

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Networks

Submitted by

Chris D T (1BM22CS080)

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 **Chris D T(1BM22CS080)**, 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.

Sandhya A Kulkarni Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
---	--

Index

Cycle - I

Sl.No.	Date	Experiment Title	Page No.
1	01/10/2024	Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.	4
2	08/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	6
3	22/10/2024	Configure default route, static route to the router	12
4	29/10/2024	Configure DHCP within a LAN and outside LAN.	15
5	29/10/2024	Configure RIP routing Protocol in Routers.	19
6	12/11/2024	Configure OSPF Routing Protocol	23
7	12/11/2024	Demonstrate the TTL/ Life of a Packet	25
8	18/12/2024	Configure Web Server & DNS	31
9	18/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	33

10	18/12/2024	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	37
11	18/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	41
12	18/12/2024	To construct a WLAN and make the nodes communicate wirelessly	45

Cycle-II

Sl.No.	Date	Experiment Title	Page No.
13	03/12/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	50
14	17/12/2024	Write a program for congestion control using Leaky bucket algorithm	52
15	24/12/2024	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	53
16	24/12/2024	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	54
17		Tool Exploration - Wireshark	55

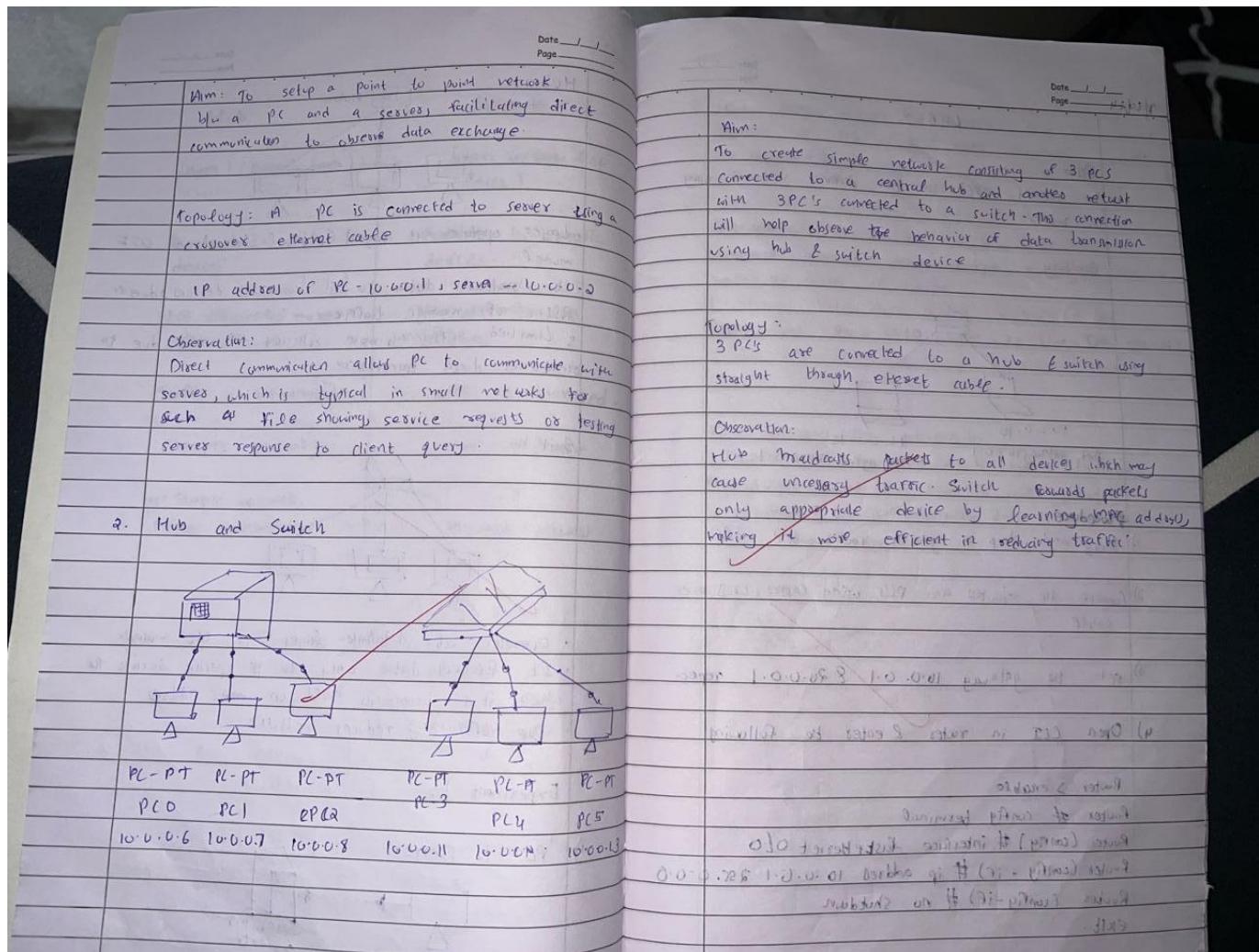
Github Link:

<https://github.com/chrisdt777/CN-LAB>

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:



Screenshot of Topology:



Screenshot of Output:

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of a ping command from PC0 to PC3. The output is as follows:

```
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=4ms TTL=128

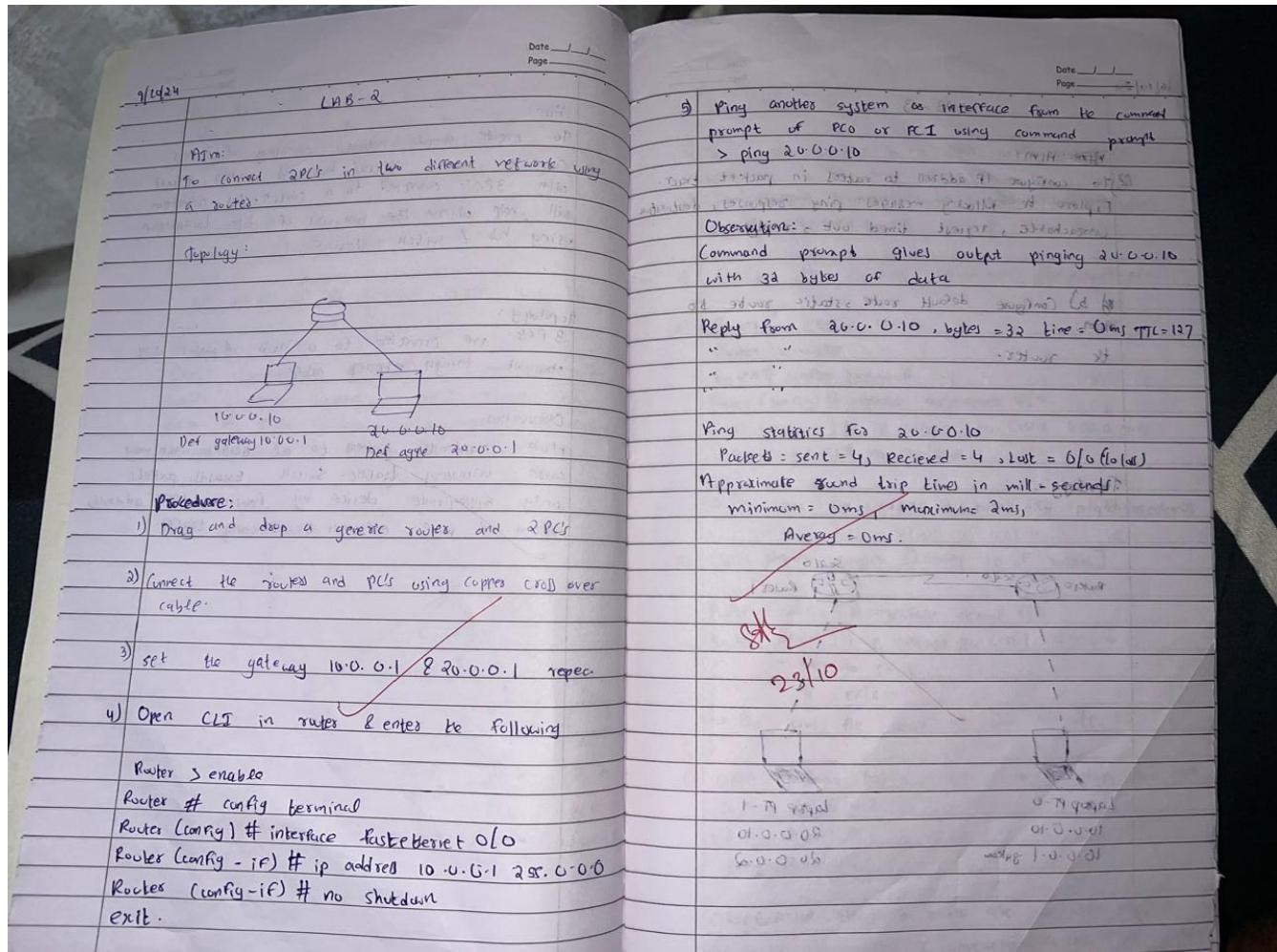
Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 4ms, Average = 4ms

PC>
```

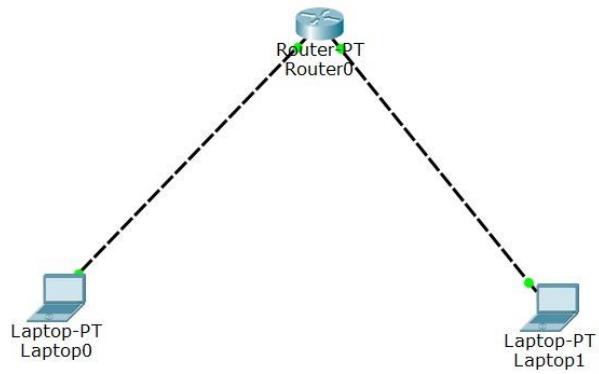
Program 2

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

Observation:



Screenshot of Topology:



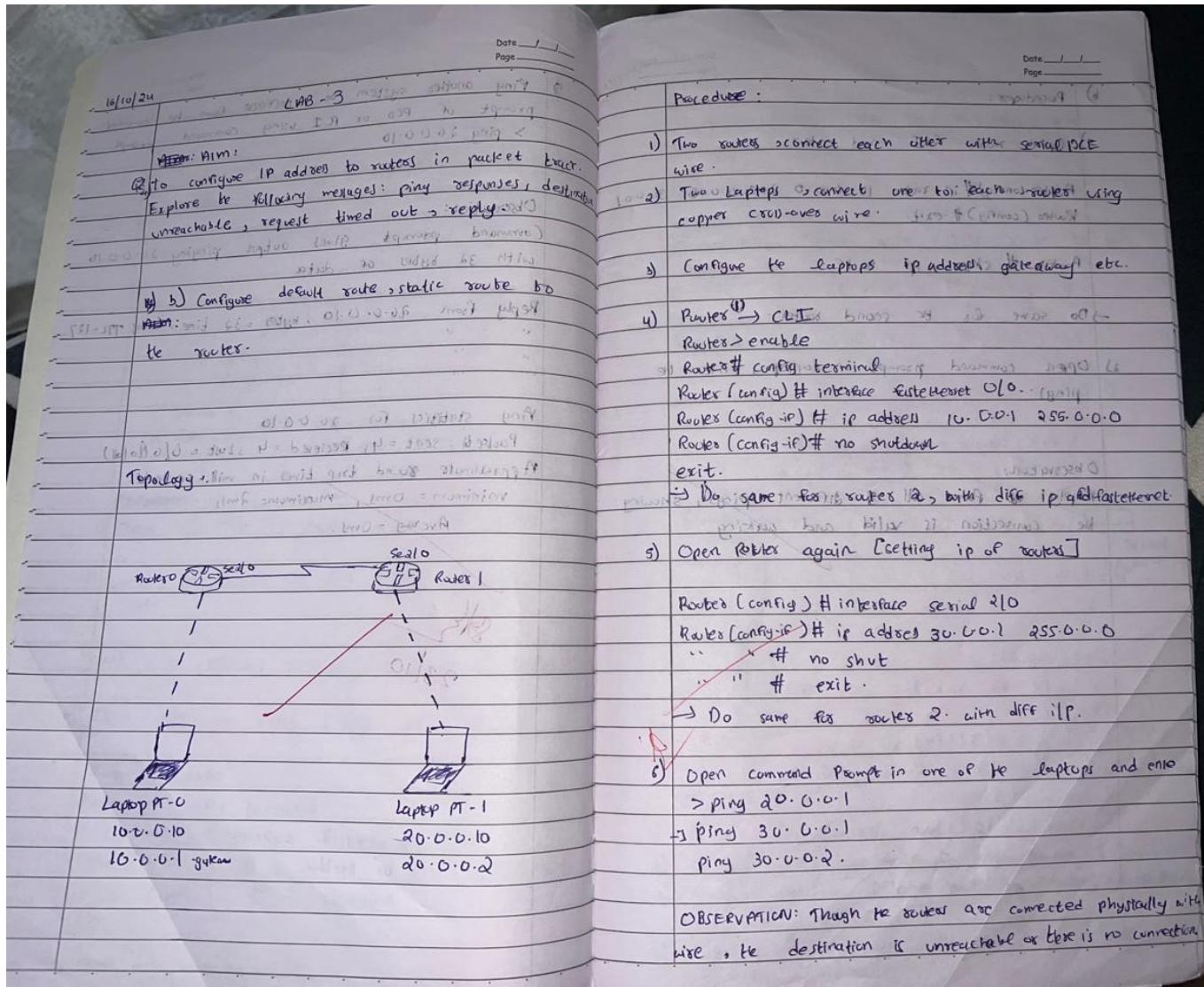
Screenshot of Output:

The screenshot shows a Cisco Packet Tracer simulation environment. At the top, there's a navigation bar with tabs: Physical, Config, Desktop, and Custom Interface. The Desktop tab is selected. Below the navigation bar is a toolbar with icons for various functions. A main window titled "Command Prompt" is open, showing the output of several ping commands. The output is as follows:

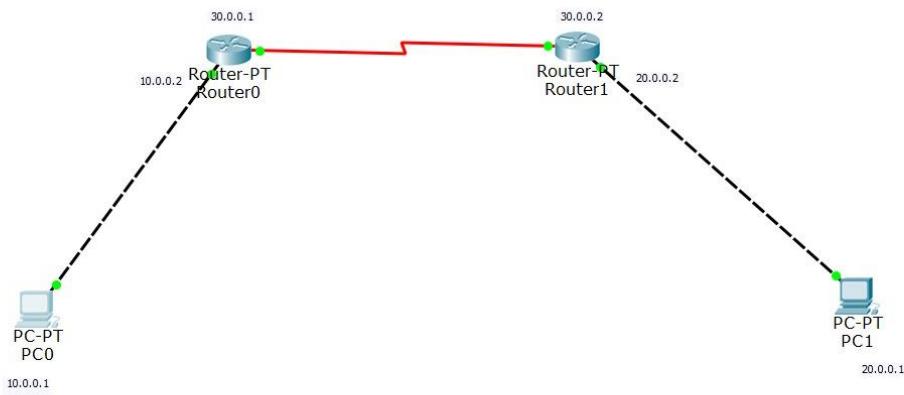
```
Pinging 20.0.0.1 with 32 bytes of data:  
Request timed out.  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
PC>ping 20.0.0.1  
  
Pinging 20.0.0.1 with 32 bytes of data:  
  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
PC>
```

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

Observation:



Screenshot of Topology:



Screenshot of Output:

PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25%  
loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 7ms, Maximum = 10ms, Average = 8ms  
  
PC>ping 20.0.0.1  
  
Pinging 20.0.0.1 with 32 bytes of data:  
  
Reply from 20.0.0.1: bytes=32 time=7ms TTL=126  
Reply from 20.0.0.1: bytes=32 time=6ms TTL=126  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=126  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=126  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0%  
loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 1ms, Maximum = 7ms, Average = 3ms  
  
PC>
```

Router0

Physical Config CLI

IOS Command Line Interface

```
Router(config-if)#  
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state  
to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface  
FastEthernet0/0, changed state to up  
  
Router(config-if)#exit  
Router(config)#interface Serial2/0  
Router(config-if)#  
%LINK-5-CHANGED: Interface Serial2/0, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface  
Serial2/0, changed state to up  
  
Router(config-if)#exit  
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.2  
Router(config)#exit  
Router#  
%SYS-5-CONFIG_I: Configured from console by console
```

Router1

Physical Config CLI

IOS Command Line Interface

```
Router(config-if)#  
%LINK-5-CHANGED: Interface FastEthernet0/0  
Router(config-if)#no shutdown  
  
Router(config-if)#  
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state  
to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface  
FastEthernet0/0, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface  
Serial2/0, changed state to up  
  
Router(config-if)#exit  
Router(config)#interface Serial2/0  
Router(config-if)#exit  
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1  
Router(config)#exit  
Router#  
%SYS-5-CONFIG_I: Configured from console by console
```

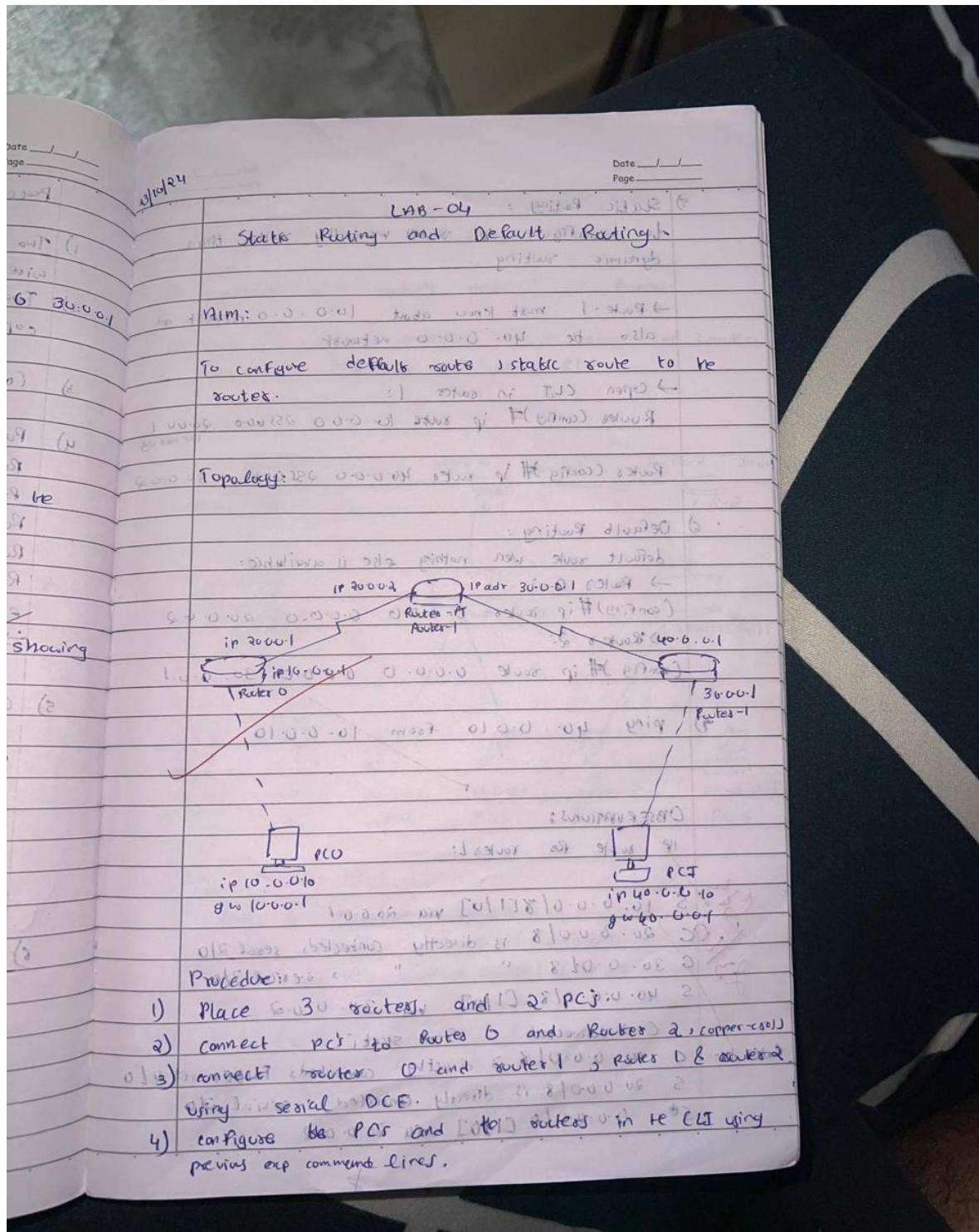
Copy Paste

Copy Paste

Program 3

Q3: Configure default route, static route to the Router

Observation:



20/11/21

5) Static Routing : ~~192.168.1.1~~
to configure ip routes manually rather than
dynamic routing.

→ Router-1 must know about 10.0.0.0 network
also the 40.0.0.0 network

→ Open CLI in Router 1:
Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
 [hop address]

Router(config)# ip router 40.0.0.0 255.0.0.0 192.168.0.1

6) Default Routing :

default route when nothing else is available:

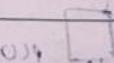
→ Router 0: ~~192.168.1.1~~
(config)# ip routes 0.0.0.0 0.0.0.0 20.0.0.2

→ Router 2:
(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

7) ping 40.0.0.10 from 10.0.0.10

OBSERVATIONS:

IP table for Router 1:



0.0.0.0/8

S 10.0.0.0/8 [1/0] via 20.0.0.1 1.0.0.1 w/o

C 20.0.0.0/8 " " " is directly connected serial 2/0

G 30.0.0.0/8 " " " , serial 3/0

S 40.0.0.0/8 [1/0] via 30.0.0.2 2.0.0.8

C → connected S → static (no form)

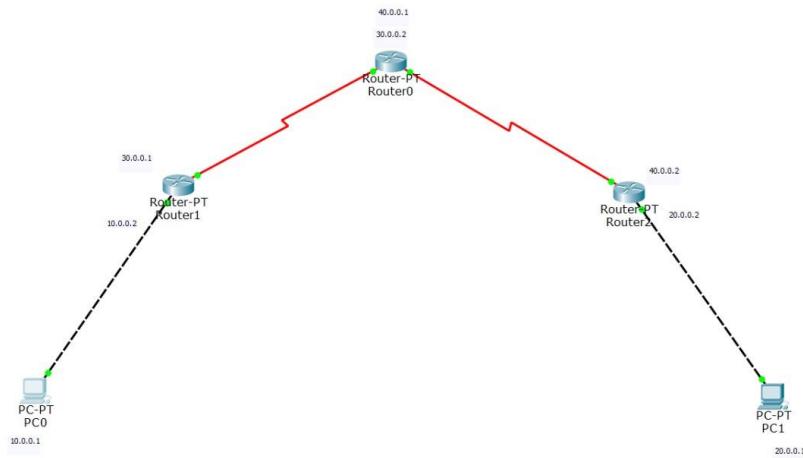
Router 0: C 10.0.0.0/8 is directly connected FastEthernet 0/0

S 20.0.0.0/8 is directly connected serial 2/0

S* 40.0.0.0/8 [1/0] via 20.0.0.2 2.0.0.8

2nd connection via serial

Screenshot of Topology:



Screenshot of Output:

```
PC0
```

```
Physical Config Desktop Custom Interface
```

```
Command Prompt X
```

```
Request timed out.
```

```
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=8ms TTL=125
Reply from 20.0.0.1: bytes=32 time=13ms TTL=125
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125

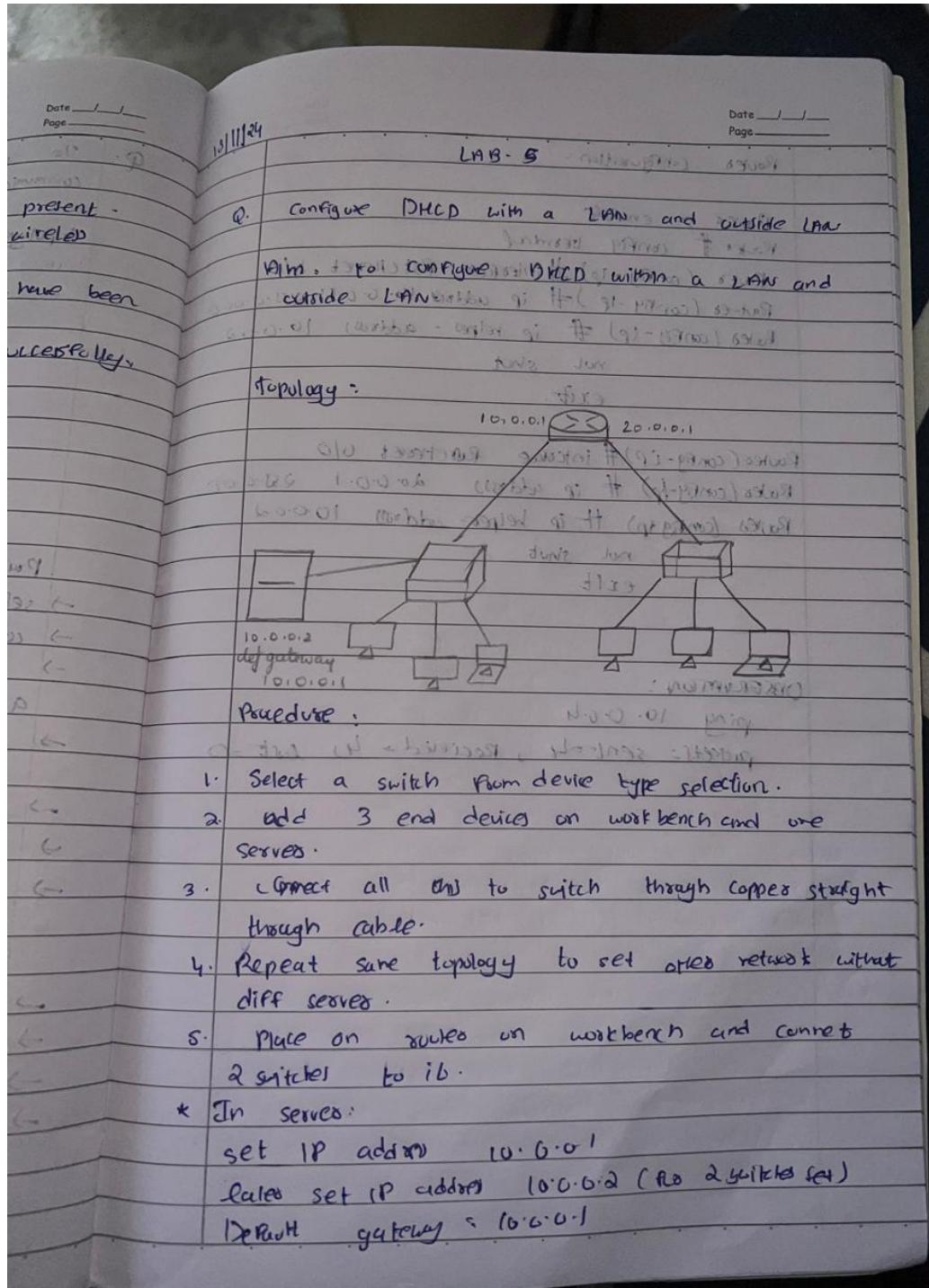
Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 13ms, Average = 10ms

PC>
```

Program 4

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

Observation:



Router configuration

```
Router > enable
Router # config terminal
Router (config) # interface fastethernet 4/0.
Router (config-ip) # ip address 10.0.1.255.0.0.0
Router (config-ip) # ip helper-address 10.0.0.2
      nol shut
      exit.
```

```
Router(config-ip) # interface fastethernet 0/0
Router(config-ip) # ip address 20.0.0.1 255.0.0.0
Router(config-ip) # ip helper-address 10.0.0.2
      nol shut
      exit
```

OBSERVATION:

ping 10.0.0.4

packets: sent=4, received=4, lost=0

round trip time = 0.000 ms

avg. round trip time = 0.000 ms

highest round trip time at 0.000 ms

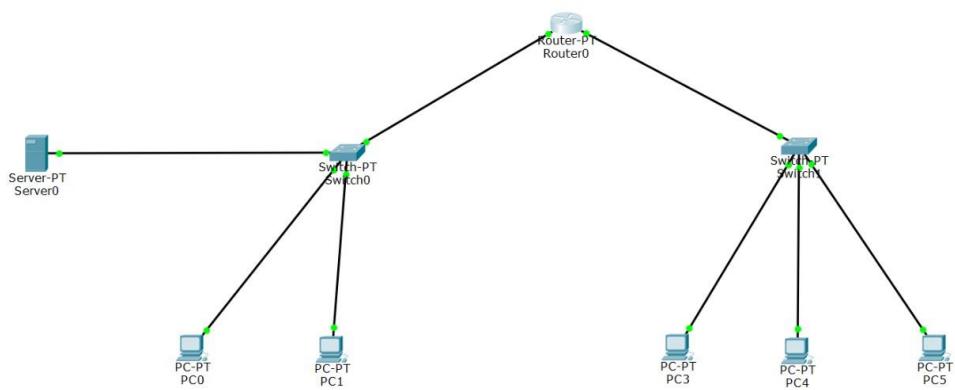
lowest round trip time at 0.000 ms

transmission times around 0.000 ms no errors no drops

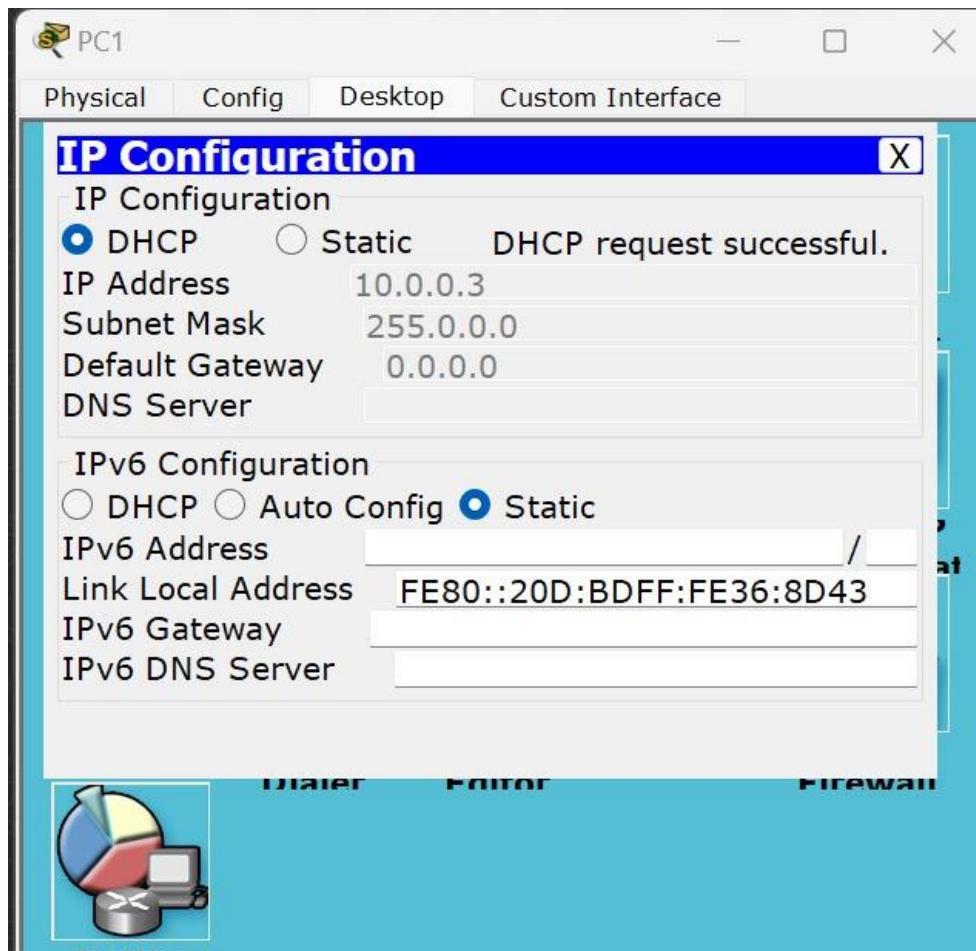
ip of 10.0.0.2

source

Screenshot of Topology:



Screenshot of Output:



Router0

Physical Config CLI

IOS Command Line Interface

```
Router(config)#int fa 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#no shutdown

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
do write memory
Building configuration...
[OK]
Router(config-if)#exit
Router(config)#int fa 01
^
% Invalid input detected at '^' marker.

Router(config)#int fa 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 20.0.0.1\
```

Copy Paste

Router0

Physical Config CLI

IOS Command Line Interface

```
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 20.0.0.1\
^
% Invalid input detected at '^' marker.

Router(config-if)#ip helper-address 20.0.0.1
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#configure terminal
Enter configuration commands, one per line. End with
CTRL/Z.
Router(config)#interface FastEthernet1/0
Router(config-if)#no shutdown

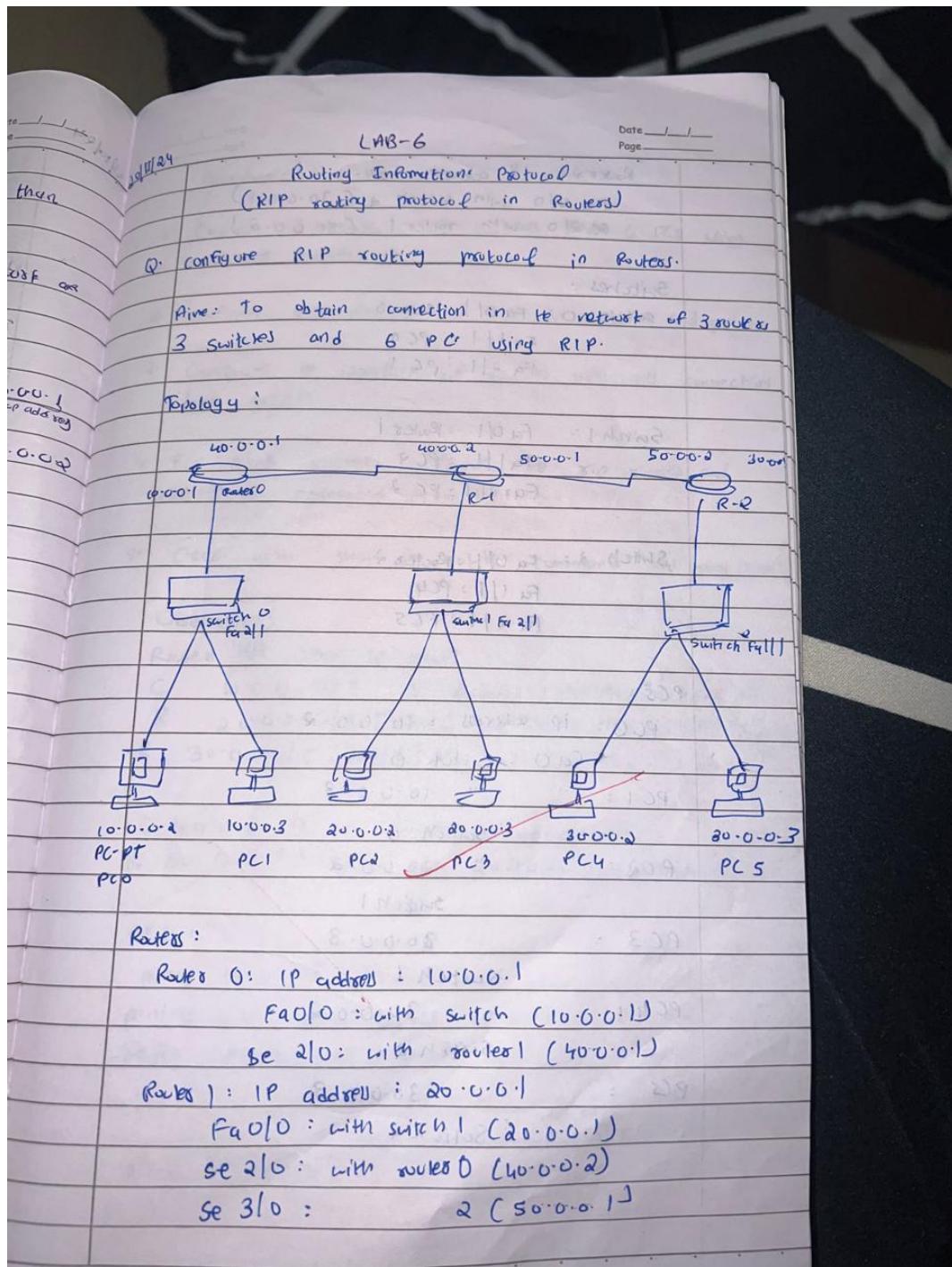
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state
to up
```

Copy Paste

Program 5:

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

Observation:



Router A: IP address : 30.0.0.1
 Fa 0/0 with switch a (30.0.0.1)
 serial with router 1 (50.0.0.2)

Switches :
 Switch 0: Fa 0/1 : Router 0
 Fa 1/1 : PC 0
 Fa 2/1 : PC 1

Switch 1: Fa 0/1 : Router 1
 Fa 1/1 : PC 2
 Fa 2/1 : PC 3

Switch 2: Fa 0/1 : Router 2
 Fa 1/1 : PC 4
 Fa 2/1 : PC 5

PCs :
 PC 0: IP address : 10.0.0.2
 Fa 0 : switch 0
 PC 1: " " 10.0.0.3
 : switch 0
 PC 2: " " 20.0.0.2
 : switch 1
 PC 3: " " 20.0.0.3
 : switch 1
 PC 4: " " 30.0.0.2
 : switch 2
 PC 5: " " 30.0.0.3
 : switch 2

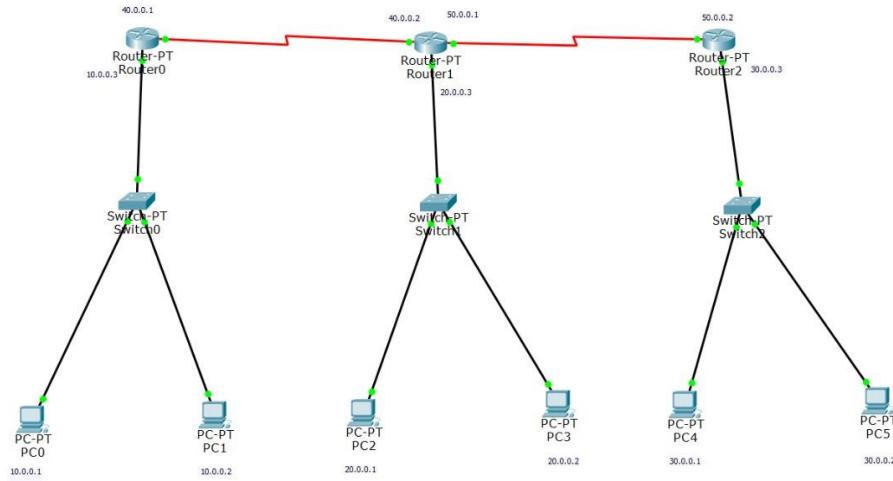
Procedure:
 1. Connect 3 routers, 3 switches and 6 PCs using appropriate connections.
 2. Configure end-devices: 6 PCs IP address and gateway.
 3. Configure the interface of routers with all connections turn green.
 4. For each router configure the static routes and add all connected networks.
 5. Check with show ip route command and ping command.

Observation:
 Router #> show ip route
 C 10.0.0.0/8 is directly connected FastEthernet 0/0.
 R 20.0.0.0/8 [12u1] via 40.0.0.2 00:00:26s serial 2/0
 R 30.0.0.0/8 [12u1] via 40.0.0.2, 00:00:26s serial 0/0
 C 40.0.0.0/8 is directly connected 2/0
 R 50.0.0.0/8 [12u1] via 40.0.0.2 00:00:26s serial 2/0

PC 0:
 ping 10.0.0.3
 pinging with 32 bytes of data.
 Reflected from 10.0.0.3 = bytes 3d or 7.7L:1/08

ping 20.0.0.2

Screenshot of Topology:



Screenshot of Output:

```
Router0#enable
Router0#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router0(config)#router rip
Router0(config-router)#network 10.0.0.0
Router0(config-router)#exit
Router0(config)#exit
Router0#
%SYS-5-CONFIG_I: Configured from console by console
```

```
Router1#enable
Router1#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#interface Serial3/0
Router1(config-if)#ip address 50.0.0.2 255.0.0.0
Router1(config-if)#exit
Router1(config)#exit
Router1#
%SYS-5-CONFIG_I: Configured from console by console
```

```
Router2#enable
Router2#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router2(config)#router rip
Router2(config-router)#network 50.0.0.0
Router2(config-router)#network 30.0.0.0
Router2(config-router)#exit
Router2(config)#exit
Router2#
%SYS-5-CONFIG_I: Configured from console by console
```

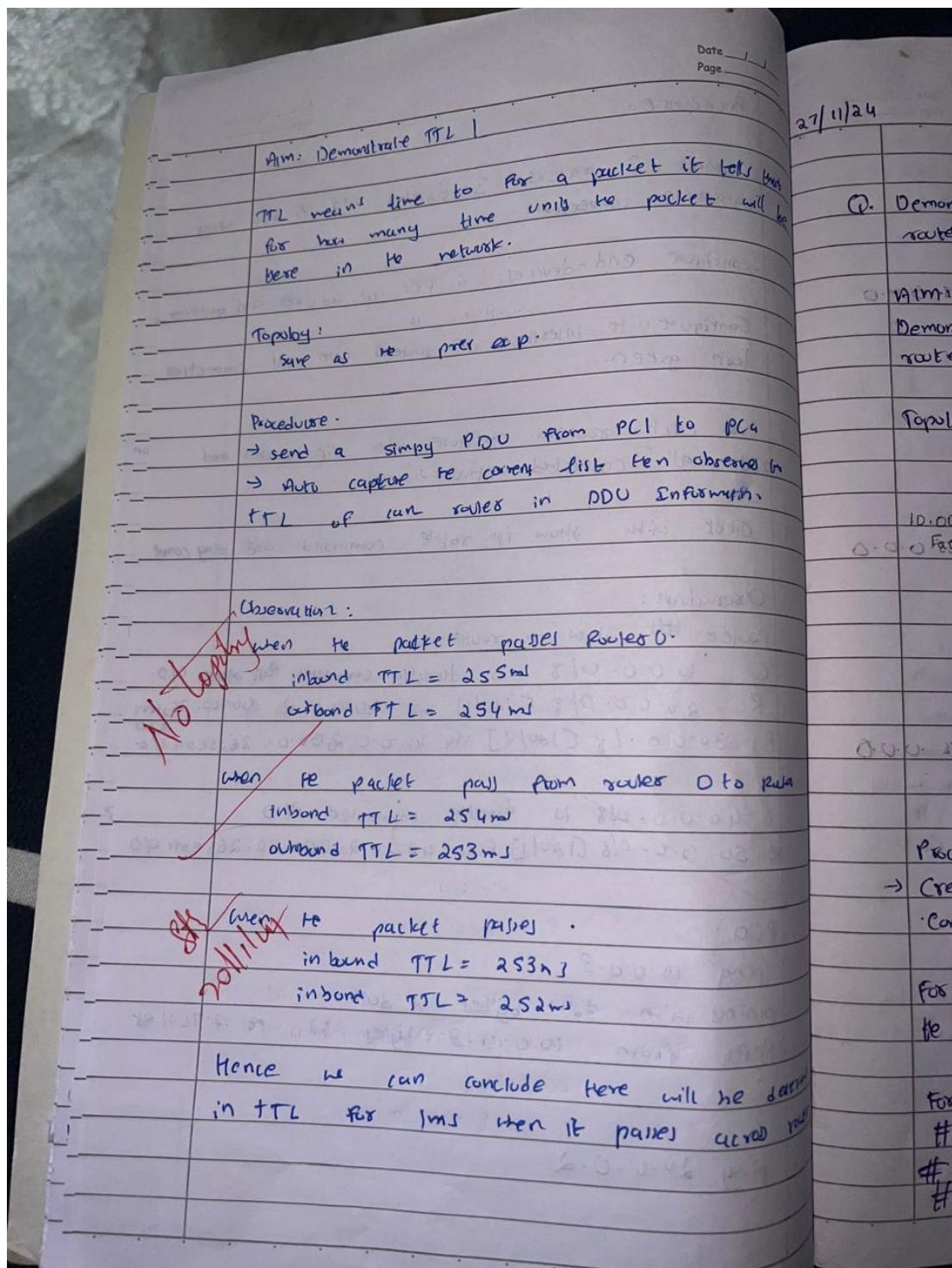
The screenshot shows a software application window titled "PC0" with tabs for "Physical", "Config", "Desktop", and "Custom Interface". A "Command Prompt" window is open, displaying the following output:

```
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25%  
loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 3ms, Maximum = 7ms, Average = 5ms  
  
PC>ping 20.0.0.1  
  
Pinging 20.0.0.1 with 32 bytes of data:  
  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=126  
Reply from 20.0.0.1: bytes=32 time=2ms TTL=126  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=126  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=126  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0%  
loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 1ms, Maximum = 2ms, Average = 1ms  
  
PC>
```

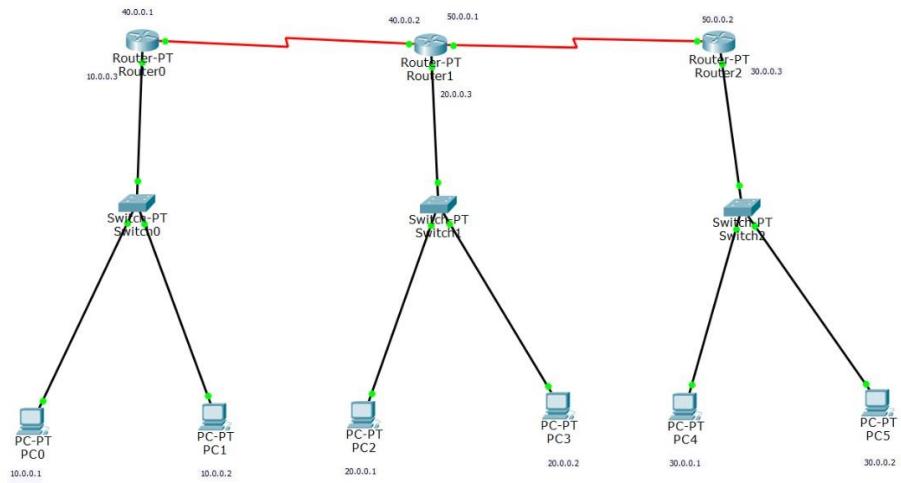
Program 6

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

Observation



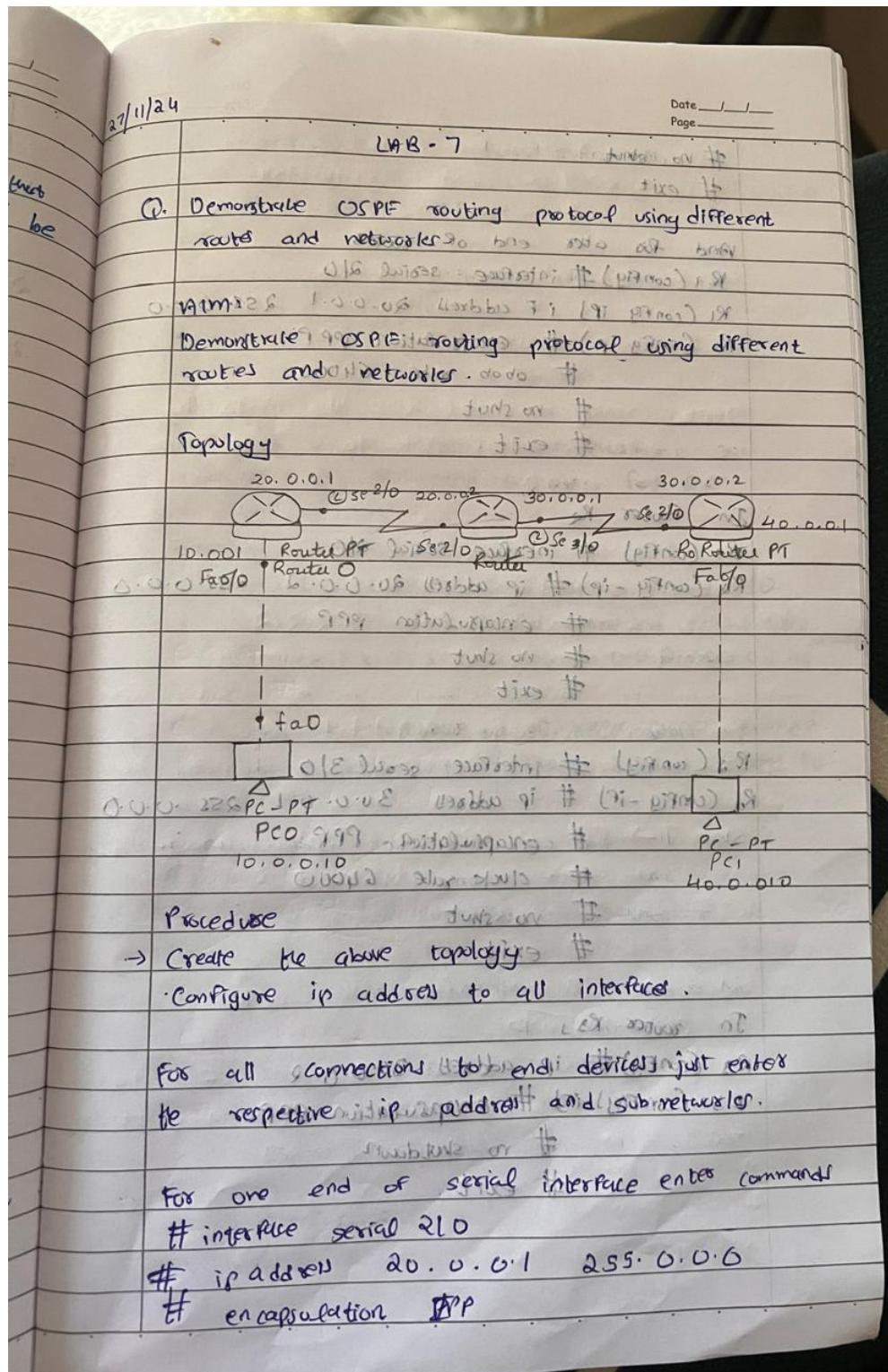
Screenshot of Topology:



Program 7

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

Observation:



- Procedure
- Create the above topology #
- Configure ip address to all interfaces.

For all connections between devices just enter the respective ip address and subnetmask.

For one end of serial interface enter commands
interface serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation PPP

no shutdown
 # exit
 And fix other end of routes!
 R1 (config) # interface serial 2/0
 R1 (config-if) # ip address 20.0.0.1 255.0.0.0
 R1 (config ip) # encapsulation PPP
 # clock rate 64000
 # no shutdown
 # exit.
 In Router R2
 R2 (config) # interface serial 2/0
 R2 (config-if) # ip address 20.0.0.2 255.0.0.0
 # encapsulation PPP
 # no shutdown
 # exit
 R2 (config) # interface serial 3/0
 R2 (config-if) # ip address 30.0.0.1 255.0.0.0
 # encapsulation PPP
 # clock rate 64000
 # no shutdown
 # exit
 In Router R3
 R3 (config) # ip address 130.0.0.2 128.0.0.0
 R3 (config-if) # encapsulation PPP
 # no shutdown
 # exit, to this route not
 0.0.0.255 130.0.0.6 metric 1

R3 (config) # interface fastethernet 0/0
 R3 (config-if) # ip address 140.0.0.1 255.0.0.0
 R3 (config-if) # no shutdown
 # exit
 Step 4 → Now check
 routing table of R3
 routers # show ip route.
 20.0.0.0 connected to R1 from R3
 Gateway of last resort is not set
 C 10.0.0.0/8 is directly connected, O/L
 C 20.0.0.0/8 is directly connected, serial 0/0
 C 1A 40.0.0.0/8 via 20.0.0.2 00:04:23

There must be 1 interface up or keep OSPF process
 UPS so is better to configure loopback address to
 routes.

R1 & no shutdown
 R1 (config-if) # interface loopback 0 #
 # ip address 112.16.1.252 255.0.0.0
 # no shutdown

R2 & no shutdown
 R2 (config-if) # interface loopback 0 #
 # ip address 112.16.253 255.0.0.0
 # no shutdown

R3 (config-if) # interface GigabitEthernet0/0
R3 (config-if) # ip address 192.168.1.253 255.255.0.0
R3 (config-if) # no shutdown

Step 5: Now check routing of R2.

Rs # show ip route to stdby unroute
.31008 qj one # 88108
a link blw R12

Step 6: Create a virtual link b/w R1 & R2 by the we create a virtual link to connect area 0 to area 0.

R1 The following is the general rule.

virtual link 12.2.2.208

area bānqāo vītēp 810-0-0-0-0-0 AC)

Ran a ~~new~~ virtual-link 10.1.1.1 to 10.0.0.1
of ~~Virtual~~ ~~Virtual~~. ~~Switches~~ at address 0.0.2 (90)

Step 7 \rightarrow R₂ and R₃ get updates about a
Now check routing table of R₃

R3# show ip routes [ctrl + c] 18

output \rightarrow I/O port number from 1 to 16

Ping ID: 0.103

printing 10. 0.6.10 with 32 by 19 of data

Reply from 10.6.0.10 by reg = 32 fine = 9 ms TTL

10-226 828 81-811 4438160 47

full on #

" " "

Screenshot of Topology:



Screenshot of Output:

```
PC0
Physical Config Desktop Custom Interface

Command Prompt
X
Ping 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms 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 = 7ms, 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=9ms 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
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 = 6ms, Maximum = 9ms, Average = 7ms

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
C 20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/32 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 40.0.0.0/8 [110/129] via 20.0.0.2, 00:18:19, Serial2/0
C 172.16.0.0/16 is directly connected, Loopback0

Router#

%SYS-6-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 40.0.0.8 [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)#

Router2

Physical Config CLI

IOS Command Line Interface

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

Gateway of last resort is not set

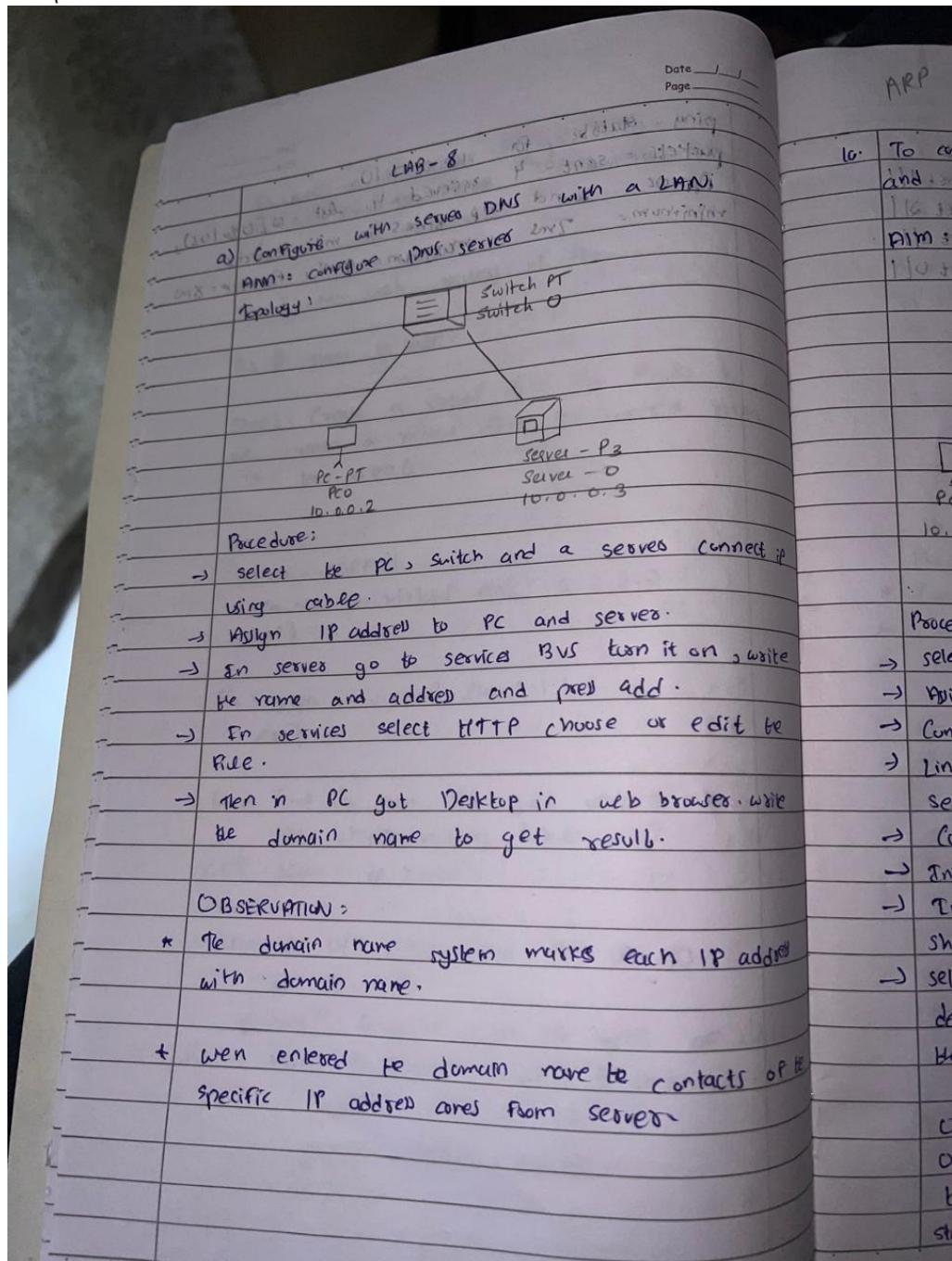
O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:00:11, Serial2/0
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:21:19, Serial2/0
  30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C     30.0.0.0/8 is directly connected, Serial2/0
C     30.0.0.1/32 is directly connected, Serial2/0
C     40.0.0.0/8 is directly 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
```

Copy Paste

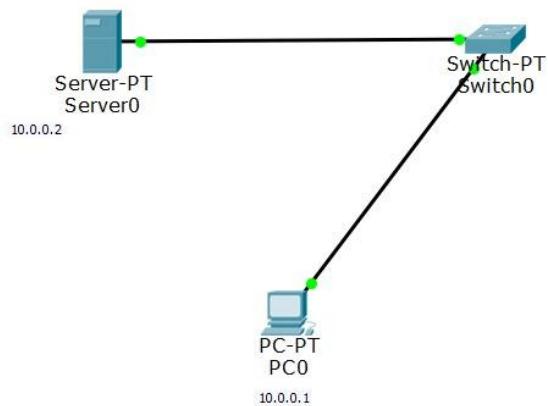
Program 8

Q8: Configure Web Server, DNS within a LAN

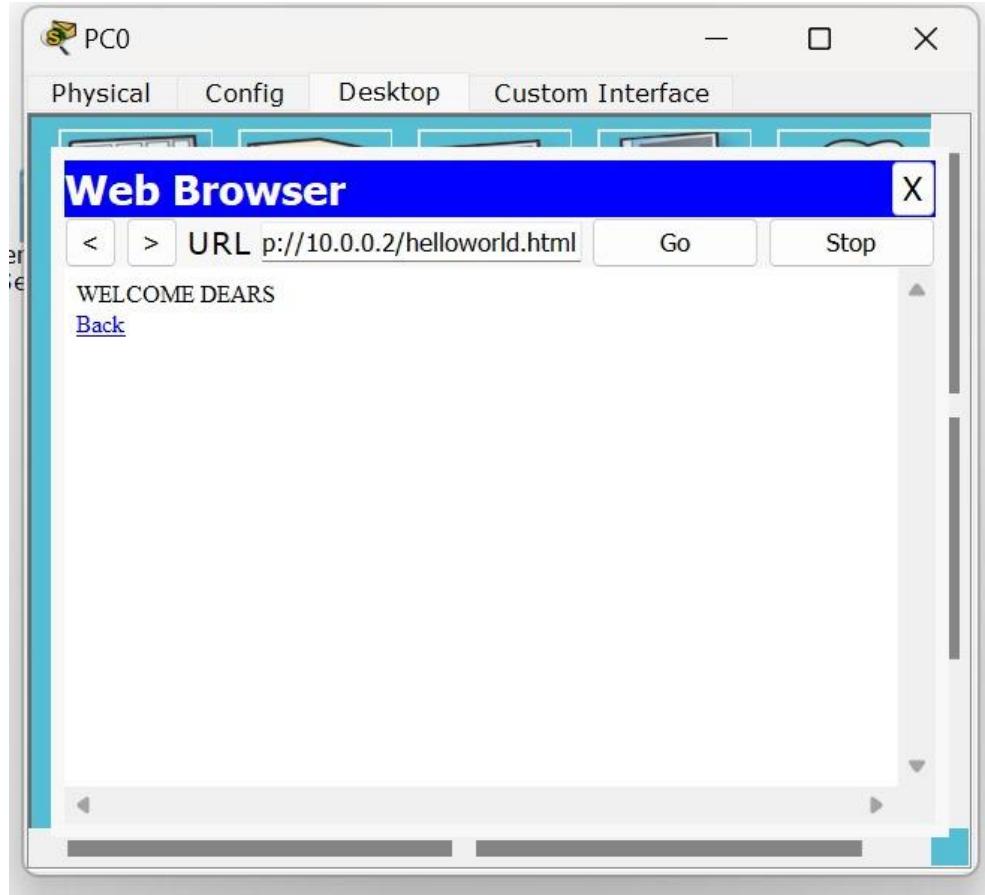
Observation:



Screenshot of Topology:



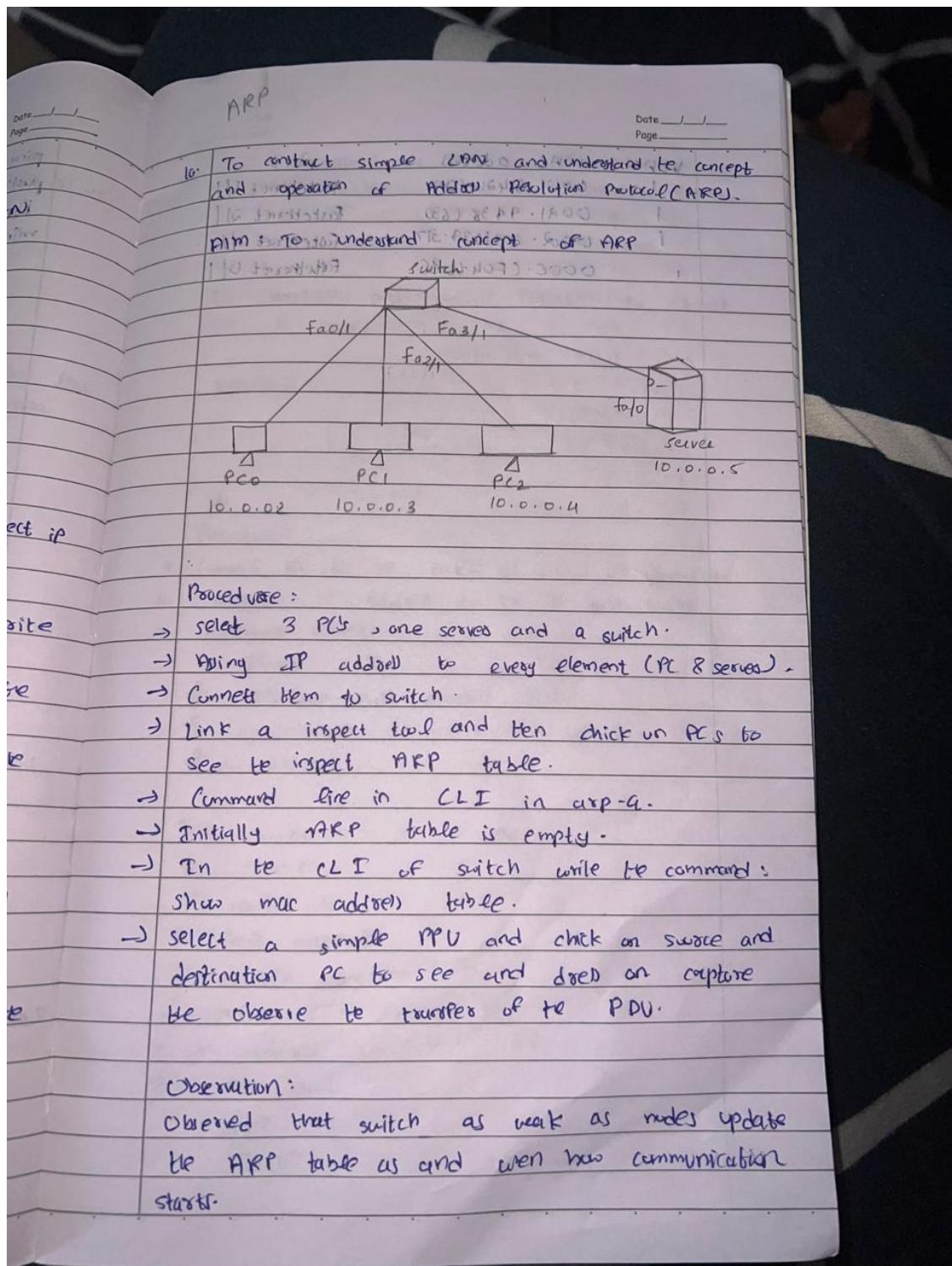
Screenshot of Output:



Program 9:

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

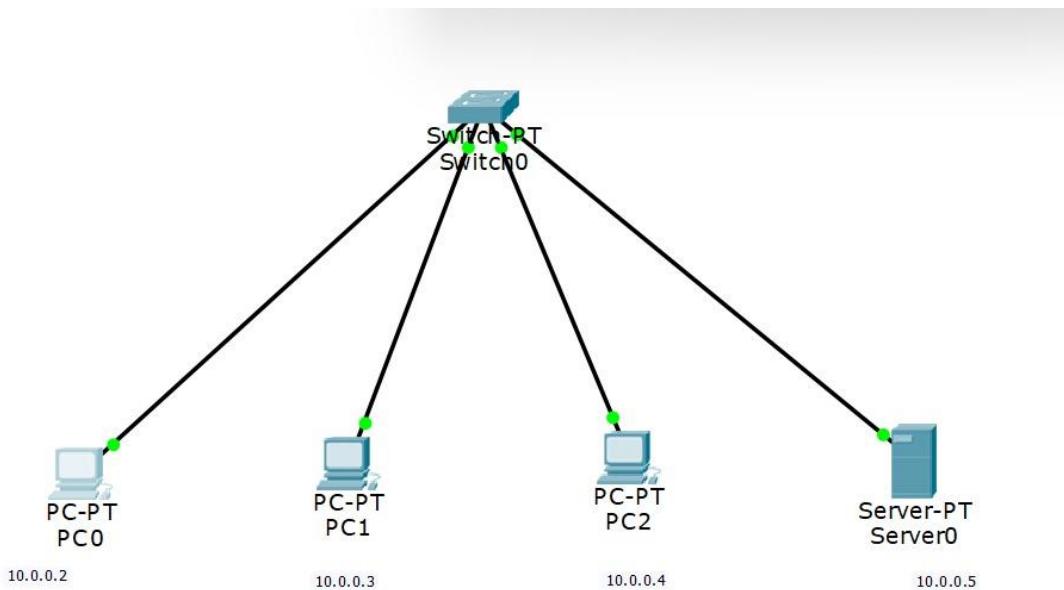
Observation:



Date _____
Page _____

VLAN	MAC address	Port
1	0001-61420-6004	Fastelinet 1/1
1	0021-9438-C630	Fastelinet 2/1
1	0002-441039-317	Fastelinet 3/1
1	000C-CF04-603C	Fastelinet 4/1

Screenshot of Topology:



Screenshot of Output:

The screenshot shows a network simulation interface with the following components:

- Command Prompt Window:** Displays the following output:

```
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.5:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>arp-a
Invalid Command.

PC>arp -a
  Internet Address      Physical Address      Type
  10.0.0.3                0030.f295.b57a    dynamic
  10.0.0.5                0030.a393.5134    dynamic
```
- ARP Table for PC0:** Shows the ARP table for host **PC0** with entries for **PC1** and **Server0**.

IP Address	Hardware Address	Interface
10.0.0.3	0030.F295.B57A	FastEthernet0
10.0.0.5	0030.A393.5134	FastEthernet0
- ARP Table for PC2:** Shows the ARP table for host **PC2** which is currently empty.

Switch0

Physical Config CLI

IOS Command Line Interface

```
Switch>show mac address table
      ^
% Invalid input detected at '^' marker.

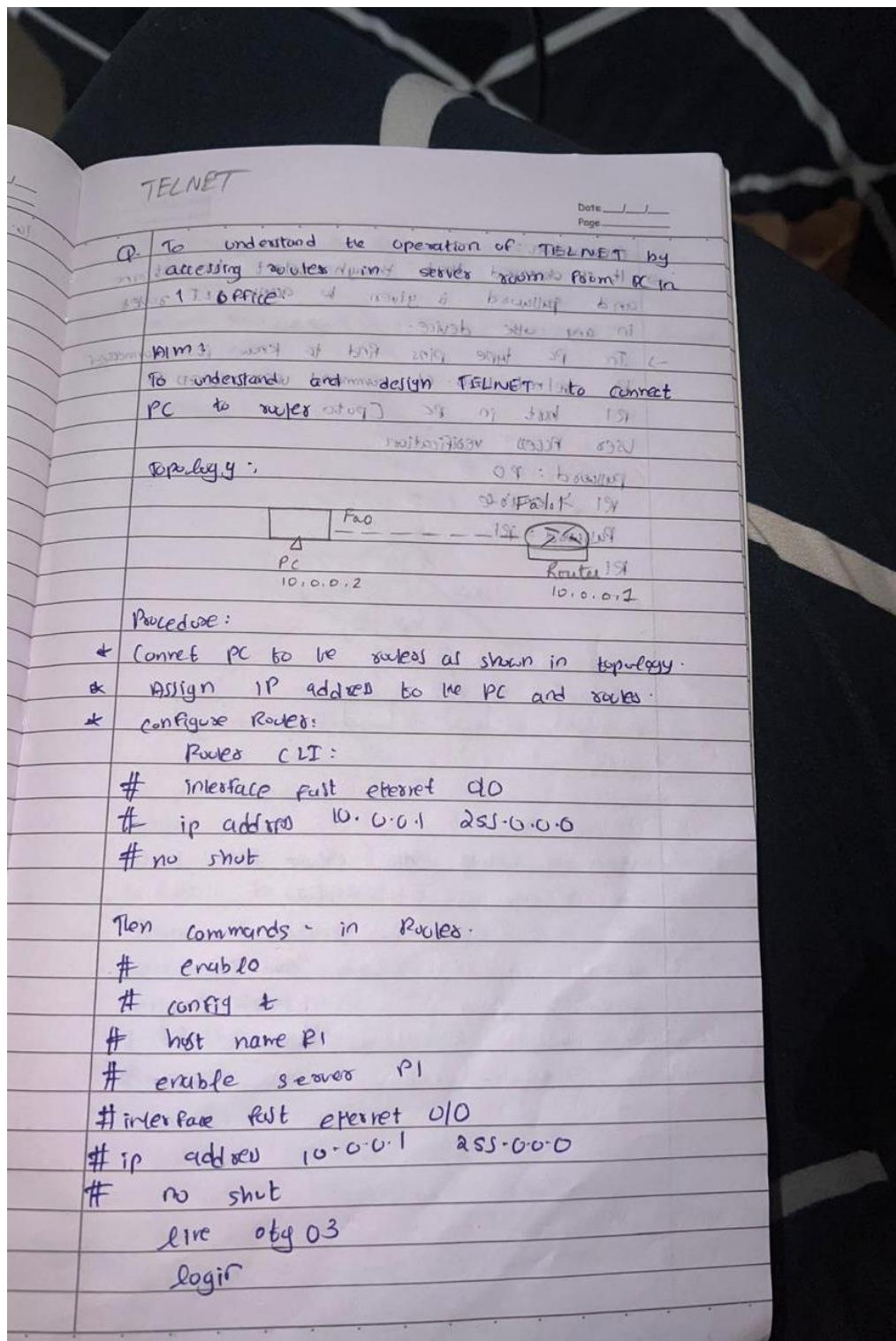
Switch>show mac address
      Mac Address Table
-----
Vlan      Mac Address          Type        Ports
----      -----
1         0005.5eb7.8cb1      DYNAMIC    Fa0/1
1         0005.5ebb.5671      DYNAMIC    Fa2/1
1         0030.a393.5134      DYNAMIC    Fa3/1
1         0030.f295.b57a      DYNAMIC    Fa1/1
Switch>arp-a
Translating "arp-a"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find
```

Copy Paste

Program 10

Q10: To understand the operation of TELNET

Observation:



Observation: It is observed that through telnet port the host name and password is given to access CLI router in any other device.

In PC type pins find to know its connected to R1 telnet to R1. Observe command user access to R1 host in PC (putty session at 39). Use access verification
password: PO
R1 > enable
Password: R1

R1

Configuration:

Router#

Router#

Router#

Router#

0.0.0.255 1.0.0.0 255.255.255.0

0.0.0.255 1.0.0.0 255.255.255.0

0.0.0.255 1.0.0.0 255.255.255.0

0.0.0.255 1.0.0.0 255.255.255.0

0.0.0.255 1.0.0.0 255.255.255.0

0.0.0.255 1.0.0.0 255.255.255.0

Screenshot of Topology:



Screenshot of Output:

```
Router#enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#enable secret chris
Router(config)#line vty 0 5
Router(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
% Login disabled on line 136, until 'password' is set
% Login disabled on line 137, until 'password' is set
Router(config-line)#password 123
Router(config-line)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

The screenshot shows a window titled "Command Prompt" from the "Packet Tracer PC Command Line 1.0". The window has tabs at the top: "Physical", "Config", "Desktop", and "Custom Interface". The "Physical" tab is selected. The main area displays the following command-line session:

```
Packet Tracer PC Command Line 1.0
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

User Access Verification

Password:
Router>enable
Password:
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

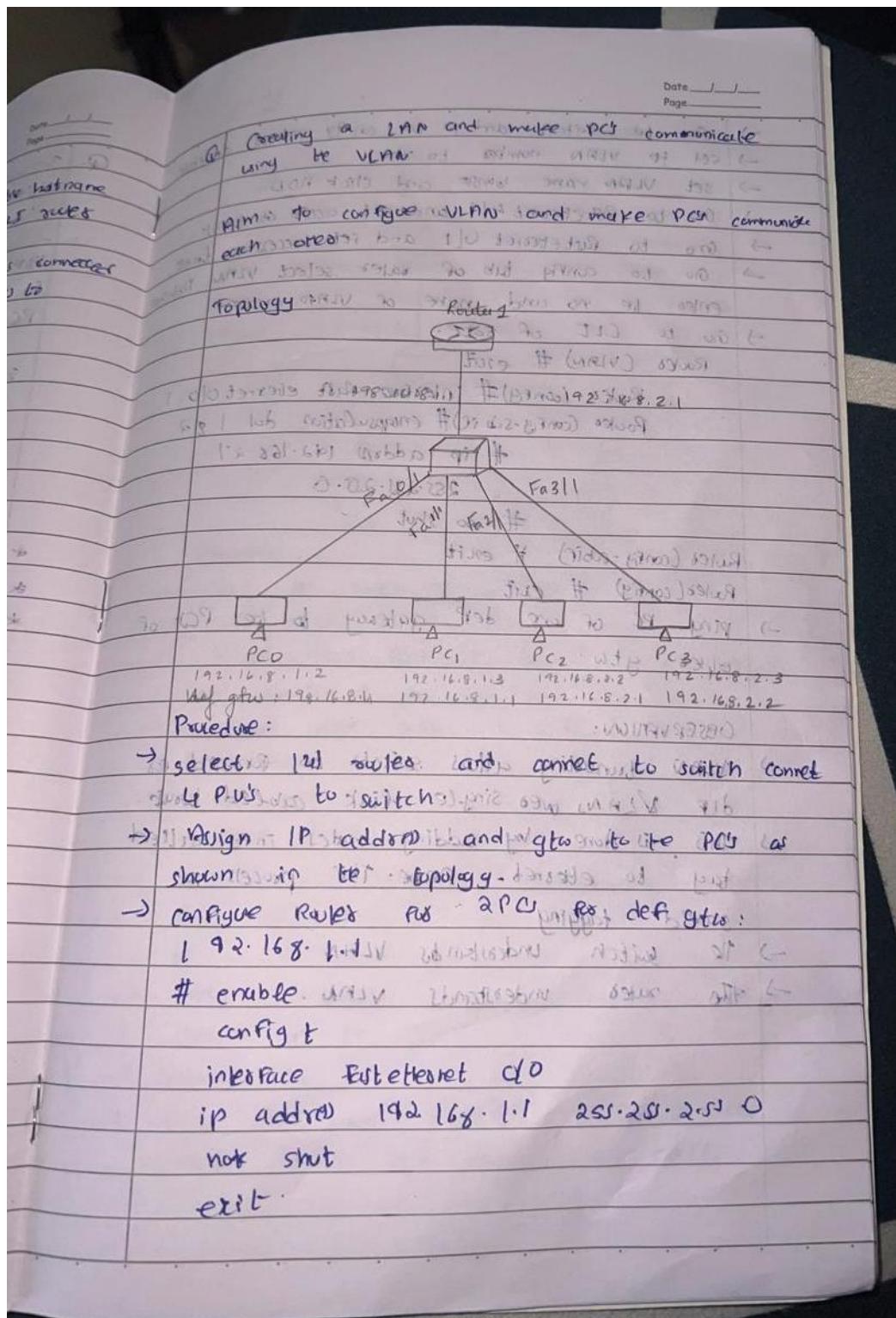
Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
Router#
```

Program 11

Q11: To construct VLAN and the PCs communicate along a VLAN

Observation:



Date _____
 Page _____

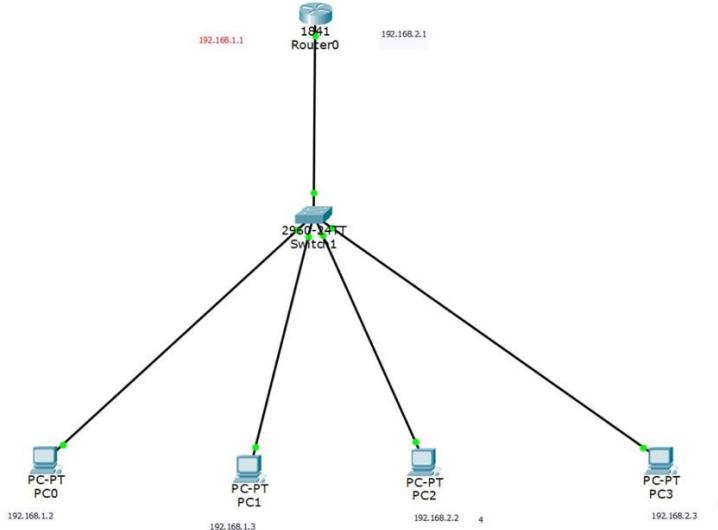
→ Go into switch and in IOS config (select) database
 → set the VLAN number to 200V 94 200
 → set VLAN name buse and click ADD -
 → Go to fast ethernet 0/1 and set access as intronuke.
 → Go to fast ethernet 0/1 and set access as trunk
 → Go to config tab of router select VLAN Database
 enter the no and name of VLAN precedent.
 → Go to CLI of route.
 Router (VLAN) # exit
 Router (config)# interface fast ethernet 0/0.1
 Router (config-subif)# encapsulation dot1q 2
 # ip address 192.168.2.1
 255.255.255.0
 # no shut
 Router (config-subif) # exit
 Router (config) # exit
 → ping PC of one def gateway to be PC of
 other gtw.

OBSERVATION:

- VLAN segmentation allows switch to forward traffic between VLANs over single link of cabled link.
- This is done by adding header in payload called tag to ethernet payload. This process is called saving tagging.
- The switch understands VLAN 802.1Q
- The router understands VLAN 802.1Q

0.0.0.128.1.1.255.255.255.0
 192.168.2.1.1.255.255.255.0
 Juniper

Screenshot of Topology:



Screenshot of Output:

The screenshot shows the Router0 CLI interface. The title bar indicates the device is Router0. The menu bar includes Physical, Config, and CLI tabs, with Config selected. The main window displays the IOS Command Line Interface. The configuration command entered is:

```
state to up
```

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed state to up

```
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#

```

At the bottom of the window are Copy and Paste buttons.

PC0

Physical Config Desktop Custom Interface

Command Prompt X

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

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

PC0

Physical Config Desktop Custom Interface

Command Prompt X

```
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=0ms TTL=128

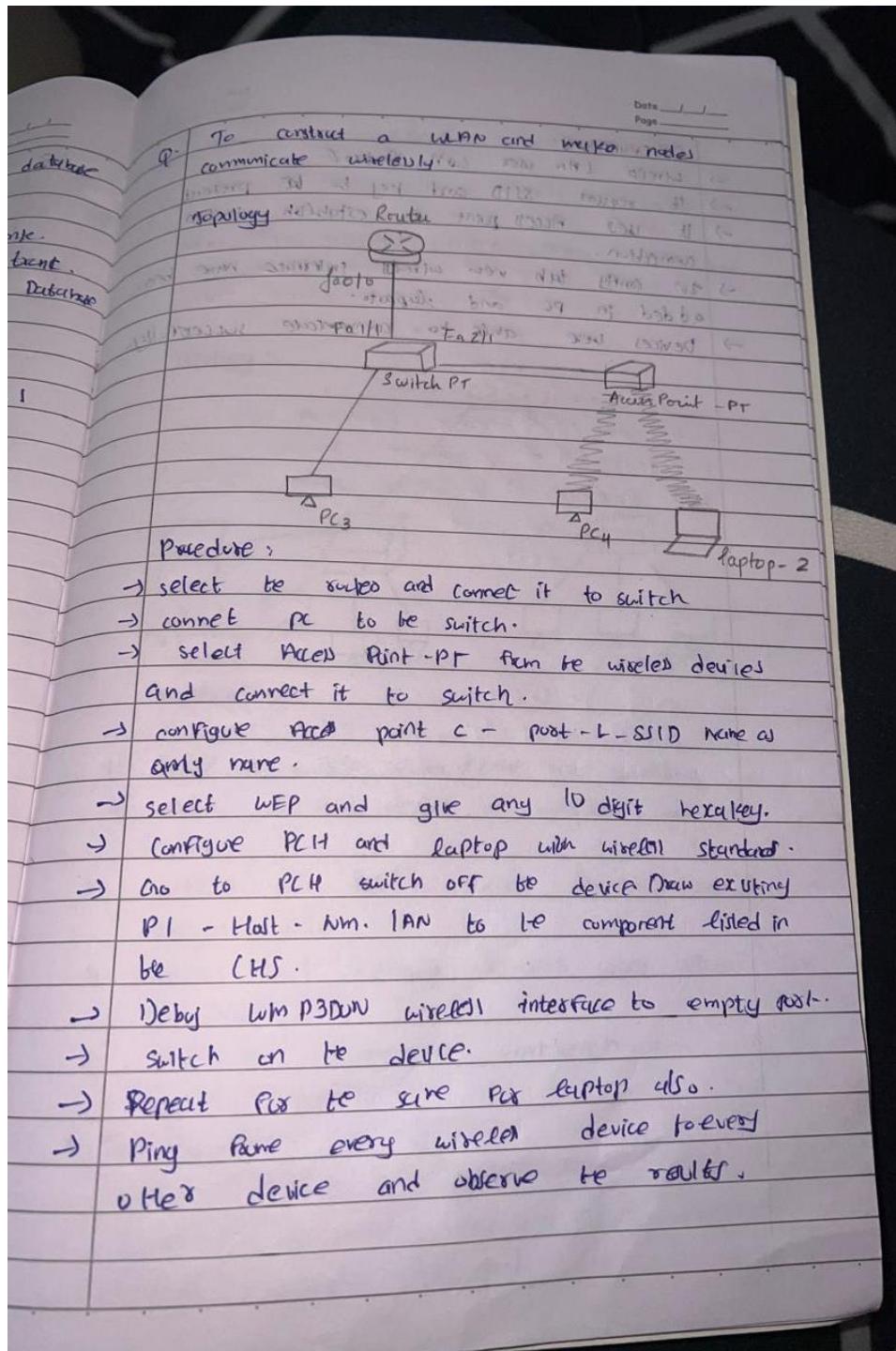
Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

Program 12

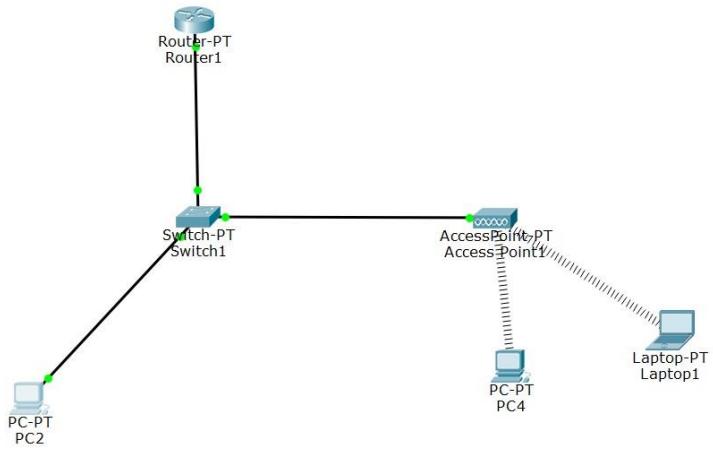
Q12: Construct a WLAN and make the nodes communicate wirelessly.

Observation:



- Date _____
Page _____
- OBSERVATION:** This experiment is about wireless LAN.
- wireless LAN uses WPA2 Protocol (minimum)
 - It requires SSID and key to be present.
 - It uses Access point to establish wireless connection.
 - In config tab new wireless interface have been added in PC and laptop.
 - Devices were able to communicate successfully.
- QUESTION:**
- Nature of the channel used by IEEE 802.11
- Nature of the power level used by IEEE 802.11
- Nature of the transmission time
- What is IEEE 802.11a - b - g - n → 5 things happen simultaneously
- What is IEEE 802.11n
- What is IEEE 802.11ac
- What is IEEE 802.11ax
- What is IEEE 802.11ad
- What is IEEE 802.11ah
- What is IEEE 802.11af
- What is IEEE 802.11ai
- What is IEEE 802.11aj
- What is IEEE 802.11ak
- What is IEEE 802.11al
- What is IEEE 802.11am
- What is IEEE 802.11an
- What is IEEE 802.11ar
- What is IEEE 802.11as
- What is IEEE 802.11av
- What is IEEE 802.11aw
- What is IEEE 802.11ay
- What is IEEE 802.11az

Screenshot of Topology:



Screenshot of Output:

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of a ping command between two hosts. The host names are visible in the window title bar: "PC2" and "PC3".

```
Request timed out.

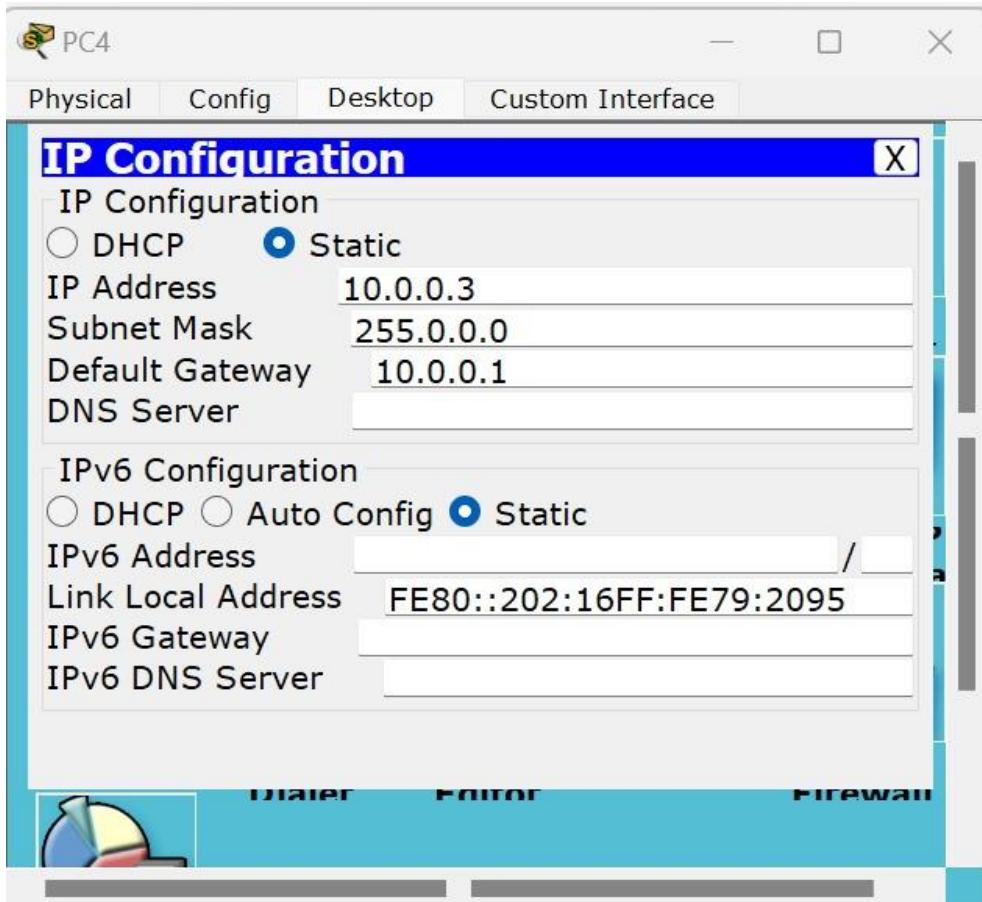
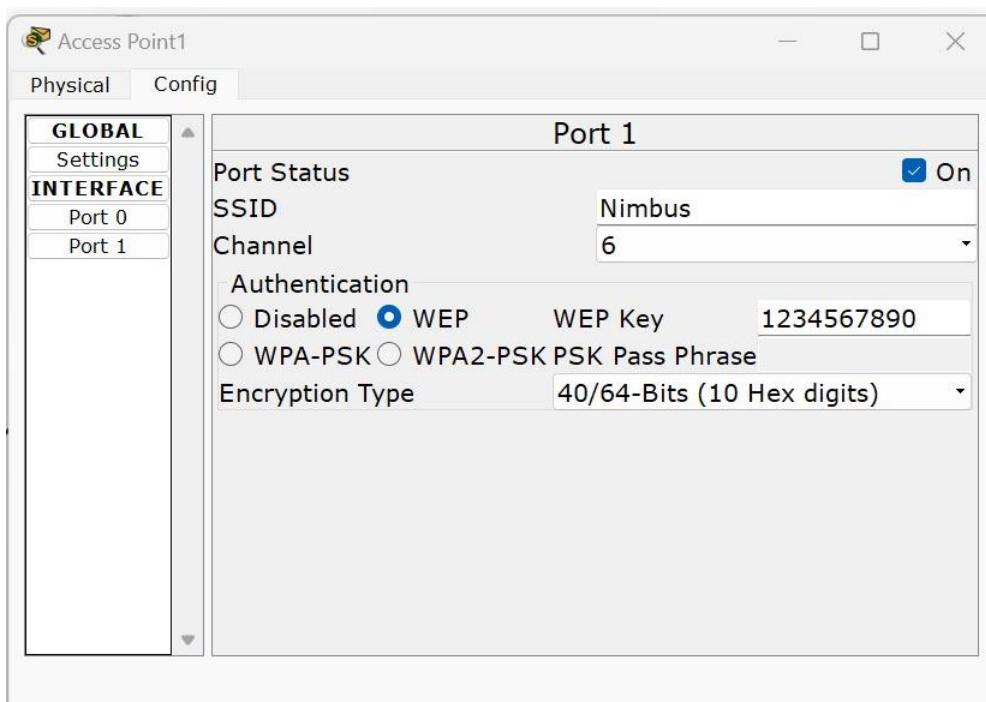
Ping statistics for 10.0.0.3:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=24ms TTL=128
Reply from 10.0.0.3: bytes=32 time=10ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=10ms 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 = 9ms, Maximum = 24ms, Average = 13ms

PC>
```



PC4

Physical Config Desktop Custom Interface

MODULES

WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
T-HOST-NM-1CF
T-HOST-NM-1CG
T-HOST-NM-1FF
T-HOST-NM-1FG
PT-HOST-NM-1W
T-HOST-NM-1W-
T-HOST-NM-3G/4
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
USB-HARD-DRI

Physical Device View

Zoom In Original Size Zoom Out



The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module

Customize Icon in Physical View 

Customize Icon in Logical View 



CYCLE-2

Program 13

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

Code and Output:

The image shows handwritten code on a lined notebook page. The title "CYCLE 2" is at the top left, and the date "11/12/25" is written vertically on the left margin. The code is organized into several functions:

- zxr(a,b):** A function that takes two binary strings, *a* and *b*, and returns a string where each character is '0' if *a[i]* equals *b[i]*, and '1' otherwise.
- mod2div(dividend, divisor):** A function that performs division of *dividend* by *divisor* using binary division. It uses a while loop to repeatedly subtract *divisor* from *dividend* until *dividend* is less than *divisor*. It handles both '0' and '1' cases for the quotient bit.
- encode(data, key):** A function that takes data and a key, appends a parity bit (0), and then performs mod2 div on the appended data with the key to get the remainder.

The code is written in a clear, cursive style, with some annotations and corrections visible.

Date 1/1
Page 1/1

```

codeword = data + remainder
printf("Remainder : ", remainder)
print("Encoded Data[ Data+ remainder], codeword")
return codeword.

def decode_data(encoded_data, key):
    remainder = mod2div(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 = "1010100100100"
key = "1101"
encoded_data = encode(data, key)
decoded_data = decode_data(encoded_data, key)

U/P :
Remainder = 1
encoded-data (data+ remainder)
= 101010010010011

Remainder after decoding = 000
No errors detected in received data.

```

Program 14

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

Code and Output:

14. Write a program for congestion control using leaky bucket algo.

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int no-of-queries = 4, storage = 0, output-pkt-size = 1;
    int input-pkt-size = 4, bucket-size = 10, size-left;
    for (int i = 0; i < no-of-queries; i++) {
        size-left = bucket-size - storage;
        if (input-pkt-size >= size-left) {
            storage += input-pkt-size;
        } else {
            cout << "Packet loss = 1\n";
            cout << "Buffer size = " << output-pkt-size << endl;
            storage -= output-pkt-size;
        }
    }
    return 0;
}
```

C/P:

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

Program 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.

Code and Output:

Q15. 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.

Client (TCP):

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name: ")
clientSocket.send(sentence.encode())
fileContent = clientSocket.recv(1024).decode()
print("From server")
print(fileContent)
clientSocket.close()
```

Servers (TCP):

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while True:
    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("Sent contents of " + sentence)
    file.close()
    connectionSocket.close()
```

UIP:

The server is ready to receive
sent contents of sentence TCP.py
The server is ready to receive
Enter file name: sentence TCP.py
Reply from server:
File contents of sentence TCP.py

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.

Code and Output:

The image shows handwritten notes for a UDP-based client-server program. On the left page, the client code (Client UDP.py) is written in Python. It imports socket, specifies a server address and port, creates a socket, sends a file name to the server, receives a reply, decodes it, and prints the content. On the right page, the server code (Server UDP.py) is shown. It binds to a port, receives messages from clients, decodes them, reads the file content, and sends it back to the client. The execution output is also handwritten, showing the interaction between the client and server.

Client (UDP.py):

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name")
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("Reply from server")
print(fileContent.decode("utf-8"))
clientSocket.close()
```

Server (UDP.py):

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = bind((serverName, serverPort))
print("The server is ready to receive")
while True:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    content = file.read(400)
    file.close()
    serverSocket.sendto(content.encode("utf-8"), clientAddress)
```

Output:

```
print ("sent contents of's ent=1")
print (sentence)
file.close()
UP:
The server is ready to receive
sent contents of's ent=1
The server is ready to receive:
Enter file name: server UDP.py
Reply from server:
```

Program 17

Q17: Tool Exploration-Wireshark

Code and Output:

11/125
17. Tool Exploration - Wireshark

It is a powerful and 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.

Key features:

1. Packet Capture: Captures live network traffic from various interfaces.
2. Protocol Analysis: Supports 100's of protocols.
3. Filtering: Offers powerful filters to isolate specific packets or traffic types.
4. Visualization: Displays packet details with hierarchical layers.

Use case of wireshark

1. Network trouble shooting
2. Security Analysis
3. Protocol Study

Common:

- * 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 address.
- * udp: Show only UDP traffic.