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

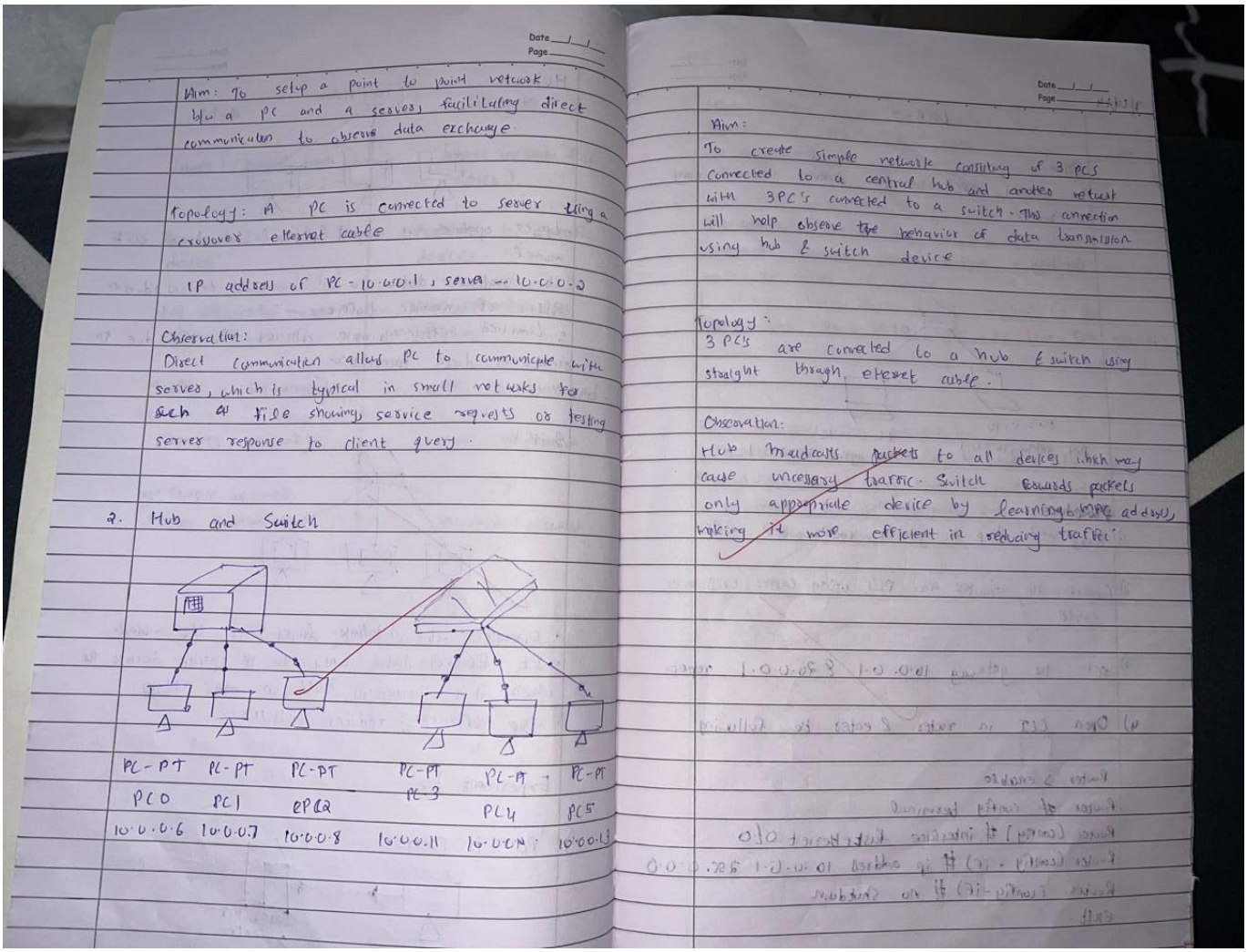
Cycle-II

Github Link:
https://github.com/chrisdt777/CN-LAB

Program 1

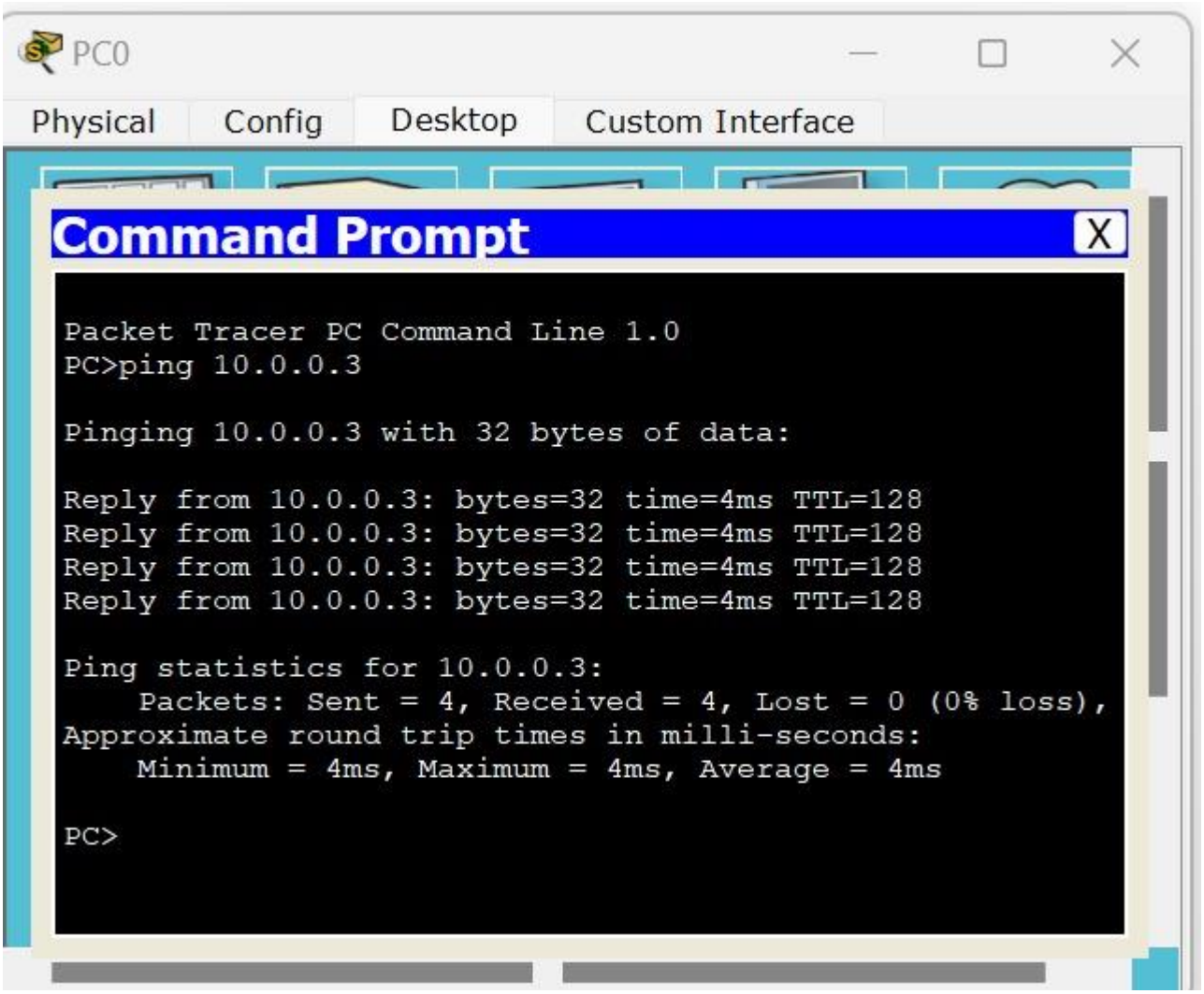
Q1: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Observation:



Screenshot of Topology:

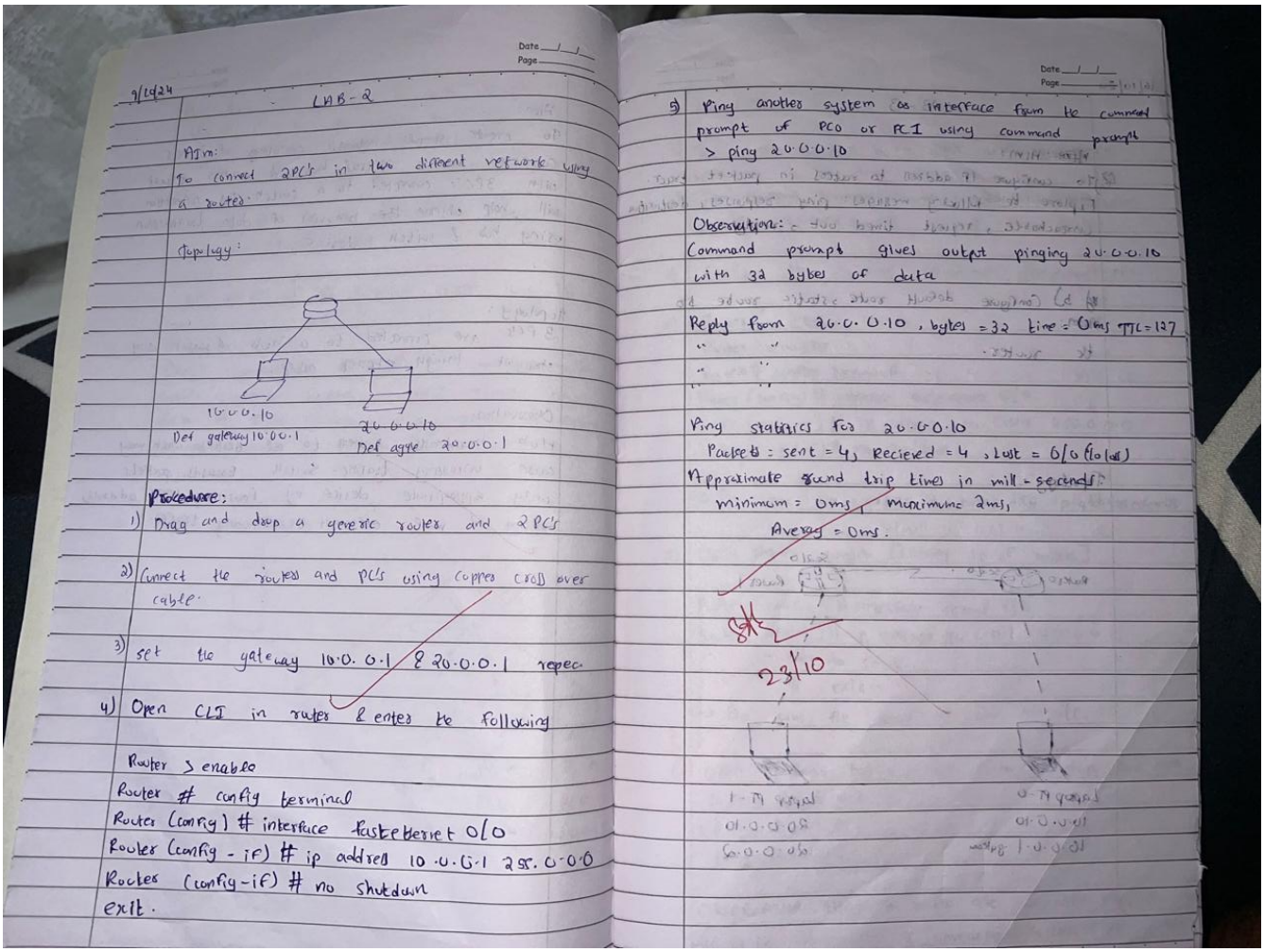
Screenshot of Output:



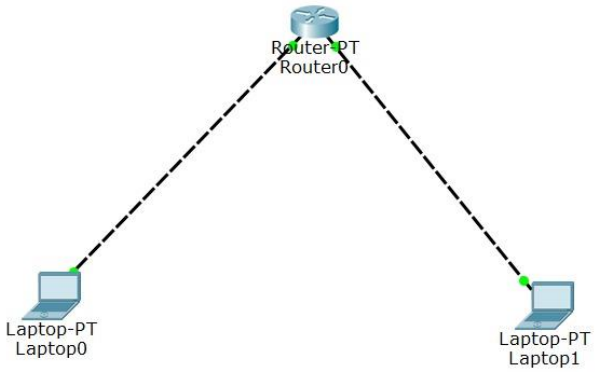
Program 2

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

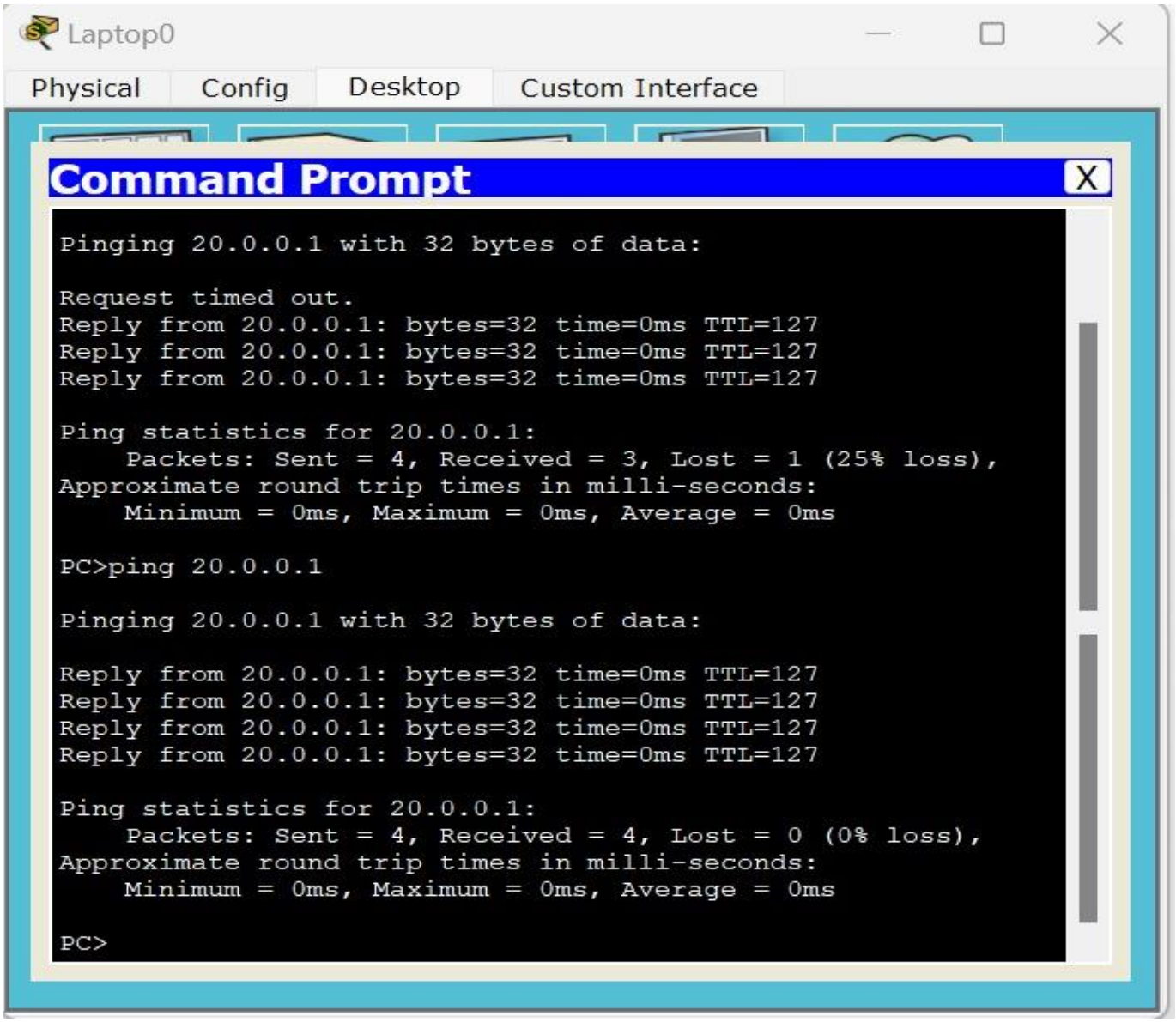
Observation:



Screenshot of Topology:

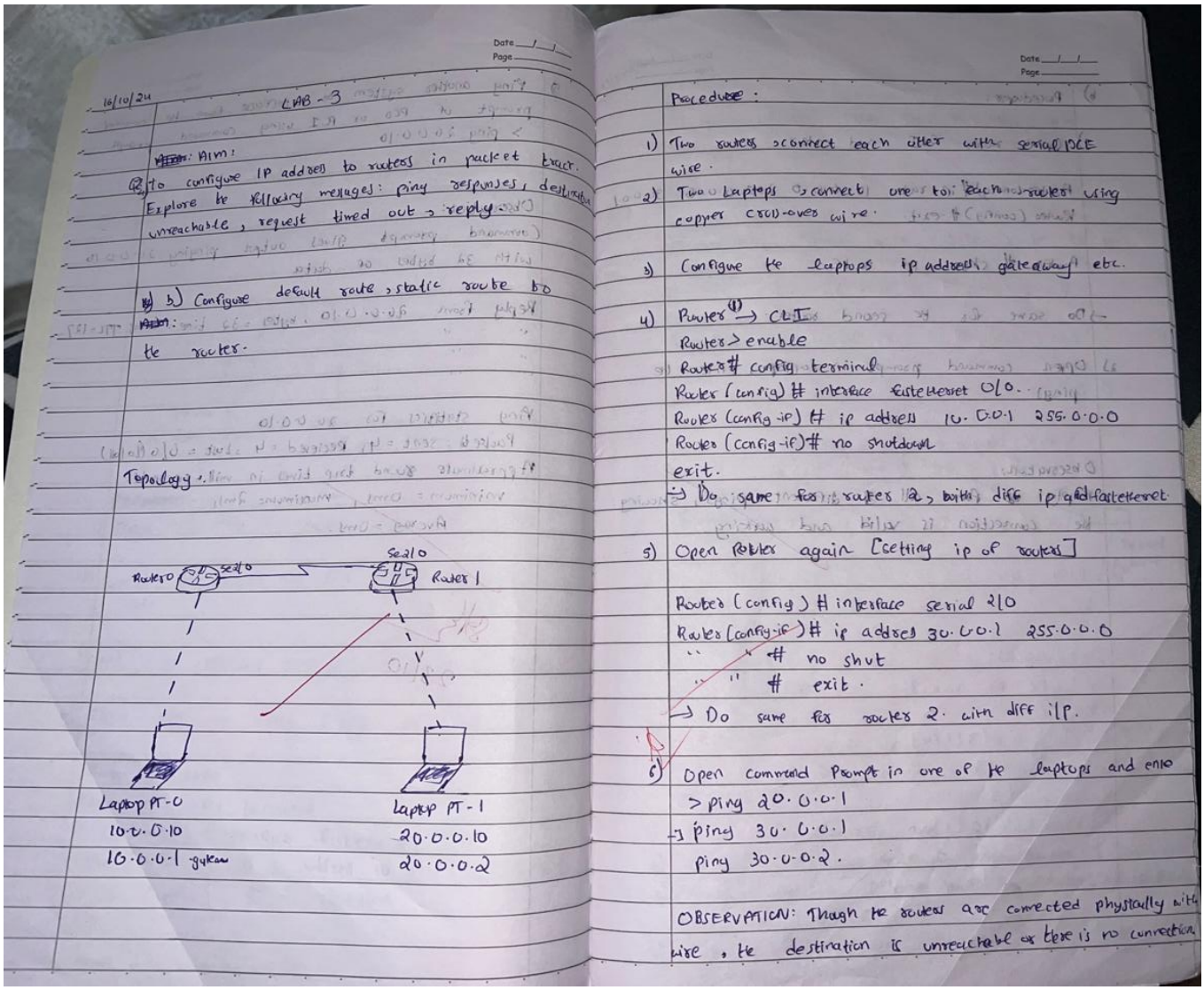


Screenshot of Output:

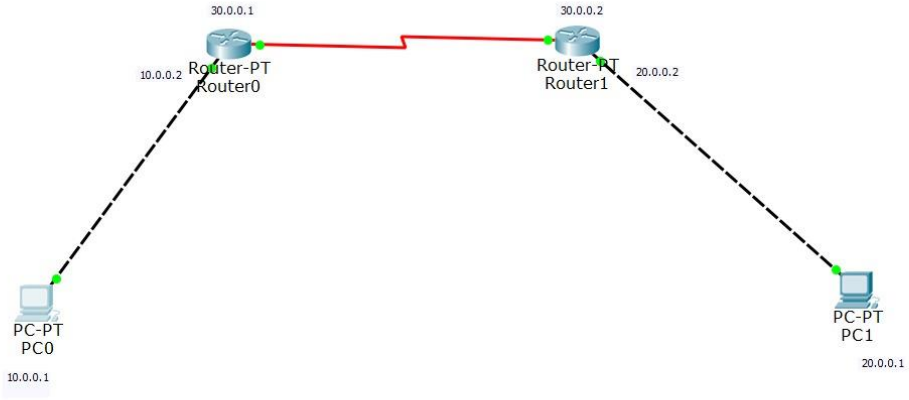


Q2 (b): To connect two PC's on two networks and two routers

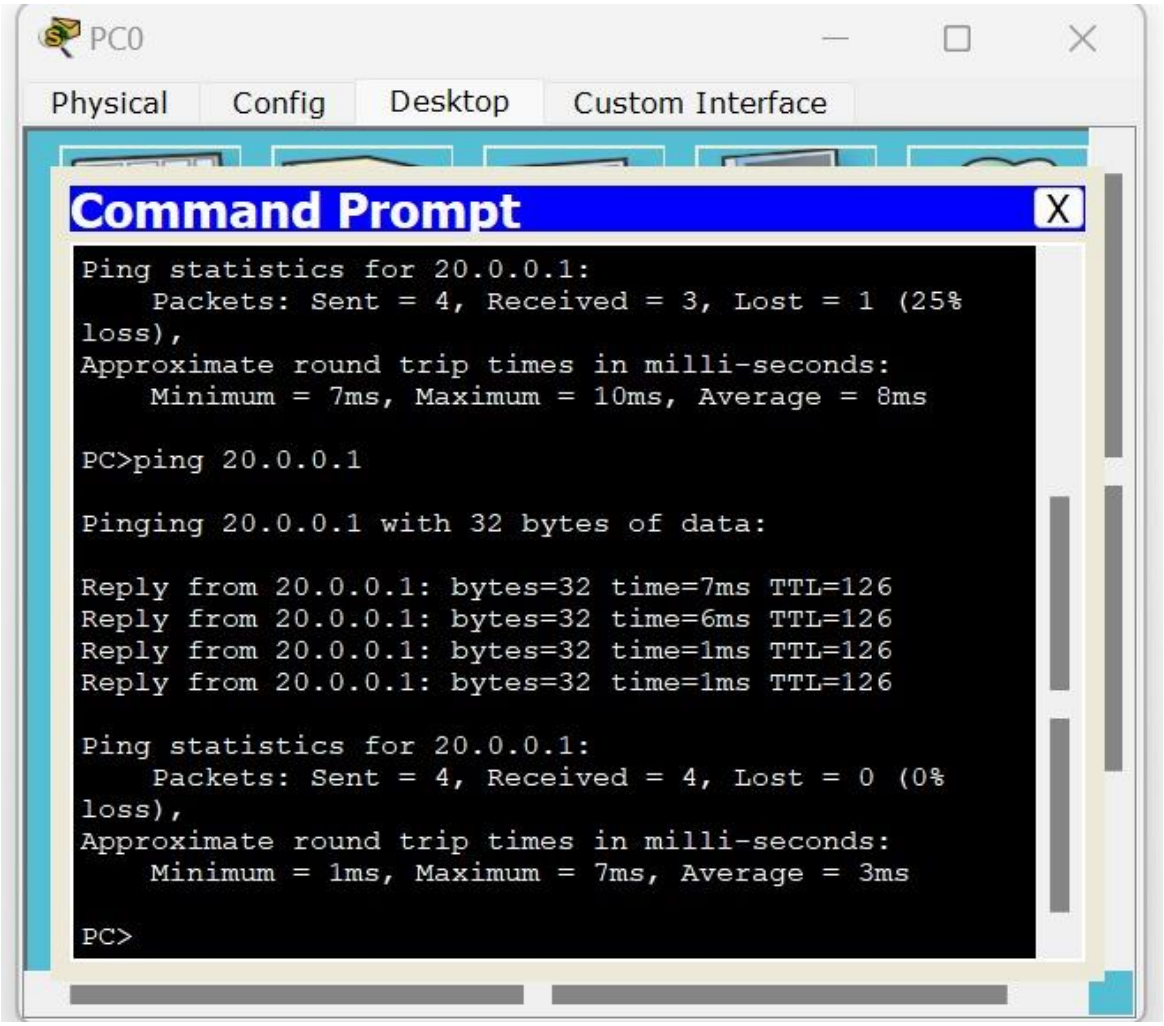
Observation:

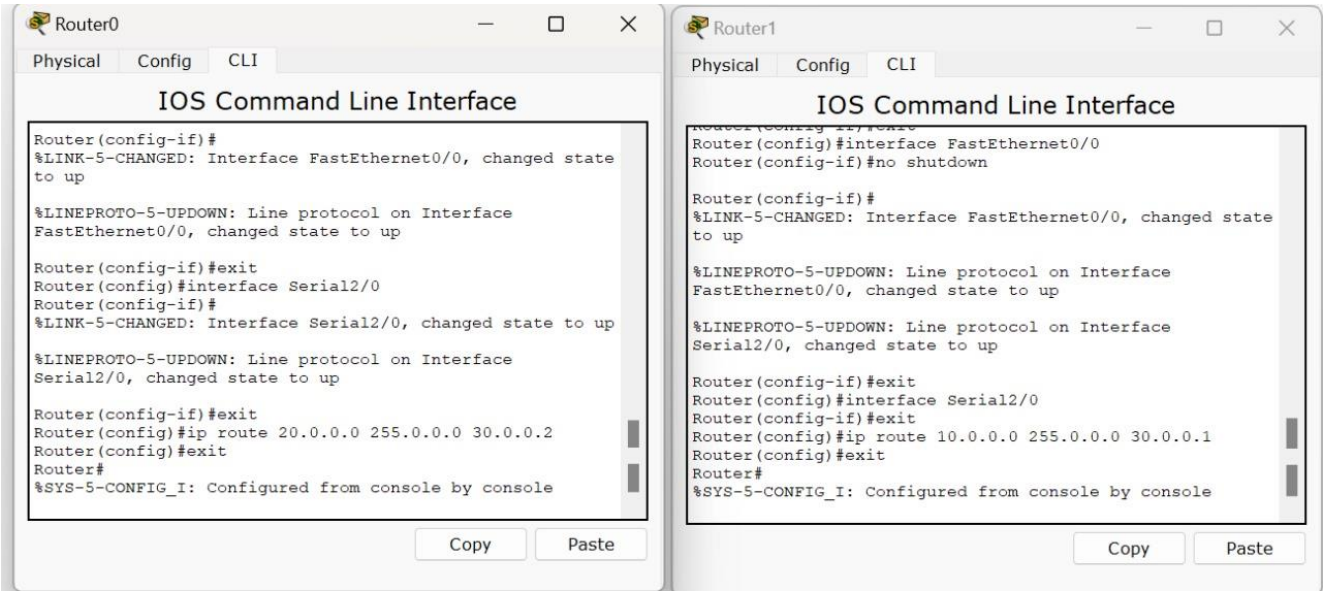


Screenshot of Topology:



Screenshot of Output:

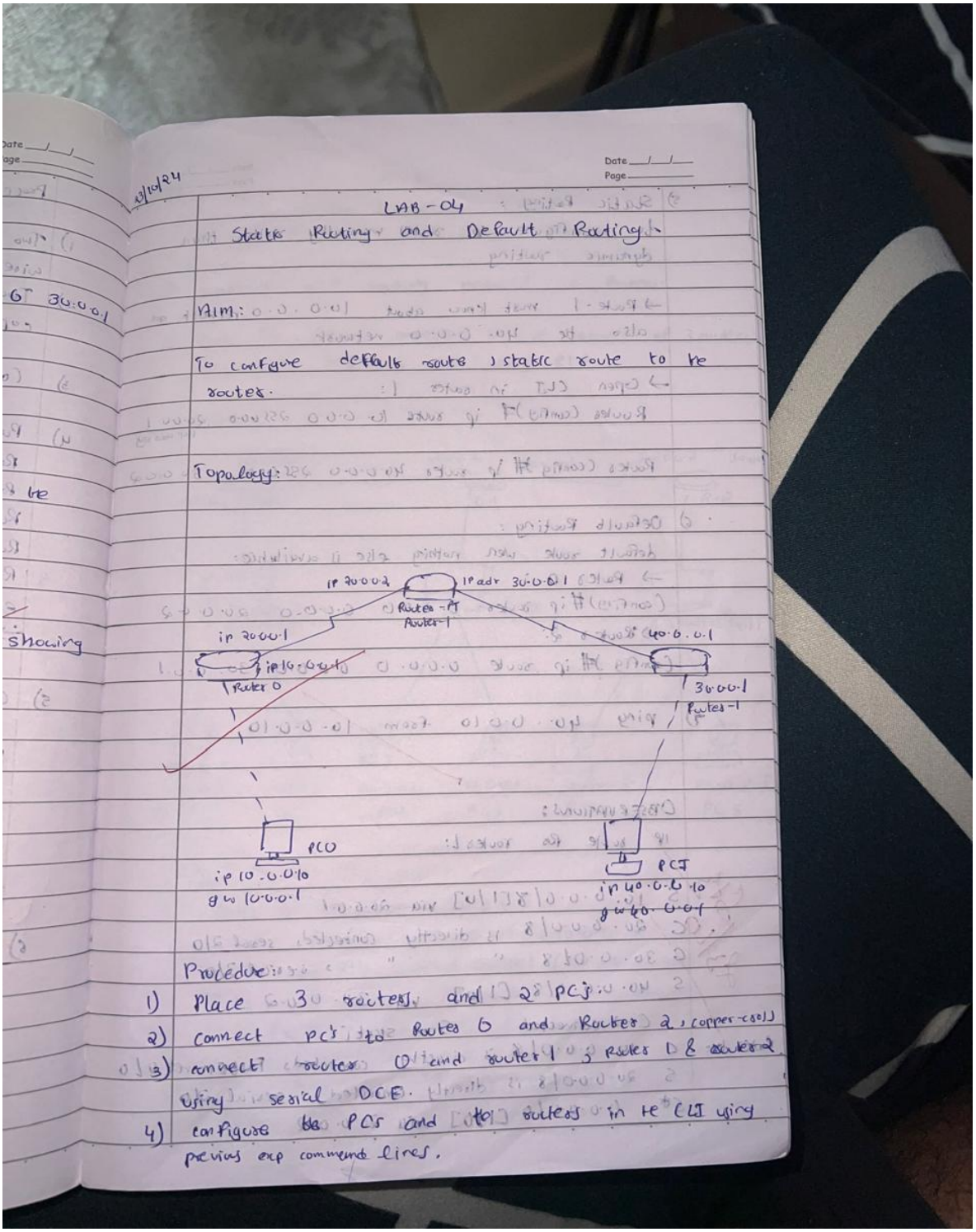


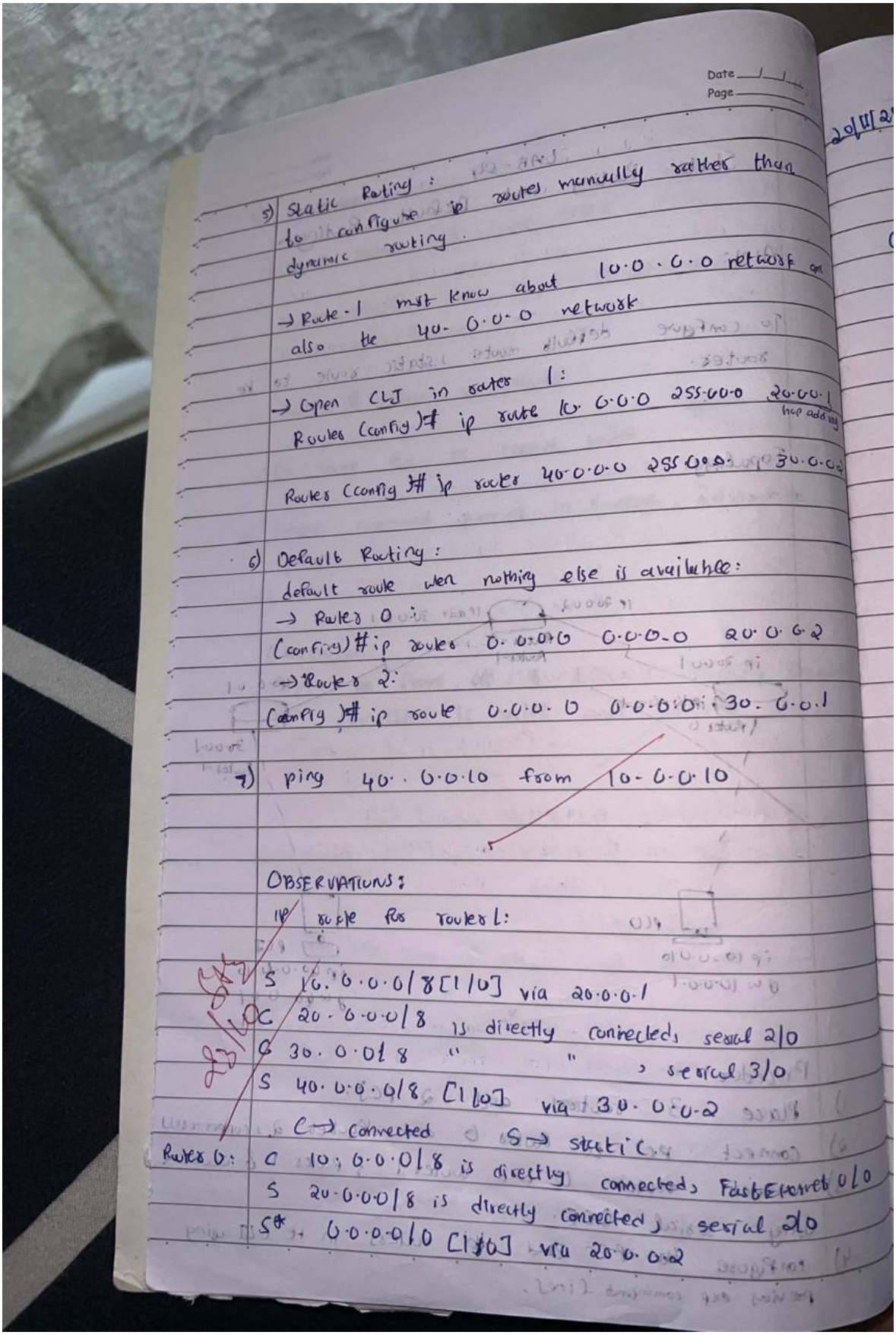


Program 3

Q3: Configure default route, static route to the Router

Observation:





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

→ Route-1 must know about 10.0.0.0 network and 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)

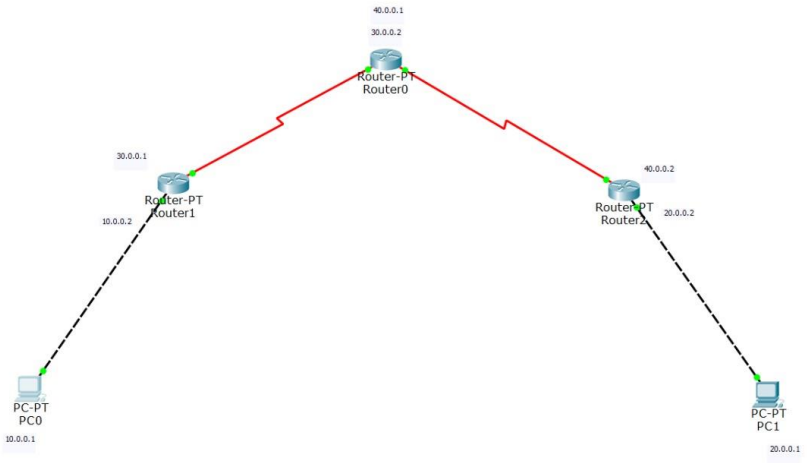
Router (config)# ip router 40.0.0.0 255.0.0.0 30.0.0.2

6) Default Routing :
default route when nothing else is available:
→ Router 0 :
(config)# ip route 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

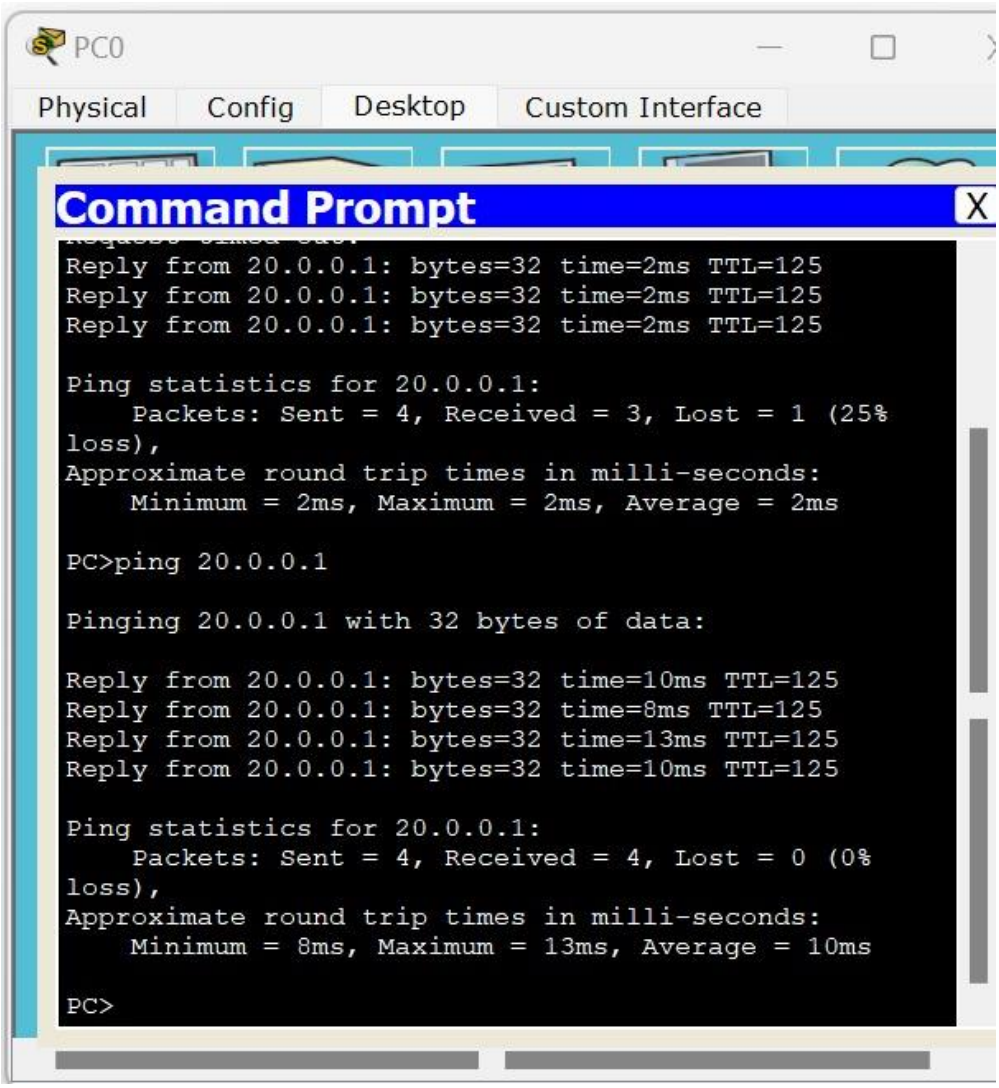
7) ping 40.0.0.10 from 10.0.0.10

OBSERVATIONS:
ip route for router 1:
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 " " , serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
C → connected S → static
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* 0.0.0.0/0 [1/0] via 20.0.0.2

Screenshot of Topology:



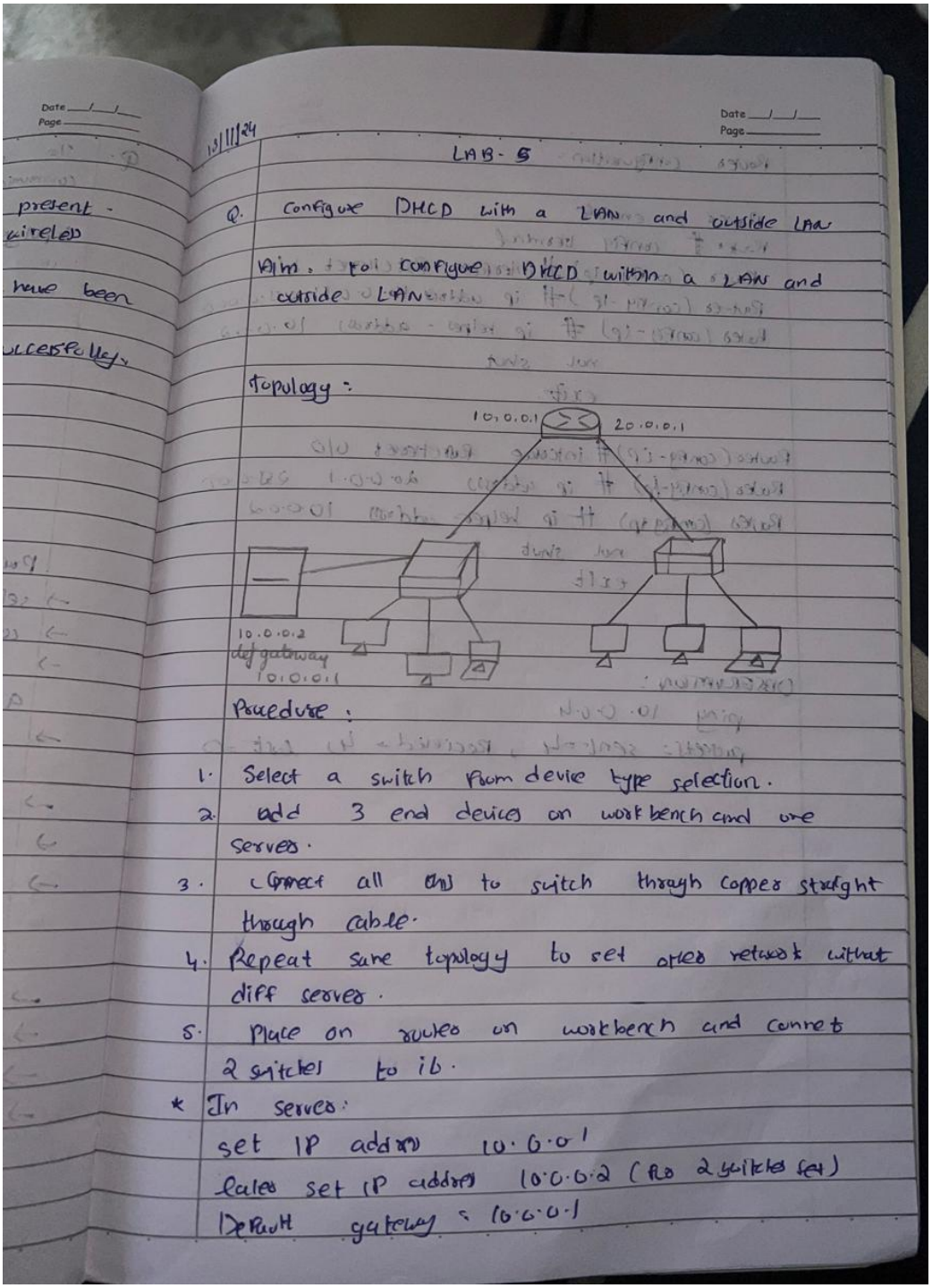
Screenshot of Output:

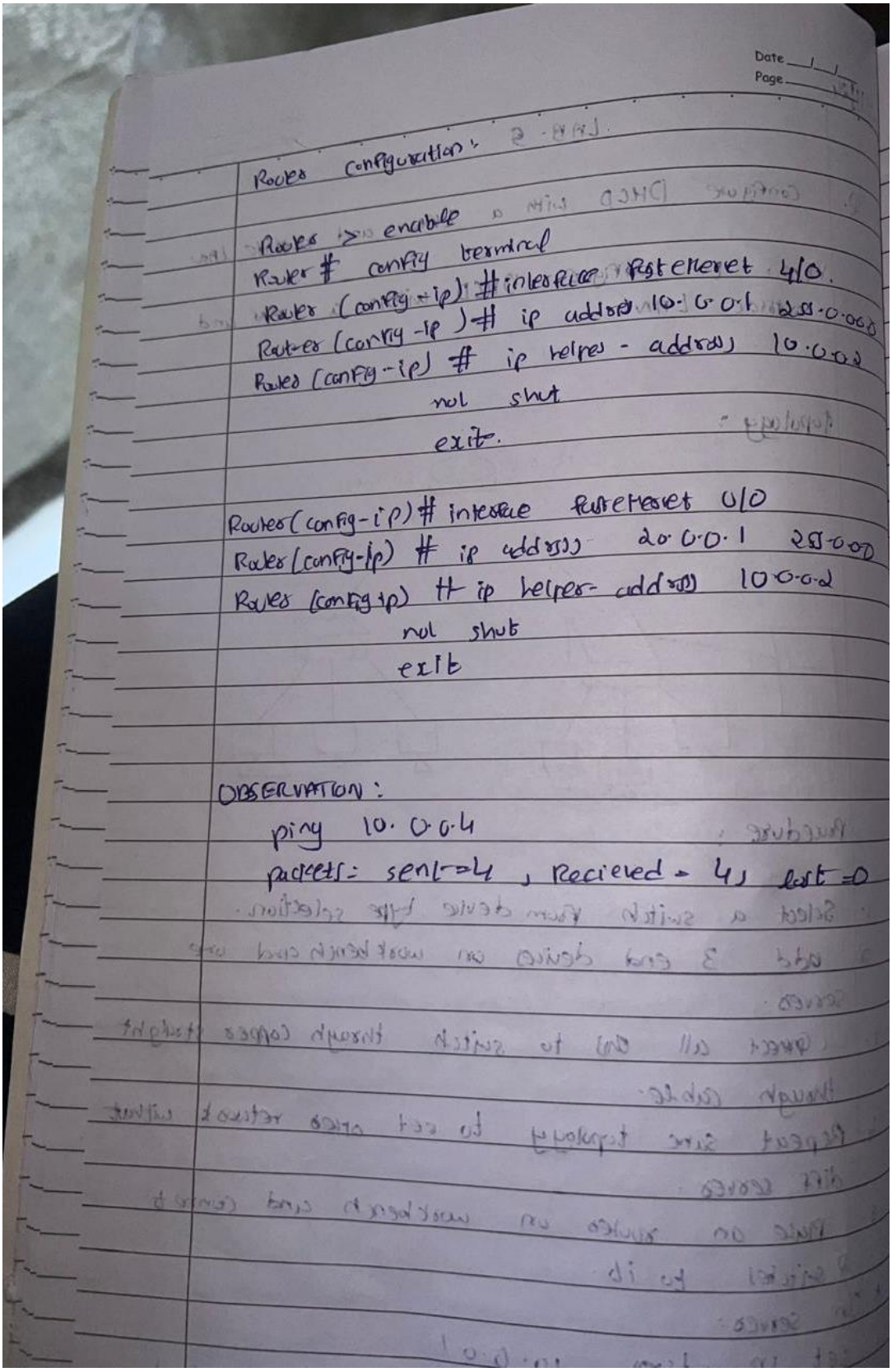


Program 4

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

Observation:





Router configuration:

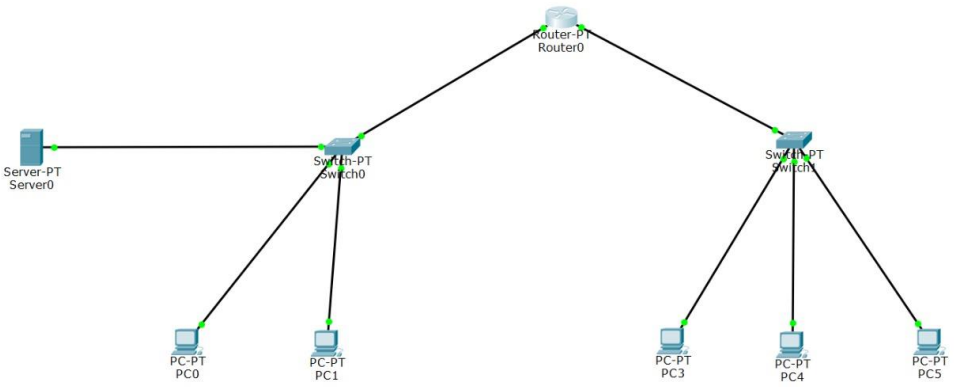
Router > enable
Router # config terminal
Router (config-ip) # interface fastethernet 4/0.
Router (config-ip) # ip address 10.0.0.1 255.0.0.0
Router (config-ip) # ip helper-address 10.0.0.2
no shut
exit.

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
no shut
exit

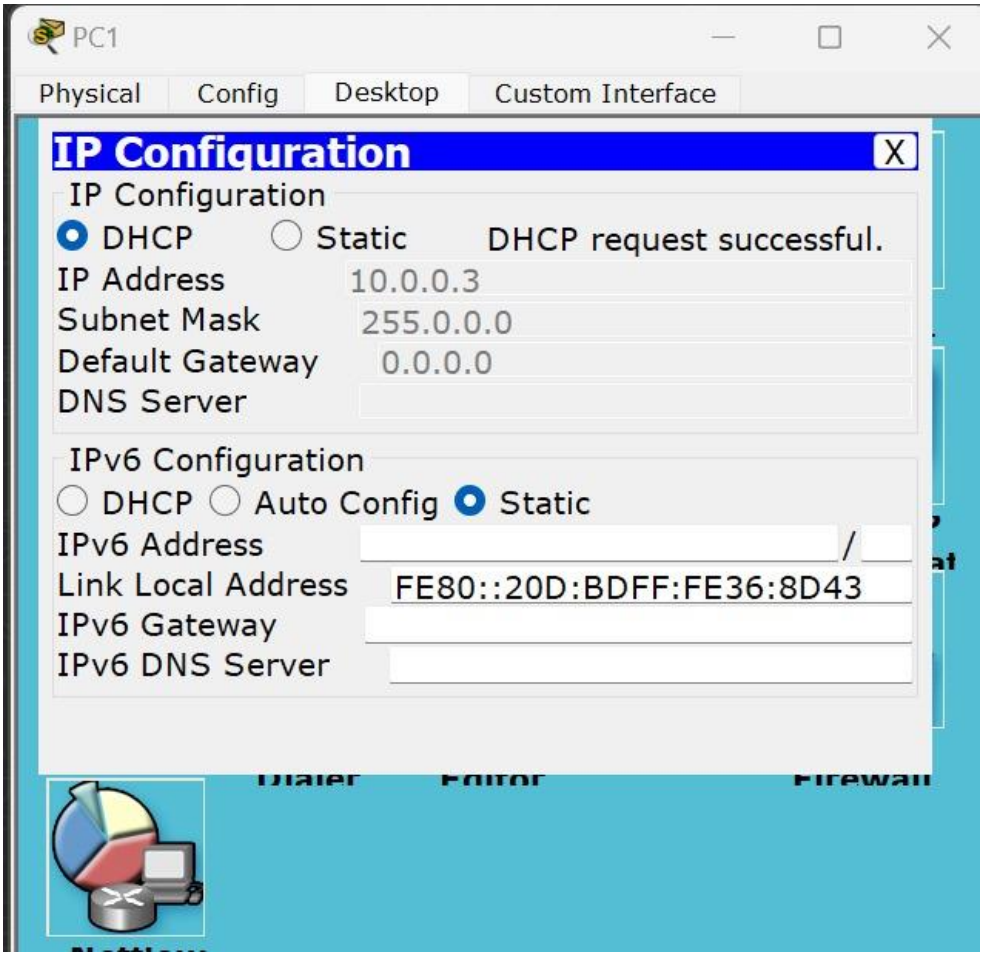
OBSERVATION:

ping 10.0.0.4
packets: sent=4, Recieved = 4, lost = 0

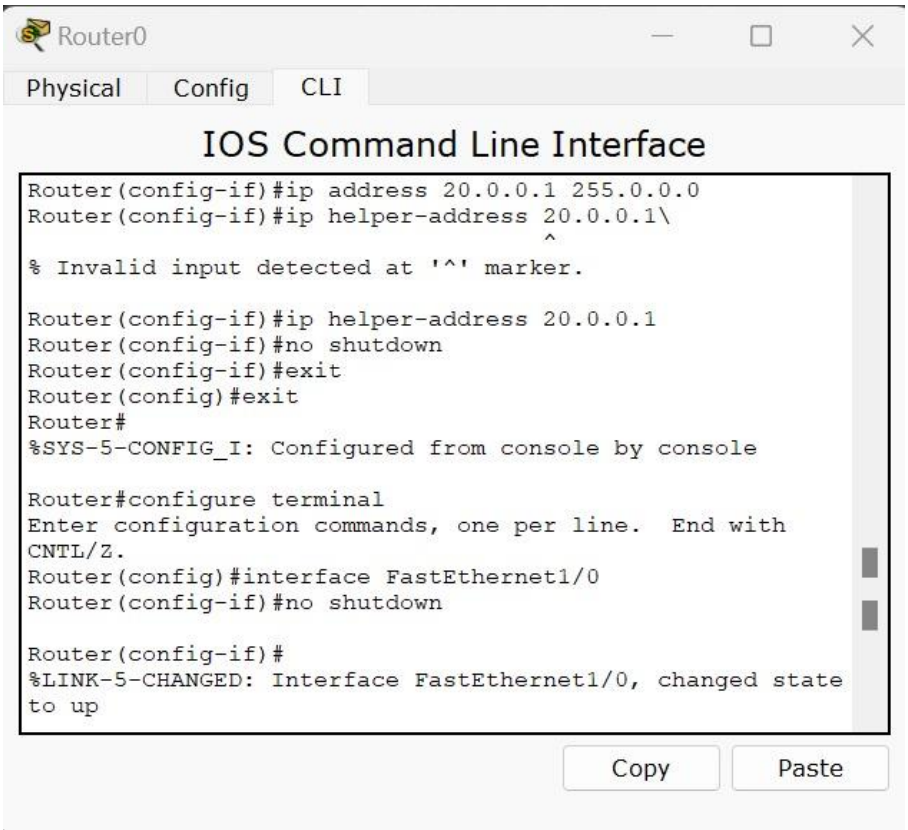
Screenshot of Topology:



Screenshot of Output:



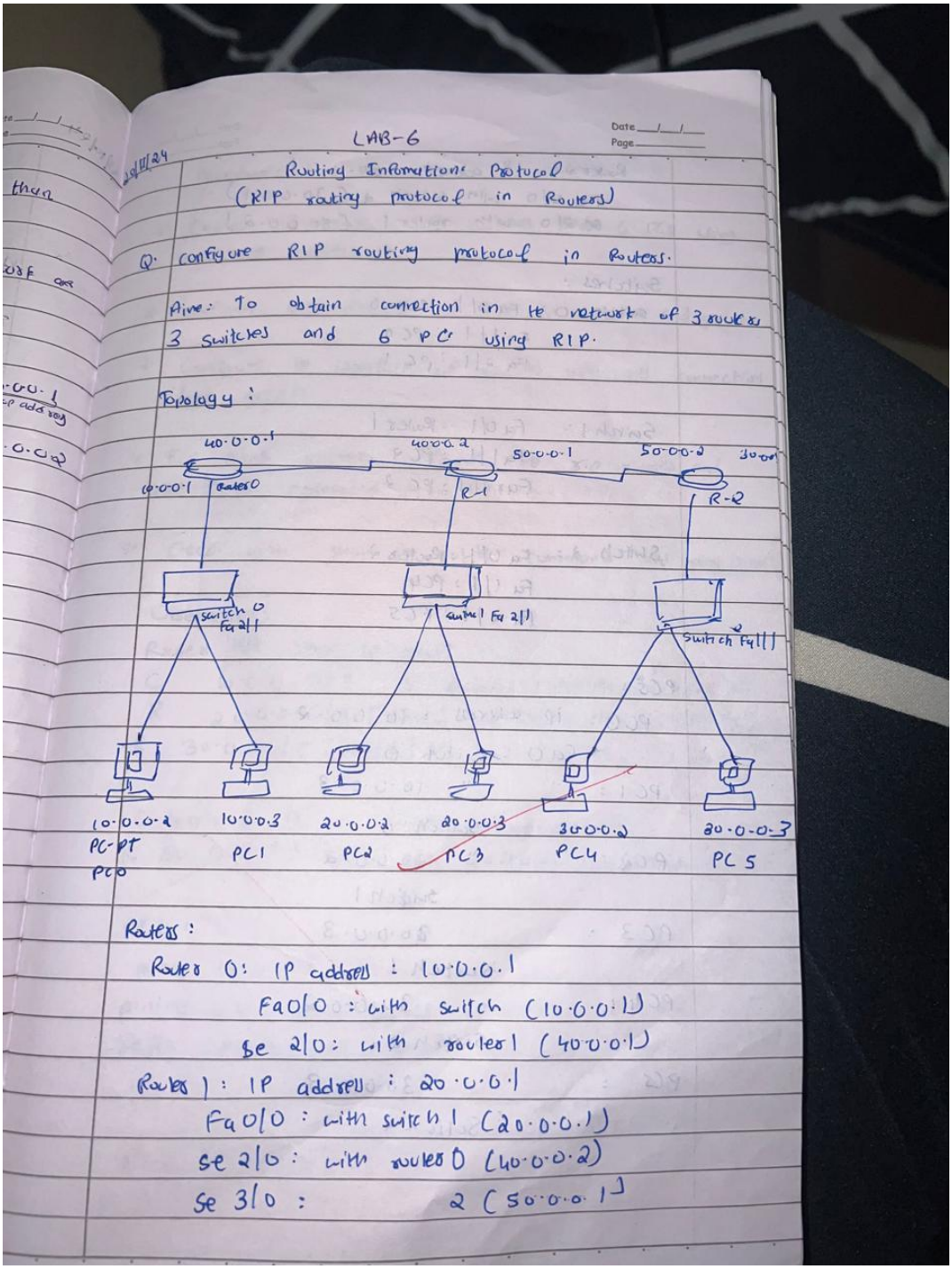

Router0
Physical
Config
CLI
IOS Command Line Interface
Router(config)#int fa 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: 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
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 CNTL/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 2: IP address : 30.0.0.1
Fa 0/0 with switch 2 (30.0.0.1)
se 2/0 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

PC's:
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:

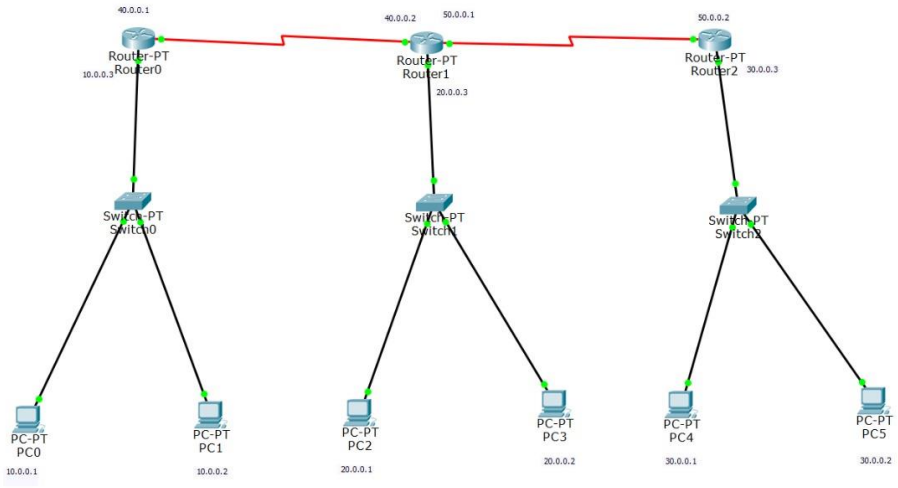
1. Connect 3 routers, 3 switches and 6 PCs using appropriate connections.
2. configure end-devices: 6 PCs ip address and gateway
3. Configure the interface of routers with all connections turn green.
4. For each routers configure the rip routes and add all connected networks.
5. Check with show ip route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

Observation:
Routers ## show ip route
C 10.0.0.0/8 is directly connected, FastEthernet 0/0.
R 20.0.0.0/8 [120/1] via 40.0.0.2 00:00:26, Serial 2/0
R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:26, Serial 2/0

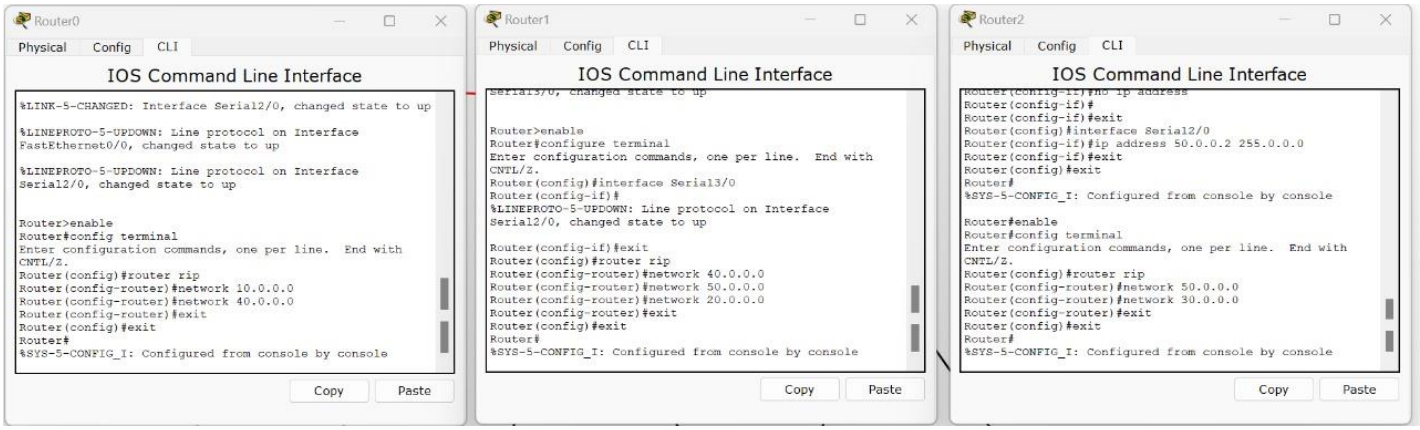
C 40.0.0.0/8 is directly connected 2/0
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:26 Serial 2/0

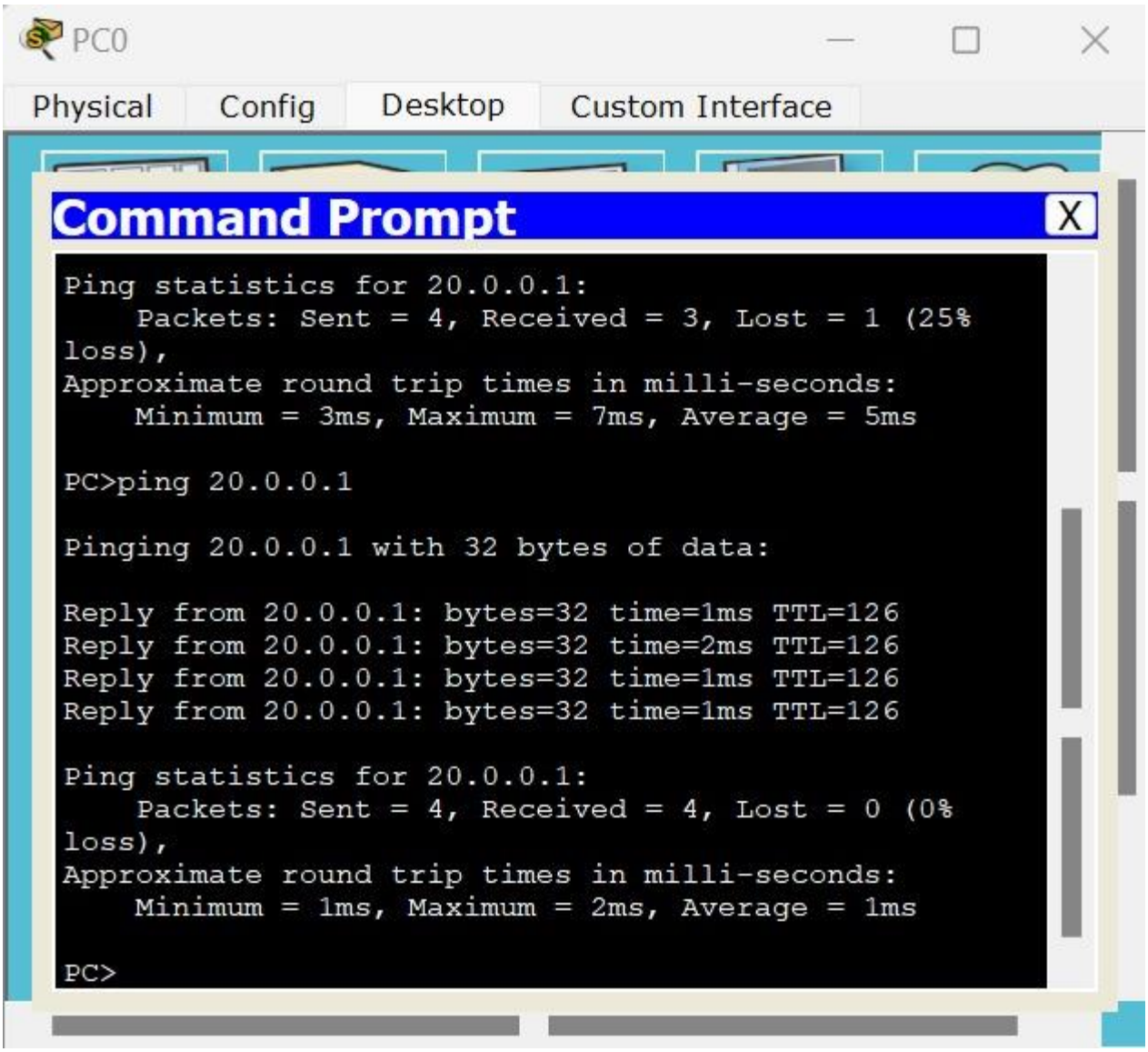
PC 0:
ping 10.0.0.3
pining with 32 bytes of data.
Reply from 10.0.0.3 : bytes 32 ... 7.7L:/8
:
:
ping 20.0.0.2

Screenshot of Topology:



Screenshot of Output:

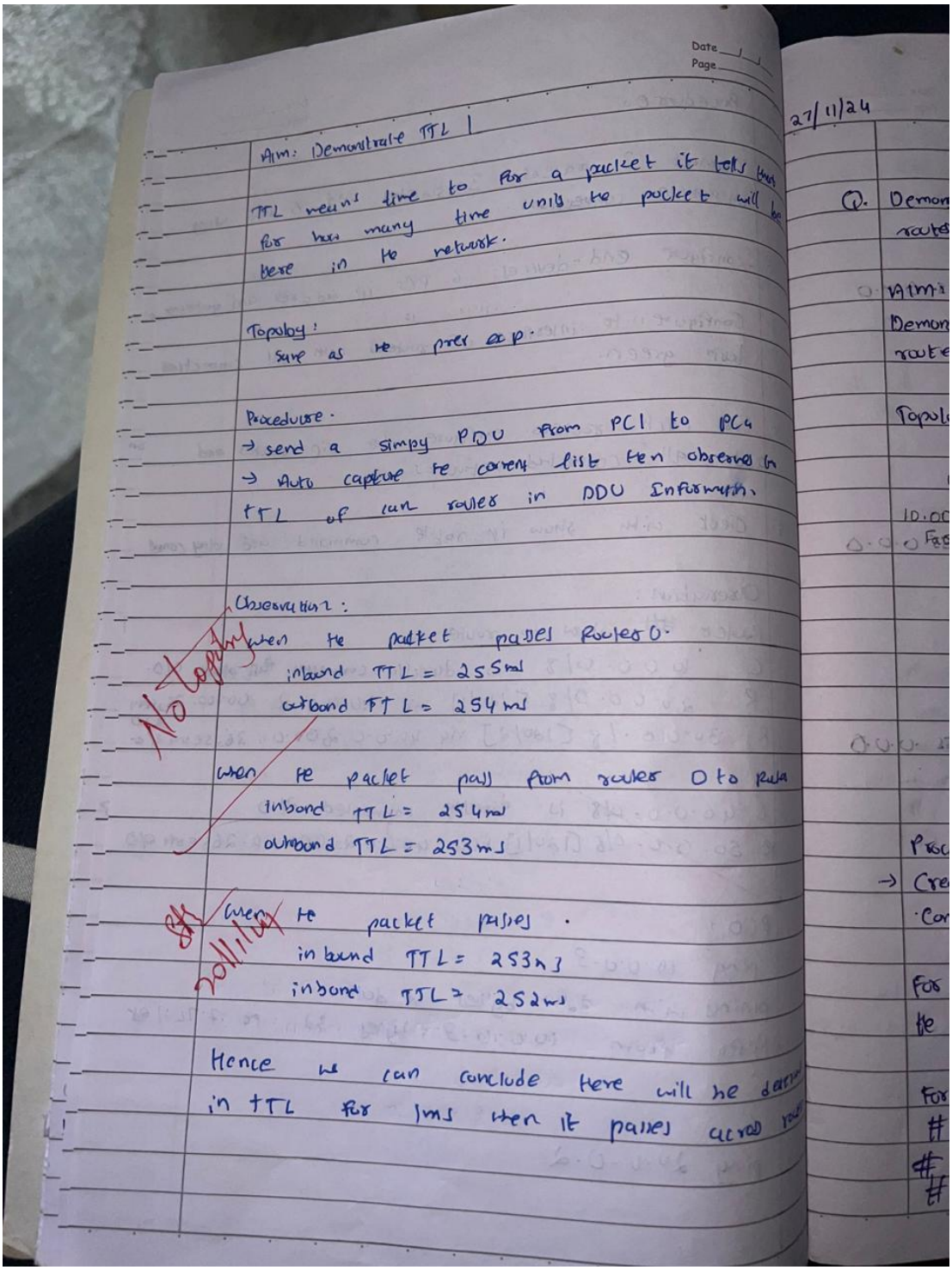



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 3ms, Maximum = 7ms, Average = 5ms
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 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>

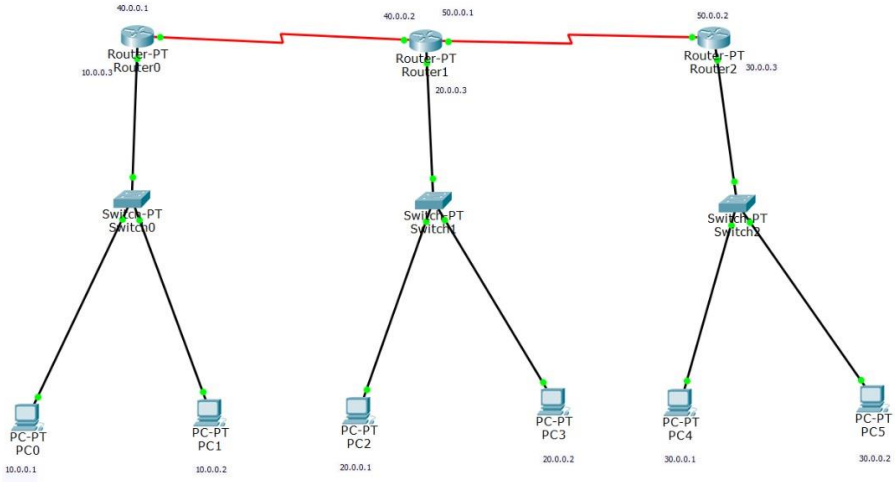
Program 6

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

Observation



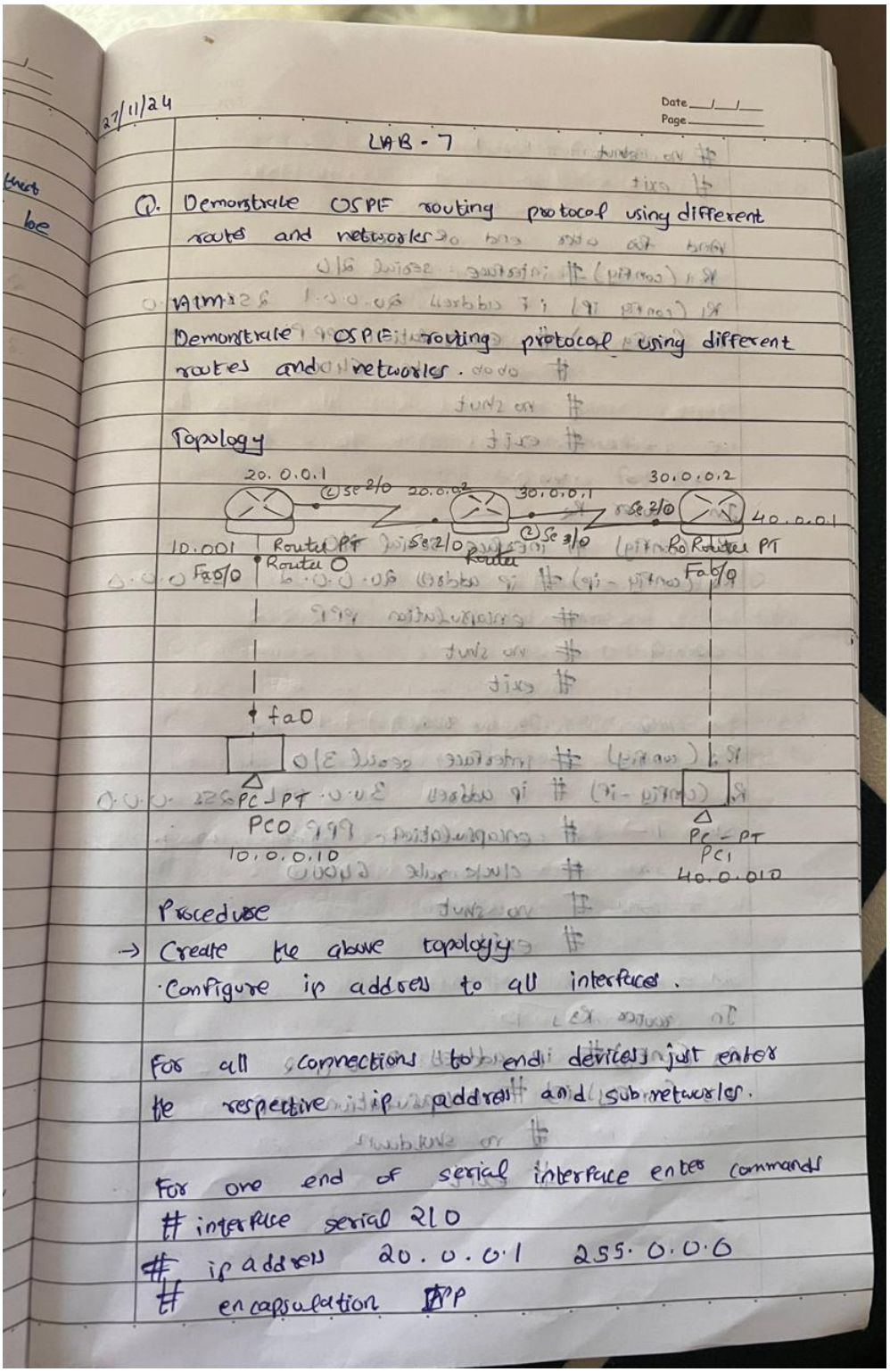
Screenshot of Topology:

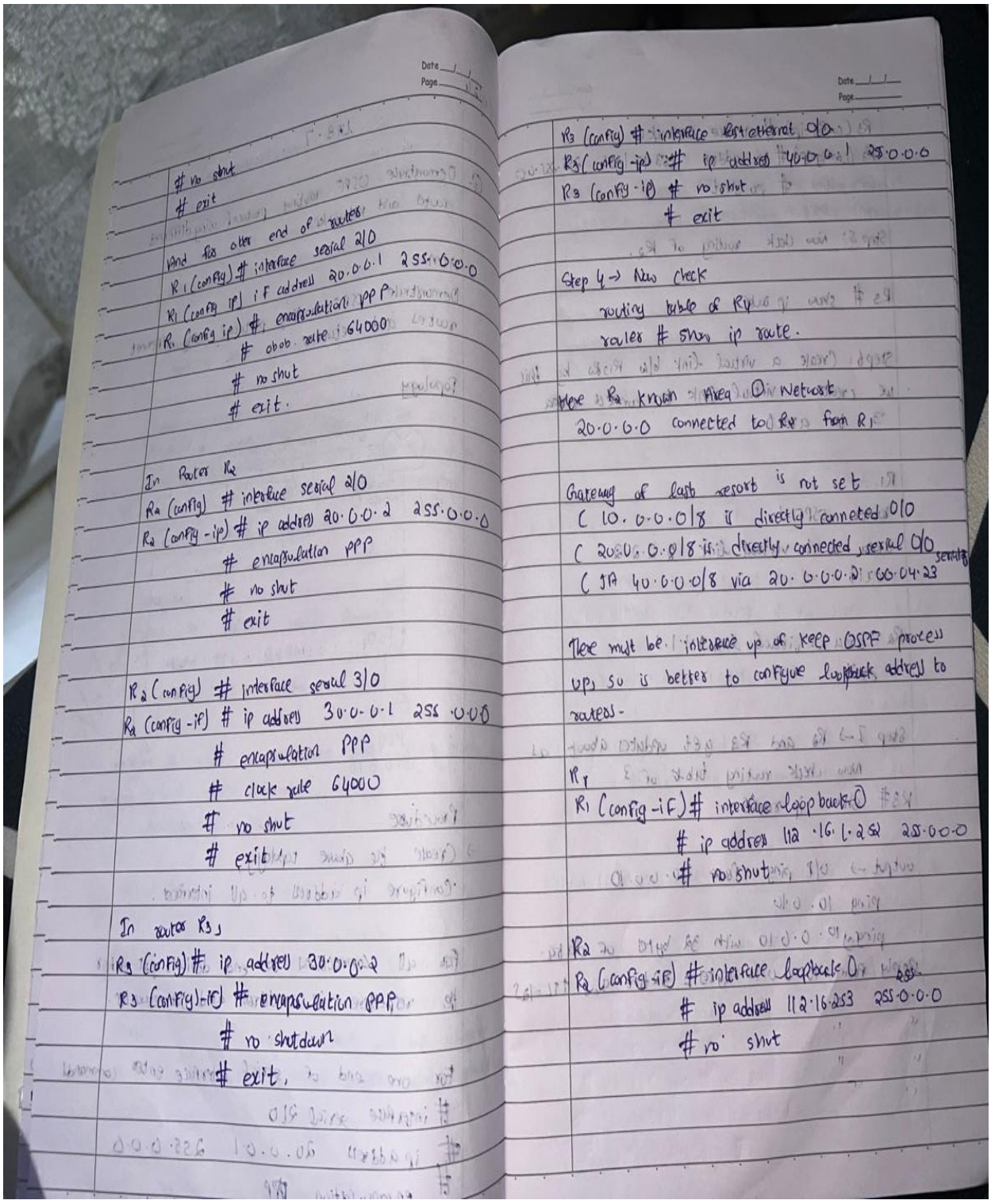


Program 7

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

Observation:





no shut
exit
Hand Rs other end of routers.
R1 (config)# interface serial 2/0
R1 (config if) # if address 20.0.0.1 255.0.0.0
R1 (config if) # encapsulation ppp
clock rate 64000
no shut
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 shut
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 shut
exit

In router R3,
R3 (config)# ip address 30.0.0.2
R3 (config-if) # encapsulation ppp
no shutdown
exit.

R3 (config) # interface fastethernet 0/0
R3 (config-if) # ip address 40.0.0.1 255.0.0.0
R3 (config-if) # no shut
exit

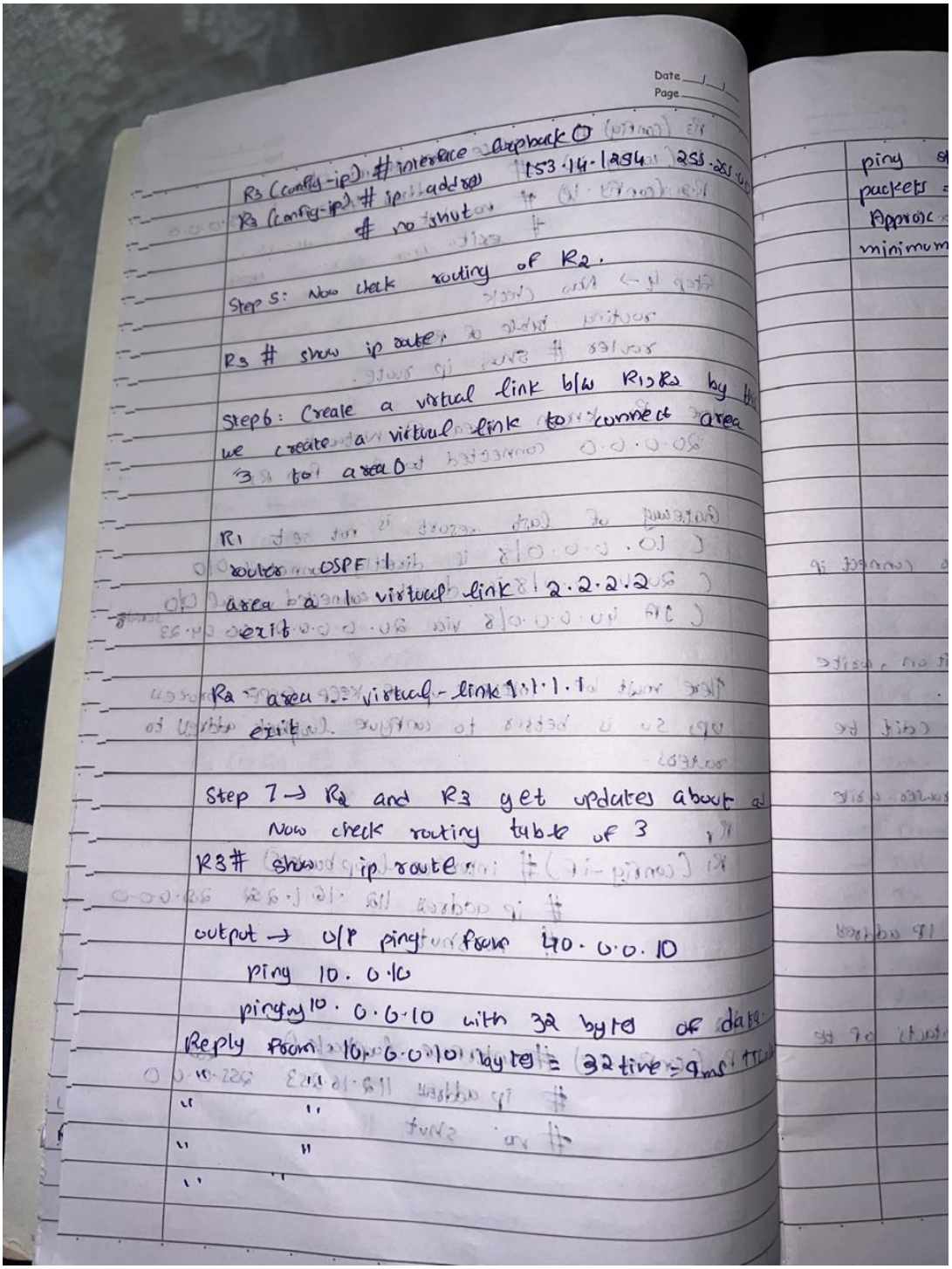
Step 4 → Now check
routing table of R1
router # show ip route.
Here R2 known area O Network
20.0.0.0 connected to R2 from R1

Gateway of last resort is not set
C 10.0.0.0/8 is directly connected 0/0
C 20.0.0.0/8 is directly connected, serial 0/0
O IA 40.0.0.0/8 via 20.0.0.2, 00:04:23 serial

There must be interface up of keep OSPF process up, so is better to configure loopback address to routers.

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

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



R3 (config-if) # interface loopback 0
R3 (config-if) # ip address 153.14.1.254 255.255.0.0
no shut

Step 5: Now check routing of R2.

R3 # show ip route

Step 6: Create a virtual link b/w R1, R3 by the
we create a virtual link to connect area
3 to area 0

R1
router OSPF 1
area 2 virtual link 2.2.2.2
exit

R2 - area 2 virtual-link 1.1.1.1
exit

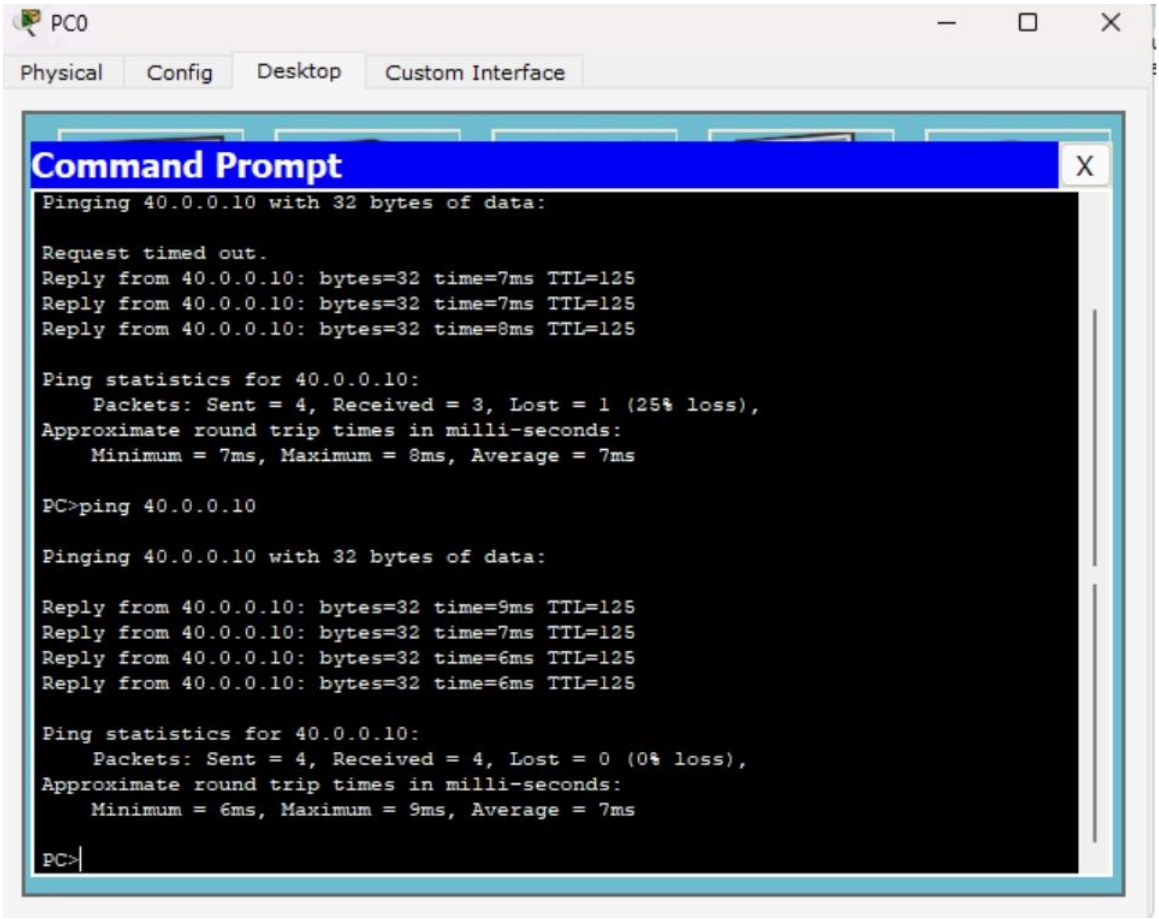
Step 7 → R2 and R3 get updates about
Now check routing table of 3
R3# show ip route

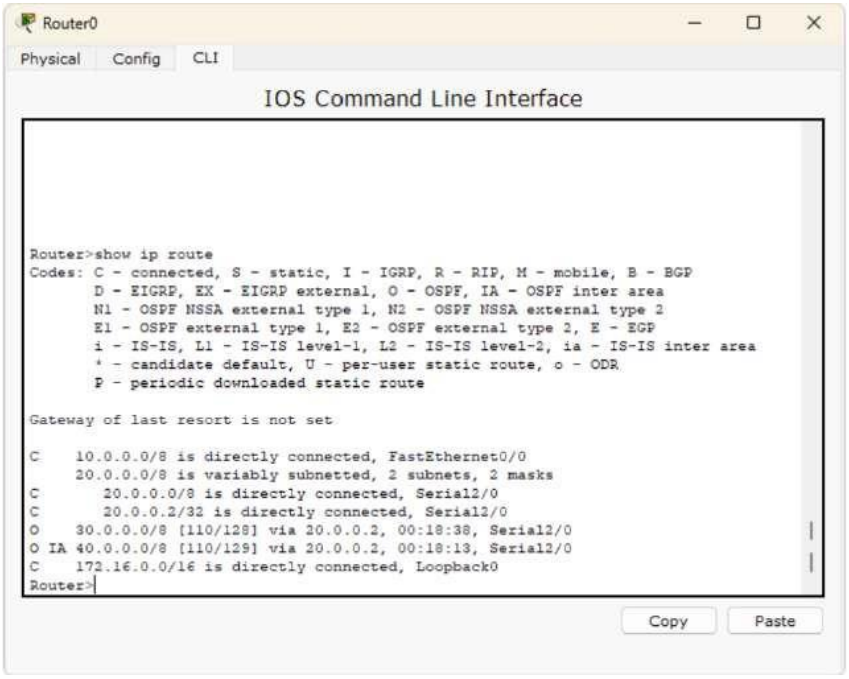
output → o/p ping from 10.0.0.10
ping 10.0.10
ping 10.0.0.10 with 32 byte of data
Reply from 10.0.0.10 byte = 32 time = 9ms TTL
"
"
"

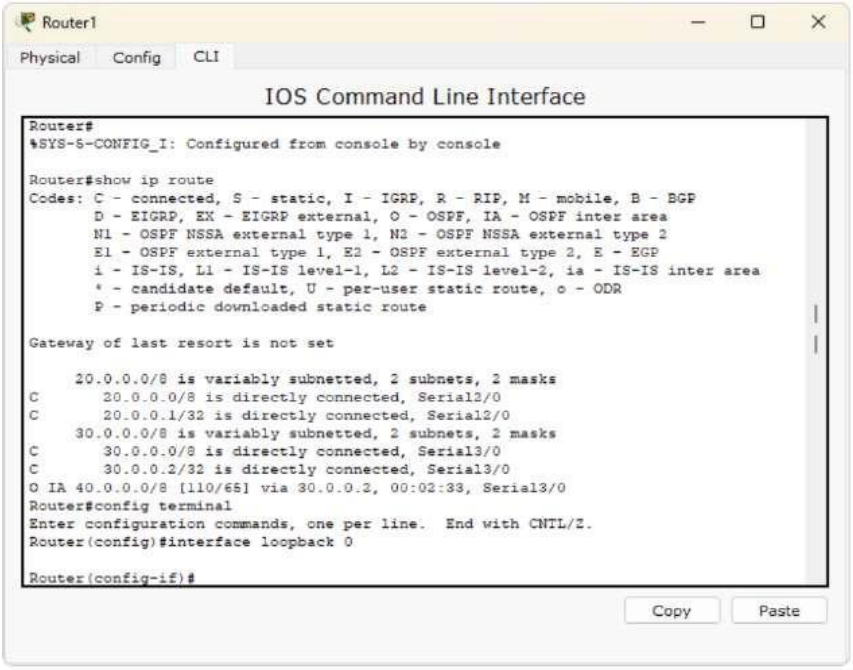
piny
packets
Approx
minimum

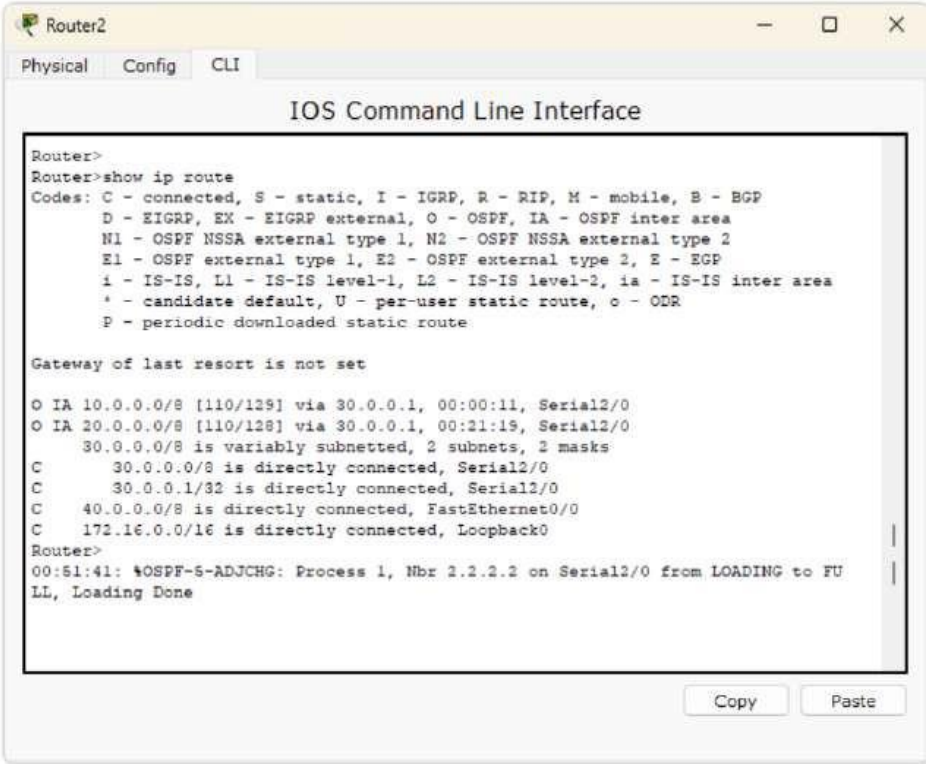
Screenshot of Topology:

Screenshot of Output:




Router0
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
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>
Copy
Paste


Router1
Physical
Config
CLI
IOS Command Line Interface
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)#
Copy
Paste

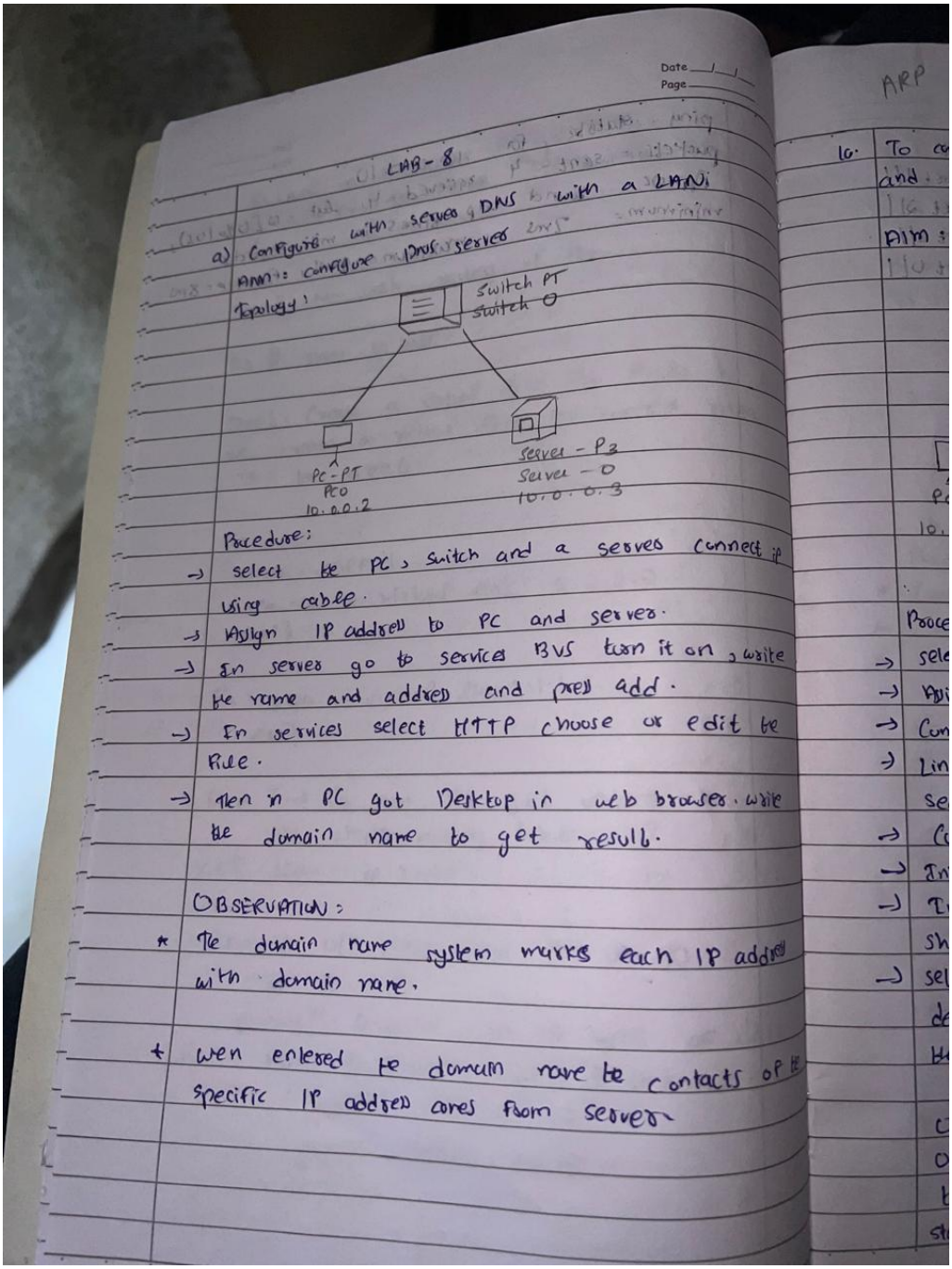

Router2
Physical
Config
CLI
IOS Command Line Interface
Router>
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external 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
Copy
Paste

Program 8

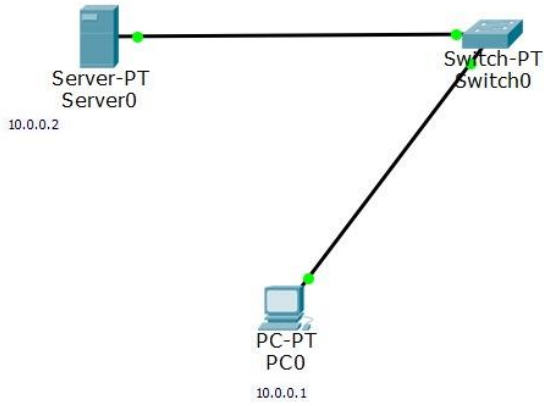
Q8: Configure Web Server, DNS within a LAN

Observation:

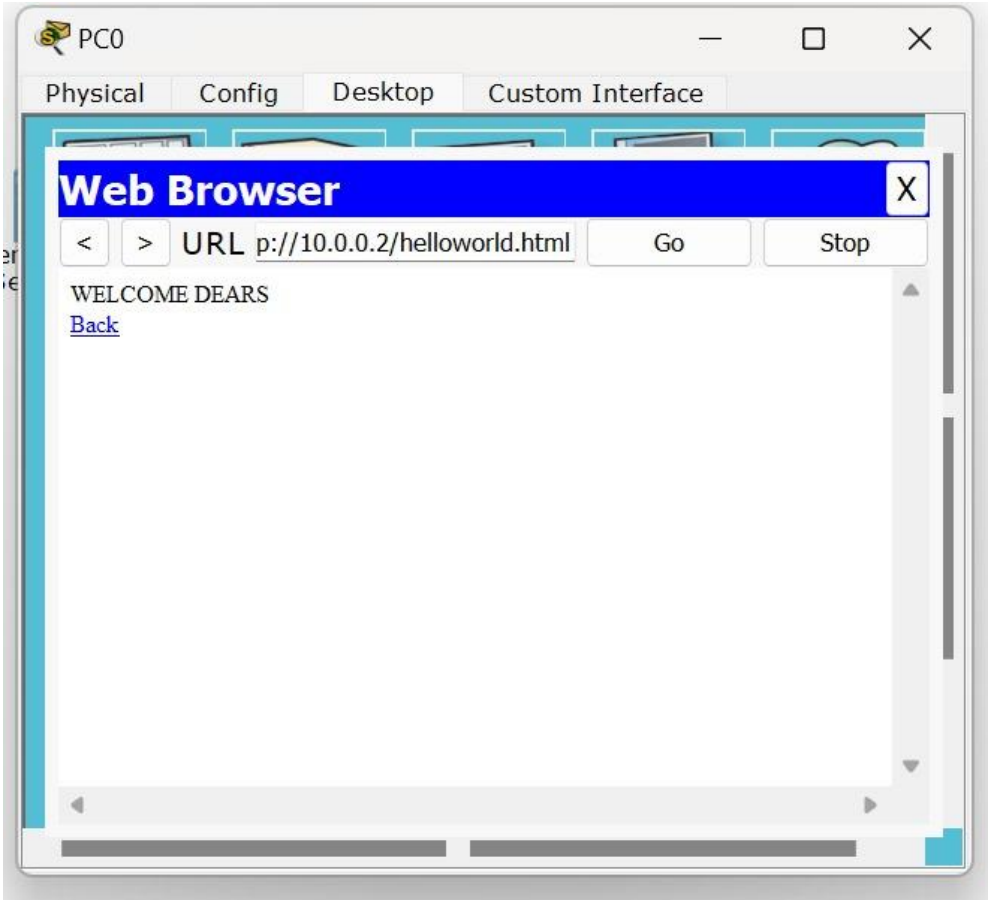
\



Screenshot of Topology:



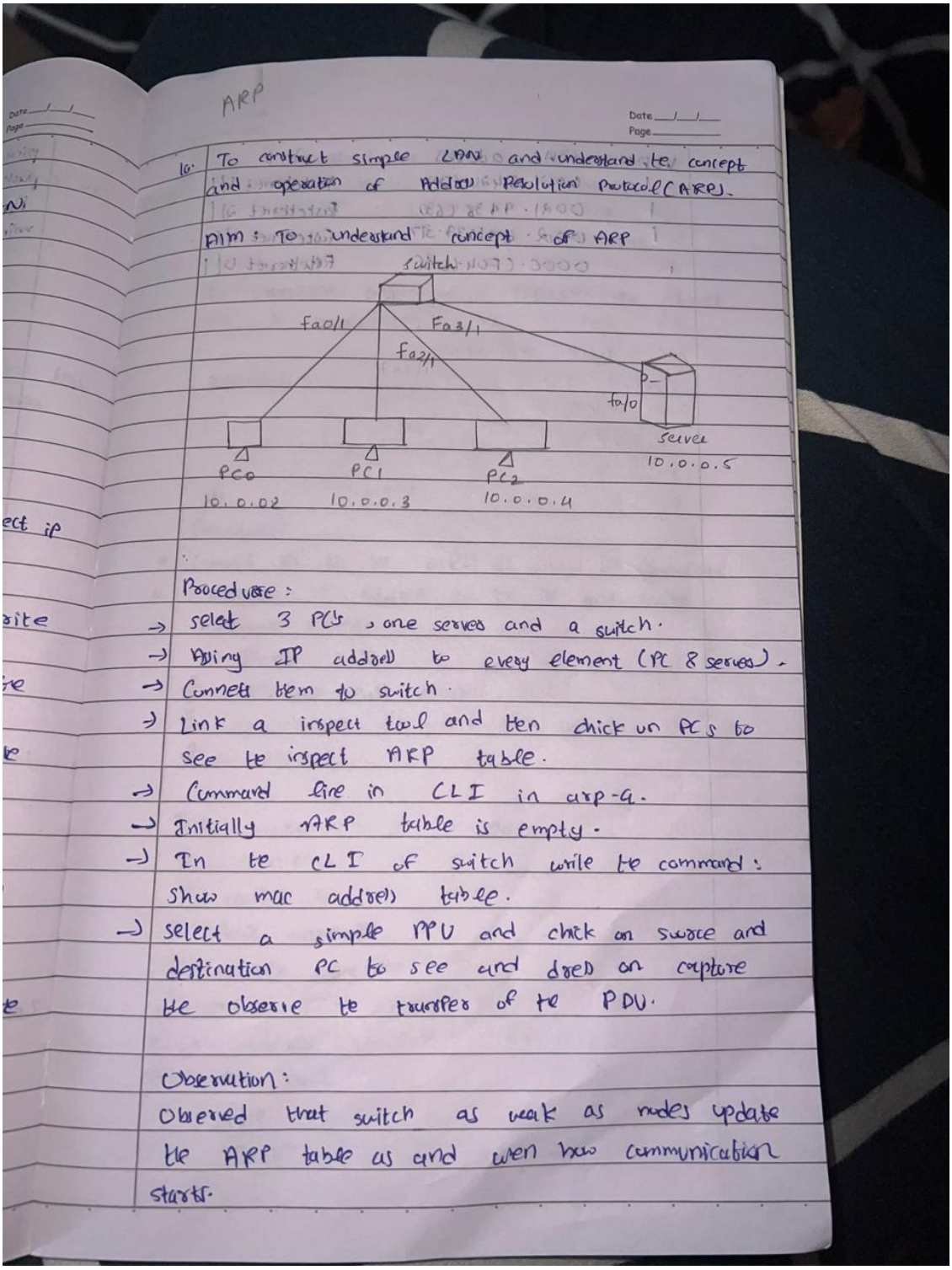
Screenshot of Output:

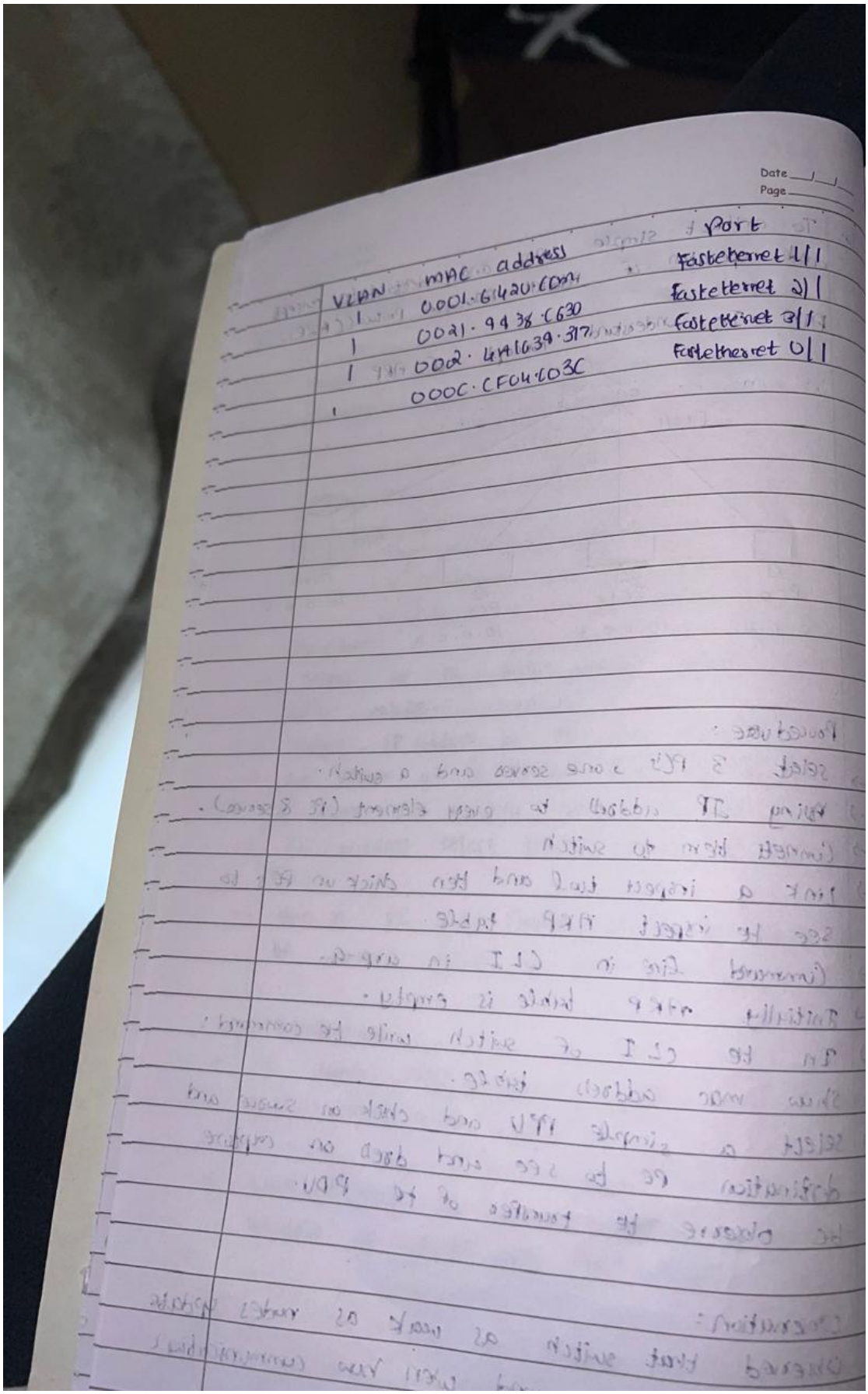


Program 9:

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

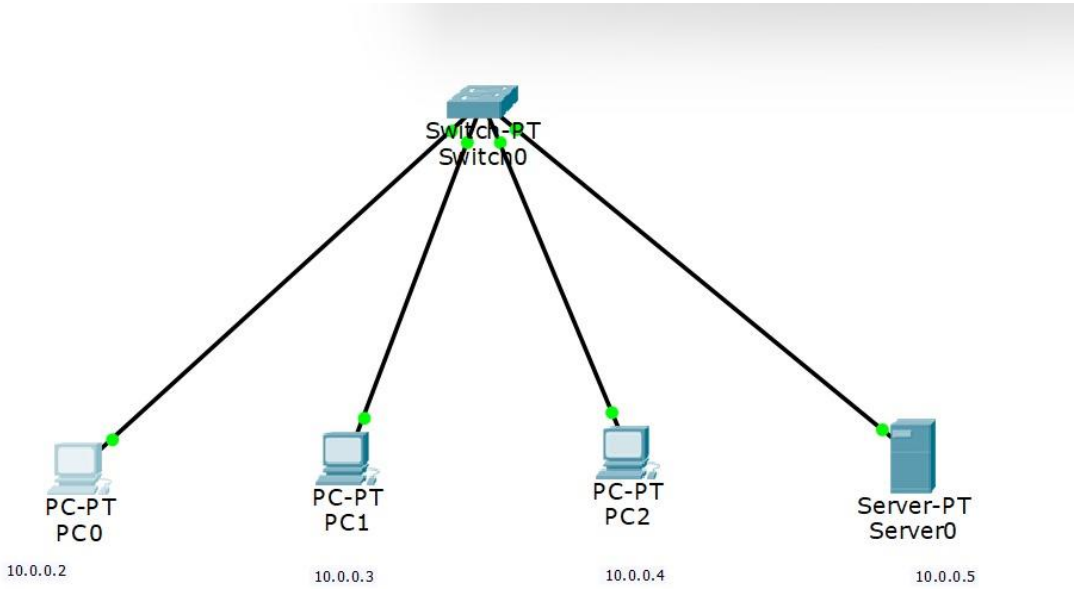
Observation:



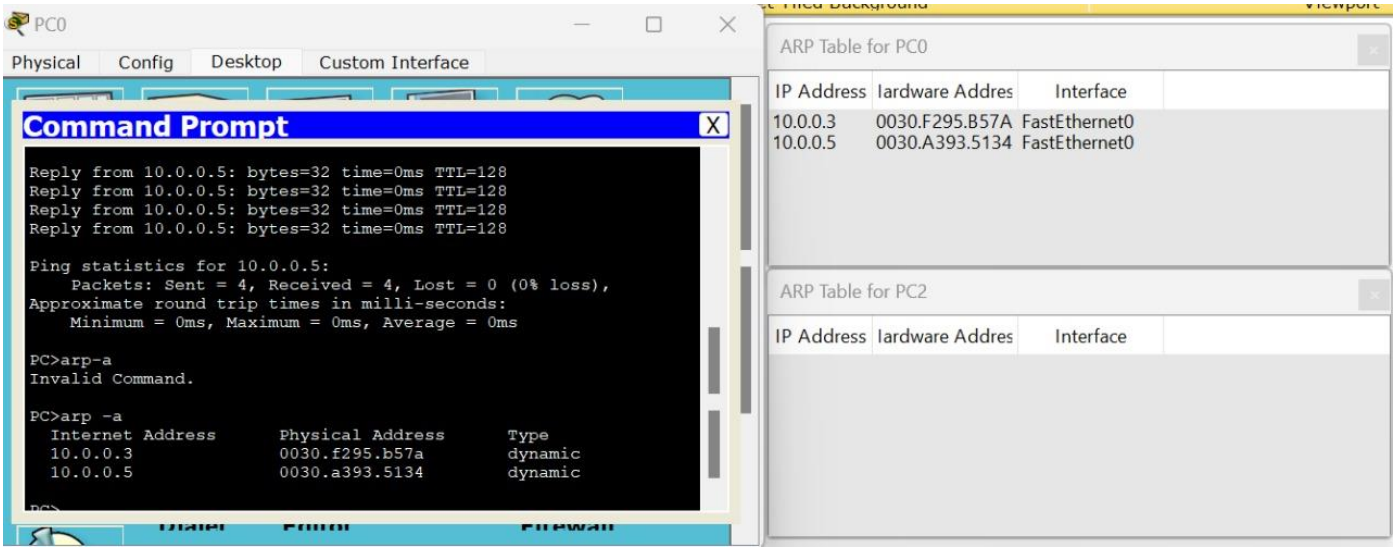


VLAN	MAC address	Port
1	0001.6420.C0A1	FastEthernet 1/1
1	0021.9438.C630	FastEthernet 2/1
1	0002.4A1639.317	FastEthernet 3/1
1	000C.CF04.603C	FastEthernet 0/1

Screenshot of Topology:



Screenshot of Output:

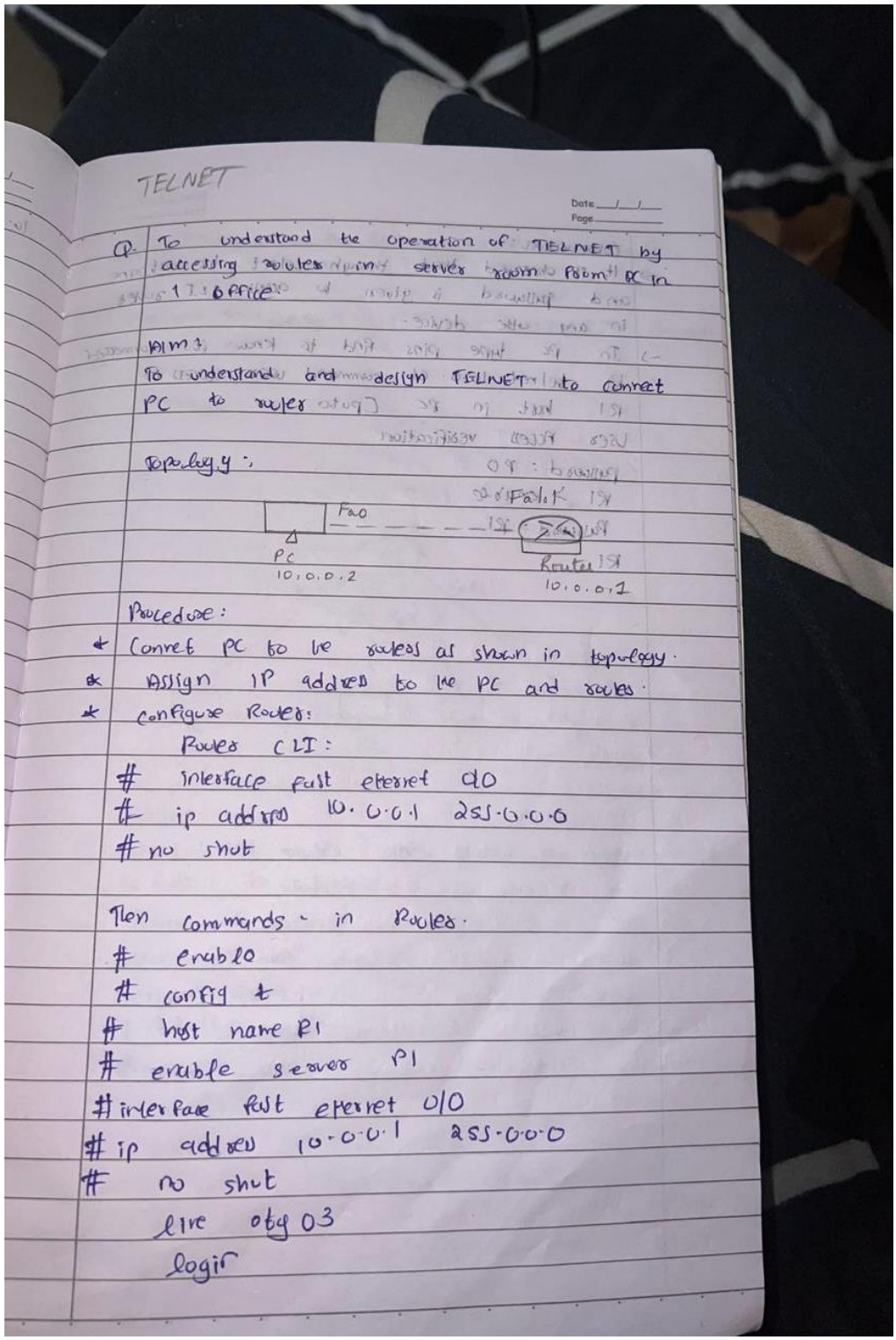


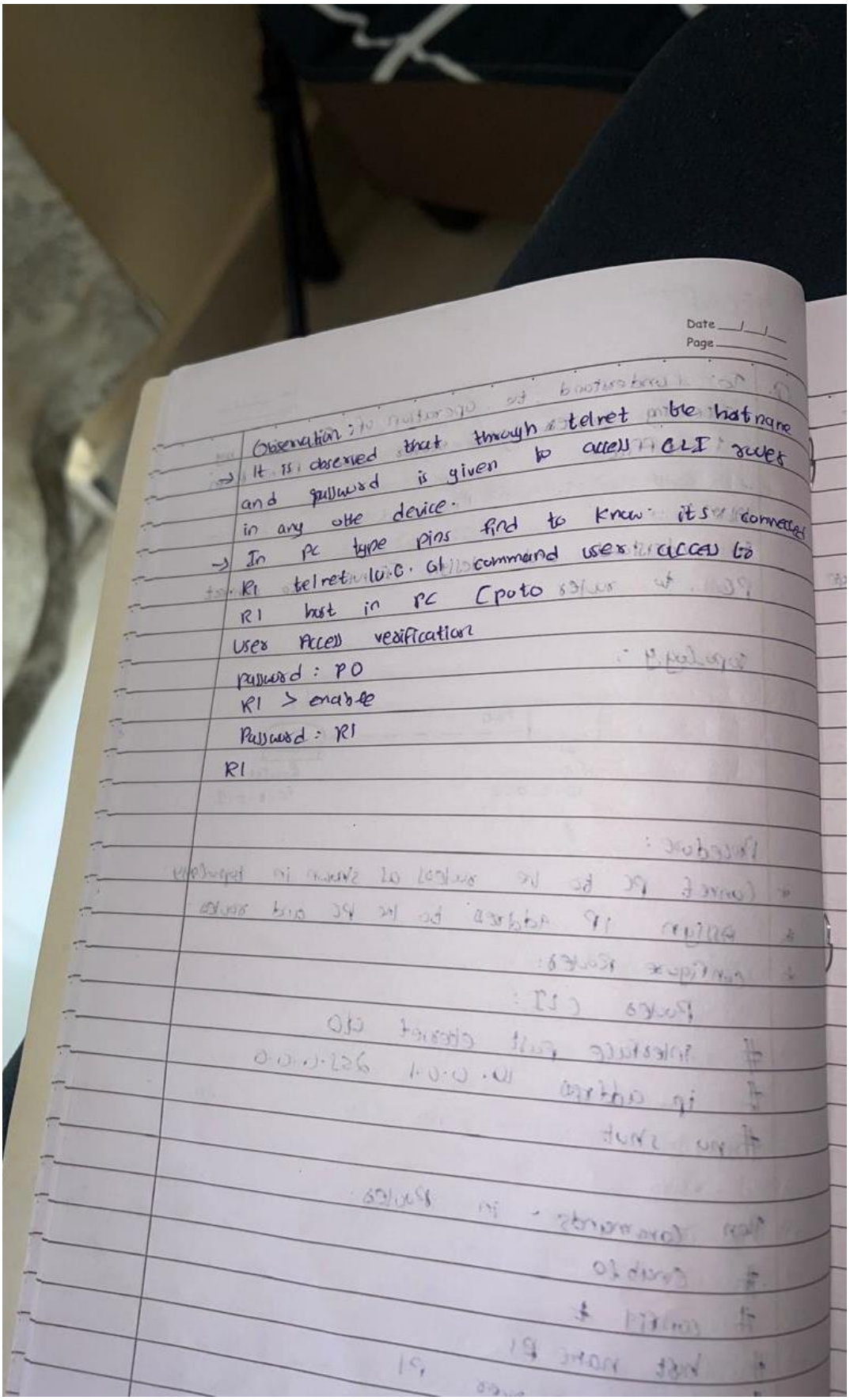

Switch0
Physical
Config
CLI
IOS Command Line Interface
Switch>show mac address table
^
% Invalid input detected at '^' marker.
Switch>show mac address
Mac Address Table
Vlan Mac Address Type Ports
1 0005.5eb7.8cb1 DYNAMIC Fa0/1
1 0005.5ebb.5671 DYNAMIC Fa2/1
1 0030.a393.5134 DYNAMIC Fa3/1
1 0030.f295.b57a 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:




Date __/__/__
Page

Observation :

→ It is observed that through telnet the hostname and password is given to access CLI router in any other device.

→ In PC type pins find to know its connected R1 telnet 10.0.0.61 command user access to R1 host in PC Cpoto

User Access verification

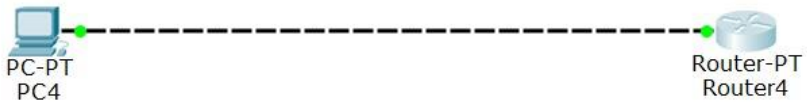
Password : PO

R1 > enable

Password : R1

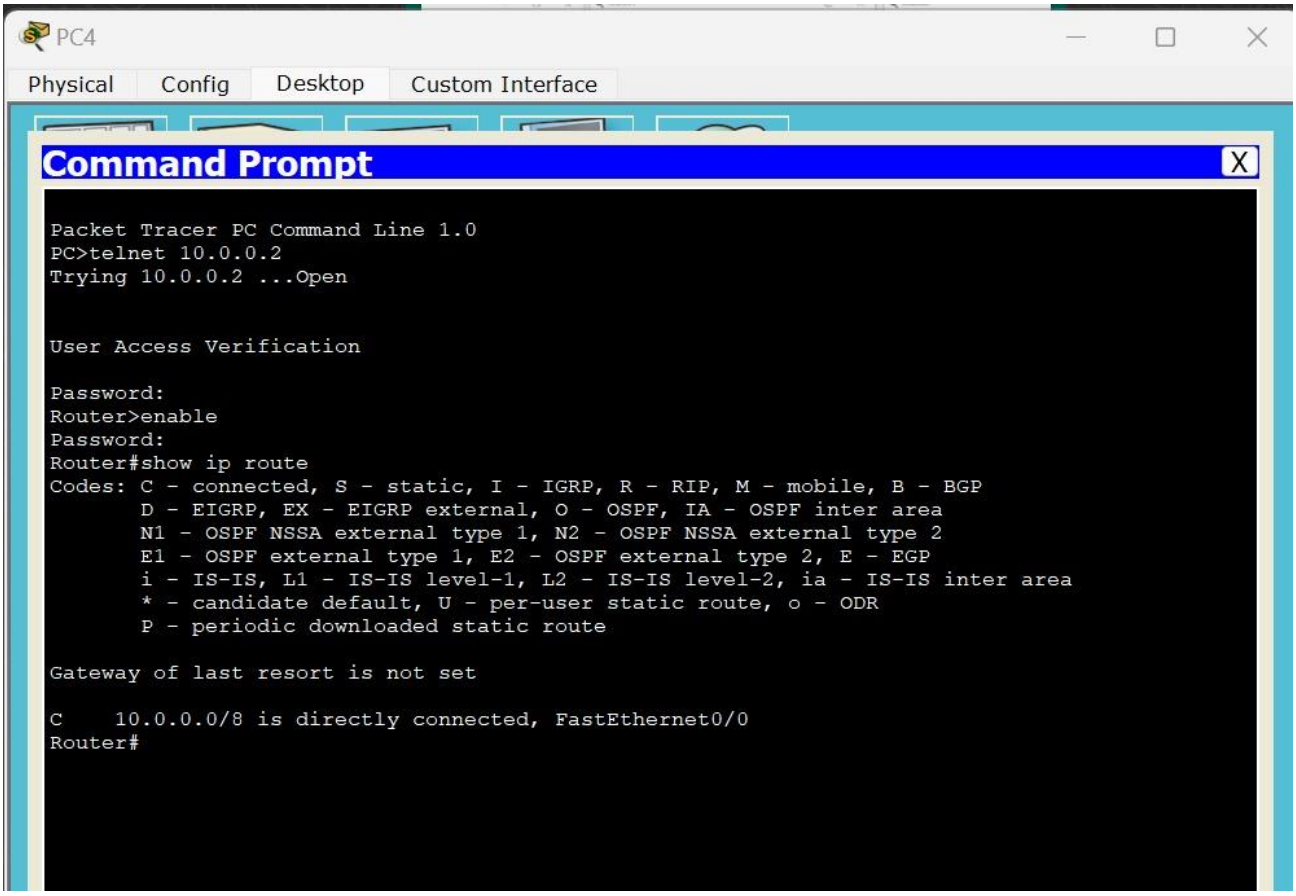
R1

Screenshot of Topology:



Screenshot of Output:

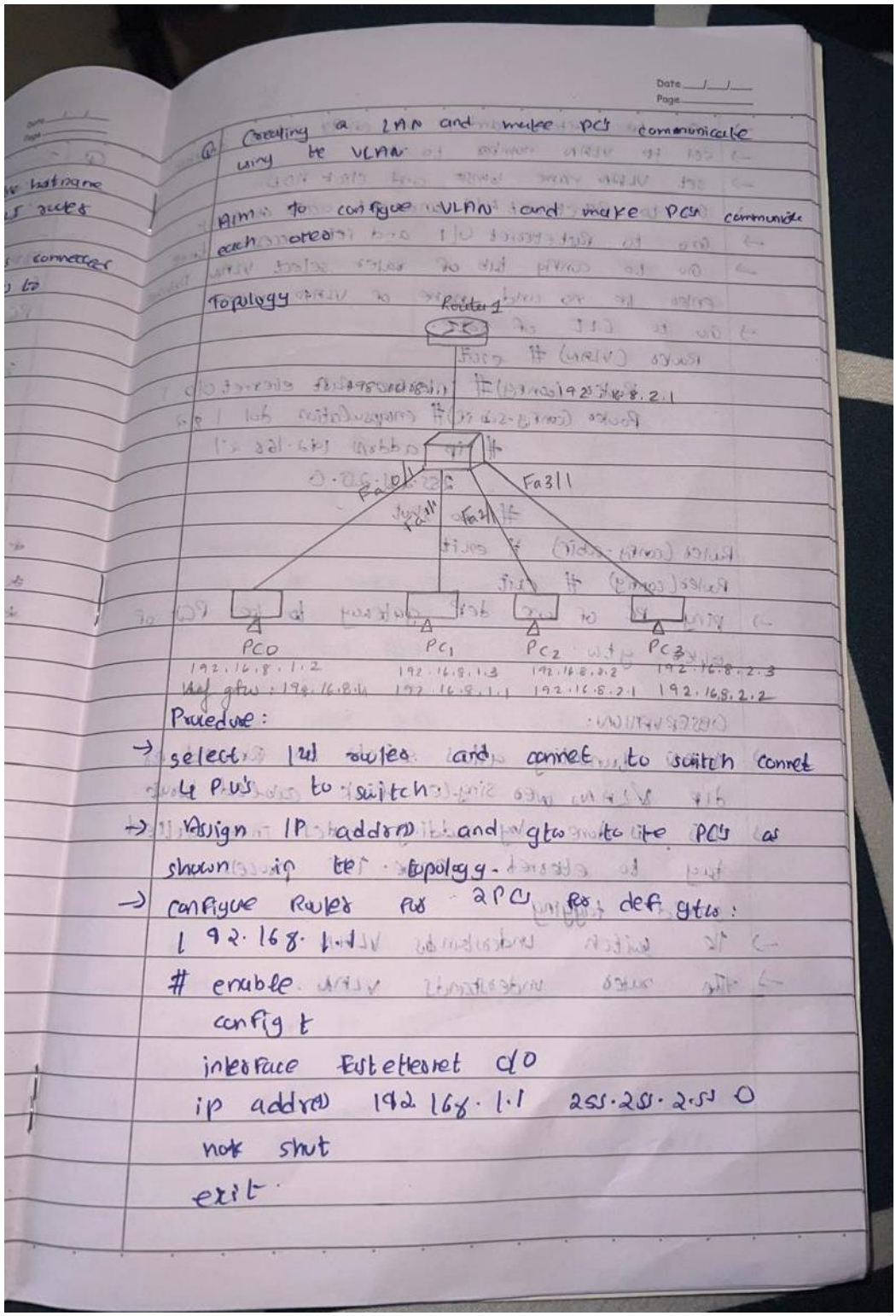
```
Router#enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#enable secret chris
Router(config)#line vty 0 5
Router(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login 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
```

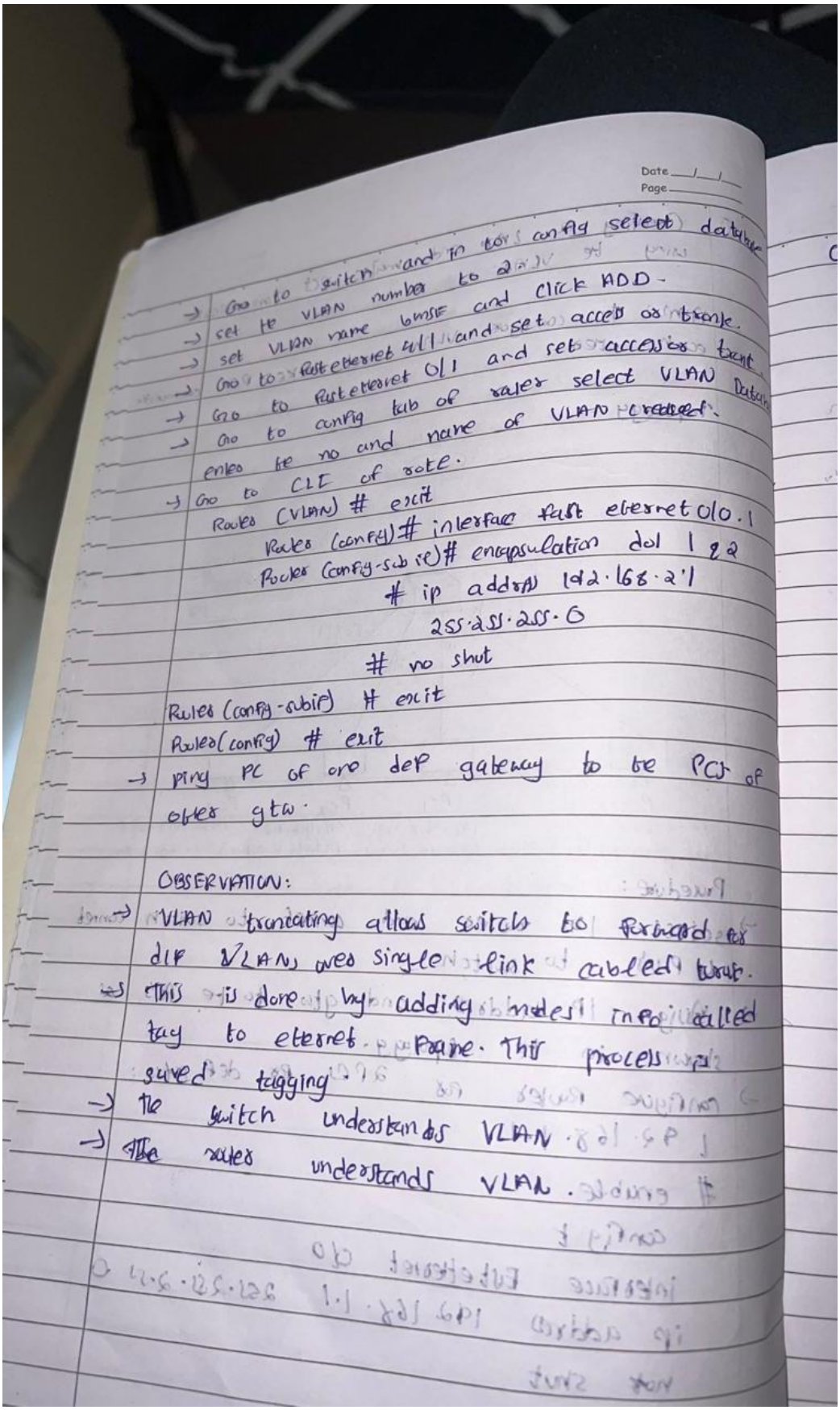

PC4
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open
User Access Verification
Password:
Router>enable
Password:
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, 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

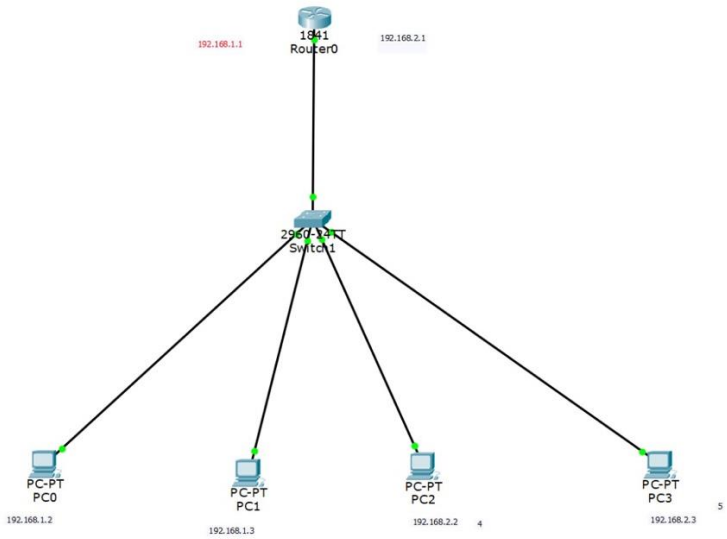
→ Go to switch and in config select database
→ set the VLAN number to 2
→ set VLAN name bmsit and click ADD.
→ Go to fastethernet 0/1 and set access or trunk.
→ Go to fastethernet 0/1 and sets access as trunk.
→ Go to config tab of router select VLAN Database
enter the no and name of VLAN created.
→ Go to CLI of router.

Router (VLAN) # exit
Router (config) # interface fast ethernet 0/0.1
Router (config-subif) # encapsulation dot1 1 q 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.

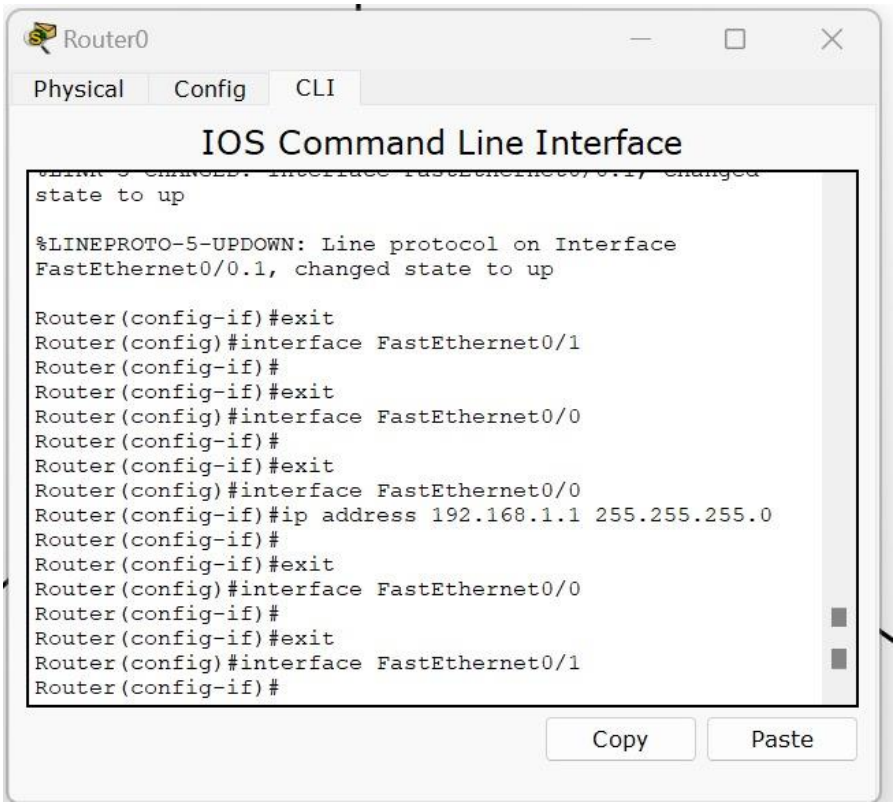
OBSERVATION:

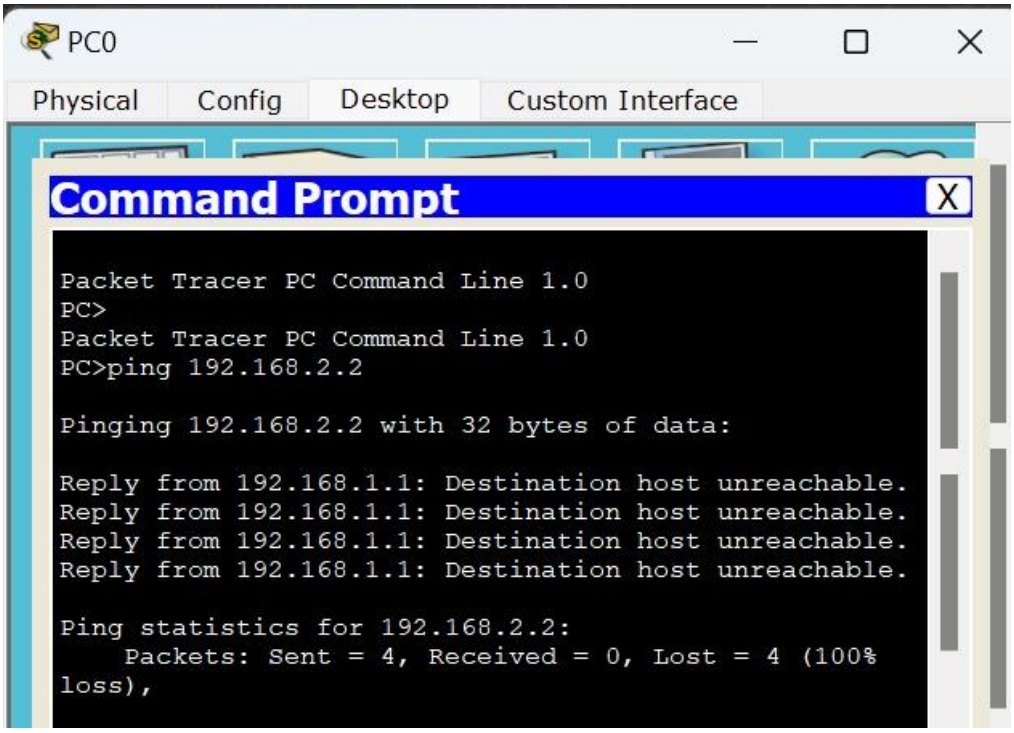
→ VLAN trunking allows switch to forward for diff VLANs over single link called trunk.
→ This is done by adding index info called tag to ethernet frame. This process is called tagging.
→ The switch understands VLAN.
→ The router understands VLAN.

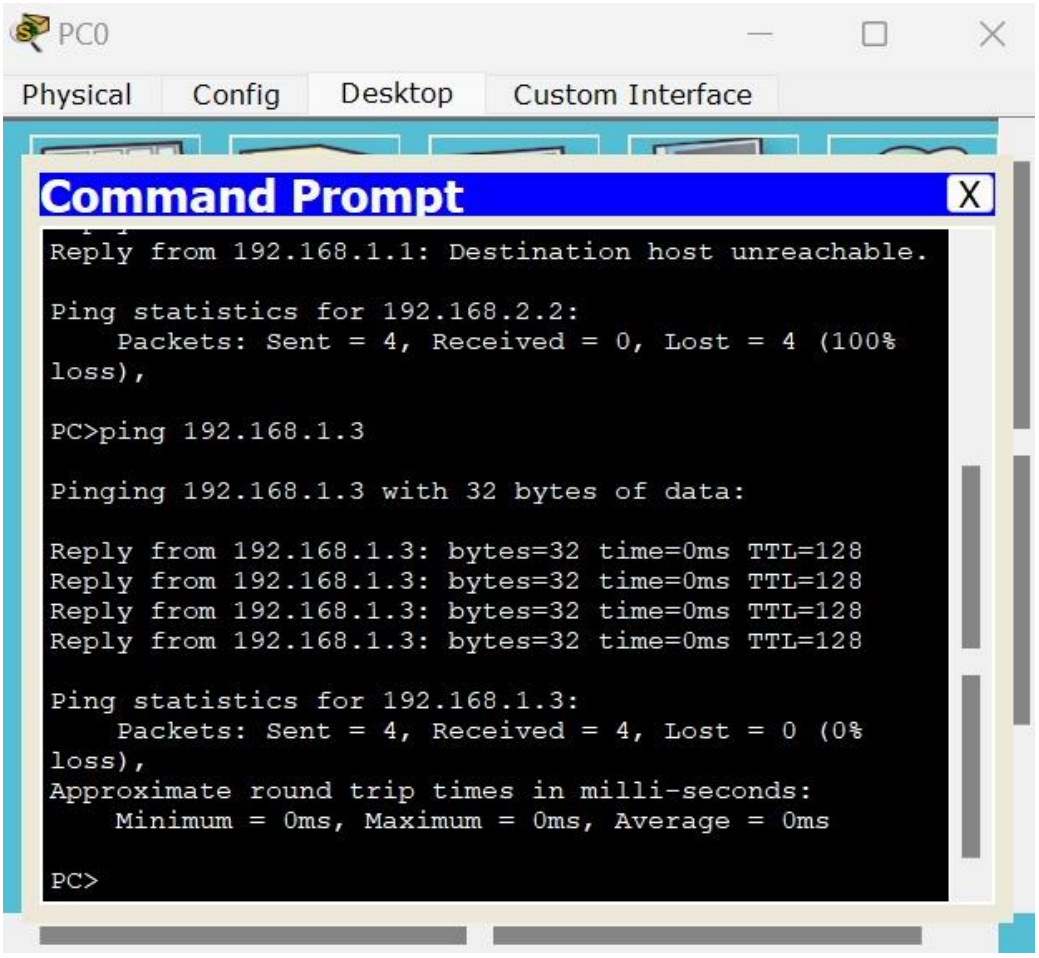
Screenshot of Topology:



Screenshot of Output:



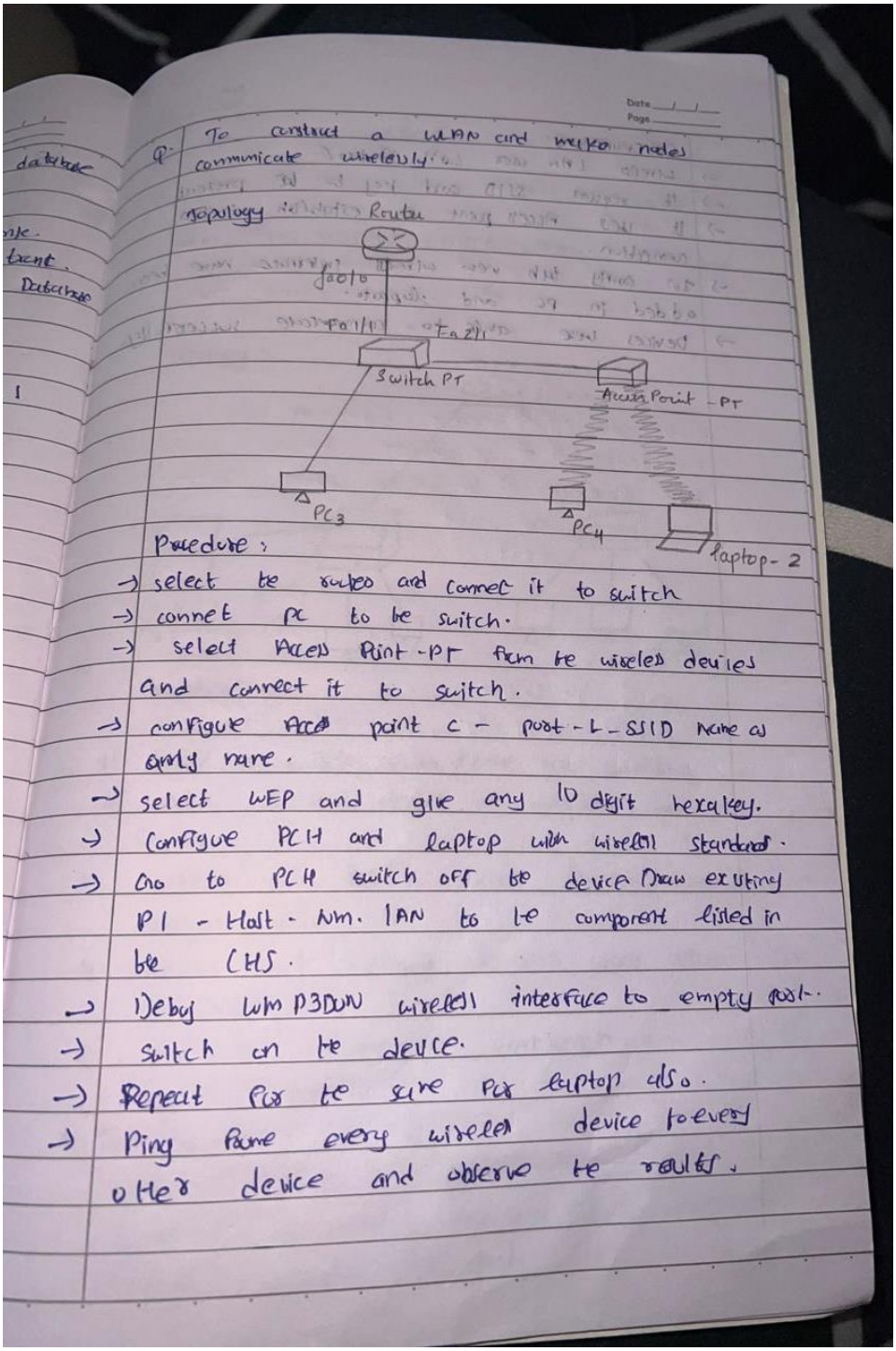

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
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),

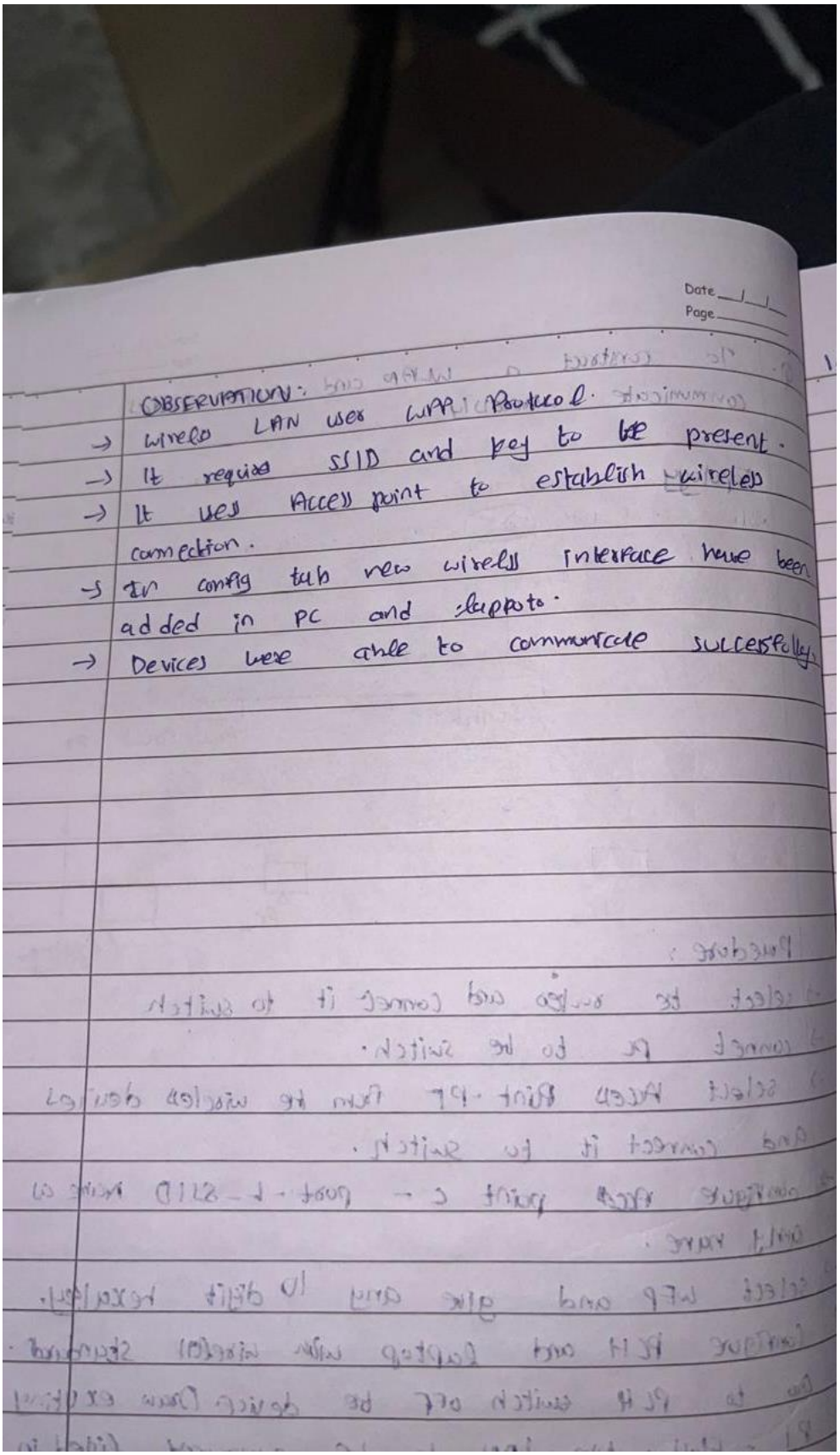

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Reply from 192.168.1.1: Destination host unreachable.
Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100%
loss),
PC>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
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:



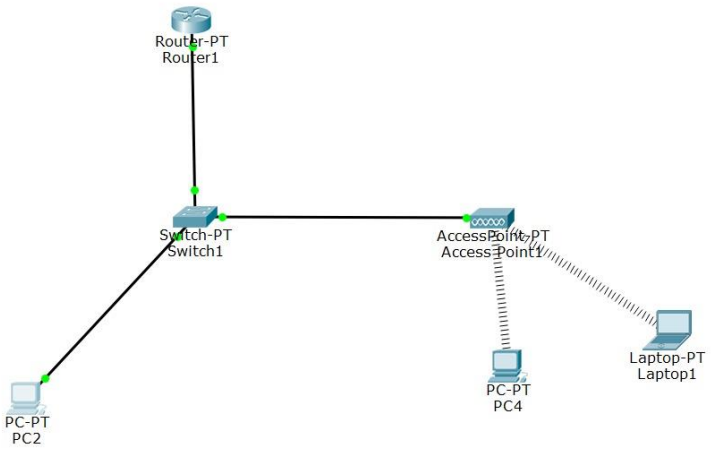


Date / /
Page

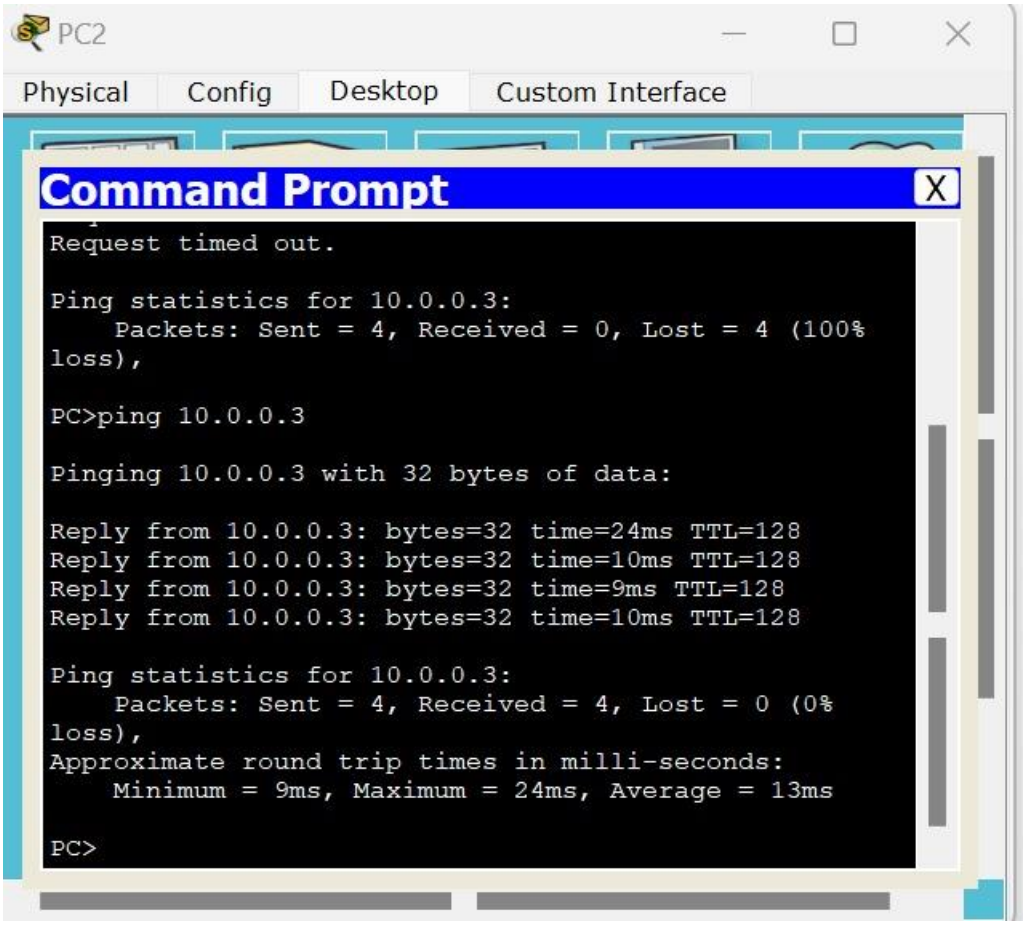
OBSERVATION:

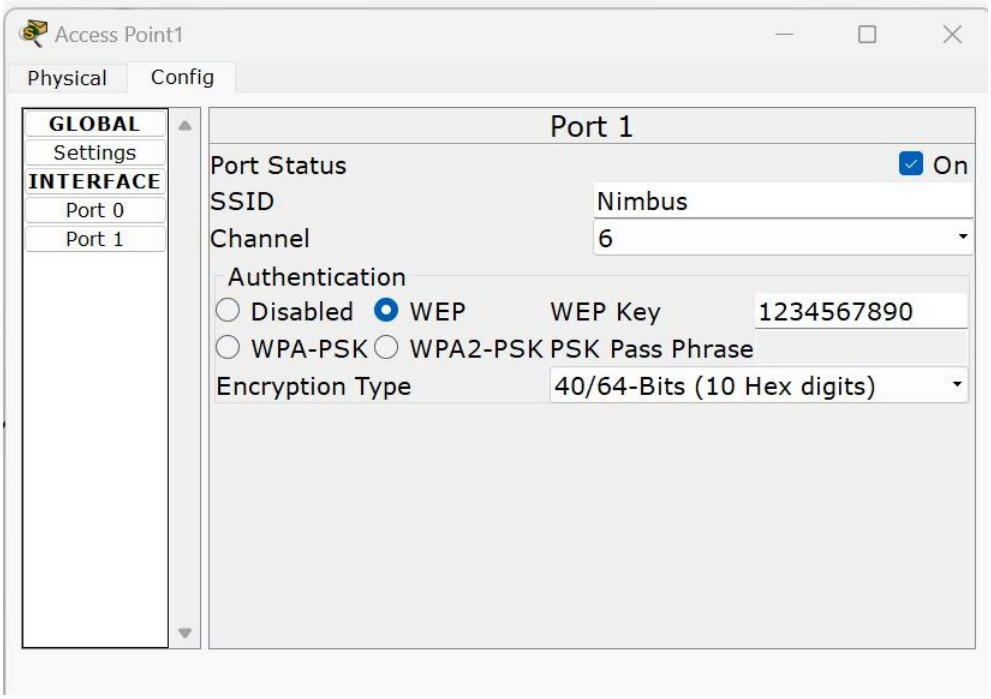
→ Wireless LAN uses WPA protocol.
→ It requires SSID and key to be present.
→ It uses Access point to establish wireless connection.
→ In config tab new wireless interface have been added in PC and laptop.
→ Devices were able to communicate successfully.

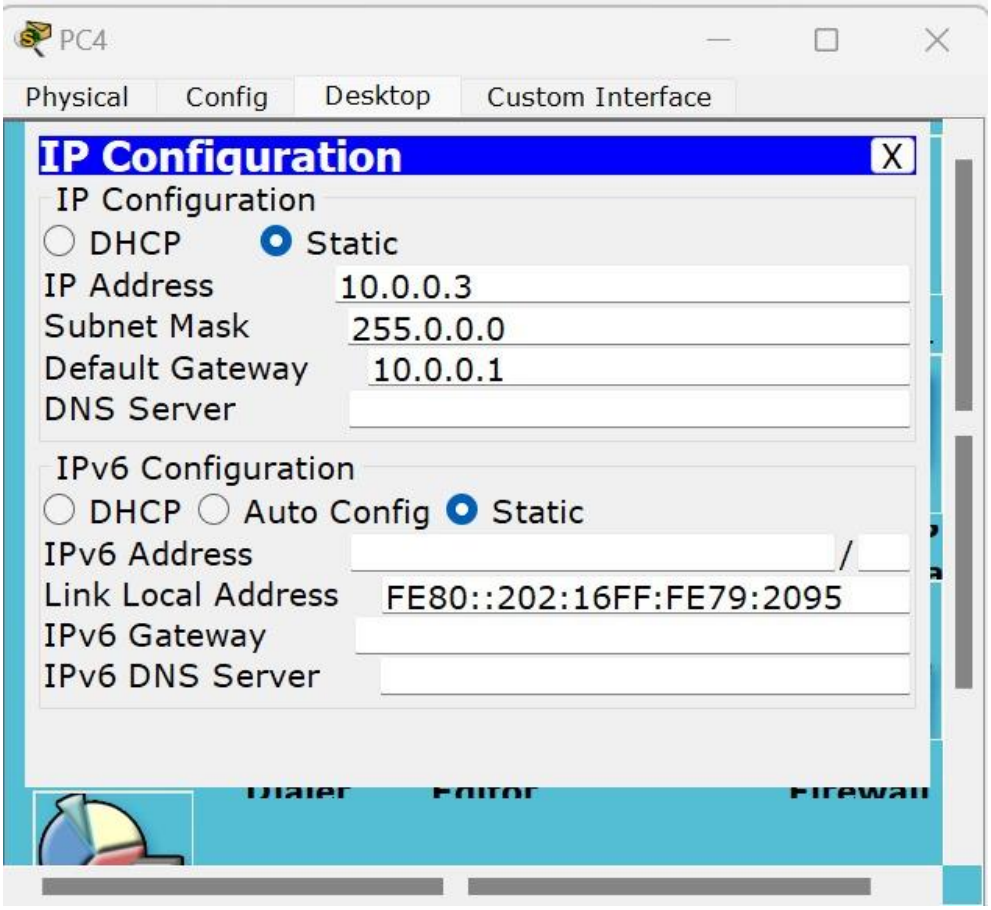
Screenshot of Topology:

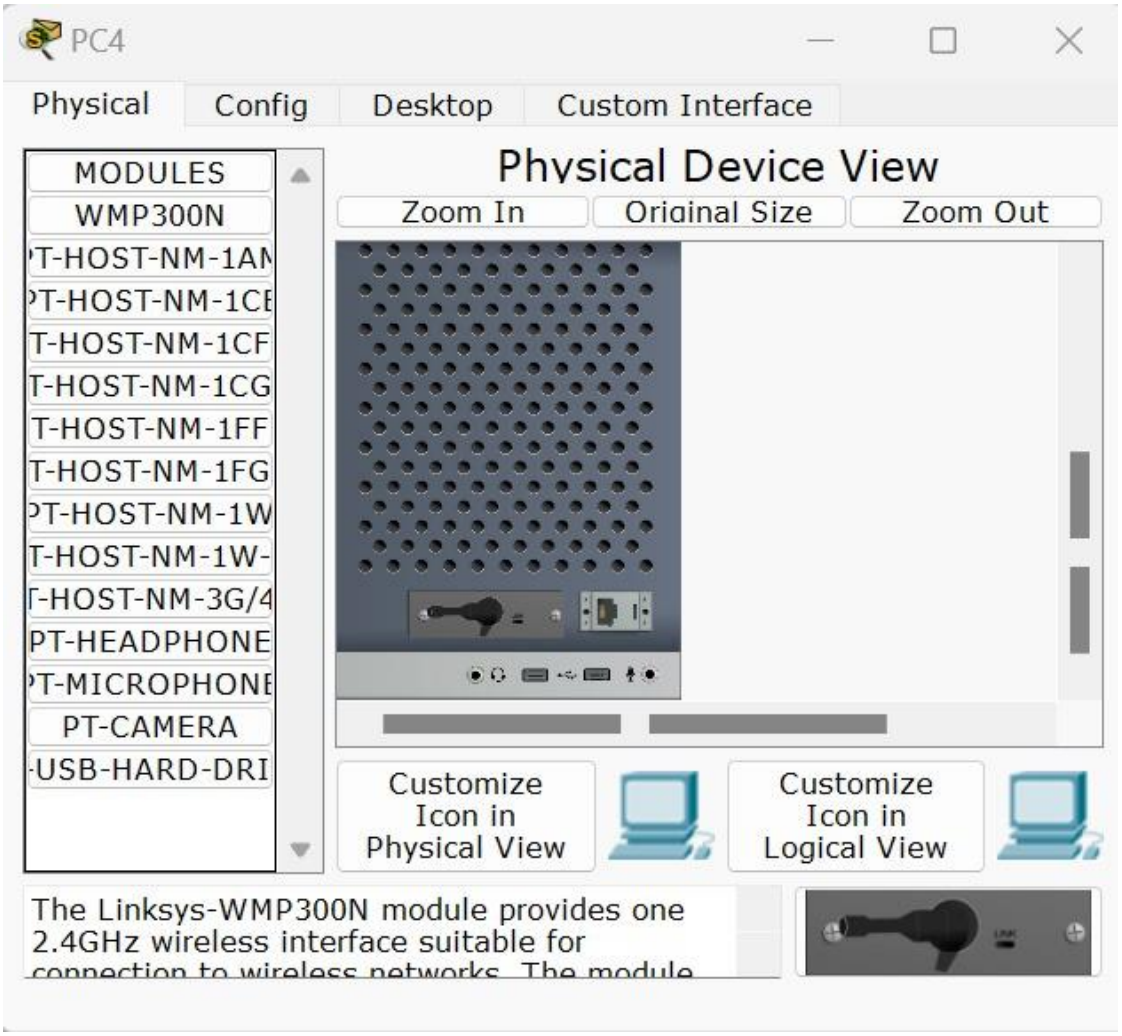


Screenshot of Output:




Access Point1
Physical
Config
GLOBAL
Settings
INTERFACE
Port 0
Port 1
Port 1
Port Status
On
SSID
Nimbus
Channel
6
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Encryption Type
40/64-Bits (10 Hex digits)


PC4
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::202:16FF:FE79:2095
IPv6 Gateway
IPv6 DNS Server

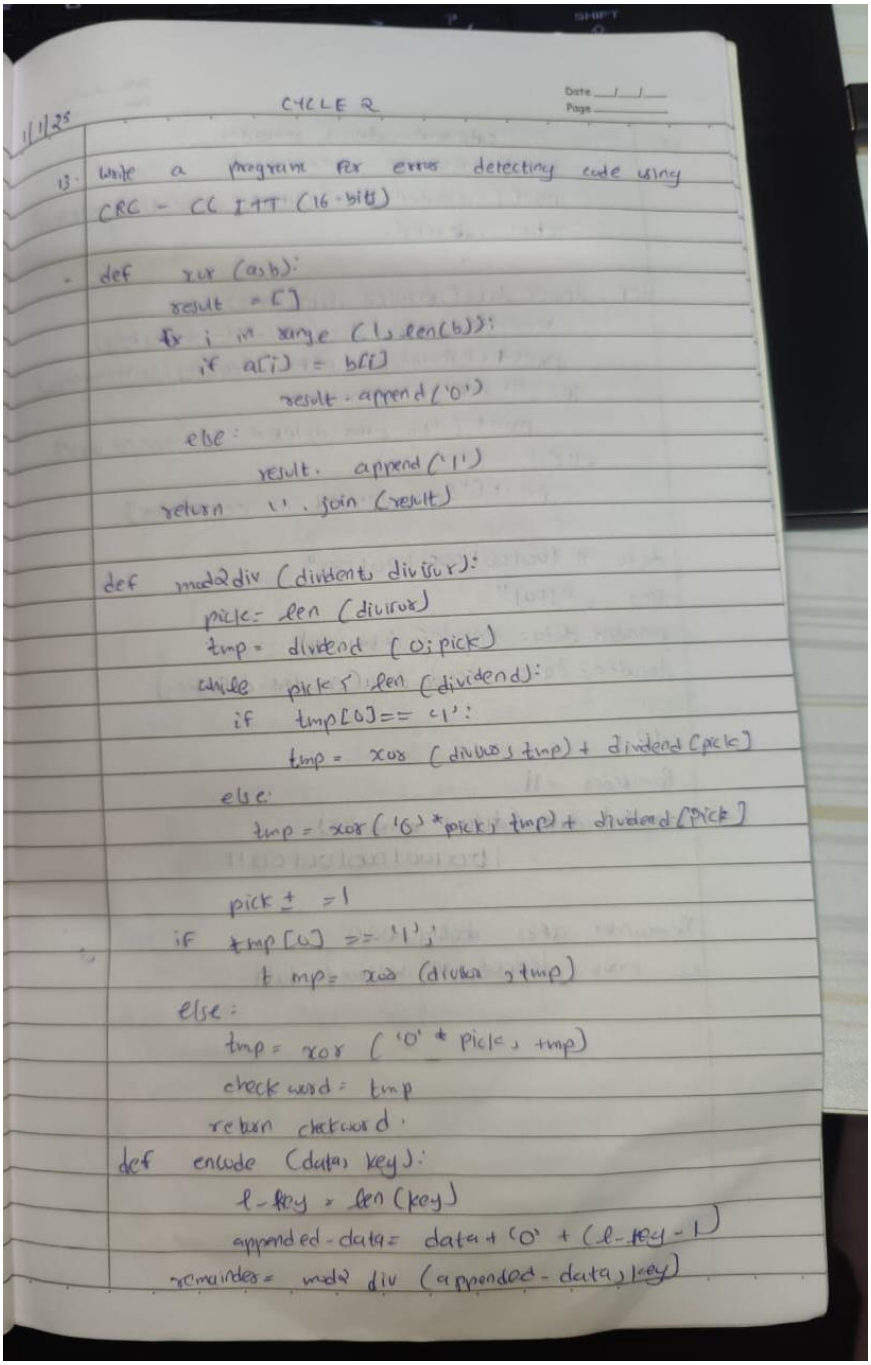

PC4
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
T-HOST-NM-1AN
T-HOST-NM-1CE
T-HOST-NM-1CF
T-HOST-NM-1CG
T-HOST-NM-1FF
T-HOST-NM-1FG
T-HOST-NM-1W
T-HOST-NM-1W-
T-HOST-NM-3G/4
PT-HEADPHONE
T-MICROPHONE
PT-CAMERA
USB-HARD-DRI
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module

CYCLE-2

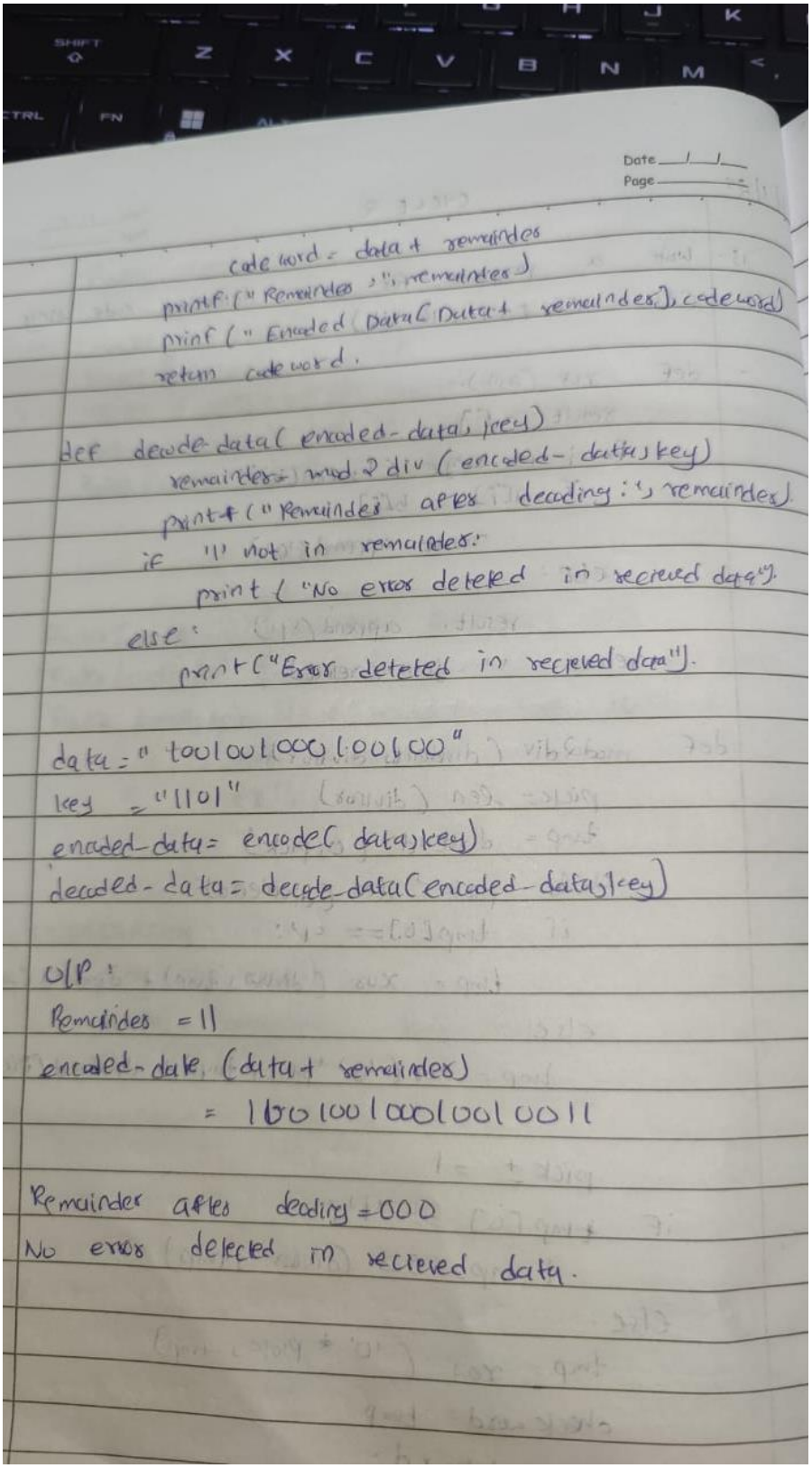
Program 13

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

Code and Output:



```
1/1/25                    CYCLE 2                    Date __/__/__
                                                     Page _____

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 '1'.join(result)

def mod2div(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)
```

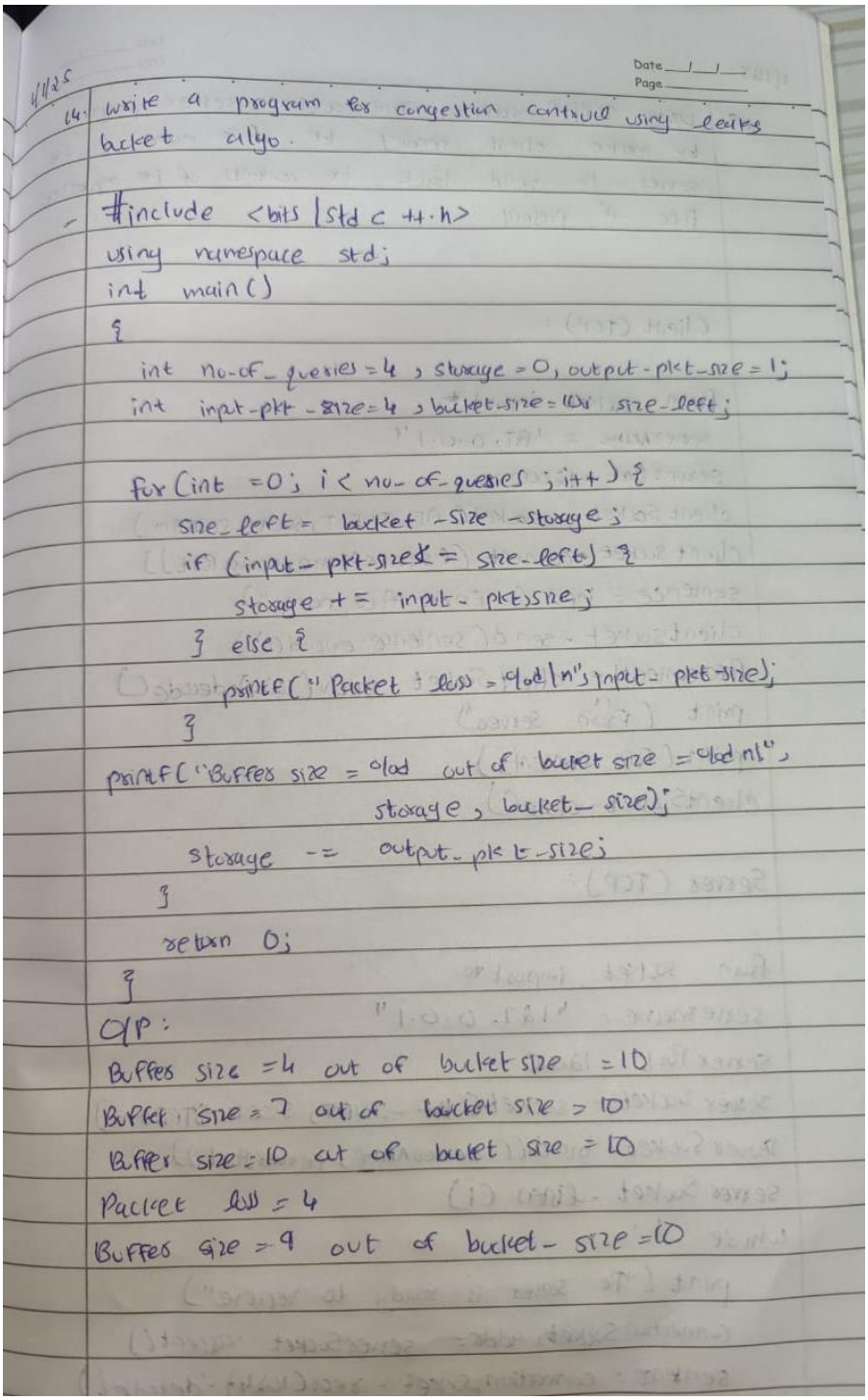


```
        codeword = data + remainder
        print("Remainder : ", remainder)
        print("Encoded Data(Data+ remainder): ", codeword)
        return codeword.

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 recieved data").
    else:
        print("Error detected in recieved data").

data = "10010001000100"
key = "1101"
encoded_data = encode(data, key)
decoded_data = decode_data(encoded_data, key)

O/P :
Remainder = 11
encoded_data (data + remainder)
        = 10010001000100111

Remainder after decoding = 000
No error detected in recieved data.
```

Program 14

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

Code and Output:

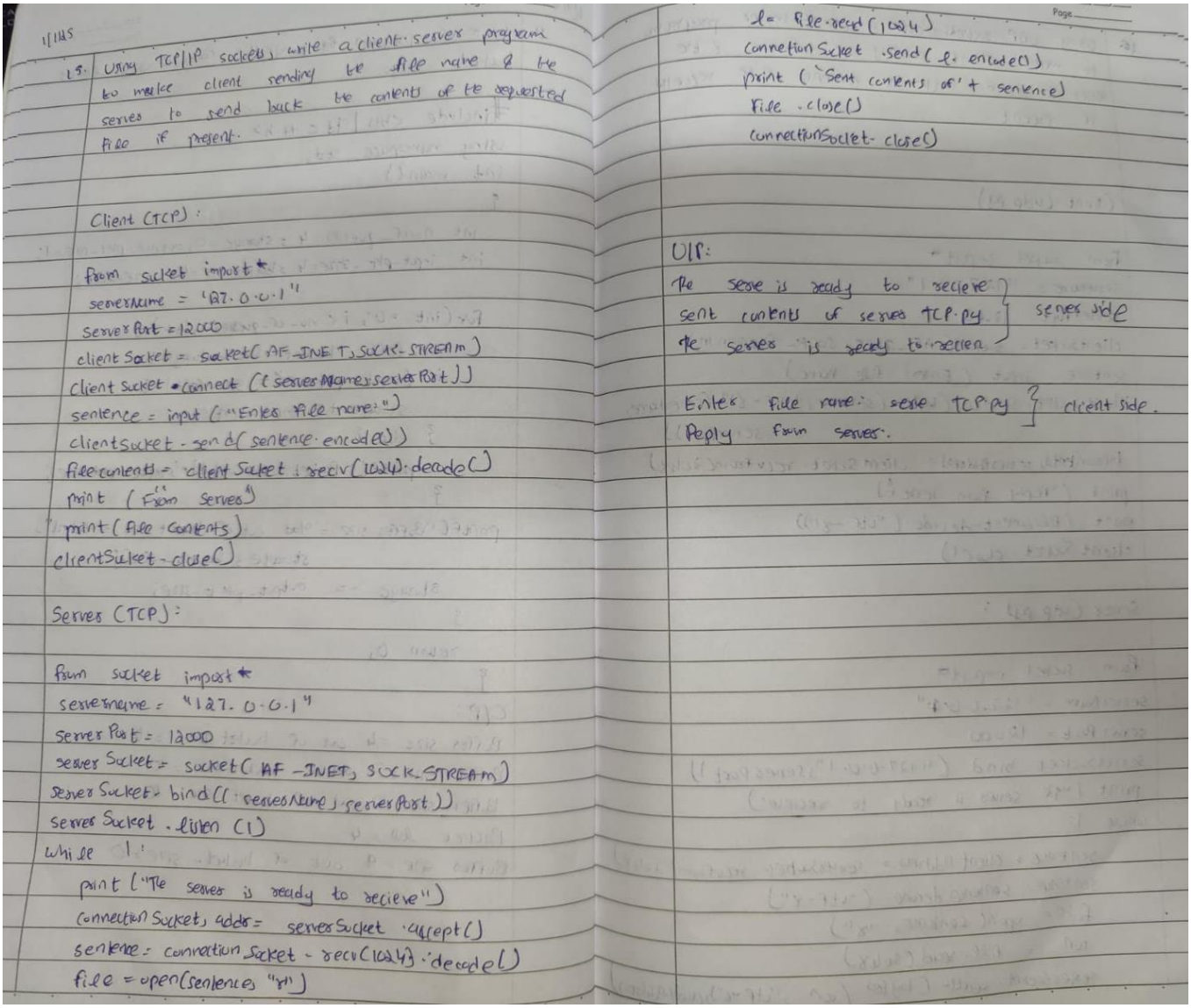


```
4/4/25
14. write a program for congestion control using leaky
bucket algo.

#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 {
      printf(" Packet loss = %d\n", input_pkt_size);
    }
  printf("Buffer size = %d out of bucket size = %d\n",
                storage, bucket_size);
    storage -= output_pkt_size;
  }
  return 0;
}
O/P:
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket_size = 10
```

Program 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.

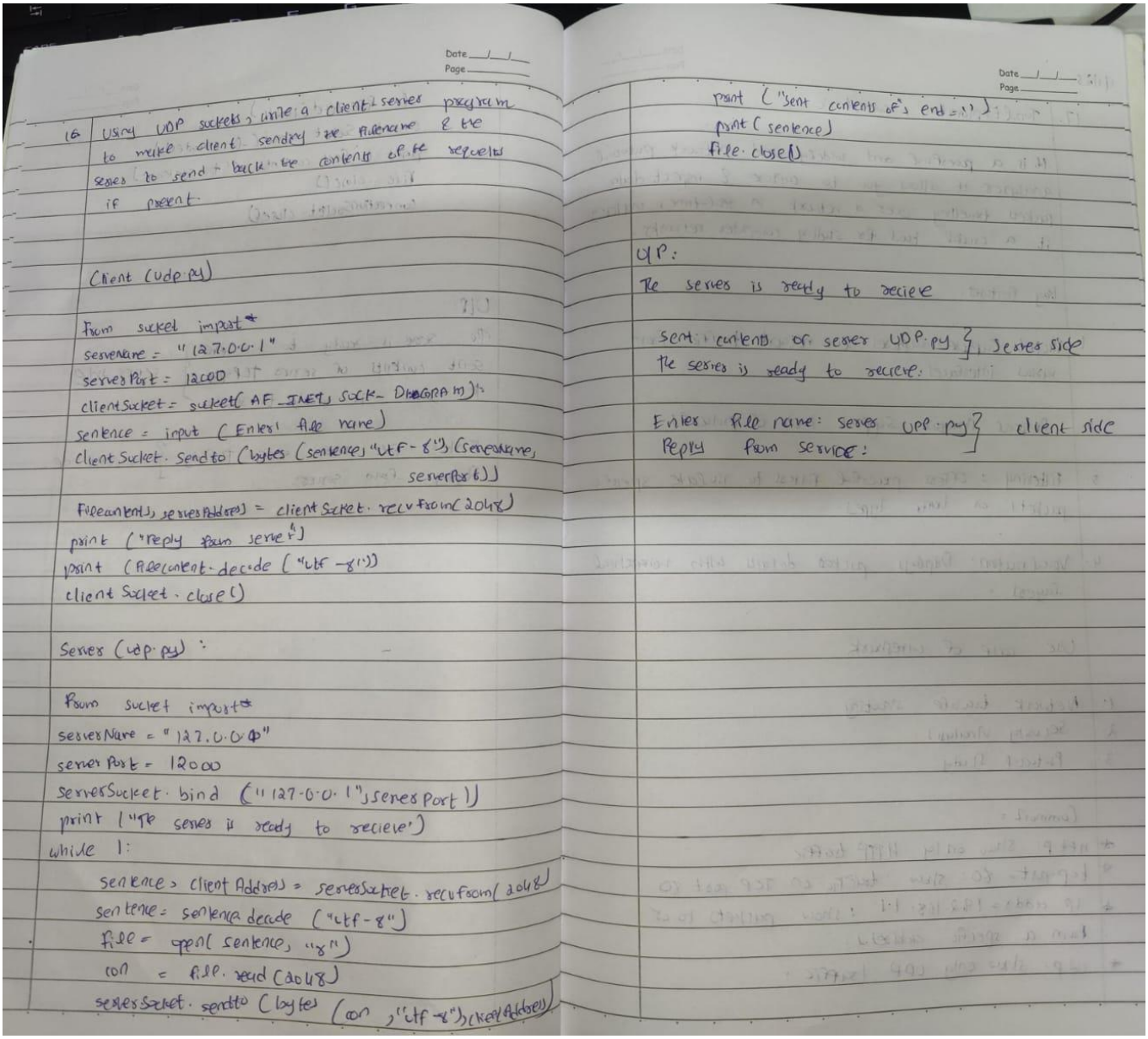
Code and Output:



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:



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

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(bytes(sentence,"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 = "127.0.0.0"
serverPort = 12000
serverSocket.bind(("127.0.0.1",serverPort))
print ("The server is ready to recieve")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence,"r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ("sent contents of", end=" ")
print (sentence)
file.close()
```

UP:

The server is ready to recieve

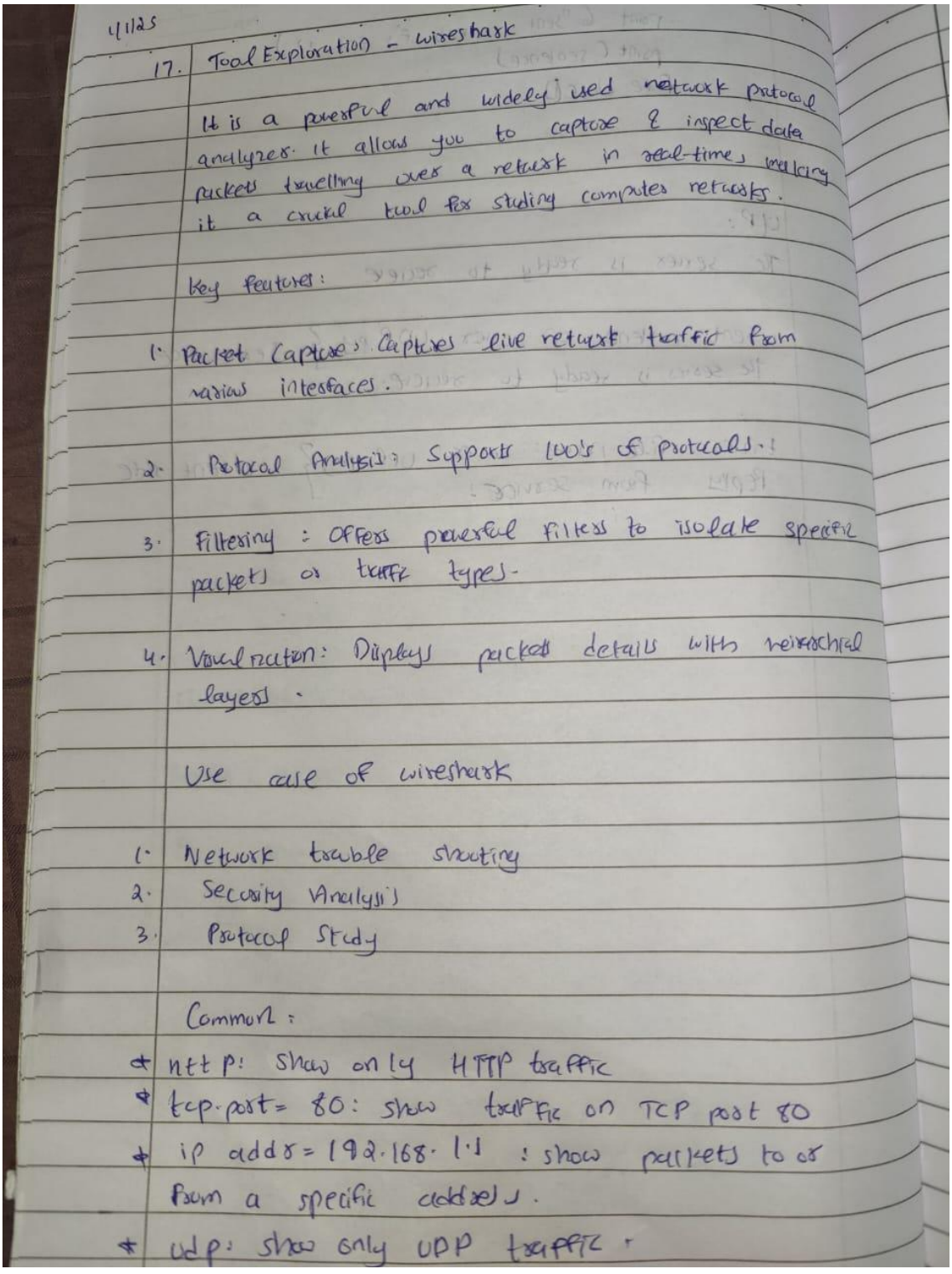
Sent contents of server UDP.py } Server side
The server is ready to recieve.

Enter file name: server UDP.py } client side
Reply from service:

Program 17

Q17: Tool Exploration-Wireshark

Code and Output:



4/1/25

17. Tool Exploration - wireshark

It is a powerful and widely used network protocol analyzer. It allow you to capture & inspect data packets travelling over a network in real-time, making it a crucial tool for studing computer networks.

Key features:

1. Packet Capture: Captures live network traffic from various interfaces.
2. Protocol Analysis: Supports 1000'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.