VISVESVARAYATECHNOLOGICALUNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

Computer Networks

Submitted by

SAIKUMAR POLICE PATIL (1BM21CS182)

in partial fulfillment for the award of the degree of

BACHELOROFENGINEERING

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 **SAIKUMAR POLICE PATIL (1BM21CS182),** 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. Shyamala G
Assistant Professor
Department of CSE
BMSCE,Bengaluru

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

Index

Sl. No.	Date	Experiment Title	Pg. No.
		CYCLE 1	
1.	16/06	Create a topology with 3 or more end devices using i)Hub ii)Switch iii)Hub & Switch Hybrid	1
2.	23/06	Configure IP address to routers & explore the following messages: ping responses, destination unreachable, request timed out, reply.	8
3.	30/06	Configure default route, static route to the Router	21
4.	14/07	Configure DHCP within a LAN and outside LAN.	33
5.	21/07	Configure Web Server, DNS within a LAN.	48
6.	21/07	Configure RIP routing Protocol in Routers	55
7.	04/08	Configure OSPF routing protocol	59
8.	11/08	To construct simple LAN & understand the concept and operation of ARP	69
9.	11/08	To construct a VLAN & make the PC's communicate among a VLAN	75
10.	18/08	To construct a WLAN & make the nodes communicate wirelessly	83
11.	18/08	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	92

12.	18/08	Demonstrate the TTL/ Life of a Packet	98

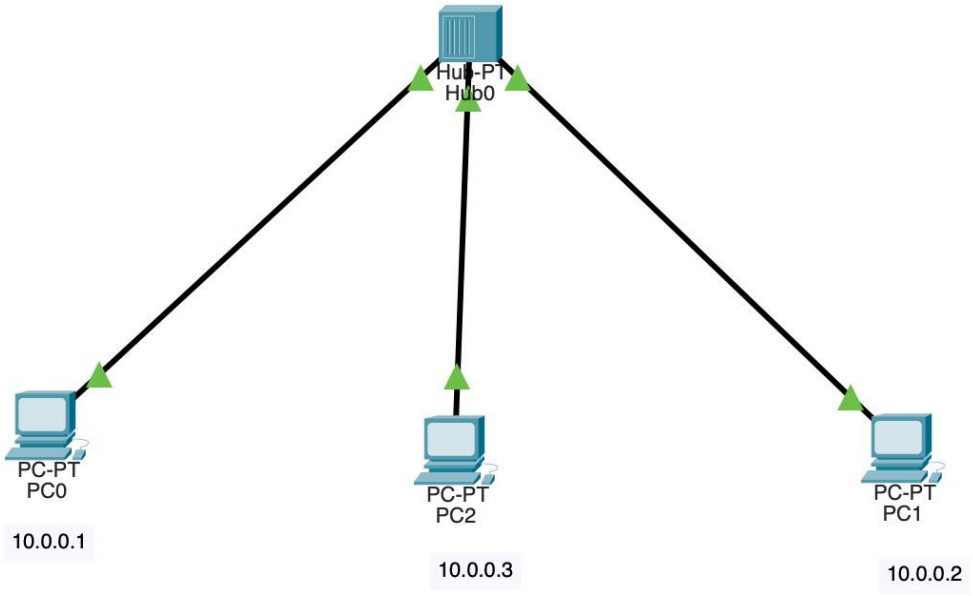
		CYCLE 2	
1.	01/09	Write a program for error detecting code using CRC-CCITT (16-bits).	105
2.	01/09	Write a program for congestion control using Leaky bucket algorithm.	112
3.	01/09	Socket programming using TCP/IP sockets	116
4.	01/09	Socket programming using UDP sockets	121

CYCLE 1

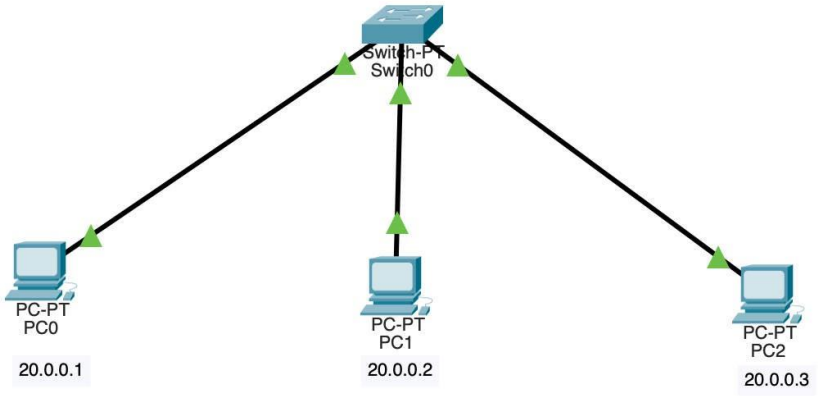
Experiment No. 1

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

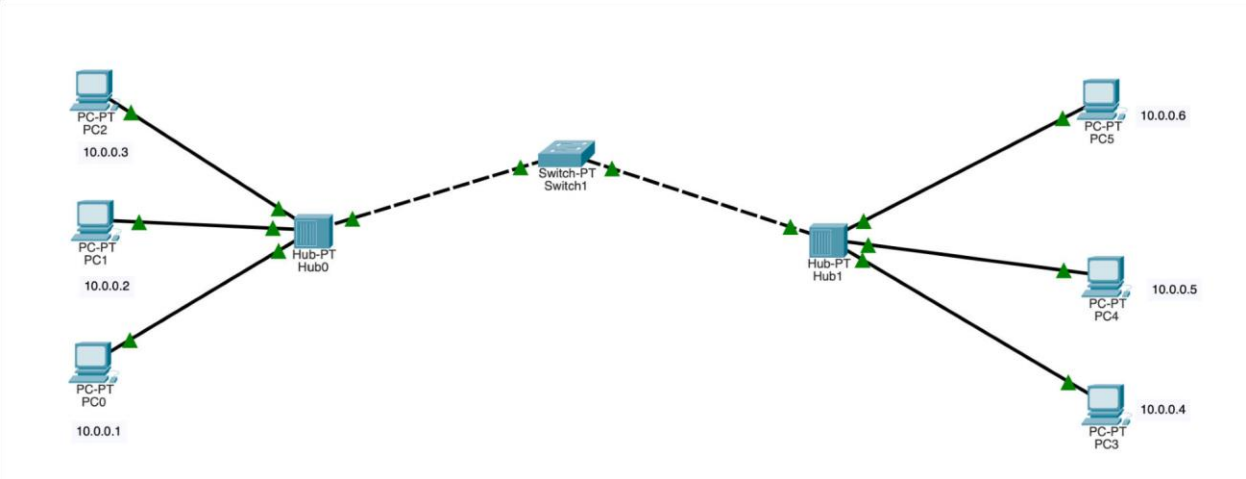
i)Hub ii)Switch iii)Hub & Switch Hybrid **Topology:**



i)Hub topology



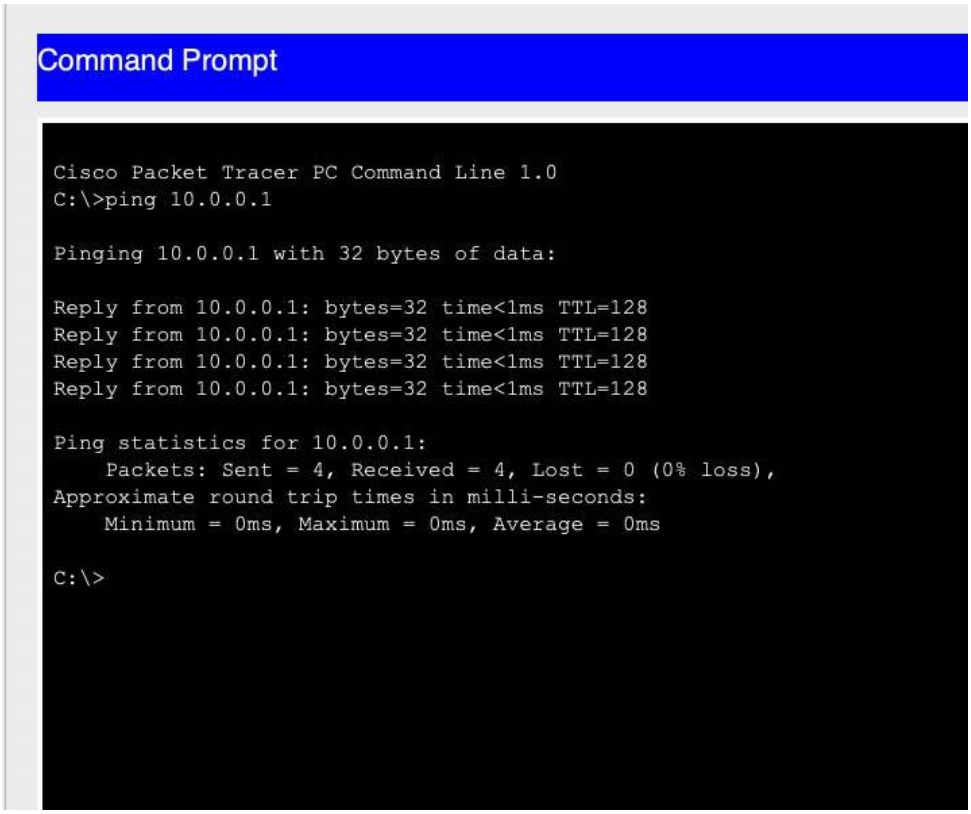
ii)Switch topology

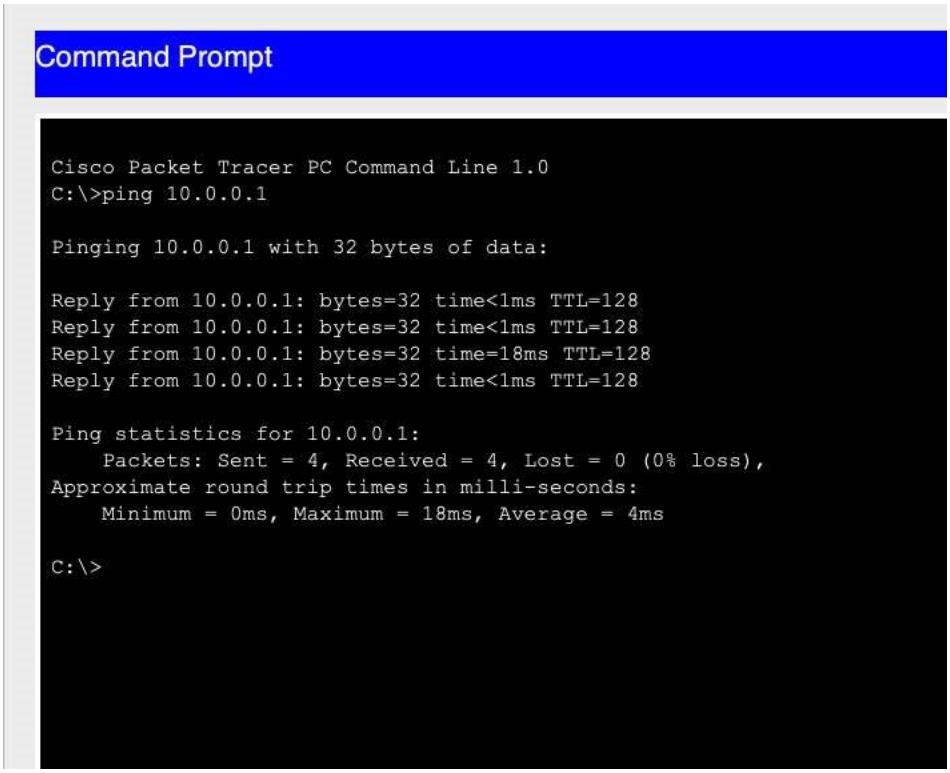


iii)Hub&Switch Hybrid topology

Pinging end devices:

i)





ii)

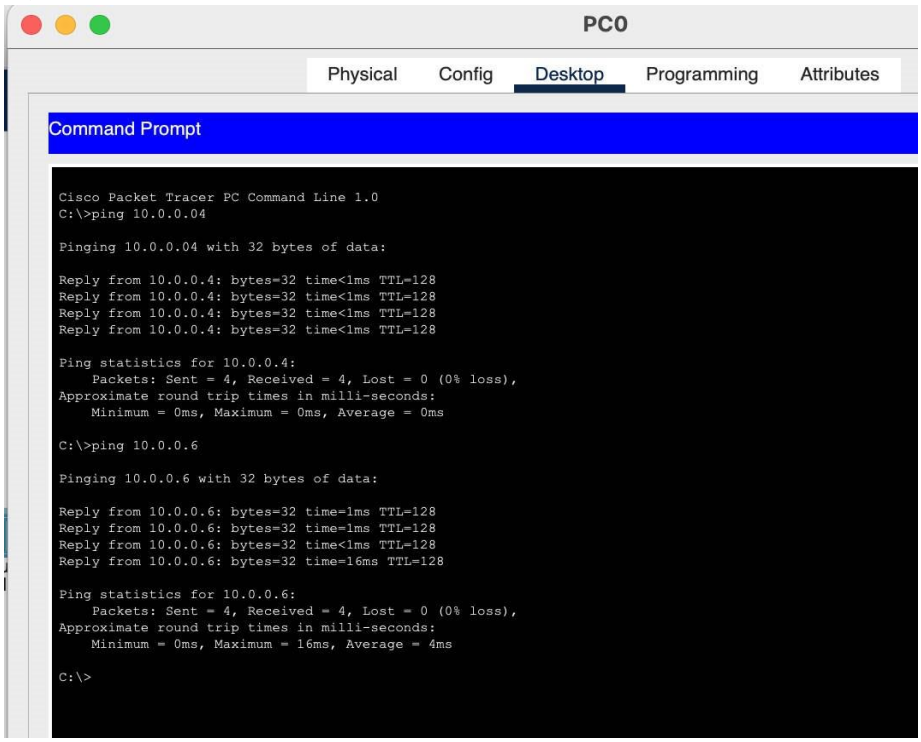
Command Prompt

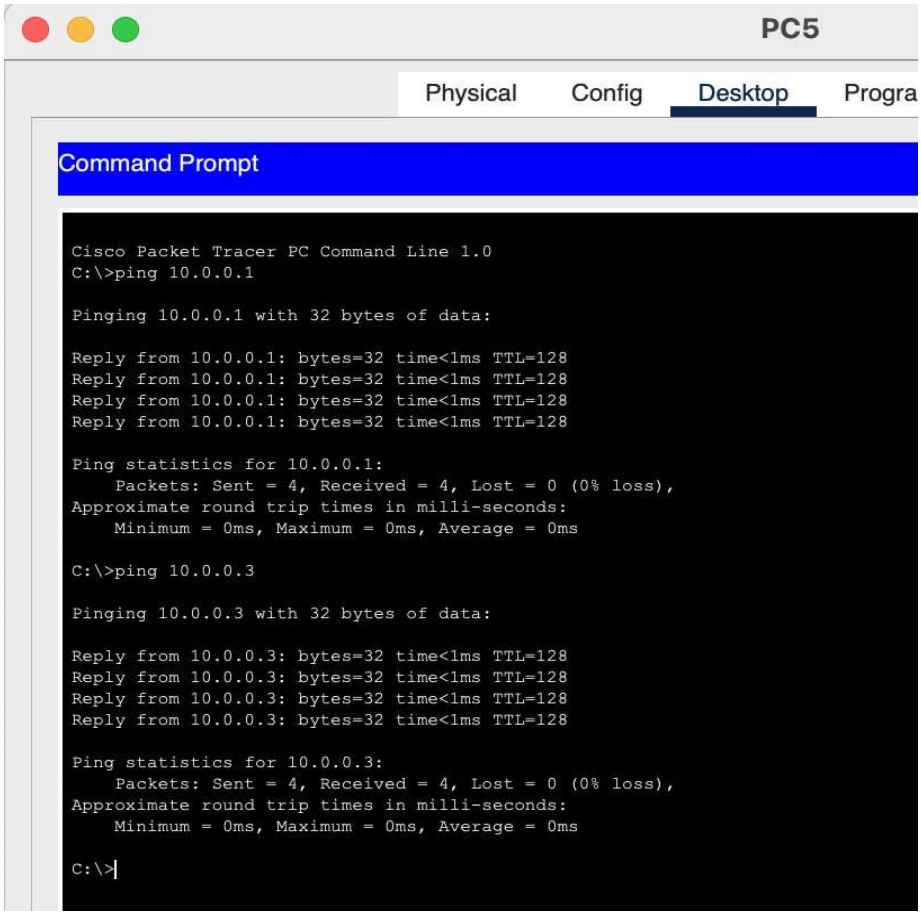
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.3

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
```

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
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
```

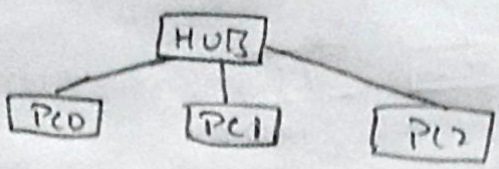
iii)


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
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:\>


PC5
Physical
Config
Desktop
Progra
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
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
C:\>

Experiment-1

Create a topology consisting of 3 or more devices, connected with one help of a hub.

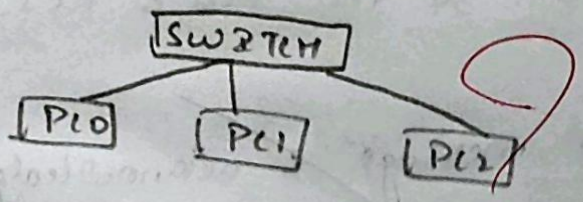


Place one hub and 3 end devices.

Observations:

Whenever a source node sends data in a network, hub recieves the data & sends the data to all the remaining nodes. The node with matching destination address accepts the data.

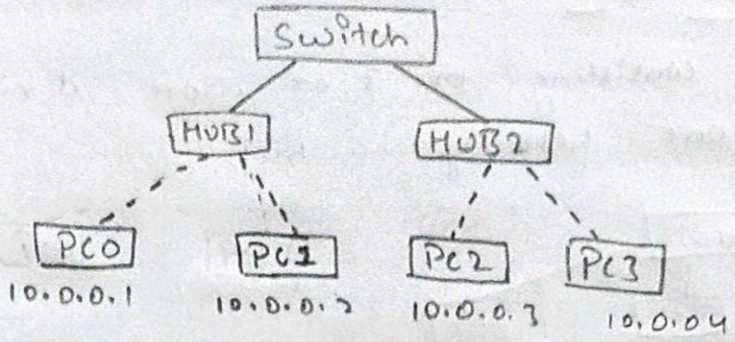
ID.



Observation:

Here the difference is that when the source nodes sends data to other node then switch only sends to nodes whose address matches.

(iii).



Combination of Hub & Switch:

→ Sending a message from PC0 to PC2.

(i). PC0 sends the message, Hub1 recieves the message & transmits to the PC & the Switch PC1 rejects the message, the destination address does not matches.

(ii). Then Switch transmits the message to the HUB2. then HUB2 sends to the PC2 & PC3 simulta-neously.

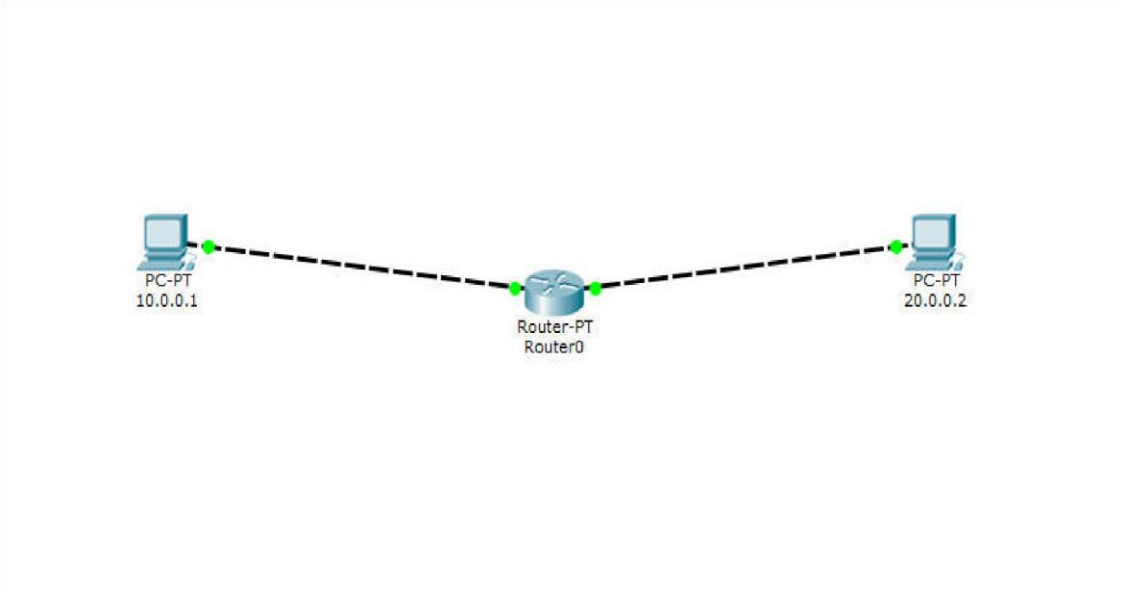
(iii). PC2 accept the message & acknowledge back & PC4 rejects the message.

30/6/23

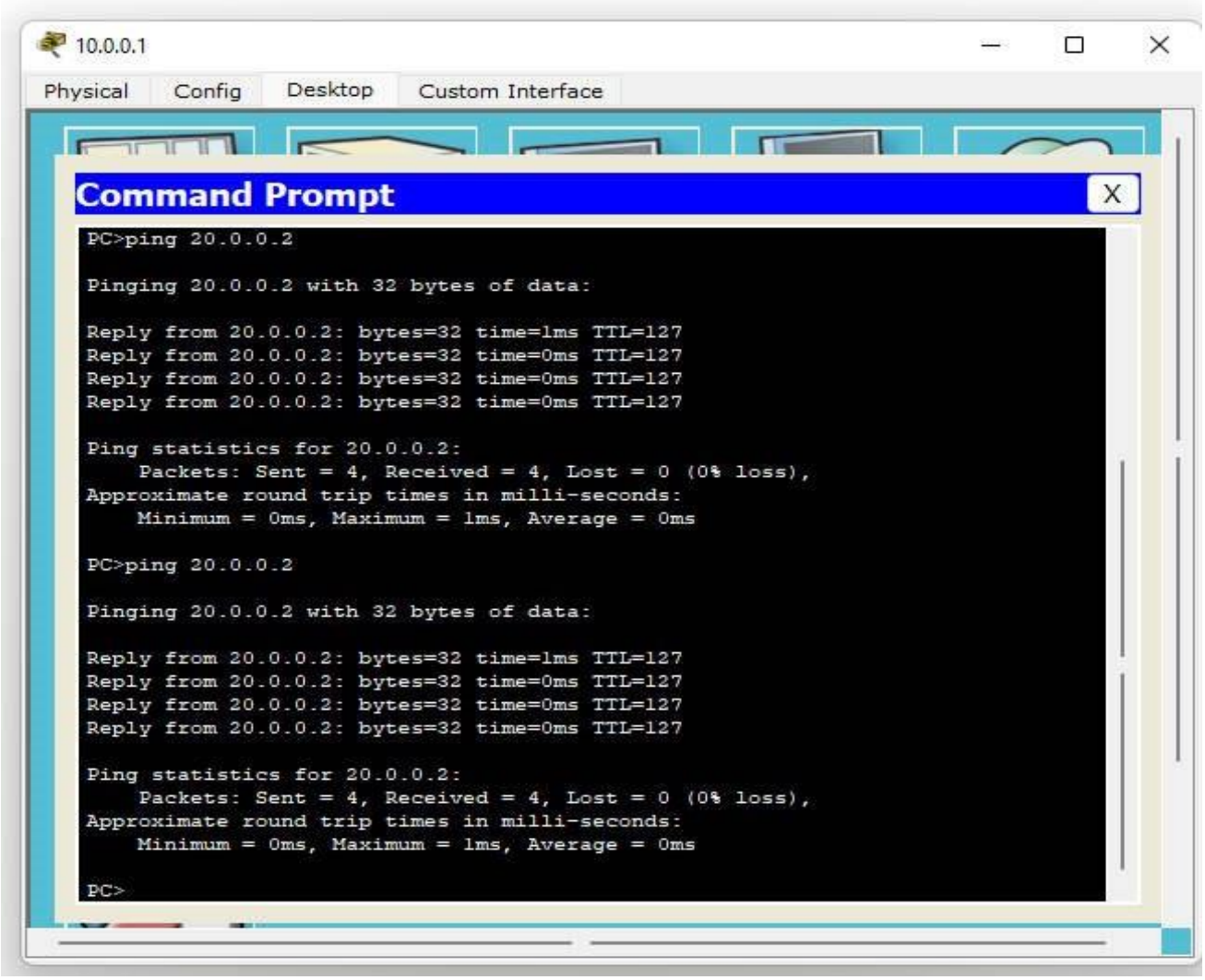
Experiment No. 2

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

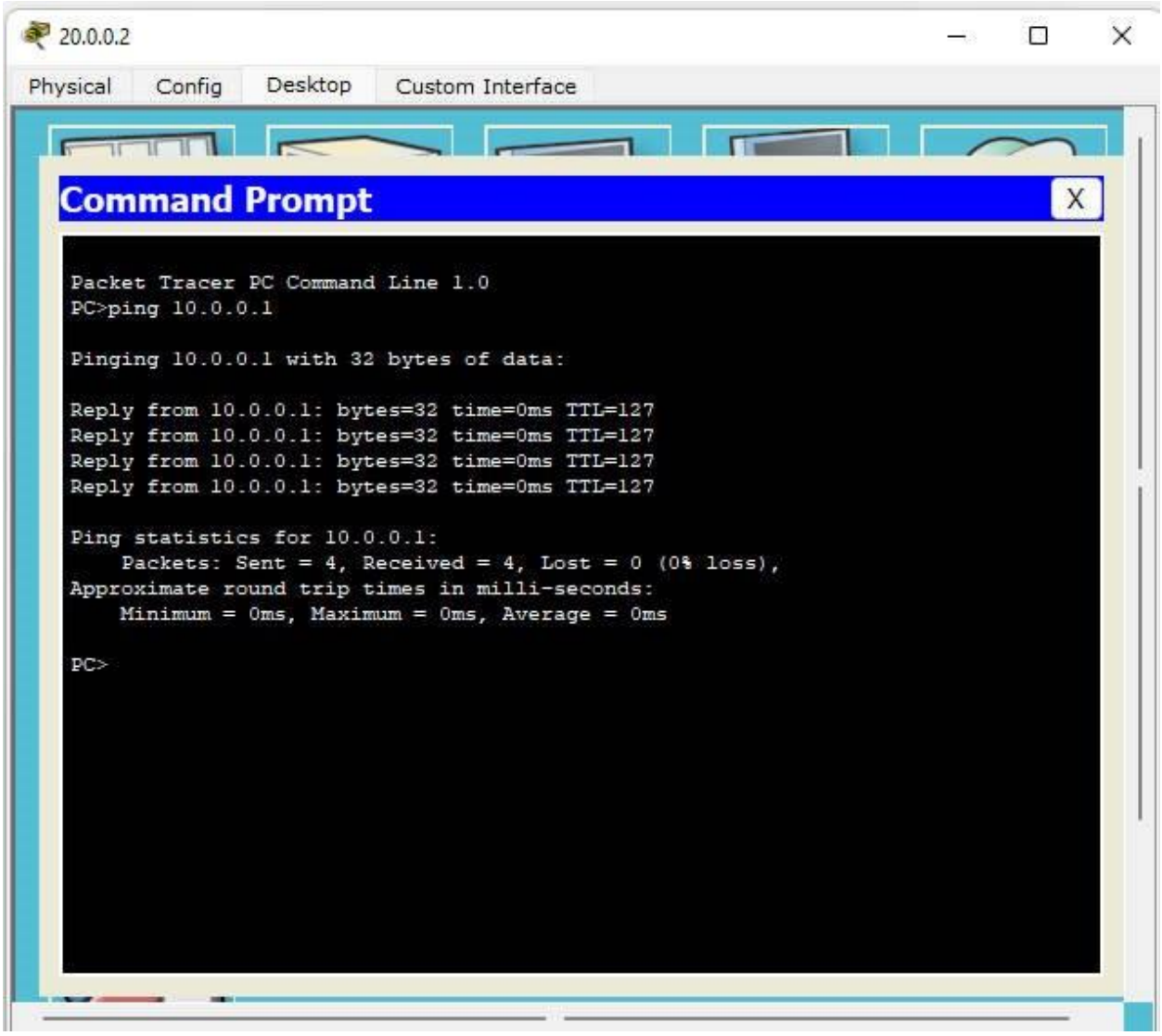
Topology: with single router



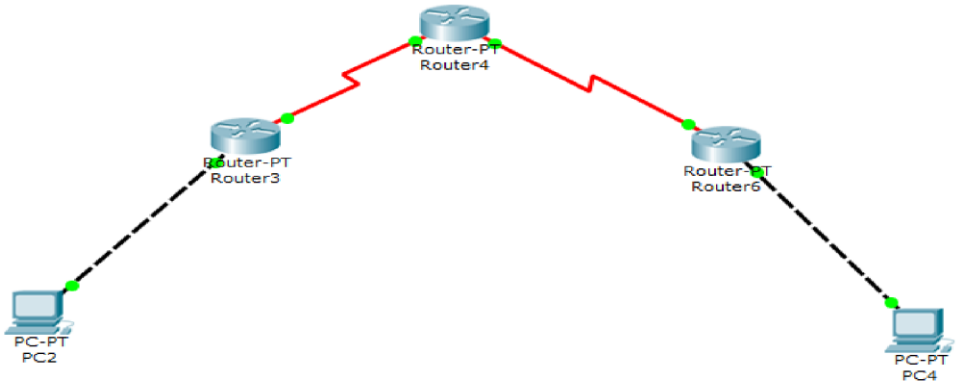
Pinging PC2 from PC1:



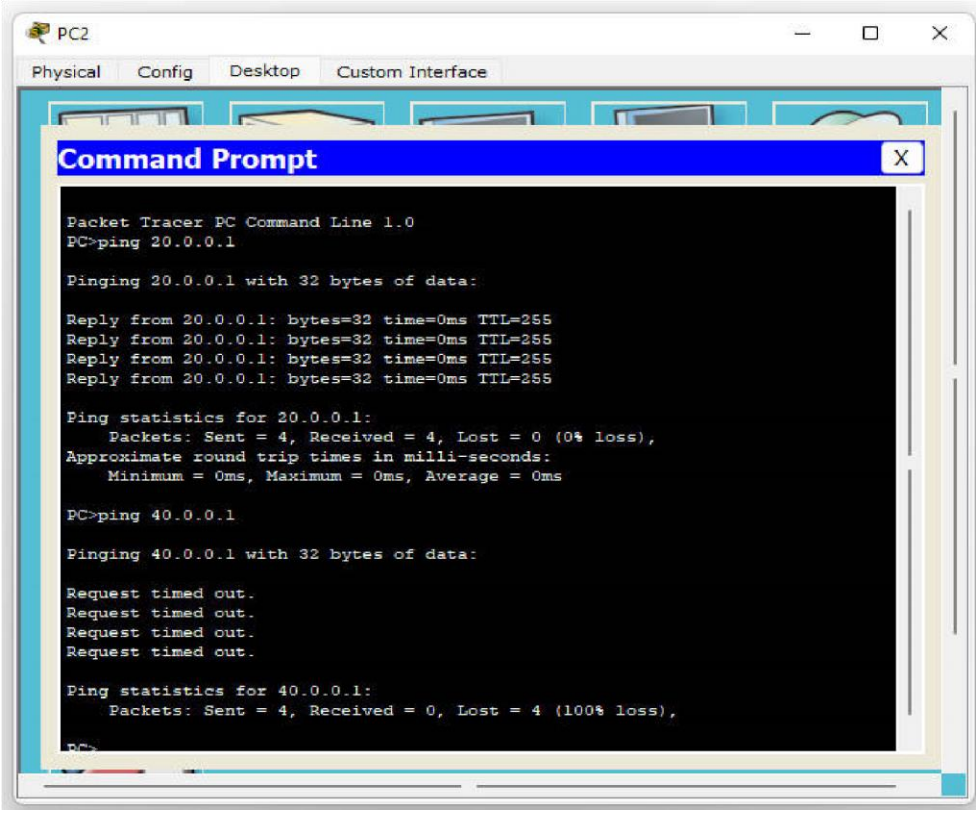
Pinging PC1 from PC2:



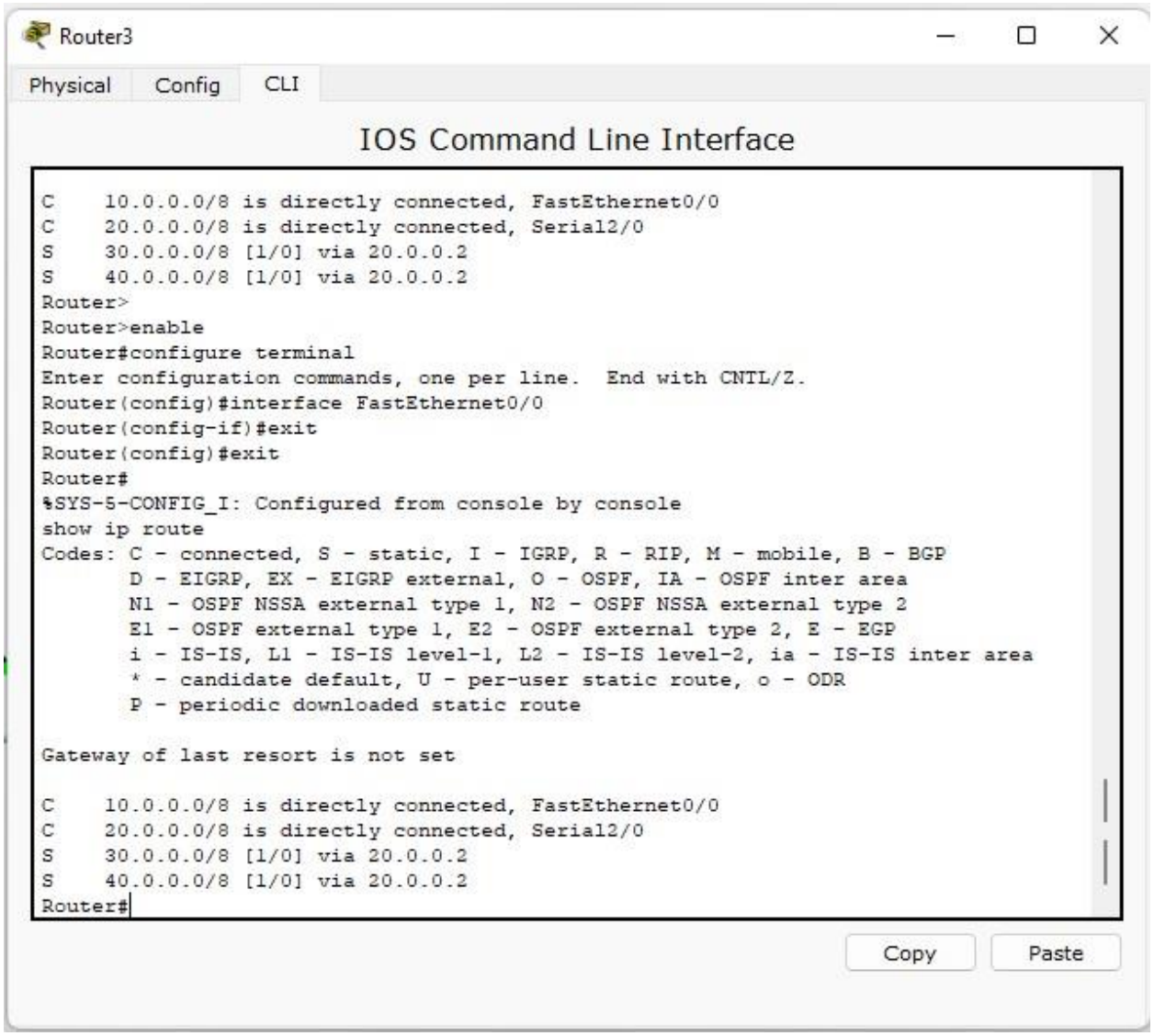
Topology: with 3 Routers

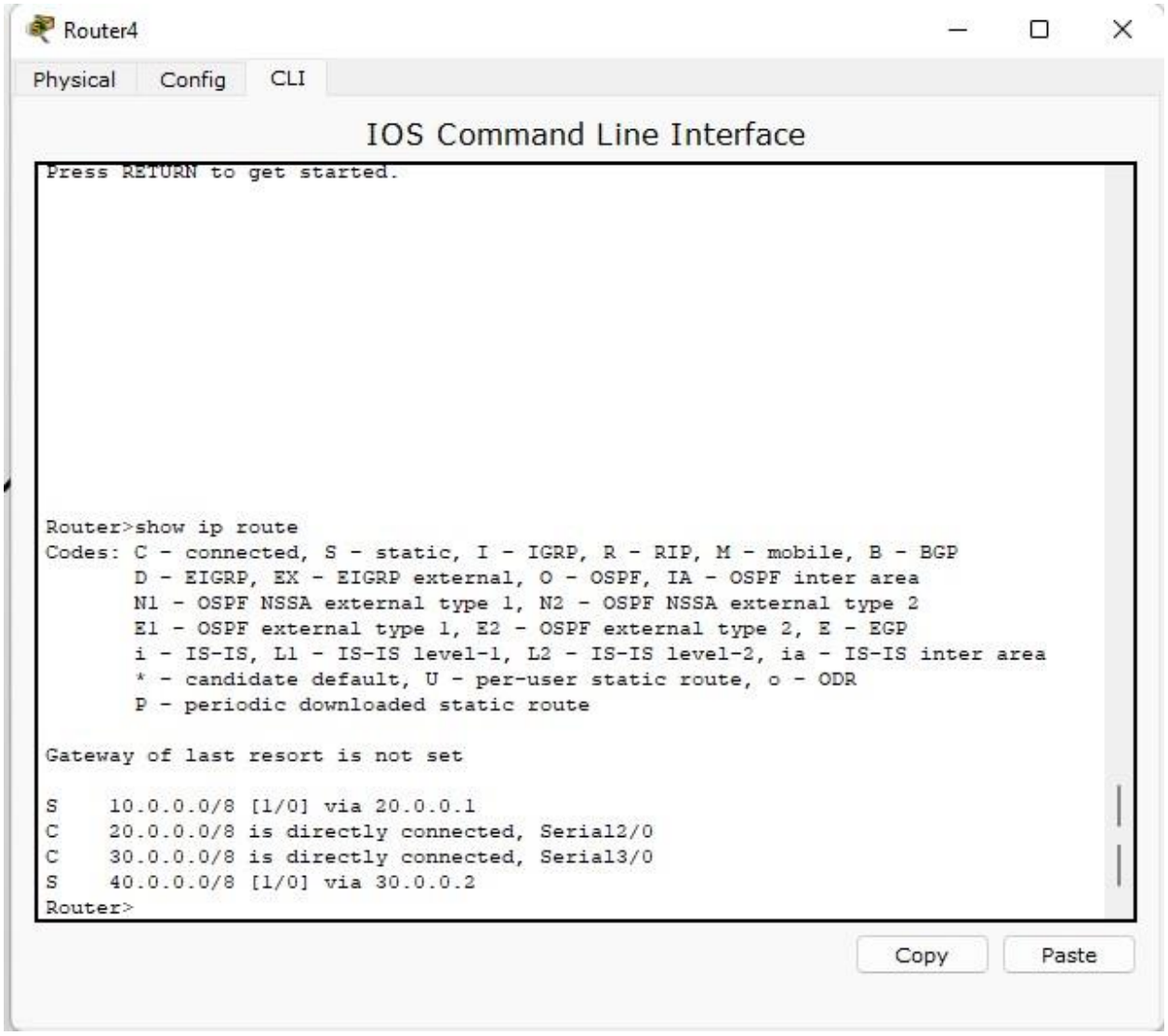


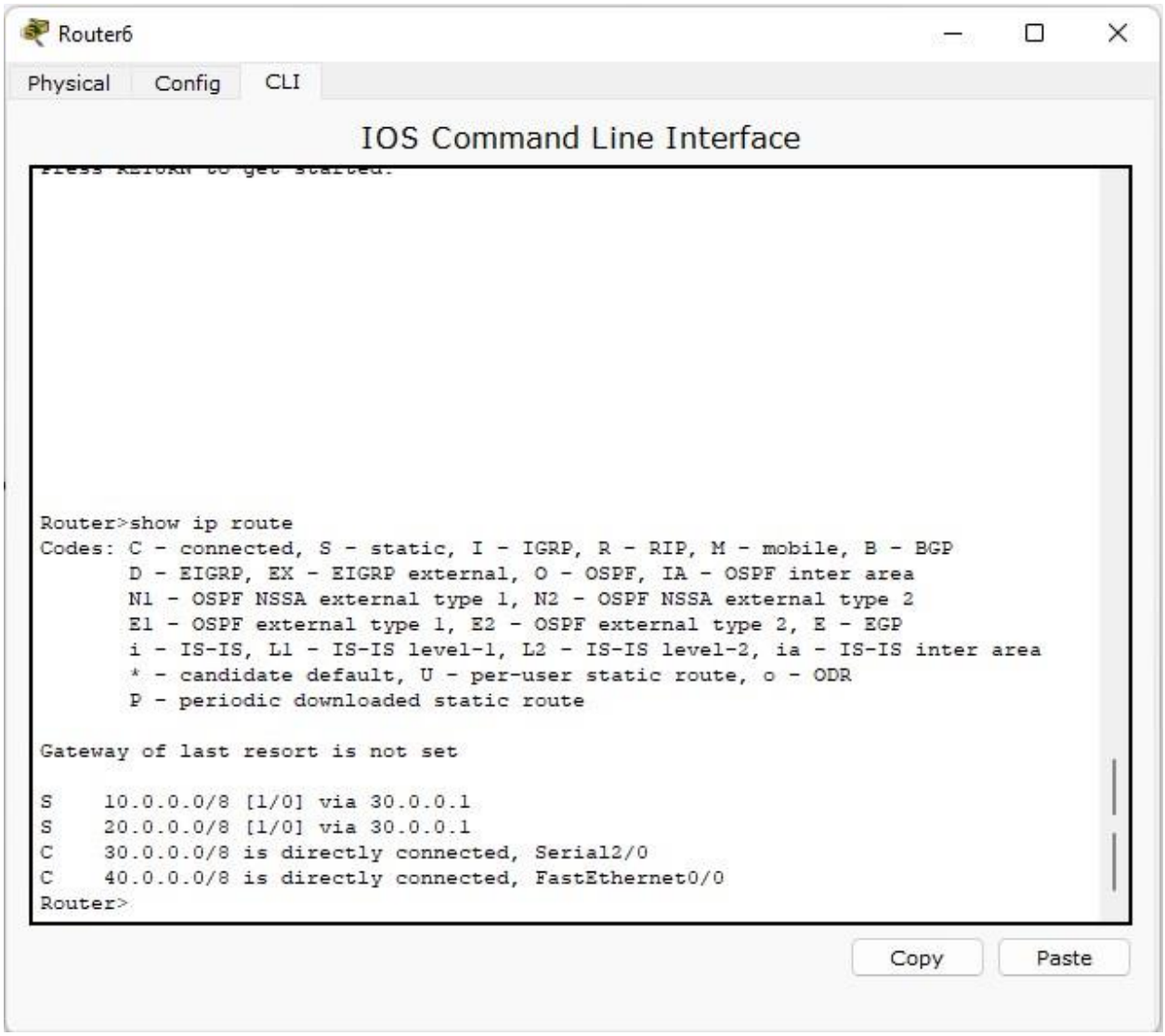
Pinging end device - in different network before setting IP route



IP route - for all routers

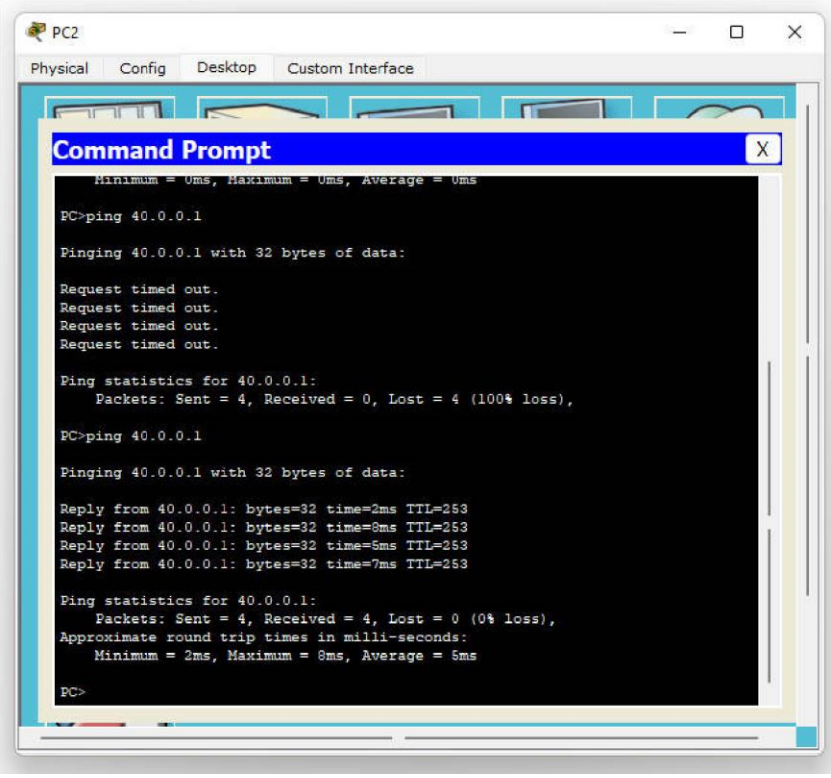



Router4
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started.
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 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

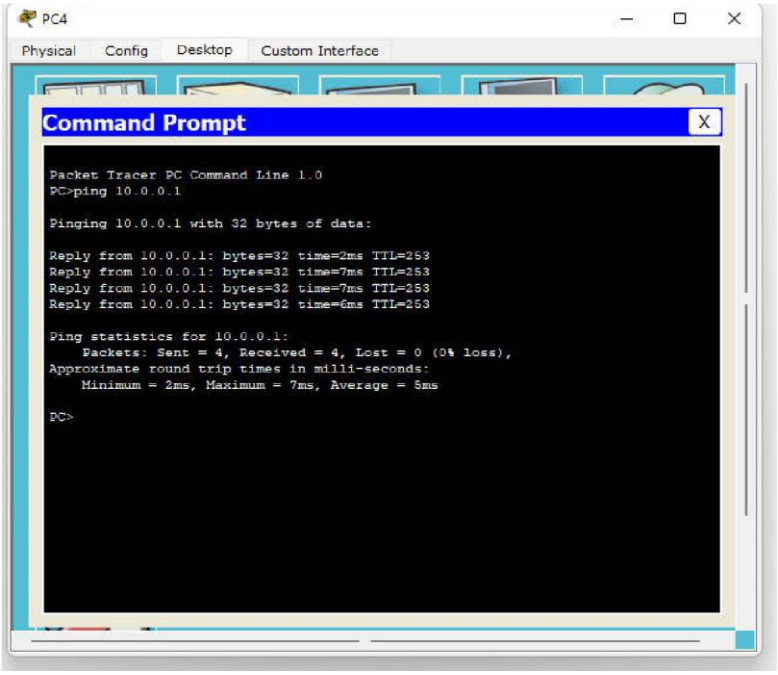

Router6
Physical
Config
CLI
IOS Command Line Interface
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, 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



Pinging PC2 from PC4

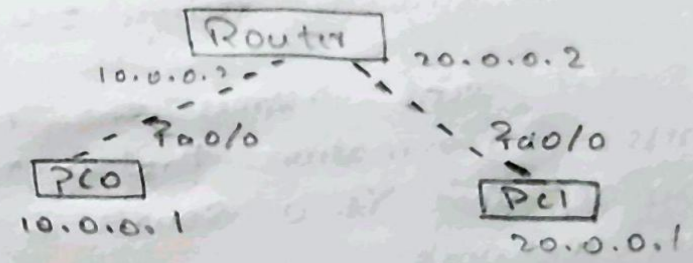


lab-2

Q. Create a topology consisting of 2 devices connected wi the help of routers.

Aim: To use one router and connect two devices of different network.

Topology:



Procedure:

1. Place one Router (Generic) & 2 end devices.
2. Connect the end devices to the Router with appropriate table.
3. Set IP address & default gateway to each device.
4. Configure the Router using the CLI, put IP address same as respective gateway address f end device.
5. Select PC0 & open the command prompt an ping the PC1 using its ip address.

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

Observation:

PC1 replies from 10.0.0.2 Byte = 32 times = 10ms
PC1 replies from 10.0.0.2 Byte = 32 times = 7ms
PC1 replies from 10.0.0.2 Byte = 32 times = 9ms
PC1 replies from 10.0.0.2 Byte = 32 times = 5ms

Number of packets sent = 4 recieved = 4 loss = 0%.

* Each data packet sent across the network contains address information that a router can use to determine that if a source & destination are on same network.

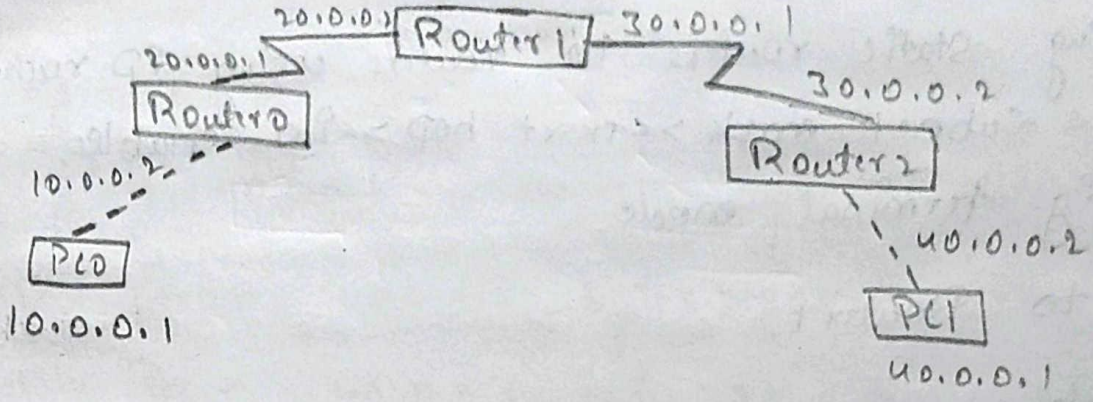
CLI for Router

-> Enable
-> configure terminal
-> Interface fa0/0
-> IP address 10.0.0.2 255.0.0.0
-> NO shutdown
-> exit.

Title: Topology consisting of 3 routers and 2 end devices.

Aim: Connecting 3 router together and end devices.

Topology



Procedure

1. Take 3 generic routers & 2 end devices.
2. Connect the end devices to the routers & blw routers using suitable wire.
3. Set IP address & dgault gateway to each end device of different networks.
4. Configure the router using the CLI, put IP address same as respective gateway address for end device.
5. Select PC0 & open the command prompt & ping PC1 with IP address.

Result: The Intermidiate Router is connected with other two Routers properly with green signal blw the 3 routers & end device

Observation: Pinging PC1 from PC0 works unexpected.

* Three Routers & 2 Pc's are connected as shown in the topology above.
* 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.

PC0 to Router 1.

> enable
config terminal
(config) # interface fa0/0
(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.

— " —

— " —

Request timed out.

Packets: Sent = 4 Recieved = 0

lost = 4 (100%)

→ Adding Static Routes

Route 0:

Router (config)# ip route 30.0.0.0 255.0.0.0 20.0.0.2.

Router (config)# ip route 40.0.0.2 255.0.0.0 20.0.0.

>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data.

Reply from 40.0.0.1 bytes = 32 times = 12ms.

Reply from 40.0.0.1 bytes = 32 times = 9ms

Reply from 40.0.0.1 bytes = 32 times = 5ms

Reply from 40.0.0.1 bytes = 32 times = 3ms

Packets sent = 4 Recieved = 4 lost = 0 (100%).

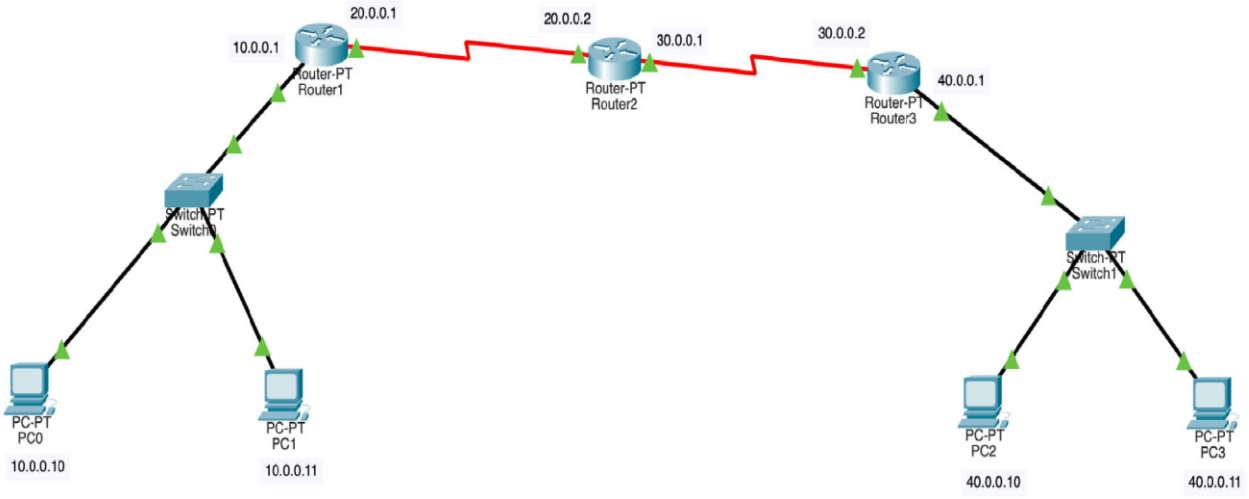
Approx round trip time.

Min = 2ms Max = 12ms Avg = 7ms.

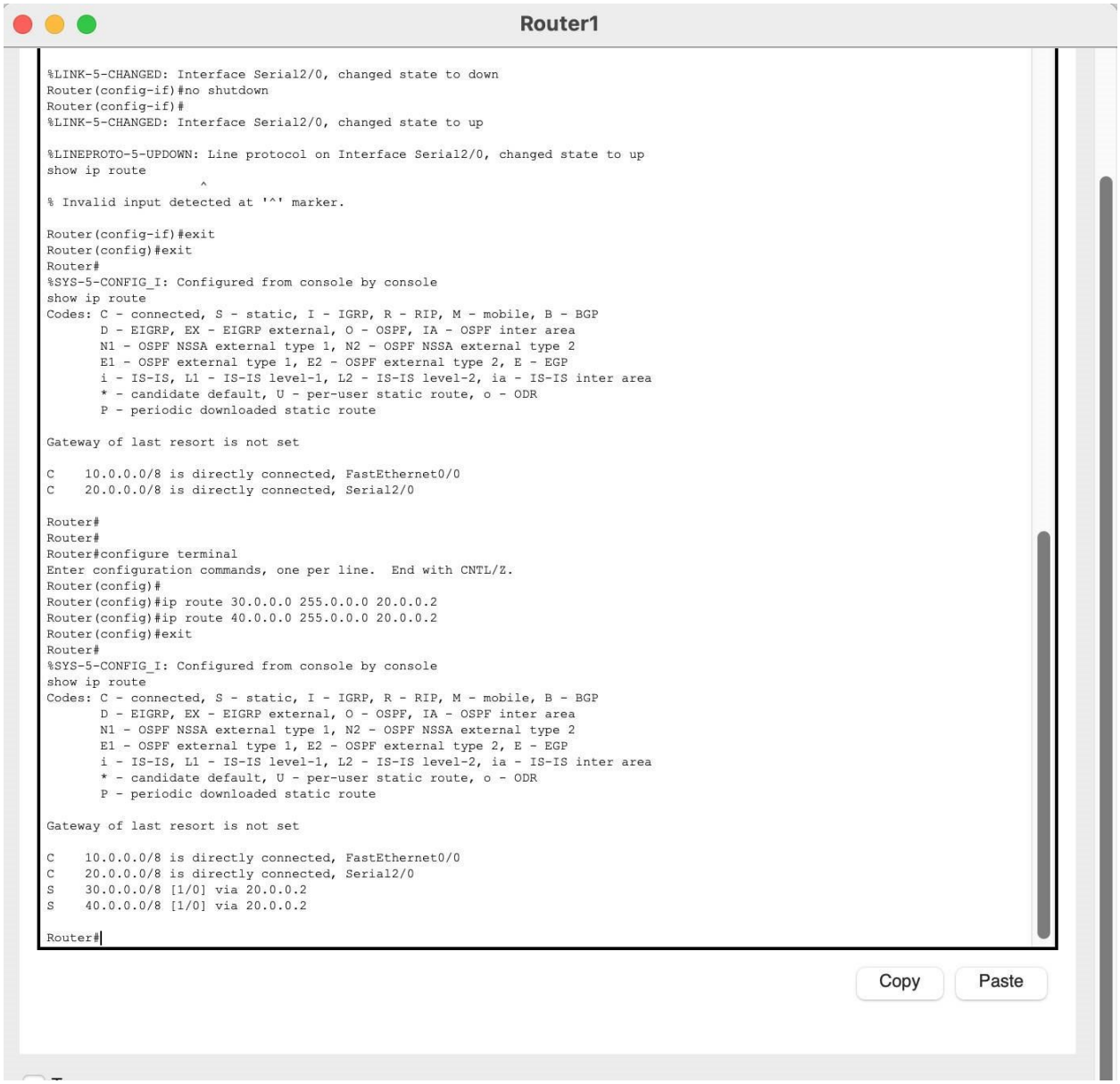
Experiment No. 3

Title:
Configure default route, static route to the Router

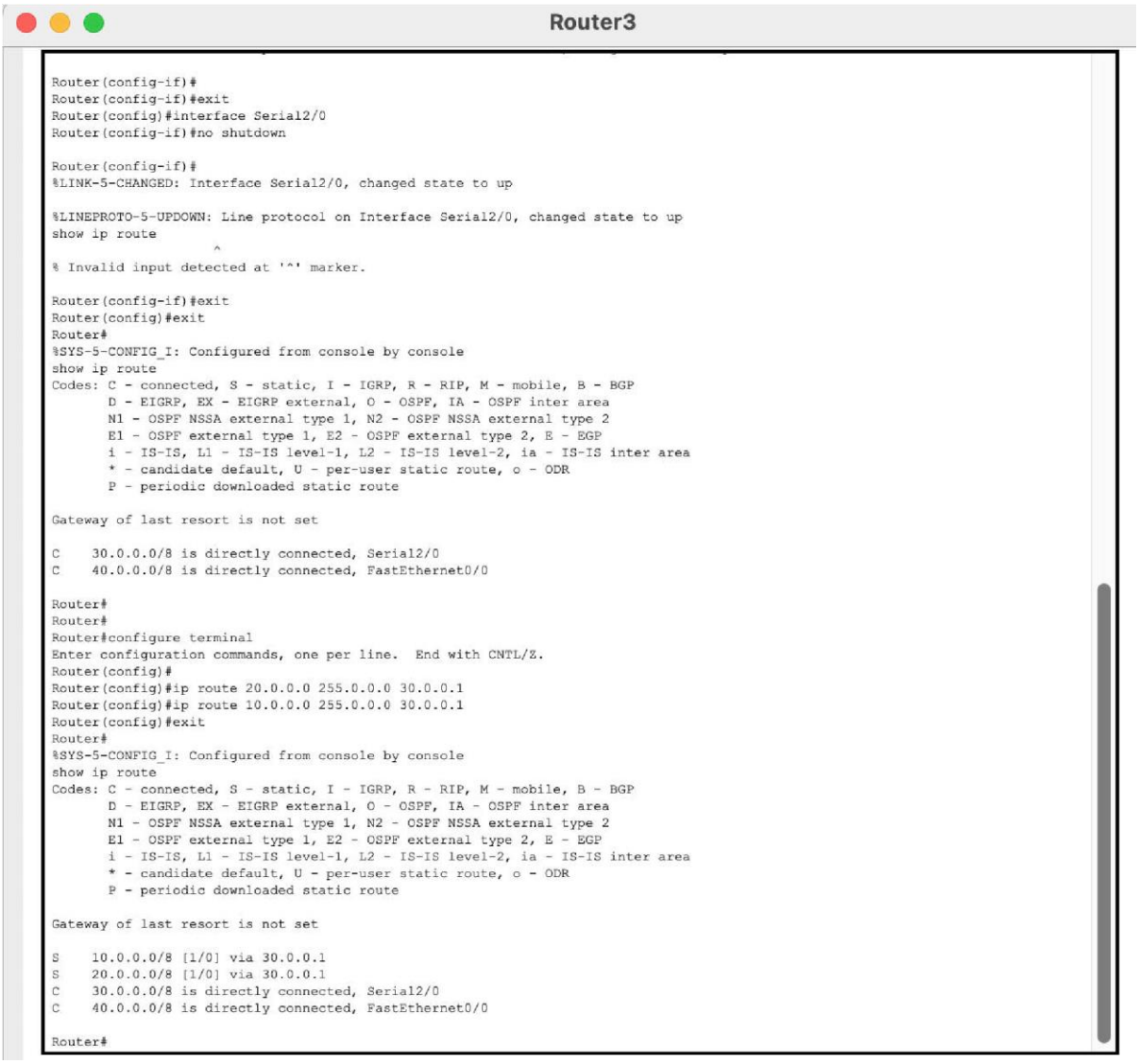
Topology:



IP Route for all routers:

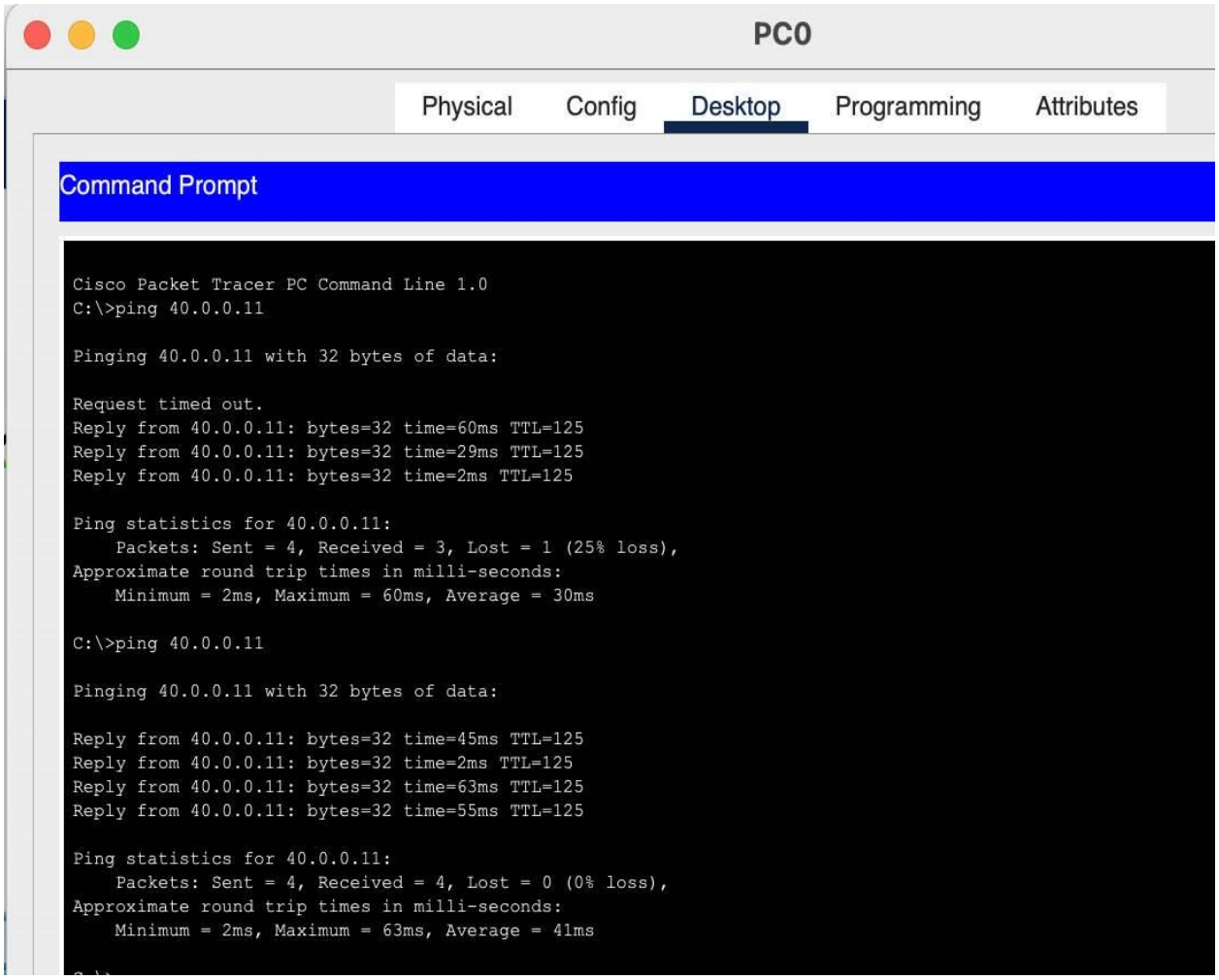



Router2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Router(config-if)#
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol 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#

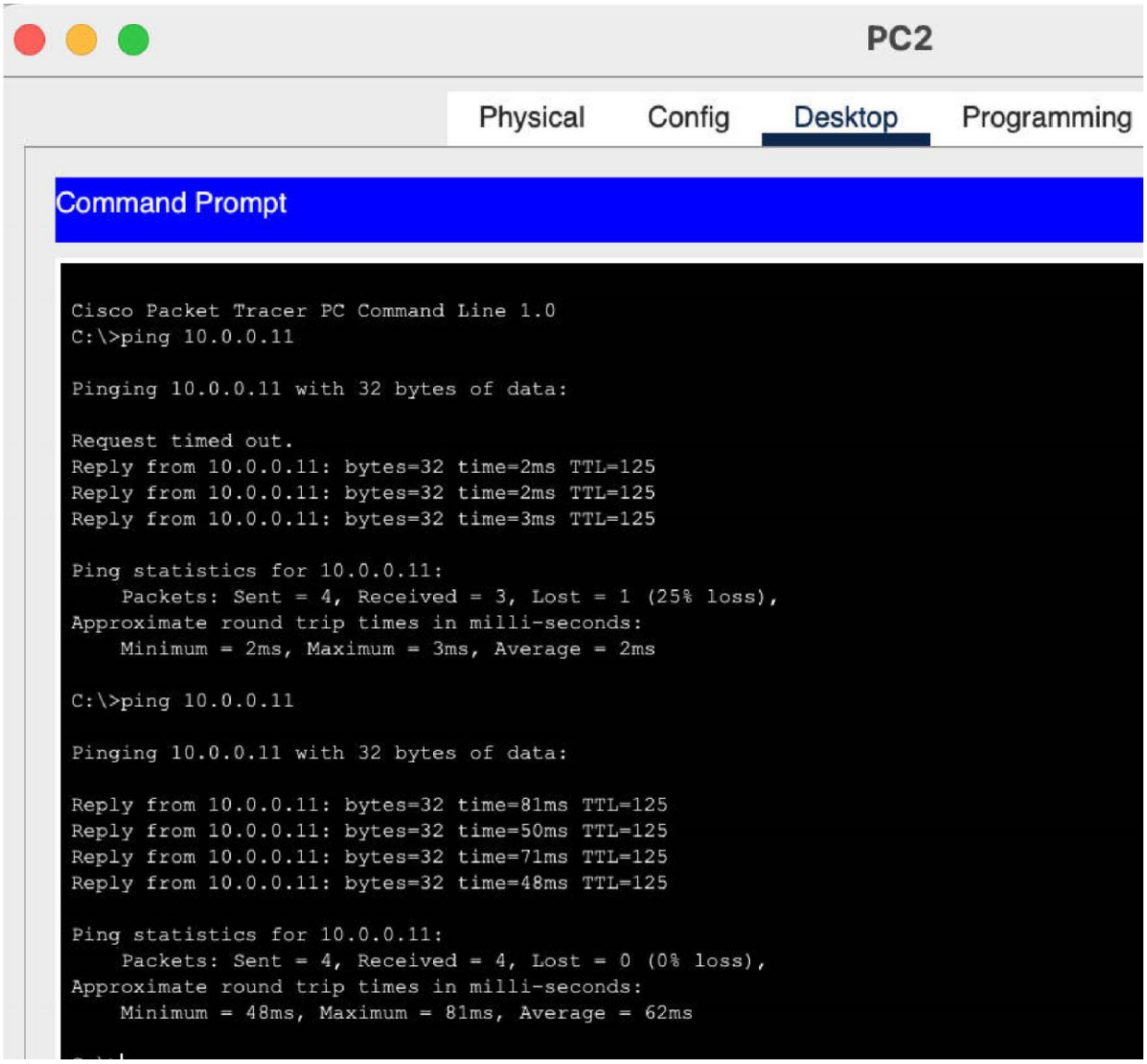

Router3
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
show ip route
^
% Invalid input 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
Router#

Pinging end devices in different network:

Ping PC3 from PC0



Ping PC1 from PC2



Lab-3 Static Routing 30-6-23.

Same topology as of lab-2

Procedure

Adding Static Router.

1. To router 1 for networks 20.0.0.0 and 40.0.0.0

```
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.1
Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.1
```

2. To router 3 for networks 10.0.0.0 & 30.0.0.0

```
Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.2
Router(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2
```

3. To router 2 for networks 10.0.0.0 & 30.0.0.0

```
Router(config)#ip route 30.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
```

Observation

The ip routes have been added to each router which can be seen by running 'show ip route' command.

Router-1

Show ip route

```
C  10.0.0.0/8 is directly connected FastEthernet0/0
S  20.0.0.0/8 [1/0] via 30.0.0.1
C  30.0.0.0/8 is directly connected Serial2/0
S  40.0.0/8 [1/0] via 30.0.0.1
```

Router 2

show ip route

S 10.0.0.0/8 [1/0] via 40.0.0.1

C 20.0.0.0/8 is directly connected FastEthernet 0/0

S 30.0.0.0/8 [1/0] via 40.0.0.1

C 40.0.0.0/8 is directly connected S/0.

Router 3

S 10.0.0.0/8 [1/0] via 30.0.0.2

S 20.0.0.0/8 [1/0] via 40.0.0.2

C 30.0.0.0/8 is directly connected Serial 2/0

C 40.0.0.0/8 is directly connected Serial 3/0

Output

The ping requests to all networks are successful.

From PC0

→ Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply From 20.0.0.1 bytes = 32 time = 10ms TTL = 125

Reply From 20.0.0.1 bytes = 32 time = 2ms TTL = 125

Reply From 20.0.0.1 bytes = 32 time = 2ms TTL = 125

Reply From 20.0.0.1 bytes = 32 time = 8ms TTL = 125

Ping Statistics For 20.0.0.1:

Packets : Sent = 4, Recieved = 4, loss = 0 [0% loss]

=> Ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data

Reply from 40.0.0.2 bytes = 32 time = 7ms TTL = 253

Reply from 40.0.0.2 bytes = 32 time = 7ms TTL = 253

Reply from 40.0.0.2 bytes = 32 time = 7ms TTL = 253

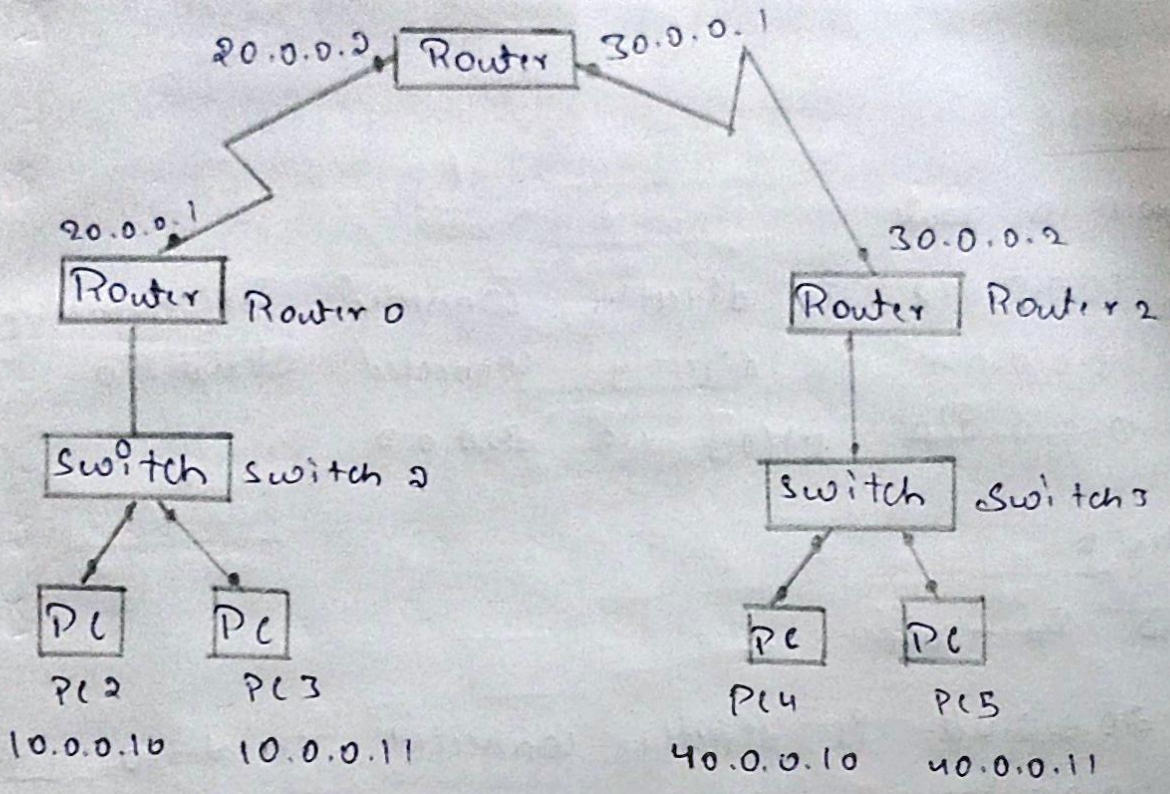
Reply from 40.0.0.2 bytes = 32 time = 7ms TTL = 253

Ping Statistics for 40.0.0.2

Packets: Sent = 4, Recieved = 4, loss = 0 [0% loss].

Lab-4 Default Routing 30-6-23

Topology



Proledure

configuring default router to router 0 & Router 2

Router 1

Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0

Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.

Router 0

Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2

Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

Configuring two static routers to router 1

Observation

The default routes to router 0 & router 2 and the static route to router 1 have been added.

Router 0

Show ip route

C 10.0.0.0/8 is directly connected FastEthernet0/0
C 20.0.0.0/8 is directly connected Serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2

Router 2

Show ip route

C 20.0.0.0/8 is directly connected, Serial 3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 1

Show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial 2/0
C 30.0.0.0/8 is directly connected, Serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

Output

Ping requests : from PC5

→ Ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data

Reply From 10.0.0.11 : bytes = 32, time = 10ms TTL = 12
Reply From 10.0.0.11 : bytes = 32, time = 6ms TTL = 12
Reply From 10.0.0.11 : bytes = 32, time = 8ms TTL = 12
Reply From 10.0.0.11 : bytes = 32, time = 8ms TTL = 12

Ping statistics for 10.0.0.11

Packets : Sent = 4, Received = 4, loss = 0 (0% loss).

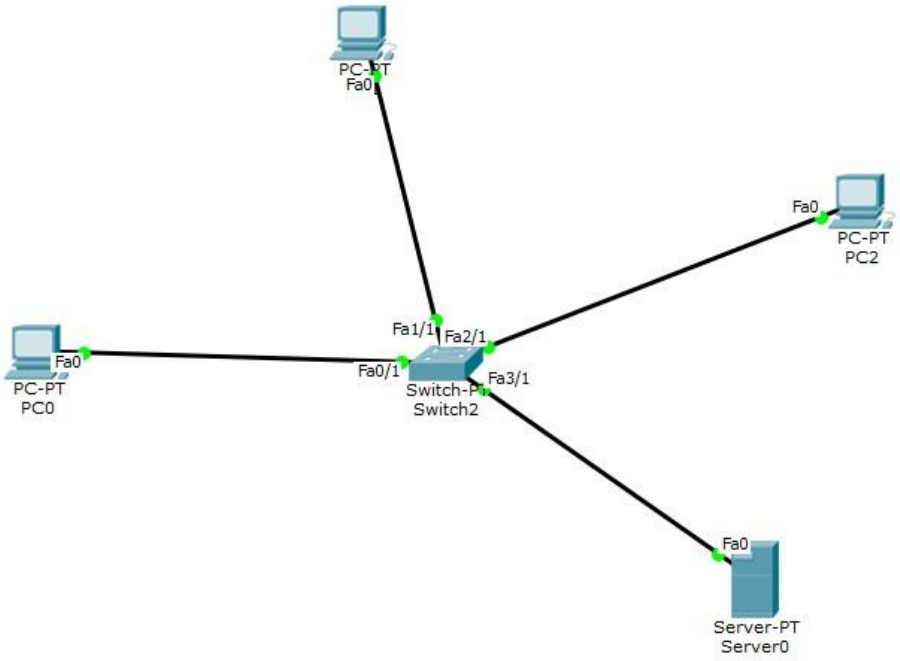
Experiment No. 4

Title:

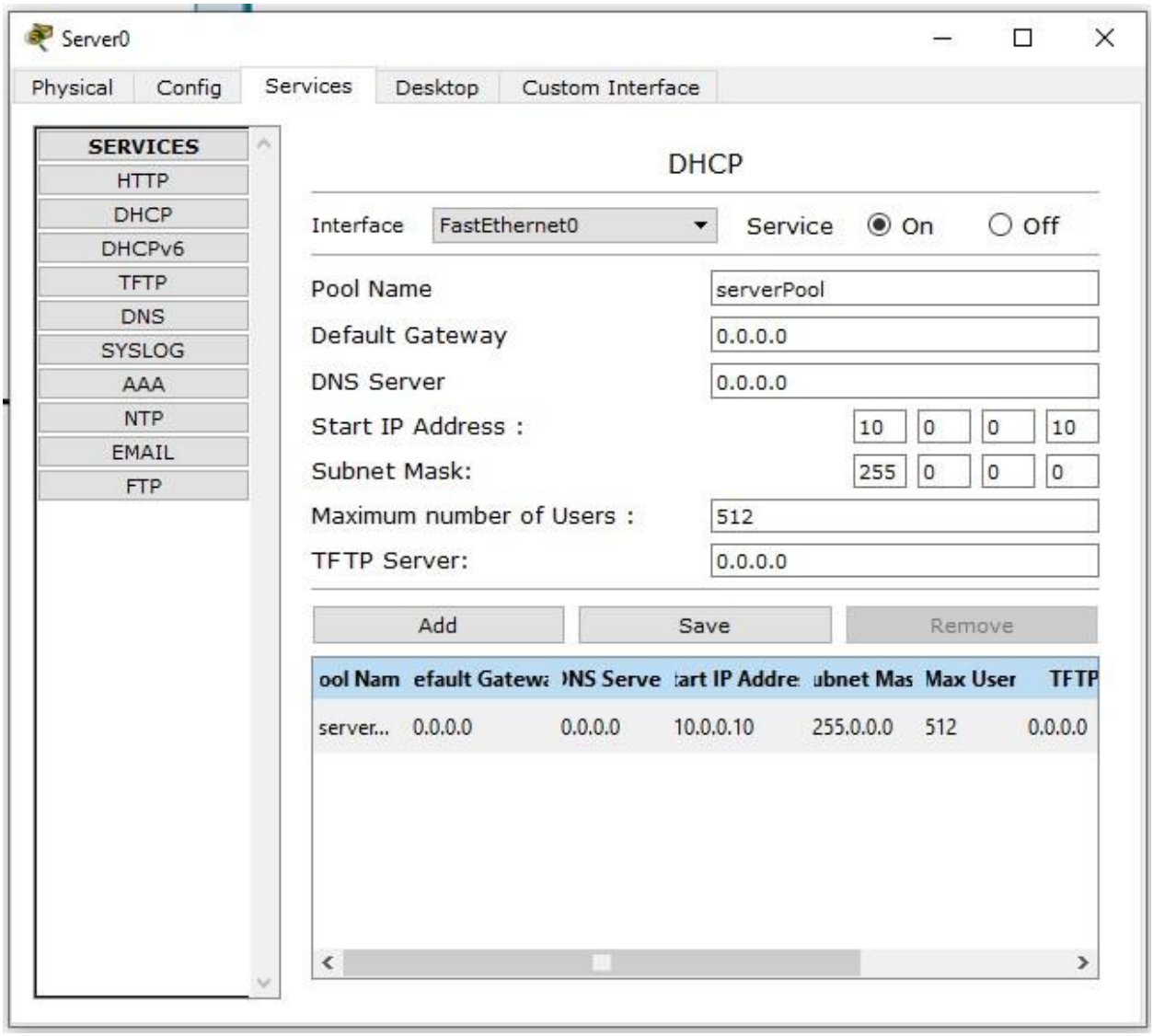
Configure DHCP within a LAN and outside LAN.

a)Within LAN

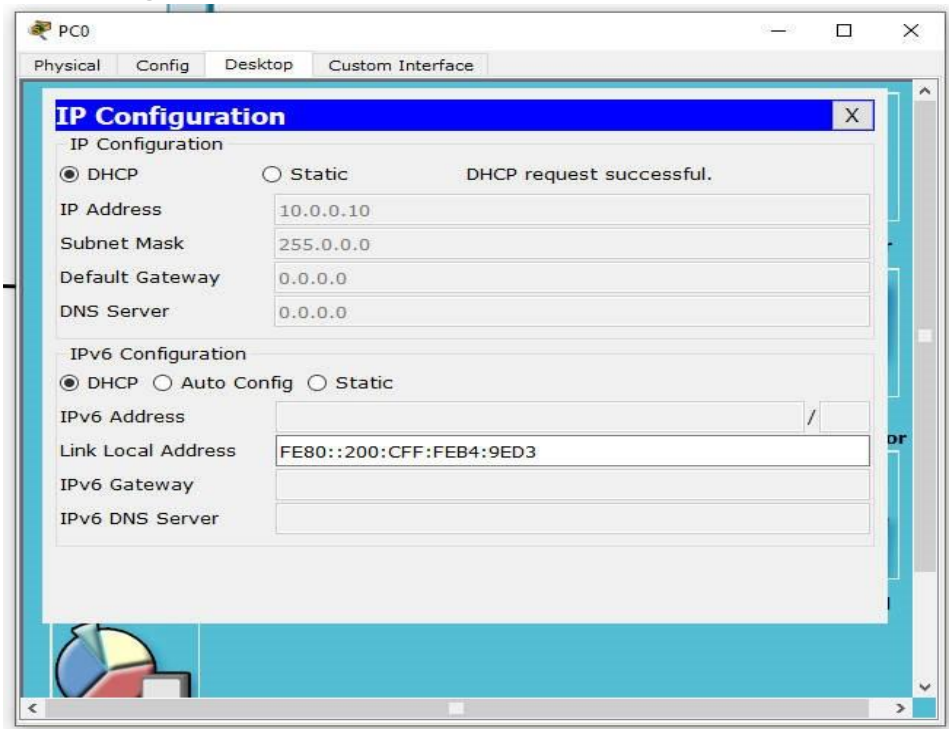
Topology:

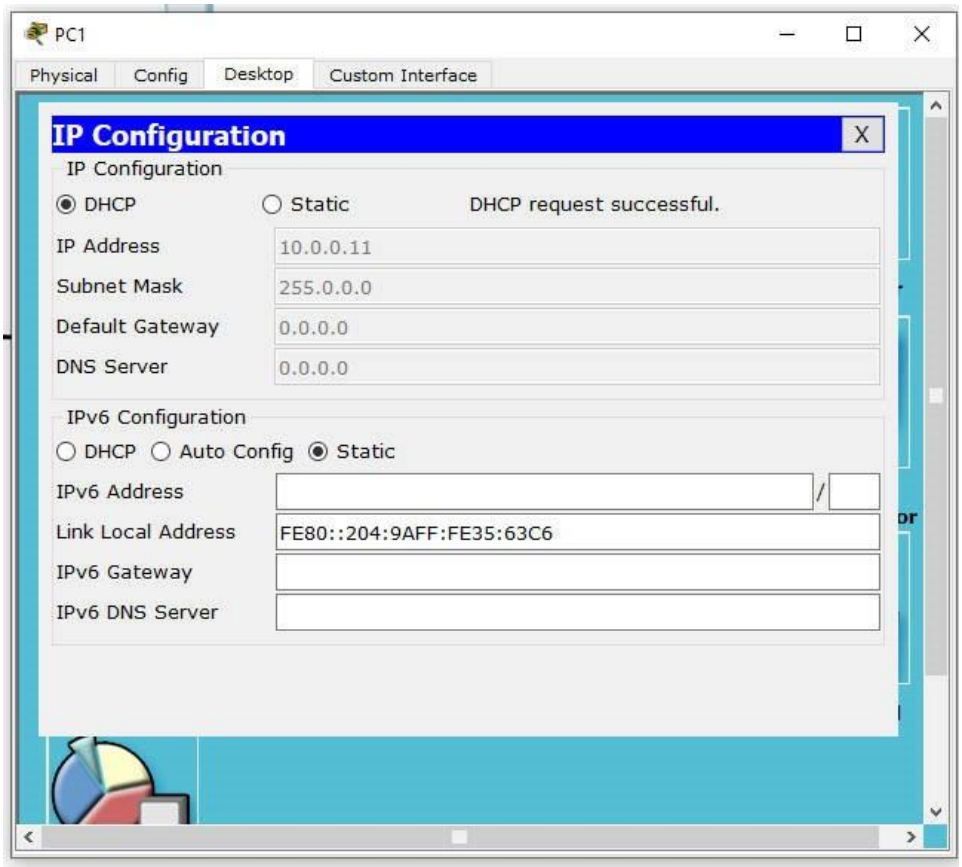


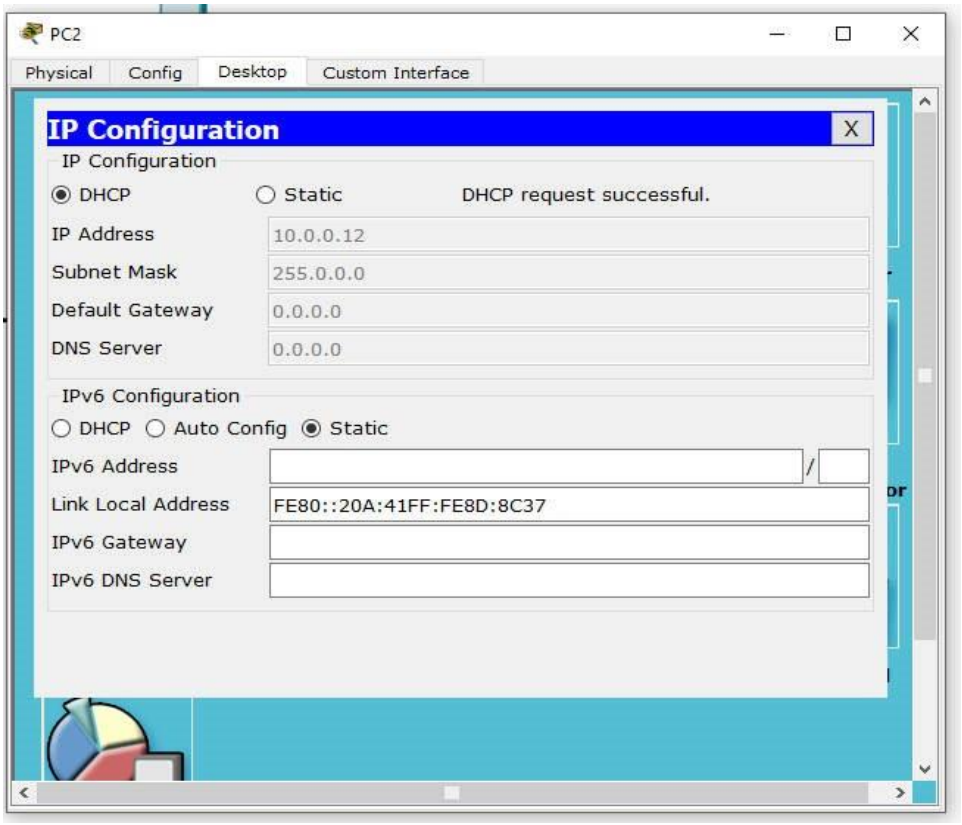
Server:


Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
serverPool
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
Start IP Address :
10 0 0 10
Subnet Mask:
255 0 0 0
Maximum number of Users :
512
TFTP Server:
0.0.0.0
Add
Save
Remove
ool Nam
efault Gatewa
NS Serve
tart IP Addre
ubnet Mas
Max User
TFTP
server...
0.0.0.0
0.0.0.0
10.0.0.10
255.0.0.0
512
0.0.0.0

Obtaining IP:



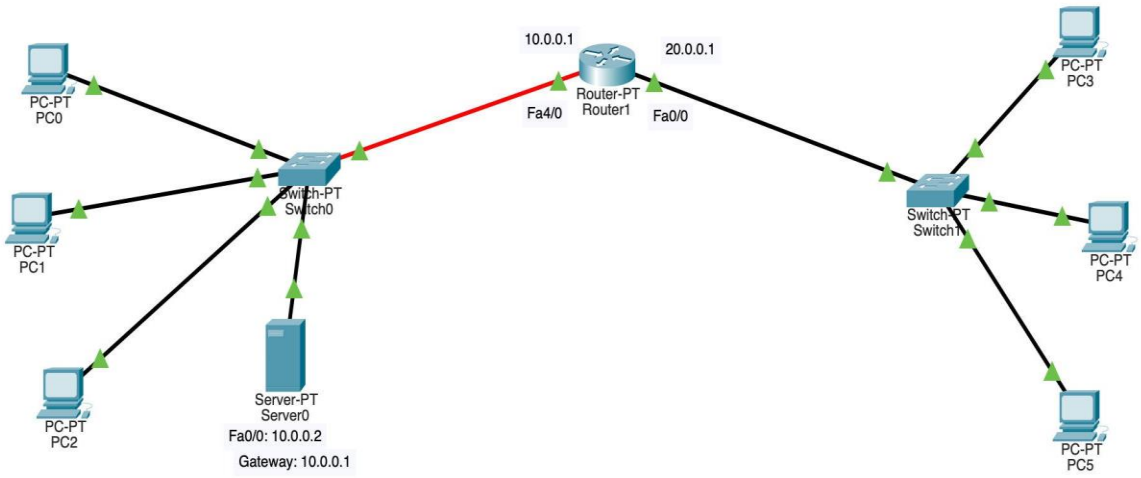



PC2
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
10.0.0.12
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::20A:41FF:FE8D:8C37
IPv6 Gateway
IPv6 DNS Server

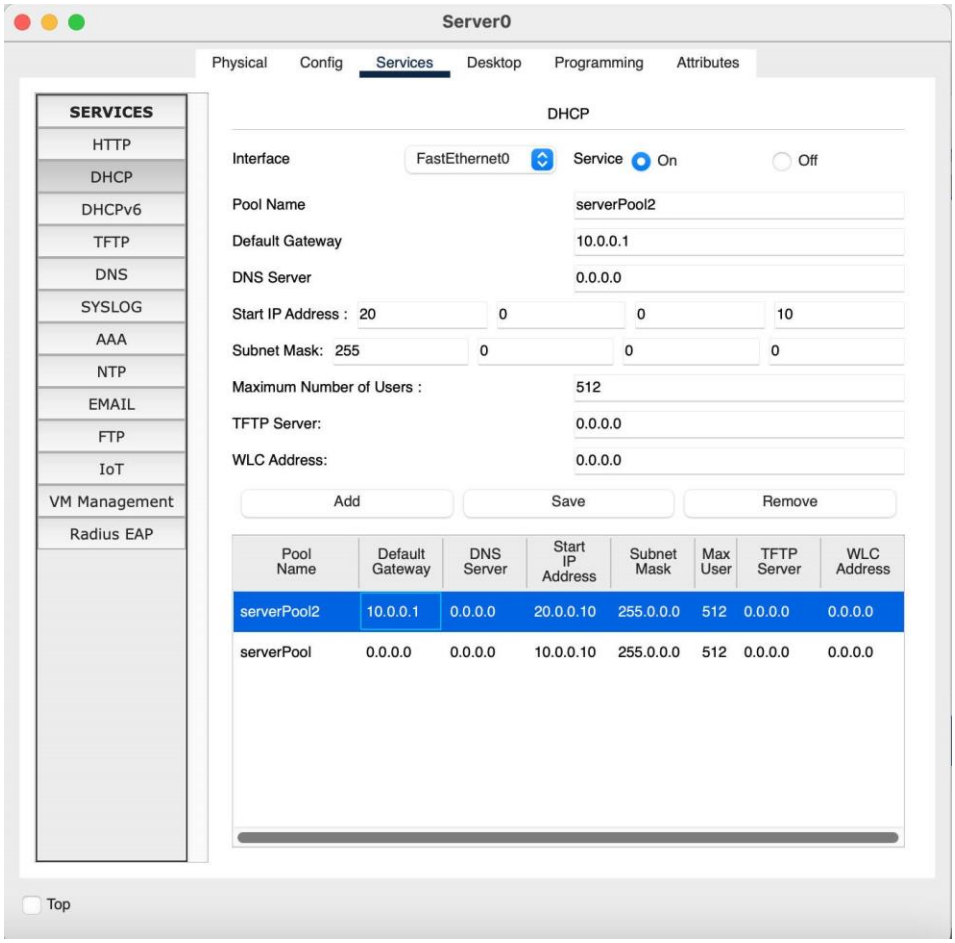
b)Outside LAN

I. Single

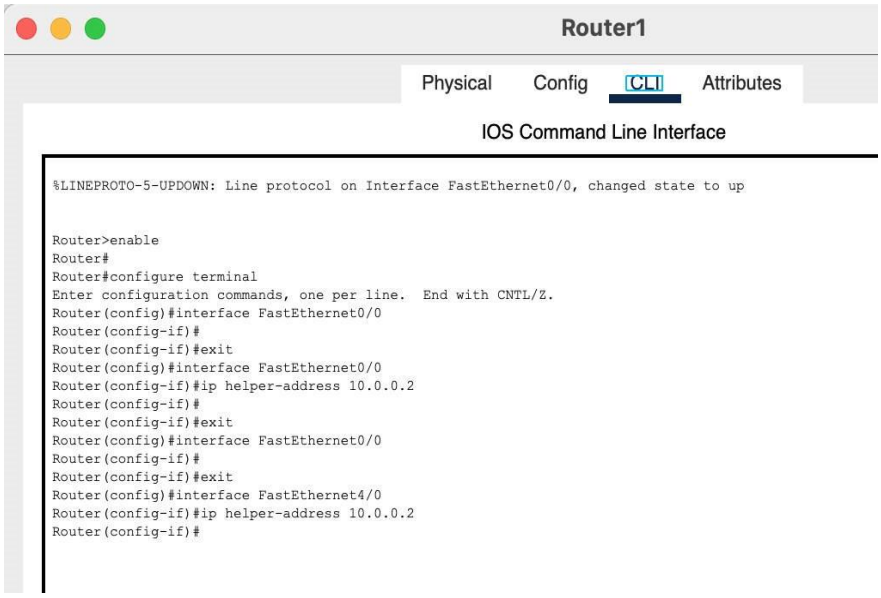
RouterTopology:



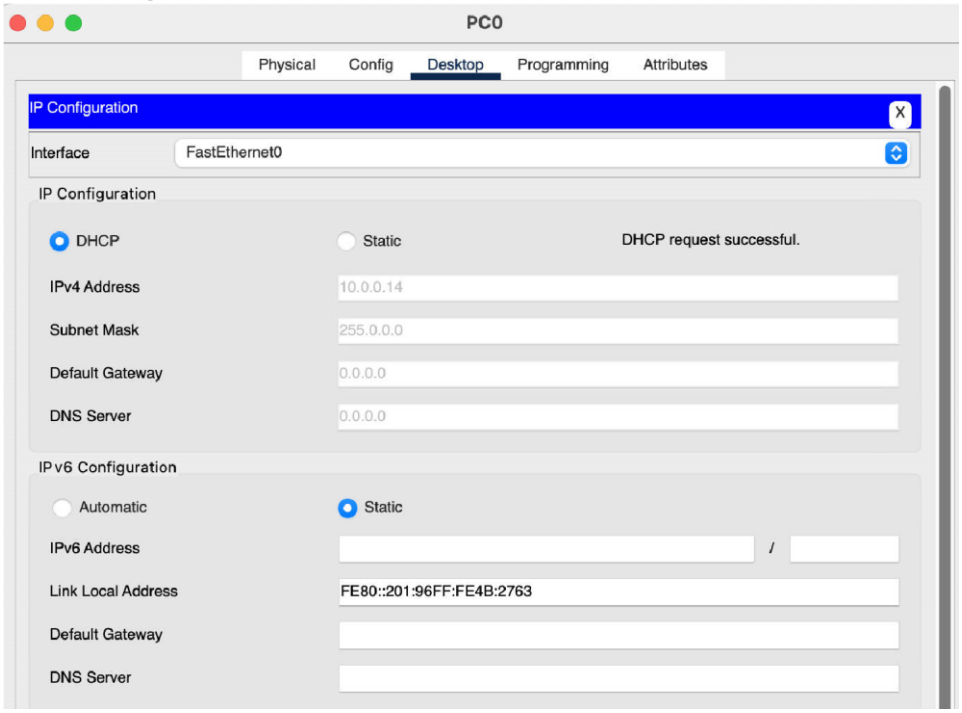
Server:

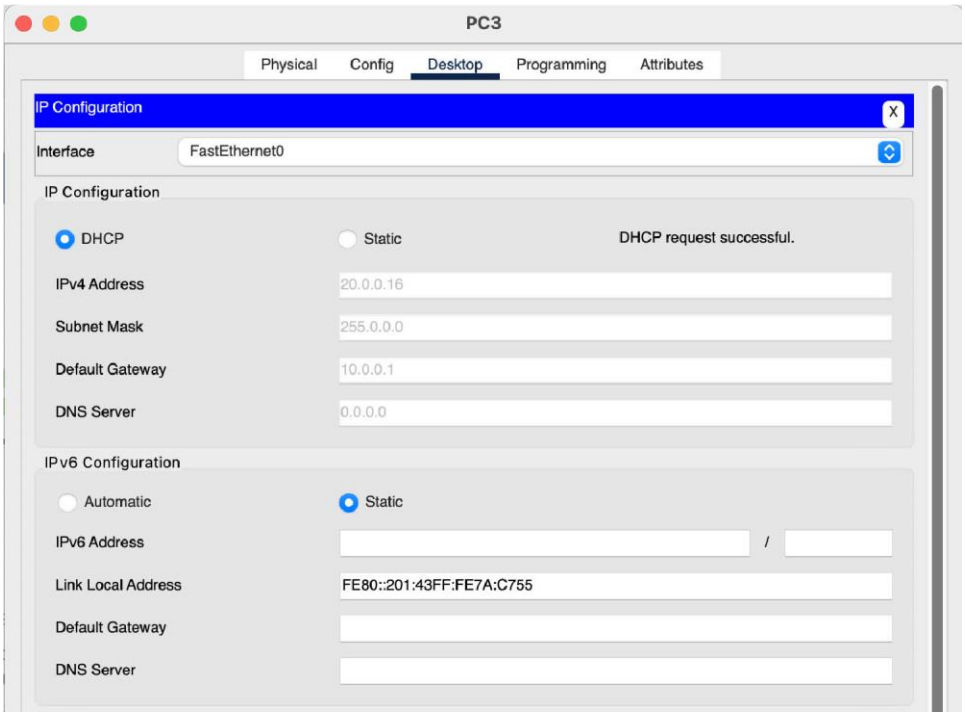


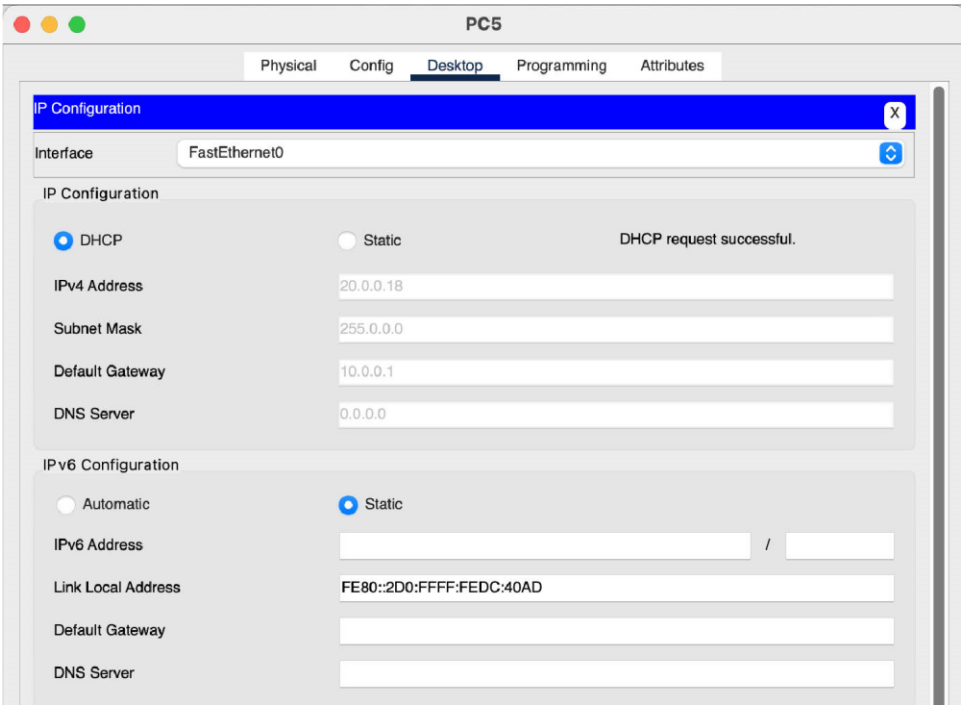
Router:



Obtaining IP:

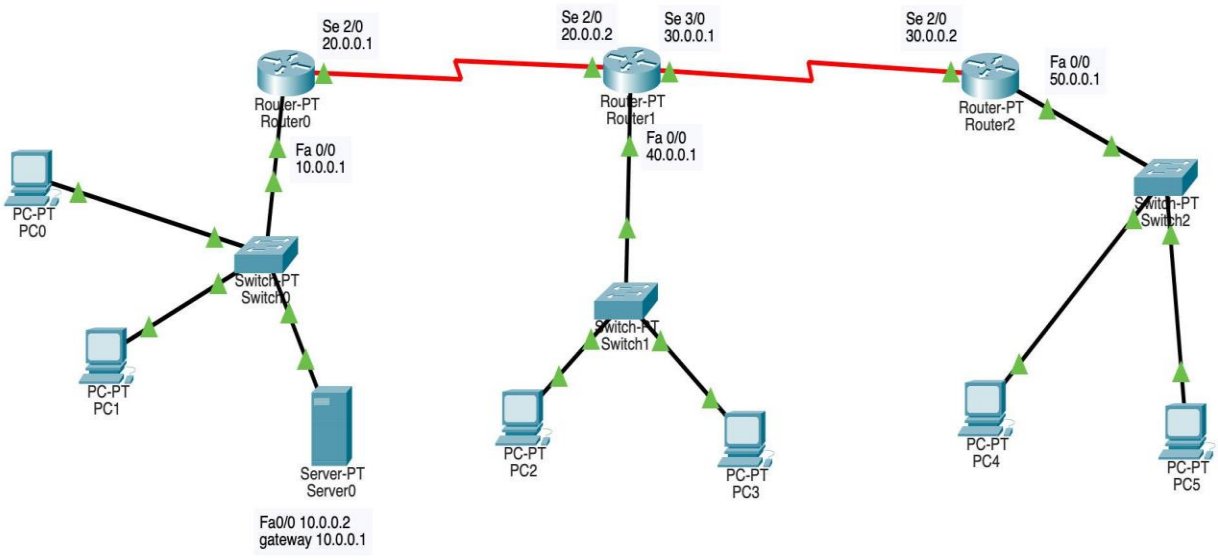




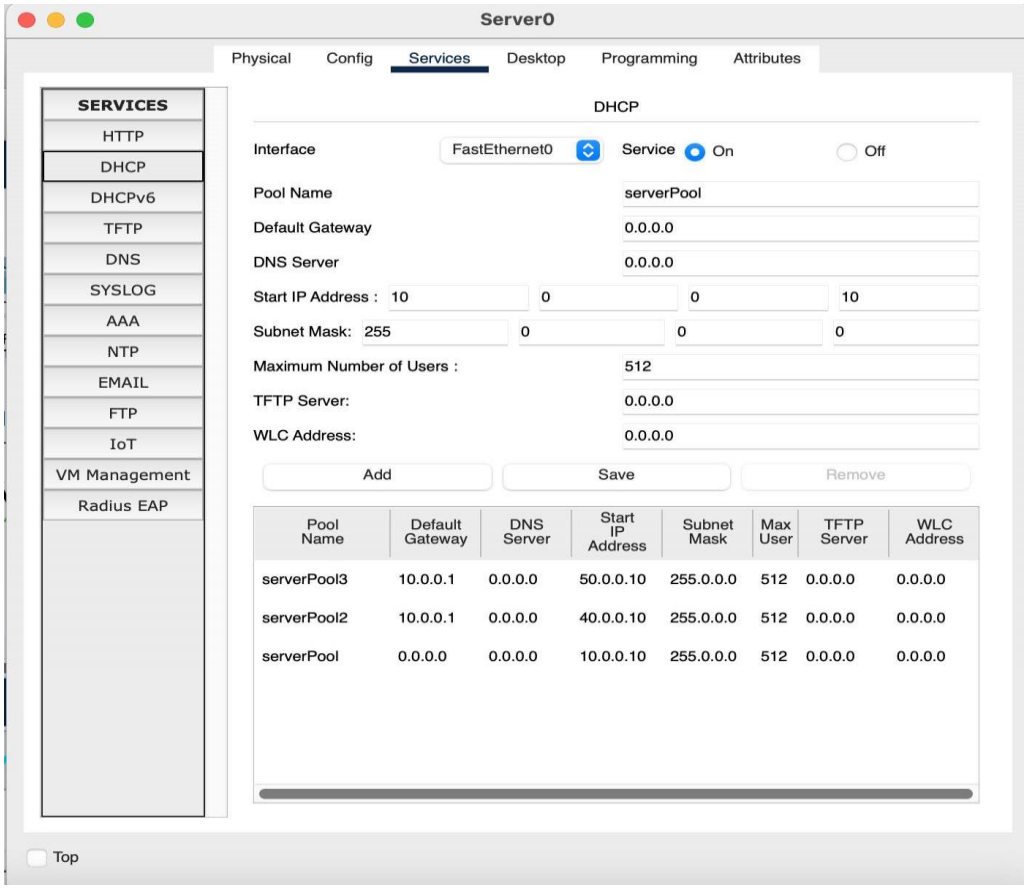


II. **Multiple**

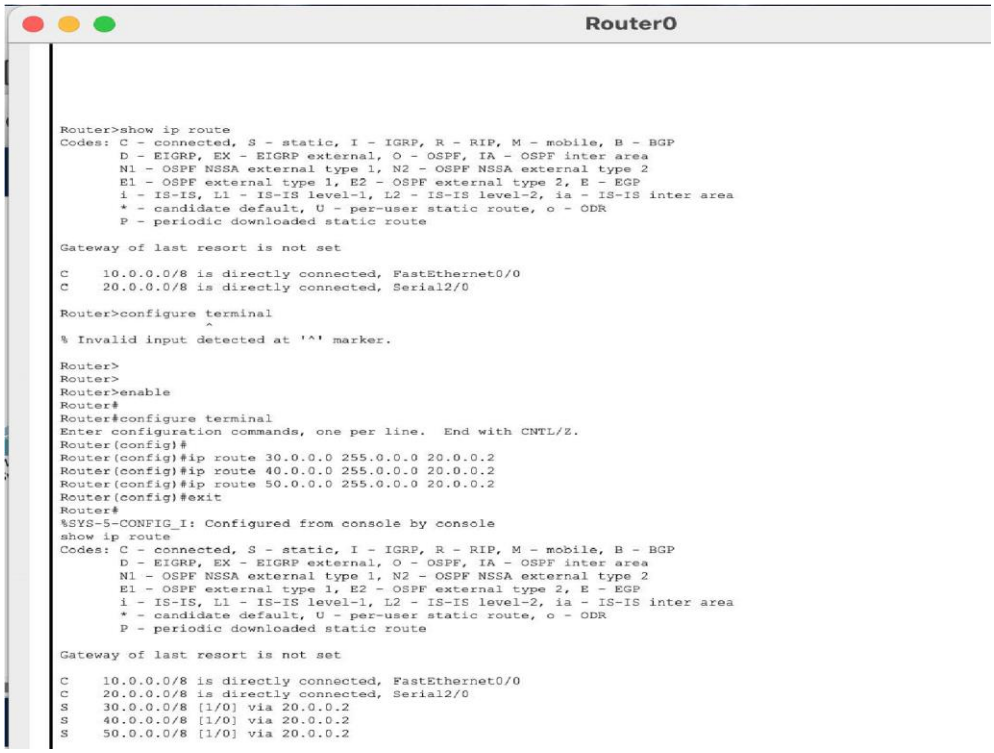
RoutersTopology:

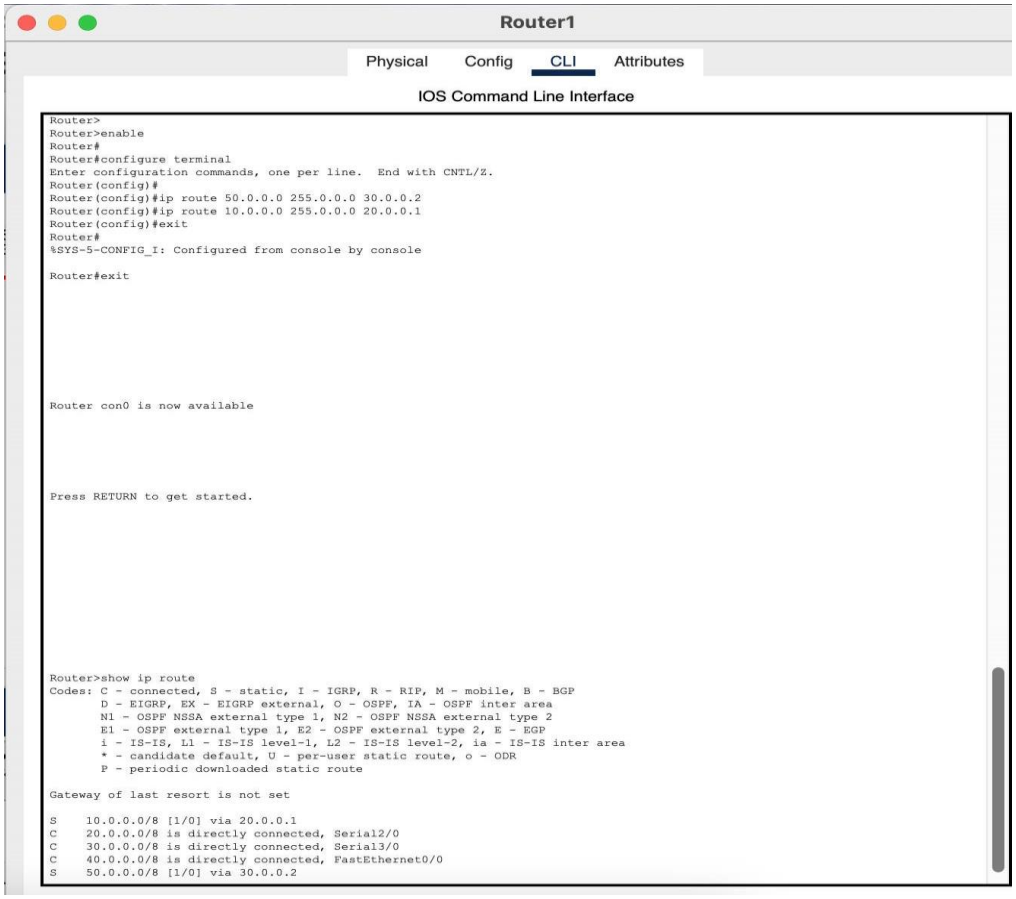


Server:



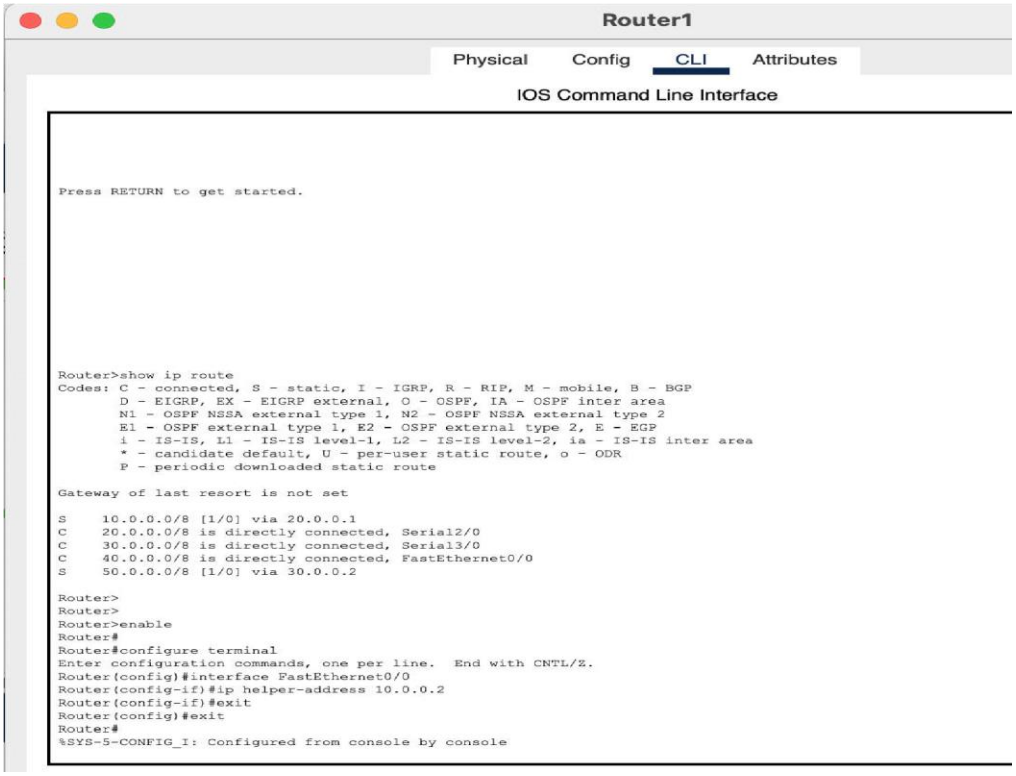
Router: setting IP route

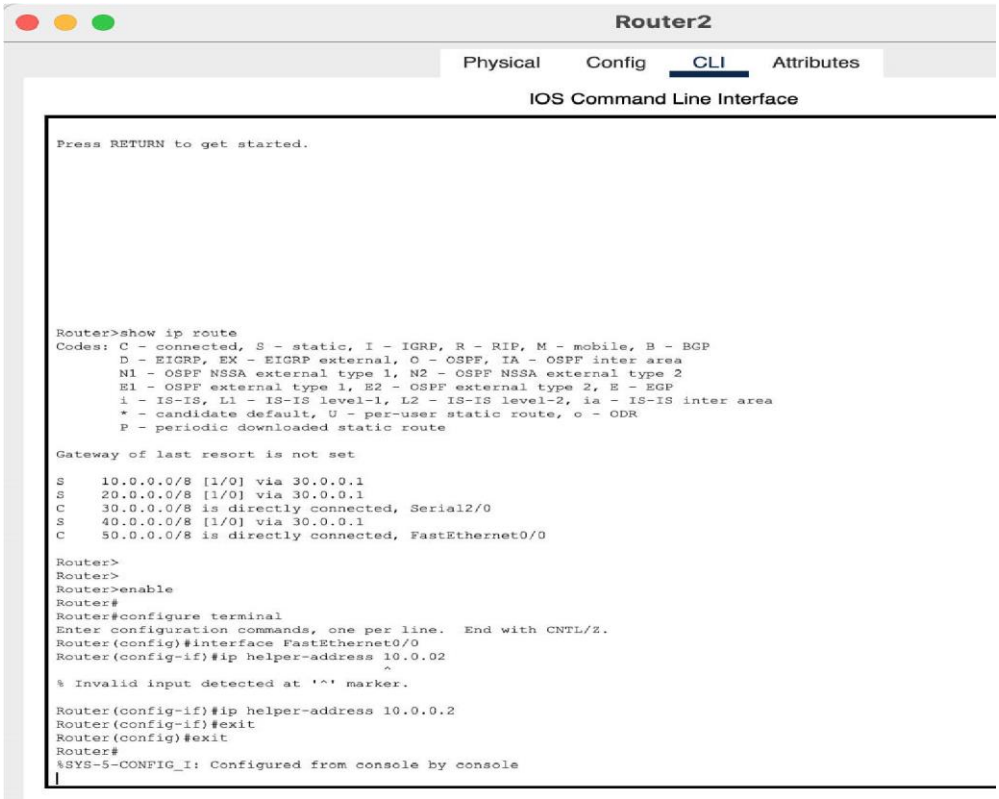



Router1
Physical
Config
CLI
Attributes
IOS Command Line Interface
Router>
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#ip route 50.0.0.0 255.0.0.0 30.0.0.2
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#exit
Router con0 is now available
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
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S 50.0.0.0/8 [1/0] via 30.0.0.2

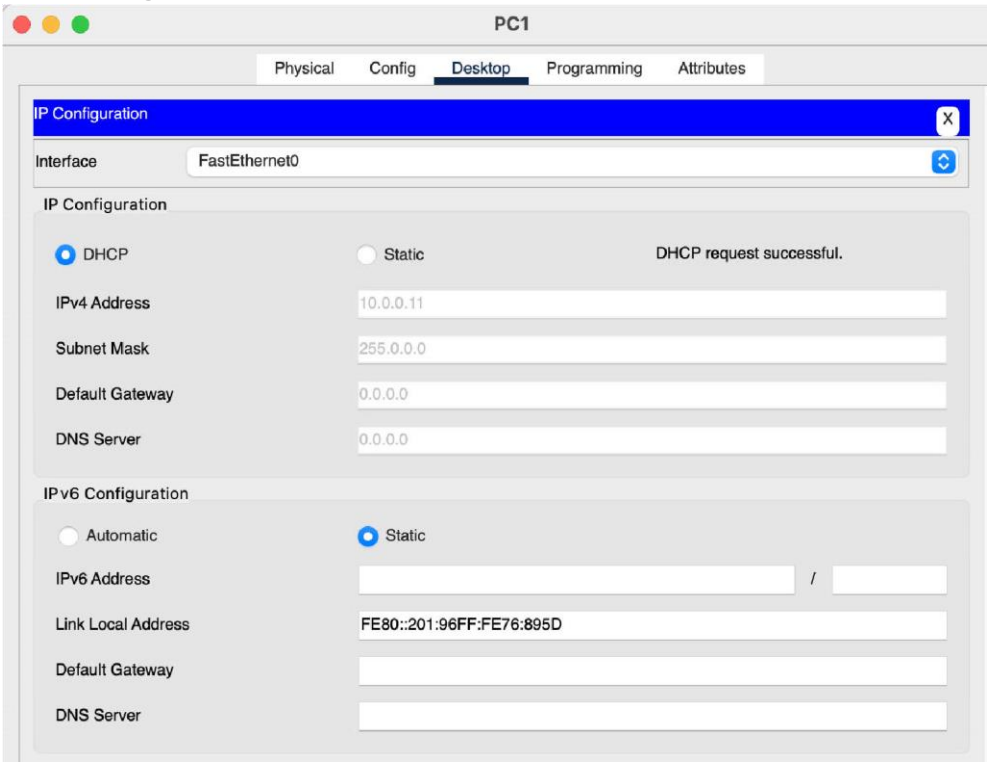

Router2
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
exit
Router con0 is now available
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 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
S 40.0.0.0/8 [1/0] via 30.0.0.1
C 50.0.0.0/8 is directly connected, FastEthernet0/0

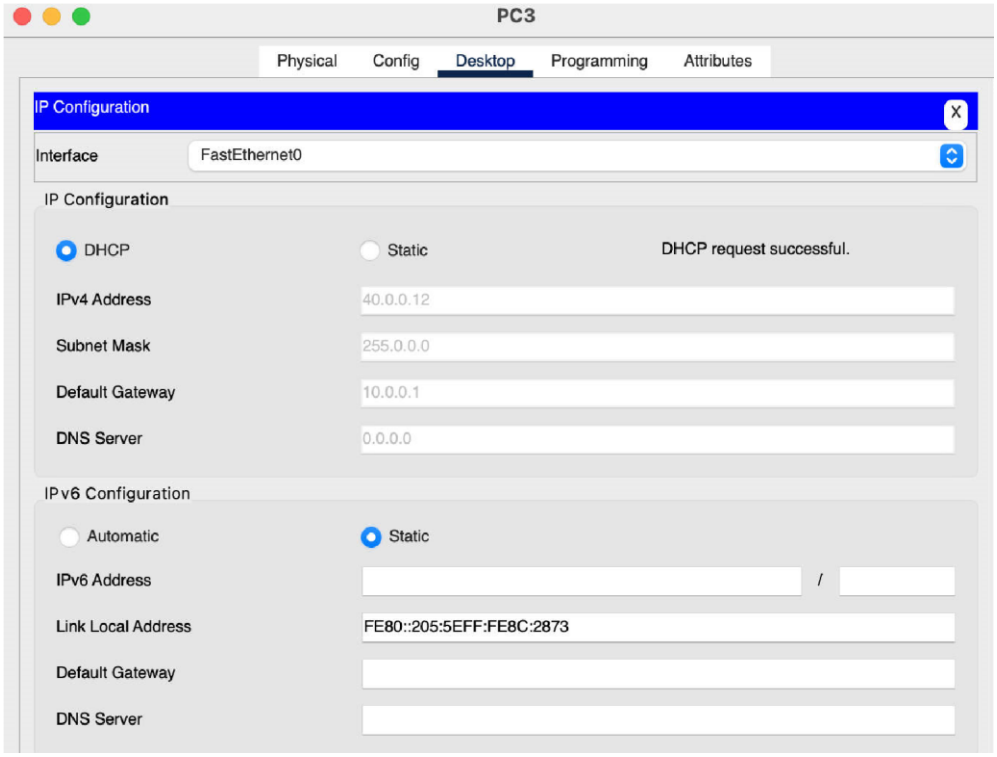
Setting IP helper address-

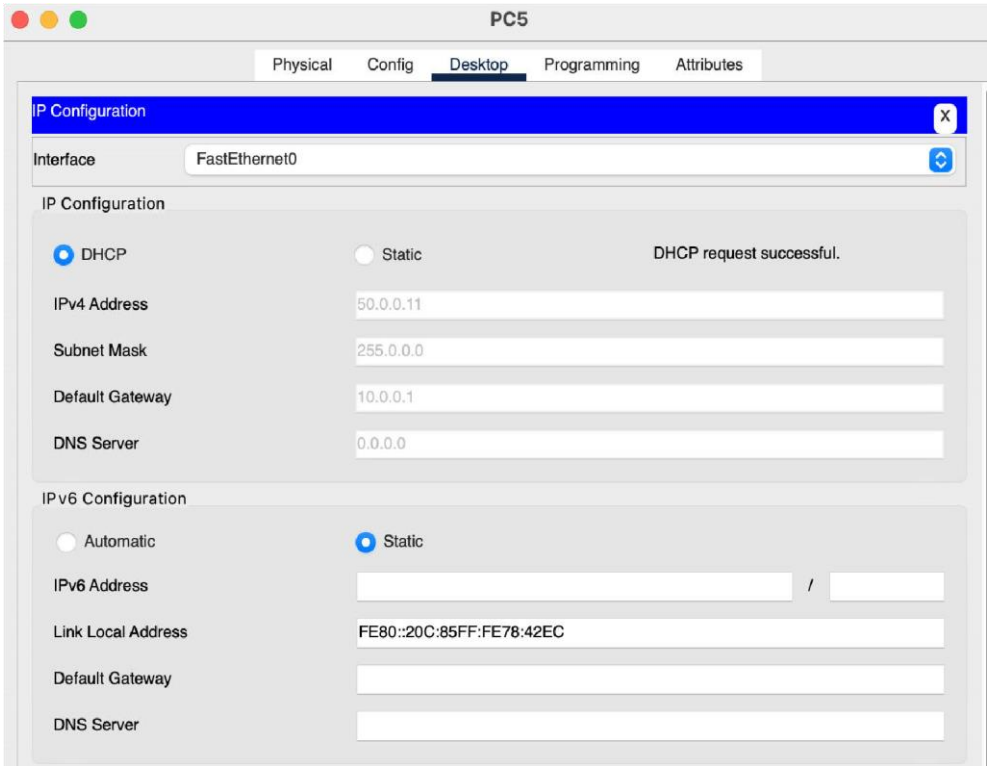




Obtaining IP:




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

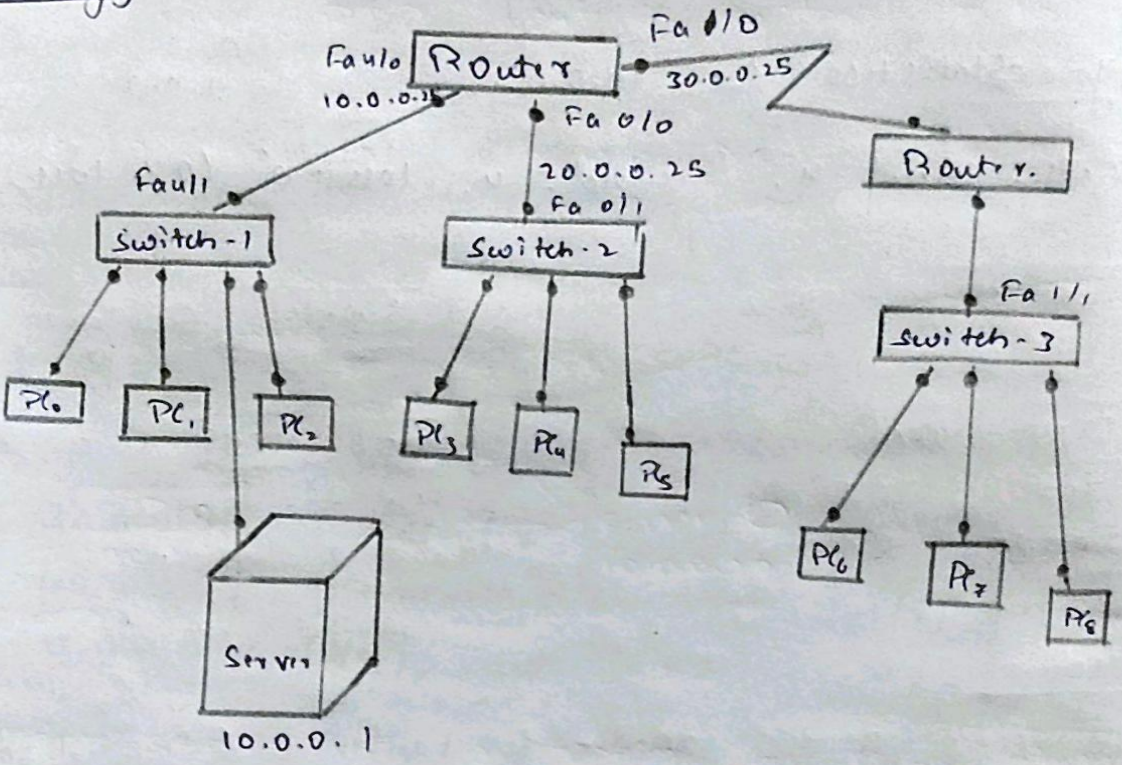
Lab-5

14-7-23

Configure DHCP within a LAN and WAN.

Aim: Dynamic host configuration protolol.

Topology:



Procedure:

- Drag and drop 9 Pc's, 3 switches, 1 server and one router (Connect every 3 Pc's to one switch and connect the server to any switch. Connect all 3 switches to the router.
- Set manually, the IP address of the server say 10.0.0.1 and create a pool with start IP address 10.0.0.2 & save.

- Give IP addresses to each connection of the router say 10.0.0.25, 20.0.0.25, 30.0.0.25.
- Use the command IP helper address to connect Pi's of different connections Ipools to the server
  - IP helper-address 10.0.0.1 (Server address).
- Create 2 other pools by giving new names & new start IP address each having the same gateway i.e., 10.0.0.25 and add to the server.
- Go to IP config tab of any PC and change from static to DHCP to allocate IP address automatically using DORA protocol.

OBSERVATION:

IP addresses are set automatically using DORA protocol by the server (discover, offer, request, acknowledge).

- > interface Serial 2/0 (For router 0)

  ip route 40.0.0.0 255.0.0.0 35.0.0.25

  > interface Serial2/0 (For router 1)

  ip route 10.0.0.0 255.0.0.0 30.0.0.26

  ip route 20.0.0.0 255.0.0.0 80.0.0.26

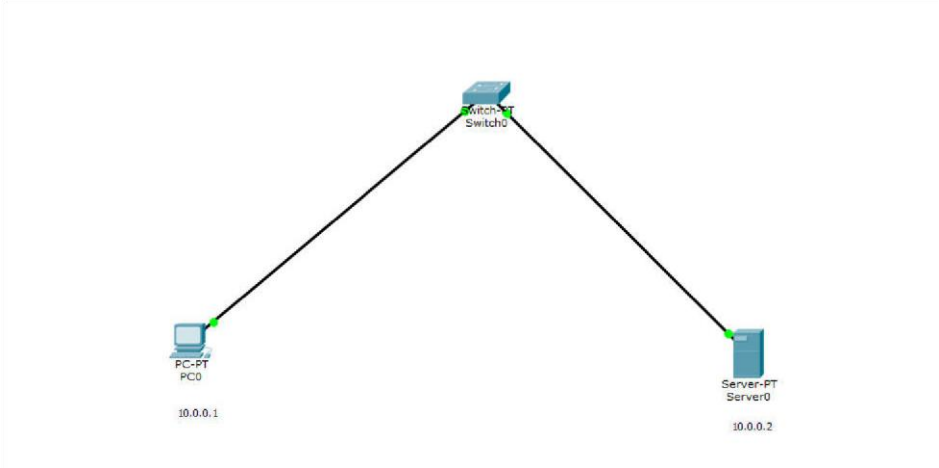
  (Static configuration for routers).

Experiment No. 5

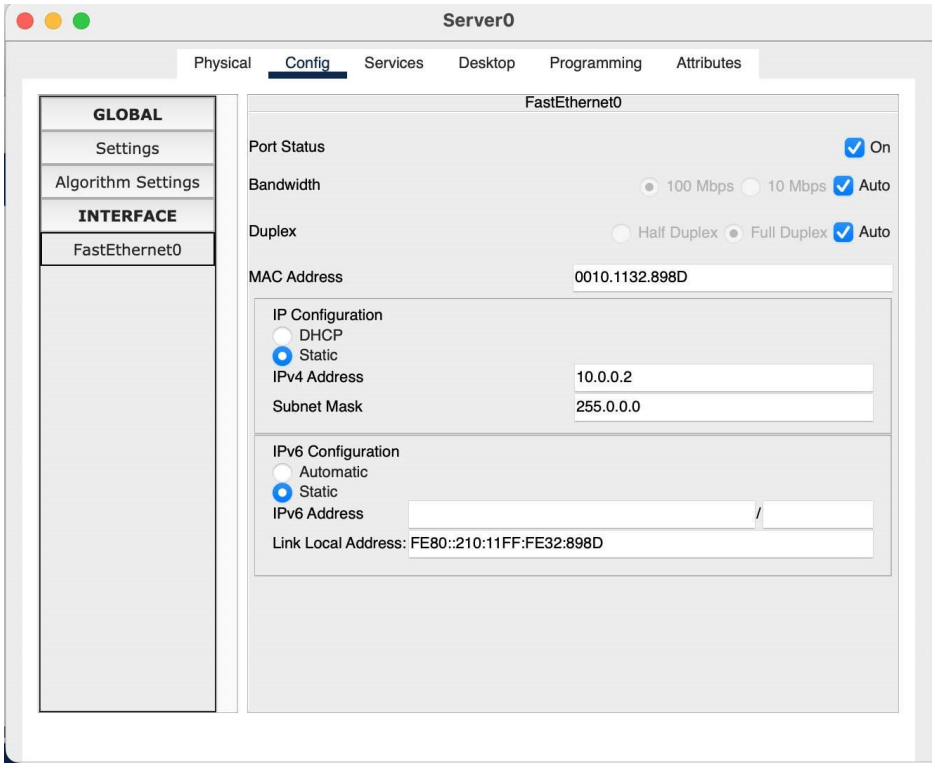
Title:

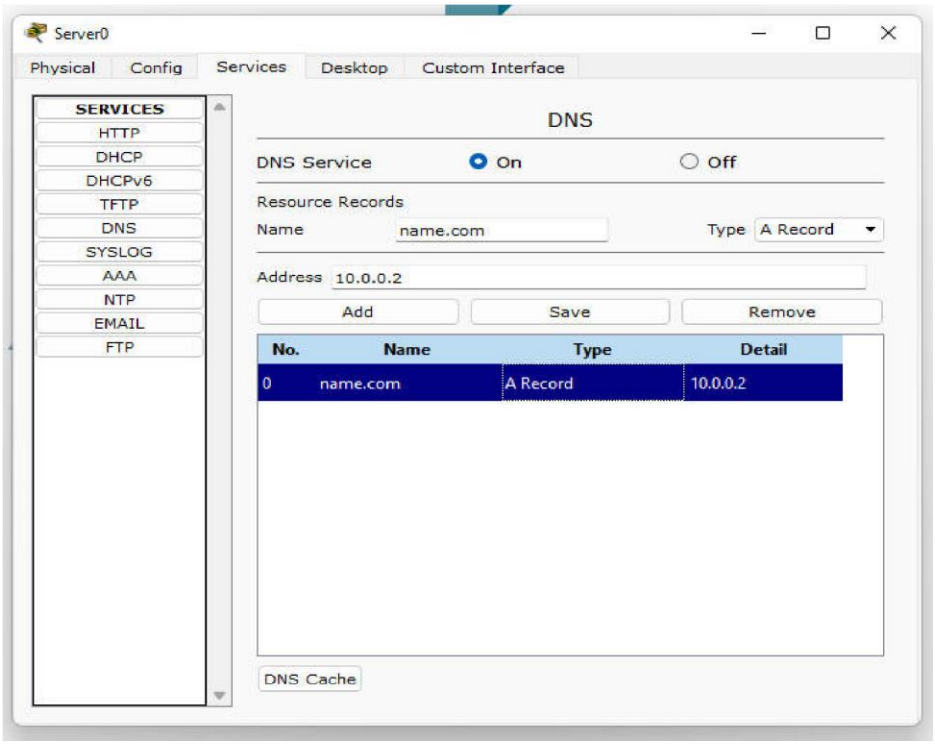
Configure Web Server, DNS within a LAN.

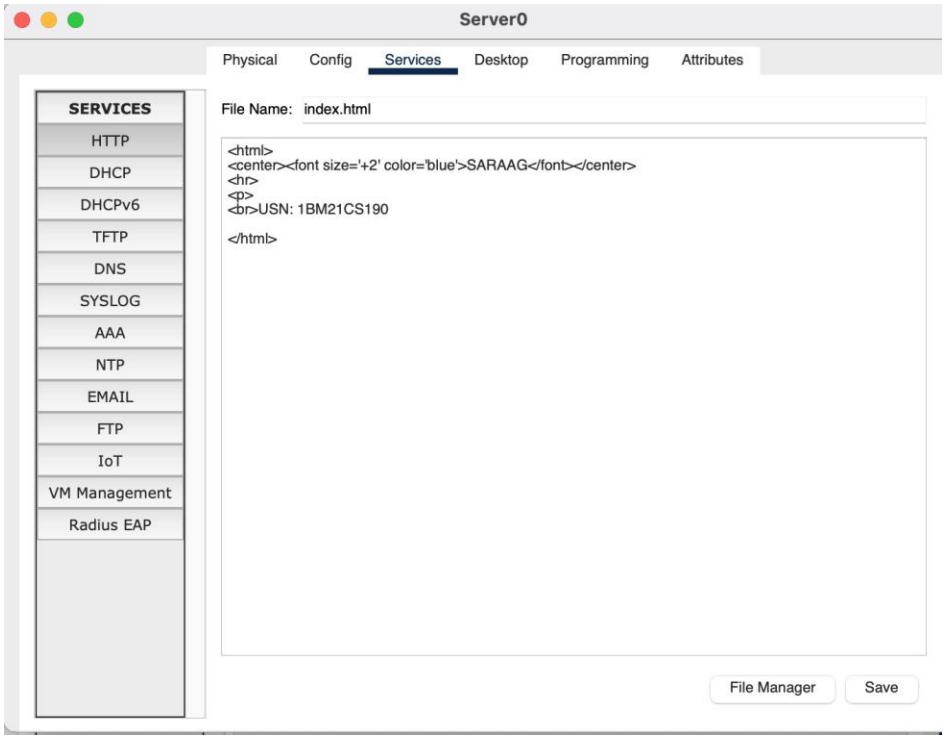
Topology:



Server:

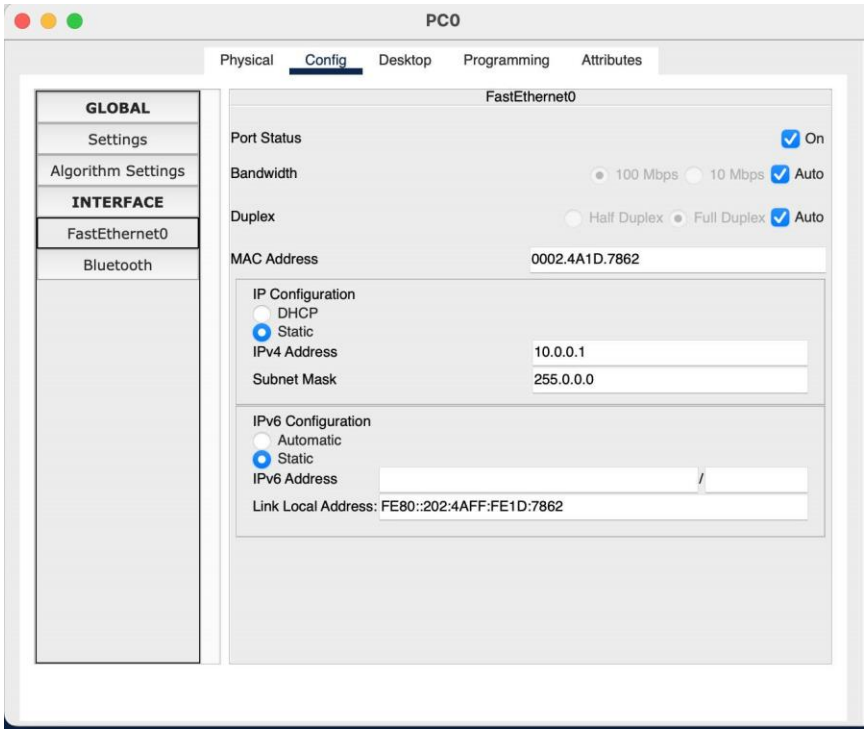



Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
name.com
Type
A Record
Address
10.0.0.2
Add
Save
Remove
No.
Name
Type
Detail
0
name.com
A Record
10.0.0.2
DNS Cache


Server0
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
File Name:
index.html
<html>
SARAAG
<hr>
<p>

USN: 1BM21CS190
</html>
File Manager
Save

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

Week-6 21-7-23

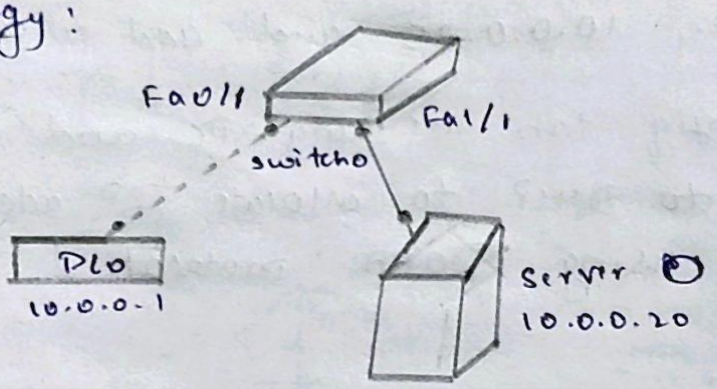
~~DNS~~ - ~~Domain~~ Name ~~System~~.

- ~~Create~~ ~~topology~~

1. Configure web server, DNS within a LAN.
2. Configure RIP routing protocol in routers.

Aim: To configure webserver, DNS within a LAN.

Topology:



Procedure: Step 1: Create a topology as shown above using a PC, Server & switch.

Step 2: ~~Step~~ Set the IP addresses. as 10.0.0.1 and 10.0.0.20 for PC and server respectively.

Step 3: In the server, under DNS service create new example.com website with add 10.0.0.20 and add under HTTP, modify the index.html file and add name and USN as

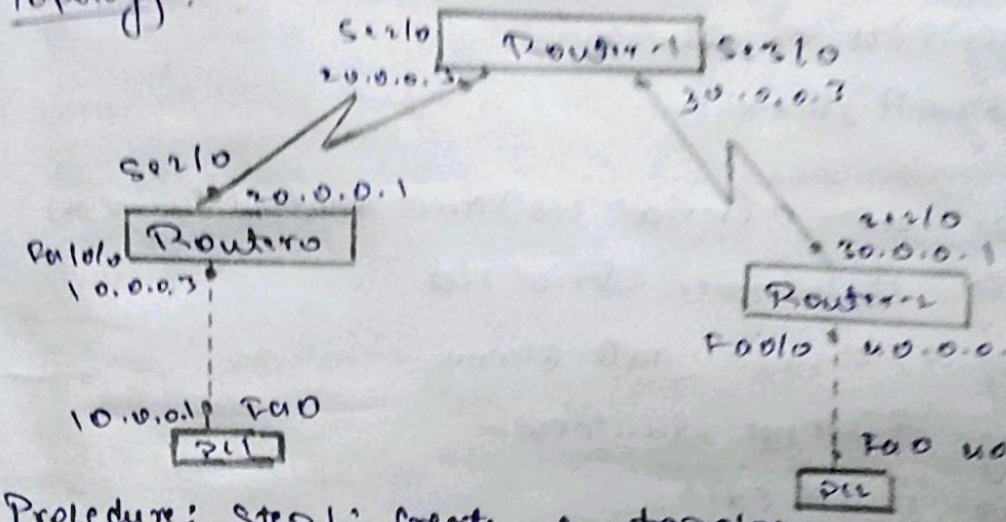
<h1> Saikumar </h1>
<h1> 1BM21CS182 </h1>

Step 4: In PC0, go to desktop → web browser and type example.com you'll be able

to see the website with domain name by using
url http://example.com
cisco packet tracer
Saikumar
192M1A05120.

① Aim: To configure RIP routing protocol in routers

Topology:



Procedure: Step 1: Create a topology as shown above using 2 PC's and 3 routers.

Step 2: Configure the IP addresses of 2 PC's as 10.0.0.1 and 40.0.0.1 for PC0 and PC1 respectively and set the gateway as 10.0.0.2 & 40.0.0.2

Step 3: Place the ip's to configure the routers for router 0.

Router > enable
Router # config terminal.
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.2 255
Router (config-if) # No shutdown.

router (config)# ip interface Serial 2/0
router (config-if)# ip address 20.0.0.1 255.0.0.0
Router (config-if)# NO shutdown

similarly, configure the ports of router 1 & router 2.

Step 4: For router 0.

router (config)# interface Serial 2/0.
router (config-if)# encapsulation PPP
router (config-if)# No shutdown
router (config-if)# exit.

Step 5: For router 0 (Serial 2/0) and router 1 (Serial 3/0).

router (config)# interface Serial 2/0
router (config-if)# clock rate 64000
router (config-if)# no shutdown
router (config-if)# exit.

Step 6: For all routers, repeat this step.

Ex: For router 0

router> enable.
router# config terminal.
router (config)# router rip
router (config-router)# network 10.0.0.0
router (config-router)# network 20.0.0.0

Similarly, do this for router 1 and router 2
then, router# show ip route.

This will result in saying that every router knows all the 4 networks in the topology. Now, you can ping from PC1 to PC2.

Result:

In command prompt of PC1

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 = 12ms TTL = 125

Reply from 40.0.0.1 : bytes = 32 time = 6ms TTL = 125

Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.1 : bytes = 32 time = 6ms TTL = 125

Ping statistics for 40.0.0.1

Packets : Sent = 4, Received = 4, Lost = 0 (0% Loss).

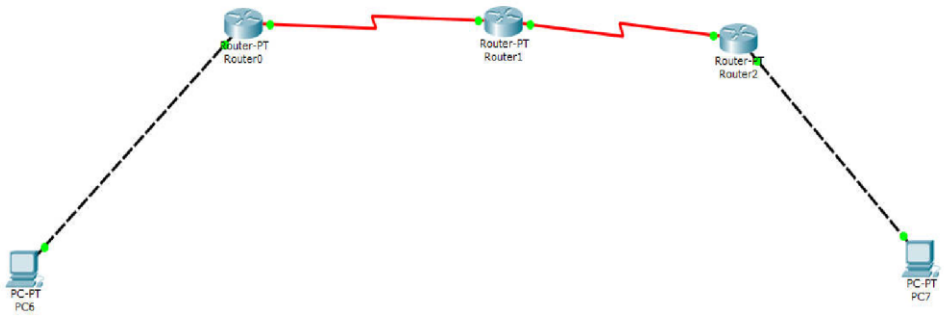
Approximate round trip times in milli seconds

Min = 2ms, max = 12ms, avg = 6ms.

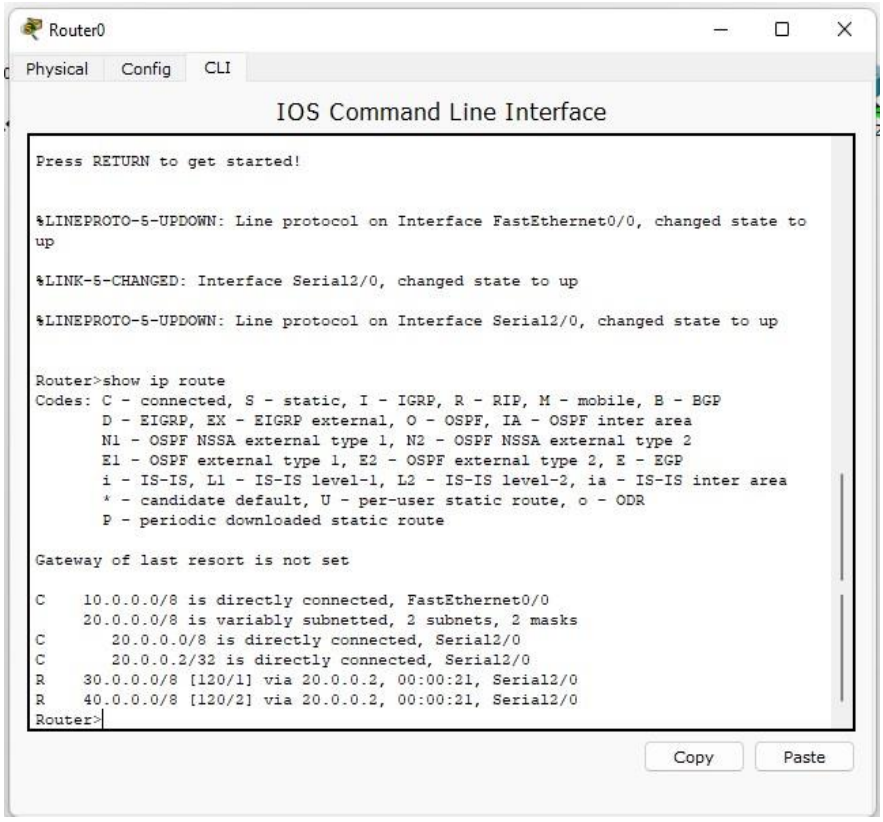
Experiment No. 6

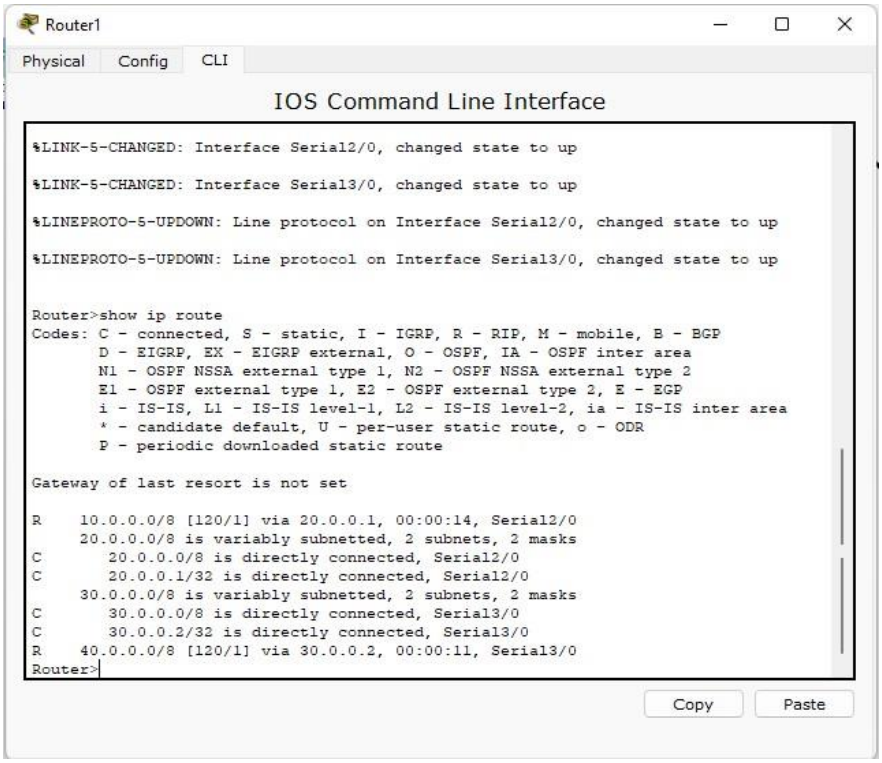
Title:
Configure RIP routing Protocol in Routers

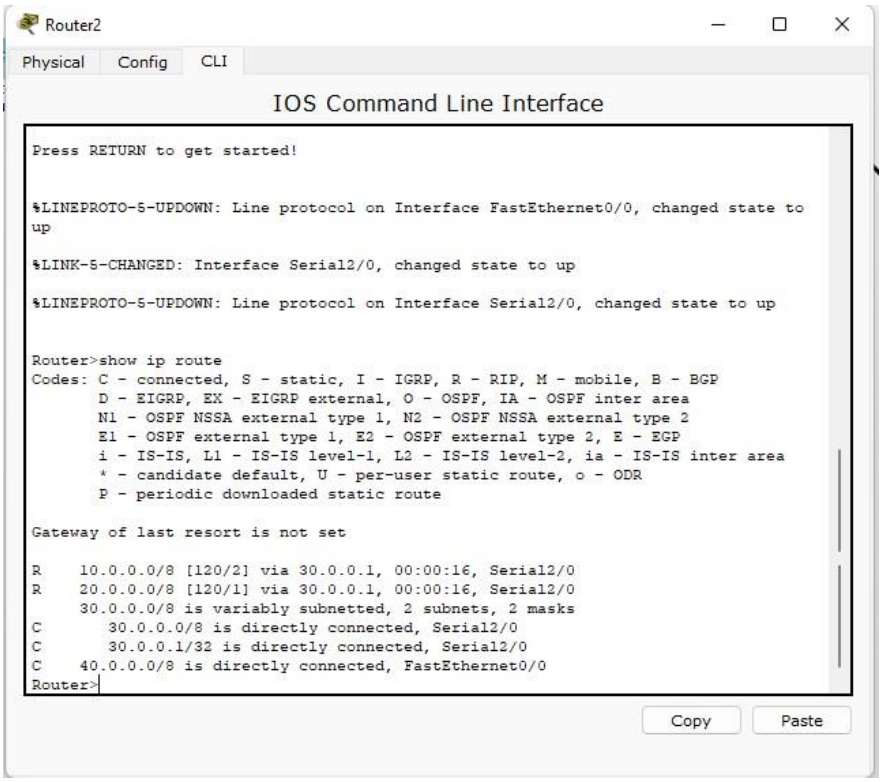
Topology:



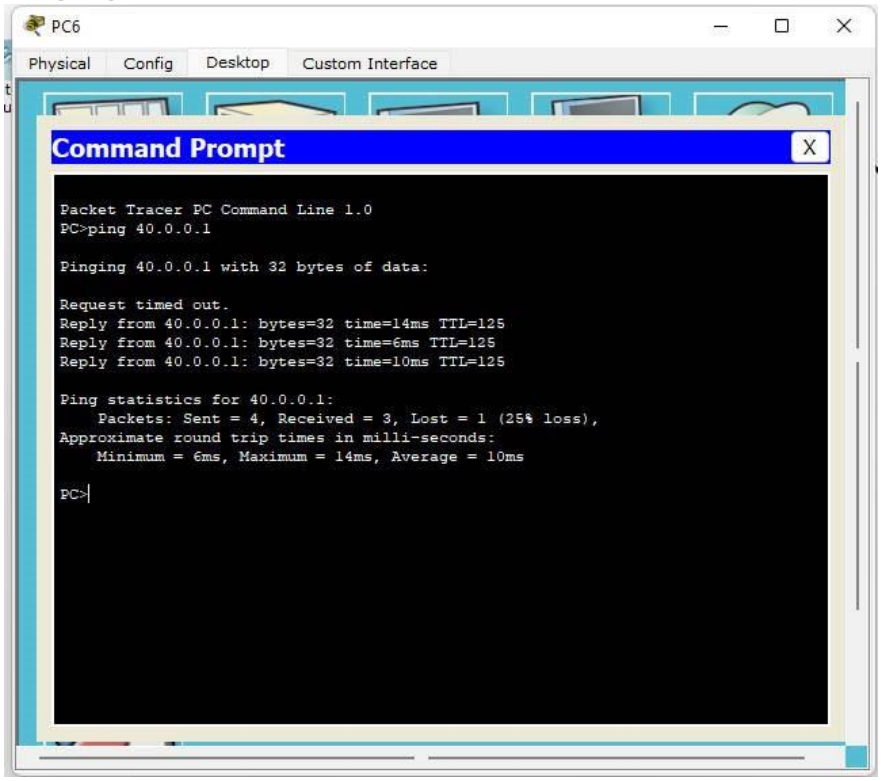
IP Route:


Router0
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
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
C 10.0.0.0/8 is directly connected, FastEthernet0/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.2/32 is directly connected, Serial2/0
R 30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:21, Serial2/0
R 40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:21, Serial2/0
Router>
Copy
Paste

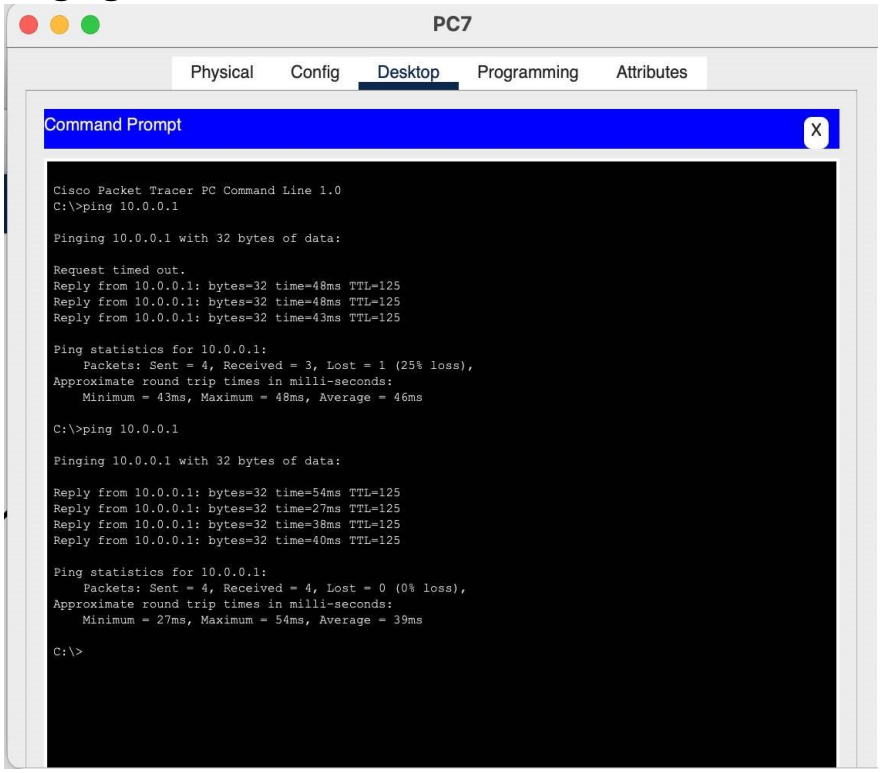

Router1
Physical
Config
CLI
IOS Command Line Interface
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
Router>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
R 10.0.0.0/8 [120/1] via 20.0.0.1, 00:00:14, Serial2/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.1/32 is directly connected, Serial2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial3/0
C 30.0.0.2/32 is directly connected, Serial3/0
R 40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:11, Serial3/0
Router>
Copy
Paste


Router2
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
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
R 10.0.0.0/8 [120/2] via 30.0.0.1, 00:00:16, Serial2/0
R 20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:16, Serial2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.1/32 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
Router>
Copy
Paste

Pinging PC7 from PC6:



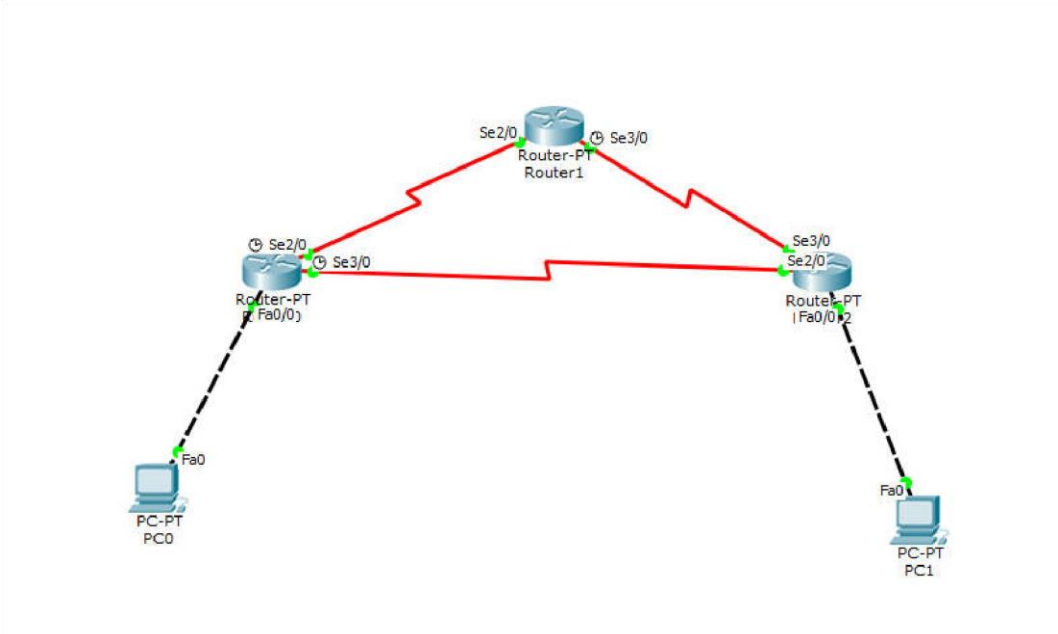
Pinging PC6 from PC7:



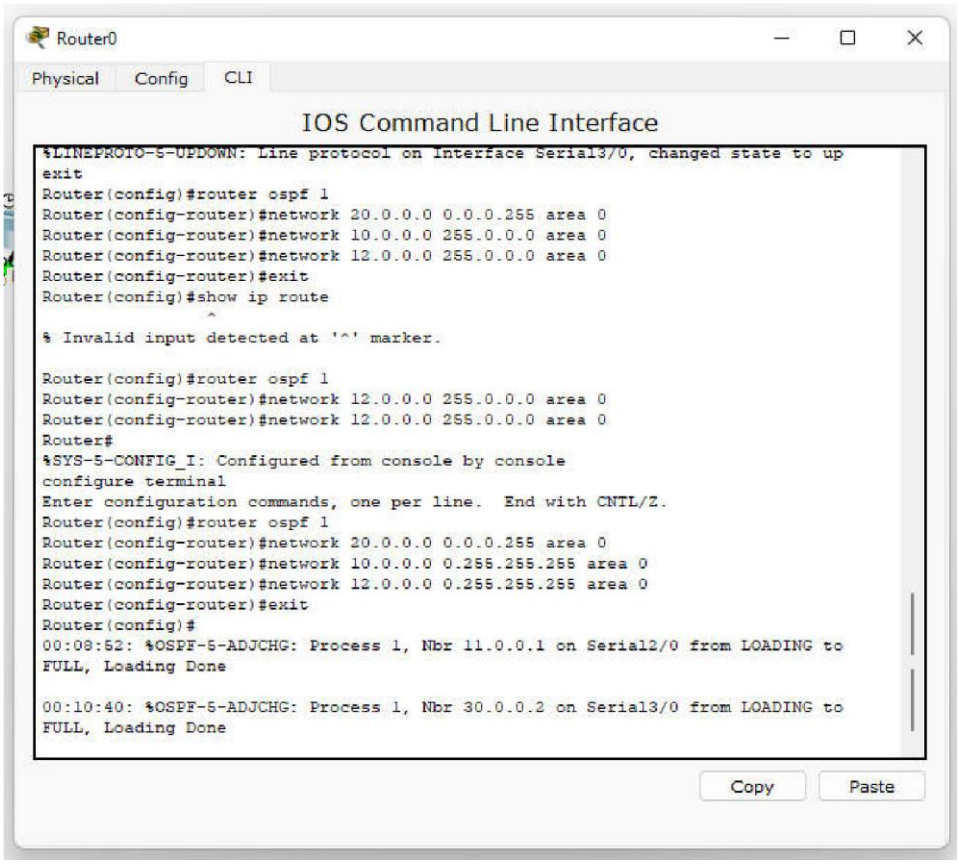
Experiment No. 7

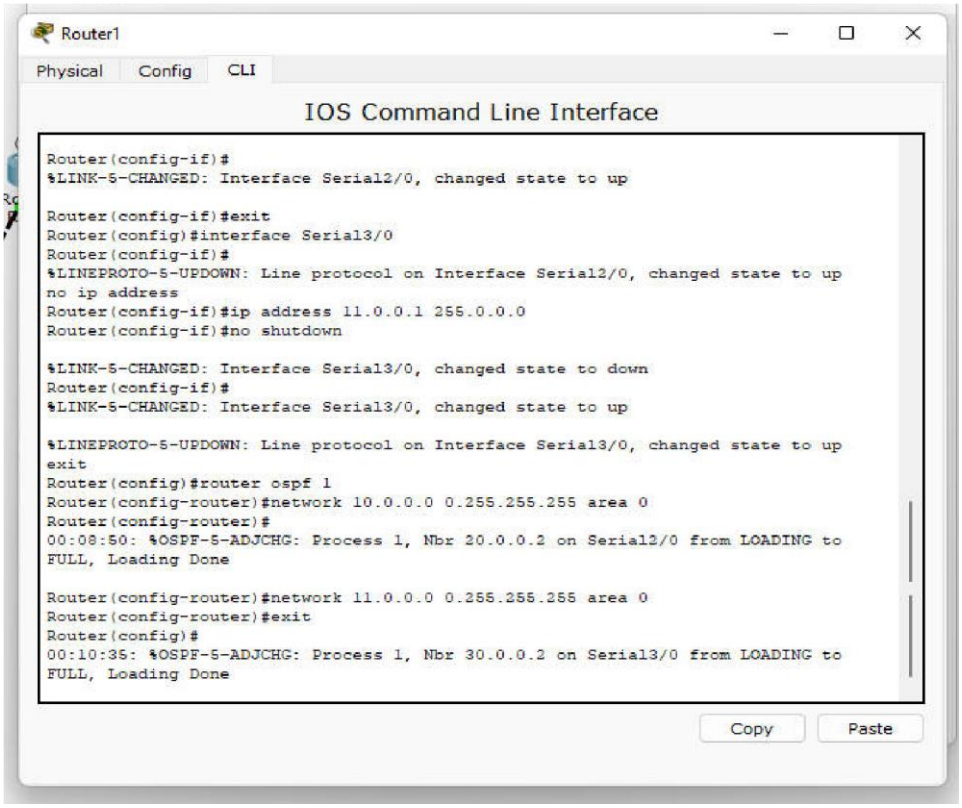
Title:
Configure OSPF routing protocol

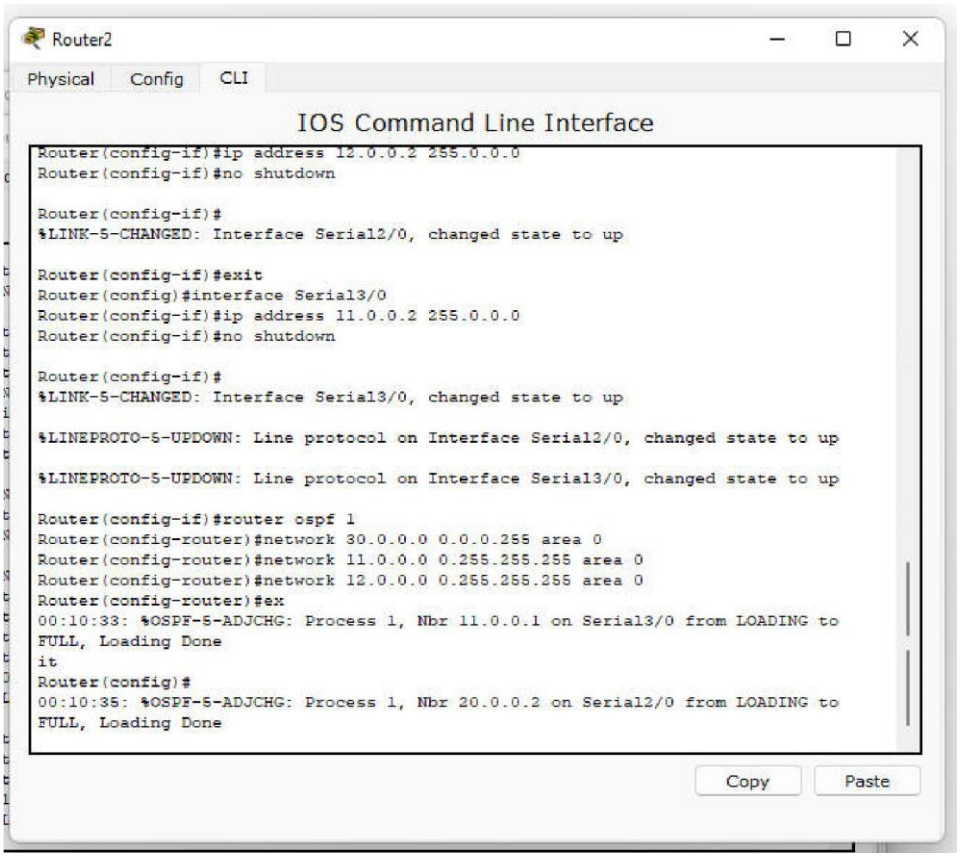
Topology:

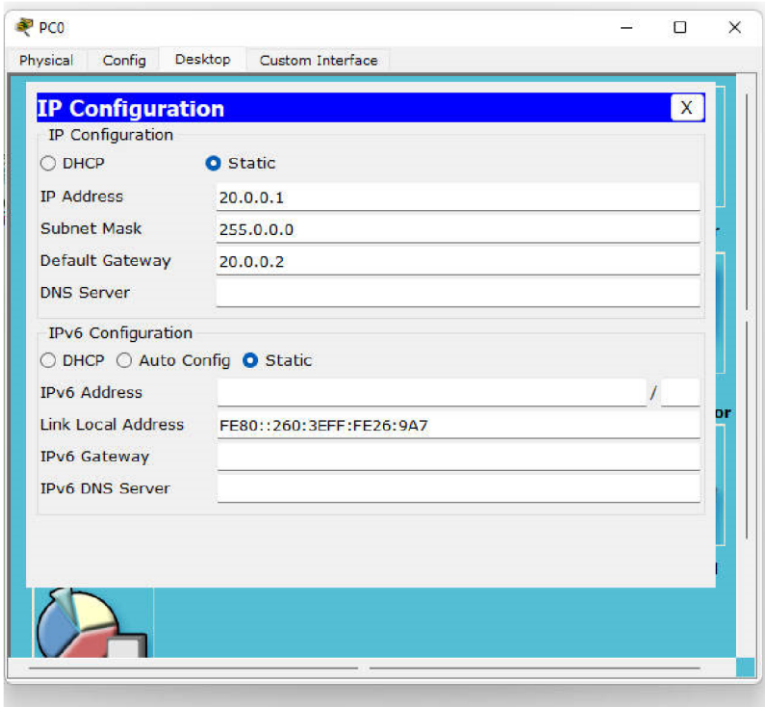


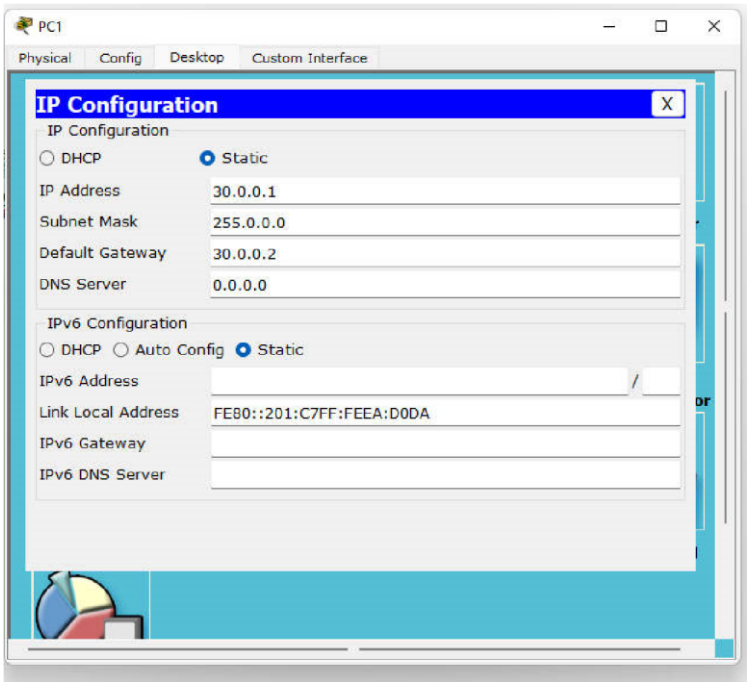
Configure Routers:

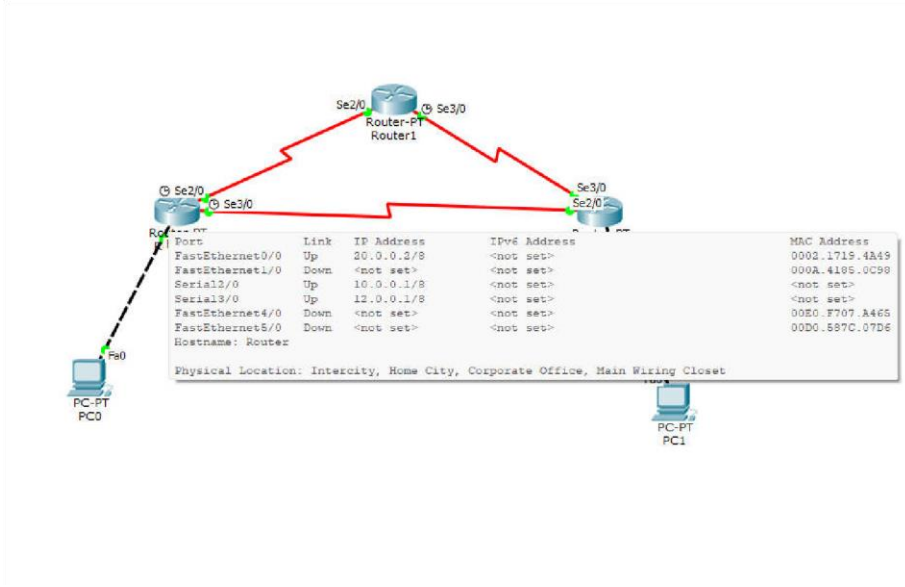

Router0
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
exit
Router(config)#router ospf 1
Router(config-router)#network 20.0.0.0 0.0.0.255 area 0
Router(config-router)#network 10.0.0.0 255.0.0.0 area 0
Router(config-router)#network 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

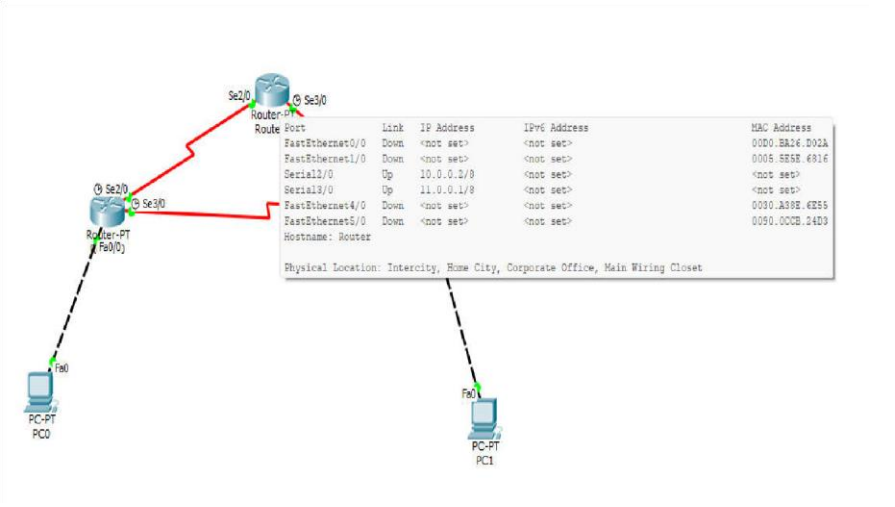

Router1
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
no ip address
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

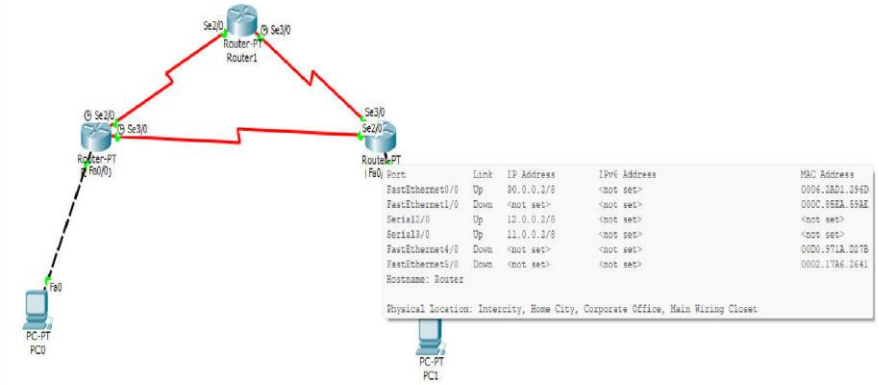

Router2
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#ip address 12.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 11.0.0.2 255.0.0.0
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

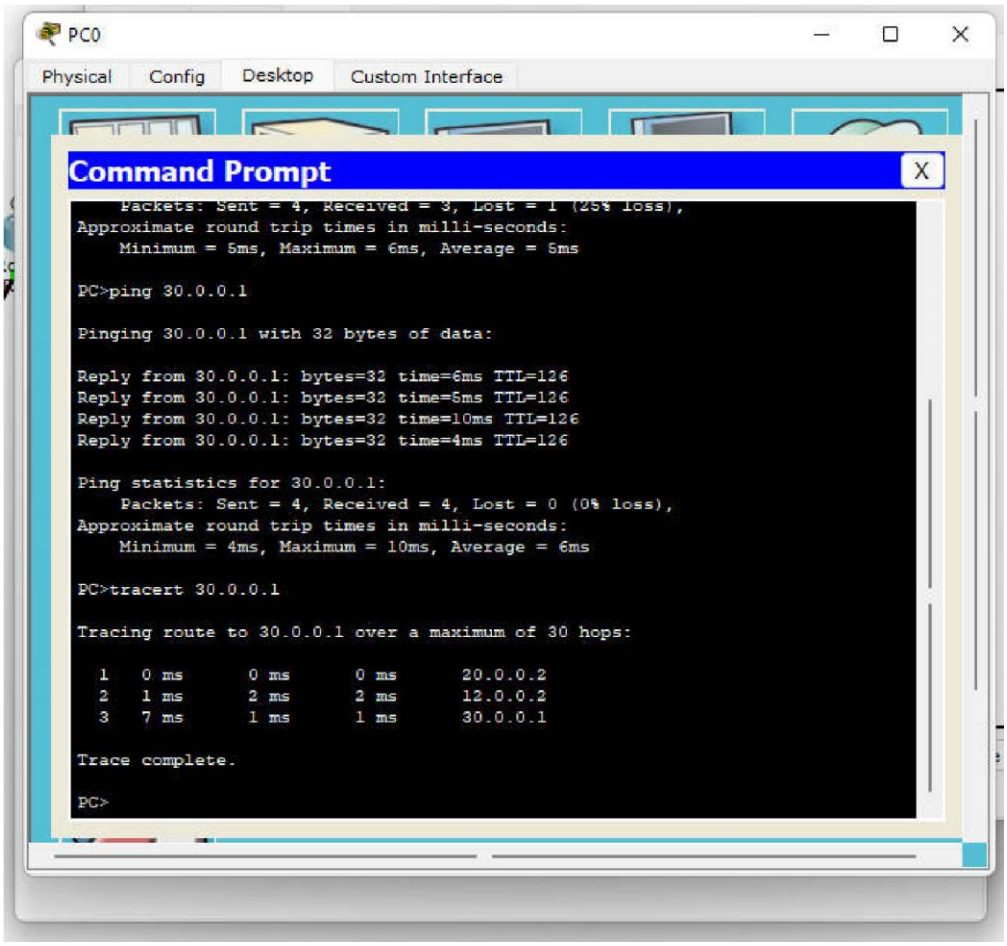

PC1
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
30.0.0.1
Subnet Mask
255.0.0.0
Default Gateway
30.0.0.2
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::201:C7FF:FEEA:D0DA
IPv6 Gateway
IPv6 DNS 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.4195.0C90
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
Fe0
PC-PT
PC0
PC-PT
PC1


Se2/0
Se3/0
Router-PT
Router
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.4E55
FastEthernet5/0 Down <not set> <not set> 0090.0CCB.24D9
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
Router-PT
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
Port Link IP Address IPv6 Address MAC Address
FastEthernet0/0 Up 80.0.0.3/8 <not set> 0006.2AD1.396D
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.D07B
FastEthernet5/0 Down <not set> <not set> 0002.17A6.3641
Hostname: Router
Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet

Output:

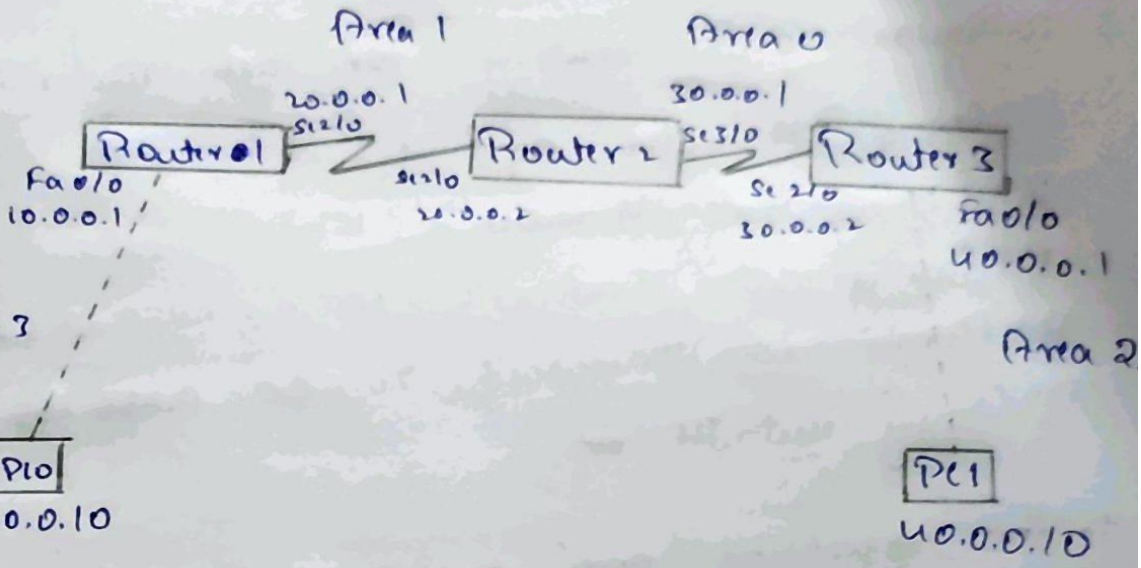

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 6ms, Average = 5ms
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=6ms 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>

4th August, 2023

(05). Configure OSPF routing protocol.

Aim: To understand how to configure OSPF routing protocol & to find.

Topology:



Procedure:

(1). Create a topology as shown above

(2). Configure each router with the IP addresses for required interfaces. 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 its better to configure loopback address to routers. It is a virtual interface that never goes down.
R1(config)# interface se2/0
R1(config-if)# interface loopback 0
R1(config-if)# ip address 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 se2/0
C 40.0.0.0/8 directly connected Fa2/0

C 30.0.0.0/8 directly connected se2/0

So a virtual link blw 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.

Ping PC0 from PC1 to check connectivity

Result:

R3 # Show ip route

O IA 20.0.0.0/8 via 30.0.0.1 se2/0

O IA 10.0.0.0/8 via 30.0.0.1 se2/0

C 40.0.0.0/8 directly connected Fa 0/0

C 30.0.0.0/8 directly connected se 2/0

R2 # Show ip route.

C 20.0.0.0/8 directly connected Se 2/0

C 30.0.0.0/8 directly connected Se 3/0

C 172.16.1.0/16 directly connected loopback 0

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 se 2/0

C 10.0.0.0/8 directly connected fa 0/0

C 20.0.0.0/8 directly connected Se 2/0.

Ping PC0 - PC1

PC1 > ping 10.0.0.10

Reply from 10.0.0.10, bytes=32 time=12ms TTL=12
Reply from 10.0.0.10, bytes=32 time=11ms TTL=12
Reply from 10.0.0.10, bytes=32 time=6ms TTL=12
Reply from 10.0.0.10, bytes=32 time=7ms TTL=12

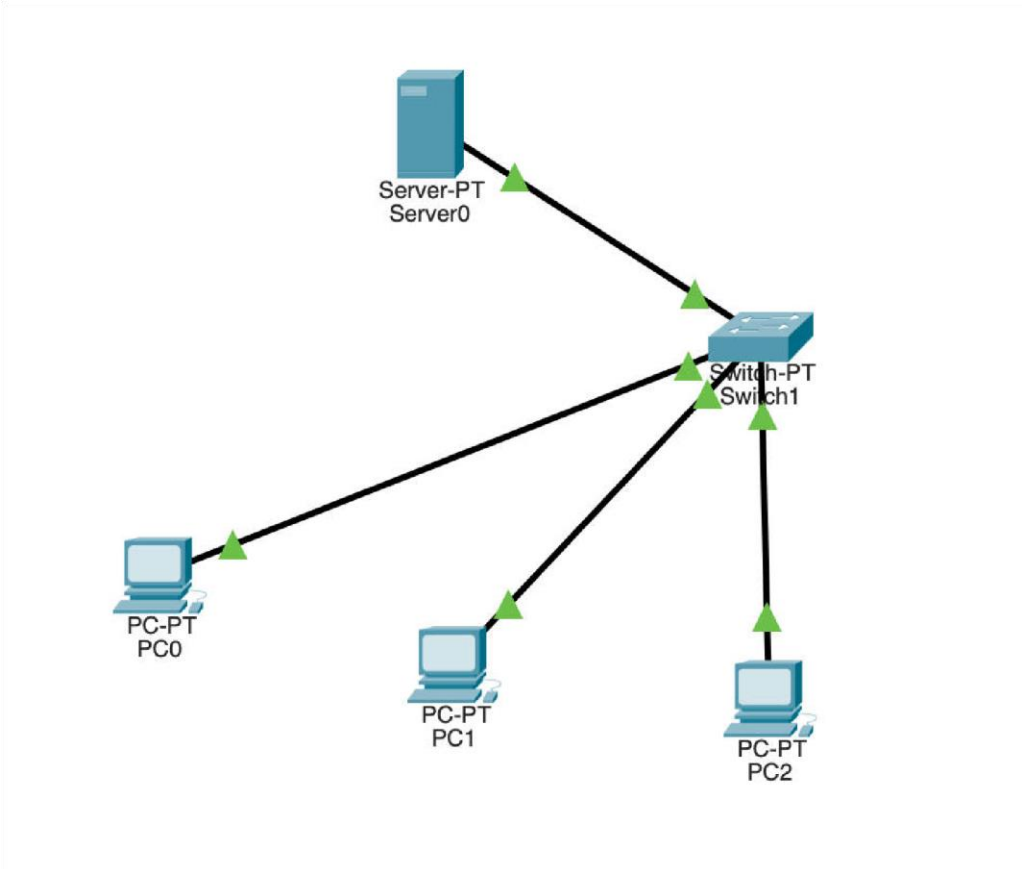
Ping statistics:

Packets: sent = 4, Recieved = 4, lost = 0 (0% loss)
min = 6ms, max = 12ms Avg = 9ms.

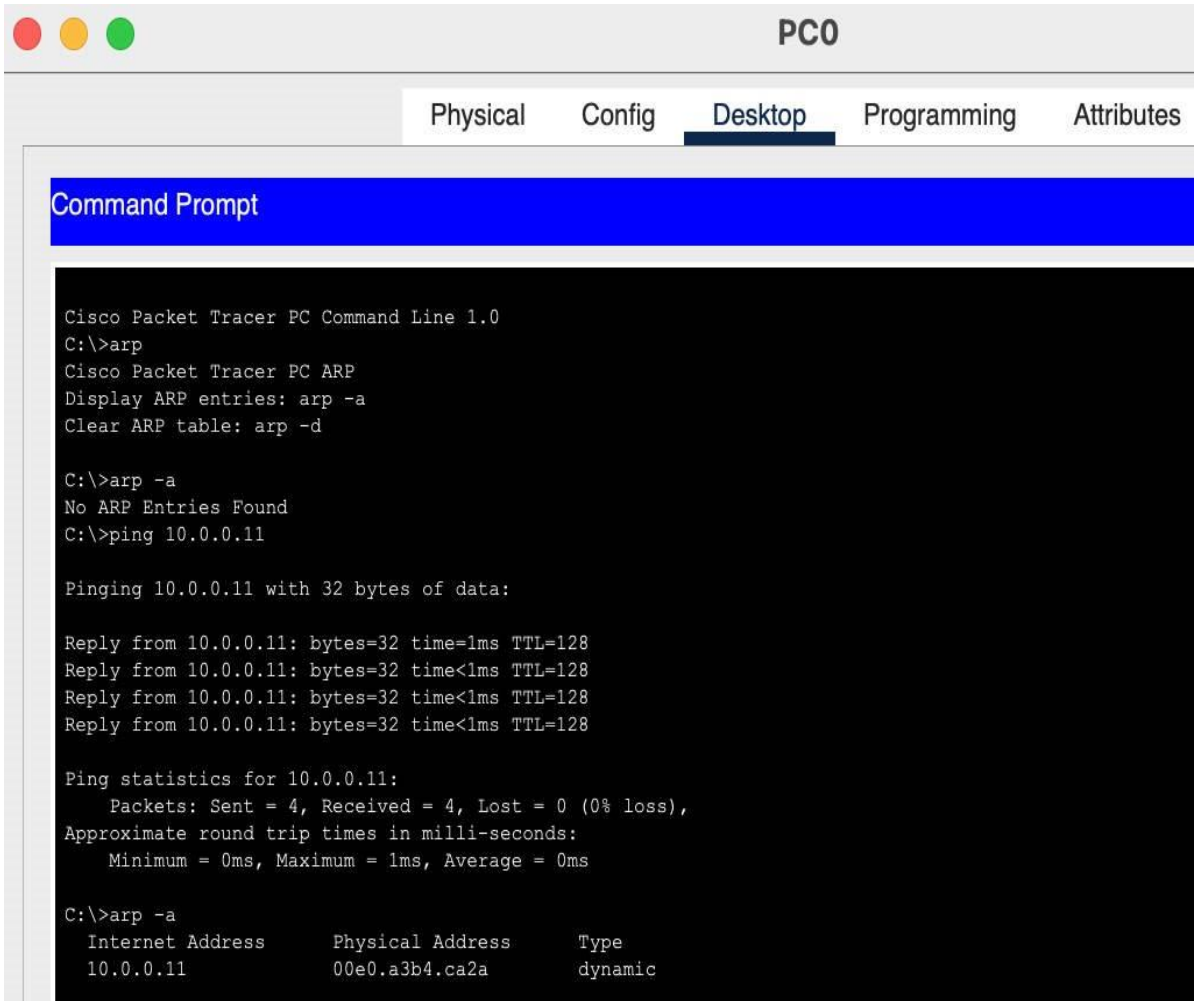
Experiment No. 8

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

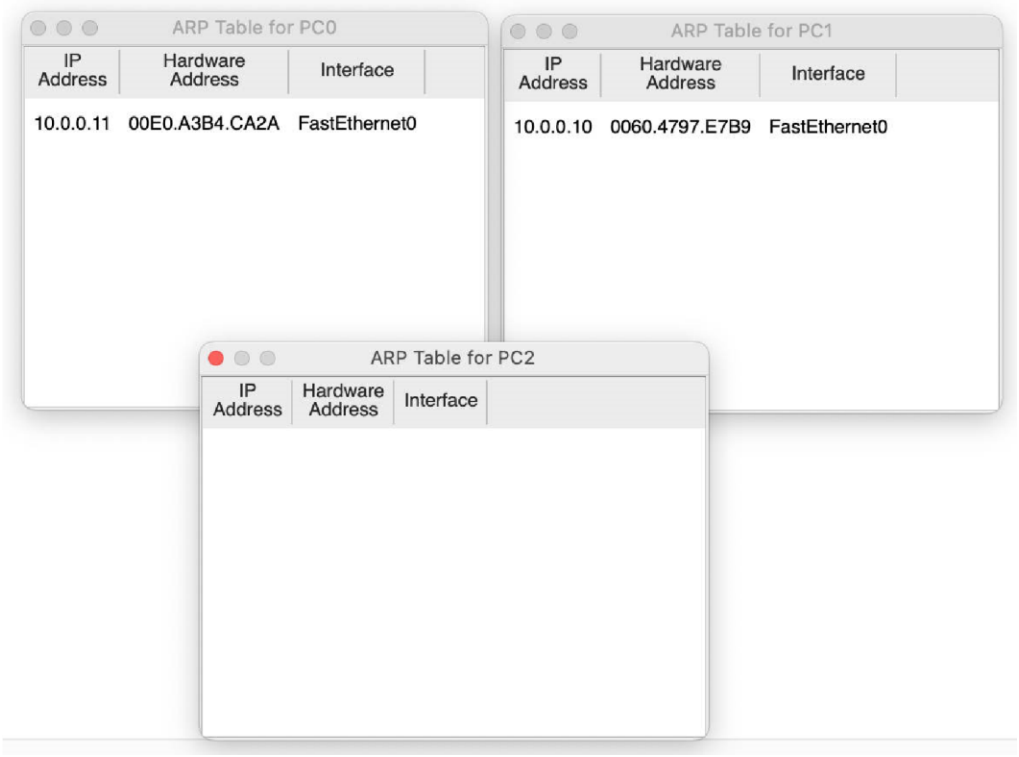
Topology:



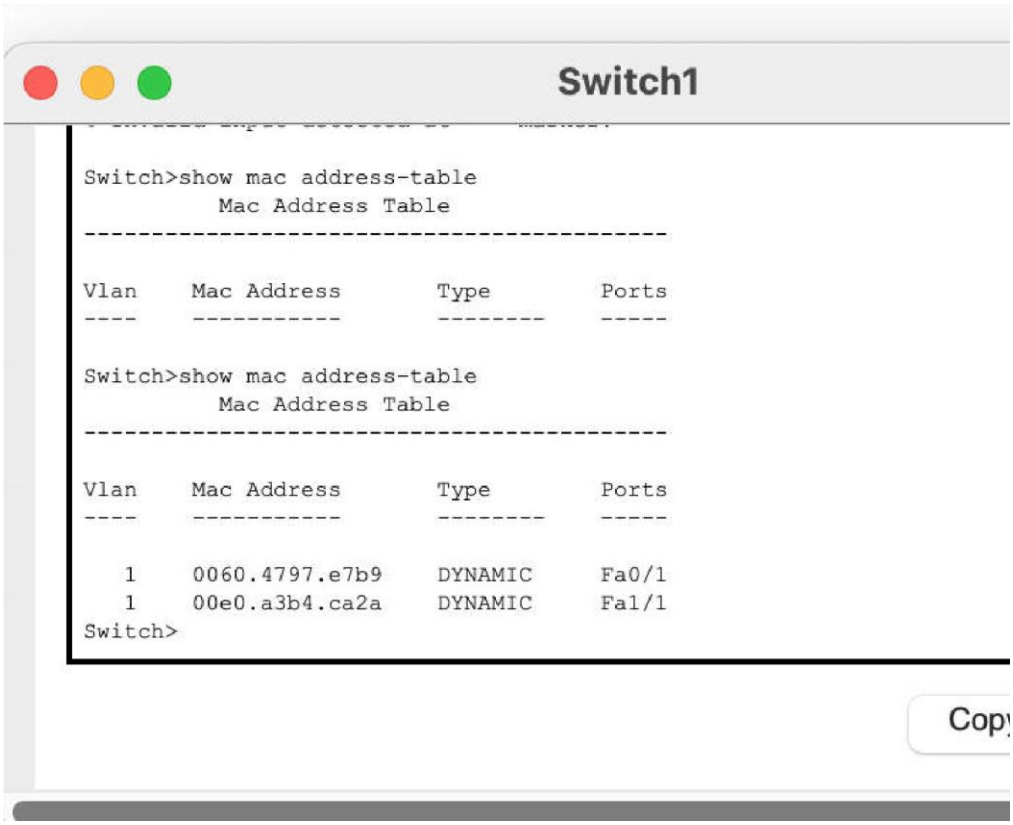
Ping PC1 from PC0:



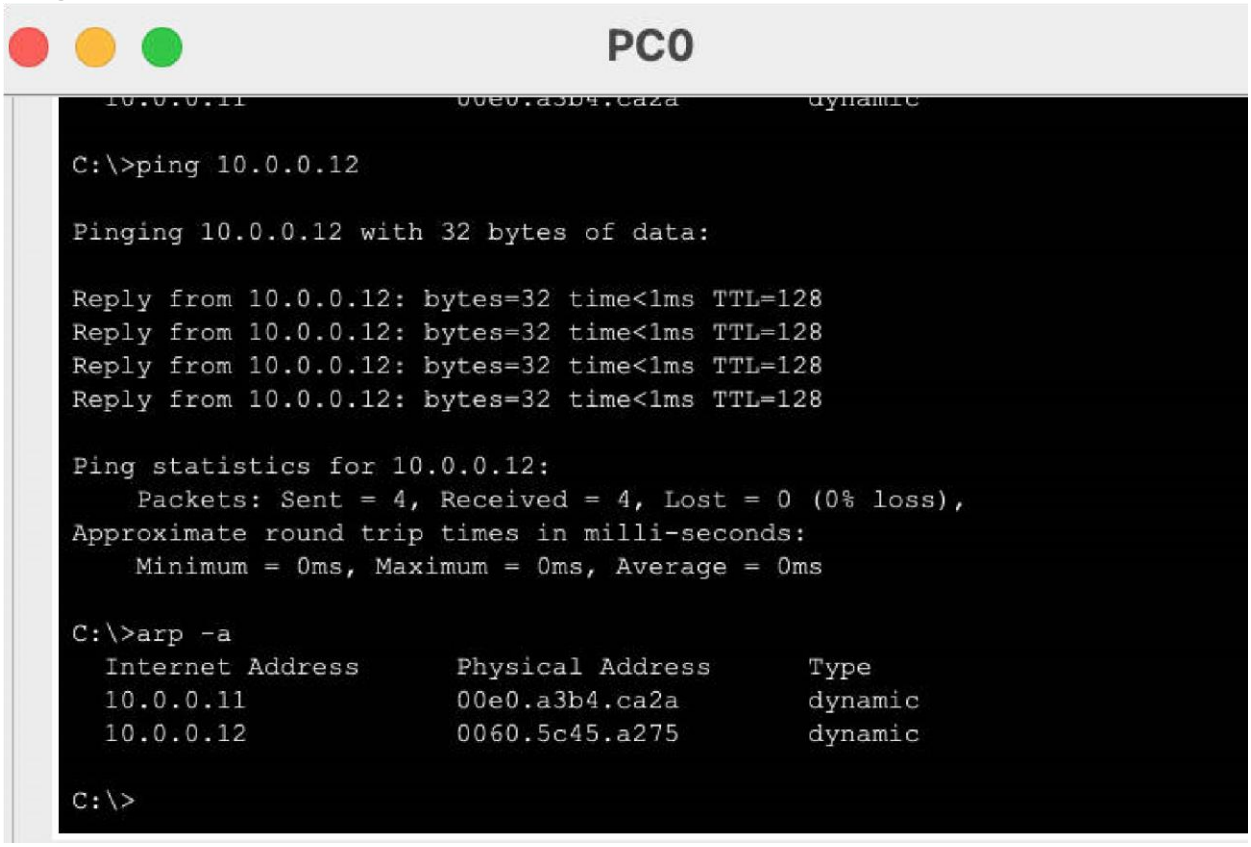
ARP table for all PCs:

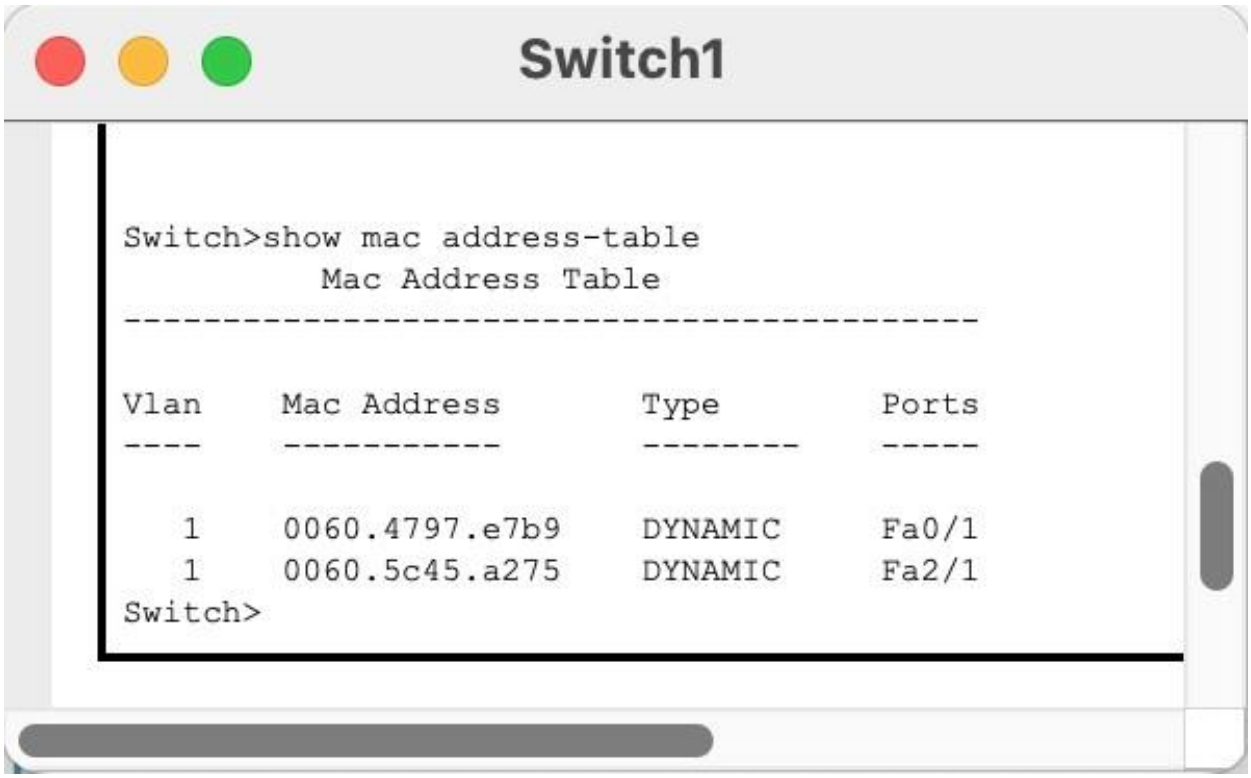


Mac Address Table:



Ping PC2 form PC0:



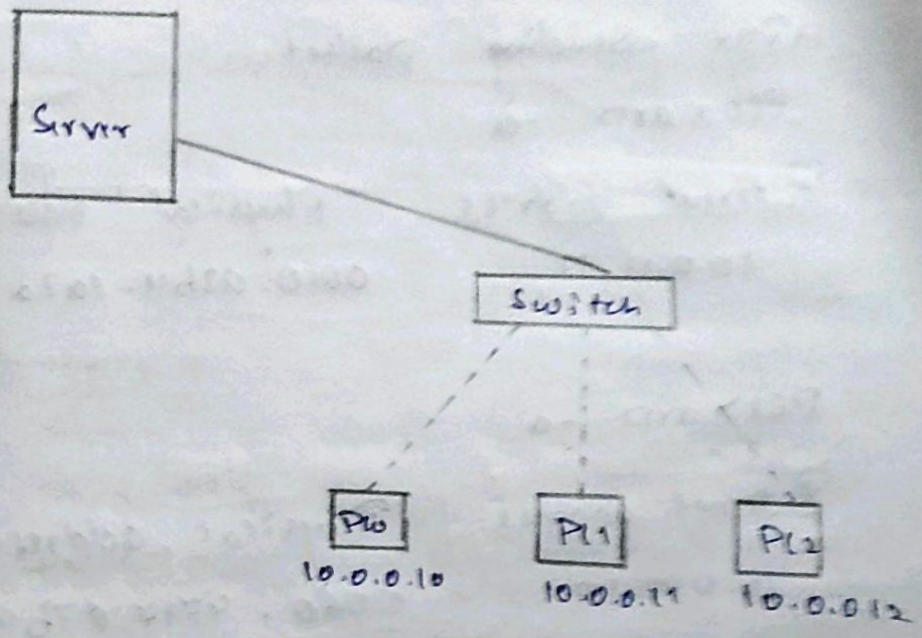


11th August 2023

(8). To construct simple LAN and understand the concept & operation of ARP.

Aim: Understand the concept & 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 table (or). Use the following commands

PC> arp -a

PC>show mac address table

4. Go to simulation mode to send packets bl end devices. Use "capture" button to go step b step to observe changes in ARP table, a and when new communication starts.

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.e7b9	dynamic

Mac Address table in Switch

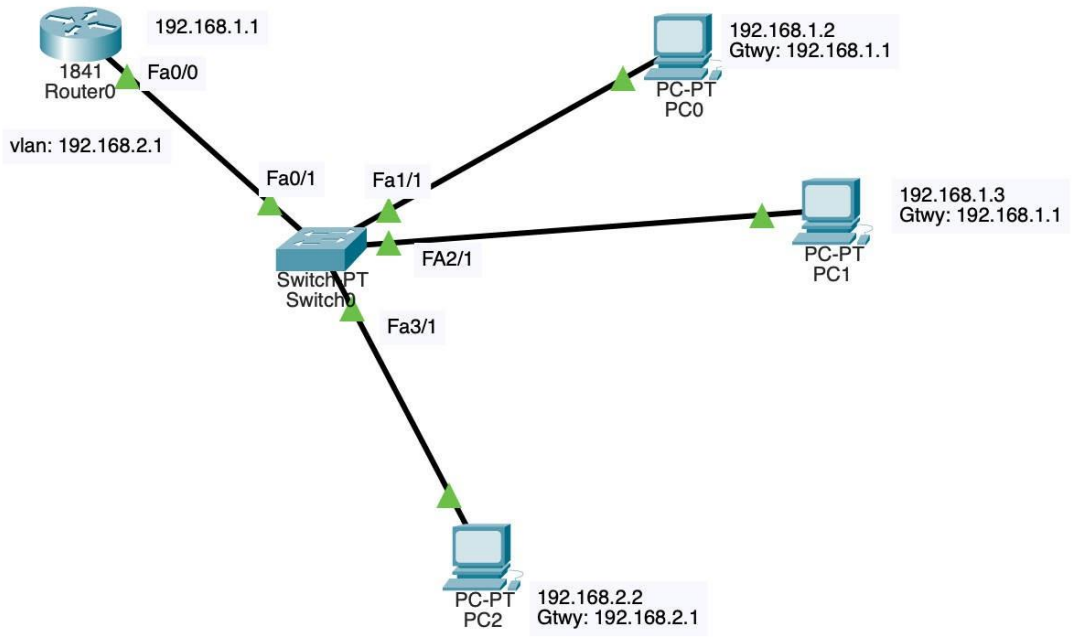
Switch> show mac address-table

Vlan	Mac address	Type	Ports
1	0060.4797.e7b9	DYNAMIC	Fa 0/1
1	00e0.a3b4.ca2a	DYNAMIC	Fa 1/1

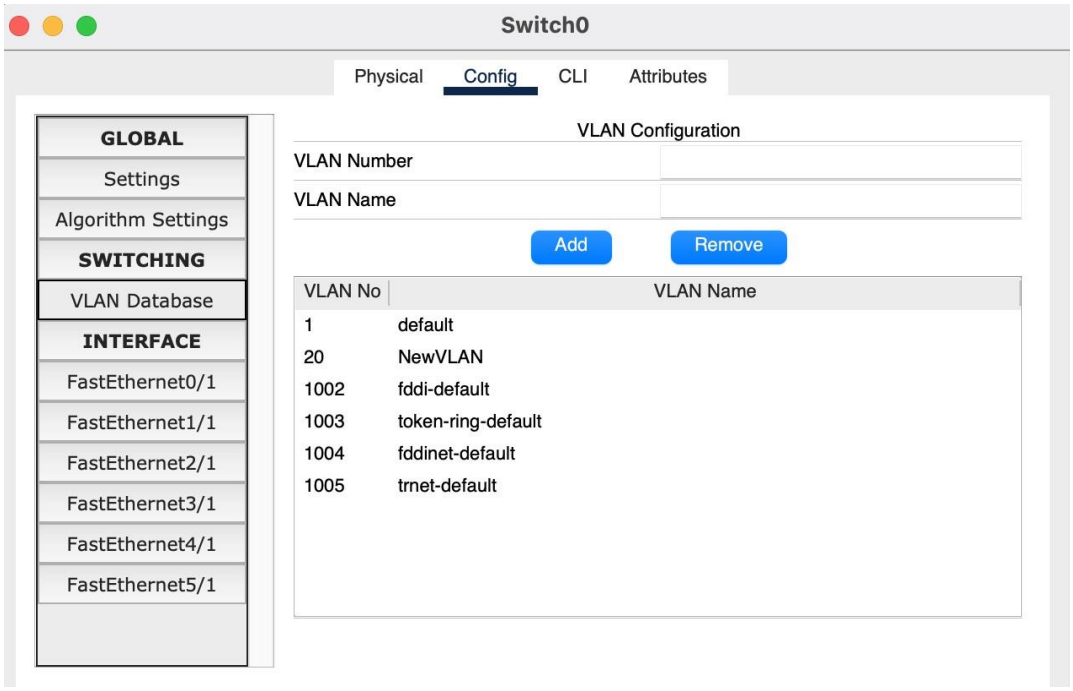
Experiment No. 9

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

Topology:



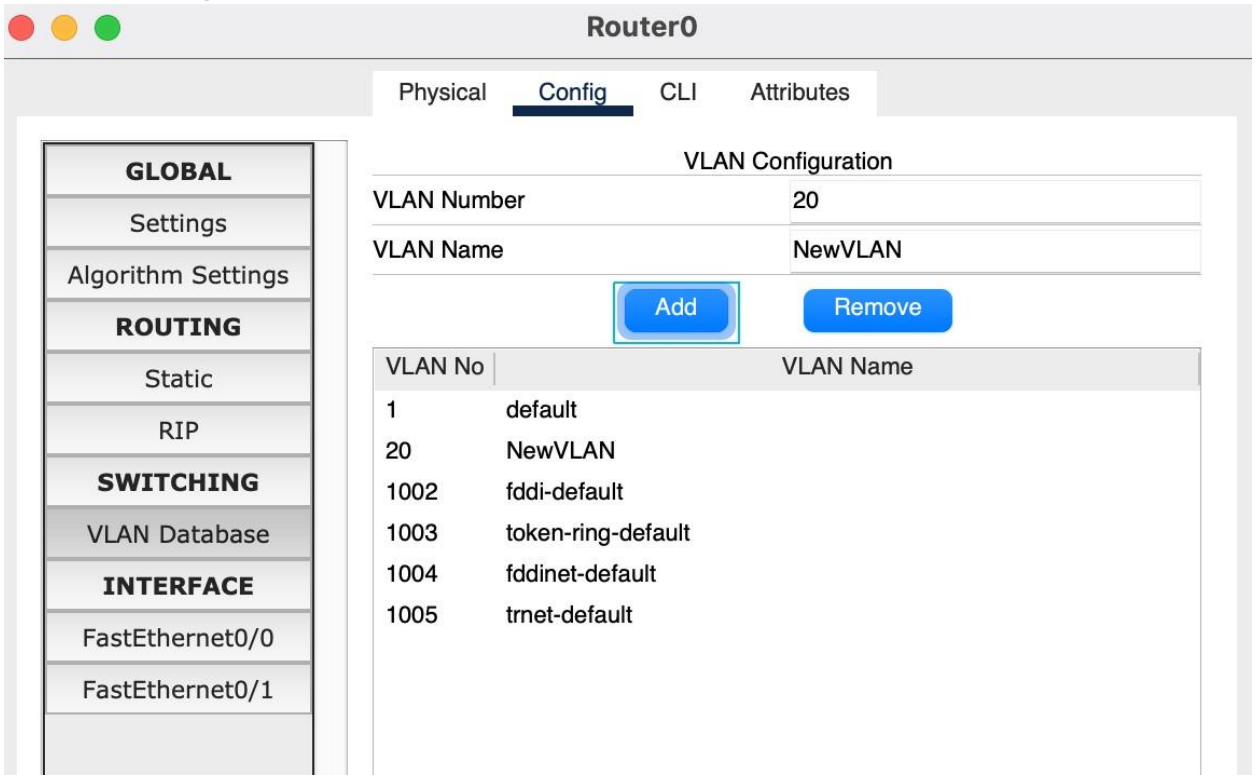
Create VLAN:

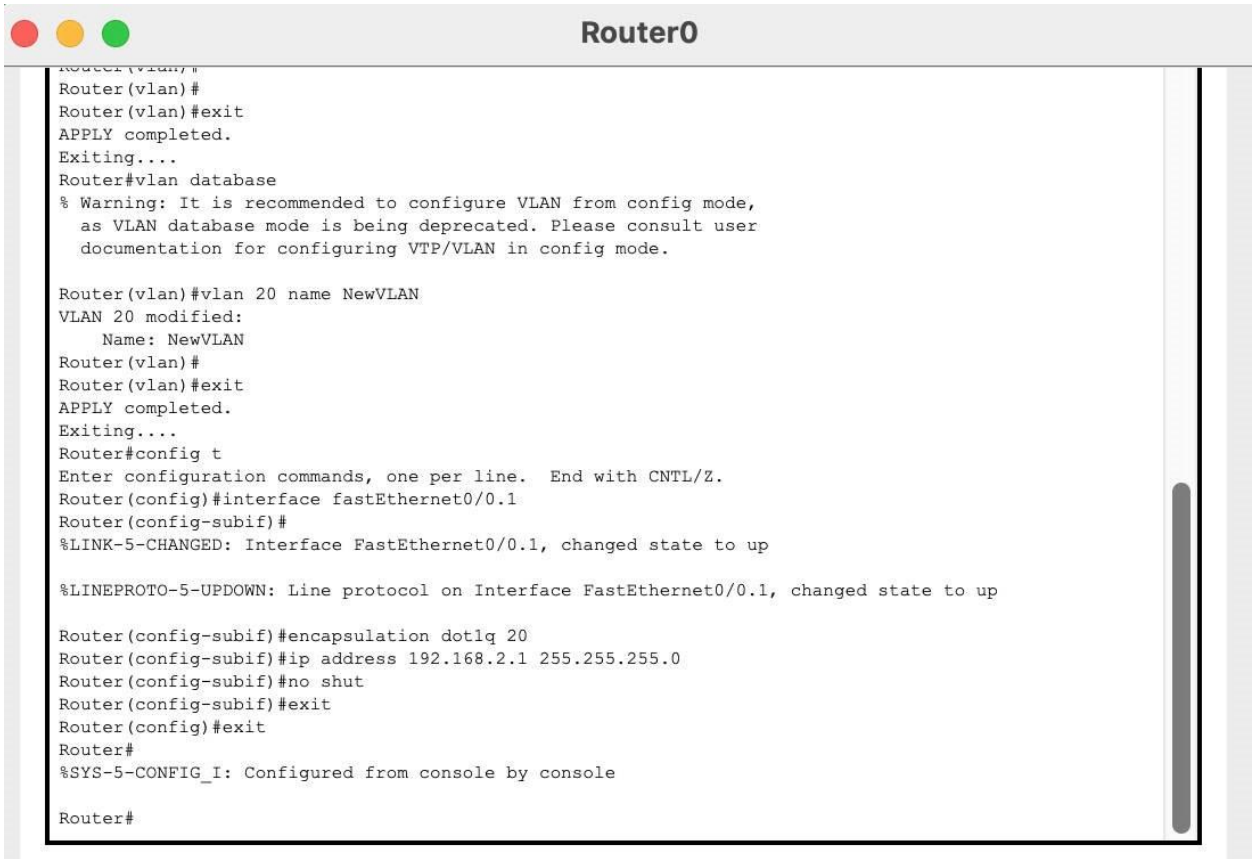


Trunking:

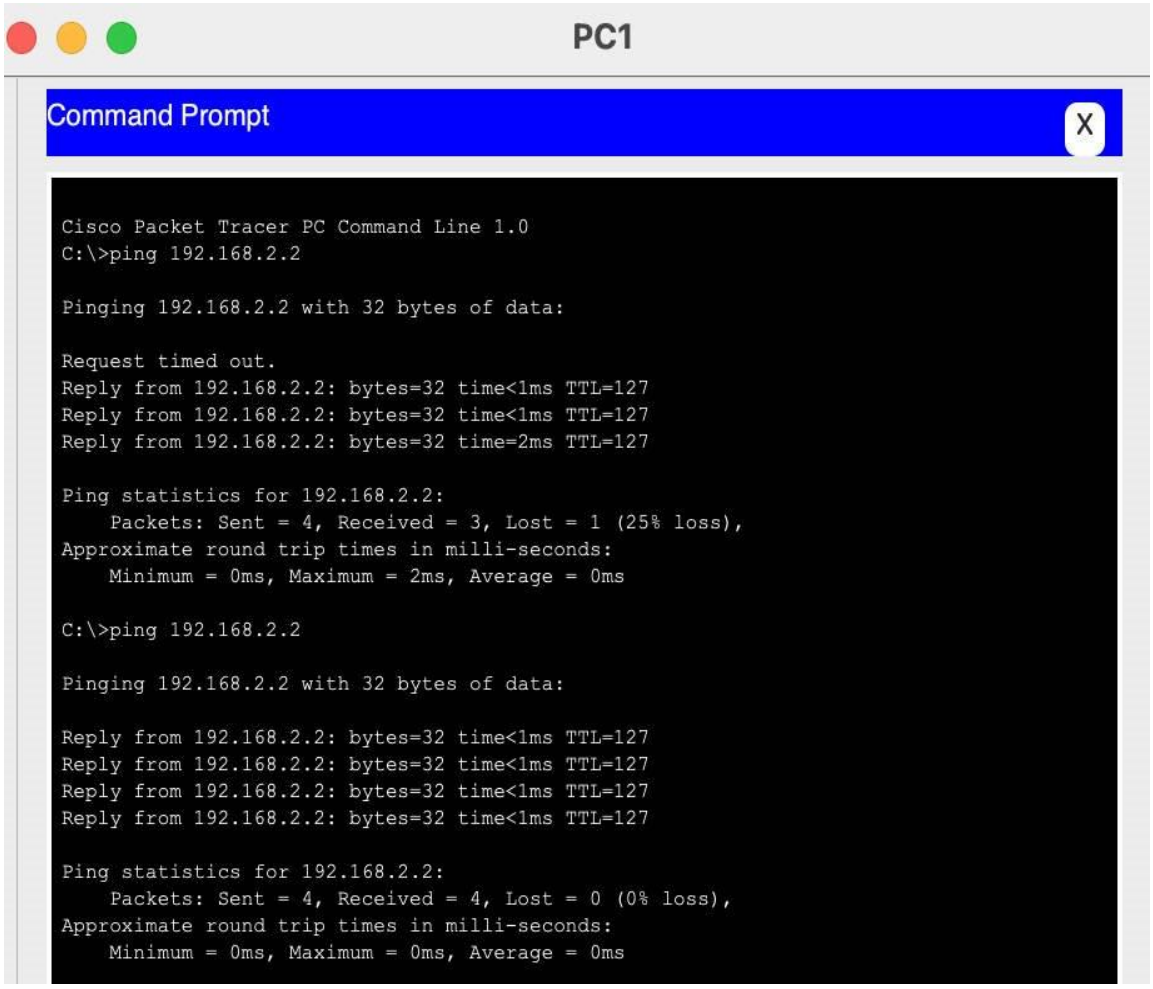
Add end devices to VLAN:

Router config:

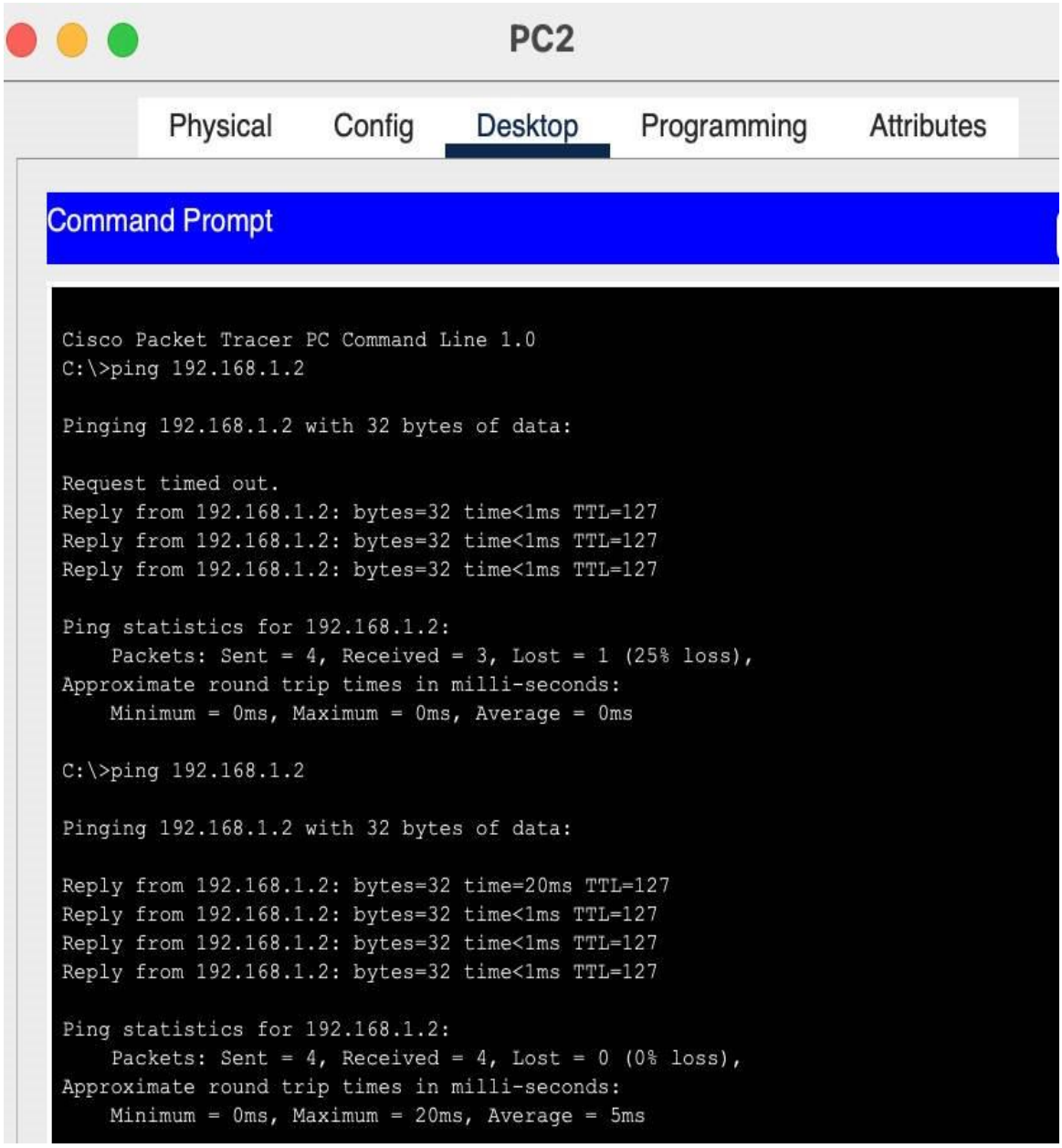




Pinging PC2(in VLAN) from PC1:



Pinging PC0 from PC2(in VLAN):

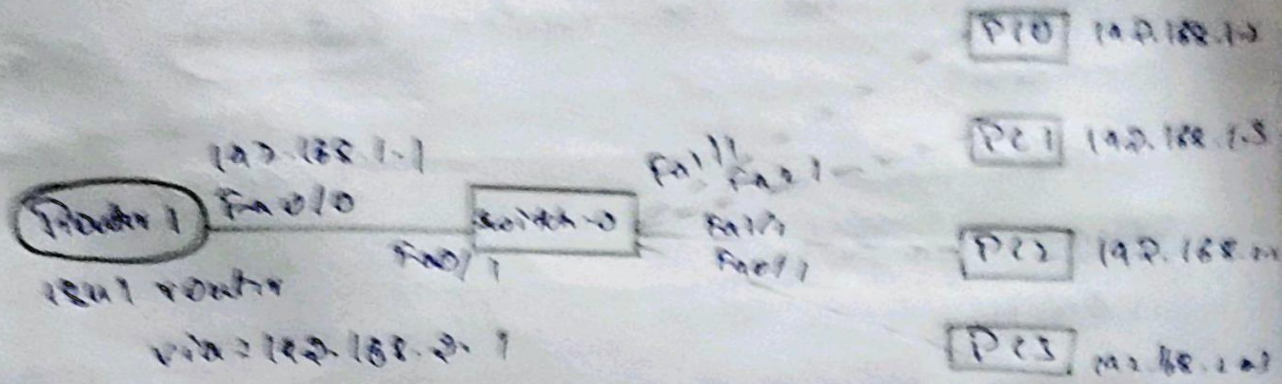


11-2-23

To create a VLAN & make the PC's communicate with existing 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 choose VLAN database. Then click on add.
   VLAN number : 20
   VLAN name : NewVLAN
3. Select the interface of the switch which connects it to router, then Fa 0/1 & make it trunk.
4. In the interfaces of the switch connecting to PC2 & PC3 [end devices of NewVLAN] select

20 : New VLAN under VLAN. This makes the switch understand PC2 & PC3 are in NewVLAN.

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

(7). Open CLI in the router. Enter the following commands

Router (Vlan)# exit
APPLY completed
Exiting

Router# config t
Router (Config)# interface FastEthernet 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

Ping the end devices using end devices in VLAN & physical LAN to check connection.

Result:

PC1> Ping 192.168.2.2

Reply from 192.168.2.2, bytes=32 time<1ms TTL=127

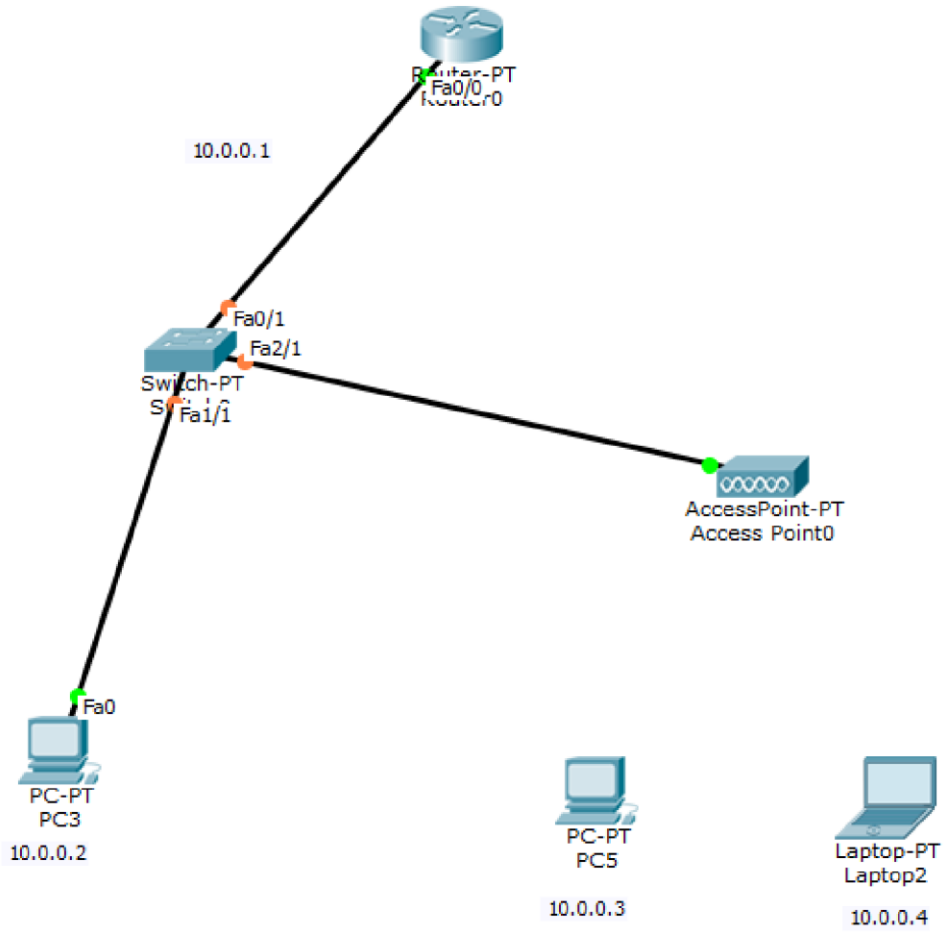
Reply from 192.168.2.2, bytes=32 time<1ms TTL=

Experiment No. 10

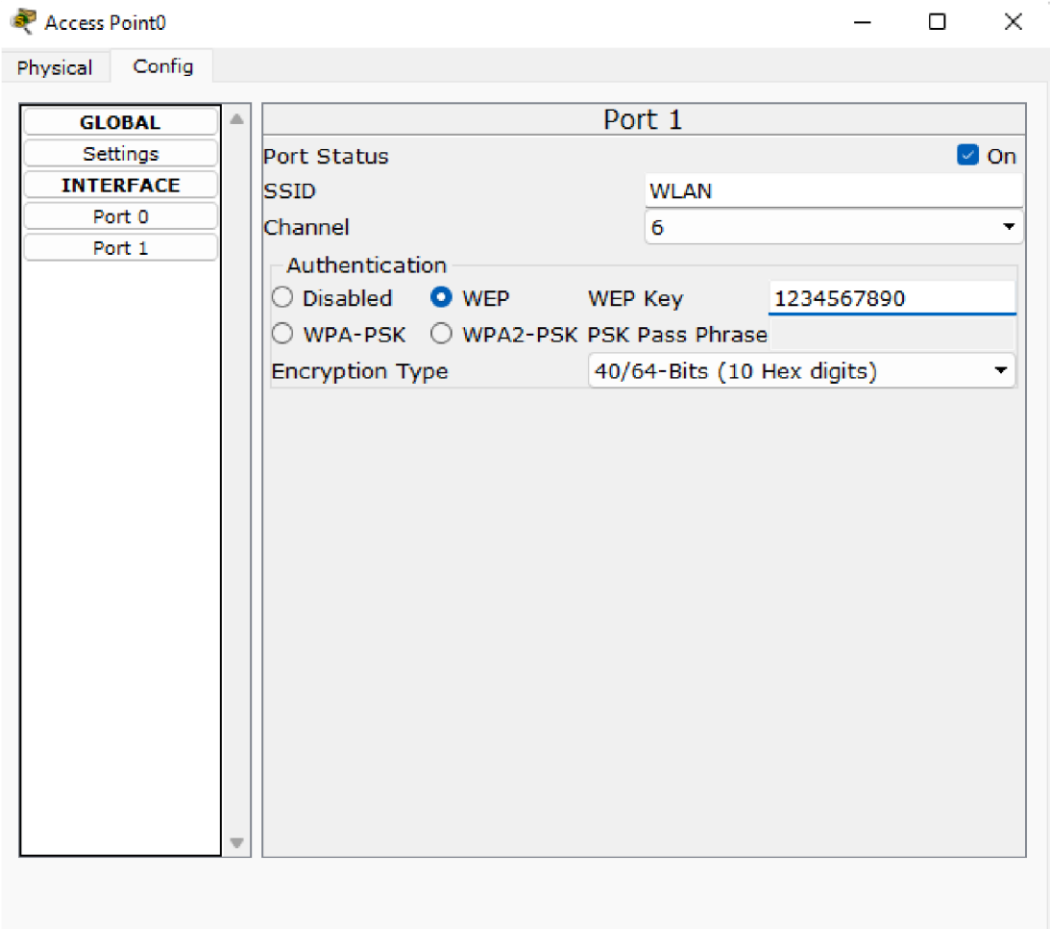
Title:

To construct a WLAN and make the nodes communicate wirelessly

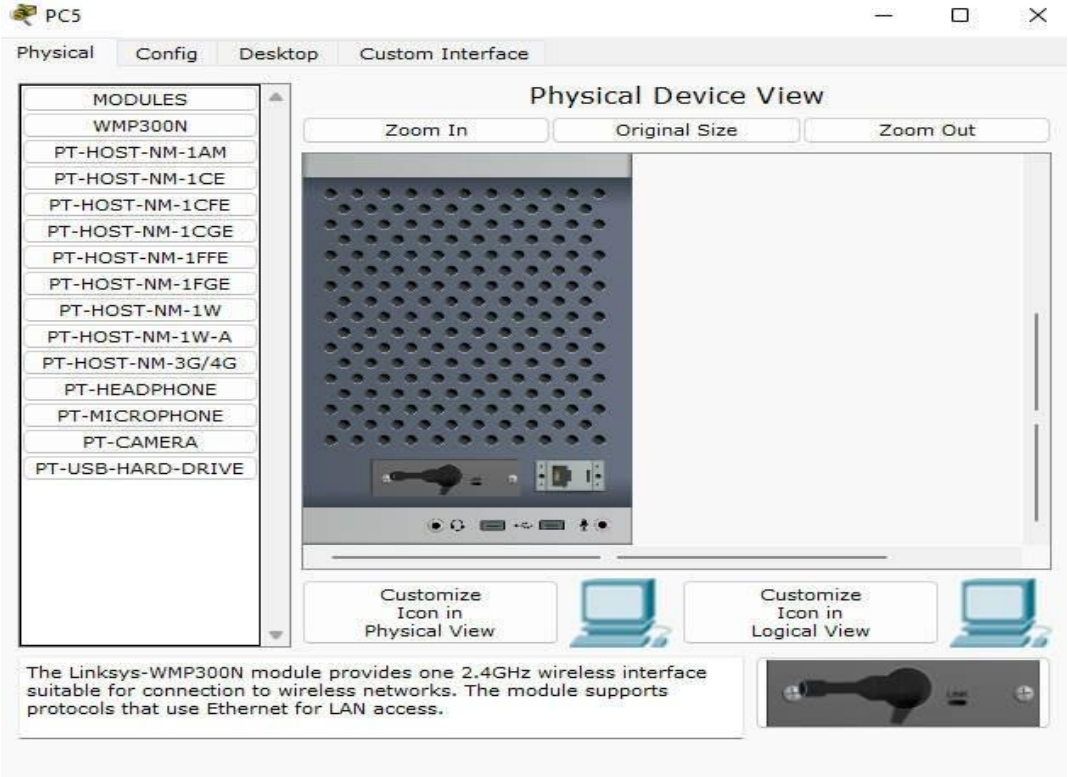
Topology:

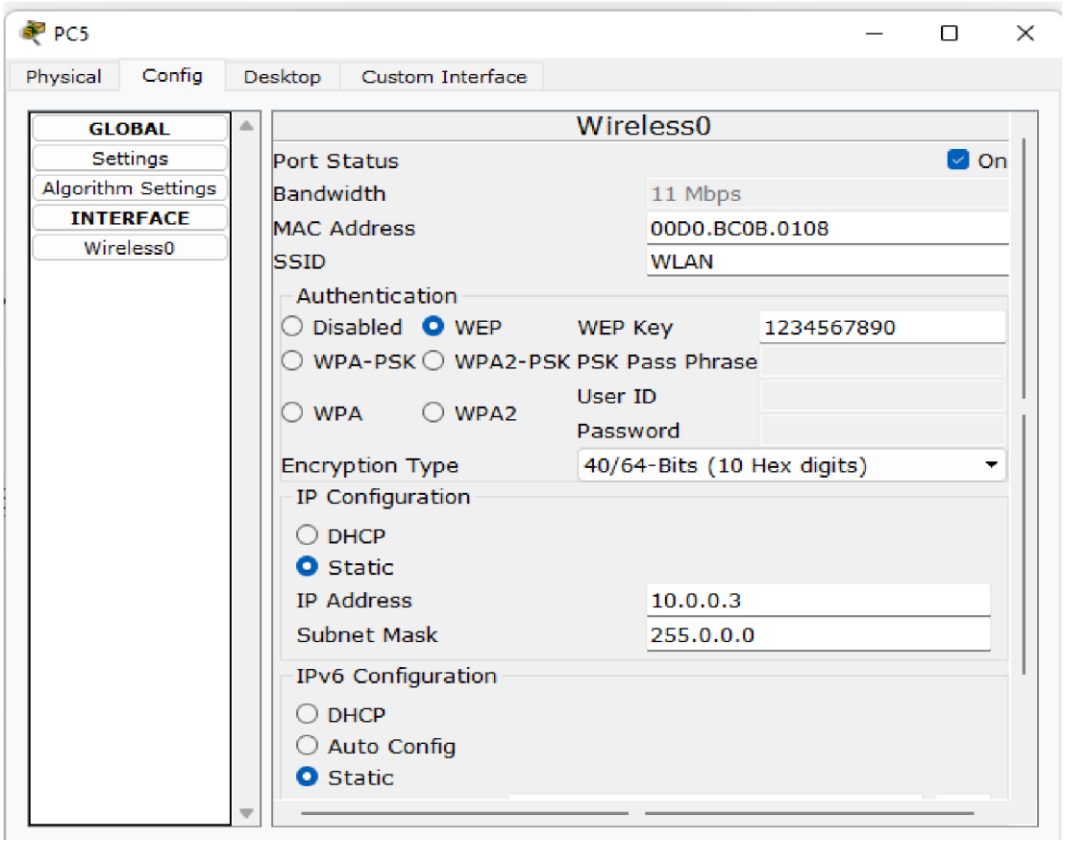


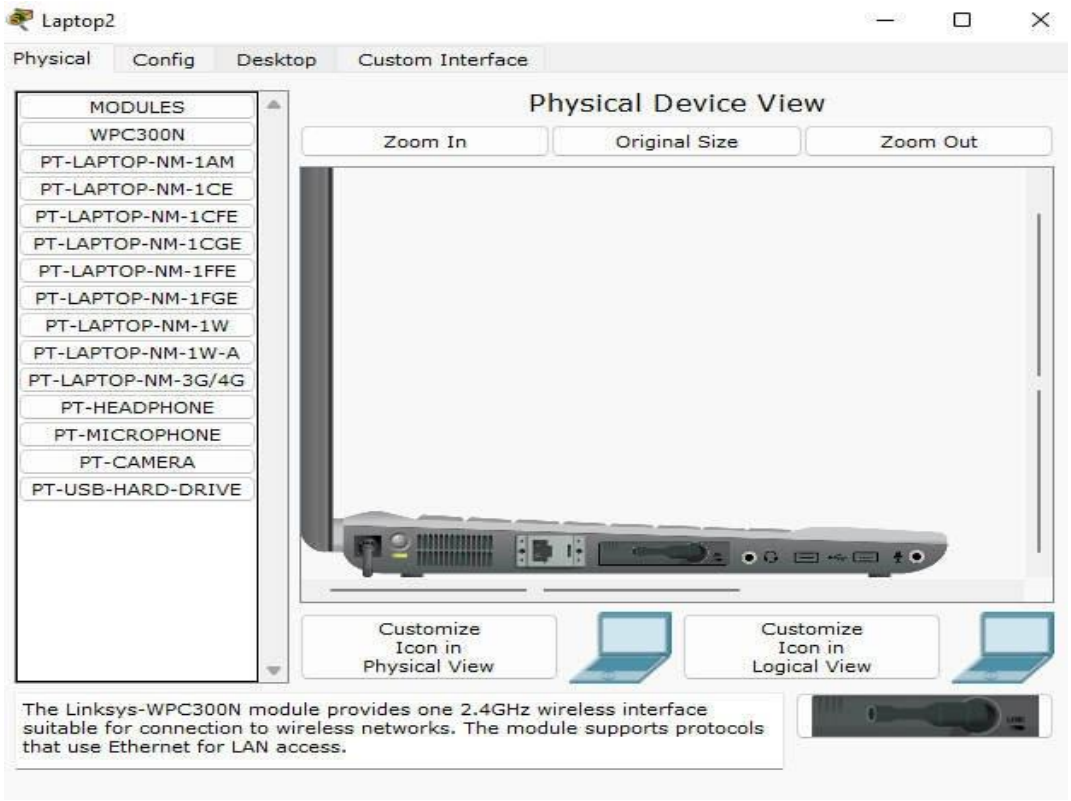
AccessPoint config:

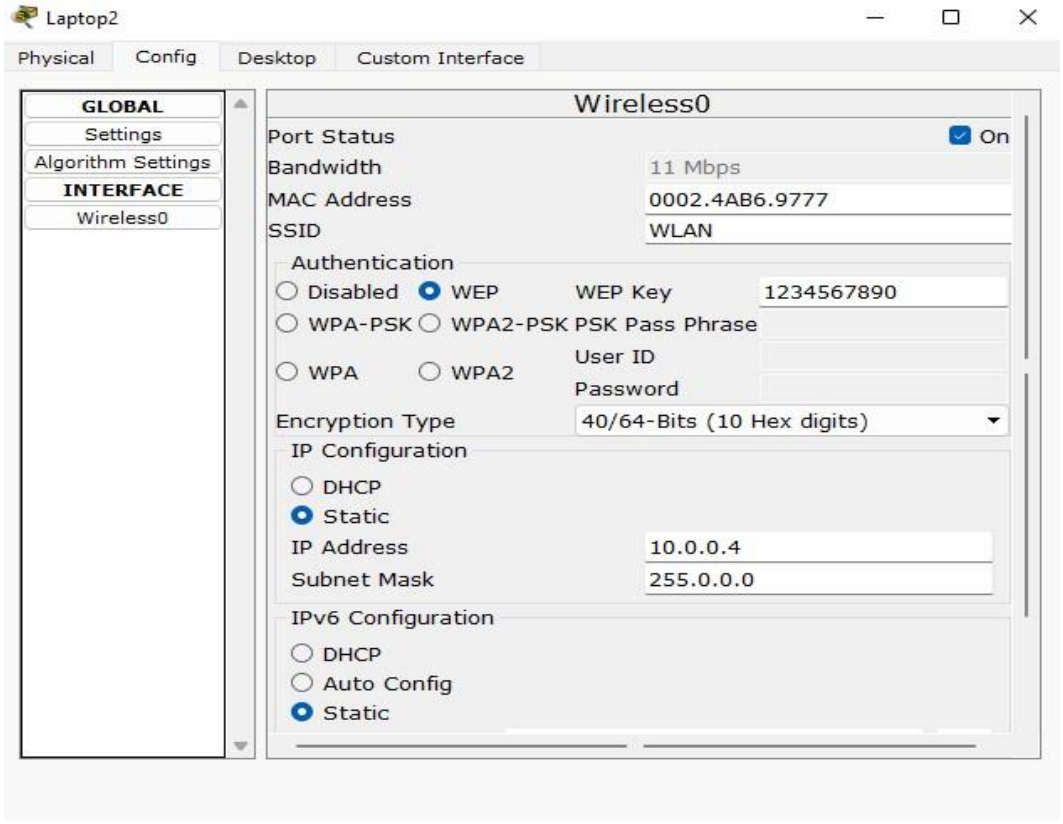


Configure wireless nodes:

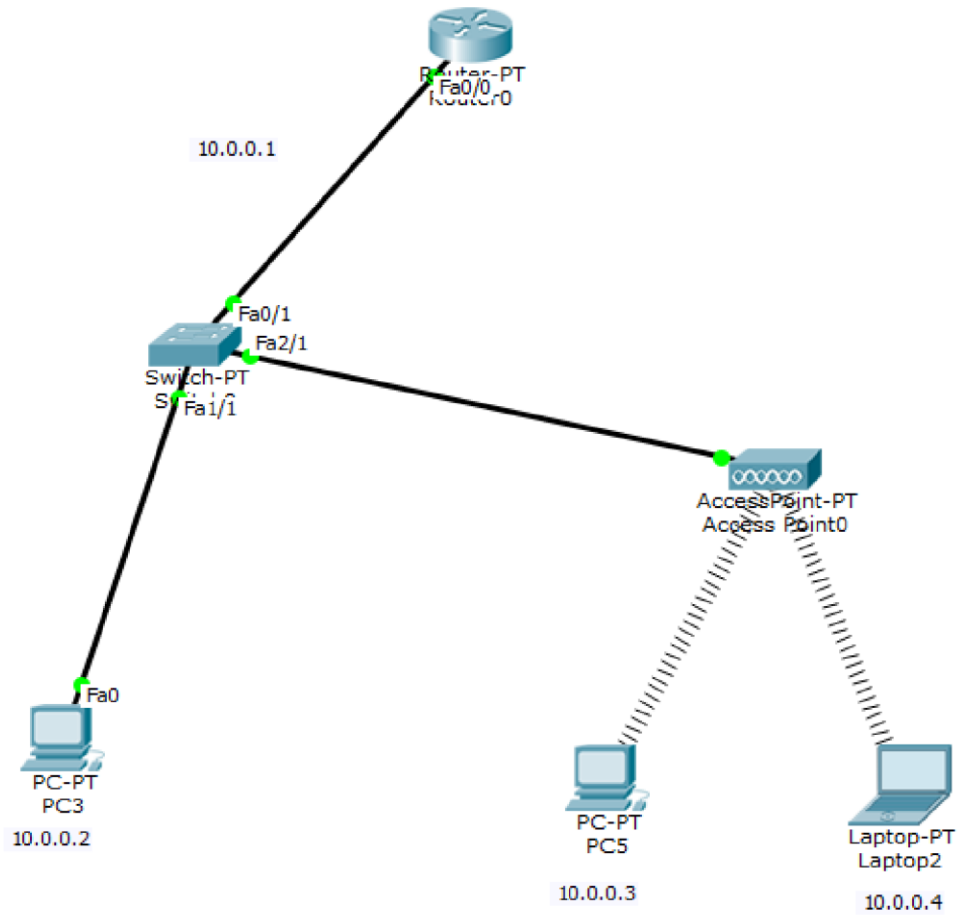



PC5
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
Wireless0
Port Status
On
Bandwidth
11 Mbps
MAC Address
00D0.BC0B.0108
SSID
WLAN
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
WPA
WPA2
User ID
Password
Encryption Type
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

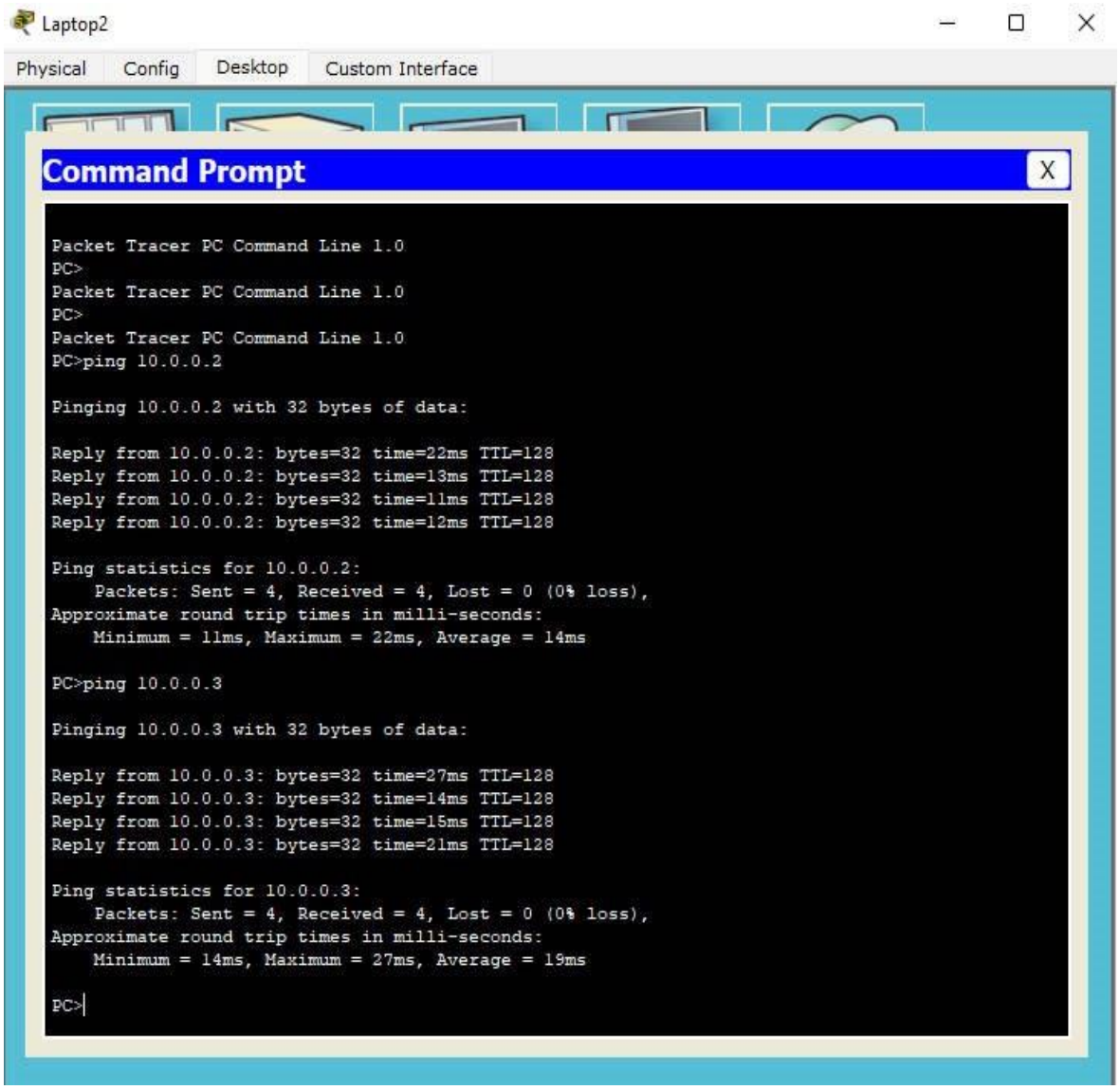

Laptop2
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

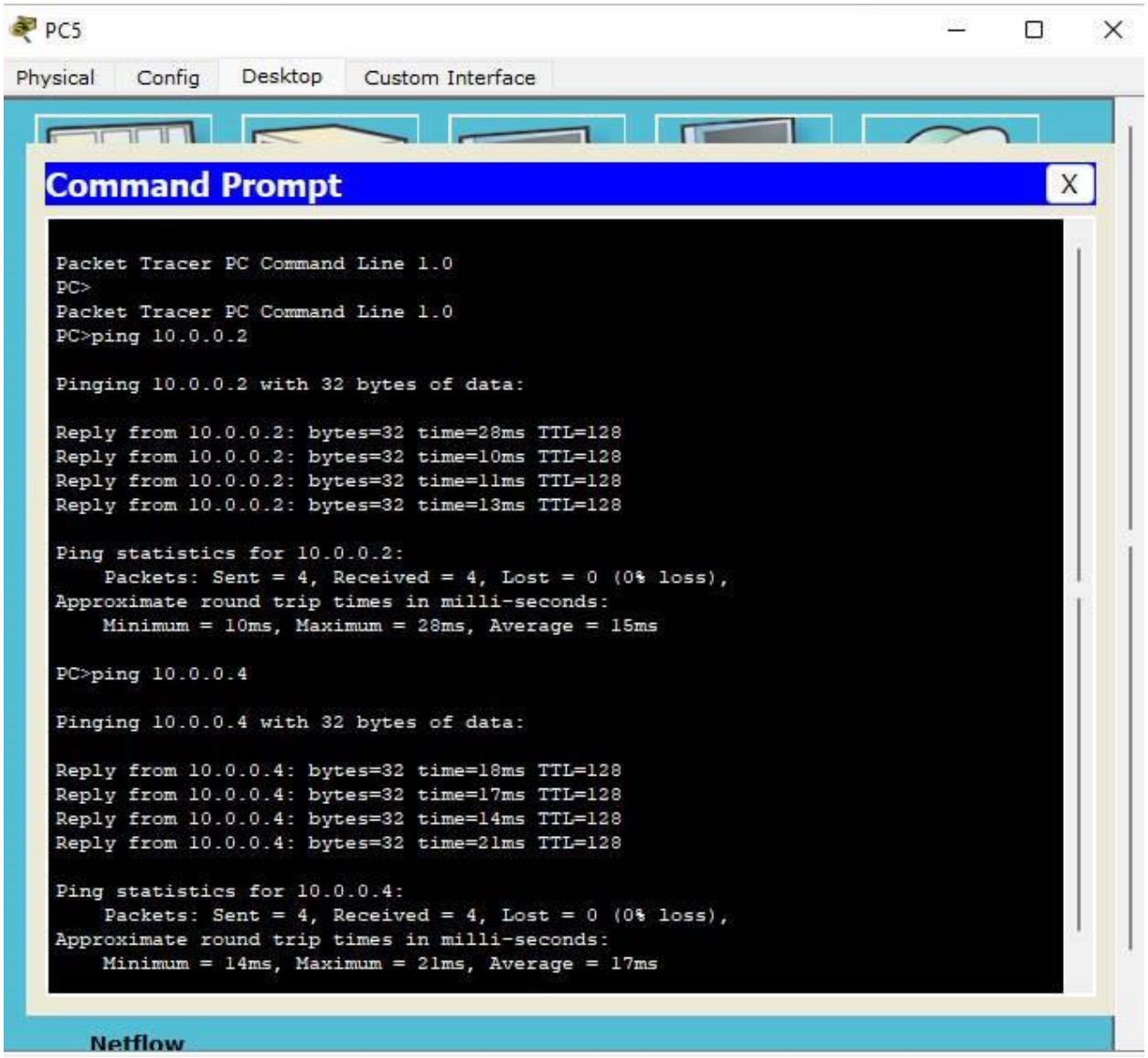

Laptop2
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
Wireless0
Port Status
On
Bandwidth
11 Mbps
MAC Address
0002.4AB6.9777
SSID
WLAN
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
WPA
WPA2
User ID
Password
Encryption Type
40/64-Bits (10 Hex digits)
IP Configuration
DHCP
Static
IP Address
10.0.0.4
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static

Final Topology:



Pinging end devices:



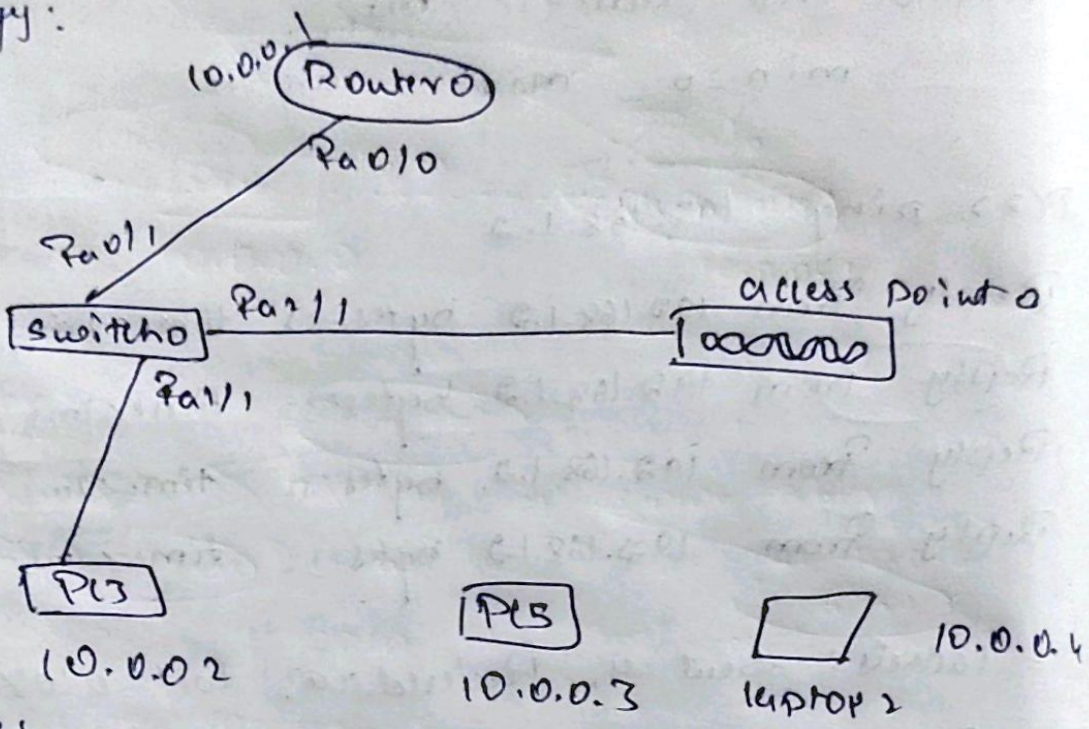

PC5
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=28ms TTL=128
Reply from 10.0.0.2: bytes=32 time=10ms TTL=128
Reply from 10.0.0.2: 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

18th august 202

10. Title: To construct a WLAN & make the nodes communicate wirelessly.

Aim: Under how to construct a WLAN.

Topology:



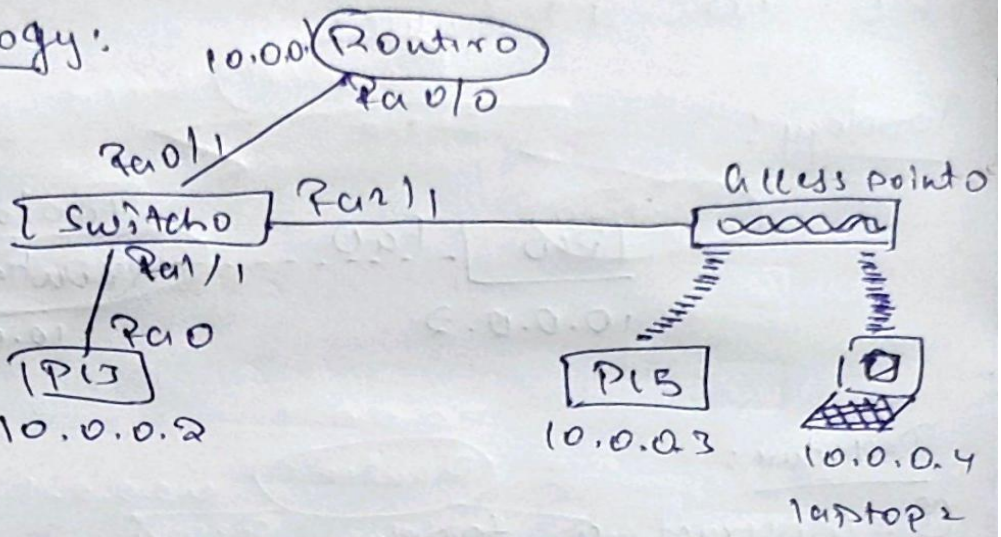
Procedure:

1. Create the topology as shown above.
2. Configure PC 3 & 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 any 10 digit hex key (1234567890, here) configure PC5 and 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 port & switch on the device.
6. In the config tab, a new wireless interface will be added. Now configure,

SSID, WEP, WEP key, IP address and gateway to the devices (PC5, laptop2).

Result:

Final topology:



Pinging end devices:

laptop 2> ping 10.0.0.2

Reply from 10.0.0.2: bytes=32 times=22ms TTL=12
Reply from 10.0.0.2: bytes=32 times=13ms TTL=120
Reply from 10.0.0.2: bytes=32 time=11ms TTL=12
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 = 14 ms.

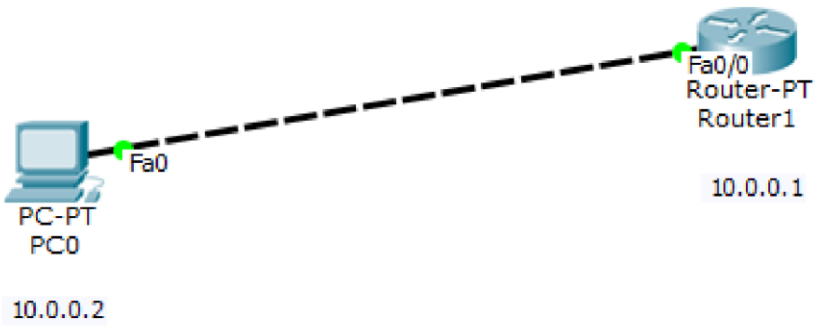
Observation:

To connect devices wirelessly, the end devices are configured with wireless interface (WMP300N component) and the corresponding SSID, WEP & WEP-key of WLAN is added. The Accesspoint is used to create a WLAN by specifying the SSID & the WEP-key.

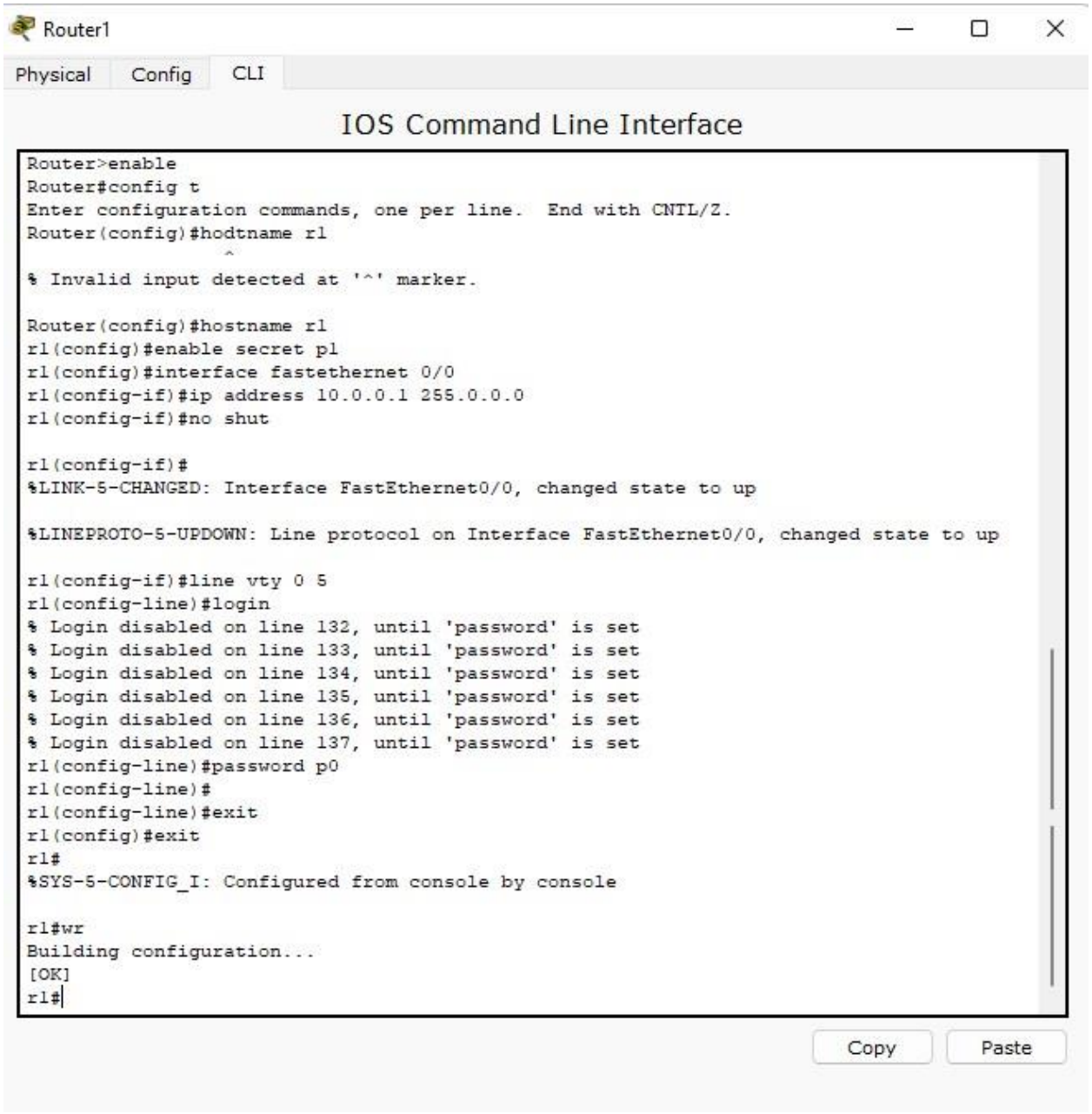
Experiment No. 11

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

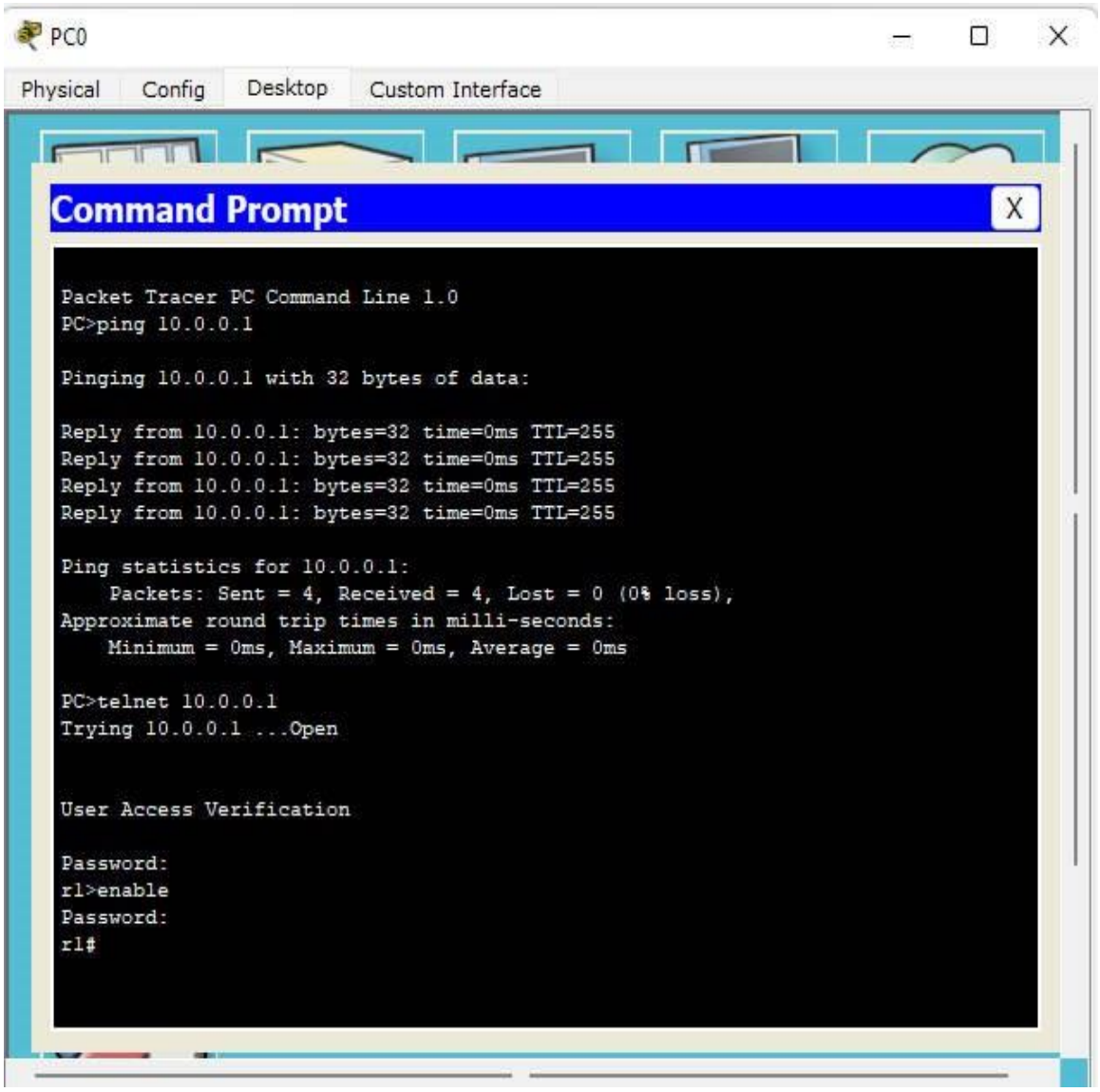
Topology:

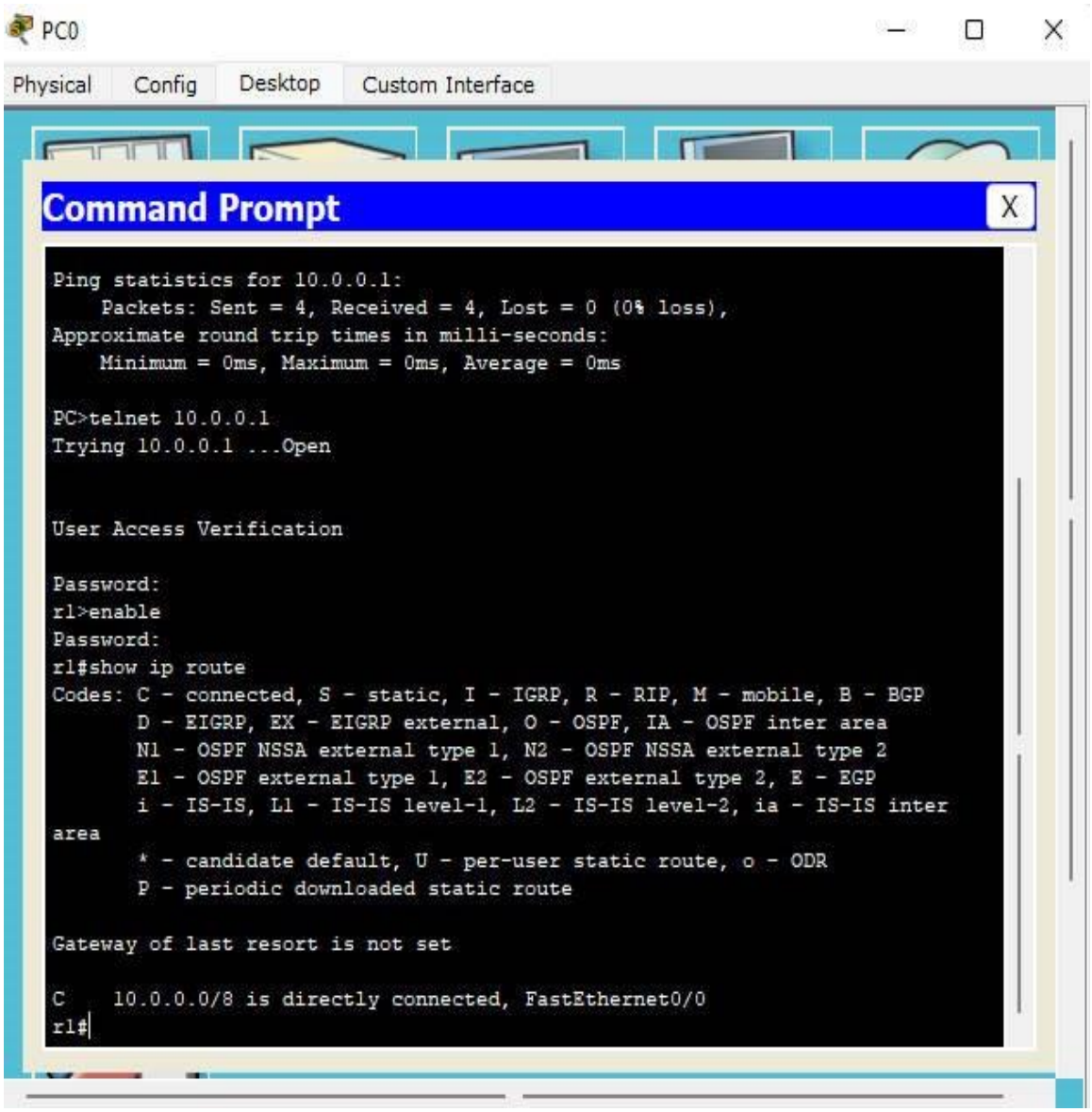


Router config:



Pinging & accession Router CLI from PC:



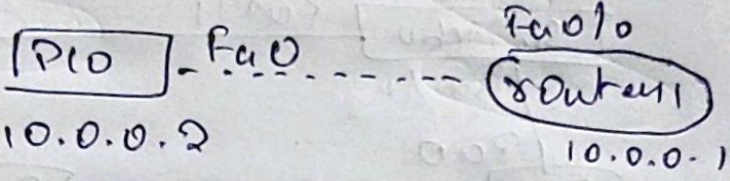

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
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).

25th-8-23

Title : To understand the operation of TELNET by allessing the router in server 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# config t.
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
                    alless for 6 users]
R1(config-line)# login
R1(config-line)# password P0
R1(config-line)# exit
R1(config)# exit.
```

r1#wr [to save changes in Router].

(M). Router 0's CLI can be opend from PC0'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 10.0.0/8 directly connected Fa0/0.

Observation:

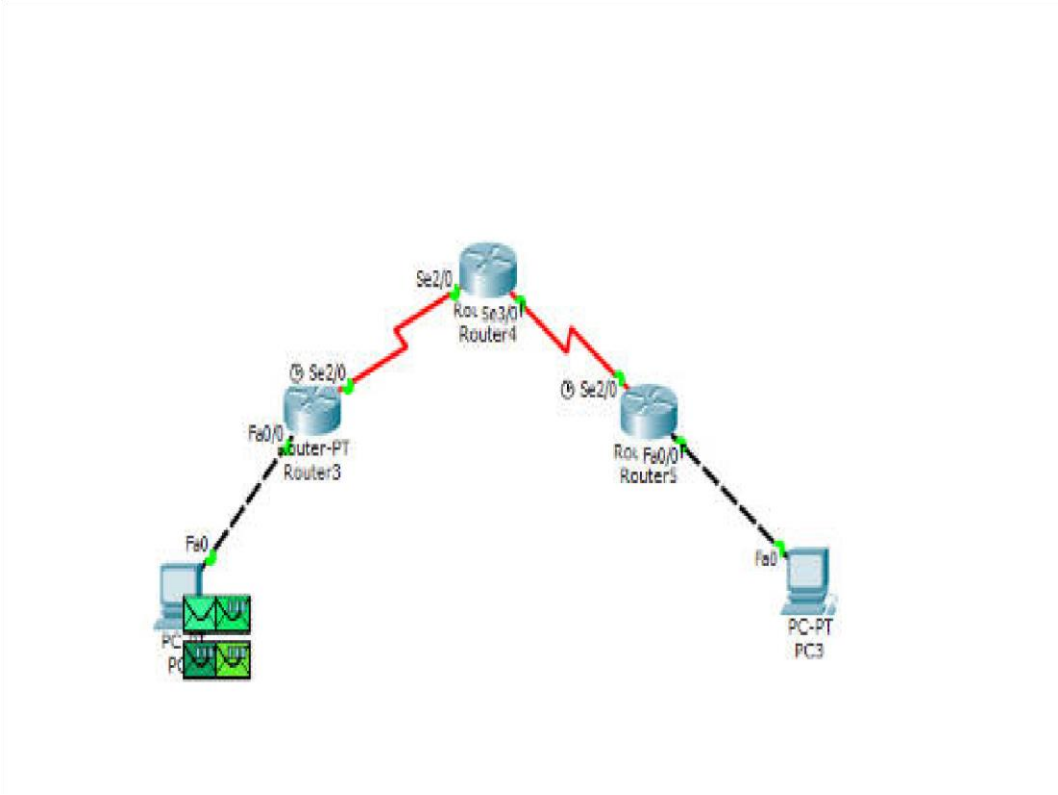
By using the "line vty 0 5" command virtual access to the router and the no. of usrs having this access can be set. The "enable secret" command is used to the password for enabling the router & under r1(config-line)# "password" command line is used to set login password.

Experiment No. 12

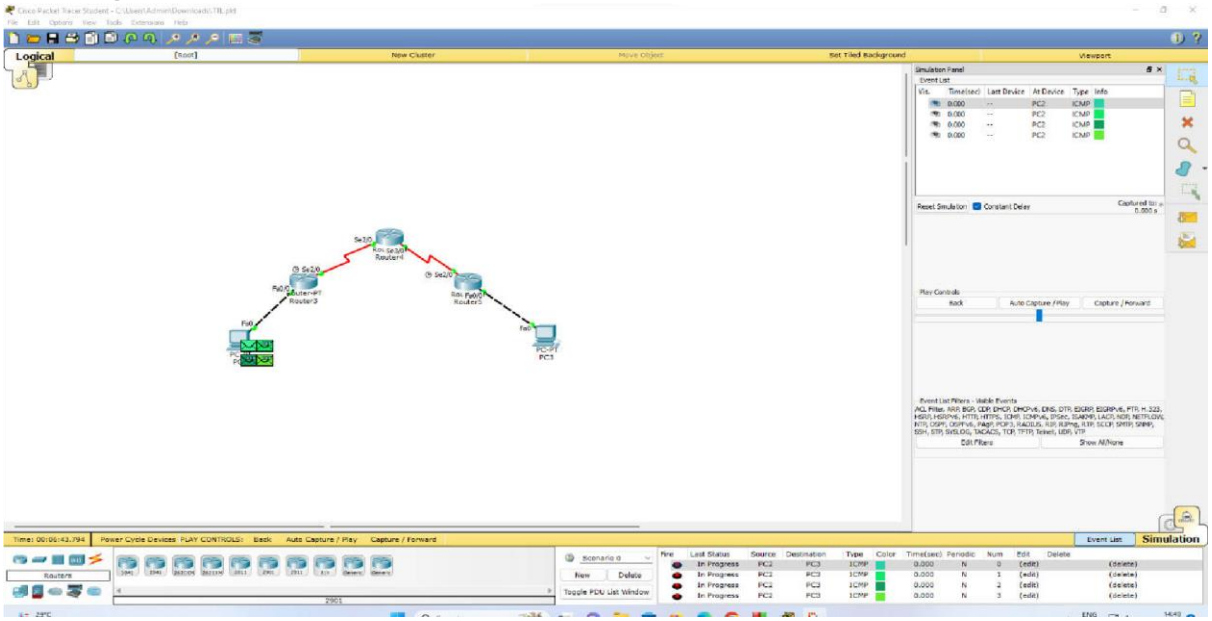
Title:

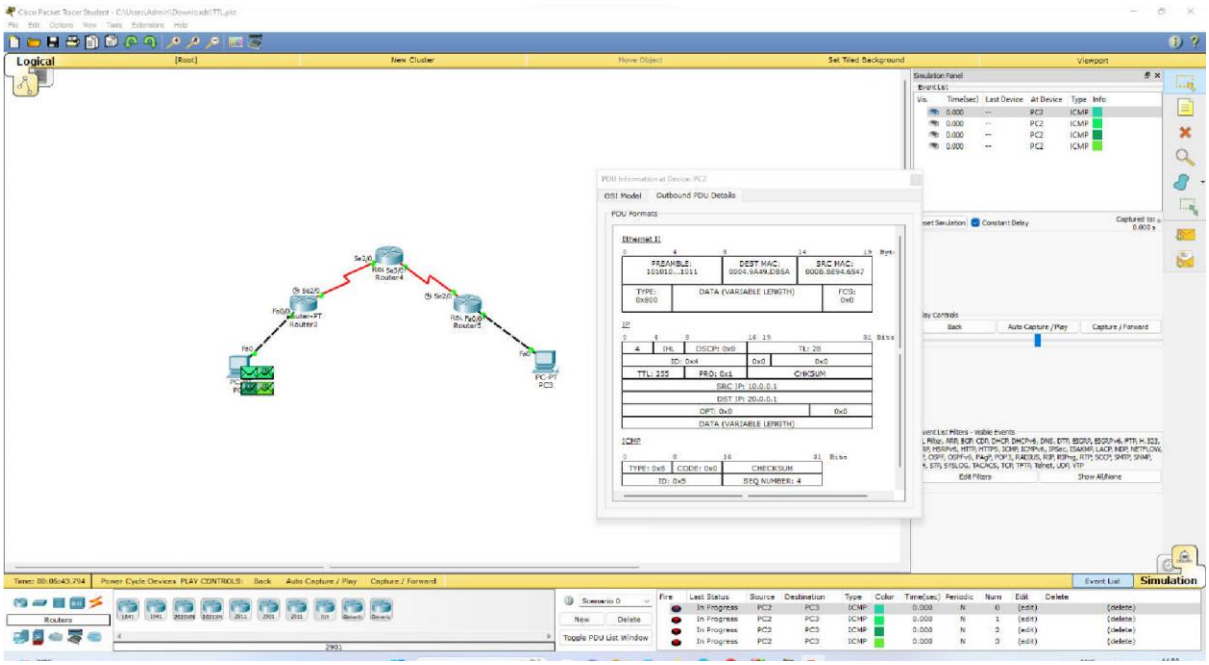
Demonstrate the TTL/ Life of a Packet

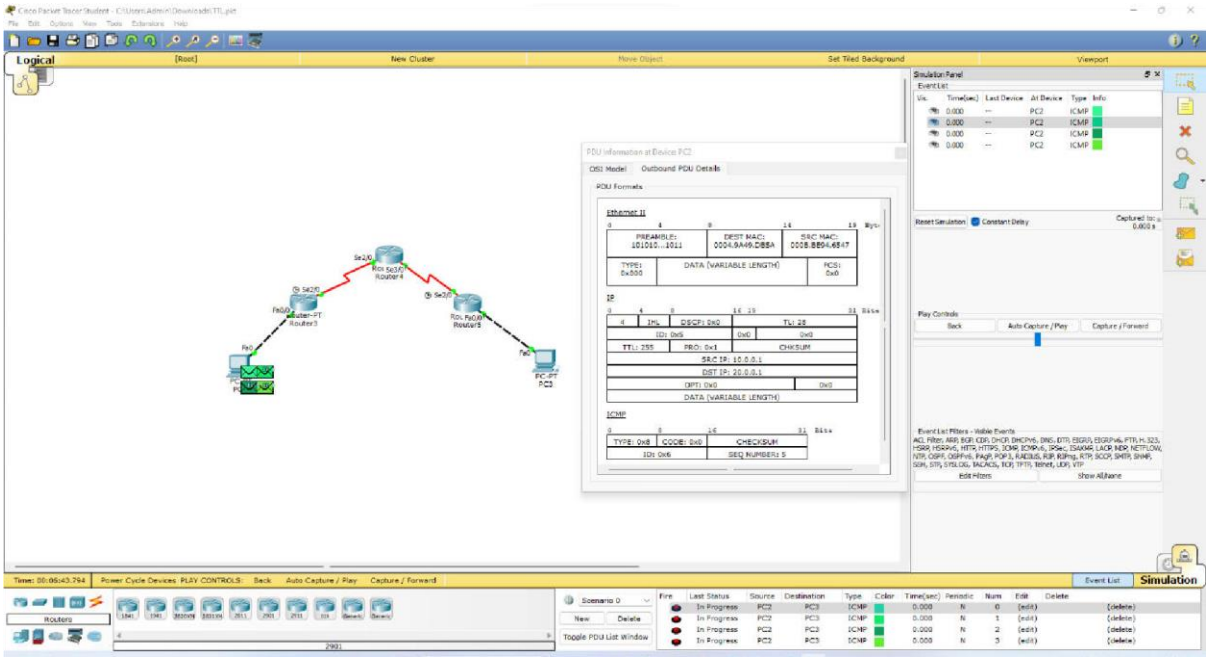
Topology:

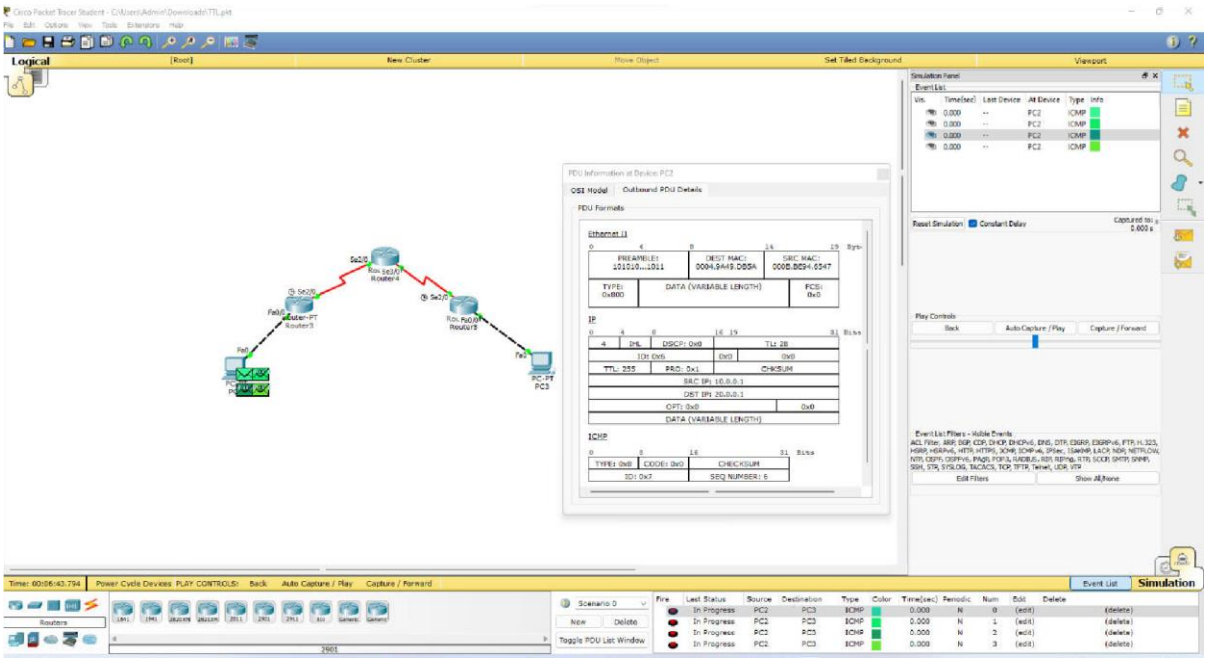


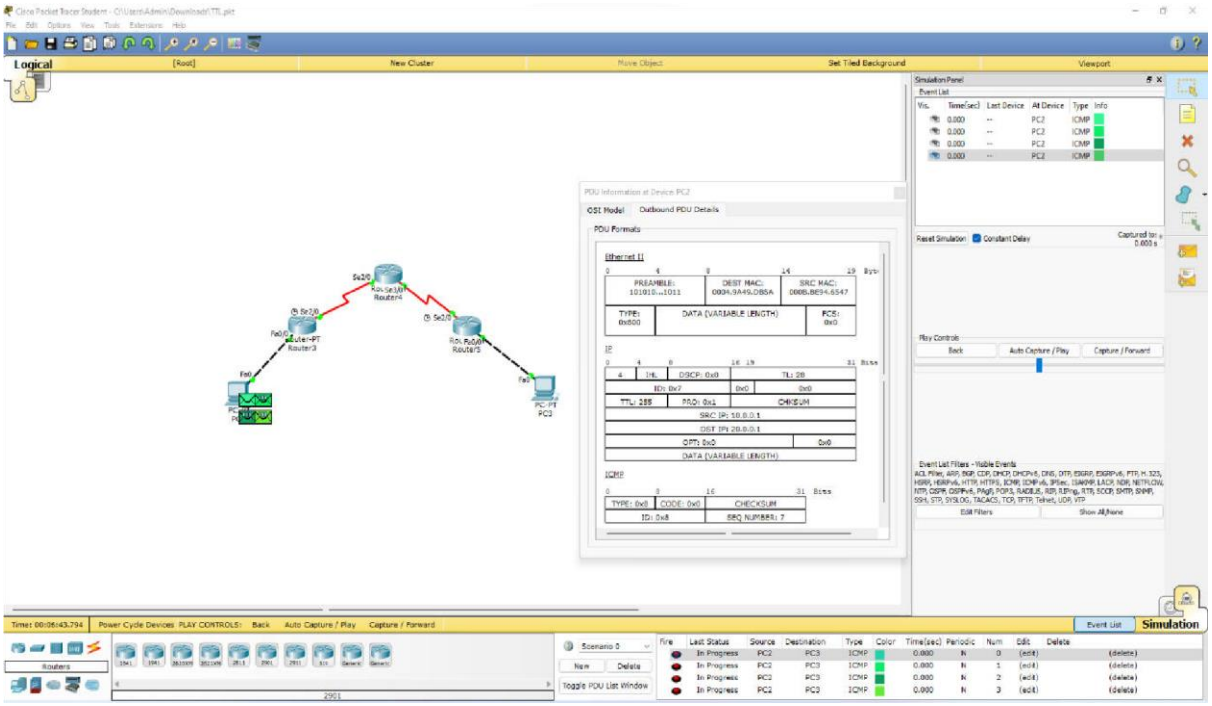
Sending PDU from one PC to another:

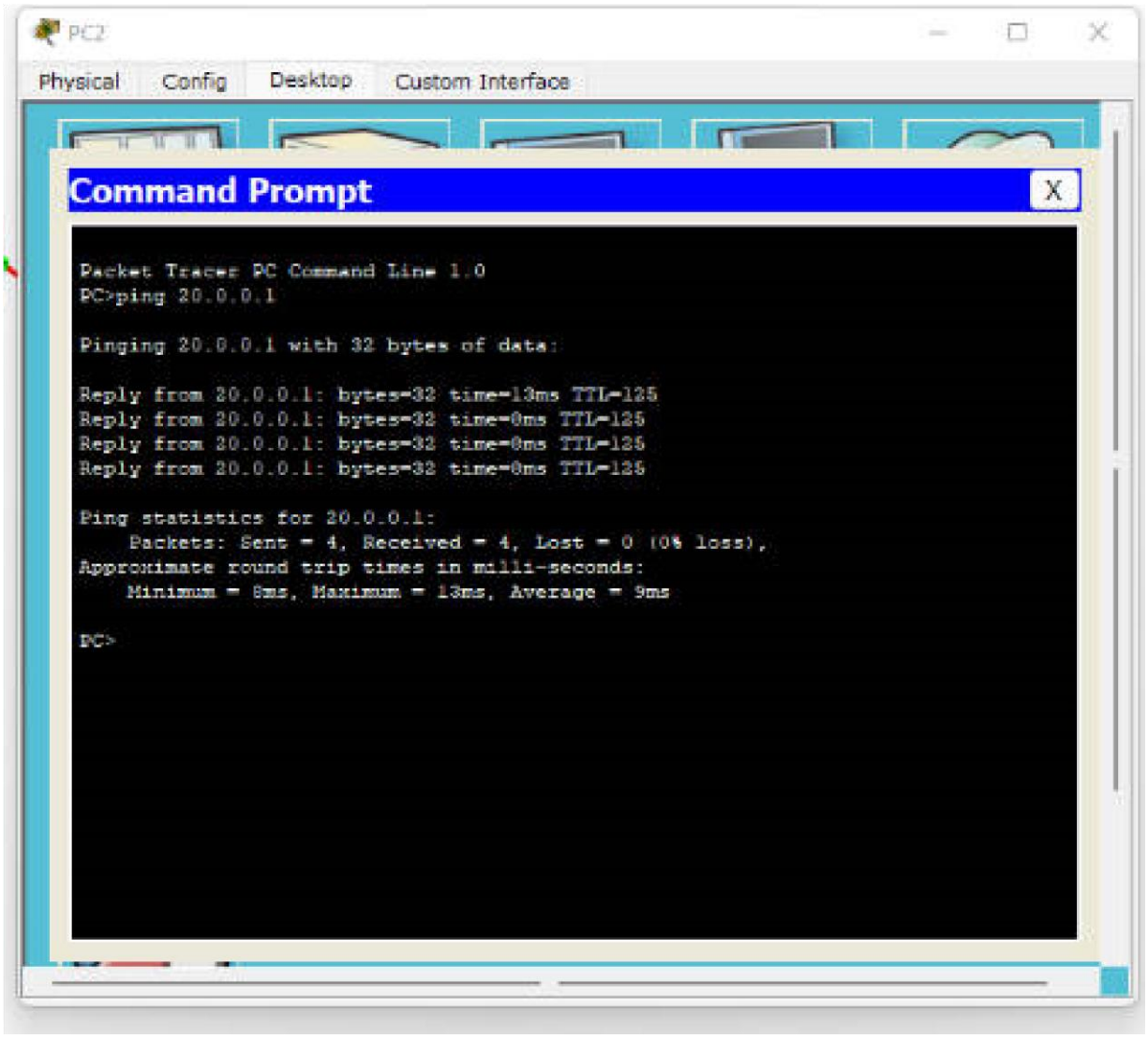






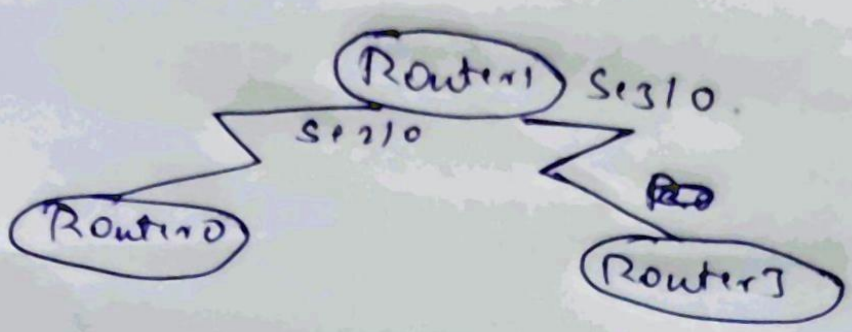


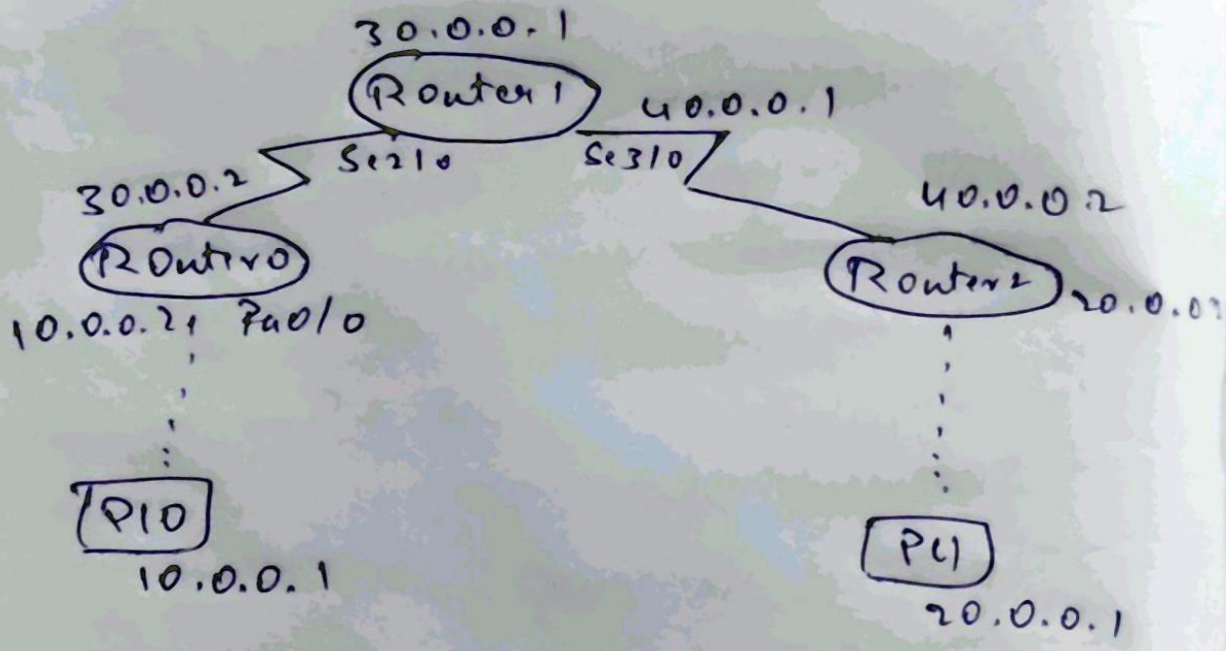



PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=13ms TTL=125
Reply from 20.0.0.1: bytes=32 time=0ms TTL=125
Reply from 20.0.0.1: bytes=32 time=0ms TTL=125
Reply 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>

(11) Title: Demonstrate the TTL/life of a Packet.

Aim: To understand TTL/life of a packet.





Procedure:

(1). Create a topology with 2PCs & 3 routers as shown above.

(2). Configure IP address as 10.0.0.1 and 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 fa0/0
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 shutdown
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 Router 1 & Router 2.

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 Router0:
Inbound PDU details - TTL = 255
Outbound PDU details - TTL = 254

PDU info at Router1:
Inbound PDU details - TTL = 254
Outbound PDU details - TTL = 253

PDU info at Router2:
Inbound PDU details - TTL = 253
Outbound 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
```

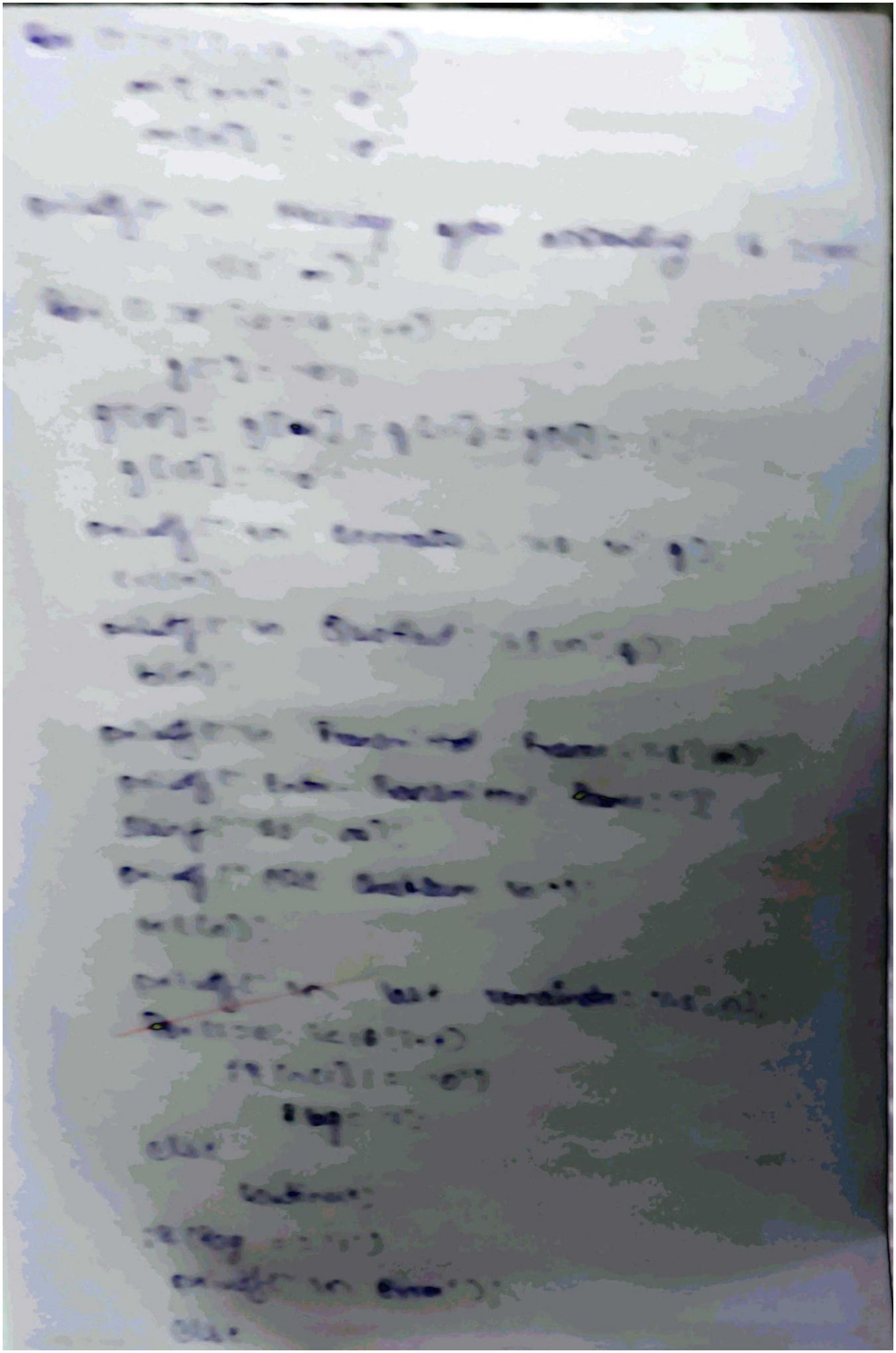
```
Enter the frame bits:1011
Message after appending 16 zeros:1011000000000000000
generator:10001000000100001


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


last remainder:0001000000100001 Error during transmission
```

CYCLE-8

Q. Write a program for error detection codes using CRC-CCITT

```
#include<stdio.h>
char m[50], g[50], r[50], q[50], temp[50];

void krt(int n)
{
    int i,j;
    for(i=0; i<n ;i++)
        temp[i]= m[i];
    for(i=0; i<17; i++)
        r[i]= m[i];
    for(i=0; i< n-17; i++)
    { if (r[0]=='1')
            q[i]='1';
            c();
    }
    else
        q[i]='0';
        Shift();
    }
    r[16]= m[i+17];
    r[17]='\0';
    for(j=0; i<=17; i++)
        temp[j]= r[j];
    }.
    q[n-17]='\0';
}
```

```
void a()
{
    int i,j;
    for (i=1; i<=10; i++)
    {
        n[i-1] = (int)temp[i] - 48] ^ (int)g[i] - 48]
                 + 48;
    }
}

void shift()
{
    int i;
    for(i=1; i<=16; i++)
        n[i-1] = n[i];
}

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

int main()
{
    int n, i=0;
    char ch, flag=0;
    printf("\n enter the frame bits: ");
    while( (ch = getc (stdin)) != '\n')
        m[i++] = ch;
    n = i;
```



```
    printf(" Frames are correct");
}
```

Output:

```
Enter Frame bits: 1011
Message after appending 16 zeroes
1011 0000 0000 0000 0000
generates: 1001 0000 0010 0001
  Quotient: 1011
transmitted: 1011 1011 0001 0110 1011
Enter transmitted frame
  1011 1011 0001 0110 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
```

(2). WAP for congestion control using leaky bucket algorithm.

=>
```
#include<stdio.h>

void main()
{
    int incoming, outgoing, bucketsize, n, store=0;
    printf("\n Enter bucket size outgoing rate
            and no. of \n");
    scanf("%d %d %d", &bucket_size, &outgoing, &n);

  while(n != 0)
  {
    printf("\n Enter the incoming packet");
    scanf("%d", &incoming);

    printf("\n Incoming packet size %d\n", incoming);
    if (incoming <= (bucket-size))
    {
        store += incoming;
        printf("Bucket buffer size %d out of
                %d", store, bucket_size);
    }
    else
    {
        printf("Dropped %d no. of packets\n",
                incoming-(bucket_size));
        printf("Bucket buffer size %d out of
                %d", store, bucket_size));
```

```
            store = bucketsize;
        }
        store -= outgoing;
        printf("After outgoing %d packet left
            out %d in buffer", store,
                bucketsize);
        n--;
    }
}
```

Output:

Enter bucket size, outgoing store and no of ip
20, 10, 2.

Enter the Incoming packet_size : 30
Incoming packet size 30
Dropped 10 number of packets
Bucket buffer size 0 out of 20
After outgoing 10 packets left out 20 in buffer

Enter the incoming packet size : 10
Incoming packet size : 10
~~patch~~ Bucket buffer size 10 out of 20
After outgoing 10 packets left and 20 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:

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerTCP.py
The server is ready to receive
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ClientTCP.py

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
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerTCP.py
The server is ready to receive

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

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

=> ClientUD.py

```
From socket import *
ServerName = "127.0.0.1"
Serverport = 12000
Clientsocket = Socket(AF_INET, SOCK_DGRAM)
Sentence = input(" In enter file name:")
Clientsocket.sendto(bytes(sentence, "utf-8"),
                    (Sentence, Serverport)
filecontents, serveraddress = clientsocket.recvfrom(2048)
print(" In Reply from Server: In")
print(" filecontents.decode("utf-8"))
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(
Sentence = Sentence.decode("utf-8")
File = open(sentence, "r")
Con = File.read(2048)
ServerSocket.bind("127.0.0.1")
ServerSocket.sendto(bytes(con,"utf-8"),Clientaddress)
print("In sort contents of", end=':')
print(Sentence)
File.close()
```

Output

ServerUDP.py

The Server is ready to recieve
Sent contents of ServerUDP.py
The Server is ready to recieve.

ClientUDP.py

```
Enter File name : ServerUDP.py
Reply From Server:
From socket import *
Serverport = 12000
From socket import *
Serverport = 12000
Serversocket = socket(AF_INET, SOCK-
                      DGRAM)
```

```
Sentence = Sentence.decode("UTF_8")
File = open(Sentence,"r")
l=File.readlines()
print("In Sent contents of ", end=":")
print(sentence)
File.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:

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerUDP.py
The server is ready to receive
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ClientUDP.py

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
```

```
/Users/mac/PycharmProjects/cn-lab/venv/bin/python /Users/mac/PycharmProjects/cn-lab/ServerUDP.py
The server is ready to receive

Sent contents of  ServerUDP.py
```

Program-1

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

=> From socket import *

Servername = "127.0.0.1"

Serverport = 12000

Clientsocket = connect (" (servername, Serverport)

Sentence = input ("\n Enter file name:")

Clientsocket. Send (sentence. encode())

Print("\n From Server:")

Print(Filecontents)

ClientSocket. close()

ServerTCP.py

From socket import *

Servername = "127.0.0."

Serversocket = Socket(AF_INET, SOCK_STREAM)

Server socket. bind ((Servername, Serverport))

while 1:

print("The Server is ready to receive"

connection socket, addr = server socket. accept()

decode()

File = open(sentence, "r")

l = File. read(1024)

```
connectionsocket.send(1.encode())
print("\n Sent Contents of ", sentence)
file.close().
```

Output:

SERVERTCP.py

```
The Server is ready to resolve
Such Contents of ServerTCP.py
The Server is ready to recieve.
```

9/9/23