VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Sanketh M Hanasi (1BM23CS242)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Academic Year 2024-25 (odd)

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 "Computer Network (23CS5PCCON)" carried out by **Sanketh M Hanasi (1BM23CS242),** who is Bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Ramya K M Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index-Cycle-I

Github Link:
github.com/SankethHanasi/CN_1BM22CS242

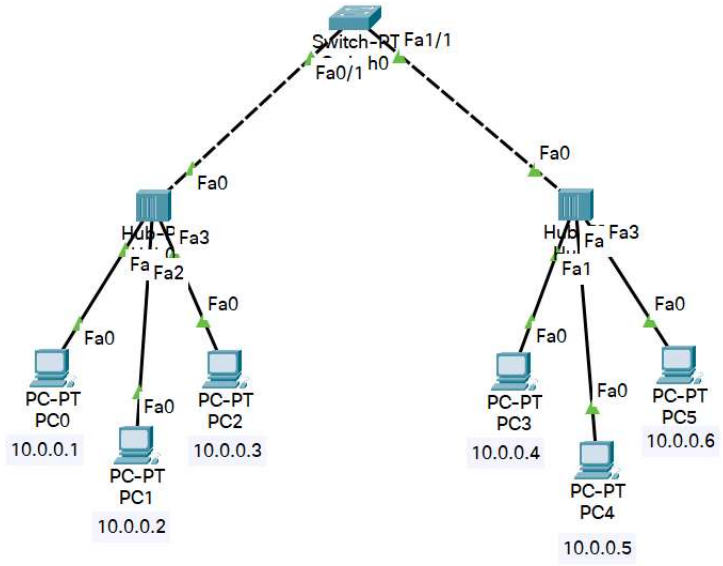
Index-Cycle-II

Cycle-I

Program 1

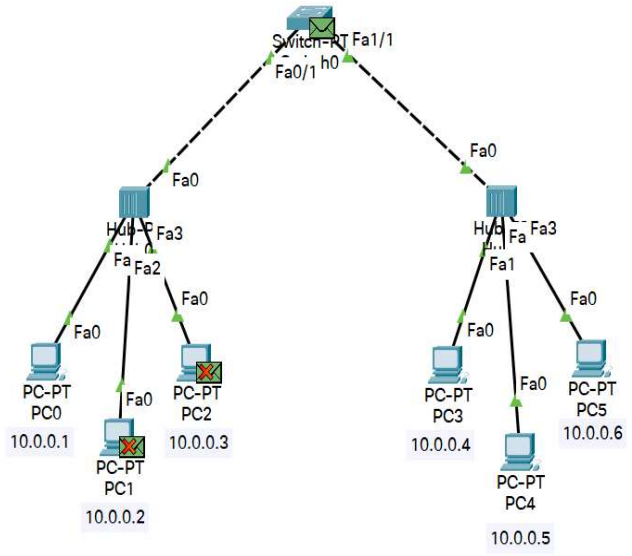
i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.

ii. Procedure along with the topology

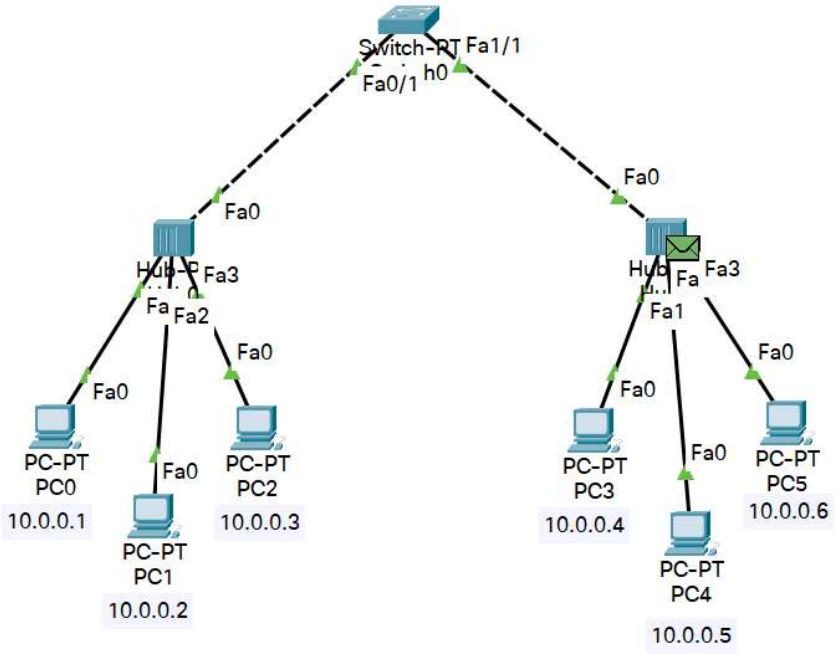


iii. Screen shots/ output

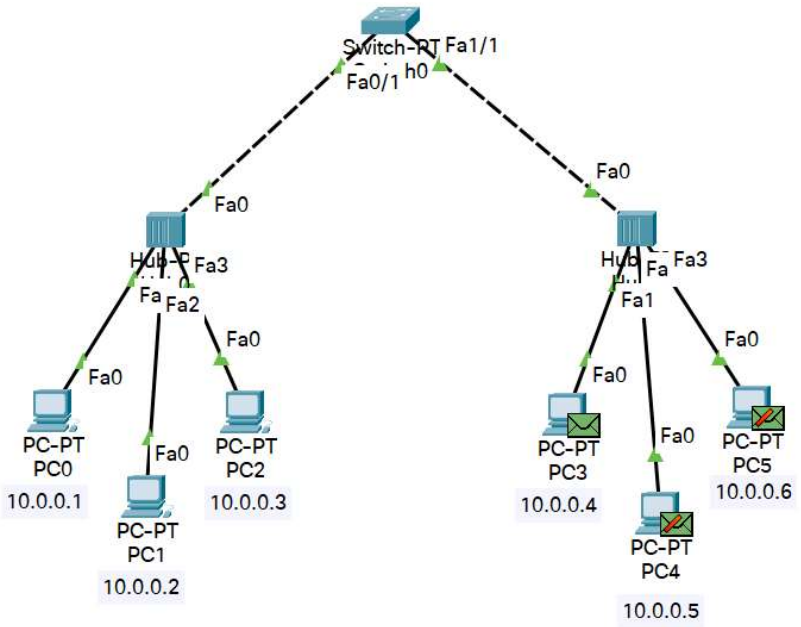
Hub behaviour at sending end



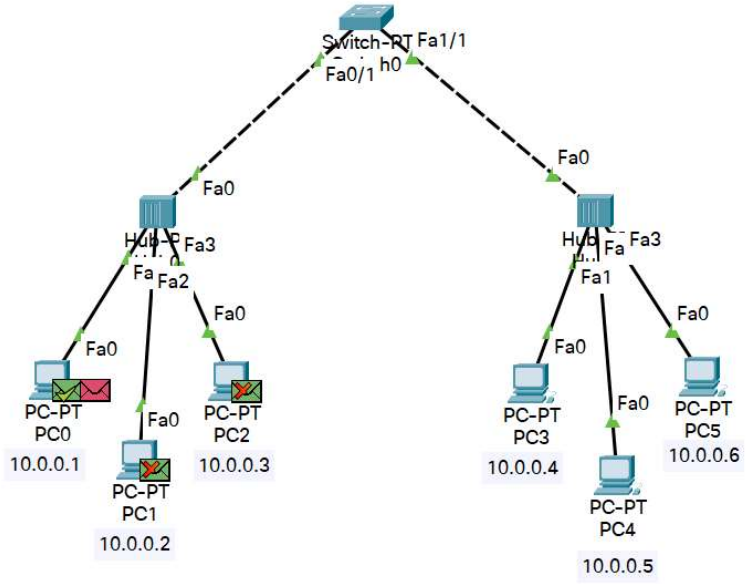
Switch behaviour



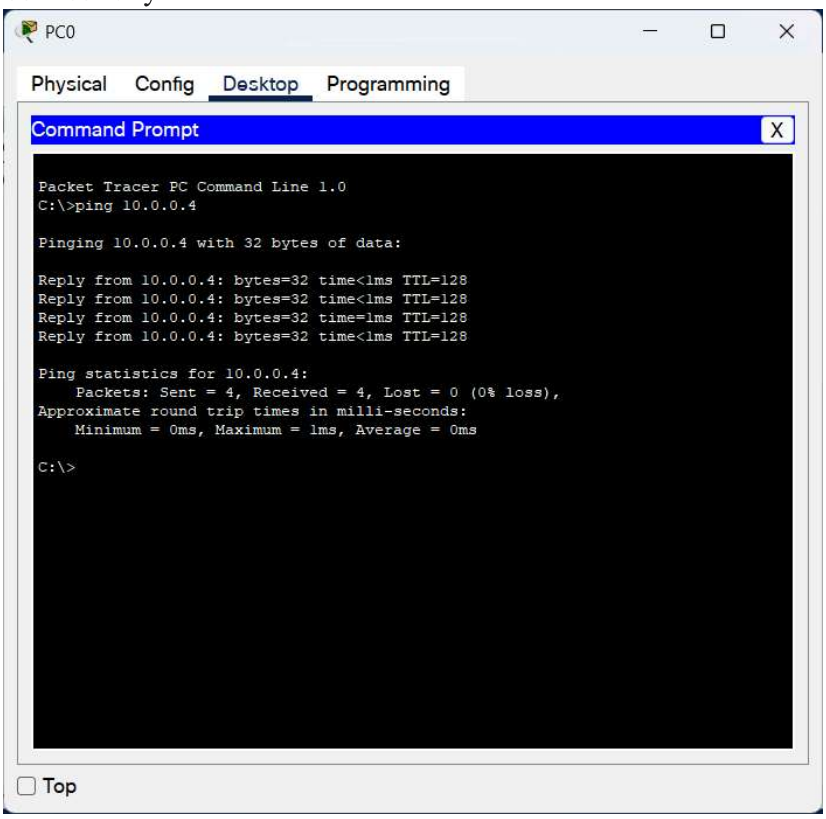
Hub behaviour at receiving end



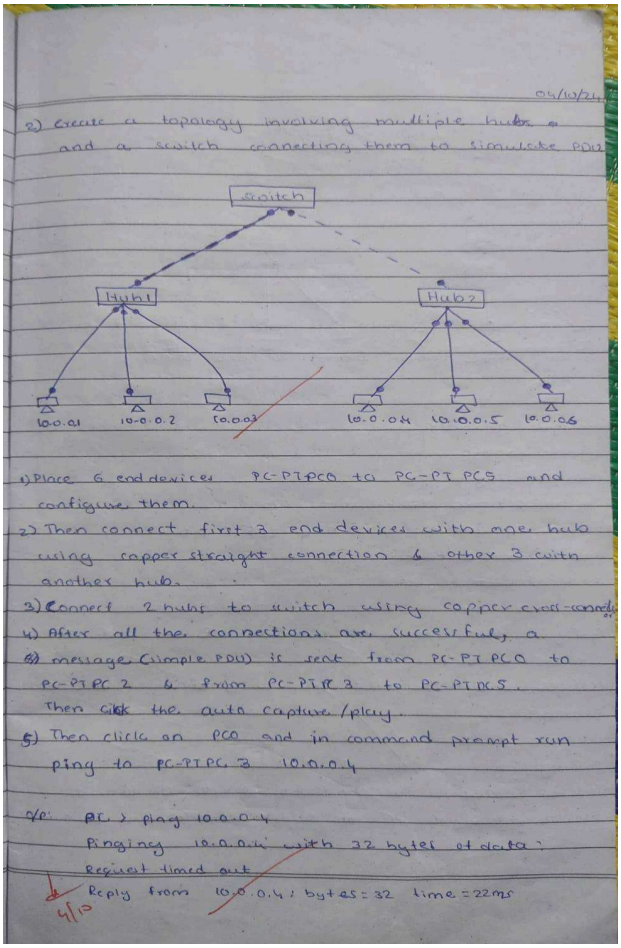
Hub behaviour when back to sender



Ping command to connectivity



iv. Observation



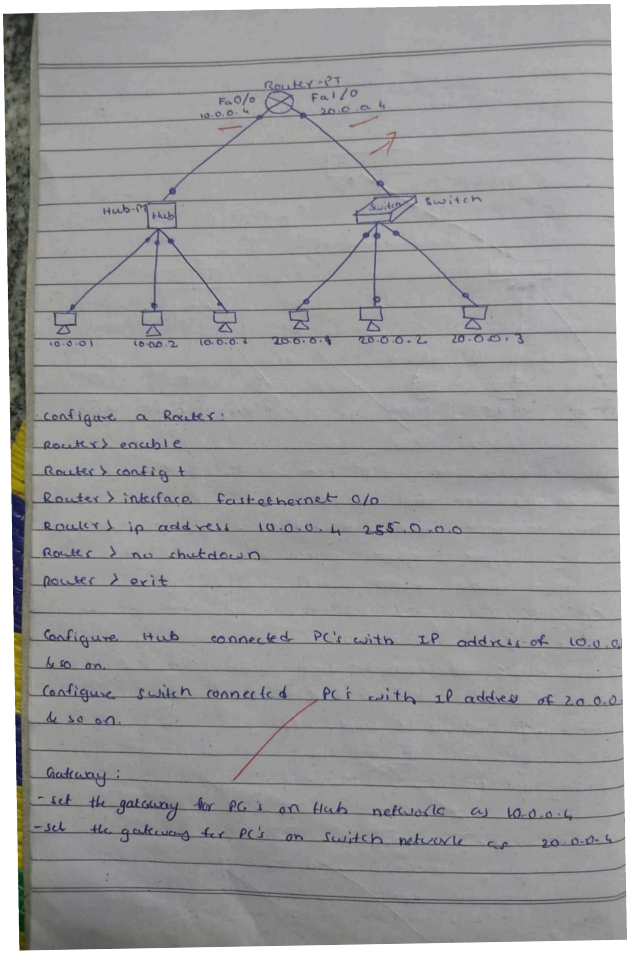
04/10/24

2) Create a topology involving multiple hubs and a switch connecting them to simulate PDU

1) Place 6 end devices PC-PT PC0 to PC-PT PC5 and configure them.
2) Then connect first 3 end devices with one hub using copper straight connection & other 3 with another hub.
3) Connect 2 hubs to switch using copper cross-connect
4) After all the connections are successful, a
5) message (simple PDU) is sent from PC-PT PC0 to PC-PT PC 2 & from PC-PT PC 3 to PC-PT PC 5.
Then click the auto capture /play.
6) Then click on PC0 and in command prompt run ping to PC-PT PC 3 10.0.0.4

o/p: PC > ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Request timed out
Reply from 10.0.0.4: bytes = 32 time = 22ms

4/10



Configure a Router:
Router> enable
Router> config t
Router> interface fastethernet 0/0
Router> ip address 10.0.0.4 255.0.0.0
Router > no shutdown
Router > exit

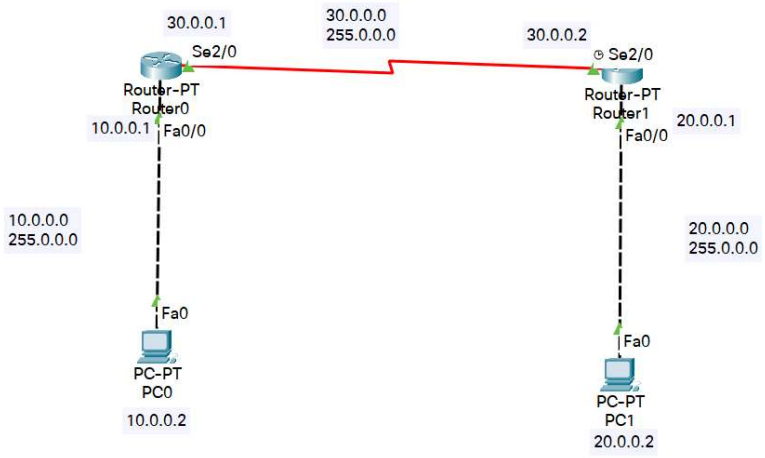
Configure Hub connected PC's with IP address of 10.0.0
& so on.
Configure switch connected PC's with IP address of 20.0.0
& so on.

Gateway:
- set the gateway for PC's on Hub network as 10.0.0.4
- set the gateway for PC's on switch network as 20.0.0.4

Program 2

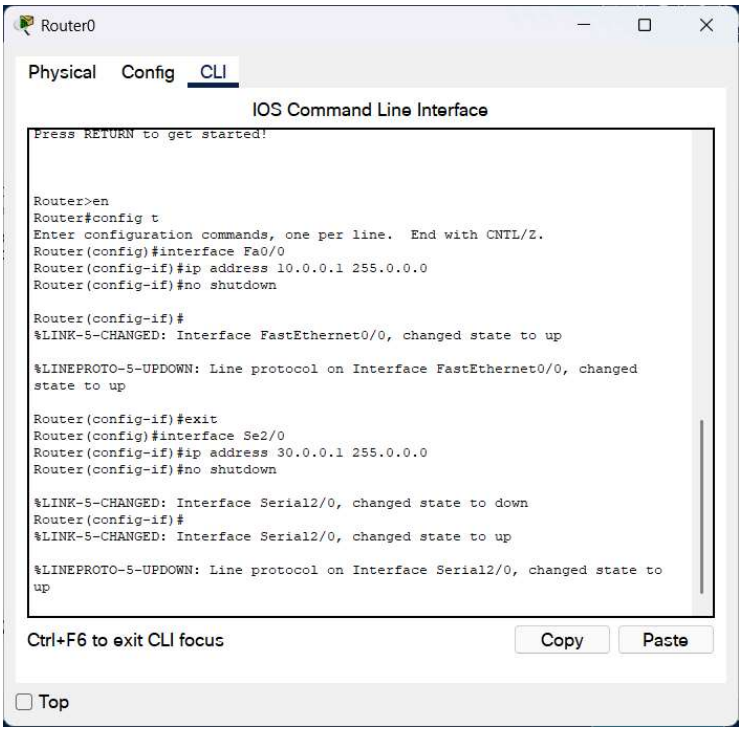
i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.

ii. Procedure along with the topology

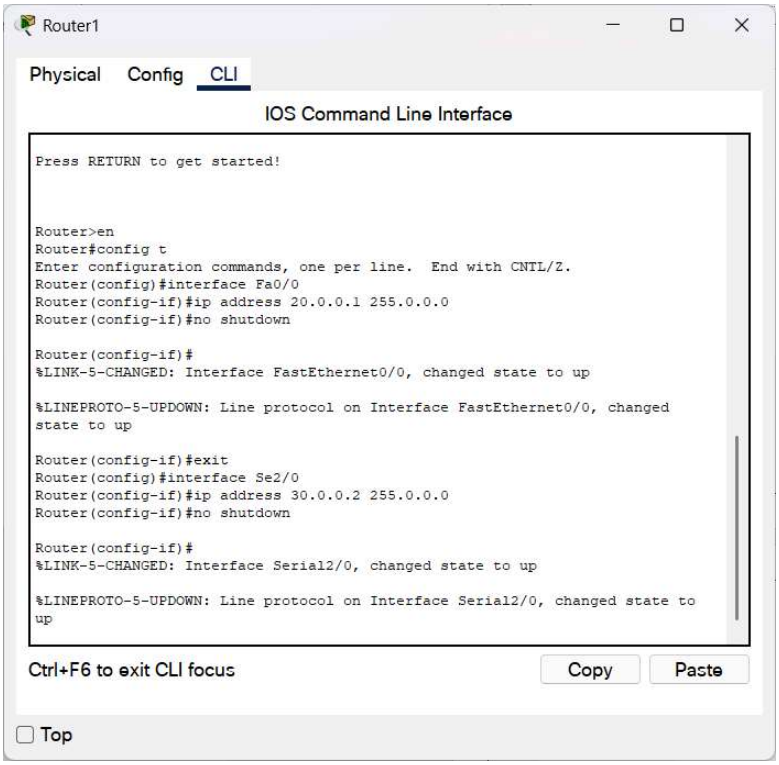


iii. Screen shots/ output

Router0 configuration



Router1 configuration



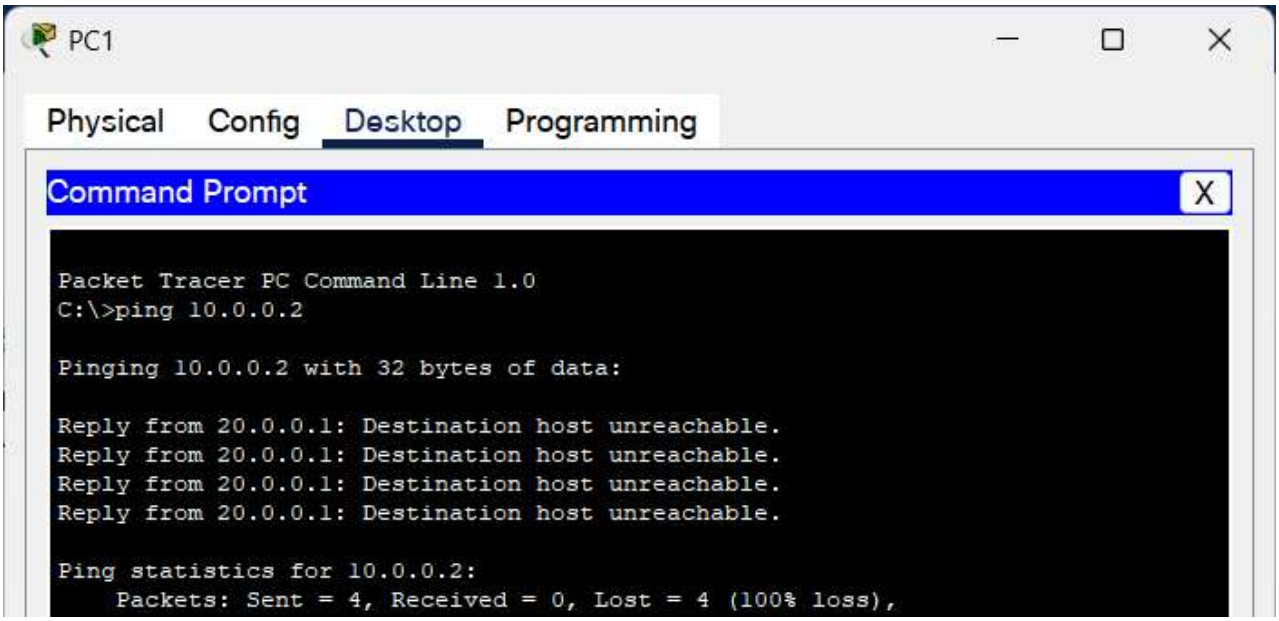
Ip route command in Router0

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.2
Router(config)#
```

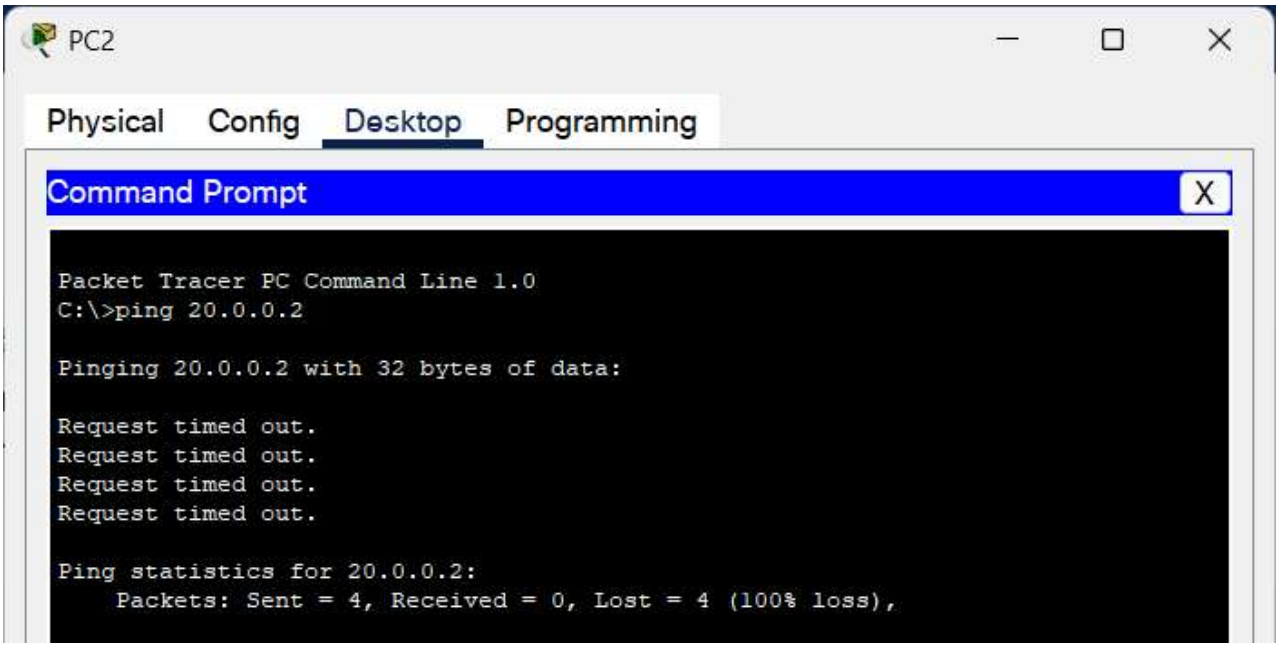
Ip route command in Router1

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#
```

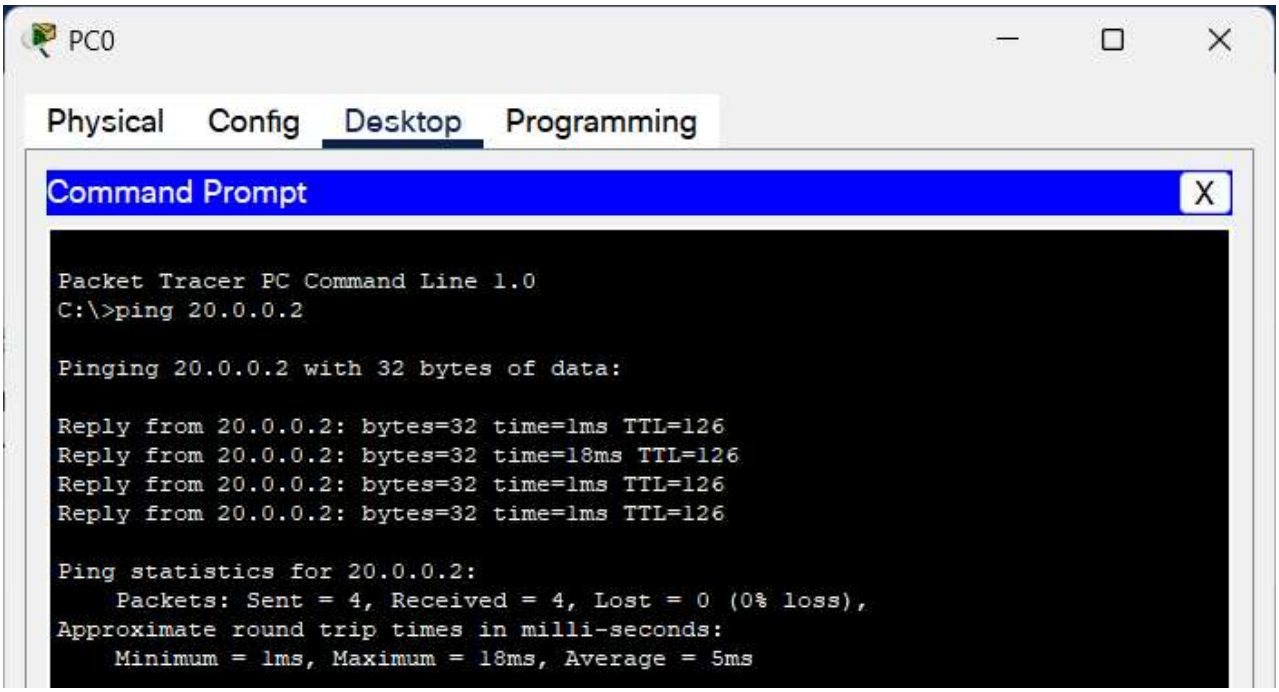
Destination host Unreachable (Before establishing network Fully)



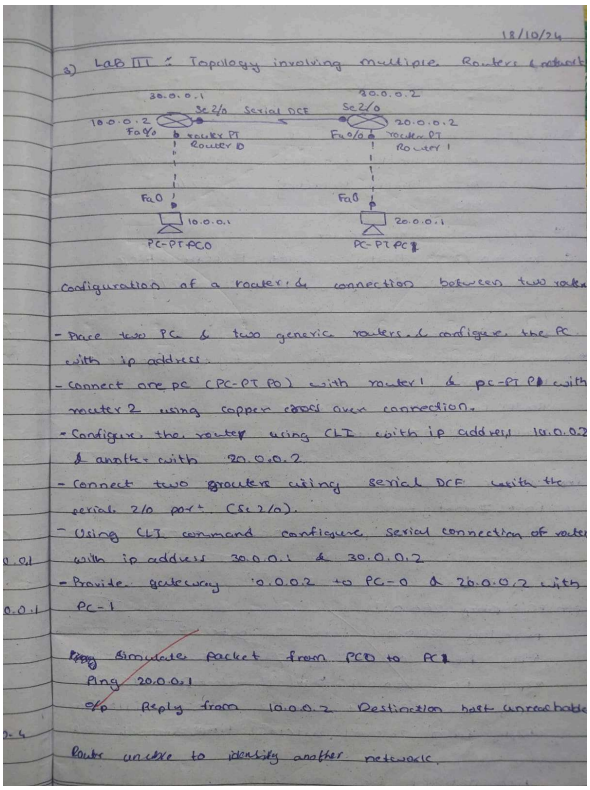
Request Timed Out

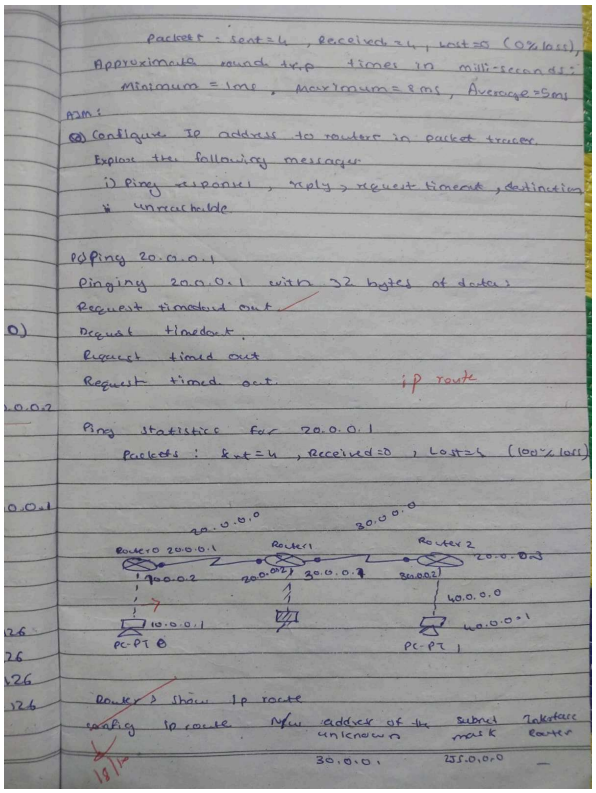


Reply from Destination



iv. Observation

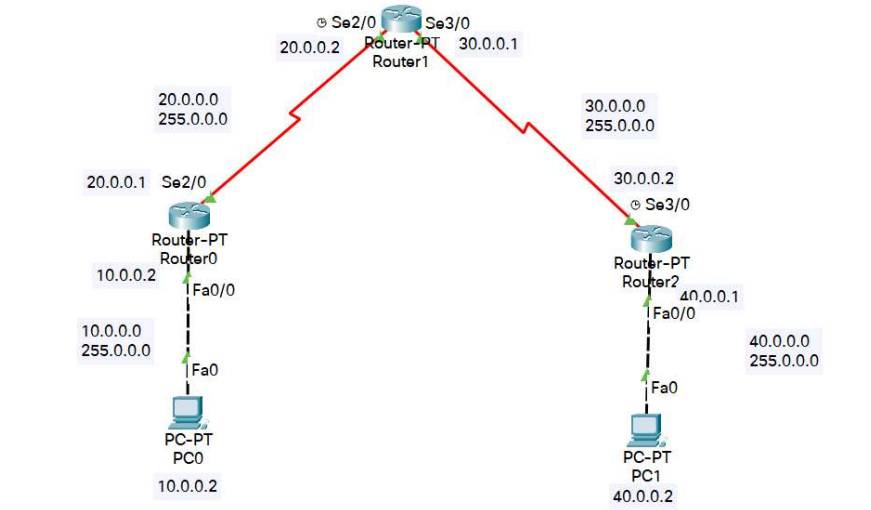




Program 3

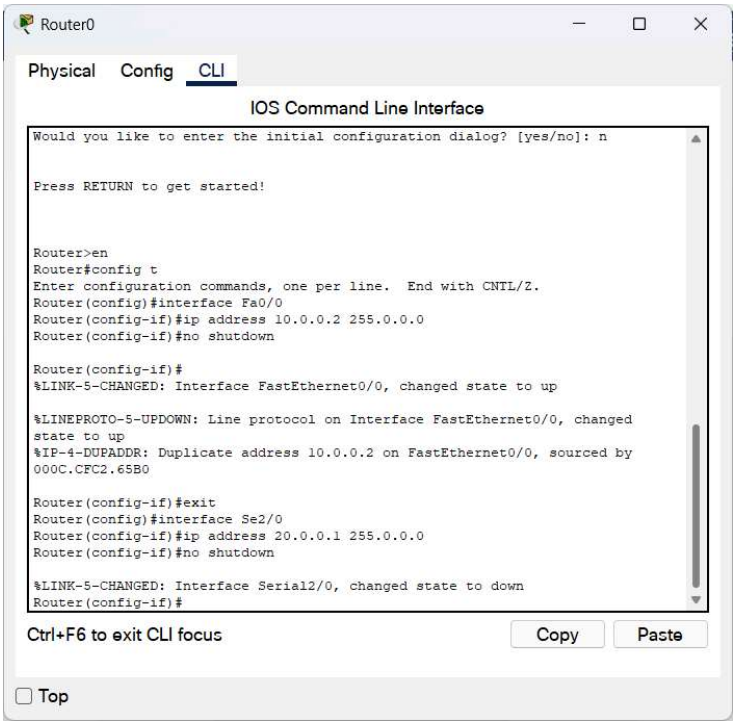
i. Configure default route, static route to the router

ii. Procedure along with the topology

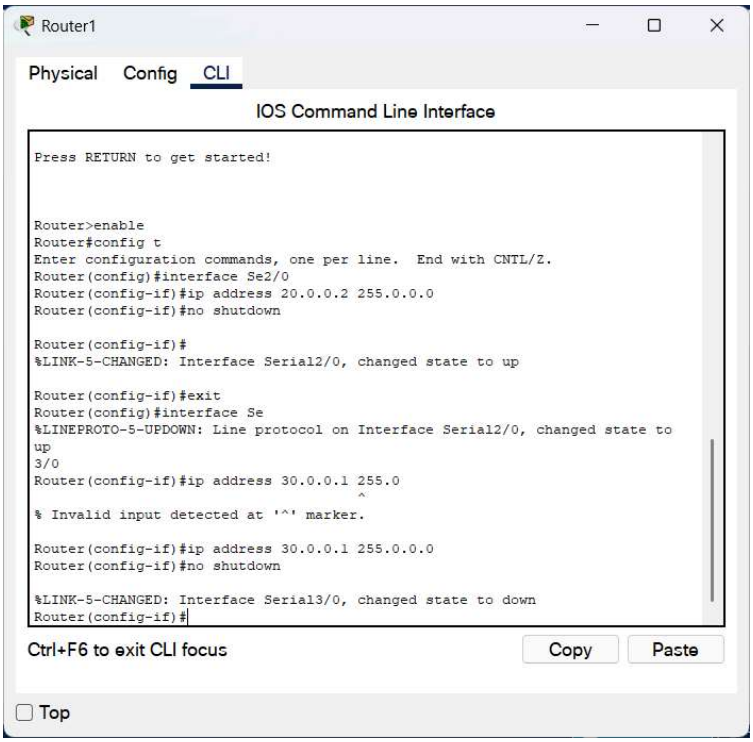


iii. Screen shots/ output

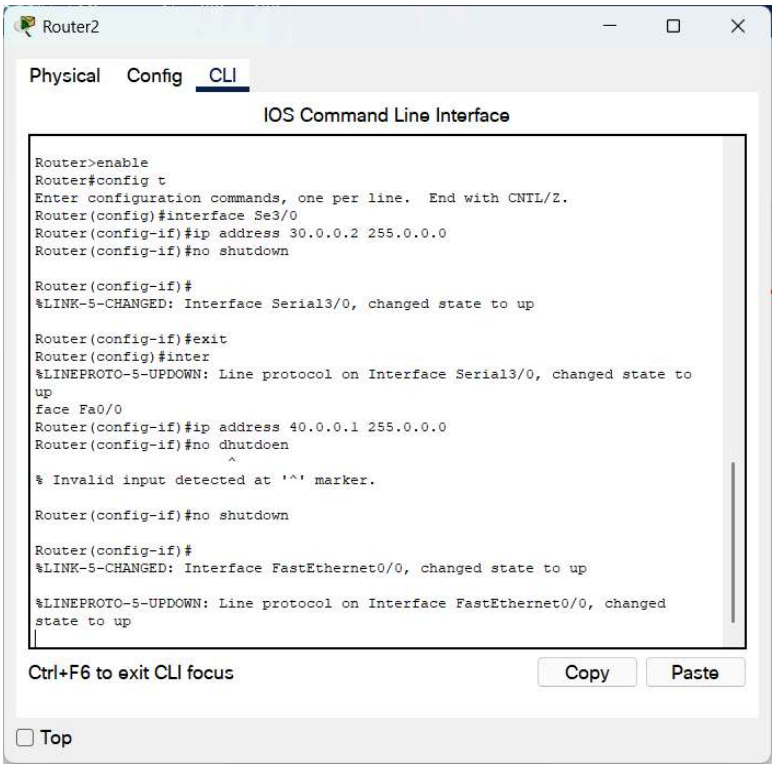
Router0 configuration



Router1 configuration

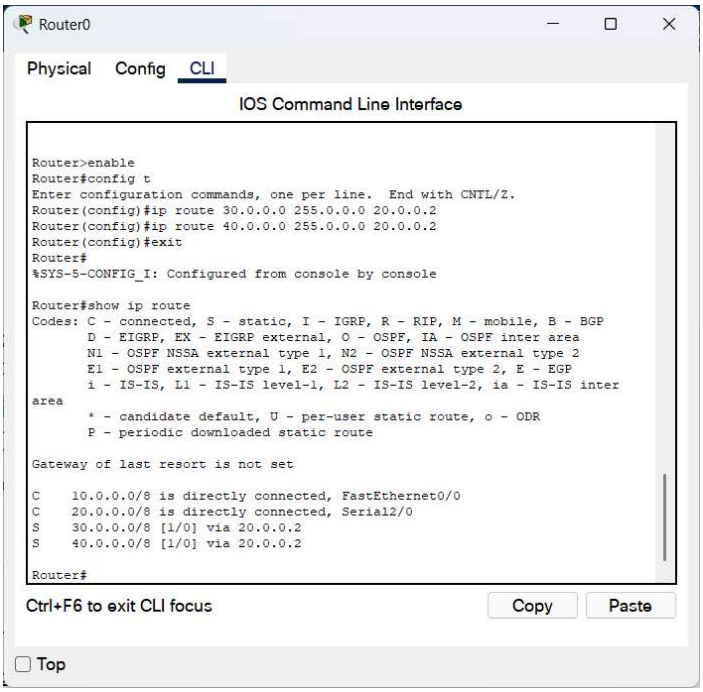


Router2 configuration

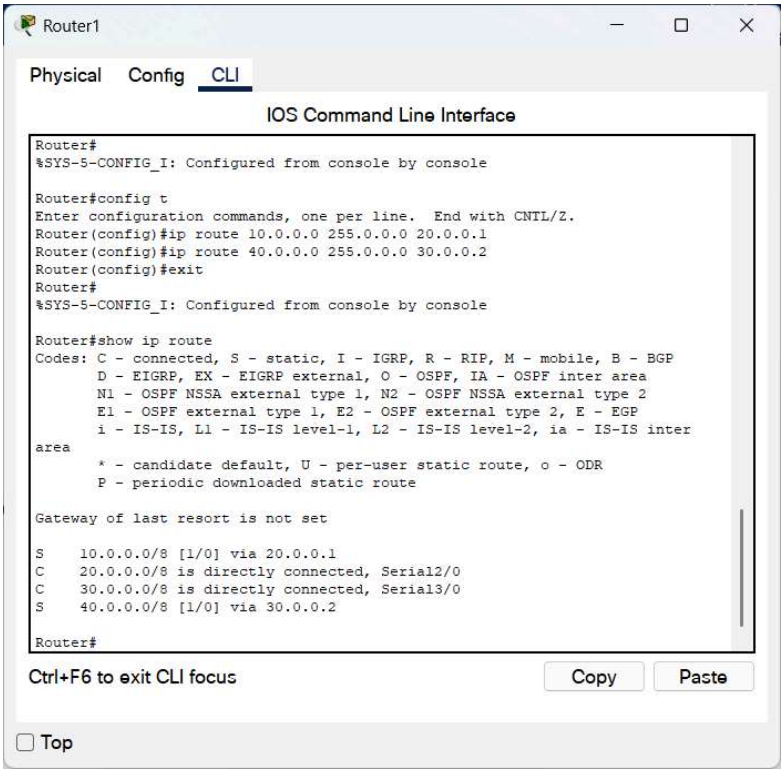


Static Routing:

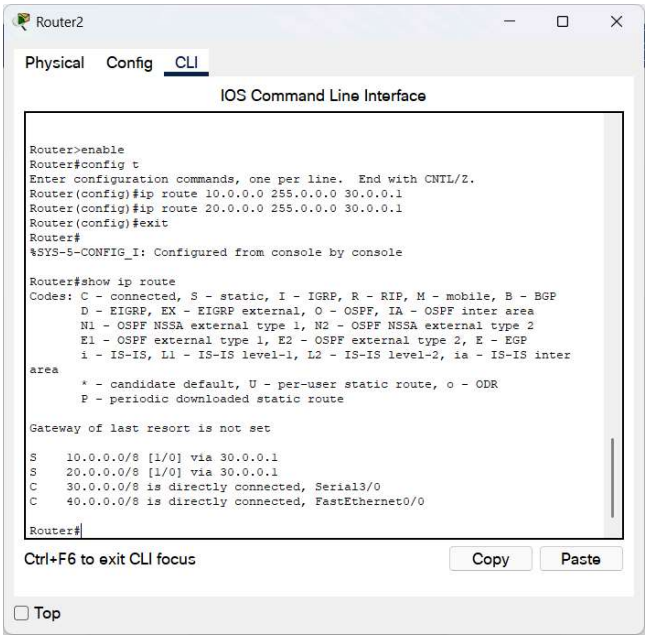
Router0



Router1

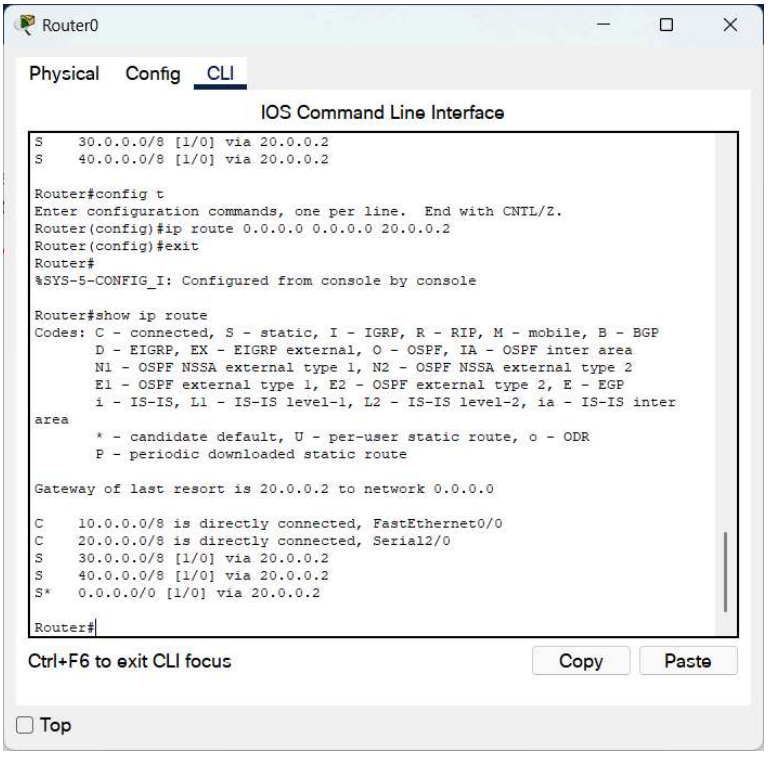


Router2

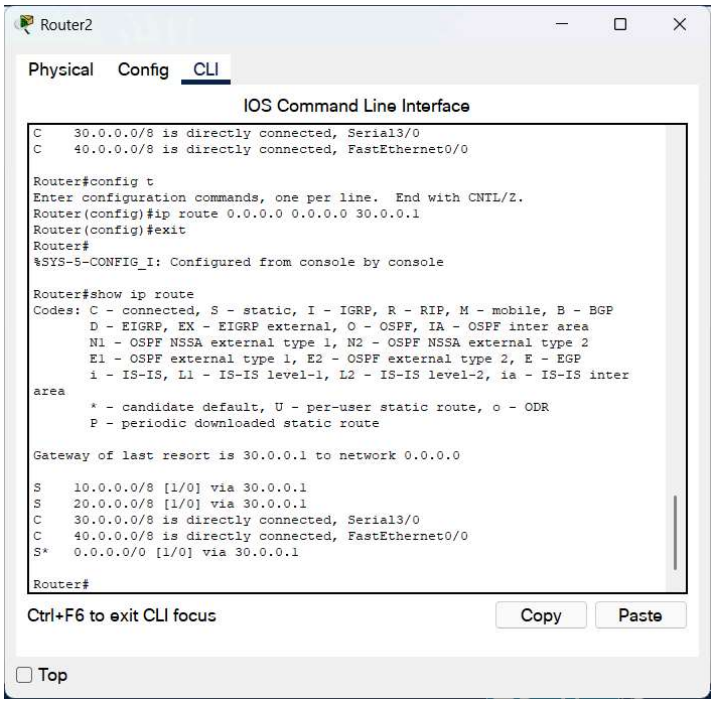


Dynamic Routing:

Route0



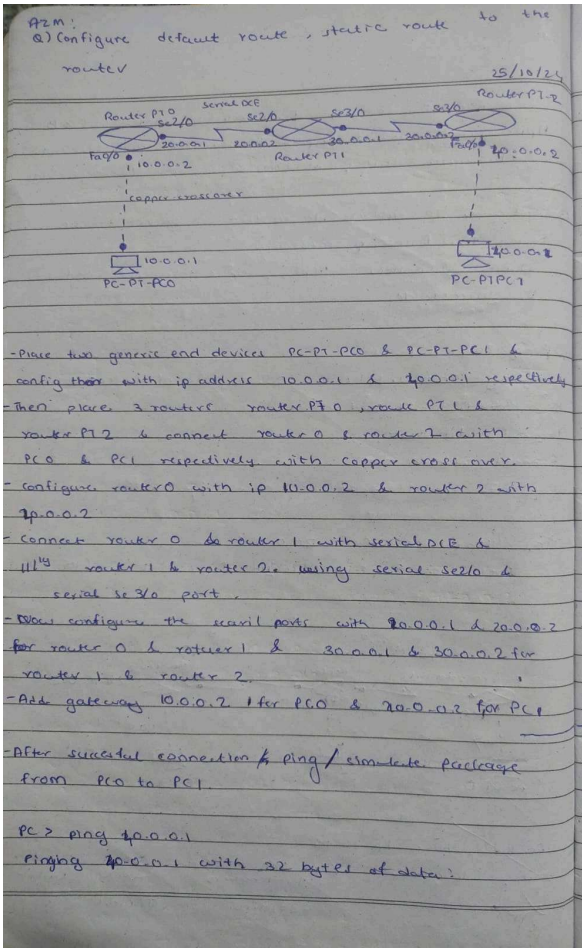
Router2



Pinging:

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=21ms TTL=125
Reply from 40.0.0.2: bytes=32 time=17ms TTL=125
Reply from 40.0.0.2: bytes=32 time=25ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 16ms
```

iv. Observation



AIM:
Q) Configure default route, static route to the router

25/10/24

- Place two generic end devices PC-PT-PC0 & PC-PT-PC1 & config their with ip address 10.0.0.1 & 40.0.0.1 respectively
- Then place 3 routers router PT 0, router PT 1 & router PT 2 & connect router 0 & router 1 with PC 0 & PC 1 respectively with copper cross over.
- configure router0 with ip 10.0.0.2 & router 2 with 40.0.0.2
- Connect router 0 & router 1 with serial DCE & router 1 & router 2. using serial se2/0 & serial se 3/0 port.
- Now configure the serial ports with 20.0.0.1 & 20.0.0.2 for router 0 & router 1 & 30.0.0.1 & 30.0.0.2 for router 1 & router 2.
- Add gateway 10.0.0.2 for PC0 & 40.0.0.2 for PC1
- After successful connection & ping / simulate. Package from PC0 to 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=8ms TTL=125
Reply from 40.0.0.1: bytes = 32 time=9ms TTL=125
Reply from 40.0.0.1: bytes = 32 time=8ms TTL=125
Reply from 40.0.0.1: bytes = 32 time=8ms TTL=125

ping statistics for 40.0.0.1:
Packets: Sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip times in milliseconds:
Minimum = 8ms, Maximum = 15ms, Average = 8ms

-> Configure all routers with

router 0
Router(config)# ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

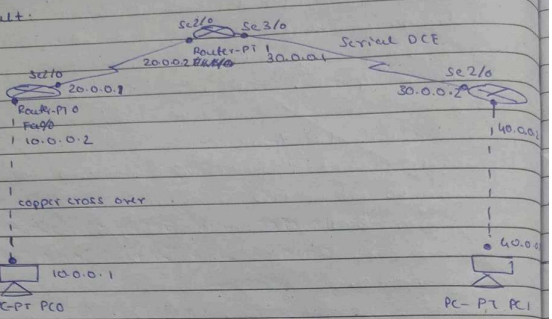
router 1
Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

router 2
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 20.0.0.1

25/10

Observation:
Default Routing is a route used by a router when it doesn't have a specific route for a destination. Static Routing involves manually setting fixed routes for specific destinations.

Default:



PC-PT PC0 - 10.0.0.1

PC-PT PC1 - 40.0.0.1

Router-PT 0 -> FastEtherNet Fa0/0 10.0.0.2

Router-PT 1 -> FastEthernet Fa0/0 40.0.0.2

PC0 -> gateway -> 10.0.0.2

PC1 -> gateway -> 40.0.0.2

PC> ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Ping statistics for 40.0.0.1

Packets: sent = 4 Received = 0, Lost = 4 (100% loss).

PC> ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data:

Reply from 10.0.0.2 Destination host unreachable

Reply from 10.0.0.2 Destination host unreachable

Reply from 10.0.0.2 Destination host unreachable

Reply from 10.0.0.2 Destination host unreachable

Router 0:

Router # show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0

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:

Router # show ip route

C 40.0.0.0/8 is directly connected, FastEthernet 0/0

C 30.0.0.0/8 is directly connected, Serial 2/0

S* 0.0.0.0/0 [1/0] via 30.0.0.2

Router 1:

Router # show ip route

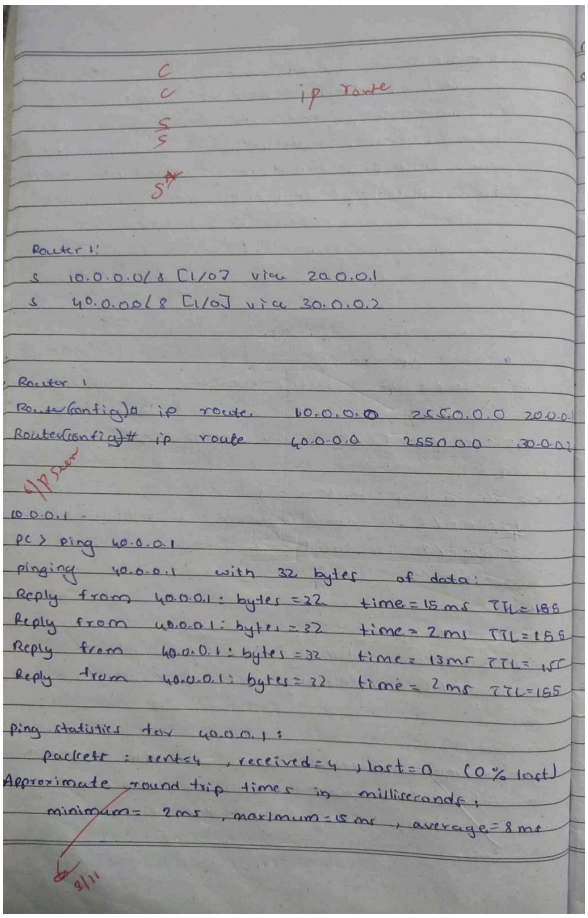
C 20.0.0.0/8 is directly connected, Serial 2/0

C 30.0.0.0/8 is directly connected, Serial 3/0

Router 0:

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.1



C
C
S
S
S

ip route

Router 1:
S 10.0.0.0/8 [1/0] via 20.0.0.1
S 40.0.0.0/8 [1/0] via 30.0.0.2

Router 1
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

App Server

10.0.0.1
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=15ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

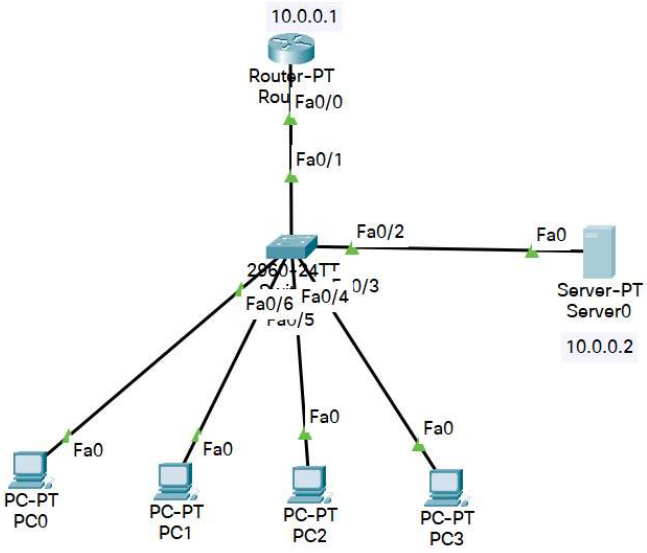
ping statistics for 40.0.0.1:
packets: sent=4, received=4, lost=0 (0% lost)
Approximate round trip times in milliseconds:
minimum=2ms, maximum=15ms, average=8ms

8/11

Program 4

i. Configure DHCP within a LAN and outside LAN.

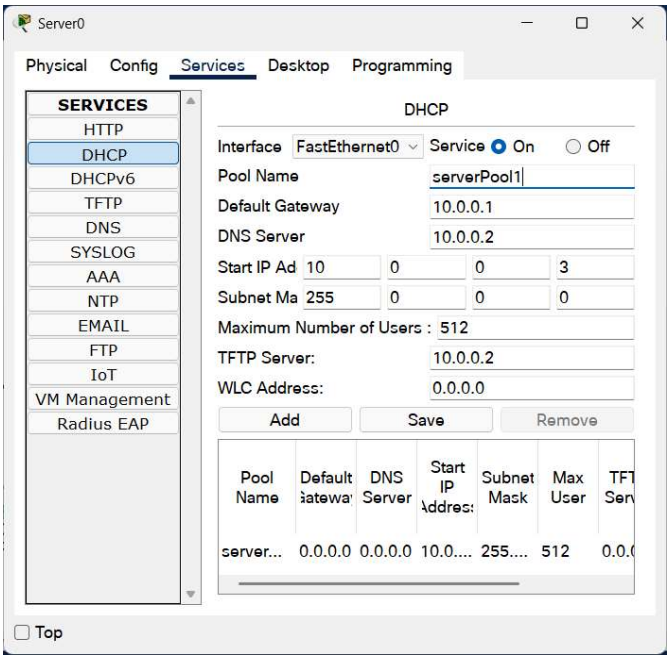
ii. Procedure along with the topology



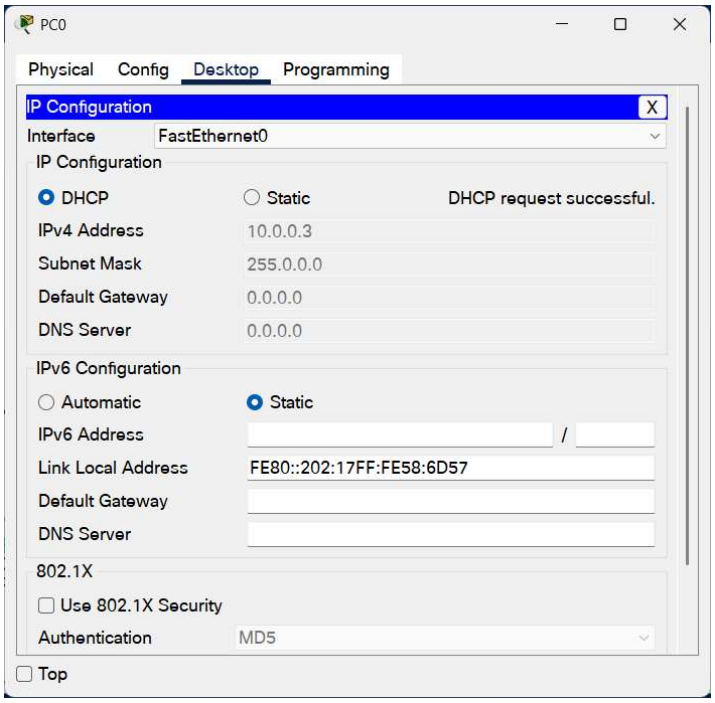
iii. Screen shots/ output

DHCP Within LAN

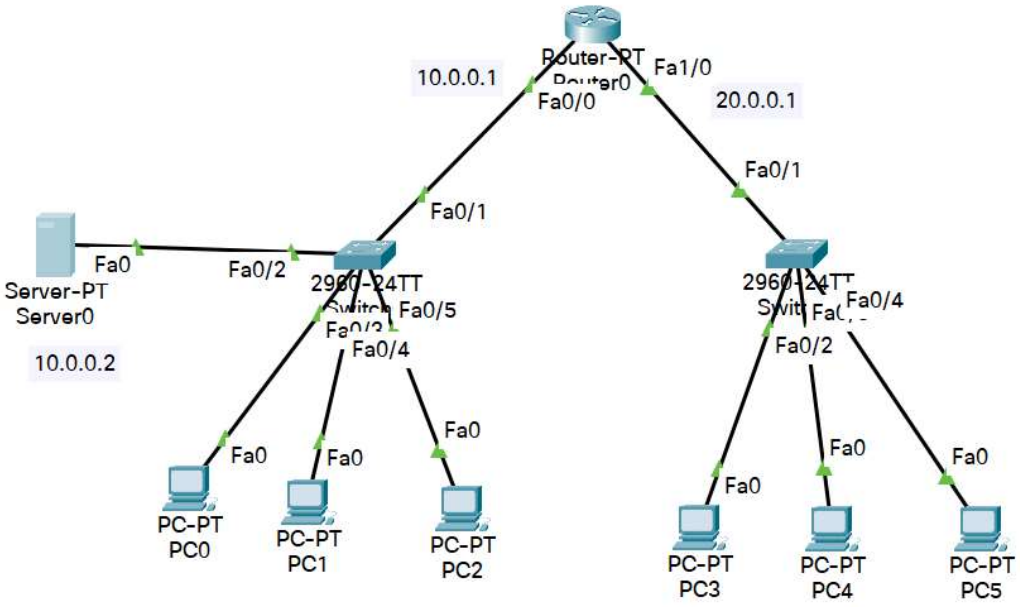
DHCP Configuration


Server0
Physical
Config
Services
Desktop
Programming
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
DHCP
Interface FastEthernet0
Service On
Off
Pool Name serverPool1
Default Gateway 10.0.0.1
DNS Server 10.0.0.2
Start IP Ad 10 0 0 3
Subnet Ma 255 0 0 0
Maximum Number of Users : 512
TFTP Server: 10.0.0.2
WLC Address: 0.0.0.0
Add
Save
Remove
Pool Name
Default Gateway
DNS Server
Start IP Address
Subnet Mask
Max User
server...
0.0.0.0
0.0.0.0
10.0....
255....
512
Top

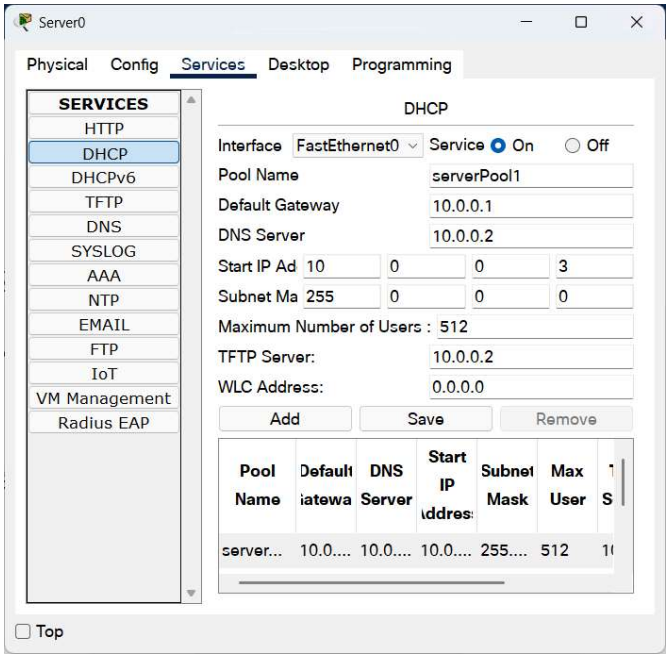
PC settings


PC0
Physical
Config
Desktop
Programming
IP Configuration
X
Interface FastEthernet0
IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address 10.0.0.3
Subnet Mask 255.0.0.0
Default Gateway 0.0.0.0
DNS Server 0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address /
Link Local Address FE80::202:17FF:FE58:6D57
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication MD5
Top

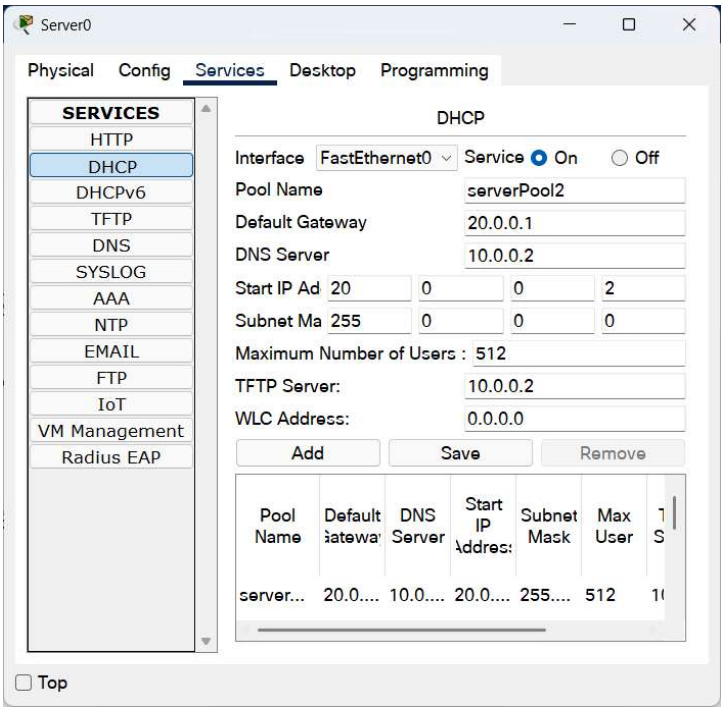
DHCP outside LAN:



DHCP configuration for inside LAN



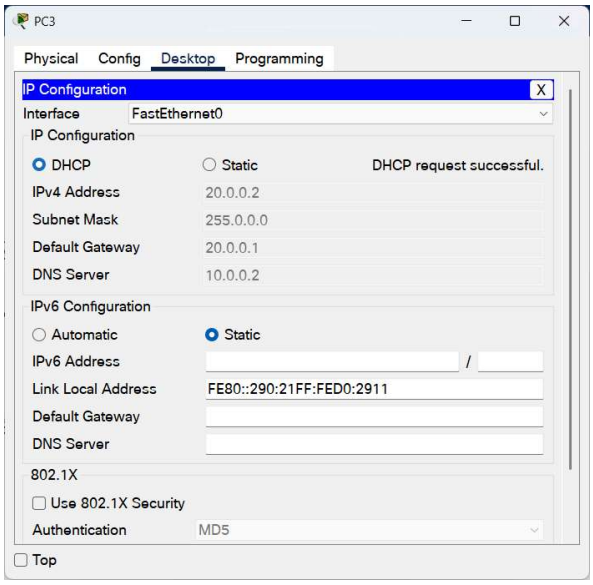
DHCP configuration for outside LAN



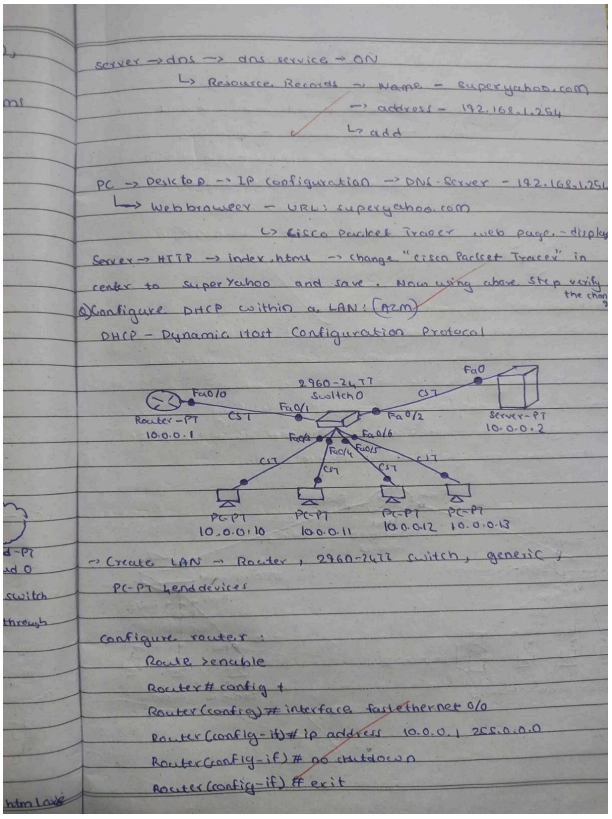
Ip helper command in Router

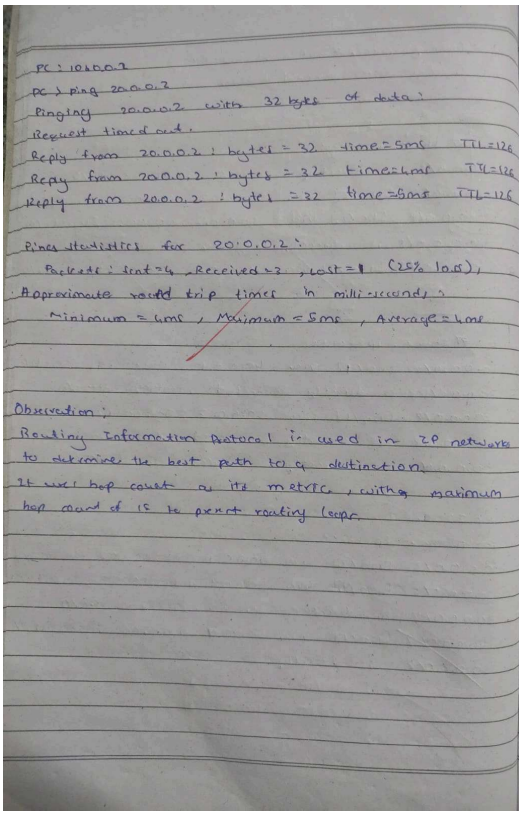
```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#
```

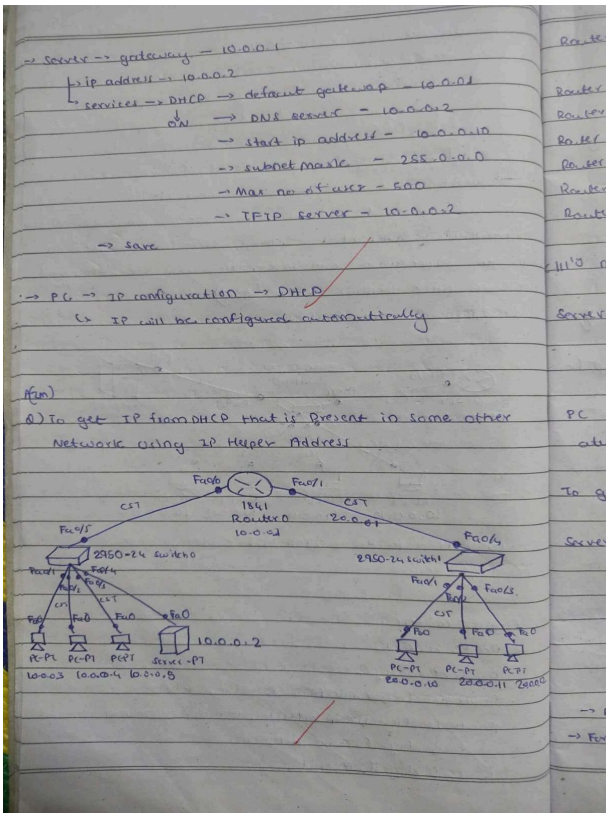
PC setting in another network

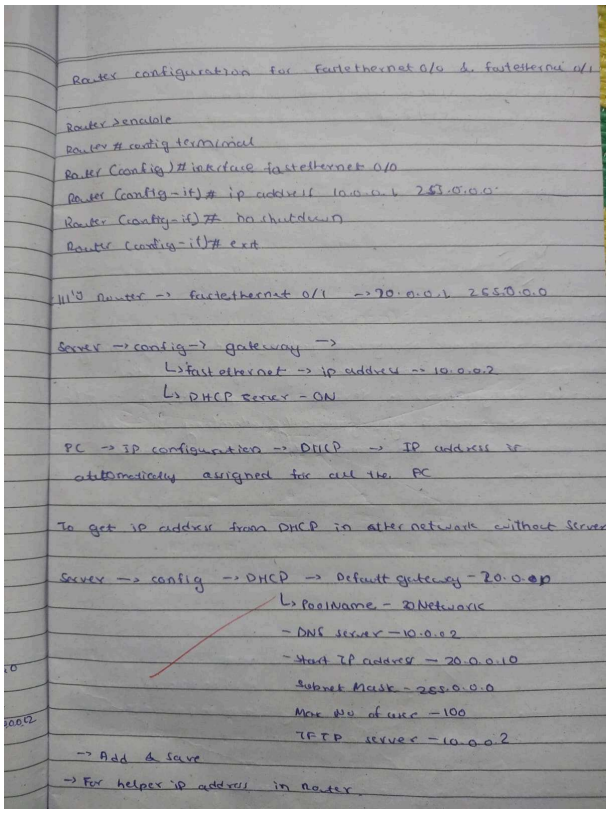


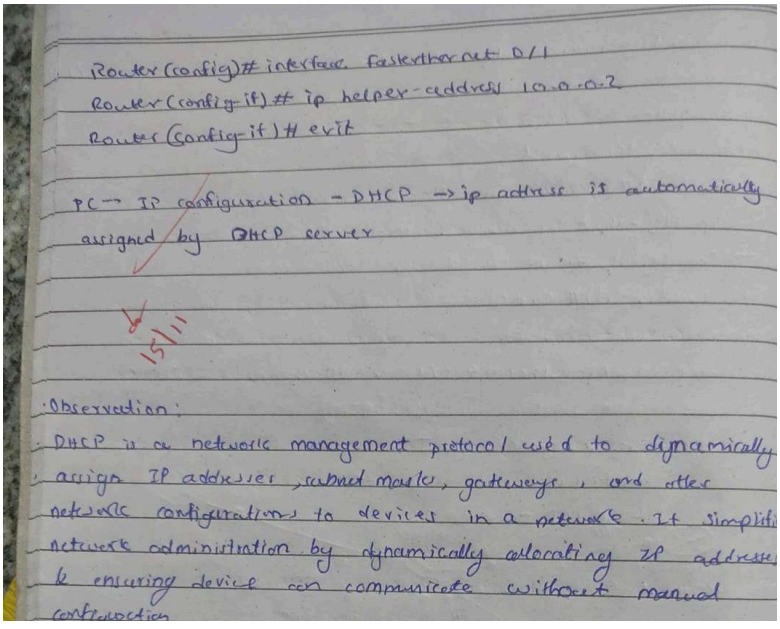
iv. Observation






→ Server → gateway → 10.0.0.1
ip address → 10.0.0.2
services → DHCP → default gateway → 10.0.0.1
ON → DNS server → 10.0.0.2
→ start ip address → 10.0.0.10
→ subnet mask → 255.0.0.0
→ Max no of user → 500
→ TFTP server → 10.0.0.2
→ save
→ PC → IP configuration → DHCP
↳ IP will be configured automatically
Fa0/0
Fa0/1
CST
1841 Router
10.0.0.1
20.0.0.1
Fa0/5
Fa0/4
2950-24 switch0
Fa0/1
Fa0/2
Fa0/3
Fa0/4
Fa0/5
2950-24 switch1
Fa0/1
Fa0/5
Fa0
PC-PT
10.0.0.3
10.0.0.4
10.0.0.5
Server-PT
10.0.0.2
PC-PT
20.0.0.10
20.0.0.11
20.0.0.12

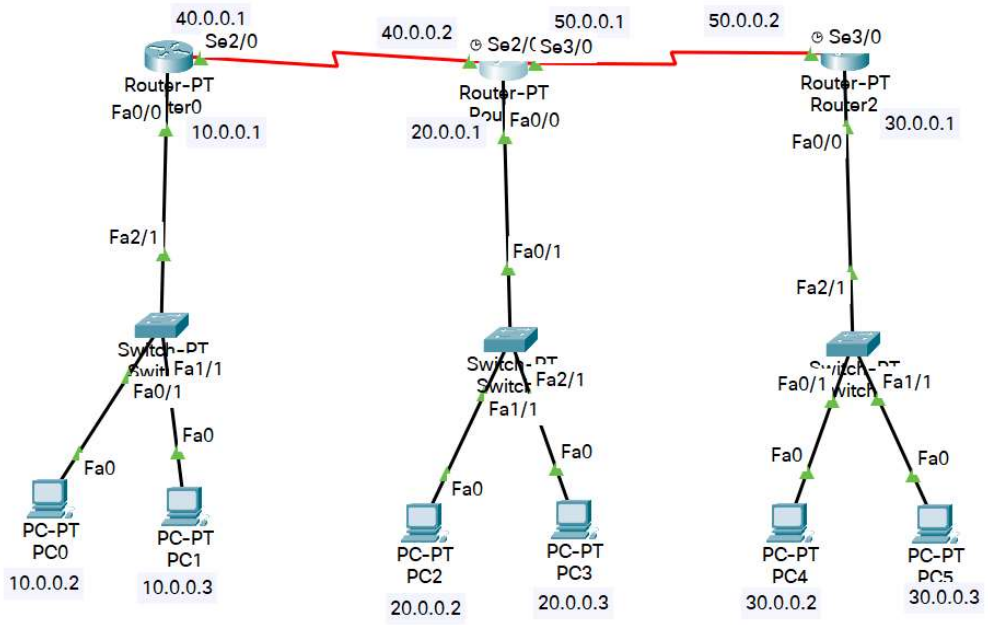

Router configuration for FastEthernet 0/0 & FastEthernet 0/1
Router >enable
Router # config terminal
Router (Config) # interface fastEthernet 0/0
Router (Config-if) # ip address 10.0.0.1 255.0.0.0
Router (Config-if) # no shutdown
Router (Config-if) # exit
Router → FastEthernet 0/1 → 20.0.0.1 255.0.0.0
Server → config → gateway →
↳ fastEthernet → ip address → 10.0.0.2
↳ DHCP Server - ON
PC → IP configuration → DHCP → IP address is automatically assigned for all the PC
To get ip address from DHCP in other network without Server
Server → config → DHCP → Default gateway - 20.0.0.0
↳ PoolName. - 2ONetwork
- DNS server - 10.0.0.2
- Start IP address - 20.0.0.10
Subnet Mask - 255.0.0.0
Max No of user - 100
TFTP server - 10.0.0.2
→ Add & save
→ For helper ip address in Router


Router (config) # interface FastEthernet 0/1
Router (config-if) # ip helper-address 10.0.0.2
Router (Config-if) # exit
PC → IP configuration → DHCP → ip address is automatically assigned by DHCP server
15/11
Observation:
DHCP is a network management protocol used to dynamically assign IP addresses, subnet masks, gateways, and other network configurations to devices in a network. It simplifies network administration by dynamically allocating IP addresses & ensuring devices can communicate without manual configuration

Program 5

i. Configure RIP routing Protocol in Routers

ii. Procedure along with the topology



iii. Screen shots/ output

Router0

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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
C    40.0.0.0/8 is directly connected, Serial2/0
```

Router1

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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 40.0.0.1, 00:00:08, Serial2/0
C    20.0.0.0/8 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:10, Serial3/0
C    40.0.0.0/8 is directly connected, Serial2/0
C    50.0.0.0/8 is directly connected, Serial3/0
```

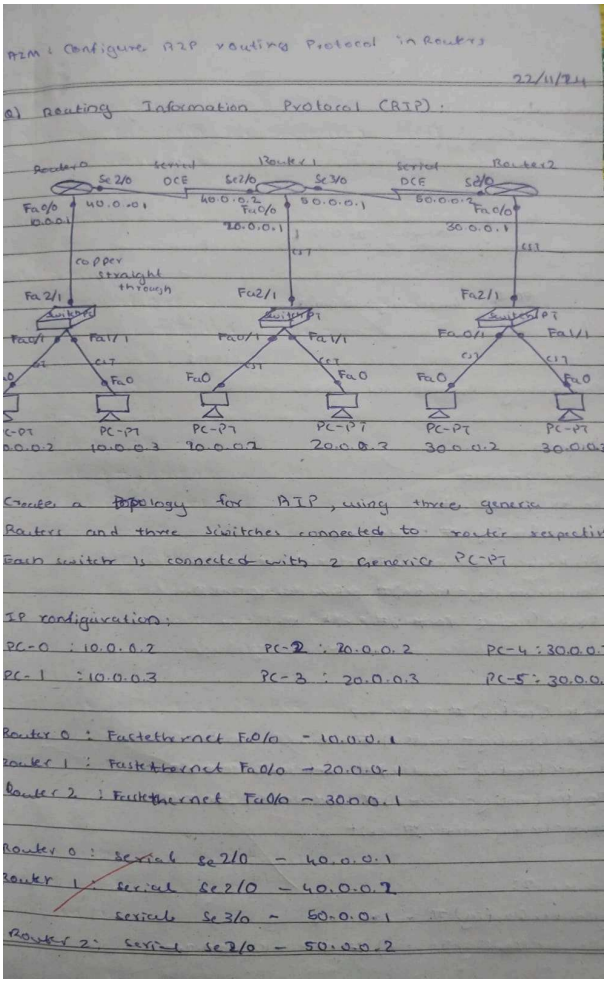
Router2

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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 50.0.0.1, 00:00:28, Serial3/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, Serial3/0
C    30.0.0.0/8 is directly connected, FastEthernet0/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, Serial3/0
C    50.0.0.0/8 is directly connected, Serial3/0
```

Pinging:

```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 9ms, Average = 6ms
```

iv. Observation



AIM : Configure RIP routing Protocol in Routers

22/11/24

a) Routing Information Protocol (RIP):

Create a topology for RIP, using three generic Routers and three switches connected to router respectively. Each switch is connected with 2 Generic PC-PT

IP configuration:

PC-0 : 10.0.0.2 PC-2 : 20.0.0.2 PC-4 : 30.0.0.2

PC-1 : 10.0.0.3 PC-3 : 20.0.0.3 PC-5 : 30.0.0.3

Router 0 : FastEthernet F0/0 - 10.0.0.1

Router 1 : FastEthernet Fa0/0 - 20.0.0.1

Router 2 : FastEthernet Fa0/0 - 30.0.0.1

Router 0 : Serial Se2/0 - 40.0.0.1

Router 1 : Serial Se2/0 - 40.0.0.2

Serial Se3/0 - 50.0.0.1

Router 2 : Serial Se3/0 - 50.0.0.2

Gateway:

For PC-0, PC-1 > Gateway : 10.0.0.1

PC-2, PC-3 > Gateway : 20.0.0.1

PC-4, PC-5 > Gateway : 30.0.0.1

Configuring Routers for RIP

Router 0:

Router > enable

Router > config t

Router (config) # router rip

Router (config-router) # network 10.0.0.0

Router (config-router) # network 40.0.0.0

Router (config-router) # exit

Router 1:

Router (config) # router rip

Router (config-router) # network 40.0.0.0

Router (config-router) # network 50.0.0.0

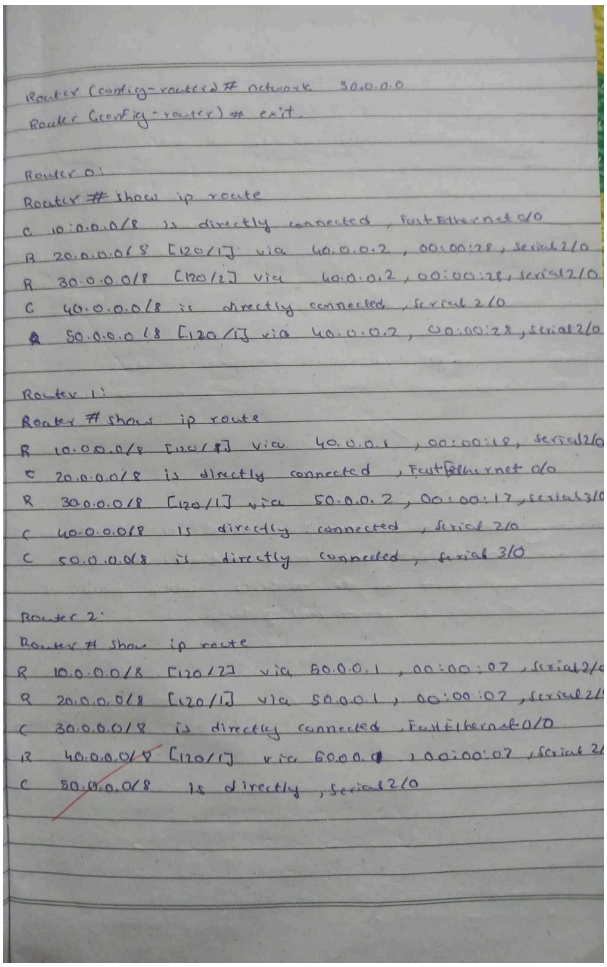
Router (config-router) # network 200.0.0

Router (config-router) # exit

Router 2:

Router (config) # router rip

Router (config-router) # network 30.0.0.0

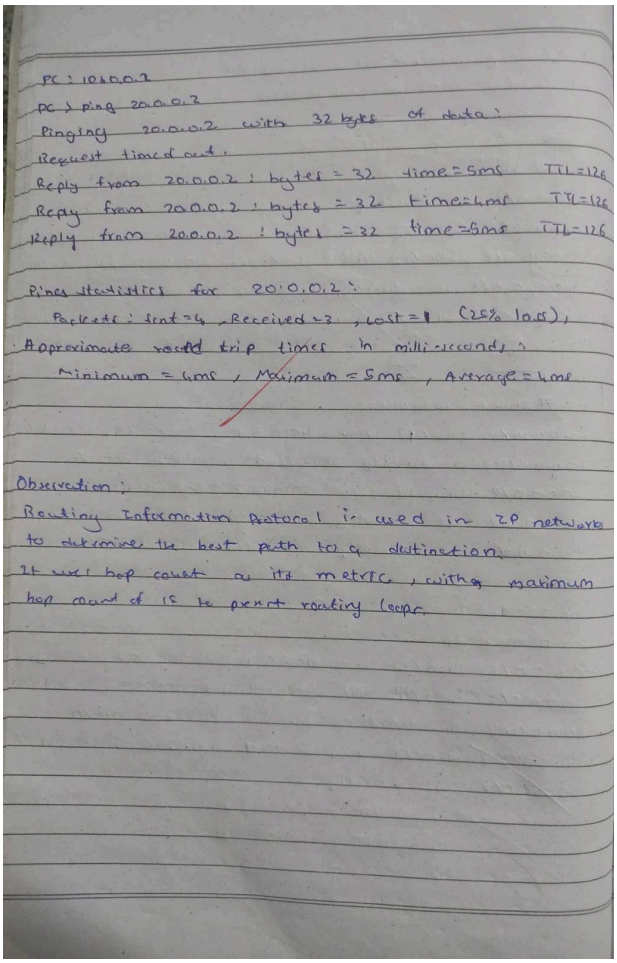


Router (config-router)# network 50.0.0.0
Router (config-router)# exit

Router 0:
Router # show ip route
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
R 20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:28, Serial 2/0
R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:28, Serial 2/0
C 40.0.0.0/8 is directly connected, Serial 2/0
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:28, Serial 2/0

Router 1:
Router # show ip route
R 10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:18, Serial 2/0
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
R 30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:17, Serial 3/0
C 40.0.0.0/8 is directly connected, Serial 2/0
C 50.0.0.0/8 is directly connected, Serial 3/0

Router 2:
Router # show ip route
R 10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:07, Serial 2/0
R 20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:07, Serial 2/0
C 30.0.0.0/8 is directly connected, FastEthernet 0/0
R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:07, Serial 2/0
C 50.0.0.0/8 is directly, Serial 2/0



PC: 10.0.0.2
PC > ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes = 32 time = 5ms TTL=126
Reply from 20.0.0.2: bytes = 32 time=4ms TTL=126
Reply from 20.0.0.2: bytes = 32 time=5ms TTL=126

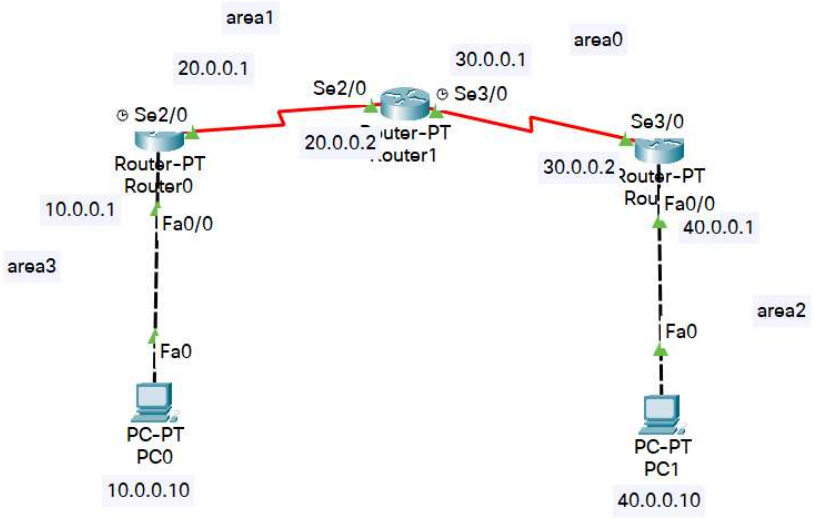
Pings statistics for 20.0.0.2:
Packets: Sent=4, Received=3, Lost=1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 5ms, Average = 4ms

Observation:
Routing Information Protocol is used in IP network to determine the best path to a destination. It uses hop count as its metric, with a maximum hop count of 15 to prevent routing loops.

Program 6

i. Configure OSPF routing protocol

ii. Procedure along with the topology



iii. Screen shots/ output

Encapsulation:

Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

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

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

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

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
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

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no dhutdown
                   ^
% Invalid input detected at '^' marker.

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

Router2

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
%IP-4-DUPADDR: Duplicate address 40.0.0.1 on FastEthernet0/0, sourced by
000D.BDDA.0123

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
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
```

OSPF Routing Protocol

Router0

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 1.1.1.1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 3
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#sho
00:27:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
w 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
O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:02, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:02, Serial2/0
```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 2.2.2.2
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

00:26:21: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on Serial3/0 from LOADING to FULL, Loading Done

00:27:18: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial2/0 from LOADING to FULL, Loading Done

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

     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
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:02:00, Serial3/0
```

Router2

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 3.3.3.3
Router(config-router)#network 40.0.0.0 0.255.255.255 area 2
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
00:26:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial3/0 from LOADING to FULL, Loading Done

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

O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:02:45, Serial3/0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial3/0
C       30.0.0.1/32 is directly connected, Serial3/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
```

Configure Loopback address

Router0

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

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

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

Router(config-if)#ip address 172.16.1.252 255.255.0.0
Router(config-if)#no shutdown
```

Router1

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

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

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

Router(config-if)#ip address 172.16.1.253 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#
```

Router2

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

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

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

Router(config-if)#ip address 172.16.1.254 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#
```

Create Virtual Link

Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#area 1 virtual-link 2.2.2.2
Router(config-router)#
```

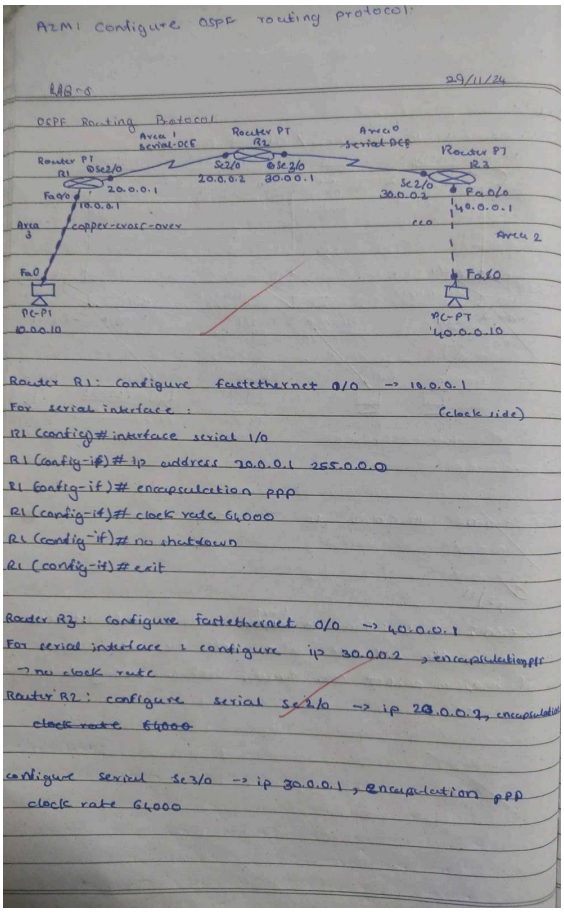
Router1

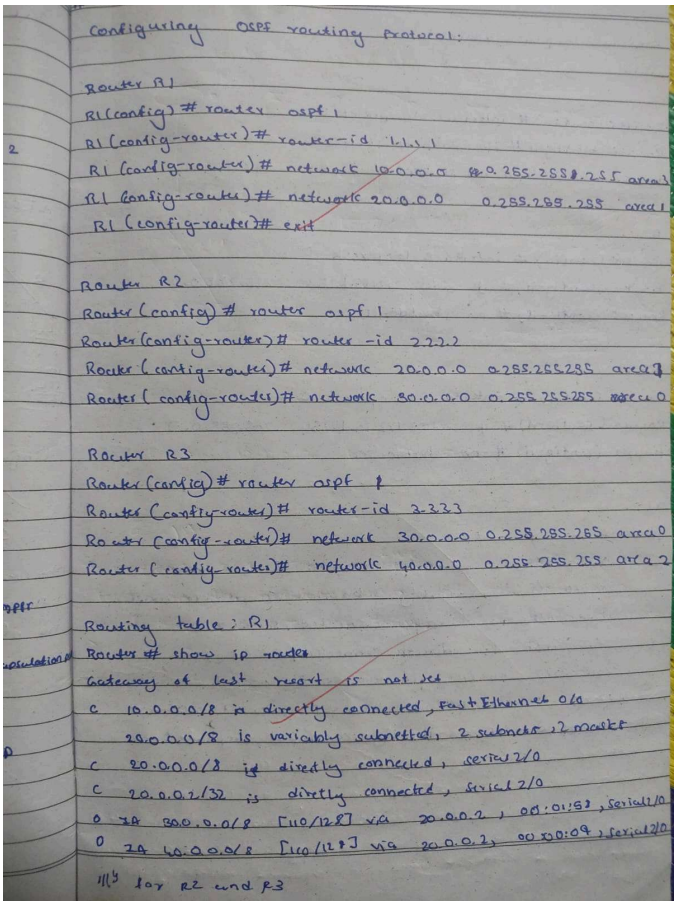
```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
01:11:01: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

01:11:11: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

Router(config)#route
01:11:21: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
r ospf 1
Router(config-router)#
01:11:31: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

Router(config-router)#area 1 v
01:11:41: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
irtual-link 1.1.1.1
Router(config-router)#
01:11:56: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on OSPF_VL0 from LOADING to
FULL, Loading Done
```

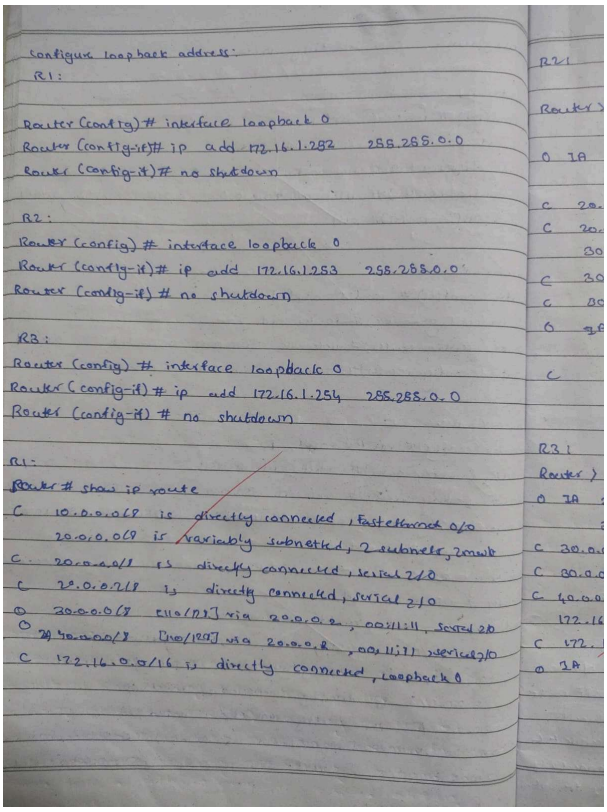
Pinging

```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=24ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 24ms, Average = 20ms
```

iv. Observation







Configure loopback address:

R1:

Router (config)# interface loopback 0
Router (config-if)# ip add 172.16.1.252 255.255.0.0
Router (config-if)# no shutdown

R2:

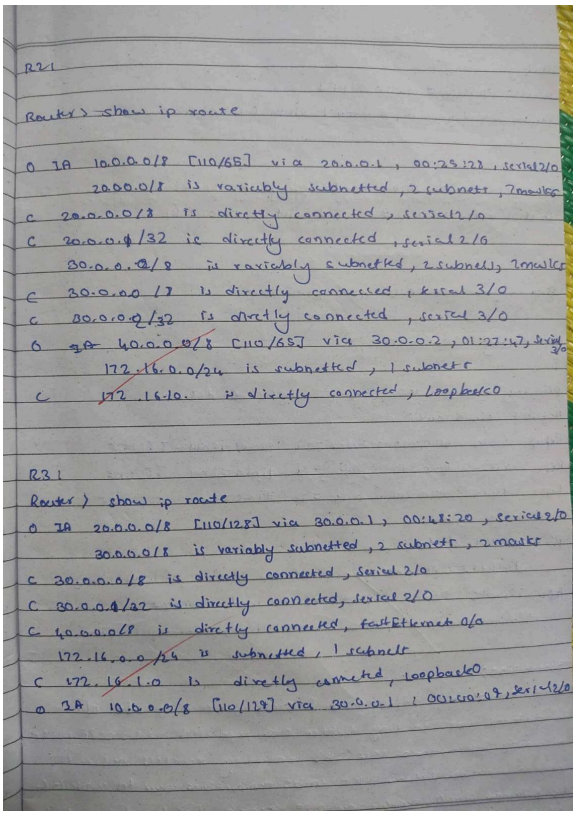
Router (config)# interface loopback 0
Router (config-if)# ip add 172.16.1.253 255.255.0.0
Router (config-if)# no shutdown

R3:

Router (config)# interface loopback 0
Router (config-if)# ip add 172.16.1.254 255.255.0.0
Router (config-if)# no shutdown

R1:

Router# show ip route
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 mask
C 20.0.0.0/8 is directly connected, Serial 2/0
C 20.0.0.2/32 is directly connected, Serial 2/0
O 30.0.0.0/8 [110/128] via 20.0.0.2, 00:11:11, Serial 2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:11:11, Serial 2/0
C 172.16.0.0/16 is directly connected, Loopback 0

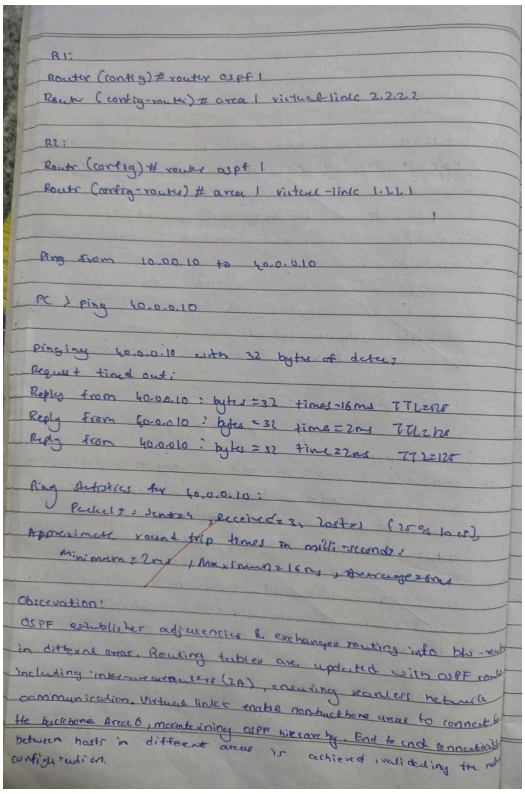


R2:

Router> show ip route
O IA 10.0.0.0/8 [110/65] via 20.0.0.1, 00:25:28, Serial 2/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 mask
C 20.0.0.0/8 is directly connected, Serial 2/0
C 20.0.0.1/32 is directly connected, Serial 2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 mask
C 30.0.0.0/8 is directly connected, Serial 3/0
C 30.0.0.2/32 is directly connected, Serial 3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 01:27:47, Serial 3/0
172.16.0.0/24 is subnetted, 1 subnet
C 172.16.10. is directly connected, Loopback0

R3:

Router> show ip route
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:48:20, Serial 2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 mask
C 30.0.0.0/8 is directly connected, Serial 2/0
C 30.0.0.1/32 is directly connected, Serial 2/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0
172.16.0.0/24 is subnetted, 1 subnet
C 172.16.1.0 is directly connected, Loopback0
O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:10:09, Serial 2/0



R1:

Router (config)# router ospf 1
Router (config-router)# area 1 virtual-link 2.2.2.2

R2:

Router (config)# router ospf 1
Router (config-router)# area 1 virtual-link 1.1.1.1

Ping from 10.0.0.10 to 40.0.0.10

PC > ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=16ms TTL=125
Reply from 40.0.0.10: bytes=32 time=2ms TTL=125
Reply from 40.0.0.10: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.10:
Packets: Sent=4, Received=3, Lost=1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum=2ms, Maximum=16ms, Average=6ms

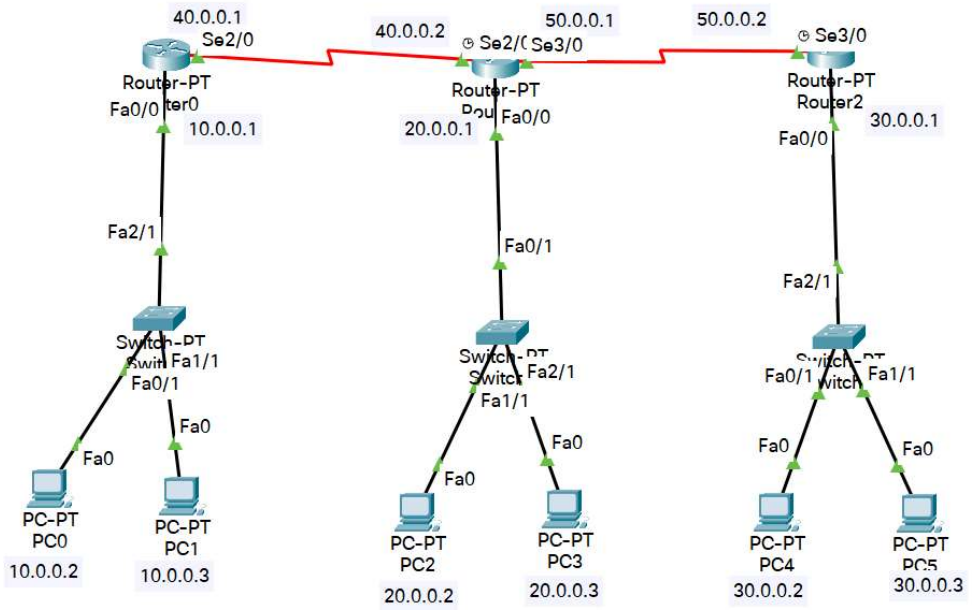
Observation:

OSPF establishes adjacencies & exchanges routing info so that routers in different areas. Routing tables are updated with OSPF routes including inter-area routes (IA), ensuring seamless network communication. Virtual links enable non-backbone areas to connect to the backbone Area 0, maintaining OSPF hierarchy. End to end connectivity between hosts in different areas is achieved, validating the network configuration.

Program 7

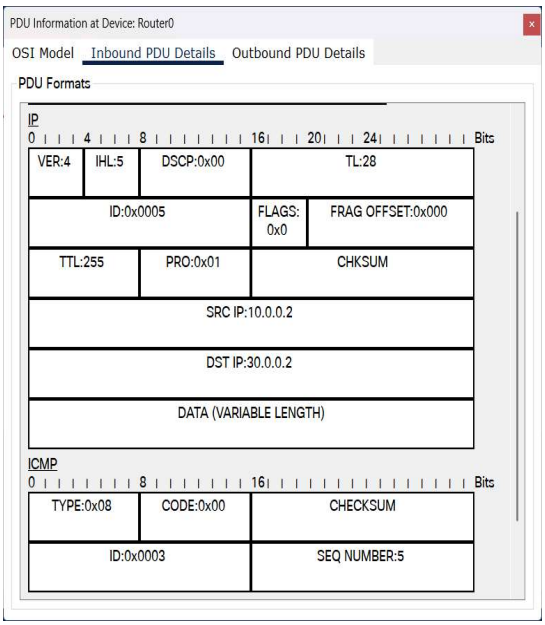
i. Demonstrate the TTL/ Life of a Packet

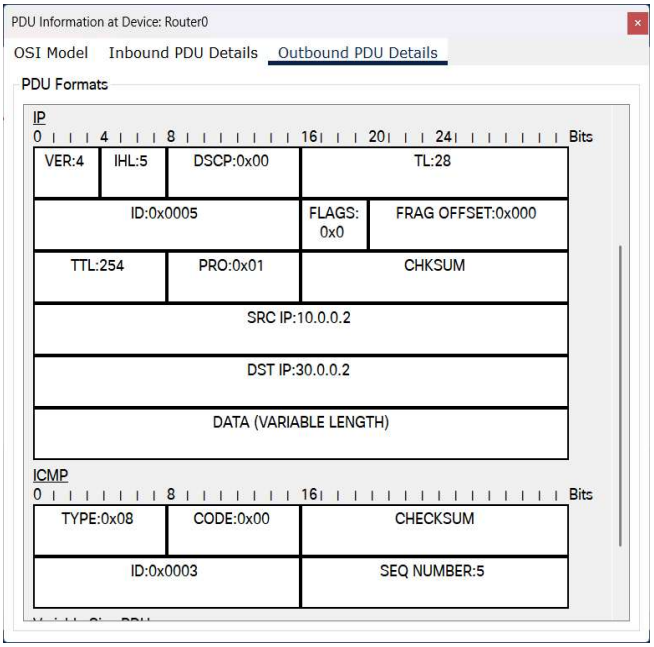
ii. Procedure along with the topology



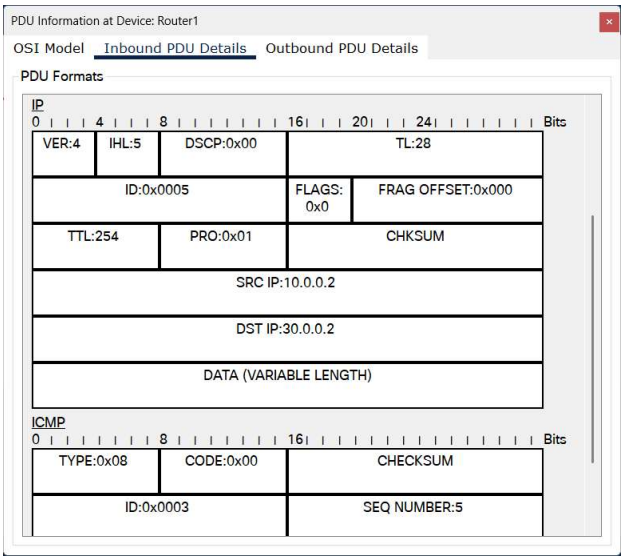
iii. Screen shots/ output

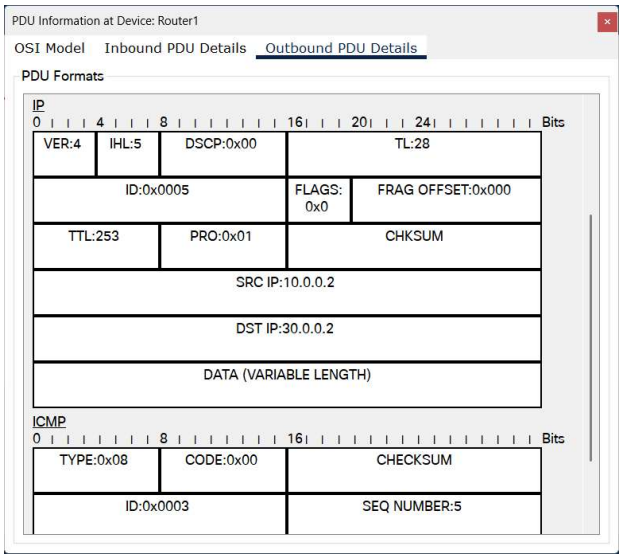
Packet at Router0



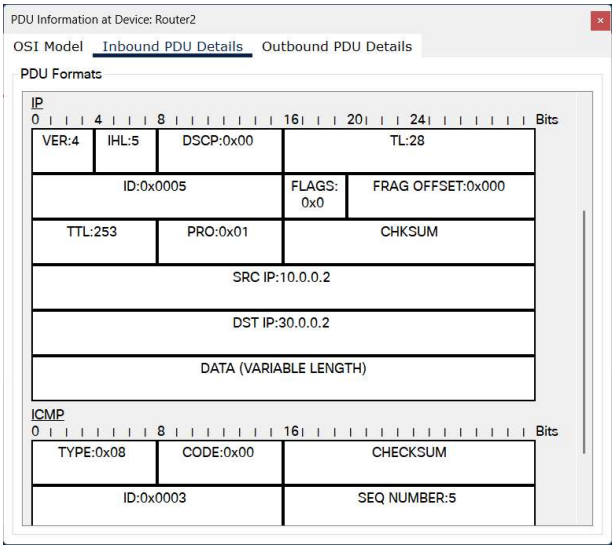


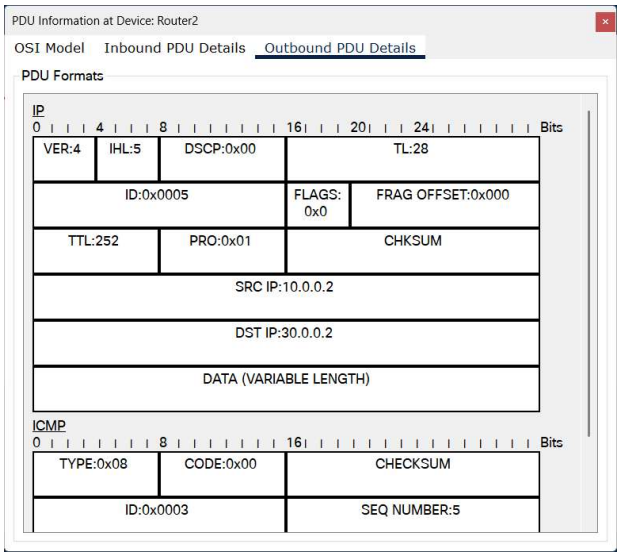
Packet at Router1



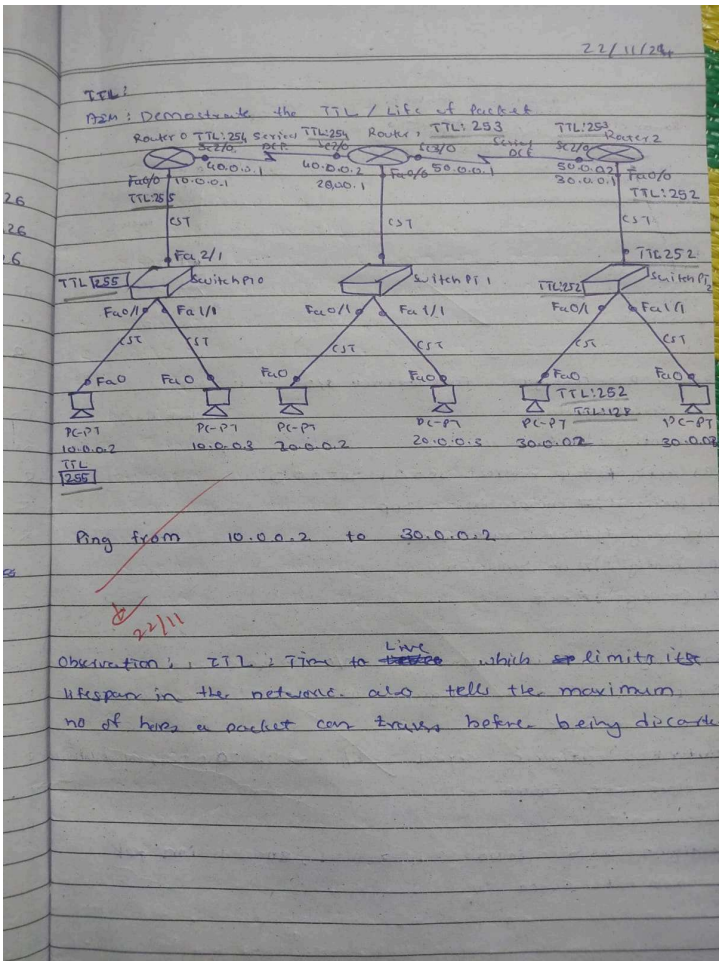


Packet at Router2





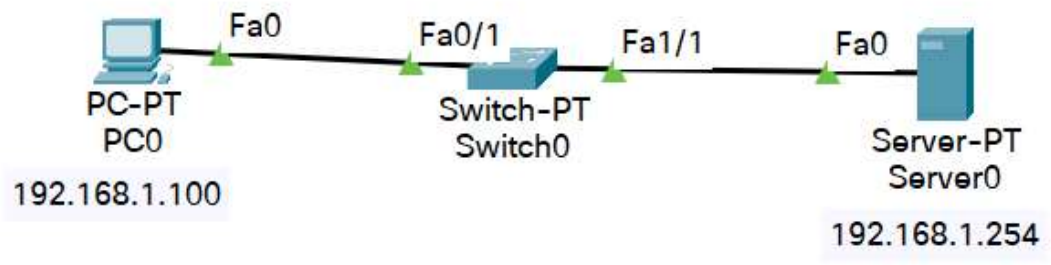
iv. Observation



Program 8

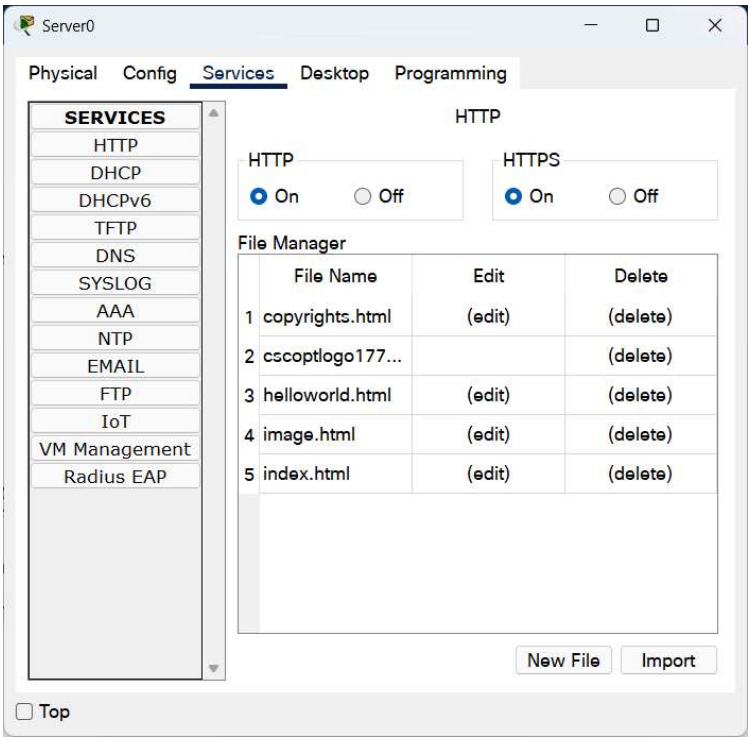
i. Configure Web Server, DNS within a LAN.

ii. Procedure along with the topology

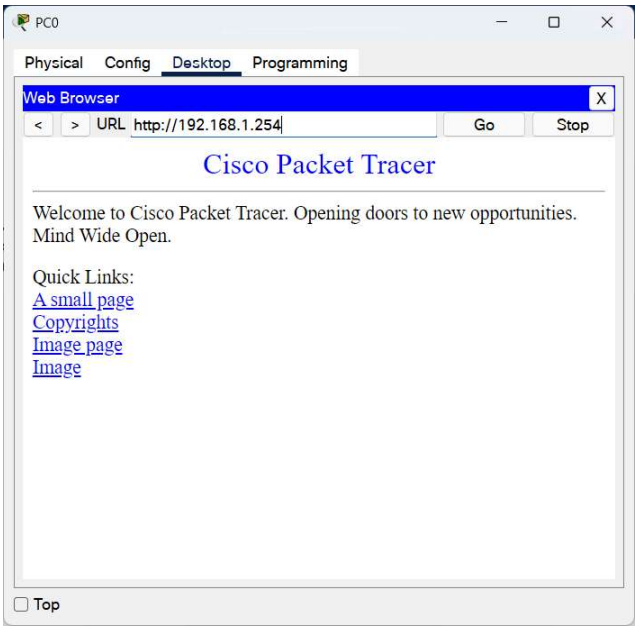


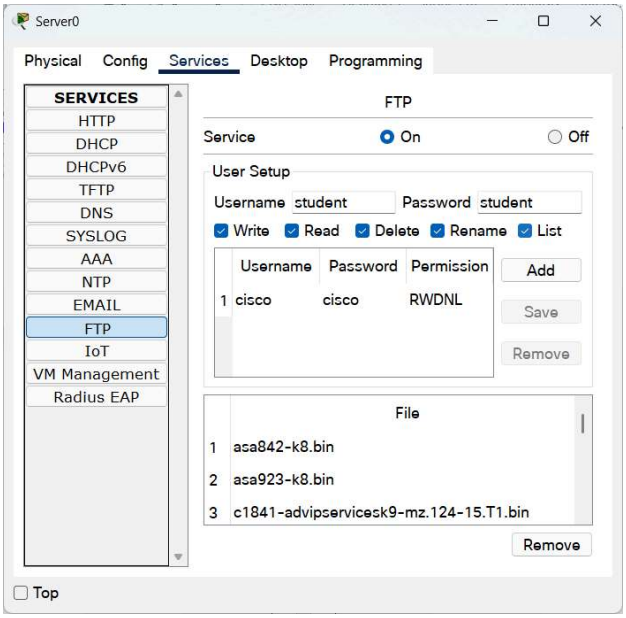
iii. Screen shots/ output

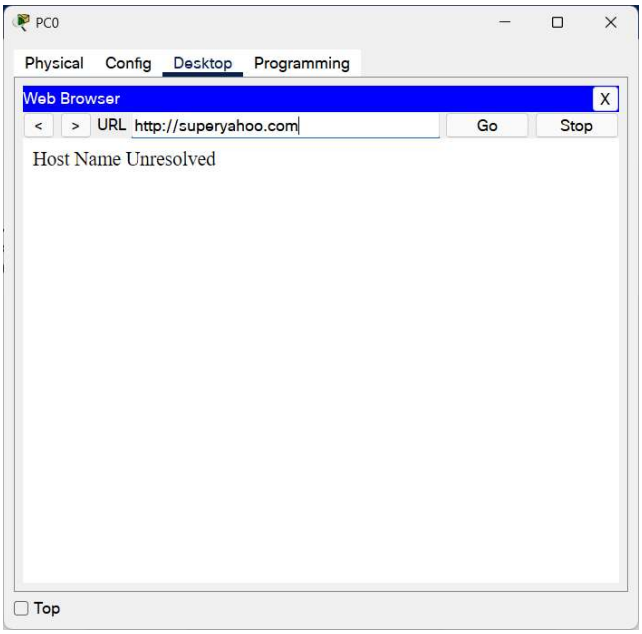
Server's services

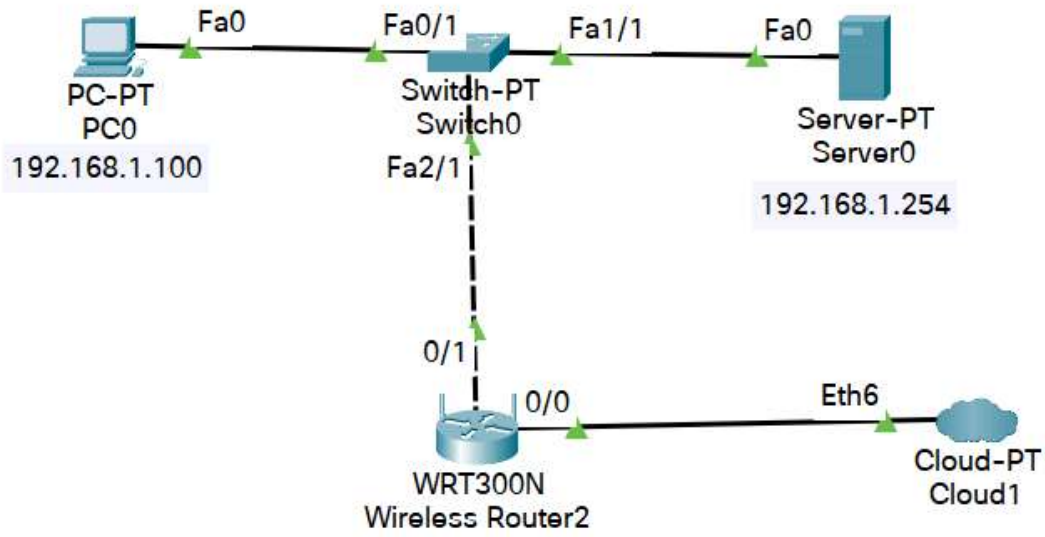


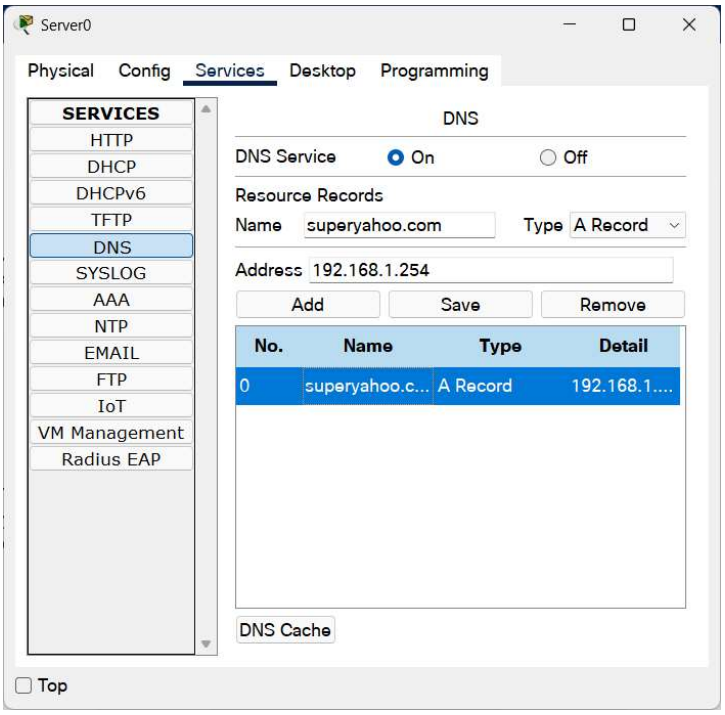
PC's Web Browser



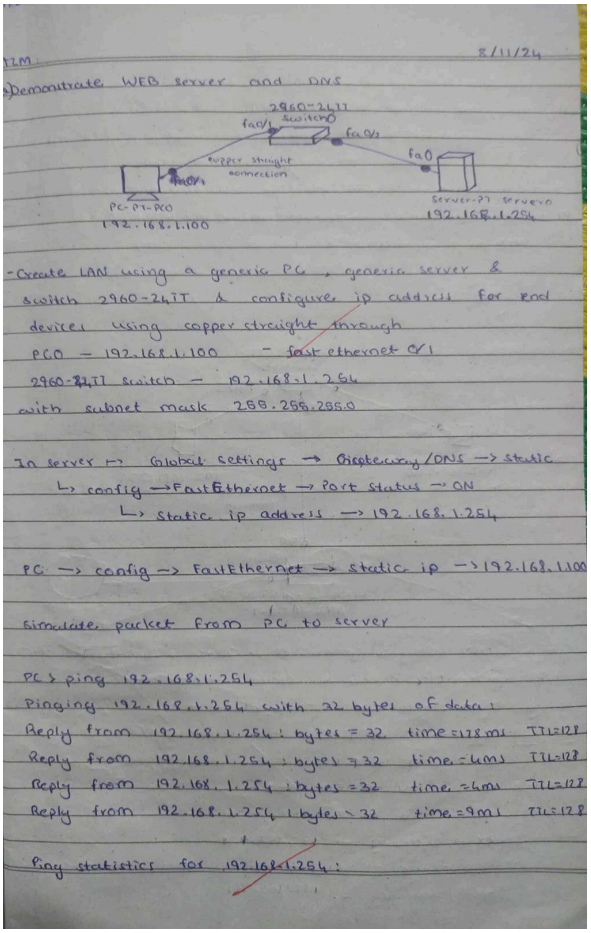



PC0
Physical
Config
Desktop
Programming
Web Browser
X
<
>
URL
http://superyahoo.com
Go
Stop
Host Name Unresolved
Top


Fa0
Fa0/1
Fa1/1
Fa0
PC-PT
PC0
192.168.1.100
Switch-PT
Switch0
Server-PT
Server0
192.168.1.254
Fa2/1
0/1
0/0
Eth6
WRT300N
Wireless Router2
Cloud-PT
Cloud1


Server0
Physical
Config
Services
Desktop
Programming
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
DNS
DNS Service
On
Off
Resource Records
Name
superyahoo.com
Type
A Record
Address
192.168.1.254
Add
Save
Remove
No.
Name
Type
Detail
0
superyahoo.c...
A Record
192.168.1....
DNS Cache
Top

iv. Observation



8/11/24

Demonstrate WEB server and DNS

-Create LAN using a generic PC, generic server & switch 2960-24TT & configure ip address for end devices using copper straight through

PC0 - 192.168.1.100 - fast ethernet 0/1

2960-24TT switch - 192.168.1.254

with subnet mask 255.255.255.0

In server → Global settings → Gateway/DNS → static

↳ config → FastEthernet → Port status → ON

↳ Static ip address → 192.168.1.254

PC → config → FastEthernet → static ip → 192.168.1.100

Simulate packet from PC to server

PC > ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

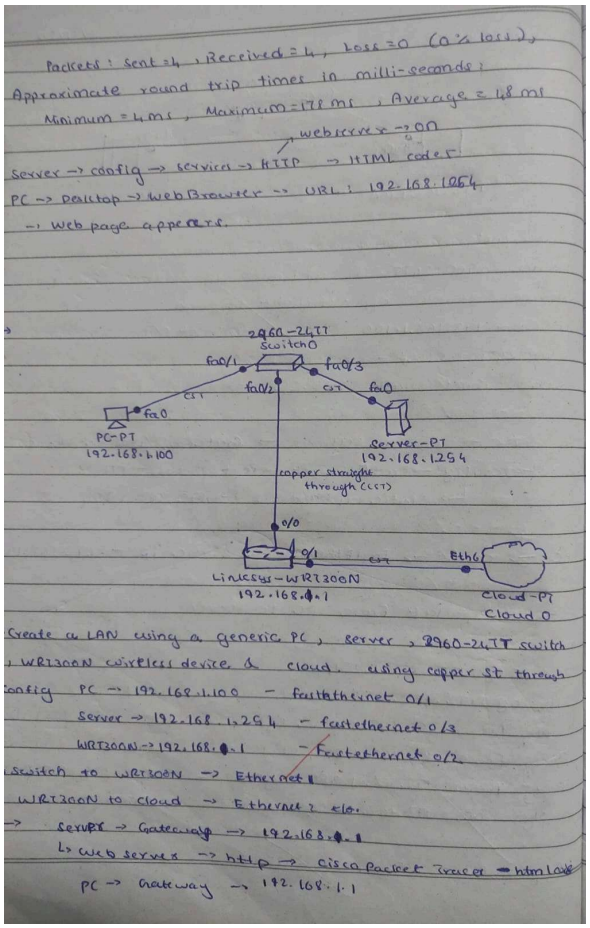
Reply from 192.168.1.254 : bytes = 32 time=178ms TTL=128

Reply from 192.168.1.254 : bytes = 32 time=4ms TTL=128

Reply from 192.168.1.254 : bytes = 32 time=4ms TTL=128

Reply from 192.168.1.254 : bytes = 32 time=9ms TTL=128

Ping statistics for 192.168.1.254:



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

Approximate round trip times in milli-seconds:

Minimum = 4ms, Maximum = 178ms, Average = 48ms

Server → config → services → HTTP → HTML code

PC → Desktop → Web Browser → URL: 192.168.1.254

→ Web page appears.

Create a LAN using a generic PC, server, 2960-24TT switch, WRT300N wireless device & cloud. using copper st through

config PC → 192.168.1.100 - fastethernet 0/1

Server → 192.168.1.254 - fastethernet 0/3

WRT300N → 192.168.0.1 - fastethernet 0/2

switch to WRT300N → Ethernet

WRT300N to cloud → Ethernet 2 etc.

→ server → Gateway → 192.168.0.1

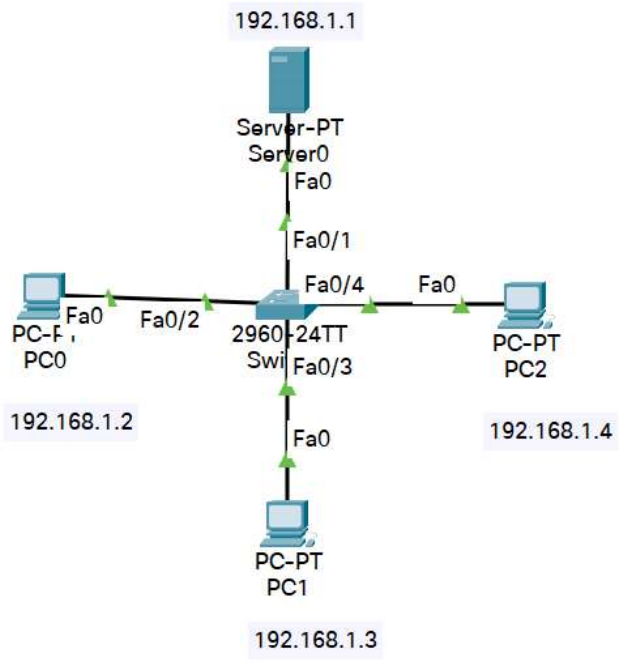
↳ web servers → http → cisco packet tracer → html code

PC → Gateway → 192.168.1.1

Program 9

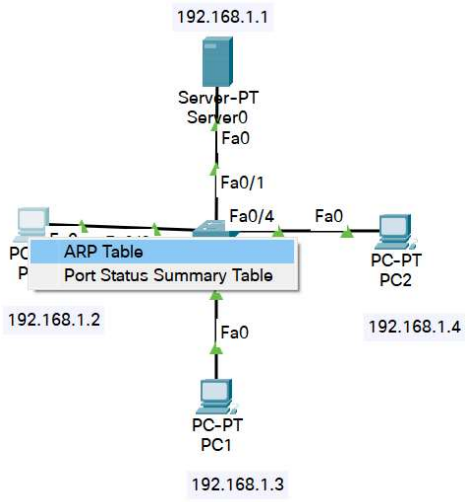
i. To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

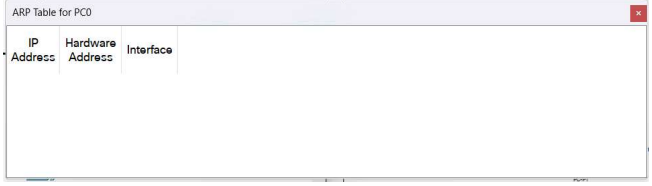
ii. Procedure along with the topology



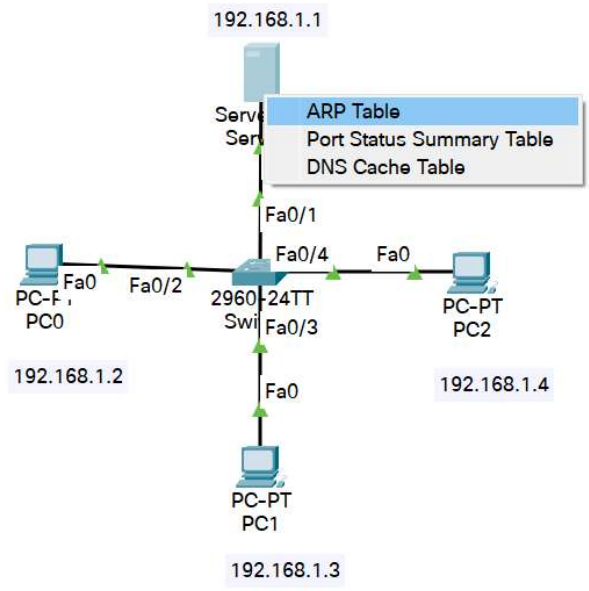
iii. Screen shots/ output

ARP Table of PC

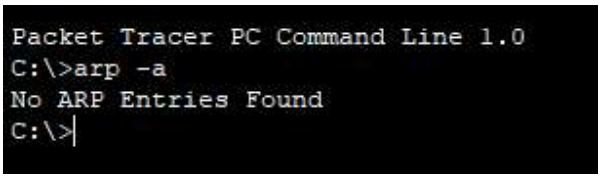




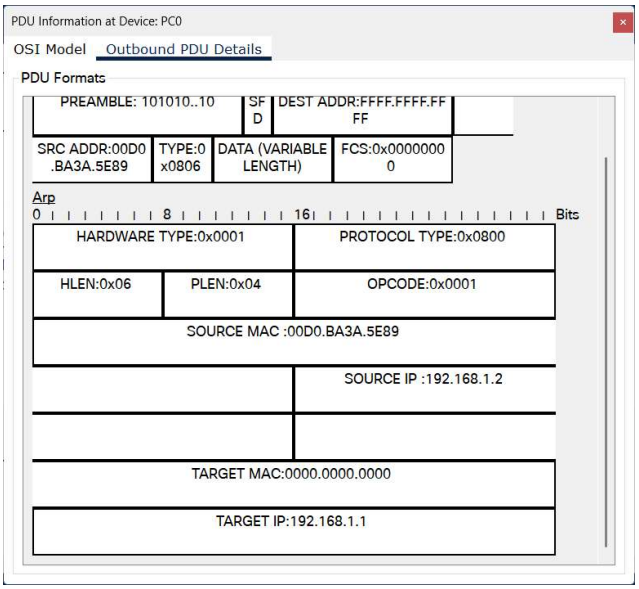
ARP Table of Server

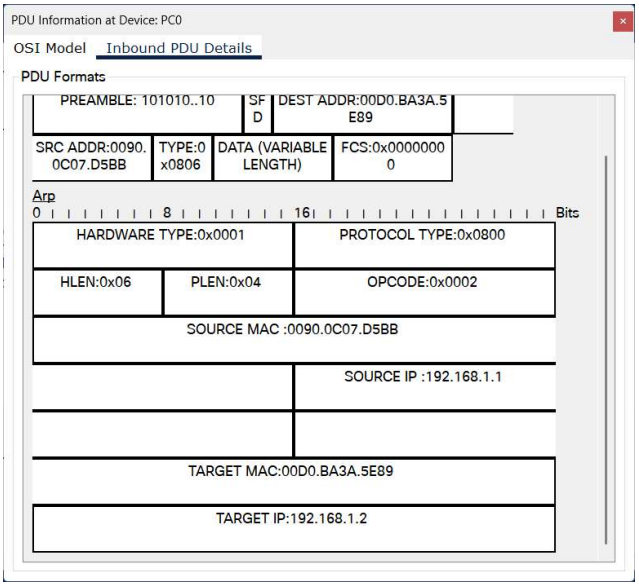


Command at PC

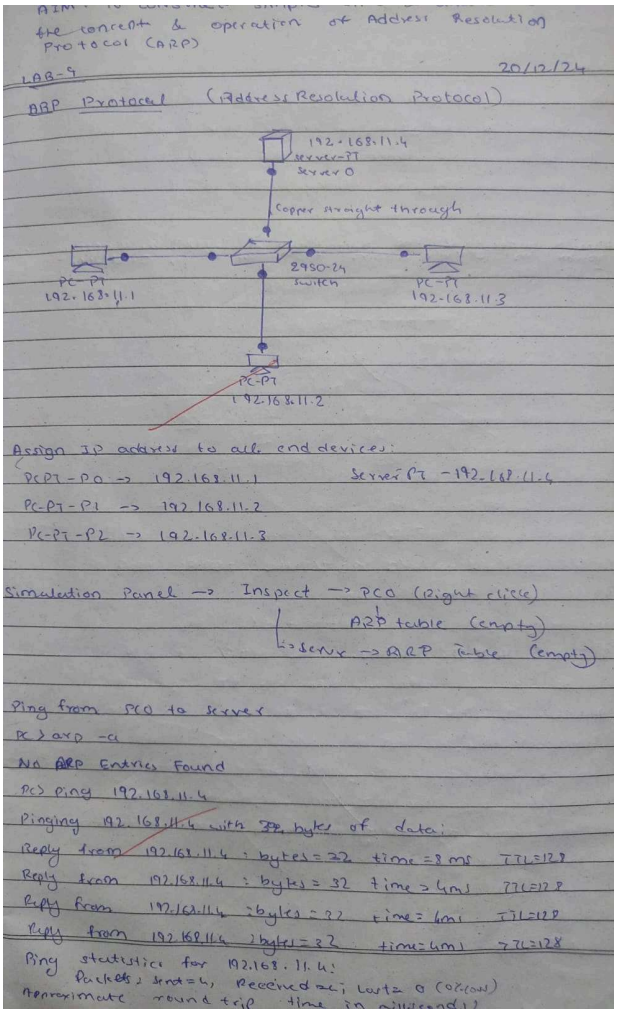


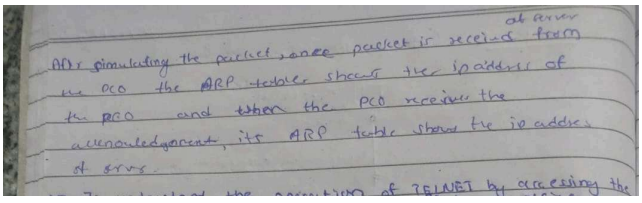
Pinging in Simulation Mode


PDU Information at Device: PC0
OSI Model
Outbound PDU Details
PDU Formats
PREAMBLE: 101010..10
SFD
DEST ADDR:FFFF.FFFF.FFFF
SRC ADDR:00D0.BA3A.5E89
TYPE:0x0806
DATA (VARIABLE LENGTH)
FCS:0x00000000
Arp
0 8 16 Bits
HARDWARE TYPE:0x0001
PROTOCOL TYPE:0x0800
HLEN:0x06
PLEN:0x04
OPCODE:0x0001
SOURCE MAC :00D0.BA3A.5E89
SOURCE IP :192.168.1.2
TARGET MAC:0000.0000.0000
TARGET IP:192.168.1.1


PDU Information at Device: PC0
OSI Model
Inbound PDU Details
PDU Formats
PREAMBLE: 101010..10
SFD
DEST ADDR:00D0.BA3A.5E89
SRC ADDR:0090.0C07.D5BB
TYPE:0x0806
DATA (VARIABLE LENGTH)
FCS:0x00000000
Arp
0 8 16 Bits
HARDWARE TYPE:0x0001
PROTOCOL TYPE:0x0800
HLEN:0x06
PLEN:0x04
OPCODE:0x0002
SOURCE MAC :0090.0C07.D5BB
SOURCE IP :192.168.1.1
TARGET MAC:00D0.BA3A.5E89
TARGET IP:192.168.1.2

iv. Observation

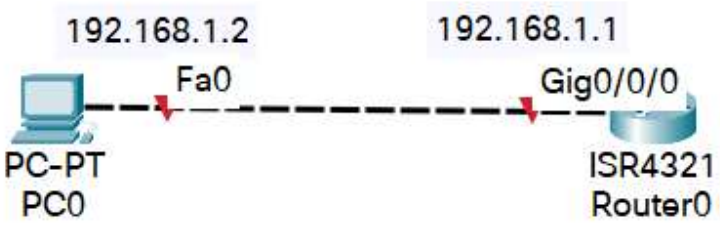




Program 10

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

ii. Procedure along with the topology

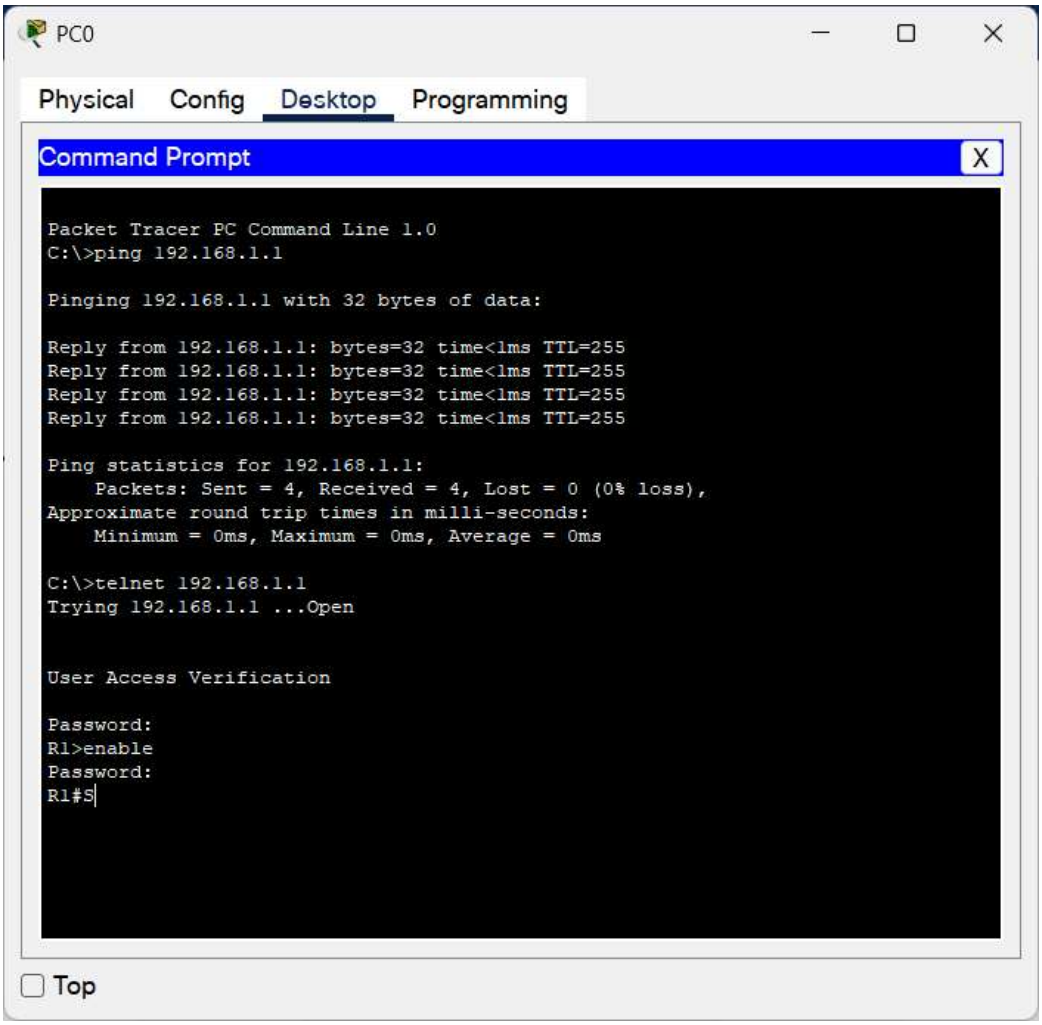


iii. Screen shots/ output

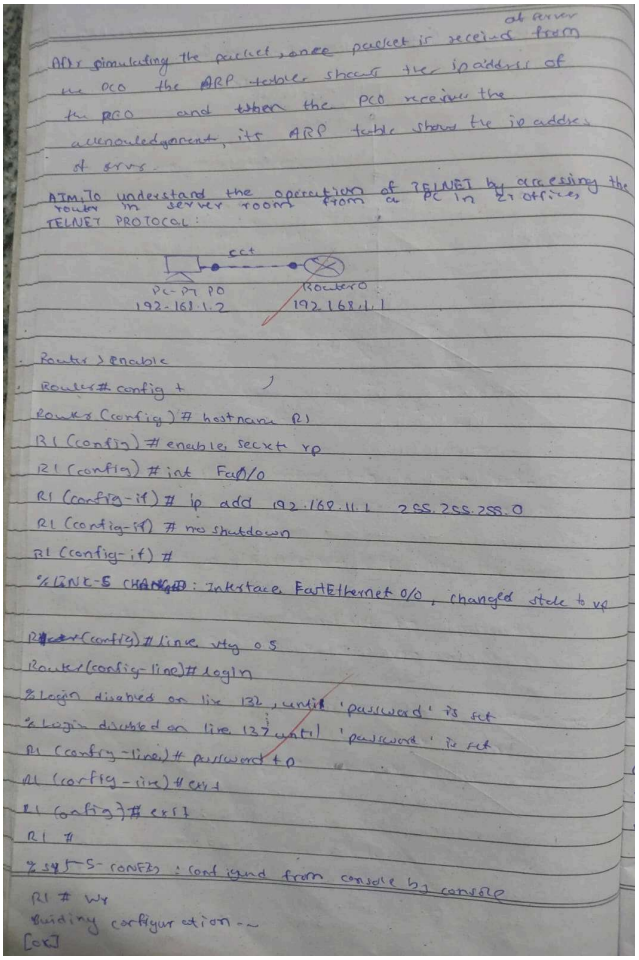
Router

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret hello
R1(config)#interface g0/0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown

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

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

R1(config-if)#line vty 0 5
R1(config-line)#login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password' is set
% Login disabled on line 7, until 'password' is set
R1(config-line)#password pass
R1(config-line)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#wr
Building configuration...
[OK]
R1#
```

PC


PC0
Physical
Config
Desktop
Programming
Command Prompt
X
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>telnet 192.168.1.1
Trying 192.168.1.1 ...Open
User Access Verification
Password:
R1>enable
Password:
R1#S
Top

iv. Observation



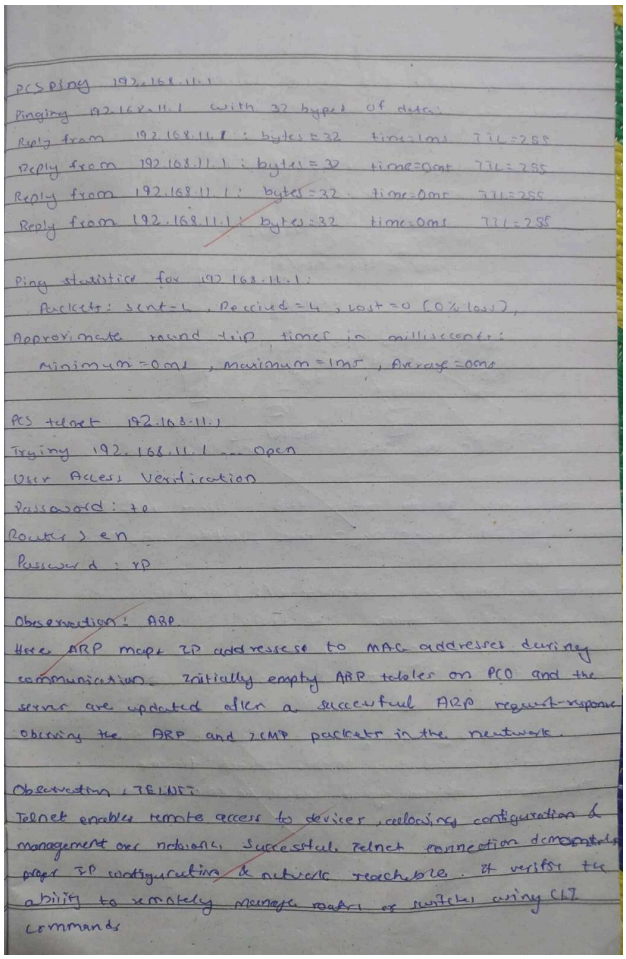
ARP: simulating the packet, once packet is received from the PC0 the ARP table shows the ip address of the PC0 and then the PC0 receives the acknowledgement, its ARP table shows the ip address of server.

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

TELNET PROTOCOL:

Router > enable
Router# config t
Router(config)# hostname R1
R1(config)# enable secret rp
R1(config)# int Fa0/0
R1(config-if)# ip add 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)#
%LINK-5 CHANGED: Interface FastEthernet 0/0, changed state to up

R1(config)# line vty 0 5
Router(config-line)# login
%Login disabled on line 132, until 'password' is set
%Login disabled on line 137, until 'password' is set
R1(config-line)# password rp
R1(config-line)# exit
R1(config)# exit
R1#
%SYS-5-CONFIG: Configured from console by console
R1# wr
Building configuration...
[OK]



PC$ ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255

Ping statistics for 192.168.1.1:
Packets: sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milliseconds:
Minimum=0ms, Maximum=1ms, Average=0ms

PC$ telnet 192.168.1.1
Trying 192.168.1.1 ... Open
User Access Verification
Password: rp
Router > en
Password: rp

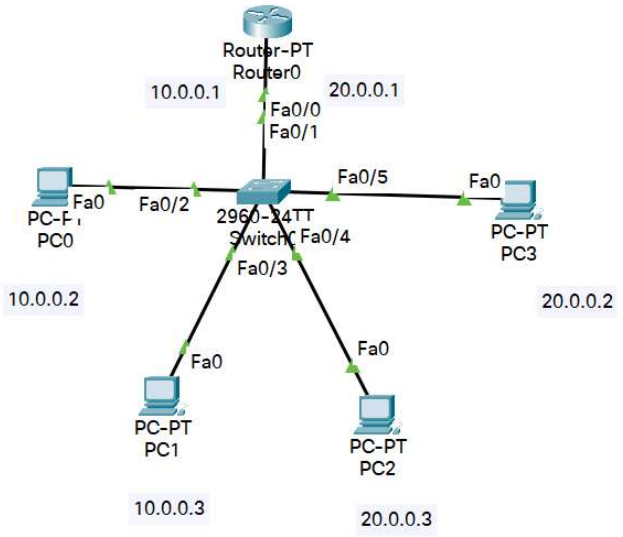
Observation: ARP
Here ARP maps IP addresses to MAC addresses during communication. Initially empty ARP tables on PC0 and the server are updated after a successful ARP request-response observing the ARP and ICMP packets in the network.

Observation: TELNET
Telnet enables remote access to devices, allowing configuration & management over networks. Successful Telnet connection demonstrates proper IP configuration & network reachability. It verifies the ability to remotely manage routers or switches using CLI commands.

Program 11

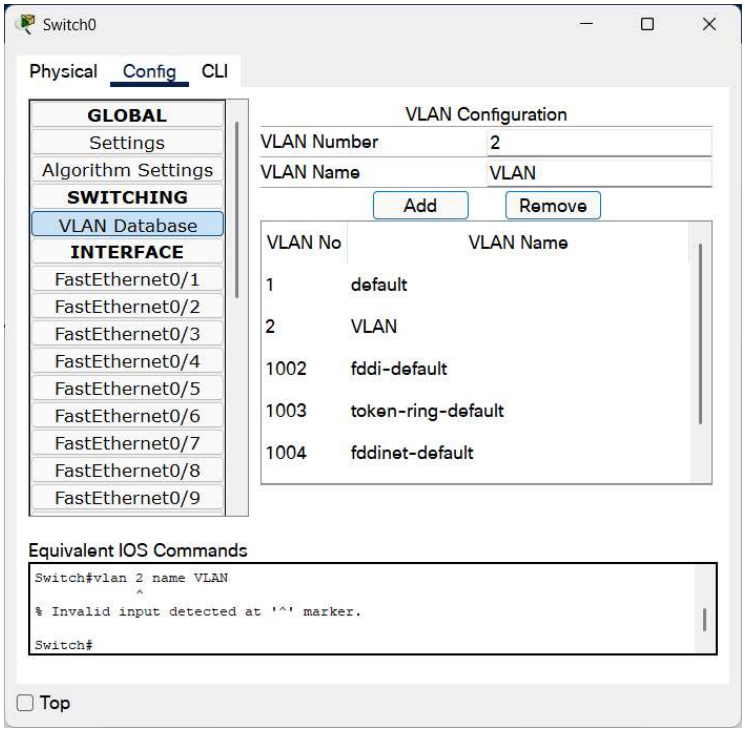
i. To construct a VLAN and make the PC's communicate among a VLAN

ii. Procedure along with the topology

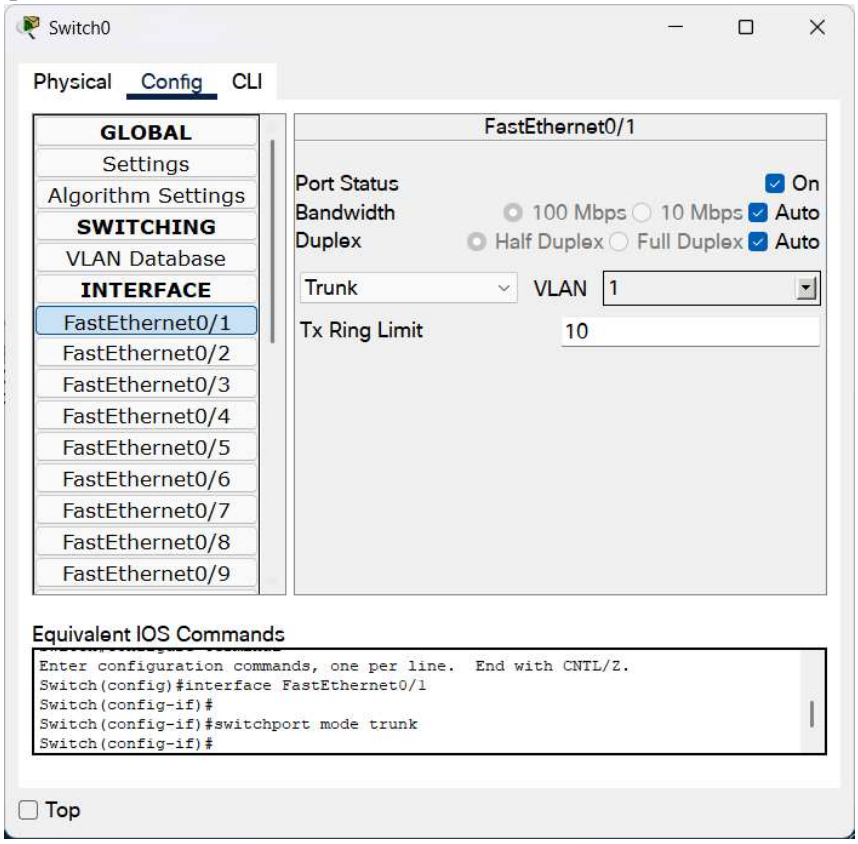


iii. Screen shots/ output

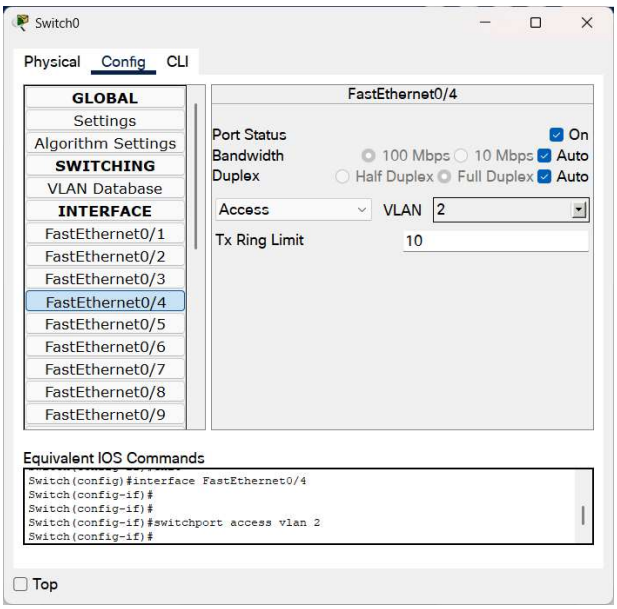
Switch Configuration

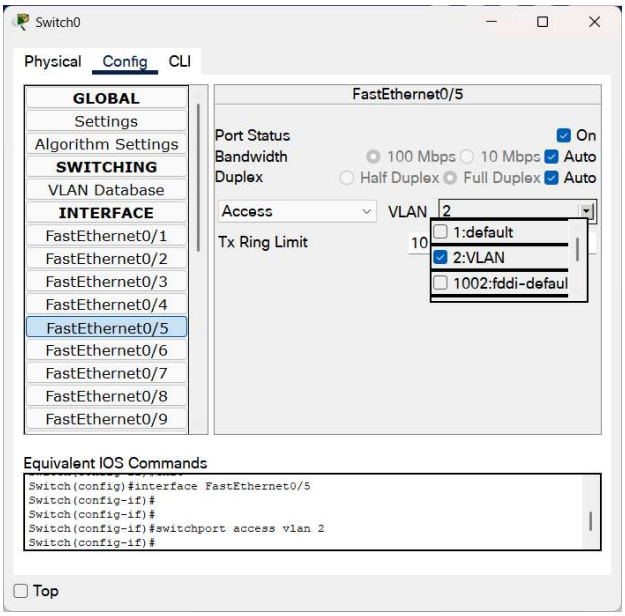


Configuration of port connected to router

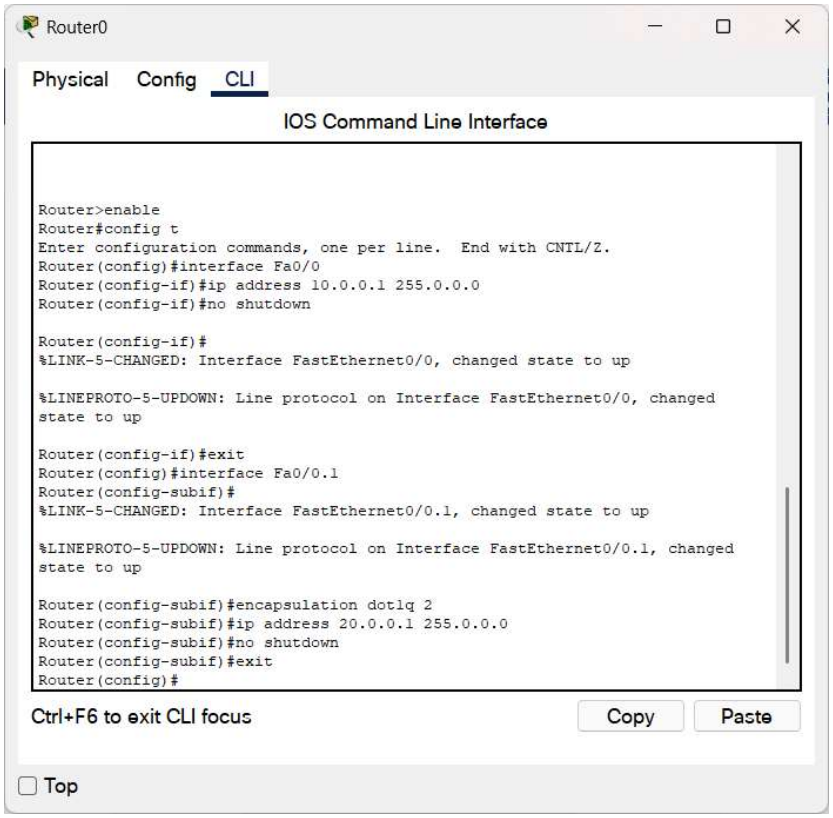


Configuration of ports connected to other networks

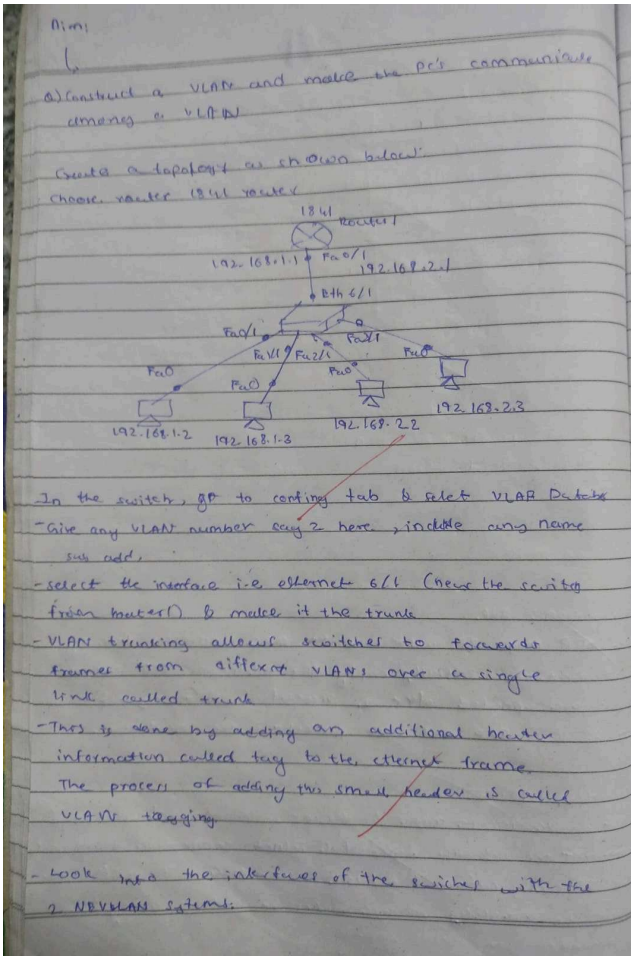


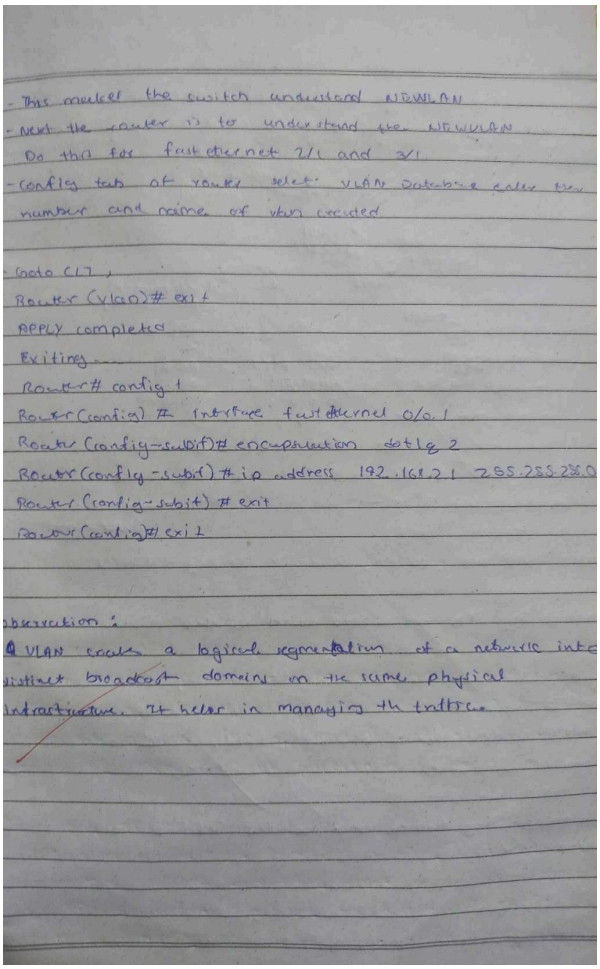


Configuartion of Router



iv. Observation

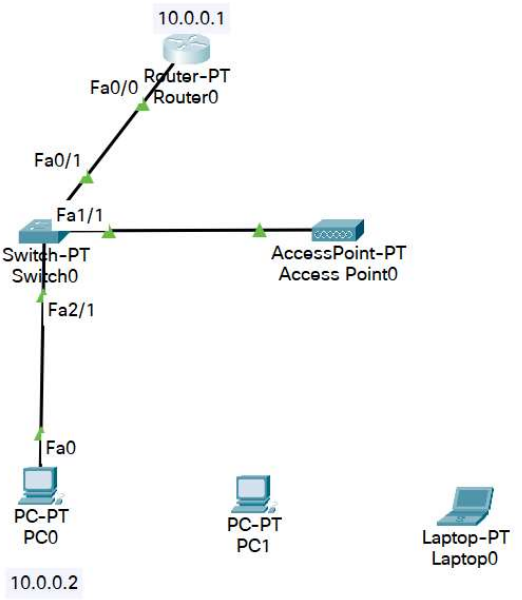




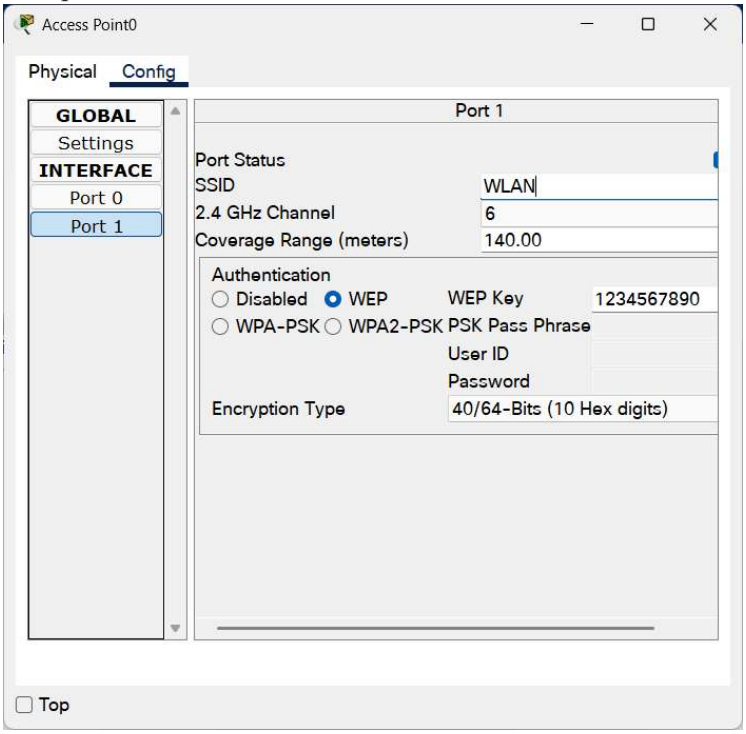
Program 12

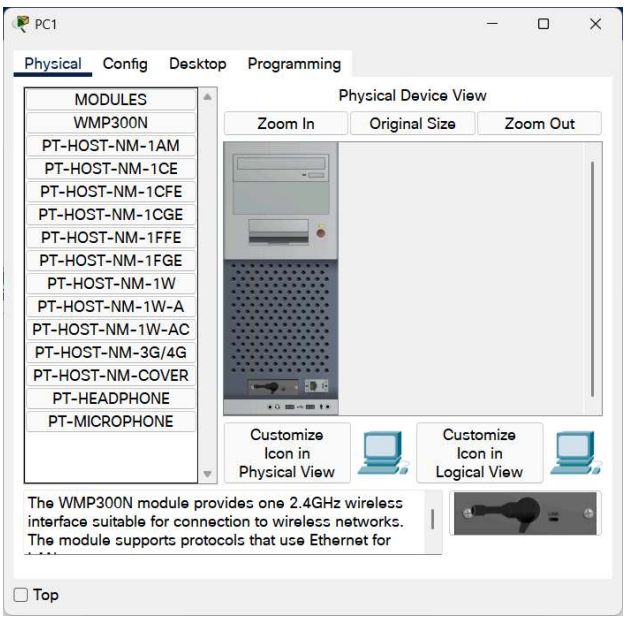
i. To construct a WLAN and make the nodes communicate wirelessly.

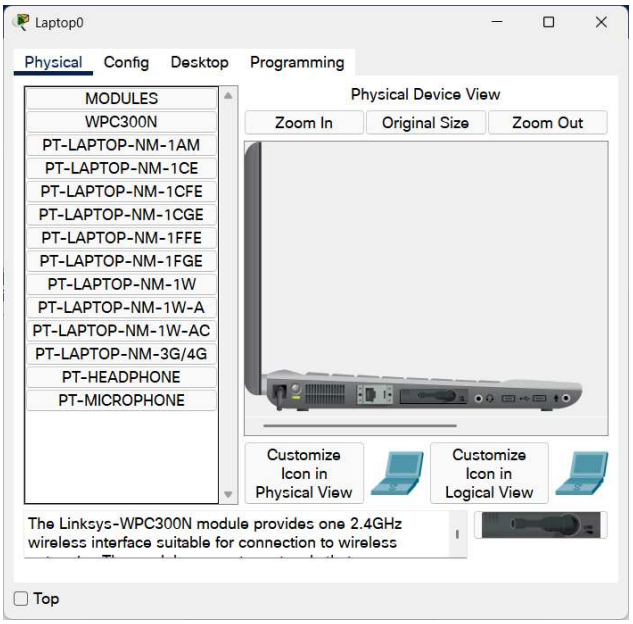
ii. Procedure along with the topology

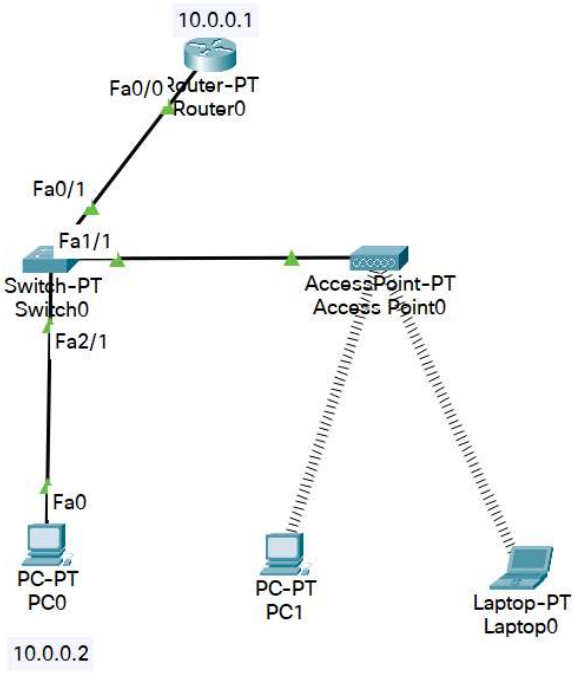


iii. Screen shots/ output

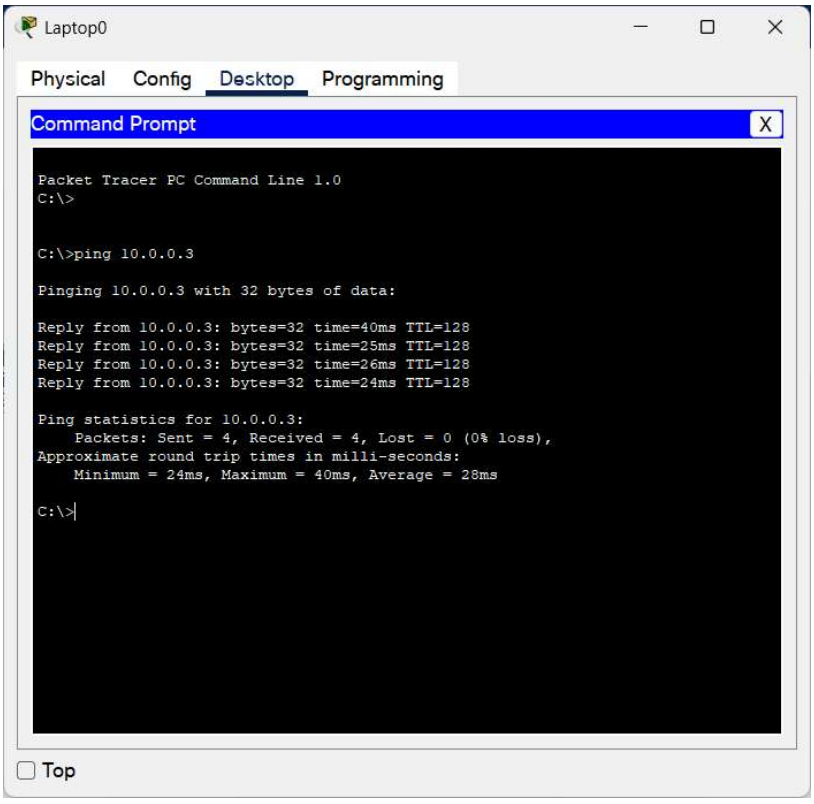



PC1
Physical
Config
Desktop
Programming
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-1W-AC
PT-HOST-NM-3G/4G
PT-HOST-NM-COVER
PT-HEADPHONE
PT-MICROPHONE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for
Top

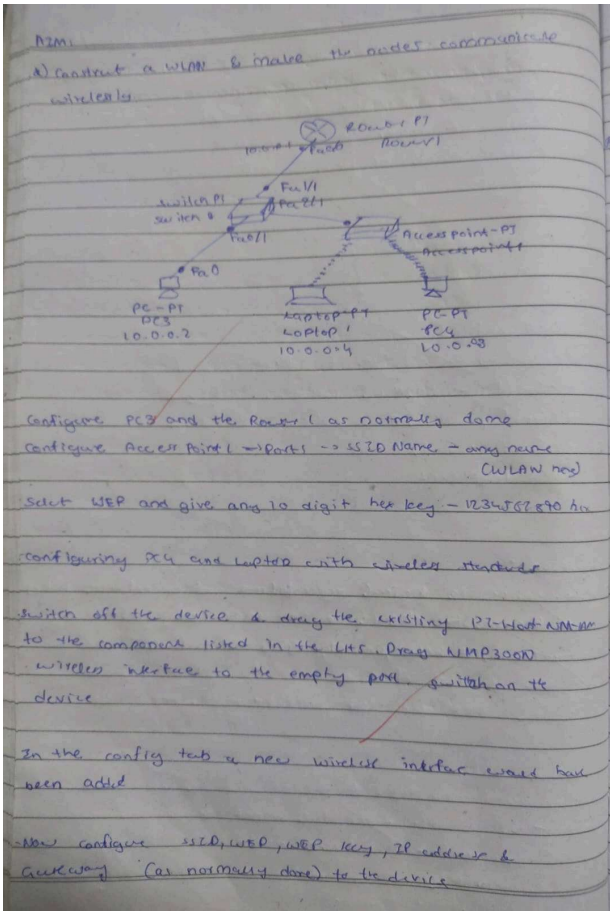

Laptop0
Physical
Config
Desktop
Programming
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-1W-AC
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
Physical Device View
Zoom In
Original 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
Top

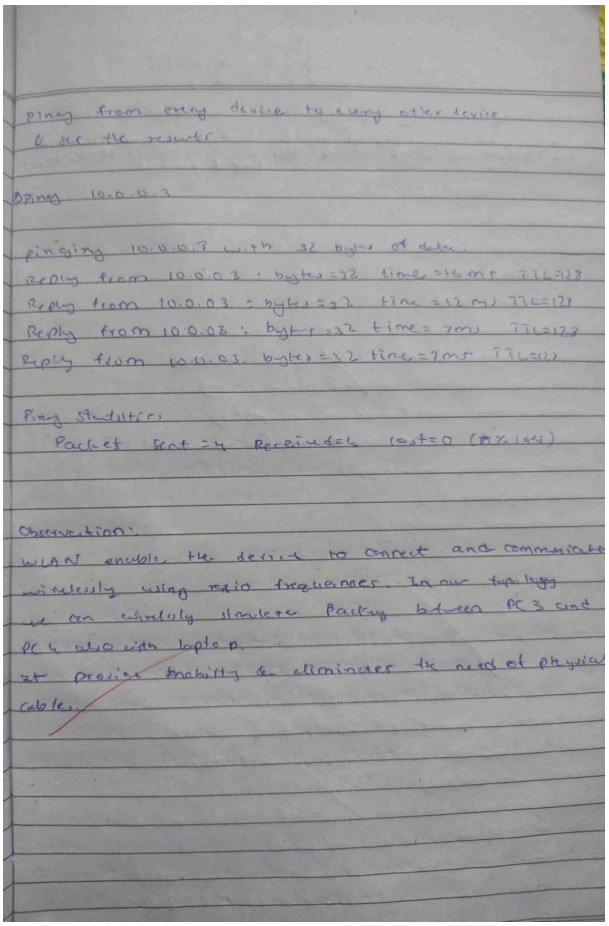

10.0.0.1
Fa0/0
Router-PT
Router0
Fa0/1
Fa1/1
Switch-PT
Switch0
Fa2/1
AccessPoint-PT
Access Point0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC1
Laptop-PT
Laptop0

Ping:


Laptop0
Physical
Config
Desktop
Programming
Command Prompt
X
Packet Tracer PC Command Line 1.0
C:\>
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=40ms TTL=128
Reply from 10.0.0.3: bytes=32 time=25ms TTL=128
Reply from 10.0.0.3: bytes=32 time=26ms TTL=128
Reply from 10.0.0.3: bytes=32 time=24ms 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 = 24ms, Maximum = 40ms, Average = 28ms
C:\>
Top

iv. Observation





Cycle-II

Program 1

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

ii. Procedure

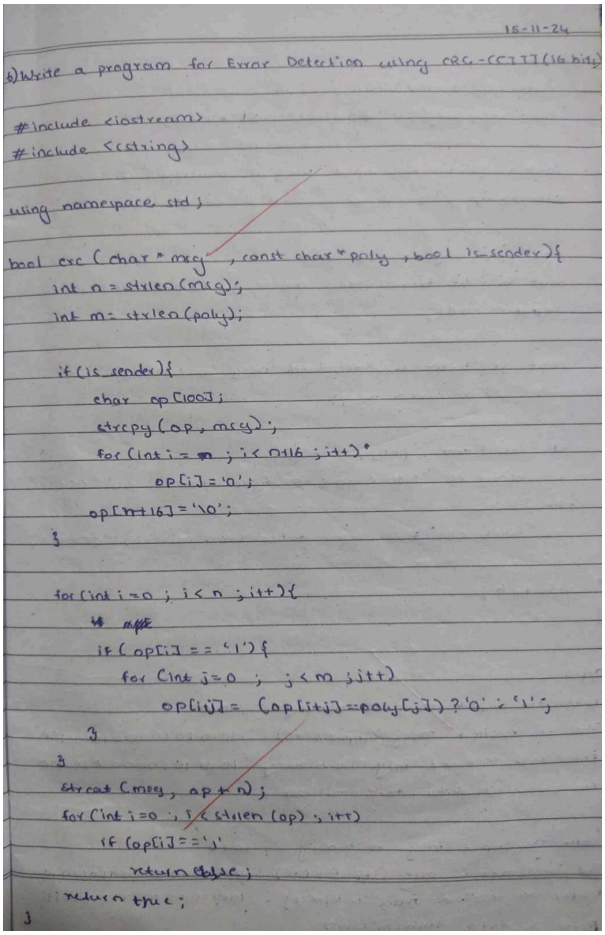
```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15  # Align the bit with CRC's uppermost bit
        for _ in range(1):  # Process the single bit
            if crc & 0x8000:  # Check if the leftmost bit is set
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
            crc &= 0xFFFF  # Ensure CRC remains 16-bit
    return crc


def append_crc_to_bitstream(bitstream: str) -> str:
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f"{crc:016b}"  # Convert CRC to a 16-bit binary string
    return bitstream + crc_bits


def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    if len(bitstream_with_crc) < 16:
        return False  # Not enough bits to contain CRC
    data, received_crc = bitstream_with_crc[:-16], bitstream_with_crc[-16:]
    calculated_crc = crc_ccitt_16_bitstream(data)
    return calculated_crc == int(received_crc, 2)


if __name__ == "__main__":
    # User input for original bitstream
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ")

    # Calculate and append CRC
    bitstream_with_crc = append_crc_to_bitstream(message_bits)
    print(f"Bitstream with CRC: {bitstream_with_crc}")

    # User input for verification
    user_bitstream = input(
        "Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): "
    )
```

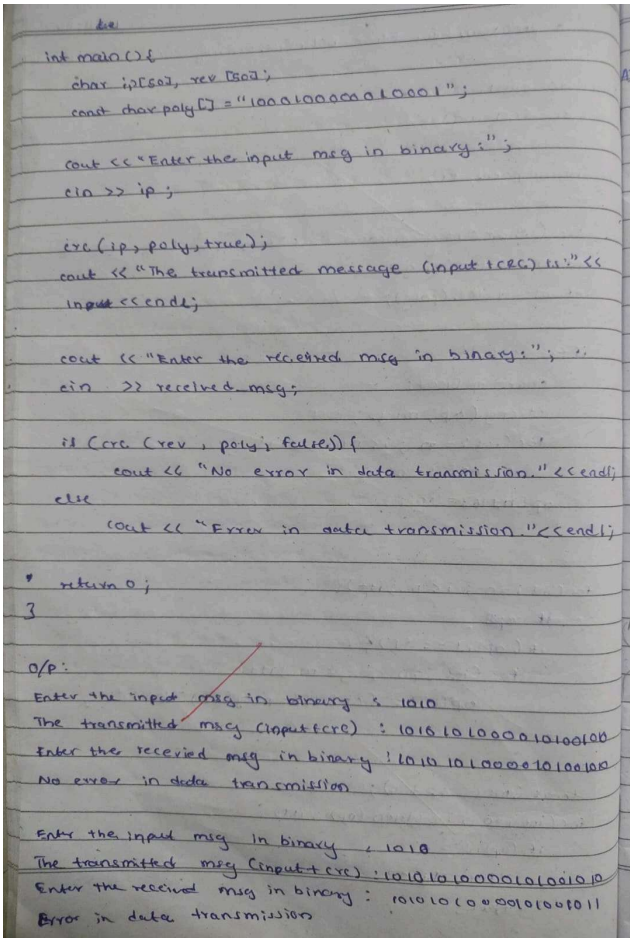
```
# Verify CRC
is_valid = verify_crc_bitstream(user_bitstream)
print(f"CRC valid: {is_valid}")
```

iii. Screen shots/ output

```
In [1]: runcell(0, 'E:/python_files/untitled2.py')

Enter the original bitstream (e.g., 11010011101100): 11111
Bitstream with CRC: 11111111111111100000

Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 11111111111111100000
CRC valid: True

In [2]: runcell(0, 'E:/python_files/untitled2.py')

Enter the original bitstream (e.g., 11010011101100): 11111
Bitstream with CRC: 11111111111111100000

Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 11111111111111100001
CRC valid: False
```

iv. Observation





Program 2

i. Write a program for congestion control using Leaky bucket algorithm

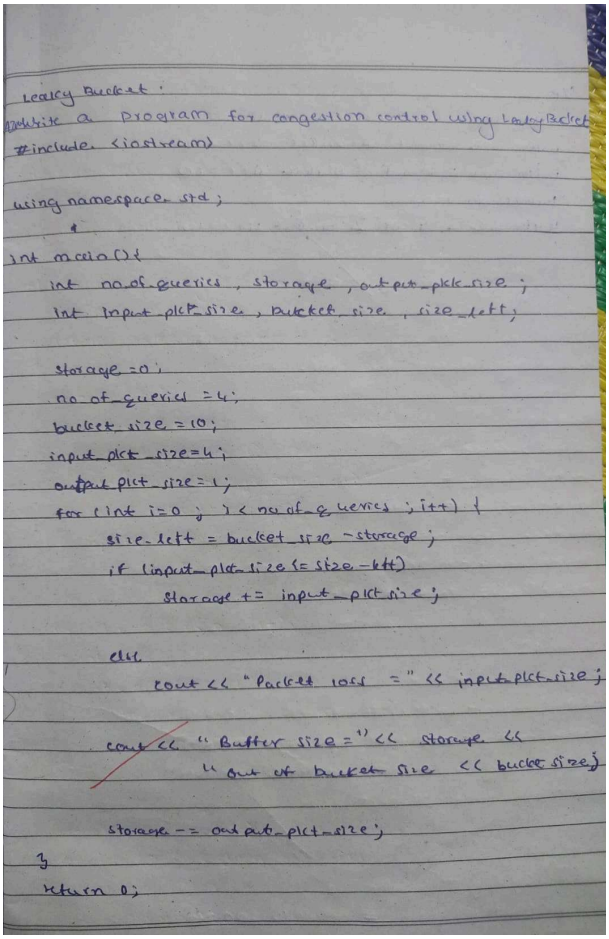
ii. Procedure

```
def main():
    # Initial packets in the bucket
    storage = 0

    # Total number of times bucket content is checked
    no_of_queries = 4

    # Total number of packets that can be accommodated in the bucket
    bucket_size = 10

    # Number of packets that enter the bucket at a time
    input_pkt_size = 4

    # Number of packets that exit the bucket at a time
    output_pkt_size = 1

    for _ in range(no_of_queries):
        # Space left in the bucket
        size_left = bucket_size - storage

        if input_pkt_size <= size_left:
            # Update storage
            storage += input_pkt_size
        else:
            print(f"Packet loss = {input_pkt_size}")

        print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

        # Remove packets from storage
        storage -= output_pkt_size

if __name__ == "__main__":
    main()
```

iii. Screen shots/ output

```
In [3]: runcell(0, 'E:/Engineering/5Sem/CN/Experiments/untitled3.py')
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```

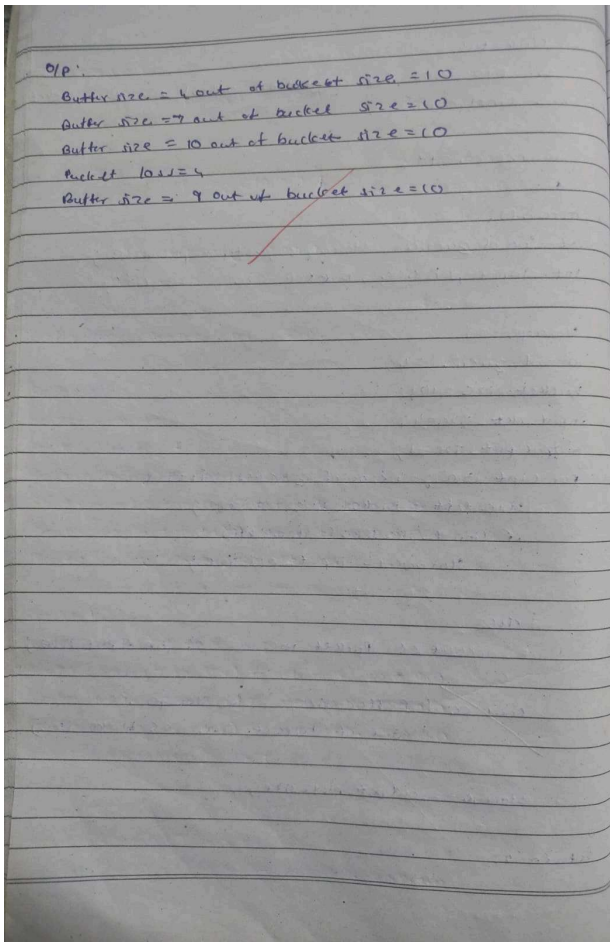
iv. Observation



Leaky Bucket:

Write a program for congestion control using Leaky Bucket

```
#include <iostream>

using namespace std;

int main(){
    int no_of_queries , storage , output_pkt_size ;
    int input_pkt_size , bucket_size , size_left;

    storage =0;
    no_of_queries =4;
    bucket_size =10;
    input_pkt_size=4;
    output_pkt_size=1;
    for (int i=0 ; i< no_of_queries ; i++) {
        size_left = bucket_size -storage;
        if (input_pkt_size <= size_left)
            storage += input_pkt_size;

        else
            cout << "Packet loss =" << input_pkt_size;

        cout << "Buffer size =" << storage <<
                " out of bucket size" << bucket_size;

        storage -= output_pkt_size;
    }
    return 0;
```



O/P:

Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10

Program 3

i. Using TCP/IP 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.

ii. Procedure

clientTCP.py

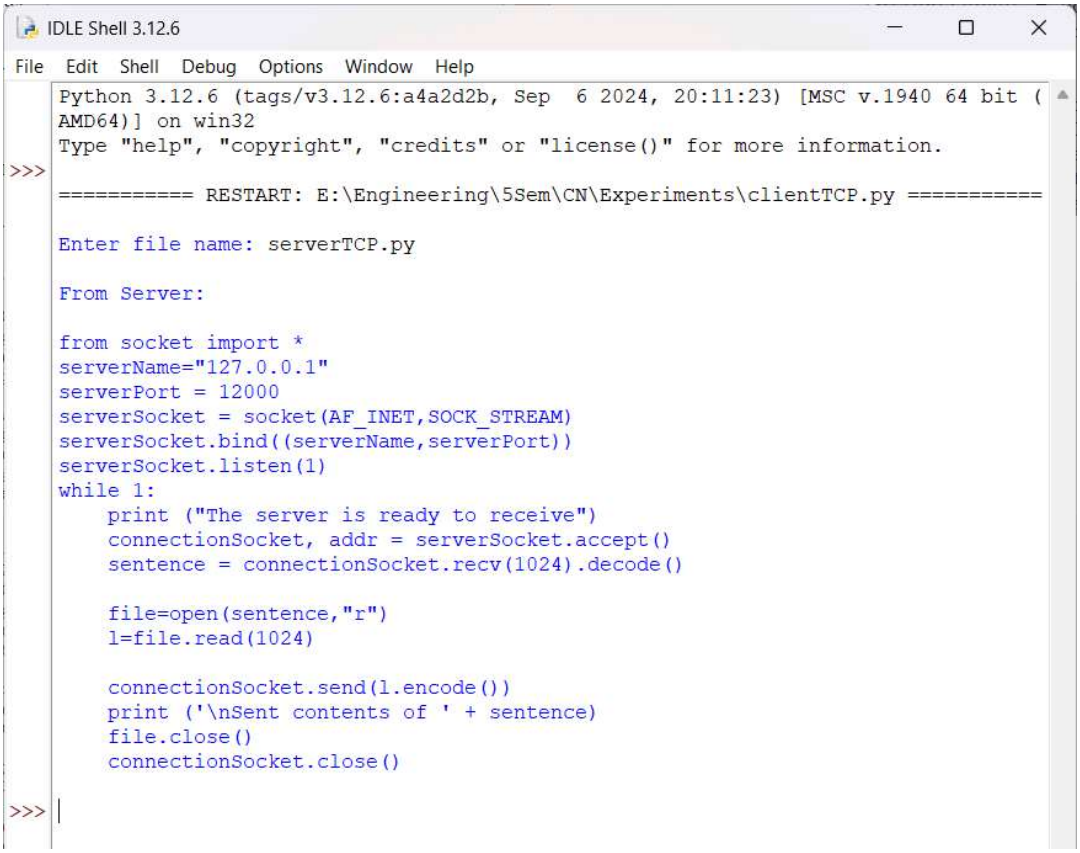
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

serverTCP.py

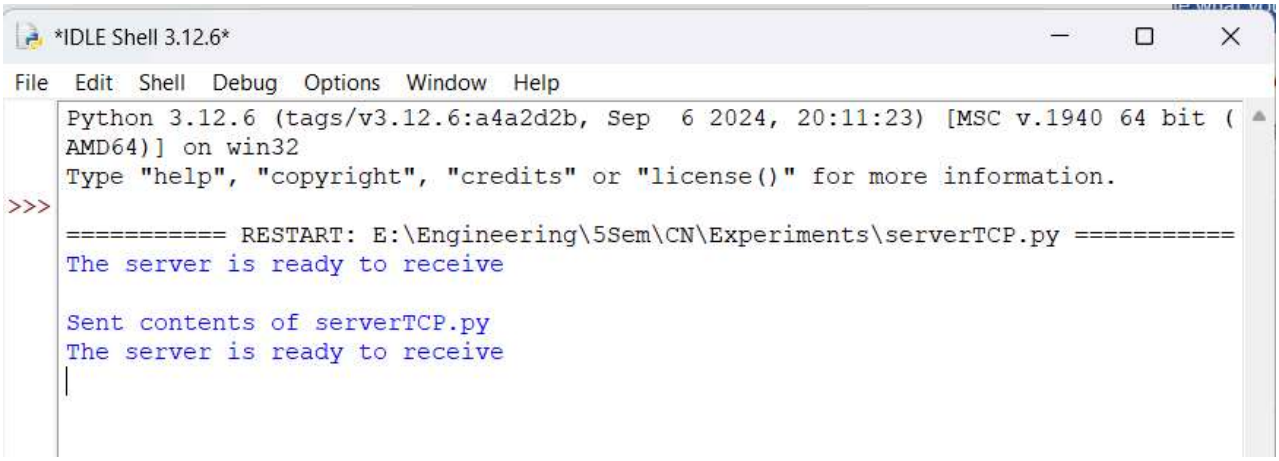
```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

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

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

iii. Screen shots/ output

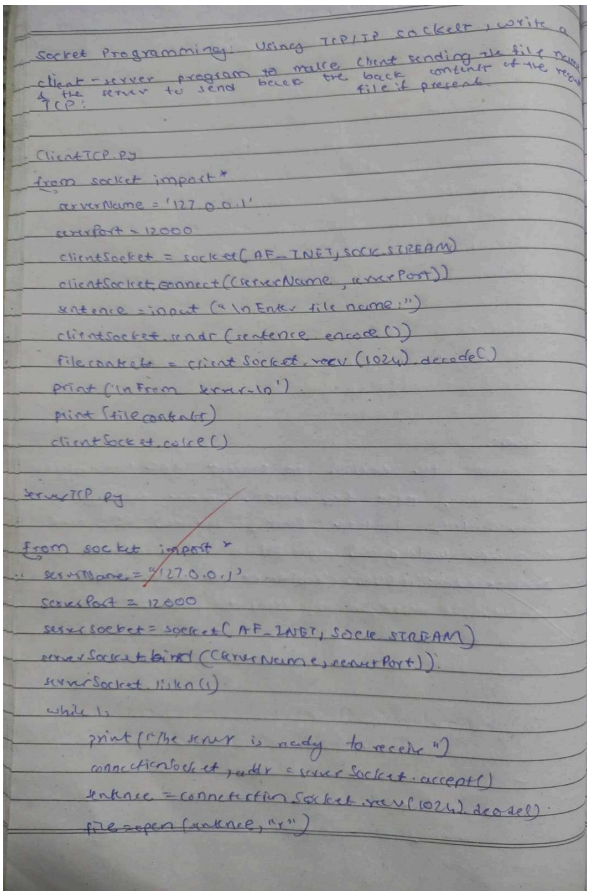
Client



Server



iv. Observation



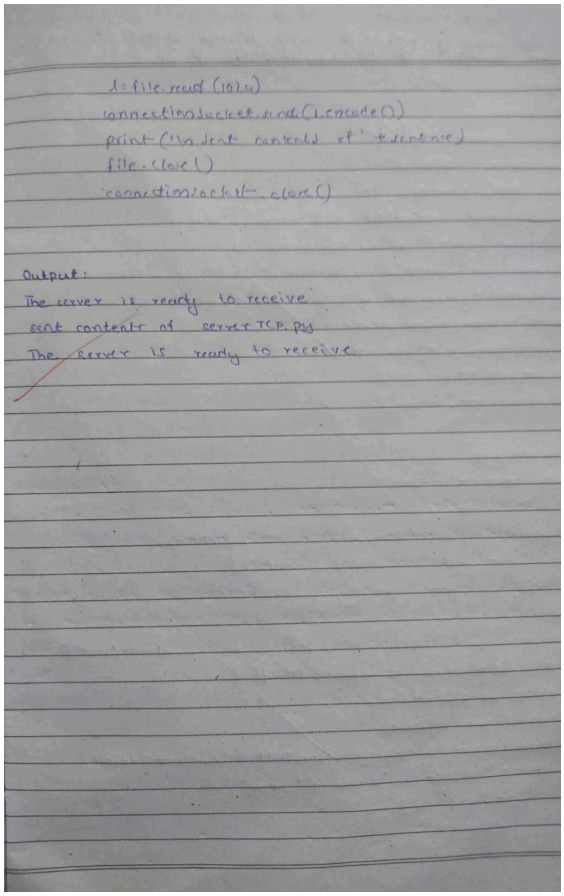
Socket Programming: Using TCP/IP sockets, write a client-server program to make client sending the file name & the server to send back contents of the requested file if present.

ClientTCP.py

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

ServerTCP.py

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



```
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Output:

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

Program 4

i. 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.

ii. Procedure

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()
```

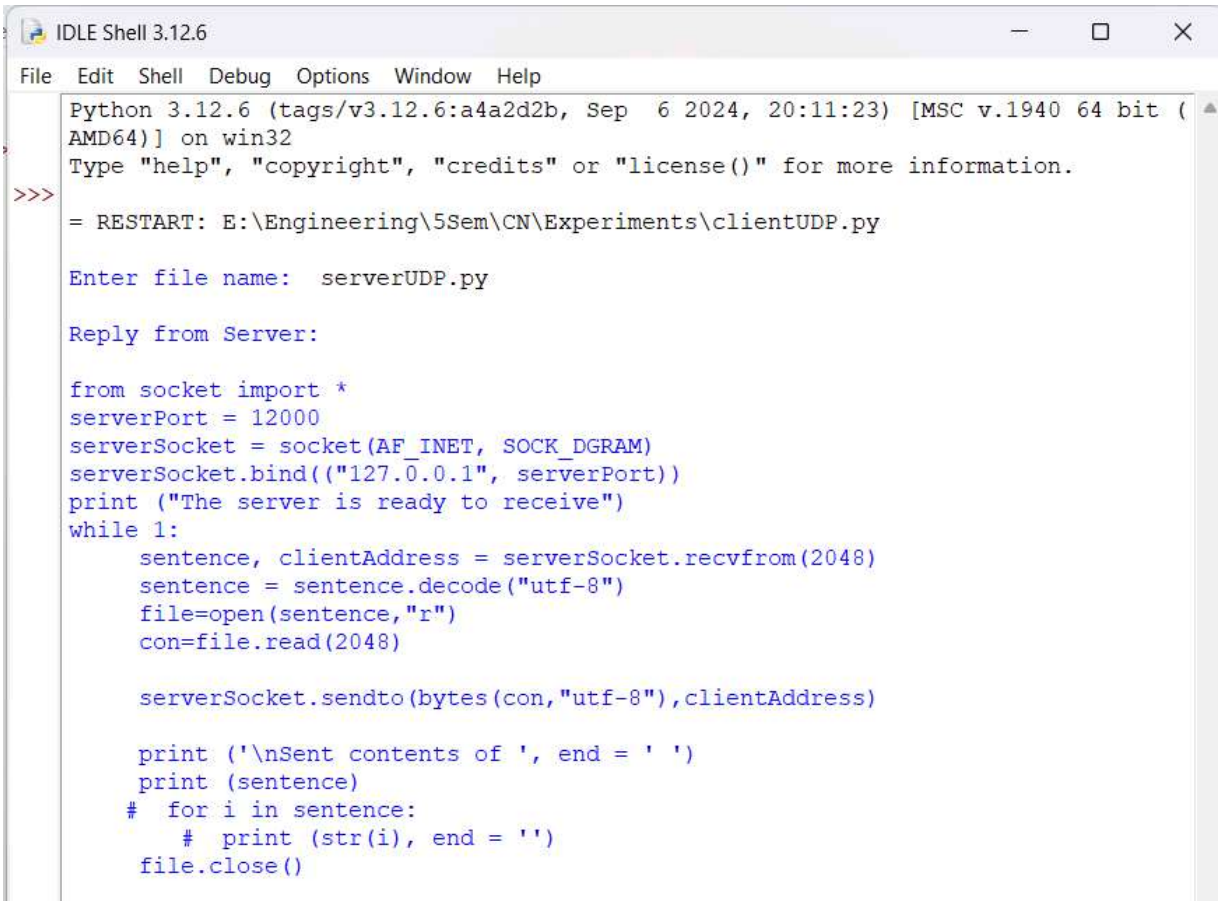
serverUDP.py

```
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:
```

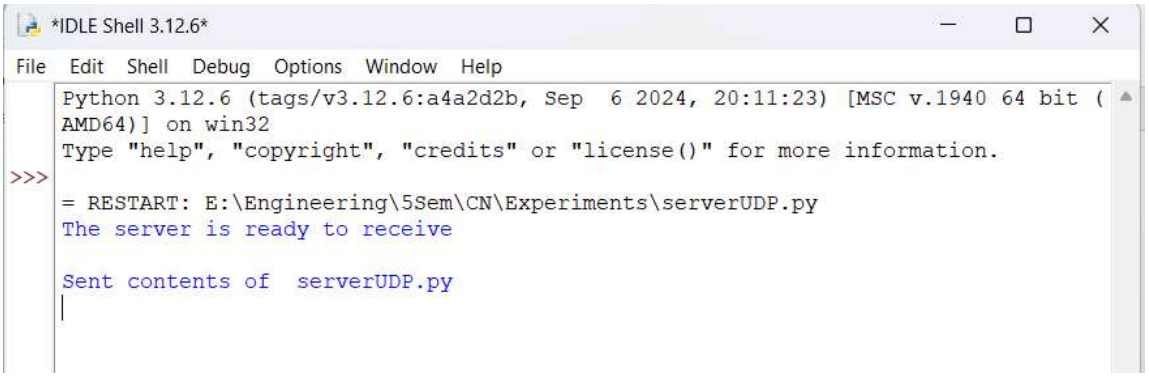
```
    #  print (str(i), end = '')
  file.close()
```

iii. Screen shots/ output

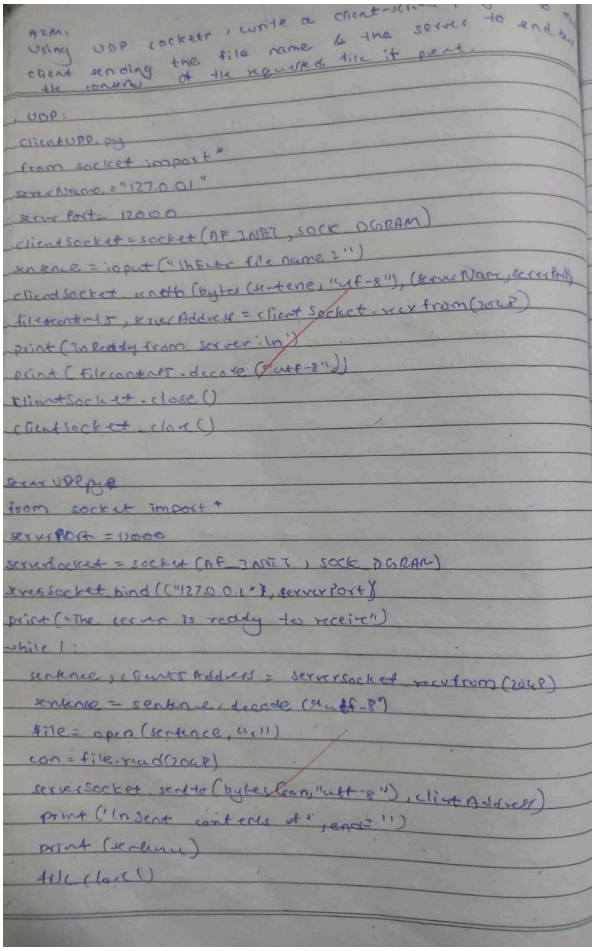
Client



Server



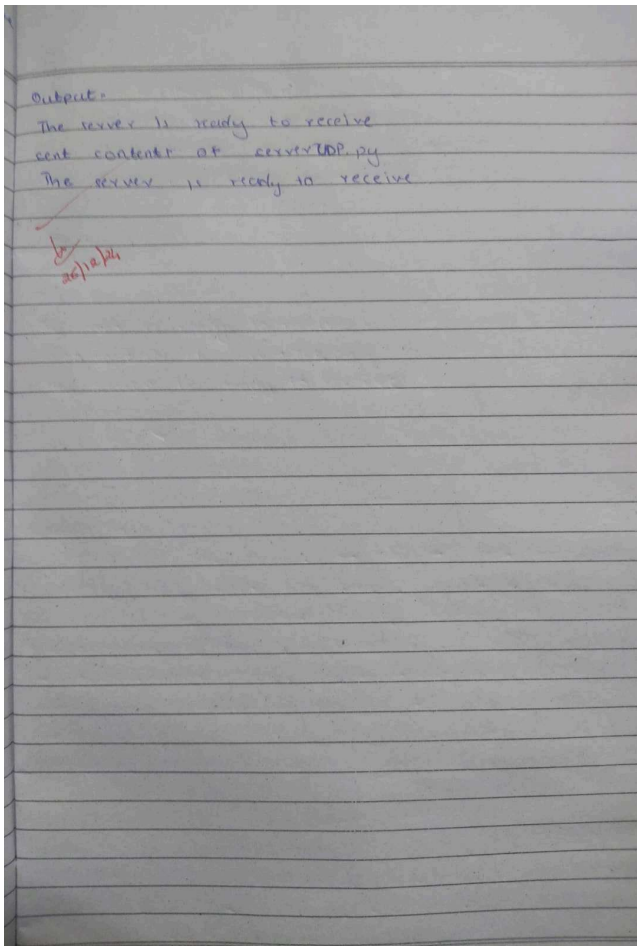
iv. Observation



Aim:
Using UDP socket, write a client-server to send and receive the content of the requested file if present.

UDP:
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))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("\nReply from server:\n")
print(fileContent.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(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", sentence)
print(con)
file.close()



Output:
The server is ready to receive
Sent contents of serverUDP.py
The server is ready to receive