VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

R SHREYA (1BM21CS152)

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 **R SHREYA (1BM21CS152),** 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 academic semester June-2023 to September-2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks (22CS4PCCON)** work prescribed for the said degree.

Name of the Lab-In charge:	Dr. Jyothi S Nayak
Nandini Vineeth	Professorand Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index Sheet

Lab Program No.	Program Details	Page No.
	CYCLE 1	
1	Create a topology and simulate sending a simple PDUfrom source to destination using hub and switch as connecting devices and demonstrate ping message.	4
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	7
3	Default routing and Static Routing	13
4	DHCP within LAN and outside LAN	18
5	Configure RIP routing Protocol in Routers	23
6	Configure OSPF routing protocol	27
7	Demonstrate the TTL/ Life of a Packet	32
8	Configure Web Server, DNS within a LAN.	35
9	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	38
10	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	41
11	To construct a VLAN and make the PC's communicate among a VLAN	44
12	To construct a WLAN and make the nodes communicate wirelessly	48
	CYCLE 2	

13	Write a program for error detecting code using CRC-CCITT (16-bits).	52
14	Write a program for congestion control using Leaky bucket algorithm.	56
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.	59
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.	63

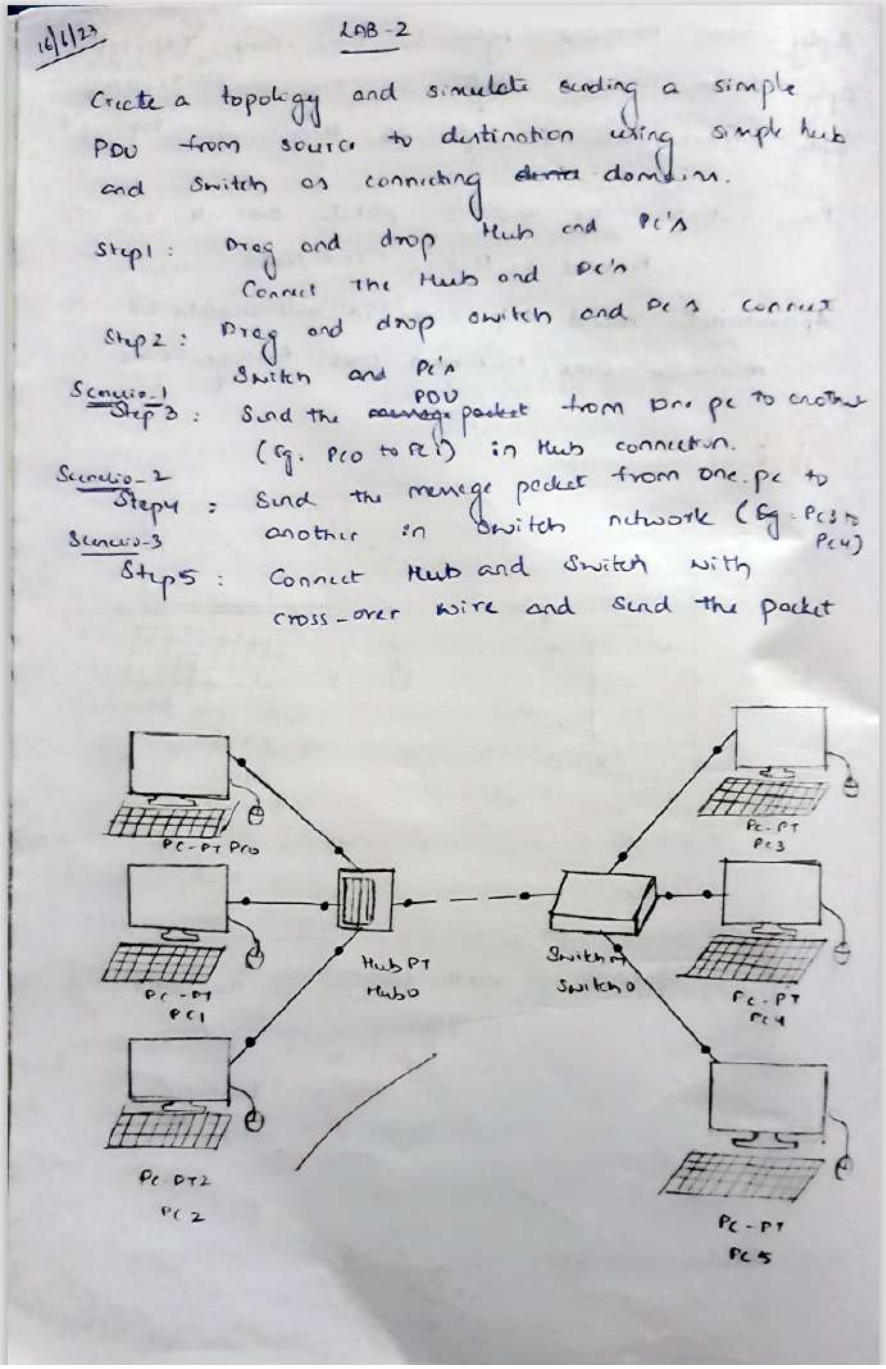
Course Outcome

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

CYCLE 1

PROGRAM -1

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



Scenario - 4:

When Switch is off the data is not transmitted between the PC's

Output:

Command prompt:

⇒ Click on Command: ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3 : bytes = 32 time = 8ms TTL = 12
Reply from 10.0.0.3 : bytes = 32 time = 4ms TTL = 1
Reply from 10.0.0.3 : bytes = 32 time = 4ms TTL = 12
Reply from 10.0.0.3 : bytes = 32 time = 4ms TTL = R

Ping statistics for 10.0.0.3:

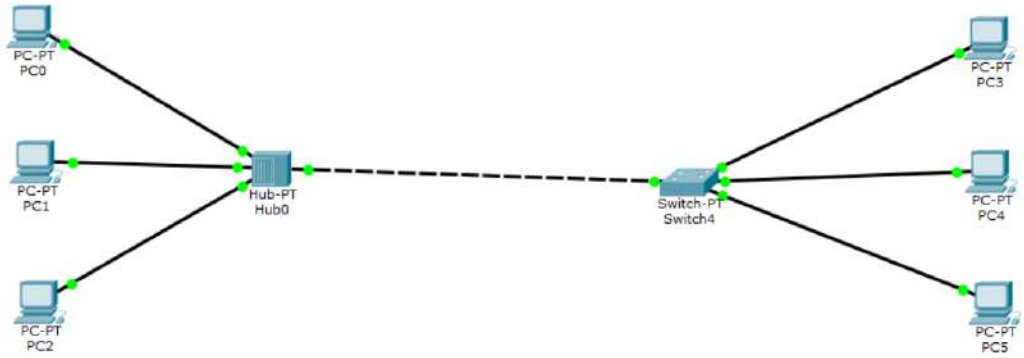
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)

Approximate round trip times in milli-seconds:

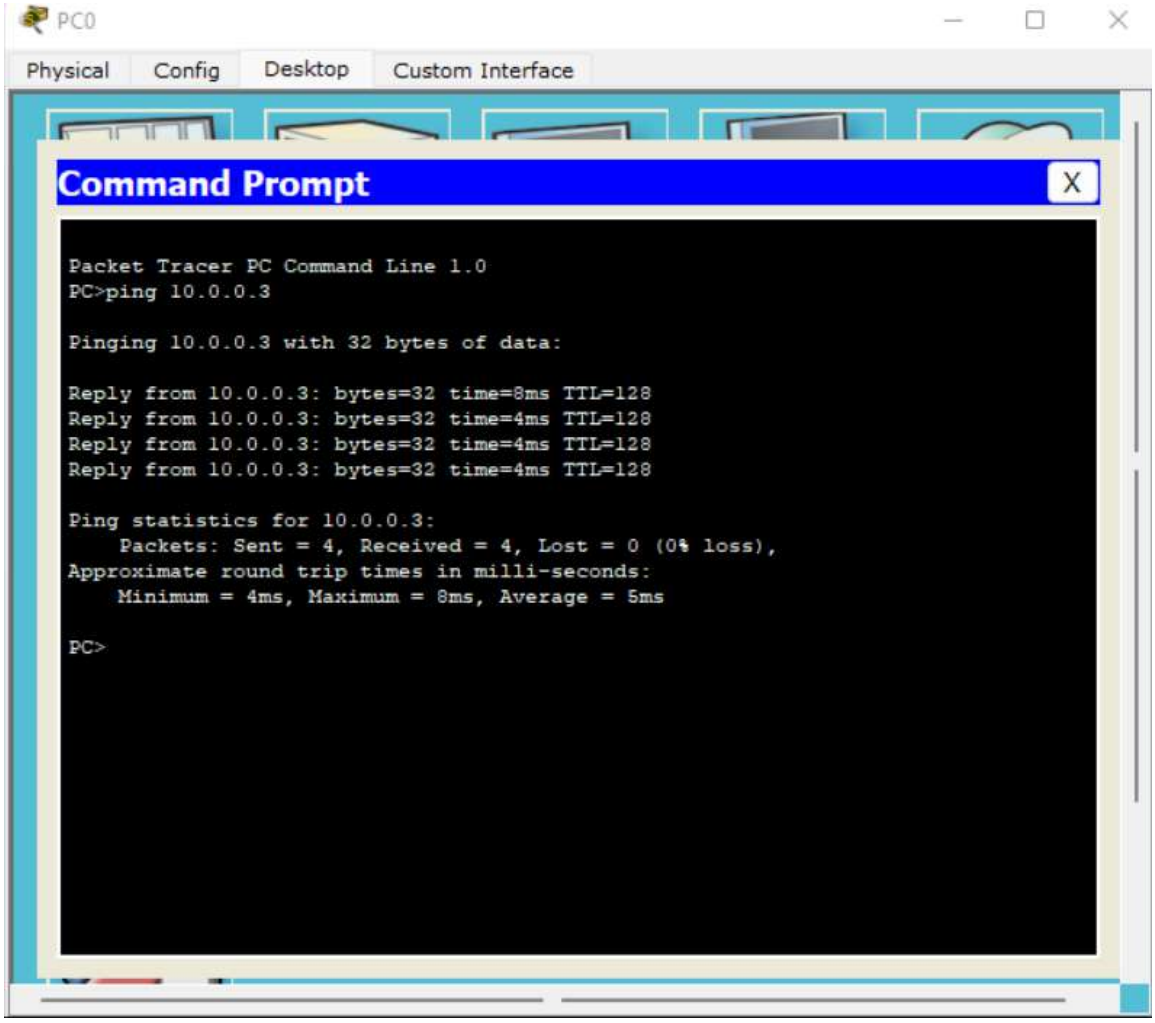
Minimum = 4ms, Maximum = 8ms, Average = 5ms

16/16

10/10

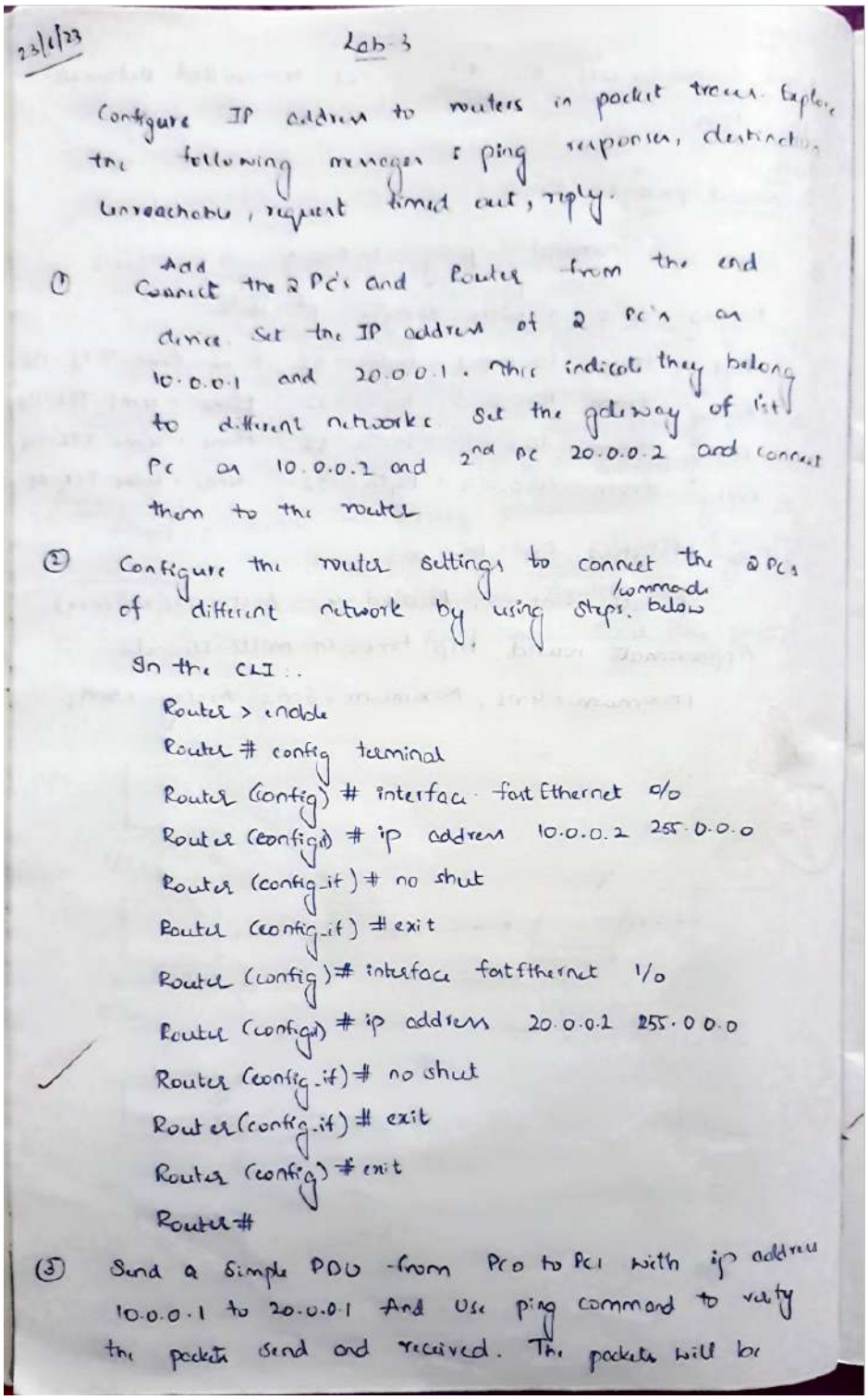

PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Hub-PT
Hub0
Switch-PT
Switch4
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5

Output


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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=8ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 8ms, Average = 5ms
PC>

PROGRAM -2

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



23/1/23

Lab 3

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

① Add and Connect the 2 PC's and Router from the end device. Set the IP address of 2 PC's as 10.0.0.1 and 20.0.0.1. This indicate they belong to different networks. Set the gateway of 1st PC as 10.0.0.2 and 2nd PC 20.0.0.2 and connect them to the router

② Configure the router settings to connect the 2 PCs of different network by using commands steps below

On the CLI..

```
Router > enable
Router # config terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface fastEthernet 1/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # exit
Router #
```

③ Send a Simple PDU from PC0 to PC1 with ip address 10.0.0.1 to 20.0.0.1 And Use ping command to verify the packets send and received. The packets will be

transmitted through the Router

④ Similarly, Connect two more PC's and Router. Config the Router following the above mentioned steps. Now, Add another generic Router to connect these two existing routers. Configure the 3rd Router following the same steps as above

⑤ Now In Command prompt from PC0

ping 30.0.0.1

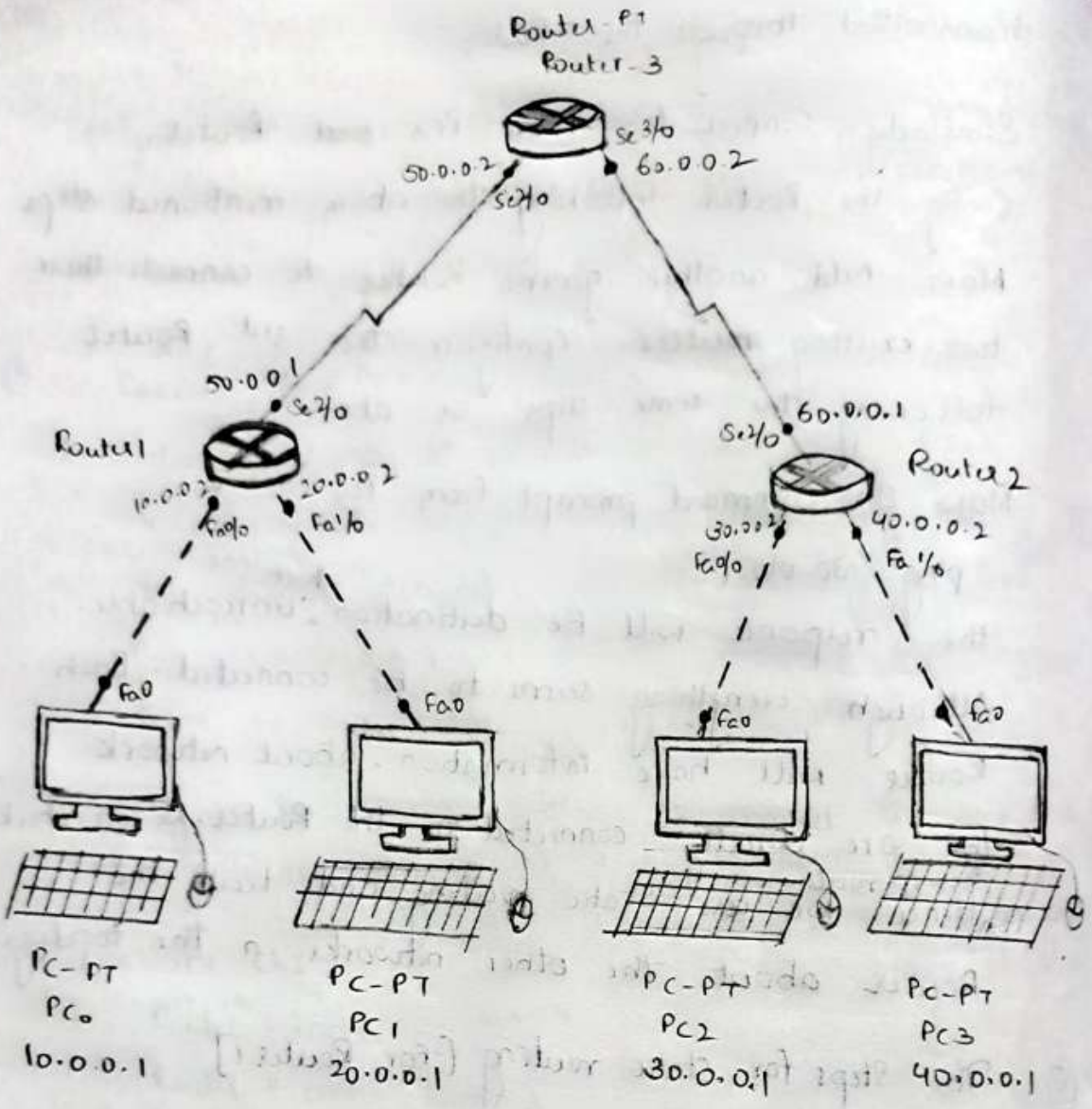
the response will be destination host unreachable. Although, everything seems to be connected. Each Router will have information only about network that are directly connected to the Router. We can check using cmd show ip route. Hence we use Static routing and teach the Router about the other networks in the topology

⑥ The Steps for Static routing [For Router1]

```
Router # config t
Router (config) # ip route 30.0.0.0 255.0.0.0 90.0.0.2
Router (config) # ip route 40.0.0.0 255.0.0.0 50.0.0.2
Router (config) # ip route 60.0.0.0 255.0.0.0 50.0.0.2
Router (config) # exit
Router #
```

⑦ Now, if we view all the networks connected to a router: Router # show ip route we get

```
S   10.0.0.0/8 [ is directly connected, fastEthernet 0/0
C   20.0.0.0/8 is directly connected, fastEthernet 1/0
```



Output:

for Destination host unreachable:

Ping 0.0.0.1

Pinging 0.0.0.1 with 32 bytes of data:

Reply from 30.0.0.2: Destination host unreachable.

Reply from 30.0.0.2: Destination host unreachable

Reply from 30.0.0.2: Destination host unreachable

Reply from 30.0.0.2: Destination host unreachable

For Reply:

ping 30.0.0.1

pinging 30.0.0.1 with 32 bytes of data:

Reply from 30.0.0.1: bytes=32 time=7ms TTL=125

Reply from 30.0.0.1: bytes=32 time=11ms TTL=125

Reply from 30.0.0.1: bytes=32 time=2ms TTL=125

Reply from 30.0.0.1: bytes=32 time=4ms TTL=125

Ping statistics for 30.0.0.1:

packets: Sent=4, Received=4, Lost=0 (0% loss),

Approximate round trip times in milli-seconds:

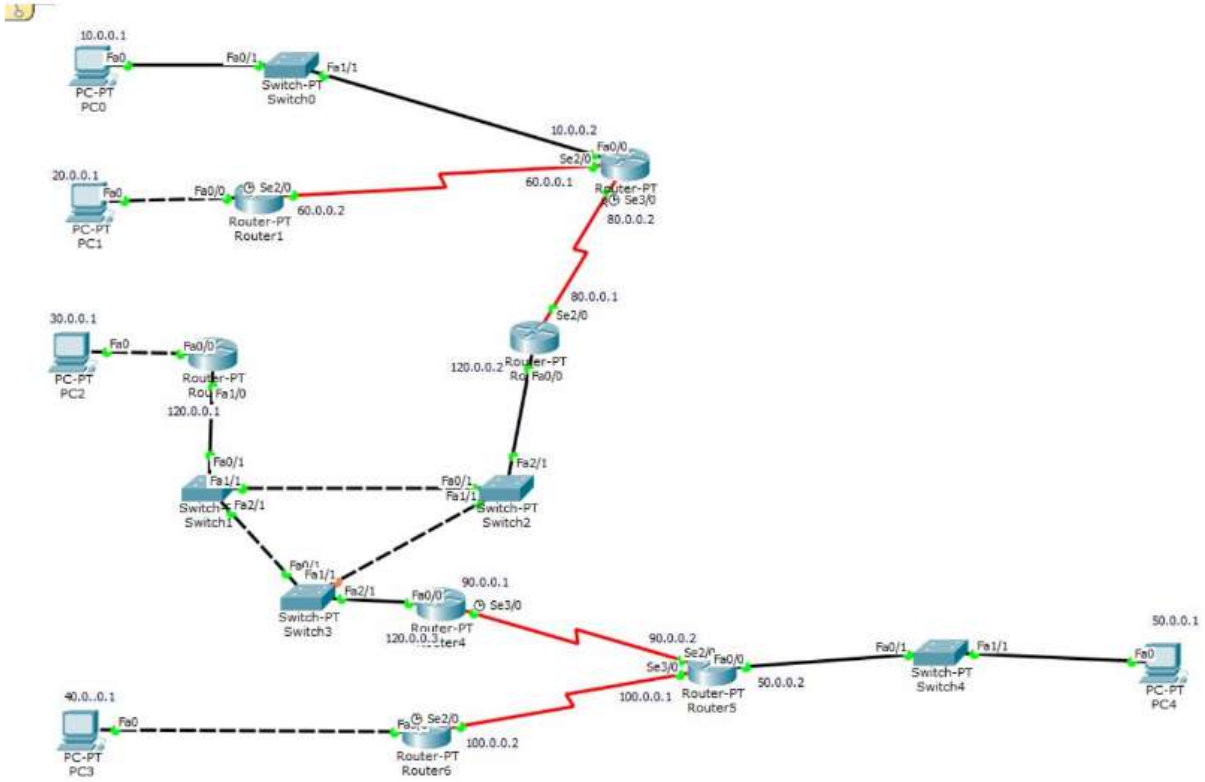
Minimum=2ms, Maximum=11ms, Average=6ms

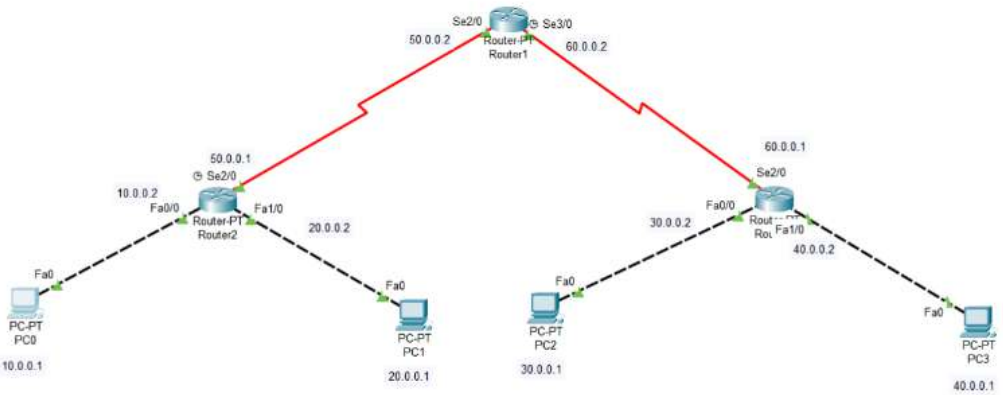
Good.

N
26/6/23

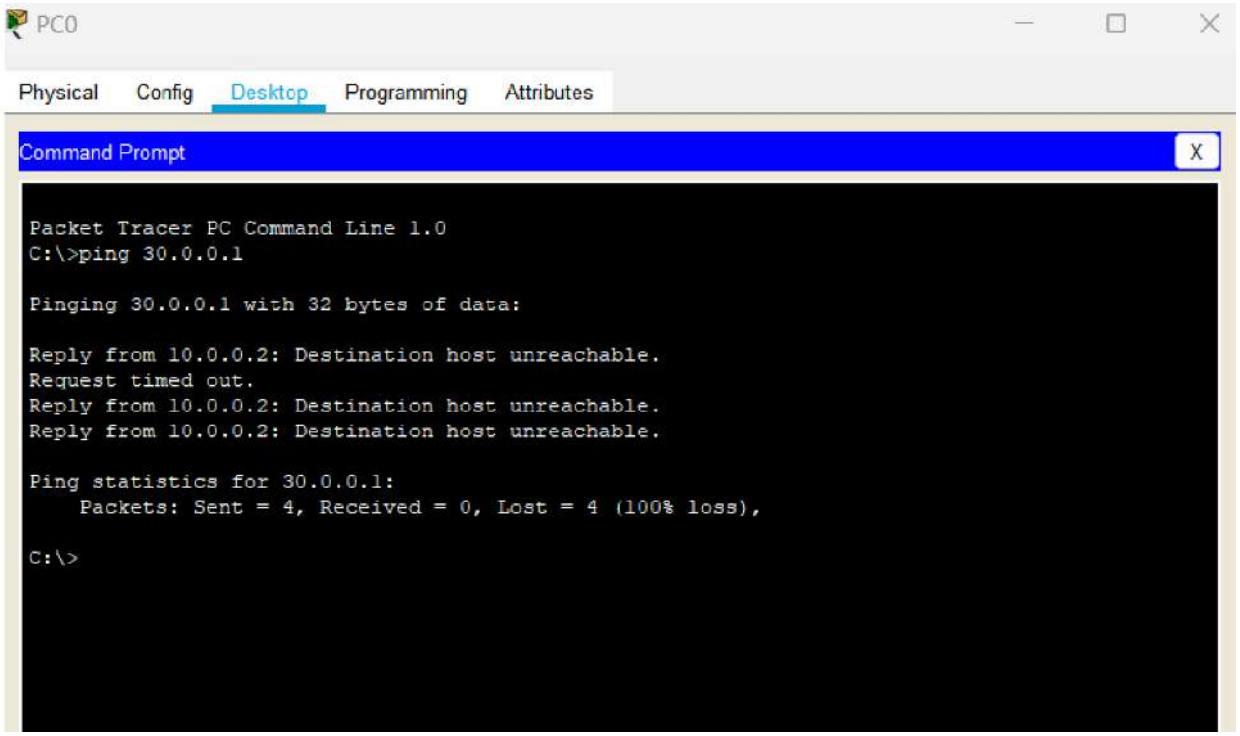
10/10

TOPOLOGY


10.0.0.1
Fa0
PC-PT
PC0
Fa0/1
Switch-PT
Switch0
Fa1/1
10.0.0.2
Fa0/0
Se2/0
60.0.0.1
20.0.0.1
Fa0
PC-PT
PC1
Fa0/0
Se2/0
Router-PT
Router1
60.0.0.2
Se3/0
80.0.0.2
80.0.0.1
Se2/0
30.0.0.1
Fa0
PC-PT
PC2
Fa0/0
Router-PT
Fa1/0
120.0.0.1
120.0.0.2
Fa0/0
Fa0/1
Fa1/1
Switch-PT
Switch1
Fa2/1
Fa2/1
Fa0/1
Fa1/1
Switch-PT
Switch2
Fa0/1
Fa1/1
Switch-PT
Switch3
Fa2/1
Fa0/0
90.0.0.1
Se3/0
Router-PT
90.0.0.2
Se2/0
Se3/0
Fa0/0
100.0.0.1
Router-PT
Router5
50.0.0.2
Fa0/1
Fa1/1
Switch-PT
Switch4
50.0.0.1
Fa0
PC-PT
PC4
40.0.0.1
Fa0
PC-PT
PC3
Se2/0
Router-PT
Router6
100.0.0.2


Se2/0
Se3/0
50.0.0.2
Router-PT
Router1
60.0.0.2
50.0.0.1
Se2/0
60.0.0.1
Se2/0
10.0.0.2
Fa0/0
Router-PT
Router2
Fa1/0
20.0.0.2
Fa0/0
30.0.0.2
Fa1/0
40.0.0.2
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
20.0.0.1
Fa0
PC-PT
PC2
30.0.0.1
Fa0
PC-PT
PC3
40.0.0.1

OUTPUT




```
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Reply from 30.0.0.1: bytes=32 time=7ms TTL=125
Reply from 30.0.0.1: bytes=32 time=11ms TTL=125
Reply from 30.0.0.1: bytes=32 time=2ms TTL=125
Reply from 30.0.0.1: bytes=32 time=4ms TTL=125

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 11ms, Average = 6ms

PC>
```

PROGRAM -3

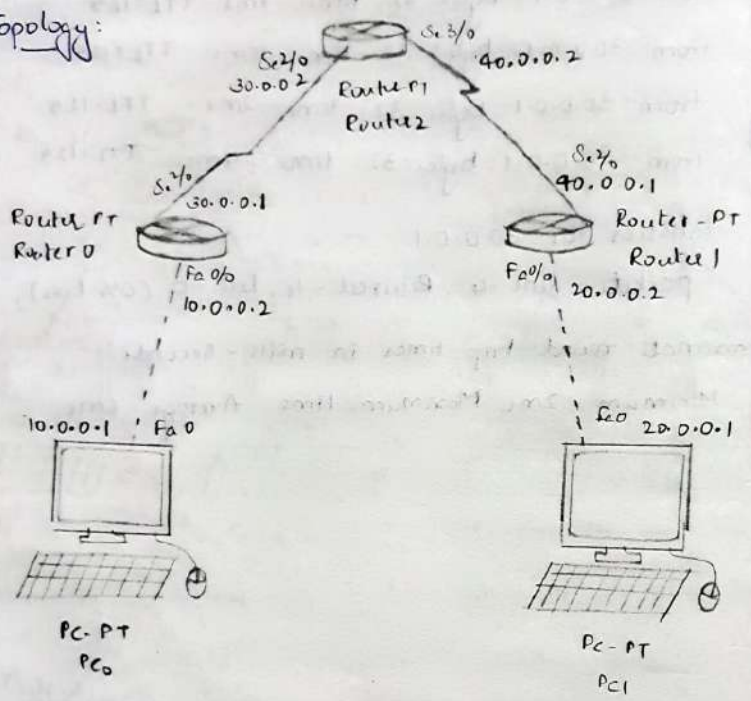
Configure default route, static route to the Router

30/6/23

Lab-4

Aim : Configure default route, Static route to the Router

Topology:



Procedure:

① Drag and drop 2 Pc's and 3 Routers from the end devices. Connect the 1 router for each of the Pc's [Pc0 and Pc1] And connect the 3rd Router to the other two Router as shown in the topology.

② Set the IP address of Pc0 as 10.0.0.1 and IP address of Pc1 as 20.0.0.1; Set the gateway as 10.0.0.2 and 20.0.0.2 for two Pc's respectively.

③ Now config the ip addrss of ports in Router 0 & Router 1 using the following commands.

```
Router > enable
Router # config t
Router (config) interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface Serial 2/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # exit
```

These are the commands for Router 0. Similarly Router 1 and Router 2 need to be configured.

④ As Router 0 and Router 1 are connected to only one side we perform default routing. using following CLI commands:

For Router 0

```
Router > enable
Router # config t
Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.2
```

For Router 1

```
Router # config t
Router (config) # ip route 0.0.0.0 0.0.0.0 40.0.0.2
```

Router # config t

Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.2

Router (config) # ip route 20.0.0.0 255.0.0.0 40.0.0.2

Router (config) # exit

Router #

⑤ Now, Check the routing information

For Router 0

Router # show ip route

C - Connected S-Static * - Candidate default

Gateway of last resort is 30.0.0.2 to network 0.0.0.0

C 10.0.0.0/8 is directly connected, Fast Ethernet 0/0

C 30.0.0.0/8 is directly connected, Serial 2/0

S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 2:

Router # show ip route

C - connected S-Static

S 10.0.0.0/8 [1/0] via 30.0.0.1

S 20.0.0.0/8 [1/0] via 40.0.0.1

C 30.0.0.0/8 is directly connected, Serial 2/0

C 40.0.0.0/8 is directly connected, Serial 3/0

Ping Commands (Output):

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 = 18ms TTL = 125

Reply from 20.0.0.1 : bytes = 32 time = 17ms TTL = 125

Reply from 20.0.0.1 : bytes = 32 time = 25ms TTL = 125

ping statistics for 20.0.0.1 :

Packets : Sent = 4 , Received = 4 , Lost = 0 (0% Loss)

Approximate round trip times in milli-seconds :

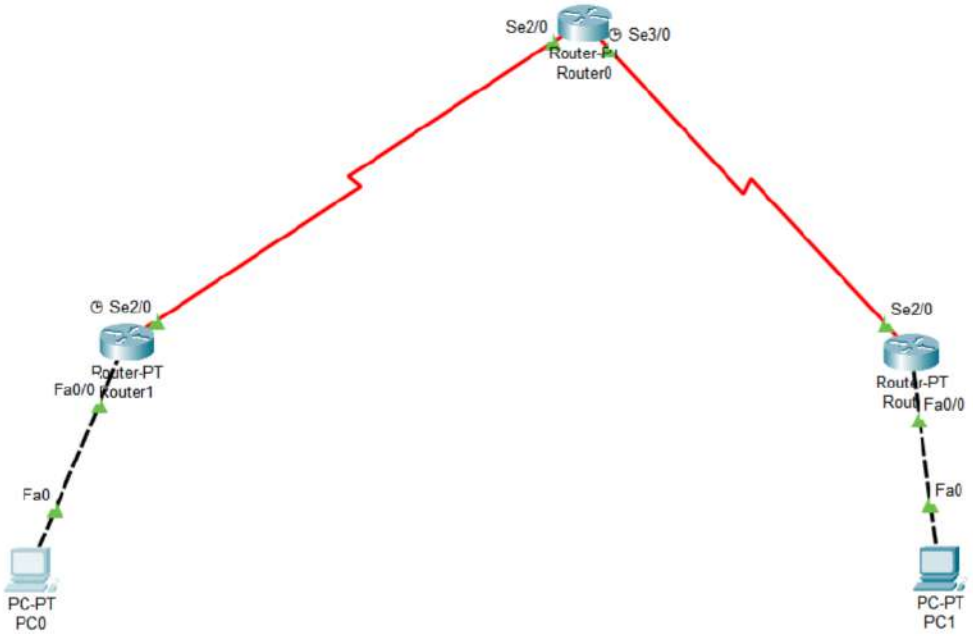
Minimum = 9ms , Maximum = 25ms , Average = 16ms

10/0

N

4/7/23

TOPOLOGY



OUTPUT

```
Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=2ms TTL=125
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=17ms TTL=125
Reply from 20.0.0.1: bytes=32 time=9ms TTL=125

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 17ms, Average = 9ms

C:\>
```

PROGRAM -4
Configure DHCP within a LAN and outside LAN

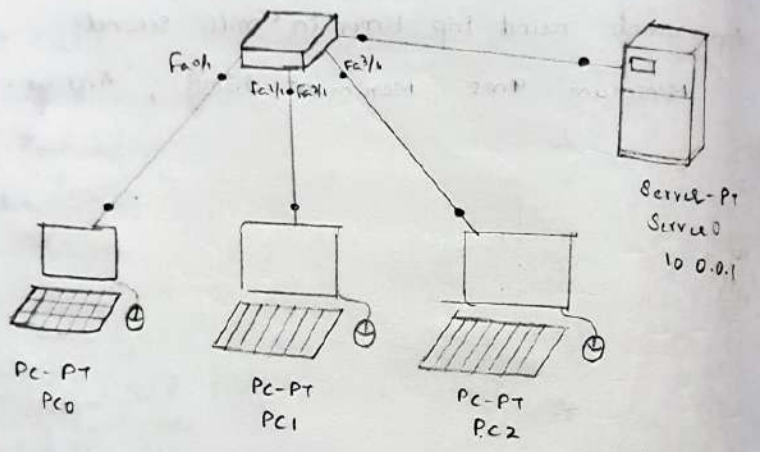
14/7/23

Lab 5

Aim: Configure DHCP within a LAN and outside LAN.

for DHCP within a LAN

Topology:



Procedure:

① Drag and drop 3 PC's and 1 Server, 1 switch from the devices. Connect the 3 PC's to the Switch.

② Connect the Server to the Switch. Give the IP address to the Server as 10.0.0.1

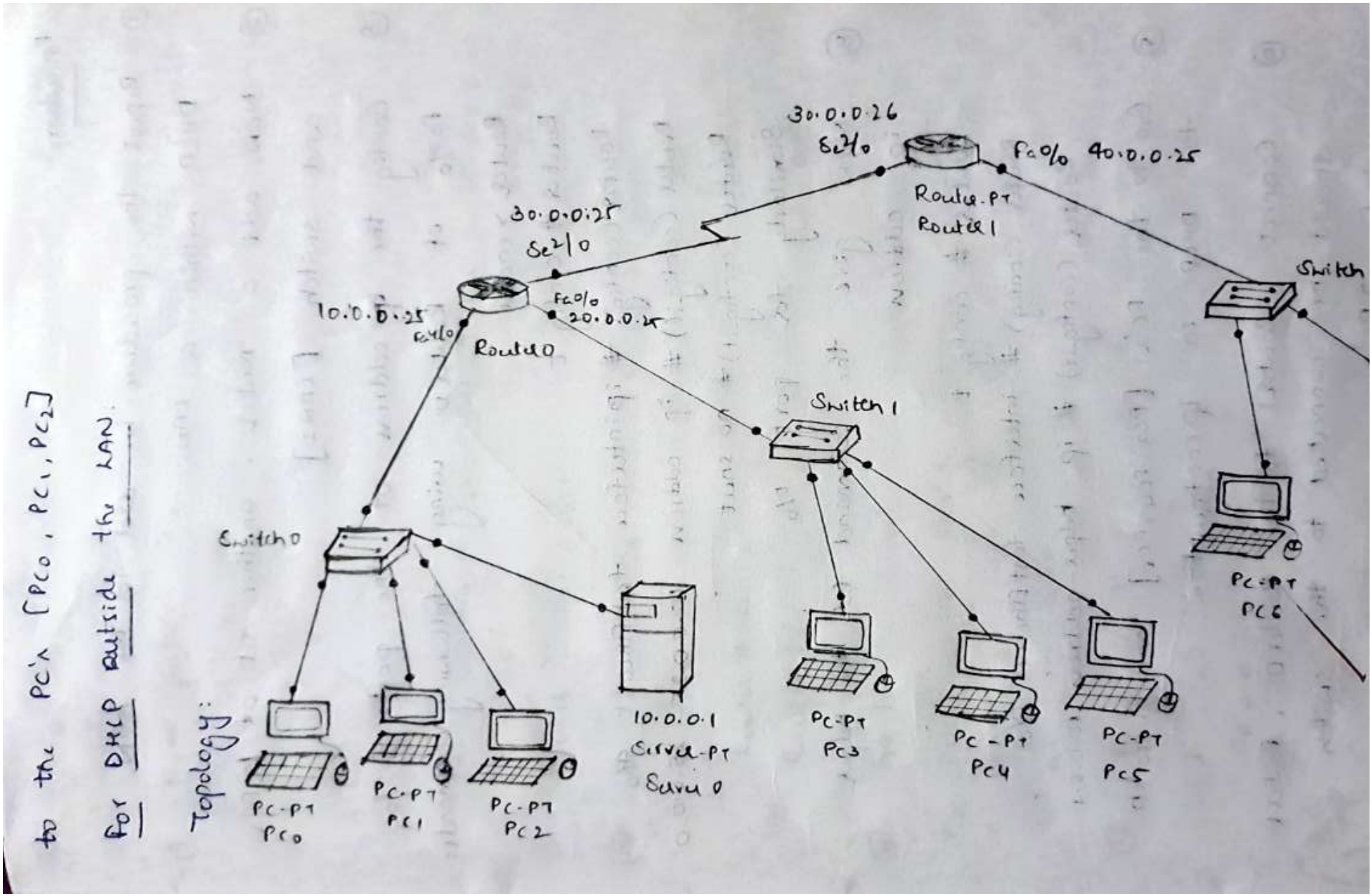
③ Go to the Services → DHCP, Add the pool with PoolName: ServerPool
Start IP Address: 10.0.0.2
Subnet Mask: 255.0.0.0

④ Now go to the PC's and switch to DHCP in IP configuration. IP address will be assigned

to the PC's [PC0, PC1, PC2]

For DHCP outside the LAN.

Topology:



Procedure:

① Repeat the procedure we did for the LAN. DHCP within a LAN.

② Now, add a router, another set of PC's and switch [LAN2]

③ Config the ip address of the ports fa4/0, fa0/0 of Router0 using following commands:

```
Router > enable
Router # config t
Router(config) # ipinterface fastEthernet 4/0
Router (config-if) # ip address 10.0.0.25 25.0.0.0
Router (config-if) # no shut
```

Similarly for port 0/0

④ Now give the command for the helper ip address:

```
Router # config t
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip helper-address 10.0.0.1
```

⑤ Go to the PC's [PC3, PC4, PC5] and g. switch to DHCP in ip configuration.

⑥ Connect Router1 to the Router0. Router1 should be connected to the switch and 2 PC's.

⑦ Config the ipaddress of Router1 with for

(8) Config the ip address of Router0 of Serial 2/0 with ip address 30.0.0.25

(9) Now perform the static routing to connect network 40 to the Router0 with the following commands:

```
Router >enable
Router #config t
Router (config) # ip route 40.0.0.0 255.0.0.0
                                    30.0.0.26
Router (config) # exit
```

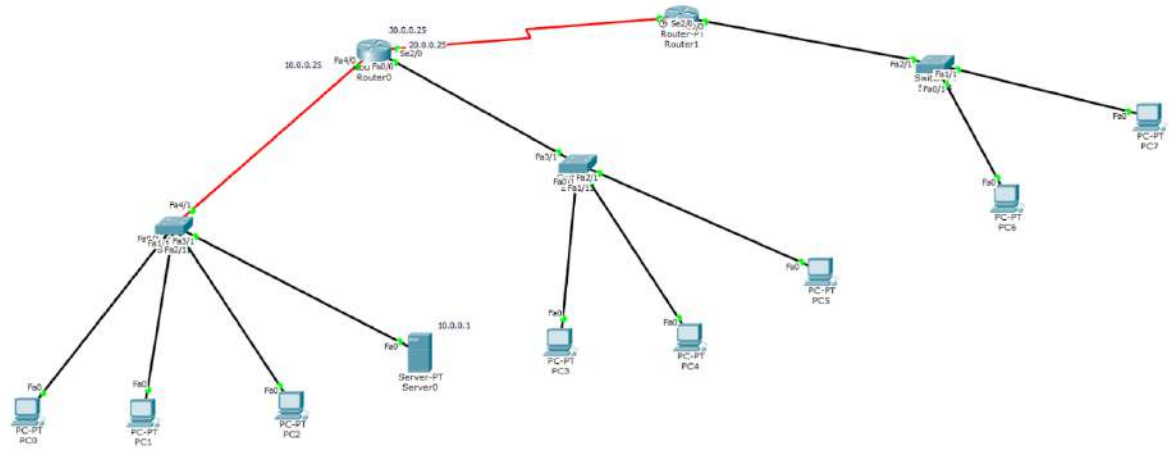
(10) Do the similar with Router1 with default or static routing

```
Router # config t
Router #(config)# ip route 0.0.0.0 0.0.0.0
                                    30.0.0.25
```

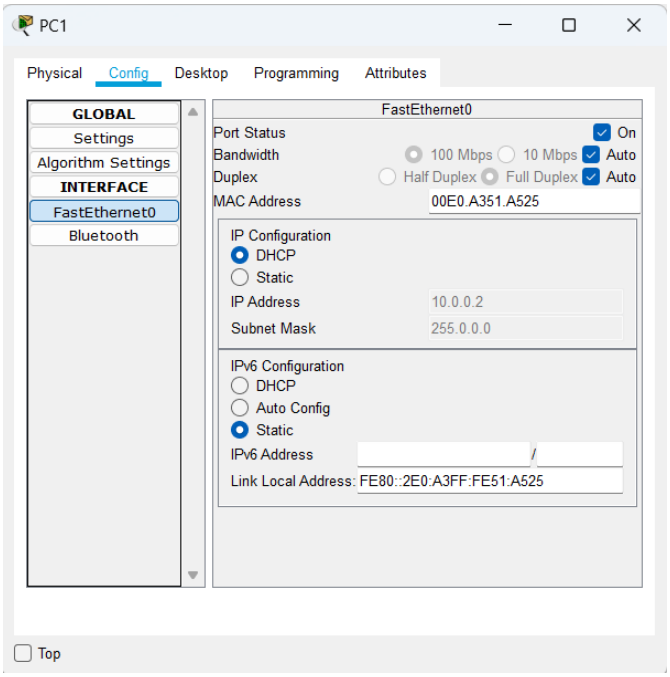
(11) Go to the Server and create 2 more pools with starting address 20.0.0.2 & 40.0.0.2. Give the Default Gateway as 10.0.0.25.

(12) Pc's in the network 40 with be provided with the ip address 40.0.0.2, 40.0.0.3. using DHCP.

10/10

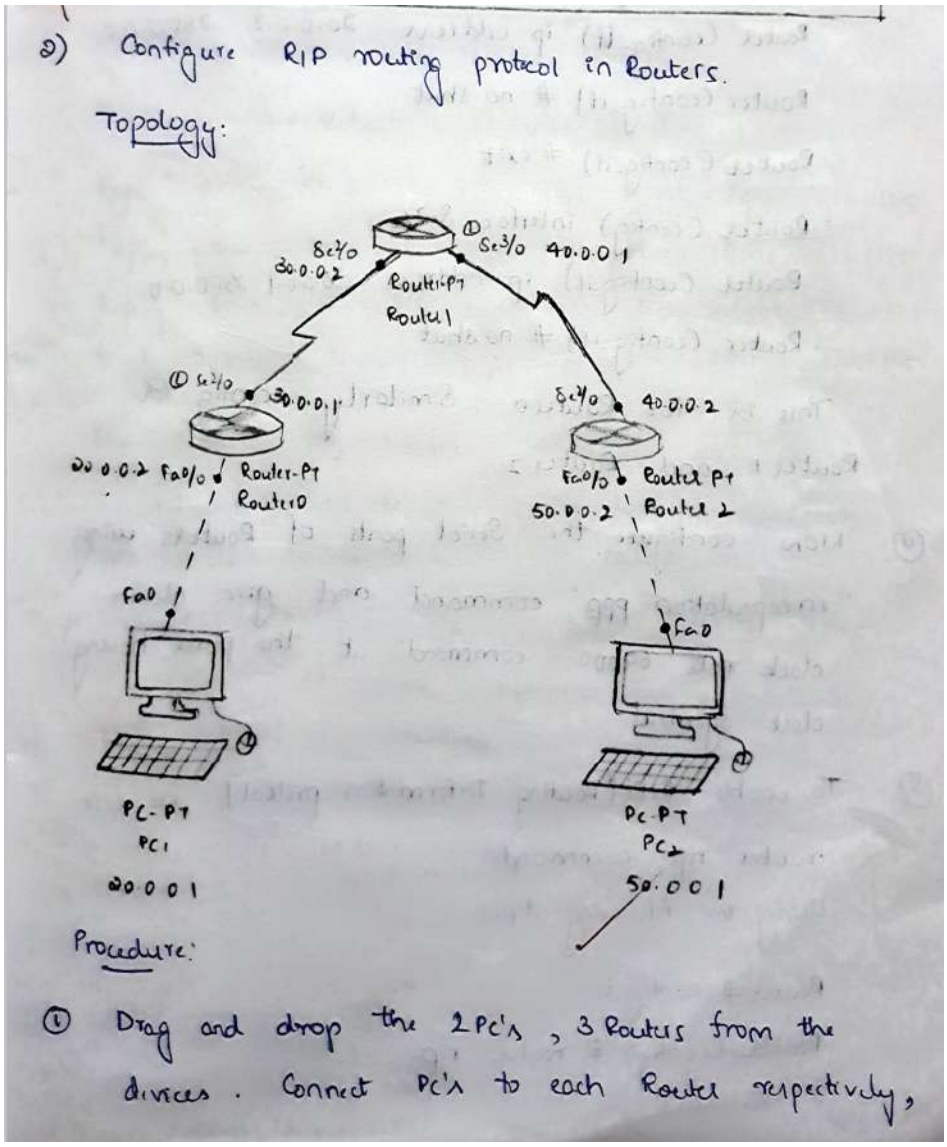

30.0.0.25
20.0.0.25
10.0.0.25
Router0
Router1
10.0.0.1
Server0
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7

OUTPUT


PC1
Physical
Config
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Bluetooth
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00E0.A351.A525
IP Configuration
DHCP
Static
IP Address
10.0.0.2
Subnet Mask
255.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2E0:A3FF:FE51:A525
Top

PROGRAM -5

Configure RIP routing Protocol in Routers



and those two Routers connected to another Router

② Configure IP address of Pc1 and Pc2 as 20.0.0.1 and 50.0.0.1 respectively.

③ Config. the Routers (All 3 routers) using following commands.

```
Router > enable
Router # config t
Router (config) interface fa0/0
Router (config-if) ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) interface Se2/0
Router (config-if) ip address 30.0.0.1 255.0.0.0
Router (config-if) # no shut
```

This is for Router0. Similarly config for Router1 and Router2

④ Now configure the Serial ports of Routers using "encapsulation ppp" command and give clock clock rate 64000 command at the ports having clock symbol

⑤ To enable RIP [Routing Information protocol] we use router rip command.

Using the following steps:

```
Router # config t
Router (config) # router rip
```

Router (config-router) # network 20.0.0.0

Router (config-router) # network 30.0.0.0

Similar should be done Router 1 & 2

(6) Now, ping Give the gateway to PC_1 as 20.0.0.2 as to PC_2 as 50.0.0.2

(7) Now, ping from PC_1 to PC_2 & check the results

Output:

Pc > ping 50.0.0.1

Pinging 50.0.0.1 with 32 bytes of data:

Reply from 50.0.0.1 : bytes = 32 time = 2ms TTL = 125

Reply from 50.0.0.1 : bytes = 32 time = 9ms TTL = 125

Reply from 50.0.0.1 : bytes = 32 time = 6ms TTL = 125

Reply from 50.0.0.1 : bytes = 32 time = 2ms TTL = 125

Ping statistics for 50.0.0.1:

Packets : Sent = 4, Received = 4, Lost = 0 (0% Loss),

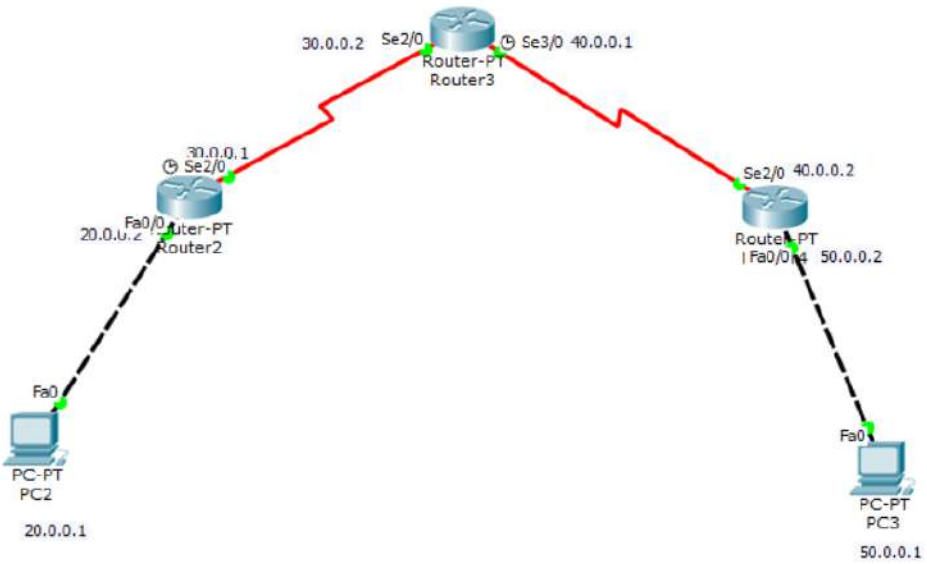
Approximate round trip time in milli-seconds:

Minimum = 2ms, Maximum = 9ms, Average = 4ms

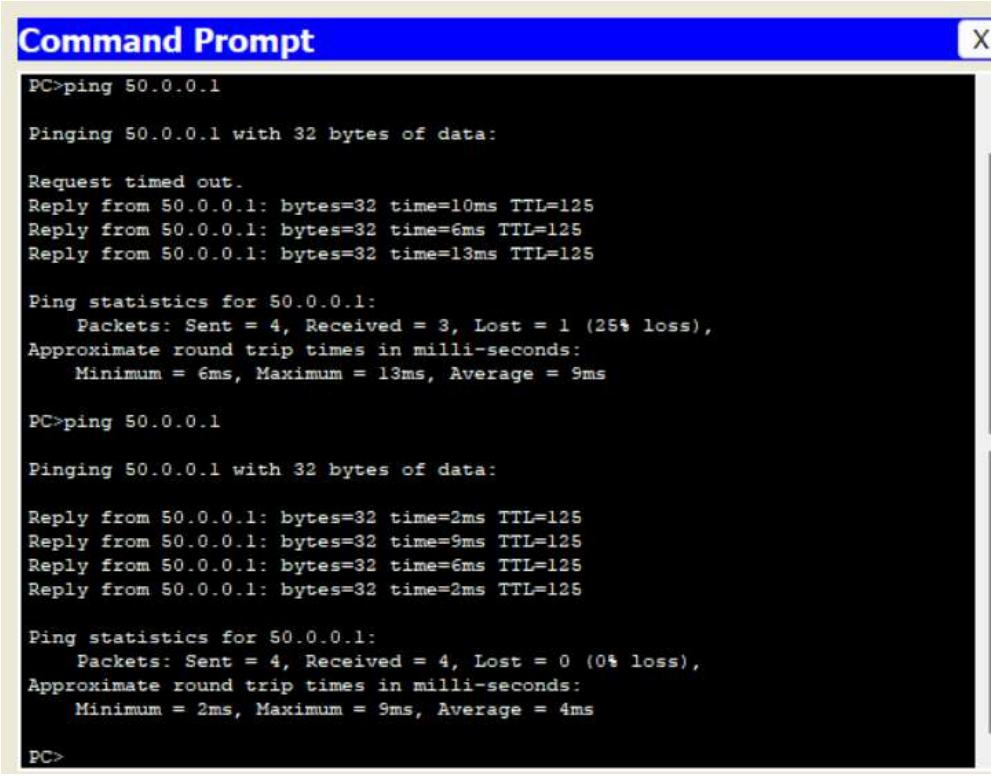
10/10

NL

25/7/23


30.0.0.2
Se2/0
Router-PT
Router3
Se3/0
40.0.0.1
30.0.0.1
Se2/0
Fa0/0
Router2
Se2/0
40.0.0.2
Router-PT
Fa0/0
50.0.0.2
Fa0
PC-PT
PC2
20.0.0.1
Fa0
PC-PT
PC3
50.0.0.1

OUTPUT


Command Prompt
PC>ping 50.0.0.1
Pinging 50.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 50.0.0.1: bytes=32 time=10ms TTL=125
Reply from 50.0.0.1: bytes=32 time=6ms TTL=125
Reply from 50.0.0.1: bytes=32 time=13ms TTL=125
Ping statistics for 50.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 13ms, Average = 9ms
PC>ping 50.0.0.1
Pinging 50.0.0.1 with 32 bytes of data:
Reply from 50.0.0.1: bytes=32 time=2ms TTL=125
Reply from 50.0.0.1: bytes=32 time=9ms TTL=125
Reply from 50.0.0.1: bytes=32 time=6ms TTL=125
Reply from 50.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 50.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 9ms, Average = 4ms
PC>

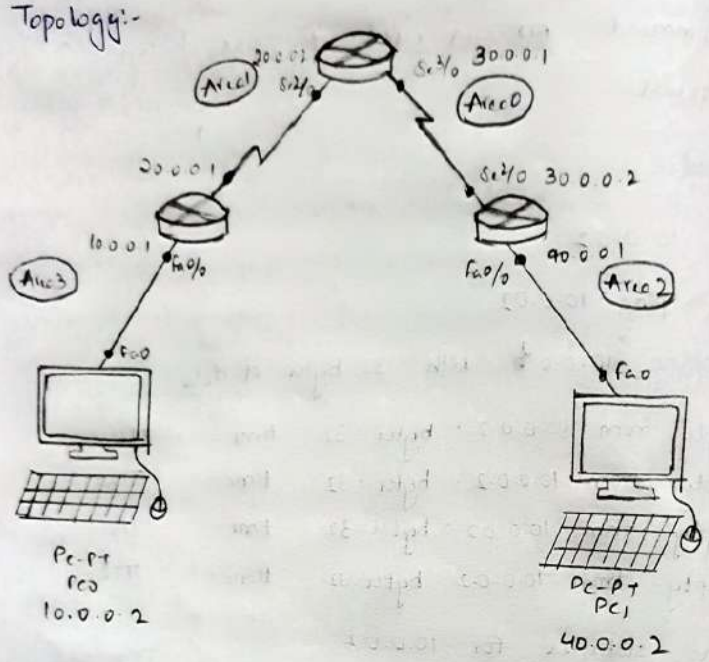
PROGRAM -6

Configure OSPF routing protocol

18/8/23

Aim: Configure OSPF routing protocol & connect Area

Topology:-



Procedure:

① Create the topology as given above with 2 PC's & 3 routers.

② Configure the ip address for PC's as 10.0.0.2 & 40.0.0.2 respectively.

③ Configure the ip address routers with ip address for all interfaces.

④ Now, for all Serial port of Routers configure using command "encapsulation ppp" & give "clock rate 64000" command at ports having clock symbol.

Eg:- For Router 2

router(config)# interface Serial 2/0

router (config-if) # ip ~~address~~ encapsulation ppp
router (config-if) # no shut
router (config-if) # exit

router (config) # interface Se3/0
router (config-if) # encapsulation ppp
router (config-if) # clock rate 64000
router (config-if) # no shut

⑤ Now, Enable ip routing by configuring ospf routing protocol in all routers,

On Router R_1,

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

Similarly configure for Router 2 and Router 3

⑥ Check the routing table of R1

Router # show ip route

Codes: C- connected, S-static, R-RIP, M-mobile, B-BGP
O- OSPF, IA-OSPF inter area N1-OSPF NSSA
extend type 1

Gateway of last resort is not set

C 10.0.0.0/8 is directly connected, fastEthernet 2/0
C 20.0.0.0/8 is directly connected, Serial 2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:04:23, Serial 2/0

(7) There must be one interface up to keep ospf process up. So its better to configure loopback address to routers. It is virtual interface never goes down once we configured.

For Router 1 :-

```
Router (config-if) # interface loopback 0
Router (config-if) # ip add 172.16.1.252 255.255.0.0
Router (config-if) # no shut
```

Do similarly for Router 2 & Router 3. Using these commands we add loopback address to the routers.

(8) Now, if we check the routing table for R_3.

R_3 # show ip route

```
Codes : C - connected, S-static, O-OSPF, IA-OSPF
Gateway of last resort is not set
O IA   20.0.0.0/8 [110/128] via 30.0.0.1, 00:18:58, Serial 2/0
C      40.0.0.0/8 is directly connected, FastEthernet 0/0
C      30.0.0.0/8 is directly connected, Serial 2/0
```

R_3 (Router 3) doesnot know about area 3, so we will create virtual link between Router 1 & Router 3

(9) Now, we have to create virtual link between Router 2, by this we create a virtual link to connect area 3 to area 0

In Router R_2:-

Router (config) # router ospf 1

Router (config-router) # area 1 virtual link 1.1.1.1

Router (config-router) # exit

(10) R_2 and R_3 get updates about area 3,

check routing table for R_3

Router # show ip route

Codes : C- connected OA-OSPF, IA-OSPF

Gateway of last resort is not set

O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:01:56, Gig 2/0

C 40.0.0.0/8 is directly connected, fast ethernet 0/0

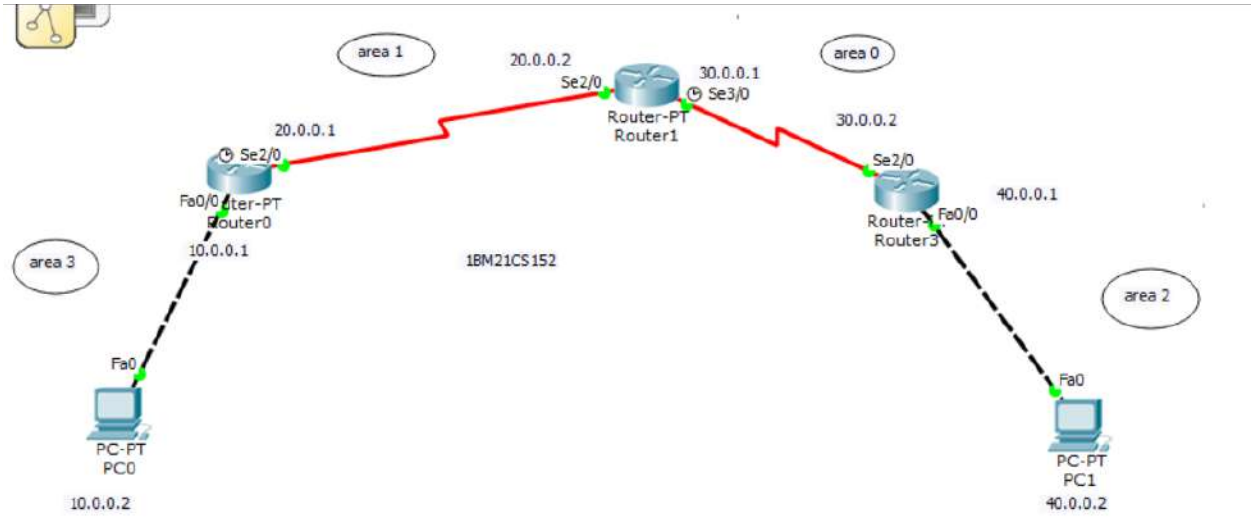
O IA 10.0.0.0/8 [110/129] via 30.0.0.1 00:01:56, Gig 2/0

C 30.0.0.0/8 is directly connected, Gig 2/0

(11) Now, ping 10.0.0.2 to 40.0.0.2

10/10

22/8/23


area 1
20.0.0.2
Se2/0
30.0.0.1
Se3/0
area 0
Router-PT
Router1
20.0.0.1
Se2/0
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Router-PT
Router0
Router-PT
Router3
Fa0/0
10.0.0.1
area 3
1BM21CS152
area 2
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
10.0.0.2
40.0.0.2

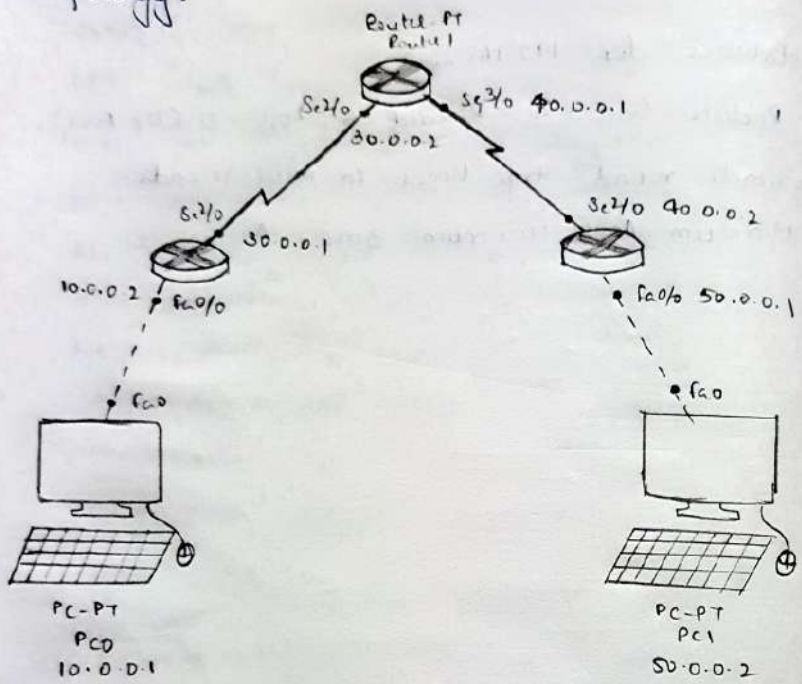
PROGRAM -7

Demonstrate the TTL/ Life of a Packet

11/6/24

Aim : Demonstrate the TTL / Life of a packet

Topology:



Procedure:

1. Create a topology as shown above with 2 PCs and 3 routers
2. Configure the IP address of PC0 and PC1 as 10.0.0.1 , 50.0.0.2 respectively.
3. Configure the IP address of routers using following commands.

```
Router # config t
Router (config) # interface fa0/0 10.0.0.2 255.0.0.0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router # exit
```

④ Configure the routers using default / static routing.

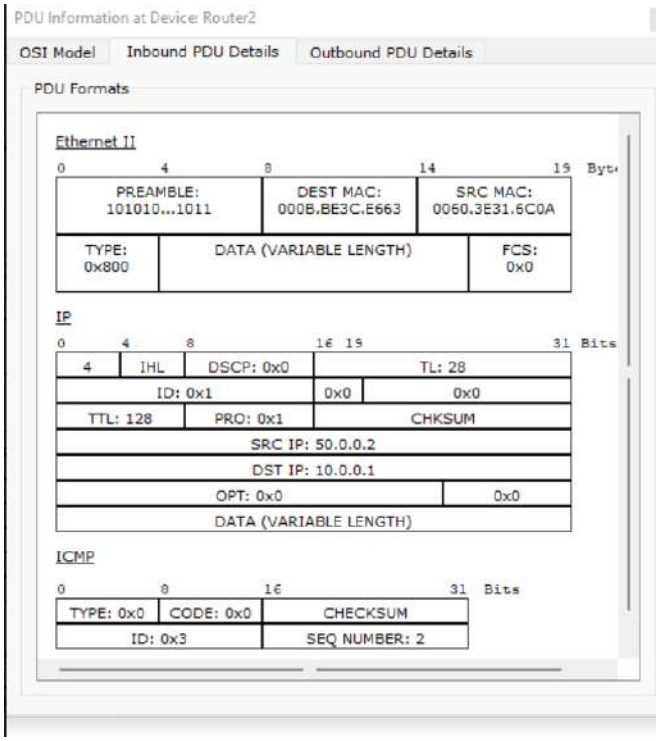
⑤ On simulation mode, send a simple PDU from one PC to another

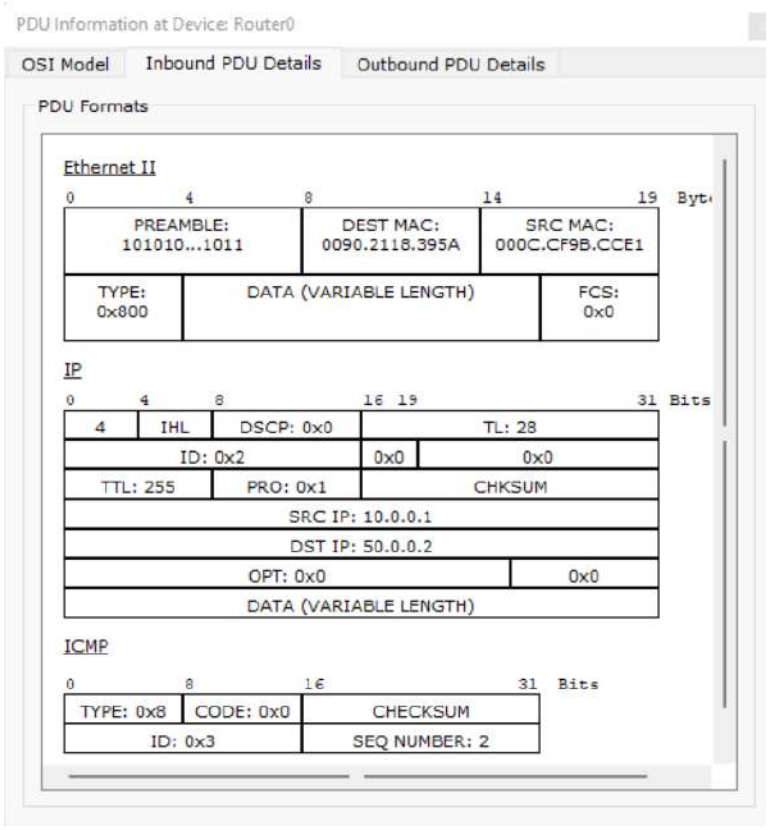
⑥ Use capture button to capture every transfer.

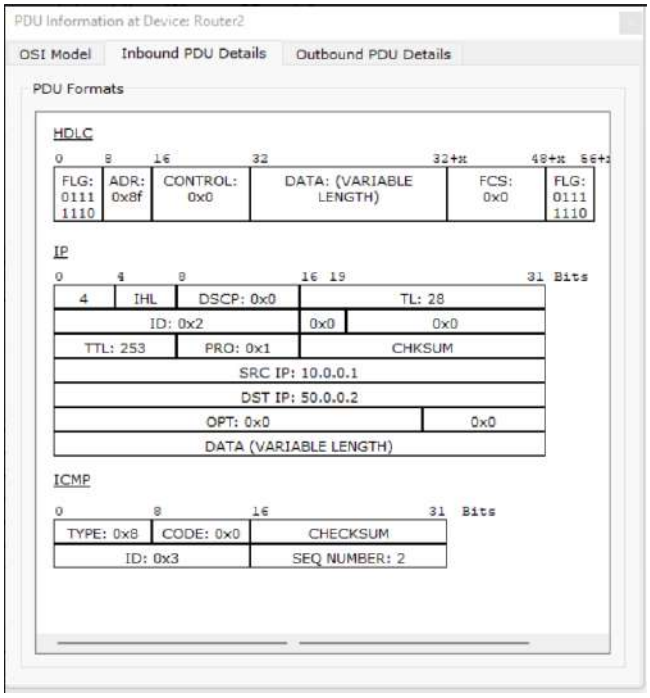
⑦ Click on PDU during every transfer to see the Inbound & outbound PDU details

10/10

18/23




PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0090.2118.395A
SRC MAC: 000C.CF9B.CCE1
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x2
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 50.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x3
SEQ NUMBER: 2


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x2
0x0
0x0
TTL: 253
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 50.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x3
SEQ NUMBER: 2

PROGRAM -8
Configure Web Server, DNS within a LAN.

21/7/23

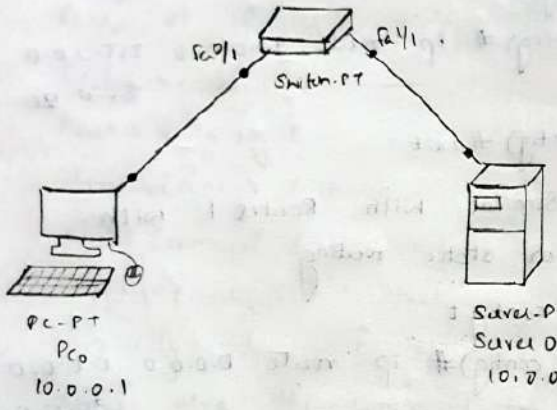
Lab-6

Aim :) Configure Webserver, DNS within a LAN

2) Configure RIP routing protocol in Routers

1)

Topology:



Procedure:

① Drag & drop the PC, Server, Switch from the devices

② Create the topology as above.

③ Configure the IP address of PC0 as 10.0.0.1

④ Configure the IP address of Server as 10.0.0.25

⑤ Open the webbrowser in PC0 and give ip address of Server.

⑥ Now, Go to the DNS in the Server and add name, URL Eg: hello, 10.0.0.25 respectively

⑦ Now, try with the name in webbrowser the index.html will be rendered.

(8) We can edit the index.html by clicking on https in Services section of Server0.

Output:

Cisco Packet Tracer

Welcome to cisco packet tracer

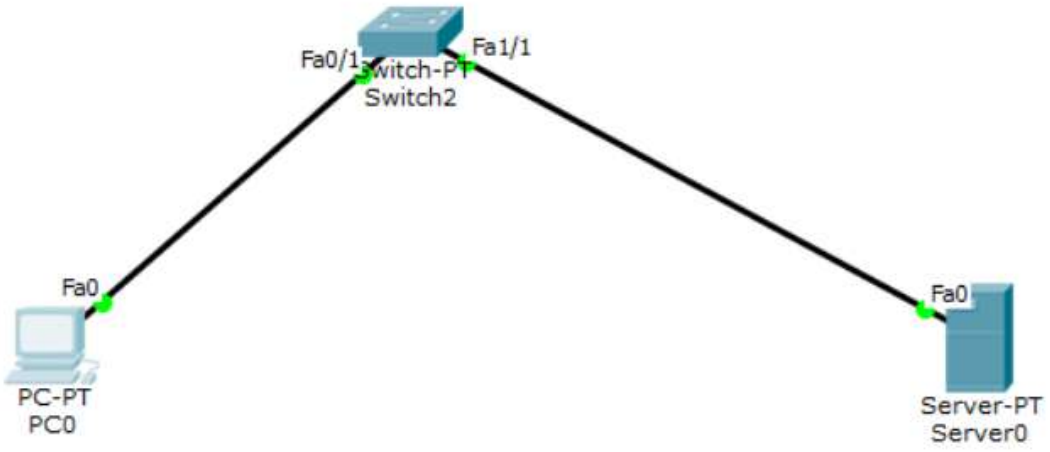
Quick Links:

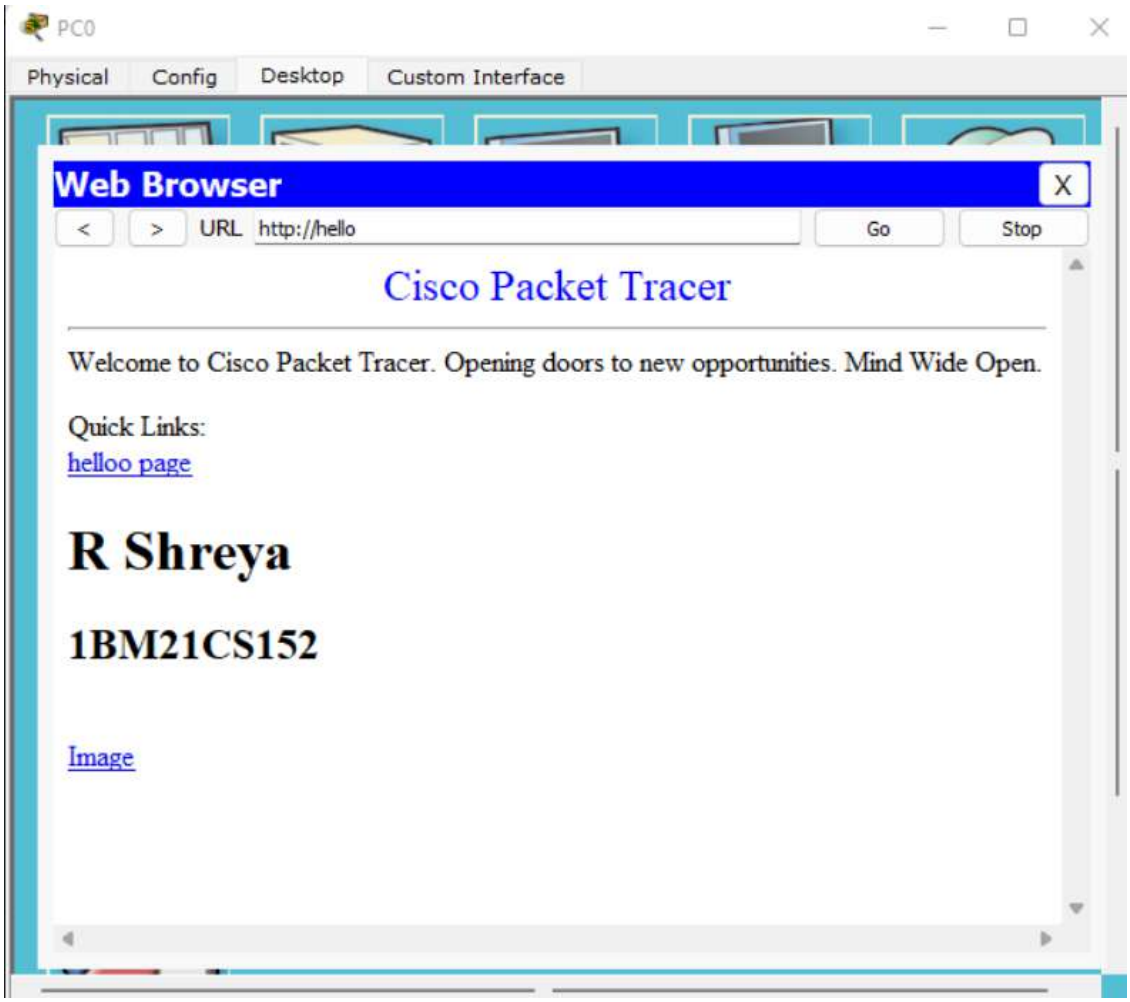
R. Shreya

1BM21CS152

10/10

29/7/23




PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL
http://hello
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
helloo page
R Shreya
1BM21CS152
Image

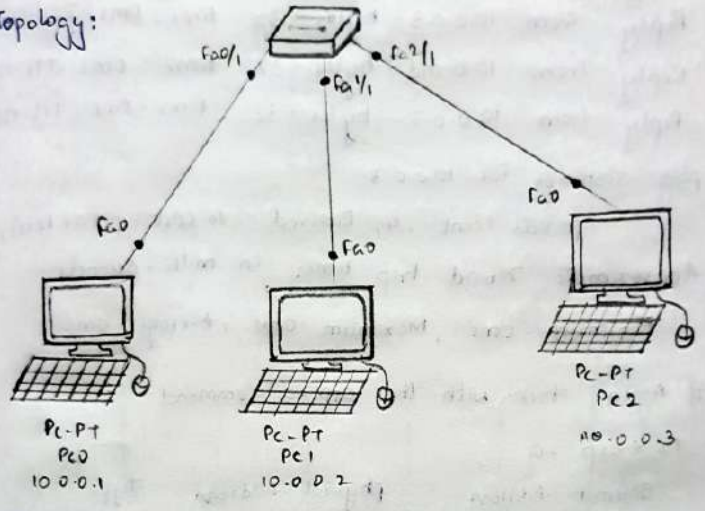
PROGRAM -9

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

4/8/23

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

Topology:



Procedure:

1. Drag and drop 3 PC's and 1 Switch from the devices.
2. Connect the devices in the topology as shown above
3. Config the IP address for the PC's PC_0, PC_1, PC_2 as 10.0.0.1, 10.0.0.2, 10.0.0.3 respectively.
4. Now, in CLI use the command "arp -a" to see ARP Table, Initially the ARP Table will be empty.
5. Also in CLI of Switch, the command - show mac address-table can be given on every transaction to see how the Switch learns from transactions and build the address table.

⑥ Now ping from one PC to another PC

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 = 0ms 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:

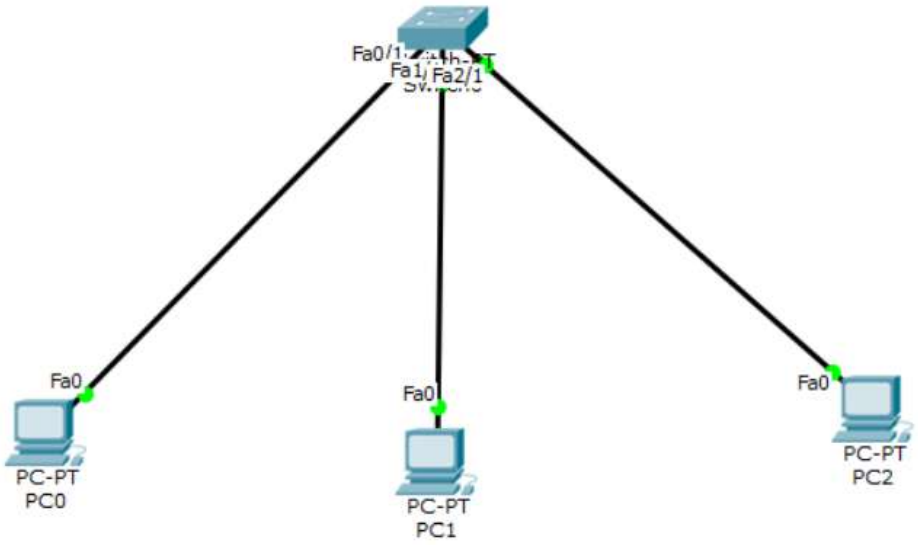
Minimum = 0ms, Maximum = 0ms, Average = 0ms

⑦ Again check with the arp -a command:

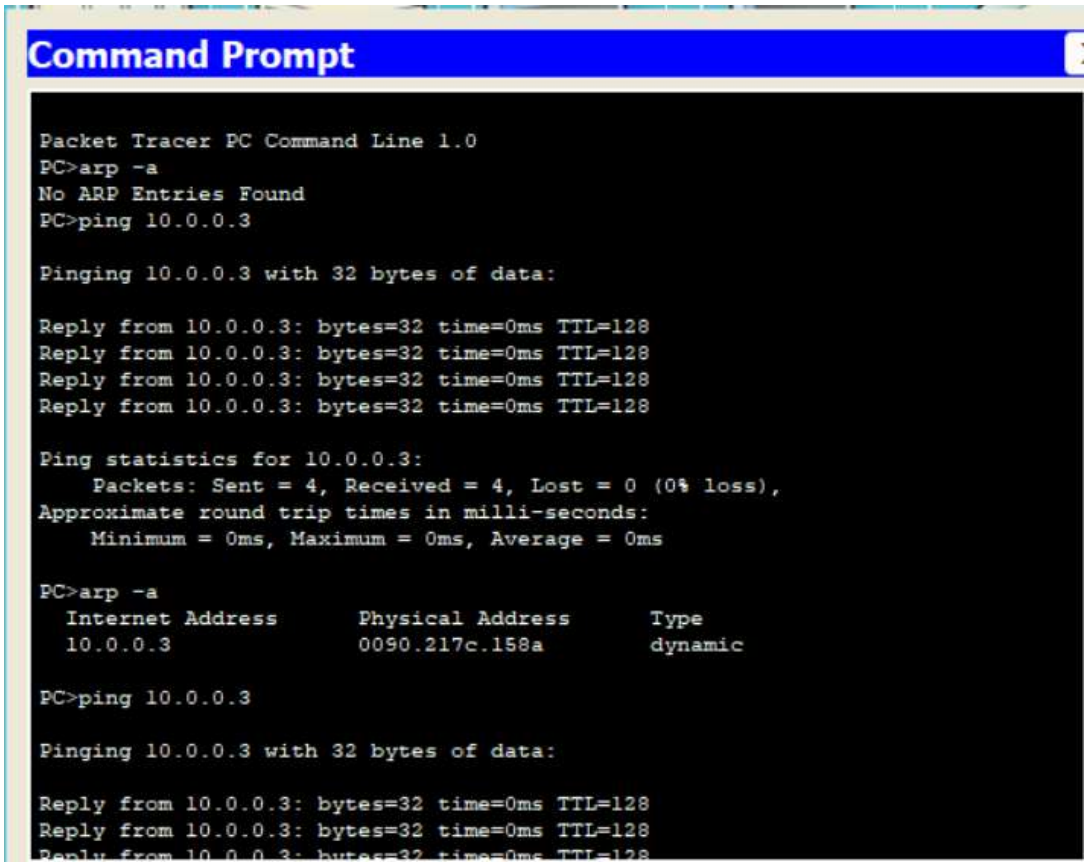
PC> arp -a

Internet Address	physical Address	Type
10.0.0.3	0090.217c.158a	dynamic

⑧ arp -d command is used to clear the table.


Fa0/1
Fa1
Fa2/1
Fa0
PC-PT
PC0
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2

OUTPUT


Command Prompt
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
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=0ms 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 = 0ms, Average = 0ms
PC>arp -a
Internet Address Physical Address Type
10.0.0.3 0090.217c.158a dynamic
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

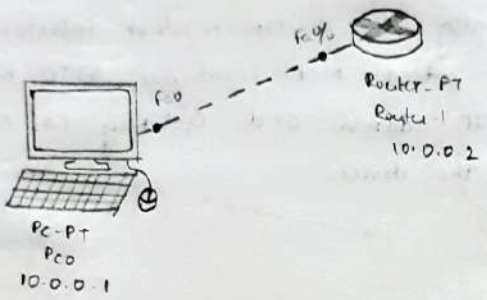
PROGRAM -10

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

13/8/23

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

Topology:



Procedure:

1. Create the topology as above
2. Give IP address of PC0 as 10.0.0.1 & config the router ip address as 10.0.0.2
3. Give the Gateway as 10.0.0.2 in the PC0
4. Now, in the Router give the following commands to create hostname, password.

```
Router > enable
Router # config
Router (config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # interface fastethernet 0/0
r1 (config-if) # ip address 10.0.0.1 255.0.0.0
r1 (config-if) # no shut
r1 (config-if) # line vty 0 5
r1 (config-line) # login
```

* R1 (config-line) # password p0
R1 (config-line) # exit
R1 (config) # exit

⑤ Command wr is used to save changes in router

Output:-

Ping 10.0.0.2

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 = TTL =
Reply from 10.0.0.2 : bytes = 32 time = TTL =
Reply from 10.0.0.2 : bytes = 32 time = TTL =
Reply from 10.0.0.2 : bytes = 32 time = TTL =

Ping statistics for 10.0.0.2 :

Packets : Sent = 4 , Received = 4 , Lost = 0 (0% loss),

Approximate round trip times in milli-seconds :

Minimum = 0 , Maximum = , Average =

Pc > telnet 10.0.0.2

Trying 10.0.0.2 ... Open

User Access Verification

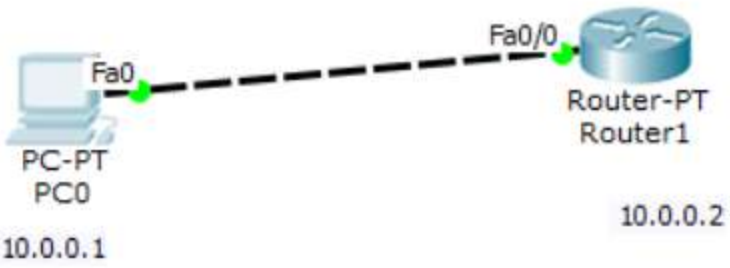
10/10 Password : P0

R1 > enable

Password : P1

22/8/23 R1 # show ip route

Codes : C - connected , S - static, I - IGRP, R - PIP, M - mobile



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

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

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms 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 = 1ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

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

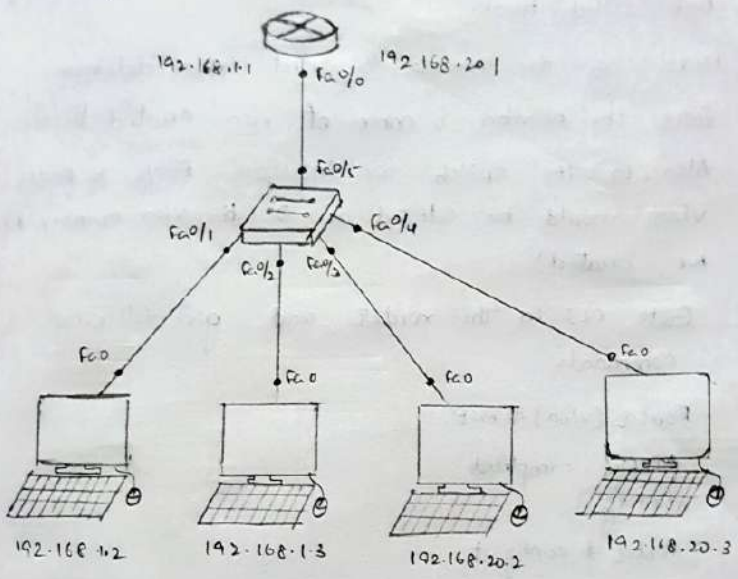
PROGRAM -11

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

11/8/22

Aim: To construct a VLAN and make the Pc's communicate among a VLAN

Topology:



Procedure:

1. Create the topology as given above.
2. Configure IP address for PC's as 192.168.1.2, 192.168.1.3, 192.168.20.2, 192.168.20.3, respectively.
3. Configure the IP address for the router using following commands:

```
Router > enable
Router # config t
Router (config) interface fa0/0
Router (config-if)# ip address 192.168.1.1 255.255.255.0
Router (config-if) # exit
```

4. On the switch go to VLAN Database & create/add

new vlan database.

(5) Now, go to Interface fastEthernet 0/5 & make it trunk, in vlan everything need to be select &. This allows different VLAN's over single link called trunk.

(6) Next, go to router & select vlan database enter the number & name of vlan created before Also, in the switch for interface fa0/3 & fa0/4 vlan should be selected as 2 (the vlan number which we created)

(7) Goto CLI in the router and give following commands..

```
Router (vlan) # exit
APPLY completed.
Exiting..
Router # config t
Router (config) # interface fastEthernet 0/0.1
Router (config-subif) # encapsulation dot1q 2
Router (config-subif) # ip address 192.168.20.1
                                   255.255.255.0
Router (config-subif) # no shut
Router (config-subif) # exit
```

Output:-

In command prompt

PC> ping 192.168.20.2

Reply from 192.168.20.2 : bytes = 32 time = 0ms TTL = 127
Reply from 192.168.20.2 : bytes = 32 time = 3ms TTL = 127
Reply from 192.168.20.2 : bytes = 32 time = 1ms TTL = 127

Ping statistics for 192.168.20.2:

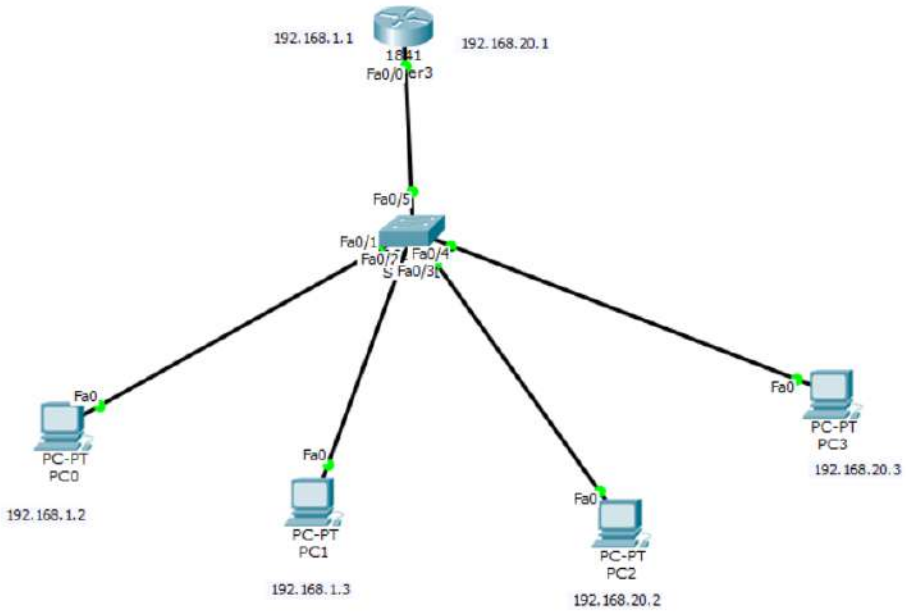
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:-

Minimum = 0ms, Maximum = 3ms, Average =

10/10

22/8/23

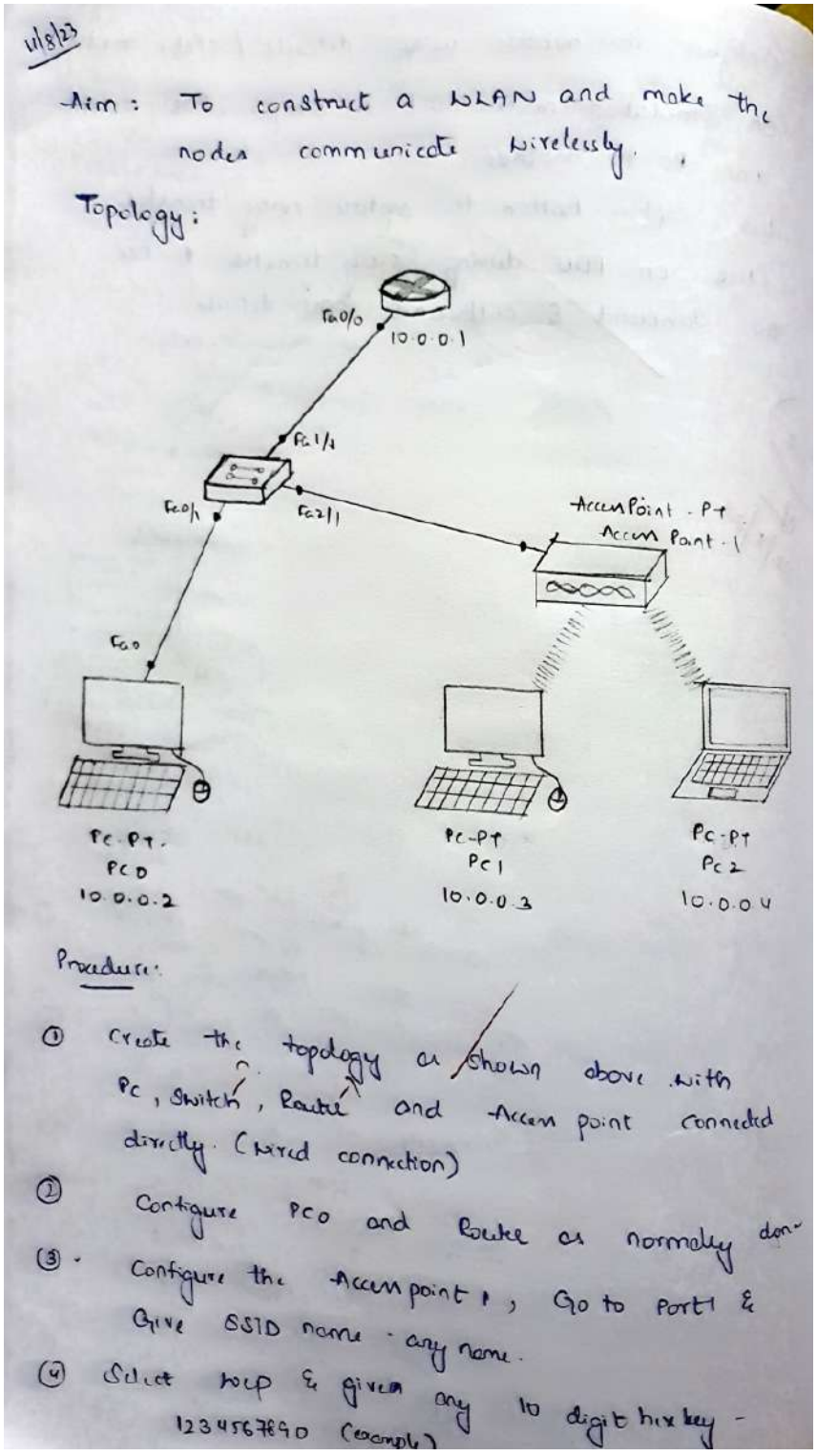


Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.2: bytes=32 time=4ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=3ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 2ms

PC>
```

PROGRAM 12

To construct a WLAN and make the nodes communicate wirelessly



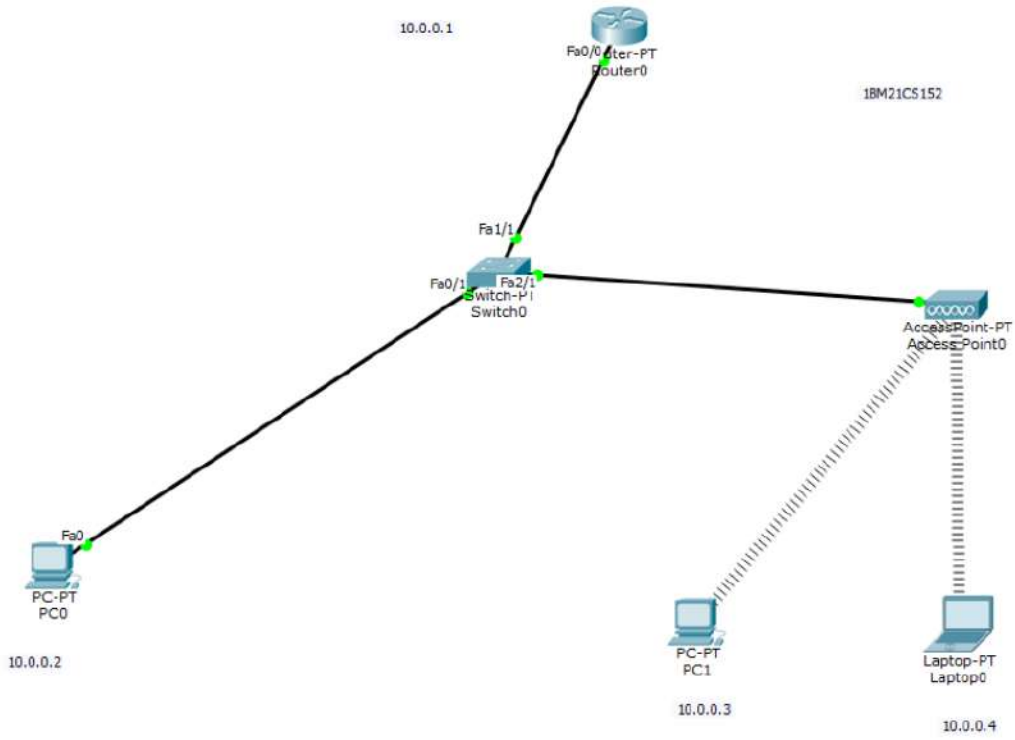
(5) Configure PCs & Laptop with Wireless Standards

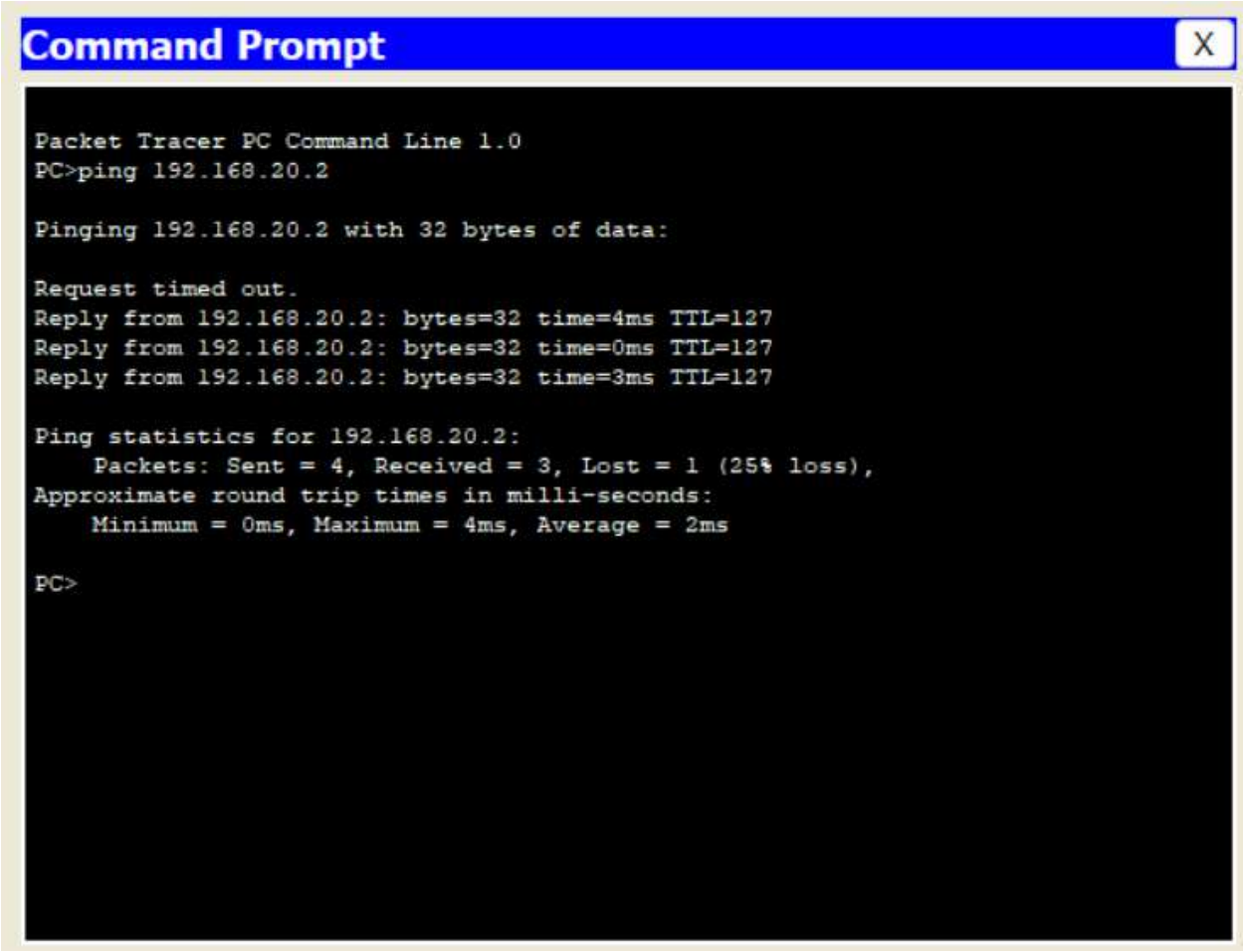
(6) Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N Wireless interface to empty port. Switch on the device.

(7) On the config tab, a new wireless interface would have been added. Now configure SSID, WEP, WEP Key, IP address and Gateway (as normally done) to the device.

10/10

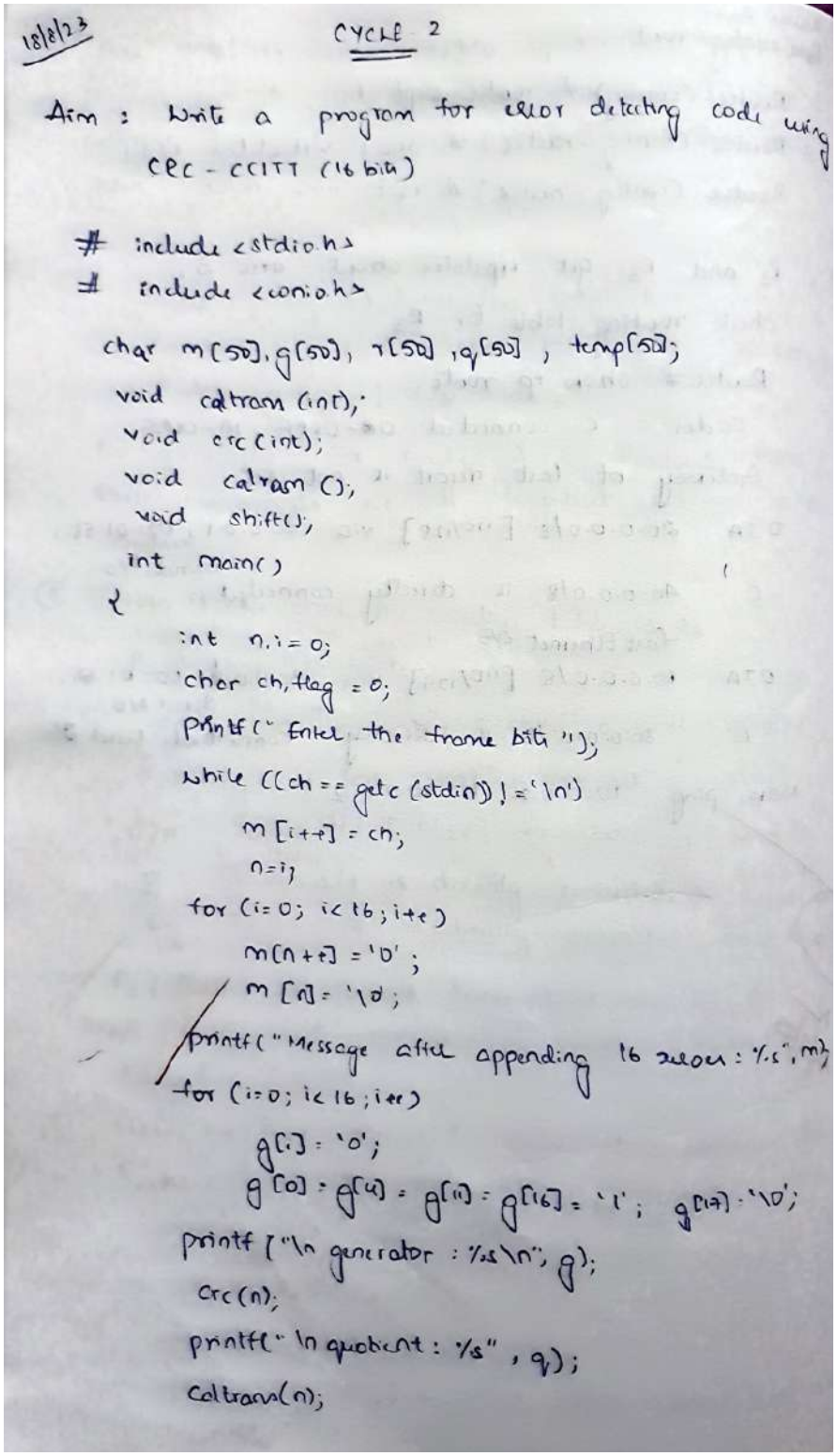
22/8/23


10.0.0.1
Fa0/0
Router-PT
Router0
1BM21CS152
Fa1/1
Fa0/1
Fa2/1
Switch-PT
Switch0
Access-Point-PT
Access-Point0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC1
10.0.0.3
Laptop-PT
Laptop0
10.0.0.4


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.2: bytes=32 time=4ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=3ms TTL=127
Ping statistics for 192.168.20.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 2ms
PC>

CYCLE - 2

Write a program for error detecting code using CRC_CCITT(16 bits)



18/8/23

CYCLE 2

Aim : Write a program for error detecting code using CRC - CCITT (16 bit)

```
#include<stdio.h>
#include<conio.h>
char m[50],g[50], r[50] ,q[50] , temp[50];
void caltrans(int);
void crc(int);
void caltrans();
void shift();
int main()
{
    int n,i=0;
    char ch,flag=0;
    printf(" Enter the frame bits ");
    while((ch=getc(stdin))!='\n')
        m[i++]=ch;
        n=i;
    for(i=0;i<16;i++)
        m[n++]='0';
        m[n]='\0';
    printf("Message after appending 16 zeros : %s",m);
    for(i=0;i<16;i++)
        g[i]='0';
        g[0]=g[4]=g[11]=g[16]='1'; g[17]='\0';
    printf("\n generator : %s\n",g);
    crc(n);
    printf("\n quotient : %s", q);
    caltrans(n);
```

```
printf("\n transmitted frame : %s", m);
printf("\n Enter transmitted frame:");
scanf("\n%s", m);
printf(" CRC checking");
crc(n);
printf("\n last remainder : %s", r);
for (i=0; i<16; i++)
    if(r[i] != '0')
        flag=01;
    else
        continue;
    if(flag==1)
        printf(" Error during transmission ");
    else
        printf(" Received frame is correct");
}
void crc(int n)
{
    int i,j;
    for(i=0; i<n; i++)
        temp[i] = m[i];
    for(i=0; i<16; i++)
        r[i] = m[i];
    for(i=0; i<n-16; i++)
    {
        if(r[0]=='1')
        {
            q[i] = '1';
            caldram();
        }
```

```
    for(j=0;j<=17;j++)
        temp[j]=r[j];
  }
  q[n-16]='\0';
}
void caltram()
{
    int i,j;
    for(i=1;i<=16;i++)
    {
        r[i-1]=(int)temp[i]-48)^((int)q[i]-48)+48;
    }
}
void shift()
{
    int i;
    for(i=1;i<=16;i++)
        r[i-1]=r[i];
}
void caltram(int n)
{
    int i,k=0;
    for(i=n-16;i<n;i++)
        m[i]=((int)m[i]-48)^((int)r[k++]-48)+48;
        m[i]='\0';
}
```

Output:

Enter frame bits : 1011

Message after appending 16 zeroes : 1011 0000 0000 0000 00

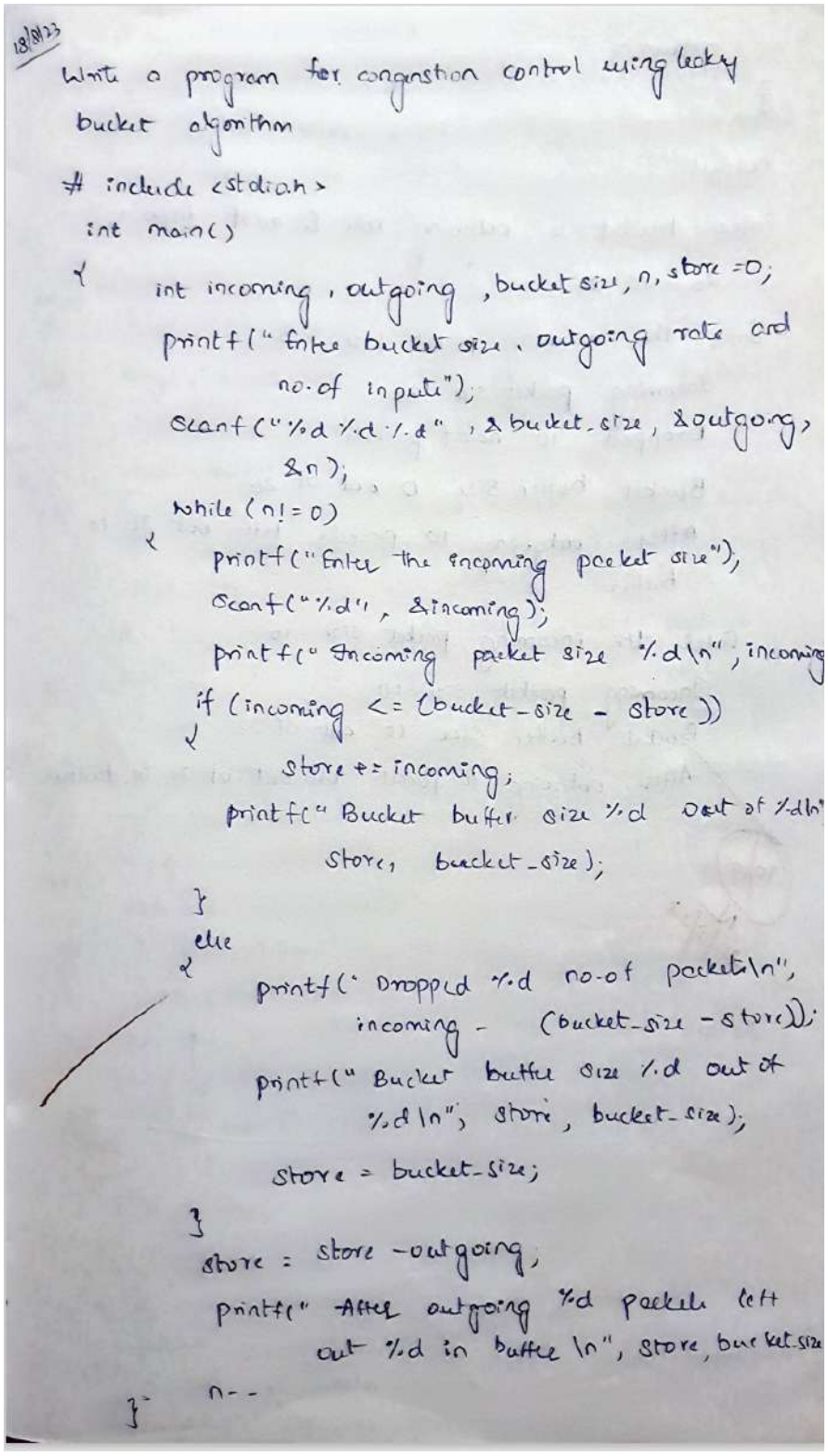
Generator : 10001000000100001

quotient : 1011

```
Enter the frame bits:1011
Message after appending 16 zeros:1011000000000000000
generator:10001000000100001


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


last remainder:0000000000000000

Received frame is correct
Process returned 0 (0x0)   execution time : 14.468 s
Press any key to continue.
```

```
Enter the frame bits:1001
Message after appending 16 zeros:1001000000000000000
generator:10001000000100001


quotient:1001
transmitted frame:10011001000100101001
Enter transmitted frame:10011001000000101001
CRC checking


last remainder:0000000100000000Error during transmission
Process returned 0 (0x0)   execution time : 19.597 s
Press any key to continue.
```

Write a program for congestion control using leaky bucket algorithm



```
18/8/23
Write a program for congestion control using leaky
bucket algorithm

# include <stdio.h>
int main()
{
    int incoming, outgoing, bucket size, n, store = 0;
    printf("Enter bucket size, outgoing rate and
            no. of inputs");
    scanf("%d %d %d", &bucket_size, &outgoing,
            &n);
    while (n != 0)
    {
        printf("Enter the incoming packet size");
        scanf("%d", &incoming);
        printf("Incoming packet size %d \n", incoming
        if (incoming <= (bucket_size - store))
        {
            store += incoming;
            printf(" Bucket buffer size %d out of %d\n"
                store, bucket_size);
        }
        else
        {
            printf(" Dropped %d no. of packets\n",
                incoming - (bucket_size - store));
            printf(" Bucket buffer size %d out of
                %d \n", store, bucket_size);
            store = bucket_size;
        }
        store = store - outgoing;
        printf(" After outgoing %d packets left
            out %d in buffer \n", store, bucket_size
        n--
    }
```

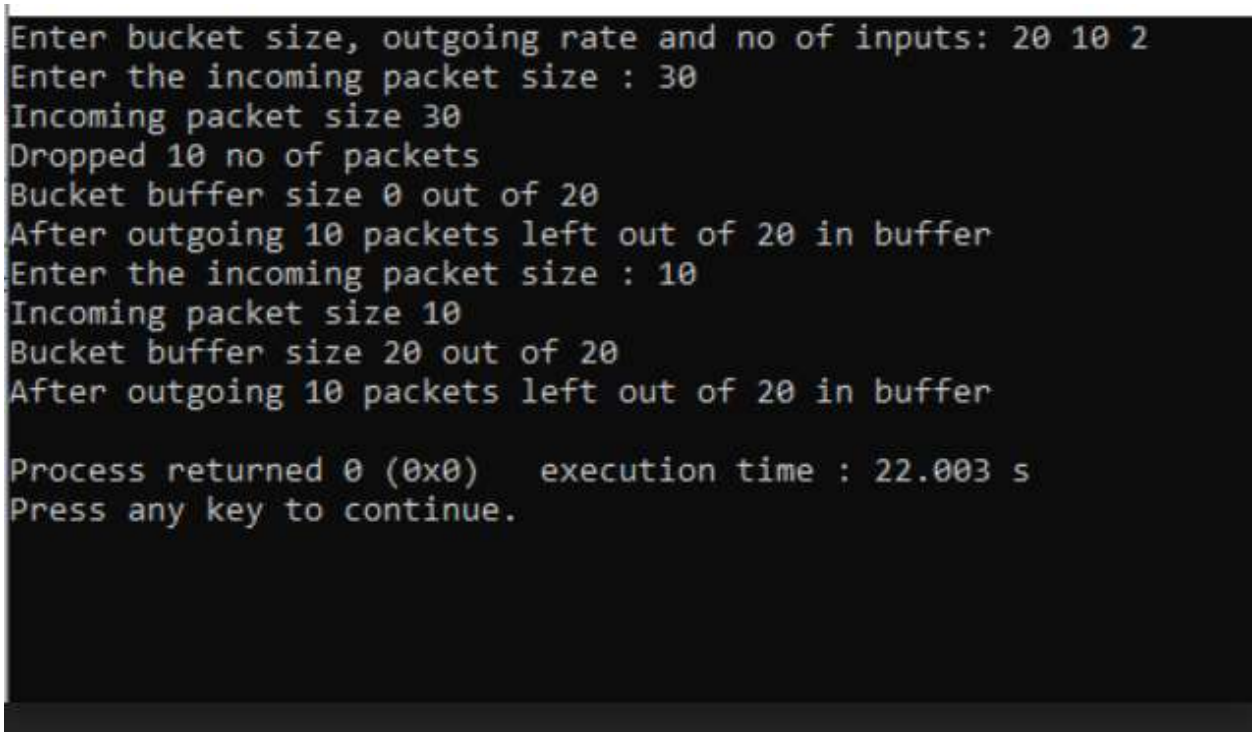
```
    return 0;
}
```

Output.

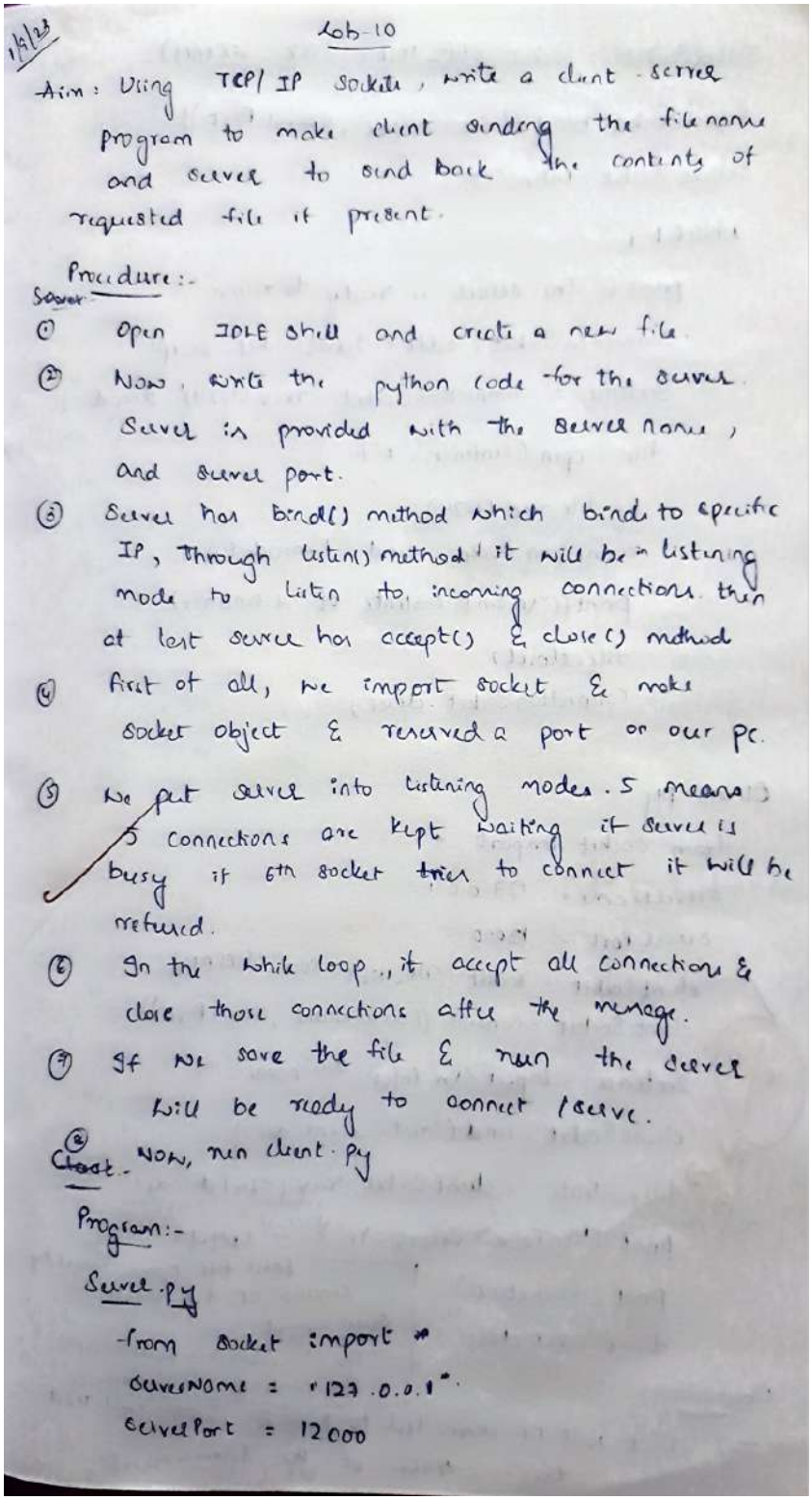
```
Enter bucket size, outgoing rate & no.of inputs.
20 10 2
Enter the incoming packet size : 30
Incoming packet size 30
Dropped 10 no.of packets
Bucket buffer size 0 out of 20
After outgoing 10 packets left out 20 in
buffer.
Enter the incoming packet size : 10
Incoming packet size 10
Bucket buffer size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

10/10

NL
22/8/23


Enter bucket size, outgoing rate and no of inputs: 20 10 2
Enter the incoming packet size : 30
Incoming packet size 30
Dropped 10 no of packets
Bucket buffer size 0 out of 20
After outgoing 10 packets left out of 20 in buffer
Enter the incoming packet size : 10
Incoming packet size 10
Bucket buffer size 20 out of 20
After outgoing 10 packets left out of 20 in buffer
Process returned 0 (0x0) execution time : 22.003 s
Press any key to continue.

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.



1/1/23

Lab-10

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

Procedure:-

Server:

1. Open IDLE shell and create a new file.
2. Now, write the python code for the server. Server is provided with the server name, and server port.
3. Server has bind() method which binds to specific IP, through listen() method it will be in listening mode to listen to incoming connections. then at last server has accept() & close() method
4. First of all, we import socket & make socket object & reserved a port on our pc.
5. We put server into listening modes. 5 means 5 connections are kept waiting if server is busy if 6th socket tries to connect it will be refused.
6. In the while loop, it accept all connections & close those connections after the message.
7. If we save the file & run the server will be ready to connect/serve.

Client:- Now, run client.py

Program:-

Server.py

```
-from socket import *
 serverName = '127.0.0.1'
 serverPort = 12000
```

```
ServerSocket = socket (AF_INET, SOCK_STREAM)
ServerSocket.bind ((ServerName, ServerPort))
ServerSocket.listen (1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = ServerSocket.accept()
    Sentence = connectionSocket.recv(1024).decode()
    file = open (Sentence, "r")
    l = file.read (1024)
    connectionSocket.send (l.encode())
    printf("\n Sent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Client.py :-

```
from socket import *
ServerName = '127.0.0.1'
ServerPort = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
clientSocket.connect ((ServerName, ServerPort))
Sentence = input ("\n Enter file name :")
clientSocket.send (Sentence.encode())
filecontents = clientSocket.recv (1024).decode()
print ("\n from Server :\n")
print (filecontents)
clientSocket.close()
```

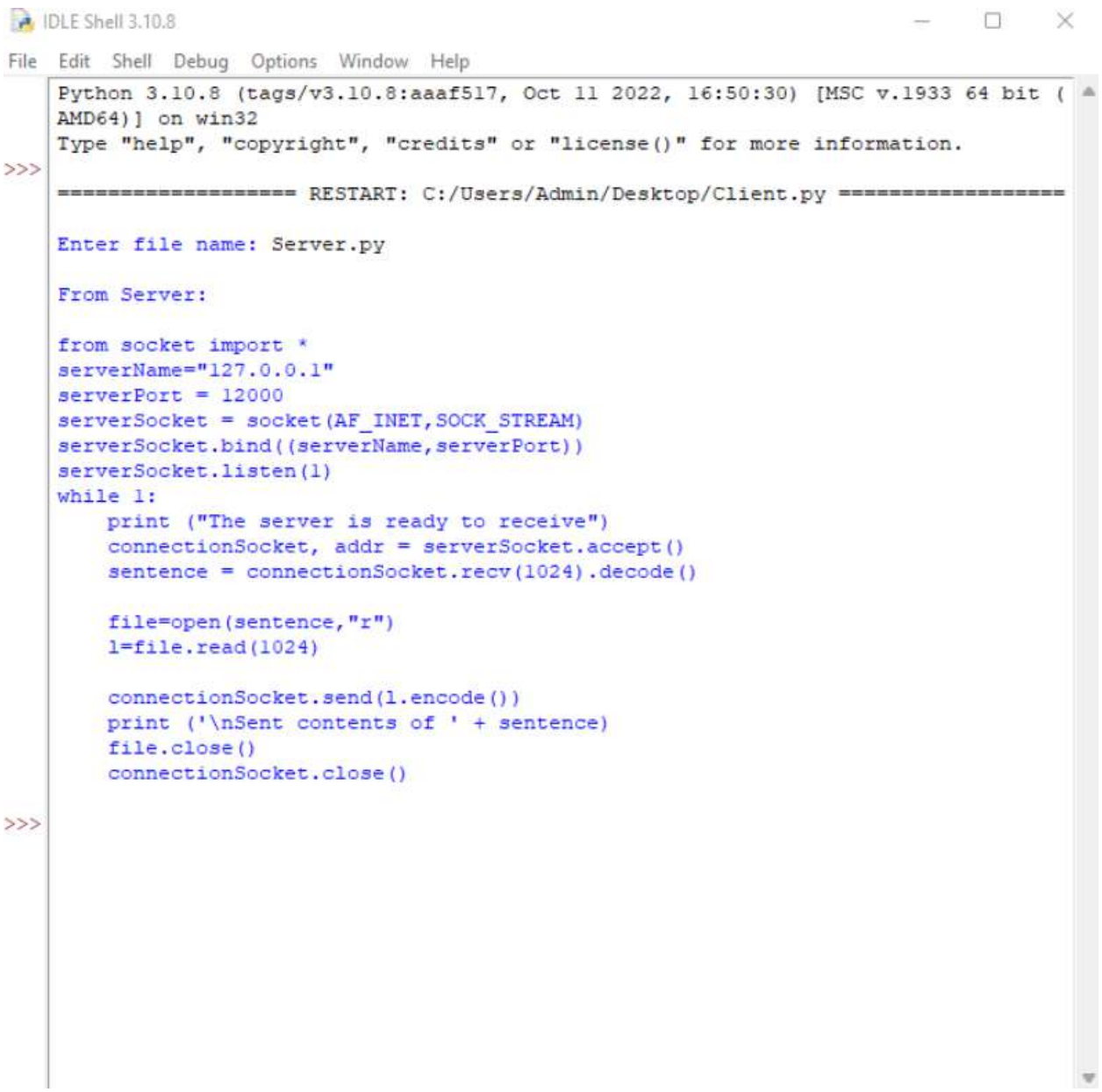
10/10

7/9/23

Client Output:-
Enter file name : server.py
contents of fo shown
Server output:-
The server is ready to receive
Sent contents of server.py

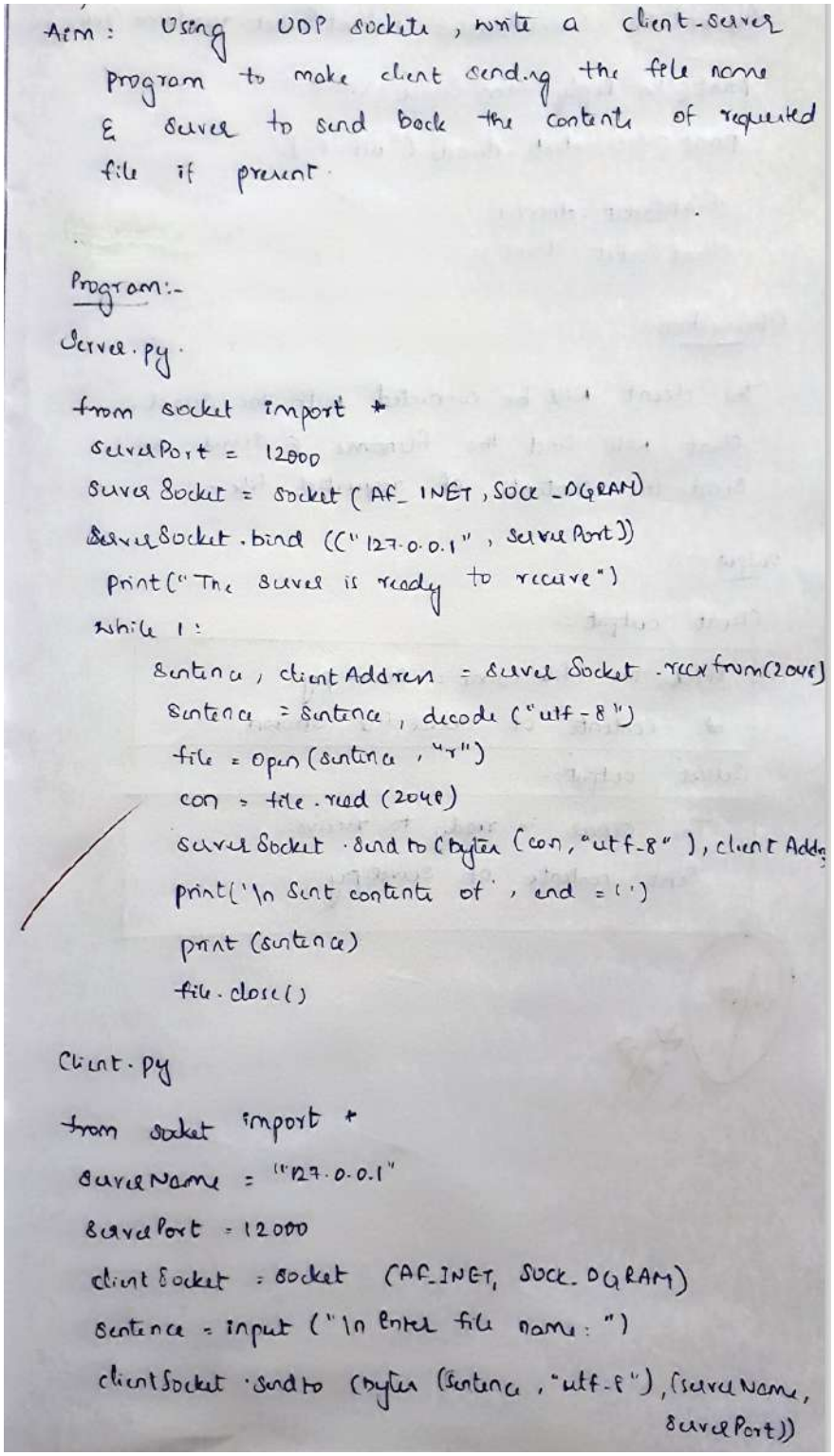
Observation:

The client will be connected to server & it will read the contents of the file.


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/Desktop/Client.py ==========
Enter file name: Server.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()
>>>


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/Desktop/UDP/server.py ==========
The server is ready to receive
Sent contents of server.py
Ln: 5 Col: 0

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.



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

Program:-

Server.py

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

Client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName,
                                               serverPort))
```

```
fileContents, serverAddress = clientSocket.recvfrom(2048)
print('\n Reply from Server :\n')
print (fileContents.decode("utf-8"))

clientSocket.close()
clientSocket.close()
```

Observation:-

The client will be connected with the server. Client will send the filename & server will send the contents of requested file.

Output

Client output :-

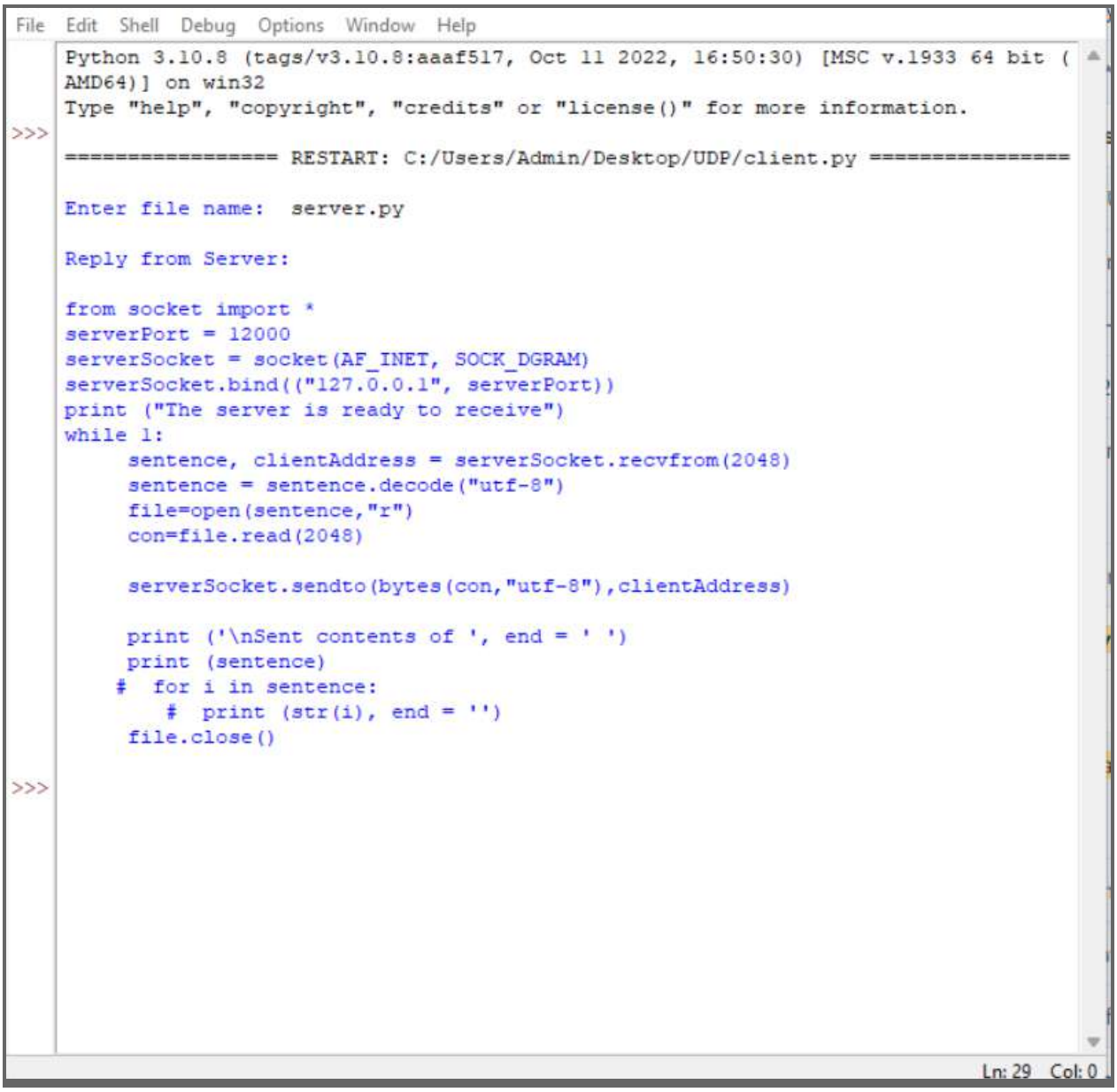
```
Enter the file name : server.py
#  Contents of server file shown
```

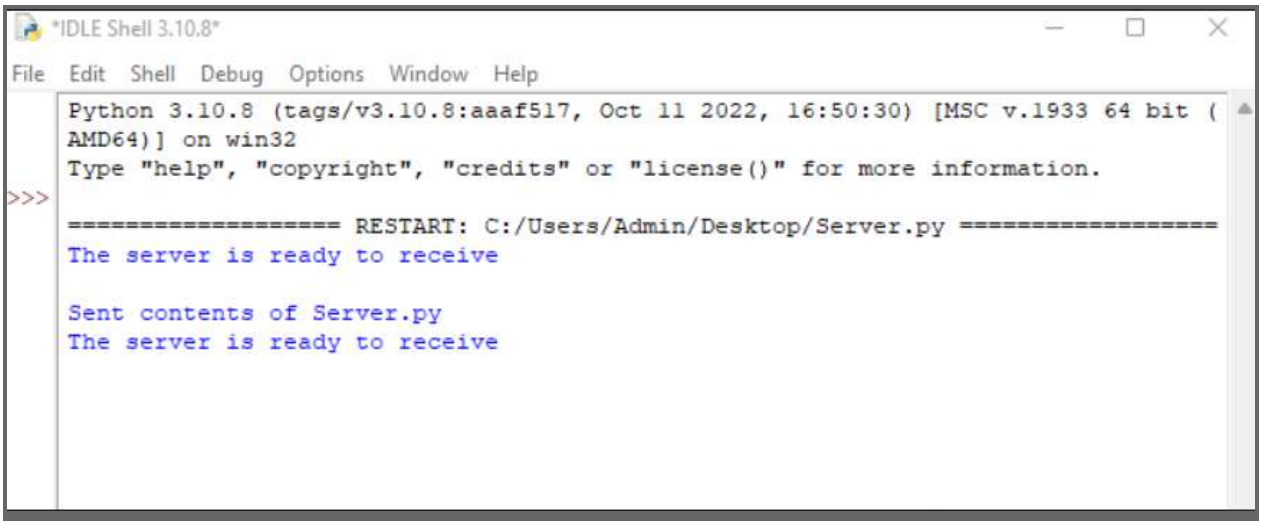
Server output:-

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

10/10

21/9/23


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/Desktop/UDP/client.py ==========
Enter file name: server.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()
>>>
Ln: 29 Col: 0


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/Desktop/Server.py ==========
The server is ready to receive
Sent contents of Server.py
The server is ready to receive