VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Networks

Submitted by

Bhuvana M (1BM22CS071)

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 **Bhuvana M (1BM22CS071),** who is a 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.

Dr. Nandhini Vineeth Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Cycle - I

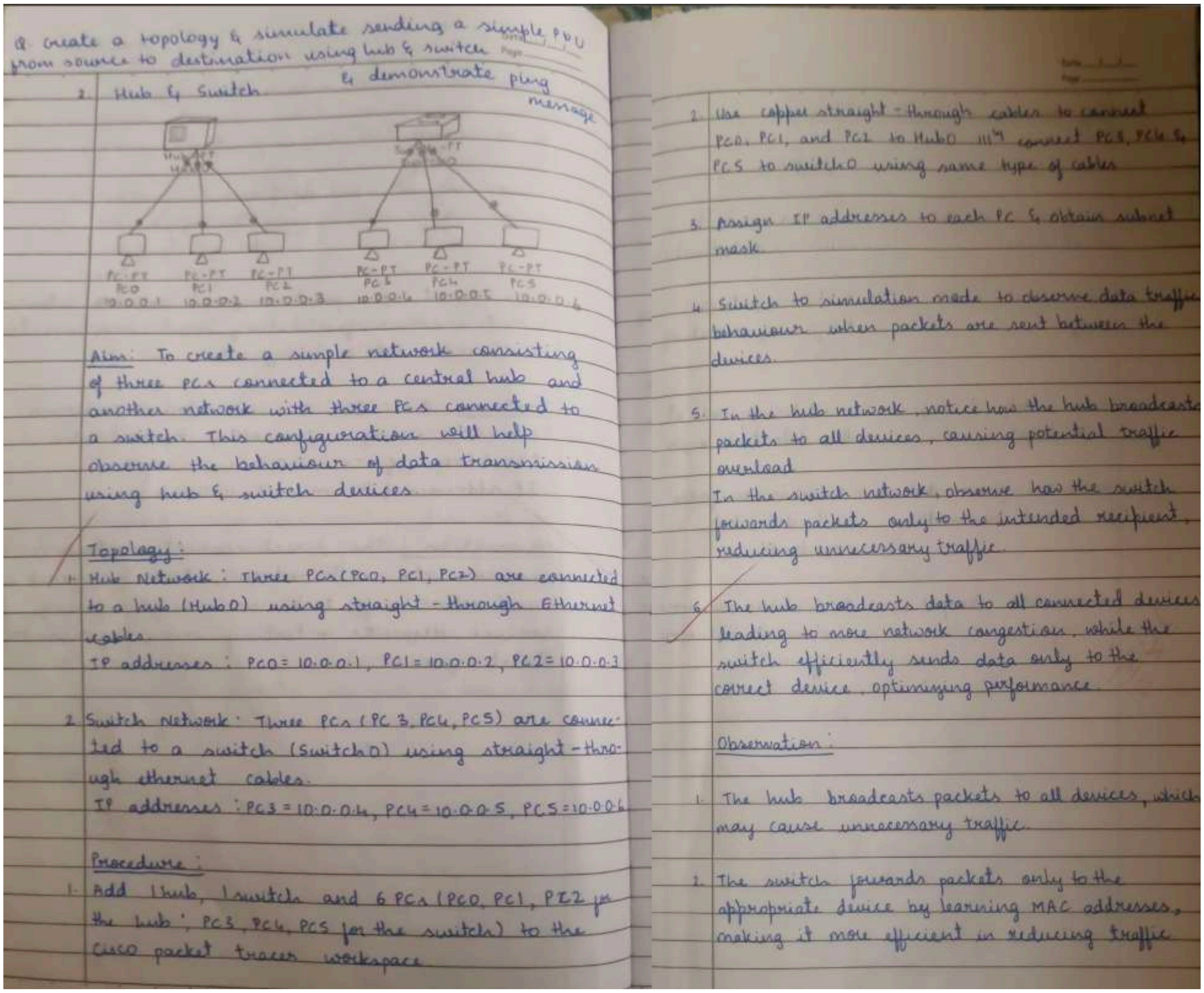
Cycle-II

Github Link:
https://github.com/Bhu06/CN-LAB.git

Program 1

Q1: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Observation:



Hub-PT
Hub0
Switch-PT
Switch0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3
PC-PT
PC3
10.0.0.4
PC-PT
PC4
10.0.0.5
PC-PT
PC5
10.0.0.6

Q. create a topology & simulate sending a simple PDU from source to destination using hub & switch & demonstrate ping message

1. Hub & Switch

Aim: To create a simple network consisting of three PCs connected to a central hub and another network with three PCs connected to a switch. This configuration will help observe the behaviour of data transmission using hub & switch devices.

Topology:

1. Hub Network: Three PCs (PC0, PC1, PC2) are connected to a hub (Hub0) using straight-through Ethernet cables.
   IP addresses: PC0 = 10.0.0.1, PC1 = 10.0.0.2, PC2 = 10.0.0.3
2. Switch Network: Three PCs (PC 3, PC4, PC5) are connected to a switch (Switch0) using straight-through ethernet cables.
   IP addresses: PC3 = 10.0.0.4, PC4 = 10.0.0.5, PC5 = 10.0.0.6

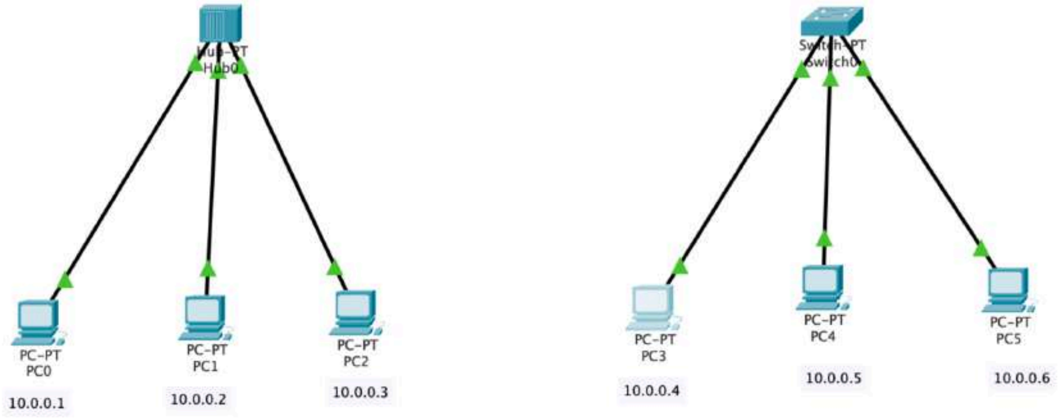
Procedure:

1. Add 1 hub, 1 switch and 6 PCs (PC0, PC1, PC2 for the hub; PC3, PC4, PC5 for the switch) to the Cisco packet tracer workspace
2. Use copper straight-through cables to connect PC0, PC1, and PC2 to Hub0 ||ly connect PC3, PC4 & PC5 to switch0 using same type of cables
3. Assign IP addresses to each PC & obtain subnet mask.
4. Switch to simulation mode to observe data traffic behaviour when packets are sent between the devices.
5. In the hub network, notice how the hub broadcasts packets to all devices, causing potential traffic overload
   In the switch network, observe how the switch forwards packets only to the intended recipient, reducing unnecessary traffic.
6. The hub broadcasts data to all connected devices leading to more network congestion, while the switch efficiently sends data only to the correct device, optimizing performance.

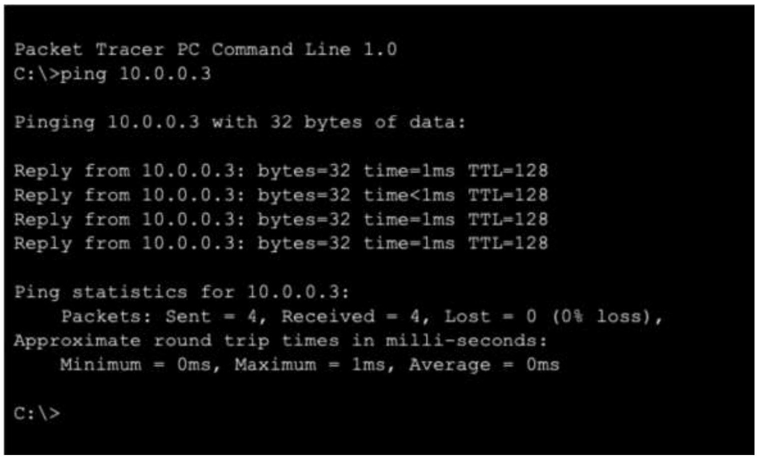
Observation:

1. The hub broadcasts packets to all devices, which may cause unnecessary traffic.
2. The switch forwards packets only to the appropriate device by learning MAC addresses, making it more efficient in reducing traffic

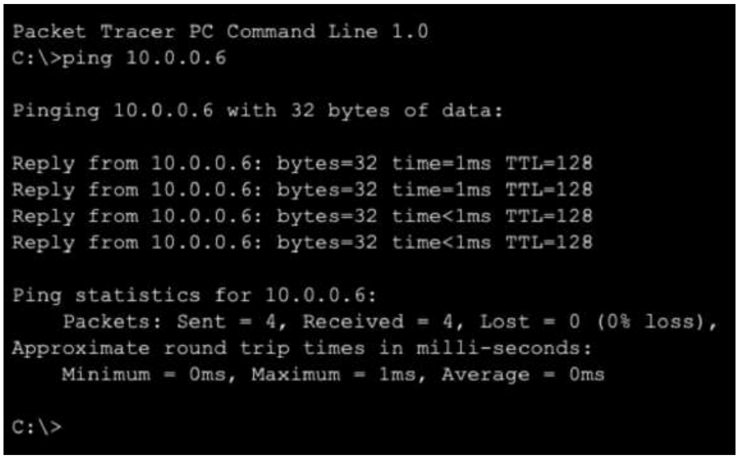
Screenshot of Topology:



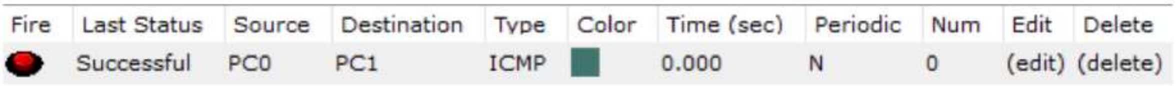
Screenshot of Output:



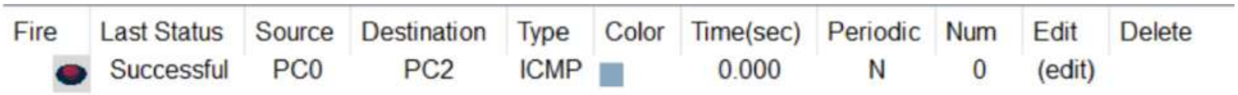
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

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

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```



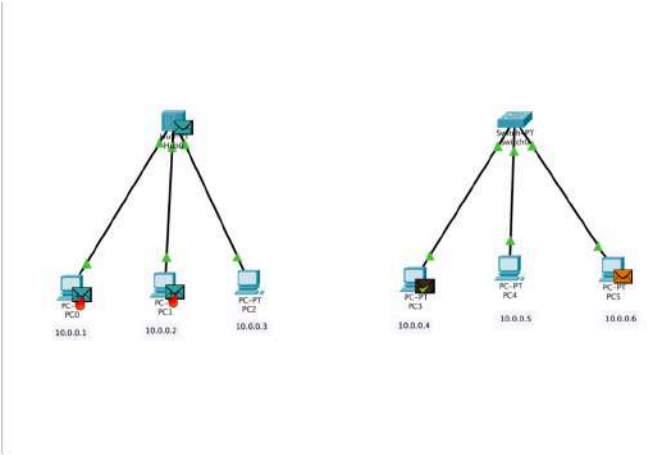
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data:

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

Ping statistics for 10.0.0.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

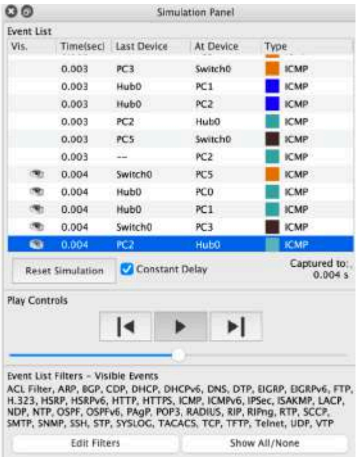


Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delete)



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC2	ICMP	🟦	0.000	N	0	(edit)	





Event List

Vis.	Time(sec)	Last Device	At Device	Type
	0.003	PC3	Switch0	🟧 ICMP
	0.003	Hub0	PC1	🟦 ICMP
	0.003	Hub0	PC2	🟦 ICMP
	0.003	PC2	Hub0	🟩 ICMP
	0.003	PC5	Switch0	🟫 ICMP
	0.003	--	PC2	🟩 ICMP
👁	0.004	Switch0	PC5	🟧 ICMP
👁	0.004	Hub0	PC0	🟩 ICMP
👁	0.004	Hub0	PC1	🟩 ICMP
👁	0.004	Switch0	PC3	🟫 ICMP
👁	0.004	PC2	Hub0	🟩 ICMP

☑ Constant Delay

Captured to:
0.004 s

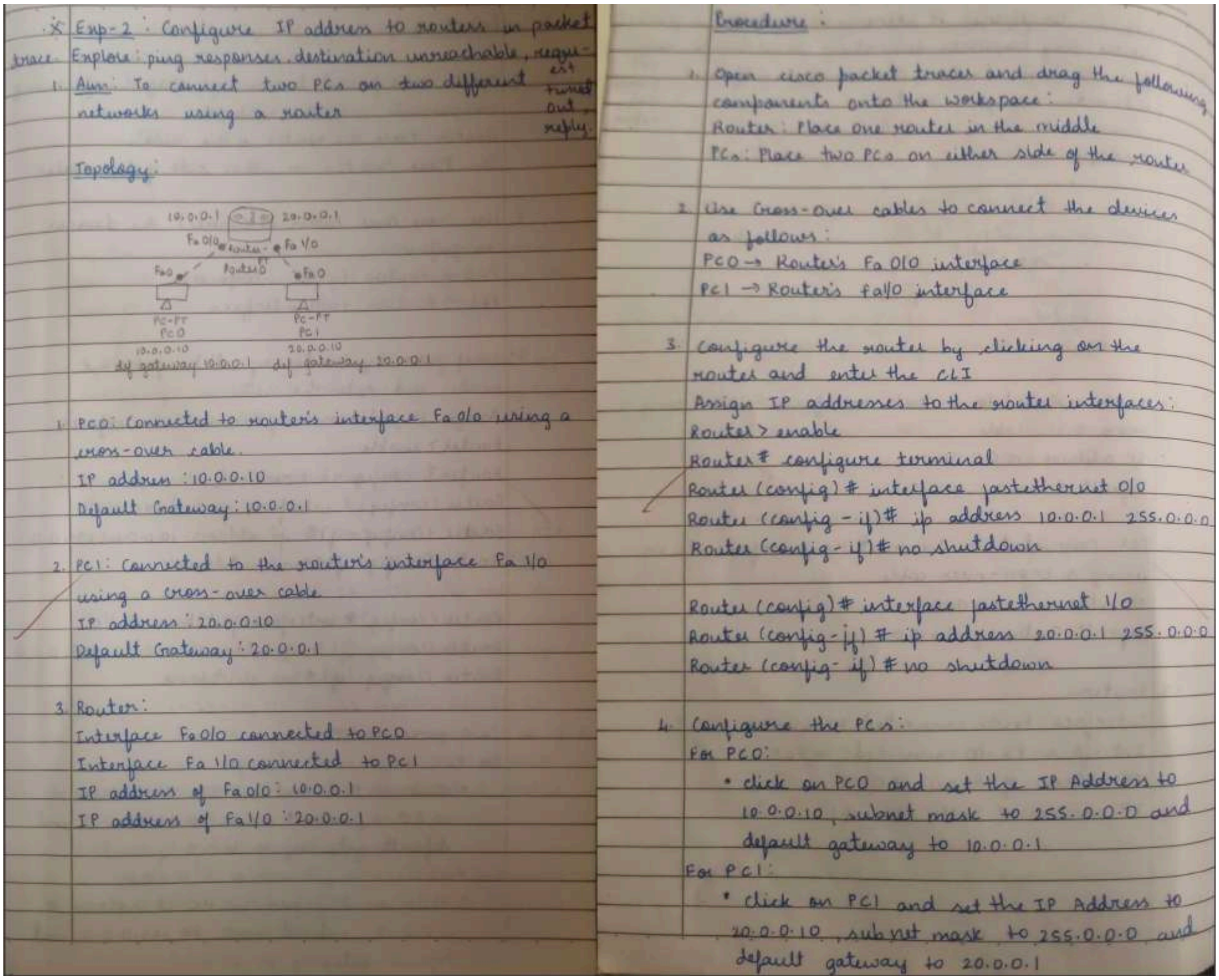
Play Controls

Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Program 2

Q2 (a): Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

Observation:



10.0.0.1
20.0.0.1
Fa 0/0
Router
Fa 1/0
Fa0
Router0
Fa 0
PC-PT
PC0
PC-PT
PC1
10.0.0.10
20.0.0.10
df gateway 10.0.0.1
df gateway 20.0.0.1

Exp-2 : Configure IP address to routers in packet trace. Explore: ping responses, destination unreachable, request timed out, reply.

1. Aim: To connect two PCs on two different networks using a router

Topology:

1. PC0: Connected to router's interface Fa 0/0 using a cross-over cable.
   IP address : 10.0.0.10
   Default Gateway : 10.0.0.1

2. PC1: Connected to the router's interface Fa 1/0 using a cross-over cable
   IP address : 20.0.0.10
   Default Gateway : 20.0.0.1

3. Router:
   Interface Fa0/0 connected to PC0
   Interface Fa 1/0 connected to PC1
   IP address of Fa 0/0 : 10.0.0.1
   IP address of Fa1/0 : 20.0.0.1

Procedure :

1. Open cisco packet tracer and drag the following components onto the workspace:
   Router: Place one router in the middle
   PCs: Place two PCs on either side of the router

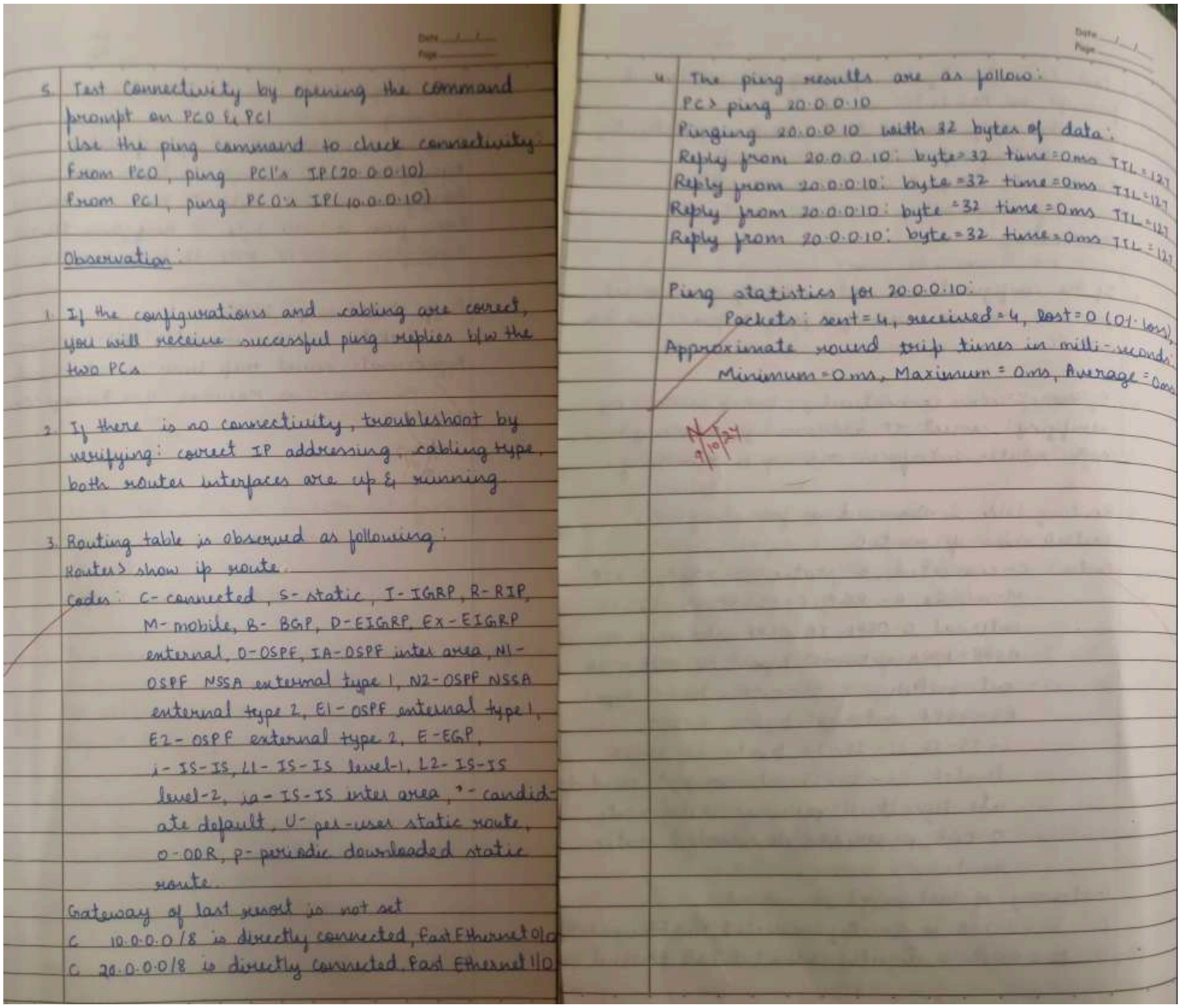
2. Use Cross-over cables to connect the devices as follows:
   PC0 → Router's Fa 0/0 interface
   PC1 → Router's fa1/0 interface

3. Configure the router by clicking on the router and enter the CLI
   Assign IP addresses to the router interfaces:

```
Router> enable
Router# configure 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)# interface fastethernet 1/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
Router (config- if) # no shutdown
```

4. Configure the PCs:
   For PC0:
   - click on PC0 and set the IP Address to 10.0.0.10, subnet mask to 255.0.0.0 and default gateway to 10.0.0.1

   For PC1:
   - click on PC1 and set the IP Address to 20.0.0.10, subnet mask to 255.0.0.0 and default gateway to 20.0.0.1



5. Test Connectivity by opening the command prompt on PC0 & PC1
Use the ping command to check connectivity:
From PC0, ping PC1's IP(20.0.0.10)
From PC1, ping PC0's IP(10.0.0.10)

Observation:

1. If the configurations and cabling are correct, you will receive successful ping replies b/w the two PCs

2. If there is no connectivity, troubleshoot by verifying: correct IP addressing, cabling type, both router interfaces are up & running.

3. Routing table is observed as following:

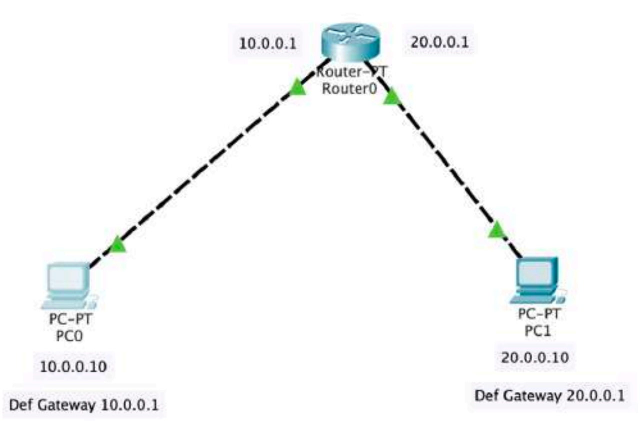
```
Router> show ip route.
Codes: C- connected, S- static, I- IGRP, R- RIP,
       M- mobile, B- BGP, D- EIGRP, EX- EIGRP
       external, O- OSPF, IA- OSPF inter area, N1-
       OSPF NSSA external type 1, N2- OSPF NSSA
       external type 2, E1- OSPF external type 1,
       E2- OSPF external type 2, E- EGP,
       i- IS-IS, L1- IS-IS level-1, L2- IS-IS
       level-2, ia- IS-IS inter area, *- candid-
       ate default, U- per-user static route,
       o- ODR, P- periodic downloaded static
       route.
Gateway of last resort is not set
C   10.0.0.0/8 is directly connected, FastEthernet0/0
C  20.0.0.0/8 is directly connected, Fast Ethernet1/0
```

4. The ping results are as follow:

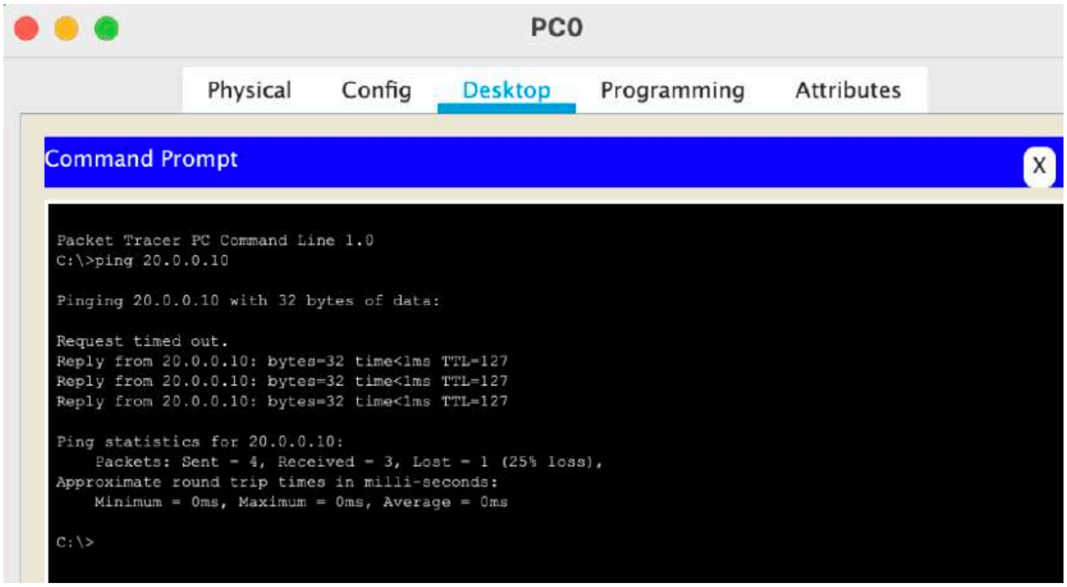
```
PC> ping 20.0.0.10
Pinging 20.0.0.10 with 32 bytes of data:
Reply from 20.0.0.10: byte=32 time=0ms TTL=127
Reply from 20.0.0.10: byte=32 time=0ms TTL=127
Reply from 20.0.0.10: byte=32 time=0ms TTL=127
Reply from 20.0.0.10: byte=32 time=0ms TTL=127

Ping statistics for 20.0.0.10:
     Packets: sent=4, received=4, lost=0 (0% loss),
Approximate round trip times in milli-seconds:
     Minimum=0ms, Maximum=0ms, Average=0ms
```

9/10/24

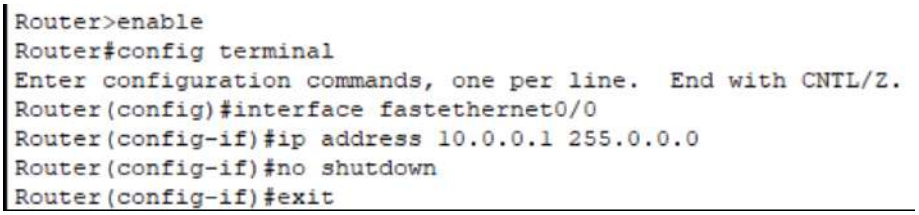
Screenshot of Topology:



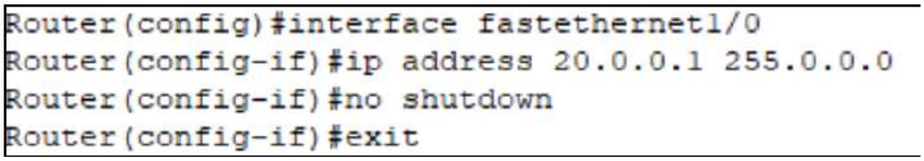
Screenshot of Output:



```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```



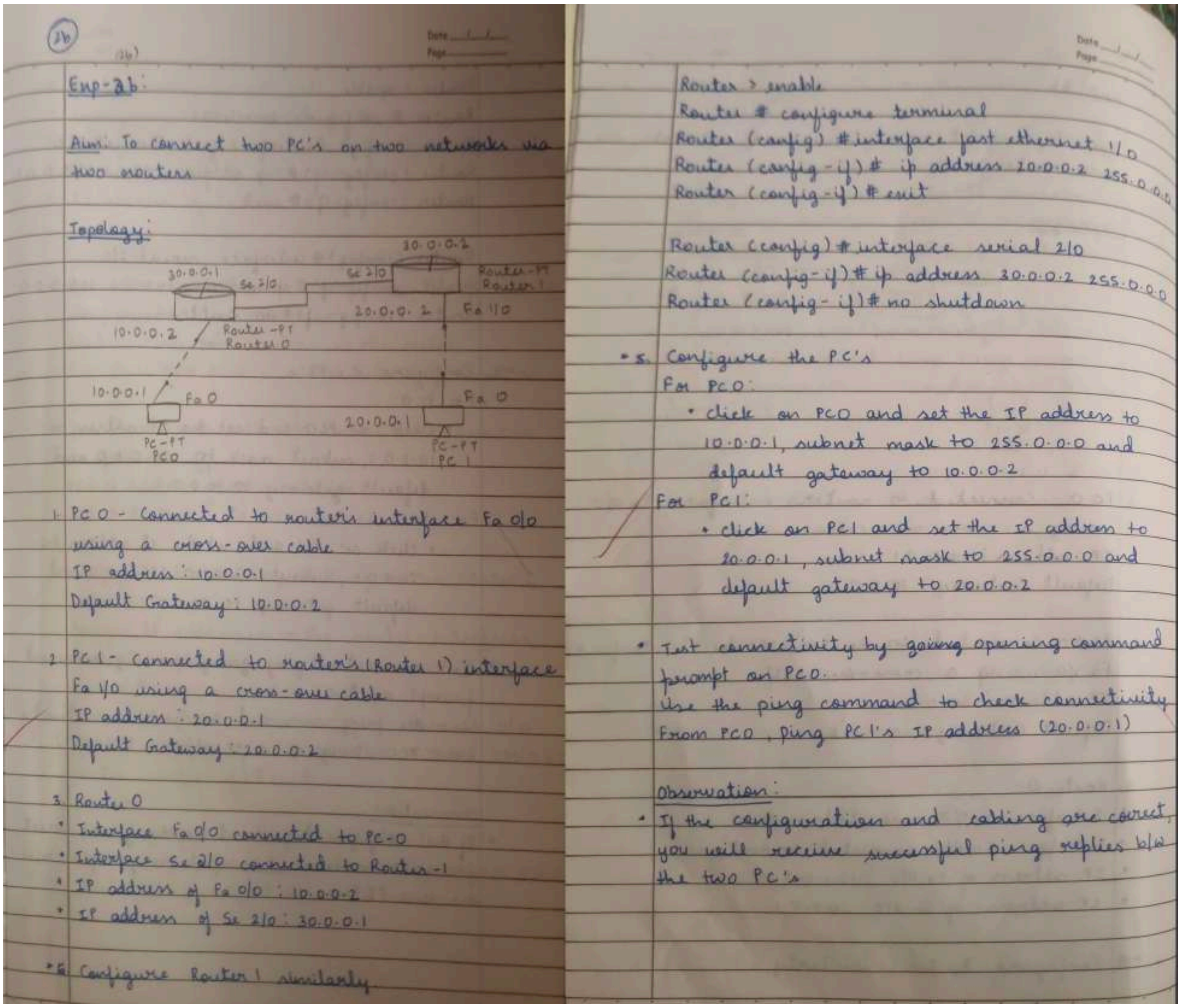
```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
```



```
Router(config)#interface fastethernet1/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
```

Q2 (b): To connect two PC's on two networks and two routers

Observation:


30.0.0.1
Se 2/0
Se 2/0
30.0.0.2
Router-PT
Router 1
20.0.0.2
Fa 1/0
10.0.0.2
Router-PT
Router 0
10.0.0.1
Fa 0
Fa 0
20.0.0.1
PC-PT
PC0
PC-PT
PC 1

2b

(2b)

Date __/__/__
Page ____

Exp-2b:

Aim: To connect two PC's on two networks via two routers

Topology:

1. PC 0 - Connected to router's interface Fa 0/0 using a cross-over cable
IP address: 10.0.0.1
Default Gateway: 10.0.0.2

2. PC 1 - connected to router's (Router 1) interface Fa 1/0 using a cross-over cable
IP address: 20.0.0.1
Default Gateway: 20.0.0.2

3. Router 0
- Interface Fa 0/0 connected to PC-0
- Interface Se 2/0 connected to Router-1
- IP address of Fa 0/0: 10.0.0.2
- IP address of Se 2/0: 30.0.0.1

• 4. Configure Router 1 similarly.

Date __/__/__
Page ____

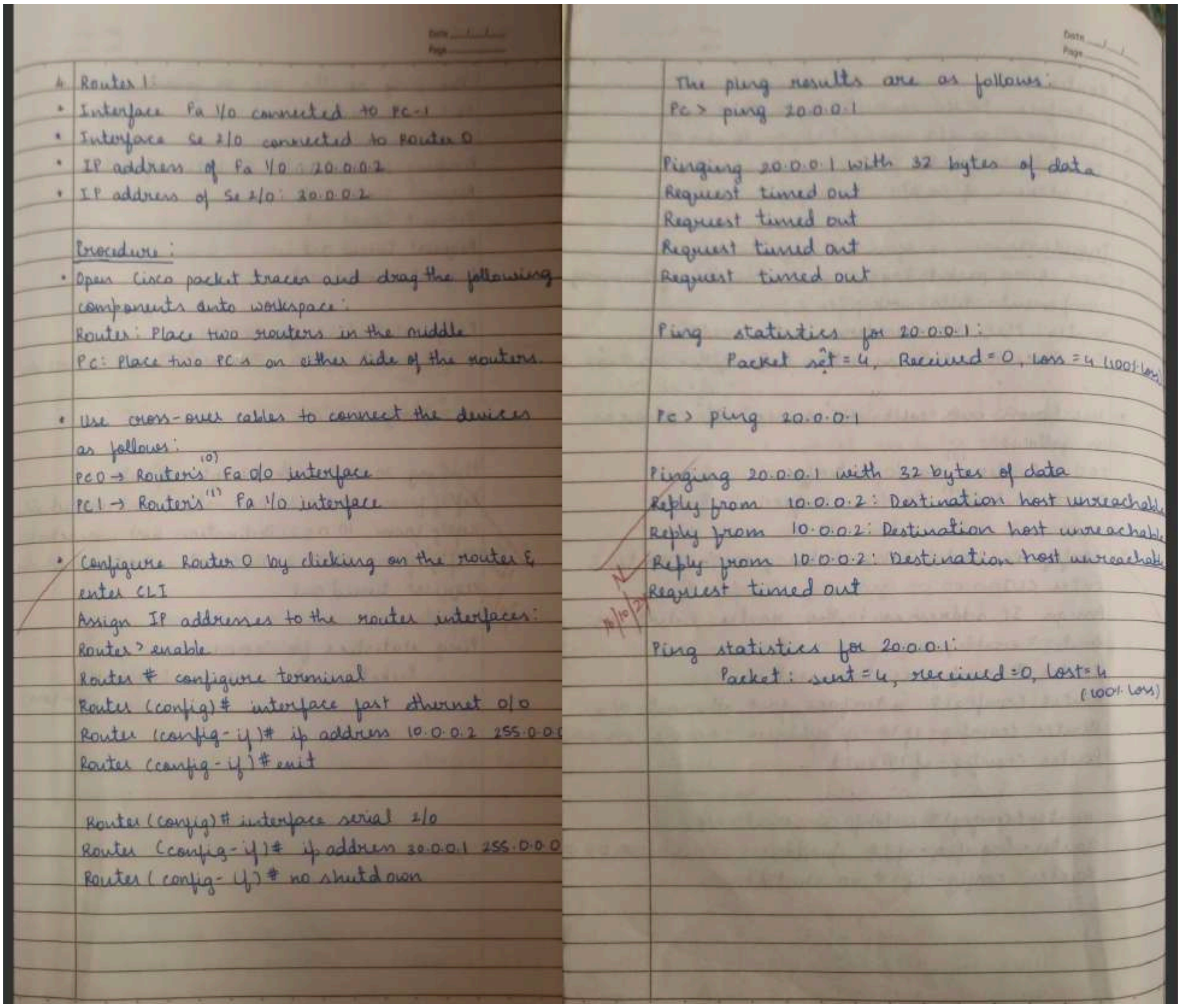
```
Router > enable
Router # configure terminal
Router (config) #interface fast ethernet 1/0
Router (config-if)# ip address 20.0.0.2 255.0.0.0
Router (config-if) # exit

Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.2 255.0.0.0
Router (config-if)# no shutdown
```

• 5. Configure the PC's
For PC0:
- click on PC0 and set the IP address to 10.0.0.1, subnet mask to 255.0.0.0 and default gateway to 10.0.0.2

For PC1:
- click on PC1 and set the IP address to 20.0.0.1, subnet mask to 255.0.0.0 and default gateway to 20.0.0.2

- Test connectivity by going opening command prompt on PC0.
Use the ping command to check connectivity
From PC0, ping PC1's IP address (20.0.0.1)

Observation:
- If the configuration and cabling are correct, you will receive successful ping replies b/w the two PC's



4. Router 1:
- Interface Fa 1/0 connected to PC-1
- Interface Se 2/0 connected to Router 0
- IP address of Fa 1/0 : 20.0.0.2
- IP address of Se 2/0: 30.0.0.2

Procedure:

- Open Cisco packet tracer and drag the following components onto workspace:
Router: Place two routers in the middle
PC: Place two PC's on either side of the routers.

- Use cross-over cables to connect the devices as follows:
PC0 → Router's(0) Fa 0/0 interface
PC1 → Router's(1) Fa 1/0 interface

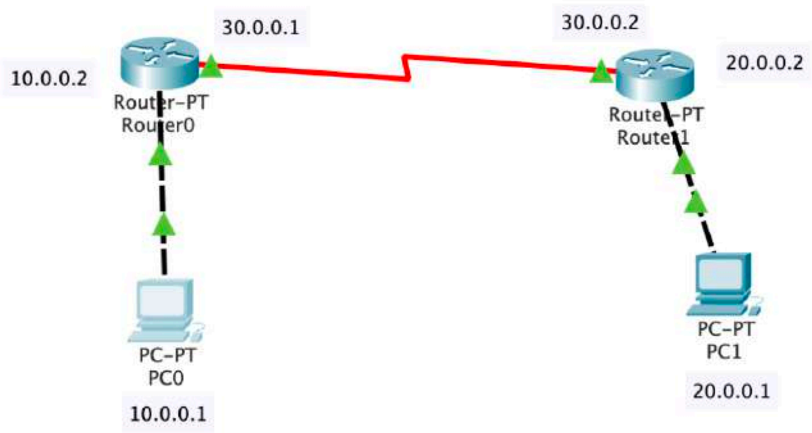
- Configure Router 0 by clicking on the router & enter CLI
Assign IP addresses to the router interfaces:

```
Router> enable
Router # configure terminal
Router (config)# interface fast ethernet 0/0
Router (config-if)# ip address 10.0.0.2 255.0.0.0
Router (config-if)# exit

Router(config)# interface serial 2/0
Router (config-if)# ip address 30.0.0.1 255.0.0.0
Router( config- if)# no shutdown
```

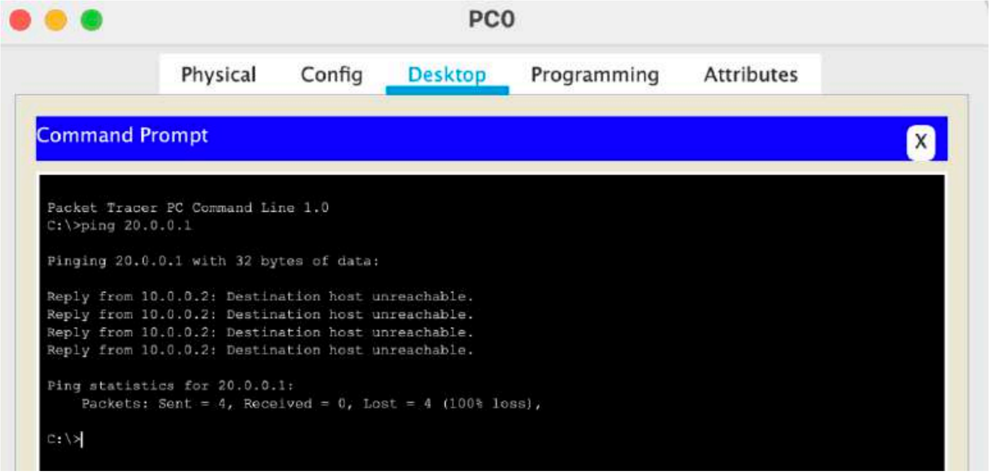
The ping results are as follows:

```
PC > ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data
Request timed out
Request timed out
Request timed out
Request timed out

Ping statistics for 20.0.0.1:
        Packet sent = 4, Received = 0, Loss = 4 (100% loss)

PC > ping 20.0.0.1

Pinging 20.0.0.1 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
Request timed out

Ping statistics for 20.0.0.1:
        Packet: sent = 4, received = 0, Lost = 4
                                        (100% Loss)
```

Screenshot of Topology:



Screenshot of Output:

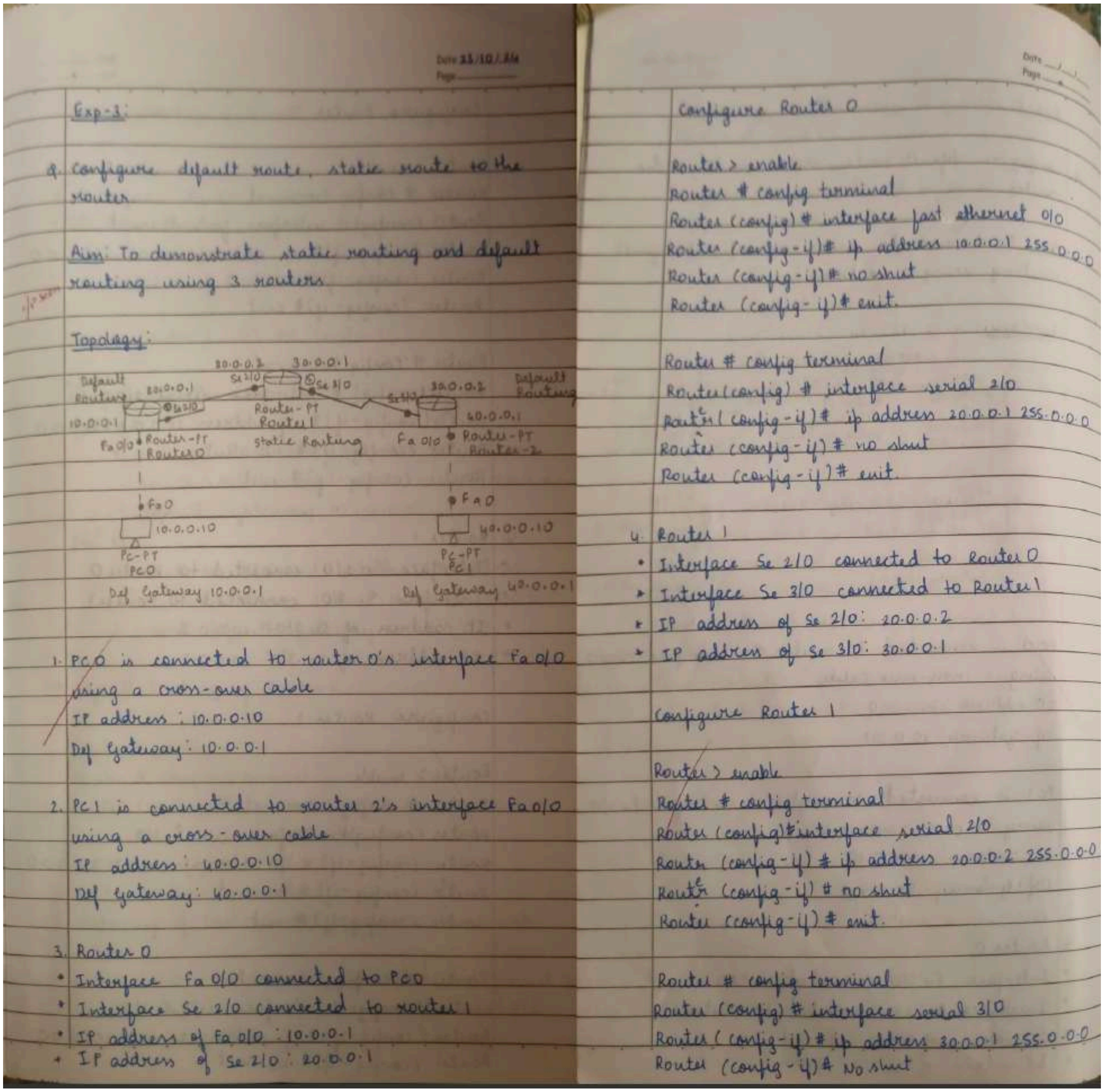


```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

Program 3

Q3: Configure default route, static route to the Router

Observation:


20.0.0.2
30.0.0.1
Default Routing
20.0.0.1
Se 2/0
Se 3/0
30.0.0.2
Default Routing
10.0.0.1
Router-PT
Router1
40.0.0.1
Fa 0/0
Router-PT
Router0
Static Routing
Fa 0/0
Router-PT
Router-2
Fa 0
Fa 0
10.0.0.10
40.0.0.10
PC-PT
PC0
PC-PT
PC1
Def Gateway 10.0.0.1
Def Gateway 40.0.0.1

Date 23/10/24

Exp-3:

Q. Configure default route, static route to the router

Aim: To demonstrate static routing and default routing using 3 routers.

Topology:

1. PC0 is connected to router 0's interface Fa 0/0 using a cross-over cable
IP address: 10.0.0.10
Def Gateway: 10.0.0.1

2. PC1 is connected to router 2's interface Fa 0/0 using a cross-over cable
IP address: 40.0.0.10
Def gateway: 40.0.0.1

3. Router 0
- Interface Fa 0/0 connected to PC0
- Interface Se 2/0 connected to router 1
- IP address of Fa 0/0 : 10.0.0.1
- IP address of Se 2/0 : 20.0.0.1

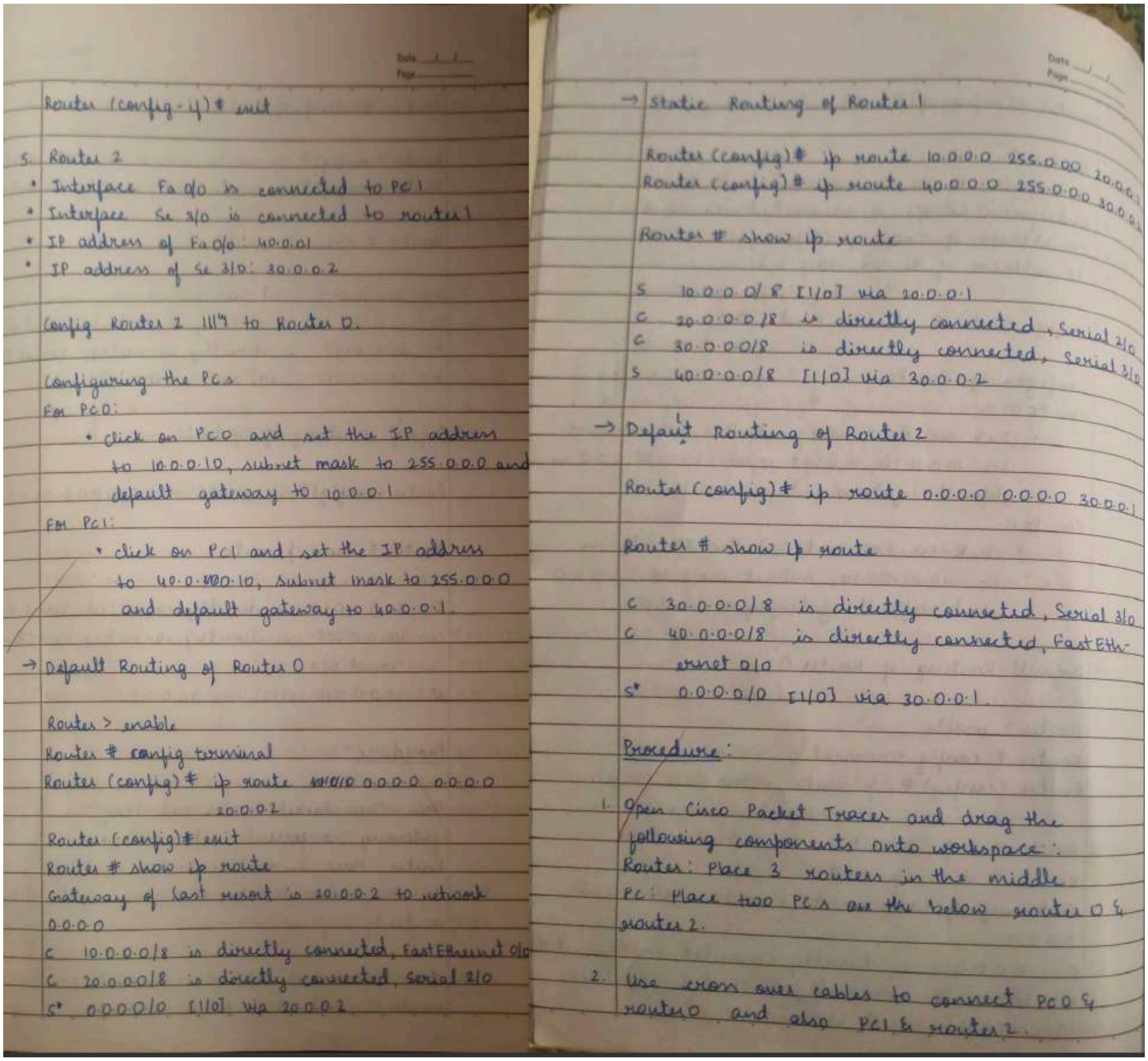
Configure Router 0

```
Router > enable
Router # config terminal
Router (config) # interface fast ethernet 0/0
Router (config-if)# ip address 10.0.0.1 255.0.0.0
Router (config-if)# no shut
Router (config-if)# exit.

Router # config terminal
Router(config) # interface serial 2/0
Router (config-if)# ip address 20.0.0.1 255.0.0.0
Router (config-if) # no shut
Router (config-if)# exit.
```

4. Router 1
- Interface Se 2/0 connected to Router 0
- Interface Se 3/0 connected to Router 1
- IP address of Se 2/0: 20.0.0.2
- IP address of Se 3/0: 30.0.0.1

Configure Router 1

```
Router > enable
Router # config terminal
Router (config)#interface serial 2/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit.

Router # config terminal
Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if)# No shut
```



Date / /
Page

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

5. Router 2
- Interface Fa 0/0 is connected to PC1
- Interface Se 3/0 is connected to router1
- IP address of Fa 0/0: 40.0.0.1
- IP address of Se 3/0: 30.0.0.2

Config Router 2 ||ly to Router 0.

Configuring the PCs

For PC0:

- click on PC0 and set the IP address to 10.0.0.10, subnet mask to 255.0.0.0 and default gateway to 10.0.0.1

For PC1:

- click on PC1 and set the IP address to 40.0.~~00~~0.10, subnet mask to 255.0.0.0 and default gateway to 40.0.0.1.

→ Default Routing of Router 0

```
Router > enable
Router # config terminal
Router (config)# ip route 0.0.0.0 0.0.0.0
                    20.0.0.2
Router (config)# exit
Router # show ip route
Gateway of last resort is 20.0.0.2 to network
0.0.0.0
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
```

Date / /
Page

→ Static Routing of 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 # show ip route

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

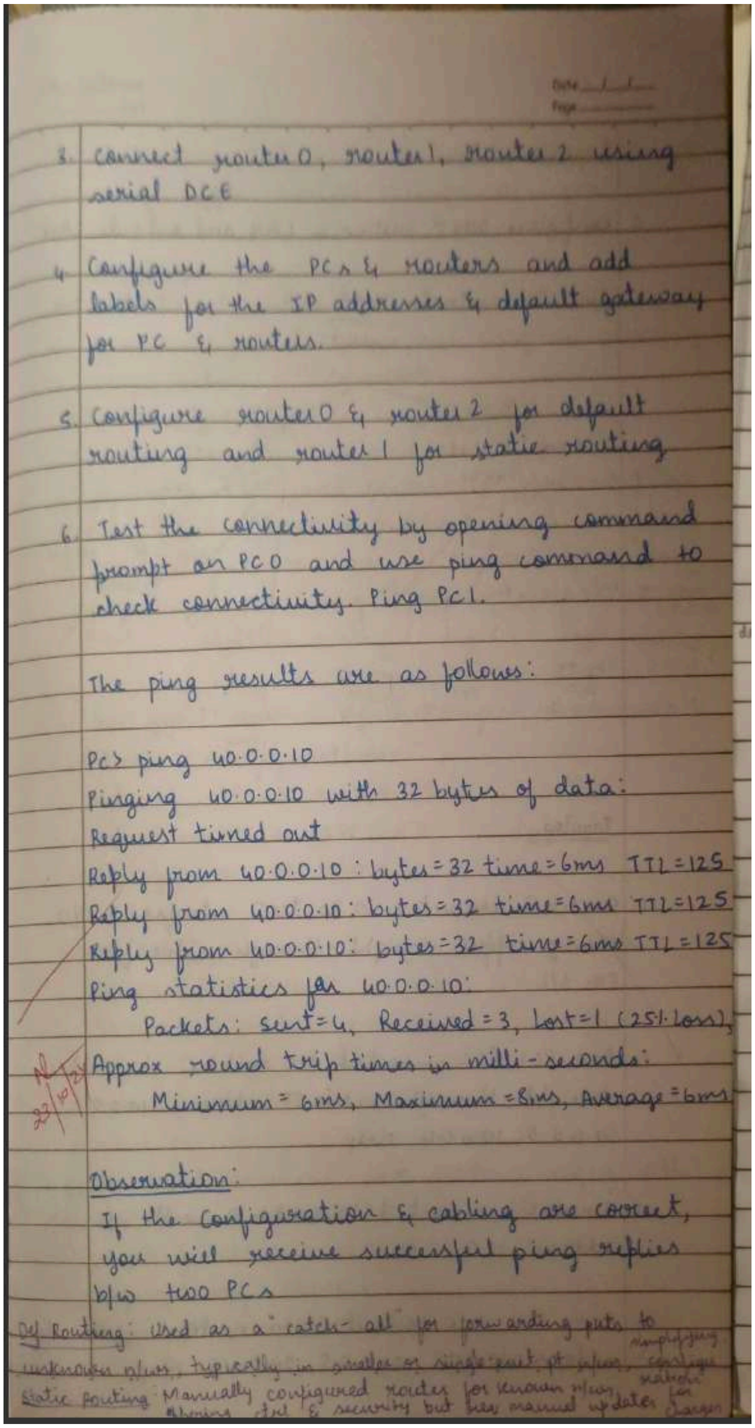
→ Default Routing of Router 2

```
Router (config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

Router # show ip route

C  30.0.0.0/8 is directly connected, Serial 3/0
C  40.0.0.0/8 is directly connected, FastEthernet 0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1.
```

Procedure:

1. Open Cisco Packet Tracer and drag the following components onto workspace:
   Router: Place 3 routers in the middle
   PC: Place two PCs are the below router 0 & router 2.

2. Use cross over cables to connect PC0 & router0 and also PC1 & router 2.



Date ___/___/___
Page ______

3. Connect router0, router1, router2 using serial DCE

4. Configure the PCs & routers and add labels for the IP addresses & default gateway for PC & routers.

5. Configure router0 & router2 for default routing and router1 for static routing

6. Test the connectivity by opening command prompt on PC0 and use ping command to check connectivity. Ping PC1.

The ping results are as follows:

```
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=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Ping statistics for 40.0.0.10:
    Packets: Sent=4, Received=3, Lost=1 (25% Loss),
Approx round trip times in milli-seconds:
    Minimum=6ms, Maximum=8ms, Average=6ms
```

22/10/24

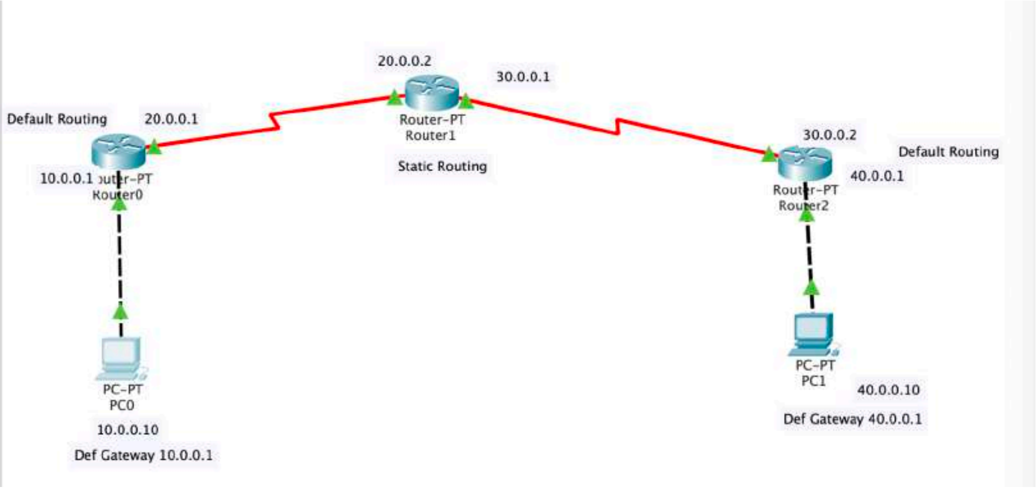
Observation:

If the configuration & cabling are correct, you will receive successful ping replies b/w two PCs

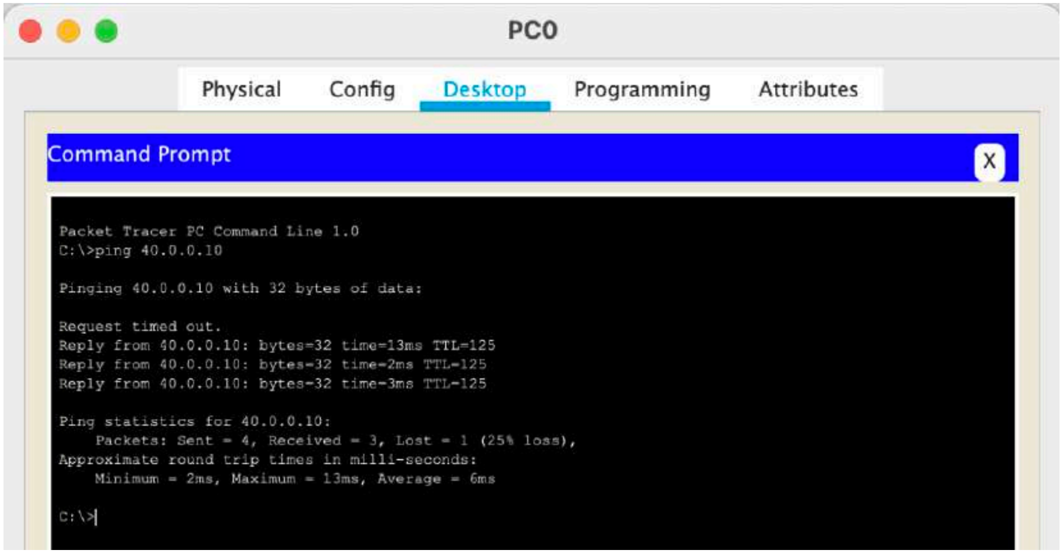
Def Routing: Used as a "catch-all" for forwarding pkts to unknown n/ws, typically in smaller or single exit pt n/ws, simplifying configu

Static Routing: Manually configured router for known n/ws, offering ctrl & security but req manual updates for n/w changes

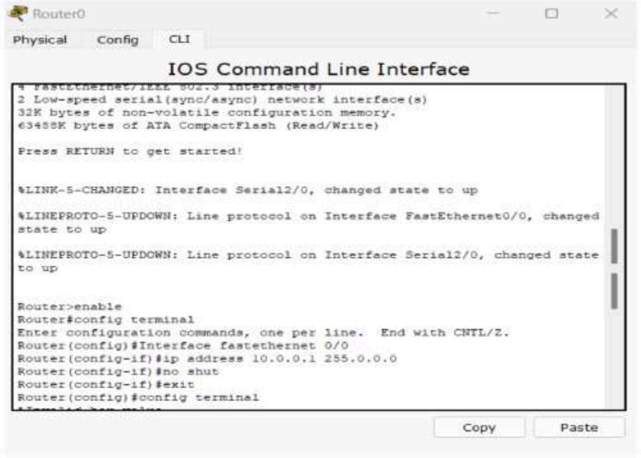
Screenshot of Topology:



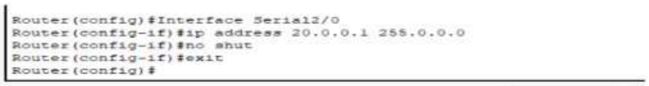
Screenshot of Output:



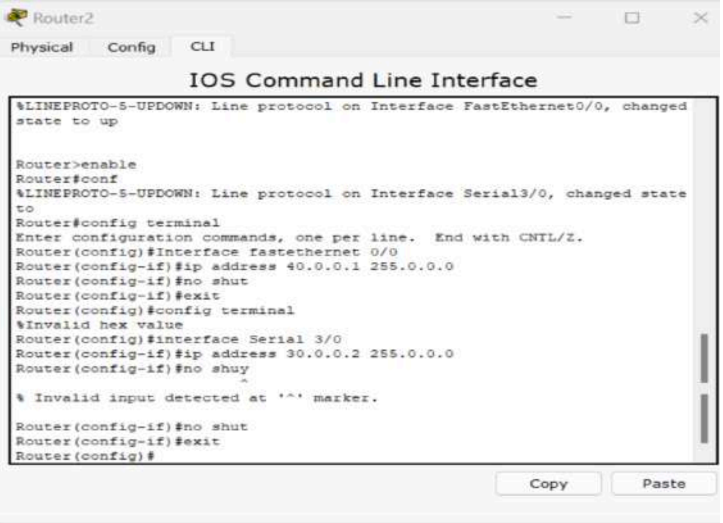
```
Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=13ms TTL=125
Reply from 40.0.0.10: bytes=32 time=2ms TTL=125
Reply from 40.0.0.10: bytes=32 time=3ms 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 = 13ms, Average = 6ms

C:\>
```



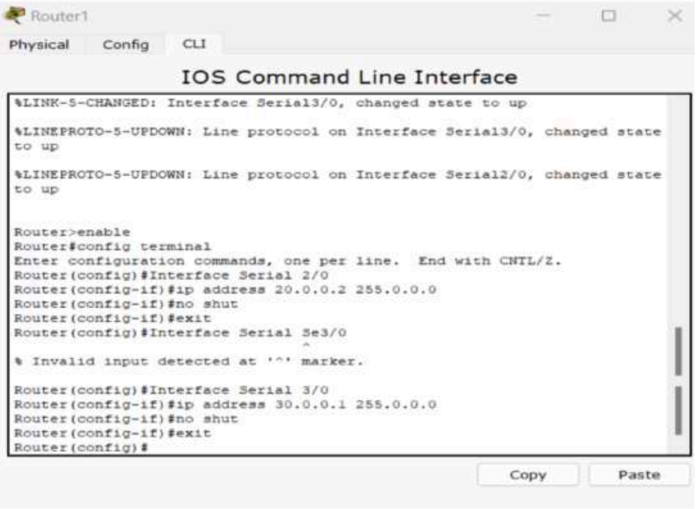
```
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)

Press RETURN to get started!


%LINK-5-CHANGED: Interface Serial2/0, changed state to up

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to up


Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#Interface fastethernet 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#config terminal
```



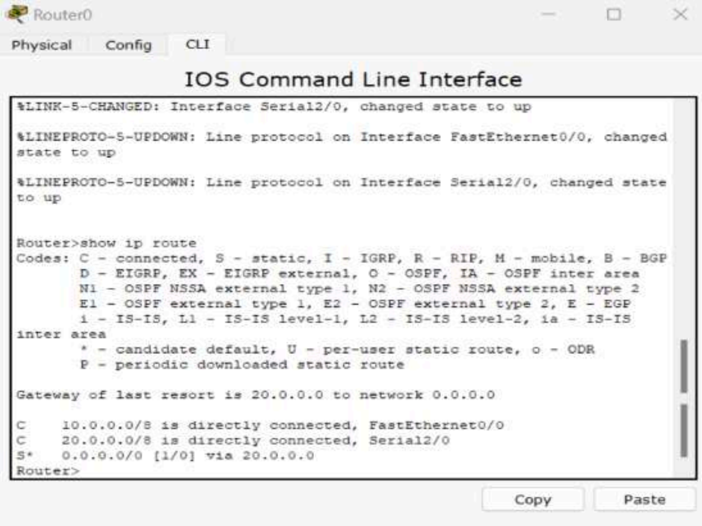
```
Router(config)#Interface Serial2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
```



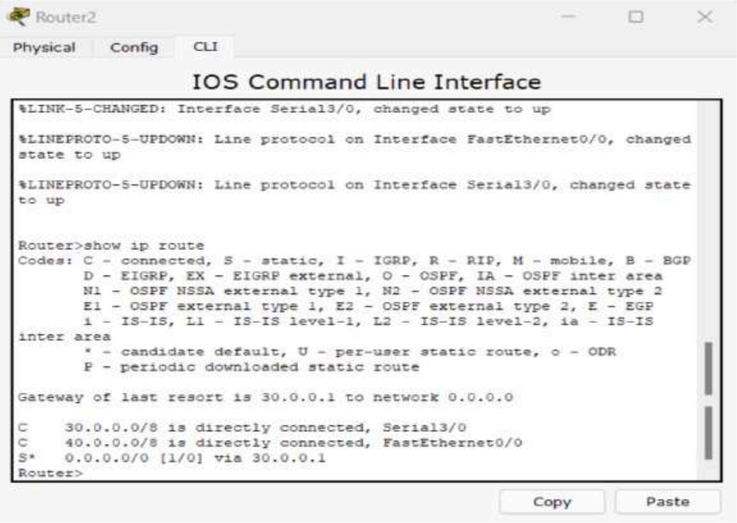
```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up


Router>enable
Router#conf
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#Interface fastethernet 0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#config terminal
%Invalid hex value
Router(config)#interface Serial 3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shuy
                        ^
% Invalid input detected at '^' marker.

Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
```



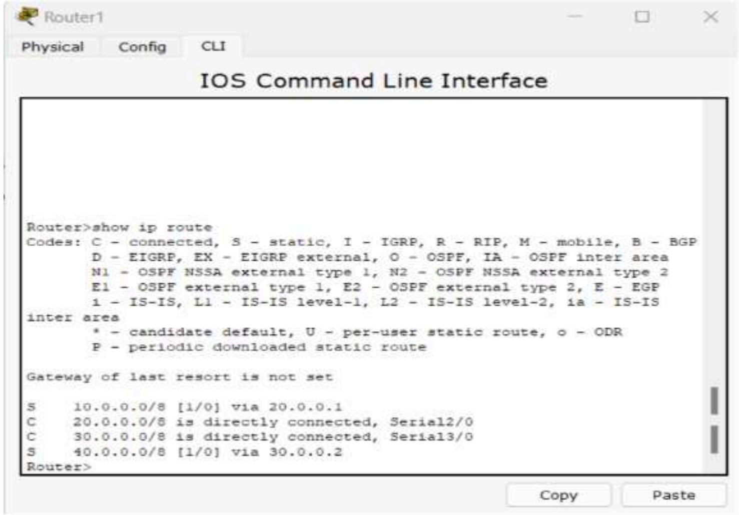
```
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to up


Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#Interface Serial 2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#Interface Serial Se3/0
                                ^
% Invalid input detected at '^' marker.

Router(config)#Interface Serial 3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
```



```
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to up


Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 20.0.0.0 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.0
Router>
```



```
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to up


Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 30.0.0.1 to network 0.0.0.0

C    30.0.0.0/8 is directly connected, Serial3/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 30.0.0.1
Router>
```

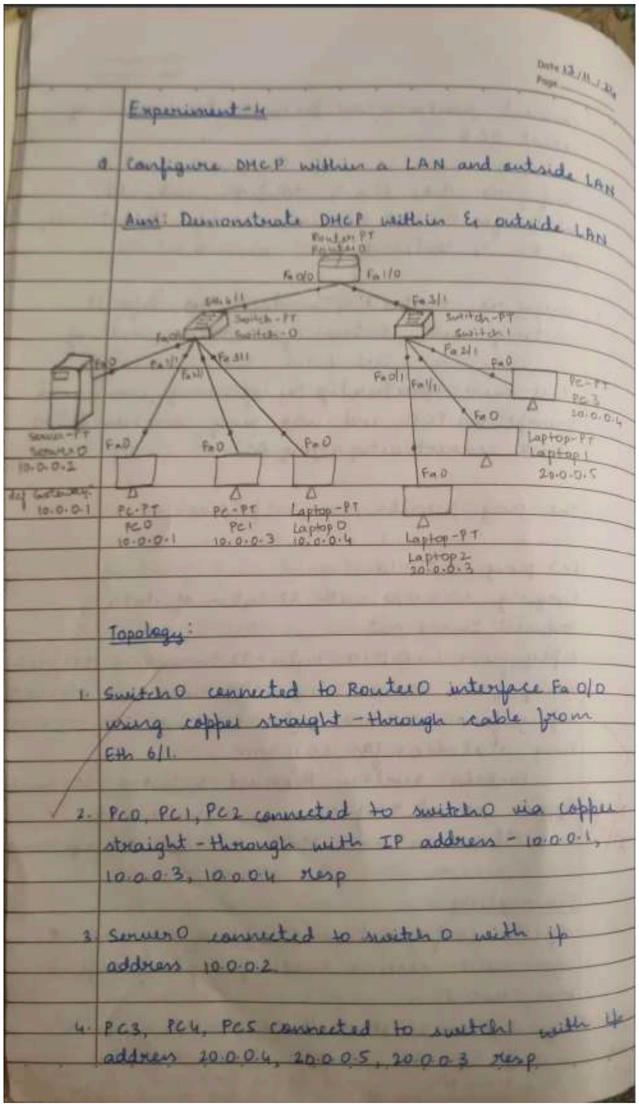


```
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

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

Program 4

Q4: Configure IP address of the host using DHCP server within and outside a LAN.

Observation:



Date 12/11/24
Page

Experiment-4

Q Configure DHCP within a LAN and outside LAN

Aim: Demonstrate DHCP within & outside LAN

Router-PT
Router0
Fa0/0
Fa1/0
Eth6/1
Switch-PT
Switch-0
Fa3/1
Switch-PT
Switch1
Fa0/1
Fa0
Fa1/1
Fa2/1
Fa3/1
Fa2/1
Fa0
Fa0/1
Fa1/1
Fa0
Server-PT
Server0
10.0.0.2
def Gateway
10.0.0.1
Fa0
Fa0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.3
Laptop-PT
Laptop0
10.0.0.4
Fa0
Laptop-PT
Laptop2
20.0.0.3
Laptop-PT
Laptop1
20.0.0.5
PC-PT
PC3
20.0.0.4

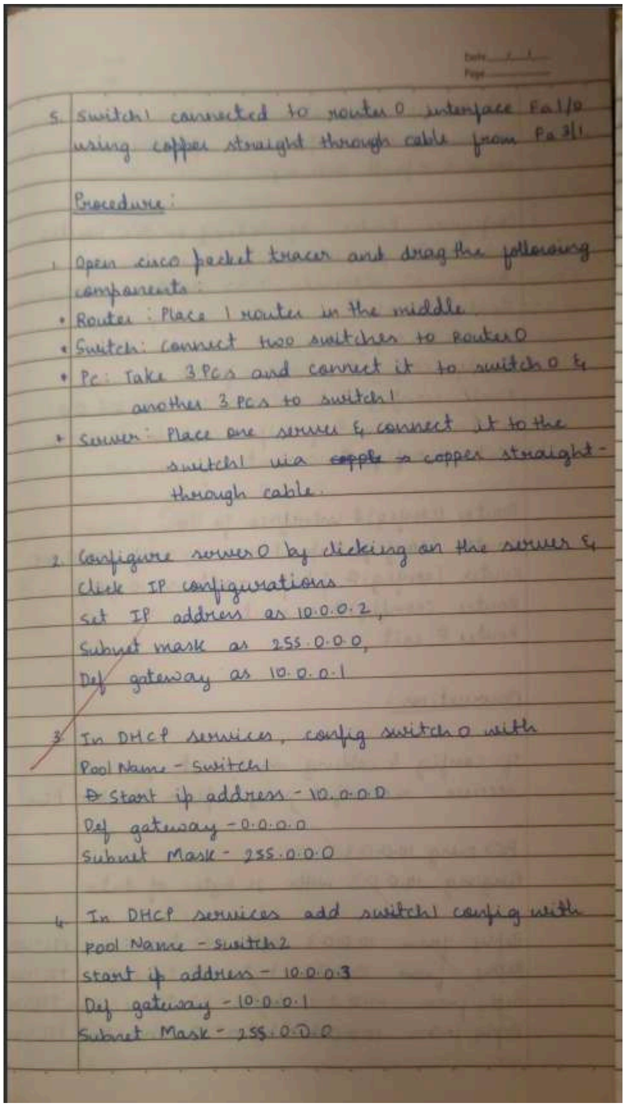
Topology:

1. Switch0 connected to Router0 interface Fa0/0 using copper straight-through cable from Eth 6/1.

2. PC0, PC1, PC2 connected to switch0 via copper straight-through with IP address - 10.0.0.1, 10.0.0.3, 10.0.0.4 resp

3. Server0 connected to switch 0 with ip address 10.0.0.2.

4. PC3, PC4, PC5 connected to switch1 with ip address 20.0.0.4, 20.0.0.5, 20.0.0.3 resp.



Date / /
Page

5. switch1 connected to router0 interface Fa1/0 using copper straight through cable from Fa3/1.

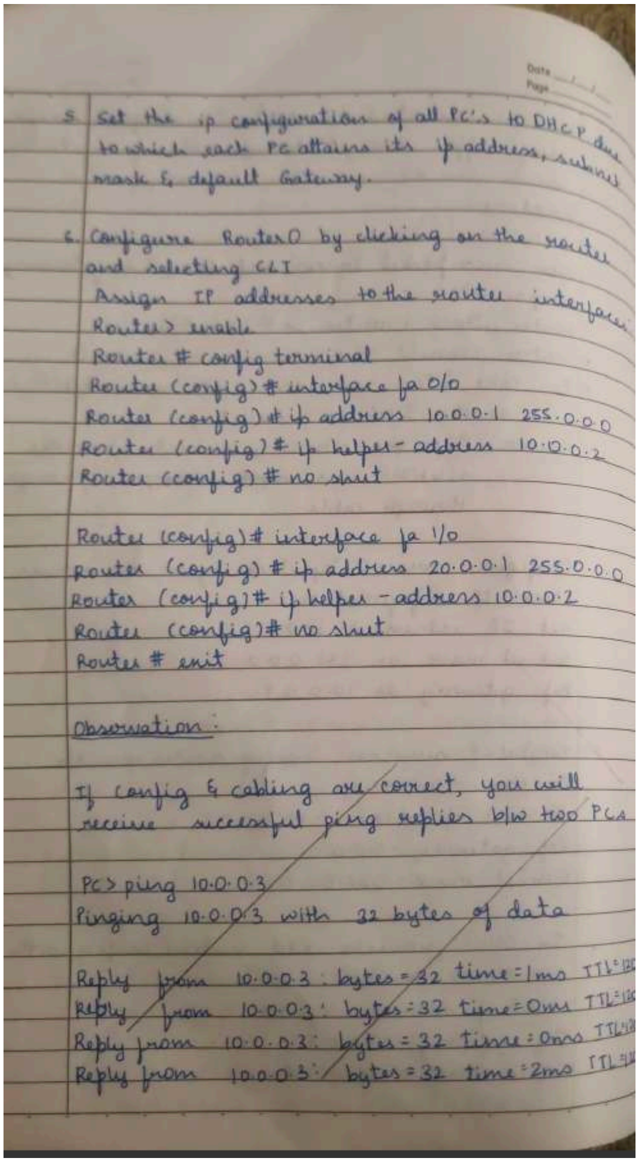
Procedure:

1. Open cisco packet tracer and drag the following components:
- Router: Place 1 router in the middle
- Switch: connect two switches to Router0
- Pc: Take 3 PCs and connect it to switch0 & another 3 PCs to switch1
- Server: Place one server & connect it to the switch1 via ~~copple~~ → copper straight-through cable.

2. Configure server0 by clicking on the server & click IP configurations
set IP address as 10.0.0.2,
Subnet mask as 255.0.0.0,
Def gateway as 10.0.0.1

3. In DHCP services, config switch0 with
Pool Name - Switch1
~~D~~ Start ip address - 10.0.0.0
Def gateway - 0.0.0.0
Subnet Mask - 255.0.0.0

4. In DHCP services add switch1 config with
Pool Name - switch2
start ip address - 10.0.0.3
Def gateway - 10.0.0.1
Subnet Mask - 255.0.0.0



Date / /
Page

5. Set the ip configuration of all PC's to DHCP due to which each PC attains its ip address, subnet mask & default Gateway.

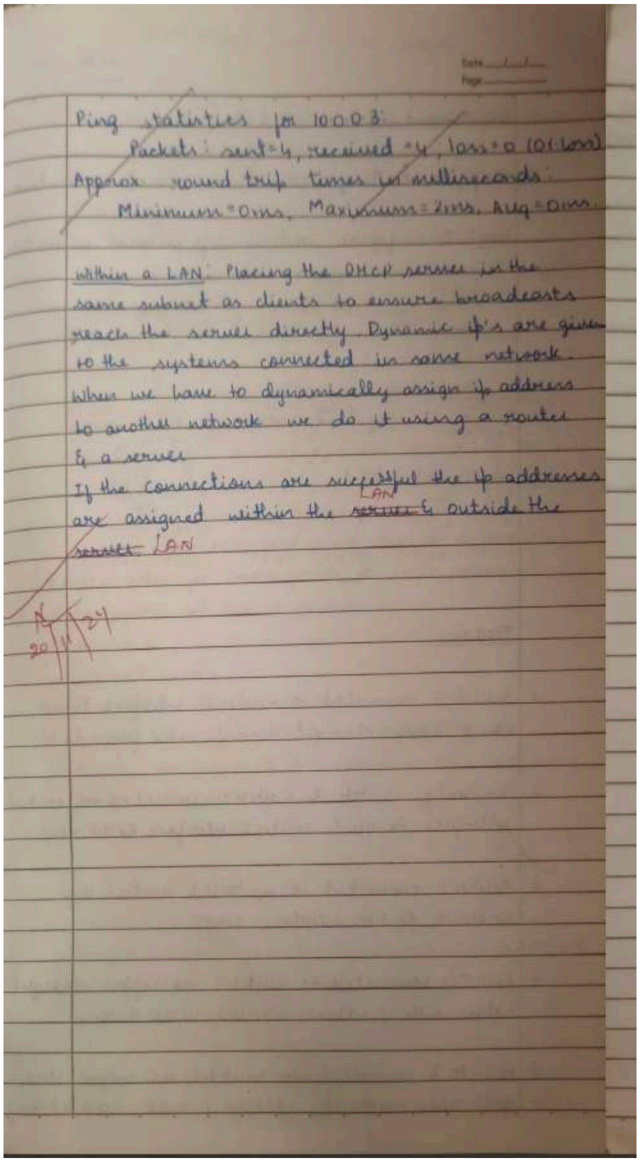
6. Configure Router0 by clicking on the router and selecting CLI
Assign IP addresses to the router interfaces

```
Router> enable
Router # config terminal
Router (config) # interface fa 0/0
Router (config) # ip address 10.0.0.1 255.0.0.0
Router (config) # ip helper-address 10.0.0.2
Router (config) # no shut

Router (config)# interface fa 1/0
Router (config) # ip address 20.0.0.1 255.0.0.0
Router (config)# ip helper -address 10.0.0.2
Router (config)# no shut
Router # exit
```

Observation:

If config & cabling are correct, you will receive successful ping replies b/w two PCs

```
PC> ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3: bytes=32 time=1ms TTL=12
Reply from 10.0.0.3: bytes=32 time=0ms TTL=12
Reply from 10.0.0.3: bytes=32 time=0ms TTL=12
Reply from 10.0.0.3: bytes=32 time=2ms TTL=12
```



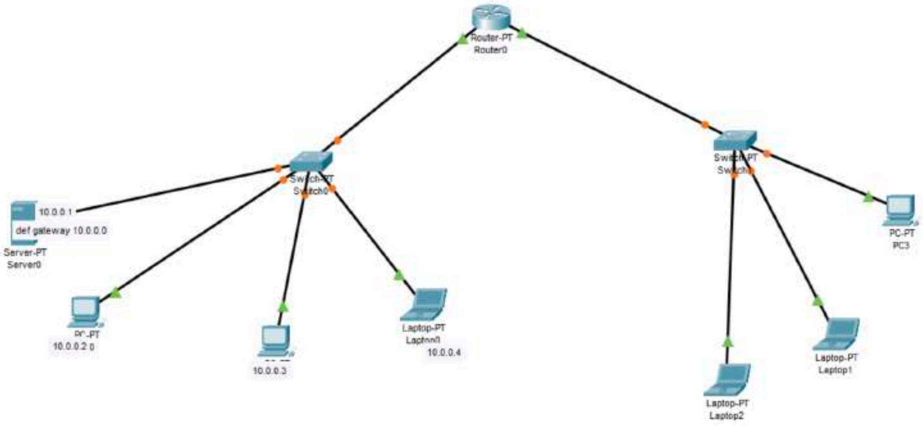
Date / /
Page

```
Ping statistics for 10.0.0.3:
    Packets: sent=4, received =4, loss=0 (0% loss)
Approx round trip times in milliseconds:
    Minimum=0ms, Maximum=2ms, Avg=0ms
```

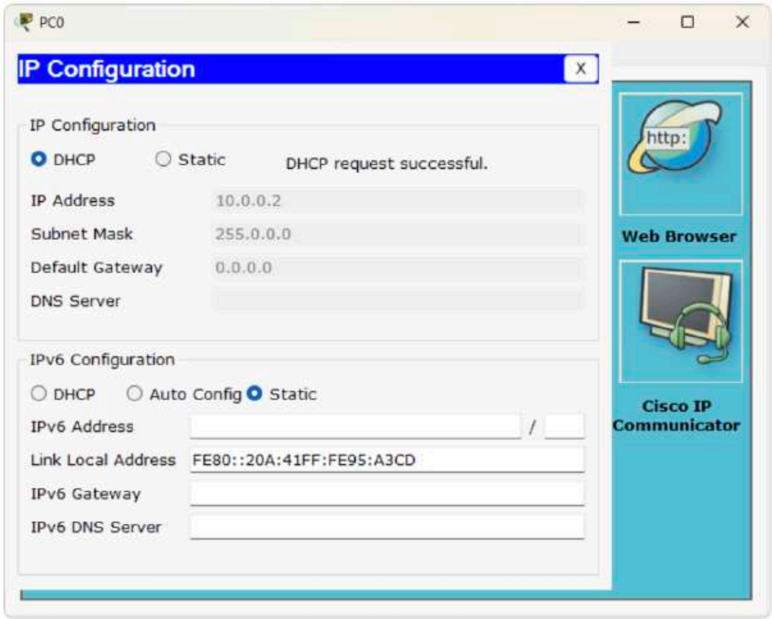
Within a LAN: Placing the DHCP server in the same subnet as clients to ensure broadcasts reach the server directly. Dynamic ip's are given to the systems connected in same network.
When we have to dynamically assign ip address to another network we do it using a router & a server
If the connections are successful the ip addresses are assigned within the ~~server~~ LAN & outside the ~~server~~ LAN

20/11/24

Screenshot of Topology:



Screenshot of Output:



IP Configuration

IP Configuration

● DHCP ○ Static DHCP request successful.

IP Address 10.0.0.2

Subnet Mask 255.0.0.0

Default Gateway 0.0.0.0

DNS Server

IPv6 Configuration

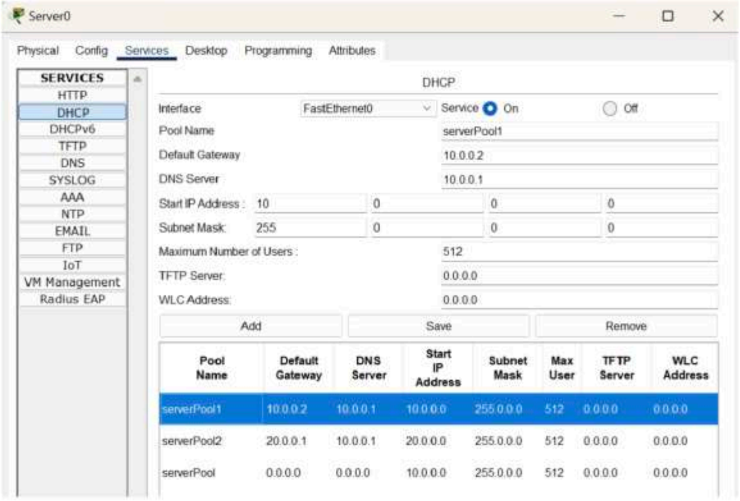
○ DHCP ○ Auto Config ● Static

IPv6 Address /

Link Local Address FE80::20A:41FF:FE95:A3CD

IPv6 Gateway

IPv6 DNS Server



Interface: FastEthernet0 Service ● On ○ Off

Pool Name: serverPool1

Default Gateway: 10.0.0.2

DNS Server: 10.0.0.1

Start IP Address : 10 0 0 0

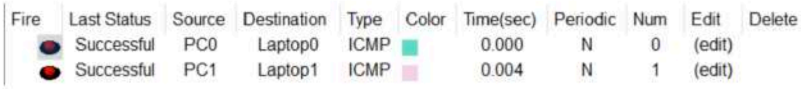
Subnet Mask: 255 0 0 0

Maximum Number of Users : 512

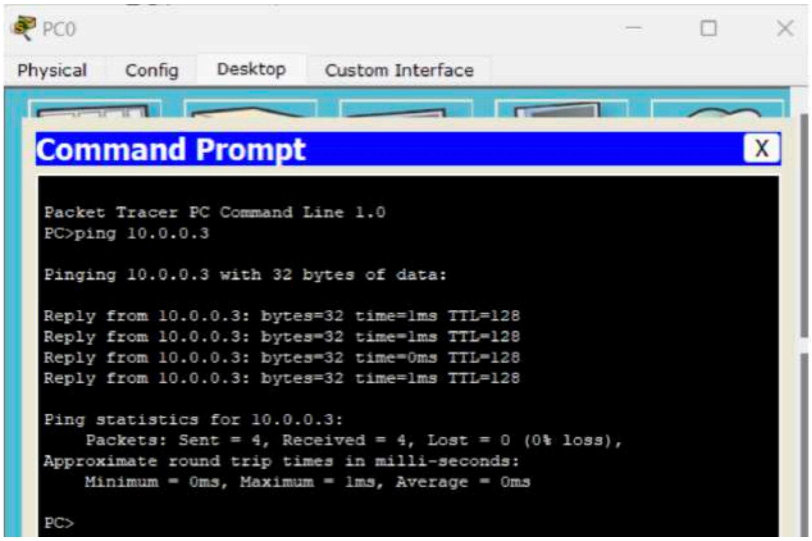
TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool1	10.0.0.2	10.0.0.1	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool2	20.0.0.1	10.0.0.1	20.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Laptop0	ICMP	🟩	0.000	N	0	(edit)	
●	Successful	PC1	Laptop1	ICMP	🟪	0.004	N	1	(edit)	

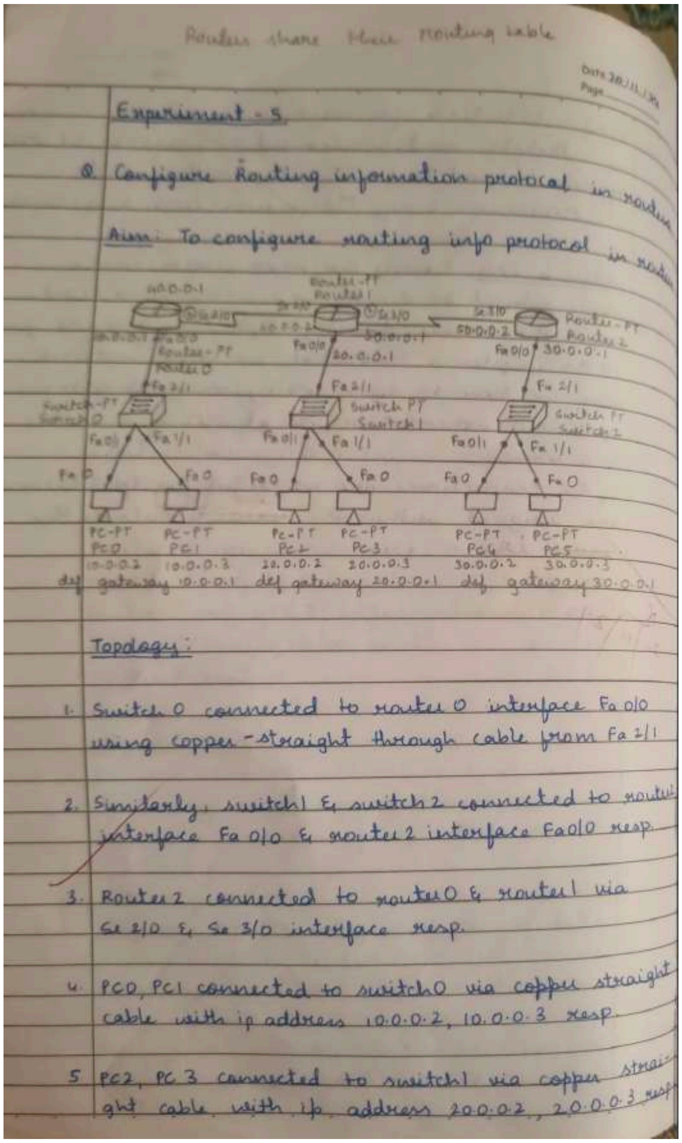


```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

Program 5:

Q5: Configure Routing Information Protocol (RIP) routing Protocol in Routers

Observation:



Routers share their routing table

Date 20/11/24
Page

Experiment - 5

Q. Configure Routing information protocol in routers

Aim: To configure routing info protocol in routers

40.0.0.1
Se2/0
Router-PT
Router1
Se2/0
Se3/0
10.0.0.1
Fa0/0
Router-PT
Router0
Fa2/1
Switch-PT
Switch0
Fa0/1
Fa1/1
Fa0
Fa0
Fa0/0
20.0.0.1
Fa2/1
Switch PT
Switch1
Fa0/1
Fa1/1
Fa0
Fa0
30.0.0.1
Se3/0
50.0.0.2
Router-PT
Router2
Fa0/0
30.0.0.1
Fa2/1
Switch PT
Switch2
Fa0/1
Fa1/1
Fa0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC1
10.0.0.3
PC-PT
PC2
20.0.0.2
PC-PT
PC3
20.0.0.3
PC-PT
PC4
30.0.0.2
PC-PT
PC5
30.0.0.3

def gateway 10.0.0.1 def gateway 20.0.0.1 def gateway 30.0.0.1

Topology:

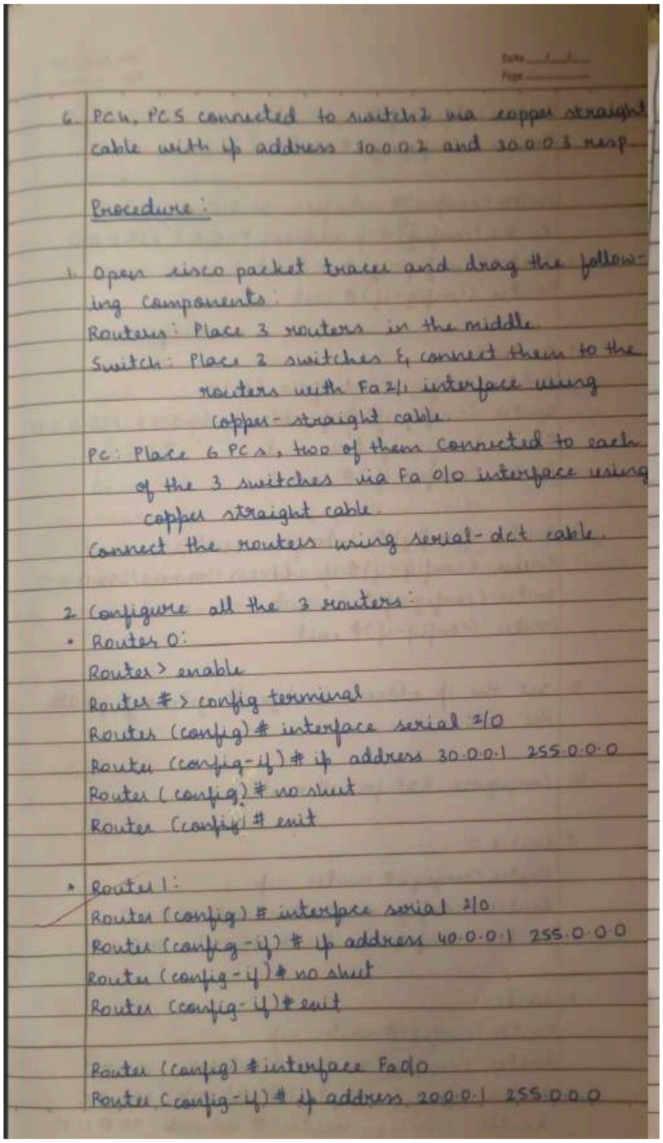
1. Switch 0 connected to router 0 interface Fa 0/0 using copper-straight through cable from Fa 2/1

2. Similarly, switch1 & switch 2 connected to router1 interface Fa 0/0 & router 2 interface Fa0/0 resp.

3. Router 2 connected to router0 & router1 via Se 2/0 & Se 3/0 interface resp.

4. PC0, PC1 connected to switch0 via copper straight cable with ip address 10.0.0.2, 10.0.0.3 resp.

5. PC2, PC 3 connected to switch1 via copper straight cable with ip address 20.0.0.2, 20.0.0.3 resp.



Date / /
Page

6. PC4, PC5 connected to switch2 via copper straight cable with ip address 30.0.0.2 and 30.0.0.3 resp.

Procedure:

1. Open cisco packet tracer and drag the following components:

Routers: Place 3 routers in the middle.

Switch: Place 3 switches & connect them to the routers with Fa2/1 interface using copper- straight cable.

PC: Place 6 PCs, two of them connected to each of the 3 switches via Fa 0/0 interface using copper straight cable.

Connect the routers using serial-dct cable.

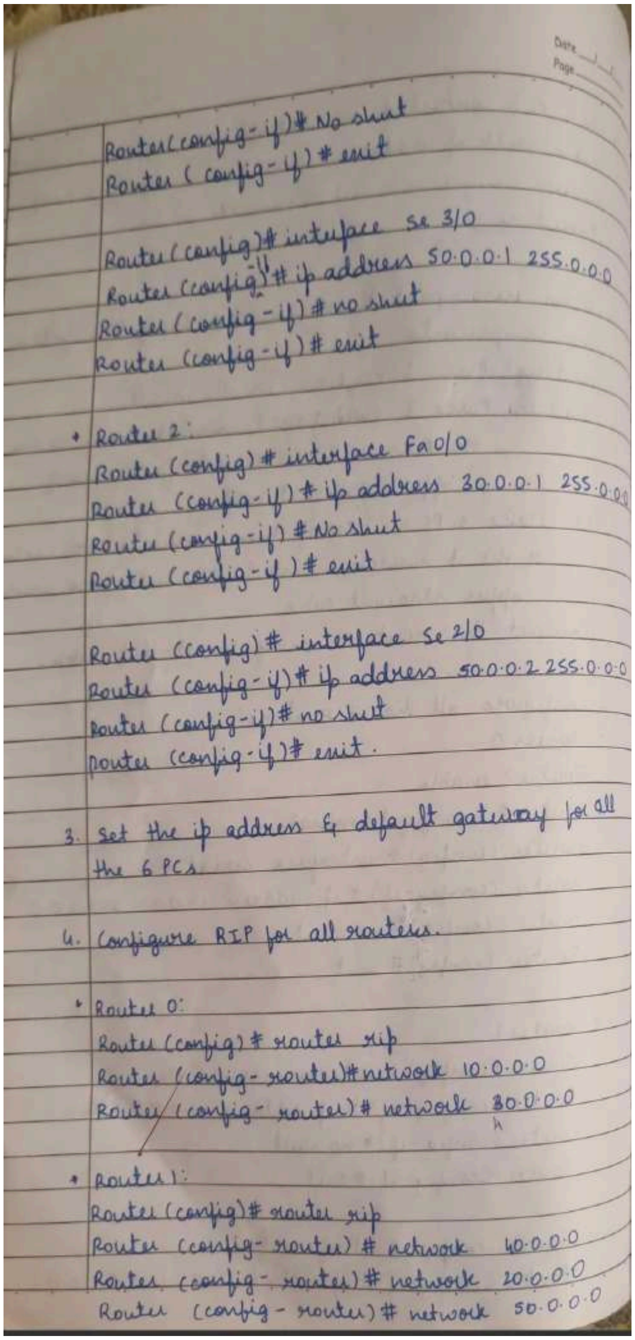
2 Configure all the 3 routers:

- Router 0:

```
Router> enable
Router #> config terminal
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router ( config) # no shut
Router (config) # exit
```

- Router 1:

```
Router (config) # interface serial 2/0
Router (config-if) # ip address 40.0.0.1 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit

Router (config) # interface Fa0/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0
```



```
Router(config-if)# No shut
Router ( config-if) # exit

Router(config)# interface Se 3/0
Router (config-if)# ip address 50.0.0.1 255.0.0.0
Router ( config - if) # no shut
Router (config - if) # exit
```

- Router 2:

```
Router (config) # interface Fa 0/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # No shut
Router (config - if ) # exit

Router (config) # interface Se 2/0
Router (config-if)# ip address 50.0.0.2 255.0.0.0
Router (config-if)# no shut
Router (config-if)# exit.
```

3. Set the ip address & default gateway for all the 6 PCs

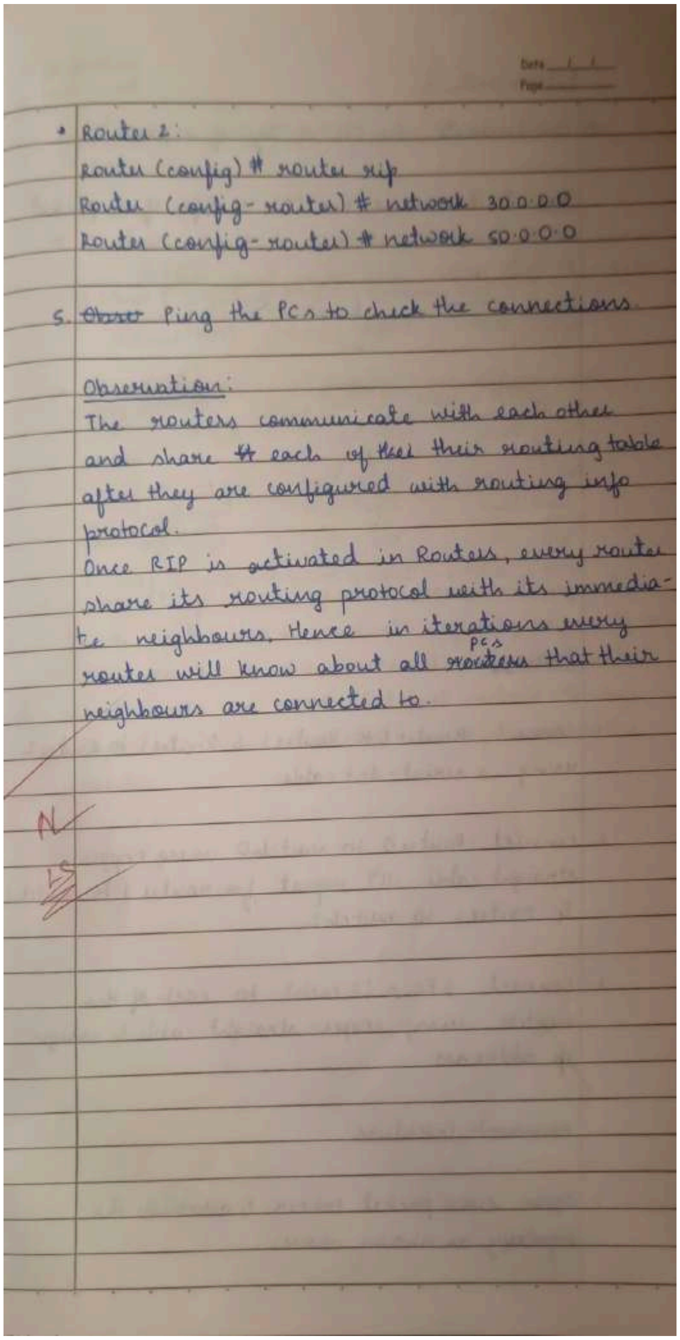
4. Configure RIP for all routers.

- Router 0:

```
Router (config) # router rip
Router (config- router)# network 10.0.0.0
Router (config- router) # network 30.0.0.0
```

- Router 1:

```
Router (config)# router rip
Router (config- router) # network 40.0.0.0
Router (config- router) # network 20.0.0.0
Router (config - router) # network 50.0.0.0
```



- Router 2:

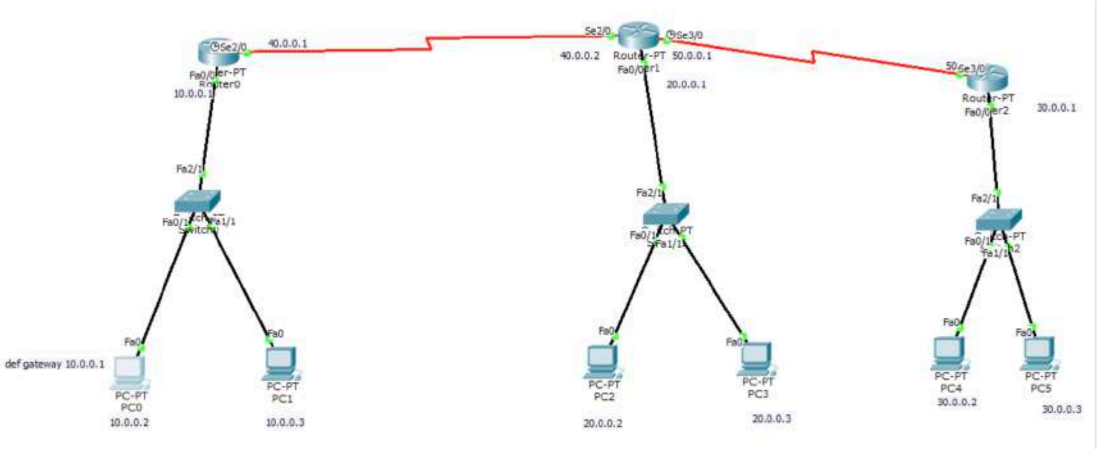
```
Router (config) # router rip
Router (config- router) # network 30.0.0.0
Router (config- router) # network 50.0.0.0
```

5. ~~Obsrv~~ Ping the PCs to check the connections.

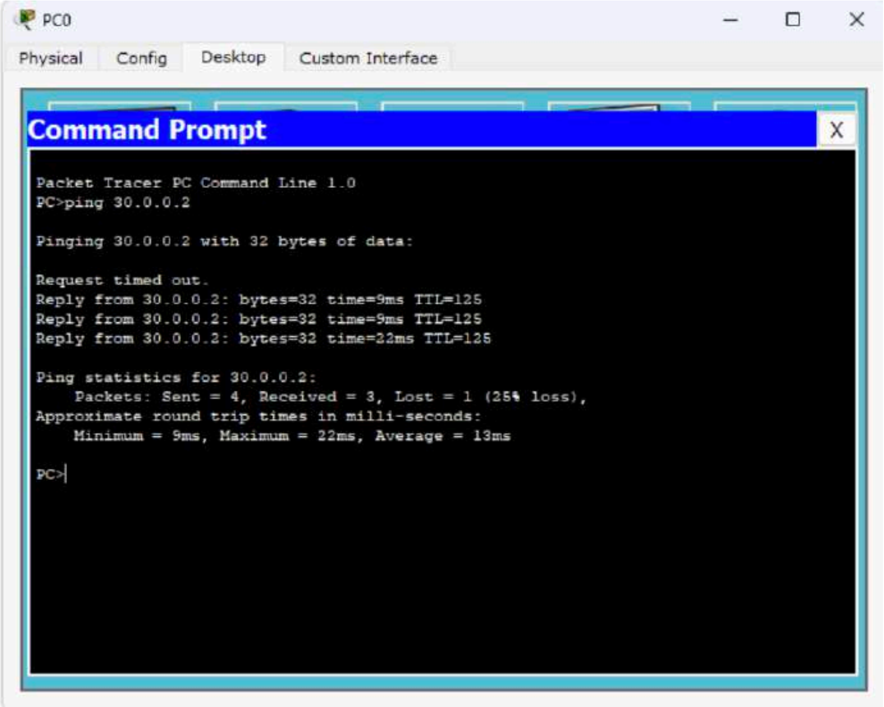
Observation:

The routers communicate with each other and share ~~th~~ each of ~~thei~~ their routing table after they are configured with routing info protocol.
Once RIP is activated in Routers, every router share its routing protocol with its immediate neighbours. Hence in iterations every router will know about all ~~routers~~ PCs that their neighbours are connected to.

Screenshot of Topology:

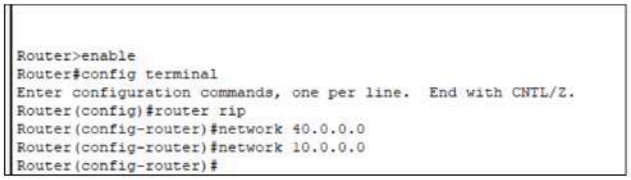


Screenshot of Output:



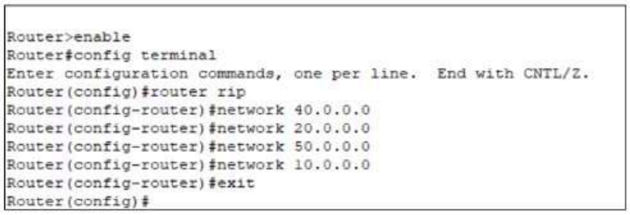
```
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=9ms TTL=125
Reply from 30.0.0.2: bytes=32 time=9ms TTL=125
Reply from 30.0.0.2: bytes=32 time=22ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 22ms, Average = 13ms

PC>
```

Router 0:



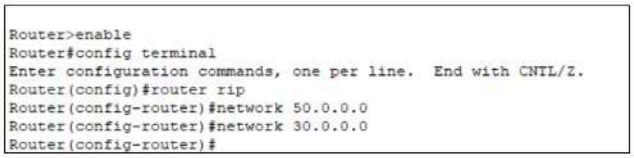
```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 10.0.0.0
Router(config-router)#
```

Router 1:

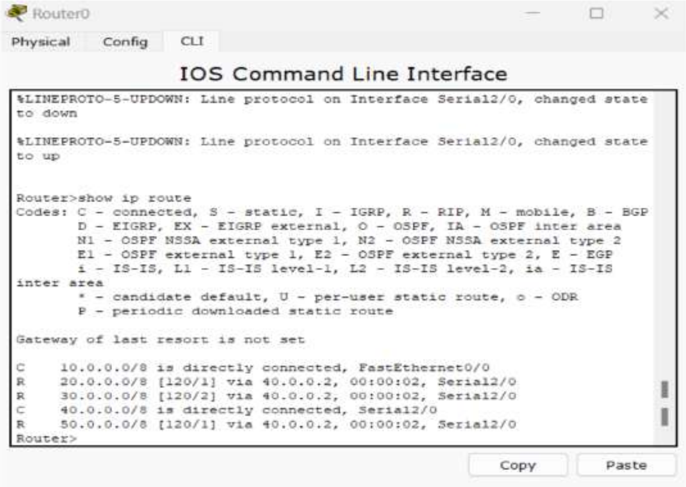


```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 10.0.0.0
Router(config-router)#exit
Router(config)#
```

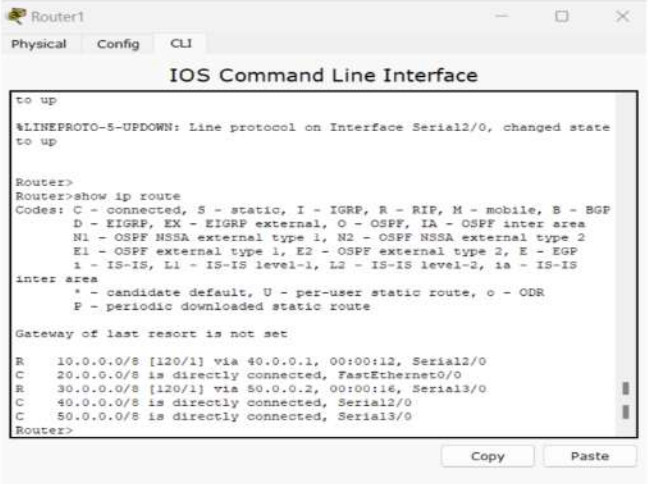
Router 2:



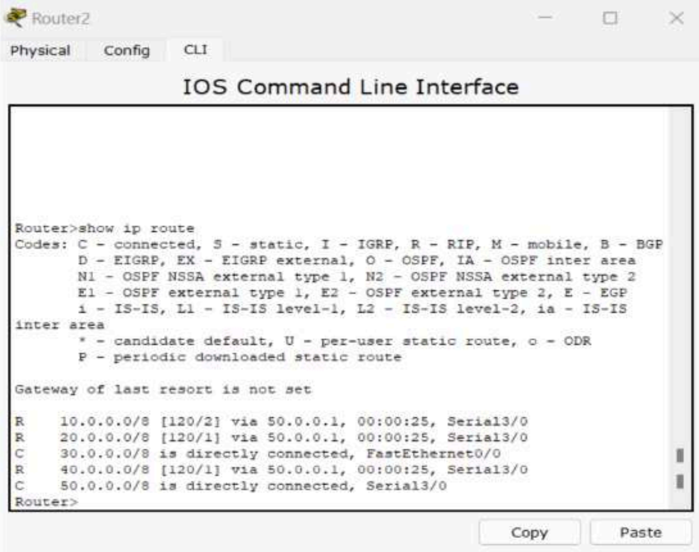
```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#
```



```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to up


Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
R    20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:02, Serial2/0
R    30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:02, Serial2/0
C    40.0.0.0/8 is directly connected, Serial2/0
R    50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:02, Serial2/0
Router>
```



```
to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to up


Router>
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, 00:00:12, 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, 00:00:16, Serial3/0
C    40.0.0.0/8 is directly connected, Serial2/0
C    50.0.0.0/8 is directly connected, Serial3/0
Router>
```

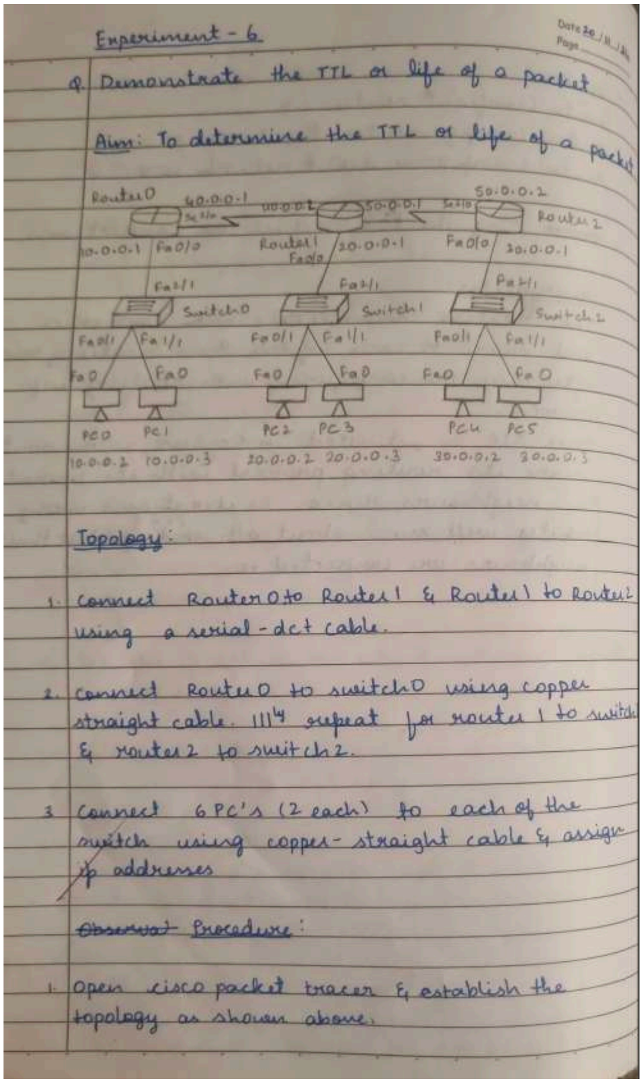


```
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - 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, 00:00:25, Serial3/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:25, 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, 00:00:25, Serial3/0
C    50.0.0.0/8 is directly connected, Serial3/0
Router>
```

Program 6

Q6: Demonstrate the Time To Live (TTL) or life of a packet.

Observation:



Experiment - 6

Date 20/11/24
Page

Q. Demonstrate the TTL or life of a packet

Aim: To determine the TTL or life of a packet

Router0
40.0.0.1
Se 2/0
40.0.0.2
50.0.0.1
Se 3/0
50.0.0.2
Router2
10.0.0.1
Fa0/0
Router1
Fa0/0
20.0.0.1
Fa0/0
30.0.0.1
Fa1/1
Fa1/1
Fa1/1
Switch0
Switch1
Switch2
Fa0/1
Fa1/1
Fa0/1
Fa1/1
Fa0/1
Fa1/1
Fa0
Fa0
Fa0
Fa0
Fa0
Fa0
PC0
PC1
PC2
PC3
PC4
PC5
10.0.0.2
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3

Topology:

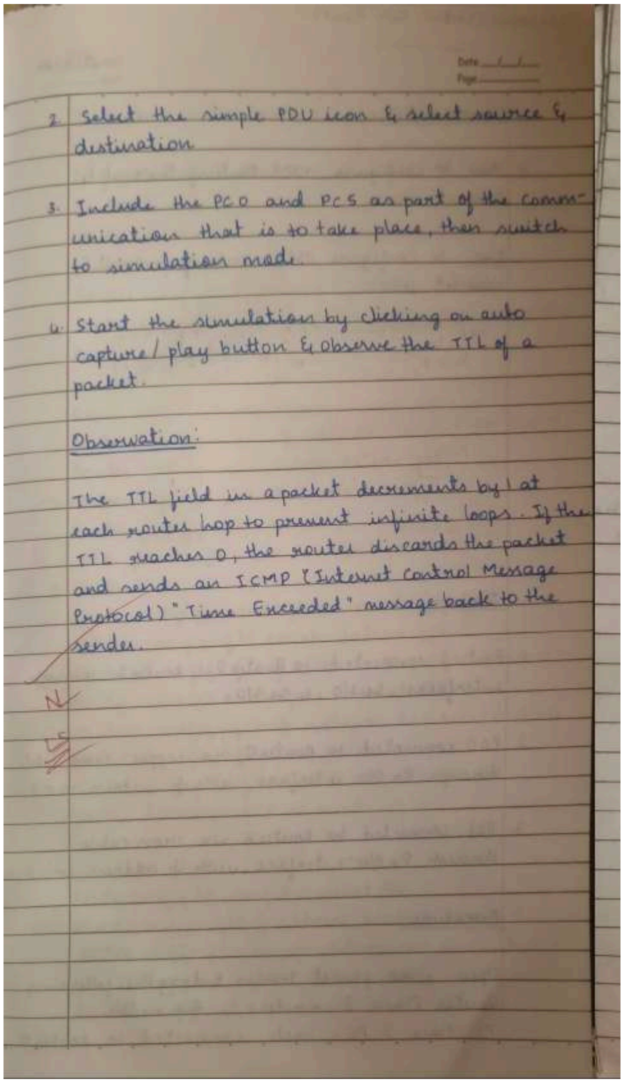
1. Connect Router0 to Router1 & Router1 to Router2 using a serial-dct cable.

2. Connect Router0 to switch0 using copper straight cable. IIIly repeat for router 1 to switch & router2 to switch2.

3. Connect 6 PC's (2 each) to each of the switch using copper-straight cable & assign ip addresses

~~Observat~~ Procedure:

1. Open cisco packet tracer & establish the topology as shown above.



Date __/__/__
Page ____

2. Select the simple PDU icon & select source & destination

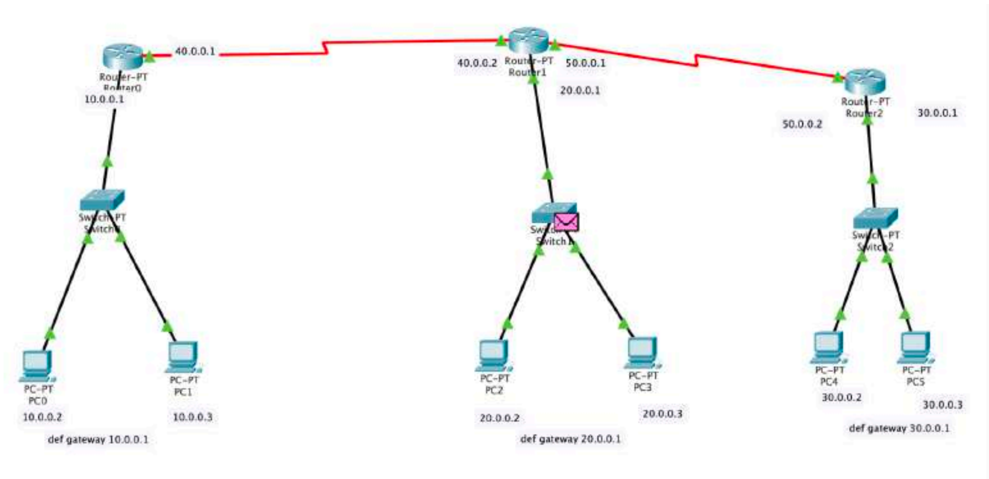
3. Include the PC0 and PC5 as part of the communication that is to take place, then switch to simulation mode.

4. Start the simulation by clicking on auto capture/play button & observe the TTL of a packet.

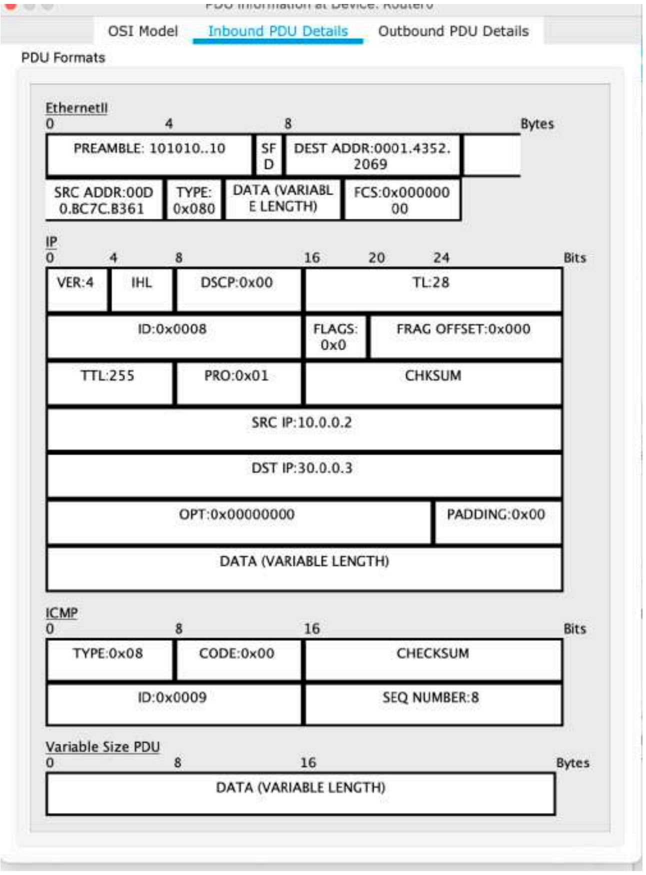
Observation:

The TTL field in a packet decrements by 1 at each router hop to prevent infinite loops. If the TTL reaches 0, the router discards the packet and sends an ICMP (Internet Control Message Protocol) "Time Exceeded" message back to the sender.

Screenshot of Topology:



Screenshot of Output:



PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

EthernetII

PREAMBLE: 101010..10 | SFD | DEST ADDR:0001.4352.2069

SRC ADDR:00D0.BC7C.B361 | TYPE: 0x080 | DATA (VARIABLE LENGTH) | FCS:0x00000000

IP

VER:4 | IHL | DSCP:0x00 | TL:28

ID:0x0008 | FLAGS: 0x0 | FRAG OFFSET:0x000

TTL:255 | PRO:0x01 | CHKSUM

SRC IP:10.0.0.2

DST IP:30.0.0.3

OPT:0x00000000 | PADDING:0x00

DATA (VARIABLE LENGTH)

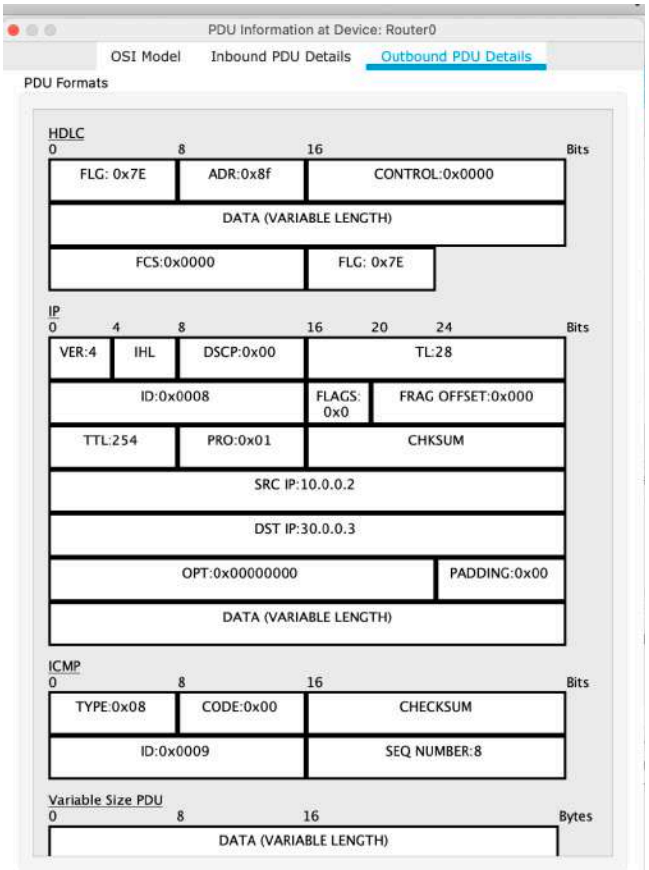
ICMP

TYPE:0x08 | CODE:0x00 | CHECKSUM

ID:0x0009 | SEQ NUMBER:8

Variable Size PDU

DATA (VARIABLE LENGTH)



PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

HDLC

FLG: 0x7E | ADR:0x8f | CONTROL:0x0000

DATA (VARIABLE LENGTH)

FCS:0x0000 | FLG: 0x7E

IP

VER:4 | IHL | DSCP:0x00 | TL:28

ID:0x0008 | FLAGS: 0x0 | FRAG OFFSET:0x000

TTL:254 | PRO:0x01 | CHKSUM

SRC IP:10.0.0.2

DST IP:30.0.0.3

OPT:0x00000000 | PADDING:0x00

DATA (VARIABLE LENGTH)

ICMP

TYPE:0x08 | CODE:0x00 | CHECKSUM

ID:0x0009 | SEQ NUMBER:8

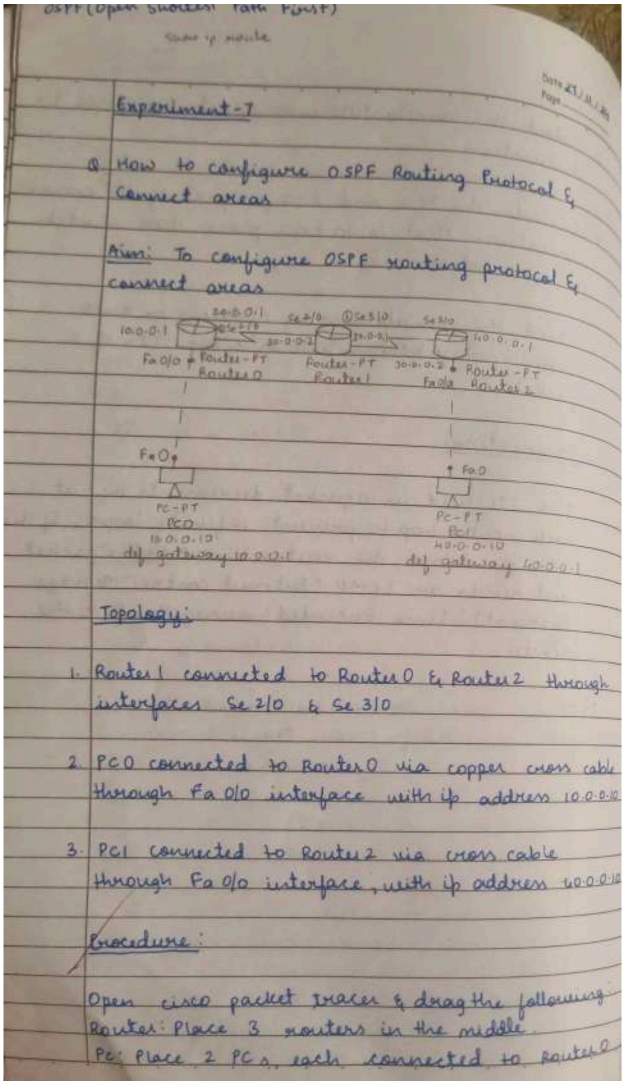
Variable Size PDU

DATA (VARIABLE LENGTH)

Program 7

Q7: To configure Open Shortest Path First (OSPF) routing protocol and connect area

Observation:



OSPF (Open Shortest Path First)

show ip route

Date 27/11/24
Page

Experiment-7

Q How to configure OSPF Routing Protocol & connect areas

Aim: To configure OSPF routing protocol & connect areas

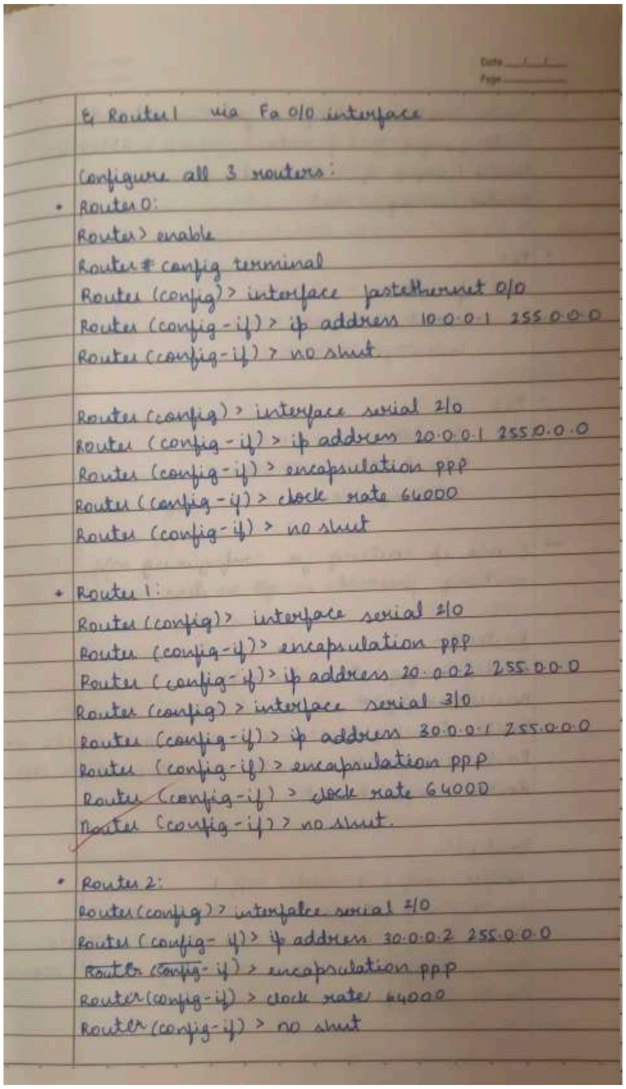
20.0.0.1
Se 2/0
10.0.0.1
20.0.0.2
Se 2/0
Se 3/0
30.0.0.1
Se 3/0
40.0.0.1
Fa 0/0
Router-PT
Router0
Router-PT
Router1
30.0.0.2
Fa 0/0
Router-PT
Router2
Fa0
PC-PT
PC0
10.0.0.10
def gateway 10.0.0.1
Fa0
PC-PT
PC1
40.0.0.10
def gateway 40.0.0.1

Topology:

1. Router1 connected to Router0 & Router2 through interfaces Se 2/0 & Se 3/0
2. PC0 connected to Router0 via copper cross cable through Fa 0/0 interface with ip address 10.0.0.10
3. PC1 connected to Router2 via cross cable through Fa 0/0 interface, with ip address 40.0.0.10

Procedure:

Open cisco packet tracer & drag the following
Router: Place 3 routers in the middle.
PC: Place 2 PCs, each connected to Router0



Date __/__/__
Page ____

& Router1 via Fa 0/0 interface

Configure all 3 routers:

- Router0:

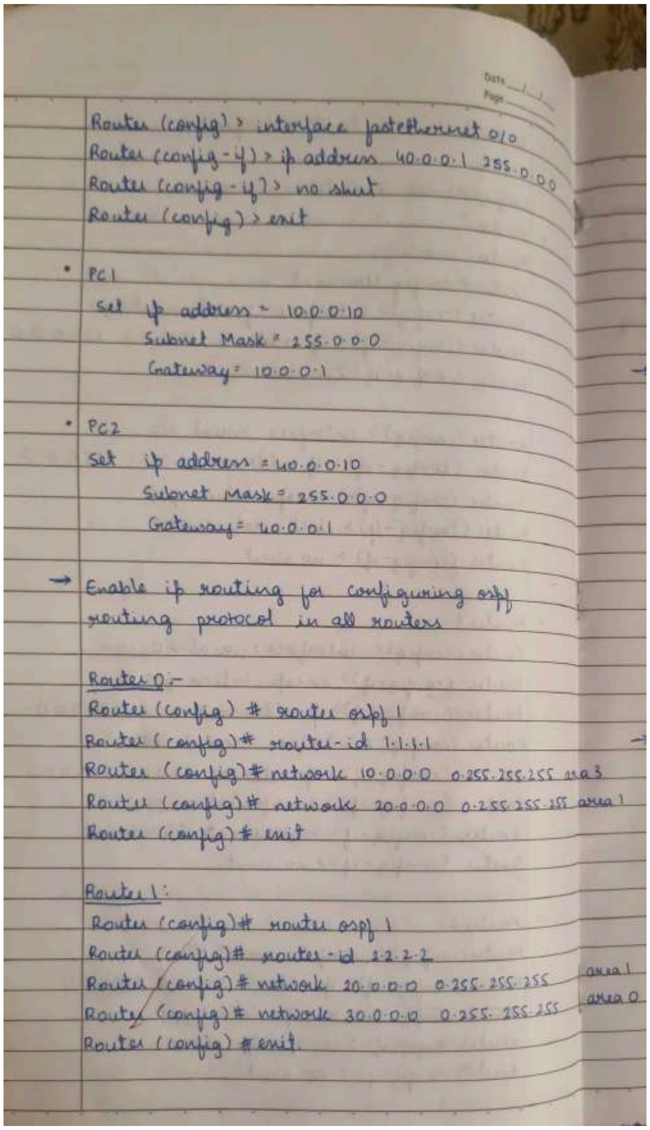
```
Router> enable
Router# config terminal
Router (config)> interface fastethernet 0/0
Router (config-if) > ip address 10.0.0.1 255.0.0.0
Router (config-if) > no shut

Router (config) > interface serial 2/0
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 shut
```

- Router1:

```
Router (config)> interface serial 2/0
Router (config-if)> encapsulation ppp
Router (config-if)> ip address 20.0.0.2 255.0.0.0
Router (config) > interface serial 3/0
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 shut.
```

- Router 2:

```
Router(config)> interfalce serial 2/0
Router (config- if)> ip address 30.0.0.2 255.0.0.0
Router (config- if) > encapsulation ppp
Router(config-if) > clock rate 64000
Router(config-if) > no shut
```



Router (config) > interface fastethernet 0/0
Router (config-if) > ip address 40.0.0.1 255.0.0.0
Router (config-if) > no shut
Router (config) > exit

- PC1

Set ip address = 10.0.0.10
Subnet Mask = 255.0.0.0
Gateway = 10.0.0.1

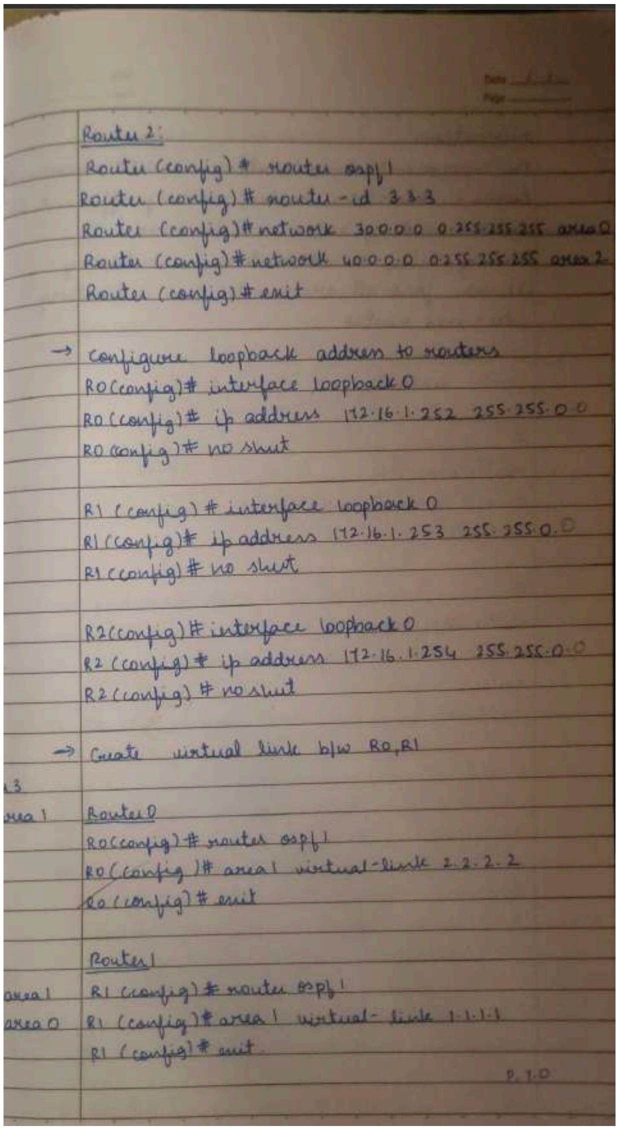
- PC2

Set ip address = 40.0.0.10
Subnet Mask = 255.0.0.0
Gateway = 40.0.0.1

→ Enable ip routing for configuring ospf routing protocol in all routers

Router 0:-
Router (config) # router ospf 1
Router (config)# router-id 1.1.1.1
Router (config)# network 10.0.0.0 0.255.255.255 ara 3
Router (config)# network 20.0.0.0 0.255.255.255 area 1
Router (config) # exit

Router 1:
Router (config)# router ospf 1
Router (config)# router-id 2.2.2.2
Router (config)# network 20.0.0.0 0.255.255.255 area 1
Router (config)# network 30.0.0.0 0.255.255.255 area 0
Router (config) # exit.



Router 2:
Router (config) # router ospf 1
Router (config) # router-id 3.3.3
Router (config)# network 30.0.0.0 0.255.255.255 area 0
Router (config) # network 40.0.0.0 0.255.255.255 area 2
Router (config) # exit

→ configure loopback address to routers
R0(config)# interface loopback 0
R0 (config)# ip address 172.16.1.252 255.255.0.0
R0 (config)# no shut

R1 (config) # interface loopback 0
R1 (config)# ip address 172.16.1.253 255.255.0.0
R1 (config) # no shut

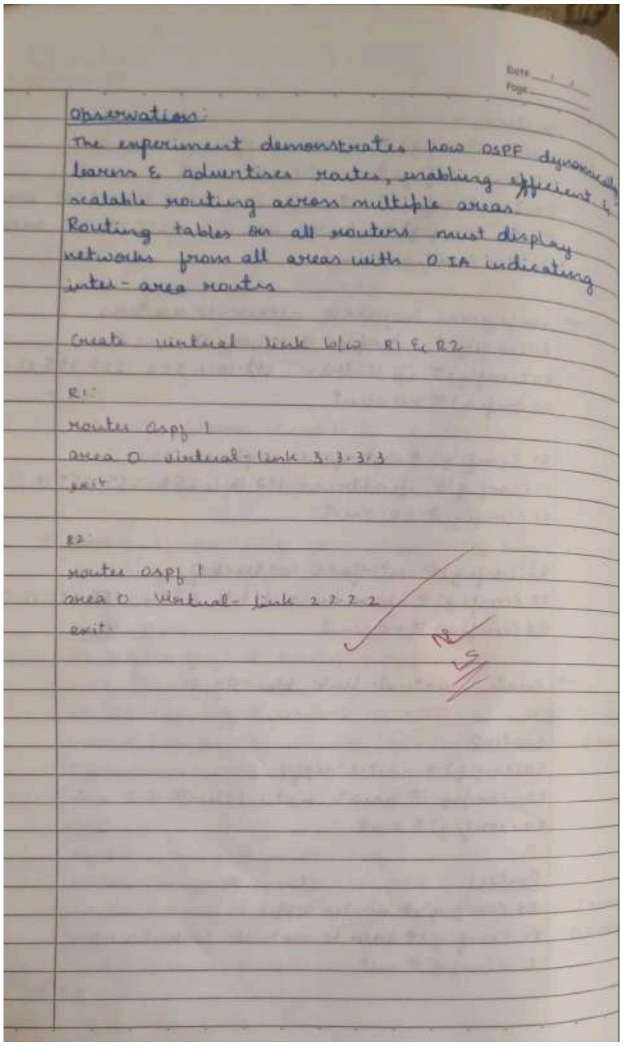
R2(config)# interface loopback 0
R2 (config)# ip address 172.16.1.254 255.255.0.0
R2 (config) # no shut

→ Create virtual link b/w R0, R1

Router 0
R0(config)# router ospf 1
R0(config)# area 1 virtual-link 2.2.2.2
R0 (config) # exit

Router 1
R1 (config) # router ospf 1
R1 (config)# area 1 virtual-link 1.1.1.1
R1 (config) # exit.

P.T.O



Date __/__/__
Page ____

Observation:

The experiment demonstrates how OSPF dynamically learns & advertises routes, enabling efficient & scalable routing across multiple areas.
Routing tables on all routers must display networks from all areas with O IA indicating inter-area routes

Create virtual link b/w R1 & R2

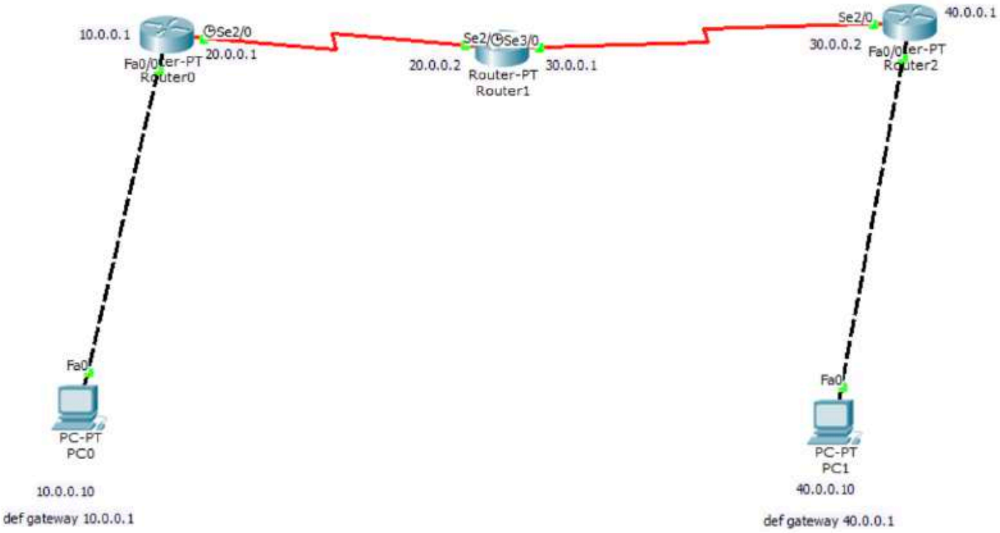
R1:

```
router ospf 1
area 0 virtual-link 3.3.3.3
exit
```

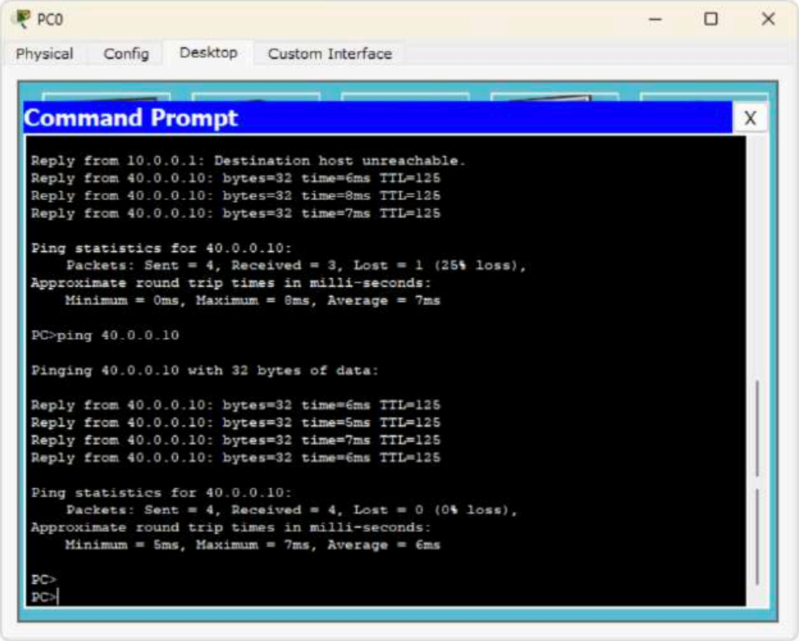
R2:

```
router ospf 1
area 0 virtual-link 2.2.2.2
exit
```

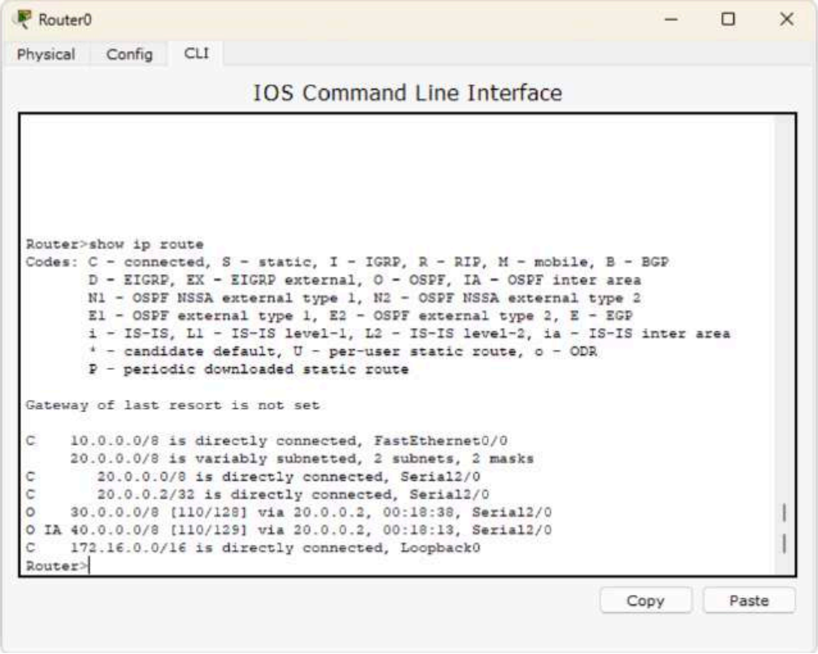
Screenshot of Topology:



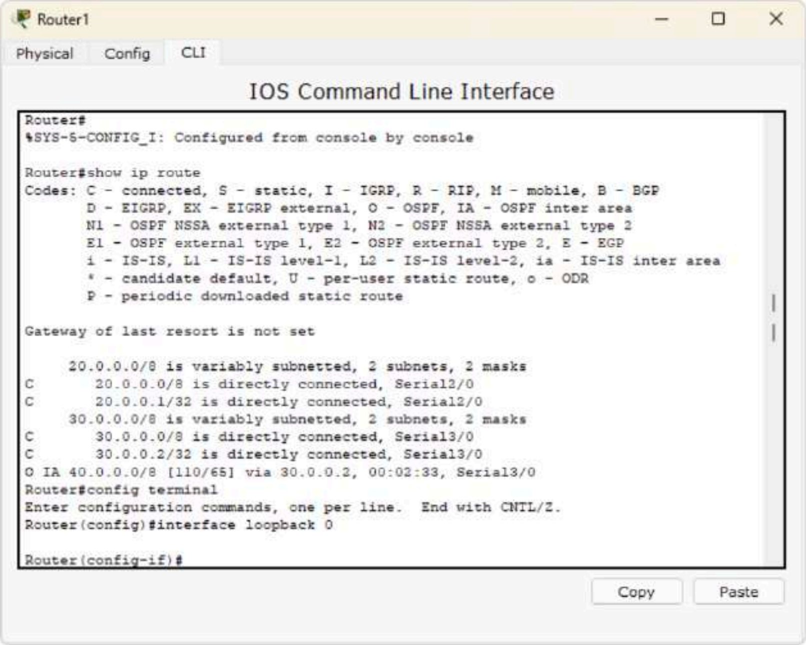
Screenshot of Output:



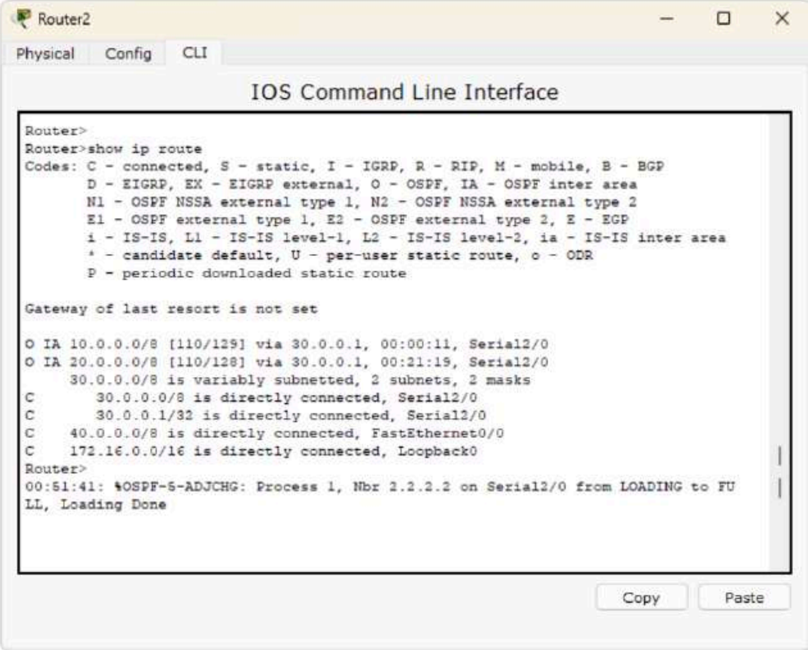
```
Reply from 10.0.0.1: Destination host unreachable.
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 8ms, Average = 7ms

PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms 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 = 5ms, Maximum = 7ms, Average = 6ms

PC>
PC>
```



```
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.0.0.0/8 is directly connected, Serial2/0
C       20.0.0.2/32 is directly connected, Serial2/0
O    30.0.0.0/8 [110/128] via 20.0.0.2, 00:18:38, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:18:13, Serial2/0
C    172.16.0.0/16 is directly connected, Loopback0
Router>
```



```
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

     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, 00:02:33, Serial3/0
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
```

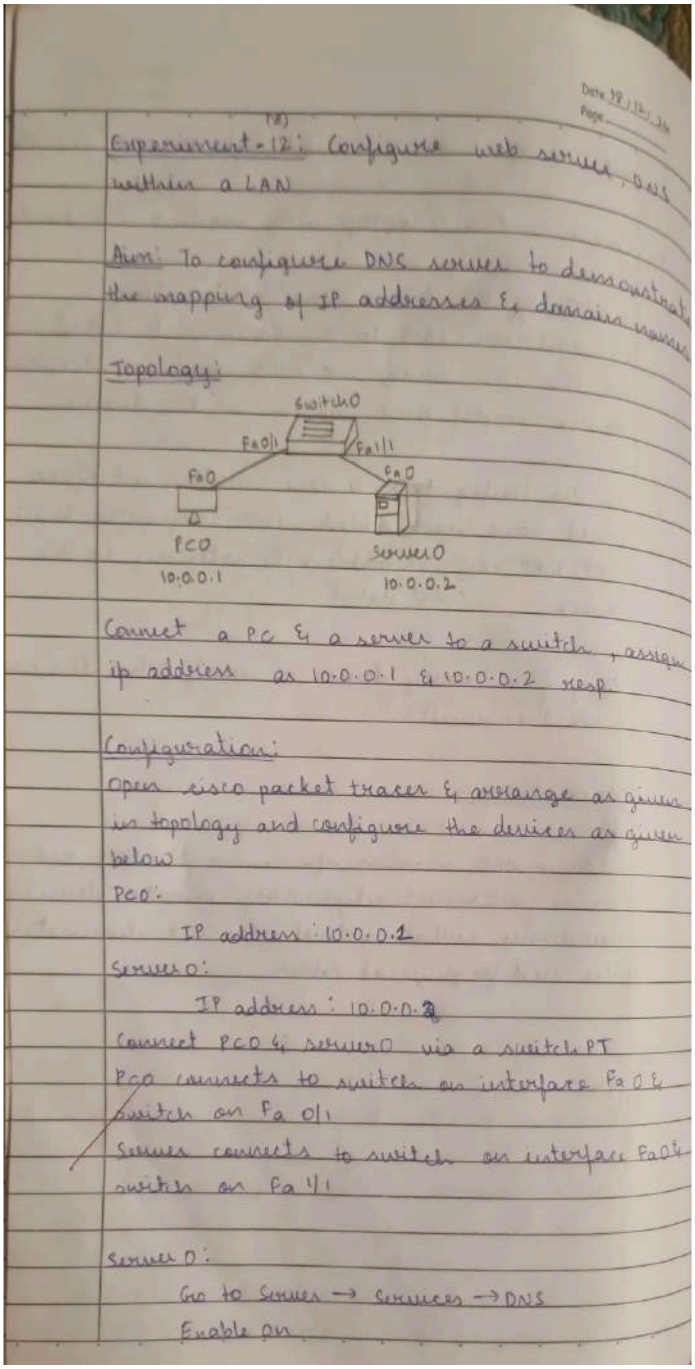


```
Router>
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 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 10.0.0.0/8 [110/129] via 30.0.0.1, 00:00:11, Serial2/0
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:21:19, Serial2/0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial2/0
C       30.0.0.1/32 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
C    172.16.0.0/16 is directly connected, Loopback0
Router>
00:51:41: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FU
LL, Loading Done
```

Program 8

Q8: Configure Web Server , DNS within a LAN

Observation:



Date 18/12/24
Page

(8)

Experiment-12: Configure web server, DNS within a LAN

Aim: To configure DNS server to demonstrate the mapping of IP addresses & domain names

Topology:

switch0
Fa0/1
Fa1/1
Fa0
Fa0
PC0
10.0.0.1
server0
10.0.0.2

Connect a PC & a server to a switch, assign ip address as 10.0.0.1 & 10.0.0.2 resp.

Configuration:

Open cisco packet tracer & arrange as given in topology and configure the devices as given below

PC0:

IP address : 10.0.0.1

Server0:

IP address : 10.0.0.2

Connect PC0 & server0 via a switch PT

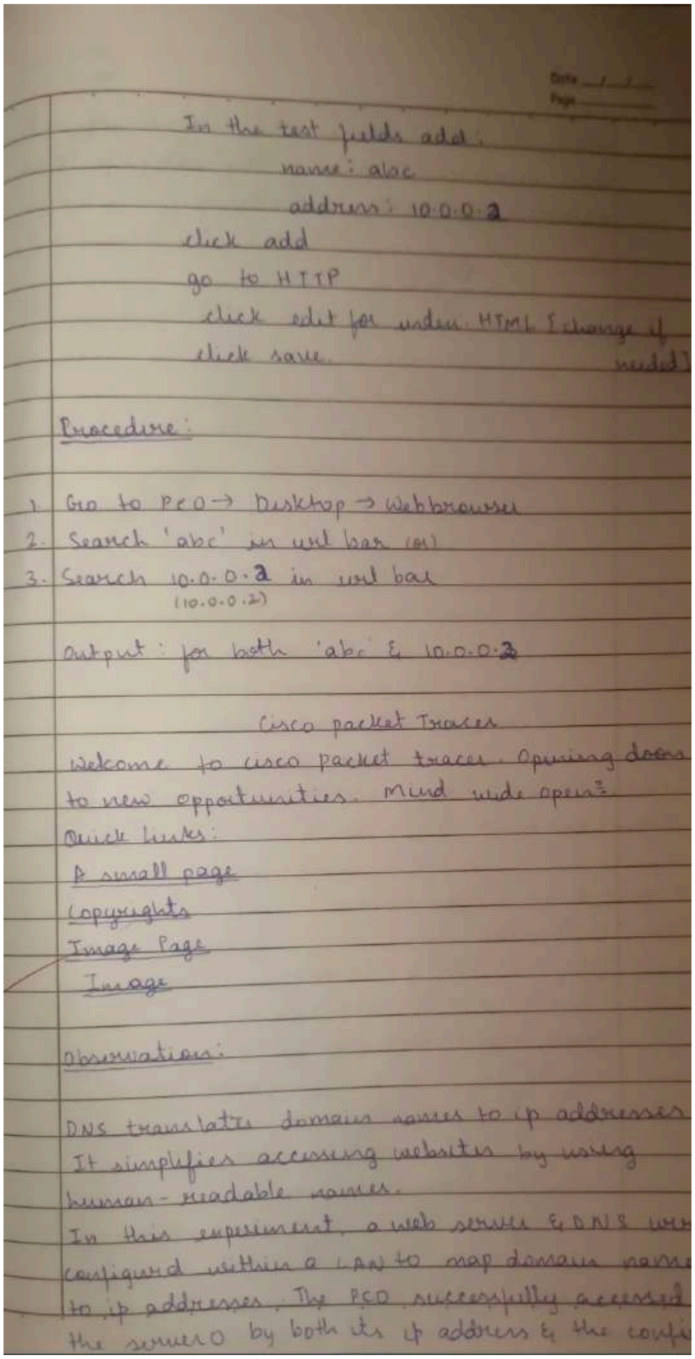
PC0 connects to switch on interface Fa0 & switch on Fa 0/1

Server connects to switch on interface Fa0 & switch on Fa 1/1

Server 0:

Go to Server → Services → DNS

Enable on



Date _/_/_
Page

In the test fields add:

name: abc

address: 10.0.0.2

click add

go to HTTP

click edit for index.HTML [change if needed]

click save.

Procedure:

1. Go to PC0 → Desktop → Webbrowser
2. Search 'abc' in url bar (or)
3. Search 10.0.0.2 in url bar (10.0.0.2)

Output: for both 'abc' & 10.0.0.2

Cisco packet Tracer

Welcome to cisco packet tracer. Opening doors to new opportunities. Mind wide open.

Quick links:

A small page

Copyrights

Image Page

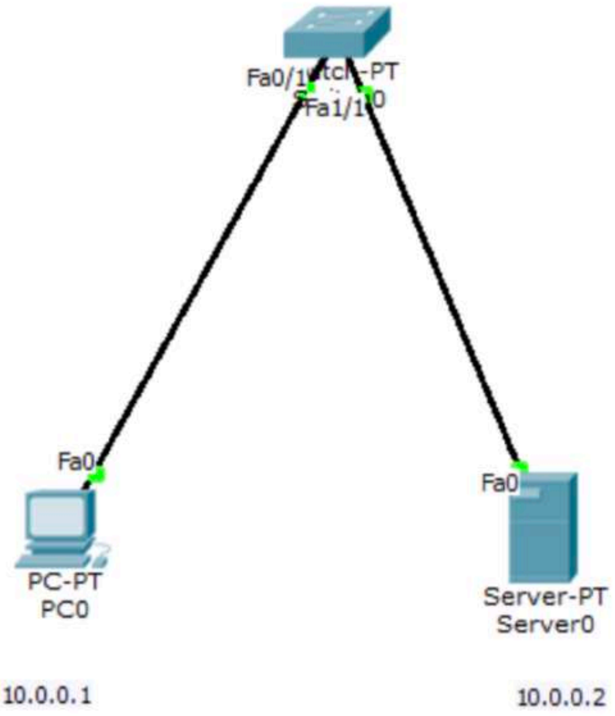
Image

Observation:

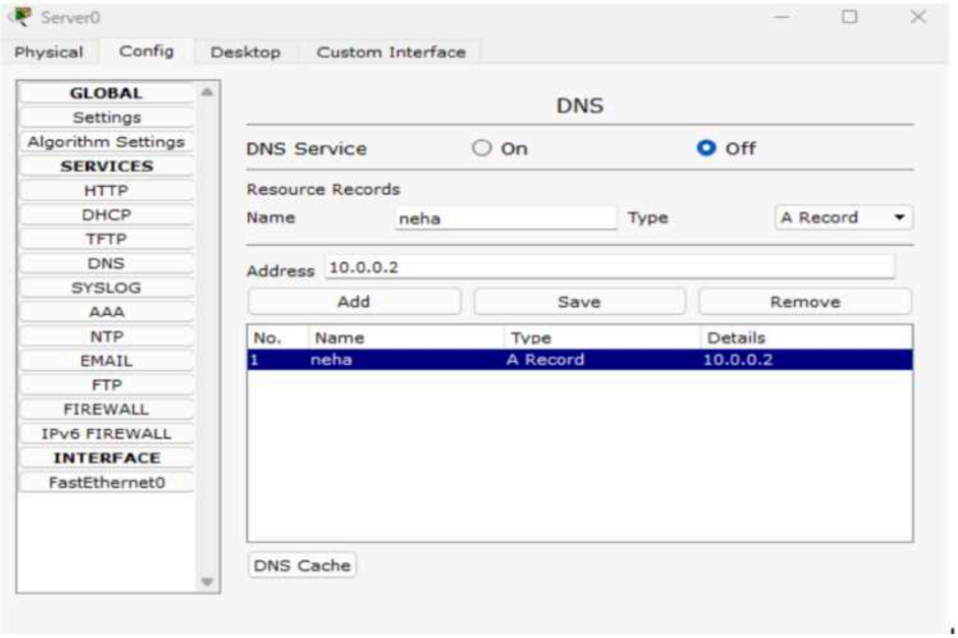
DNS translates domain names to ip addresses. It simplifies accessing websites by using human-readable names.

In this experiment, a web server & DNS wer configured within a LAN to map domain nam to ip addresses. The PC0 successfully accessed the server0 by both its ip address & the confi

Screenshot of Topology:



Screenshot of Output:



DNS

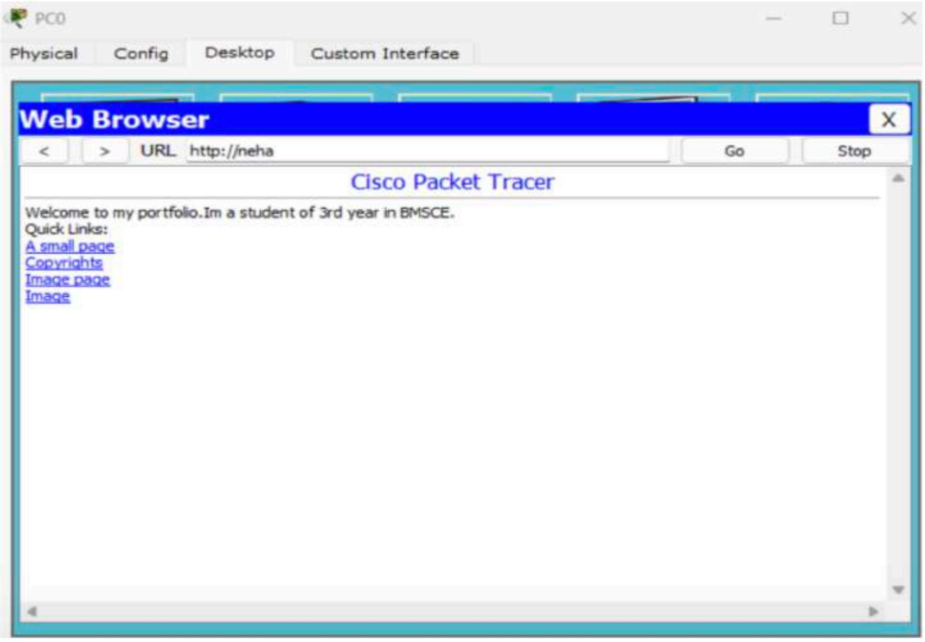
DNS Service ○ On ● Off

Resource Records

Name neha Type A Record

Address 10.0.0.2

No.	Name	Type	Details
1	neha	A Record	10.0.0.2



http://neha

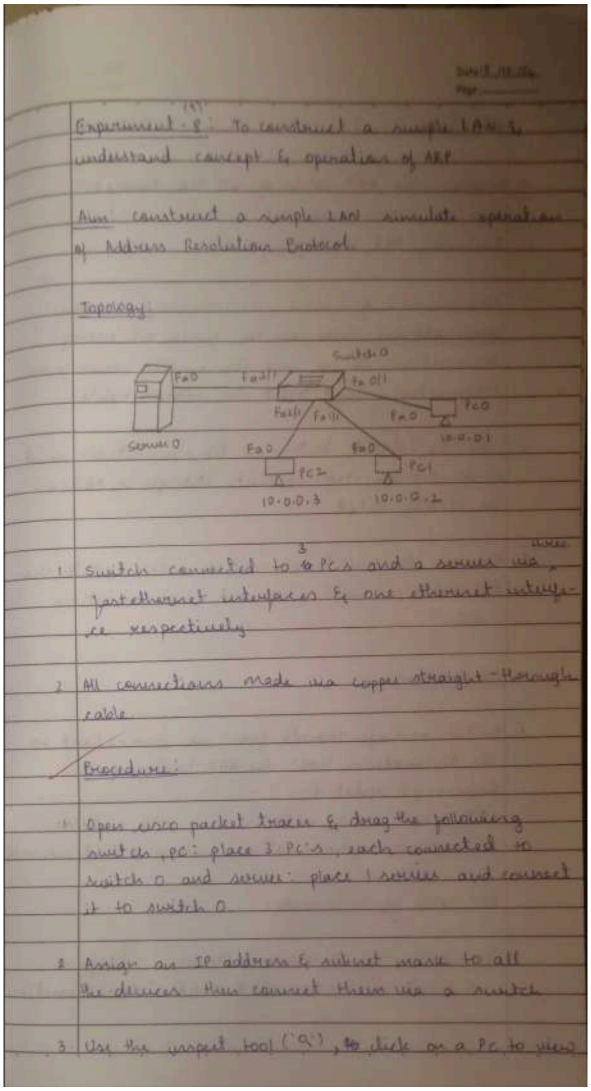
Cisco Packet Tracer

Welcome to my portfolio.Im a student of 3rd year in BMSCE.
Quick Links:
A small page
Copyrights
Image page
Image

Program 9:

Q9: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Observation:



Date 1/11/24
Page

Experiment 8 (9): To construct a simple LAN & understand concept & operation of ARP

Aim: construct a simple LAN simulate operation of Address Resolution Protocol.

Topology:

Switch0
Fa0
Fa0/1
Fa0/1
Fa0/1
Fa0/1
Fa0
PC0
10.0.0.1
Server0
Fa0
PC2
Fa0
PC1
10.0.0.3
10.0.0.2

1. Switch connected to ~~6~~ 3 PC's and a server via wires, fastethernet interfaces & one ethernet interface respectively

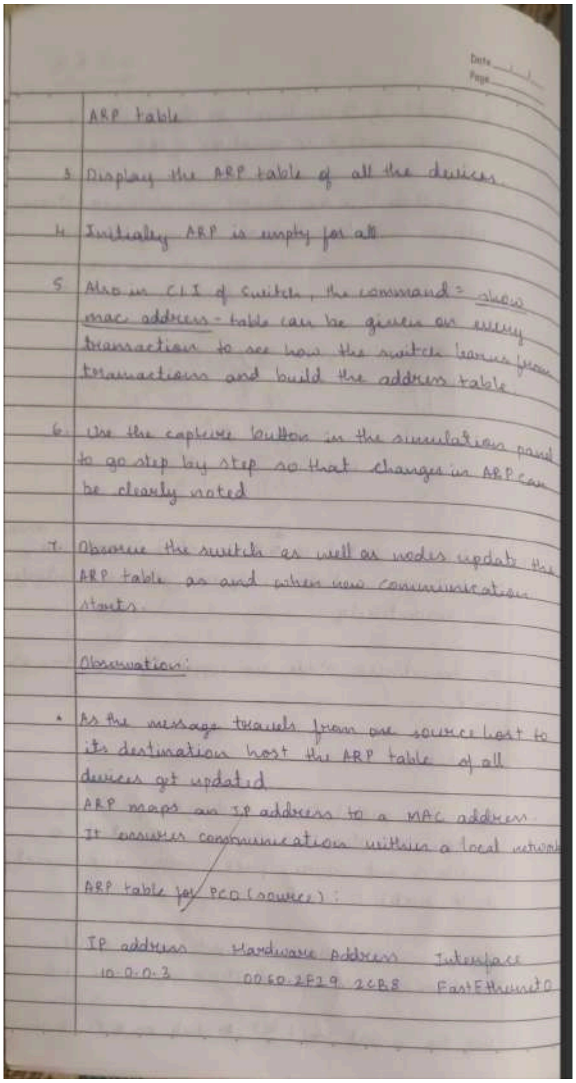
2. All connections made via copper straight-through cable.

Procedure:

1. Open cisco packet tracer & drag the following switch, pc: place 3 PC's, each connected to switch 0 and server: place 1 server and connect it to switch 0.

2. Assign an IP address & subnet mask to all the devices then connect them via a switch

3. Use the inspect tool ('Q'), ~~to~~ click on a PC to view



Date / /
Page

ARP table

3. Display the ARP table of all the devices.

4. Initially ARP is empty for all.

5. Also in CLI of switch, the command = show mac address-table can be given on every transaction to see how the switch learns from transactions and build the address table

6. Use the capture button in the simulation panel to go step by step so that changes in ARP can be clearly noted

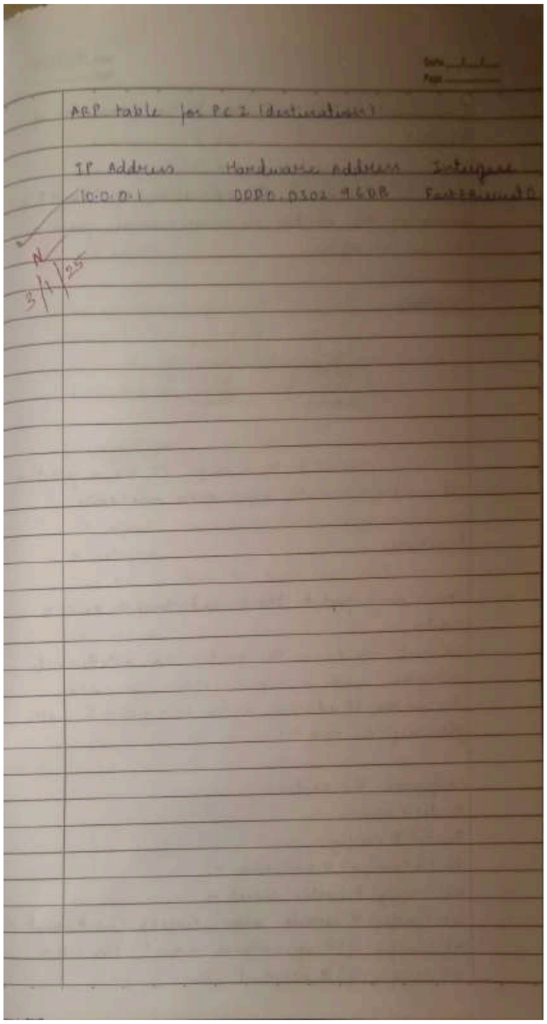
7. Observe the switch as well as nodes update the ARP table as and when new communication starts.

Observation:

- As the message travels from one source host to its destination host the ARP table of all devices get updated.
ARP maps an IP address to a MAC address.
It ensures communication within a local network

ARP table for PC0 (source):

IP address	Hardware Address	Interface
10.0.0.3	0060.2F29.2CB8	FastEthernet0



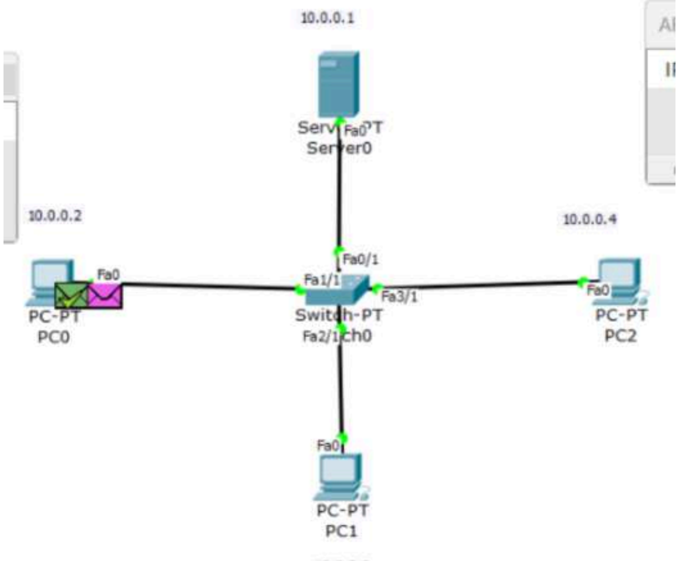
Date ______
Page ______

ARP table for PC 2 (destination)

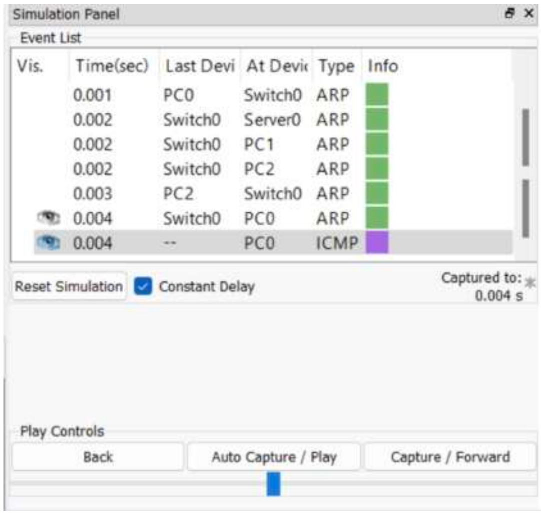
IP Address	Hardware address	Interface
10.0.0.1	00D0.D302.960B	FastEthernet0

N
3/1/25

Screenshot of Topology:



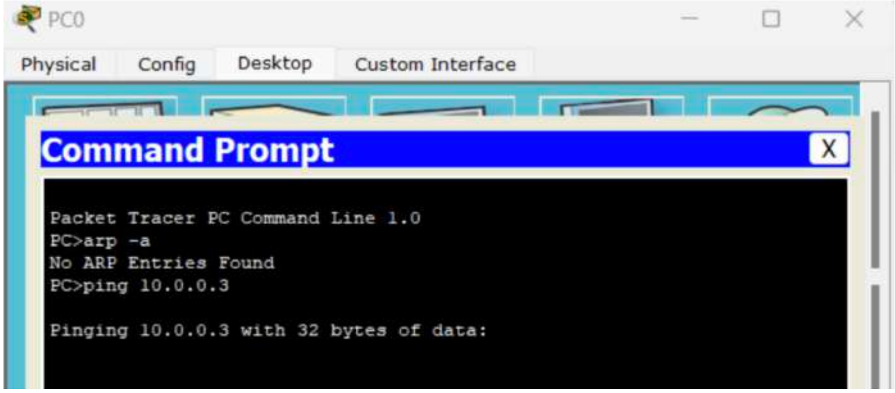
Screenshot of Output:



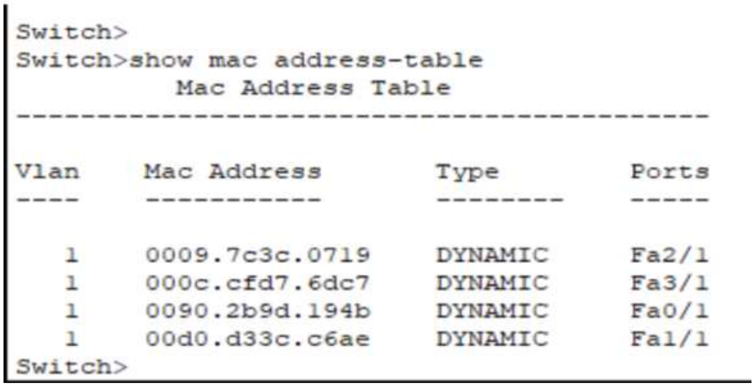
Vis.	Time(sec)	Last Devi	At Devic	Type	Info
	0.001	PC0	Switch0	ARP	🟩
	0.002	Switch0	Server0	ARP	🟩
	0.002	Switch0	PC1	ARP	🟩
	0.002	Switch0	PC2	ARP	🟩
	0.003	PC2	Switch0	ARP	🟩
	0.004	Switch0	PC0	ARP	🟩
	0.004	--	PC0	ICMP	🟪

☑ Constant Delay

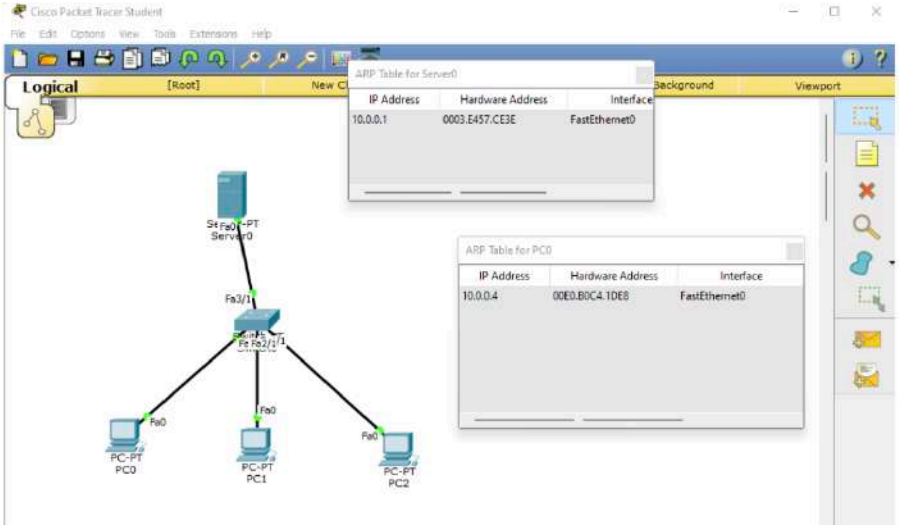
Captured to:
0.004 s



```
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:
```



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

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

   1    0009.7c3c.0719    DYNAMIC     Fa2/1
   1    000c.cfd7.6dc7    DYNAMIC     Fa3/1
   1    0090.2b9d.194b    DYNAMIC     Fa0/1
   1    00d0.d33c.c6ae    DYNAMIC     Fa1/1
Switch>
```

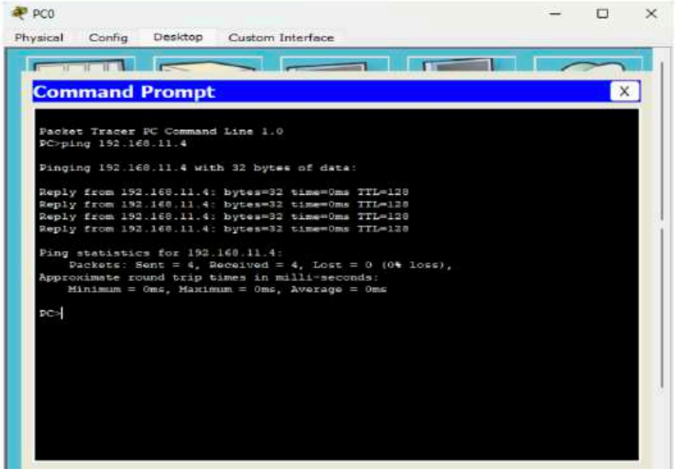


ARP Table for Server0

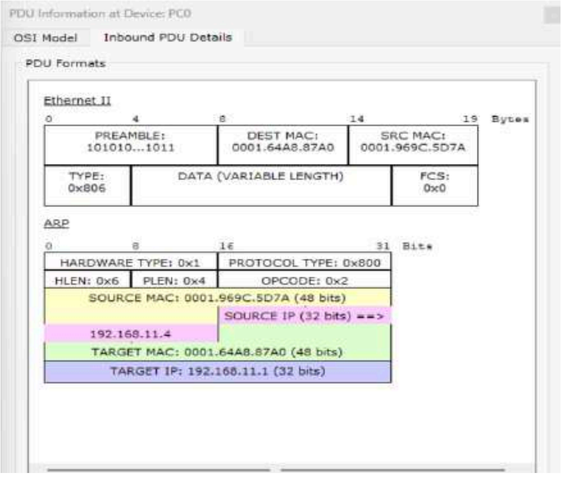
IP Address	Hardware Address	Interface
10.0.0.1	0003.E457.CE3E	FastEthernet0

ARP Table for PC0

IP Address	Hardware Address	Interface
10.0.0.4	00E0.B0C4.1DE8	FastEthernet0



```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.11.4

Pinging 192.168.11.4 with 32 bytes of data:

Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128
Reply from 192.168.11.4: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.11.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```



Ethernet II

0 4 8 14 19 Bytes

PREAMBLE: 101010...1011
DEST MAC: 0001.64A8.87A0
SRC MAC: 0001.969C.5D7A
TYPE: 0x806
DATA (VARIABLE LENGTH)
FCS: 0x0

ARP

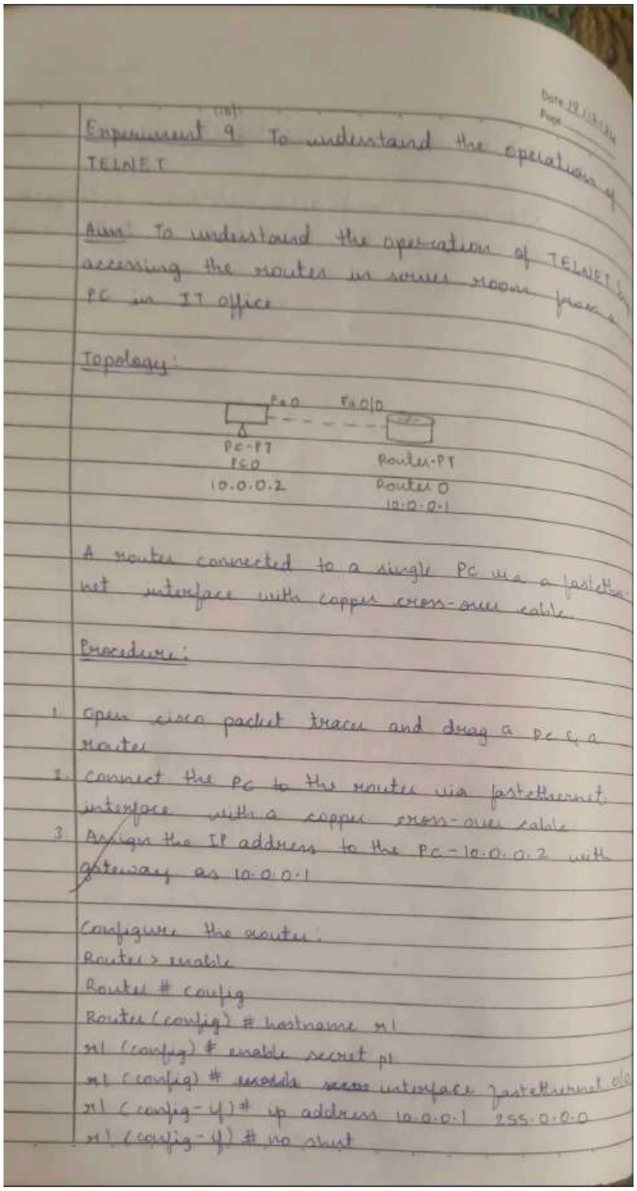
0 8 16 31 Bits

HARDWARE TYPE: 0x1
PROTOCOL TYPE: 0x800
HLEN: 0x6
PLEN: 0x4
OPCODE: 0x2
SOURCE MAC: 0001.969C.5D7A (48 bits)
SOURCE IP (32 bits) ==>
192.168.11.4
TARGET MAC: 0001.64A8.87A0 (48 bits)
TARGET IP: 192.168.11.1 (32 bits)

Program 10

Q10: To understand the operation of TELNET

Observation:



(16)

Date 19/12/24
Page

Experiment 9: To understand the operation of TELNET

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

Topology:

Fa0 Fa0/0
PC-PT
PC0
10.0.0.2
Router-PT
Router0
10.0.0.1

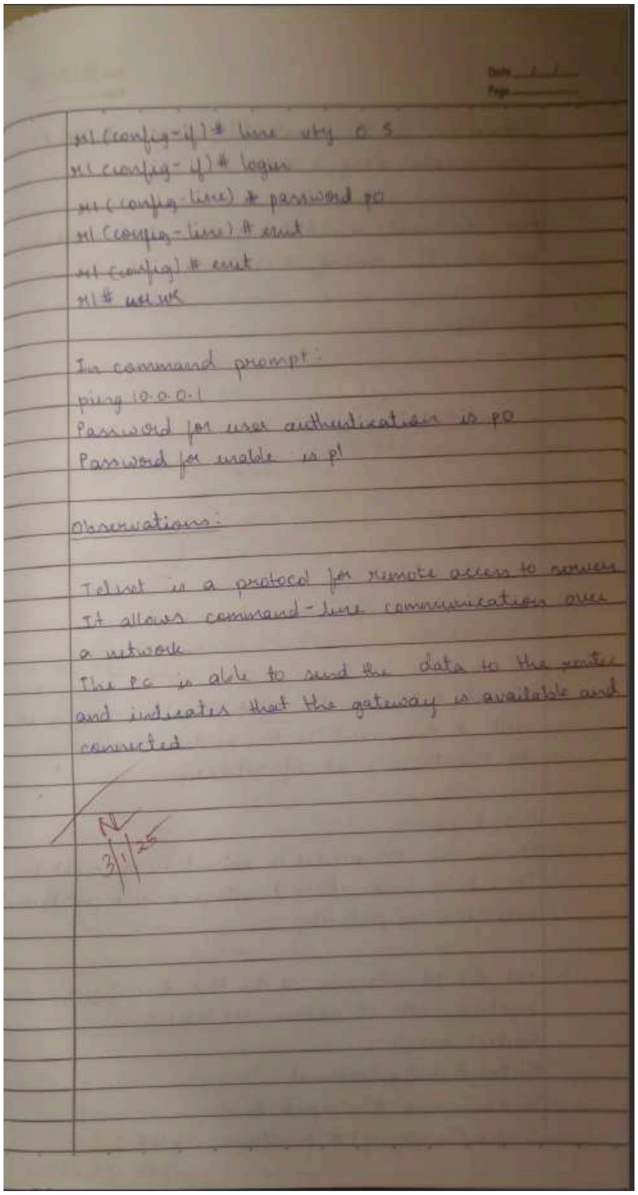
A router connected to a single PC via a fastethernet interface with copper cross-over cable.

Procedure:

1. Open cisco packet tracer and drag a PC & a router
2. Connect the PC to the router via fastethernet interface with a copper cross-over cable
3. Assign the IP address to the PC - 10.0.0.2 with gateway as 10.0.0.1

Configure the router:

```
Router> enable
Router # config
Router (config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # ~~enable~~ ~~secre~~ interface fastethernet 0/0
r1 (config-if)# ip address 10.0.0.1 255.0.0.0
r1 (config-if) # no shut
```



Date __/__/__
Page ____

```
r1 (config-if)# line vty 0 5
r1 (config- if)# login
r1 (config-line) # password p0
r1 (config-line) # exit
r1 (config) # exit
r1# wr wr
```

In command prompt:

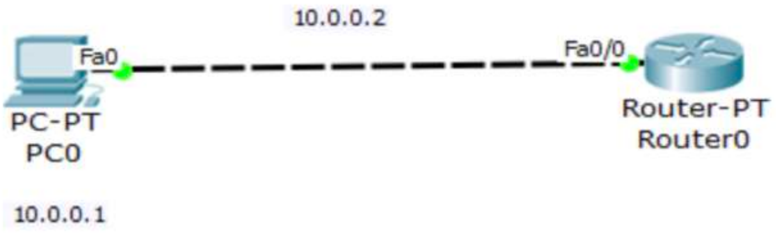
ping 10.0.0.1
Password for user authentication is p0
Password for enable is p1

Observations:

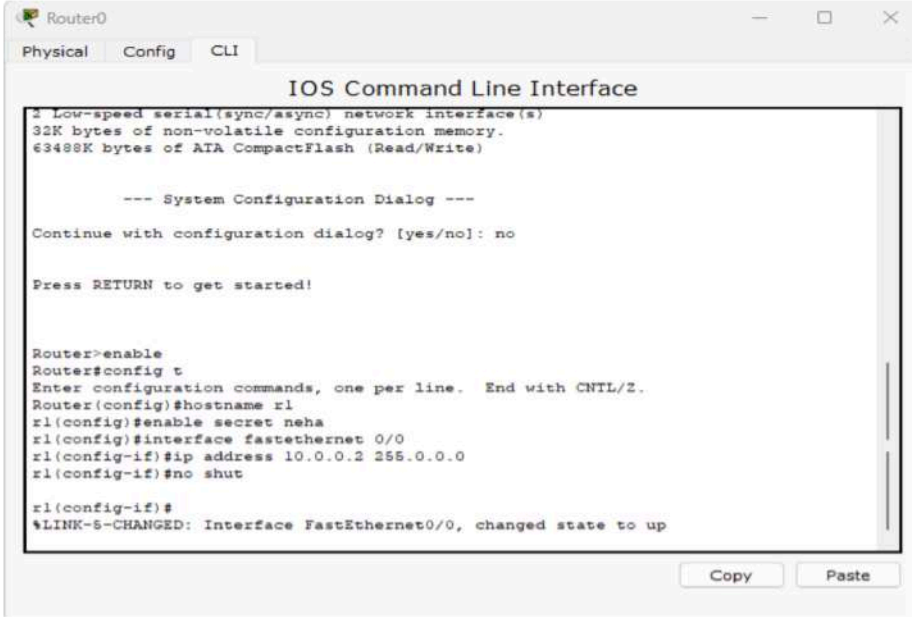
Telnet is a protocol for remote access to servers. It allows command-line communication over a network
The PC is able to send the data to the router and indicates that the gateway is available and connected

N
3/1/25

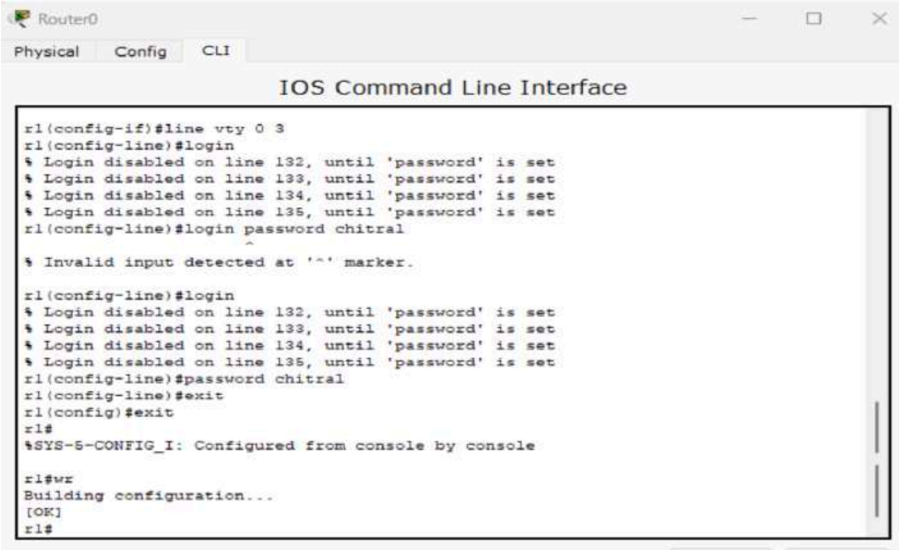
Screenshot of Topology:



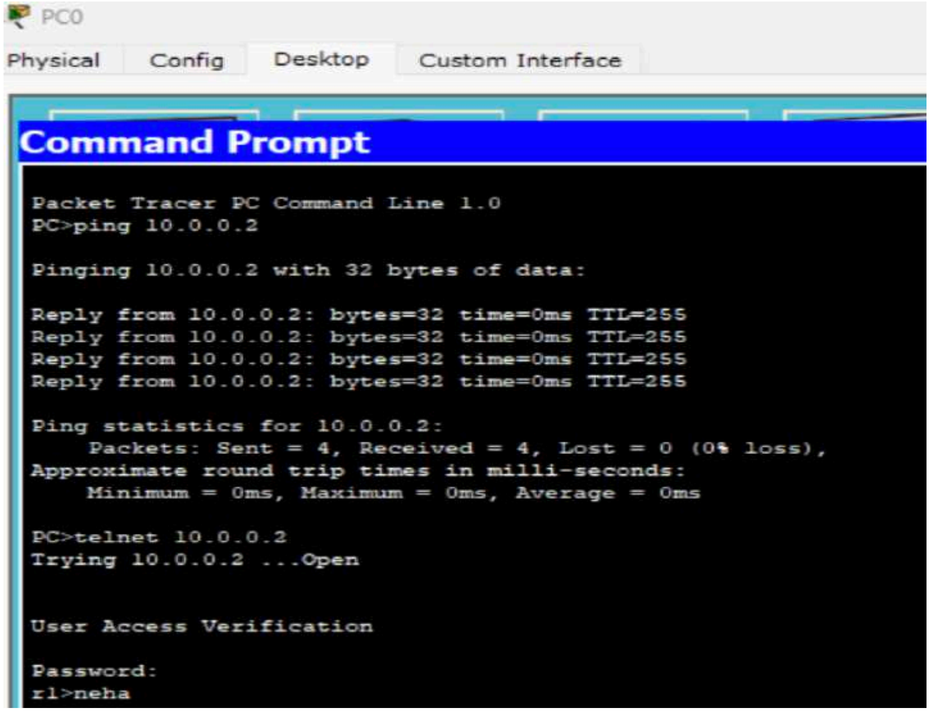
Screenshot of Output:



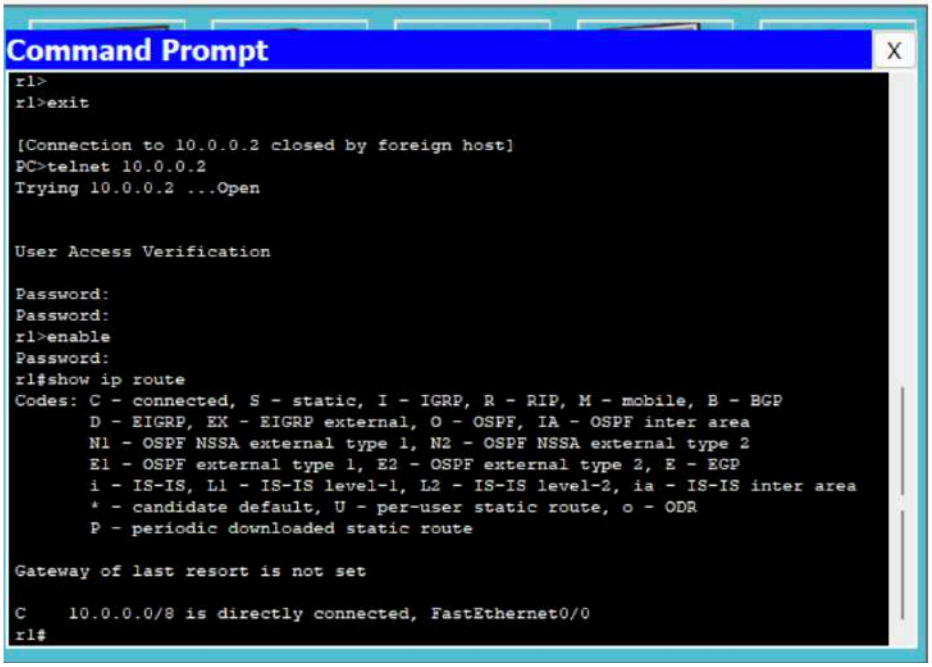
```
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)


         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname rl
rl(config)#enable secret neha
rl(config)#interface fastethernet 0/0
rl(config-if)#ip address 10.0.0.2 255.0.0.0
rl(config-if)#no shut

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



```
rl(config-if)#line vty 0 3
rl(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
rl(config-line)#login password chitral
                      ^
% Invalid input detected at '^' marker.

rl(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
rl(config-line)#password chitral
rl(config-line)#exit
rl(config)#exit
rl#
%SYS-5-CONFIG_I: Configured from console by console

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



```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
r1>neha
```

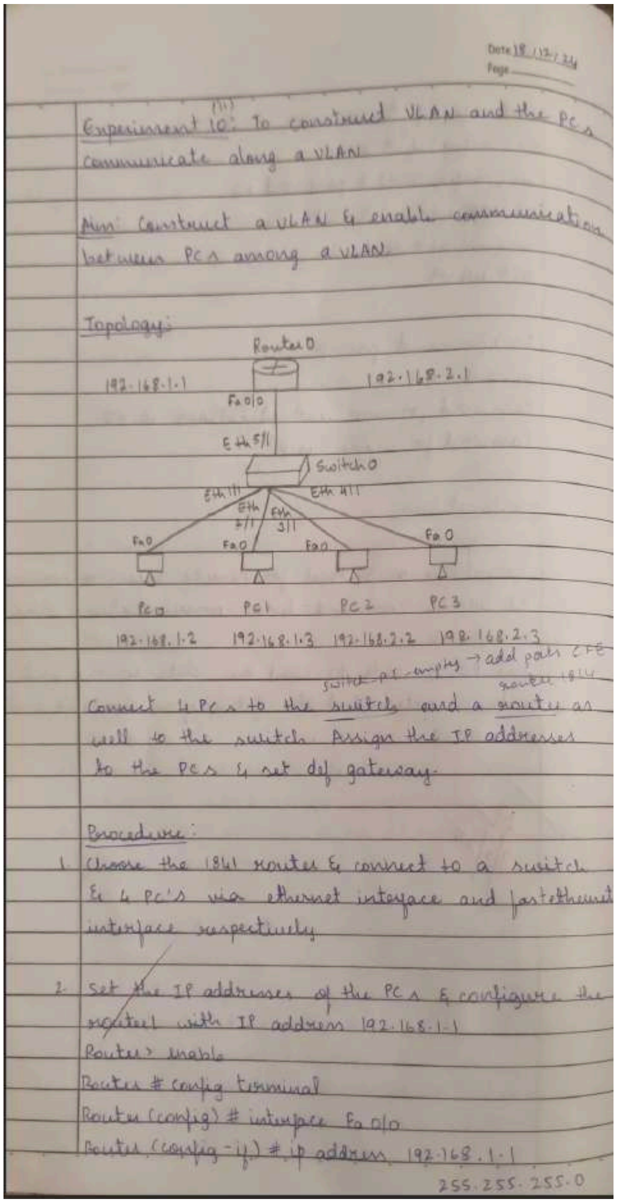


```
r1>
r1>exit

[Connection to 10.0.0.2 closed by foreign host]
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
Password:
r1>enable
Password:
r1#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
r1#
```

Program 11

Q11: To construct VLAN and the PC's communicate along a VLAN

Observation:



Date 18/12/24
Page

Experiment ~~10~~ (11): To construct VLAN and the PCs communicate along a VLAN

Aim: Construct a VLAN & enable communication between PCs among a VLAN.

Topology:

Router0
192.168.1.1
192.168.2.1
Fa0/0
Eth5/1
Switch0
Eth1/1
Eth 2/1
Eth 3/1
Eth 4/1
Fa0
Fa0
Fa0
Fa 0
PC0
PC1
PC2
PC3
192.168.1.2
192.168.1.3
192.168.2.2
198.168.2.3

switch-PT-empty → add ports CFE
router 1841

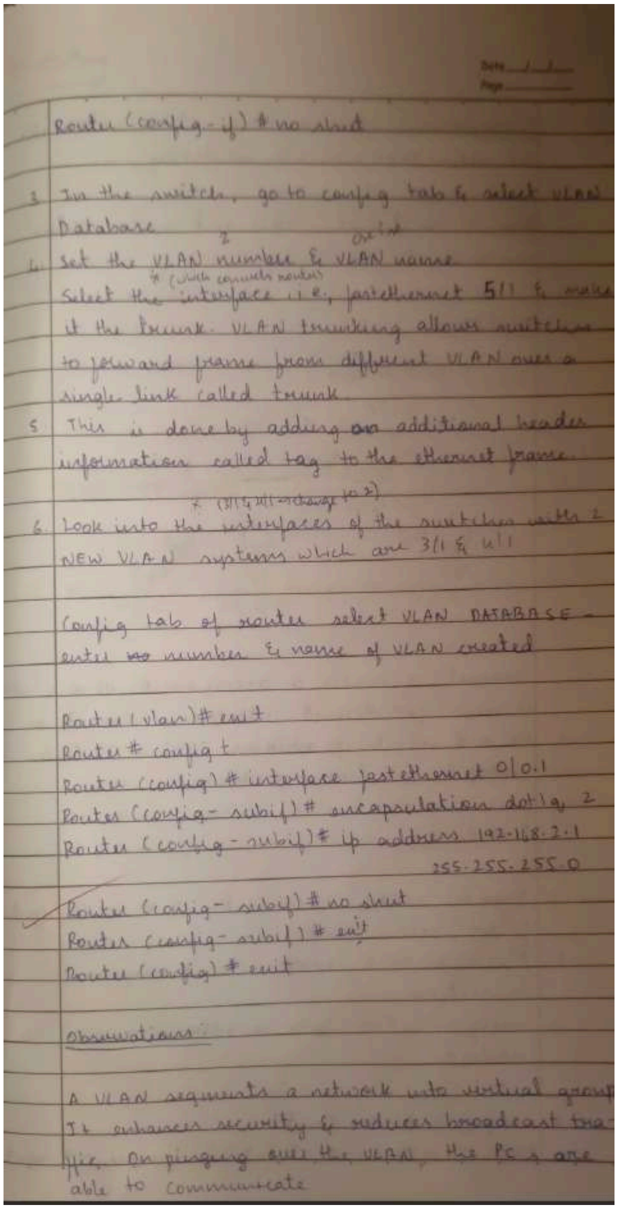
Connect 4 PCs to the switch and a router as well to the switch. Assign the IP addresses to the PCs & set def gateway.

Procedure:

1. Choose the 1841 router & connect to a switch & 4 PC's via ethernet interface and fastethernet interface respectively.

2. Set the IP addresses of the PCs & configure the router with IP address 192.168.1.1

```
Router> enable
Router # config terminal
Router (config) # interface Fa 0/0
Router (config-if) # ip address 192.168.1.1
                                 255.255.255.0
```



Date
Page

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

3. In the switch, go to config tab & select VLAN Database

4. Set the VLAN number (2) & VLAN name (vlan2).
Select the interface (which connects router) i.e., fastethernet 5/1 & make it the trunk. VLAN trunking allows switches to forward frame from different VLAN over a single link called trunk.

5. This is done by adding ~~an~~ additional header information called tag to the ethernet frame.

6. Look into the interfaces of the switches with 2 NEW VLAN systems which are 3/1 & 4/1 (3/1 & 4/1 → change to 2)

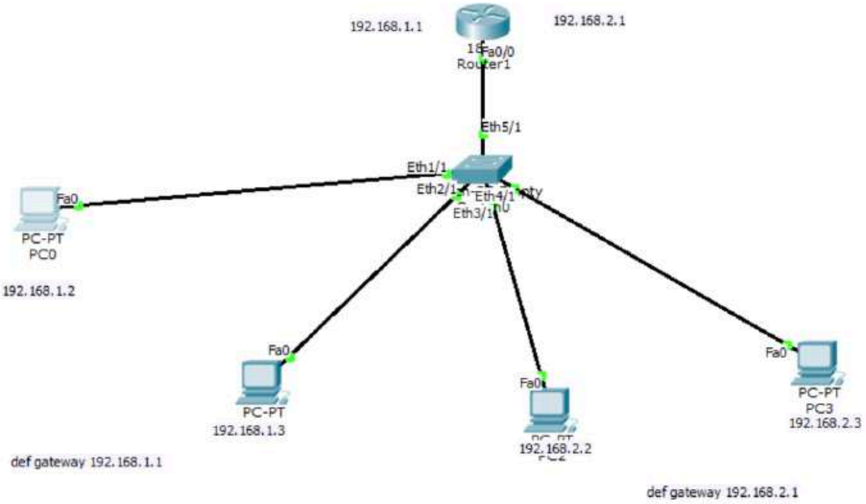
Config tab of router select VLAN DATABASE - enter ~~no~~ number & name of VLAN created

```
Router (vlan) # exit
Router # config t
Router (config) # interface fastethernet 0/0.1
Router (config-subif) # encapsulation dot1q 2
Router (config-subif) # ip address 192.168.2.1
                                   255.255.255.0
Router (config-subif) # no shut
Router (config-subif) # exit
Router (config) # exit
```

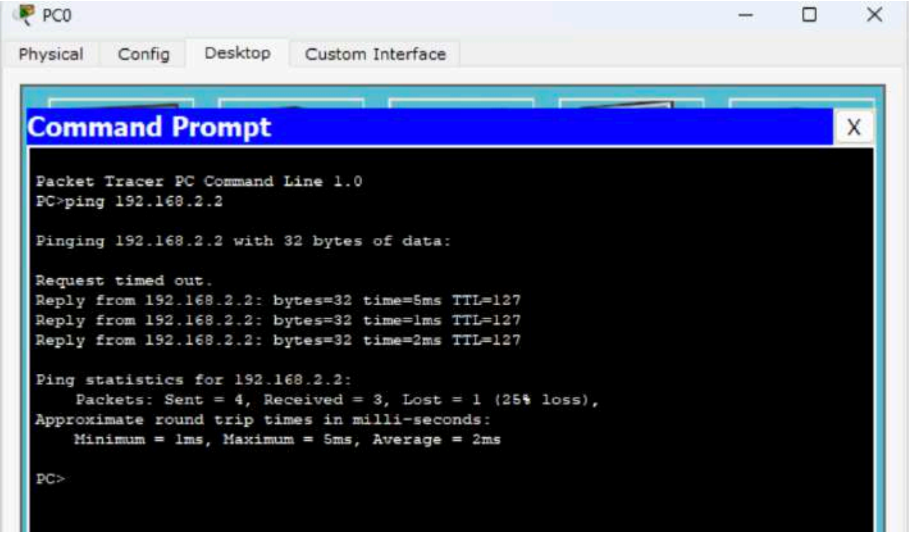
Observations:

A VLAN segments a network into virtual groups. It enhances security & reduces broadcast traffic. On pinging over the VLAN, the PCs are able to communicate

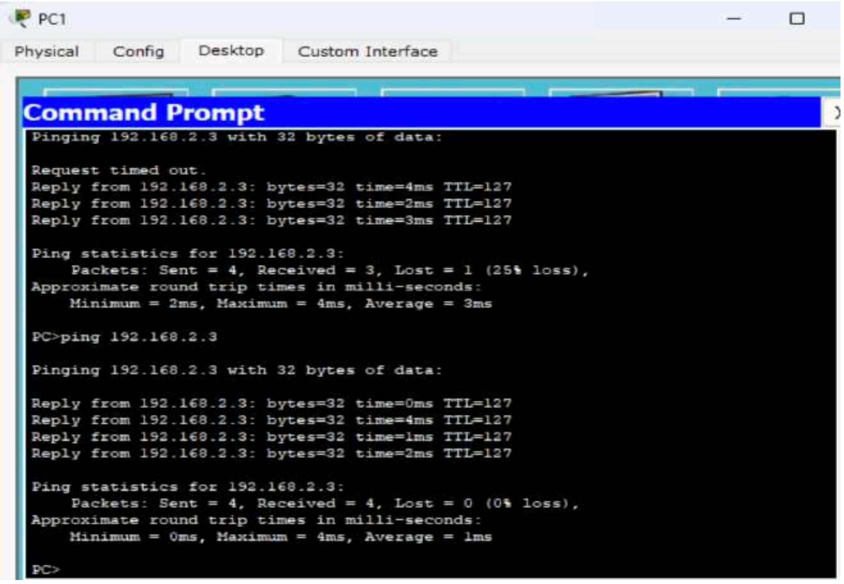
Screenshot of Topology:



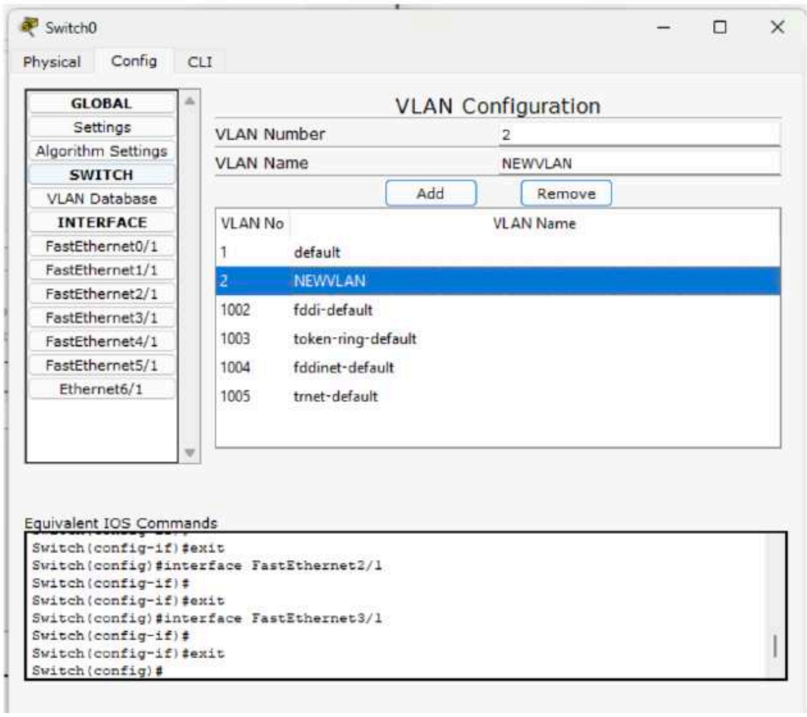
Screenshot of Output:



```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=5ms TTL=127
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 5ms, Average = 2ms

PC>
```



```
Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=4ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 3ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=4ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>
```



VLAN Configuration

VLAN Number 2

VLAN Name NEWVLAN

VLAN No	VLAN Name
1	default
2	NEWVLAN
1002	fddi-default
1003	token-ring-default
1004	fddinet-default
1005	trnet-default

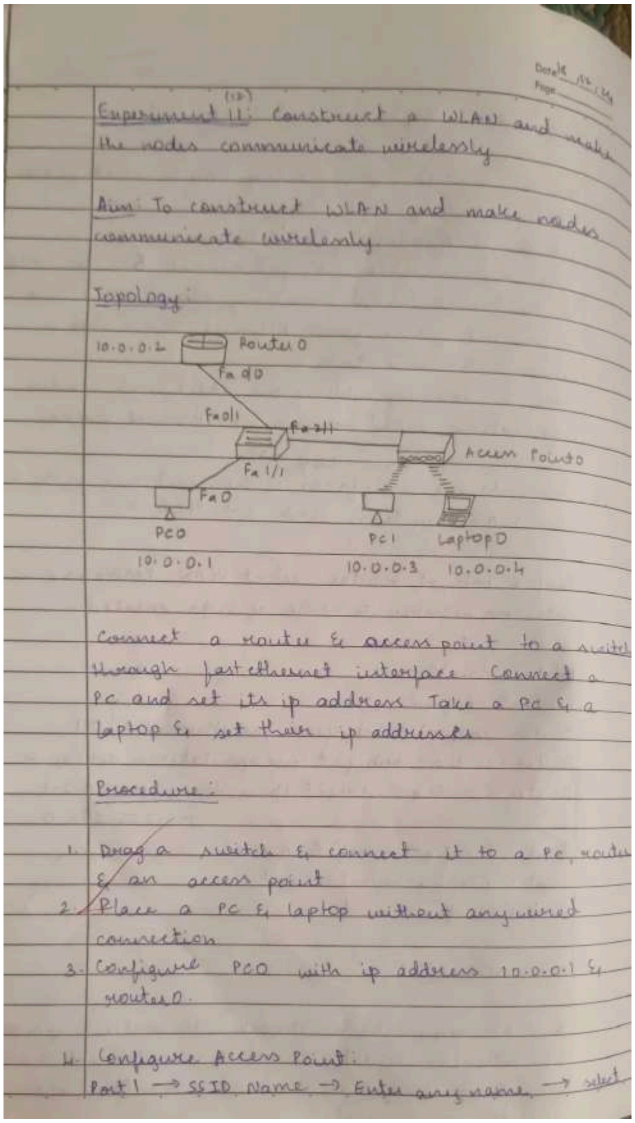
Equivalent IOS Commands

```
Switch(config-if)#exit
Switch(config)#interface FastEthernet2/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#interface FastEthernet3/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#
```

Program 12

Q12: Construct a WLAN and make the nodes communicate wirelessly.

Observation:



Date 18/12/24
Page

Experiment 11 (12): Construct a WLAN and make the nodes communicate wirelessly

Aim: To construct WLAN and make nodes communicate wirelessly.

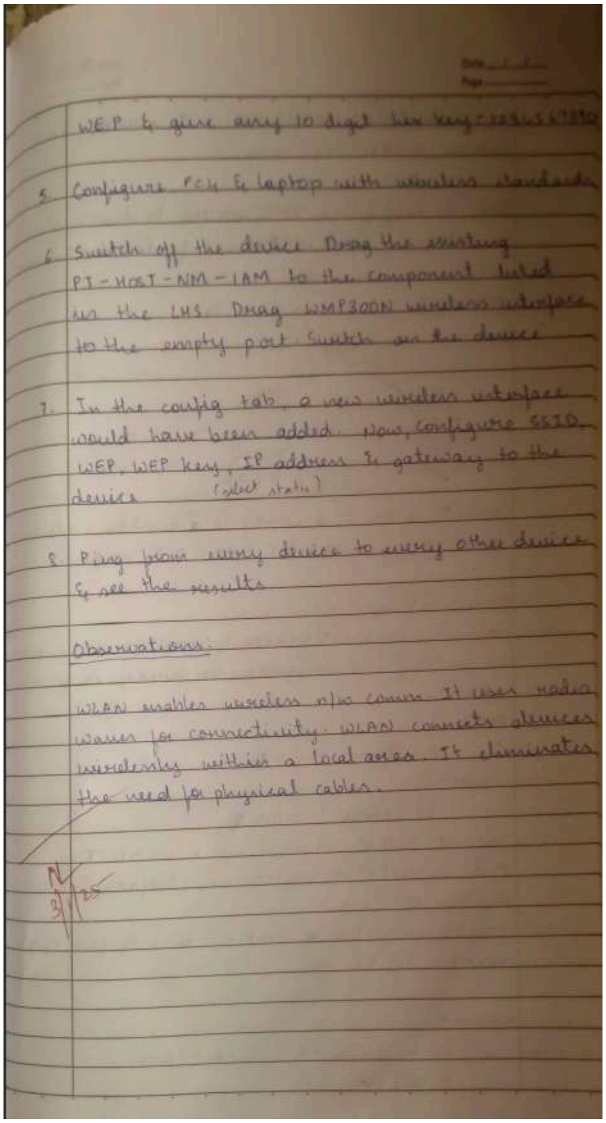
Topology:

10.0.0.2
Router0
Fa0/0
Fa0/1
Fa2/1
Access Point0
Fa1/1
Fa0
PC0
10.0.0.1
PC1
10.0.0.3
Laptop0
10.0.0.4

Connect a router & access point to a switch through fast ethernet interface. Connect a PC and set its ip address. Take a PC & a laptop & set their ip addresses

Procedure:

1. Drag a switch & connect it to a PC, router & an access point
2. Place a PC & laptop without any wired connection
3. Configure PC0 with ip address 10.0.0.1 & router0.
4. Configure Access Point:
Port 1 → SSID Name → Enter any name → select



Date
Page

WEP & give any 10 digit hex key-1234567890

5. Configure PC1 & laptop with wireless standards

6. Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to the empty port. Switch on the device

7. In the config tab, a new wireless interface would have been added. Now, configure SSID, WEP, WEP key, IP address & gateway to the device (select static)

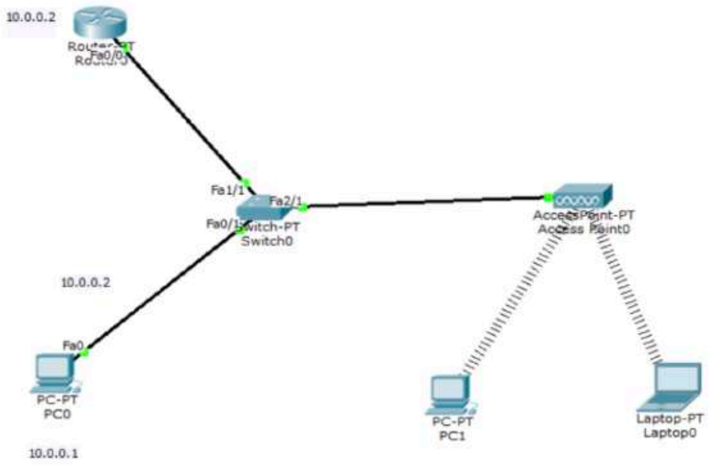
8. Ping from every device to every other device & see the results

Observations:

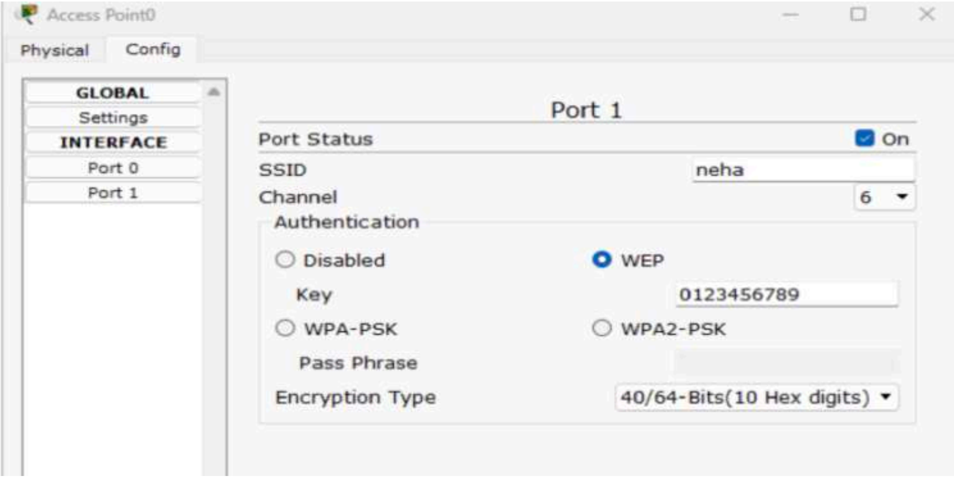
WLAN enables wireless n/w comm. It uses radio waves for connectivity. WLAN connects devices wirelessly within a local area. It eliminates the need for physical cables.

3/1/25

Screenshot of Topology:



Screenshot of Output:



Port 1

Port Status ☑ On

SSID neha

Channel 6

Authentication

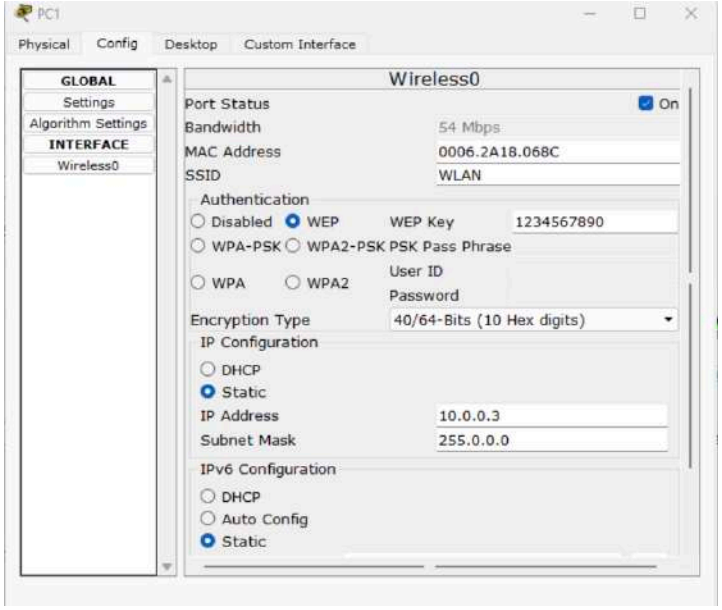
○ Disabled ● WEP

Key 0123456789

○ WPA-PSK ○ WPA2-PSK

Pass Phrase

Encryption Type 40/64-Bits(10 Hex digits)



Wireless0

Port Status ☑ On

Bandwidth 54 Mbps

MAC Address 0006.2A18.068C

SSID WLAN

Authentication

○ Disabled ● WEP WEP Key 1234567890

○ WPA-PSK ○ WPA2-PSK PSK Pass Phrase

○ WPA ○ WPA2 User ID

Password

Encryption Type 40/64-Bits (10 Hex digits)

IP Configuration

○ DHCP

● Static

IP Address 10.0.0.3

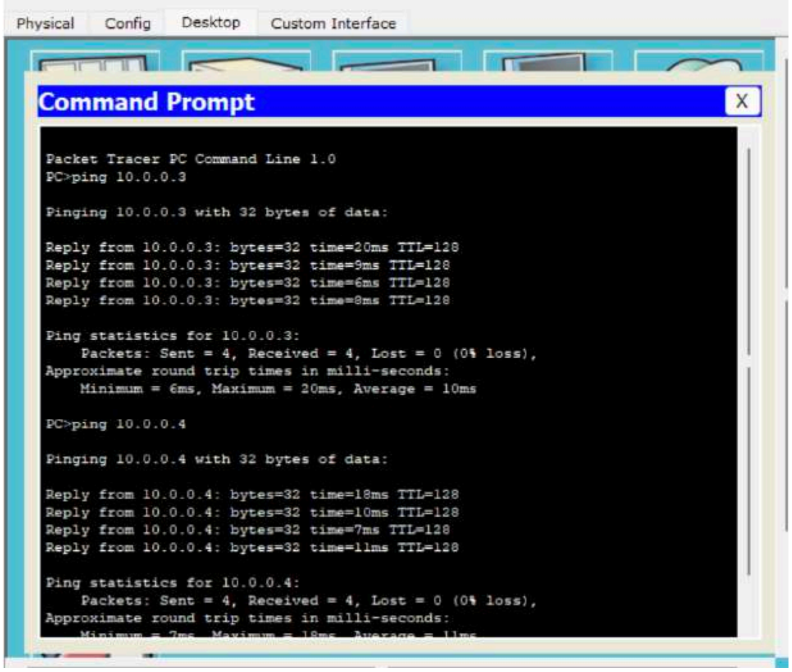
Subnet Mask 255.0.0.0

IPv6 Configuration

○ DHCP

○ Auto Config

● Static



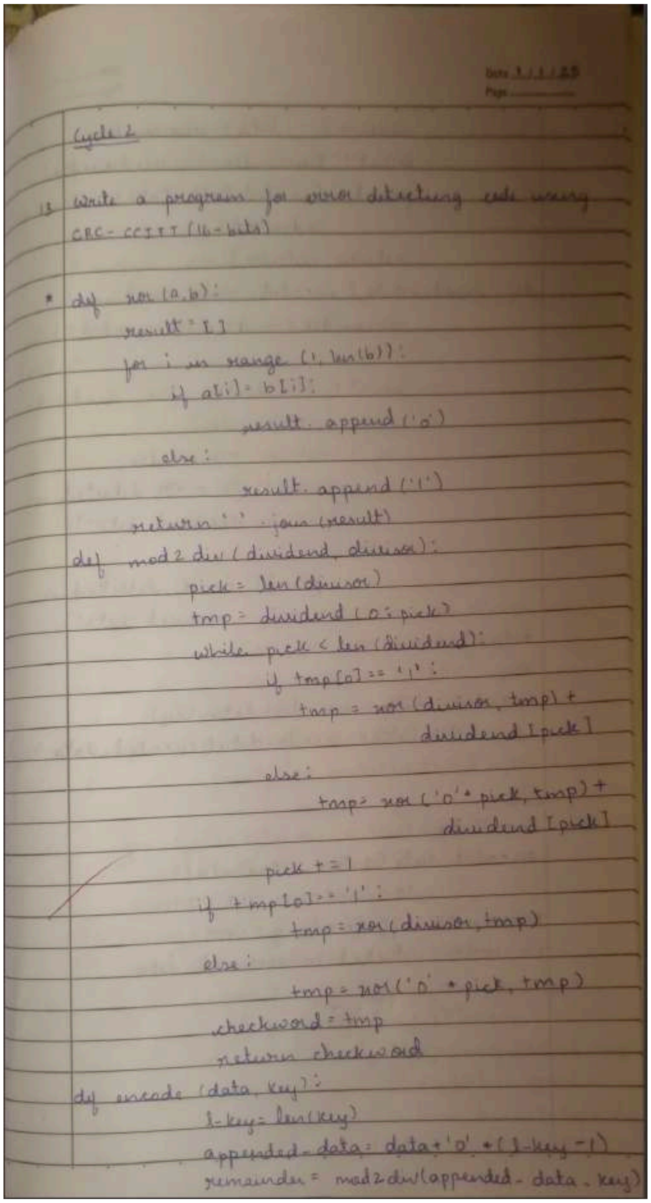
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=20ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=8ms 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 = 6ms, Maximum = 20ms, Average = 10ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=18ms TTL=128
Reply from 10.0.0.4: bytes=32 time=10ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms TTL=128
Reply from 10.0.0.4: bytes=32 time=11ms 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 = 7ms, Maximum = 18ms, Average = 11ms
```

CYCLE-2

Program 13

Q13: Write a program for error detecting code using CRC-CCITT (16-bits).

Observation:

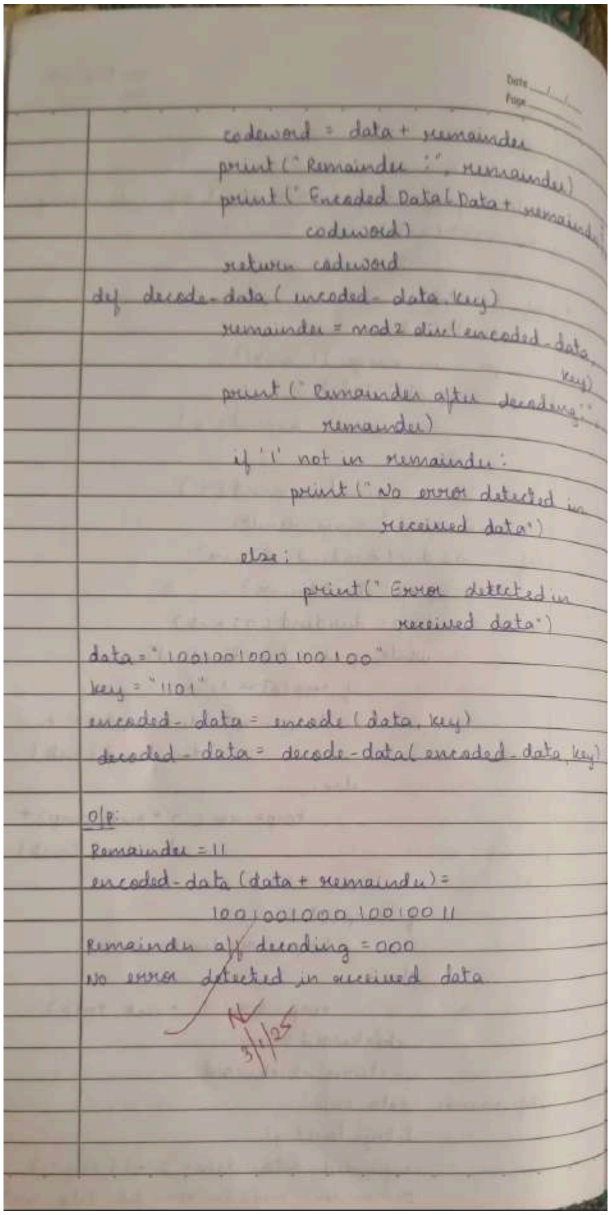


Date 3/1/25
Page

Cycle 2

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

```
* def xor (a,b):
      result = []
      for i in range (1, len(b)):
          if a[i] = b[i]:
                  result. append ('0')
          else:
                  result. append ('1')
      return ' ' . join (result)
def mod2 div (dividend, divisor):
          pick = len (divisor)
          tmp = dividend (0 : pick]
          while pick < len (dividend):
                  if tmp[0] == '1' :
                      tmp = xor (divisor, tmp) +
                                      dividend [pick]
                  else:
                      tmp = xor ('0' * pick, tmp) +
                                      dividend [pick]
                  pick += 1
          if tmp[0] == '1' :
                  tmp = xor(divisor, tmp)
          else:
                  tmp = xor('0' * pick, tmp)
          checkword = tmp
          return checkword
def encode (data, key):
          l-key = len(key)
          appended-data = data + '0' * (l-key - 1)
          remainder = mod2div(appended-data, key)
```



Date __/__/__
Page ____

```
            codeword = data + remainder
            print ("Remainder :", remainder)
            print (" Encoded Data (Data + remainder)",
                    codeword)
            return codeword
def decode-data ( encoded-data, key)
            remainder = mod2 div(encoded-data,
                                              key)
            print (" Remainder after decoding:",
                    remainder)
            if '1' not in remainder:
                    print (" No error detected in
                            received data")
            else;
                    print(" Error detected in
                            received data")
data = "1001001000100100"
key = "1101"
encoded-data = encode (data, key)
decoded-data = decode-data(encoded-data, key)
```

O/P:

```
Remainder = 11
encoded-data (data + remainder) =
                100100100010010011
Remainder aft decoding = 000
No error detected in received data
```

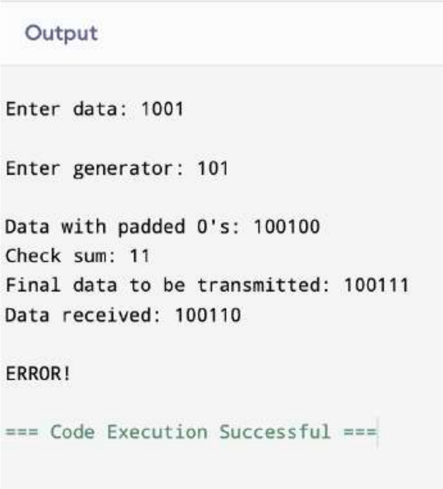
N
3/1/25

Code:

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

char data[28], gen_poly[10], check[28];
int data_len, i, j;

void XOR() {
   for (j = 0; j < N; j++) {
      check[j] = (check[j] == gen_poly[j]) ? '0' : '1';
   }
}

void crc() {
   for (i = 0; i < N; i++) {
      check[i] = data[i];
   }
   do {
      if (check[0] == '1') {
         XOR();
      }
      for (j = 0; j < N - 1; j++) {
         check[j] = check[j + 1];
      }
      check[j] = data[i++];
   } while (i <= data_len + N - 1);
}

void receiver() {
   printf("\nData received: ");
   scanf("%s", data);

   crc();
   for (i = 0; i < N - 1; i++) {
      if (check[i] == '1') {
         break;
      }
   }
   if (i < N - 1) {
      printf("\nERROR!");
   } else {
      printf("\nNO ERROR!");
   }
}
```

```
int main() {
    printf("\nEnter data: ");
    scanf("%s", data);
    printf("\nEnter generator: ");
    scanf("%s", gen_poly);

    data_len = strlen(data);

    // Append N-1 zeros to the data
    for (i = data_len; i < data_len + N - 1; i++) {
        data[i] = '0';
    }
    data[data_len + N - 1] = '\0'; // Null-terminate the string

    printf("\nData with padded 0's: %s", data);

    crc();

    printf("\nCheck sum: ");
    for (i = 0; i < N - 1; i++) {
        printf("%c", check[i]);
    }

    // Append checksum to data
    for (i = data_len; i < data_len + N - 1; i++) {
        data[i] = check[i - data_len];
    }
    data[data_len + N - 1] = '\0'; // Null-terminate the string

    printf("\nFinal data to be transmitted: %s", data);

    receiver();
    return 0;
}
```

Output:

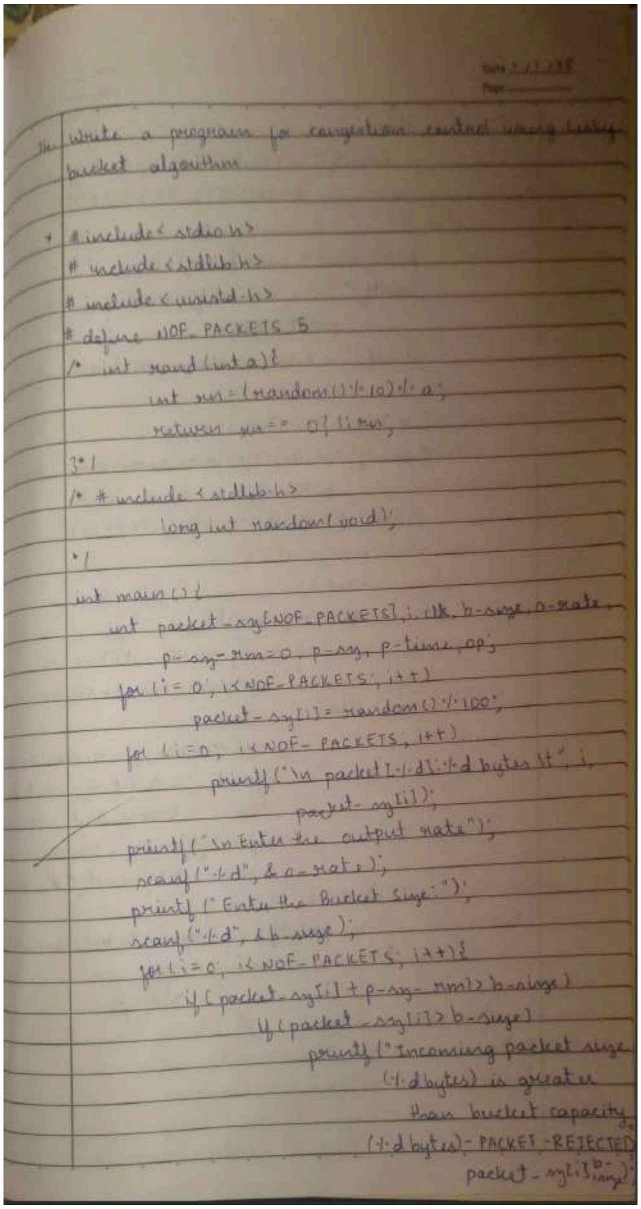


```
Enter data: 1001

Enter generator: 101

Data with padded 0's: 100100
Check sum: 11
Final data to be transmitted: 100111
Data received: 100110

ERROR!

=== Code Execution Successful ===
```

Program 14

Q14: Write a program for congestion control using Leaky bucket algorithm.

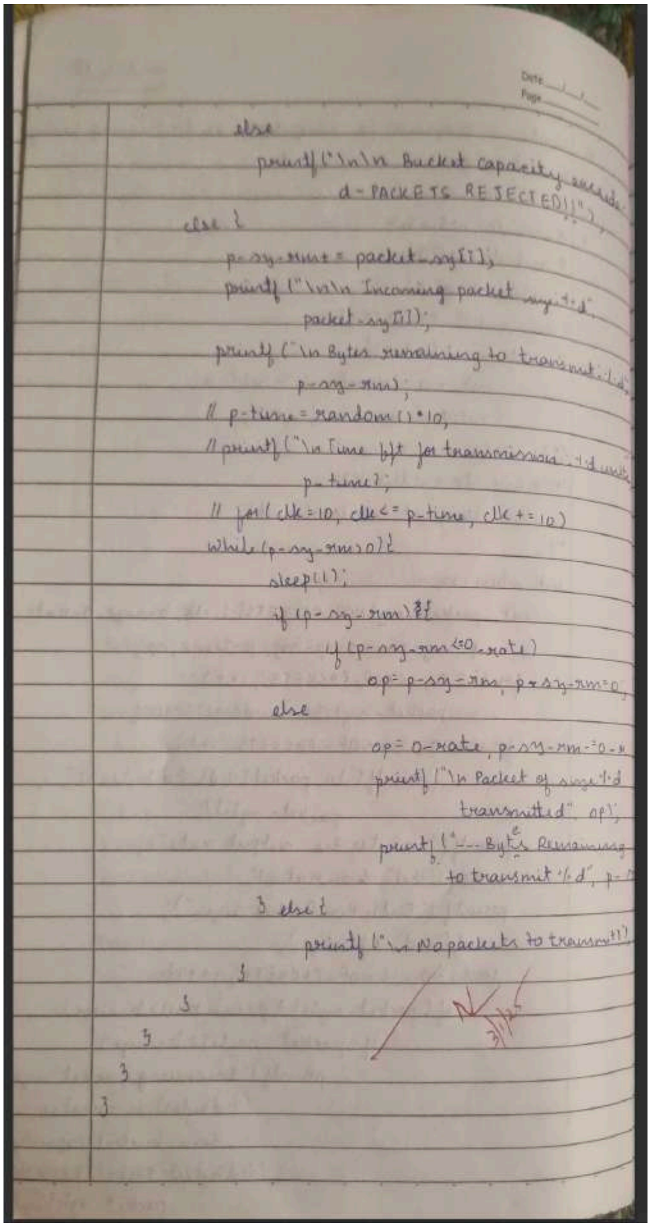
Observation:



Date 1/11/25

Q14 Write a program for congestion control using leaky bucket algorithm

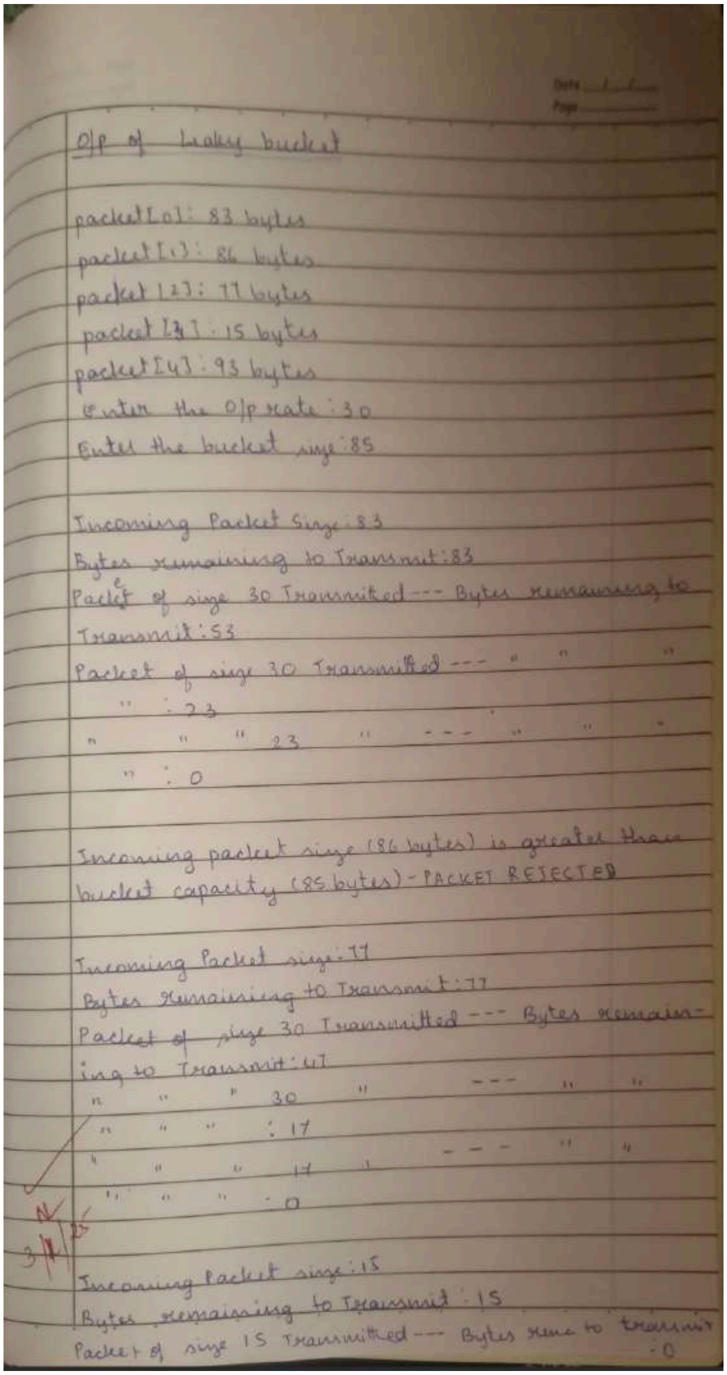
```
#include< stdio.h>
# include <stdlib.h>
# include < unistd.h>
# define NOF_PACKETS 5
/*  int rand (int a){
        int rn=(random()%10)%a;
        return rn== 0? 1:rn;
}*/
/* # include < stdlib.h>
        long int random(void);
*/
int main(){
    int packet_sz[NOF_PACKETS], i, clk, b-size, o-rate,
        p-sz-rm=0, p-sz, p-time, op;
    for (i= 0; i<NOF-PACKETS; i++)
            packet-sz[i]= random()%100;
    for (i=0; i<NOF- PACKETS; i++)
            printf("\n packet[%d]:%d bytes \t", i,
                        packet-sz[i]);
    printf("\n Enter the output rate");
    scanf("%d", &o-rate);
    printf(" Enter the Bucket Size:");
    scanf("%d", &b-size);
    for(i=0; i< NOF_PACKETS; i++){
        if (packet_sz[i] + p-sz- rm)> b-size)
            if (packet_sz[i]> b-size)
                printf("Incoming packet size
                        (%dbytes) is greater
                        than bucket capacity
                (%d bytes)- PACKET -REJECTED",
                        packet_sz[i], b-size);
```



Date __/__/__

```
        else
            printf("\n\n Bucket capacity exceeded
                    d - PACKETS REJECTED!!");
        else {
            p-sz-rm+= packet_sz[i];
            printf("\n\n Incoming packet size: %d",
                    packet-sz[i]);
            printf("\n Bytes remaining to transmit: %d",
                    p-sz-rm);
            // p-time= random()*10;
            // printf("\n Time left for transmission: %d units",
                    p_time);
            // for( clk=10; clk<= p_time; clk+=10)
            while(p-sz-rm>0){
                sleep(1);
                if (p-sz-rm){
                    if (p-sz-rm<=o-rate)
                        op= p-sz-rm; p-sz-rm=0;
                else
                        op= o-rate, p-sz-rm-=o-rate;
                        printf("\n Packet of size %d
                                transmitted", op);
                        printf("---Bytes Remaining
                                to transmit %d", p-
                } else {
                    printf("\n No packets to transmit!");
                }
            }
        }
    }
}
```

3/1/25



O/p of Leaky bucket

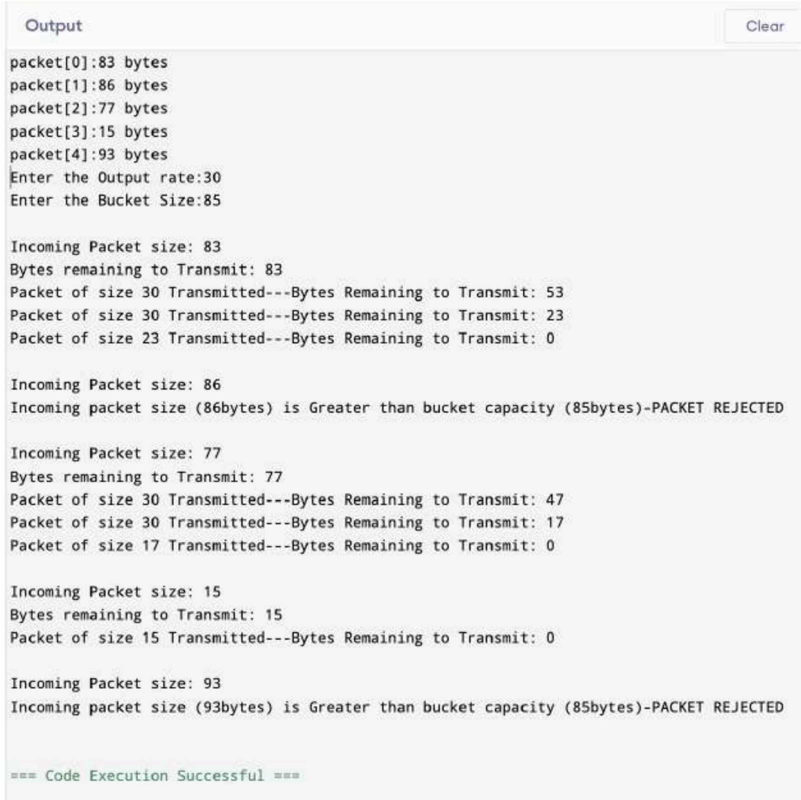
```
packet[0]: 83 bytes
packet[1]: 86 bytes
packet[2]: 77 bytes
packet[3]: 15 bytes
packet[4]: 93 bytes
Enter the O/p rate: 30
Enter the bucket size: 85

Incoming Packet Size: 83
Bytes remaining to Transmit: 83
Packet of size 30 Transmitted --- Bytes remaining to
Transmit: 53
Packet of size 30 Transmitted --- "     "      "
    "   : 23
 "      "     " 23    "       ---   "     "     "
    "   : 0

Incoming packet size (86 bytes) is greater than
bucket capacity (85 bytes) - PACKET REJECTED

Incoming Packet size: 77
Bytes Remaining to Transmit: 77
Packet of size 30 Transmitted --- Bytes remain-
ing to Transmit: 47
 "     "     "  30    "           ---   "    "
  "    "    "   : 17
 "     "     "  17    "         - - -   "    "
  "    "    "   : 0

3/4/25

Incoming Packet size: 15
Bytes remaining to Transmit: 15
Packet of size 15 Transmitted --- Bytes rem to transmit
                                               : 0
```

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h> // for sleep function

#define NOF_PACKETS 5

// Function to simulate sending packets
void send_packet(int packet_size, int output_rate) {
    while (packet_size > 0) {
        int sent = (packet_size < output_rate) ? packet_size : output_rate;
        printf("Packet of size %d Transmitted---", sent);
        packet_size -= sent;
        printf("Bytes Remaining to Transmit: %d\n", packet_size);
        sleep(1);  // Simulate time delay between packets
    }
}

int main() {
    int output_rate, bucket_size, incoming_packet_size;
    int i, packet_size[NOF_PACKETS];

    // Input number of packets and their sizes
    for(i = 0; i < NOF_PACKETS; i++) {
        packet_size[i] = rand() % 100;  // Random packet size between 0 and 99
        printf("packet[%d]:%d bytes\n", i, packet_size[i]);
    }

    printf("Enter the Output rate:");
    scanf("%d", &output_rate);

    printf("Enter the Bucket Size:");
    scanf("%d", &bucket_size);

    for(i = 0; i < NOF_PACKETS; i++) {
        printf("\nIncoming Packet size: %d\n", packet_size[i]);
        if(packet_size[i] > bucket_size) {
            printf("Incoming packet size (%dbytes) is Greater than bucket capacity
(%dbytes)-PACKET REJECTED\n", packet_size[i], bucket_size);
```

```
            continue;
        }

        printf("Bytes remaining to Transmit: %d\n", packet_size[i]);
        send_packet(packet_size[i], output_rate);
    }

    return 0;
}
```

Output:

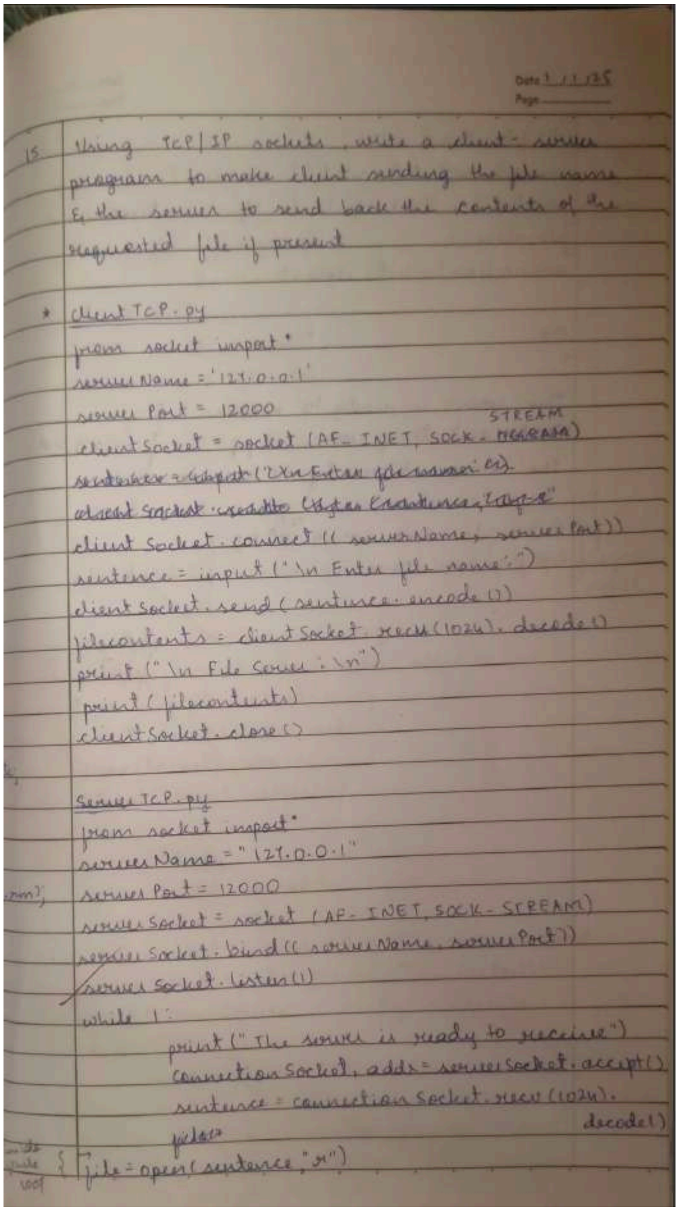


```
packet[0]:83 bytes
packet[1]:86 bytes
packet[2]:77 bytes
packet[3]:15 bytes
packet[4]:93 bytes
Enter the Output rate:30
Enter the Bucket Size:85

Incoming Packet size: 83
Bytes remaining to Transmit: 83
Packet of size 30 Transmitted---Bytes Remaining to Transmit: 53
Packet of size 30 Transmitted---Bytes Remaining to Transmit: 23
Packet of size 23 Transmitted---Bytes Remaining to Transmit: 0

Incoming Packet size: 86
Incoming packet size (86bytes) is Greater than bucket capacity (85bytes)-PACKET REJECTED

Incoming Packet size: 77
Bytes remaining to Transmit: 77
Packet of size 30 Transmitted---Bytes Remaining to Transmit: 47
Packet of size 30 Transmitted---Bytes Remaining to Transmit: 17
Packet of size 17 Transmitted---Bytes Remaining to Transmit: 0

Incoming Packet size: 15
Bytes remaining to Transmit: 15
Packet of size 15 Transmitted---Bytes Remaining to Transmit: 0

Incoming Packet size: 93
Incoming packet size (93bytes) is Greater than bucket capacity (85bytes)-PACKET REJECTED


=== Code Execution Successful ===
```

Program 15

Q15: Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

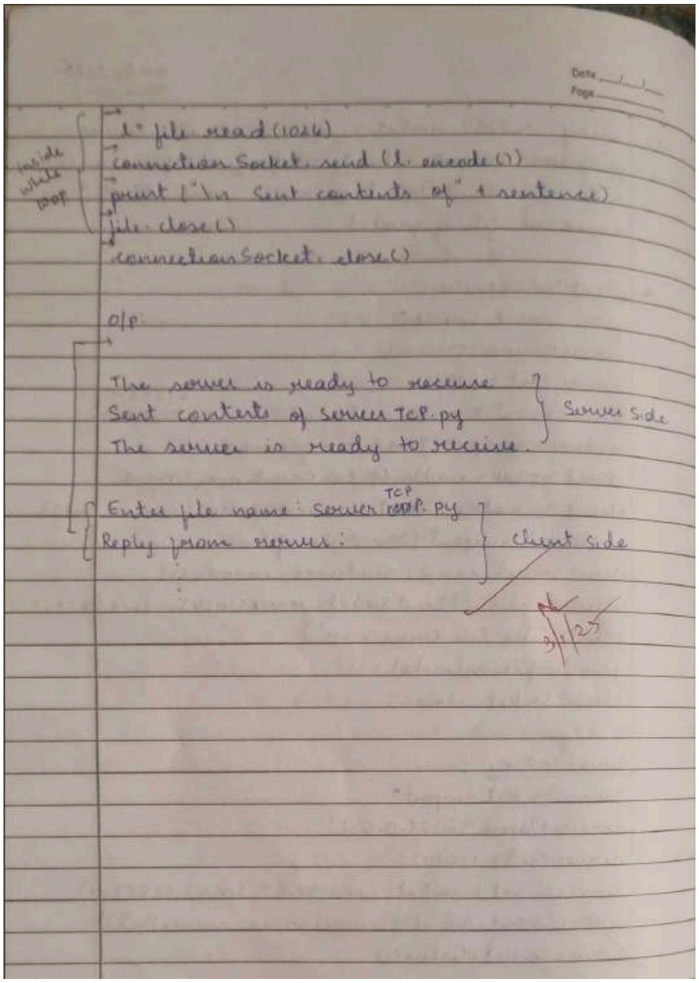
Observation:



Date 1/1/25

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

```
* clientTCP.py
from socket import*
serverName = '127.0.0.1'
server Port = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
~~sentence = input('\n Enter file name: ')~~
~~client socket.sendto(bytes(sentence, 'utf-8'))~~
client Socket.connect ((serverName, serverPort))
sentence = input ("\n Enter file name: ")
client Socket.send(sentence.encode())
filecontents = client Socket.recv(1024).decode()
print ("\n File Server: \n")
print (filecontents)
clientSocket.close()

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



Date __/__/__

```
inside while loop {
    l = file.read(1024)
    connection Socket.send (l.encode())
    print ("\n Sent contents of" + sentence)
    file.close()
connectionSocket.close()
```

O/P.

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

Enter file name: server TCP.py      ]
Reply from server:                  }  Client side
```

3/1/25

Code:

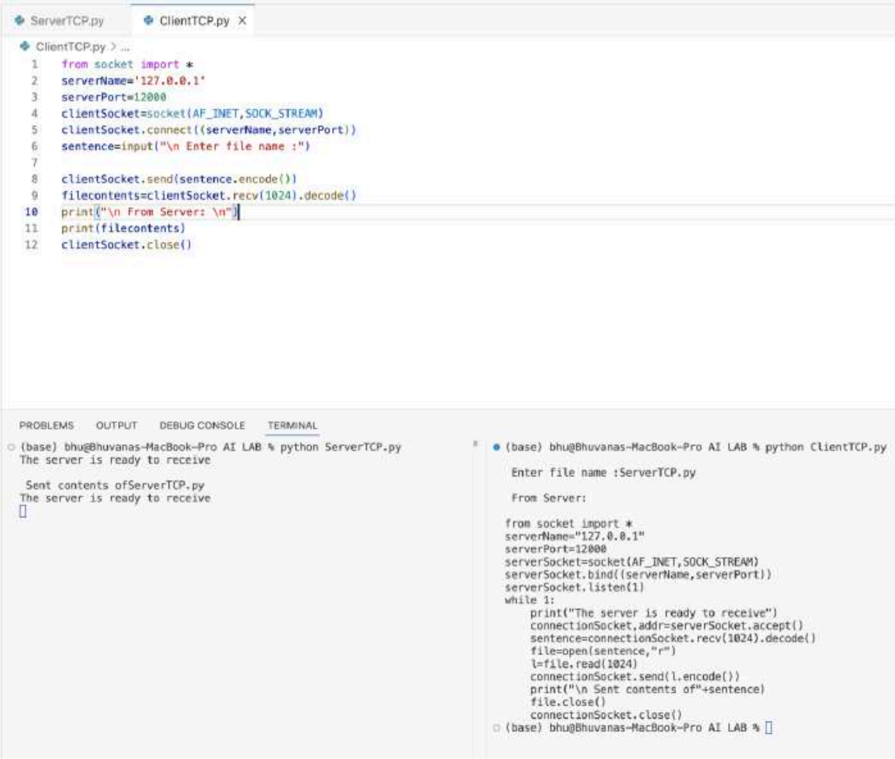
ServerTCP.py

```
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("\n Sent contents of"+sentence)
   file.close()
   connectionSocket.close()
```

ClientTCP.py

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

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

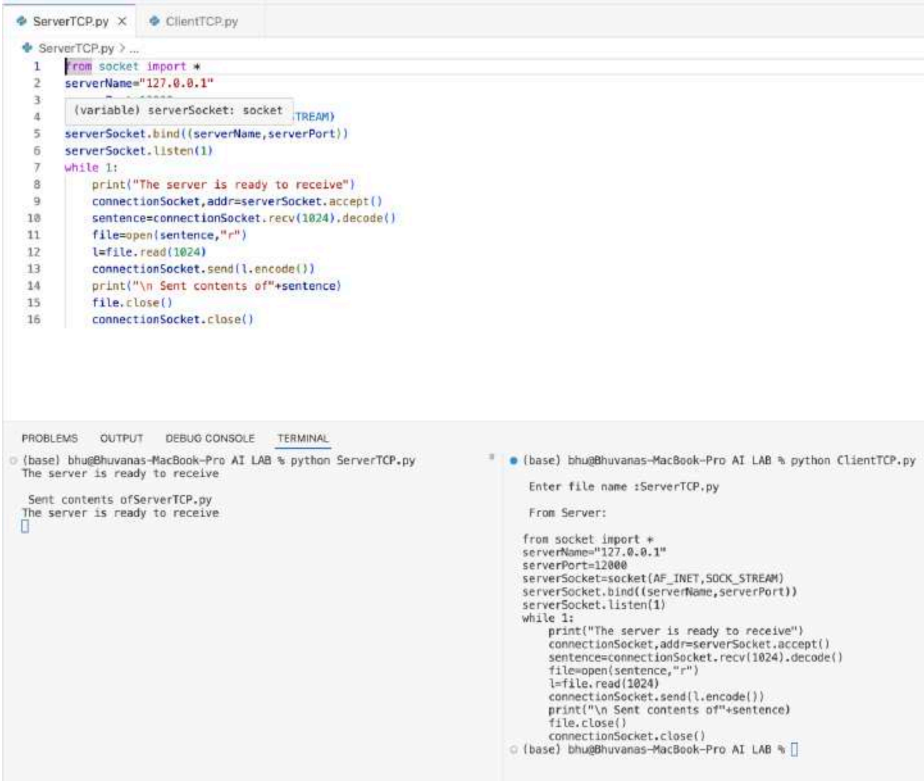
Output:



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

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

```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ServerTCP.py
The server is ready to receive

 Sent contents ofServerTCP.py
The server is ready to receive
```

```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ClientTCP.py

 Enter file name :ServerTCP.py

 From Server:

from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents of"+sentence)
    file.close()
    connectionSocket.close()
(base) bhu@Bhuvanas-MacBook-Pro AI LAB %
```



```
from socket import *
serverName="127.0.0.1"

(variable) serverSocket: socket   TREAM)
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("\n Sent contents of"+sentence)
    file.close()
    connectionSocket.close()
```

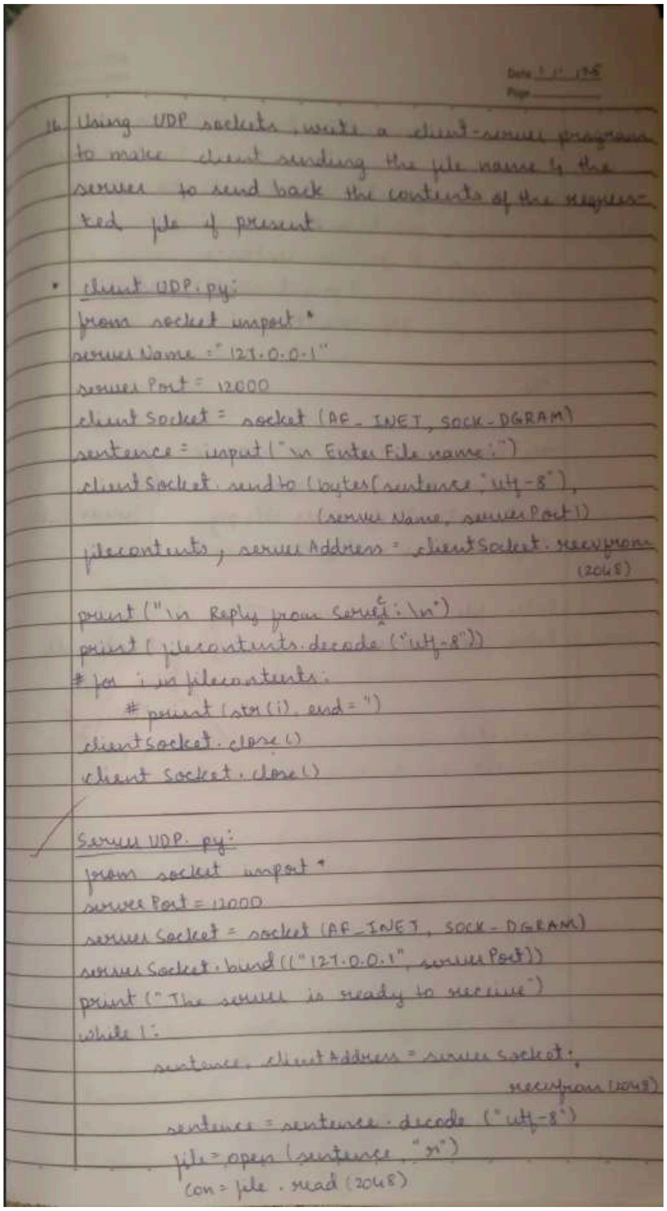
```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ServerTCP.py
The server is ready to receive

 Sent contents ofServerTCP.py
The server is ready to receive
```

```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ClientTCP.py

 Enter file name :ServerTCP.py

 From Server:

from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents of"+sentence)
    file.close()
    connectionSocket.close()
(base) bhu@Bhuvanas-MacBook-Pro AI LAB %
```

Program 16

Q16 : Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

Observation:



Date 1/1/25

Page

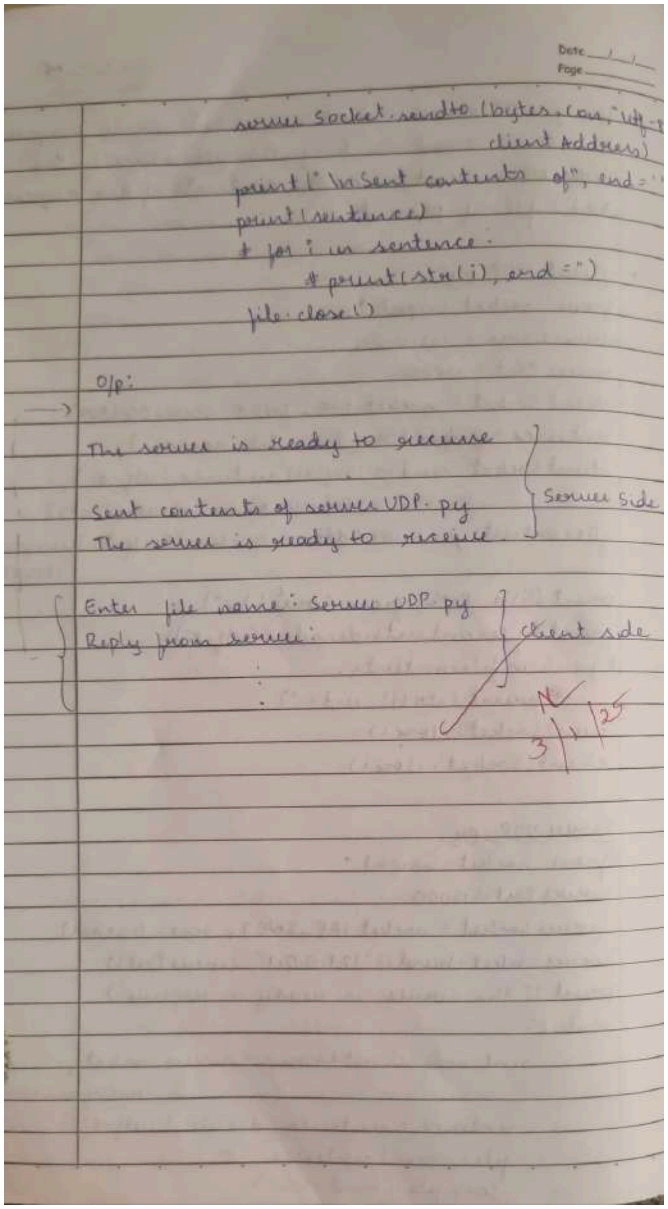
16 Using UDP sockets, write a client-server program to make client sending the file name & the server to send back the contents of the requested file if present

- 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 (filecontents.decode ("utf-8"))
# for i in filecontents:
    # print (str(i), end = "")
clientSocket.close()
client Socket.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)
```



Date / /

Page

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

O/p:

```
The server is ready to receive

Sent contents of server UDP.py
The server is ready to receive
```
} Server Side

```
Enter file name: Server UDP.py
Reply from server:
        :
        :
```
} Client side

3/1/25

Code:

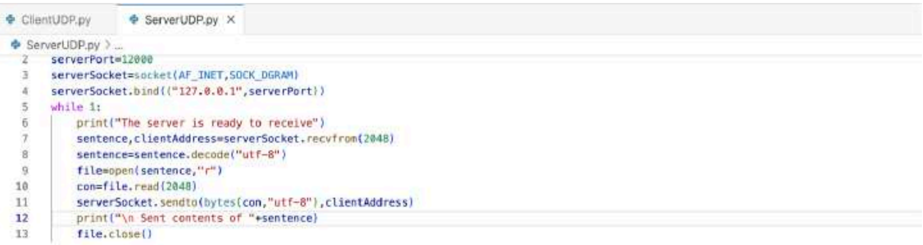
ServerUDP.py

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

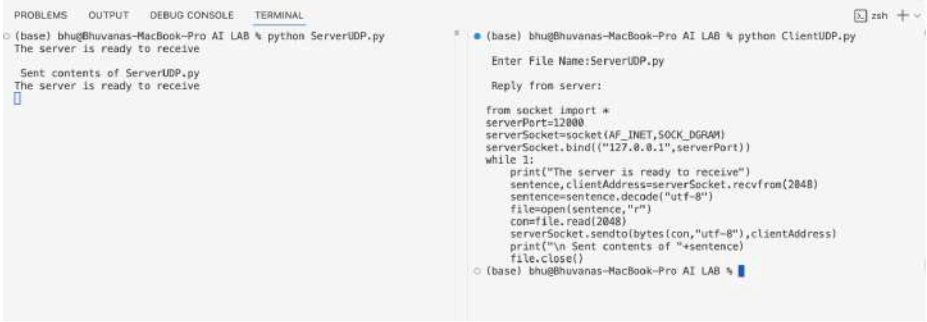
ClientUDP.py

```
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(filecontents.decode("utf-8"))
clientSocket.close()
```

Output:

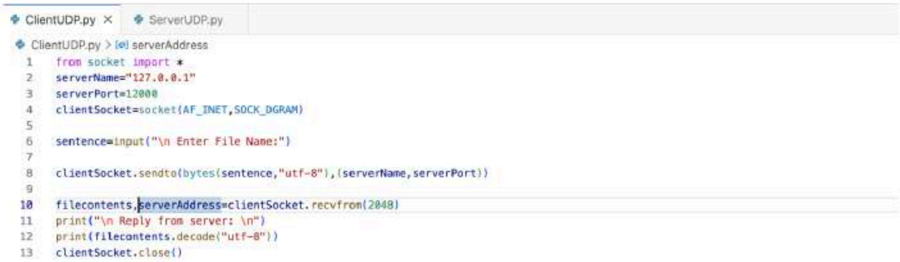


```
serverPort=12000
serverSocket=socket(AF_INET,SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",serverPort))
while 1:
    print("The server is ready to receive")
    sentence,clientAddress=serverSocket.recvfrom(2048)
    sentence=sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)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

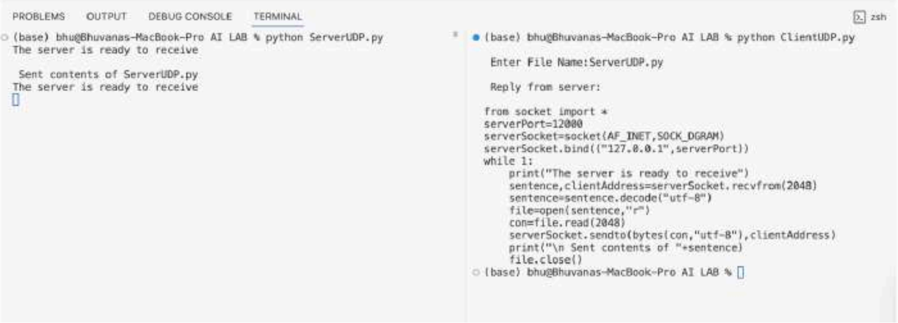


```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ServerUDP.py
The server is ready to receive

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

```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ClientUDP.py

 Enter File Name:ServerUDP.py

 Reply from server:

from socket import *
serverPort=12000
serverSocket=socket(AF_INET,SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",serverPort))
while 1:
    print("The server is ready to receive")
    sentence,clientAddress=serverSocket.recvfrom(2048)
    sentence=sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
    print("\n Sent contents of "+sentence)
    file.close()
(base) bhu@Bhuvanas-MacBook-Pro AI LAB %
```



```
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(filecontents.decode("utf-8"))
clientSocket.close()
```



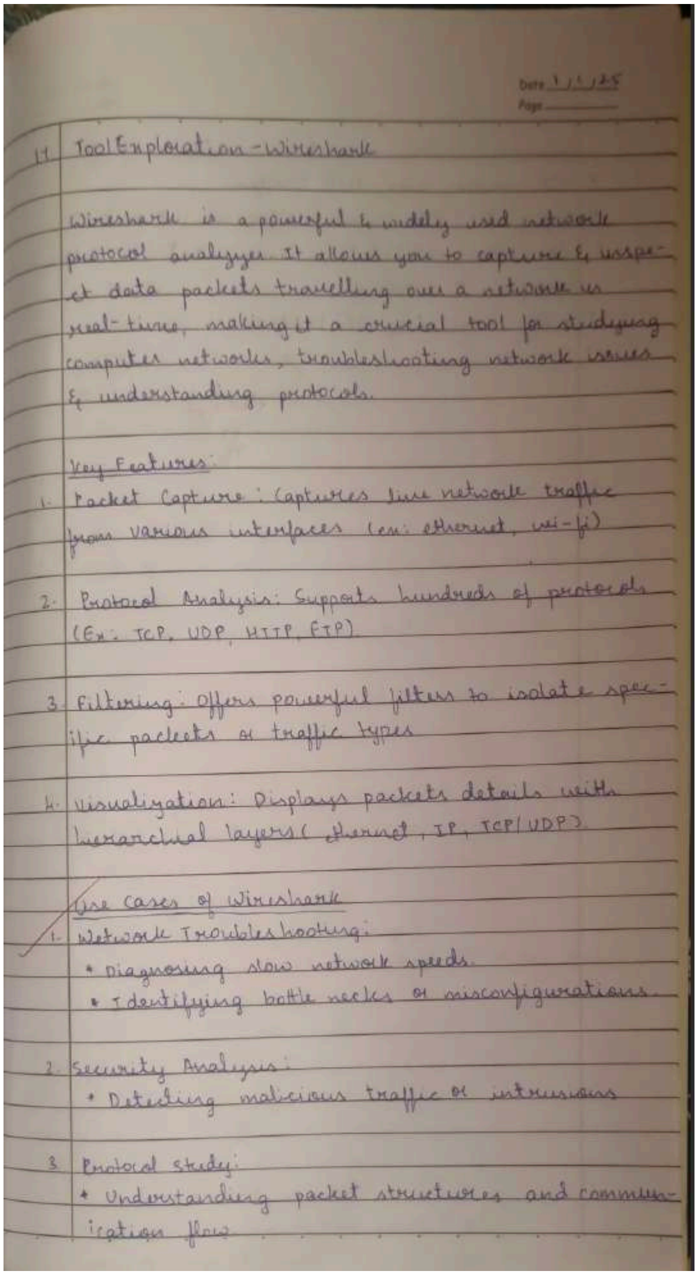
```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ServerUDP.py
The server is ready to receive

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

```
(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ClientUDP.py

 Enter File Name:ServerUDP.py

 Reply from server:

from socket import *
serverPort=12000
serverSocket=socket(AF_INET,SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",serverPort))
while 1:
    print("The server is ready to receive")
    sentence,clientAddress=serverSocket.recvfrom(2048)
    sentence=sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
    print("\n Sent contents of "+sentence)
    file.close()
(base) bhu@Bhuvanas-MacBook-Pro AI LAB %
```

Program 17

Q17: Tool Exploration-Wireshark

Observation



Date 1/1/25
Page

17 Tool Exploration-Wireshark

Wireshark is a powerful & widely used network protocol analyzer. It allows you to capture & inspect data packets travelling over a network in real-time, making it a crucial tool for studying computer networks, troubleshooting network issues & understanding protocols.

Key Features:

1. Packet Capture: Captures live network traffic from various interfaces (ex: ethernet, wi-fi)

2. Protocol Analysis: Supports hundreds of protocols (Ex: TCP, UDP, HTTP, FTP)

3. Filtering: Offers powerful filters to isolate specific packets or traffic types

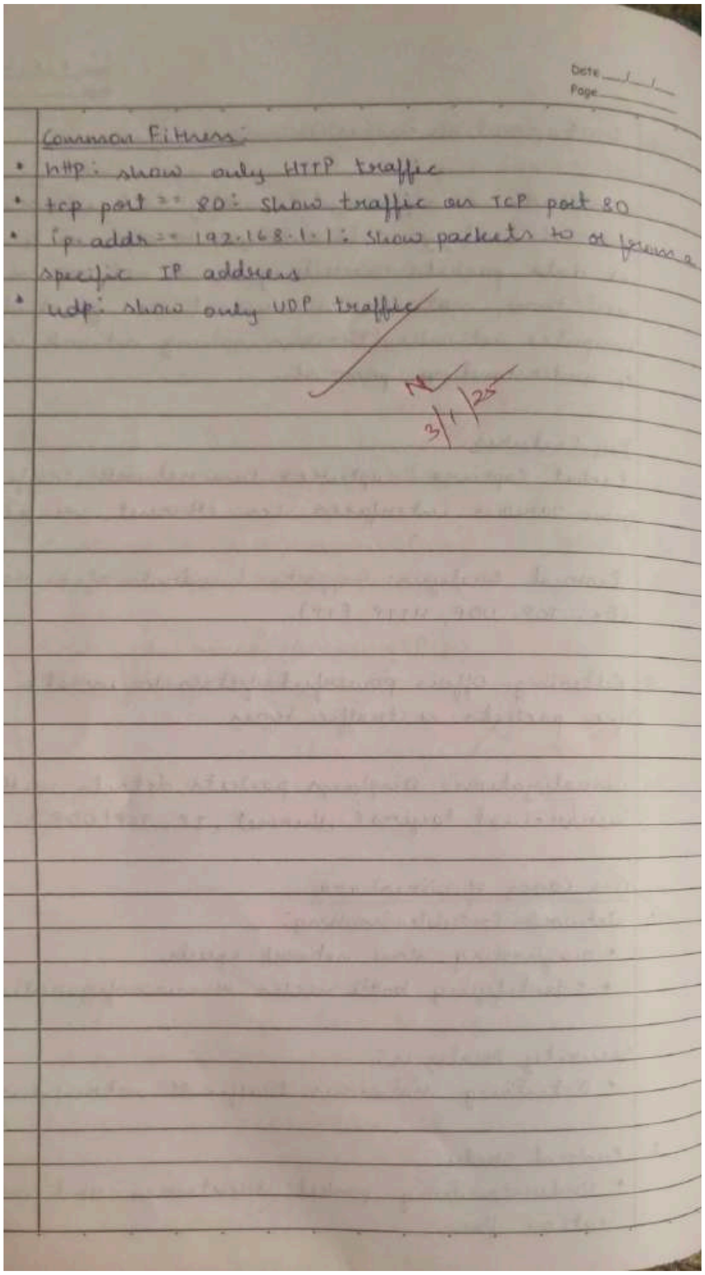
4. Visualization: Displays packets details with hierarchical layers (ethernet, IP, TCP/UDP)

Use cases of Wireshark

1. Network Troubleshooting:
   - Diagnosing slow network speeds.
   - Identifying bottle necks or misconfigurations.

2. Security Analysis:
   - Detecting malicious traffic or intrusions

3. Protocol study:
   - Understanding packet structures and communication flow



Date __/__/__
Page

Common Filtress:
- http: show only HTTP traffic
- tcp port == 80: Show traffic on TCP port 80
- ip.addr == 192.168.1.1: Show packets to or from a specific IP address
- udp: show only UDP traffic

3/1/25