VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Pooja Gaikwad(1BM22CS194)

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 **Pooja Gaikwad (1BM22CS194),** 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.

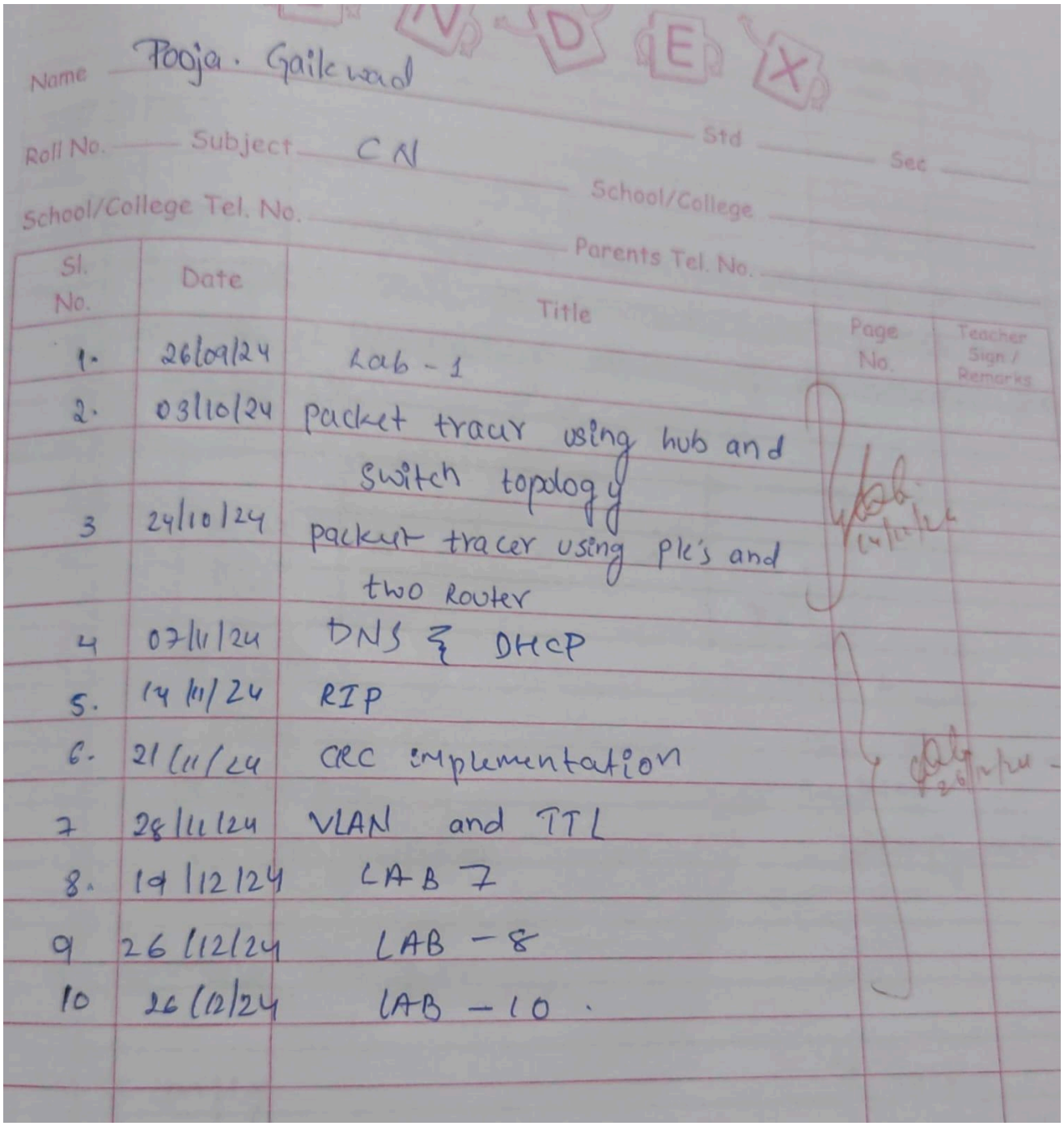
Shyamala G Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index-Cycle-I

Github Link:https://github.com/poojagaikwad10/CN

Index-Cycle-II

Index :



Name Pooja. Gaikwad

Roll No. Subject CN Std Sec

School/College Tel. No. School/College

Parents Tel. No.

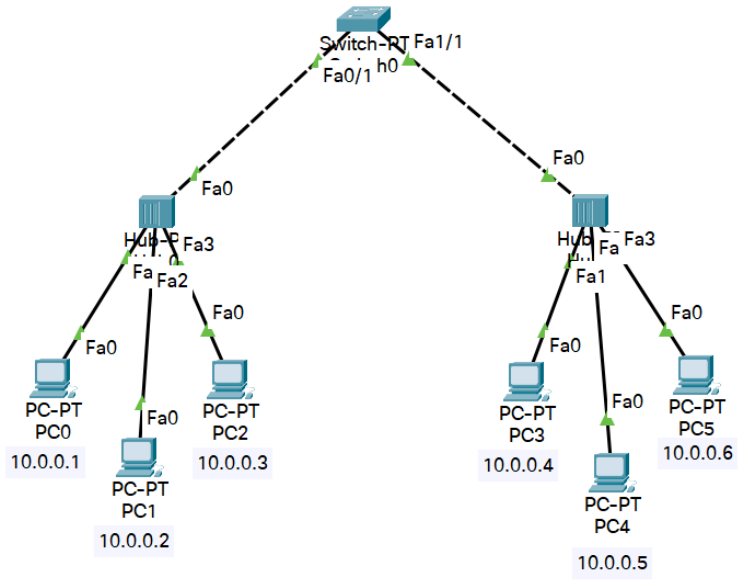
Sl. No.	Date	Title	Page No.	Teacher Sign / Remarks
1.	26/09/24	Lab - 1		
2.	03/10/24	packet tracer using hub and switch topology		
3	24/10/24	packet tracer using pc's and two Router		
4	07/11/24	DNS & DHCP		
5.	14/11/24	RIP		
6.	21/11/24	CRC implementation		
7	28/11/24	VLAN and TTL		
8.	19/12/24	LAB 7		
9	26/12/24	LAB - 8		
10	26/12/24	LAB - 10.		

Cycle-I

Program 1

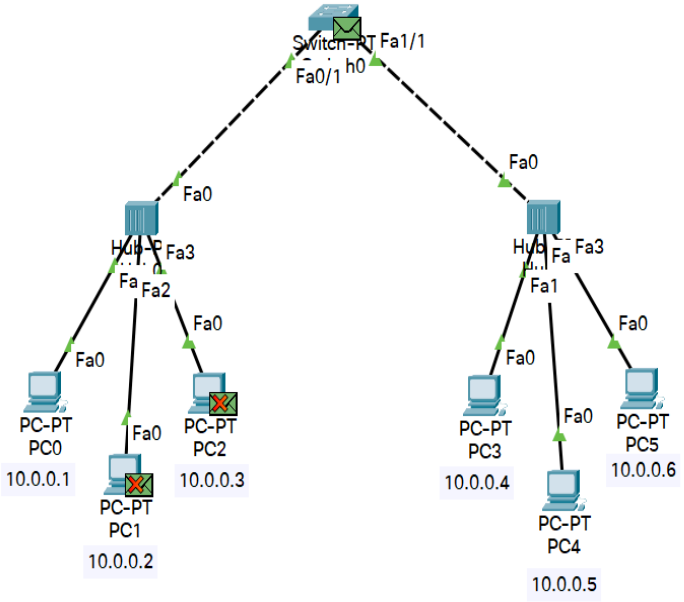
i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.

ii. Procedure along with the topology

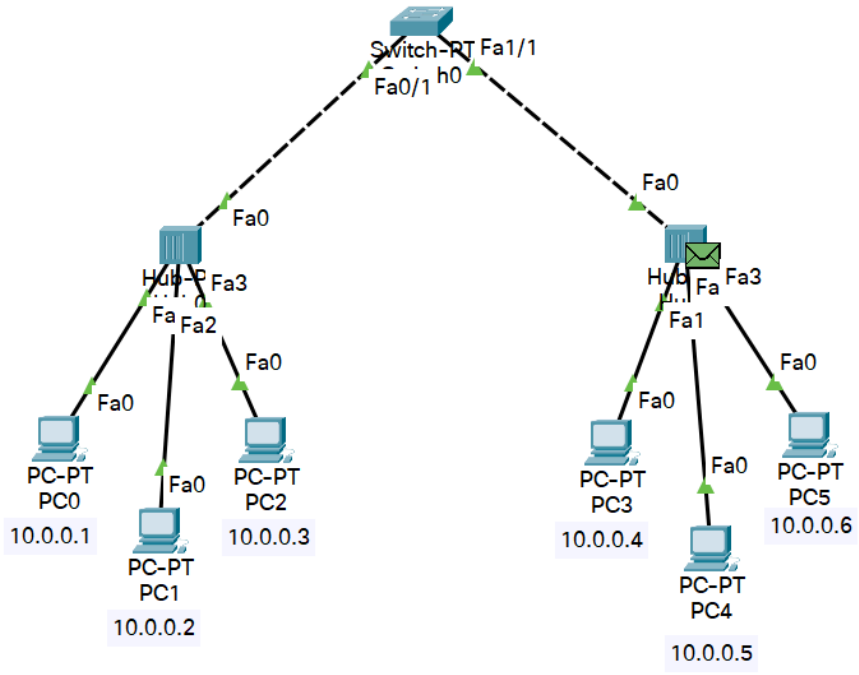


iii. Screen shots/ output

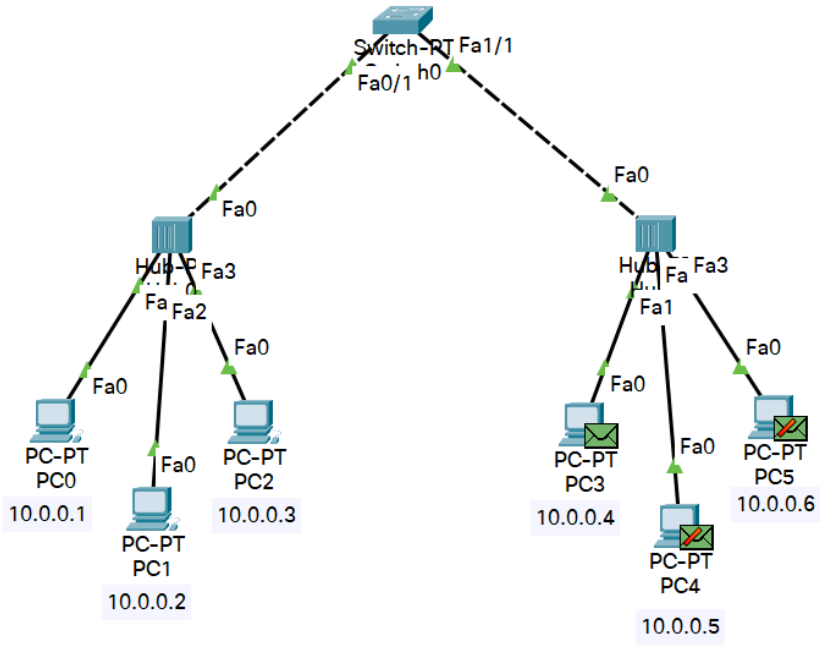
Hub behaviour at sending end



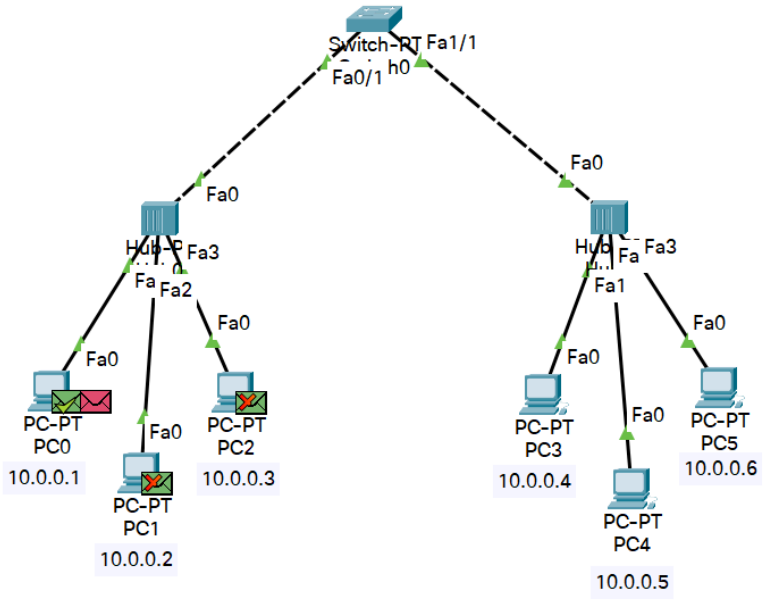
Switch behaviour



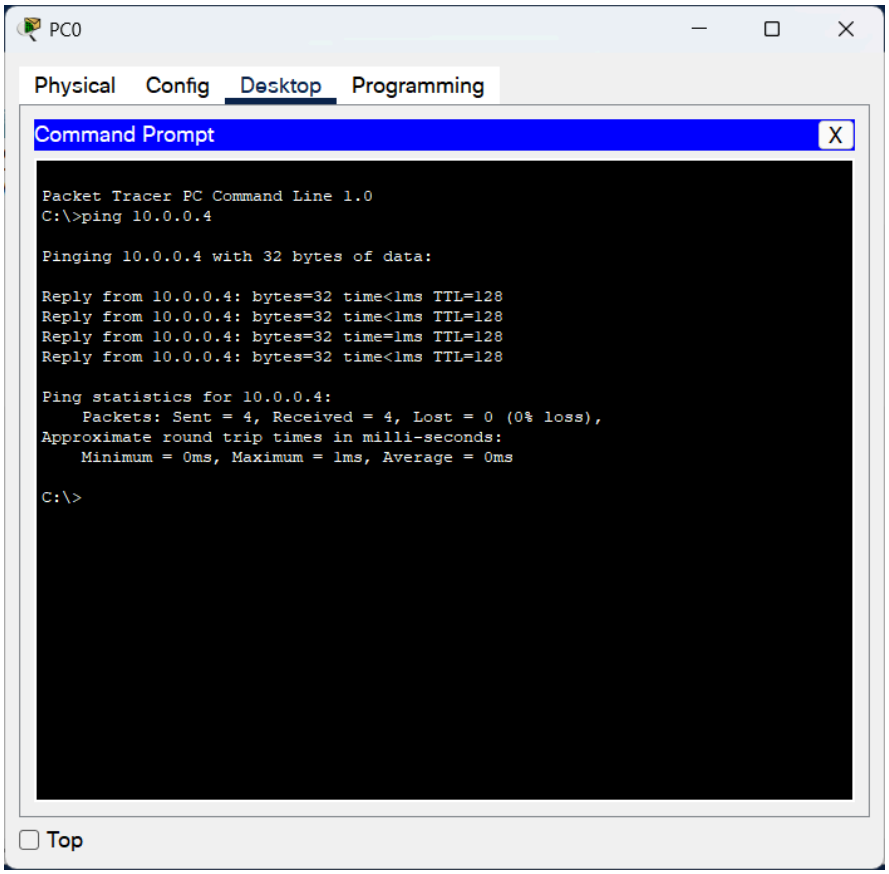
Hub behaviour at receiving end



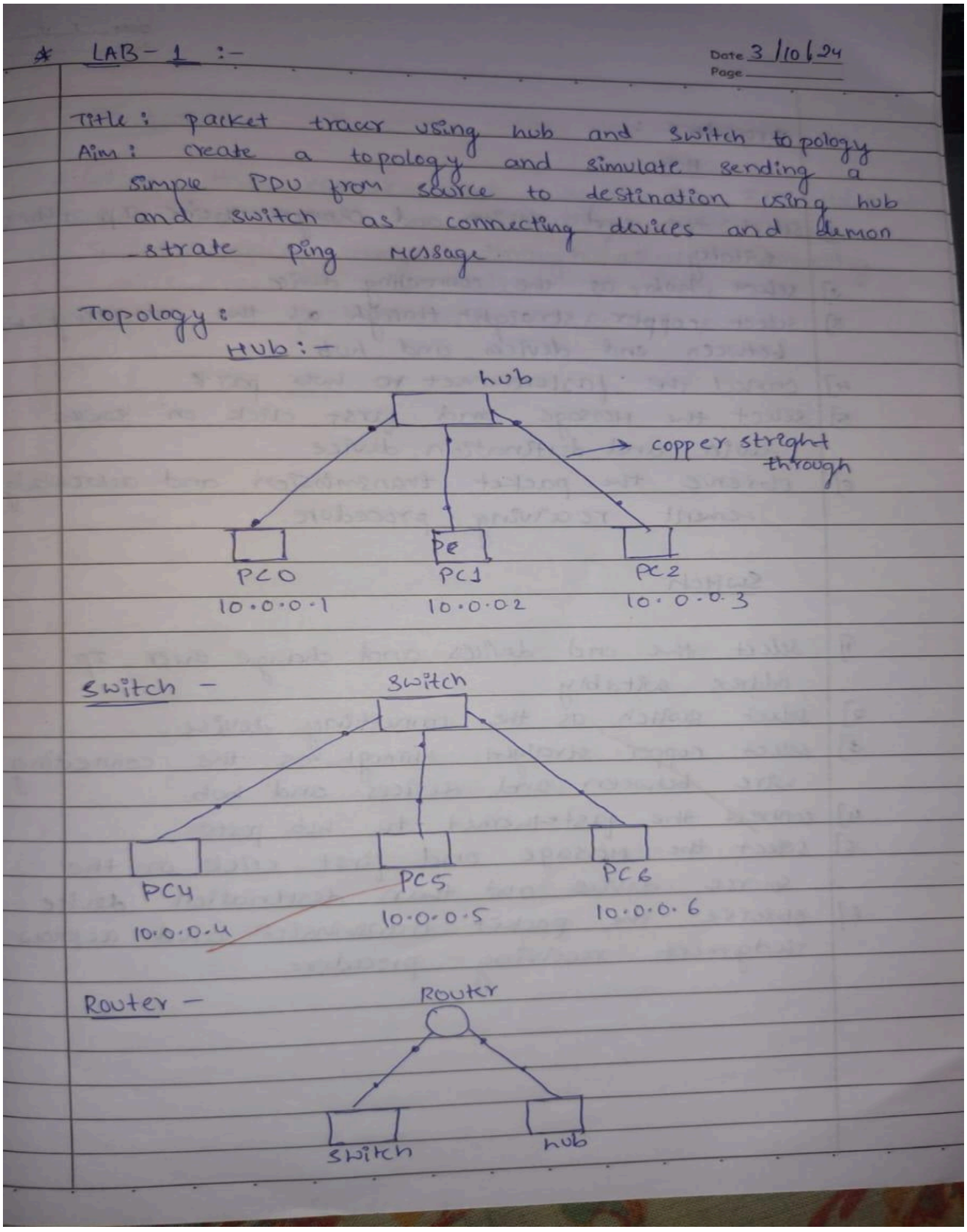
Hub behaviour when back to sender



Ping command to connectivity


PC0
Physical
Config
Desktop
Programming
Command Prompt
X
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4
Pinging 10.0.0.4 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 = 1ms, Average = 0ms
C:\>
Top

iv. Observation



Date __/__/__
Page

procedure :

Hub

1] Select the end devices and change their Ip addresses suitably
2] select hub as the connecting device
3] select copper straight-through as the connecting wire between end devices and hub
4] connect the fasterthernet to hub port
5] select the message and first click on source device and destination device
6] observe the packet transmission and acknowledgement receiving procedure.

Switch

1] Select the end devices and change their Ip address suitably
2] select switch as the connecting device.
3] select copper straight through as the connecting wire between and devices and hub.
4] connect the fastethernet to hub posts.
5] select the message and first click on the source device and than destination device
6] observe the packet transmission and acknowledgment receiving procedure.

Date __/__/__
Page _____

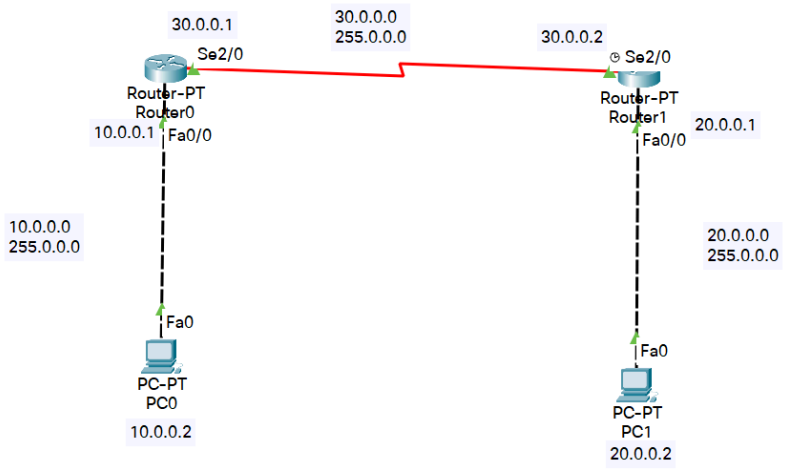
Router :

1] Select the router and change the Ip addresses
2] select router as the connecting device
3] select copper straight - through as the connecting wire between hub, switch & router
4] connect the fast ethernet to Router port.

Program 2

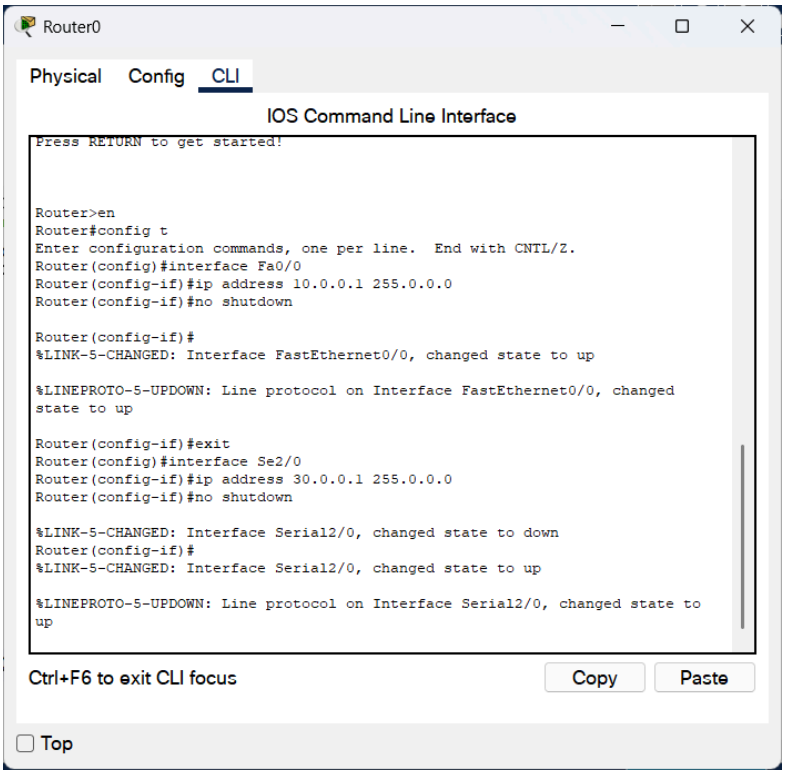
i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.

ii. Procedure along with the topology

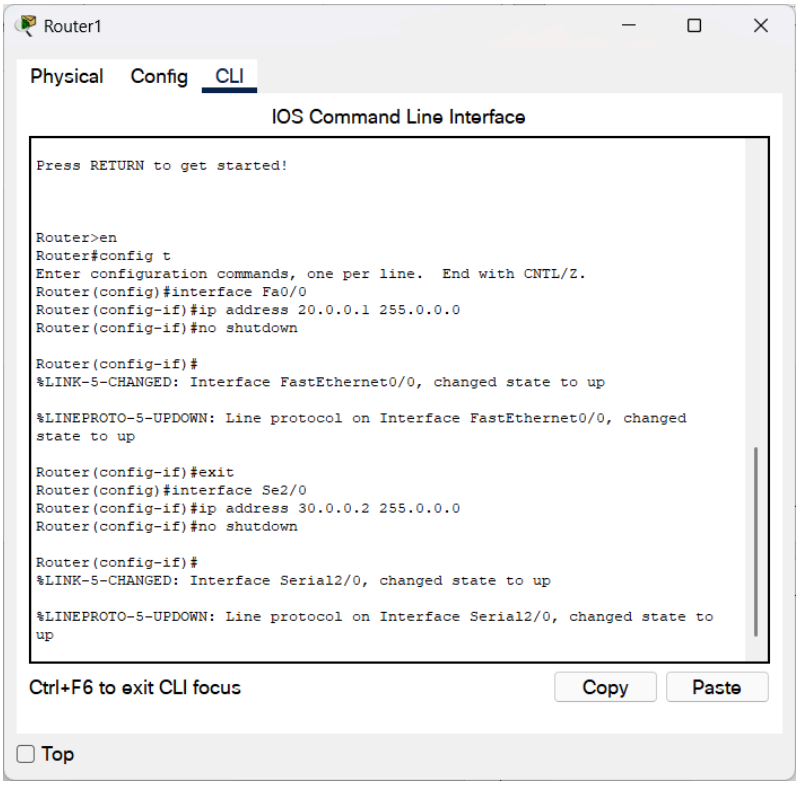


iii. Screen shots/ output

Router0 configuration



Router1 configuration



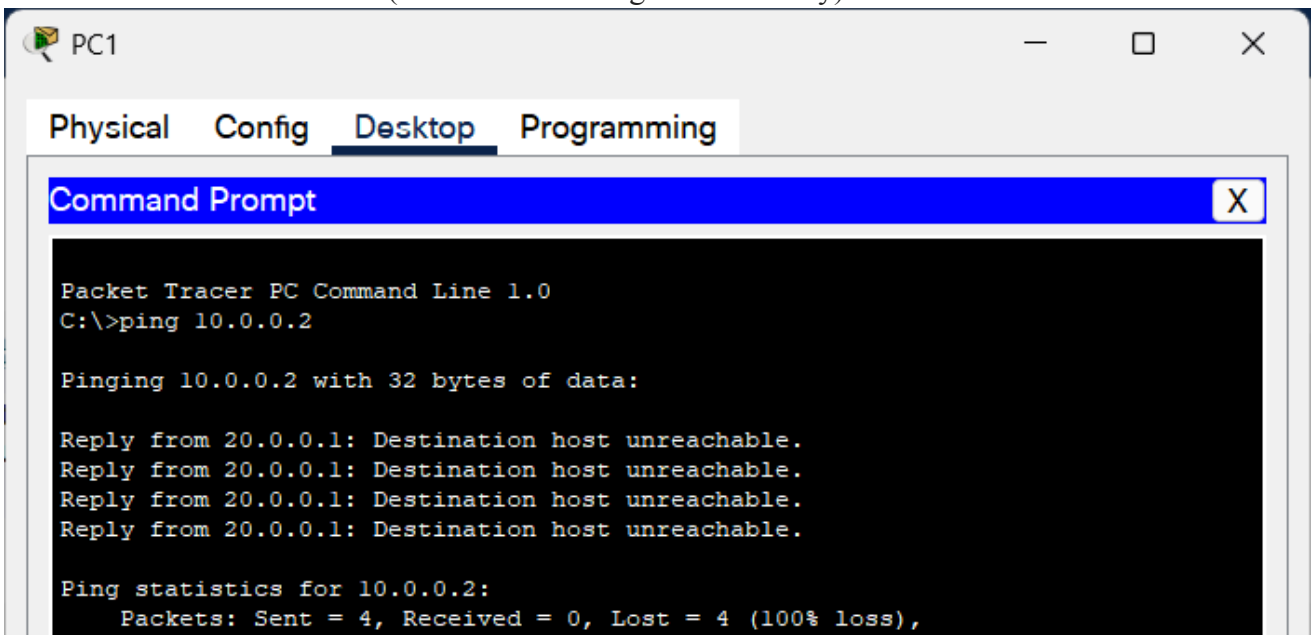
Ip route command in Router0

```
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)#
```

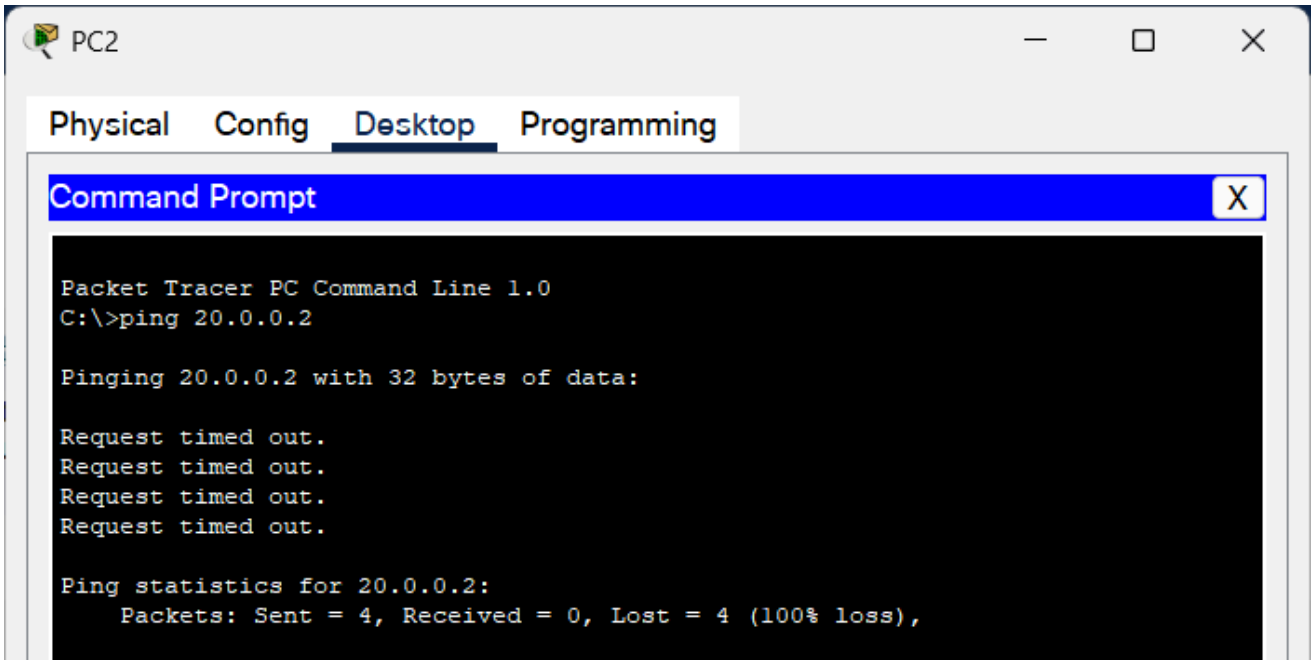
Ip route command in Router1

```
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)#
```

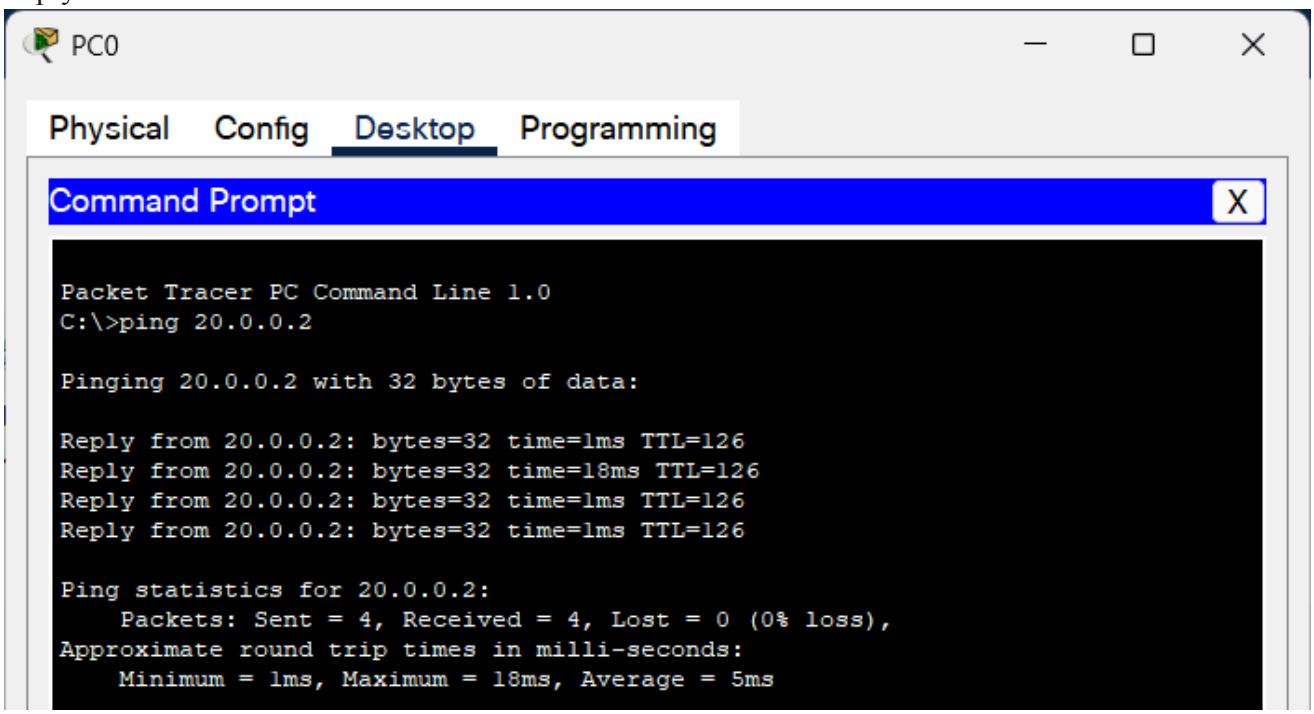
Destination host Unreachable (Before establishing network Fully)



Request Timed Out



Reply from Destination



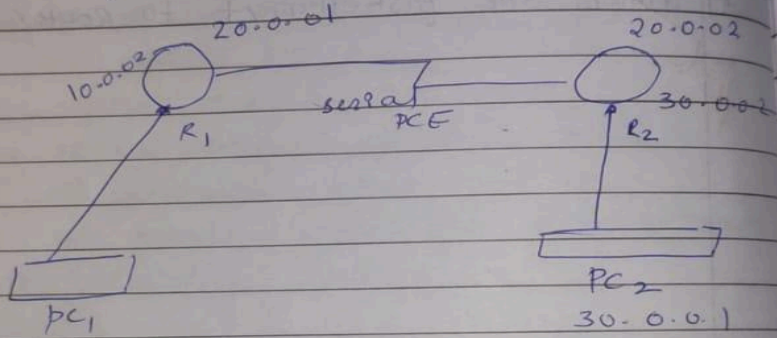
iv. Observation

Date 24/10/24
Page

* LAB-03:

* Title :- packet tracer using PC's and two Router

Aim:- create a topology and simulate sending a simple PDC from source to destination using routers as connecting devices and demonstrating ping message.



* Router procedure :-

1. select a Generic Router a configure them
2. Both the port's need to be configured
   a. interface
      - FastEthernet 0/0
        ip address 10.0.0.2
   b. Serial
      - Serial 2/0
        ip address 20.0.0.1
3. Repeat the same for router 2 also
4. NOW use serial PCE cable to connect Router 2 but the Router 2
   serial
      - Serial 2/0
        ip address 20.0.0.2

Date __/__/__
Page _____

5. The serial cable1 turns green

* PC's

1. Select PC-PT from End devices namely PC-PT PC, PC-PT PC.
2. NOW click on PC-PT PC1 and click on desktop & configure the ip address
3. NOW for the same PC PCPT PC1 configure the gateway 10.0.0.2
4. DO the same for PC PT PC2 giving different ip address & gateway.
5. Use a cross over copperwire to connect these PC's to the Router which makes own fast-ethernet cable thus 2 part of the Router are connected successfully.

Ping Message

Select PC PTC1 and click on desktop
Select PCPTPC1 and click on desktop
Select commdant port and type ping 30.0.0.1
It says sucessful.

* observations:

1. Request time out
   - when the gateway it not connected properly

2. Destination unreachable
   - when the ip address is not configured

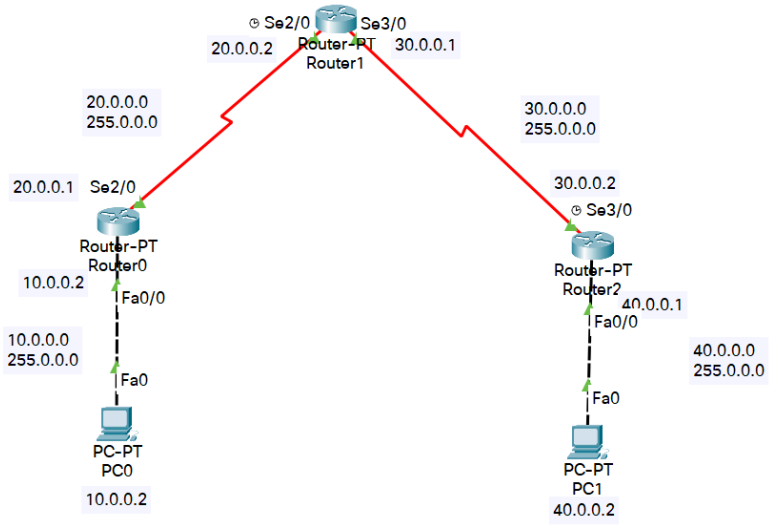
when we ask for ip route it shows the commands of CLI for Router 1:

enable
configure t
hostname R2

Program 3

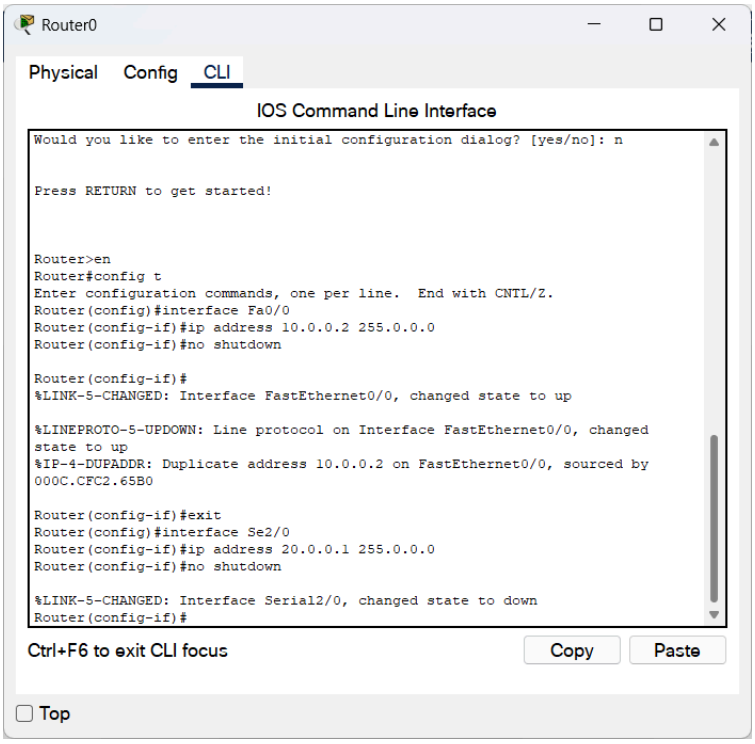
i. Configure default route, static route to the router

ii. Procedure along with the topology

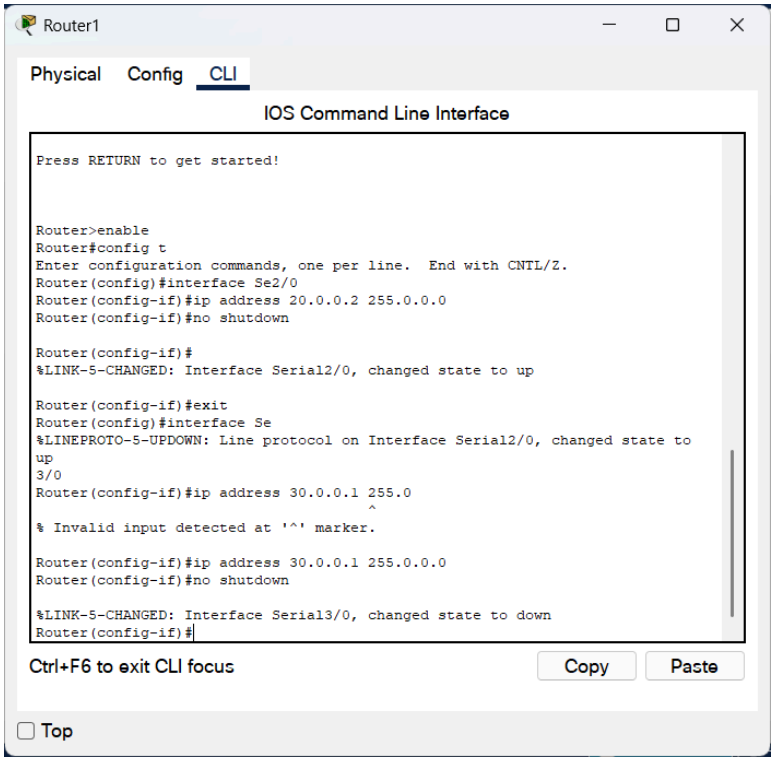


iii. Screen shots/ output

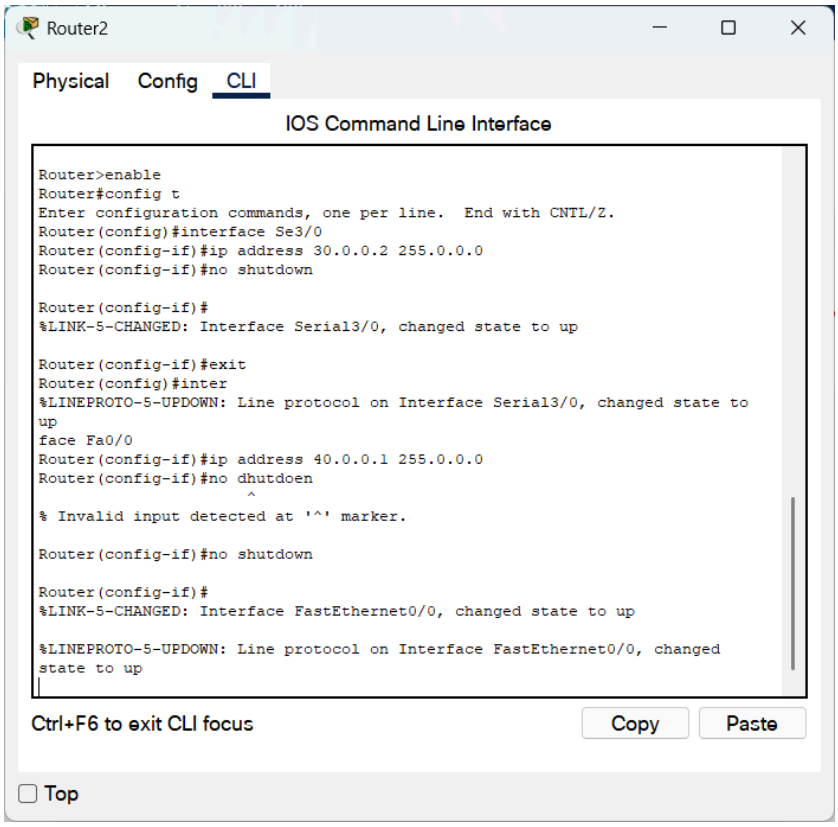
Router0 configuration



Router1 configuration

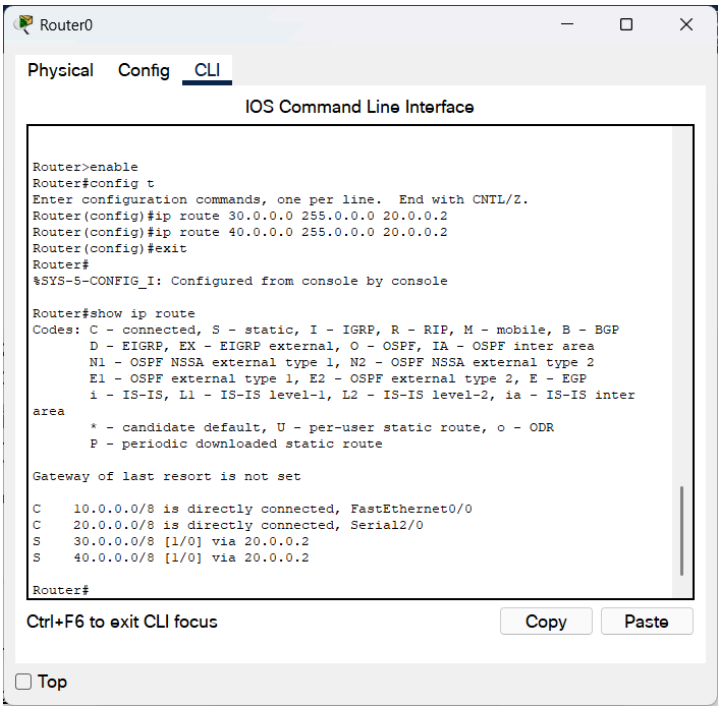


Router2 configuration

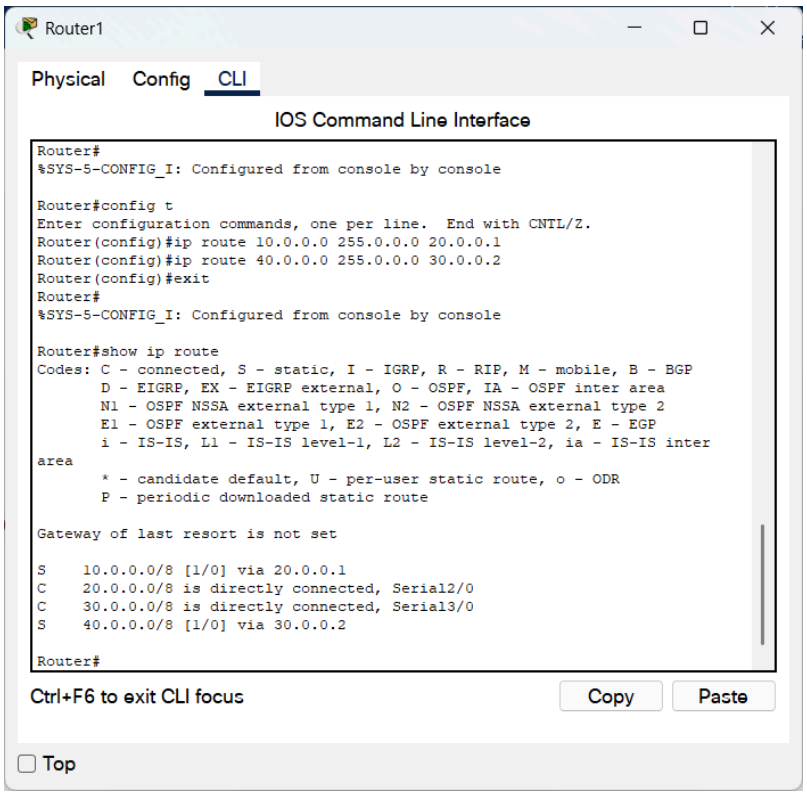


Static Routing:

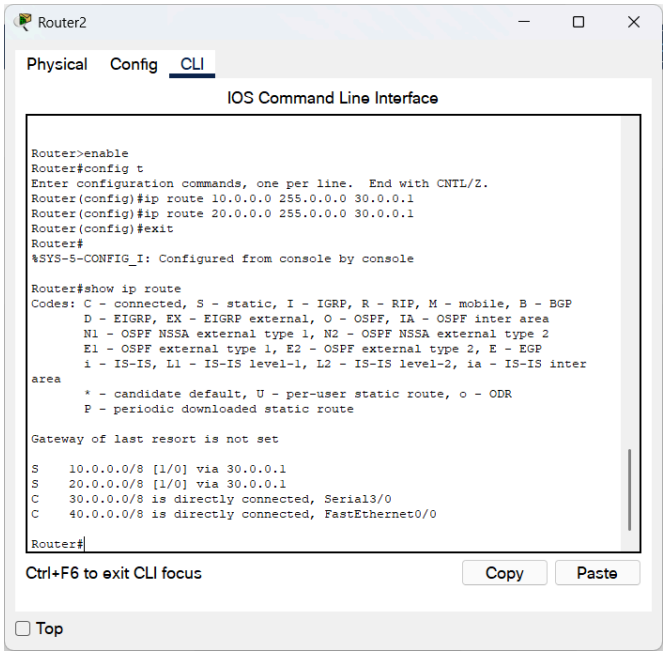
Router0



Router1

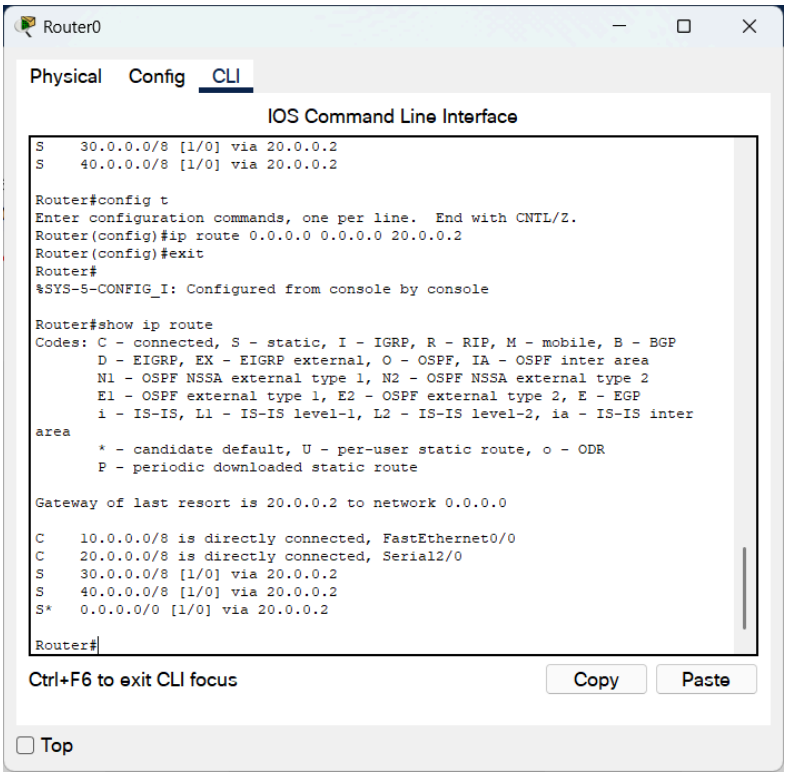


Router2

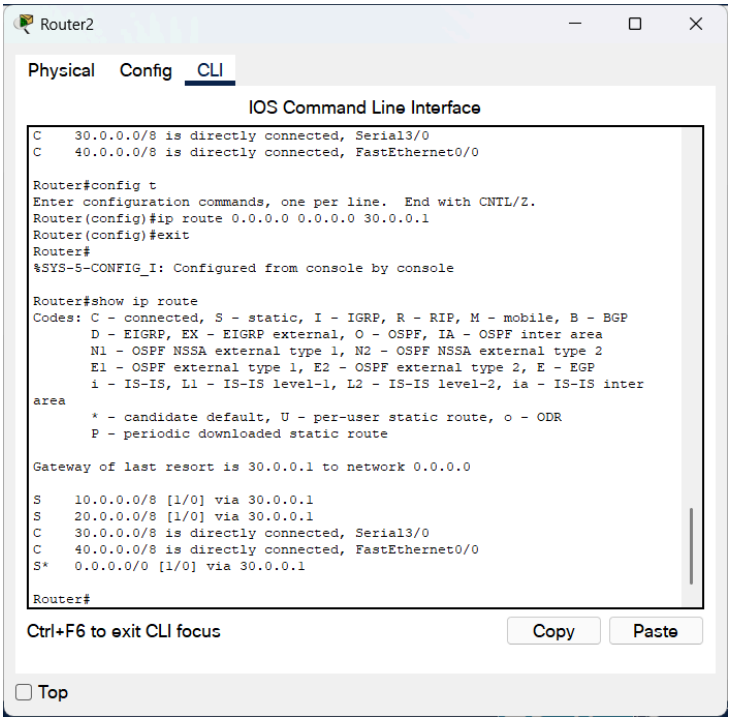


Dynamic Routing:

Route0



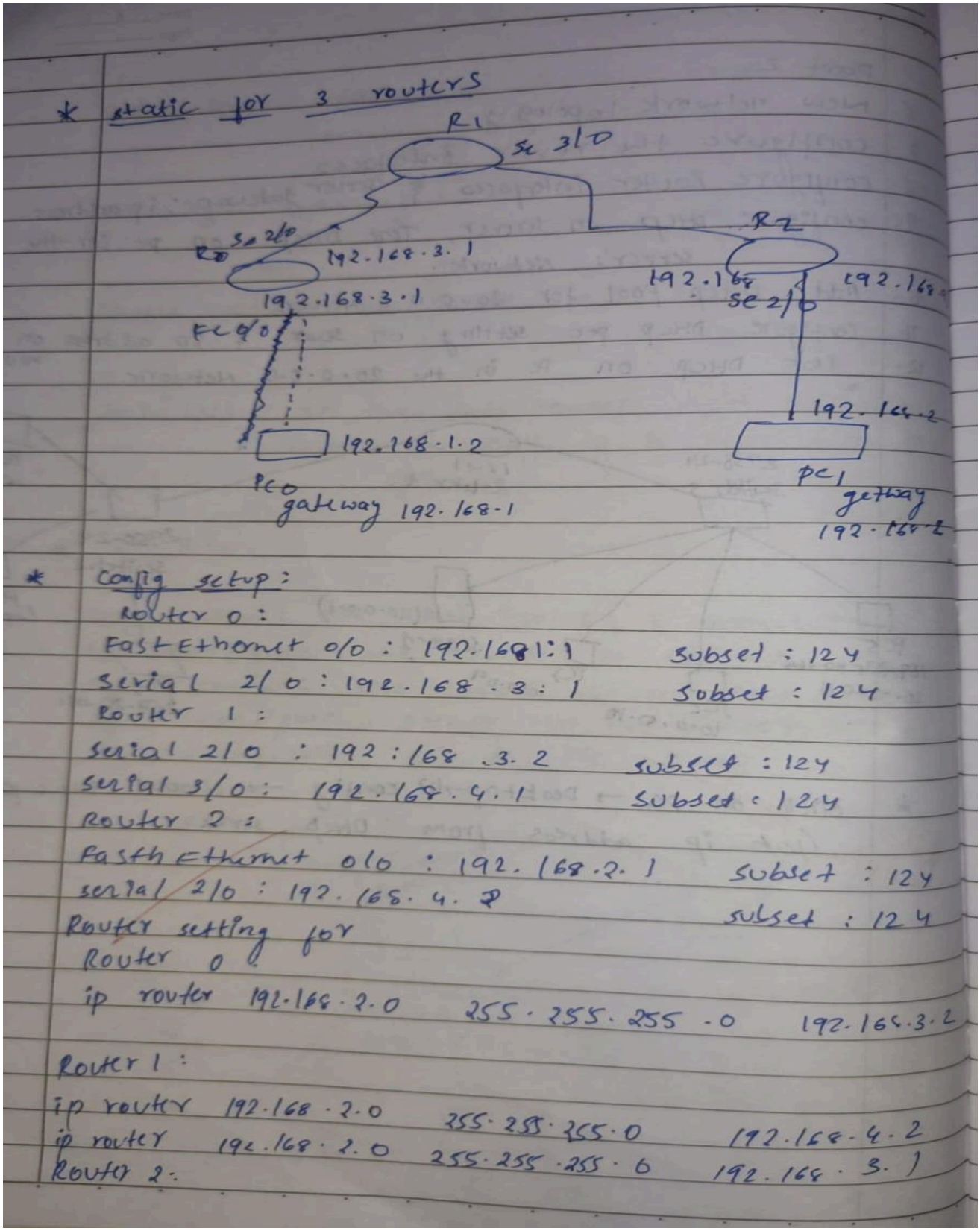
Router2



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



* Config setup:

Router 0:

FastEthernet 0/0 : 192.168.1.1 Subset : 124

Serial 2/0 : 192.168.3.1 Subset : 124

Router 1:

Serial 2/0 : 192.168.3.2 subset : 124

Serial 3/0 : 192.168.4.1 Subset : 124

Router 2:

Fasth Ethernet 0/0 : 192.168.2.1 subset : 124

Serial 2/0 : 192.168.4.2 subset : 124

Router setting for

Router 0

ip router 192.168.2.0 255.255.255.0 192.168.3.2

Router 1:

ip router 192.168.2.0 255.255.255.0 192.168.4.2

ip router 192.168.2.0 255.255.255.0 192.168.3.1

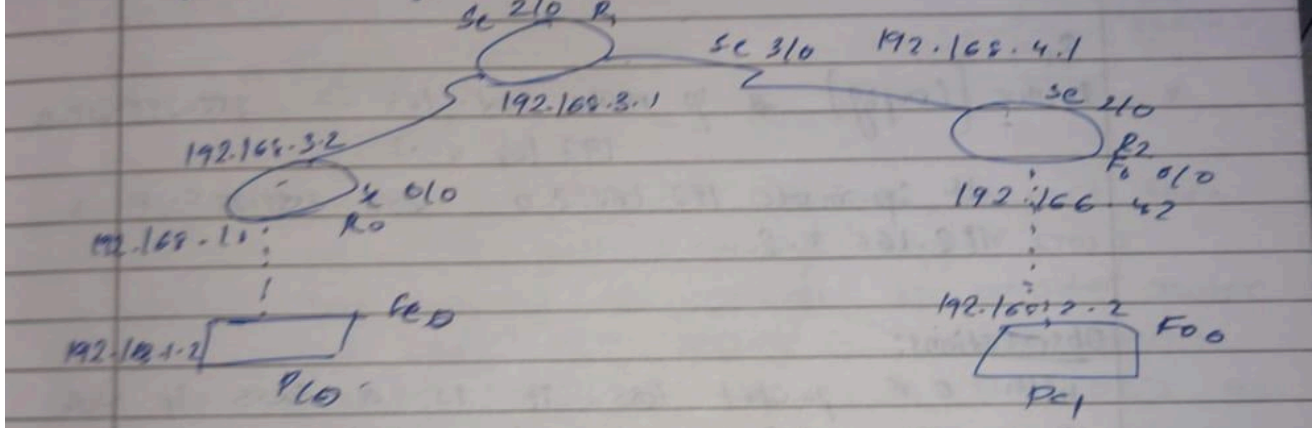
Router 2:

Date / /
Page

ip router 192.168.1.0 255.255.255.0 192.168.4.1

observations:
Ping successful from 192.168.1.2 to 192.168.2.2 with 0% packet config

* Default routing b/w devices:



config setup

R0 :-
Fa 0/0 : 192.168.1.12/24
Se 2/0 : 192.168.3.2/24

PC0 :
F00 : 192.168.1.2/24

R1 :-
Se 2/0 : 192.168.3.12.4
Se 3/0 : 192.168.4.1/2.4

PC1
F00 : 192.168.22/24

R2 :
Se 2/0 : 192.168.4.2/24
Fa 0/0 : 192.168.2.1/24

Date
Page

R_0 :
Router (config) # ip route 0.0.0.0 0.0.0.0
192.168.3.1

R_2 :
Router (config) # ip route 0.0.0.0 0.0.0.0
192.168.4.1

R_1 :
Router (config) # ip route 192.168.20 255.255.255.0
192.168.4.2
ip route 192.168.2.0 255.255.255.0
192.168.3.2

Observations:
with 0% packet loss it is a suss if we ping from 192.168.1.2 to 192.168.2.2

R_0 sends packet R_1 via 192.168.3.0 as if is the only route. It can't be sent to any non existent network, thus it will not fail

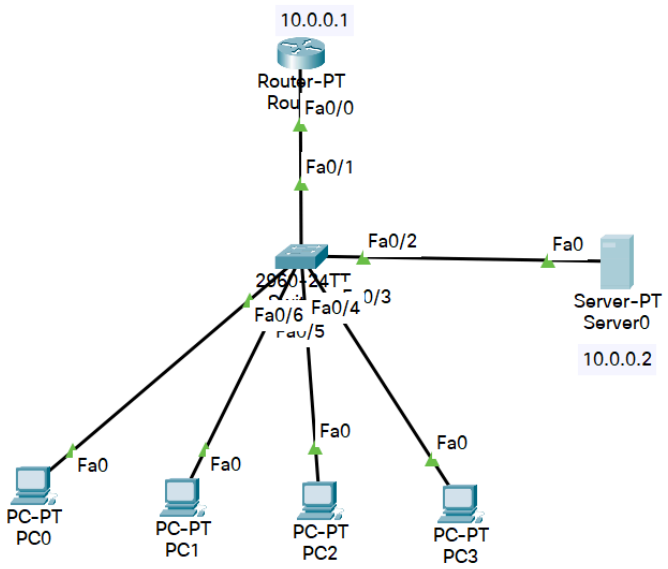
R_1 is configured to send the pack to R_2 through 192.168.4.0. If it was sent through another route (192.168.0.0)

R_2 Sends the pack to 192.168.2.2 as default gateway is set and packet goes from 192.168.1.2 to 192.168.1.1 it is the default gateway
The request can time out if the router is not configured (192.168.2.0 to 192.168.1.0)

Program 4

i. Configure DHCP within a LAN and outside LAN.

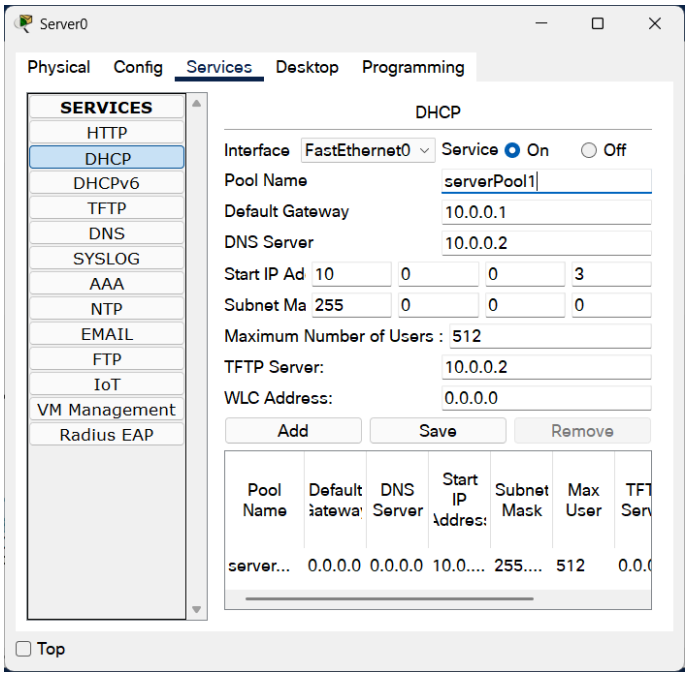
ii. Procedure along with the topology



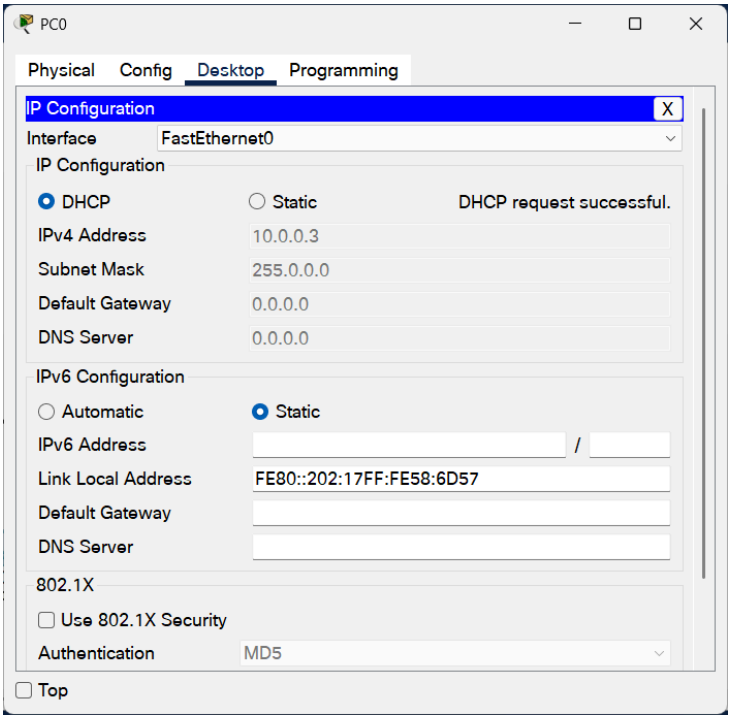
iii. Screen shots/ output

DHCP Within LAN

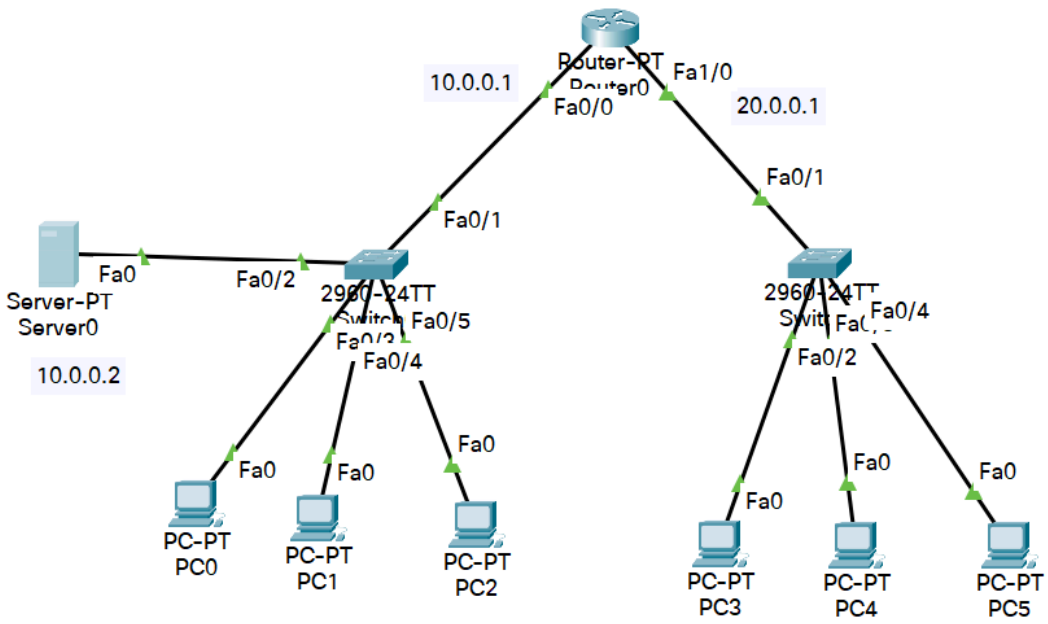
DHCP Configuration



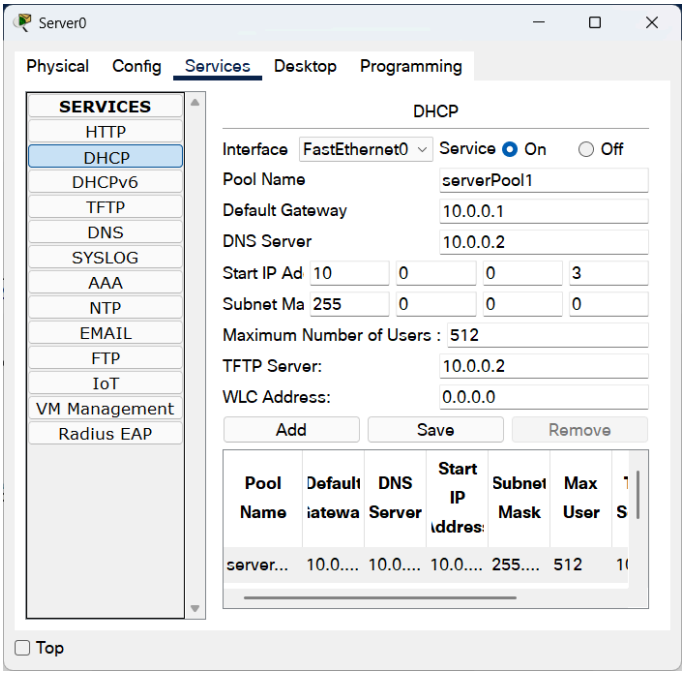
PC settings



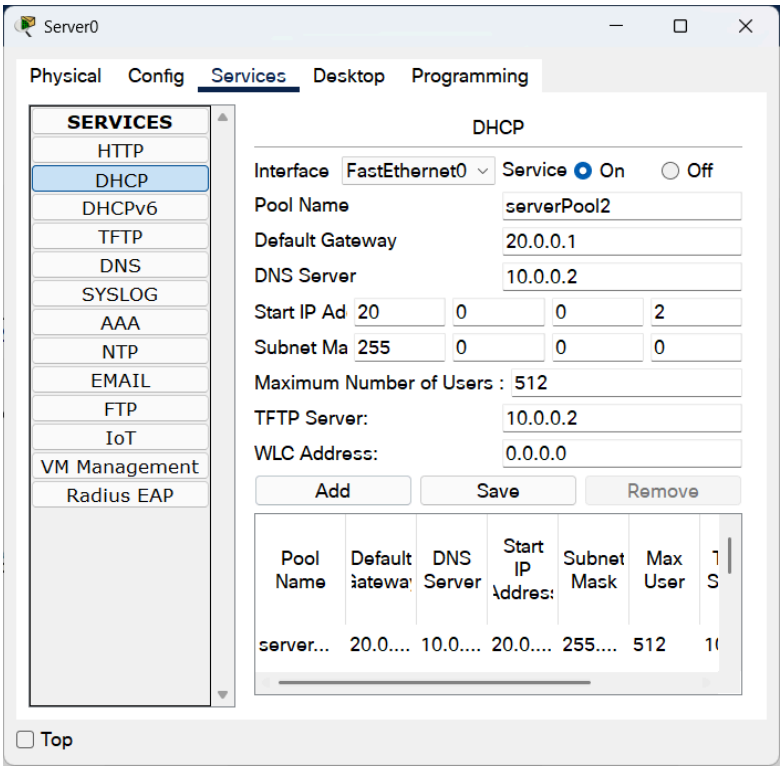
DHCP outside LAN:



DHCP configuration for inside LAN



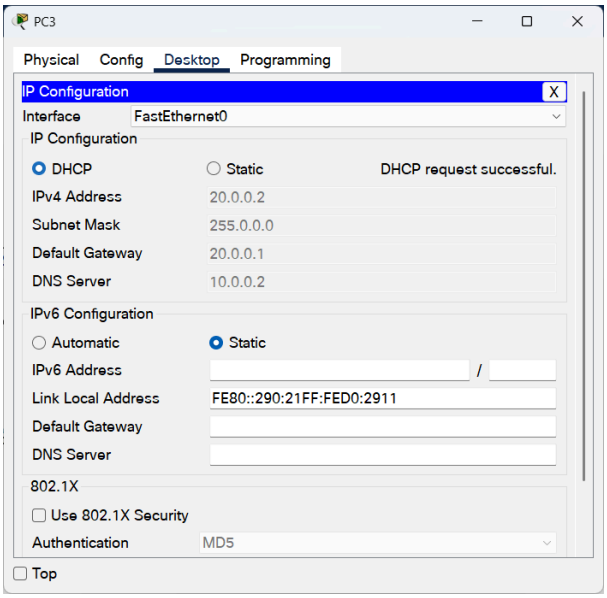
DHCP configuration for outside LAN



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)#
```

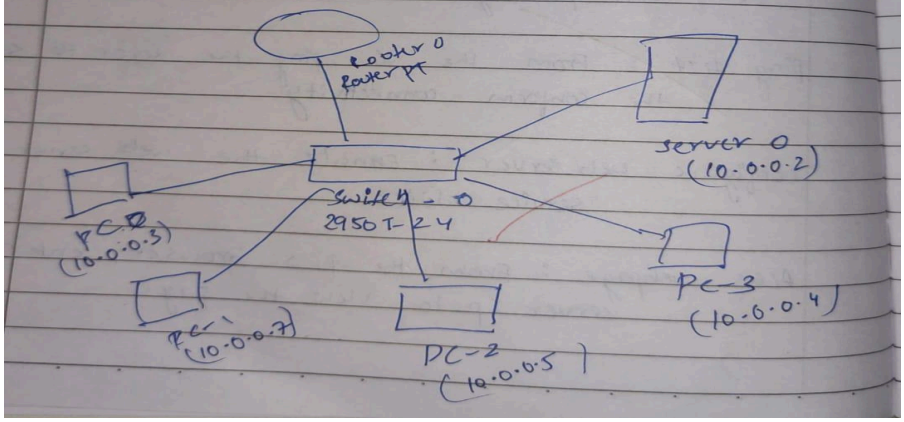
PC setting in another network



iv. Observation

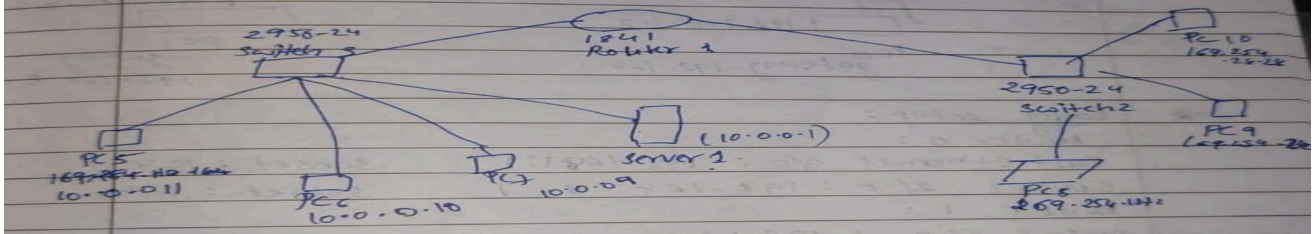
* DHCP:

Step 1: create the LAN Topology & connect them
S_2 : configure the router Ip Address
S_3 : configure the server as a DHCP server
S_4 : Configure DHCP in the server
S_5 : configure PCs to use DHCP



Part - 2

6. New network topology.
7. configure the router interfaces.
8. configure Router interfaces & server gateway ip address.
9. configure DHCP on server, Test DHCP on PC in the server's network.
10. Add DHCP pool for 20.0.0.0 Network.
11. Configure DHCP pool setting on sever & ip address on router.
12. Test DHCP on PC in the 20.0.0.0 Network.

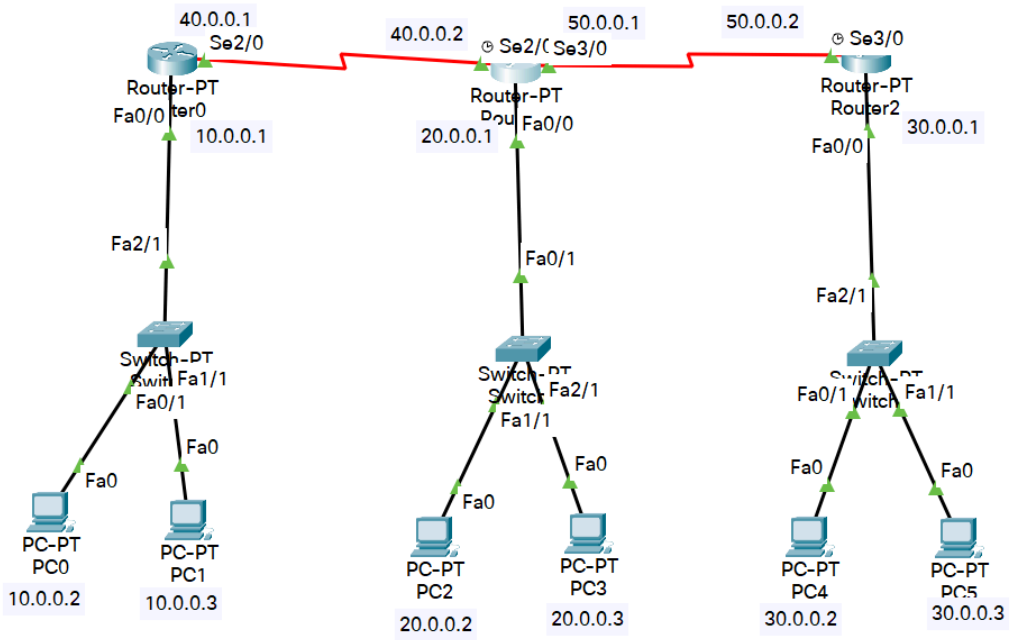


* click on PC → Desktop → IP config → choose DHCP.
Got ip address from DHCP server.

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
```

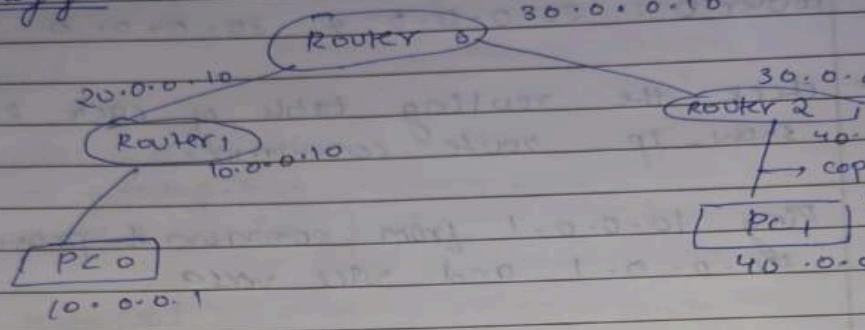
iv. Observation

Date 14/11/24
Page

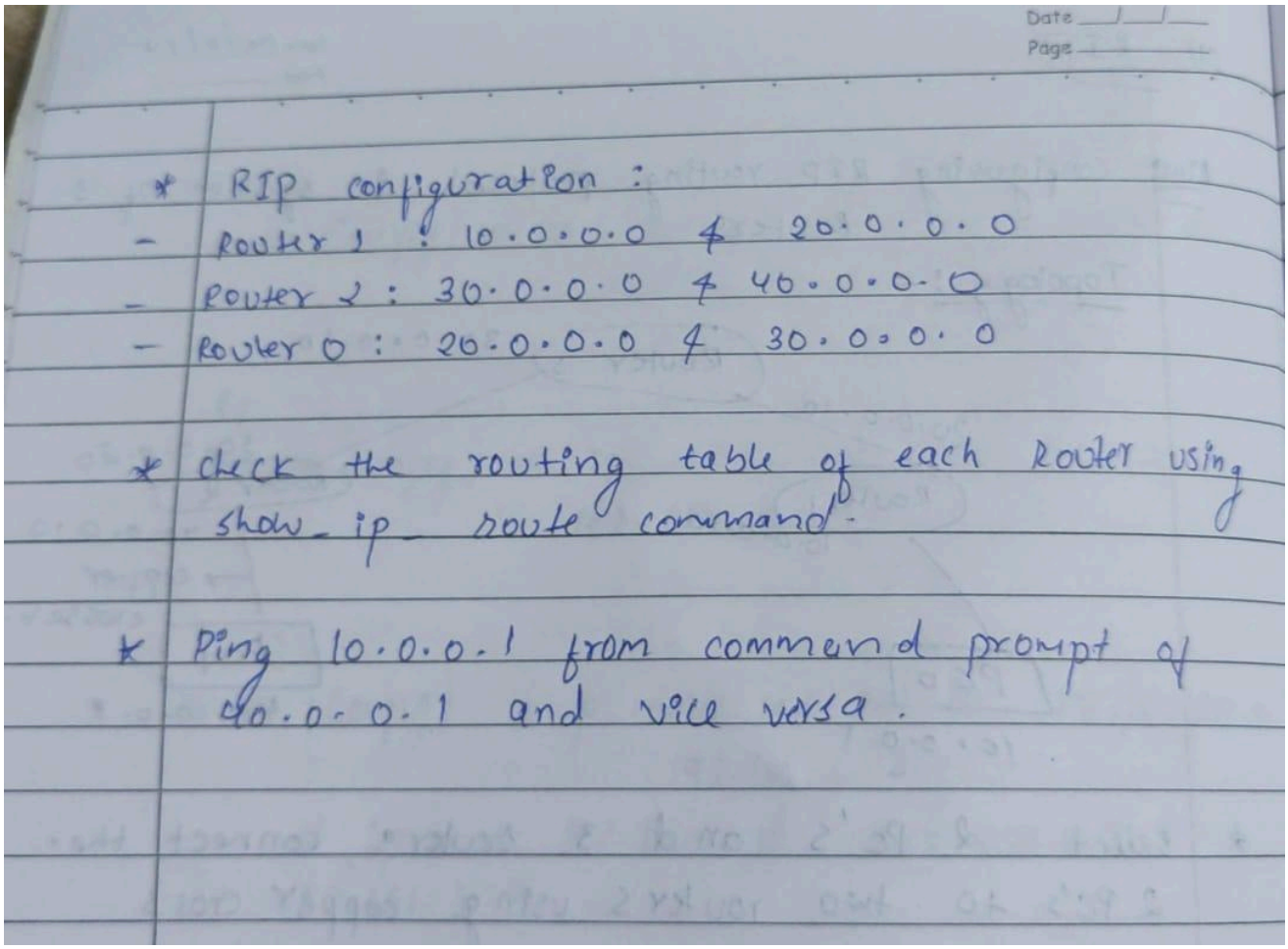
RIP

Aim: configuring RIP routing protocol in system of 3 Routers

Topology :-



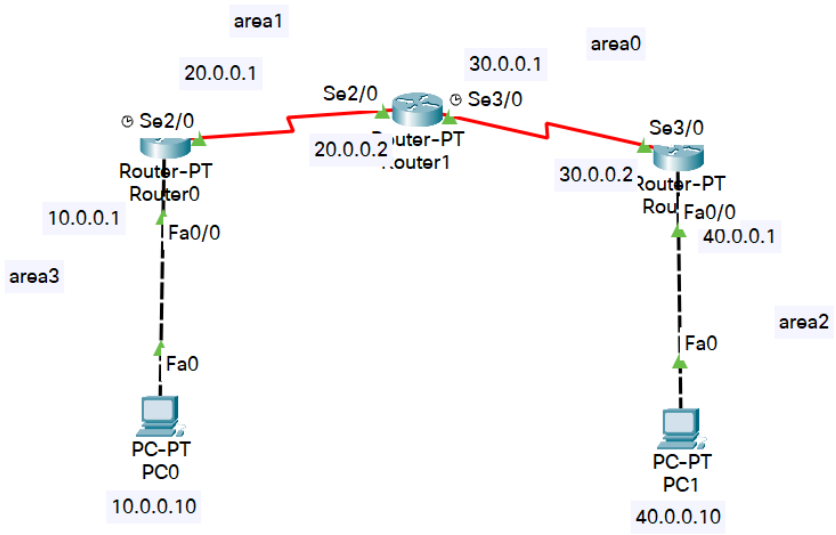
* Select 2 Pc's and 3 routers, connect the 2 Pc's to two routers using copper cross wires and connect the routers to another router to another router with serial DCE with timer.
* set the ip address of both the pc's as 10.0.0.1 and 40.0.0.1 respectively and their gateways as 10.0.0.10 and 40.0.0.10 respectively
* For router 1, Fast ethernet 0/0 : 10.0.0.10 255.0.0.0
  - serial 2/0 : 20.0.0.1 255.0.0.0
  - clock rate 64000
* For router 2, fast ethernet 0/0 : 40.0.0.1 255.0.0.0
  - serial 3/0 : 30.0.0.2 255.0.0.0
* For router 0, serial 2/0 : 20.0.0.2 255.0.0.0
  - serial 3/0 : 30.0.0.1 255.0.0.0
  - clock rate 64000



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
```

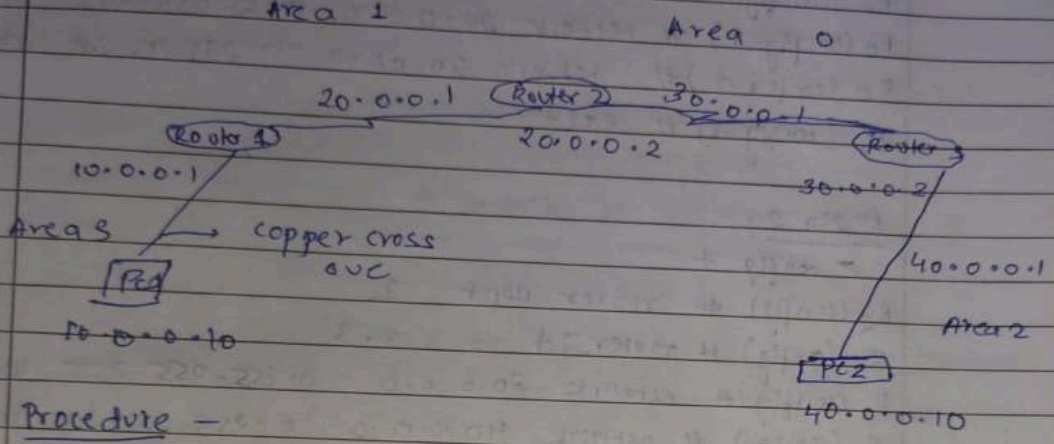
iv. Observation

* OSPF:

Date 13/12/24
Page

* Aim - configuring ospf protocol for a system of 3 routers.

Topology:



Procedure -

1] Select the two Pc's & three routers & join the 2 pc's to the two routers with copper-cross over wires.

2] Join the 2 routers to the third router with clocked copper wire

3] configure the pc's and gateways with IP's

4] configure the routers as per the topology above with the ip addresses.

5] Encapsulation ppp and clock rate need to be set as done in rip protocal experiment.

6] configuring each router with ospf protocol

For router 1 :-

> enable

> config t

R1(config)# router ospf 1

R1(config-r)# router-id 1.1.1.1

R1(config-r)# network 10.0.0.0 0.255.255.255 area 31

R1(config-r)# network 20.0.0.0 0.255.255.255 area 15

R1(config-r)# exit

Router 0

> config +

R0 (config) # router ospf 1

R0 (config-r) # router-id 2.2.2.2

R0 (config-r) # network 20.0.0.0 0.255.255.255 area

R0 (config-r) # network 30.0.0.0 0.255.255.255 area

R0 (config-r) # exit

Router 2.

> config +

R2 (config) # router ospf 1

R2 (config) # router_id 3.3.3.3

R2 (config) # network 30.0.0.0 0.255.255.255 area

R2 (config) # network 40.0.0.0 0.255.255.255 area

R2 (config-r) # exit

2) configuring the interfaces.

R1 (config-if) # interface loop back 0

R1 (config-if) # ip add 172.16.1.252 255.255.0.0

R1 (config-if) # no shutdown.

R1 (config-if) # interface loop back 0

R2 (config-if) # ip address 172.16.1.253 255.255.0.0

R2 (config-if) # no shutdown.

R3# show ip route

C 40.0.0.0/8 is directly connected

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

30.0.0.1/32 is directly connected, serial 1/0

C 40.0.0.1/32 is directly connected.

8] In Router R_1

R_1 (config) # router ospf 2

R_1 (config-router) # area 1 virtual-link 2.2.2.2

In Router R_2

R_2 (config) # router ospf 1

R_2 (config-router) # area 1 virtual-link 1.1.1.1

R_2 (config-router) # exit.

Now a virtual link is established between area 3 & area 0.

9] show ip route must be configured in all routers

For router 2 :

O IA 10.0.0.0/8 via 20.0.0.1 00:00:01, serial 2/0

20.0.0.0/8 is vertically subnetted, 2 subnetted 2 Masks.

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 Masks

C 20.0.0.0/8 is directly connected serial 2/0

C 20.0.0.1/32 is directly connected, serial 2/0

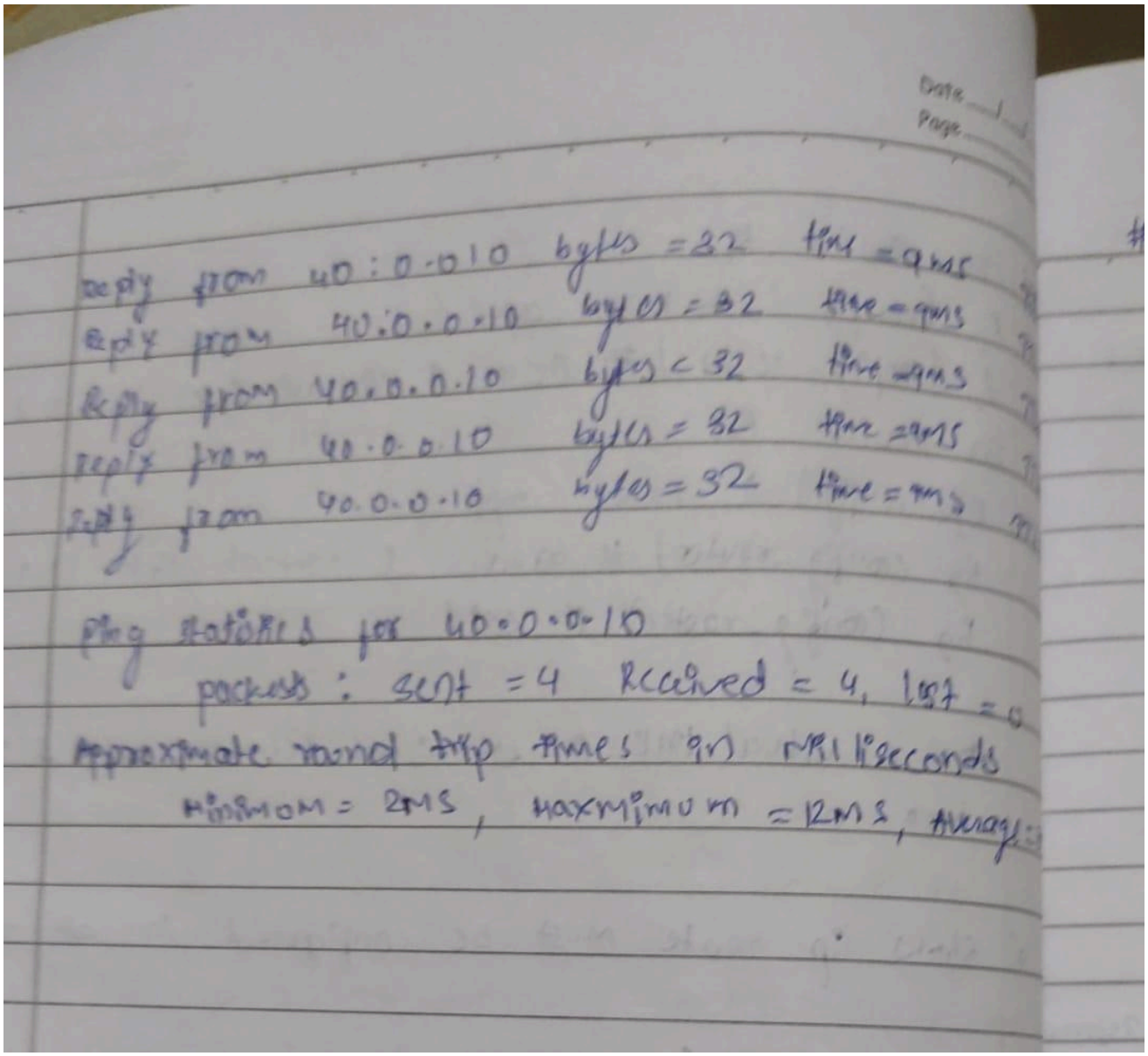
30.0.0.0/8 is variably subnetted, 2 subnets

C 172.16.0.0/16 is directly connected Loopback.

Result :

> ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data



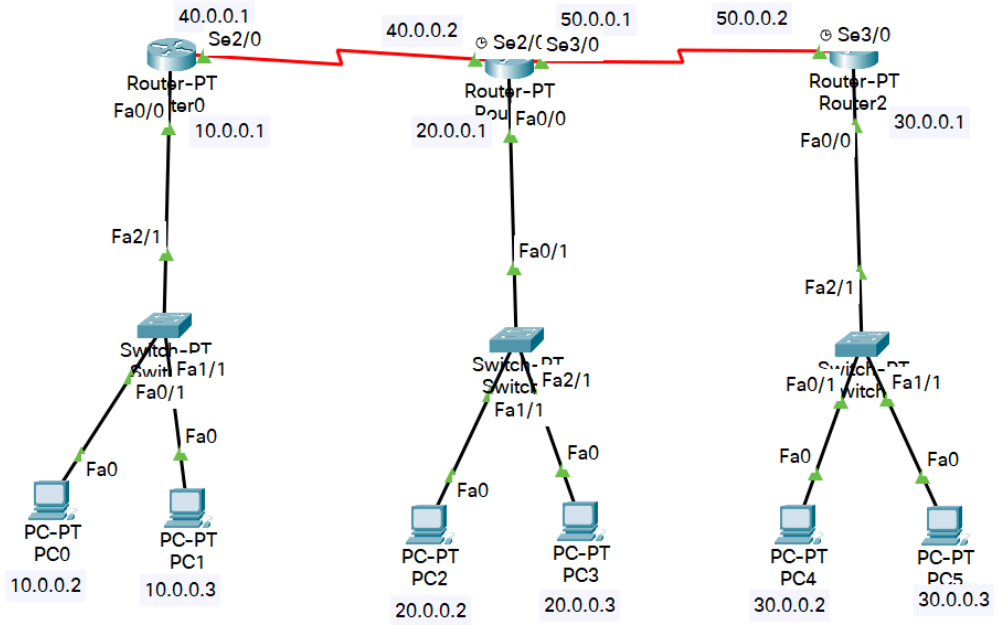
Reply from 40:0.0.10 bytes =32 time =9ms
Reply from 40.0.0.10 bytes =32 time =9ms
Reply from 40.0.0.10 bytes < 32 time =9ms
reply from 40.0.0.10 bytes = 32 time =9ms
reply from 40.0.0.10 bytes = 32 time = 9ms

ping statistics for 40.0.0.10
packets: sent = 4 Received = 4, lost = 0
Approximate round trip times in milliseconds
Minimum = 2ms, Maximum = 12ms, Average:

Program 7

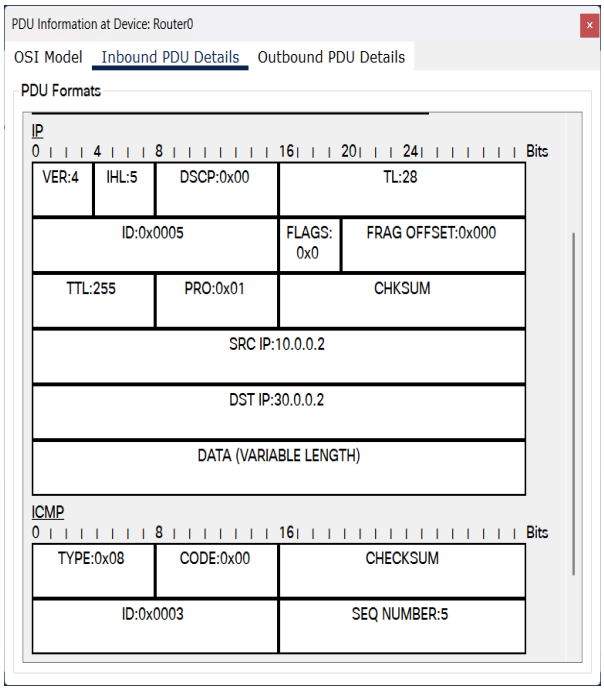
i. Demonstrate the TTL/ Life of a Packet

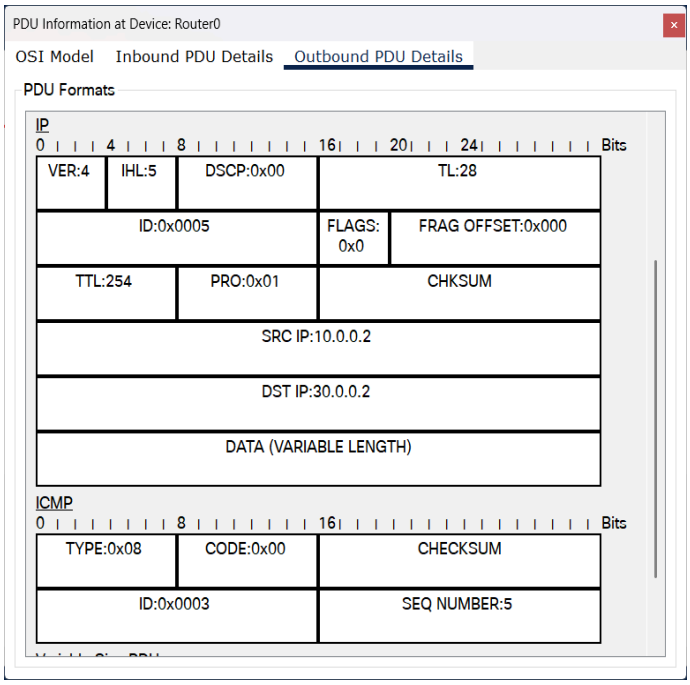
ii. Procedure along with the topology



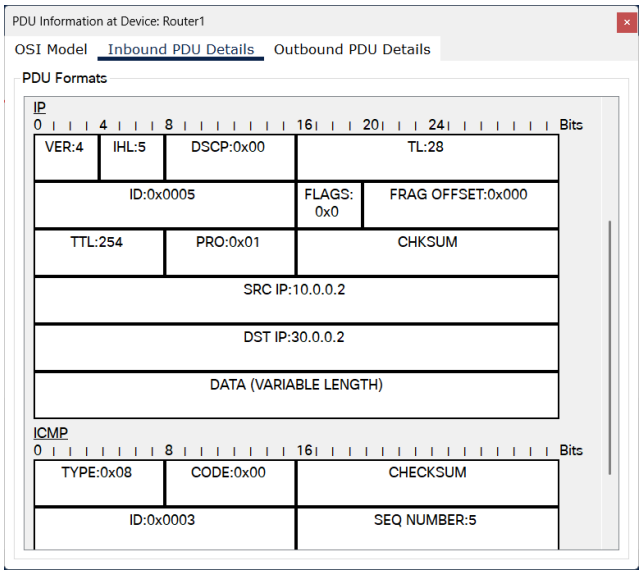
iii. Screen shots/ output

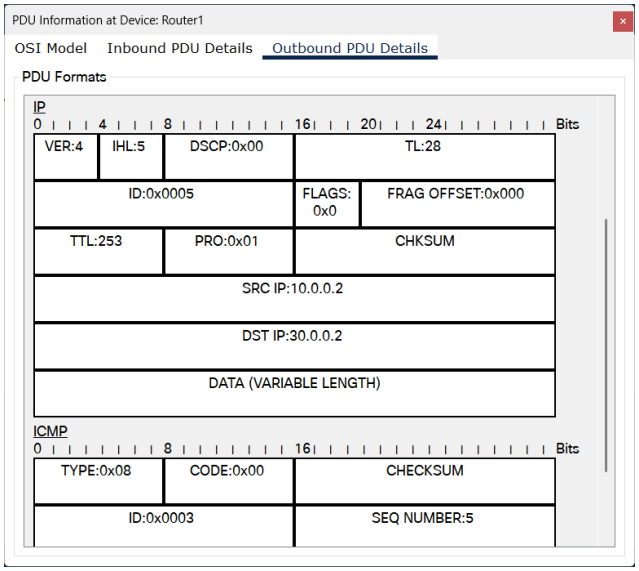
Packet at Router0



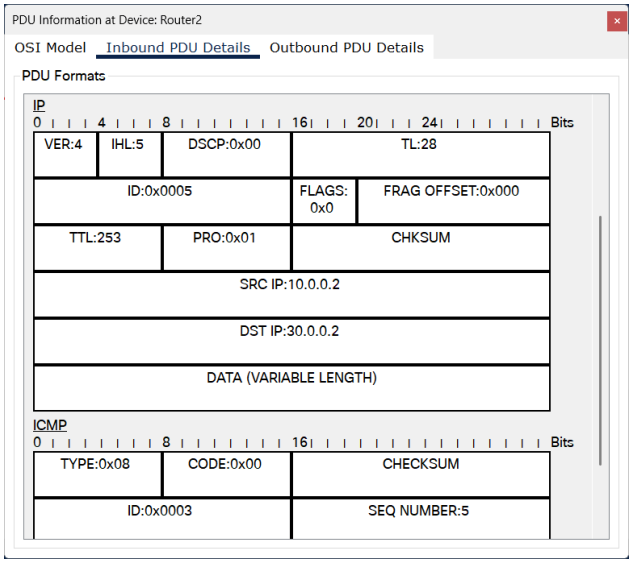


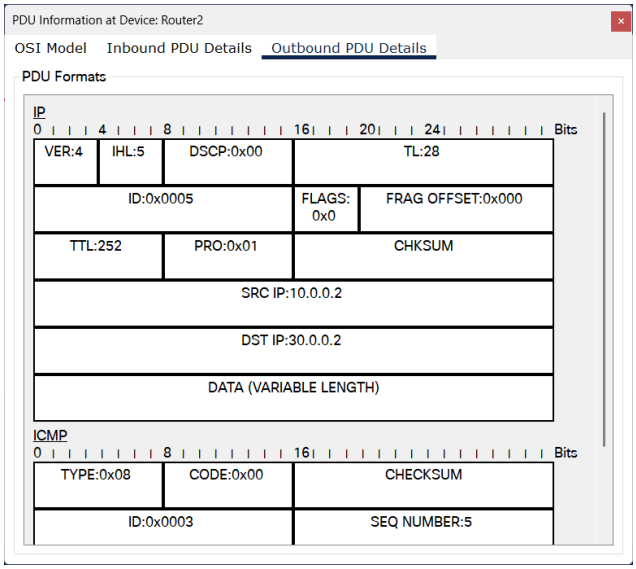
Packet at Router1





Packet at Router2





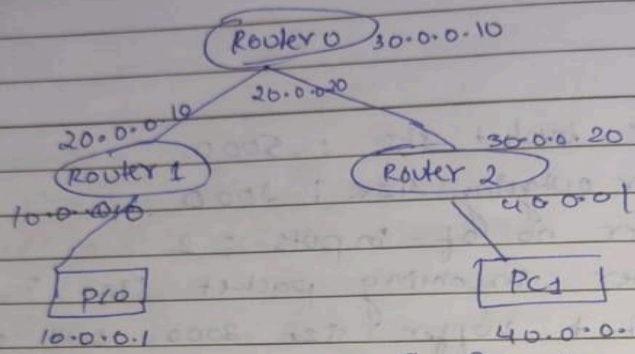
iv. Observation

Date / /
Page

LAB-07 :

* Aim - To demonstrate the TTL / life of a packet packet

Topology :



Procedure : 1] create a 2PC & 3 router configuration as shown in the topology

2] use Serial DTE between Routers & copper cross overs b/w router & pc. 3] configure the Ip address & gateway of pc and configure all the routers.

For routers

```
> enable
# config t
# interface fastEthernet 0/0
# ip address 10.0.0.10 255.0.0.0
# no shut
# exit
# ip router 30.0.0.0 255.0.0.0 20.0.0.200
# ip router 40.0.0.0 255.0.0.0 20.0.0.20
# exit
```

For router 1 :—

```
> enable
# config t
# interface serial 2/0
# ip address 20.0.0.20 255.0.0.0
# no shut
# exit
# interface serial 3/0
```

Date __/__/__
Page

ip address 30.0.0.10 255.0.0.0
no shut
exit
ip route 10.0.0.0 255.0.0.0 20.0.0.10
ip route 40.0.0.0 255.0.0.0 20.0.0.20
exit

* For router 2

> enable
config t
interface fastethernet 0/0
ip address 30.0.0.0 20.255.0.0.0
no shut
exit
interface fastethernet 0/0
ip address 10.0.0.10 255.0.0.0
no shut
exit
ip route 10.0.0.255.0.0 30.0.0.10
ip route 20.0.0.0 255.0.0.0 30.0.0.10

4] select simulation mode, select simple PDU and select source of destination PC's

5] click on capture button to send PDU and acknowledgment from pc to router & router to PC

6] click on PDU during every transfer to see the inbound and outbound PDU details observe the difference in the TTL

* Result

outbound PDU details : TTL = 255-

PDU information at Router 0 / Inbound PDU details TTL = 255
outbound = 254

PDU information at Router 1 / Inbound PDU details TTL = 254
outbound 253

PDU information at Router 2 / Inbound PDU details TTL = 253
outboond = 252

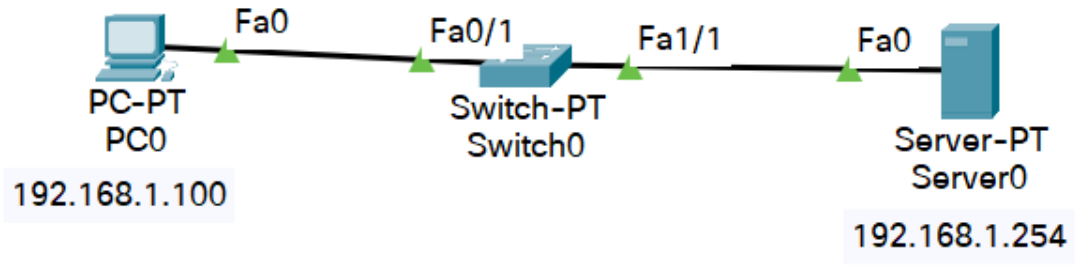
* Observation :

The TTL is reduced by 1 in every router TTL is a mechanism which limits the number of hops b/w source and destination.

Program 8

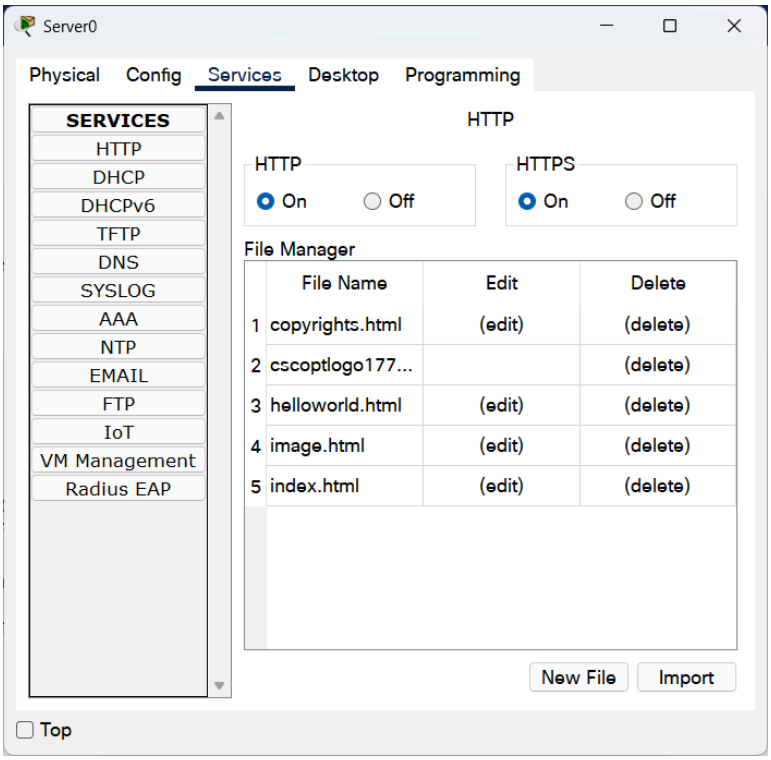
i. Configure Web Server, DNS within a LAN.

ii. Procedure along with the topology

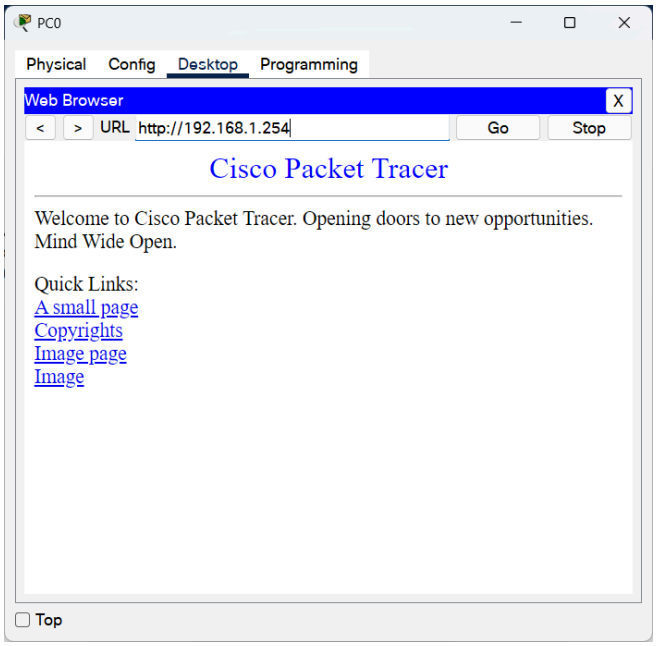


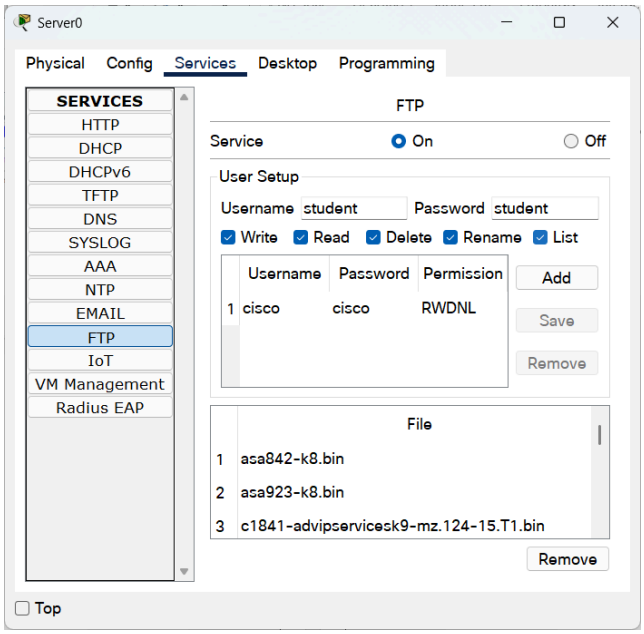
iii. Screen shots/ output

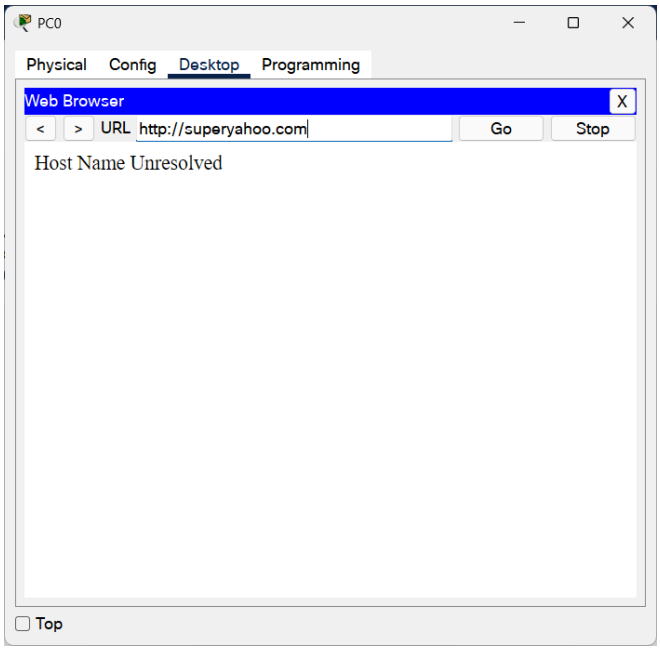
Server's services

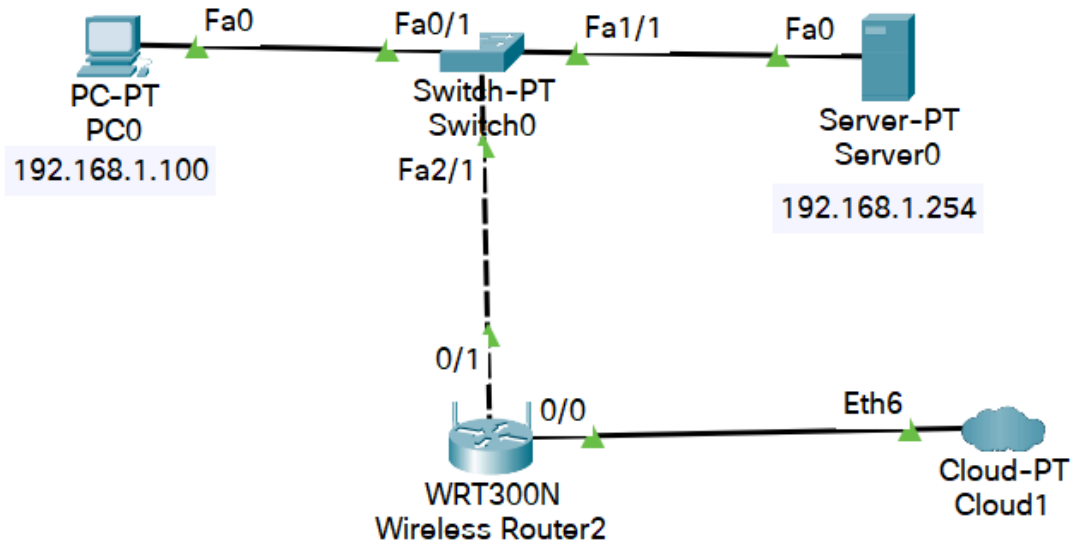


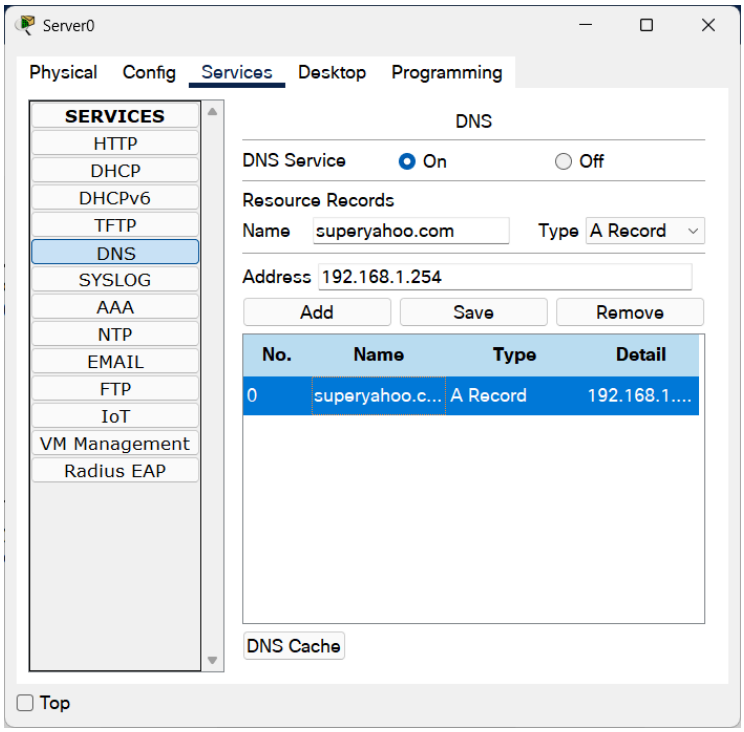
PC's Web Browser


PC0
Physical
Config
Desktop
Programming
Web Browser
X
<
>
URL
http://192.168.1.254
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image
Top

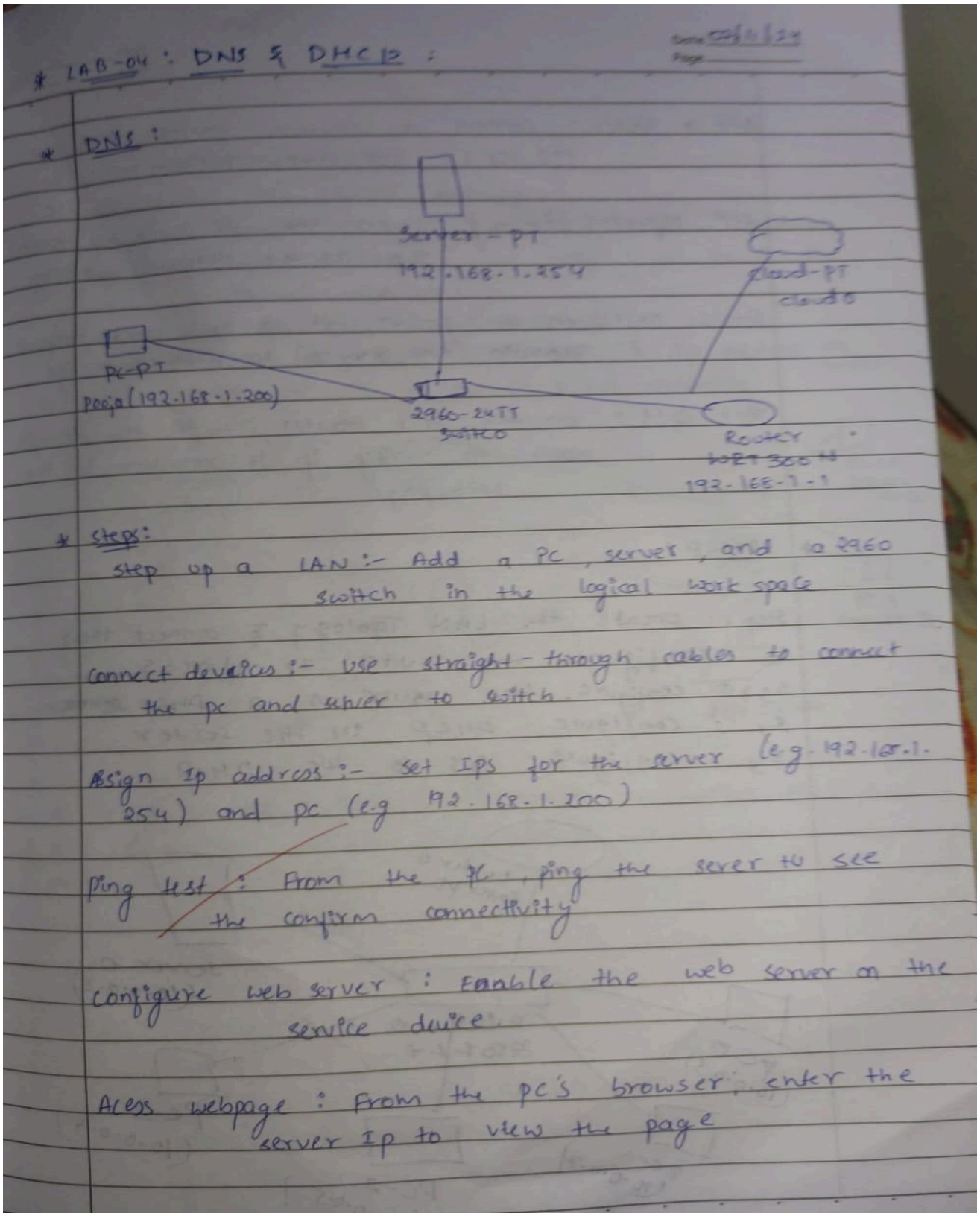

Server0
Physical
Config
Services
Desktop
Programming
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
FTP
Service
On
Off
User Setup
Username
student
Password
student
Write
Read
Delete
Rename
List
Username
Password
Permission
1
cisco
cisco
RWDNL
Add
Save
Remove
File
1
asa842-k8.bin
2
asa923-k8.bin
3
c1841-advipservicesk9-mz.124-15.T1.bin
Remove
Top


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



Page

Add a Router: Connect a wireless router with Ip 192.168.1.1 for more realism.

Set default Gateway: on the PC, set the router's Ip as the deafault gateway

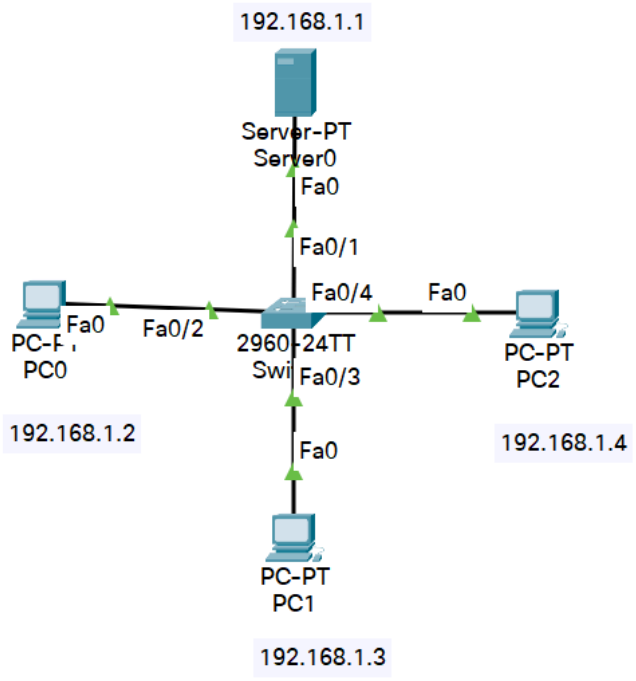
Enable DNS: on the server, st up DNS and map a domain (e.g xyyz) to the server Ip

Test DNS: In the PC's browser, type the domain name to verify if it resolves to thy web page.

Program 9

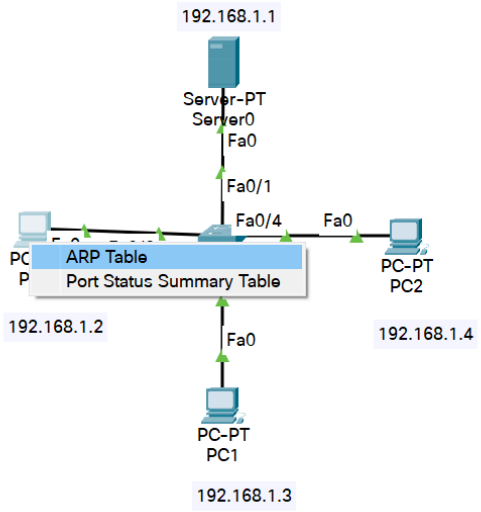
i. To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

ii. Procedure along with the topology

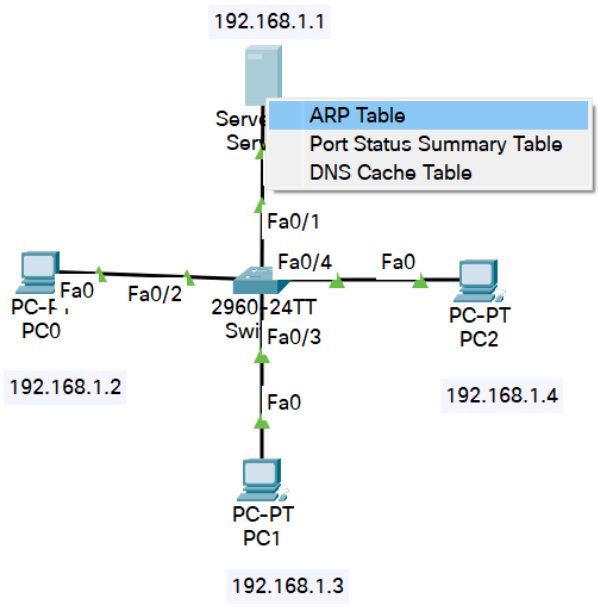


iii. Screen shots/ output

ARP Table of PC



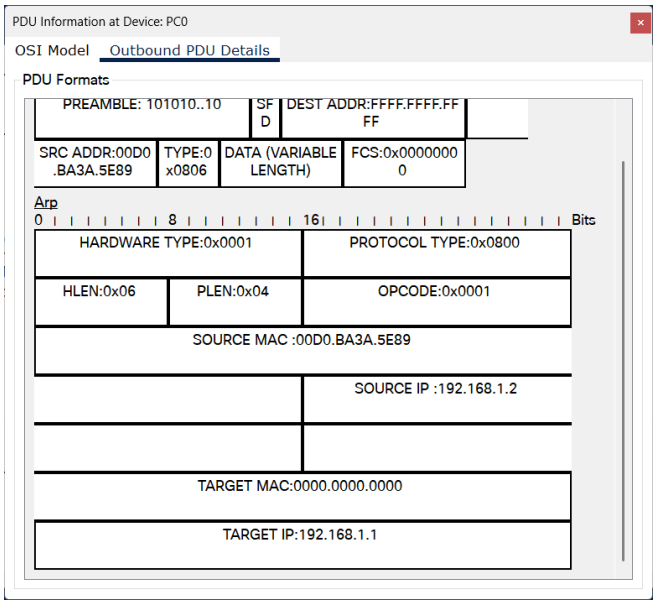
ARP Table of Server

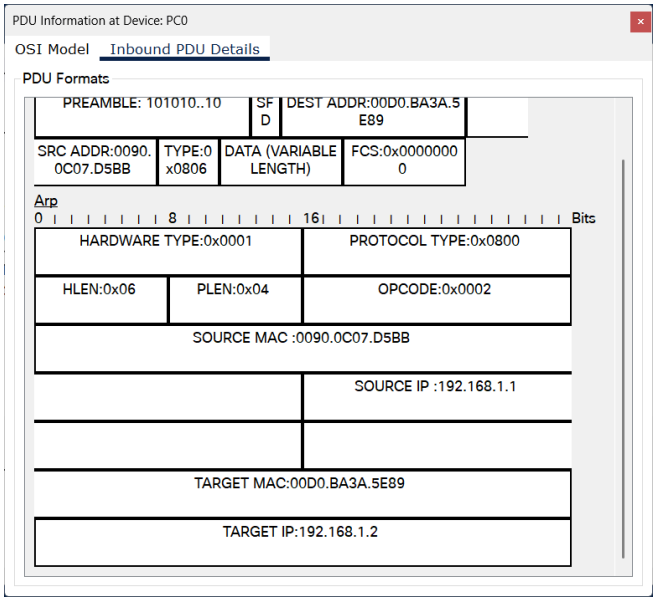


Command at PC

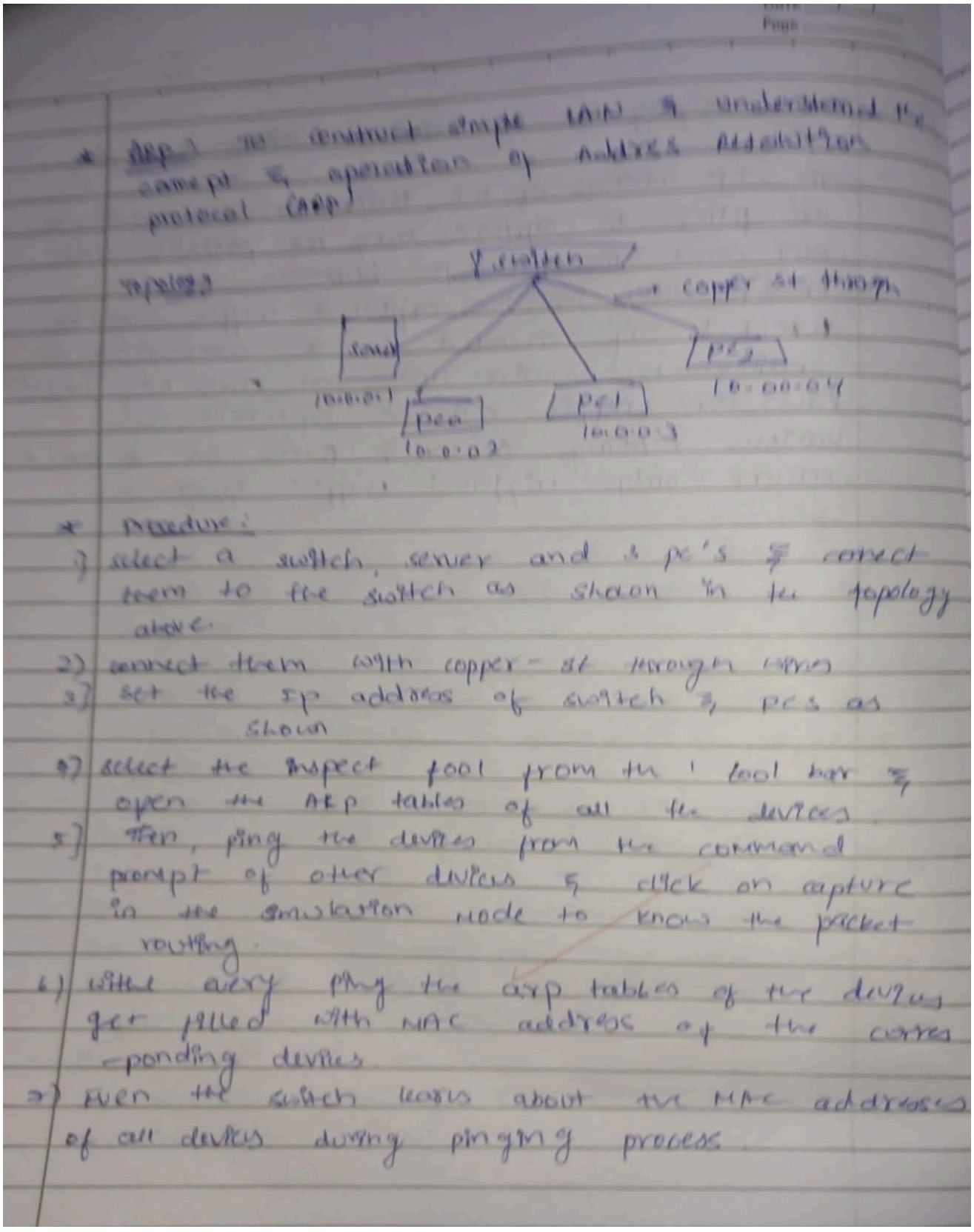
```
Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
C:\>
```

Pinging in Simulation Mode


PDU Information at Device: PC0
OSI Model
Outbound PDU Details
PDU Formats
PREAMBLE: 101010..10
SF D
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
SF D
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



Date __/__/__
Page

8) once you have pinged all the device, you can check the arp table of each device in command prompt of pcs
   >arp - a

9) In the switch → cli, you can check the MAC address of the devices as follows
   switch > show MAC address-table
   MAC address table.

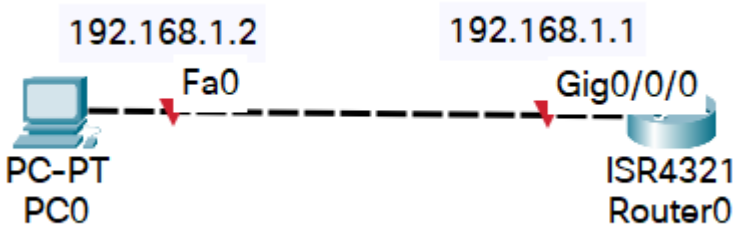
* Observation:
ARP protocal is communication protocol used for discovering the link layer address, such as a MAC address. After pinging, every device learns about the MAC address of the pinged devices and the switch stors these MAC addresses in the ARP table for future pinging ARP. Learns about the MAC address by pinging all the devices and the right Ip address responds with the acknowledgement.

26/2/24

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

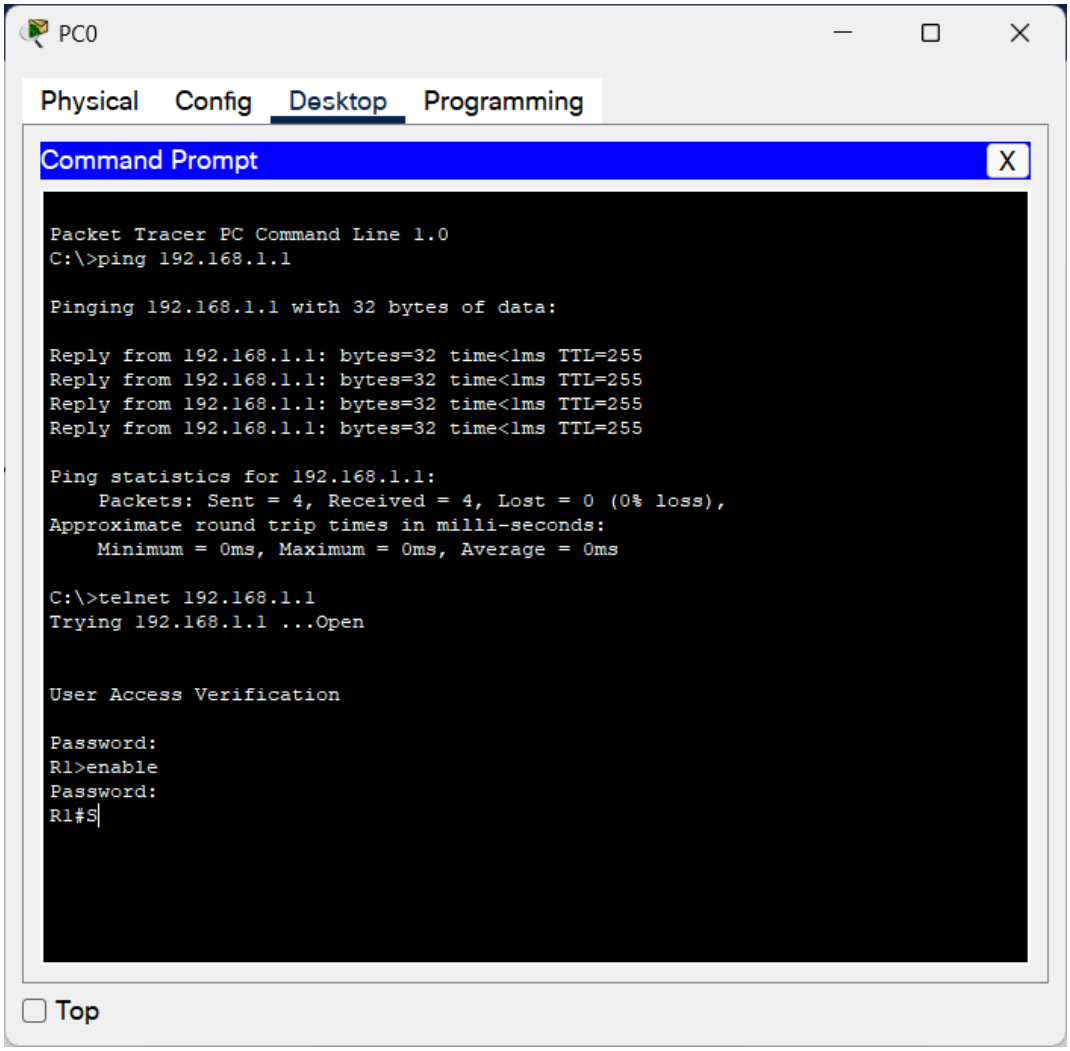


iii. Screen shots/ output

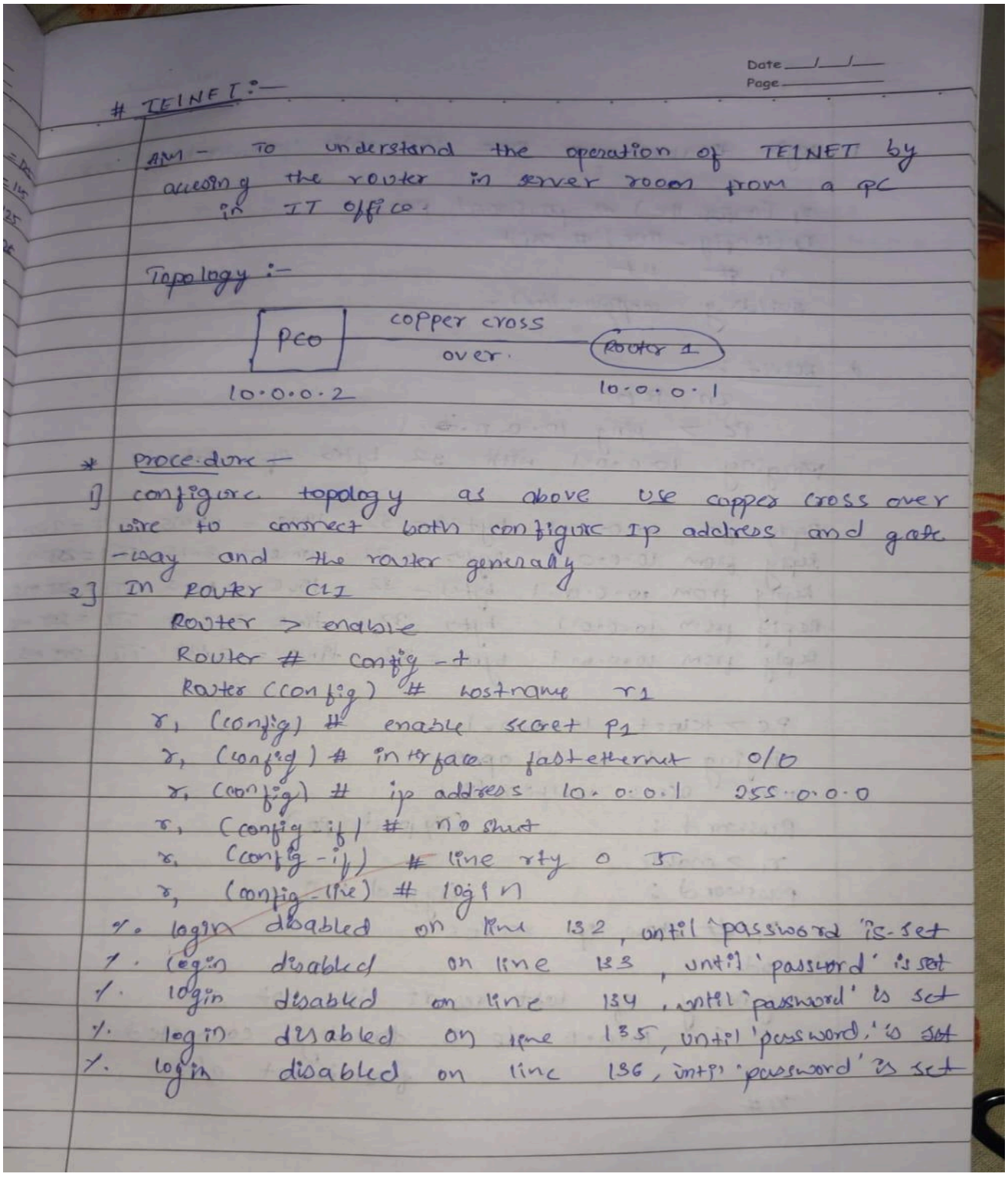
Router

```
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



Date __/__/__
Page

TELNET :-

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

Topology :-

* Procedure -

1] configure topology as above use copper cross over wire to connect both configure Ip address and gate-way and the router generally

2] In Router CLI

Router > enable
Router # config -t
Router (config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # interface fastethernet 0/0
r1 (config) # ip address 10.0.0.1 255.0.0.0
r1 (config-if) # no shut
r1 (config-if) # line vty 0 5
r1 (config-line) # login
% login disabled on line 132, until 'password' is set
% login disabled on line 133, until 'password' is set
% login disabled on line 134, until 'password' is set
% login disabled on line 135, until 'password' is set
% login disabled on line 136, until 'password' is set

Date / /
Page

1- login disabled on line 157 until 'password' is set

r1 (config-line) # password p0
r1 (config-line) # exit
r1 # wr
Building configuration.

* Result :—

In PC0

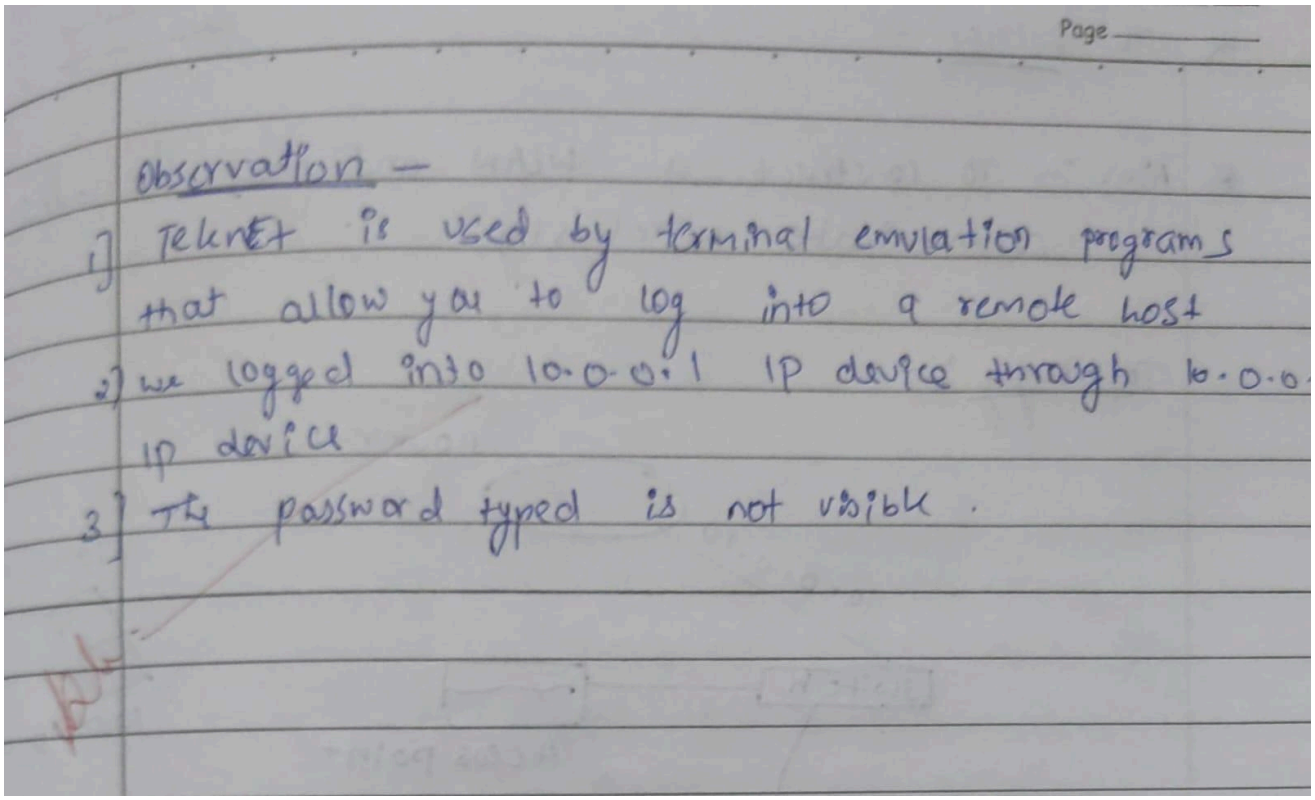
PC > ping 10.0.0.1

pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1 bytes = 32 time = 21MS TTL = 255MS
Reply from 10.0.0.1 bytes = 32 time = 6MS TTL = 255MS
Reply from 10.0.0.1 bytes = 32 time = 0MS TTL = 255 MS
Reply from 10.0.0.1 bytes = 32 time = 0 MS TTL = 255 MS
Reply from 10.0.0.1 bytes = 32 time = 0 MS TTL = 255 MS

PC > telnet 10.0.0.1
Trying 10.0.0.1 open
user acess verification
Password : (Typed p0)
r1 > enable
password : (Typed P1)
r1 # show ip route.
code :
gateway of last resort is not set
C 10.0.0.0/8 is directly connected.
fastethernet 0/0.

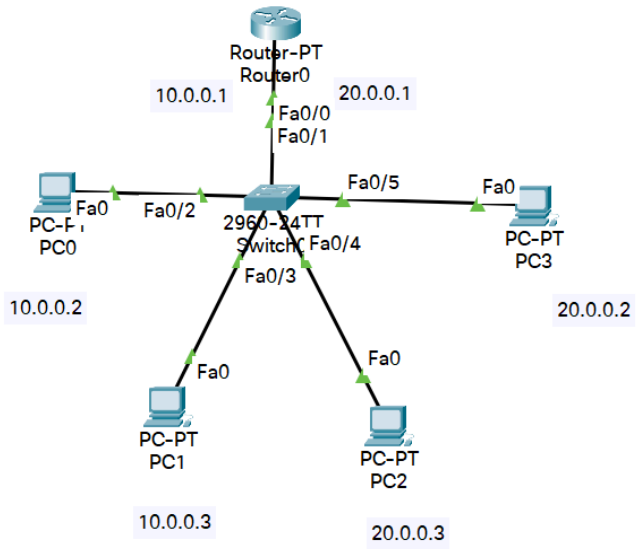
r1 #



Program 11

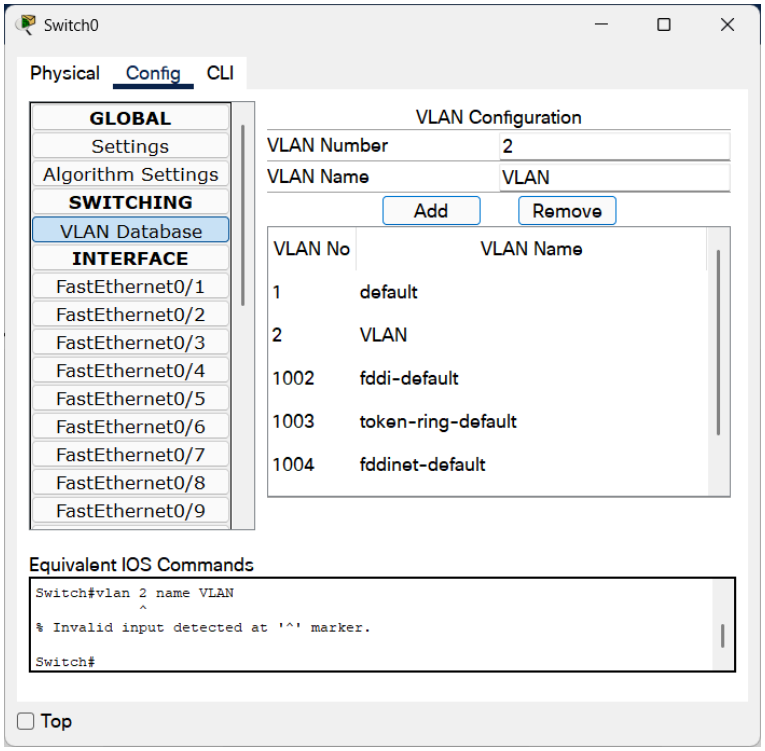
i. To construct a VLAN and make the PC's communicate among a VLAN

ii. Procedure along with the topology

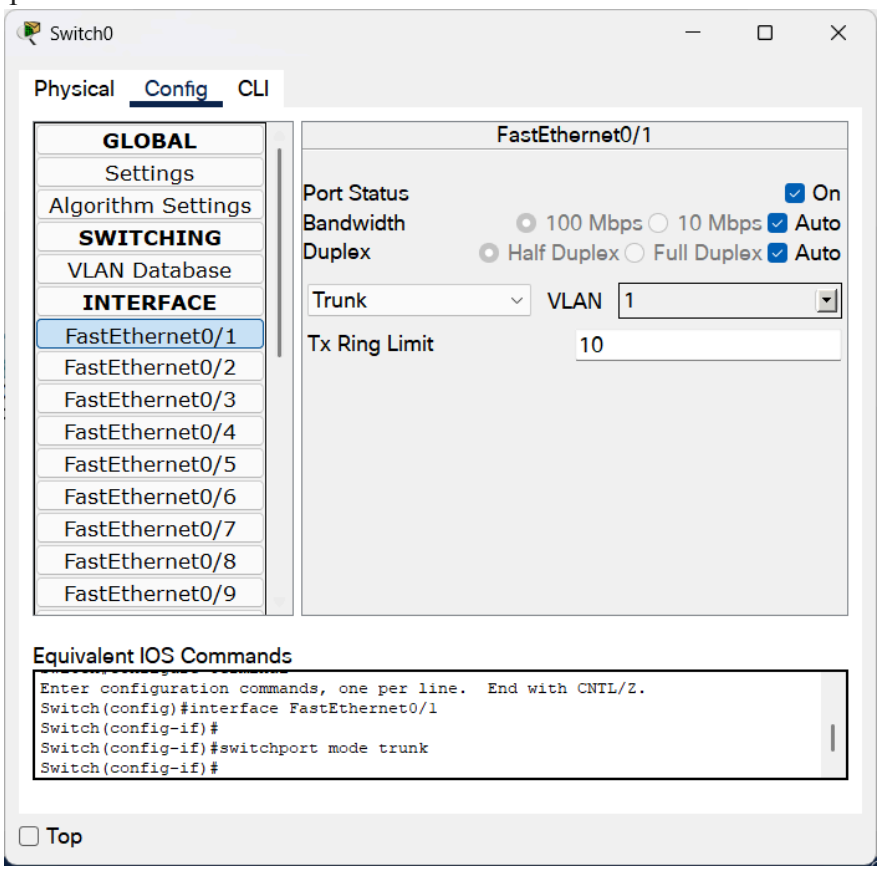


iii. Screen shots/ output

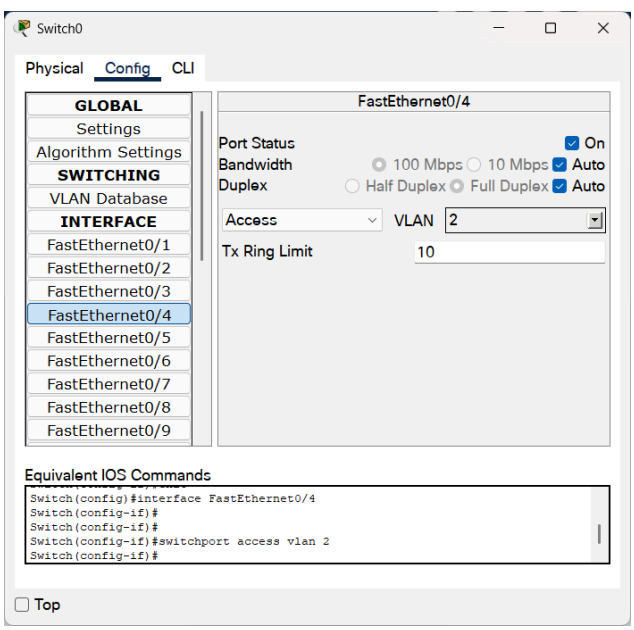
Switch Configuration

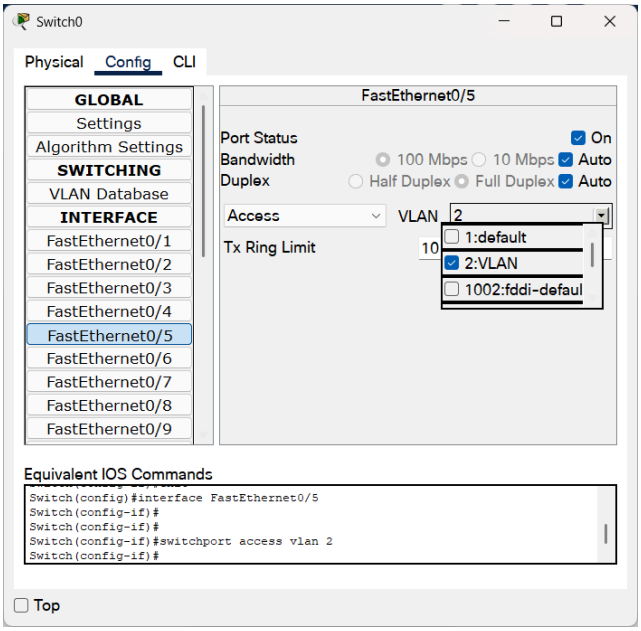


Configuration of port connected to router

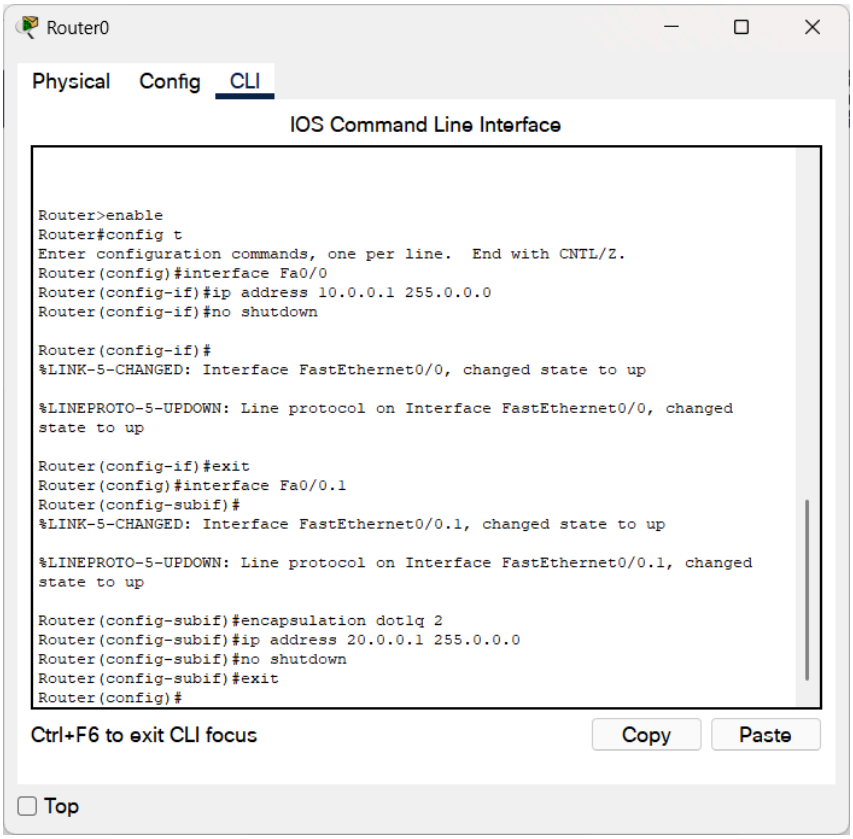


Configuration of ports connected to other networks

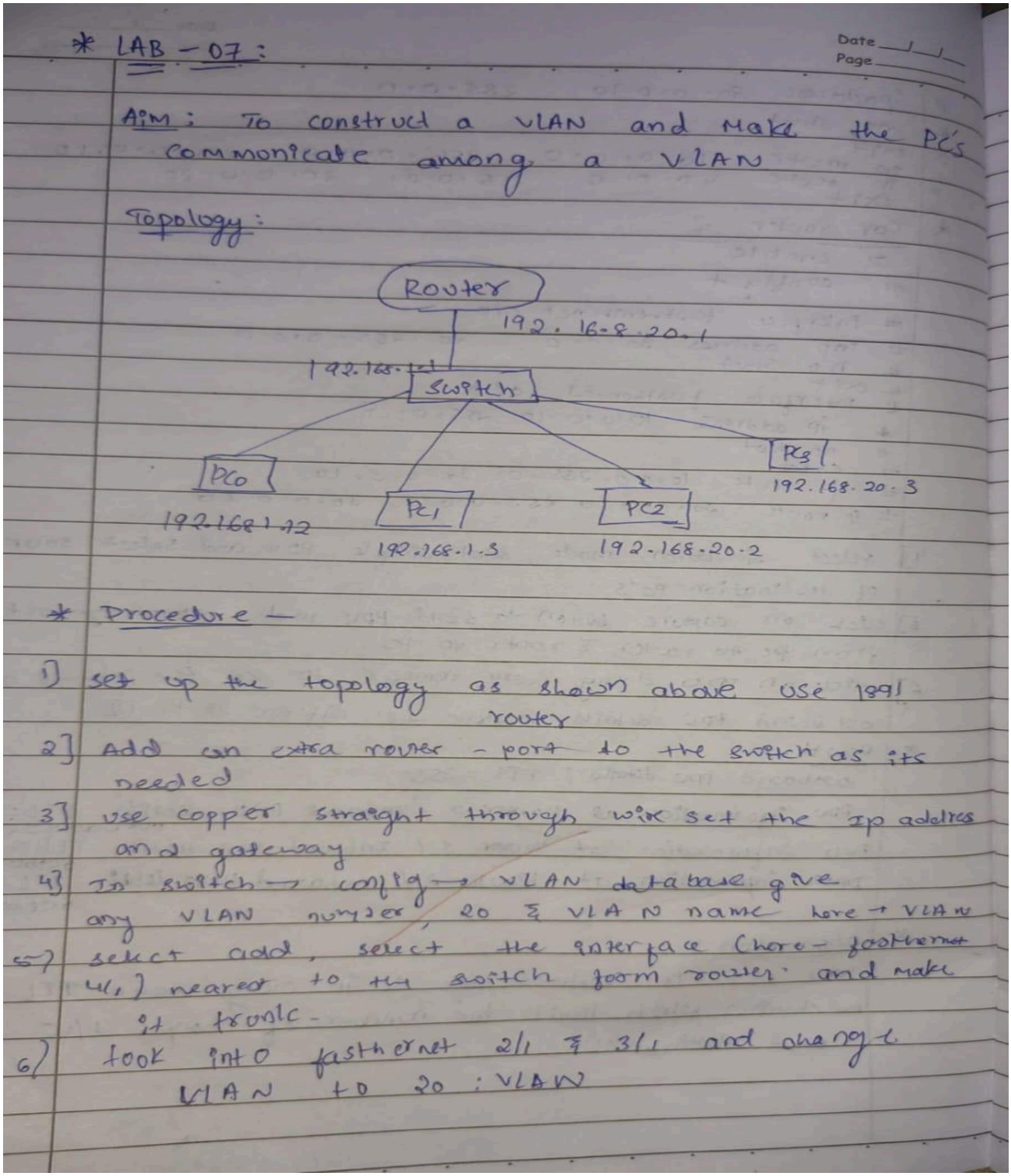




Configuartion of Router



iv. Observation



Date / /
Page

7. Router select VLAN database enter the number and name of the VLAN created.

In CLI of router (VLAN) # exit
Apply command
exiting ----
Router # config
Router (config) # interface fastethernet 0/0
Router (conf-if) # ip address 192.168.2.1
Router (config-if) # no shut 255.255.255.0
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 192.168.1.1
Router (config-if) # no shut 255.255.255.0
Router (config) # interface fastethernet 0/0.1
Router (config-subif) # encapsulation dot1q
Router (config-subif) # ip address 192.168.10.0
255.255.255.0
Router (config-subif) # no shut
Router (config-subif) # exit.

Result:—

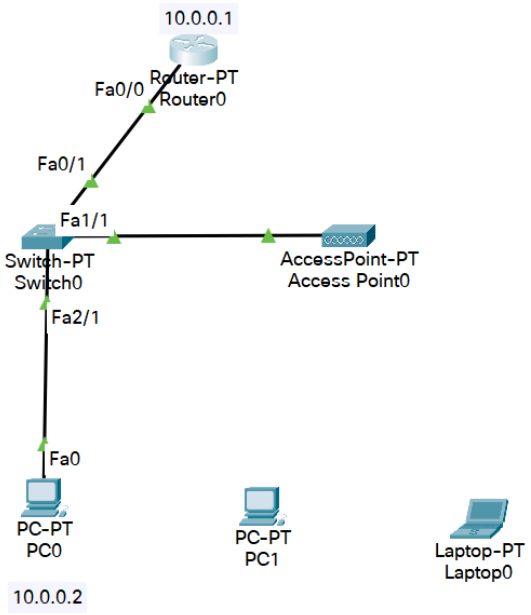
(in pc0)
PC > ping 192.168.20.3

pinging 192.168.20.3 with 32 bytes of data
Reply from 192.168.20.3 : bytes = 32 time = 1ms TTL
Reply from 192.168.20.3 : bytes = 32 time = 1ms TTL
Reply from 192.168.20.3 : bytes = 32 time = 0ms TTL
Reply from 192.168.20.3 : bytes = 32 time = 0ms TTL

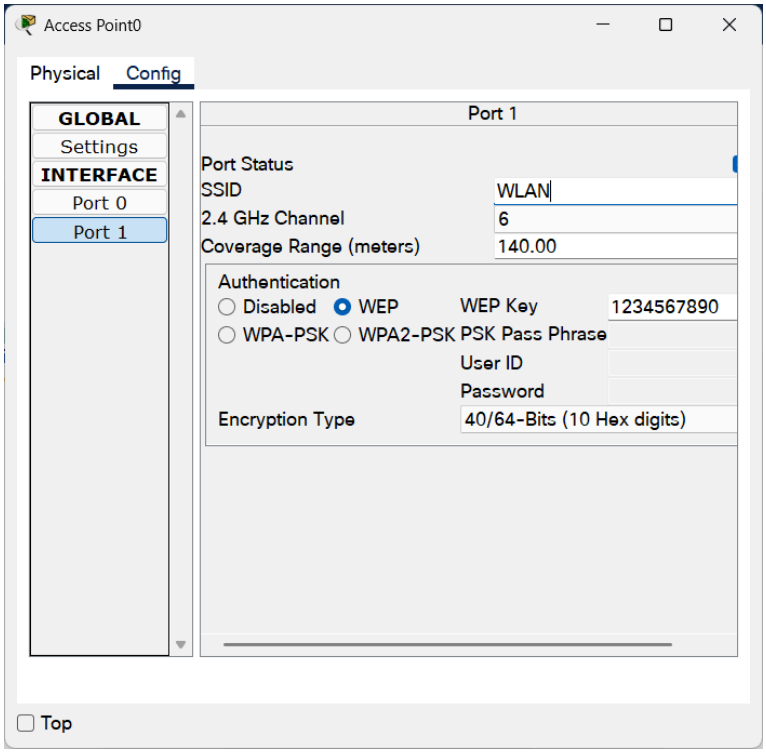
ping statistics for 192.168.20.3
Packets Sent = 4 Received = 4 lost = 0
Approximate round trip time in milliseconds
Minimum = 0 ms, Maximum = 1ms, Average = 0 ms

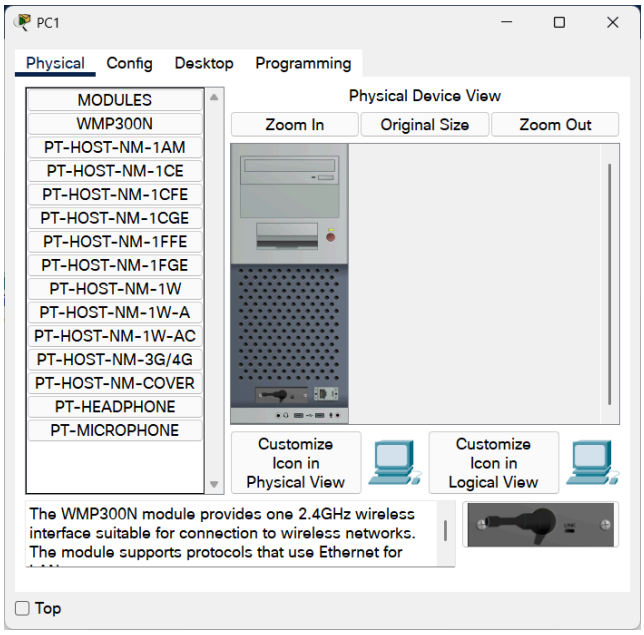
Program 12

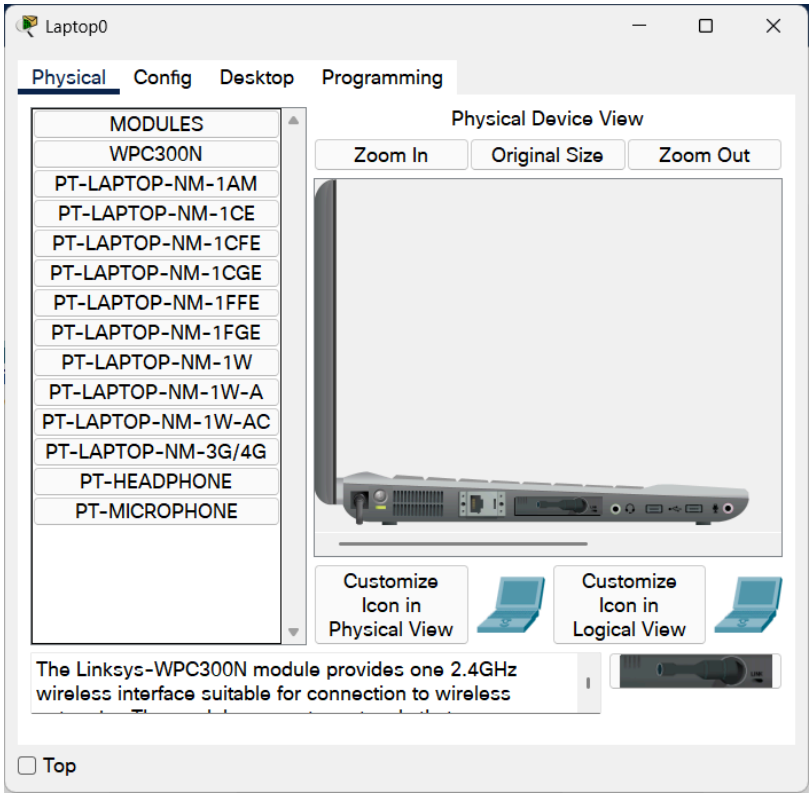
i. To construct a WLAN and make the nodes communicate wirelessly.
ii. Procedure along with the topology

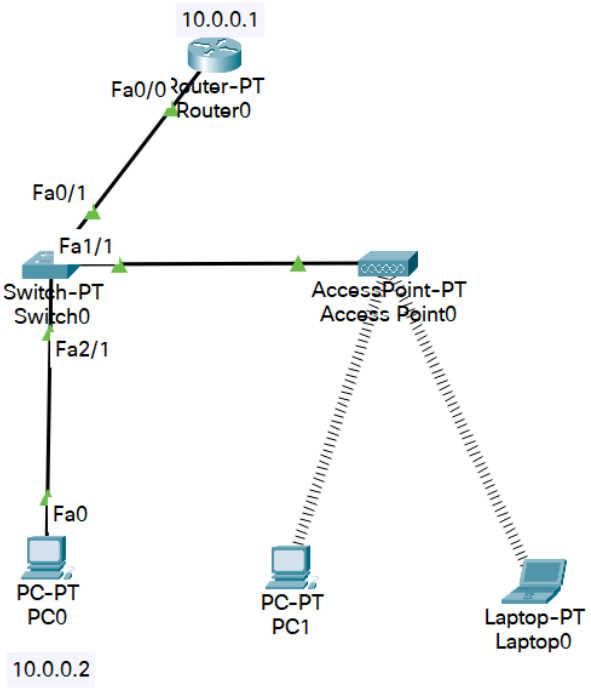


iii. Screen shots/ output

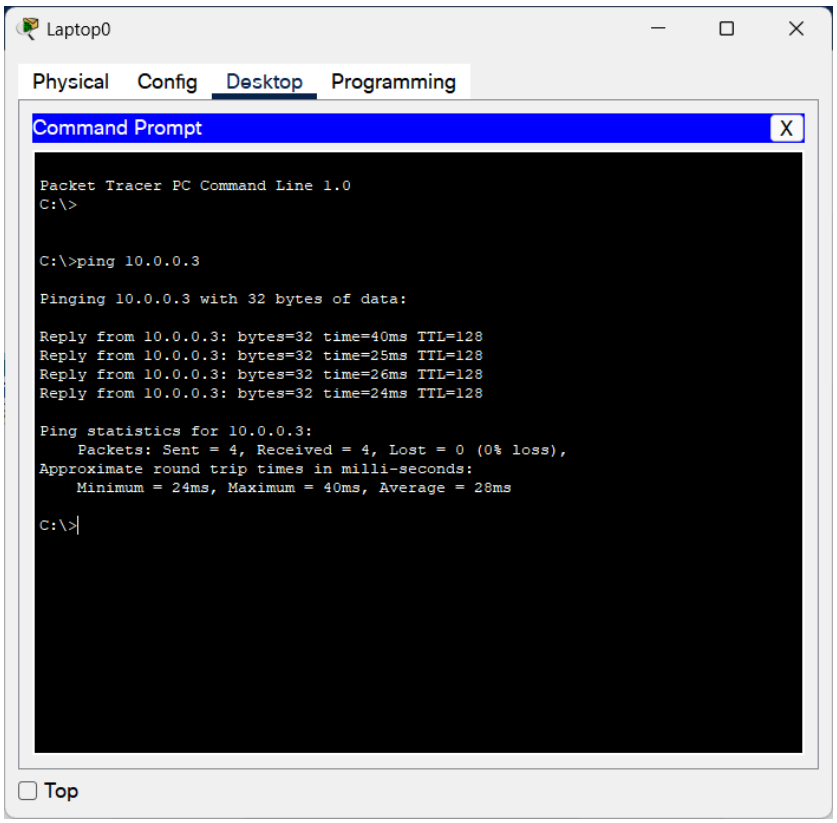



PC1
Physical
Config
Desktop
Programming
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-1W-AC
PT-HOST-NM-3G/4G
PT-HOST-NM-COVER
PT-HEADPHONE
PT-MICROPHONE
Physical Device View
Zoom In
Original Size
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

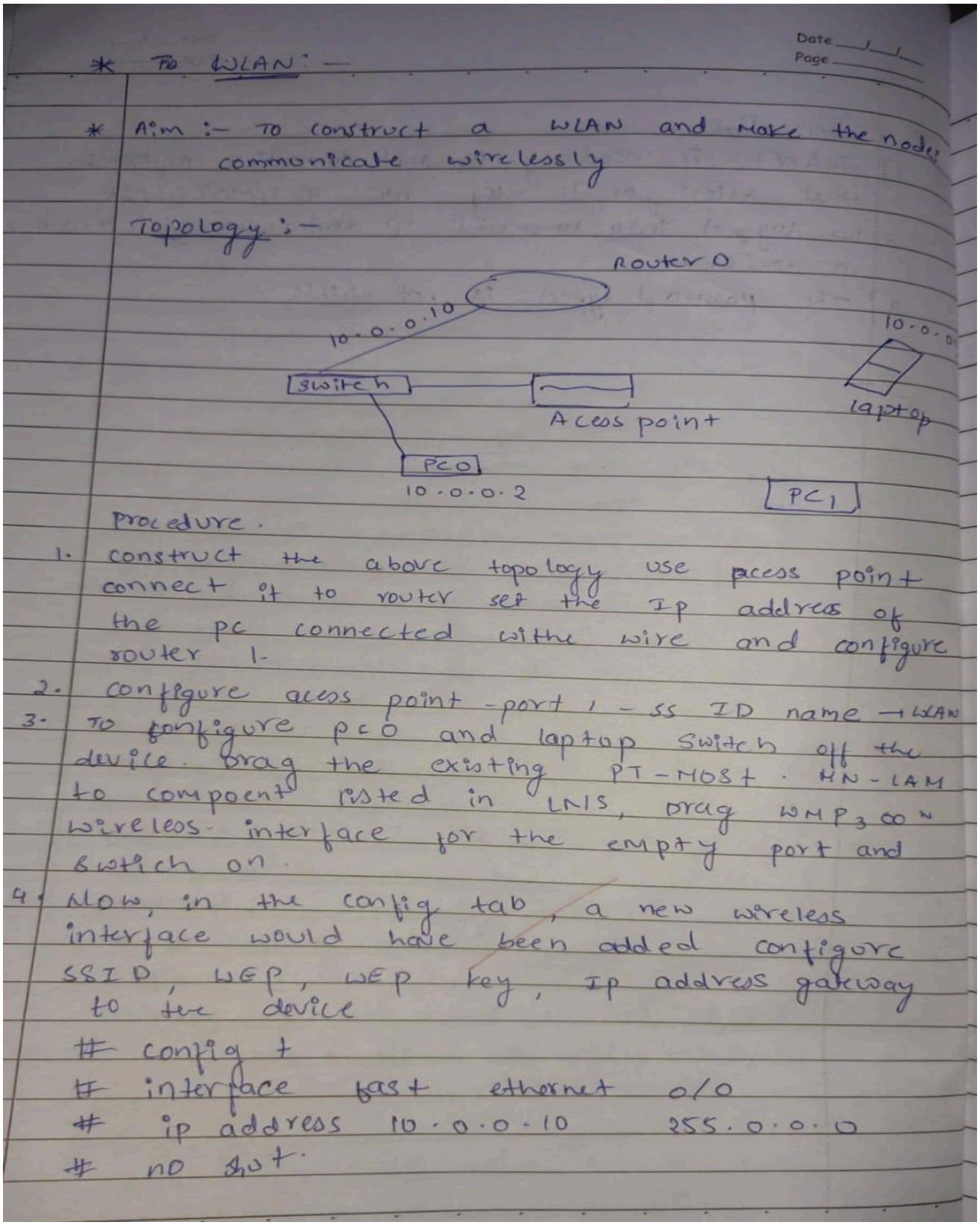

Laptop0
Physical
Config
Desktop
Programming
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-1W-AC
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
Physical Device View
Zoom In
Original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



Date __/__/__
Page

* To WLAN :-

* Aim :- To construct a WLAN and make the nodes communicate wirelessly

Topology :-

Procedure.

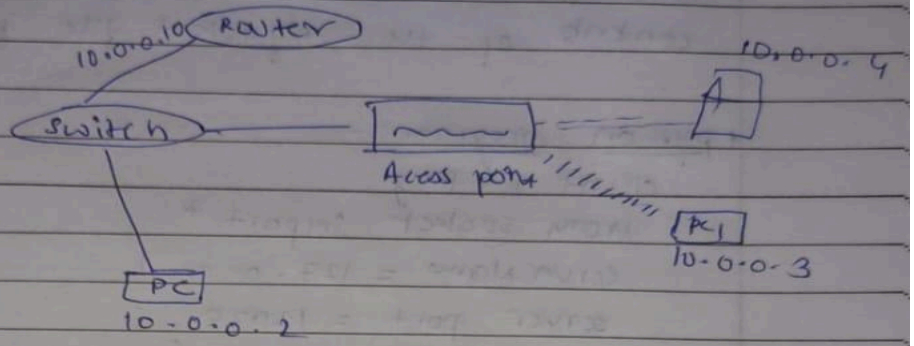
1. construct the above topology use access point connect it to router set the Ip address of the pc connected with wire and configure router 1-
2. configure acess point - port 1 - ss ID name → WLAN
3. To configure pc0 and laptop switch off the device. drag the existing PT-HOST. HN-LAM to compoent listed in LNIS, drag WMP300N wireless interface for the empty port and switch on.
4. Now, in the config tab, a new wireless interface would have been added configure SSID, WEP, WEP key, Ip address gateway to the device

```
# config t
# interface fast ethernet 0/0
# ip address 10.0.0.10 255.0.0.0
# no shut.
```

Date __/__/__
Page _____

Result :—

Topology :—



Result in PC0

Pc > Ping 10.0.0.3

Pinging 10.0.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 bytes 32 time 12ms TTL: 12

Reply from 10.0.0.3 bytes 32 time = 13ms TTL: 12

Reply from 10.0.0.3 bytes 32 time = 6ms TTL: 12

Reply from 10.0.0.3 bytes 32 time = 0ms TTL: 12

Ping statistics for 10.0.0.3

packets sent = 4, Received = 4, Lost = 0

Approx round trip time in Milliseconds

Minimum = 6ms, Maximum = 21, Average = 12

* Observation :—

1. wireless local area network wlan is a group of allocated computes or other devices that form a network based on radio transmission rather than wired connections
2. After the WLAN is setup, the lined connection appears in the topology from the acss. port.

Cycle-II

Program 1

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

ii. Procedure

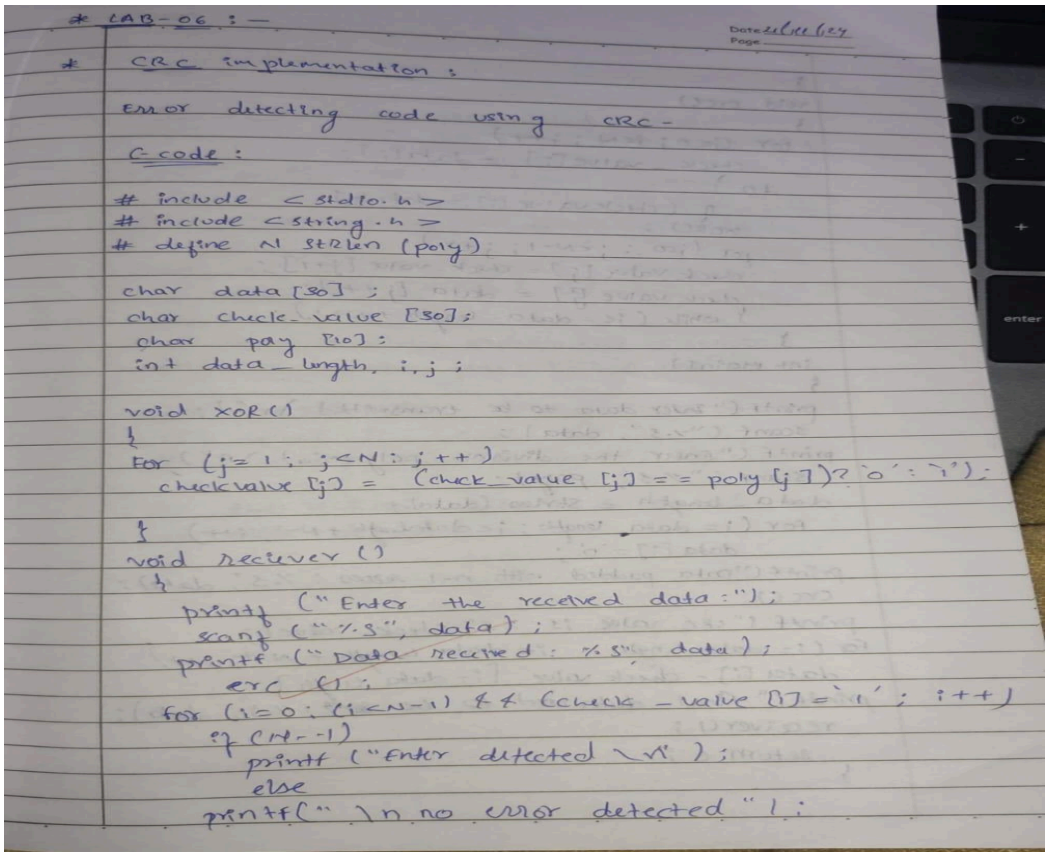
```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15  # Align the bit with CRC's uppermost bit
        for _ in range(1):  # Process the single bit
            if crc & 0x8000:  #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


# Example usage:
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., 1101001110110011011011011000011): "
)

# 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., 1101001110110011011011011000011): 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., 1101001110110011011011011000011): 11111111111111100001
CRC valid: False
```

iv. Observation



* LAB - 06 :-

* CRC implementation :

Error detecting code using CRC -

C code :

```
# include <stdio.h>
# include <string.h>
# define N strlen(poly)

char data[30];
char check_value[30];
char poly[10];
int data_length, i, j;

void XOR()
{
For (j=1; j<N; j++)
  checkvalue[j] = (check_value[j] == poly[j])? '0' : '1');

}
void receiver()
  {
    printf("Enter the received data:");
    scanf("%s", data);
    printf("Data received: %s", data);
      crc();
  for (i=0; (i<N-1) && (check_value[i]=='1'); i++)
    if (i<N-1)
      printf("Enter detected \n");
      else
    printf("\n no error detected");
```

```
}
void crc()
{
 for (i=0; i<N ; i++)
    check_value[i] = data[i];
  do {
      if (checkvalue[i] = data[0]
     xOR();
    for (j=0 : j<N-1; j++)
     check value[j] = check_value[j+1];
     check value[j] = data[j++];
    } while (i<=data.length + N+1);
 }
int main()
{
printf("Enter data to be transmitted \n");
 scanf("%s", data);
printf("Enter the divisor polynomial) \n");
 scanf("%s", poly);
data_length = strlen(data);
 for (i= data_length; i<datalength + N-1; i++)
     data[i] = '0';
printf("Data padded with n-1 zeroes : %s", data);
 crc()
printf("CRC value is %s", check_value);
For (i= data_length ; i< data_length + N-1; i++)
 data[i] = check value [i- data_length];
printf(" final data word to be sent : %s", data);
 receiver();
  return 0;
}
```

Output:

Enter data to be transmitted : 101010
Enter the divisor polynomial : 1011

Data padded with n-1 zeroes : 101010000
CRC value is : 001
Final codeword to be sent : 101010001

Enter the received data : 100010001
error detected

Enter data to be transmitted : 101100
Enter the divisor polynomial : 1001

Data padded with n-1 zeroes : 101100000
CRC value is : 001
Final codeword to be sent : 101100001

Enter the received data : 101100001
No error detected.

Program 2

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

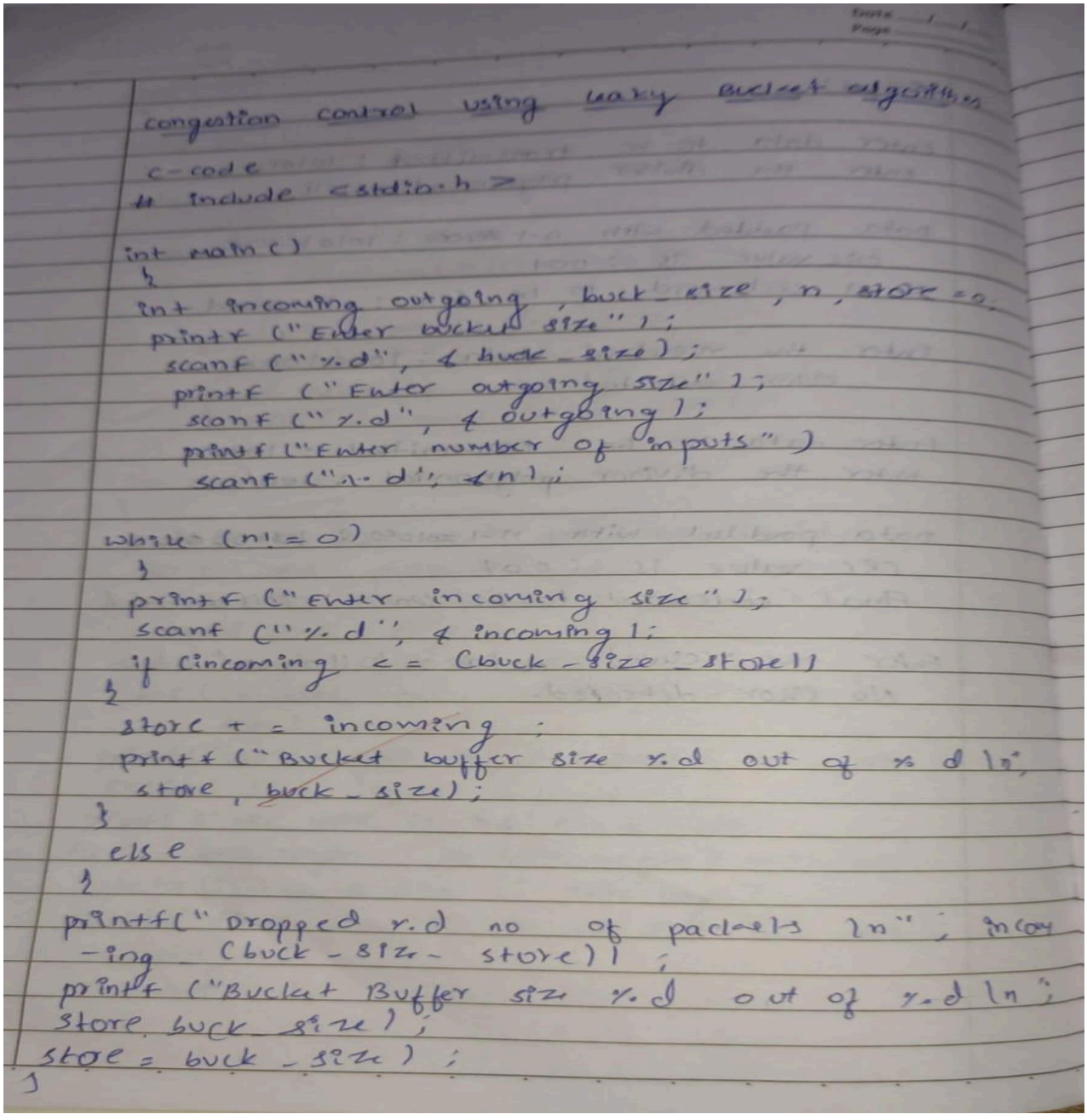
ii. Procedure

```
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



congestion control using leaky bucket algorithm

c-code

```
#include <stdio.h>

int main()
{
int incoming, outgoing, buck_size, n, store=0;
printf("Enter bucket size");
scanf("%d", &buck_size);
printf("Enter outgoing size");
scanf("%d", &outgoing);
printf("Enter number of inputs")
scanf("%d", &n);

while(n!=0)
{
printf("Enter incoming size");
scanf("%d", &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 : 5000

Enter outgoing size : 2000

Enter no of inputs : 2

Enter incoming packet size : 3000

Bucket buffer size 3000 out of 5000

After outgoing 1000 packets left out of 5000 in buffer

Enter incoming packet size : 1000

Bucket buffer size 2000 out of 5000

After outgoing 0 packets left out of 5000 in buffer.

-

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

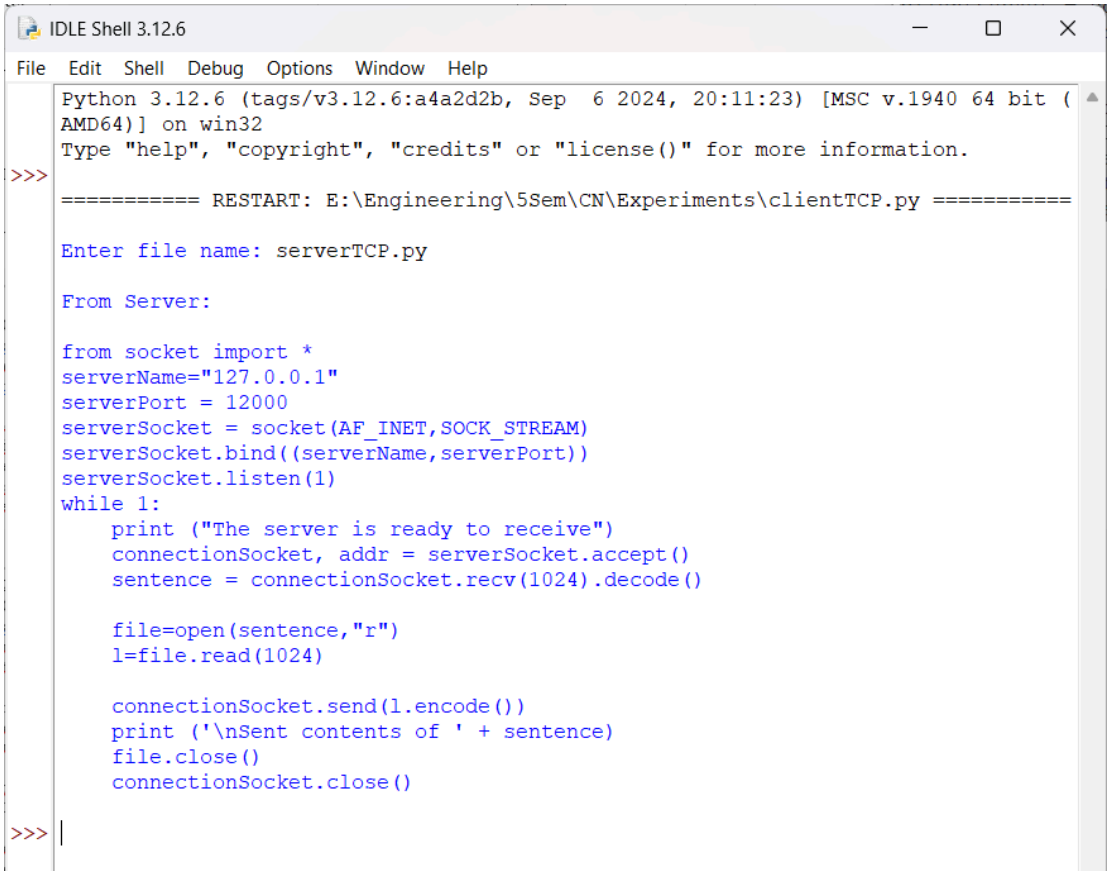
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

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

serverTCP.py

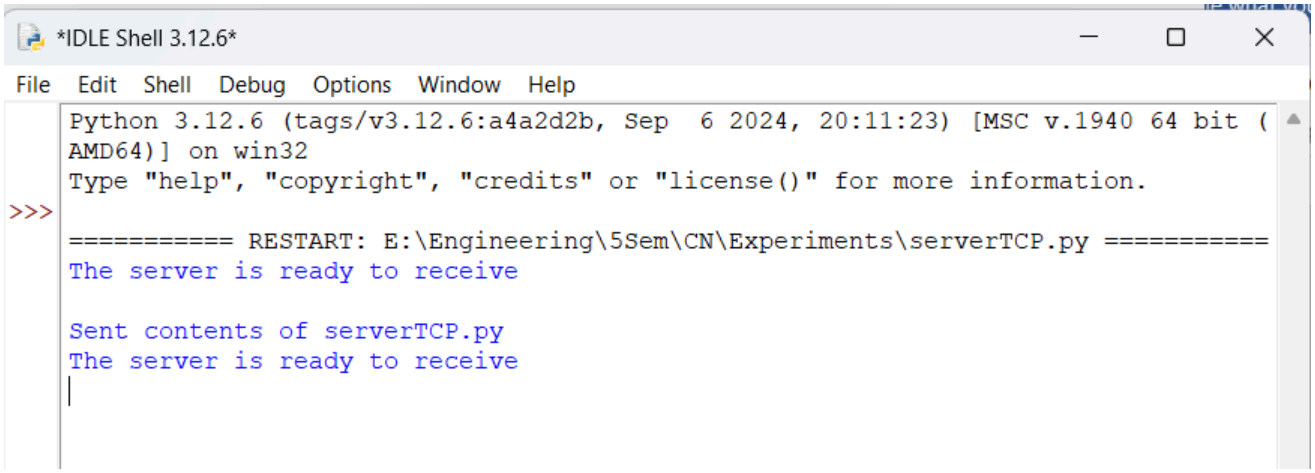
```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    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

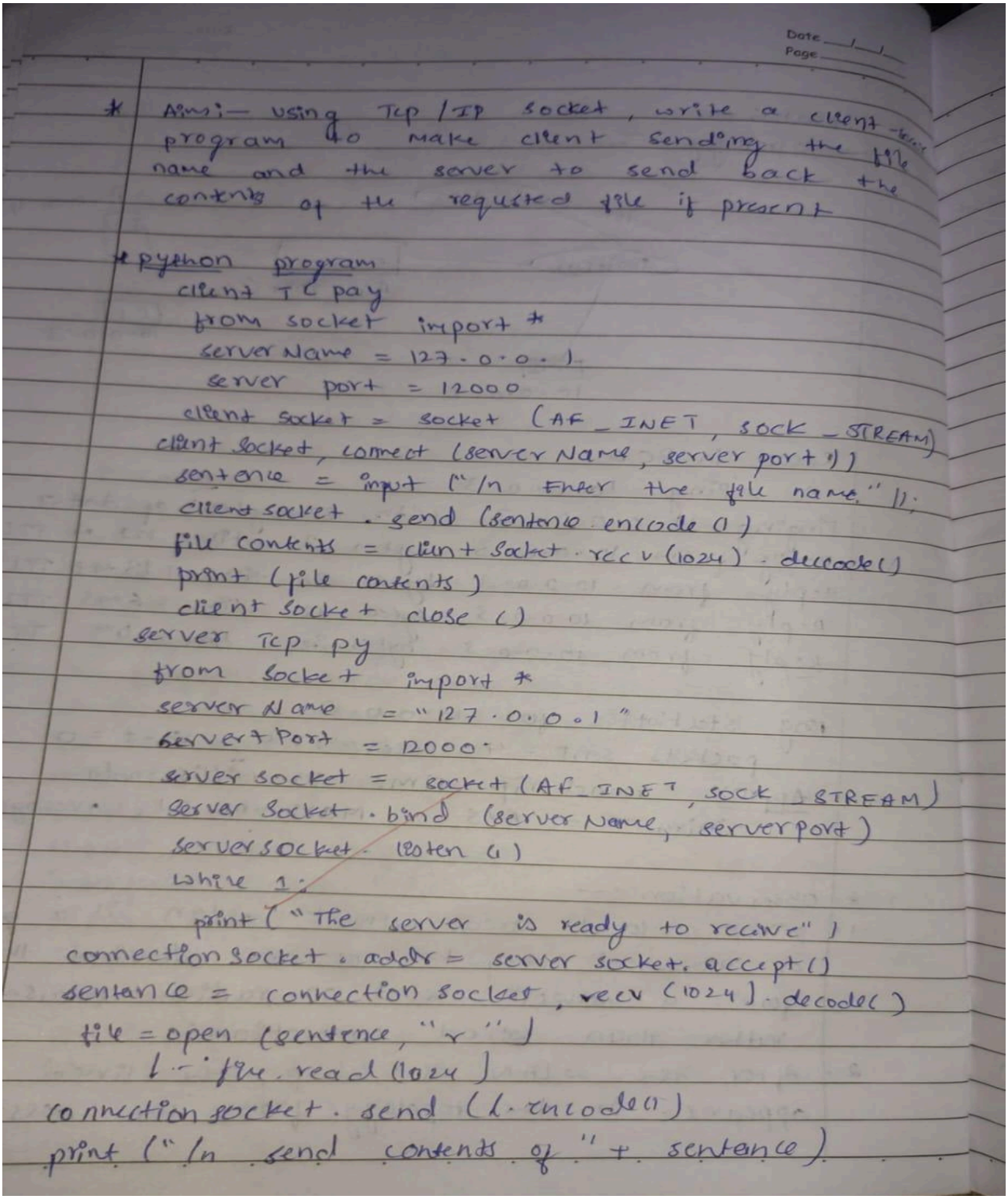
Client



Server



iv. Observation



Aim:— Using TCP/IP socket, 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

Python program

client TC.py

```
from socket import *
serverName = 127.0.0.1
server port = 12000
client socket = socket (AF_INET, SOCK_STREAM)
client socket, connect (serverName, server port))
sentence = input ("\n Enter the file name.");
client socket . send (sentence encode ())
file contents = clint Socket . recv (1024) . decode()
print (file contents)
client socket close ()
```

server TCP.py

```
from socket import *
server Name = "127.0.0.1"
server Port = 2000.
server socket = socket (AF_INET, SOCK_STREAM)
server Socket . bind (server Name, server port)
server socket . listen (1)
while 1:
print ("The server is ready to receive")
connection socket . addr = server socket. accept()
sentance = connection socket, recv (1024) . decode()
file = open (sentence, "r")
l = file. read (1024)
connection socket . send (l. encode())
print ("\n send contents of " + sentence)
```

file close()
connection socket . close()

Result :

Client window :

Enter the file name : serverTcp.py

Contents of the file are displayed

server window :

The server is ready to receive

sent contents of serverTcp.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()
```

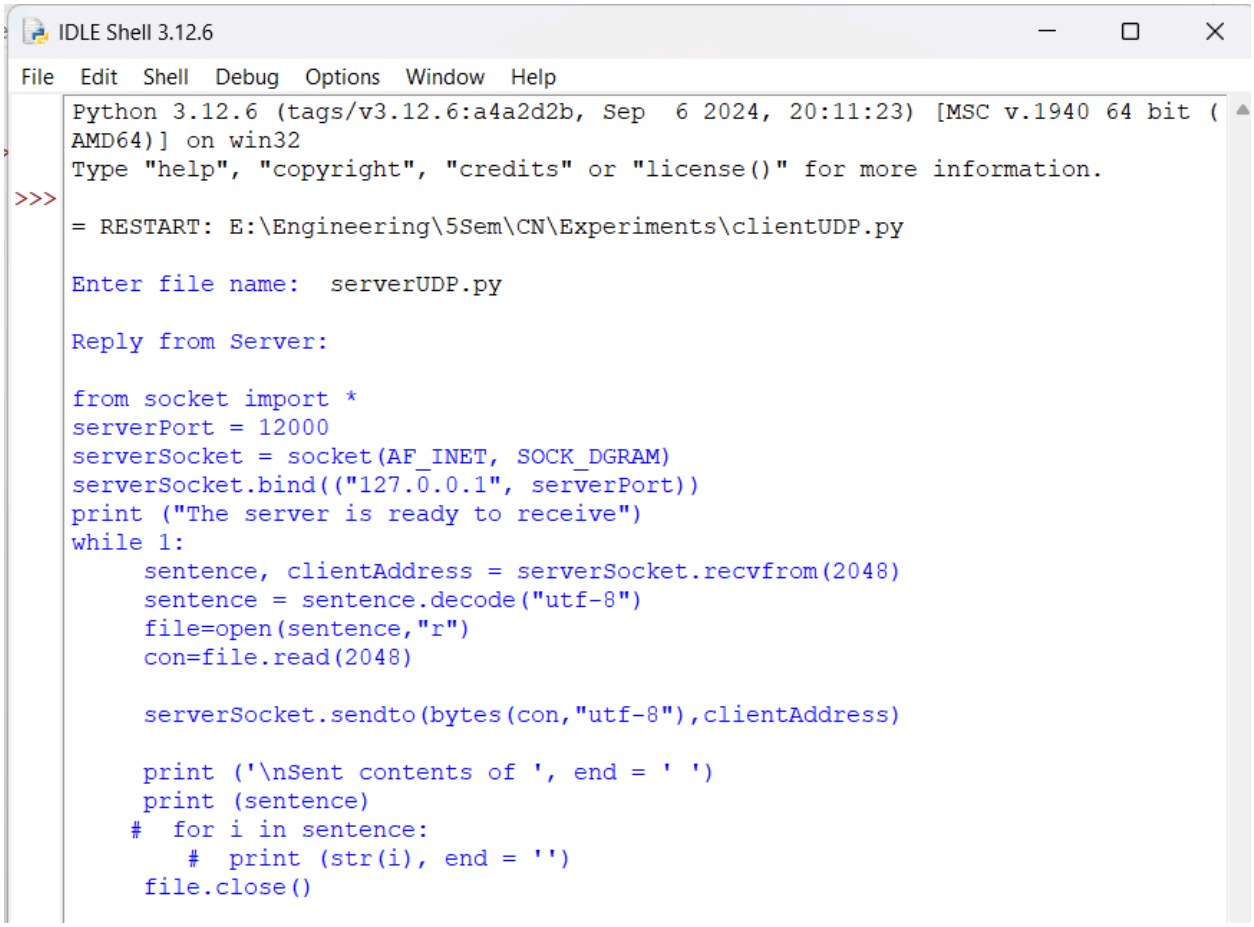
serverUDP.py

```
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:
```

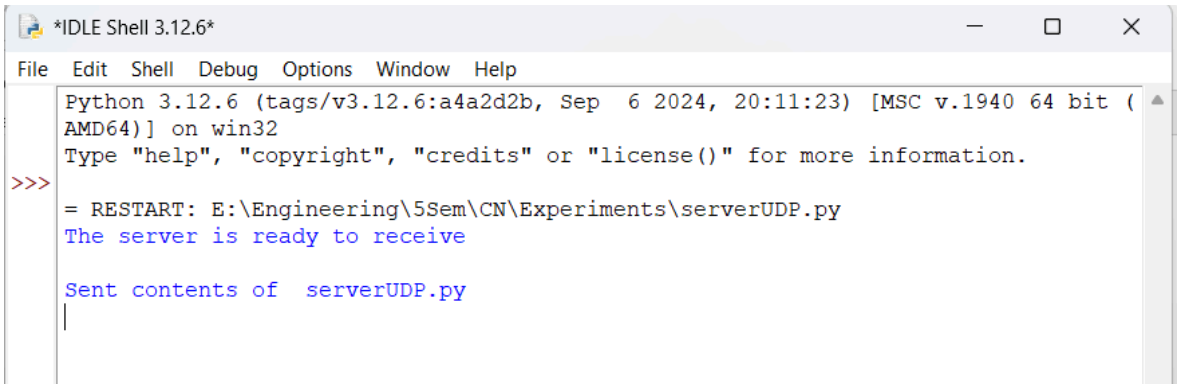
```
    #  print (str(i), end = '')
  file.close()
```

iii. Screen shots/ output

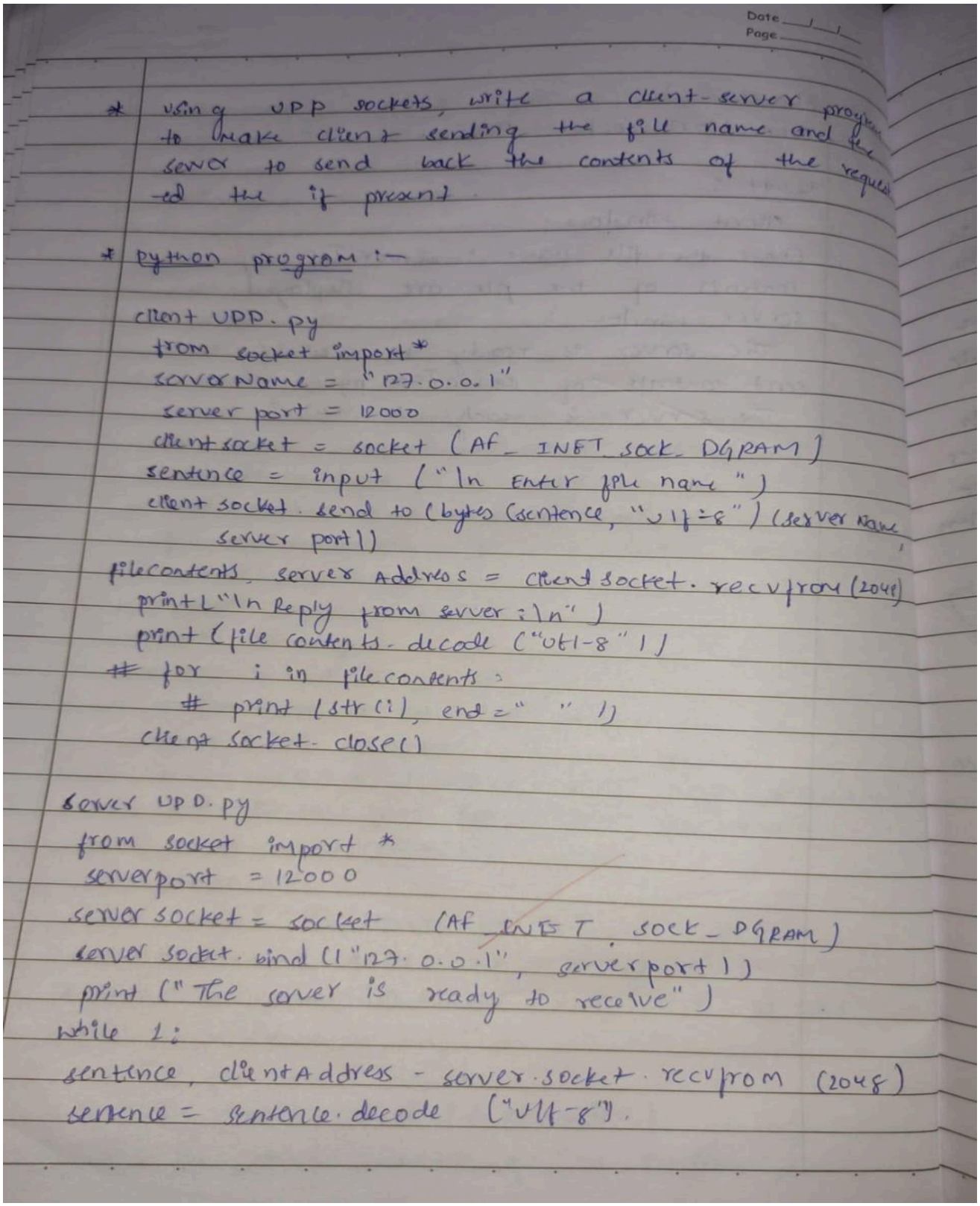
Client



Server



iv. Observation



* 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

* python program :-

client UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverport = 12000
clientsocket = socket (Af_INET, sock_DGRAM)
sentence = input ("\n Enter file name")
clientsocket.sendto (bytes (sentence, "utf-8") (servername
    serverport))
filecontents, serverAddress = clientsocket.recvfrom (2048)
print ("\n Reply from server :\n")
print (file contents.decode ("utf-8"))
# for i in filecontents:
    # print (str (i), end = " ")
clientsocket.close()
```

Server UDP.py

```
from socket import *
serverport = 12000
serversocket = socket (AF_INET, SOCK_DGRAM)
serversocket.bind (("127.0.0.1", serverport))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serversocket.recvfrom (2048)
sentence = sentence.decode ("utf-8").
```

```
file = open (sentence, "r")
con = file.read (2048)

serverSocket.sendto (bytes(con, "utf-8"),
                     clientAddress)
print ("In sent contents of:", end = ' ')
print (sentence)
# for i in sentence:
#     print (str(i), end = ' ')
file.close()
```

Result -

Client window:

Enter the file name : serverTcp.py

Contents of the file are displayed

server window:

The server is ready to receive

sent contents of serverTcp.py

The server is ready to receive.