# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**“JnanaSangama”, Belgaum -590014, Karnataka.**

**LAB REPORT**
**on**

# COURSE TITLE

***Submitted by***

**NAGARAJ SUNAGAR (1BM20CS090)**

***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**
**October-2022 to Feb-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 **NAGARAJ SUNAGAR (1BM20CS090),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **COMPUTER NETWORKS- (20CS5PCCON)** work prescribed for the said degree.

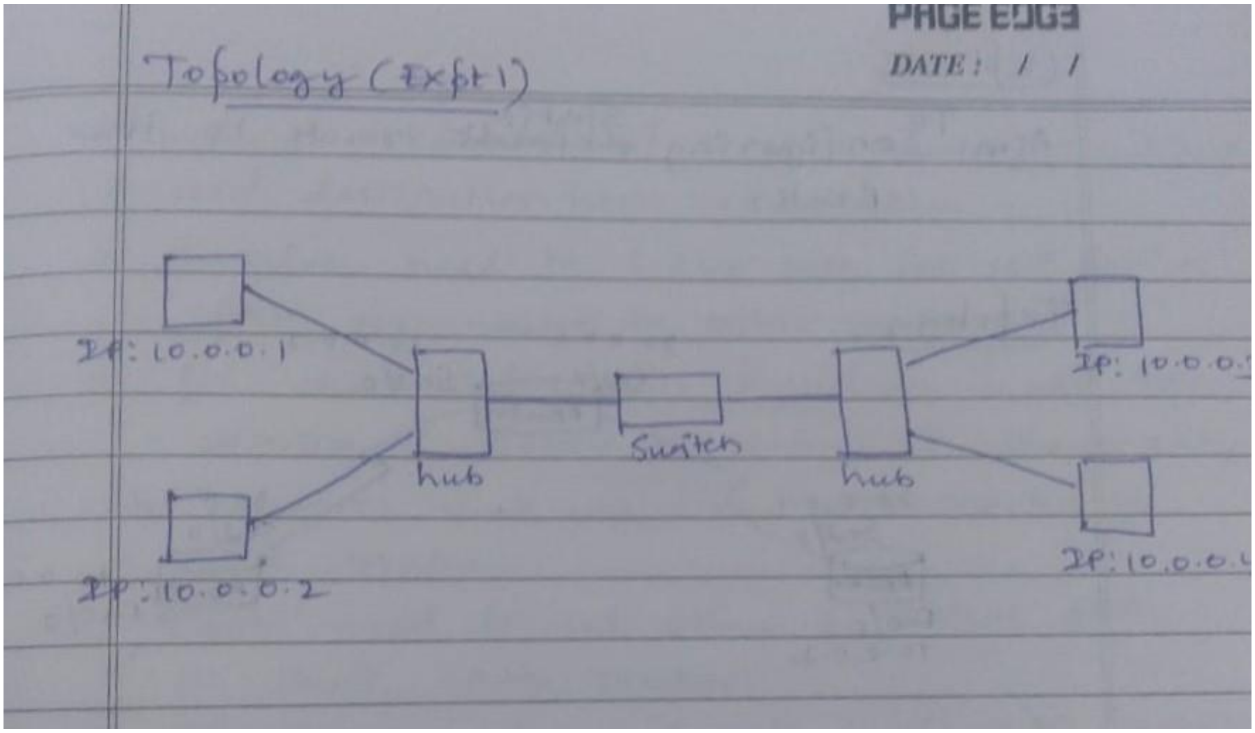
| | |
|---|---|
| **Prof. Lohit J J** | **Dr. Jyothi S Nayak** |
| Assistant Professor | Professor and Head |
| Department of CSE | Department of CSE |
| BMSCE, Bengaluru | BMSCE, Bengaluru |

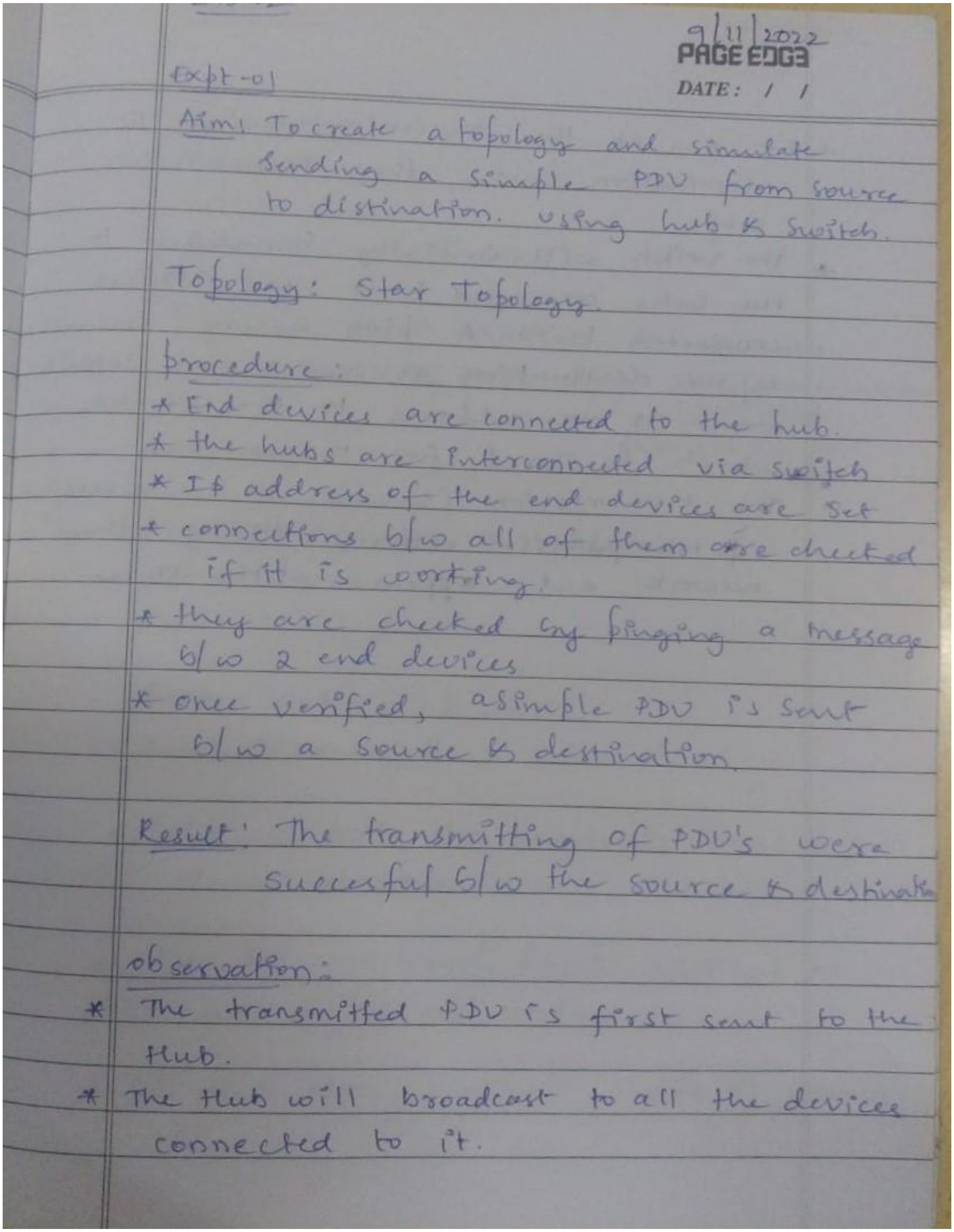
`

# Index

# Experiment No 1

Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.





9/11/2022

PAGE EDGE

DATE: / /

Expt-01

Aim: To create a topology and simulate Sending a Simple PDU from source to destination. using hub & Switch.

Topology: Star Topology.

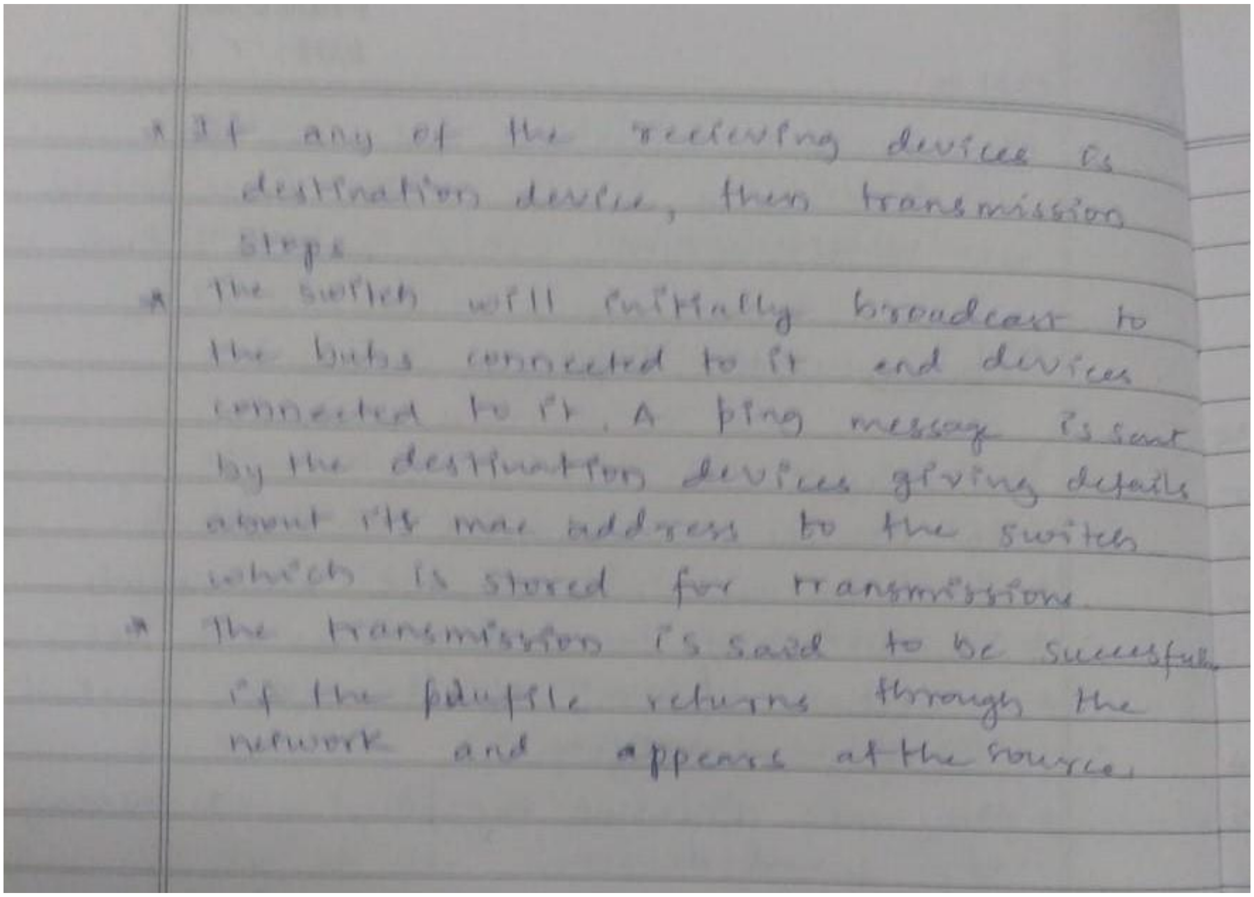
procedure:

* End devices are connected to the hub.
* the hubs are interconnected via switch
* IP address of the end devices are set
* connections b/w all of them are checked if it is working.
* they are checked by pinging a message b/w 2 end devices
* once verified, a simple PDU is sent b/w a source & destination.

Result: The transmitting of PDU's were succesful b/w the source & destination

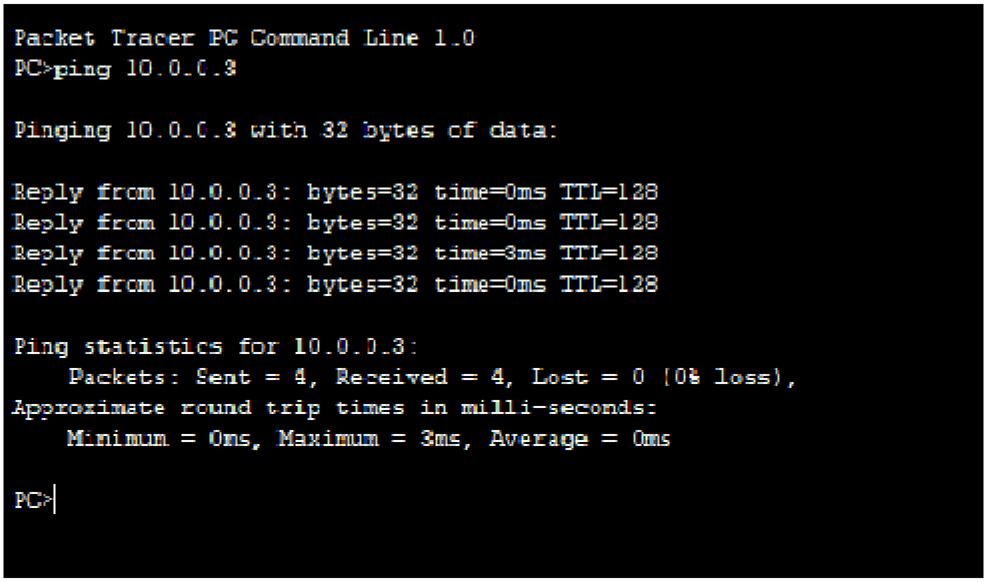
observation:

* The transmitted PDU is first sent to the Hub.
* The Hub will broadcast to all the devices connected to it.



* If any of the receiving devices is destination device, then transmission steps.
* The switch will initially broadcast to the hubs connected to it and devices connected to it. A ping message is sent by the destination devices giving details about its mac address to the switch which is stored for transmission.
* The transmission is said to be successful if the pdufile returns through the network and appears at the source.

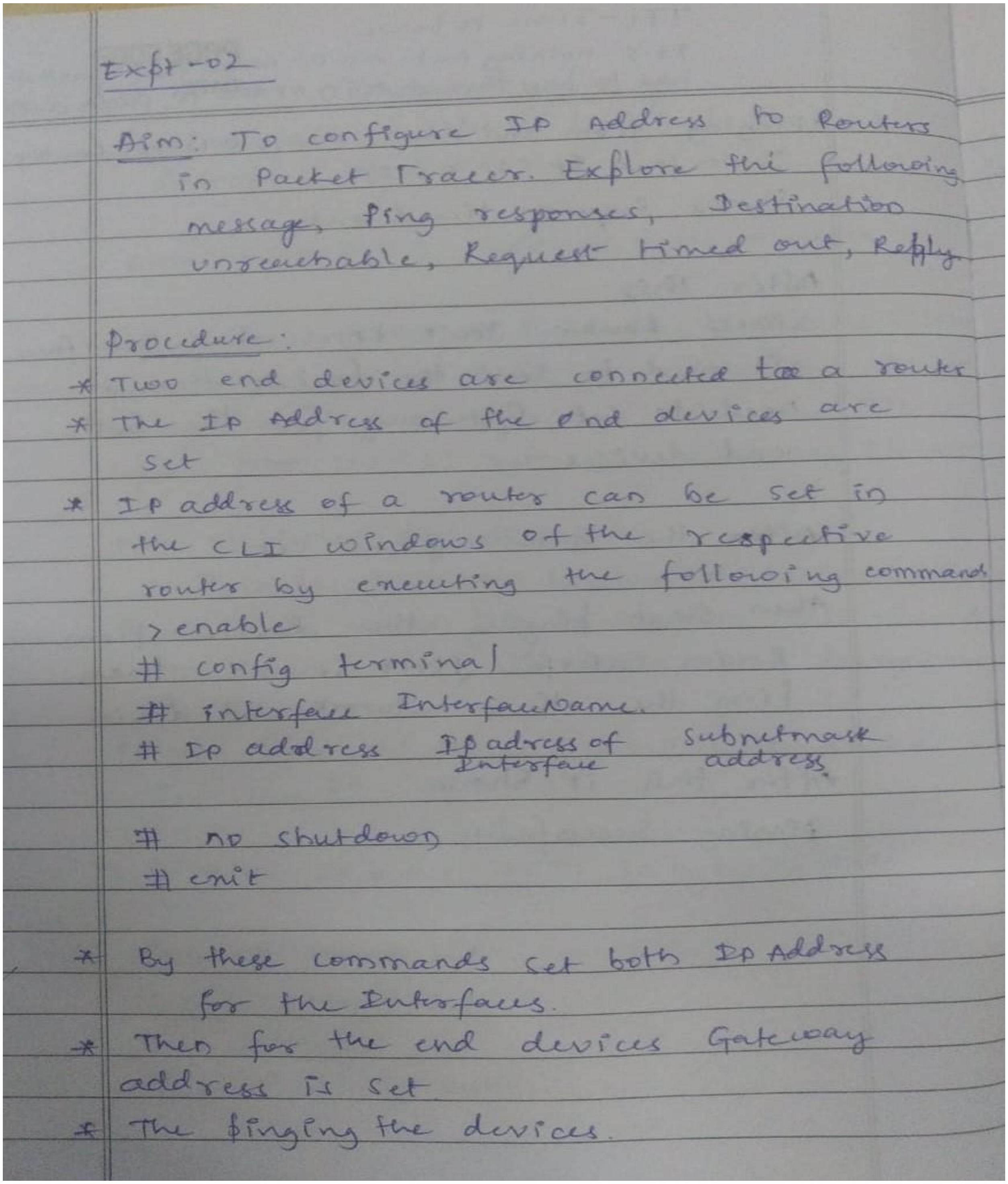
**OUTPUT:**



```
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=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 0ms

PC>
```

## Experiment 2

Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply



Expt-02

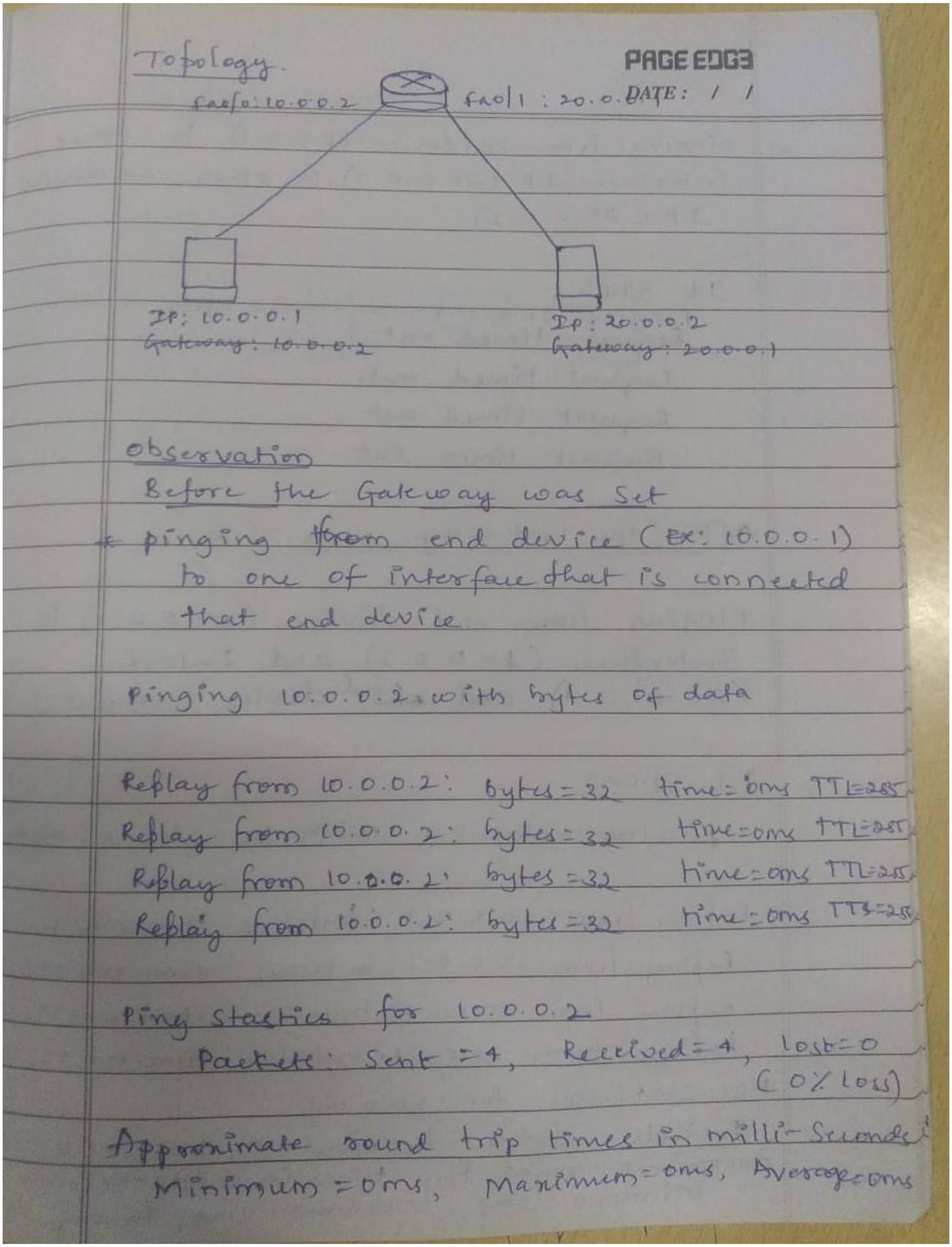
Aim: To configure IP Address to Routers in Packet Tracer. Explore the following message, Ping responses, Destination unreachable, Request timed out, Reply.

Procedure:

* Two end devices are connected too a router
* The IP Address of the end devices are set
* IP address of a router can be set in the CLI windows of the respective router by executing the following commands

```
> enable
# config terminal
# interface InterfaceName.
# IP address   IP adress of Interface   Subnetmask address.

# no shutdown
# exit
```

* By these commands set both IP Address for the Interfaces.
* Then for the end devices Gateway address is set
* The pinging the devices.



Topology.

fa0/0: 10.0.0.2 fa0/1 : 20.0.0.1

IP: 10.0.0.1
~~Gateway: 10.0.0.2~~

IP: 20.0.0.2
~~Gateway: 20.0.0.1~~

observation

Before the Gateway was Set

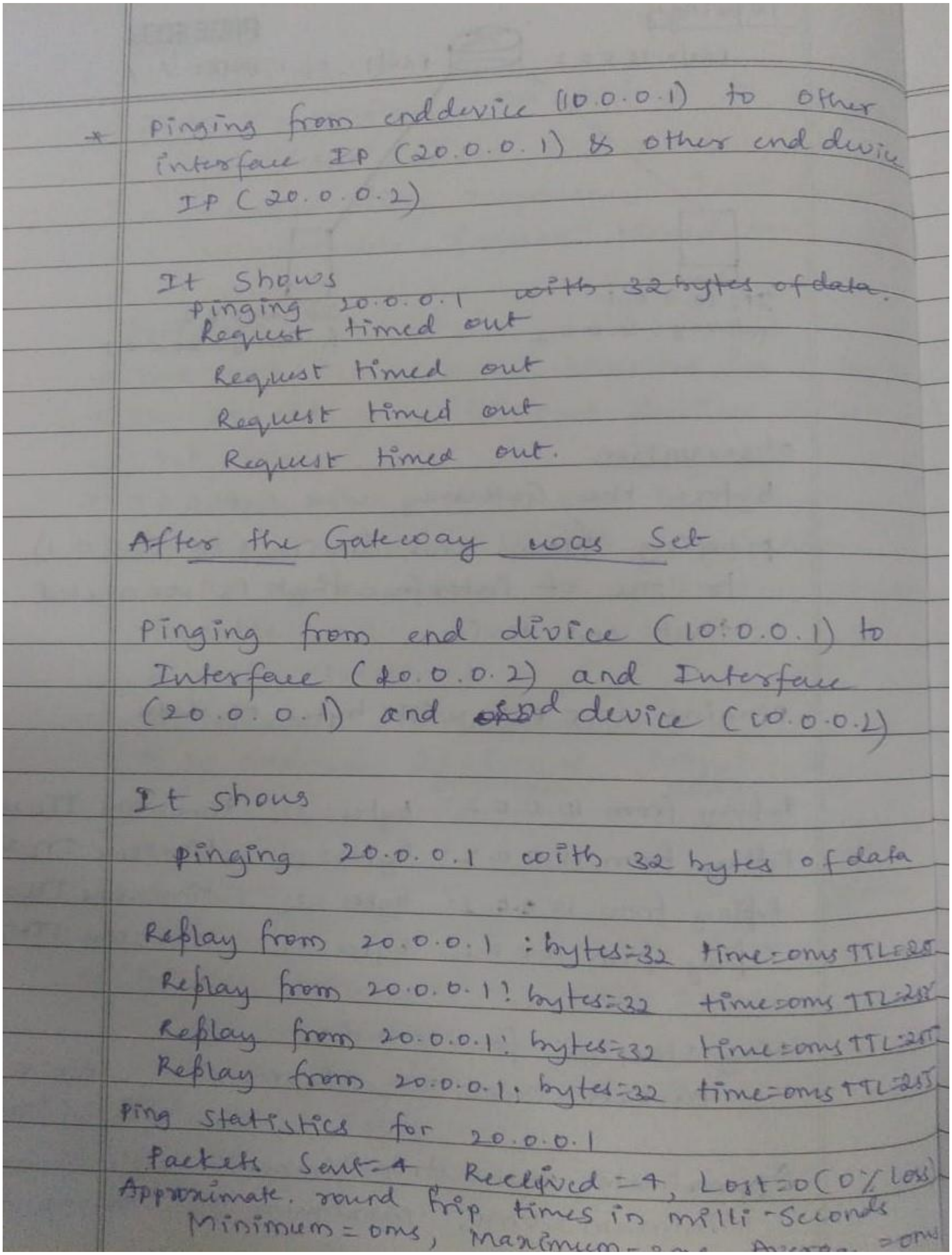
* pinging from end device (Ex: 10.0.0.1) to one of interface that is connected that end device

Pinging 10.0.0.2 with bytes of data

```
Replay from 10.0.0.2: bytes=32   time=0ms TTL=255
Replay from 10.0.0.2: bytes=32   time=0ms TTL=255
Replay from 10.0.0.2: bytes=32   time=0ms TTL=255
Replay from 10.0.0.2: bytes=32   time=0ms TTS=255
```

Ping stastics 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



* Pinging from enddevice (10.0.0.1) to other interface IP (20.0.0.1) & other end device IP (20.0.0.2)

It Shows

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

After the Gateway was Set

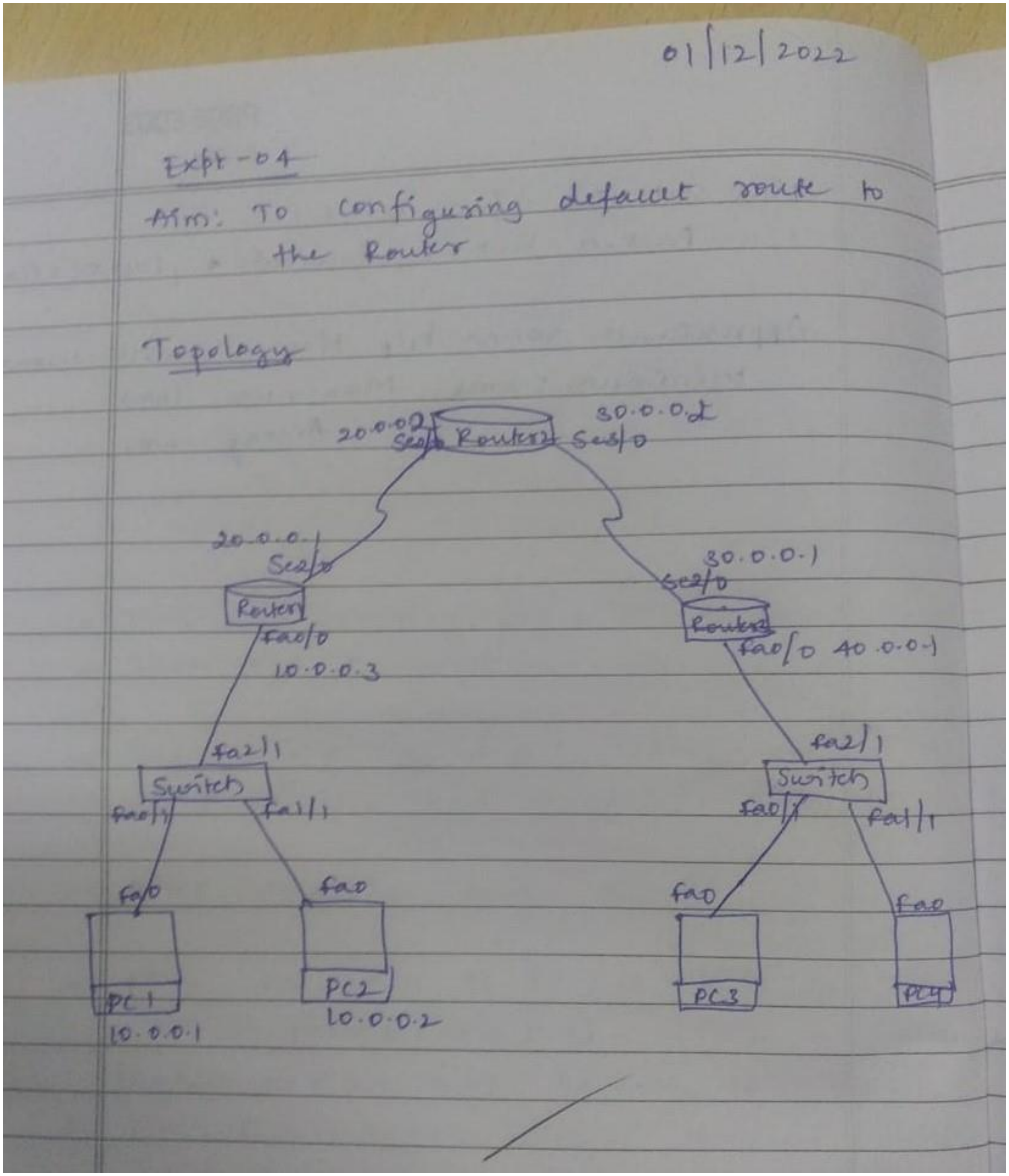
Pinging from end divice (10:0.0.1) to Interface (20.0.0.2) and Interface (20.0.0.1) and end device (10.0.0.2)

It shows

```
  pinging 20.0.0.1 with 32 bytes of data

Replay from 20.0.0.1 : bytes=32 time=oms TTL=255
 Replay from 20.0.0.1? bytes=32  time=oms TTL=255
 Replay from 20.0.0.1: bytes=32  time=oms TTL=255
 Replay from 20.0.0.1; bytes=32  time=oms TTL=255
Ping statistics for 20.0.0.1
 Packets Sent=4  Received =4, Lost=0 (0% loss)
Approximate. round trip times in milli-Seconds
    Minimum = oms, Maximum = oms, Average =oms
```

# Experiment 3

Configuring default route to the Router

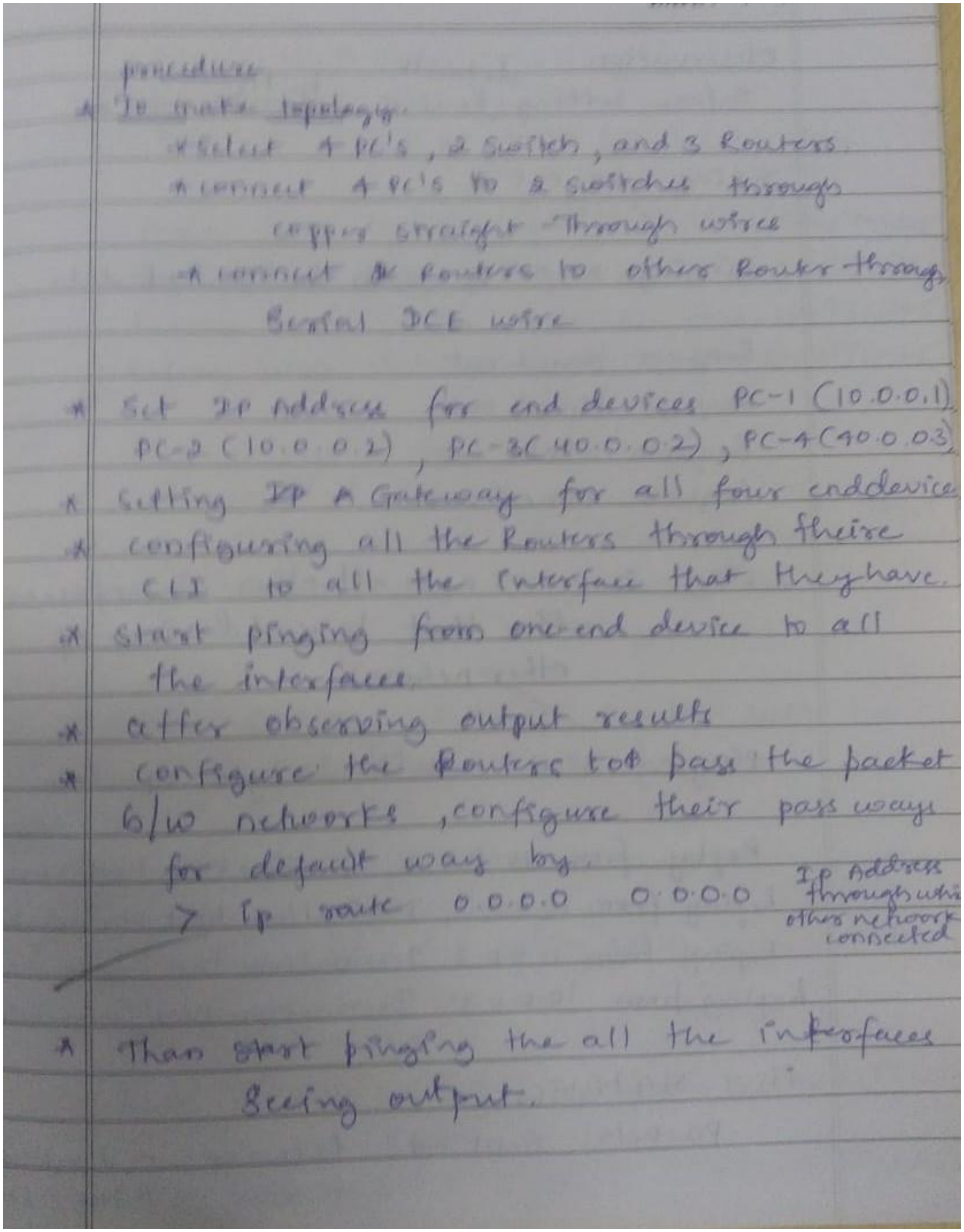


01/12/2022

Expt - 04

Aim: To configuring default route to the Router

Topology



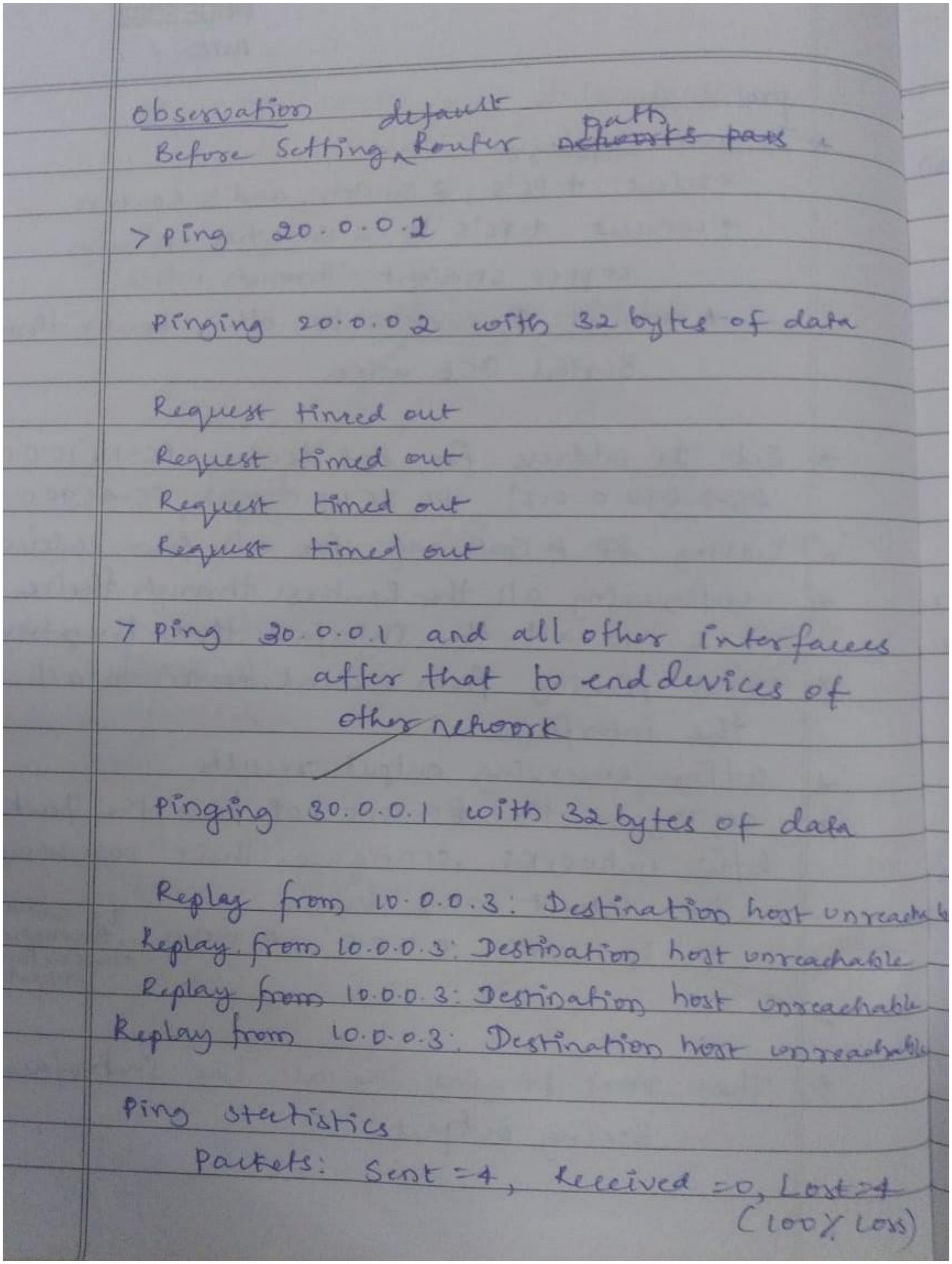
procedure

* To make topology
  * Select 4 PC's , 2 Switch , and 3 Routers.
  * connect 4 PC's to 2 switches through copper straight Through wire
  * connect ~~A~~ Routers to other Router through Serial DCE wire

* Set IP Address for end devices PC-1 (10.0.0.1), PC-2 (10.0.0.2), PC-3(40.0.0.2), PC-4(40.0.0.3)
* Setting IP & Gateway for all four enddevice
* configuring all the Routers through theire CLI to all the interface that theyhave.
* Start pinging from one end device to all the interfaces.
* after observing output results
* configure the Routers to~~o~~ pass the packet b/w networks , configure their pass ways for default way by.

  > Ip route 0.0.0.0 0.0.0.0 IP Address through which other network connected

* Than start pinging the all the interfaces Seeing output.

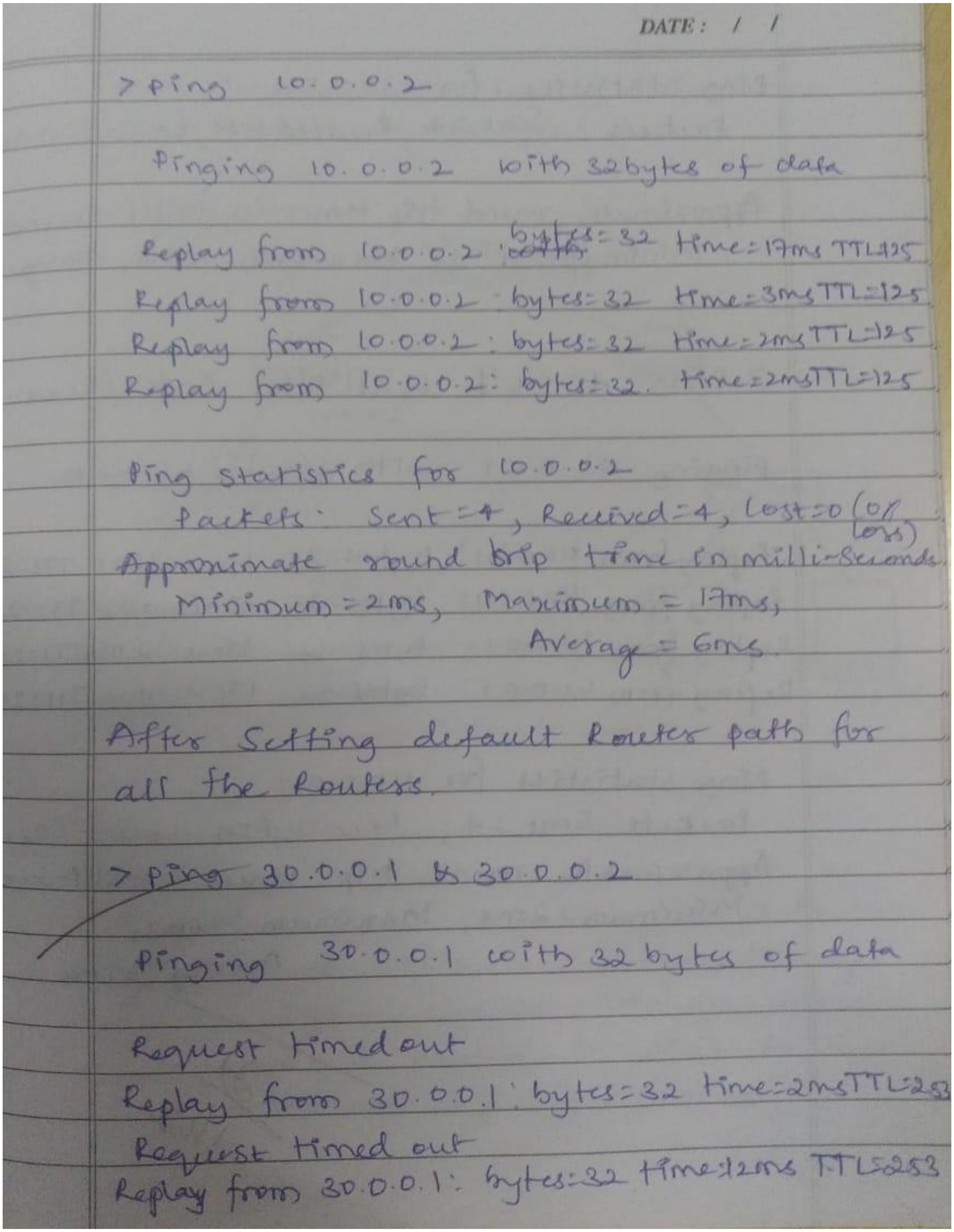


Observation
Before Setting default Router ~~networks pass~~ path.

```
> ping 20.0.0.2

pinging 20.0.0 2 with 32 bytes of data

Request timed out
Request timed out
Request timed out
Request timed out
```

> ping 20.0.0.1 and all other interfaces after that to end devices of other network

```
pinging 30.0.0.1 with 32 bytes of data

Replay from 10.0.0.3: Destination host unreachab
Replay from 10.0.0.3: Destination host unreachable
Replay from 10.0.0.3: Destination host unreachable
Replay from 10.0.0.3: Destination host unreachable

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



DATE : / /

> ping 10.0.0.2

Pinging 10.0.0.2 with 32bytes of data

Replay from 10.0.0.2 : bytes= 32 time=17ms TTL=125
Replay from 10.0.0.2 : bytes=32 time=3ms TTL=125
Replay from 10.0.0.2 : bytes=32 time=2ms TTL=125
Replay from 10.0.0.2: bytes=32. time=2ms TTL=125

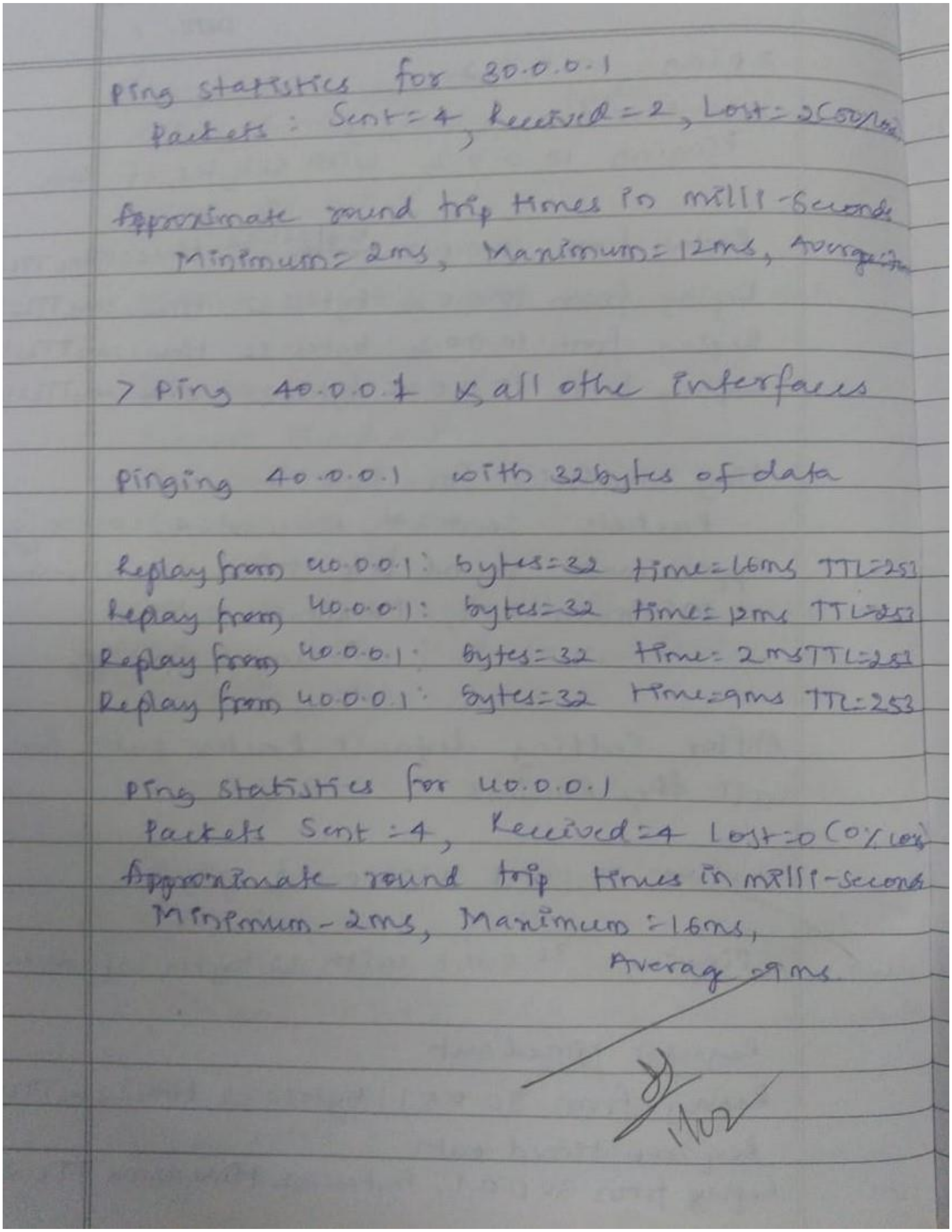
Ping statistics for 10.0.0.2
Packets: Sent =4, Received=4, Lost=0 (0% loss)
Approximate round trip time in milli-Seconds
Minimum=2ms, Maximum = 17ms,
Average = 6ms.

After Setting default Router path for all the Routers.

> ping 30.0.0.1 & 30.0.0.2

Pinging 30.0.0.1 with 32 bytes of data

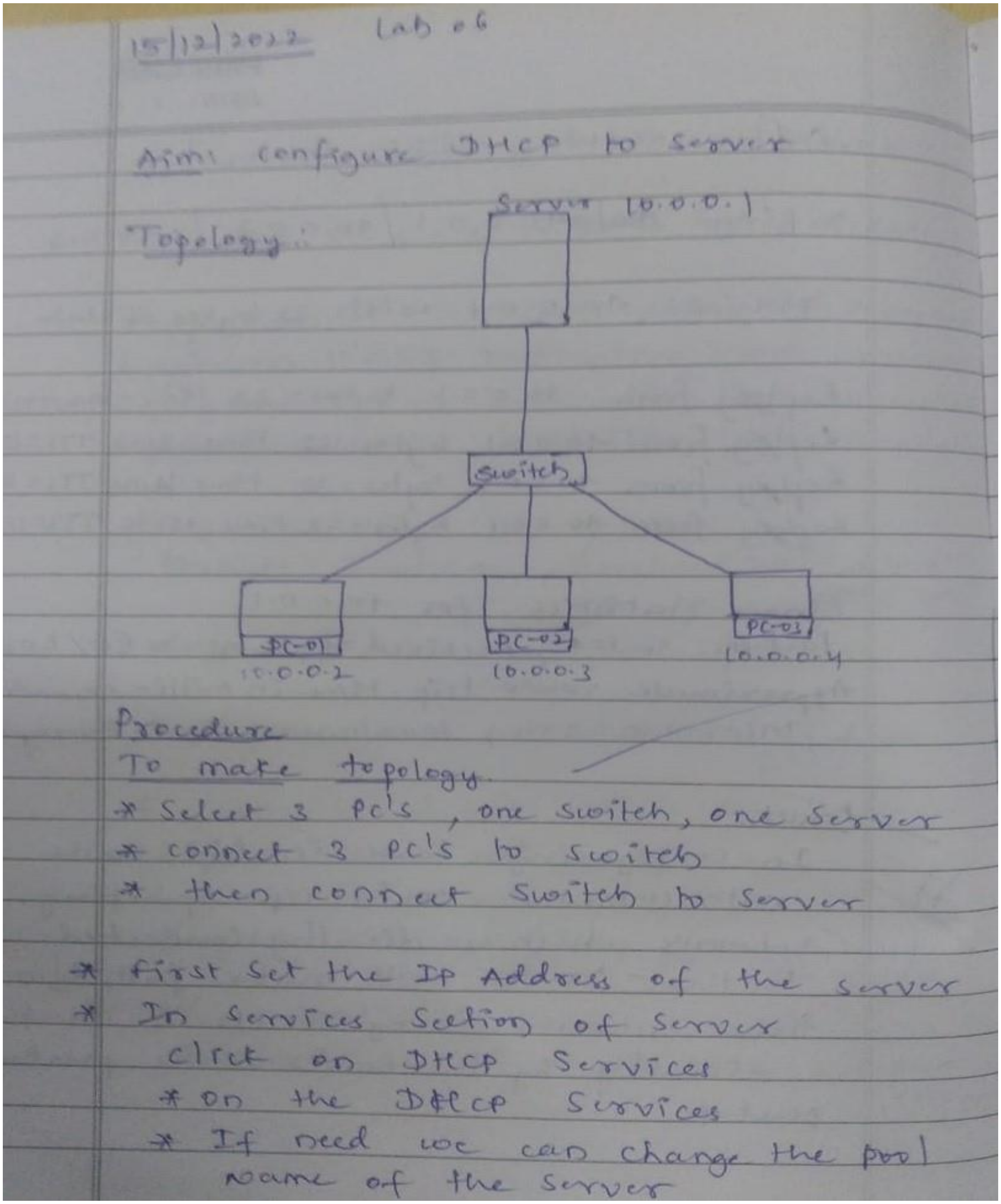
Request timed out
Replay from 30.0.0.1: bytes=32 time=2ms TTL=253
Request timed out
Replay from 30.0.0.1: bytes=32 time=2ms TTL=253



```
ping statistics for 30.0.0.1
   Packets : Sent=4, Received=2, Lost=2(50% lo

Approximate round trip times in milli-Seconds
     Minimum=2ms, Maximum=12ms, Average=7m


> Ping 40.0.0.1 & all othe Interfaces

Pinging 40.0.0.1 with 32bytes of data

Replay from 40.0.0.1: bytes=32 time=16ms TTL=253
Replay from 40.0.0.1: bytes=32 time=12ms TTL=253
Replay from 40.0.0.1: bytes=32 time=2ms TTL=253
Replay from 40.0.0.1: bytes=32 time=9ms TTL=253

Ping Statistics for 40.0.0.1
  Packets Sent =4, Received=4 Lost=0(0% loss
Approximate round trip times in milli-seconds
  Minimum-2ms, Maximum =16ms,
                              Average =9ms.
```

## Experiment 4

Configuring DHCP within a LAN in a packet Tracer


Server 10.0.0.1
switch
PC-01
10.0.0.2
PC-02
10.0.0.3
PC-03
10.0.0.4

15/12/2022 Lab 06

Aim: configure DHCP to server

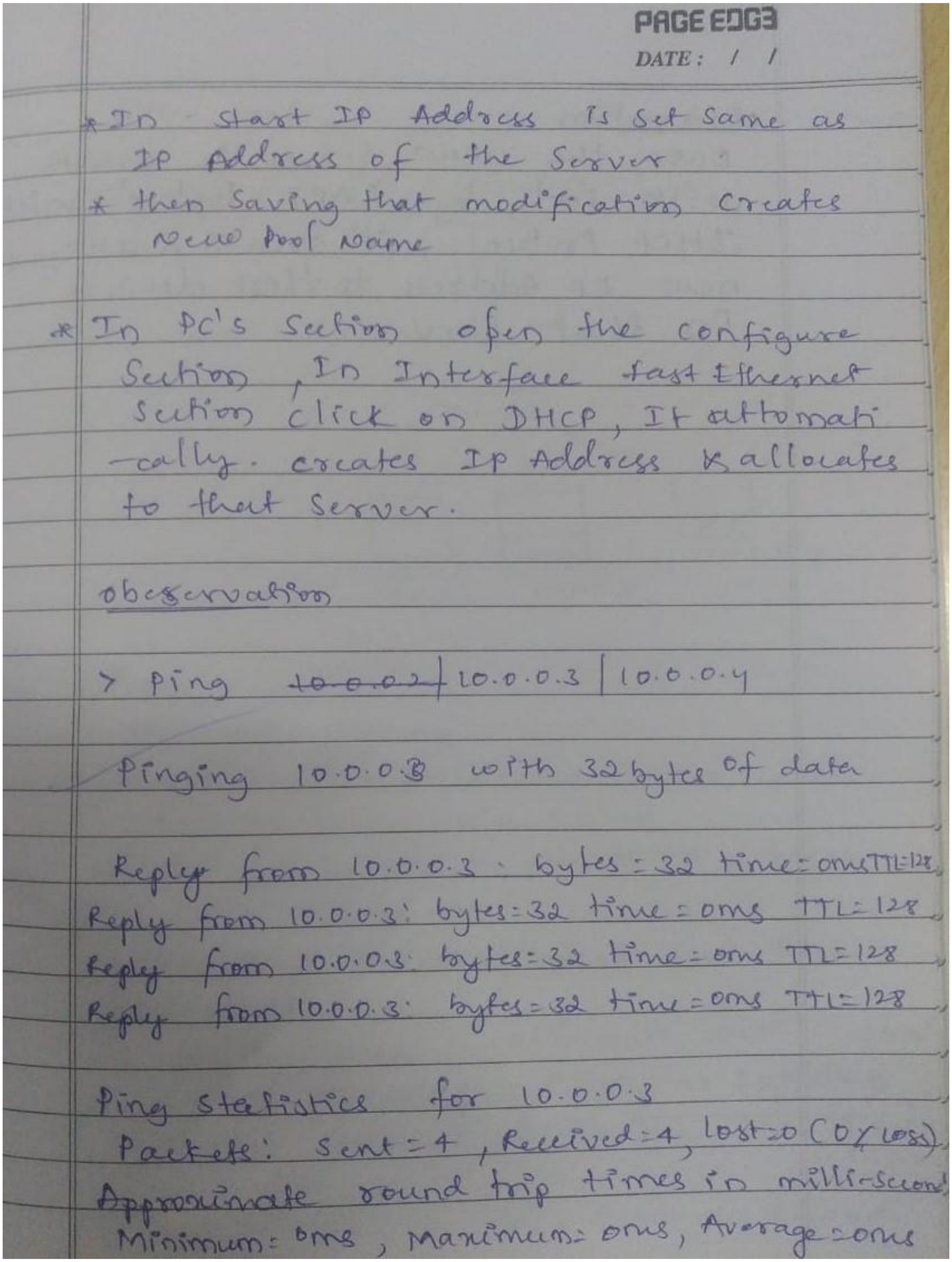
Topology:

Procedure

To make topology.

* Select 3 PC's, one switch, one Server
* connect 3 PC's to switch
* then connect Switch to Server

* first set the IP Address of the server
* In services section of server
  click on DHCP services
  * on the DHCP services
  * If need we can change the pool
    name of the server



PAGE EDG3
DATE: / /

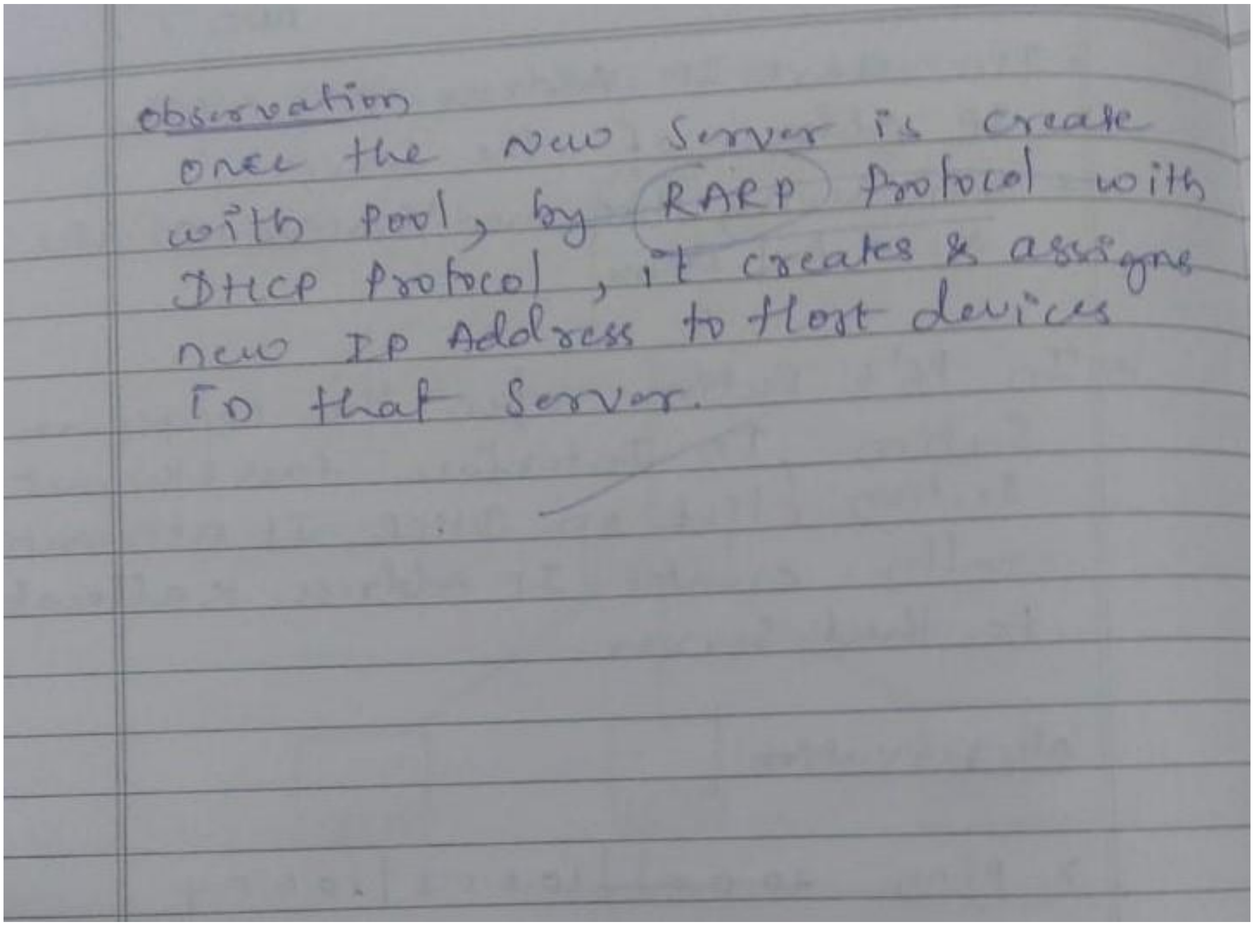
* In Start IP Address is Set Same as IP Address of the Server
* then Saving that modification Creates new Pool name

* In PC's Section open the configure Section, In Interface fast Ethernet Section click on DHCP, It attomatically. creates IP Address & allocates to that Server.

<u>observation</u>

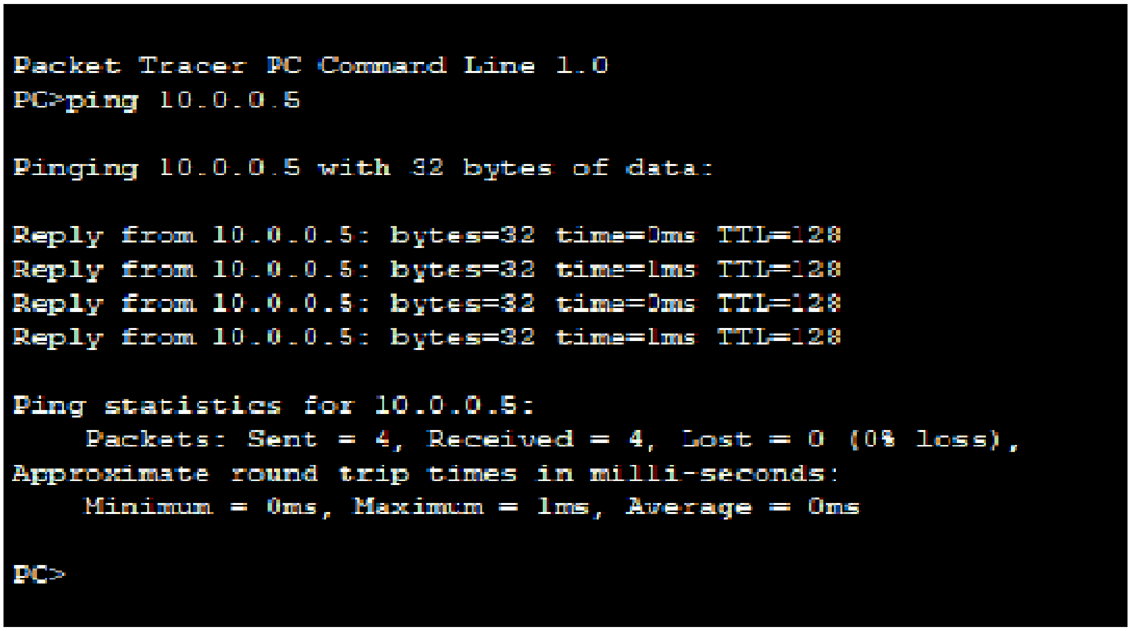
> ping ~~10.0.0.2~~ | 10.0.0.3 | 10.0.0.4

```
Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : bytes = 32 time = omsTTL=128
Reply from 10.0.0.3: bytes=32 time = oms TTL=128
Reply from 10.0.0.3: bytes=32 time = oms TTL=128
Reply from 10.0.0.3: bytes = 32 time = oms TTL=128

Ping Statistics for 10.0.0.3
Packets: Sent = 4 , Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-second
Minimum = oms , Maximum = oms, Average = oms
```



observation
once the New Server is create
with pool, by RARP Protocol with
DHCP Protocol, it creates & assigne
new IP Address to Host devices
To that Server.

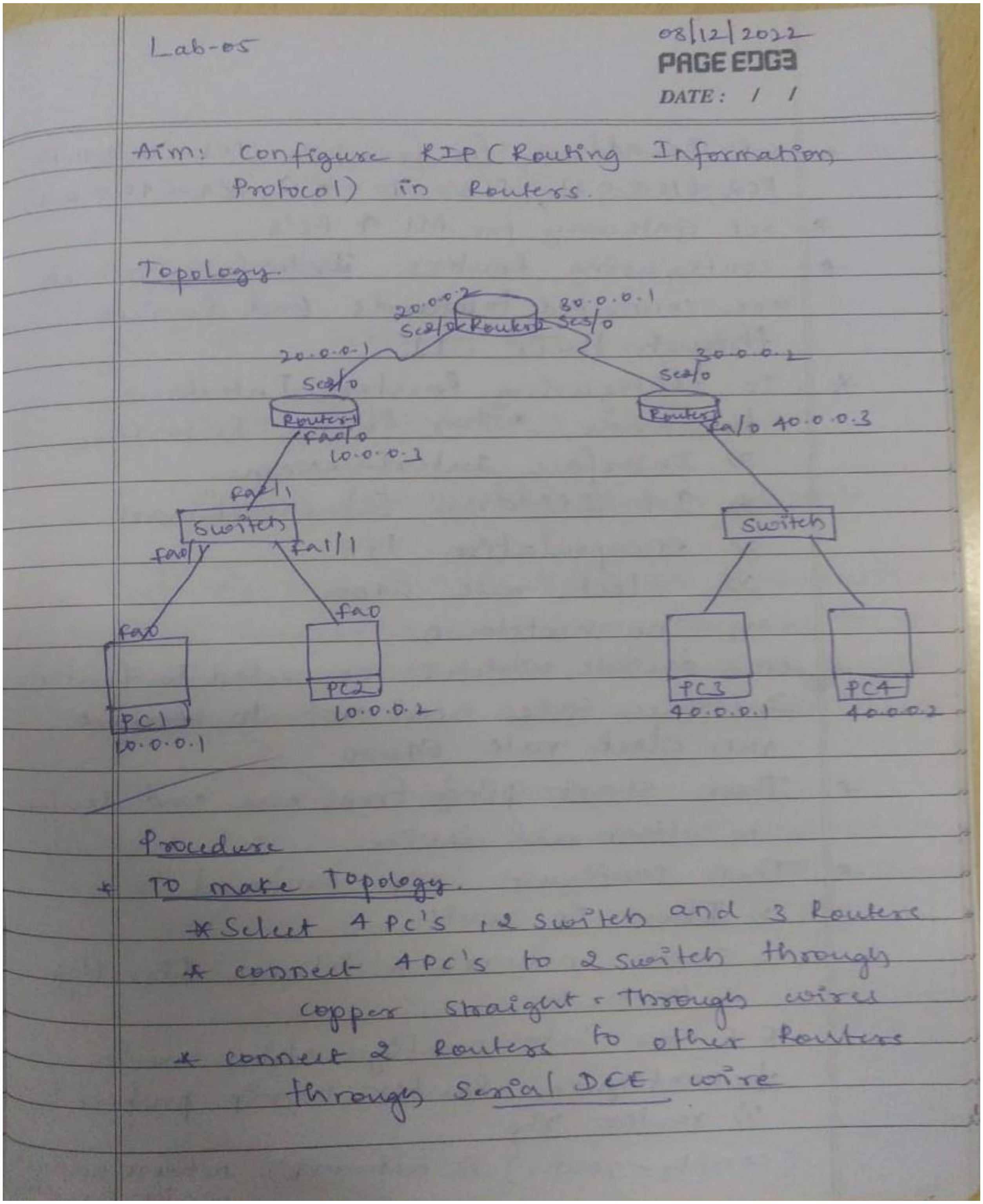
**OUTPUT:**



```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=1ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=1ms 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 = 1ms, Average = 0ms

PC>
```

# Experiment 5

Configuring RIP Routing Protocol in Routers



Lab-05

08/12/2022

PAGE EDG3

DATE : / /

Aim: Configure RIP (Routing Information Protocol) in Routers.

Topology.

20.0.0.2
Sc2/0
Router2
30.0.0.1
Sc3/0
20.0.0.1
Sc3/0
30.0.0.2
Sc2/0
Router-1
fa0/0
10.0.0.3
Router3
fa/0 40.0.0.3
fa1/1
Switch
fa0/1
fa1/1
fa0
fa0
PC1
10.0.0.1
PC2
10.0.0.2
Switch
PC3
40.0.0.1
PC4
40.0.0.2

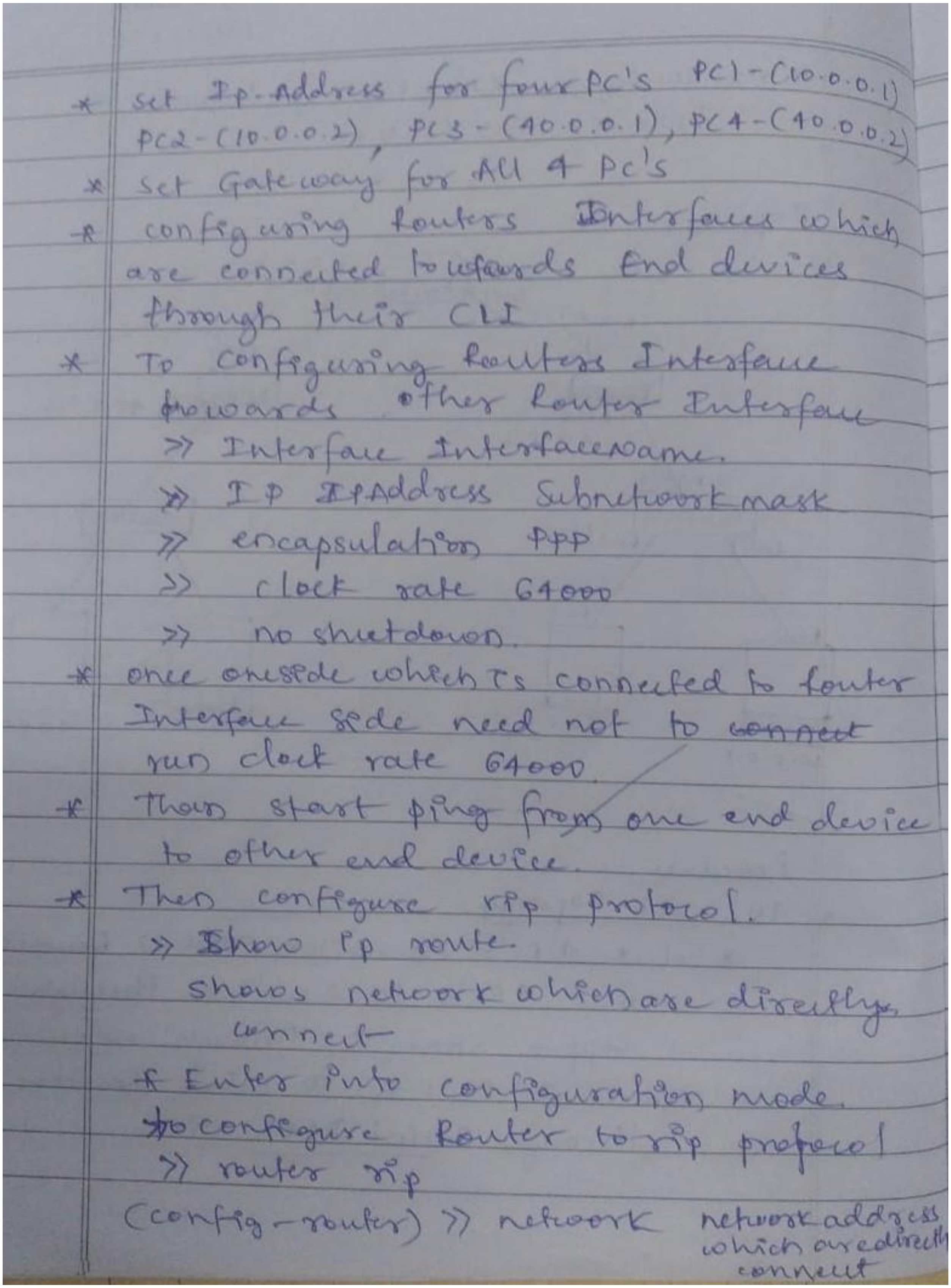
Procedure

* To make Topology.

* Select 4 PC's, 2 switch and 3 Routers

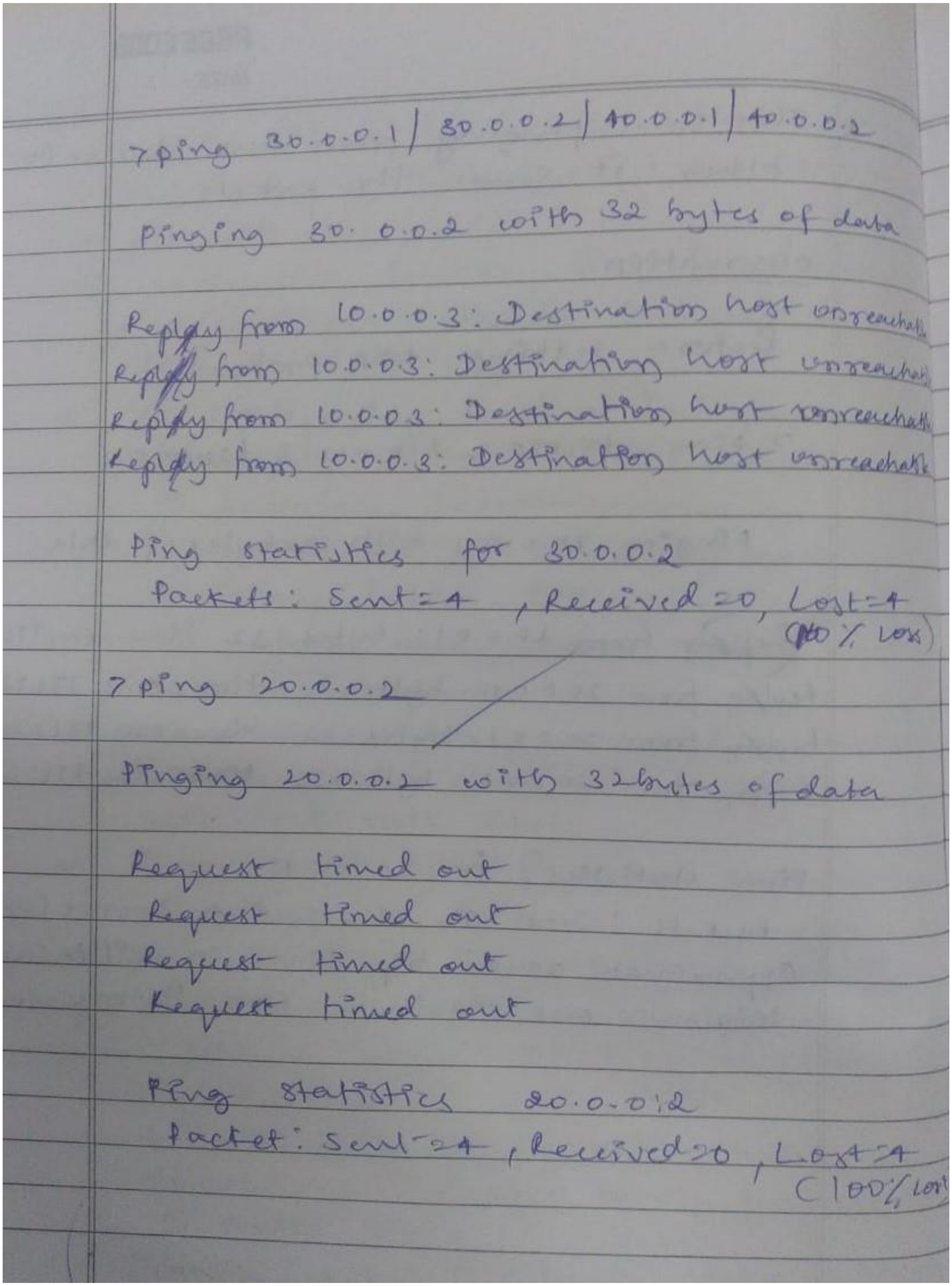
* connect 4 PC's to 2 switch through copper straight - Through wires

* connect 2 Routers to other Routers through Serial DCE wire

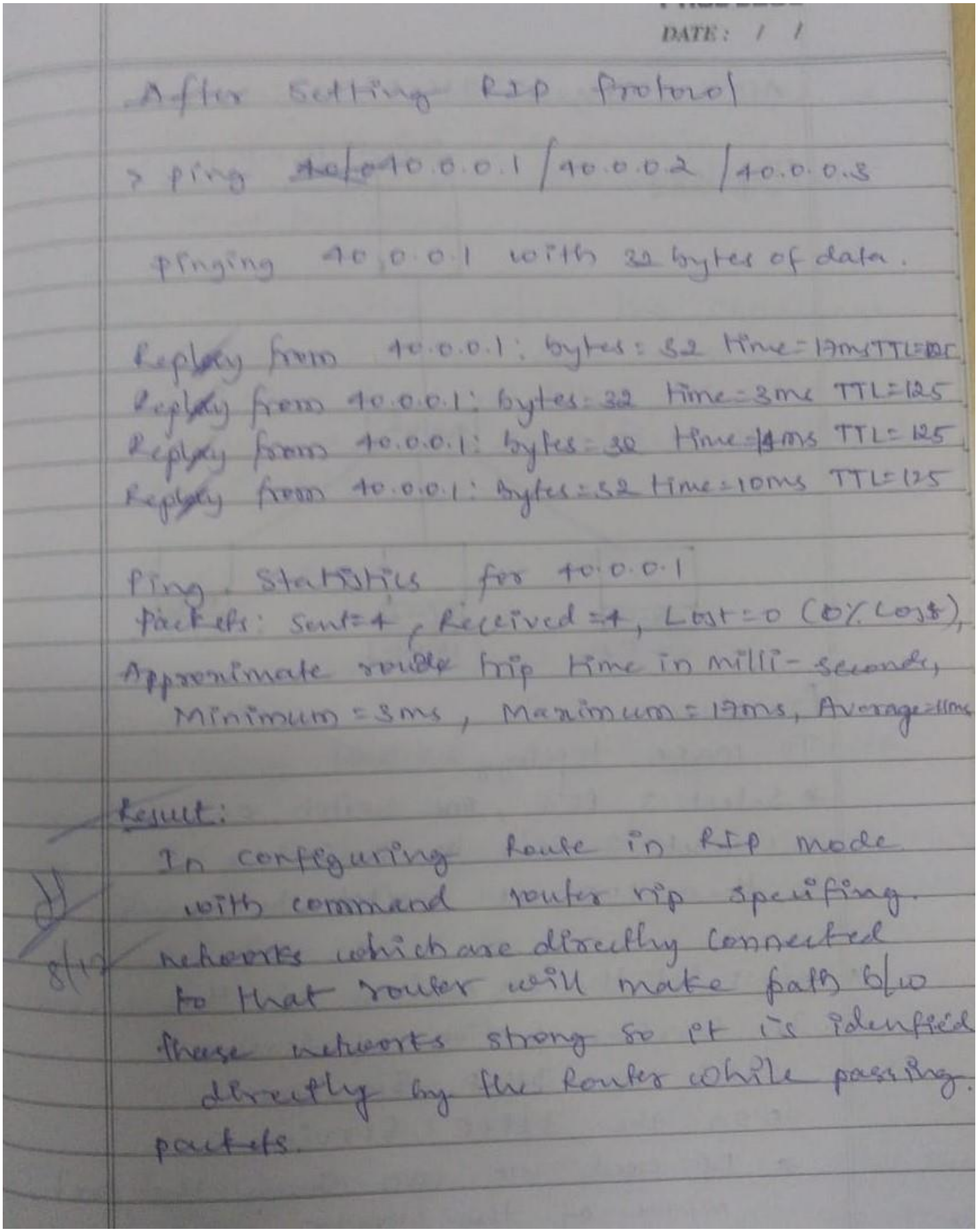


* set Ip-Address for four PC's PC1-(10.0.0.1) PC2-(10.0.0.2), PC3-(40.0.0.1), PC4-(40.0.0.2)
* set Gateway for All 4 PC's
* configuring Routers Interfaces which are connected towfards End devices through their CLI
* To configuring Roulters Interface towards other Router Interface
  >> Interface InterfaceName.
  >> IP IPAddress Subnetwork mask
  >> encapsulation PPP
  >> clock rate 64000
  >> no shutdown.
* once oneside which is connected to Router Interface side need not to ~~connect~~ run clock rate 64000.
* Then start ping from one end device to other end device.
* Then configure rip protocol.
  >> Show ip route.
  shows network which are directly connect

* Enter into configuration mode.
* to configure Router to rip protocol
  >> router rip
(config-router) >> network network address which are directly connect



```
>ping 30.0.0.1/ 30.0.0.2/ 40.0.0.1/ 40.0.0.2

Pinging 30. 0.0.2 with 32 bytes of data

Reply from 10.0.0.3: Destination host unreachable
Reply from 10.0.0.3: Destinating host unreachable
Reply from 10.0.0.3: Destination host unreachable
Reply from 10.0.0.3: Destination host unreachable

Ping statistics for 30.0.0.2
 Packets: Sent=4 , Received =0, Lost=4
                                (100% Loss)
>ping 20.0.0.2

Pinging 20.0.0.2 with 32bytes of data

Request timed out
Request timed out
Request timed out
Request timed out

Ping statistics 20.0.0;2
 Packet: Sent=4 , Received=0 , Lost=4
                                (100% Loss
```



DATE : / /

After Setting RIP Protocol

```
> ping ~~40.0~~ 40.0.0.1 / 40.0.0.2 / 40.0.0.3

pinging 40.0.0.1 with 32 bytes of data.

Reply from 40.0.0.1: bytes=32 time=17ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms TTL=125
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125

Ping statistics for 40.0.0.1
Packets: Sent=4, Received=4, Lost=0 (0% Loss),
Approximate round trip time in milli-seconds,
  Minimum=3ms, Maximum=17ms, Average=11ms
```

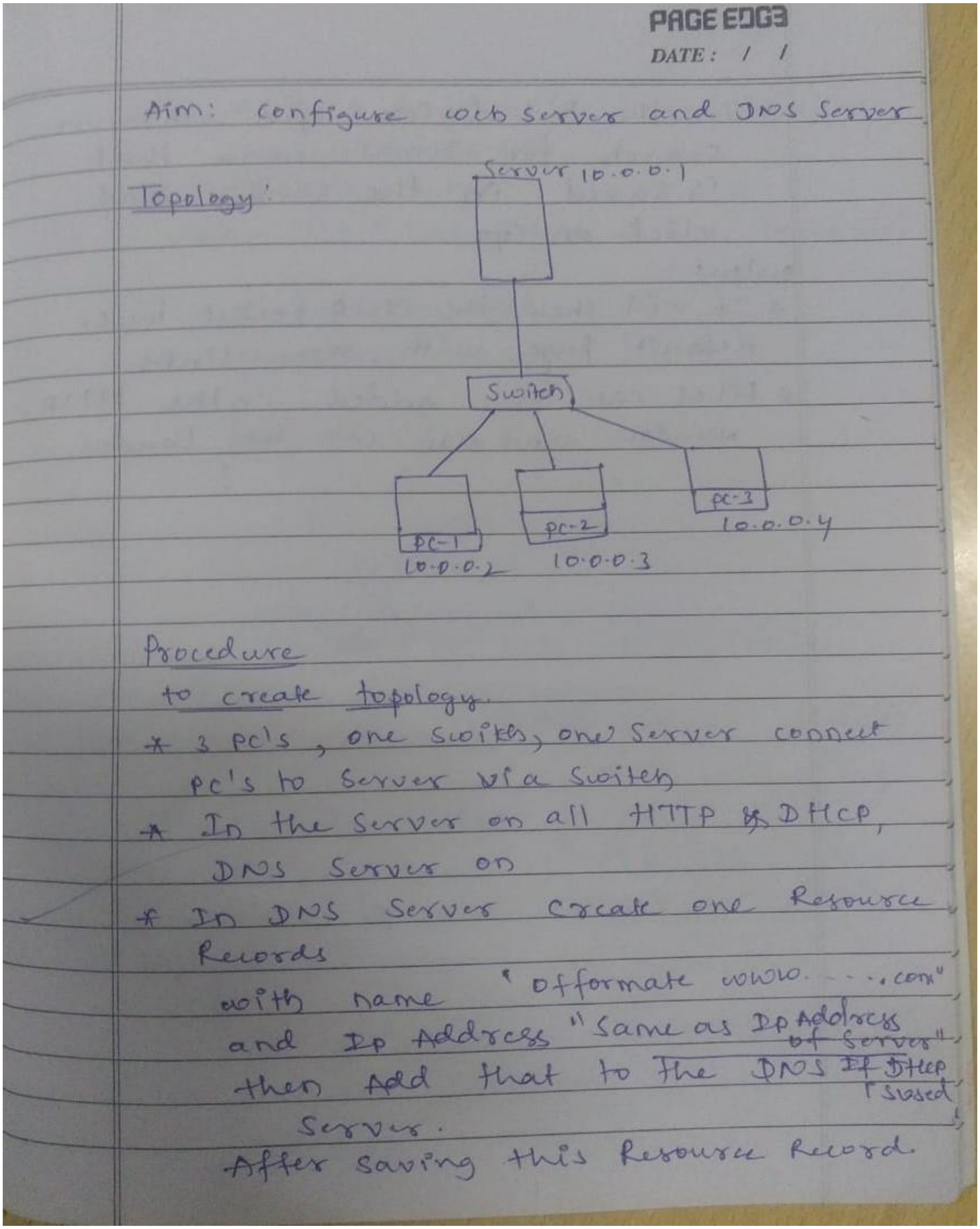
Result:

In configuring Route in RIP mode with command router rip specifing. networks which are directly connected to that router will make path b/w these networks strong so it is identified directly by the Router while passing packets.

8/10

## Experiment 6

Demonstration of WEB server and DNS using Packet Tracer



PAGE EDGE
DATE : / /

Aim: configure web server and DNS Server

Topology:

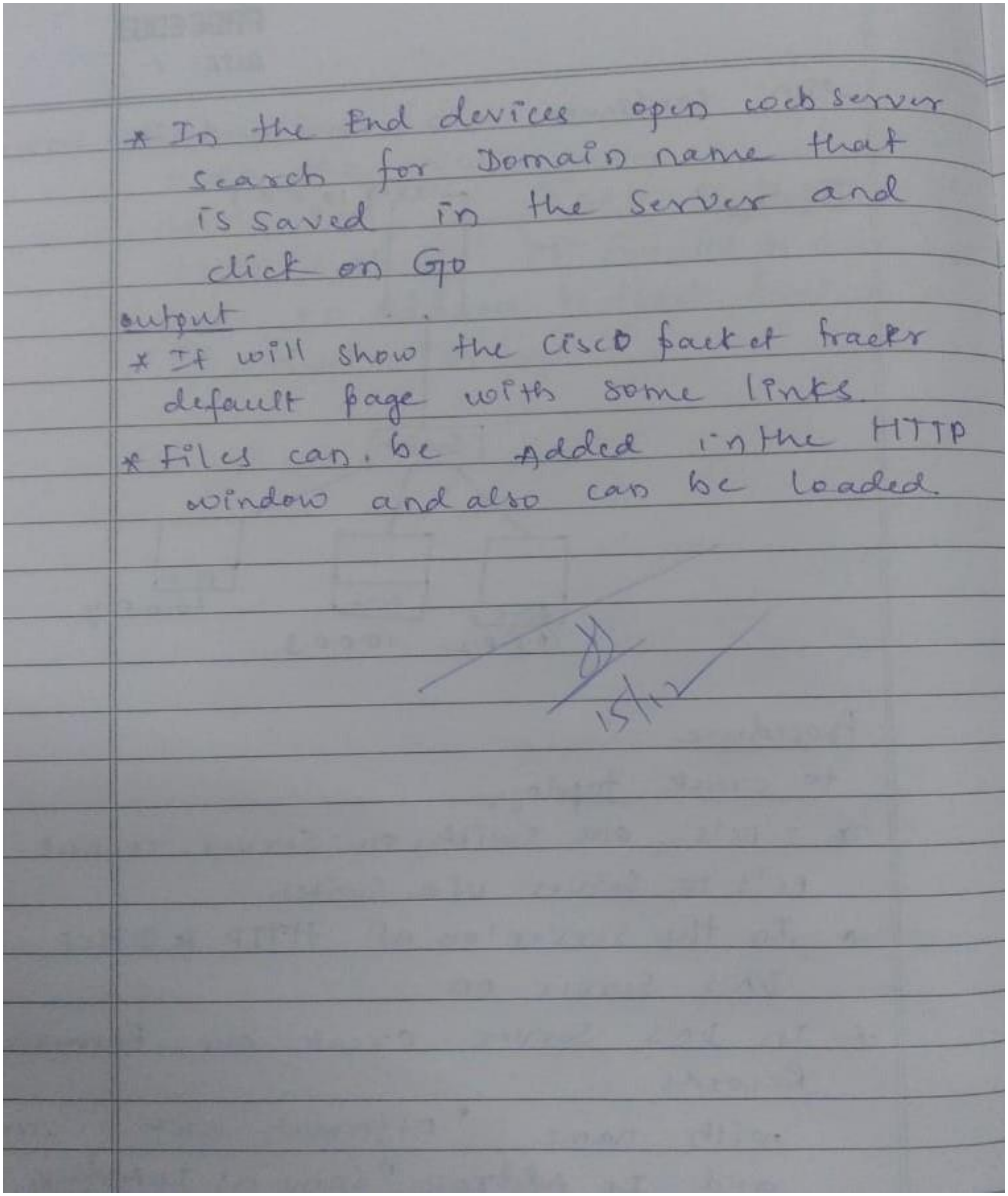
Server 10.0.0.1
Switch
PC-1
10.0.0.2
PC-2
10.0.0.3
PC-3
10.0.0.4

Procedure

to create topology.

* 3 PC's, one switch, one Server connect PC's to Server via switch
* In the Server on all HTTP & DHCP, DNS Server on
* In DNS Server create one Resource Records with name "offormate www. . . . .com" and Ip Address "Same as Ip Address of Server" then Add that to The DNS If DHCP is used Server.

After saving this Resource Record.



* In the End devices open web Server search for Domain name that is saved in the Server and click on Go

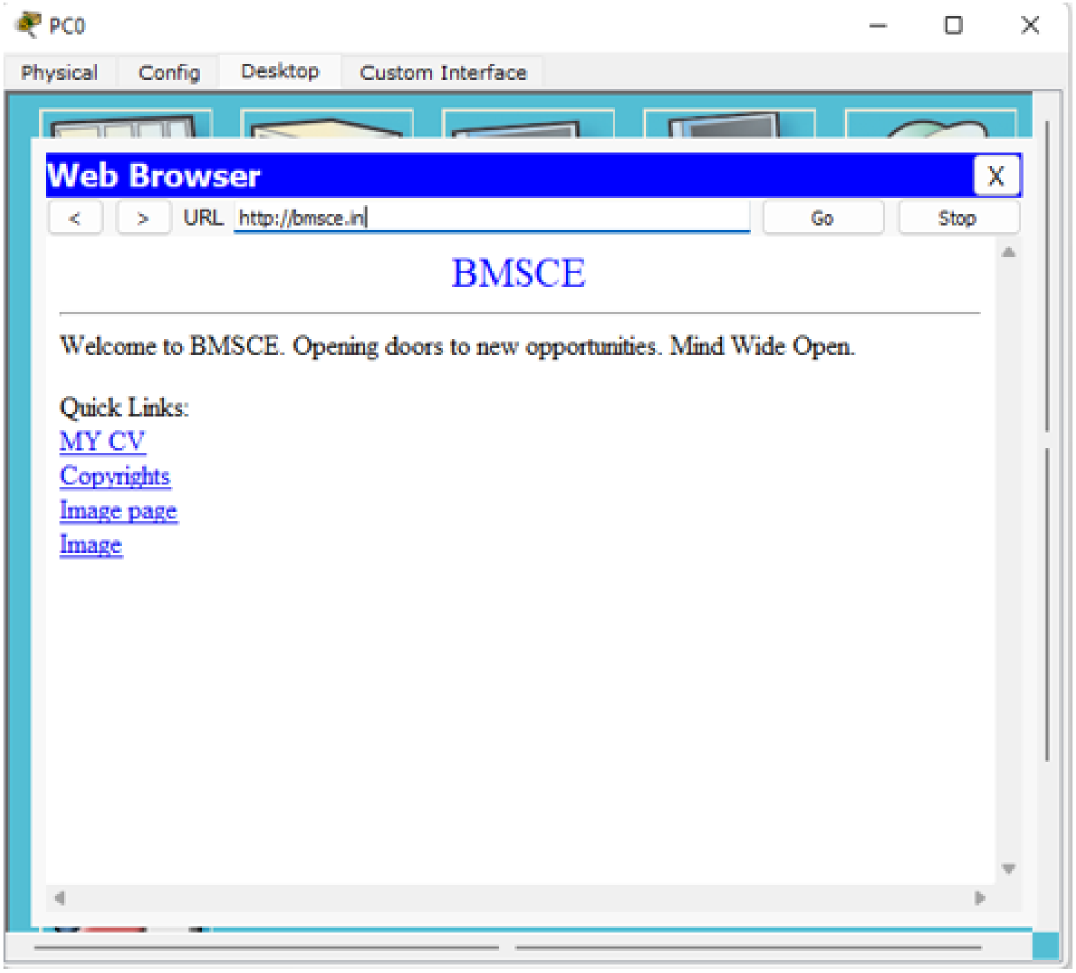
output

* It will show the cisco packet tracker default page with some links.

* Files can. be Added in the HTTP window and also can be loaded.

15/12

**OUTPUT:**



BMSCE

Welcome to BMSCE. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
MY CV
Copyrights
Image page
Image

## Cycle-2

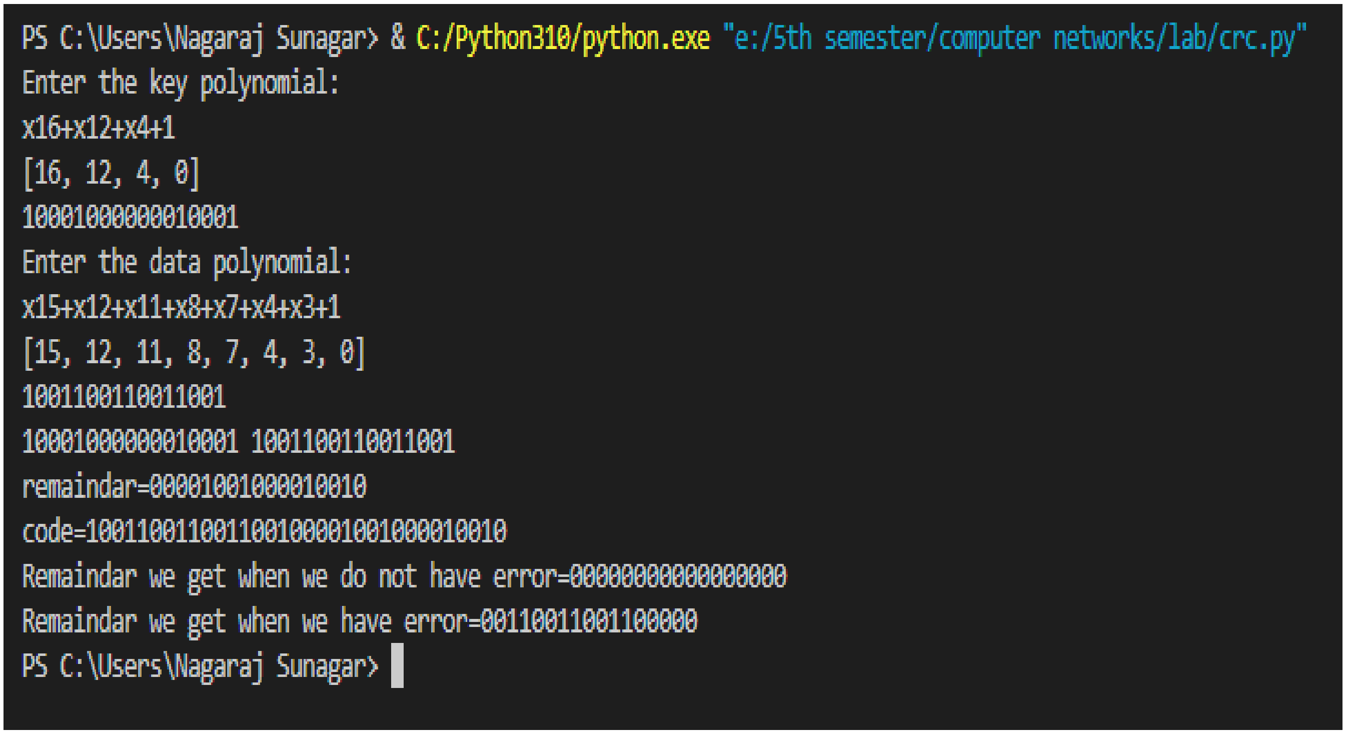
### 7. Write a program for error detecting code using CRC-CCITT (16-bits).

```
def xor1(a, b):
   x = ""
   for i in range(1, len(a)):
      if a[i] == b[i]:
         x += "0"
      else:
         x += "1"
   return x

def modulo2(divident, divisor):
   divlen = len(divisor)
   temp = divident[0:divlen]
   while(divlen < len(divident)):
      if temp[0] == "1":
         temp = xor1(temp, divisor)+divident[divlen]
      else:
         temp = temp[1:divlen]+divident[divlen]
      divlen += 1
   if temp[0] == "1":
      temp = xor1(temp, divisor)
   if len(temp) < len(divisor):
      return "0"+temp
   return temp


def encode(data, key):
   append = data+"0"*(len(key))
   rem = modulo2(append, key)
   print("remaindar="+rem)
   code = data+rem
   print("code="+code)
```

```
    # Checking the logic:

    rem = modulo2(code, key)
    print("Remaindar we get when we do not have error="+rem)
    code = code.replace("011", "101")
    rem = modulo2(code, key)
    print("Remaindar we get when we have error="+rem)


def polytobin(string):
    keys = []
    key = ""
    for i in string:
        if i == '+':
            keys.append(int(key[1:]))
            key = ""
            continue
        key += i
    if key != "":
        keys.append(0)
    binary = ""
    j = 0
    print(keys)
    for i in range(keys[0], -1, -1):
        if i == (keys[j]):
            binary += "1"
            j += 1
        else:
            binary += "0"
    print(binary)
    return binary

string = input("Enter the key polynomial:\n")
key = polytobin(string)
string = input("Enter the data polynomial:\n")
```

```
data = polytobin(string)
print(key, data)
encode(data, key)
```

### Output:

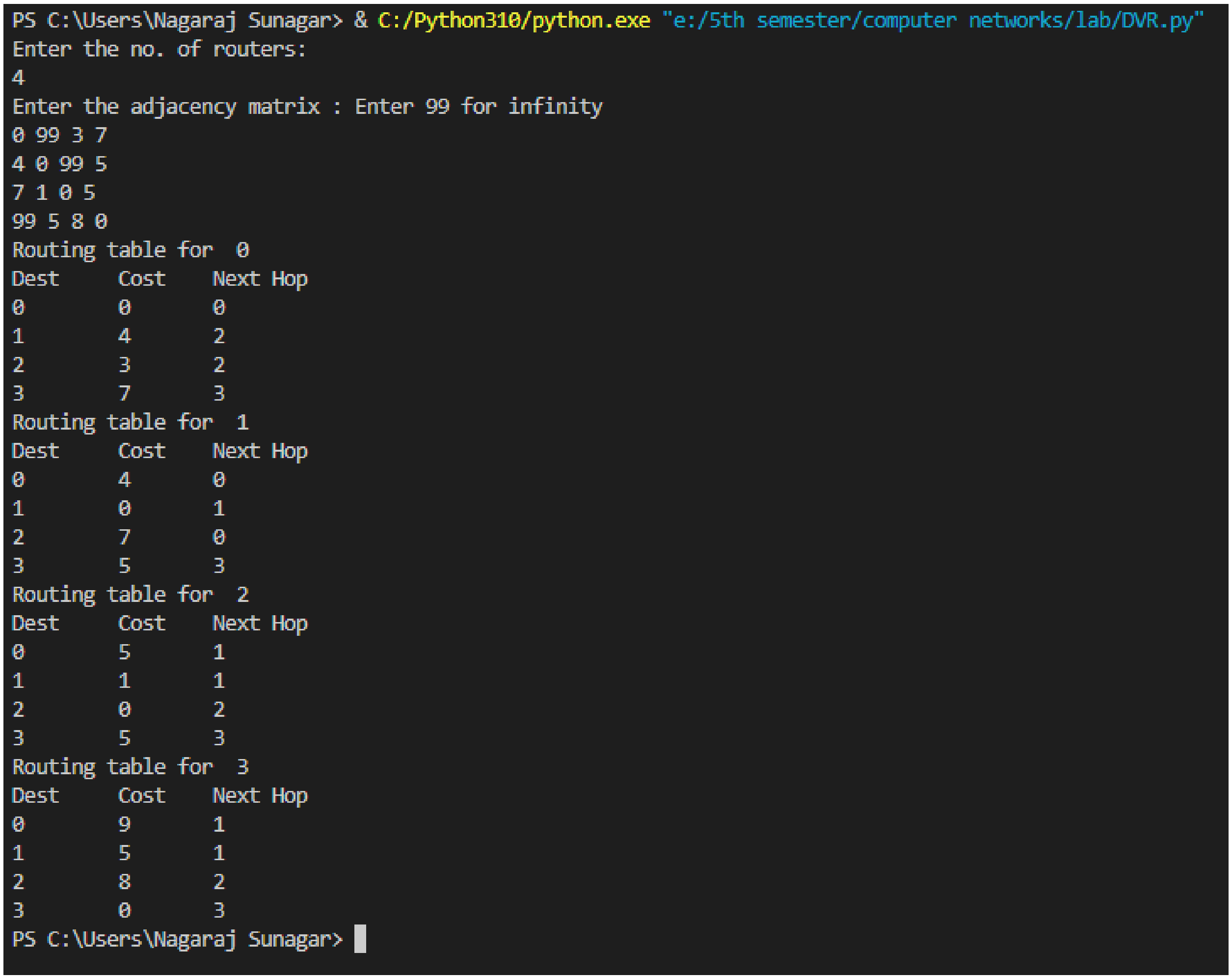


```
PS C:\Users\Nagaraj Sunagar> & C:/Python310/python.exe "e:/5th semester/computer networks/lab/crc.py"
Enter the key polynomial:
x16+x12+x4+1
[16, 12, 4, 0]
10001000000010001
Enter the data polynomial:
x15+x12+x11+x8+x7+x4+x3+1
[15, 12, 11, 8, 7, 4, 3, 0]
1001100110011001
10001000000010001 1001100110011001
remaindar=00001001000010010
code=100110011001100100001001000010010
Remaindar we get when we do not have error=00000000000000000
Remaindar we get when we have error=00110011001100000
PS C:\Users\Nagaraj Sunagar>
```

**8. Write a program for distance vector algorithm to find suitable path for transmission.**

```
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = []

    def add_edge(self, s, d, w):
        self.graph.append([s, d, w])

    def print_solution(self, dist, src, next_hop):
        print("Routing table for ", src)
        print("Dest \t Cost \t Next Hop")
        for i in range(self.V):
            print("{0} \t {1} \t {2}".format(i, dist[i], next_hop[i]))

    def bellman_ford(self, src):
        dist = [99] * self.V
        dist[src] = 0
        next_hop = {src: src}
        for _ in range(self.V - 1):
            for s, d, w in self.graph:
                if dist[s] != 99 and dist[s] + w < dist[d]:
                    dist[d] = dist[s] + w
                    if s == src:
                        next_hop[d] =d
                    elif s in next_hop:
                        next_hop[d] = next_hop[s]
        for s, d, w in self.graph:
            if dist[s] != 99 and dist[s] + w < dist[d]:
                print("Graph contains negative weight cycle")
                return   self.print_solution(dist, src, next_hop)

def main():
    matrix = []
    print("Enter the no. of routers:")
    n = int(input())
    print("Enter the adjacency matrix : Enter 99 for infinity")
    for i in range(0,n):
        a = list(map(int, input().split(" ")))
        matrix.append(a)

    g = Graph(n)
    for i in range(0,n):
        for j in range(0,n):
            g.add_edge(i,j,matrix[i][j])

    for k in range(0, n):
        g.bellman_ford(k)
main()
```

OUTPUT:



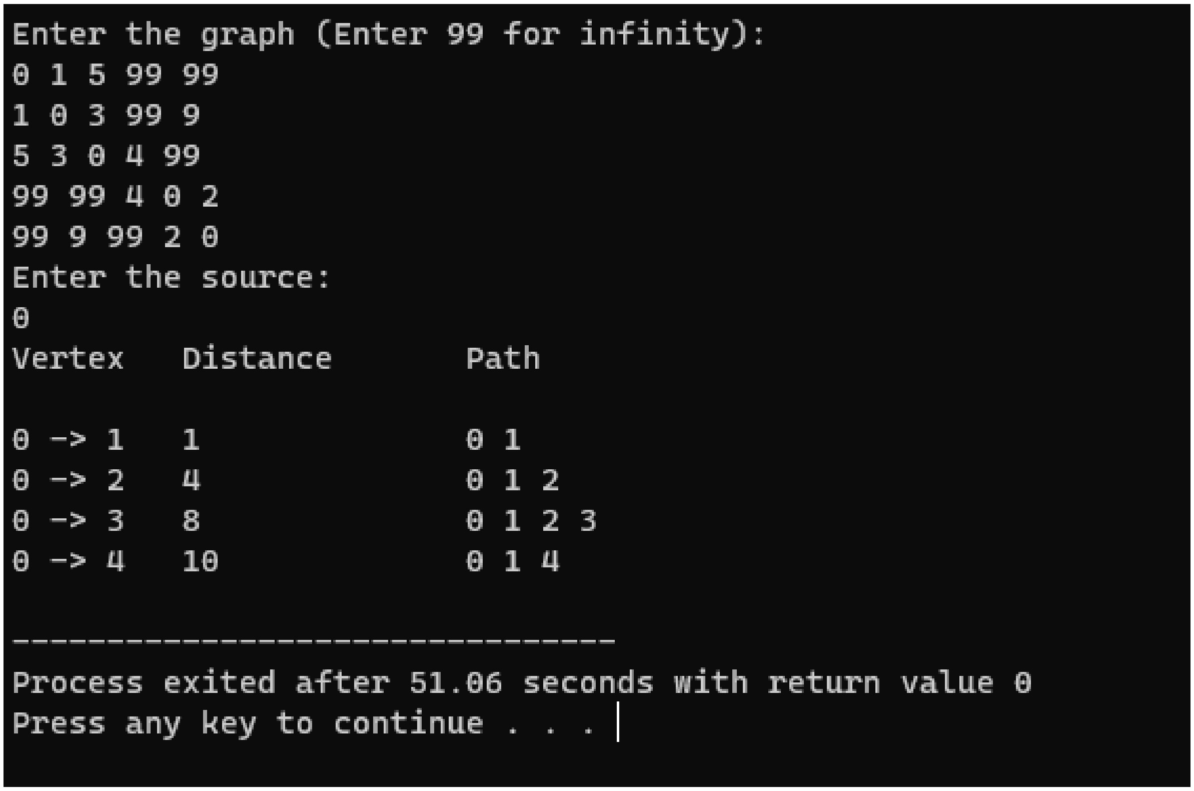
```
PS C:\Users\Nagaraj Sunagar> & C:/Python310/python.exe "e:/5th semester/computer networks/lab/DVR.py"
Enter the no. of routers:
4
Enter the adjacency matrix : Enter 99 for infinity
0 99 3 7
4 0 99 5
7 1 0 5
99 5 8 0
Routing table for  0
Dest     Cost    Next Hop
0        0       0
1        4       2
2        3       2
3        7       3
Routing table for  1
Dest     Cost    Next Hop
0        4       0
1        0       1
2        7       0
3        5       3
Routing table for  2
Dest     Cost    Next Hop
0        5       1
1        1       1
2        0       2
3        5       3
Routing table for  3
Dest     Cost    Next Hop
0        9       1
1        5       1
2        8       2
3        0       3
PS C:\Users\Nagaraj Sunagar> 
```

**9. Implement Dijkstra’s algorithm to compute the shortest path for a given topology.**

```
#include<bits/stdc++.h>
using namespace std;
#define V 5

int minDistance(int dist[], bool sptSet[])
{
   int min = 9999, min_index;
   for (int v = 0; v < V; v++)
      if (sptSet[v] == false && dist[v] <= min)
         min = dist[v], min_index = v;
   return min_index;
}

void printPath(int parent[], int j)
{
   if (parent[j] == - 1)
      return;
   printPath(parent, parent[j]);
   cout<<j<<" ";
}

void printSolution(int dist[], int n, int parent[])
{
   int src = 0;
   cout<<"Vertex\t Distance\tPath"<<endl;
   for (int i = 1; i < V; i++)
   {
      cout<<"\n"<<src<<" -> "<<i<<" \t "<<dist[i]<<"\t\t"<<src<<" ";
      printPath(parent, i);
   }
}

void dijkstra(int graph[V][V], int src)
{
   int dist[V];

   bool sptSet[V];
   int parent[V];
   for (int i = 0; i < V; i++)
   {
      parent[0] = -1;
      dist[i] = 9999;
      sptSet[i] = false;
   }

   dist[src] = 0;

   for (int count = 0; count < V - 1; count++)
   {
      int u = minDistance(dist, sptSet);
      sptSet[u] = true;
      for (int v = 0; v < V; v++)
         if (!sptSet[v] && graph[u][v] &&
             dist[u] + graph[u][v] < dist[v])
         {
             parent[v] = u;
             dist[v] = dist[u] + graph[u][v];
         }
   }

   printSolution(dist, V, parent);
}

int main()
{
   int graph[V][V];
   cout<<"Enter the graph (Enter 99 for infinity): "<<endl;
   for(int i = 0; i<V; i++)
   {
      for(int j = 0; j<V; j++)
```

```
            cin>>graph[i][j];
    }
    cout<<"Enter the source: "<<endl;
    int src;
    cin>>src;

    dijkstra(graph, src);
    cout<<endl;
    return 0;
}
```

**OUTPUT:**



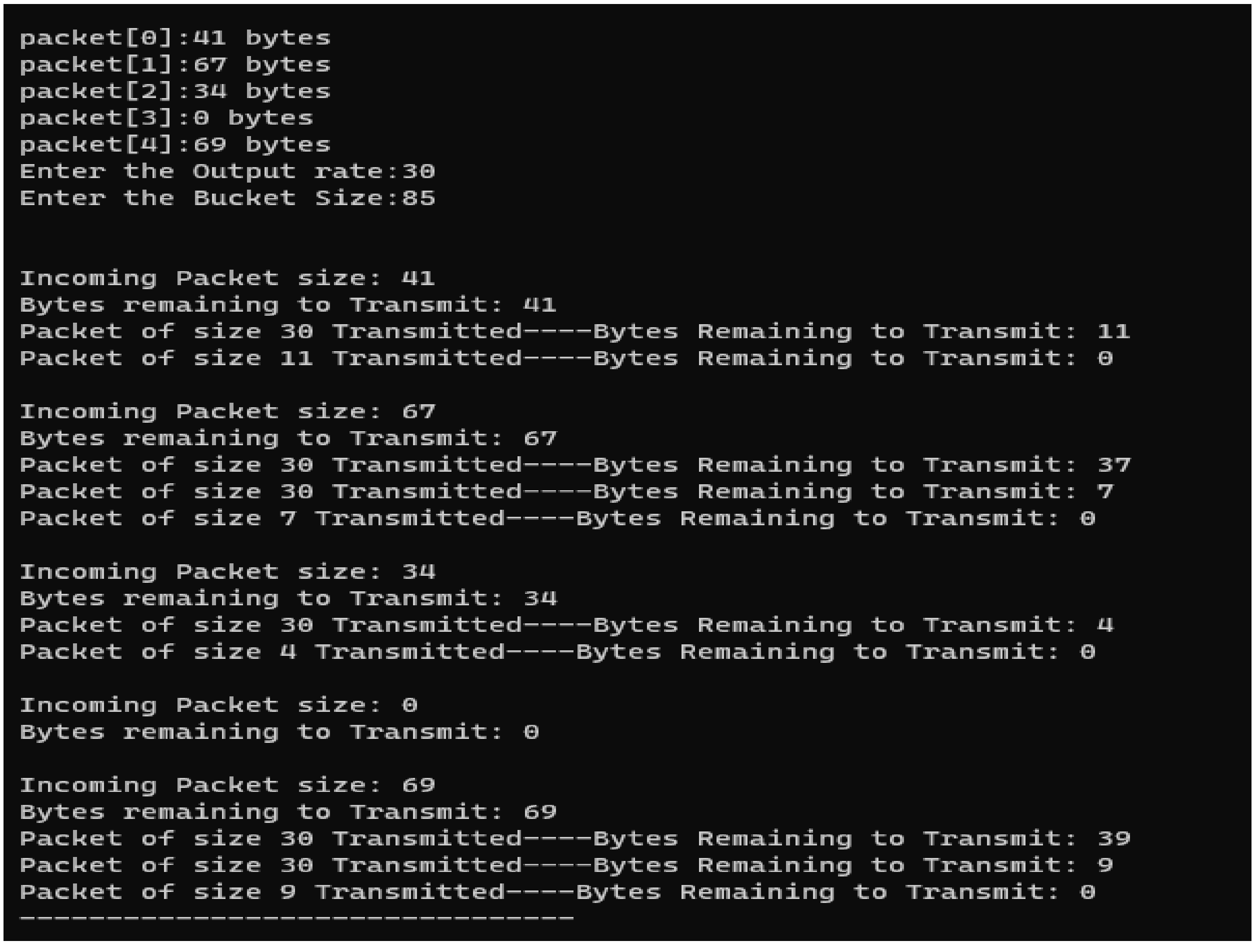
```
Enter the graph (Enter 99 for infinity):
0 1 5 99 99
1 0 3 99 9
5 3 0 4 99
99 99 4 0 2
99 9 99 2 0
Enter the source:
0
Vertex   Distance       Path

0 -> 1   1              0 1
0 -> 2   4              0 1 2
0 -> 3   8              0 1 2 3
0 -> 4   10             0 1 4

--------------------------------
Process exited after 51.06 seconds with return value 0
Press any key to continue . . .
```

**10. Write a program for congestion control using Leaky bucket algorithm**

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define NOF_PACKETS 5

int main()
{
   int packet_sz[NOF_PACKETS], i, b_size, o_rate, p_sz_rm = 0, p_sz, op;
   for (i = 0; i < NOF_PACKETS; ++i)
      packet_sz[i] = rand() % 100;
   for (i = 0; i < NOF_PACKETS; ++i)
      printf("\npacket[%d]:%d bytes\t", i, packet_sz[i]);
   printf("\nEnter the Output rate:");
   scanf("%d", &o_rate);
   printf("Enter the Bucket Size:");
   scanf("%d", &b_size);
   for (i = 0; i < NOF_PACKETS; ++i)
   {
      if ((packet_sz[i] + p_sz_rm) > b_size)
         if (packet_sz[i] > b_size) /*compare the packet siz with bucket size*/
            printf("\n\nIncoming packet size (%dbytes) is Greater than bucket capacity (%dbytes)-PACKET REJECTED", packet_sz[i], b_size);
         else
            printf("\n\nBucket capacity exceeded-PACKETS REJECTED!!");
      else
      {
         p_sz_rm += packet_sz[i];
         printf("\n\nIncoming Packet size: %d", packet_sz[i]);
         printf("\nBytes remaining to Transmit: %d", p_sz_rm);
         while (p_sz_rm > 0)
         {
            sleep(1);
            if (p_sz_rm)
            {
               if (p_sz_rm <= o_rate) /*packet size remaining comparing with output rate*/
```

```
                  op = p_sz_rm, p_sz_rm = 0;
               else
                  op = o_rate, p_sz_rm -= o_rate;
               printf("\nPacket of size %d Transmitted", op);
               printf("----Bytes Remaining to Transmit: %d", p_sz_rm);
            }
            else
            {
               printf("\nNo packets to transmit!!");
            }
         }
      }
   }
}
```

**OUTPUT:**



```
packet[0]:41 bytes
packet[1]:67 bytes
packet[2]:34 bytes
packet[3]:0 bytes
packet[4]:69 bytes
Enter the Output rate:30
Enter the Bucket Size:85


Incoming Packet size: 41
Bytes remaining to Transmit: 41
Packet of size 30 Transmitted----Bytes Remaining to Transmit: 11
Packet of size 11 Transmitted----Bytes Remaining to Transmit: 0

Incoming Packet size: 67
Bytes remaining to Transmit: 67
Packet of size 30 Transmitted----Bytes Remaining to Transmit: 37
Packet of size 30 Transmitted----Bytes Remaining to Transmit: 7
Packet of size 7 Transmitted----Bytes Remaining to Transmit: 0

Incoming Packet size: 34
Bytes remaining to Transmit: 34
Packet of size 30 Transmitted----Bytes Remaining to Transmit: 4
Packet of size 4 Transmitted----Bytes Remaining to Transmit: 0

Incoming Packet size: 0
Bytes remaining to Transmit: 0

Incoming Packet size: 69
Bytes remaining to Transmit: 69
Packet of size 30 Transmitted----Bytes Remaining to Transmit: 39
Packet of size 30 Transmitted----Bytes Remaining to Transmit: 9
Packet of size 9 Transmitted----Bytes Remaining to Transmit: 0
--------------------------------
```

**11. 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.**

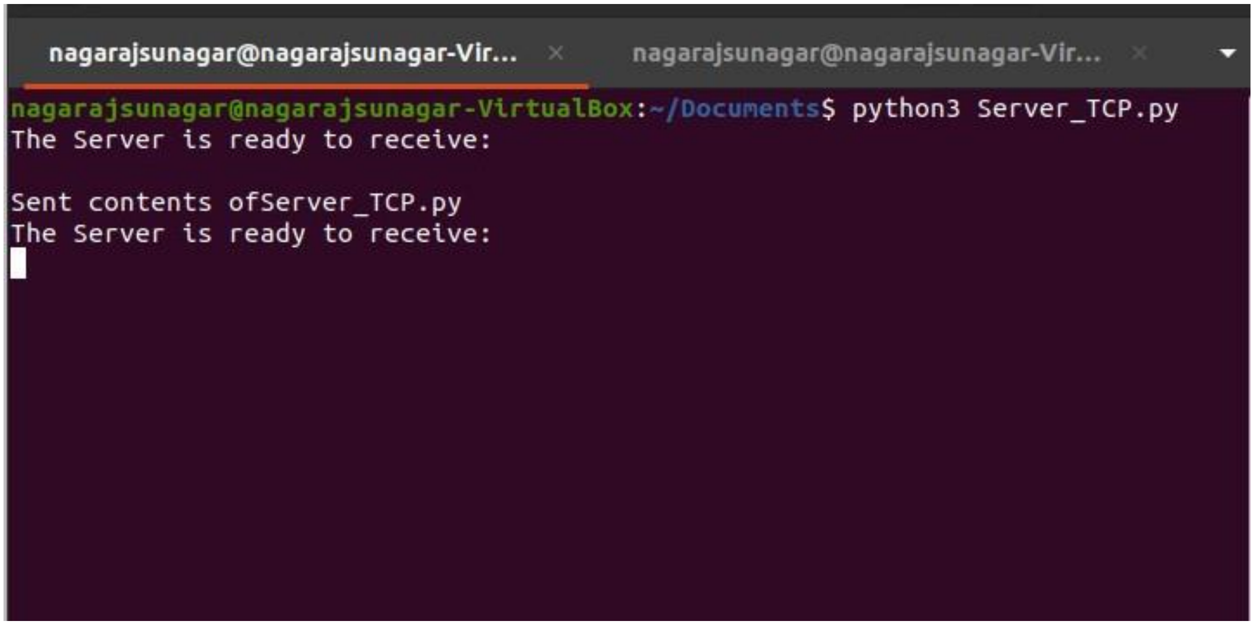
**SERVER:**

```
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()
```

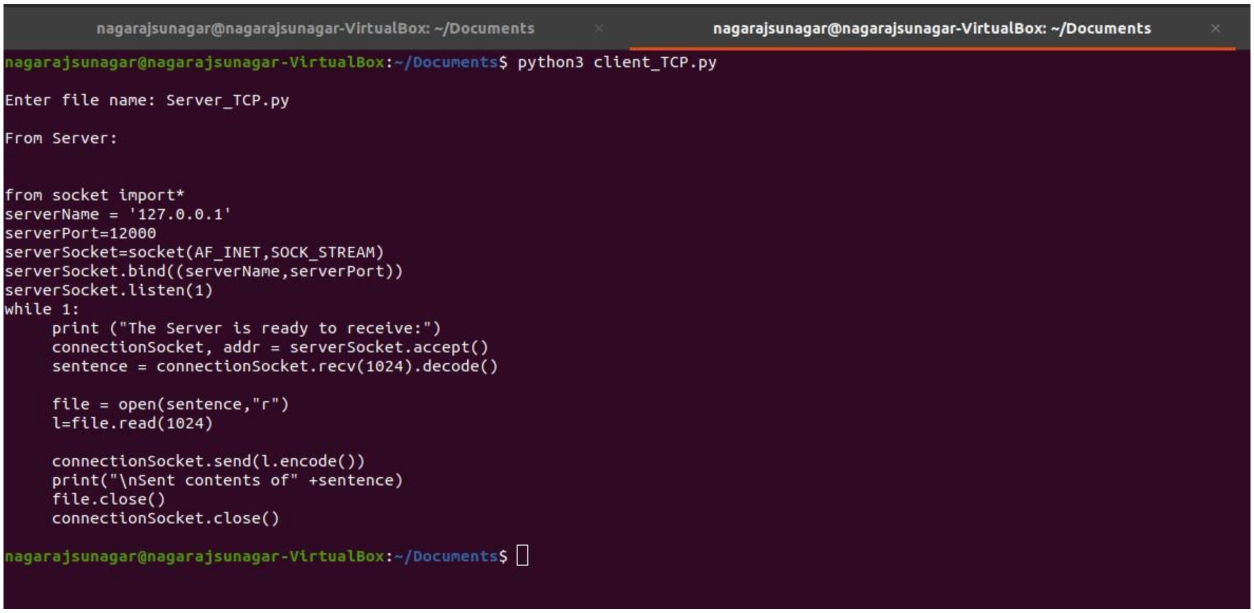
**CLIENT:**

```
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()
```

OUTPUT:


nagarajsunagar@nagarajsunagar-Vir...
nagarajsunagar@nagarajsunagar-Vir...

```
nagarajsunagar@nagarajsunagar-VirtualBox:~/Documents$ python3 Server_TCP.py
The Server is ready to receive:

Sent contents ofServer_TCP.py
The Server is ready to receive:
```


nagarajsunagar@nagarajsunagar-VirtualBox: ~/Documents
nagarajsunagar@nagarajsunagar-VirtualBox: ~/Documents

```
nagarajsunagar@nagarajsunagar-VirtualBox:~/Documents$ python3 client_TCP.py

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("\nSent contents of" +sentence)
     file.close()
     connectionSocket.close()

nagarajsunagar@nagarajsunagar-VirtualBox:~/Documents$
```

**12. 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**.

SERVER:

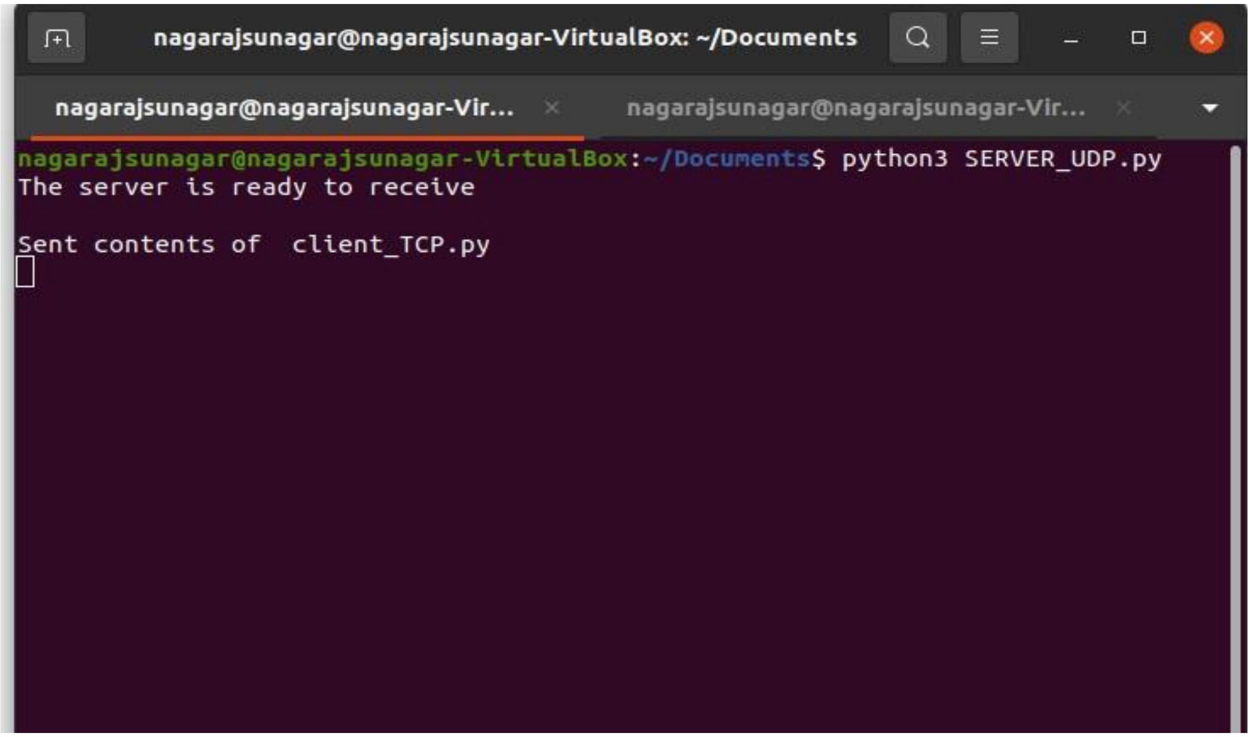
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")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

    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)

    print ('\nSent contents of ', end = ' ')
    print (sentence)
   #  for i in sentence:
      #  print (str(i), end = '')
    file.close()
```

CLIENT:

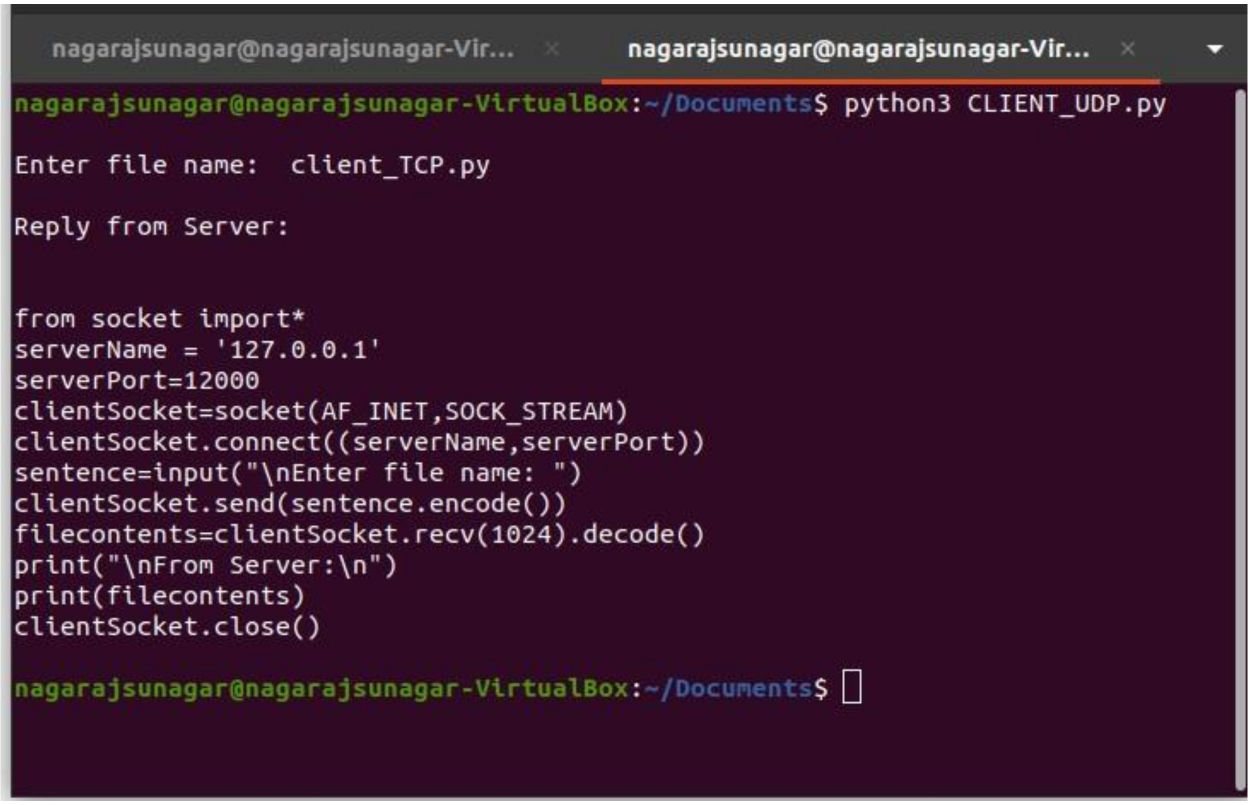
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name:  ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode("utf-8"))
# for i in filecontents:
   # print(str(i), end = '')
clientSocket.close()
```

clientSocket.close()

OUTPUT:


nagarajsunagar@nagarajsunagar-VirtualBox: ~/Documents
nagarajsunagar@nagarajsunagar-Vir...
nagarajsunagar@nagarajsunagar-Vir...

```
nagarajsunagar@nagarajsunagar-VirtualBox:~/Documents$ python3 SERVER_UDP.py
The server is ready to receive

Sent contents of  client_TCP.py
```


nagarajsunagar@nagarajsunagar-Vir...
nagarajsunagar@nagarajsunagar-Vir...

```
nagarajsunagar@nagarajsunagar-VirtualBox:~/Documents$ python3 CLIENT_UDP.py

Enter file name:  client_TCP.py

Reply from Server:


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()

nagarajsunagar@nagarajsunagar-VirtualBox:~/Documents$ 
```