VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

Prajwal Dhage (1BM21CS133)

in partial fulfillment 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
June-2023 to September-2023

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 NETWORKS" carried out by

Prajwal Dhage (1BM21CS133), who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in ComputerScience and Engineering** of the Visvesvaraya Technological University, Belgaum during the academic semester June-2023 to September-2023. The Lab report has been approved as it satisfiesthe academic requirements in respect of a **COMPUTER NETWORKS (22CS4PCCON)** work prescribed for the said degree.

Swathi Sridharan
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

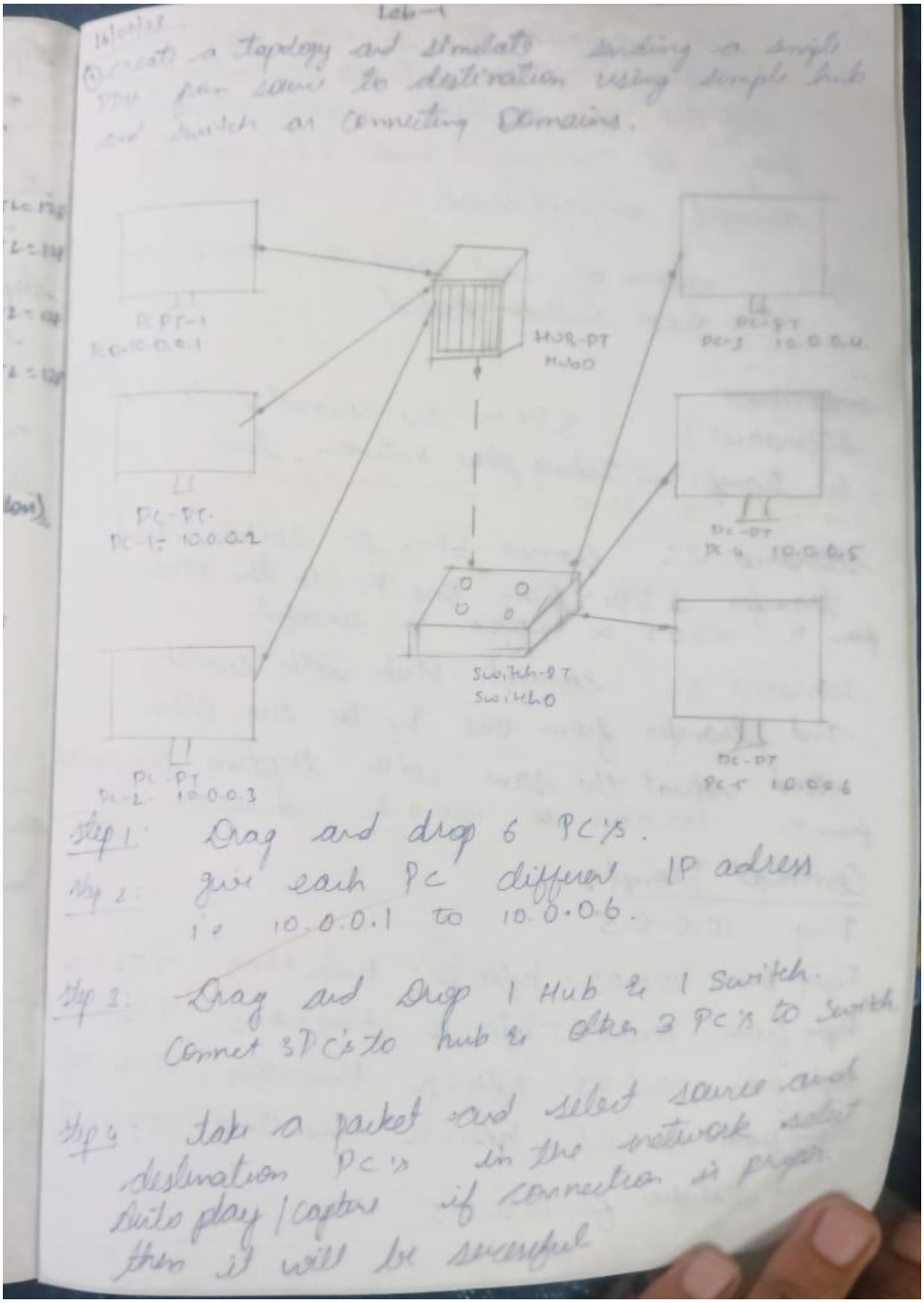
INDEX SHEET

Course Outcomes

CO1	Apply the fundamental concepts of communication in networking.
CO2	Analyze the various protocols, techniques in TCP/IP network architectur
CO3	Develop programs that demonstrate the functionalities of physical, Data Link, Network, Transport or Application layer

Experiment 1

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



Step 5: take a packet then select the source and destination in the network connected by only Hub in simulation mode then select play. if connection is proper event will be successful in real mode

Step 6: switch off and repeat the same and the event will be failed

~~Practical~~

Scenario 1: 3 PC's are connected to Hub. & transfer is taking place between two PC's from 10.0.0.1 to 10.0.0.2 → sucessful

Scenario 2: connect 3 PC's to switch and transfer a PDU from one PC to the other from PC 10.0.0.5 to 10.0.0.6 → sucessful

Scenario 3: connect Hub with switch and transfer from one PC to the other and repeat the same with toggling the switch from PC 10.0.0.1 to 10.0.0.6 → sucessful

Command Prompt:

Ping 10.0.0.5

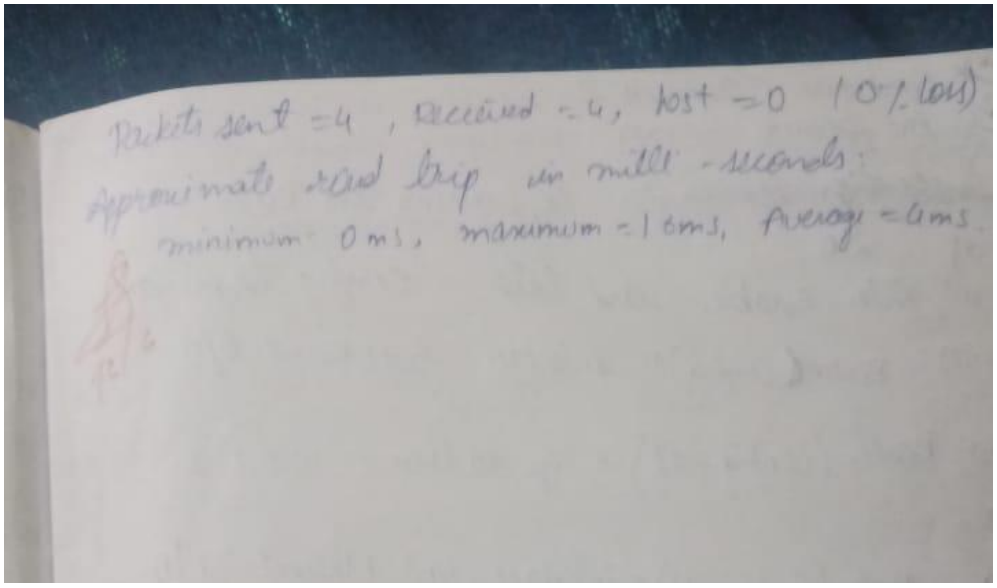
Reply from 10.0.0.5: bytes=32 time=16ms TTL=128

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

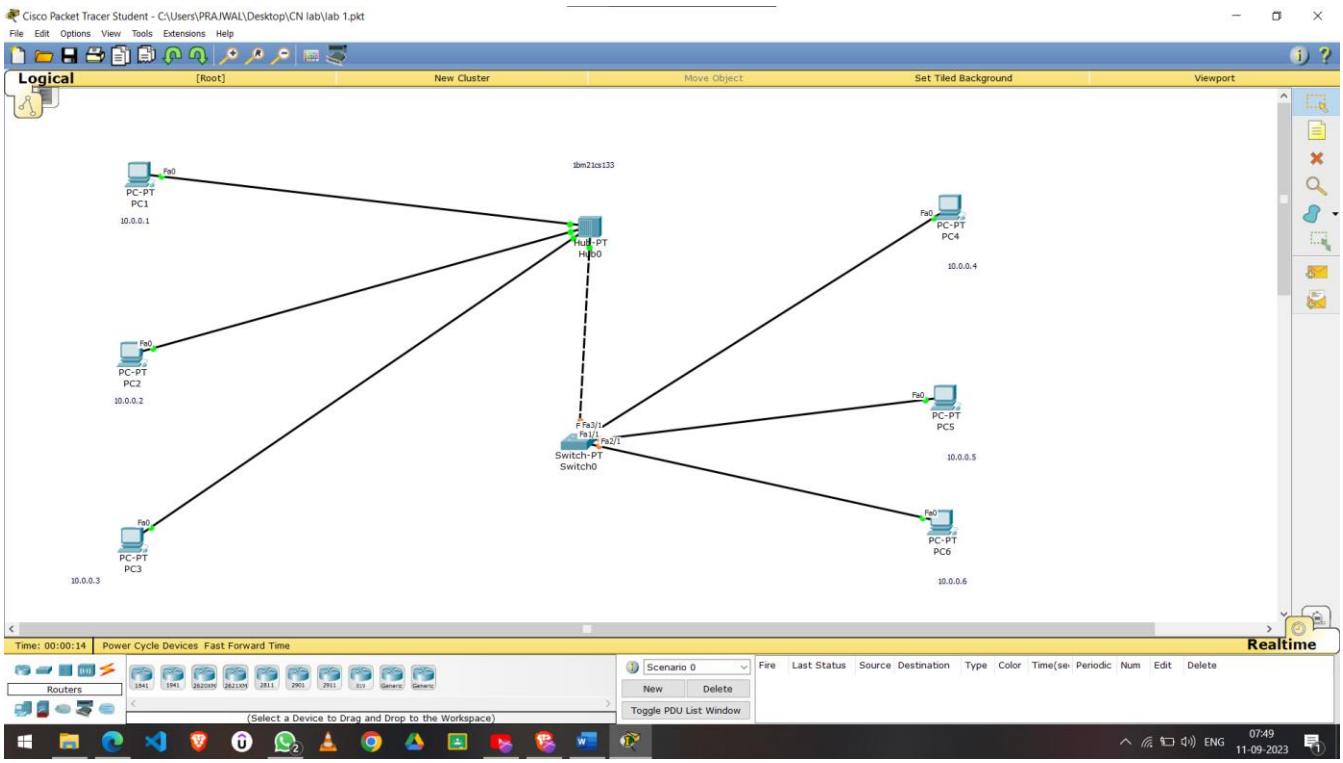
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128

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

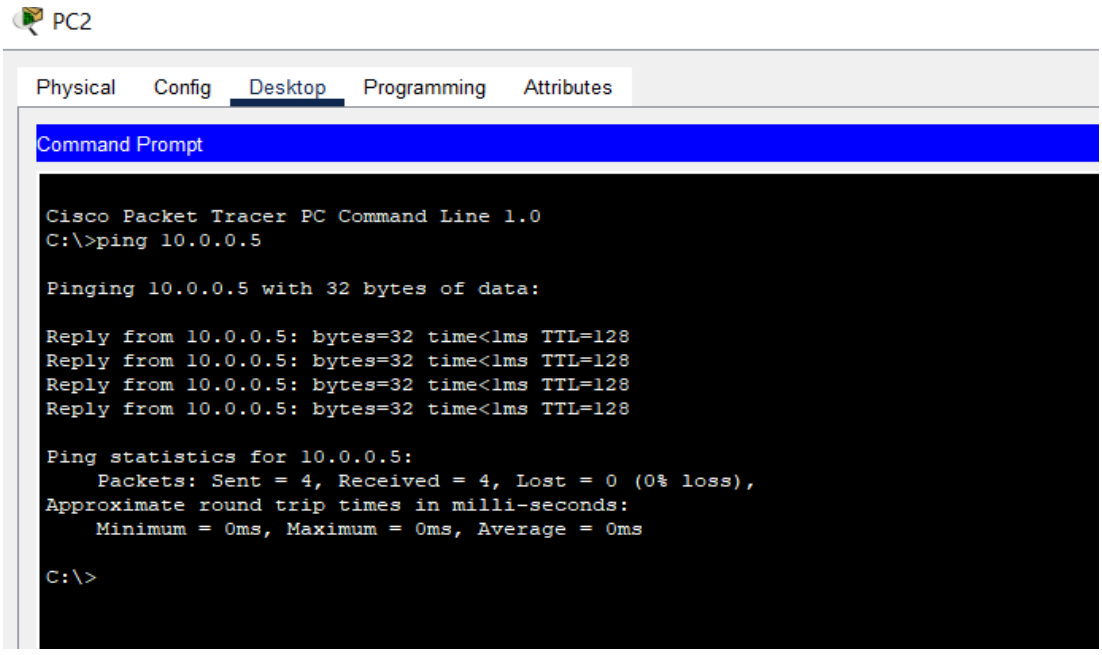
Ping statistics for 10.0.0.5


Packets sent = 4, received = 4, lost = 0 (0% loss)
Approximate round trip in milli-seconds:
minimum = 0ms, maximum = 1ms, Average = 0ms

Topology:

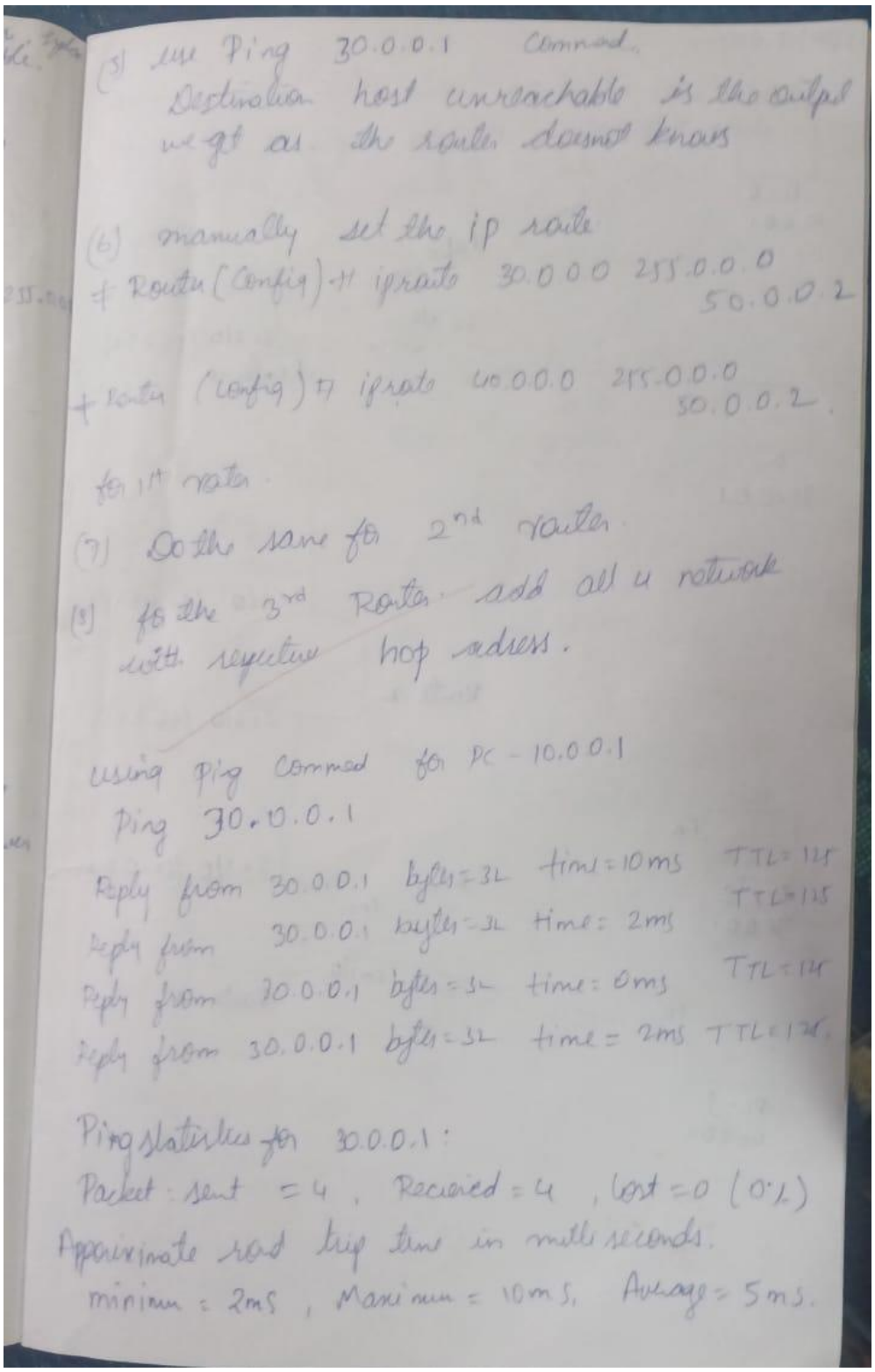


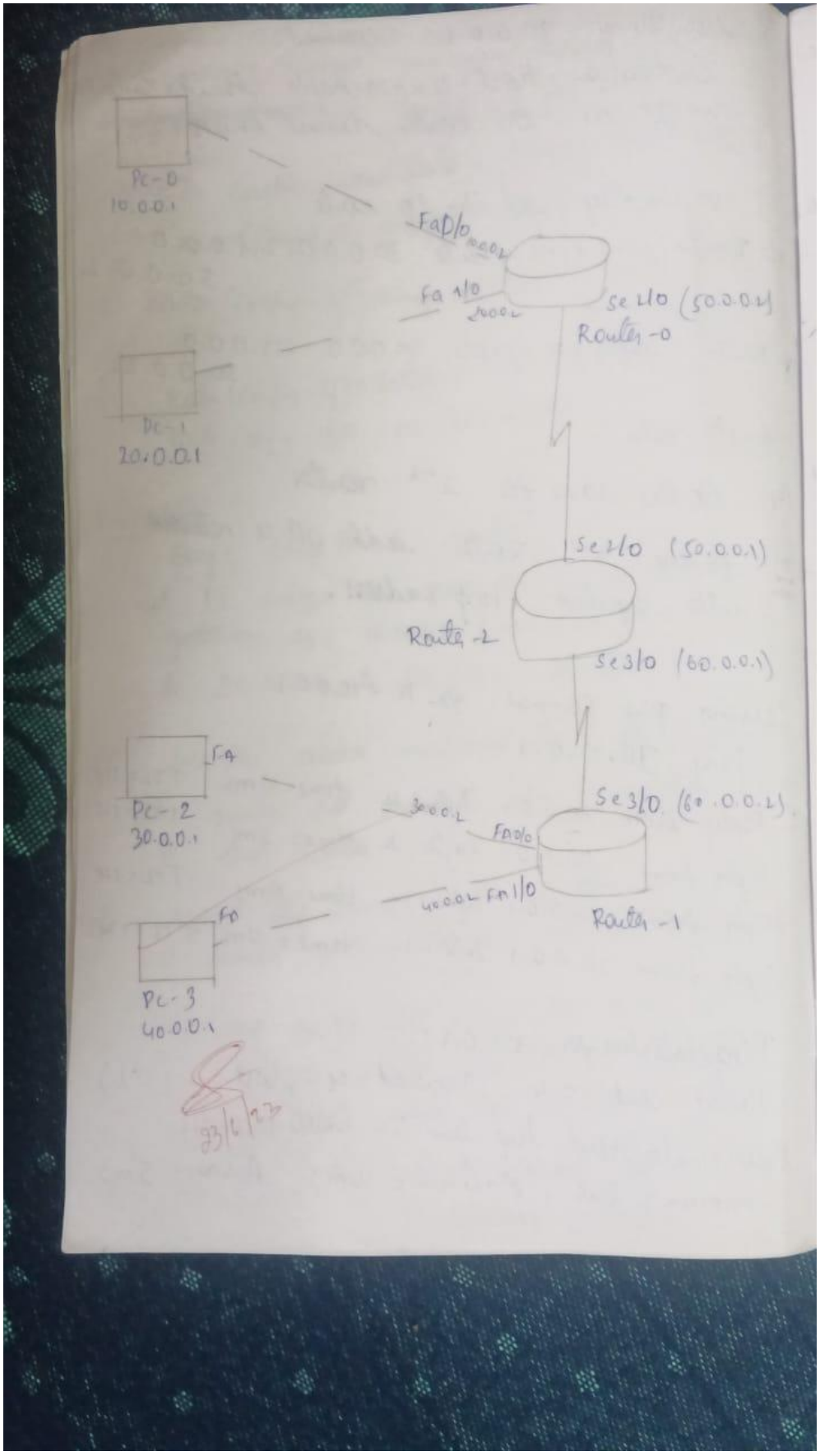
Output:



Experiment 2

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




Pc-0
10.0.0.1
Fa0/0
Fa 1/0
Se 2/0 (50.0.0.2)
Router-0
Pc-1
20.0.0.1
Se2/0 (50.0.0.1)
Router-2
Se3/0 (60.0.0.1)
Fa
Pc-2
30.0.0.1
Se3/0 (60.0.0.2)
Fa0/0
Fa1/0
Router-1
Fa
Pc-3
40.0.0.1
23/6/22

(2) Configure IP address to routers in packet Tracer. Explore the following messages: ping response, destination unreachable.

Configuring the Router

(i) No.

(ii) enter enable and later config terminal

(iii) Router(config)# interface fastEthernet 0/0

(iv) Router(config-if)# ip address 10.0.0.2 255.

(v) Router(config-if)# interface fast Ethernet 1/0

Router(config-if)# no shut down.

Procedure:

(1) Drag & Drop PC & router.

set PC adress to both PC & set their gateway as 10.0.0.1 & 10.0.0.2 for 1 PC & 20.0.0.1 & 20.0.0.2 to other PC

(2) use the above mentioned steps to configure the router and establish the connection between PC and router.

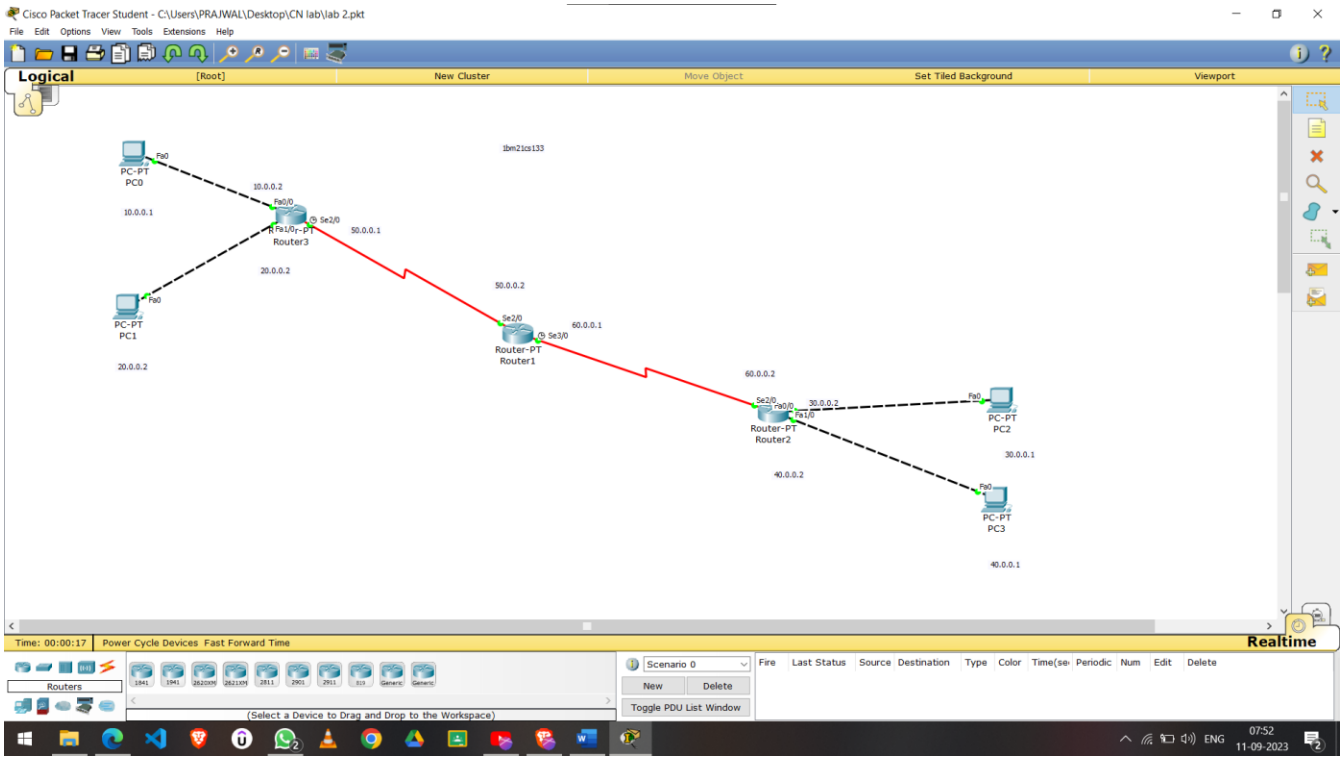
(3) repeat the same with another 2 PC's & a router with IP adress 30.0.0.1 & 40.0.0.1

(4) ~~Set up~~ Drag and drop the other router connect both router to the new router by serial ports. using the command

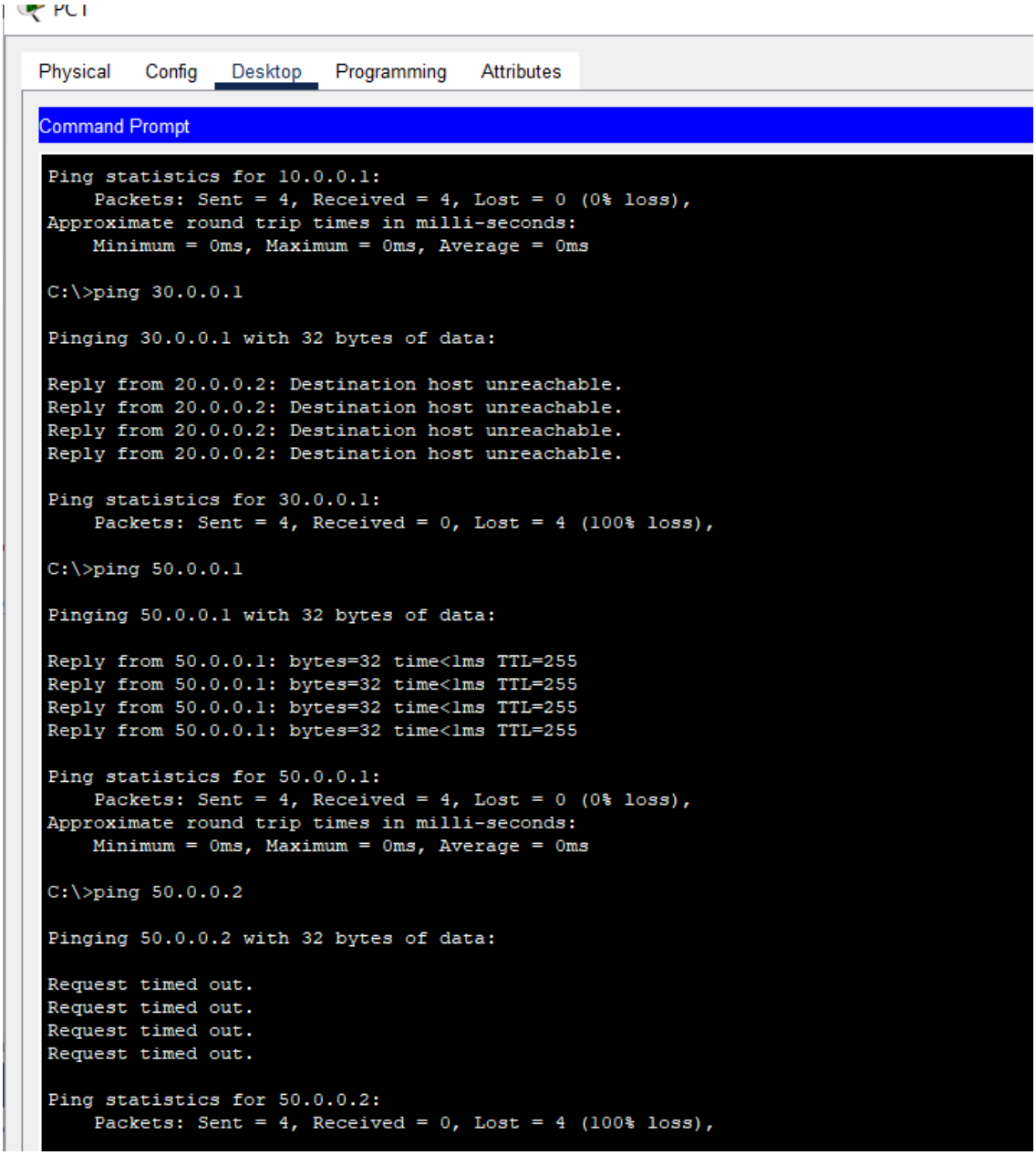
Router(config)# interface serial 2/0

Router(config-if)# ip address 50.0.0.2

Topology:

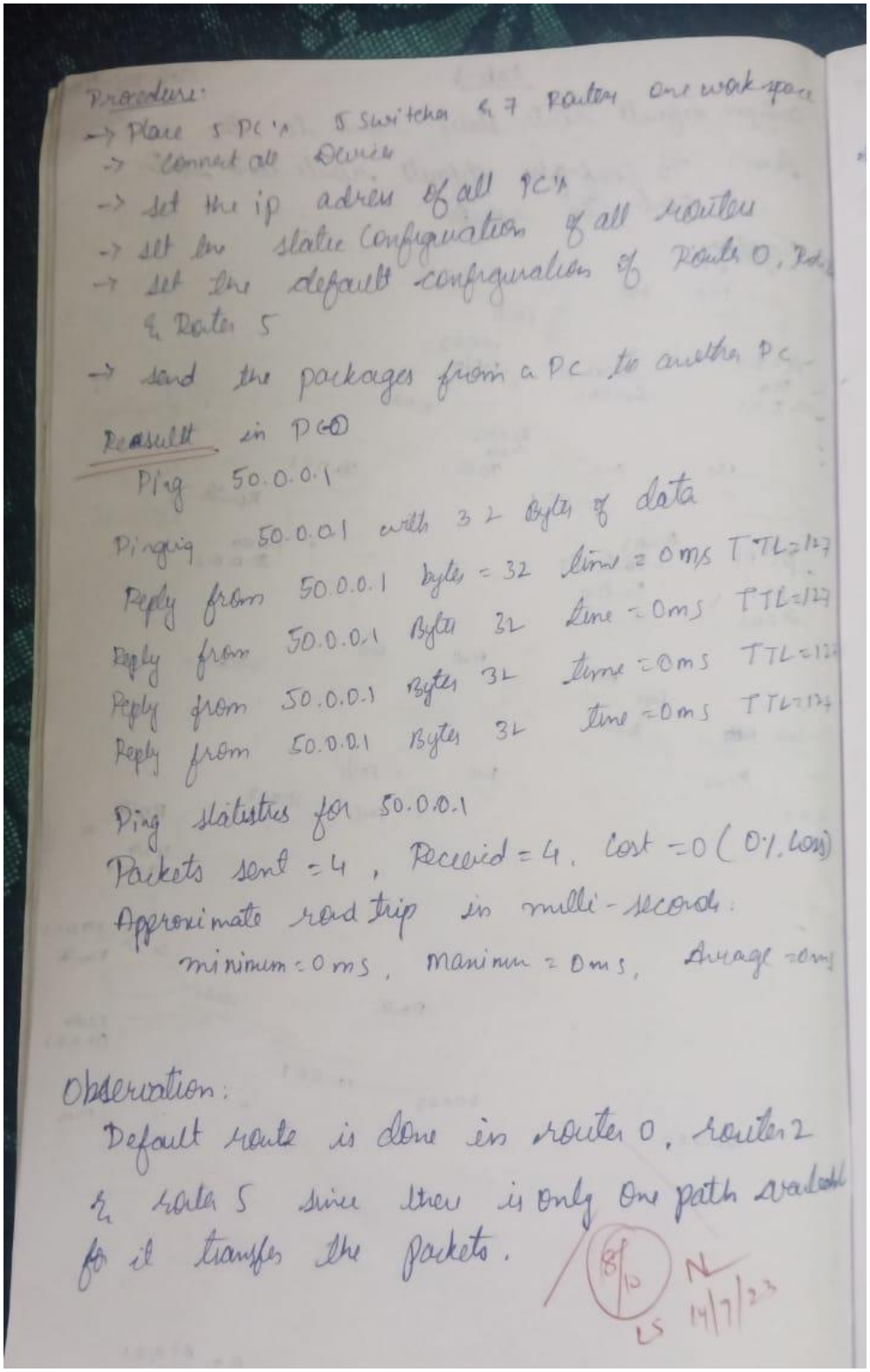

Cisco Packet Tracer Student - C:\Users\PRAJWAL\Desktop\CN lab\lab 2.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
1bm21cs133
PC-PT PC0
PC-PT PC1
Router-PT Router3
Router-PT Router1
Router-PT Router2
PC-PT PC2
PC-PT PC3
10.0.0.2
10.0.0.1
20.0.0.2
20.0.0.2
50.0.0.1
50.0.0.2
60.0.0.1
60.0.0.2
30.0.0.2
30.0.0.1
40.0.0.2
40.0.0.1
Time: 00:00:17
Power Cycle Devices Fast Forward Time
Realtime
Routers
(Select a Device to Drag and Drop to the Workspace)
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Periodic
Num
Edit
Delete
ENG
07:52
11-09-2023

Output:


PC1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 20.0.0.2: Destination host unreachable.
Reply from 20.0.0.2: Destination host unreachable.
Reply from 20.0.0.2: Destination host unreachable.
Reply from 20.0.0.2: Destination host unreachable.
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 50.0.0.1
Pinging 50.0.0.1 with 32 bytes of data:
Reply from 50.0.0.1: bytes=32 time<1ms TTL=255
Reply from 50.0.0.1: bytes=32 time<1ms TTL=255
Reply from 50.0.0.1: bytes=32 time<1ms TTL=255
Reply from 50.0.0.1: bytes=32 time<1ms TTL=255
Ping statistics for 50.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 50.0.0.2
Pinging 50.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 50.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

Experiment 3

Configure default route, static route to the Router



Procedure:
-> Place 5 PC's 5 Switches & 7 Routers one work space
-> Connect all Devices
-> Set the ip adress of all PC's
-> Set the static configuration of all routers
-> Set the default configuration of Route 0, Rd. & Router 5
-> Send the packages from a PC to another PC

Result in PC0

Ping 50.0.0.1

Pinging 50.0.0.1 with 32 Bytes of data
Reply from 50.0.0.1 bytes = 32 time = 0ms TTL=127
Reply from 50.0.0.1 Bytes 32 time = 0ms TTL=127
Reply from 50.0.0.1 Bytes 32 time = 0ms TTL=127
Reply from 50.0.0.1 Bytes 32 time = 0ms TTL=127

Ping statistics for 50.0.0.1
Packets sent = 4, Received = 4, Lost = 0 (0% Loss)
Approximate round trip in milli-seconds:
minimum = 0ms, Maximum = 0ms, Average = 0ms

Observation:
Default route is done in router 0, router 2 & router 5 since there is only one path available for it transfer the packets.

8/10
14/7/23

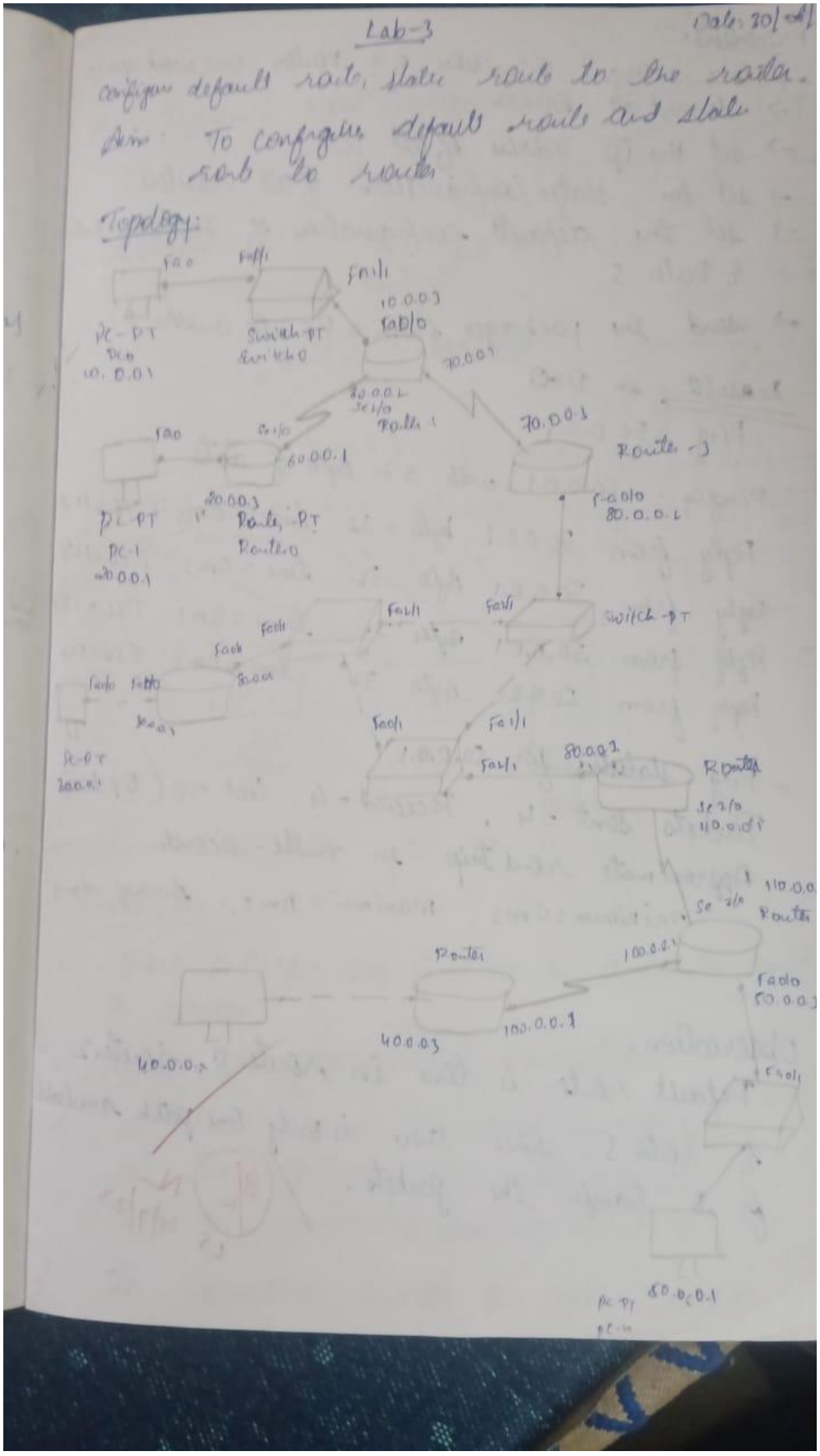
Lab-3

Date: 20/

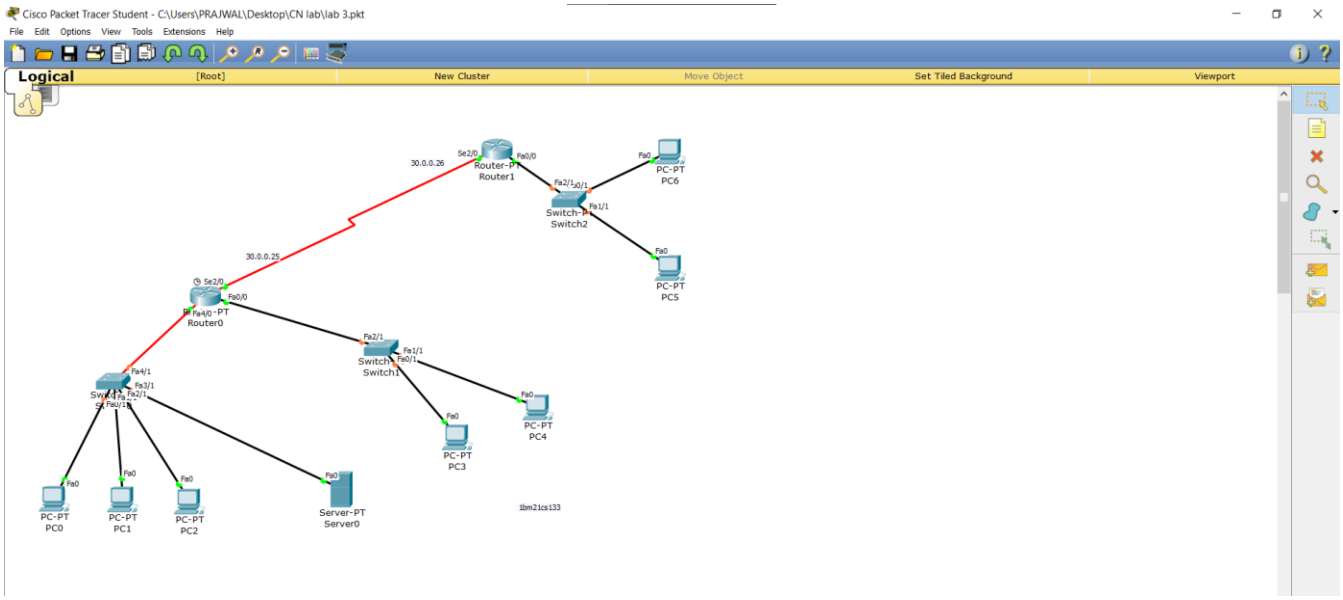
configure default route, static route to the router.

Aim: To configure default route and static route to router

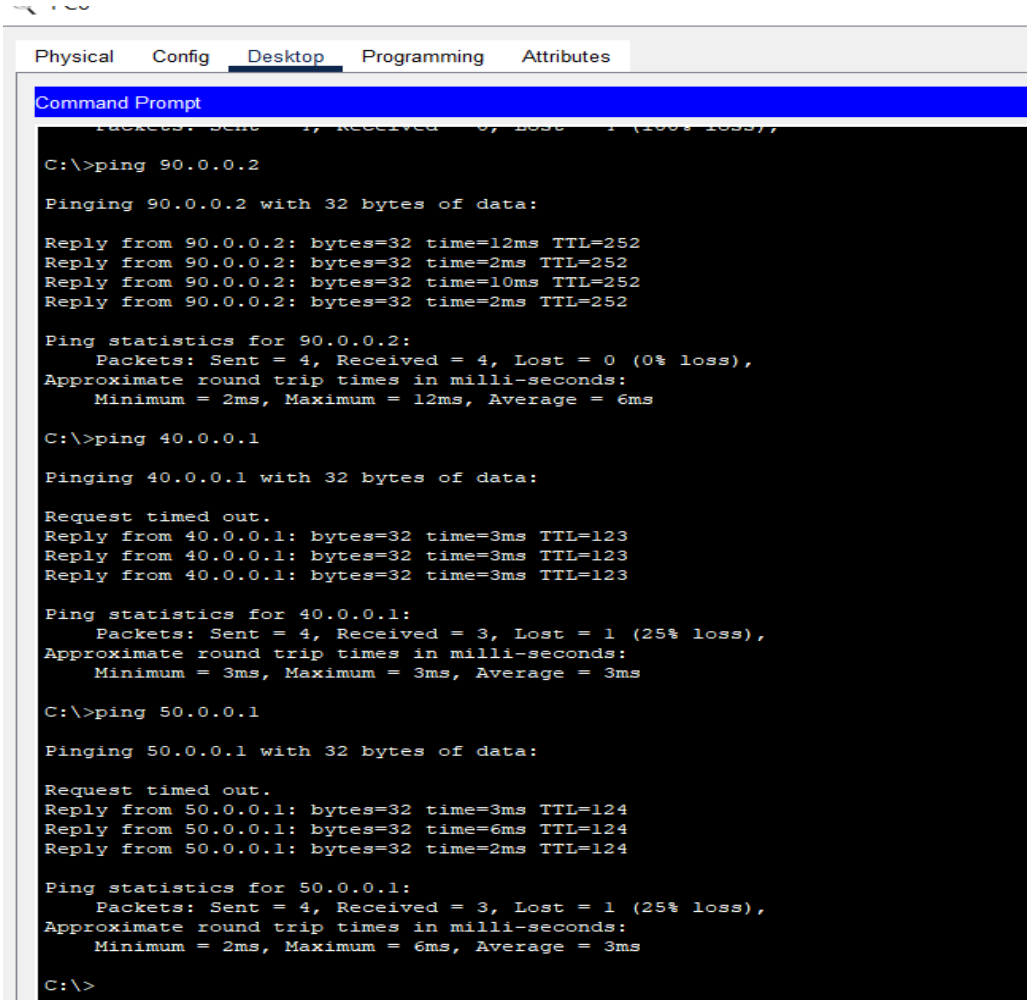
Topology:



Topology:

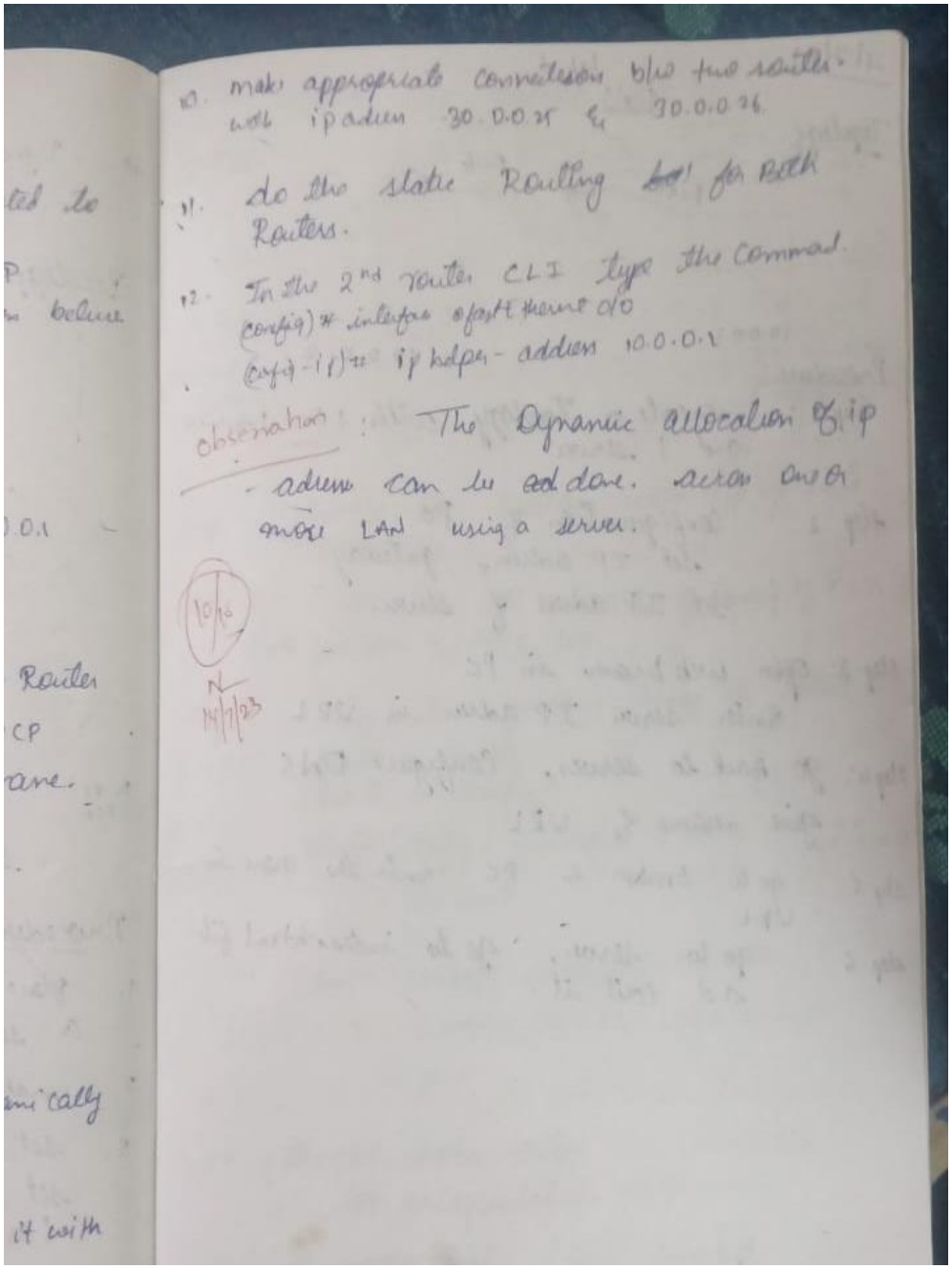


Output:



Experiment 4

Configure DHCP within a LAN and outside LAN

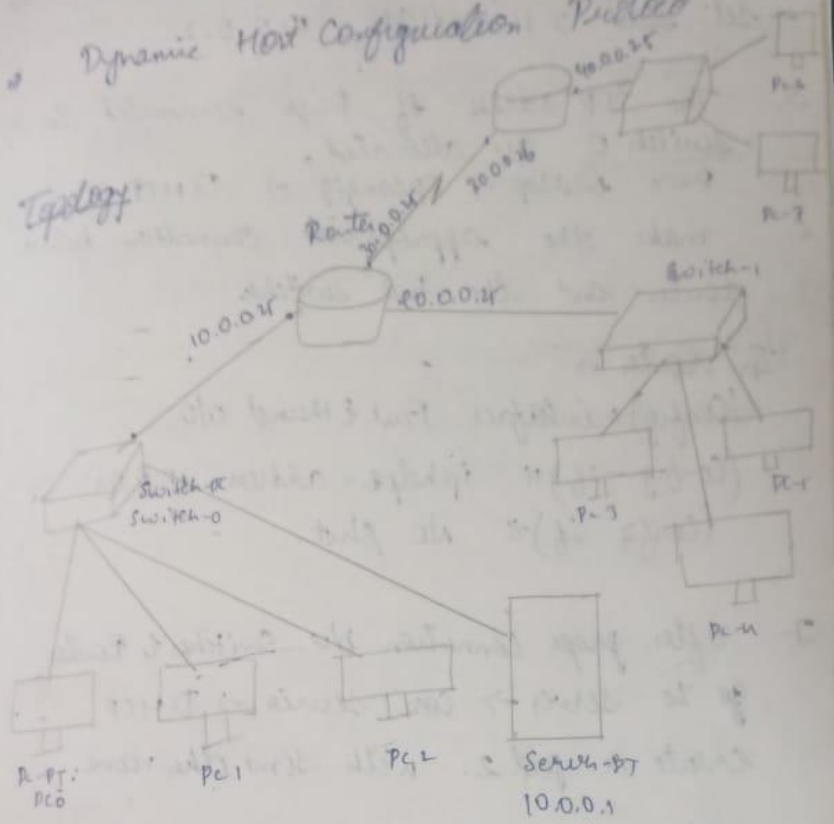


14/9/23

Lab-9

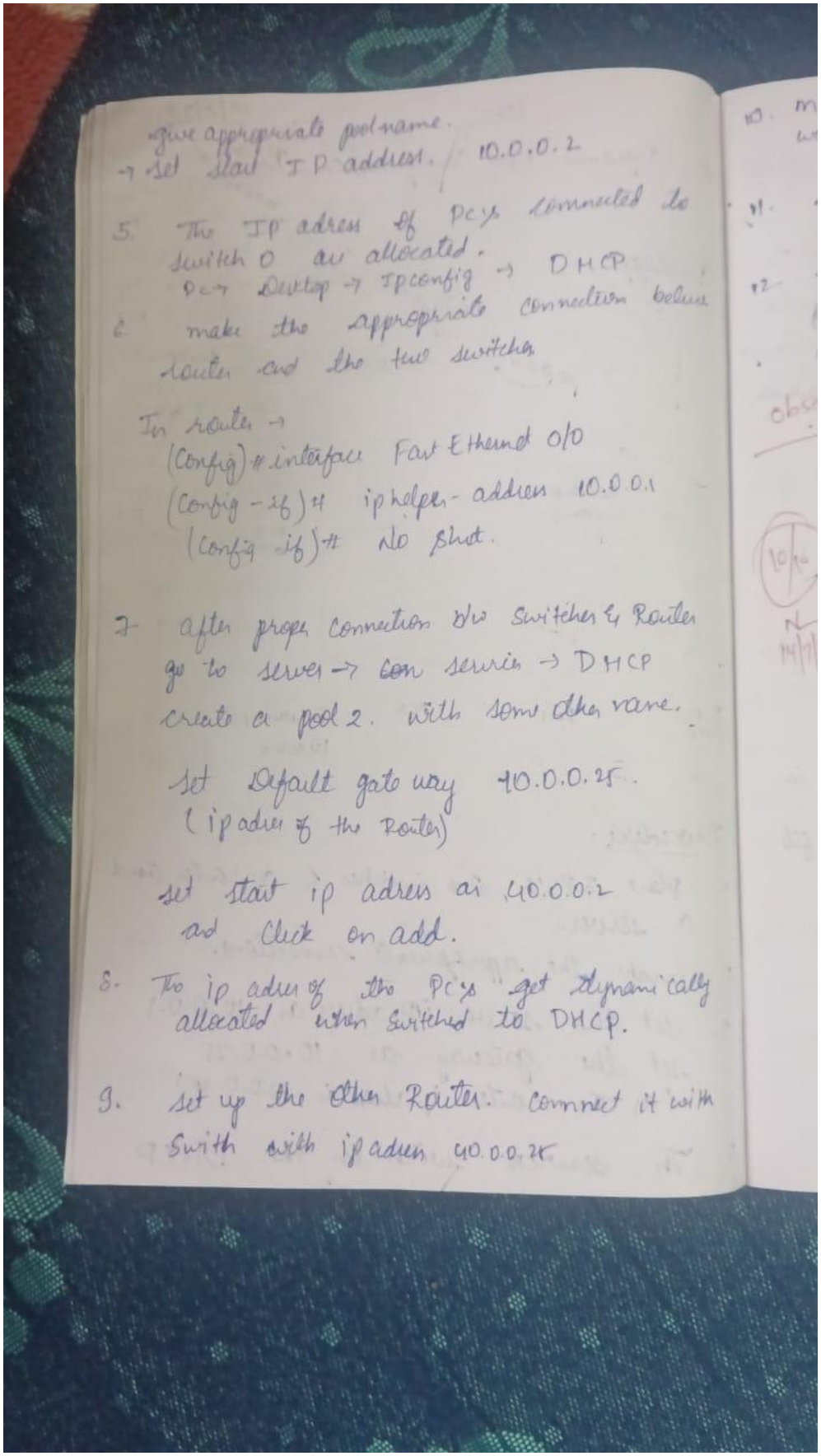
Dynamic Host Configuration Protocol

Topology:



Procedure:

1. Place 3 PC's, 2 Switches & a router and a server.
2. make the appropriate connections.
3. set the server ip adress as 10.0.0.1
   set the gateway as 10.0.0.25
   (as the router ip adress is 10.0.0.25)
4. In services switch on the DHCP



give appropriate pool name.
→ set start IP address. 10.0.0.2

5. The IP adress of PCs connected to switch 0 are allocated.
PC → Desktop → IP config → DHCP

6. make the appropriate connection below router and the two switches

In router →
(Config) # interface Fast Ethernet 0/0
(Config-if) # ip helper-address 10.0.0.1
(Config-if) # No shut.

7. after proper connection b/w switches & Router go to server → Con service → DHCP
create a pool 2. with some other name.

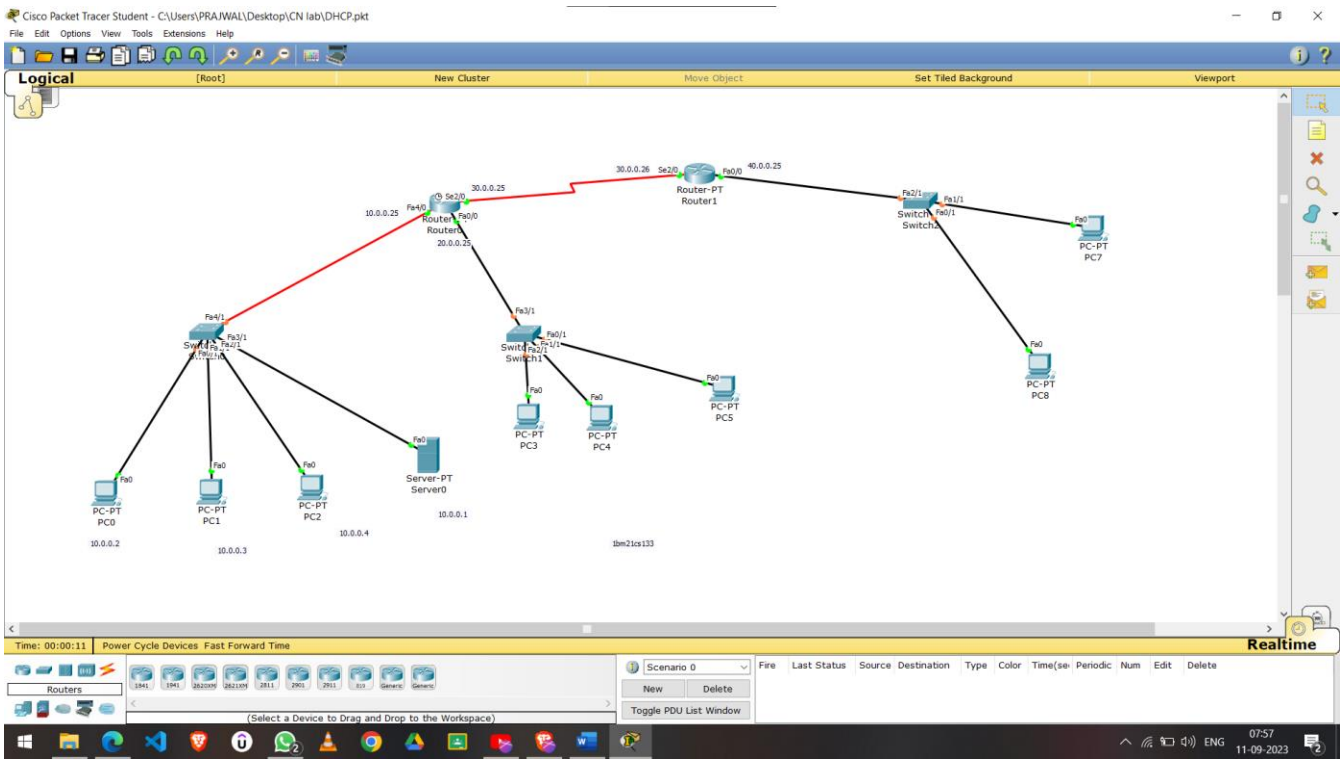
set Default gate way 40.0.0.25.
(ip adres of the Router)

set start ip adress as 40.0.0.2
and Click on add.

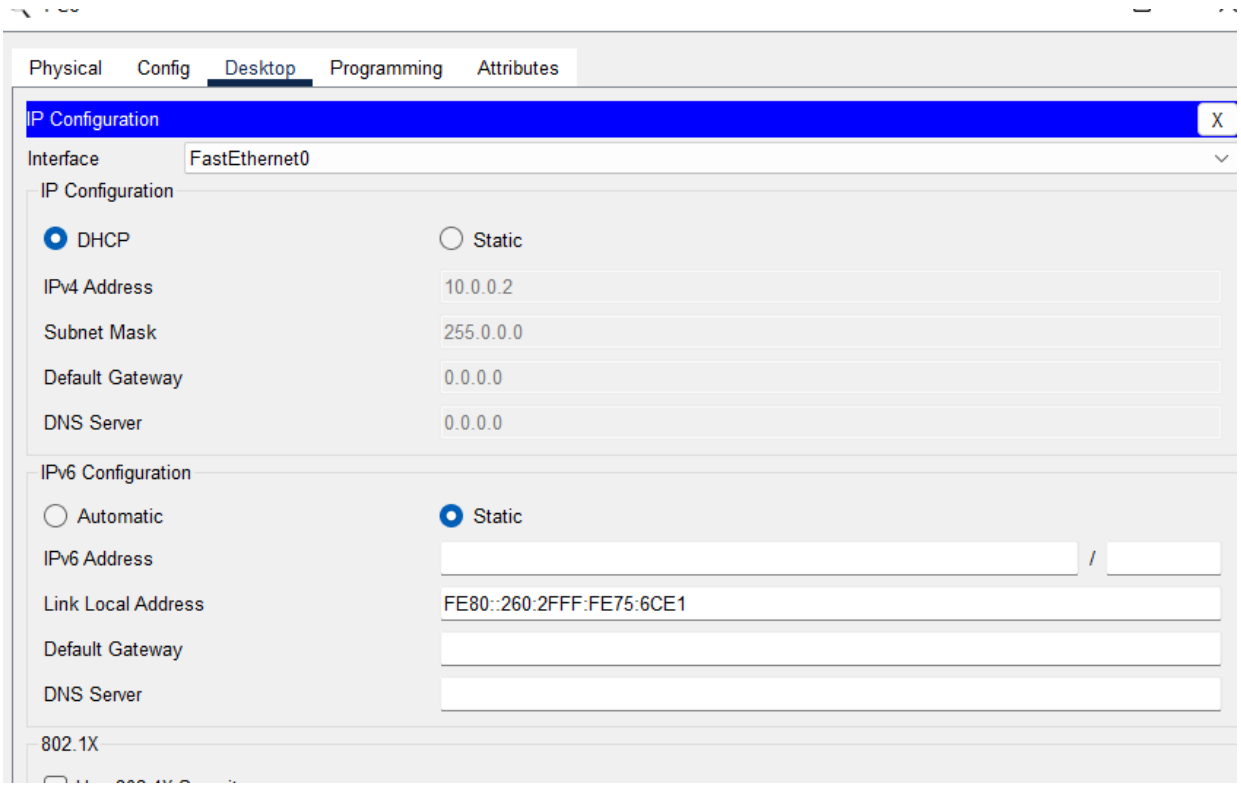
8. The ip adres of the PCs get dynamically allocated when switched to DHCP.

9. set up the other Router. Connect it with Swith with ip adres 40.0.0.25

Topology:

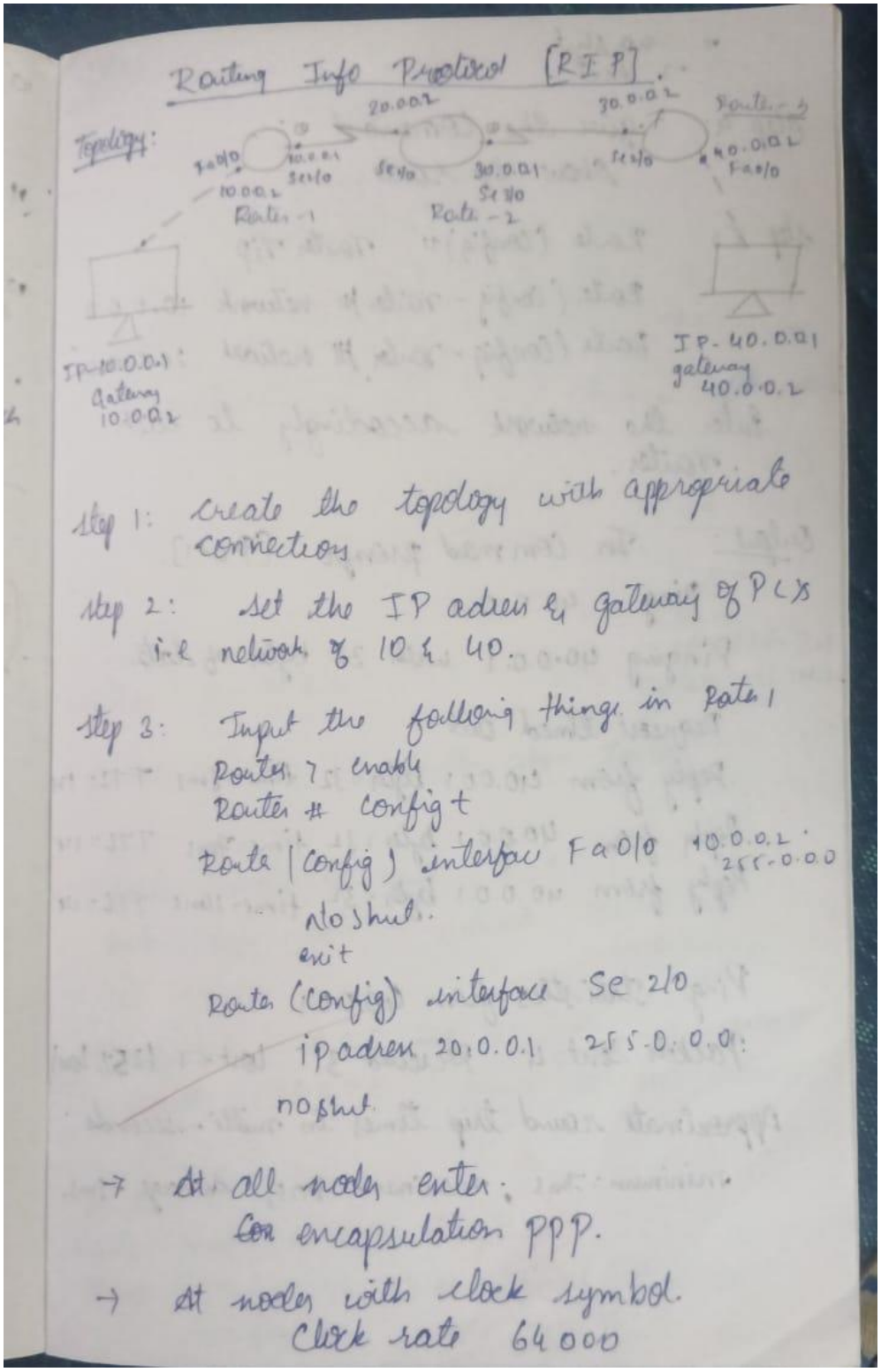


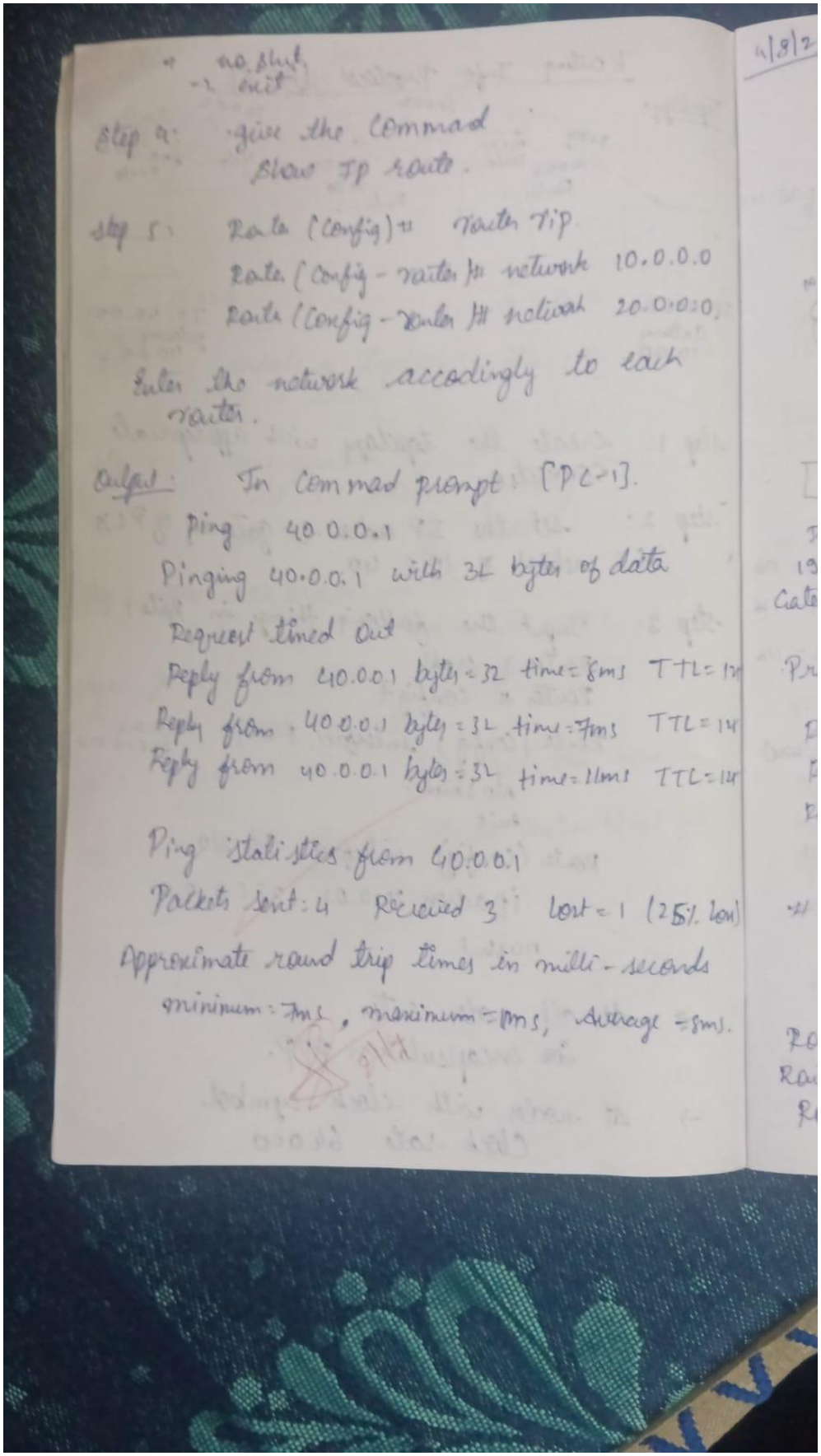
Output:



Experiment 5

Configure RIP routing Protocol in Routers





no shut
exit

Step 4: give the command
show ip route.

Step 5: Router (config)# router rip
Router (config-router)# network 10.0.0.0
Router (config-router)# network 20.0.0.0

Enter the network accordingly to each router.

Output: In command prompt: [PC-1].

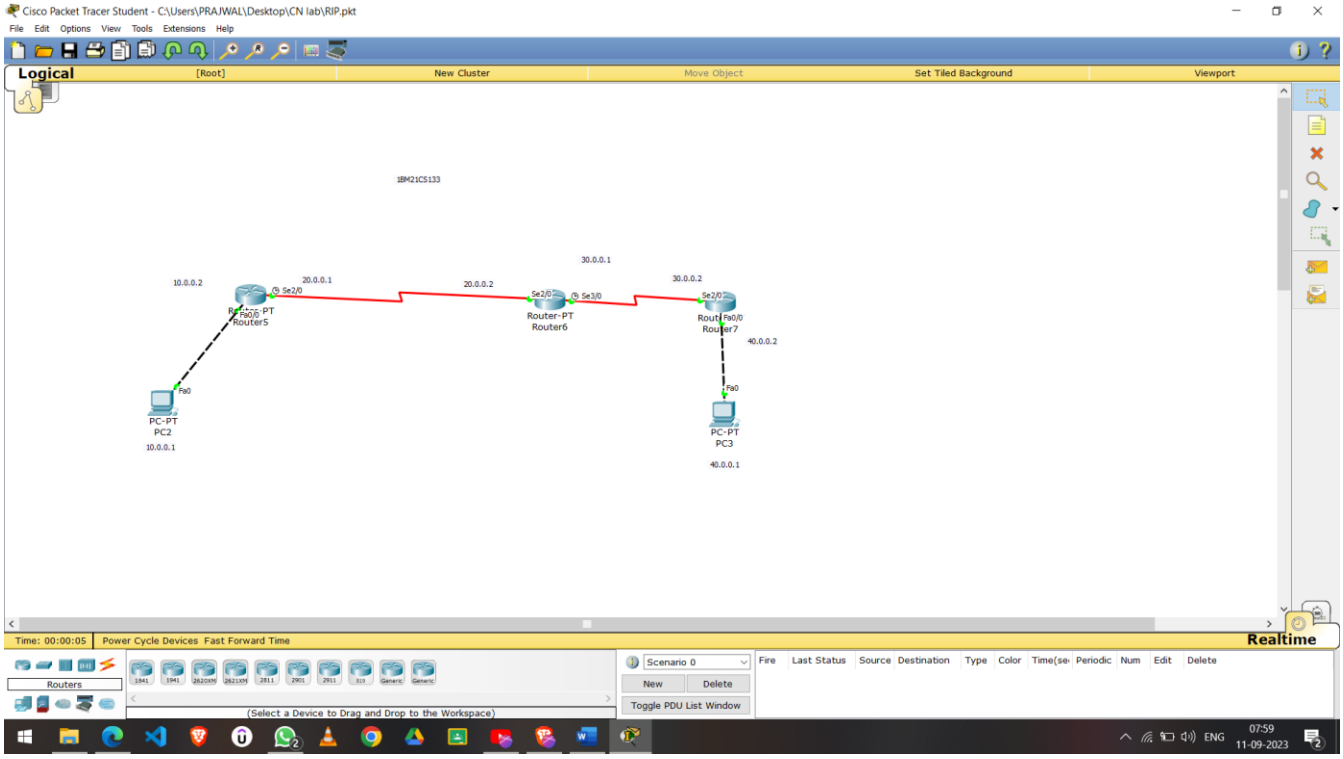
Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

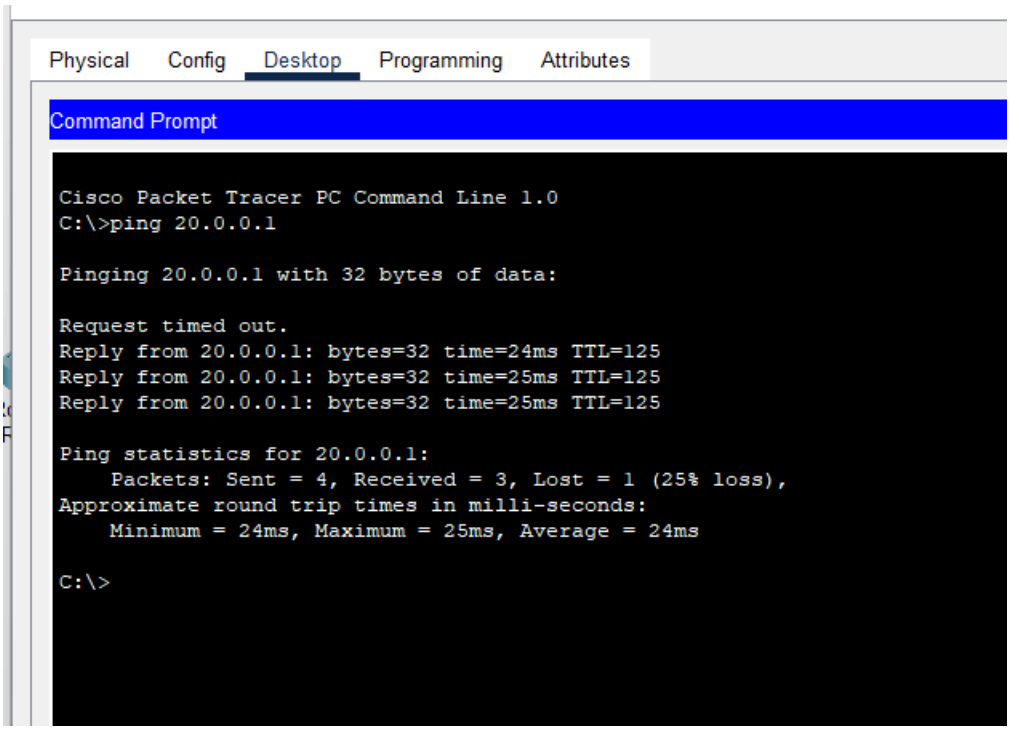
Request timed out
Reply from 40.0.0.1 bytes=32 time=8ms TTL=12
Reply from 40.0.0.1 bytes=32 time=7ms TTL=12
Reply from 40.0.0.1 bytes=32 time=11ms TTL=12

Ping statistics from 40.0.0.1
Packets Sent: 4 Received 3 Lost = 1 (25% loss)
Approximate round trip times in milli-seconds
minimum = 7ms, maximum = 11ms, Average = 8ms.

Topology:

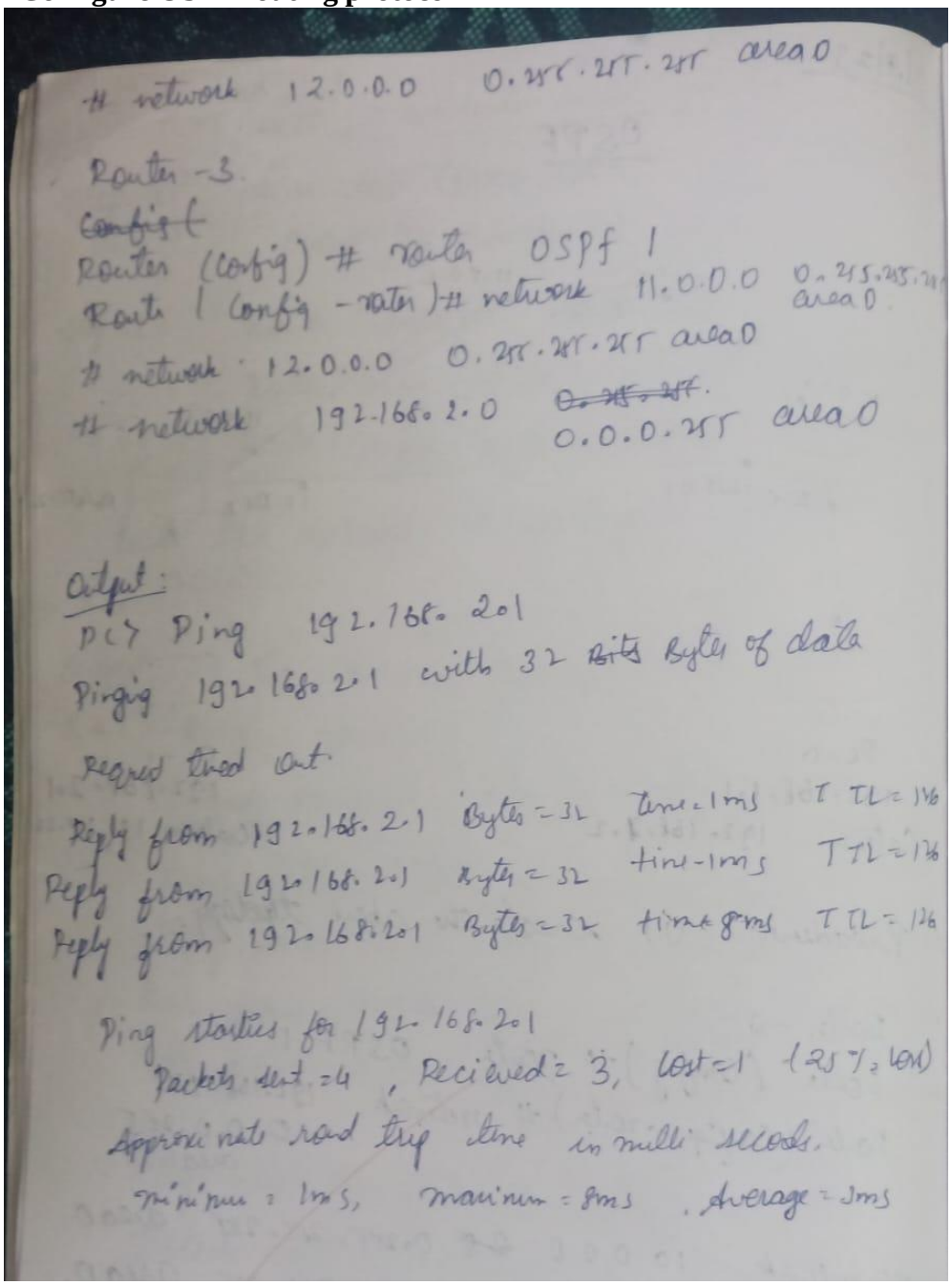


Output:



Experiment 6

Configure OSPF routing protocol



network 12.0.0.0 0.255.255.255 area 0

Router -3

~~Config t~~

Router (config) # router OSPf 1

Router (config-router)# network 11.0.0.0 0.255.255.255 area 0

network 12.0.0.0 0.255.255.255 area 0

network 192.168.2.0 ~~0.255.255.~~ 0.0.0.255 area 0

Output:

PC> Ping 192.168.2.1

Pinging 192.168.2.1 with 32 ~~Bits~~ Bytes of data

Request timed out.

Reply from 192.168.2.1 Bytes=32 time=1ms TTL=126

Reply from 192.168.2.1 Bytes=32 time=1ms TTL=126

Reply from 192.168.2.1 Bytes=32 time=8ms TTL=126

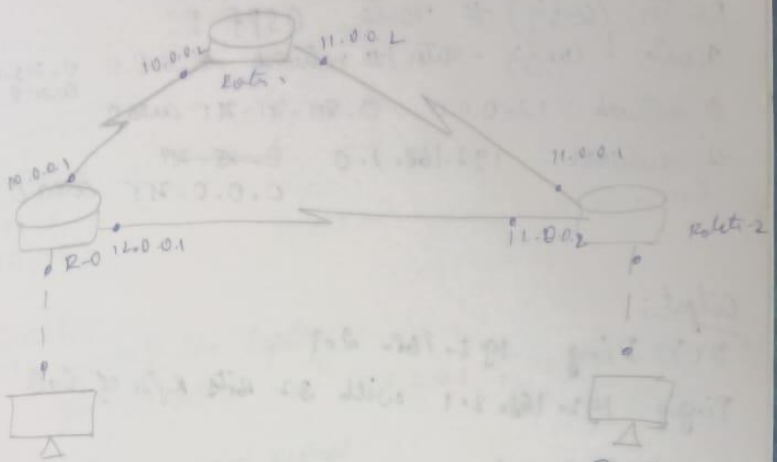
Ping statics for 192.168.2.1

Packets sent=4, Recieved=3, Lost=1 (25% loss)

Approximate round trip time in milli seconds.

minimum = 1ms, maximum = 8ms, Average = 3ms

OSPF



PC-0
192.168.1.1
Gateway - 192.168.1.2

PC-1
192.168.2.1
Gateway 192.168.2.2

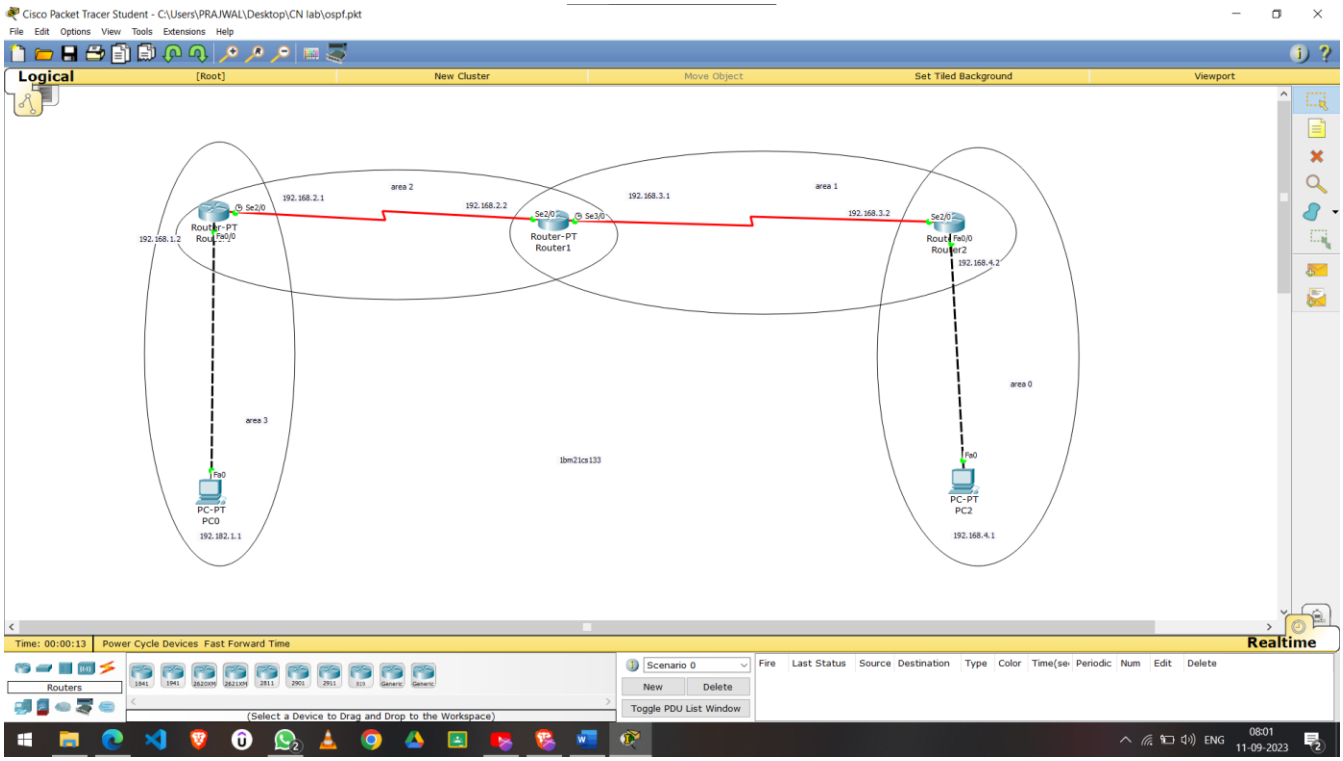
Procedure: (1) create the above topology:

Router-0
Router (config)# router OSPF 1
Router (config-router)# network 192.168.1.0 0.0.0.255 area 0.

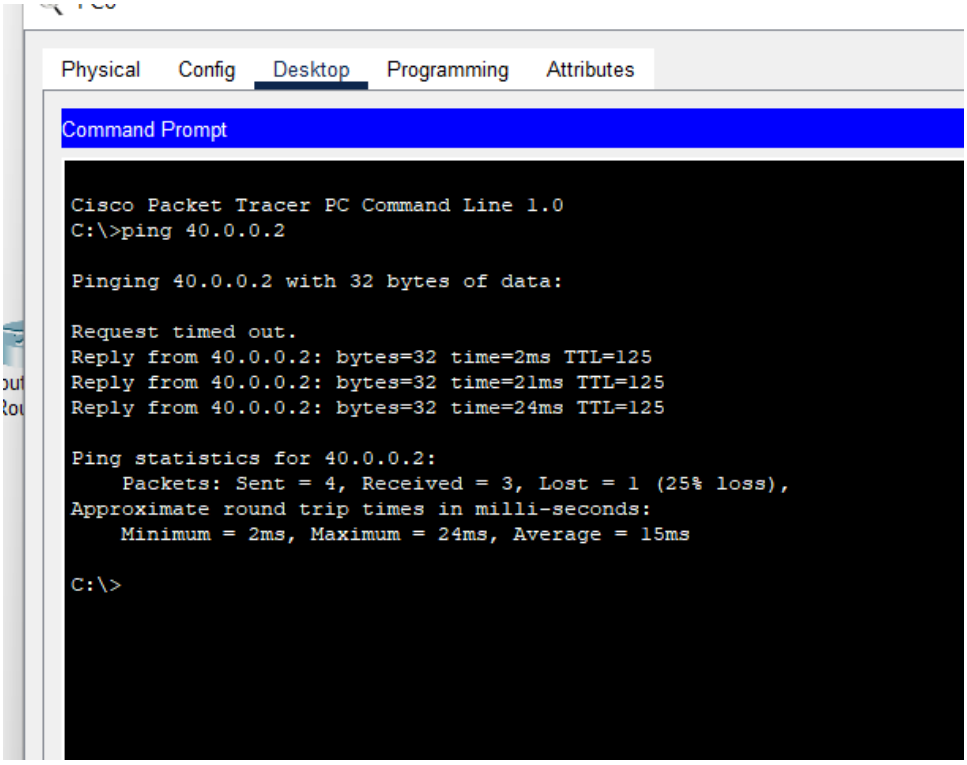
network 10.0.0.0 ~~255~~ 0.255.255.255 area 0
network 12.0.0.0 0.255.255.255 area 0.

Router-1
Router (config)# router OSPF 1
Router (config-router)# network ~~192.168.~~
10.0.0.0 0.255.255.255 area 0

Topology:

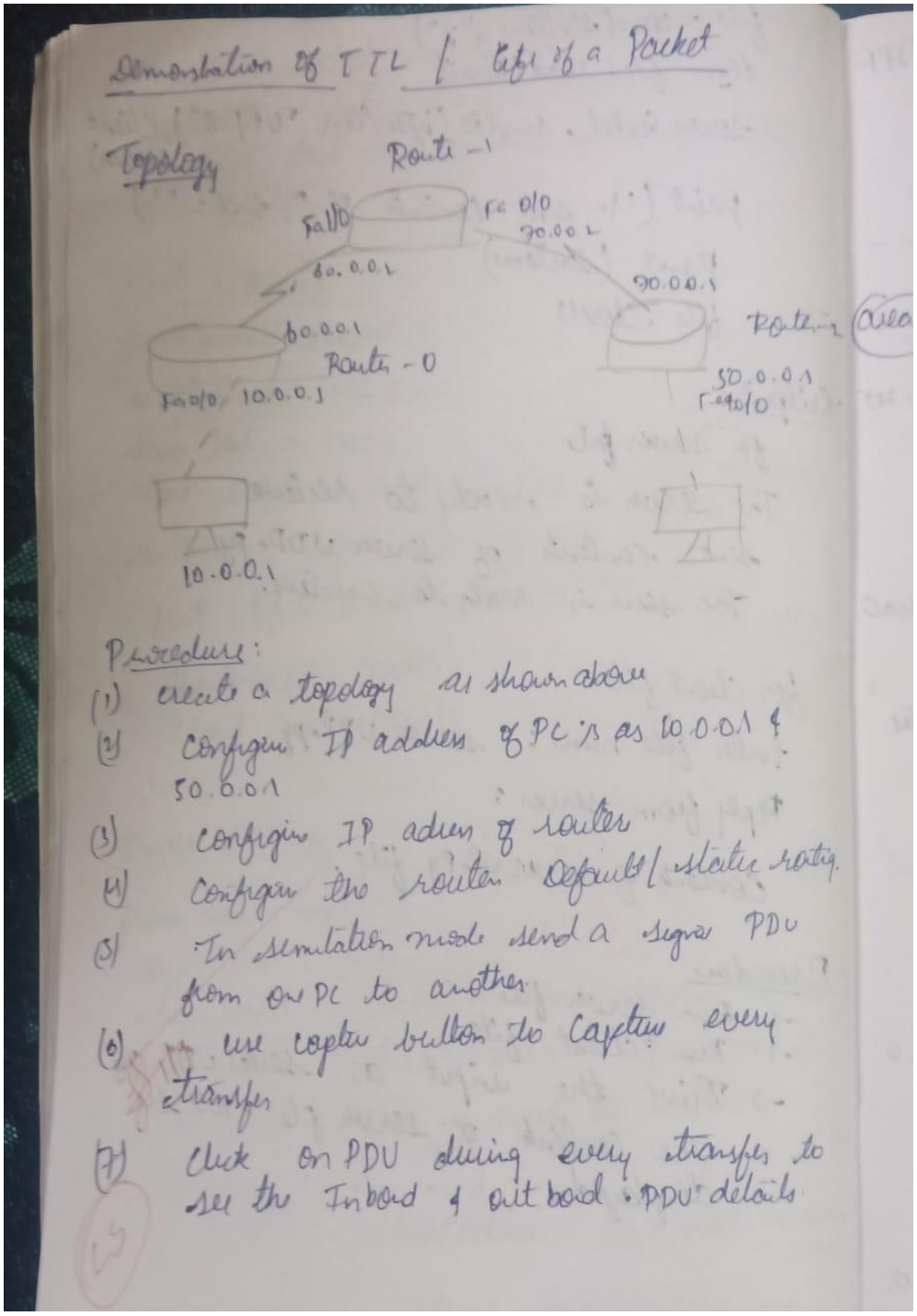


Output:

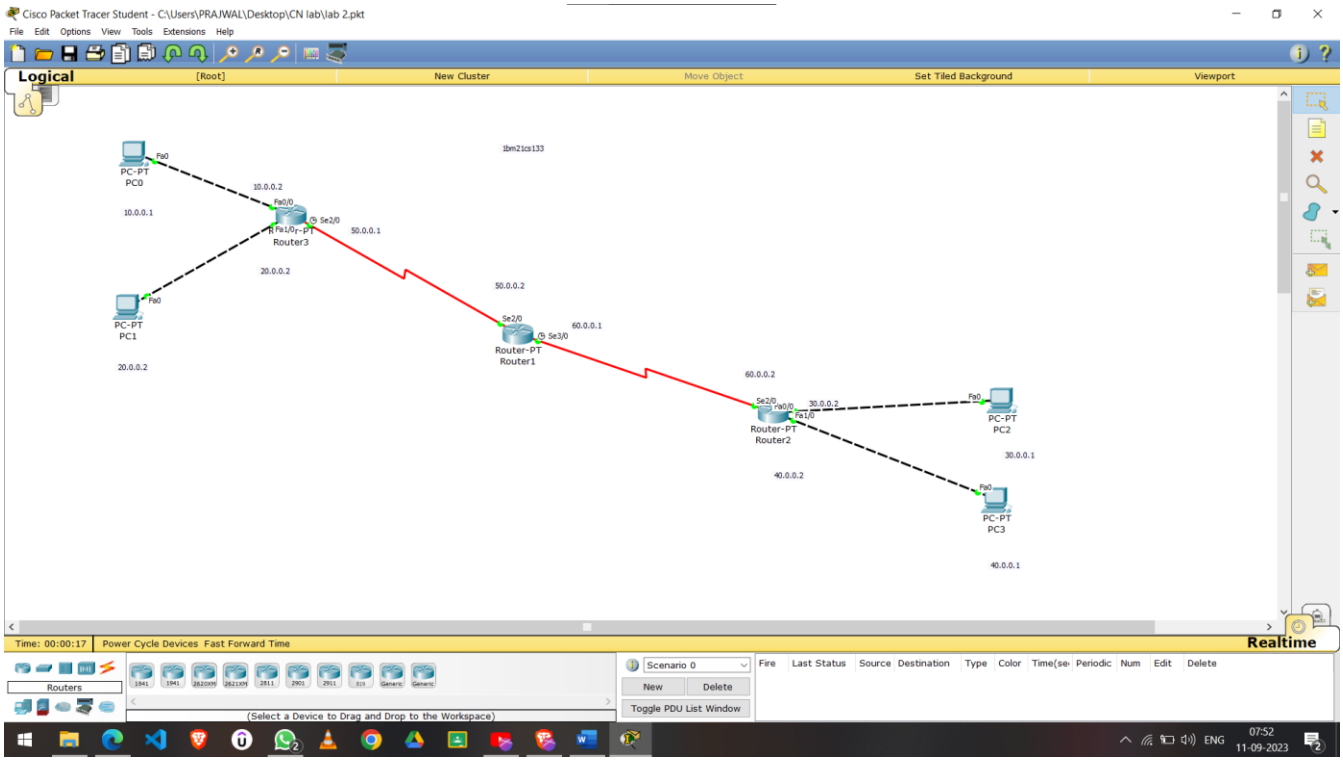


Experiment 7

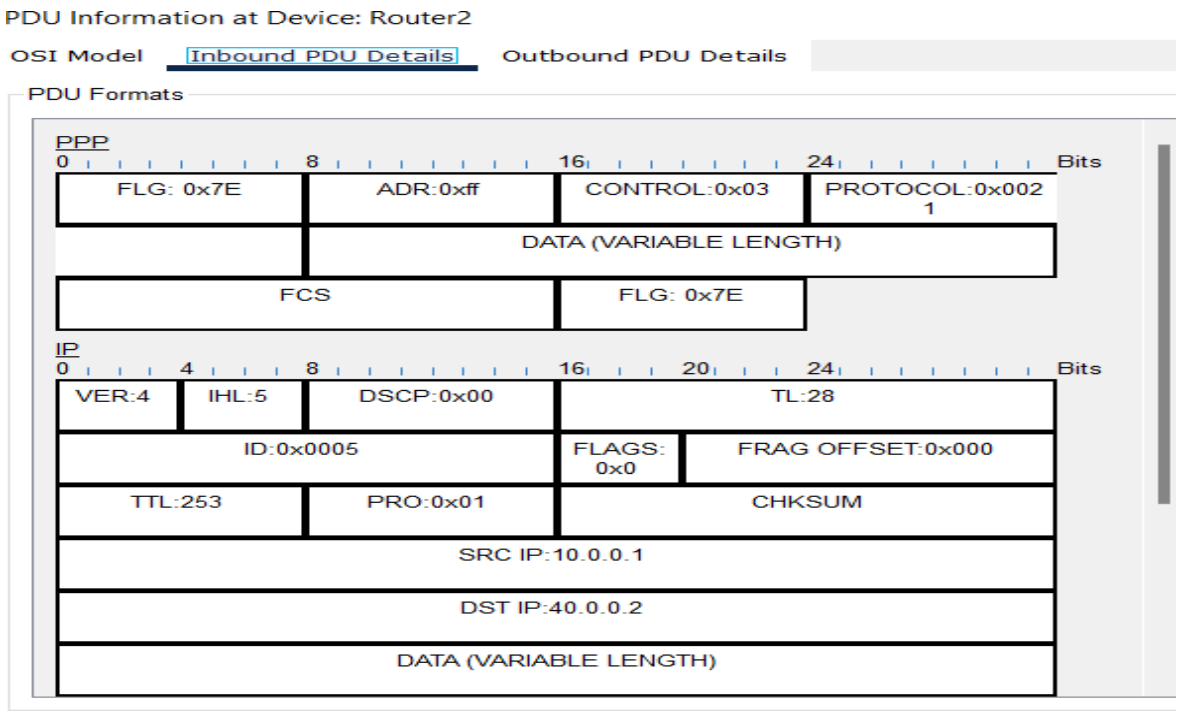
Demonstrate the TTL/ Life of a Packet

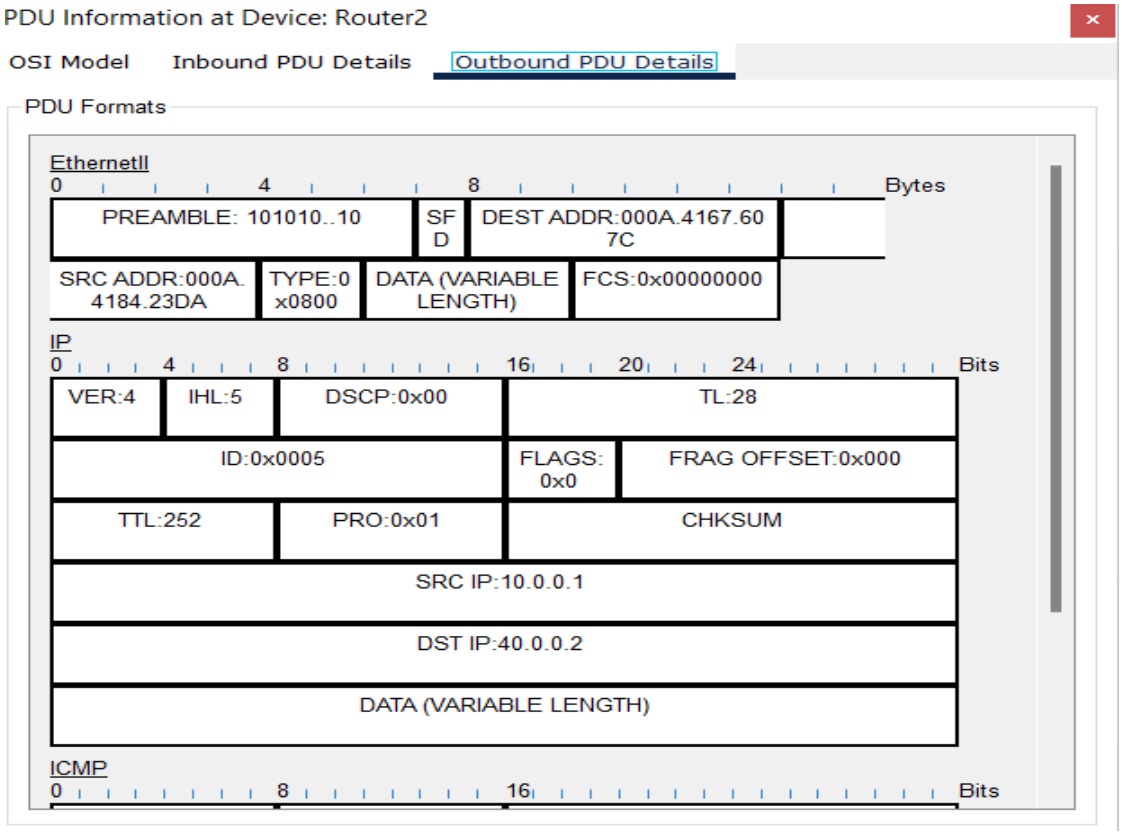


Topology:


Cisco Packet Tracer Student - C:\Users\PRAJWAL\Desktop\CN lab\lab 2.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
lbm21cs133
PC-PT PC0
Fa0
10.0.0.2
10.0.0.1
Fa0/0
Fa1/0
Se2/0
Router-PT Router3
50.0.0.1
20.0.0.2
PC-PT PC1
Fa0
20.0.0.2
50.0.0.2
Se2/0
Se3/0
Router-PT Router1
60.0.0.1
60.0.0.2
Se2/0
Fa0/0
Fa1/0
Router-PT Router2
30.0.0.2
PC-PT PC2
Fa0
30.0.0.1
40.0.0.2
PC-PT PC3
Fa0
40.0.0.1
Time: 00:00:17
Power Cycle Devices Fast Forward Time
Realtime
Routers
(Select a Device to Drag and Drop to the Workspace)
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(se
Periodic
Num
Edit
Delete
ENG
07:52
11-09-2023

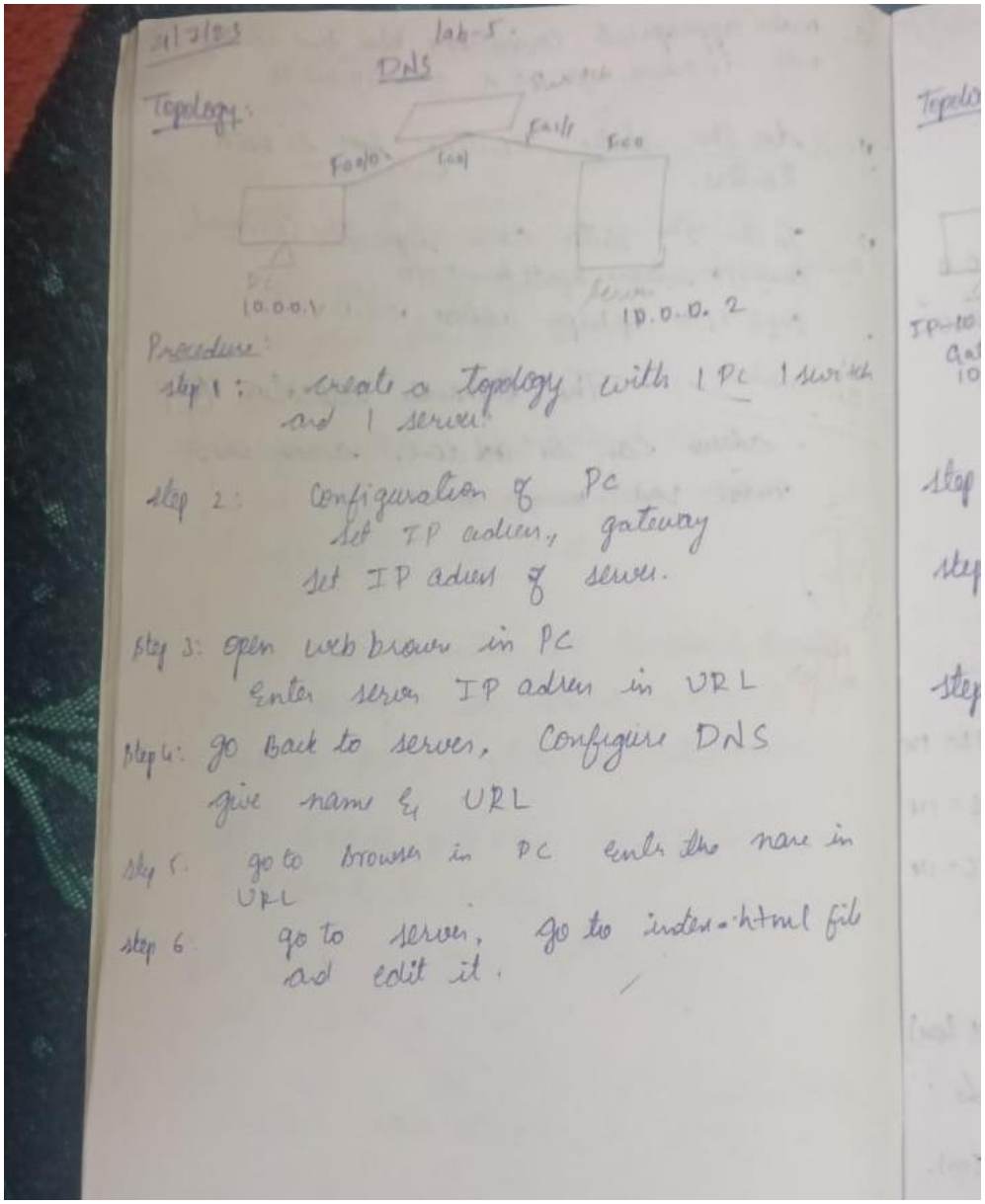
Output:


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
0 8 16 24 Bits
FLG: 0x7E
ADR:0xff
CONTROL:0x03
PROTOCOL:0x0021
DATA (VARIABLE LENGTH)
FCS
FLG: 0x7E
IP
0 4 8 16 20 24 Bits
VER:4
IHL:5
DSCP:0x00
TL:28
ID:0x0005
FLAGS: 0x0
FRAG OFFSET:0x000
TTL:253
PRO:0x01
CHKSUM
SRC IP:10.0.0.1
DST IP:40.0.0.2
DATA (VARIABLE LENGTH)

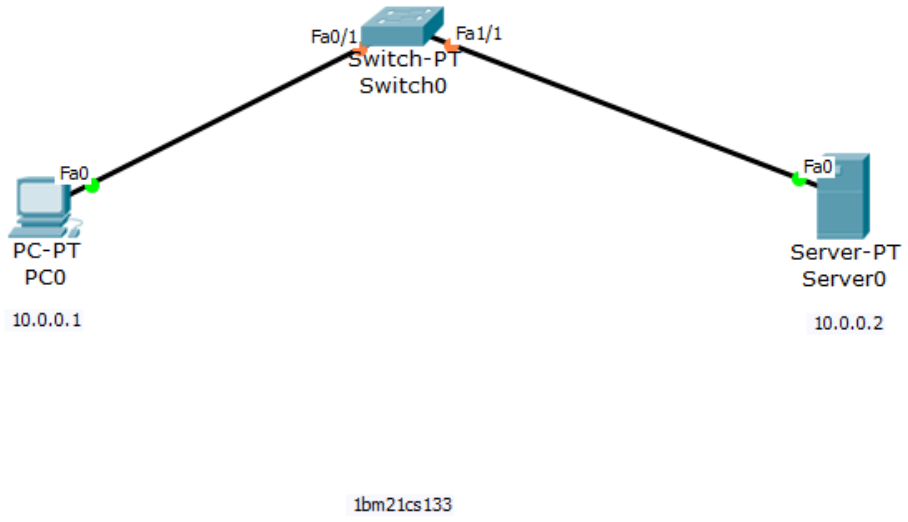

PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
EthernetII
0 4 8 Bytes
PREAMBLE: 101010..10
SFD
DEST ADDR:000A.4167.607C
SRC ADDR:000A.4184.23DA
TYPE:0x0800
DATA (VARIABLE LENGTH)
FCS:0x00000000
IP
0 4 8 16 20 24 Bits
VER:4
IHL:5
DSCP:0x00
TL:28
ID:0x0005
FLAGS: 0x0
FRAG OFFSET:0x000
TTL:252
PRO:0x01
CHKSUM
SRC IP:10.0.0.1
DST IP:40.0.0.2
DATA (VARIABLE LENGTH)
ICMP
0 8 16 Bits

Experiment 8

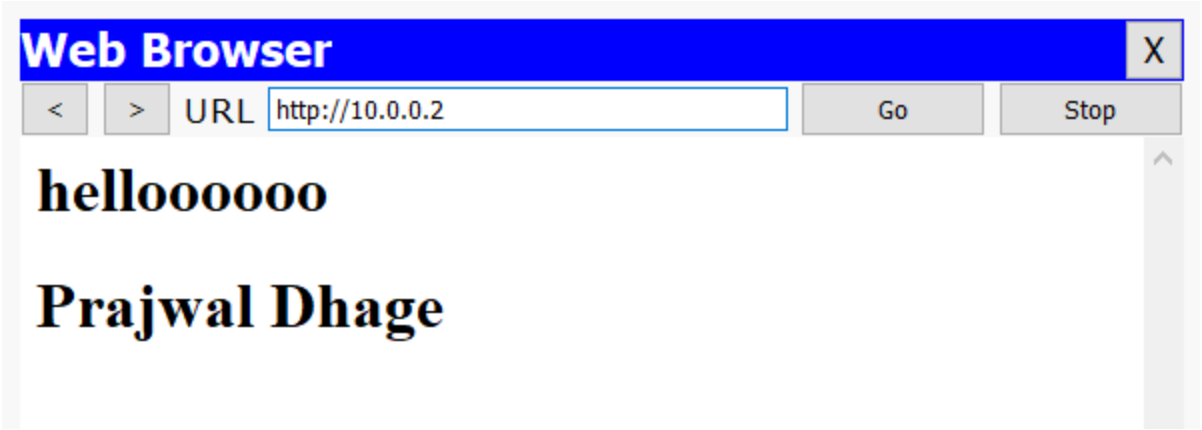
Configure Web Server, DNS within a LAN.



Topology:

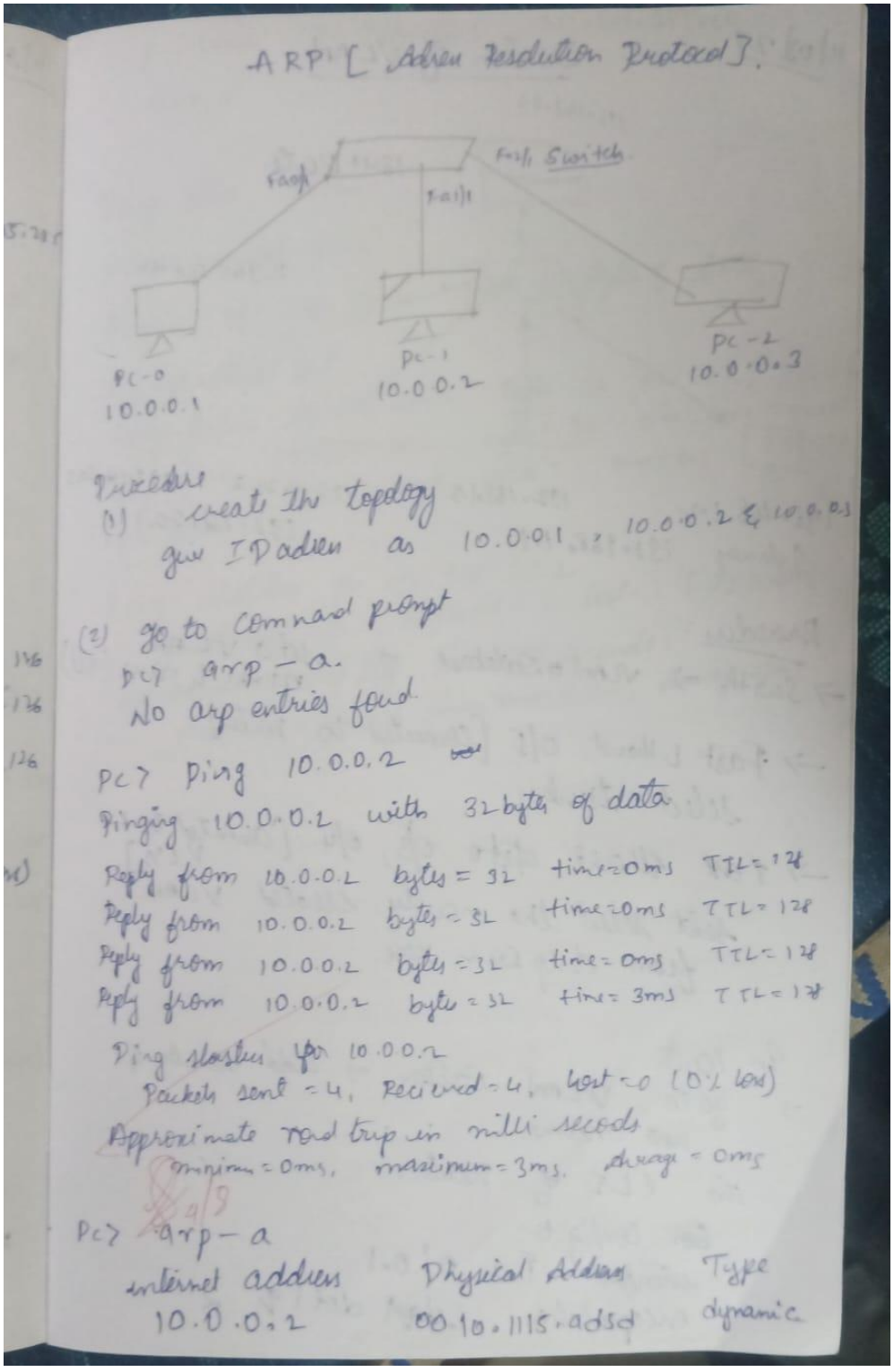


Output:



Experiment 9

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



ARP [Adress Resolution Protocol].

Procedure

(1) create the topology
give IP adress as 10.0.0.1, 10.0.0.2 & 10.0.0.3

(2) go to comnand prompt

PC> arp -a.
No arp entries found

PC> Ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

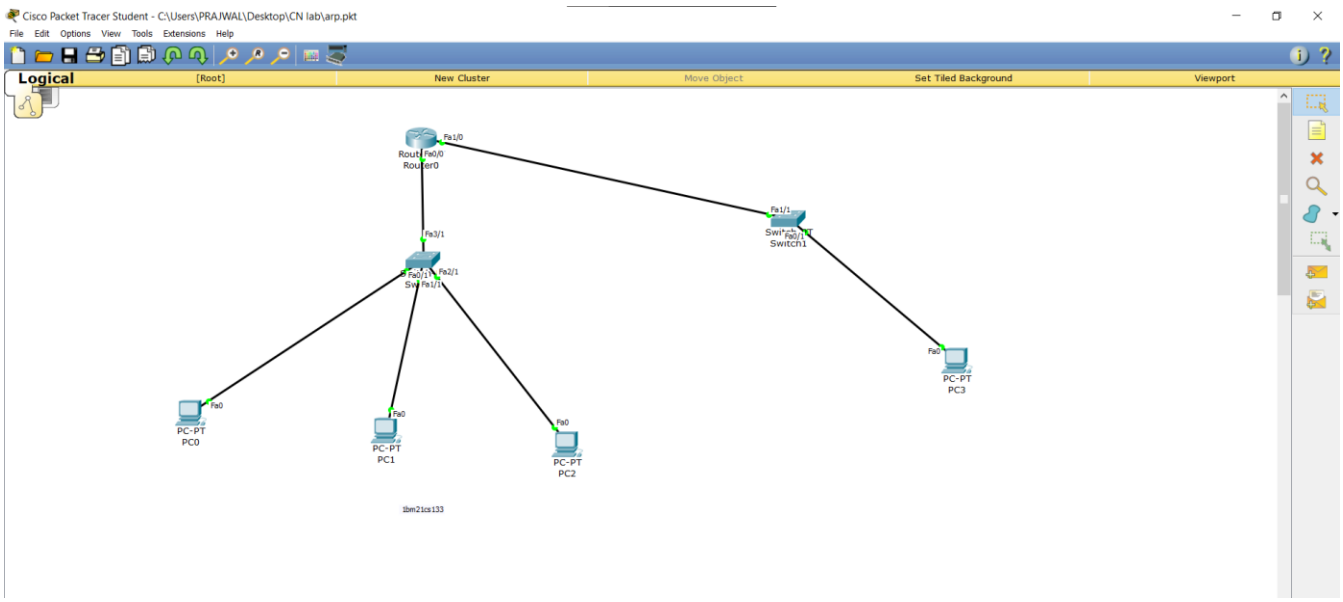
Reply from 10.0.0.2 bytes = 32 time=0ms TTL=128
Reply from 10.0.0.2 bytes = 32 time=0ms TTL=128
Reply from 10.0.0.2 bytes = 32 time=0ms TTL=128
Reply from 10.0.0.2 bytes = 32 time=3ms TTL=128

Ping statistics for 10.0.0.2
Packets sent = 4, Recieved = 4, Lost = 0 (0% loss)
Approximate round trip in milli seconds
minimum = 0ms, maximum = 3ms, Average = 0ms

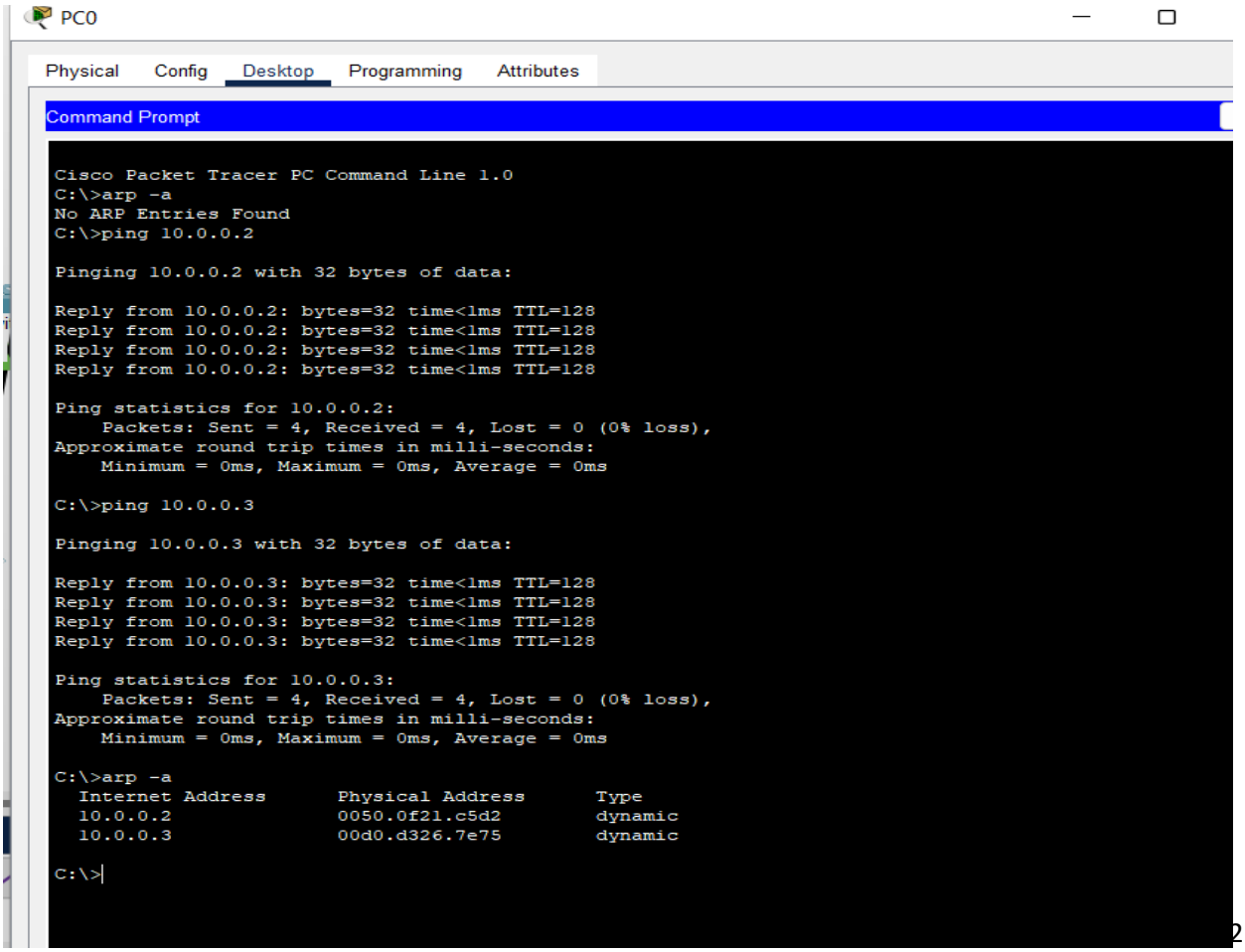
PC> arp -a

internet address	Physical Address	Type
10.0.0.2	00.10.1115.adsd	dynamic

Topology:

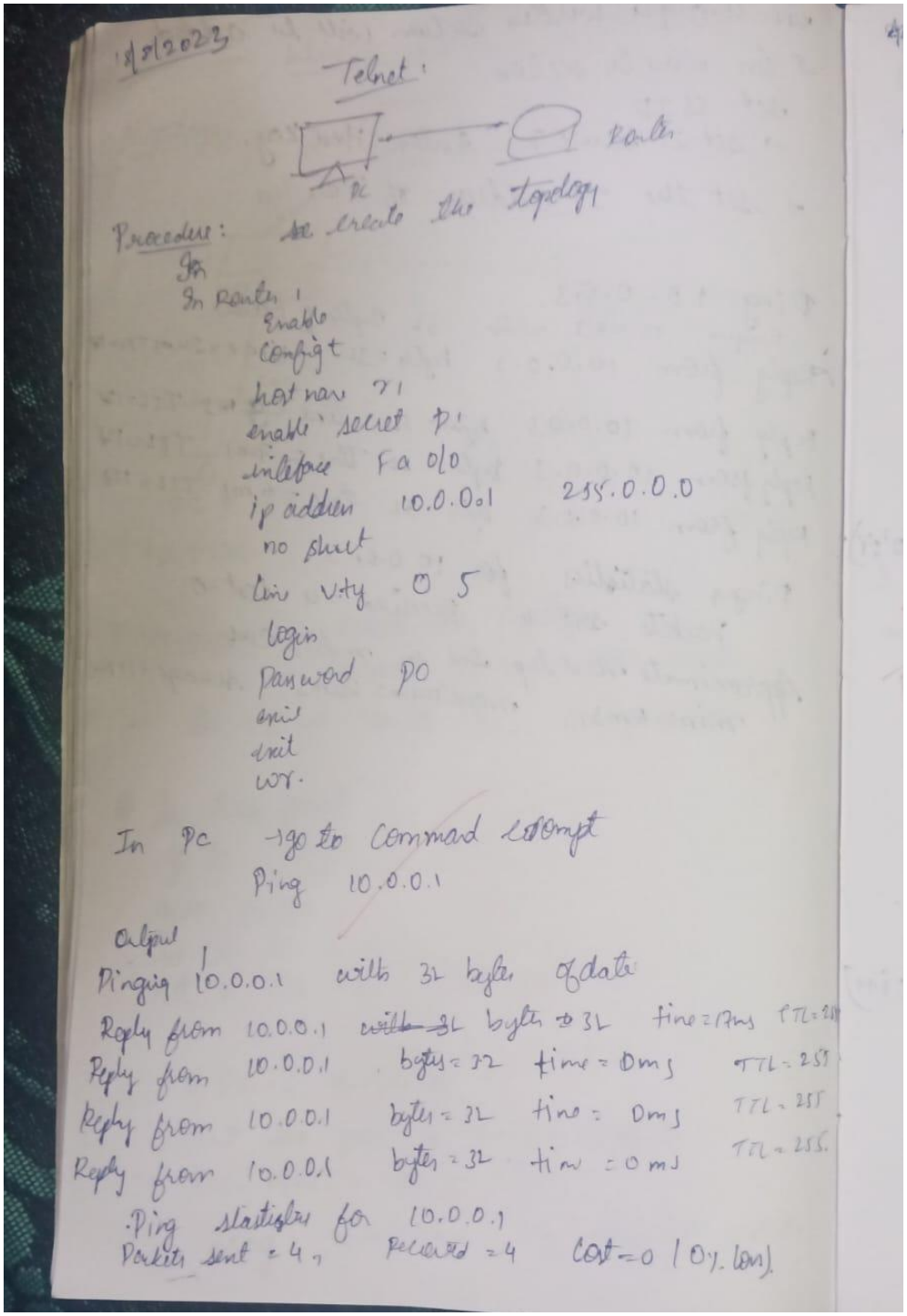


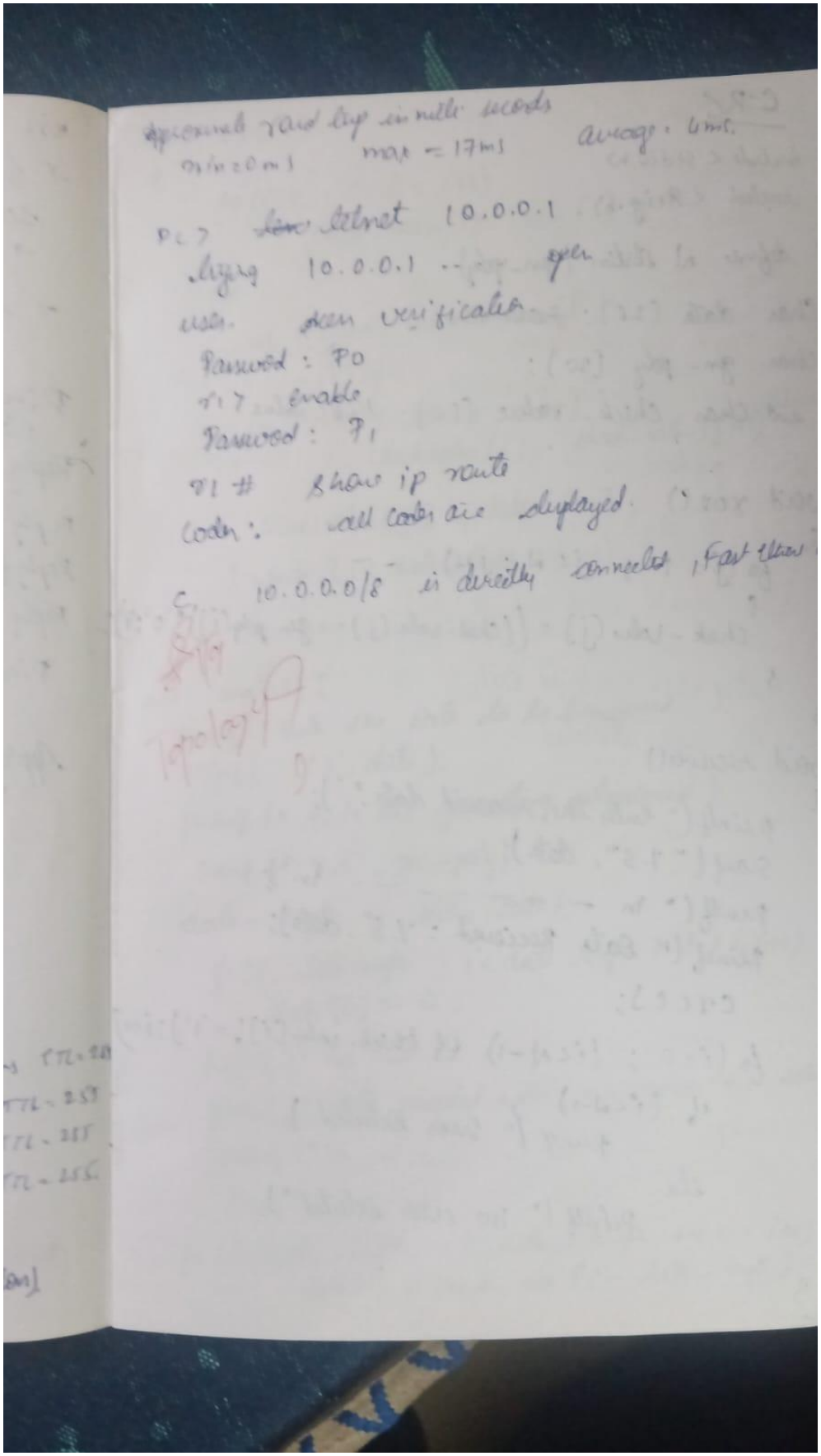
Output:



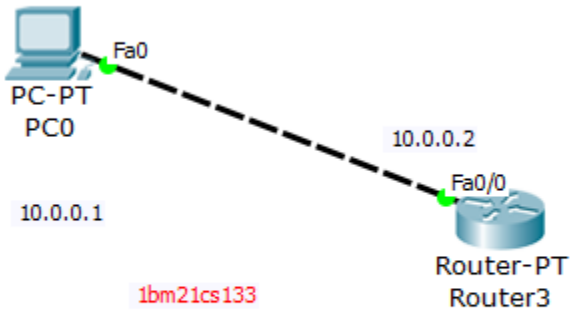
Experiment 10

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

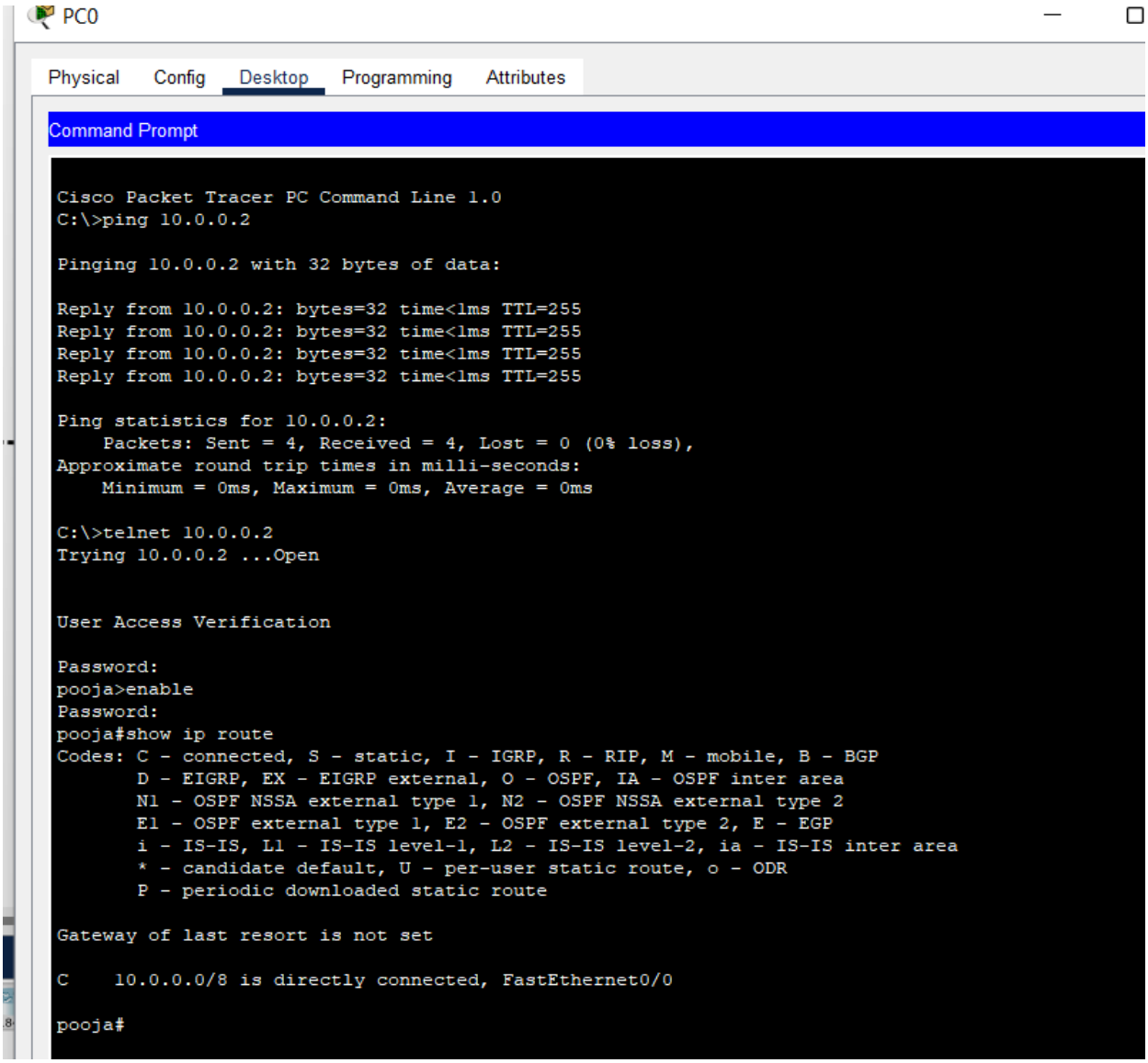



Approximate round trip in milli seconds
min = 0ms
max = 17ms
average = 6ms.
PC> telnet 10.0.0.1
Trying 10.0.0.1 ... open
user access verification
Password: P0
R1> enable
Password: P1
R1# show ip route
Codes: all codes are displayed
C 10.0.0.0/8 is directly connected, FastEthernet
Topology

Topology:

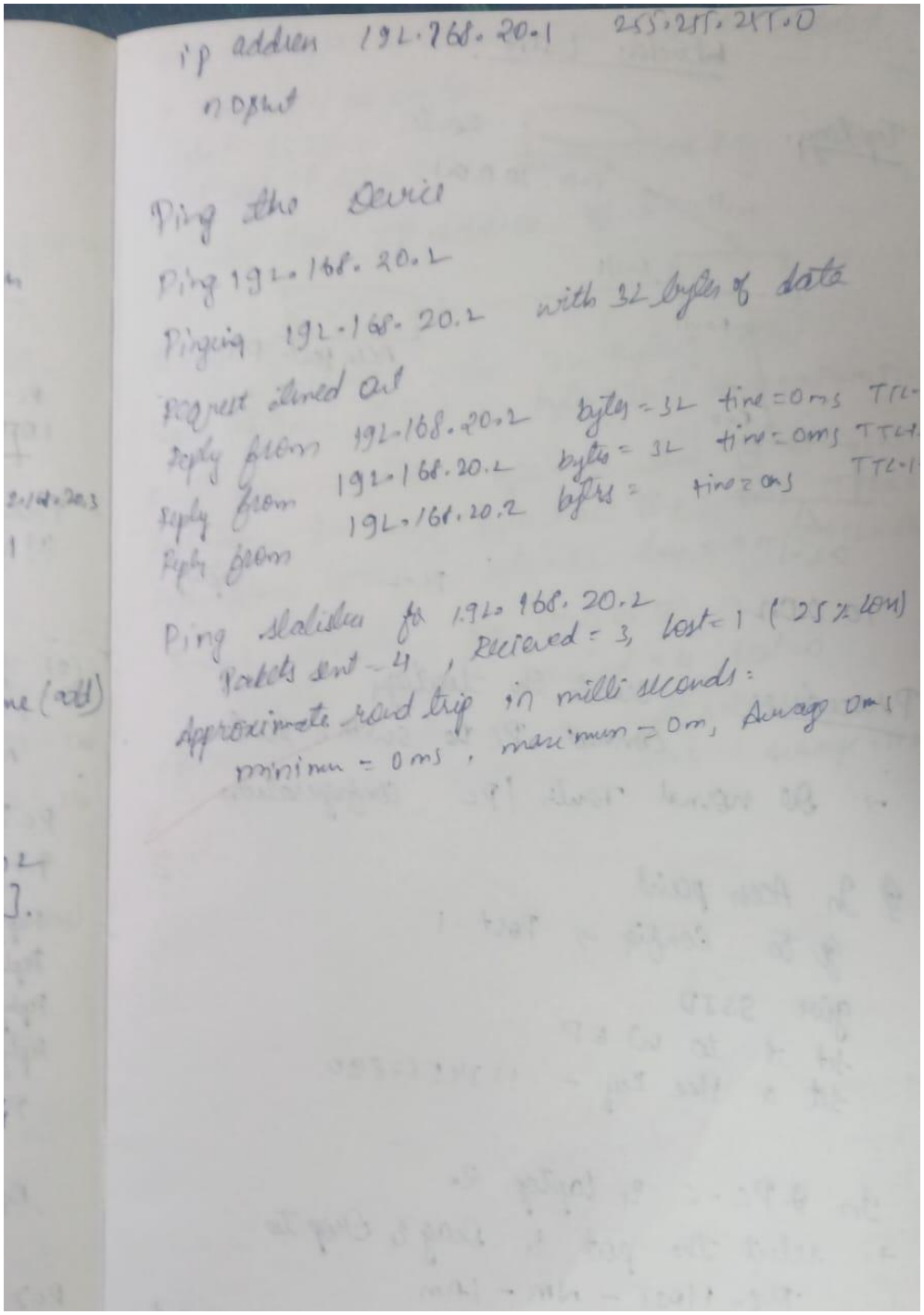


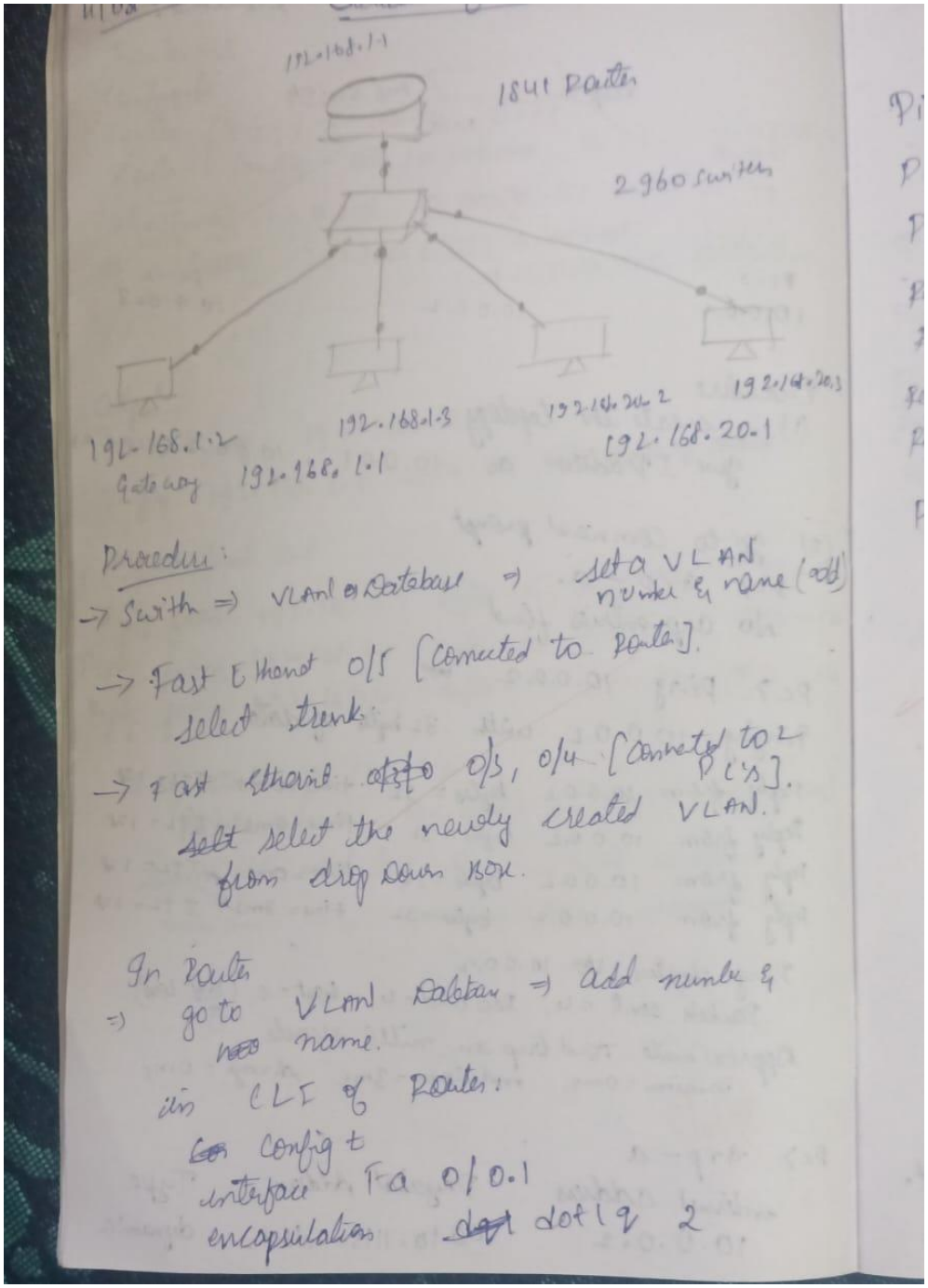
0utput:


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time<1ms TTL=255
Reply from 10.0.0.2: bytes=32 time<1ms TTL=255
Reply from 10.0.0.2: bytes=32 time<1ms TTL=255
Reply from 10.0.0.2: bytes=32 time<1ms 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
C:\>telnet 10.0.0.2
Trying 10.0.0.2 ...Open
User Access Verification
Password:
pooja>enable
Password:
pooja#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
pooja#

Experiment 11

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





Procedure:

→ Switch ⇒ VLAN Database ⇒ set a VLAN number & name (add)

→ Fast Ethernet 0/5 [connected to Router]. select trunk.

→ Fast Ethernet 0/3, 0/4 [connected to PC's]. select the newly created VLAN. from drop down box.

In Router

⇒ go to VLAN Database ⇒ add number & name.

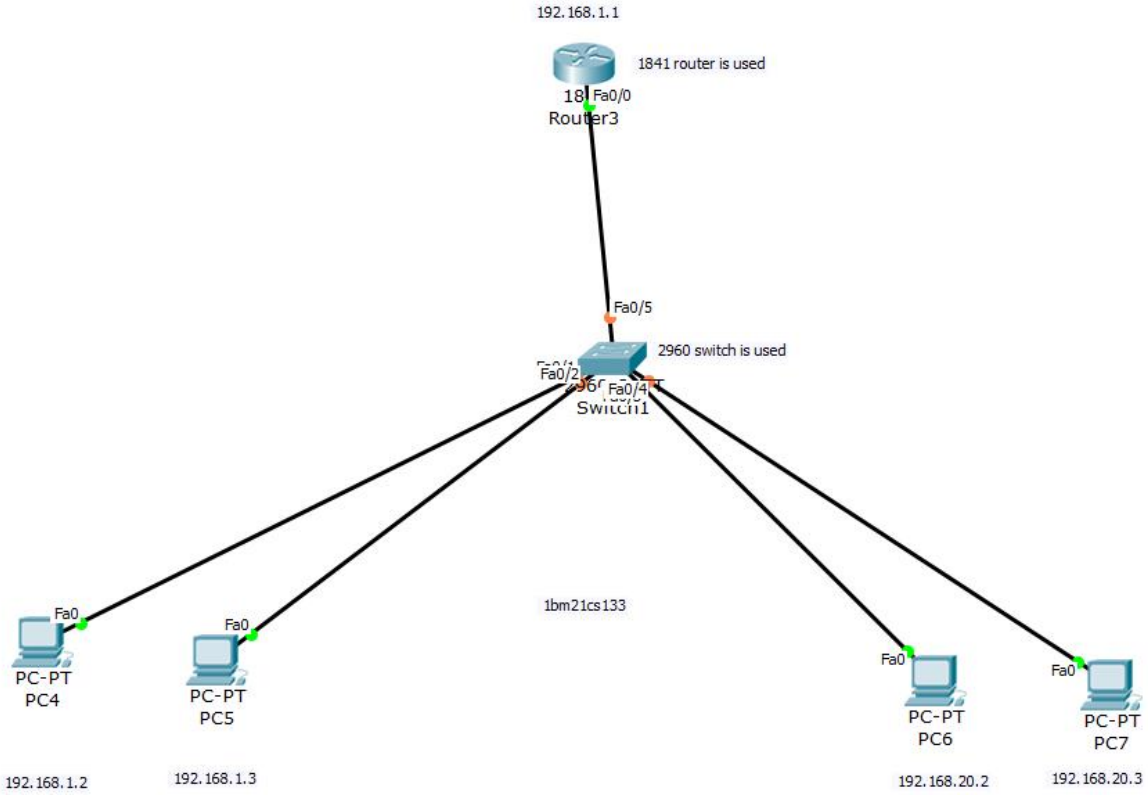
in CLI of Router:

Config t

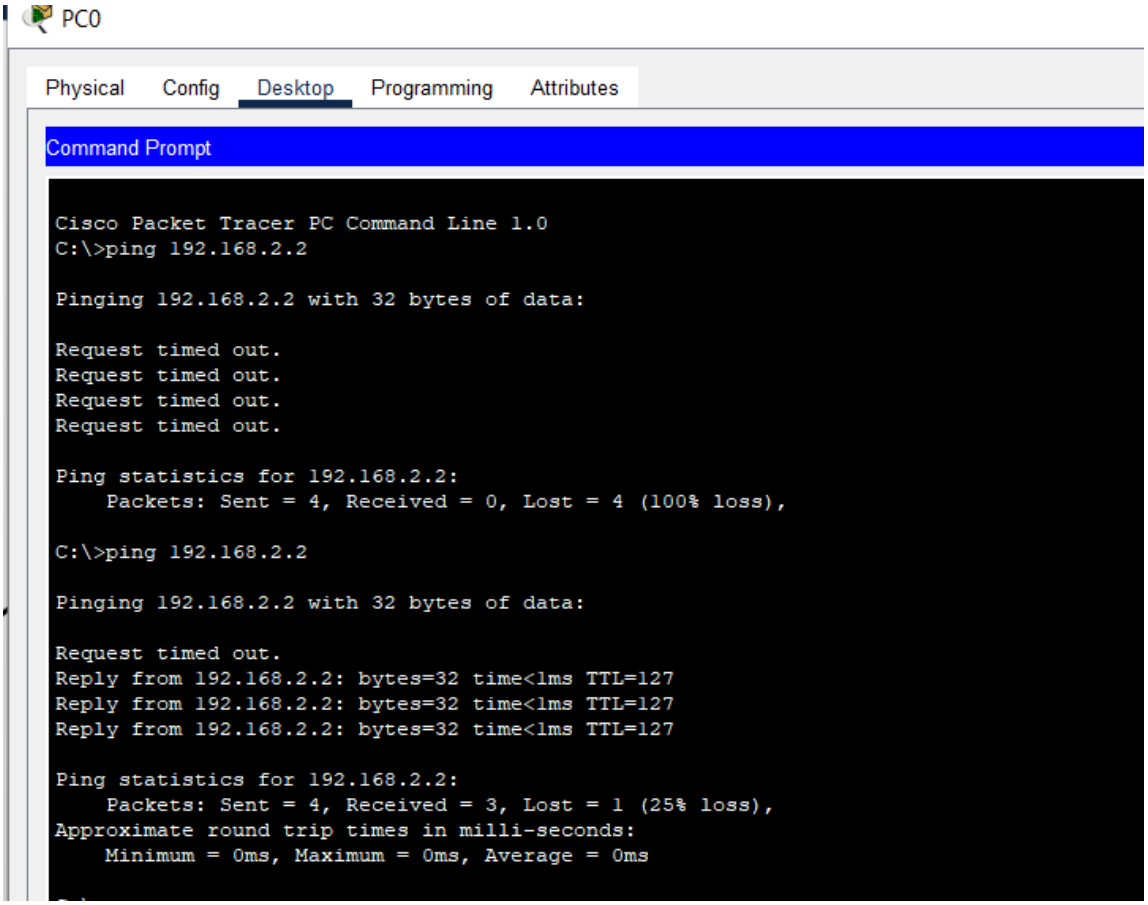
interface Fa 0/0.1

encapsulation dot1q 2

Topology:

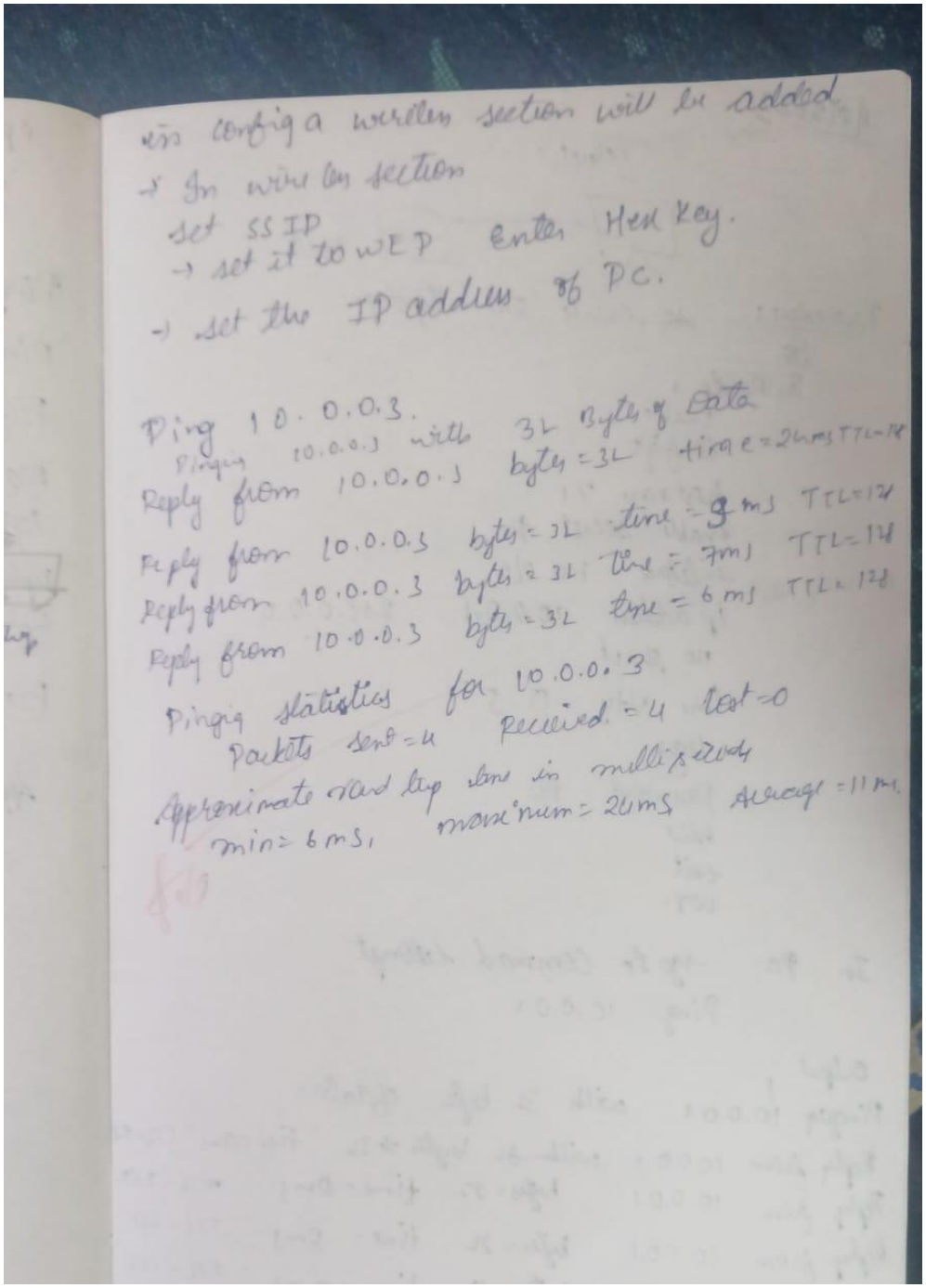

192.168.1.1
1841 router is used
Fa0/0
Router3
Fa0/5
2960 switch is used
Fa0/2
Fa0/4
Switch1
1bm21cs133
Fa0
PC-PT
PC4
192.168.1.2
Fa0
PC-PT
PC5
192.168.1.3
Fa0
PC-PT
PC6
192.168.20.2
Fa0
PC-PT
PC7
192.168.20.3

Output:


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>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<1ms TTL=127
Reply from 192.168.2.2: bytes=32 time<1ms TTL=127
Reply from 192.168.2.2: bytes=32 time<1ms 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 = 0ms, Maximum = 0ms, Average = 0ms

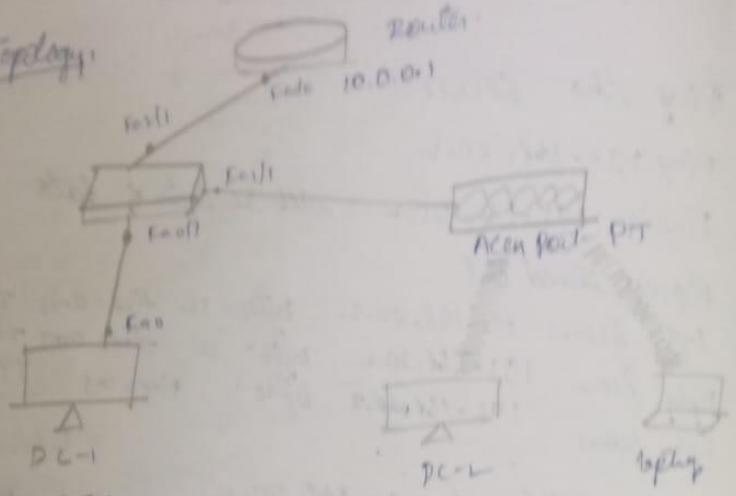
Experiment 12

To construct a WLAN and make the nodes communicate wirelessly



Wireless LAN

Topology:



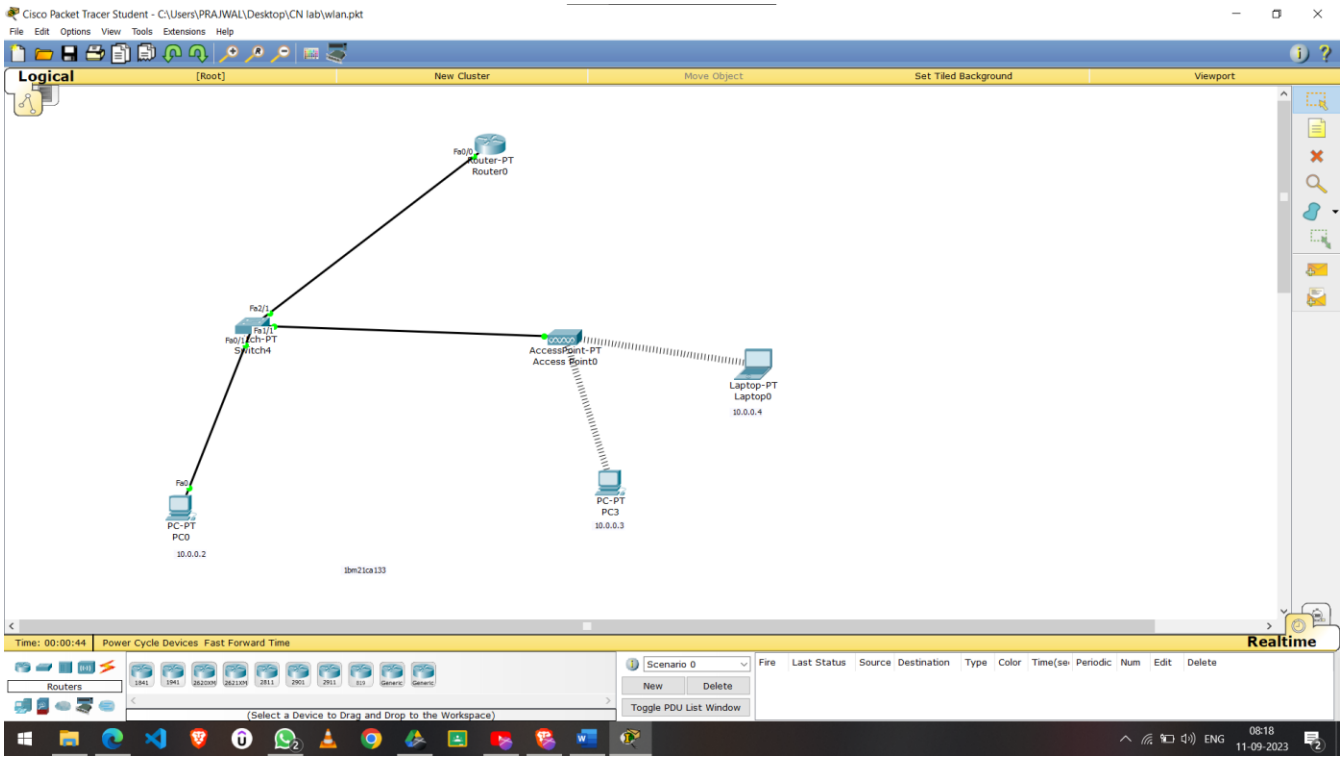
Procedure: Create the topology
Connect PC to switch & Router

→ Do normal route / PC configuration

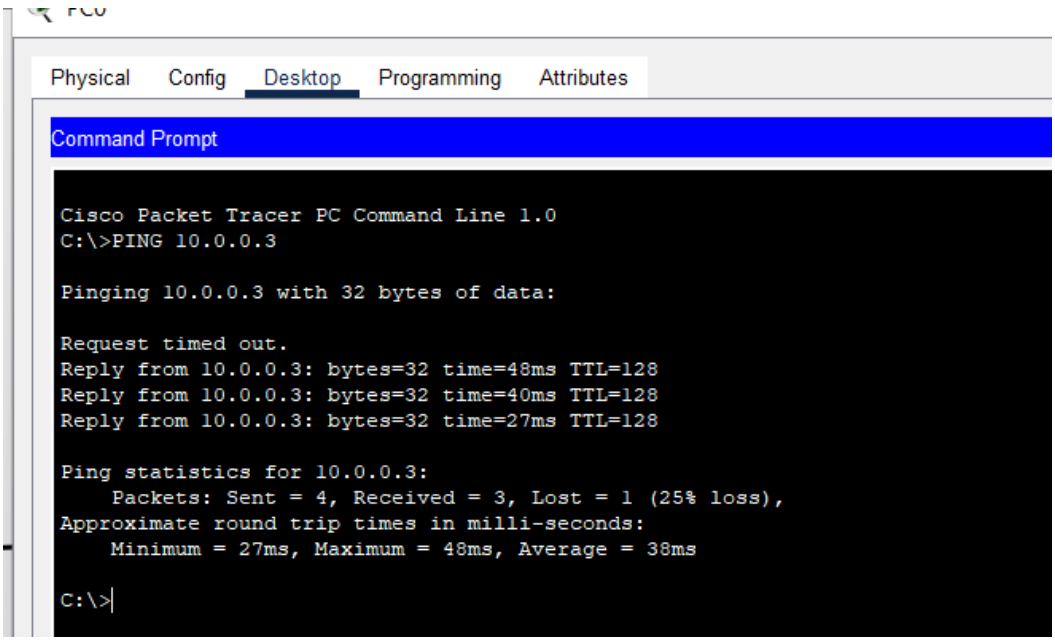
In Acess point
go to Config → Port-1
give SSID
Set it to WEP
Set a Hex Key - 1234567890.

In PC-C & Laptop 2.
→ select the port & Drag & Drop to
PC-HOST-NM-1AM.
→ Drag WMP300N & Drop at port.

Topology:

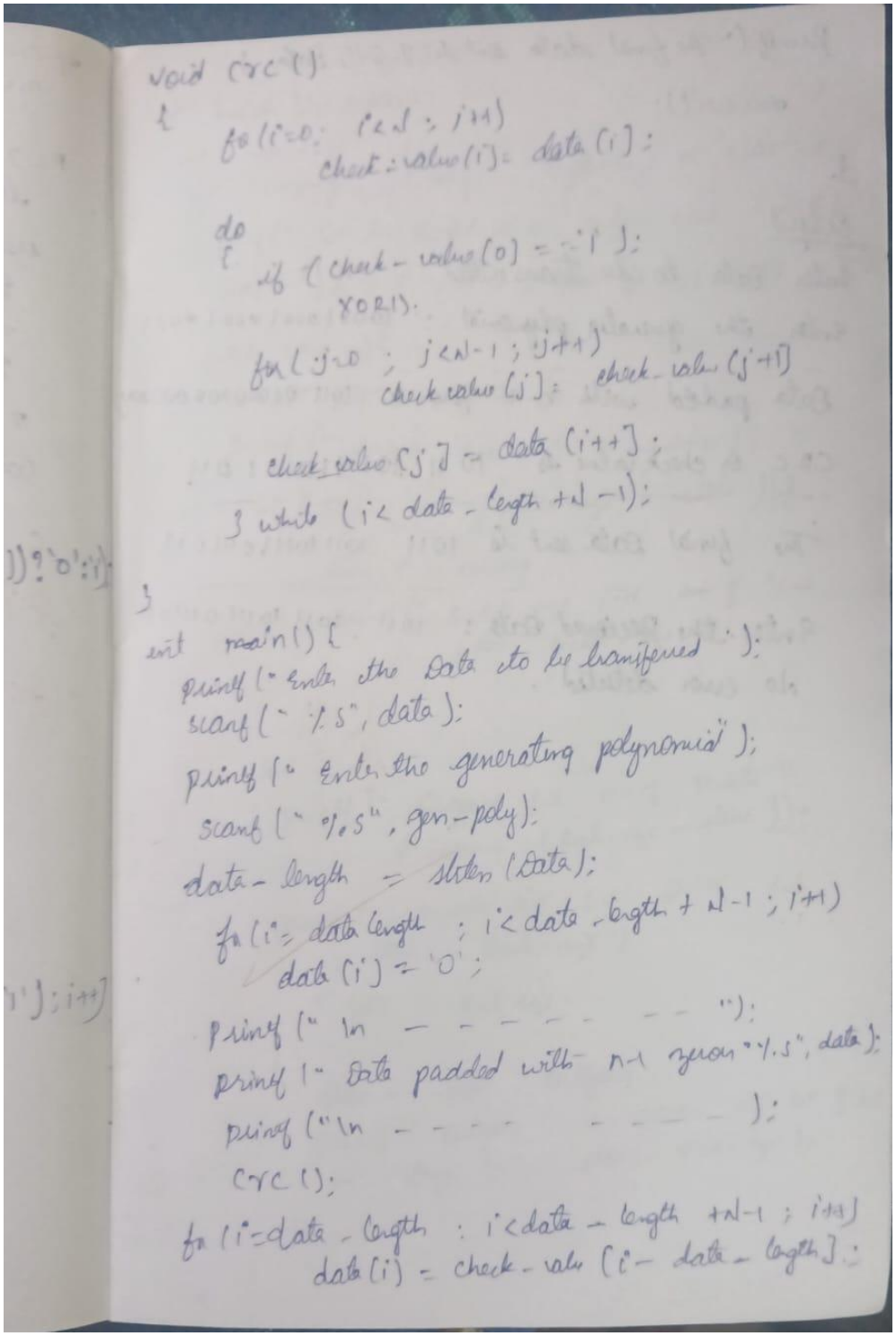


Output:

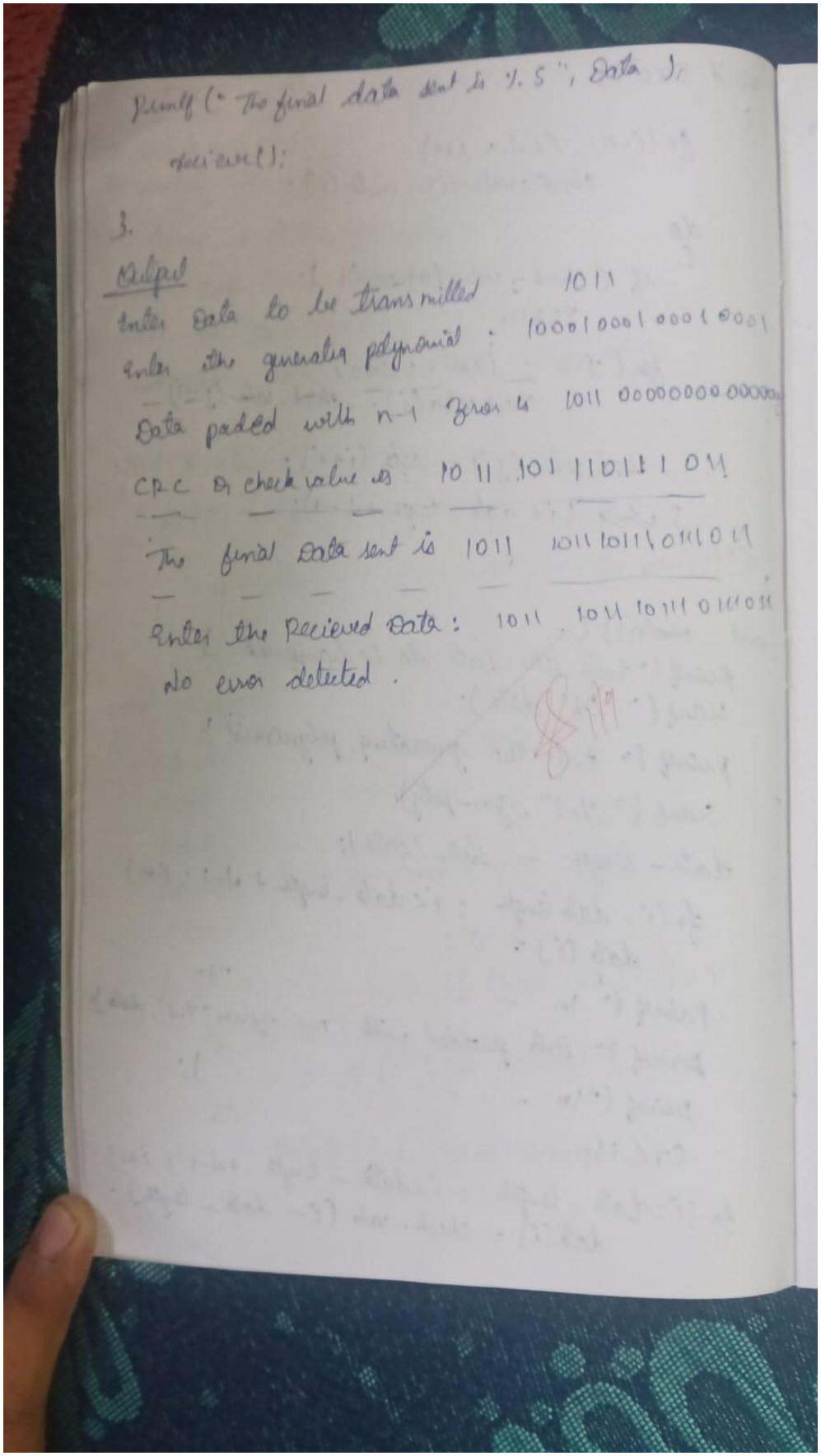


Experiment 13

Write a program for error detecting code using CRCCCITT (16-bits).



```
void crc()
{
    for(i=0; i<n; i++)
        check_value[i] = data[i];

    do
    {
        if(check_value[0] == '1');
            XOR1();
        for(j=0; j<n-1; j++)
            check_value[j] = check_value[j+1];

        check_value[j] = data[i++];
    } while(i<data_length+n-1);

}
int main(){
    printf("Enter the Data to be transferred");
    scanf("%s", data);
    printf("Enter the generating polynomial");
    scanf("%s", gen_poly);
    data_length = strlen(data);
    for(i=data_length; i<data_length+n-1; i++)
        data[i] = '0';
    printf("\n ------------");
    printf("Data padded with n-1 zeros : %s", data);
    printf("\n ------------");
    crc();
    for(i=data_length; i<data_length+n-1; i++)
        data[i] = check_value[i-data_length];
```



printf("The final data sent is %s", Data);
receive();

3.

Output

Enter Data to be transmitted : 1011

Enter the generator polynomial : 10001000100010001

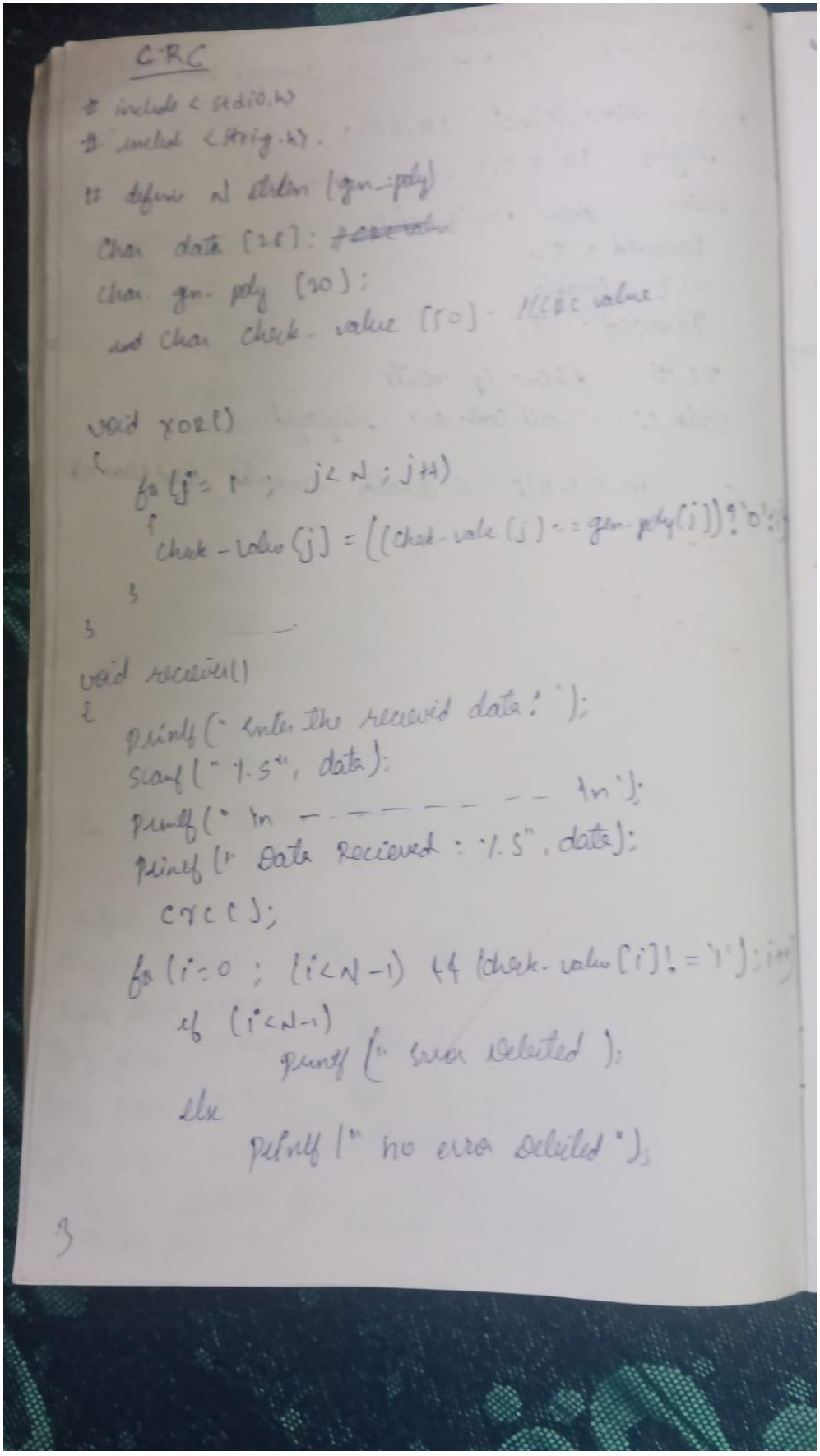
Data padded with n-1 zeros is 1011 000000000000000

CRC or check value is 10 11 101 1101 1 011

The final data sent is 1011 1011101101011

Enter the recieved data : 1011 1011101101011

No error detected.



CRC

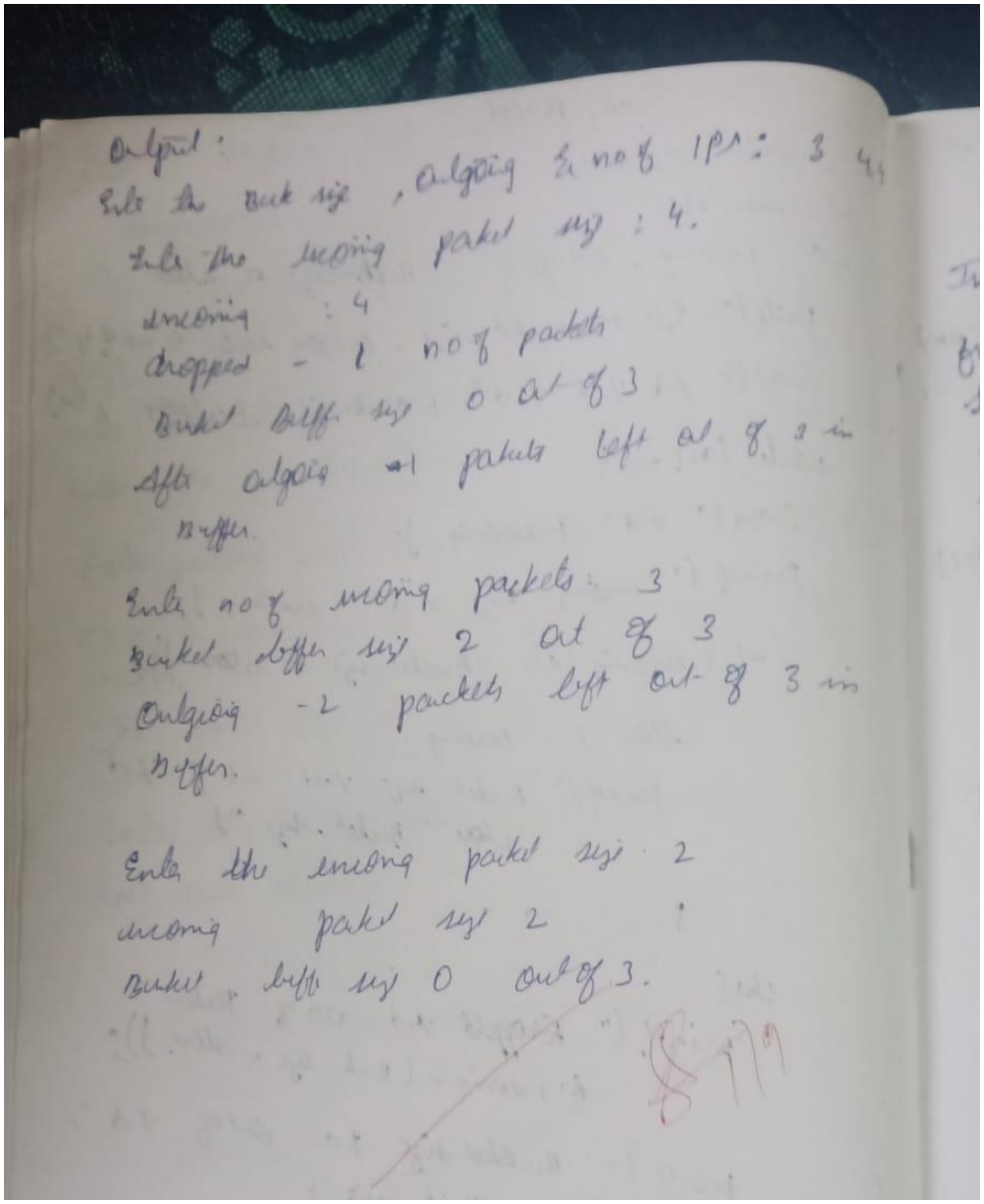
```
#include <stdio.h>
#include <string.h>
#define N strlen(gen_poly)
char data[28];
char gen_poly[20];
int data_length, i, j;
void XOR()
{
    for(j = 1; j < N; j++)
    check_value[j] = ((check_value[j] == gen_poly[j])?'0':'1');
}
void receive()
{
    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n-----------\n");
    printf("Data Recieved : %s", data);
    crc();
    for(i=0; (i<N-1) && (check_value[i]!='1'); i++);
    if (i<N-1)
        printf("Error Detected");
    else
        printf("No error detected");
}
```

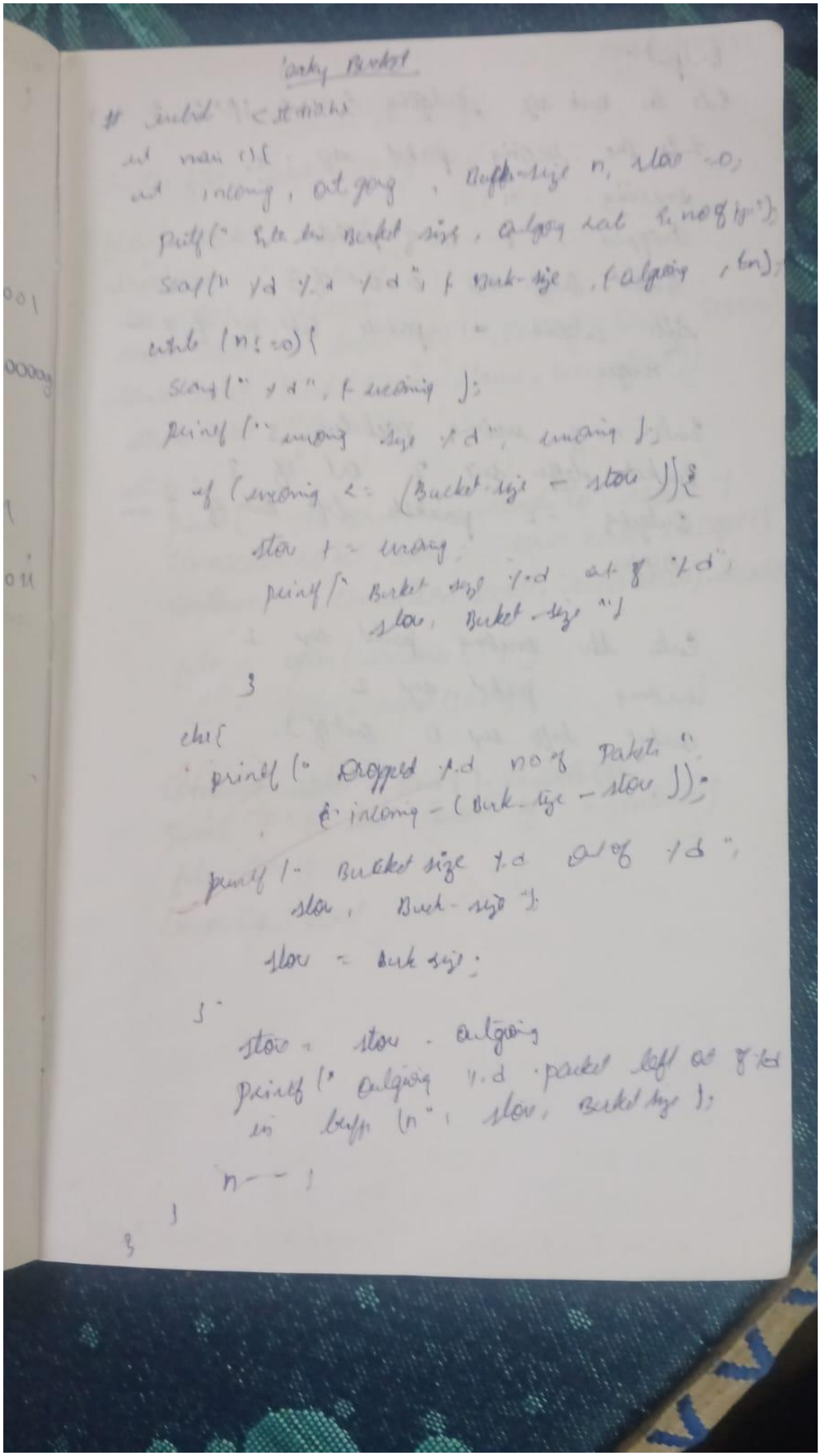
Output:

```
Enter the frame bits:1011
Message after appending 16 zeros:10110000000000000000
generator:10001000000100001


quotient:1011
transmitted frame:10111011000101101011
Enter transmitted freme:10111011000101101011
CRC checking


last remainder:0000000000000000

Received freme is correct
```

Experiment 14

Write a program for congestion control using Leaky bucket algorithm.





Leaky Bucket

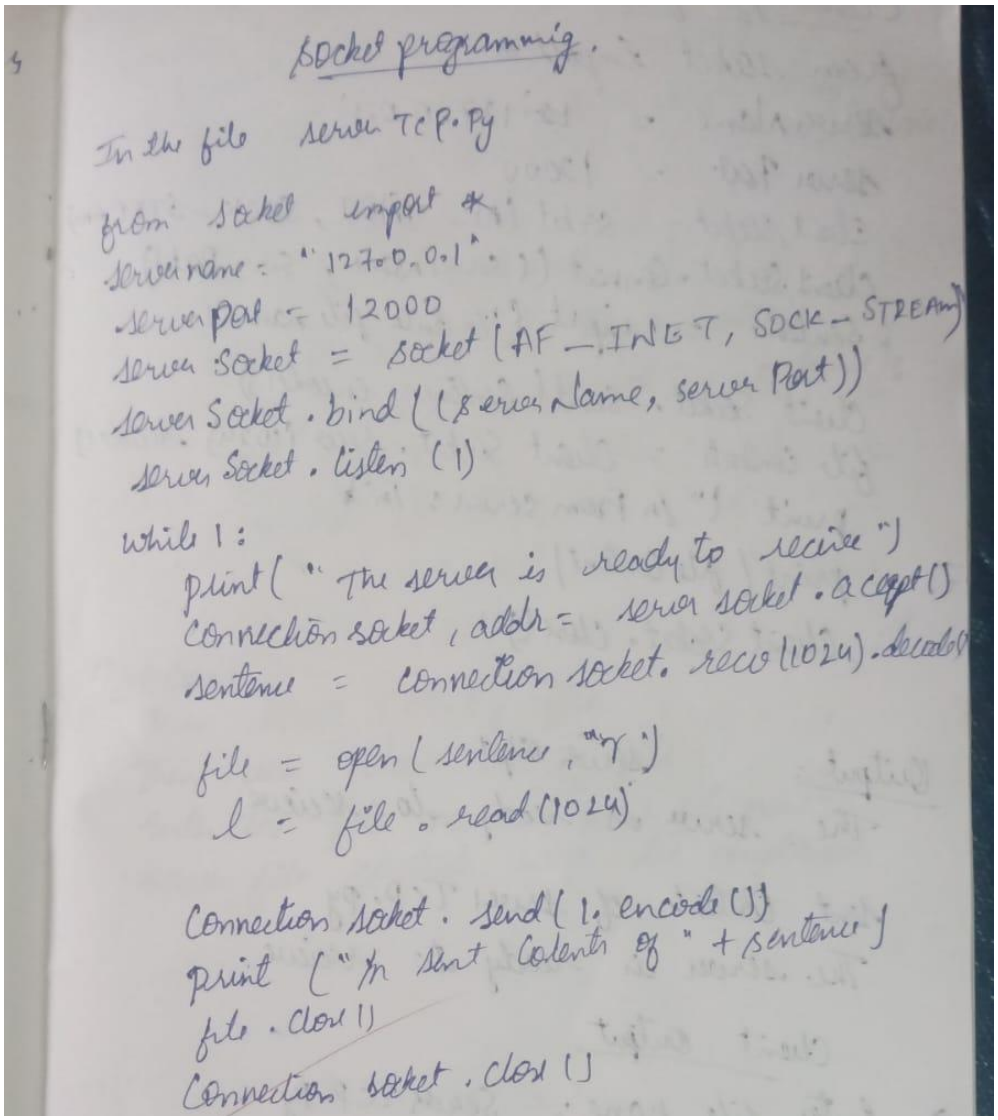
```
#include<stdio.h>
int main(){
int incoming, outgoing, buffer_size, n, store = 0;
printf("Enter bucket size, outgoing rate and no of inputs");
scanf("%d %d %d", &buffer_size, &outgoing, &n);

while(n!=0){
  scanf("%d", &incoming);
  printf("\n incoming size = %d", incoming);
  if(incoming <= (buffer_size - store)){
      store += incoming;
      printf("Buffer size %d out of %d",
             store, buffer_size);
  }
  else{
  printf("Dropped %d no of Packets",
          incoming - (buffer_size - store));
  printf("Buffer size %d out of %d",
          store, buffer_size);
      store = buffer_size;
  }
     store = store - outgoing
     printf("After outgoing %d packet left out of %d
     in buffer\n", store, buffer_size);
     n--;
  }
}
```

Output:

```
Enter bucket size, outgoing rate and no of inputs: 10 10 2
Enter the incoming packet size : 30
Incoming packet size 30
Dropped 20 no of packets
Bucket buffer size 0 out of 10
After outgoing 0 packets left out of 10 in buffer
Enter the incoming packet size : 10
Incoming packet size 10
Bucket buffer size 10 out of 10
After outgoing 0 packets left out of 10 in buffer
```

Experiment 15

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



Socket programming.

In the file server TCP.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 recive")
    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()
```

Client.Py file

```
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: server otput

The server is ready to recieve

sent contents of server TCP.Py

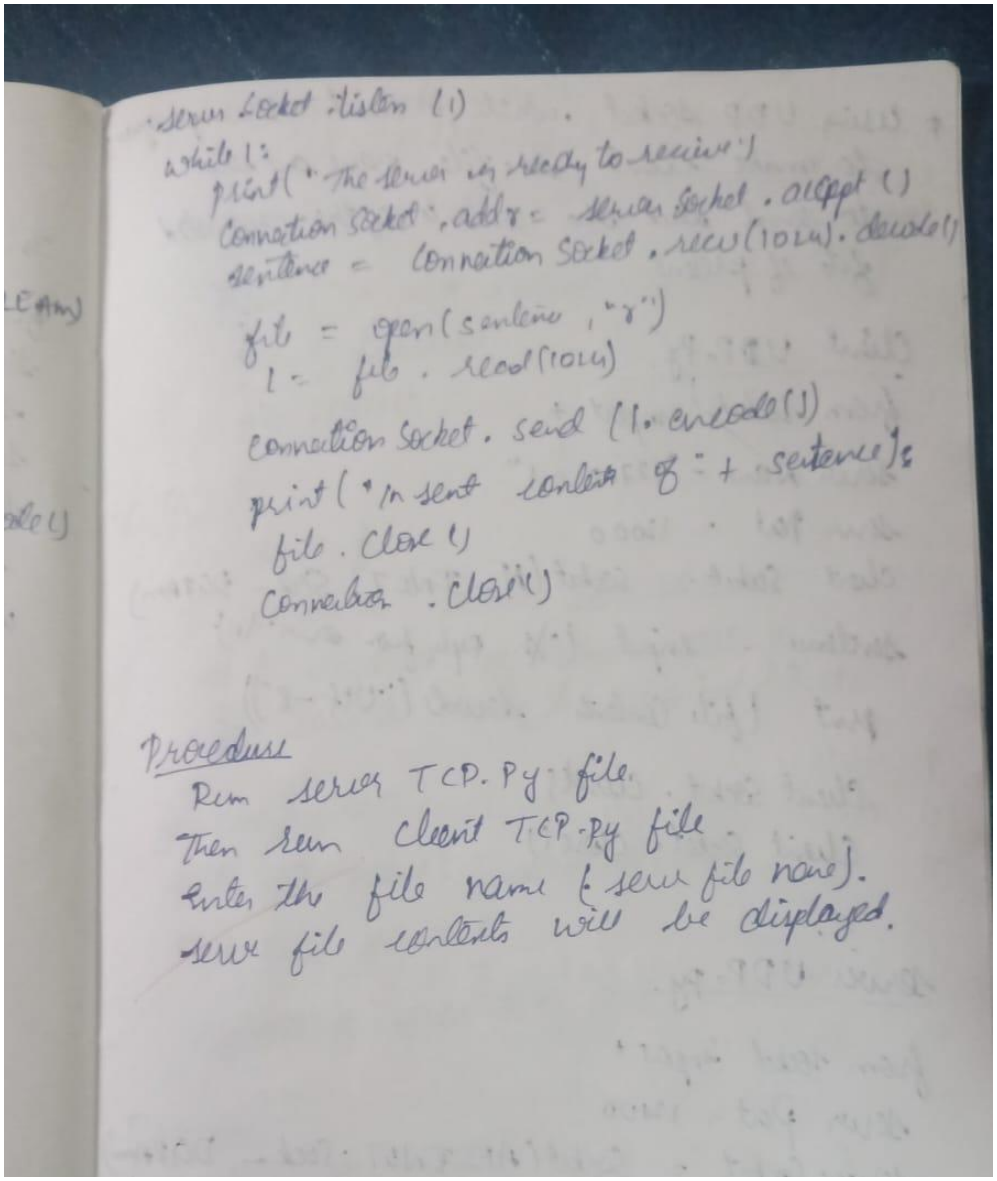
The server is ready to recieve.

Client Output

⇒ Enter file name: Server TCP.Py

from Server:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
```

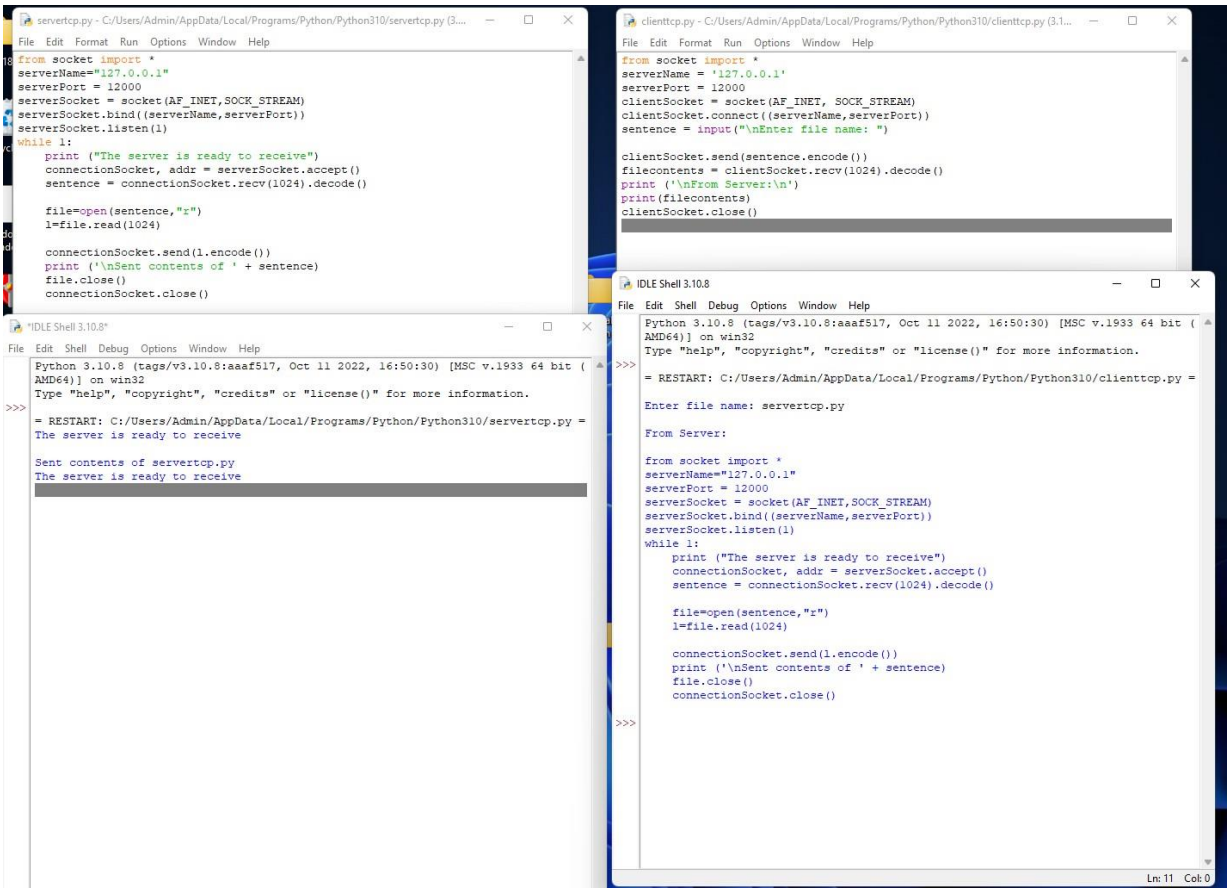


```
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print("\n sent contents of " + sentence)
    file.close()
    connection.close()
```

Procedure

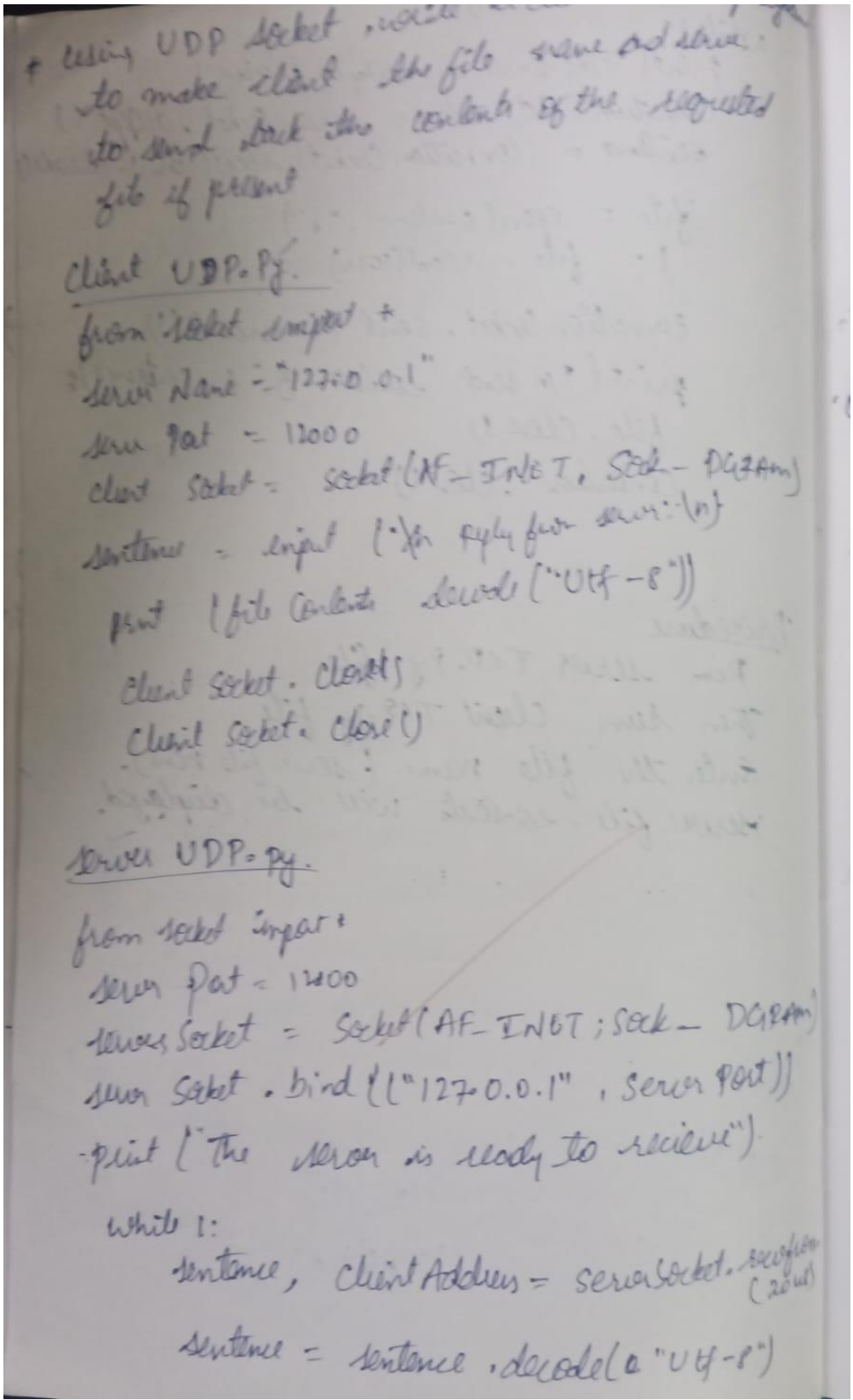
Run server TCP.py file.
Then run client TCP.py file.
Enter the file name (server file name).
Server file contents will be displayed.

Output:


servertcp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/servertcp.py (3....
File Edit Format Run Options Window Help
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,"r")
l=file.read(1024)
connectionSocket.send(l.encode())
print ('\nSent contents of ' + sentence)
file.close()
connectionSocket.close()
clienttcp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/clienttcp.py (3.1...
File Edit Format Run Options Window Help
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/servertcp.py =
The server is ready to receive
Sent contents of servertcp.py
The server is ready to receive
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/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 ('\nSent contents of ' + sentence)
file.close()
connectionSocket.close()
>>>
Ln: 11 Col: 0

Experiment 16

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



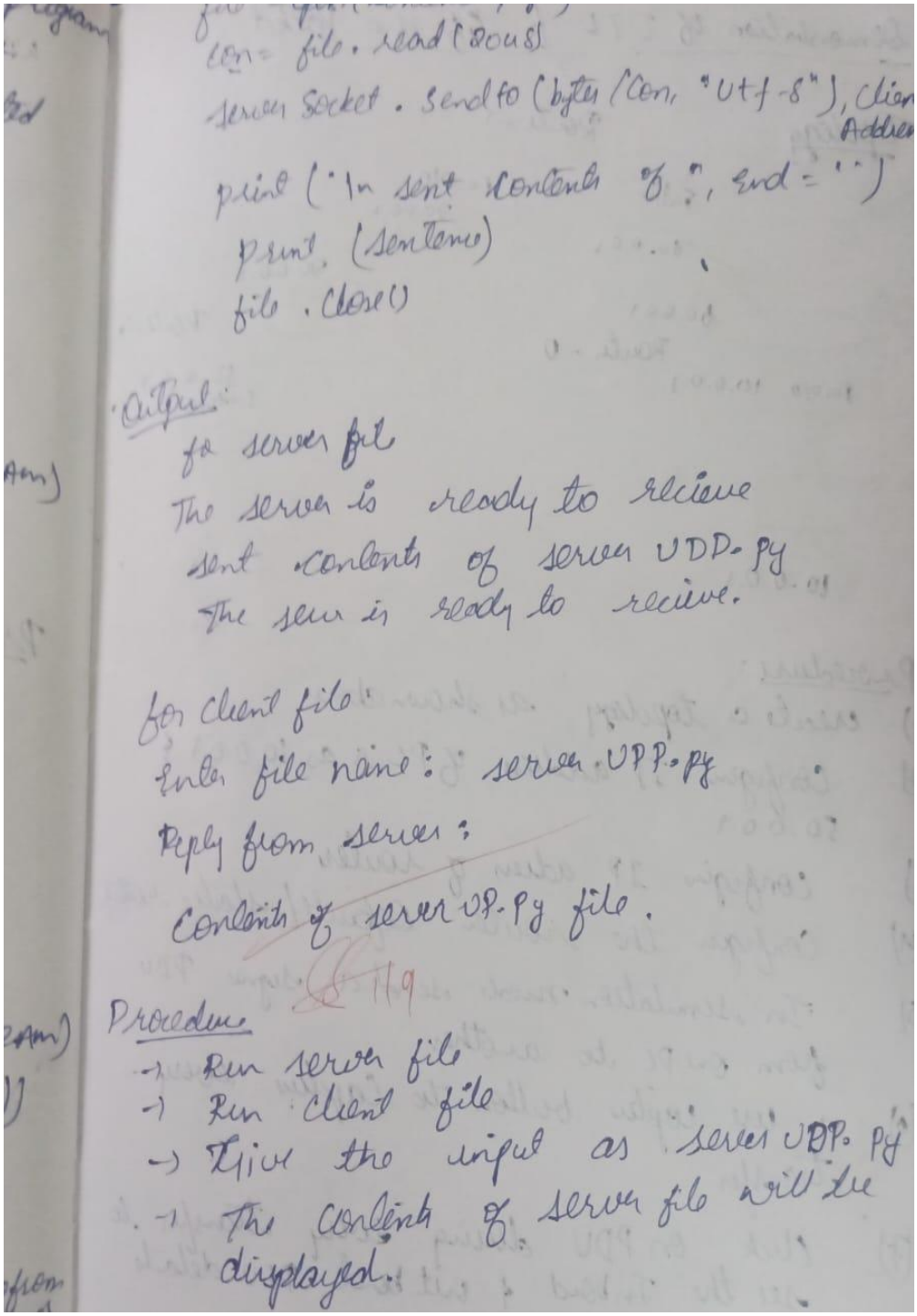
* using UDP socket, write a program to make client send the file name and server to send back the contents of the requested file if present

Client UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Input lowercase sentence:\n")
print("From Server:\n")
print(filecontents.decode("UTF-8"))
clientSocket.close()
clientSocket.close()
```

Server UDP.py

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



con = file.read(2048)
serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
print("In sent contents of :", end = ' ')
print(sentence)
file.close()

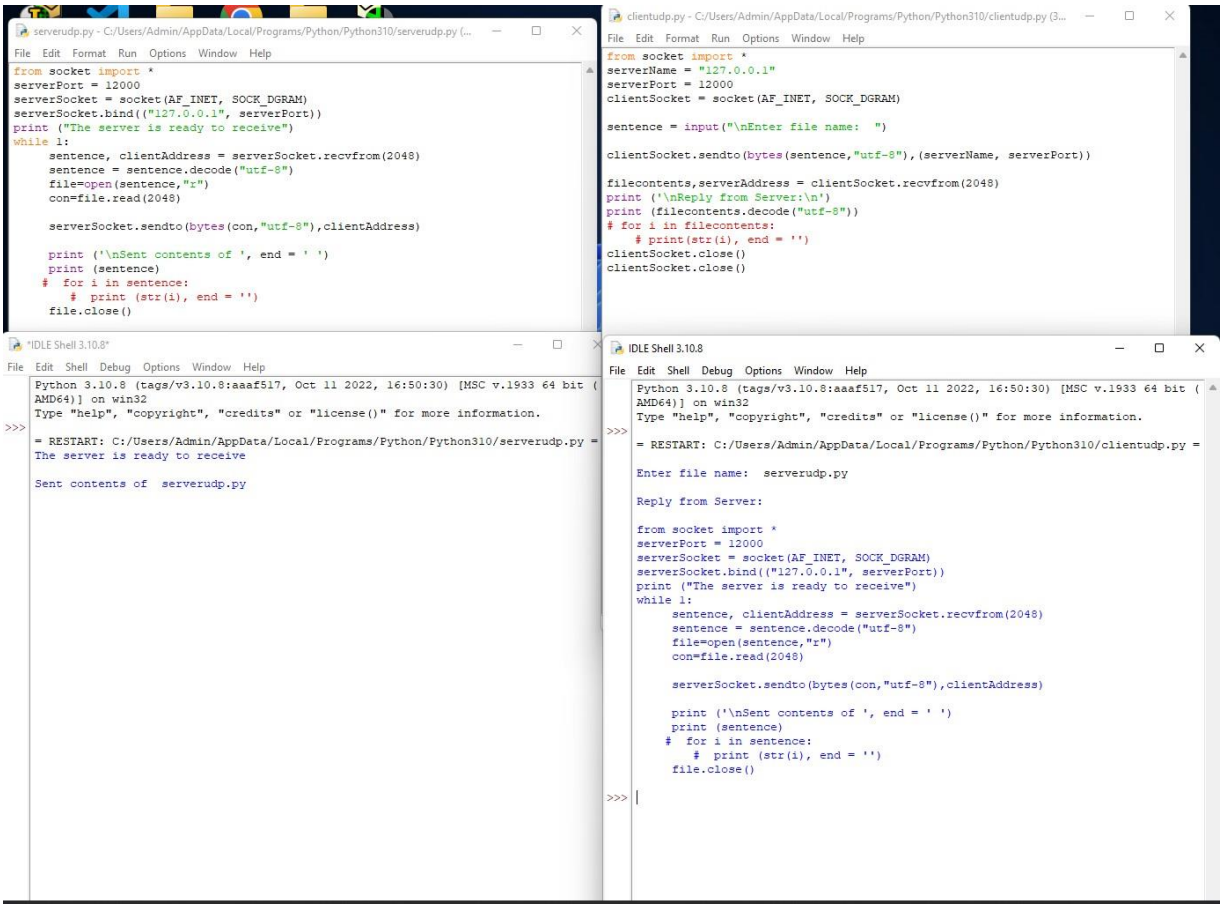
Output:

for server file
The server is ready to recieve
sent contents of server UDP.py
The ser is ready to recive.

for client file
Enter file name: server.UPP.py
Reply from server:
Contents of server UP.py file.

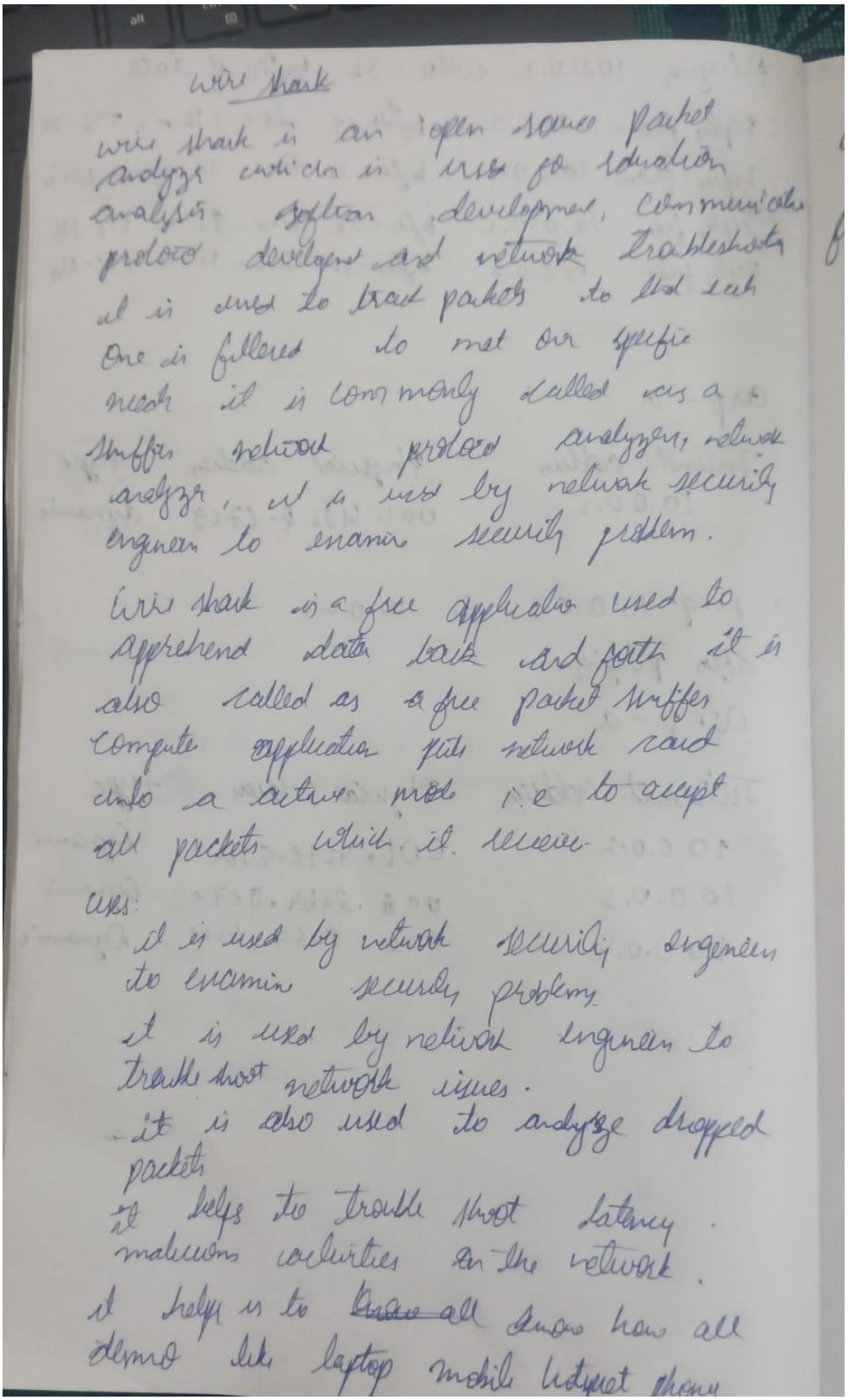
Procedure
-> Run server file
-> Run client file
-> Give the input as server UDP.py
-> The content of server file will be displayed.

Output:


serverudp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverudp.py (...
File Edit Format Run Options Window Help
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
clientudp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientudp.py (3...
File Edit Format Run Options Window Help
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode("utf-8"))
# for i in filecontents:
# print(str(i), end = '')
clientSocket.close()
clientSocket.close()
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverudp.py =
The server is ready to receive
Sent contents of serverudp.py
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/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))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
>>>

Experiment 17

Tool Exploration -Wireshark



Wireshark

wireshark is an open source packet analyzer which is used for education analysis software development, communication protocol development and network troubleshooting it is used to track packets so that each one is filtered to meet our specific needs it is commonly called as a sniffer network protocol analyzer, network analyzer. it is used by network security engineers to examine security problems.

wireshark is a free application used to apprehend data back and forth it is also called as a free packet sniffer computer application puts network card into a active mode i.e to accept all packets which it receives

uses:

- it is used by network security engineers to examine security problems
- it is used by network engineers to trouble shoot network issues.
- it is also used to analyze dropped packets
- it helps to trouble shoot latency.
- malicious activities on the network.
- it helps us to know all about how all devices like laptop mobile hotspot phone

desktop switch router communication
phone in a local network or the rest of the world

functionability of wireshark
it is similar to TCP dump in networking
it is a graphic end, sort and filtering
function it also monitors the unicast
traffic which is not sent to network
is a method to monitor the network
traffic when it is enabled switch sends
copies of all network packets present at one
port to another.

features:
it is a multi ~~programming~~ platform software i.e
it can run on the linux, window etc
→ it is standard 3 pane packet Browser
→ it perform deep inspection of hut on protocol
→ it even has standard features often which
makes ease to user to view the Data
→ it can ~~replace~~ .
→ it is useful in IP analysis.