password p0
r1(config-line) # exit
r1 (config) # exit
r1 # wr [to save changes in router]
```

Date __/__/__
Page _____

(4) Router 0's CLI can be opened from PC 0's command prompt using following commands..

```
PC > telnet 10.0.0.1
password:
    r1> enable
password:
    r1#
```

Result:

```
PC0> Telnet 10.0.0.1
    Trying 10.0.0.1 ... open
    User Access Verification
        password: p0
        r1> enable
        Password: p1
        r1# show ip route
            C 10.0.0.0/8 directly connected
                                Fa0/0
```

Observation:

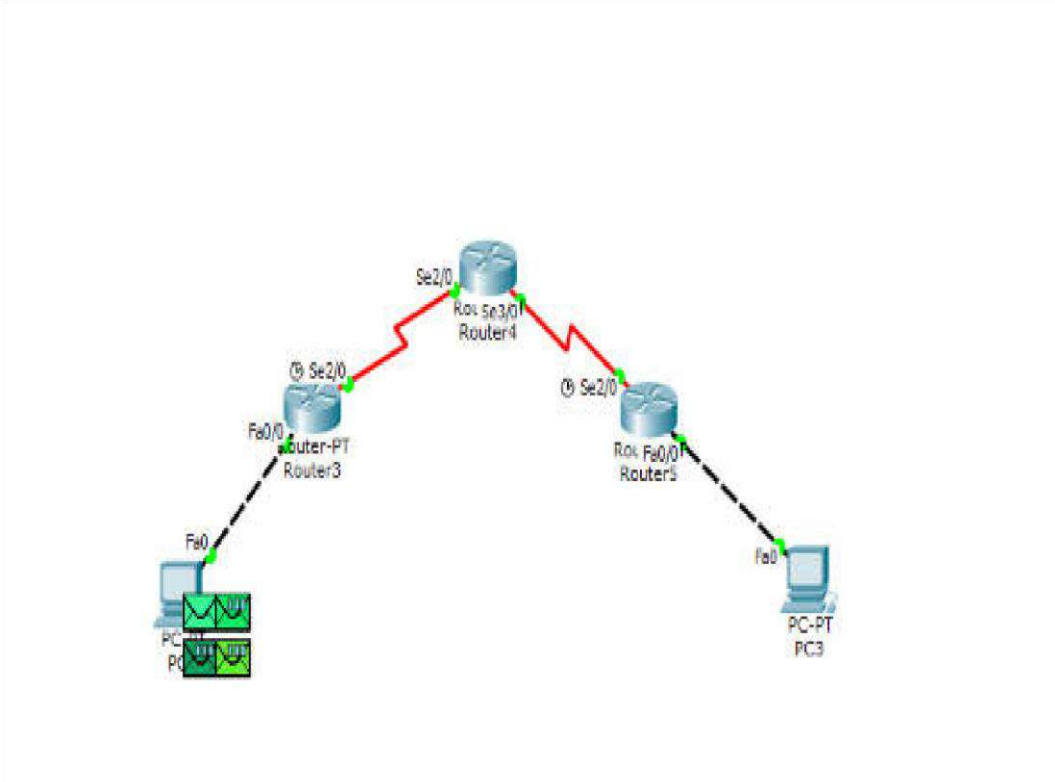
By using the "line vty 0 5" command virtual access to the router and the no. of users having this access can be set. The "enable secret" command is used to set the password for enabling the router and under r1(config-line) "password" command is used to set login password.

Experiment No. 12

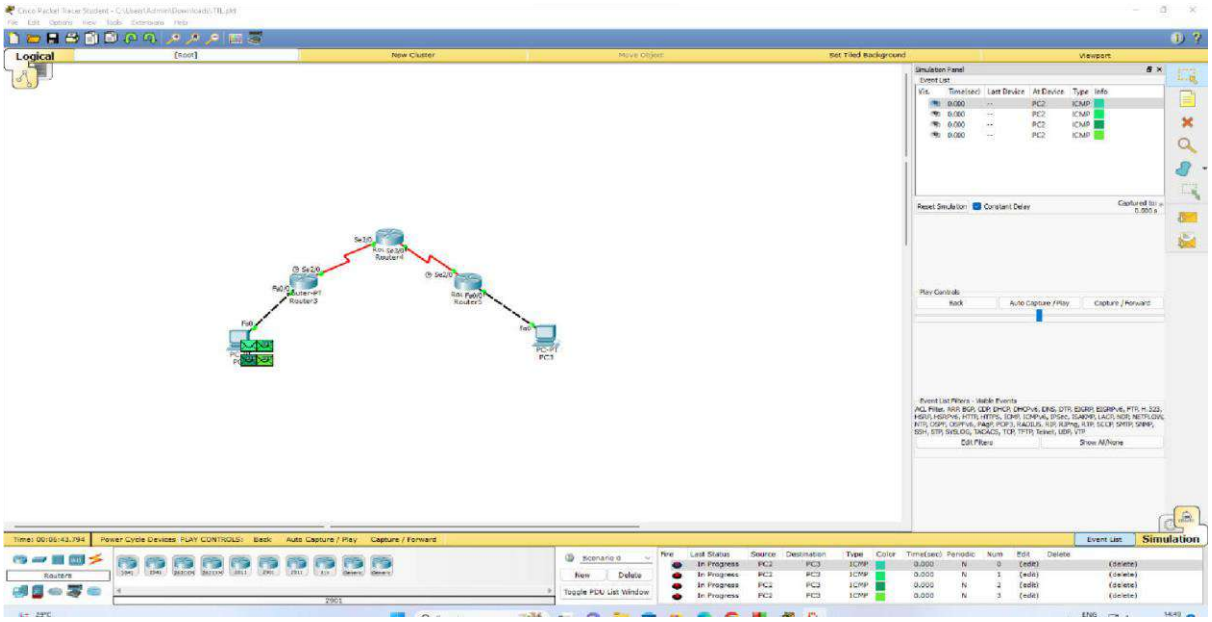
Title:

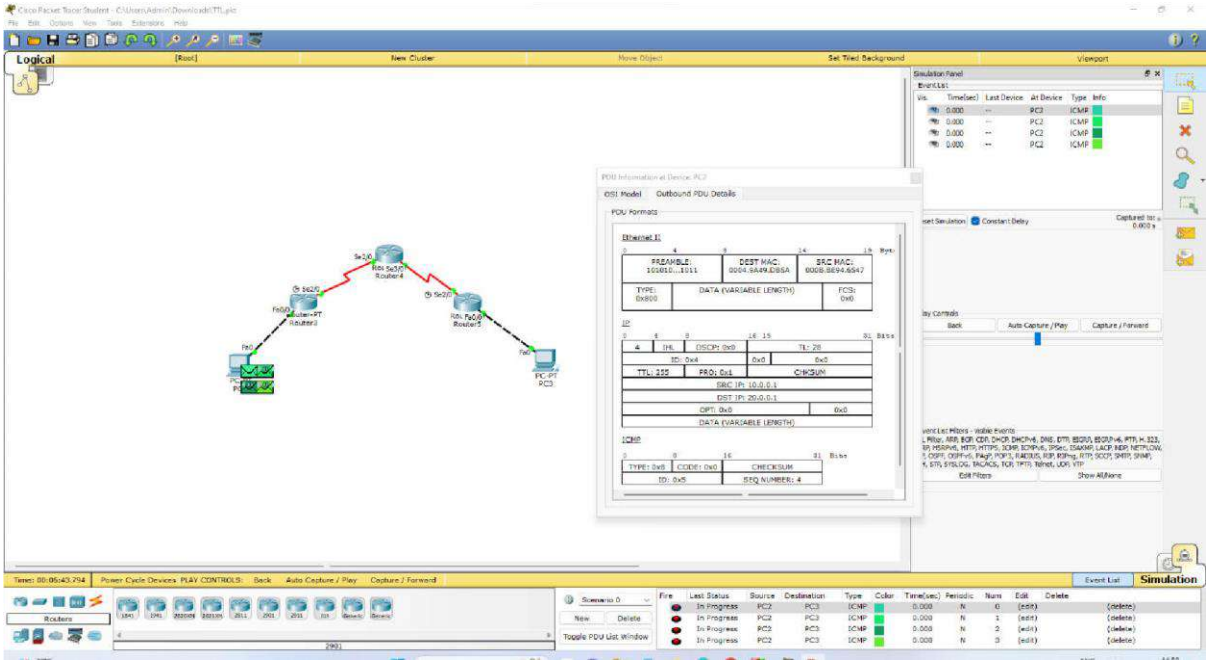
Demonstrate the TTL/ Life of a Packet

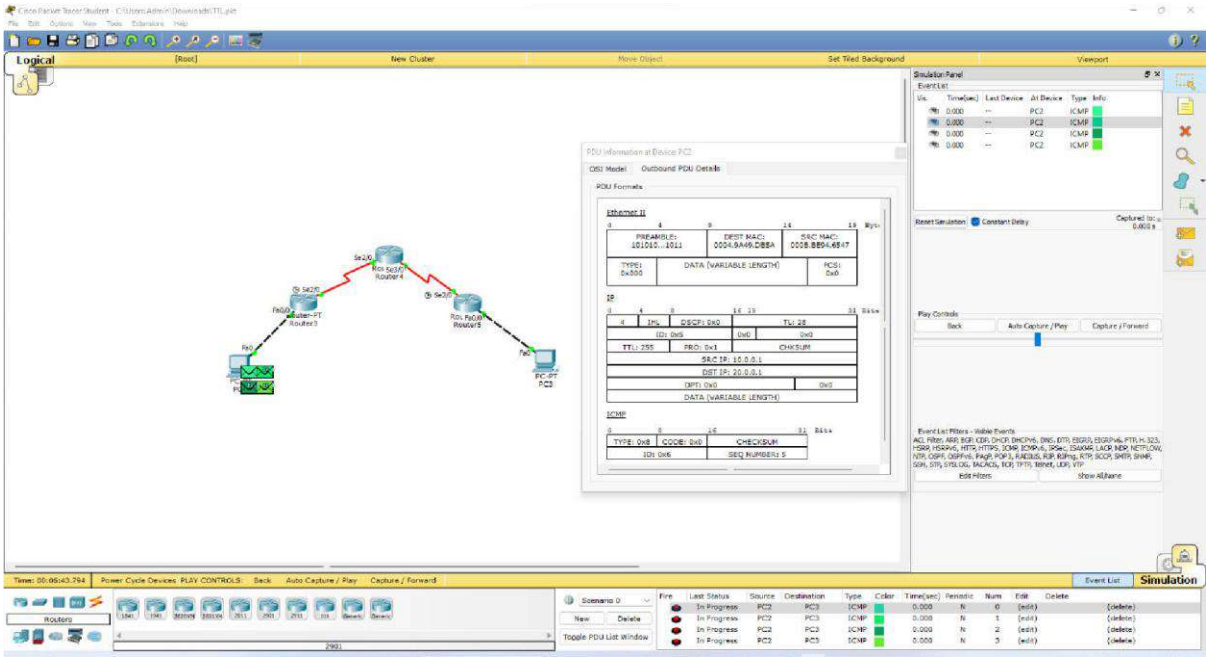
Topology:

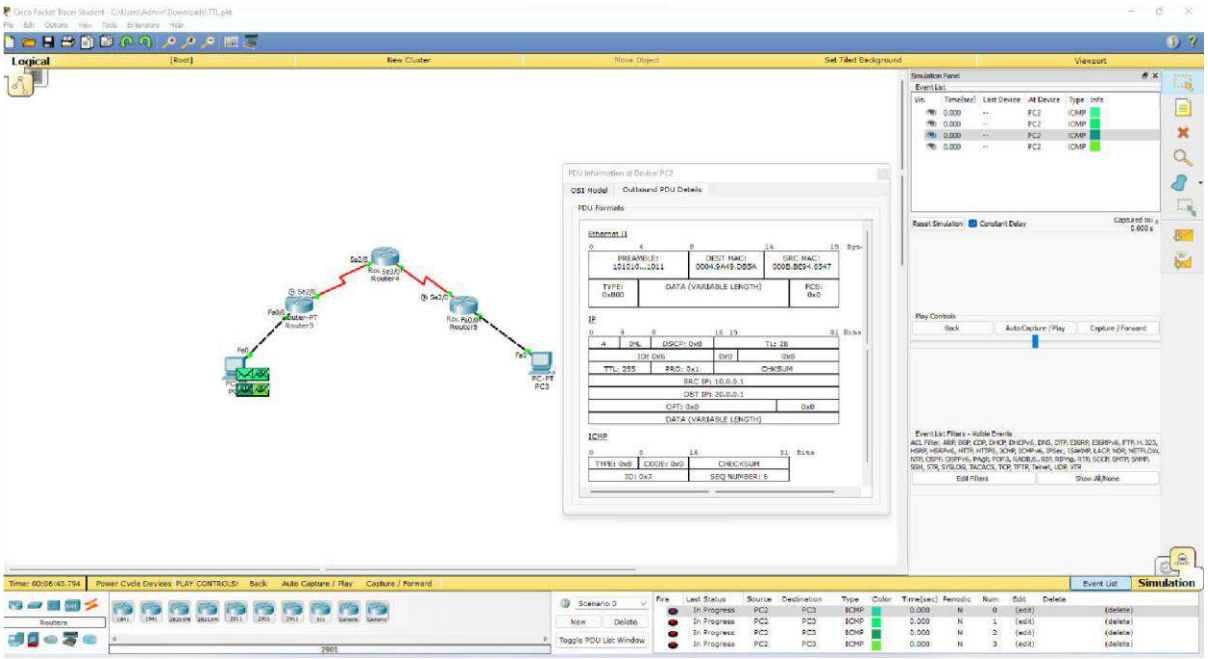


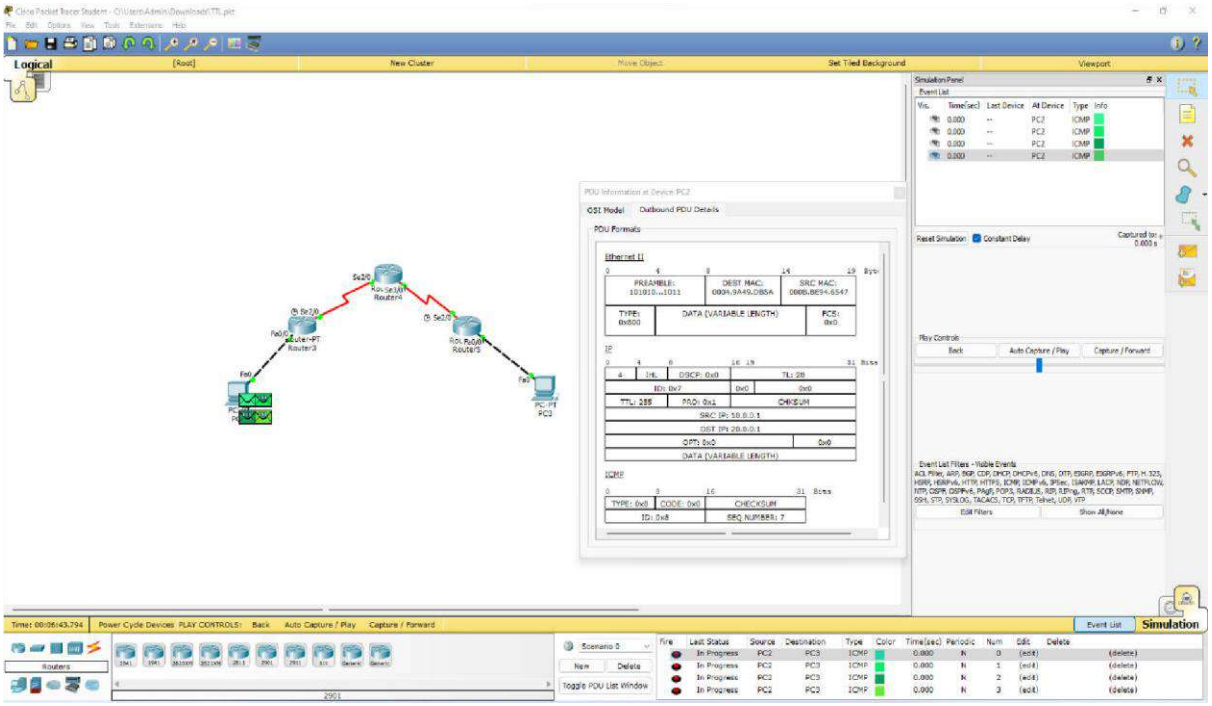
Sending PDU from one PC to another:

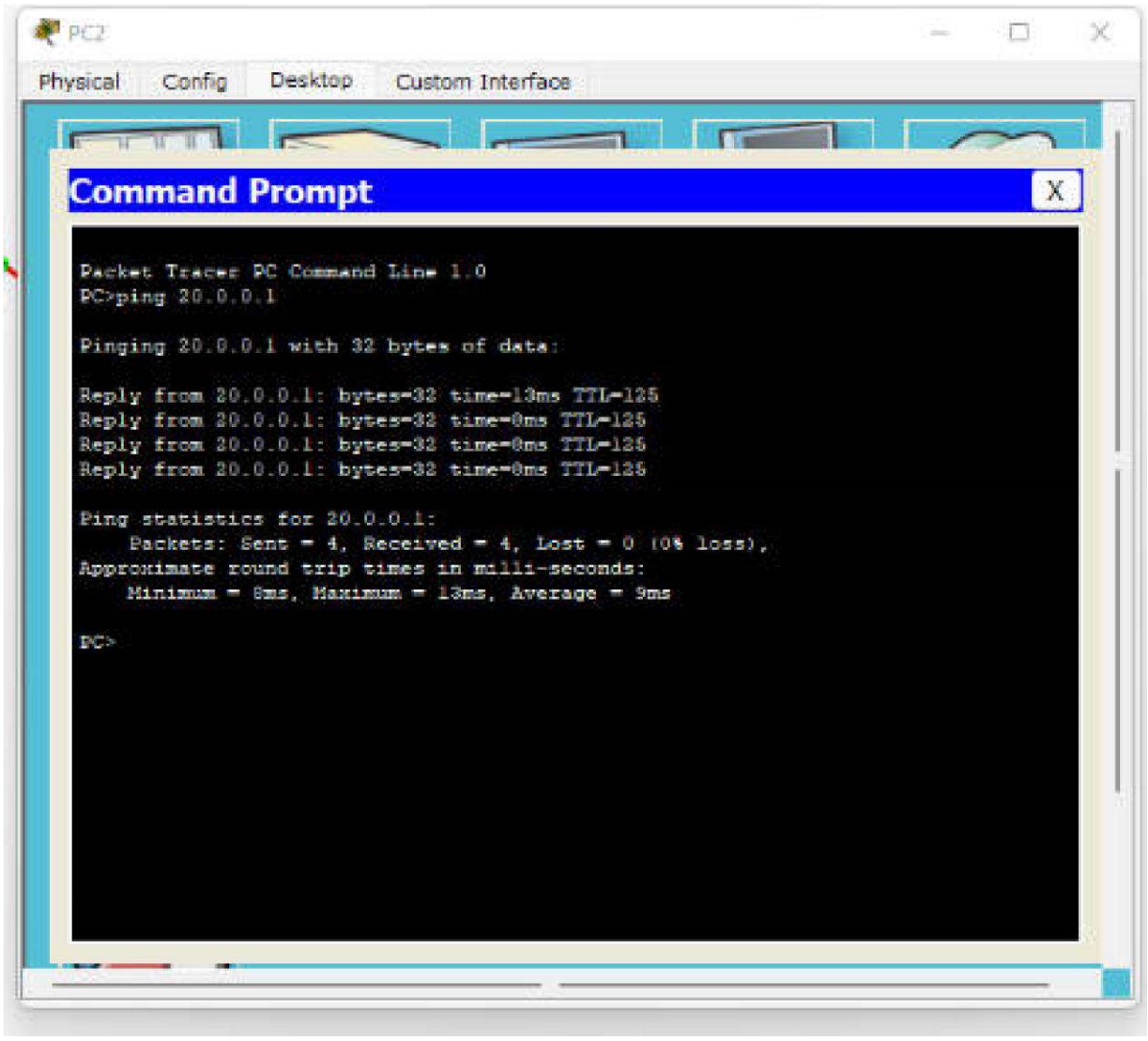










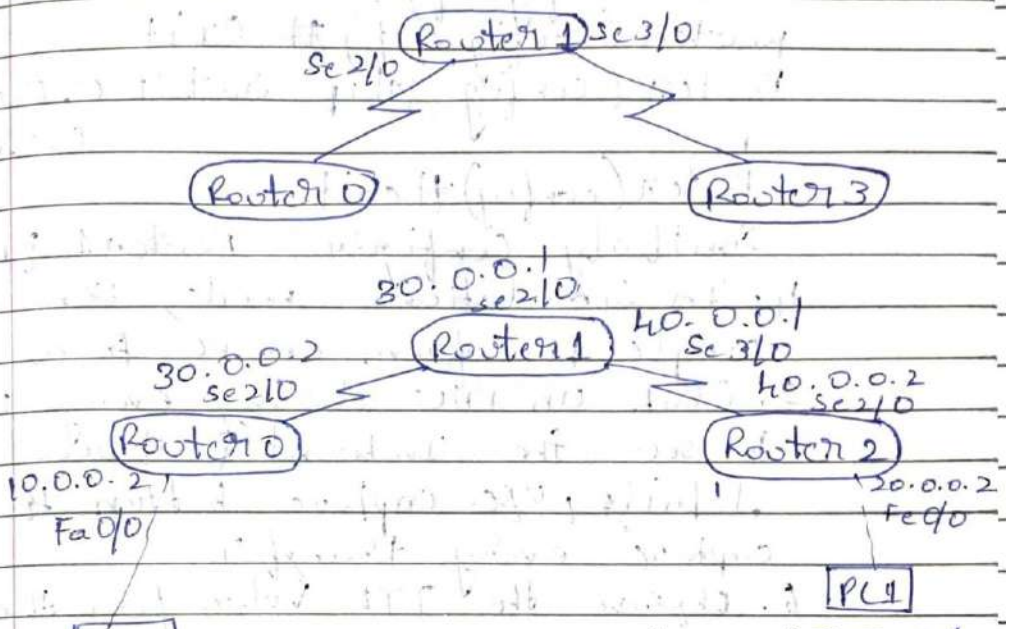

PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=13ms TTL=125
Reply from 20.0.0.1: bytes=32 time=0ms TTL=125
Reply from 20.0.0.1: bytes=32 time=0ms TTL=125
Reply from 20.0.0.1: bytes=32 time=0ms TTL=125
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 13ms, Average = 9ms
PC>

Date __/__/__
Page _____

(12) Title: Demonstrate the TTL/Life of a Packet

Aim: To understand TTL/Life of a packet

Topology:



Procedure:

1. Create a topology with 2 pcs & 3 routers as shown above.
2. Configure IP address as 10.0.0.1 & 20.0.0.1 for PC0 & PC1 respectively.
3. Configure IP addresses for routers and set default gateway.

Router 0

Router # config t
Router (config) # interface fa 0/0
Router (config-if) # ip address 10.0.0.2
255.0.0.0

Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface Se2/0
Router(config-if)# ip address 30.0.0.2 255.0.0.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# ip router 0.0.0.0 0.0.0.0 30.0.0.1
Router(config)# exit

Similarly configure Router1 & Router2

4. In simulation mode, send a simple PDU from one PC to another.
5. Click on PDU during every transfer to see the Inbound & outbound PDU details, use capture button to capture every transfer.
6. Observe the TTL value for the PDU when it crosses each router.

Result:

Sending PDU from PC0 to PC1

PDU information at PC0:
outbound PDU details - TTL = 255

PDU info. at Router 0:
Inboond PDU details - TTL = 255
outbound PDU details - TTL = 254

PDU info at Router 1:
Inbound PDU details - TTL = 254
outbound PDU details - TTL = 253

PDU info at Router 2:
Inboond PDU details - TTL = 253
outboond PDU details - TTL = 252

Observation:

The value of TTL decreases/changes by 1 every time it crosses a router

CYCLE 2

Experiment No. 1

Title:

Write a program for error detecting code using CRC-CCITT (16-bits).

Code:

```
#include<stdio.h>
char m[50],g[50],r[50],q[50],temp[50];
void caltrans(int); void
crc(int); void calram(); void
shiftl(); int main() { int n,i=0;
char ch,flag=0; printf("Enter
the frame bits:");
while((ch=getc(stdin))!='\n')
m[i++]=ch; n=i;
for(i=0;i<16;i++)
m[n++]='0';
m[n]='\0';
printf("Message after appending 16 zeros:%s",m);
for(i=0;i<=16;i++)
g[i]='0';
g[0]=g[4]=g[11]=g[16]='1';g[17]='\0';
printf("\ngenerator:%s\n",g); crc(n);
printf("\n\nquotient:%s",q);
caltrans(n);
printf("\ntransmitted frame:%s",m);
printf("\nEnter transmitted freme:");
scanf("\n%s",m); printf("CRC
checking\n"); crc(n); printf("\n\nlast
remainder:%s",r);
for(i=0;i<16;i++) if(r[i]!='0')
flag=1; else continue; if(flag==1)
```

```
printf("Error during transmission");
else
printf("\n\nReceived freme is correct");
} void crc(int
n)
{
int i,j;
for(i=0;i<n;i++)
temp[i]=m[i];
for(i=0;i<16;i++)
r[i]=m[i]; for(i=0;i<n-
16;i++)
{
if(r[0]=='1')
{
q[i]='1';
calram();
}
else
{
q[i]='0';
shiftl();
} r[16]=m[17+i];
r[17]='\0';
for(j=0;j<=17;j++)
temp[j]=r[j]; }
q[n-16]='\0';
} void
calram()
{
int i,j;
for(i=1;i<=16;i++)
r[i-1]=((int)temp[i]-48)^((int)g[i]-48)+48;
}
void shiftl()
{
```

```
int i;
for(i=1;i<=16;i++)
r[i-1]=r[i];
} void caltrans(int n) { int i,k=0; for(i=n-
16;i<n;i++) m[i]=((int)m[i]-
48)^((int)r[k++]-48)+48;
m[i]='\0';
}
```

Output:

```
Enter the frame bits:1011
Message after appending 16 zeros:1011000000000000000
generator:10001000000100001


quotient:1011
transmitted frame:10111011000101101011
Enter transmitted freme:10111011000101101011
CRC checking


last remainder:0000000000000000

Received freme is correct
```

```
Enter the frame bits:1011
Message after appending 16 zeros:1011000000000000000
generator:10001000000100001


quotient:1011
transmitted frame:10111011000101101011
Enter transmitted freme:101
CRC checking


last remainder:0001000000100001 Error during transmission
```

(13) Title:
Write a program for error detecting code using CRC-CCITT

Code:

```
#include<stdio.h>
char m[50], g[50], r[50], q[50], temp[50];
void caltrans(int);
void crc(int);
void calram();
void shift();
int main(){
int n, i=0, flag=0;
char ch;
printf("Enter the frame bits:");
while((ch=getc(stdin))!='\n')
m[i++]=ch;
n=i;
for(i=0; i<16; i++)
m[n++]='0';
r[i]='\0';
printf("message after appending 16 zeros: %s", m);
for(i=0; i<=16; i++)
g[i]='0';
g[0]=g[1]=g[11]=g[16]='1';
g[17]='\0';
printf("\n generator %s\n", g);
crc(n);
printf("\n Quotient: %s\n", q);
caltrans(n);
```

```
printf("In Transmitted frame : %s", m);
printf("Enter transmitted frame :");
    Scanf("%s", m);
printf("CRC checking");
    crc(n);
printf("last remainder = %s", r);
    for(i=0; i<16; i++)
        if(r[i] != '0')
            flag=1;
        else
            continue;
        if(flag==1)
    printf("Error during transmission
                                    ");
    else
printf("Received frame is correct");
}
void crc(int n){
    int i, j;
    for(i=0; i<n; i++)
        temp[i]=m[i];
    for(i=0; i<n+16; i++){
        if(r[0] == '1')
            caldram(n);
    }
    else
        q[i]='0';
        shift();
    }
    for(j=0; j<=17; j++)
        temp[j]=r[j];
    }
    q[n-1] = '1'|'0';
```

```
void caltrem(){
    int i,j;
    for(j=1;i<=16;i++)
    r[i-1]=((int)temp[i]-48)^
           ((int)g[i]-48)+48;
}
void shift(){
  int i;
for(i=1;i<=16;i++)
    r[i-1]=r[i];
}
void caltrans(int n){
    int i,k=0;
    for(i=n-16;i<n;i++)
m[i]=((int)m[i]-48)^
          ((int)r[k++]-48)+48;
    m[i]='10';
}
```

Output:

Enter frame bits: 1011

Message after appending 16 zeros:

1011 0000 0000 0000 0000

Generator: 10001 0000 0001 00001

Quotient: 1011

Transmitted frame: 1011 1011 0001 0111 0101 1011

Enter Transmitted frame: 1011 1011 0001 0111 0101 1011

Last remainder: 0000 0000 0000 0000

Recieved frame is correct.

Experiment No. 2

Title:

Write a program for congestion control using Leaky bucket algorithm.

Code:

```
#include<stdio.h>

int main(){
   int incoming, outgoing, buck_size, n, store = 0;
   printf("Enter bucket size, outgoing rate and no of inputs: ");
   scanf("%d %d %d", &buck_size, &outgoing, &n);

   while (n != 0) {
      printf("Enter the incoming packet size : ");
      scanf("%d", &incoming);
      printf("Incoming packet size %d\n", incoming);
      if (incoming <= (buck_size - store)){
         store += incoming;
         printf("Bucket buffer size %d out of %d\n", store, buck_size);
      } else { printf("Dropped %d no of packets\n", incoming - (buck_size -
         store)); printf("Bucket buffer size %d out of %d\n", store, buck_size);
         store = buck_size;
      }
      store = store - outgoing;
      printf("After outgoing %d packets left out of %d in buffer\n", store,
buck_size); n--
      ;
   }
}
```

Output:

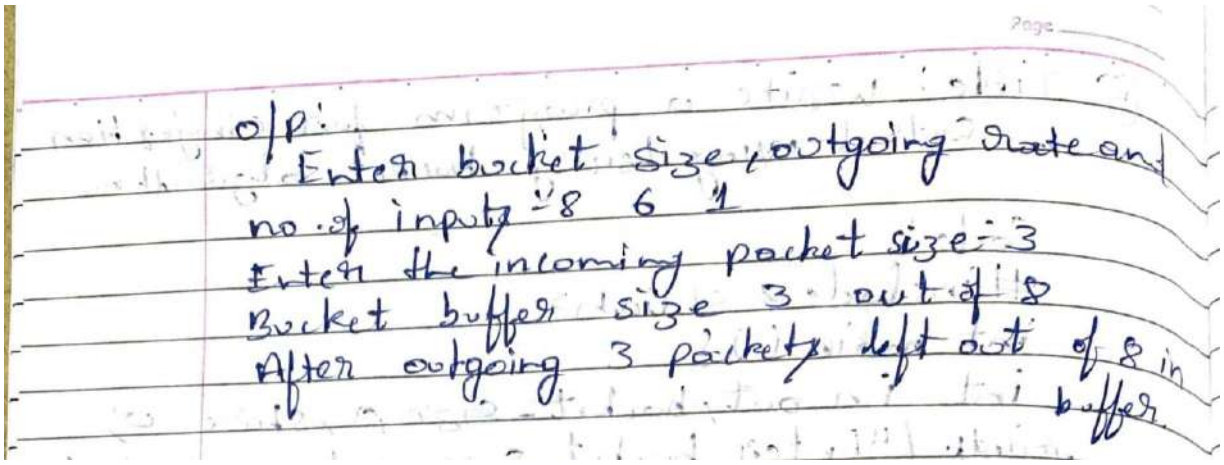
```
Enter bucket size, outgoing rate and no of inputs: 8 6 4
Enter the incoming packet size : 3
Incoming packet size 3
Bucket buffer size 3 out of 8
After outgoing -3 packets left out of 8 in buffer
Enter the incoming packet size : 3
Incoming packet size 3
Bucket buffer size 0 out of 8
After outgoing -6 packets left out of 8 in buffer
Enter the incoming packet size : 4
Incoming packet size 4
Bucket buffer size -2 out of 8
After outgoing -8 packets left out of 8 in buffer
Enter the incoming packet size : 3
Incoming packet size 3
Bucket buffer size -5 out of 8
After outgoing -11 packets left out of 8 in buffer
```

Date __/__/__
Page

② Title: write a program for congestion control using leaky bucket algorithm.

Code:

```
#include<stdio.h>
int main(){
int in, out, bucket_size, n, store=0;
printf("Enter bucket size, outgoing rate & no. of inputs:");
scanf("%d %d %d", &bucket_size, &out, &n);
while(n!=0){
printf("Enter the incoming packet size:");
scanf("%d", &in);
printf("Incoming packet size %d\n", in);
if(in<=(bucket_size-store)){
store+=in;
printf("Bucket buffer size %d out of %d\n", store, bucket_size);
}
else{
printf("Dropped %d no. of packets\n", in-(bucket_size-store));
printf("Bucket buffer size %d out of %d\n", store, bucket_size);
store=bucket_size;
}
store-=out;
printf("After outgoing %d packets left out of %d in buffer\n", store, bucket);
n--;
}
}
```



o/p:
Enter bucket size, outgoing rate and no. of inputs: 8 6 4
Enter the incoming packet size: 3
Bucket buffer size 3 out of 8
After outgoing 3 packets left out of 8 in buffer

Experiment No. 3

Title:

Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

Code:

ClientTCP.py

```
from socket import * serverName =
'127.0.0.1' serverPort = 12000 clientSocket =
socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode()) filecontents =
clientSocket.recv(1024).decode() print ('\nFrom
Server:\n') print(filecontents) clientSocket.close()
```

ServerTCP.py

```
from socket import *
    serverName="127.0.0.1" serverPort = 12000
    serverSocket = socket(AF_INET,SOCK_STREAM)
    serverSocket.bind((serverName,serverPort))
    serverSocket.listen(1) while 1:
        print ("The server is ready to receive")
        connectionSocket, addr = serverSocket.accept()
        sentence = connectionSocket.recv(1024).decode()
        file=open(sentence,"r")
        l=file.read(1024)
        connectionSocket.send(l.encode())
        print ('\nSent contents of ' +
        sentence) file.close()
        connectionSocket.close()
```

Output:

```
The server is ready to receive
```

```
Enter file name: ServerTCP.py

From Server:

from socket import *

serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

```
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file = open(sentence, "r")
    l = file.read(1024)

    connectionSocket.send(l.encode())
    print('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()


Process finished with exit code 0
```

```
The server is ready to receive

Sent contents of ServerTCP.py
The server is ready to receive
```

(3) Title:

use TCP/IP Sockets, write a client-server program to make client sending the file name and the server to send back the contents of requested file if present.

Code:

client TCP.py

```
from socket import *
ServerName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((ServerName, serverPort))
Sentence = input("\n Enter file name:")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print(fileContents)
clientSocket.close()
```

Server TCP.py

```
from socket import
ServerName = "127.0.0.1"
Server_Port = 12000
ServerSockt = socket(AF_INET,
                     SOCK_STREAM)
Server Socketgbind ((ServerName,
                     ServerPort))
Server Socket.ligten (1)
while (1):
    printf ("The Server is ready to
                     recieve")
Connection Socket, addr = server Socket.
                     accept()
Sentence = Connection Socket.recv (1024).
                     decode()
file = open (sentence; "r")
l = file, read (1024)
Connection Socket: send (l. encode())
print ("In sent Contents of " + sentence)
    file. (close())
Connection Socket. (close())
```

~~output~~:

Server TCP.py
The server is ready to Recieve
sent contents of Server TCP.py
client TCP.py

Enter file name : server TCP.py
From Server:
from socket import *
Server Name = '127. 0.0.1'

```
serverport = 12000
ServerSocket = socket(AF_INET,
                      SOCK_STREAM)
ServerSocket.bind((ServerName,
                   ServerPort))
ServerSocket.ligen()
while 1:
print("The Server is ready to
                   receive")
ConnectionSaket, addr = serverSocket.
                        accept()
Sentence = connectionSocket.recv(1024).
                        decode()
file = open(sentence, "or")
    d = file.read(1024)
ConnectionSocket.send(Clientcode())
print('In Sent Events of' + sentence)
    file.close()
    connectionSocket.close()
```

Experiment No. 4

Title:

Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Code:

ClientUDP.py from
socket import *
serverName =
"127.0.0.1" serverPort =
12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName,
serverPort)) filecontents,serverAddress =
clientSocket.recvfrom(2048) print ('\nReply from Server:\n') print
(filecontents.decode("utf-8")) # for i in filecontents:
# print(str(i), end = '')
clientSocket.close()
clientSocket.close()

ServerUDP.py from socket import * serverPort =
12000 serverSocket = socket(AF_INET,
SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",
serverPort)) print ("The server is ready to
receive") while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)

```
        # for i in sentence:
            # print (str(i), end = '')
        file.close()
```

Output:

```
The server is ready to receive
```

```
Enter file name:  ServerUDP.py

Reply from Server:

from socket import *

serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
```

```
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)

    print('\nSent contents of ', end=' ')
    print(sentence)
    #  for i in sentence:
    #  print (str(i), end = '')
    file.close()


Process finished with exit code 0
```

```
The server is ready to receive

Sent contents of  ServerUDP.py
```

(4) Title:

Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Code:

Client UDP.py

```
from Socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET,
                      SOCK_DGRAM)
Sentence = input("\n Enter file name:")
clientSocket.sendto(bytes(sentence,
"utf-8"),(serverName, serverPort))
fileContent, server, Address = clientSocket.
                               recvfrom(2048)
print("\n Reply from Server:\n")
print(fileContents.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

Server UDP.py

```
from Socket import *
serverport = 12000
ServerSocket = socket(AF_INET,
                      SOCK_DGRAM)
ServerSocket.bind("127.0.0.1", serverport)
print("The Server is ready to receive")
```

```
while 1:
Sentence, clientAddress = serverSocket.
                          recvfrom(2048)
Sentence = Sentence.decode("utf-8")
    file.open(sentence, "r")
        con = file.read(2048)
serverSocket.sendto(bytes(con, "utf-8"),
                          clientAddress)
    print("\n Sent contents of ', end = ' ')
        print (sentence)
        file.close()
```

o/p:

```
    serverUDP.py
The server is ready to recieve
Sent contents of serverUDP.py
        clientUDP.py

Enter file name: serverUDP.py
    from socket import *
        serverPort = 12000
serverSocket = socket(AF_INET,
                      SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",
                   serverPort))
while 1:
print("The server is ready to
                          receive")
Sentence, clientAddress = server
                  Socket.recvfrom(2048)
Sentence = sentence.decode("utf-8")
    file.open(sentence, "r")
        l = file.read(2048)
```

Date __/__/__
Page _____

```
serverSocket.sendto(bytes(1,"utf-8"),
                    clientAddress)
print("\n Sent contents of ", end="")
print(sentence)
file.close()
```