VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Computer Networks

Submitted by

PULKIT RAINA (1BM21CS148)

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 **PULKIT RAINA (1BM21CS148),** 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 OPERATING SYSTEMS **(22CS4PCOPS)** work prescribed for the said degree.

Name of the Lab-In charge: Nandini Vineeth
Designation: Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index Sheet

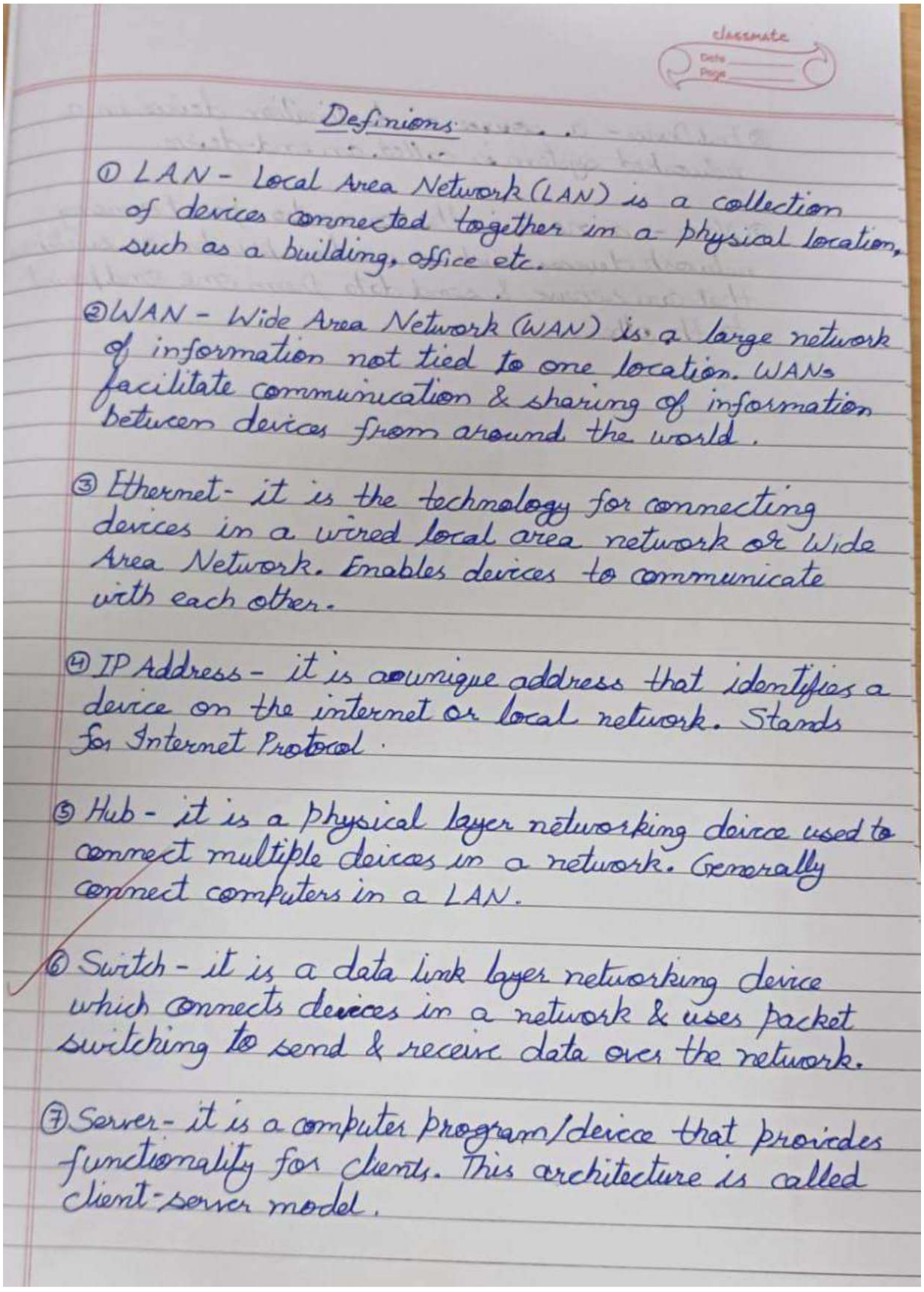
Lab Program No.	Program Details	Page No.
1	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message	1
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	7
3	Configure default route, static route to the Router	13
4	Configure DHCP within a LAN and outside LAN	17
5	Configure RIP routing Protocol in Routers	21
6	Configure OSPF routing protocol	25
7	Demonstrate the TTL/ Life of a Packet	30
8	Configure Web Server, DNS within a LAN	34
9	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	36
10	To understand the operation of TELNET by accessing the router in server room from a PC in IT office	38
11	To construct a VLAN and make the PC's communicate among a VLAN	41
12	To construct a WLAN and make the nodes communicate wirelessly	44

13	Write a program for error detecting code using CRC-CCITT (16-bits)	47
14	Write a program for congestion control using Leaky bucket algorithm	51
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	54
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	57

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

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



Definions

① LAN - Local Area Network (LAN) is a collection of devices connected together in a physical location, such as a building, office etc.

② WAN - Wide Area Network (WAN) is a large network of information not tied to one location. WANs facilitate communication & sharing of information between devices from around the world.

③ Ethernet - it is the technology for connecting devices in a wired local area network or Wide Area Network. Enables devices to communicate with each other.

④ IP Address - it is a unique address that identifies a device on the internet or local network. Stands for Internet Protocol.

⑤ Hub - it is a physical layer networking device used to connect multiple devices in a network. Generally connect computers in a LAN.

⑥ Switch - it is a data link layer networking device which connects devices in a network & uses packet switching to send & receive data over the network.

⑦ Server - it is a computer program/device that provides functionality for clients. This architecture is called client-server model.

⑧ End Device - a source or destination device in a networked system is called an end-device.

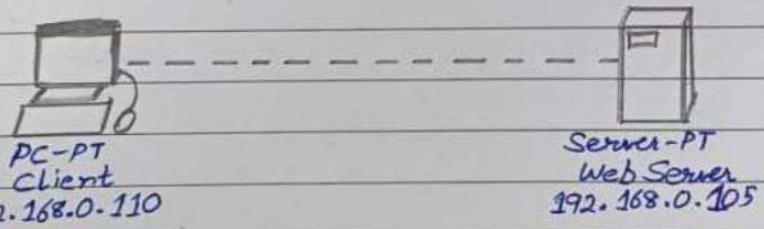
⑨ Node - a node is the connection point among network devices such as routers, printers or switches that can receive & send data from one end point to the other.

09.06.2023

WEEK - 1

STEPS TO SEND A SIMPLE TEXT MESSAGES

1. Add a client end-device & a web server end-device.
2. Connect both using a Copper Cross-over cable
3. Set the client's DNS server to 192.168.0.105. Set the IP Address under the FastEthernet to 192.168.0.110.
4. Select the Web Server & IP address is to be set to 192.168.0.105.
5. Select the DNS services & set the domain name as "www.firstlab.com" & IP address as 192.168.0.105 & add.
6. Ensure DNS service is ON.
7. Add Simple PDU ~~tool~~ is used to send a simple one-time message from PC to the server & vice-versa
8. The log values are displayed in the PDU List Window




```
PC > ping 192.168.0.110
Pinging 192.168.0.110 with 32 bytes of data:

Reply from 192.168.0.110: bytes = 32 time = 4ms TTL = 128
Reply from 192.168.0.110: byte = 32 time = 2ms TTL = 128
Reply from 192.168.0.110: byte = 32 time = 4ms TTL = 128
Reply from 192.168.0.110: byte = 32 time = 4ms TTL = 128
```

Ping Statistics for 192.168.0.110:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milliseconds:
Minimum = 2ms, Maximum = 4ms, Average = 3ms

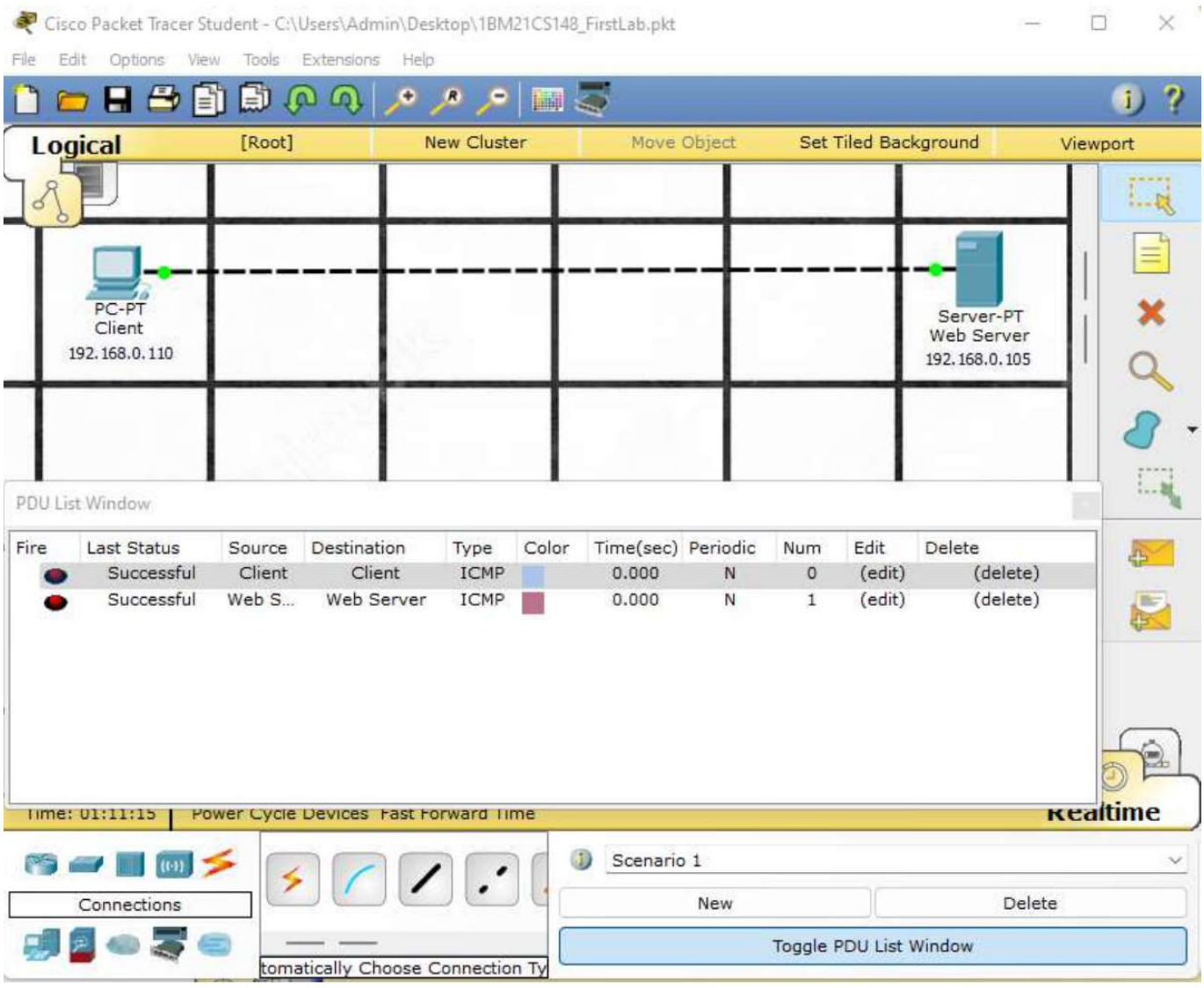
PC > ping 10.0.0.2

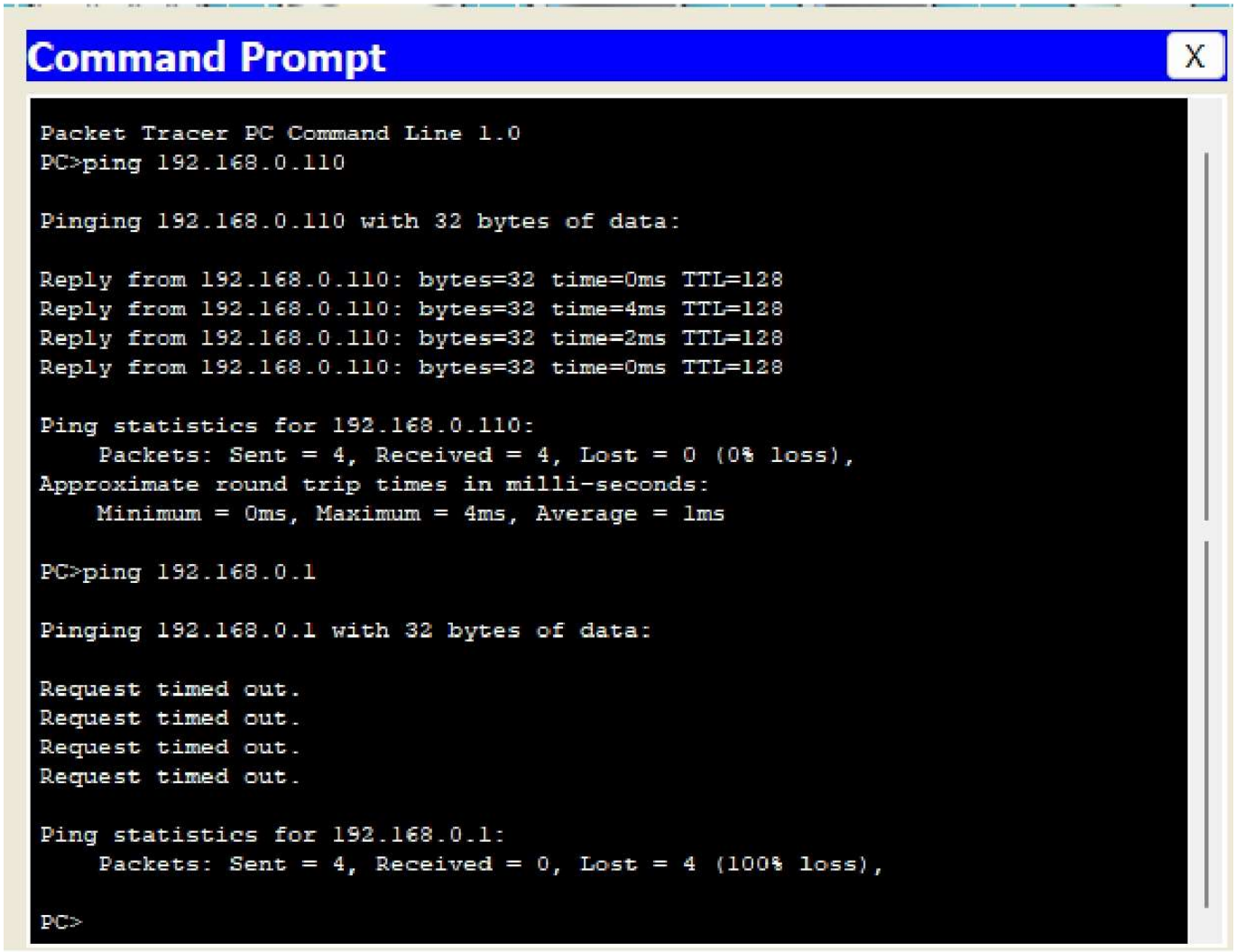
Pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

9/6/23


Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS148_FirstLab.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC-PT
Client
192.168.0.110
Server-PT
Web Server
192.168.0.105
PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
Successful Client Client ICMP 0.000 N 0 (edit) (delete)
Successful Web S... Web Server ICMP 0.000 N 1 (edit) (delete)
Time: 01:11:15
Power Cycle Devices
Fast Forward Time
Realtime
Connections
tomatically Choose Connection Ty
Scenario 1
New
Delete
Toggle PDU List Window


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.0.110
Pinging 192.168.0.110 with 32 bytes of data:
Reply from 192.168.0.110: bytes=32 time=0ms TTL=128
Reply from 192.168.0.110: bytes=32 time=4ms TTL=128
Reply from 192.168.0.110: bytes=32 time=2ms TTL=128
Reply from 192.168.0.110: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.110:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
PC>ping 192.168.0.1
Pinging 192.168.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>

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

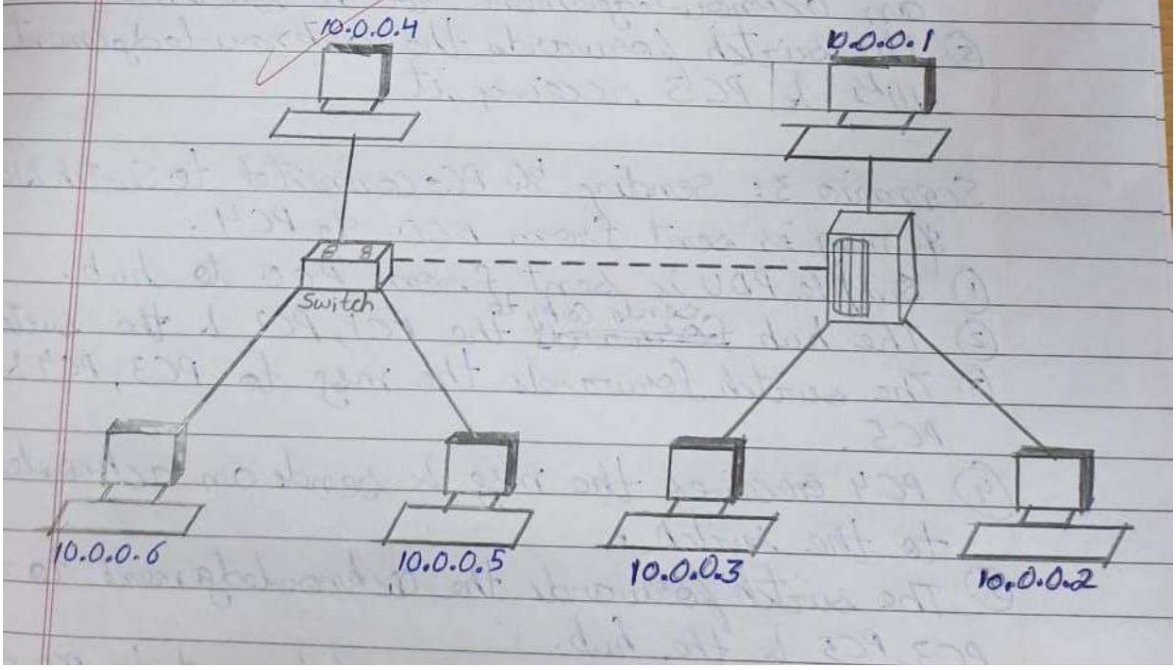
WEEK - 2

Q. Create a topology & simulate sending a simple PDU from source to destination using simple hub & switch as connecting domains.

Sol^n:

STEPS:

1. Add a generic hub & connect 3 PCs to it using Copper Straight-through wire.
2. Add a generic switch & connect 3 PCs to it using Copper Straight-through wire.
3. Wait for the switch-PC connections to be established.
4. Connect the hub & switch using a Copper Cross-over wire.
5. Set the IP Addresses for all the PCs (end-devices) as 10.0.0.1, 10.0.0.2 & so on.
6. Send a simple PDU message from one PC to another & wait for the simulation to run after clicking Auto capture.



Scenario 1 : Sending b/w PCs connected to Hub
If msg is sent from PC1 to PC2
① Simple PDU is sent from PC1 to Hub.
② Hub sends the copies of the msg to PC0, PC2 & switch.
③ The PC2 receives the msg & sends back acknowledgement to the Hub.
④ The Hub further forwards the acknowledgement to PC0, PC1 & switch.
⑤ The PC1 receives it & the transfer is completed

Scenario 2: Sending b/w PCs connected to Switch.
If msg is sent from PC3 to PC4
① Simple PDU is sent from PC3 to switch.
② The switch forwards the msg to PC5, P4 & the hub.
③ The hub forwards it to the PC0, PC1 & PC2.
④ The PC4 accepts the message & sends an acknowledgement to the switch.
⑤ The switch forwards the acknowledgement to & PC3 receives it.

Scenario 3: Sending b/w PCs connected to Switch & Hub
If msg is sent from PC0 to PC4.
① Simple PDU is sent from PC0 to hub.
② The hub ~~forwards~~ Sends copy to the PC1, PC2 & the switch.
③ The switch forwards the msg to PC3, PC4 & PC5.
④ PC4 receives the msg & sends an acknowledgement to the switch.
⑤ The switch forwards the acknowledgement to the PC3, PC5 & the hub.
⑥ Hub sends copy of the acknowledgement to PC0, PC1 &.

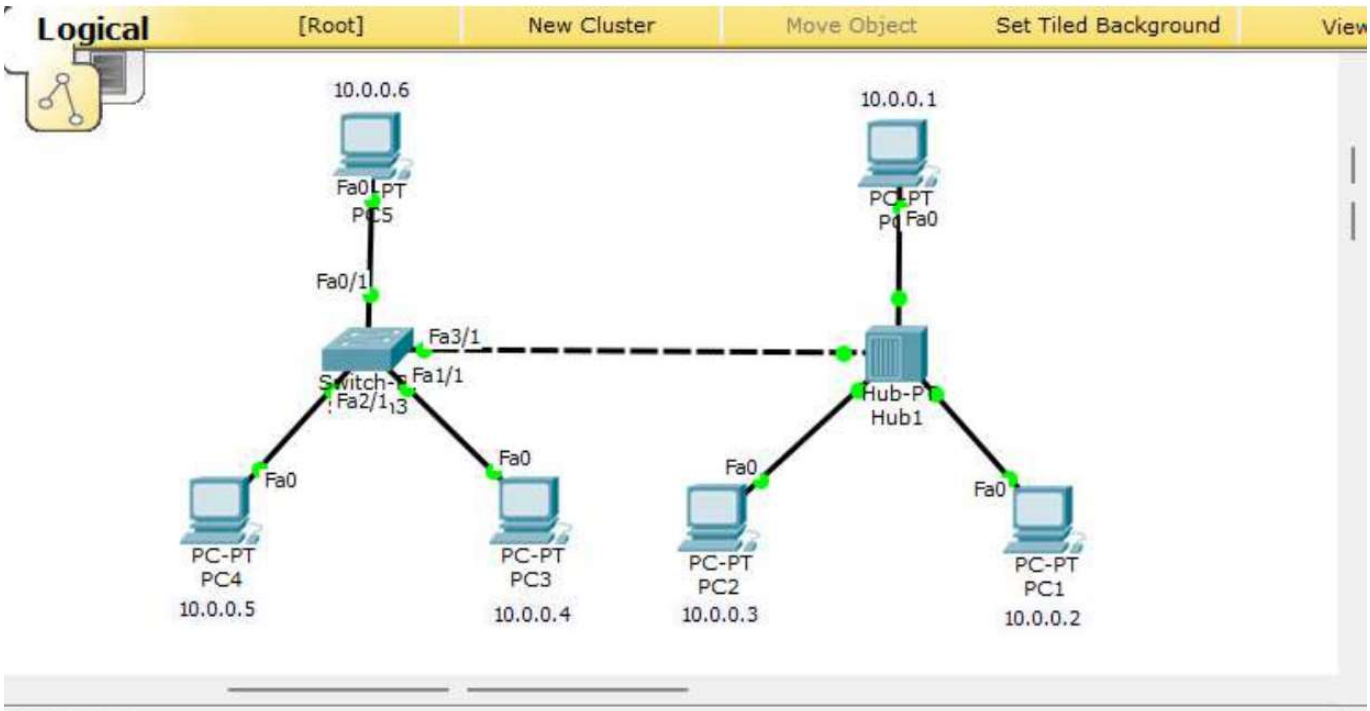
⑨ PC0 receives the acknowledgement thereby completing the transfer of msg.

PC> ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

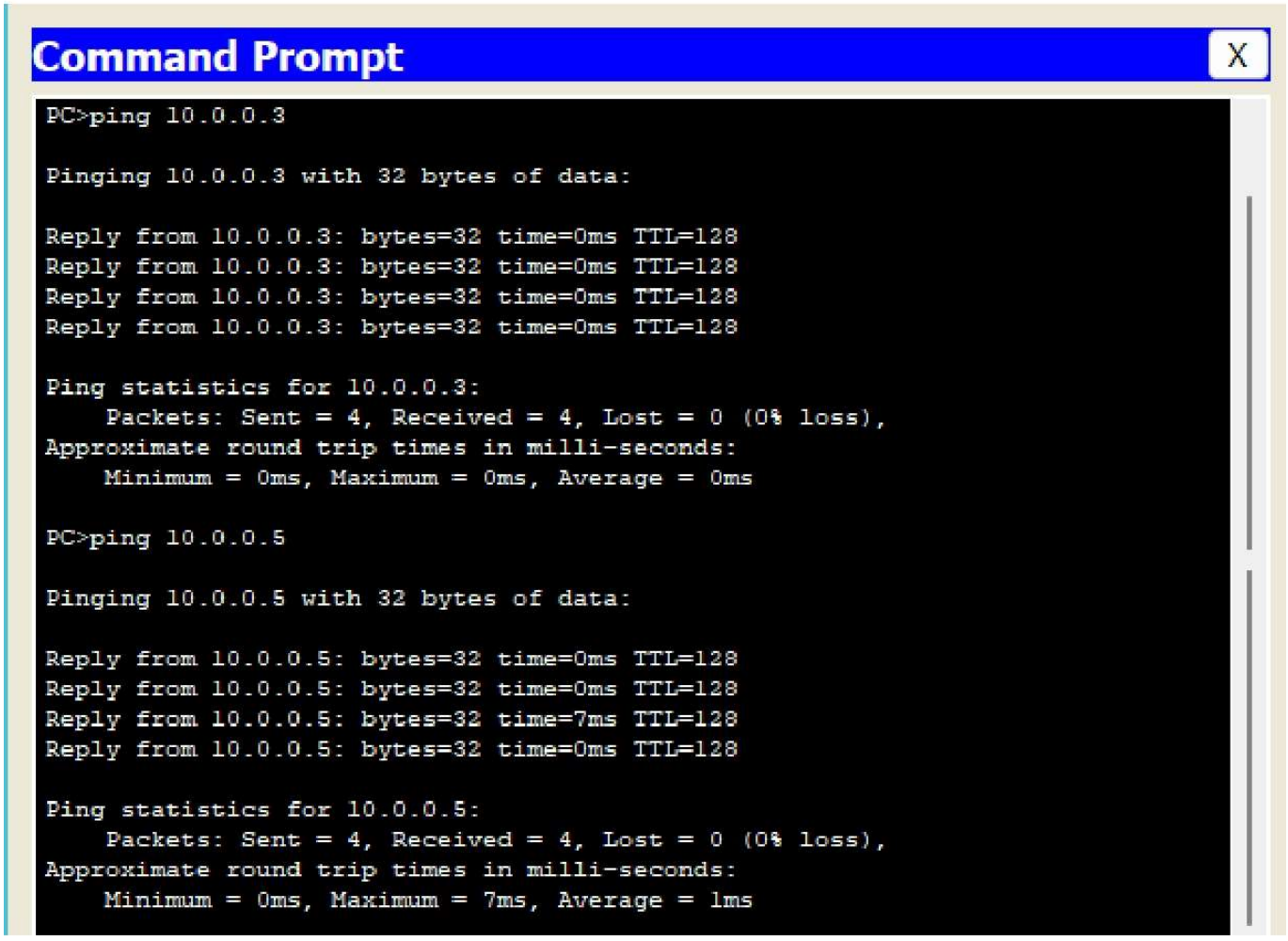
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128

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



PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC4	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC5	PC2	ICMP		0.153	N	1	(edit)	(delete)


Command Prompt
X
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>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=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=7ms 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 times in milli-seconds:
Minimum = 0ms, Maximum = 7ms, Average = 1ms

```
PC>ping 10.0.0.45

Pinging 10.0.0.45 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.45:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.6:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```

3. Configure default route, static route to the Router

WEEK - 3

Q. Configure default route, static route to router.

Soln: STEPS:

① Add 2 PCs & connect them to a generic Router.
② Configure the PCs by setting their IP addresses to 10.0.0.1 & 20.0.0.1 respectively.
③ Set the gateways as 10.0.0.2 & 20.0.0.2 respectively.
④ & Go to the Cmd Line Interface in the router and enter 'no' for 'continue with configuration dialogue'.
⑤ Type 'config terminal' & enter twice.
⑥ Type 'interface fastEthernet 0/0' & enter.
⑦ Type 'ip address 10.0.0.2 255.0.0.0'
⑧ Type 'no shut' & the connection is now established.
⑨ Repeat steps "1 to 8" for the other PC. 20.0.0.1.
⑩ Repeat steps ① to ⑧ for PC3 & PC4.
⑪ Connect router 1 & router 2 via a router 3.
⑫ All route to router connections via Serial DTE & PC to route via Copper cross-over.
⑬ Go to the Router 1 CLI & type the following:

```
enable ↵
config t ↵
interface Serial2/0 ↵
ip address 50.0.0.2 255.0.0.0 ↵
no shut ↵
```

⑭ Go to the Router 3 CLI & type:

```
enable ↵
config t ↵
interface Serial2/0 ↵
ip address 50.0.0.1 255.0.0.0 ↵
no shut ↵
```

The connection b/w @ Router 1 & 3 is green now.

(15) Repeat steps (13) & (14) for Router 2 to 3 similarly.

(16) Go to Router 1 CLI & type

show ip route.

It shows only the direct connections.

(17) We statically connect routers to the PCs by typing following in the CLI: (for router 1):

ip route 30.0.0.0 255.0.0.0 50.0.0.1 ↵

ip route 40.0.0.0 255.0.0.0 40.0.0.1 ↵

(for router 2):

ip route 10.0.0.0 255.0.0.0 60.0.0.1 ↵

ip route 20.0.0.0 255.0.0.0 60.0.0.2 ↵

(for router 3):

ip route		10.0.0.0	255.0.0.0	50.0.0.1
"	"	20.0.0.0	"	"
"	"	30.0.0.0	"	60.0.0.1
"	"	40.0.0.0	"	"

(18) Now data transfer b/w PCs is successful.

(19) Before statically connecting routers pinging b/w PCs NOT directly connected was unsuccessful.

```
PC> Ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:

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)
```

After statically defining the route:

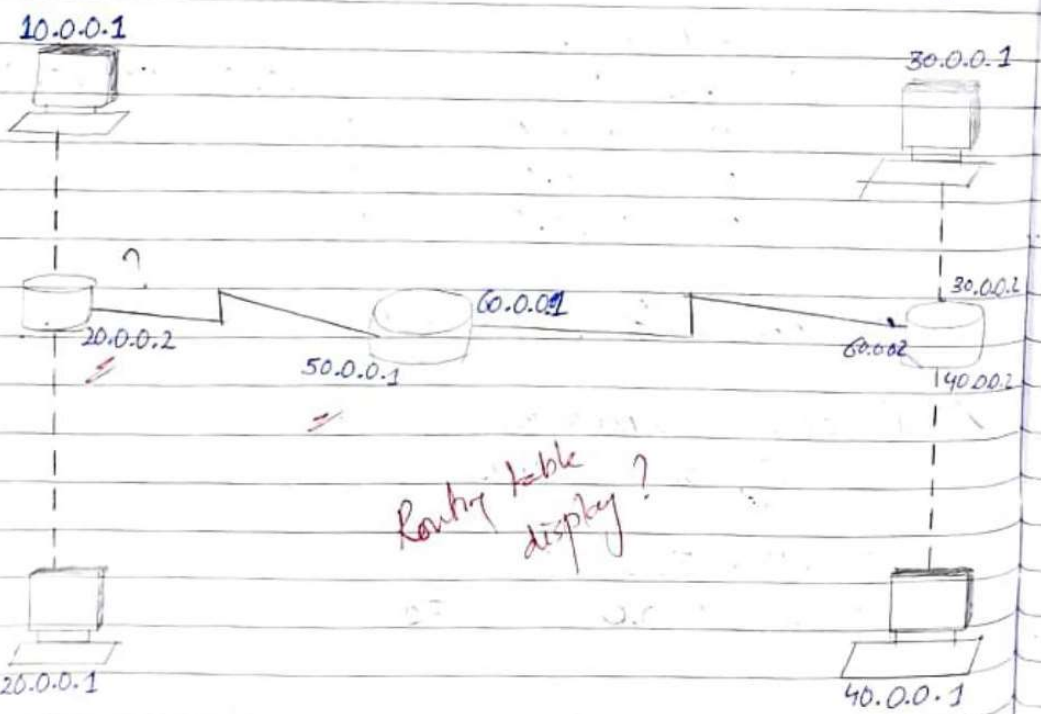
PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Pinging

from 40.0.0.1 with 32 bytes of data:

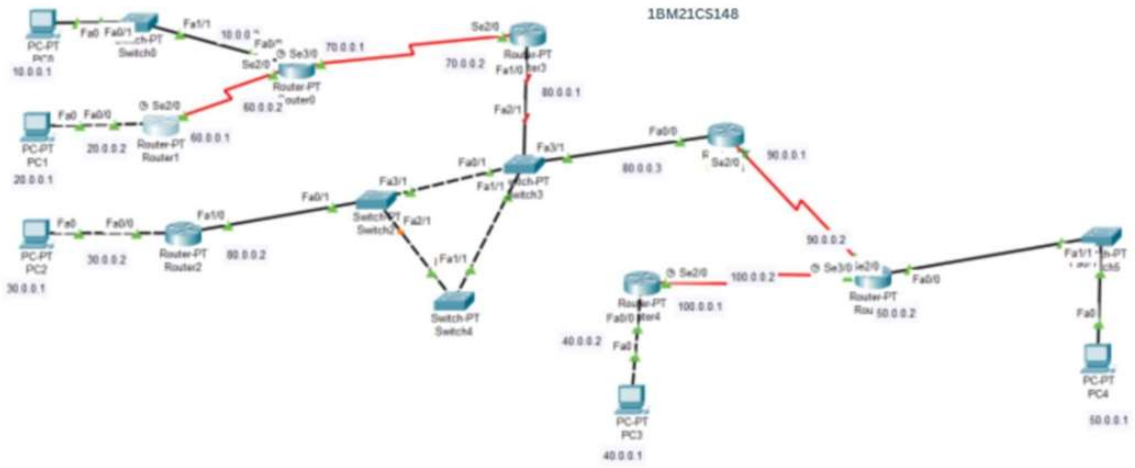
Reply from 40.0.0.1: bytes = 32 times = 2ms TTL = 125



9/10

Routing table display?

26/6/23

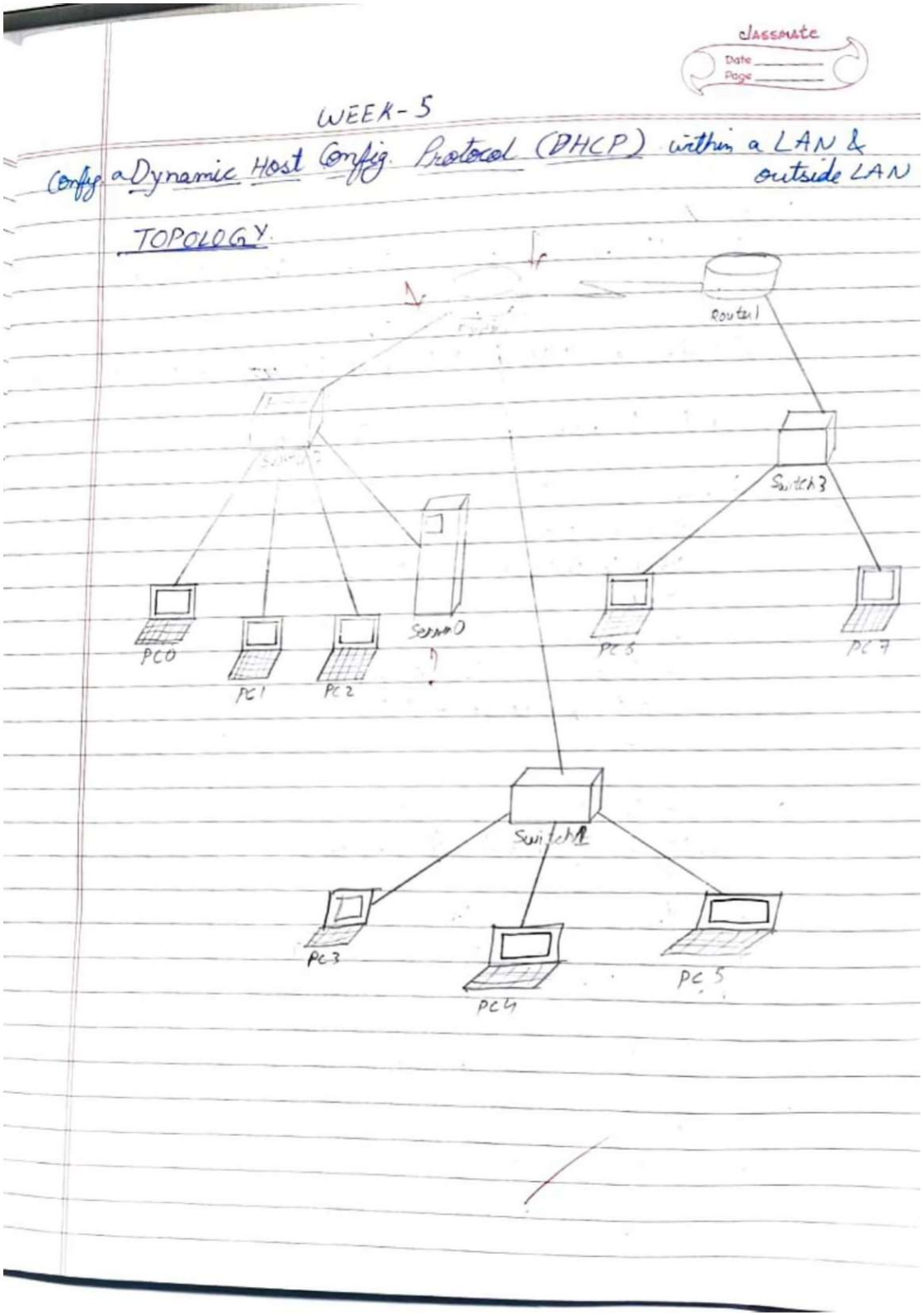


Physical Config Desktop Programming Attributes

Command Prompt

```
C:\>ping 90.0.0.2

Pinging 90.0.0.2 with 32 bytes of data:

Reply from 90.0.0.2: bytes=32 time=12ms TTL=252
Reply from 90.0.0.2: bytes=32 time=2ms TTL=252
Reply from 90.0.0.2: bytes=32 time=10ms TTL=252
Reply from 90.0.0.2: bytes=32 time=2ms TTL=252

Ping statistics for 90.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 12ms, Average = 6ms

C:\>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=3ms TTL=123
Reply from 40.0.0.1: bytes=32 time=3ms TTL=123
Reply from 40.0.0.1: bytes=32 time=3ms TTL=123

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 3ms, Average = 3ms

C:\>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=3ms TTL=124
Reply from 50.0.0.1: bytes=32 time=6ms TTL=124
Reply from 50.0.0.1: bytes=32 time=2ms TTL=124

Ping statistics for 50.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 6ms, Average = 3ms

C:\>
```

4. Configure DHCP within a LAN and outside LAN



Create the topology as shown.

> In the server
Set IP address : 10.0.0.1 (& gateway 10.0.0.25)

Services : DHCP → on
Services → DHCP → Default gateway → 10.0.0.25

Start IP address 10.0.0.2 (save)

Service pool 1
gateway → 10.0.0.25
Start ip address - 20.0.0.2
add
Service pool 2
gateway → 10.0.0.25
Start ip address - 40.0.0.2
add.

> In router 0 :-
- Set 2 network ip address
- config t.
- Interface fa 4/0
- ip address 10.0.0.25 255.0.0.0
- no shut.

11th fa 20.0.0.25

- exit
- interface fa 0/0
- ip helper-address 10.0.0.1
- no shut.

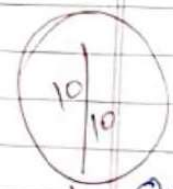
→ Static route (for 40.0... ip route)
config t
ip route 40.0.0.0 255.0.0.0 30.0.0.2

In router 1
set ip address
config t
interface fa0/0
ip address 40.0.0.26 255.0.0.0
no shut

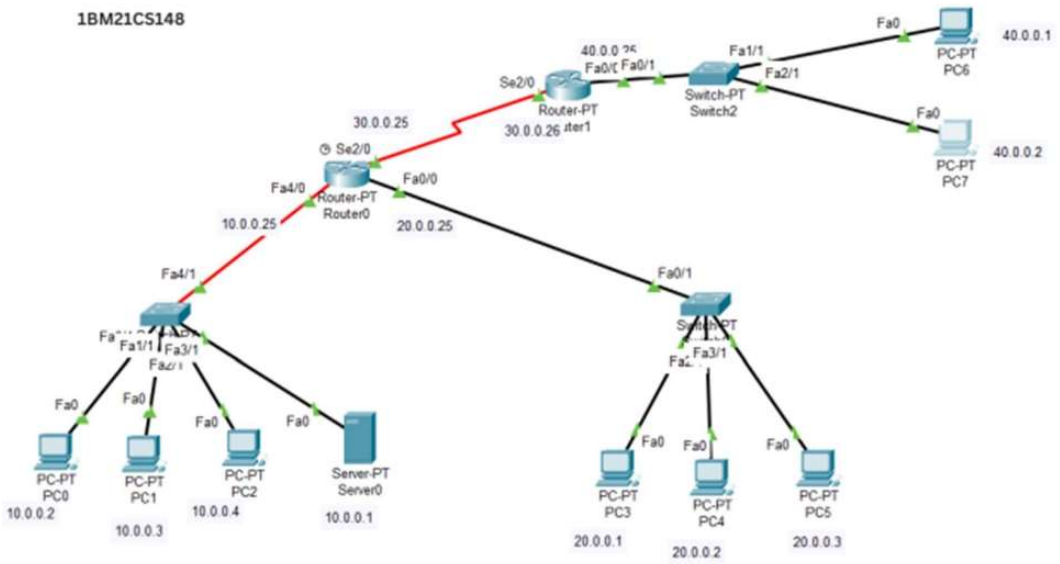
config t
interface se2/0
ip address 30.0.0.0 255.0.0.0
no shut

→ Static route for 10 & 20 network
config t
ip route 10.0.0.0 255.0.0.0 30.0.0.25
ip route 20.0.0.0 255.0.0.0 30.0.0.25

Setup helper address
config t
interface fa0/0
ip helper-address 10.0.0.1
no shut



Output:
In any PC
Desktop → IP config → DHCP
(Dynamic IP address is assigned to all PC by server).



Physical Config Desktop Programming Attributes

IP Configuration X

Interface FastEthernet0

IP Configuration

DHCP Static

IPv4 Address 10.0.0.2

Subnet Mask 255.0.0.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

Automatic Static

IPv6 Address /

Link Local Address FE80::260:2FFF:FE75:6CE1

Default Gateway

DNS Server

802.1X

5. Configure RIP routing Protocol in Routers

Output Website:

Cisco Packet Tracer

Welcome to Cisco Packet Tracer.

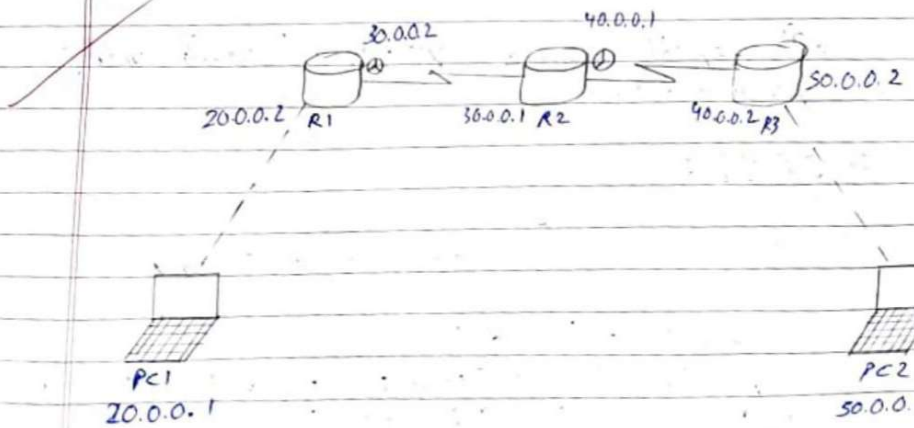
Quick Links:
A small Page.
Copyright.

Name:
Pulkit Raina

USN:
1BM21CS148

Q-7. Q: Config. RIP routing Protocol in Routers

Topology.



STEPS:

① Create the ~~other~~ topology as shown.
② Configure all IP address of PCs & routers as shown.
③ Set the getway of PC1 as 20.0.0.2 & of PC2 as 30.0.0.2 resp.
④ Config routers to router using following cmds:

```
interface serial ↵
ip address XX 255.0.0.0 ↵
encapsulation ppp
clock rate 64000   <only for ones clocked
```

⑤ Now configure all routers with following cmds:

```
config t
router rip
network 10.0.0.0
network 20.0.0.0
```

⑥ Repeat same steps for all routers making respectively changes

⑦ ping 50.0.0.1

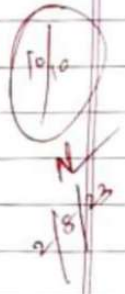
```
Pinging 50.0.0.1 with 32 bytes of data

Request timed out
Reply from 50.0.0.1: bytes=32 time= 2ms TTL = 126
Reply from 50.0.0.1: bytes=32 time= 7ms TTL = 126
Reply from 50.0.0.1: bytes=32 time= 8ms TTL = 126
Ping statistics for 50.0.0.1:
```

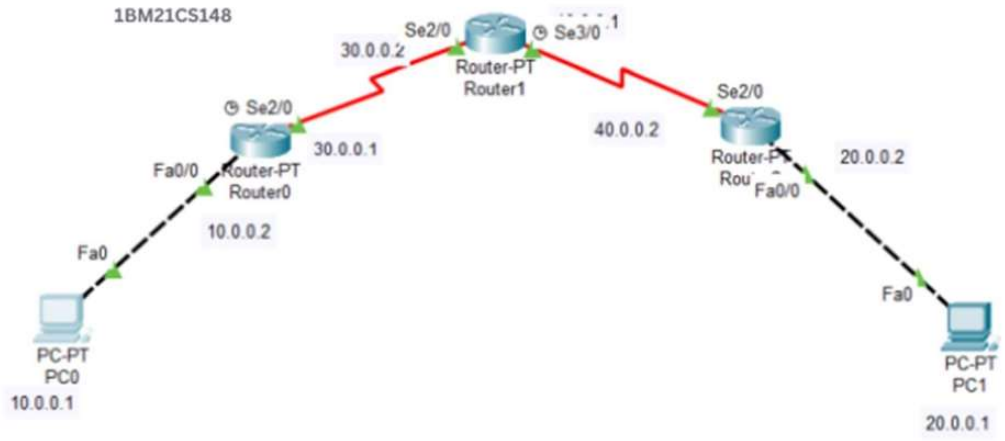
~~OUTPUT:~~

OBSERVATION:-

For huge networks IP route can't be set statically, so ip route is set using routing info. protocol (rip).

The packets are encapsulated using PPP (Point to point protocol).





Physical Config Desktop Programming Attributes

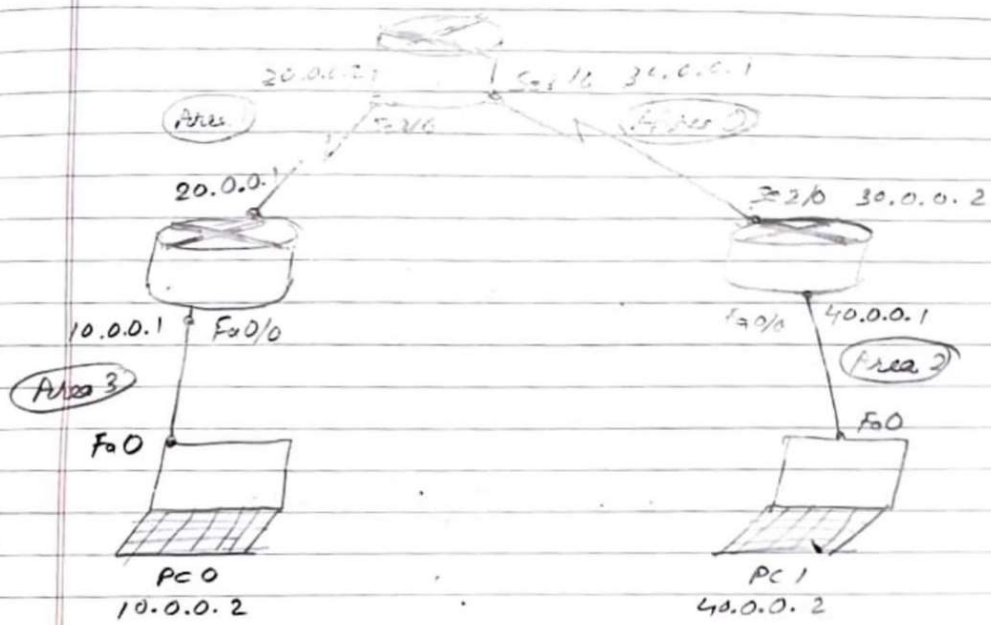
Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.1: bytes=32 time=24ms TTL=125
Reply from 20.0.0.1: bytes=32 time=25ms 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 = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 24ms, Maximum = 25ms, Average = 24ms

C:\>
```

6. Configure OSPF routing protocol

Config OSPF routing protocol & connect Areas.

Topology:



PROCEDURE:

1. Create the topology as given above with 2 PCs & 3 routers.
2. Config the ip address for PCs as 10.0.0.2 & 40.0.0.2 resp.
3. Config. the routers with ip address for all interfaces.
4. Now, for all Se port of routers, config using cmd "encapsulation ppp" & give "clock rate 64000" command at ports having clock symbol.

Eg:- For Router 2

```
router(config)# interface se2/0
router(config-if)# ip encapsulation ppp
```

$\frac{x}{100} = \frac{9.5}{10}$

router(config-if)# no shut
router(config-if)# exit

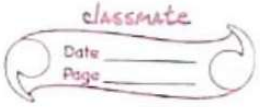
router(config)# interface Se3/0
router(config-if)# encapsulation ppp
router(config-if)# clock rate 64000
router(config-if)# no shut

⑤ Now, enable ip routing by configuring OSPF routing protocol in all routers.

In Router R1,
Router(config)# router ospf 1
Route(config-router)# router-id 1.1.1.1 area 3
Route(config-router)# network 10.0.0.0 0255.255.255.
Router(config-router)# network 20.0.0.0 0.255.255.255 area
Router(config-router)# exit
Like config four Router 2 & 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 external type 1.

Gateway of last resort is NOT set
C 10.0.0.0/8 is directly connected, fa 2/0
C 20.0.0.0/8 is directly connected, se 2/0
O IA 40.0.0.0/8 via 20.0.0.2, 00:04:23, Se 2/0
O IA 30.0.0.0/8 via 20.0.0.2, 00:07:29, Se 2/0
R2 knows area 0, network 20.0.0.0 connected to R2 from R1, so R1 learns through this network.



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

For Router 1:

```
Router(config-if)# interface loopback 0
Router(config-if)# ip add 172.16.1.252 255.255.255.0
Router(config-if)# no shut
```

Do 111^{b} for Router 2 & Router 3. Using these cmds. we add loopback address to the routers

(4) Now, if we check the routing table for R3,

```
R3 # 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, Se2/0
C   40.0.0.0/8 is directly connected, Fa 0/0
C   30.0.0.0/8 is directly connected Se 0/0
```

R3 (Router 3) doesn't know about area 3, so we will create virtual link between Router 1 & Router 3

(5) Now, we have to create virtual link b/w Router 1 & 2; with this we create a virtual link to connect area 3 to area 0.

In Router R1:

Router(config)# router ospf 1
Router(config-router)# area 1 virtual link 2.2.2.2

In Router R2:

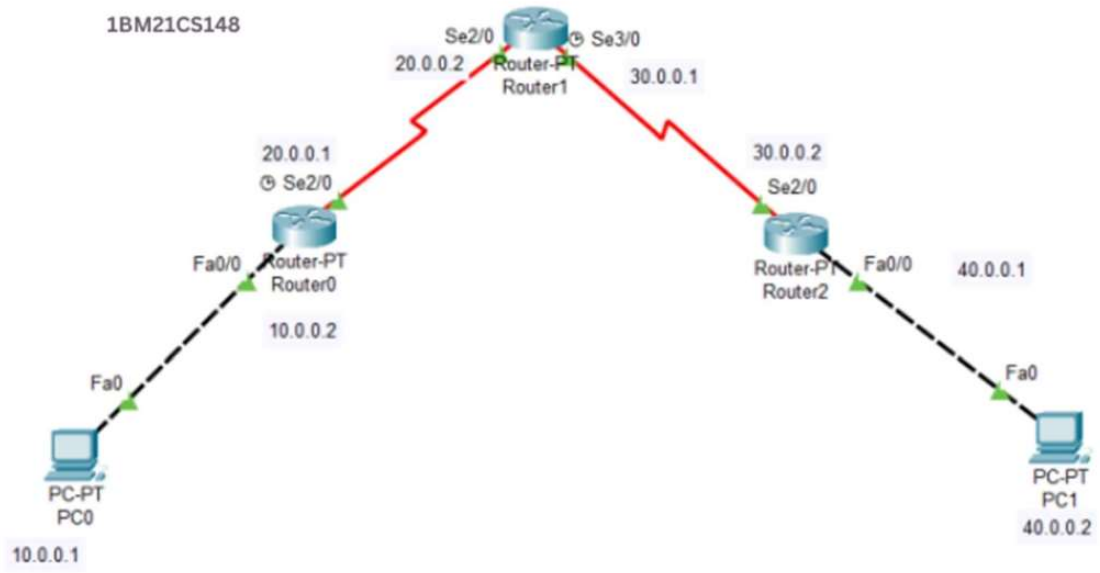
Router(config)# router ospf 1
Router(config-router)# area 1 virtual link 1.1.1.1
Router(config-router)# exit.

(10) R_2 & R_3 get updates about area 3, check routing table for R3.

Router# show ip route.
Codes: C - connected, O - OSPF, IA - OSPF,
Gateway of last resort is NOT set

O IA 20.0.0.0/8 [110/123] via 30.0.0.1, 00:01:56, Se 2/0
C 40.0.0.0/8 is directly connected Fa 0/0
O IA 10.0.0.0/8 [110/128] via 30.0.0.1, 00:01:56, Se 2/0
C 30.0.0.0/8 is directly connected, Se 2/0

(11) Now ping 10.0.0.2 to 40.0.0.2



Physical Config Desktop Programming Attributes

Command Prompt

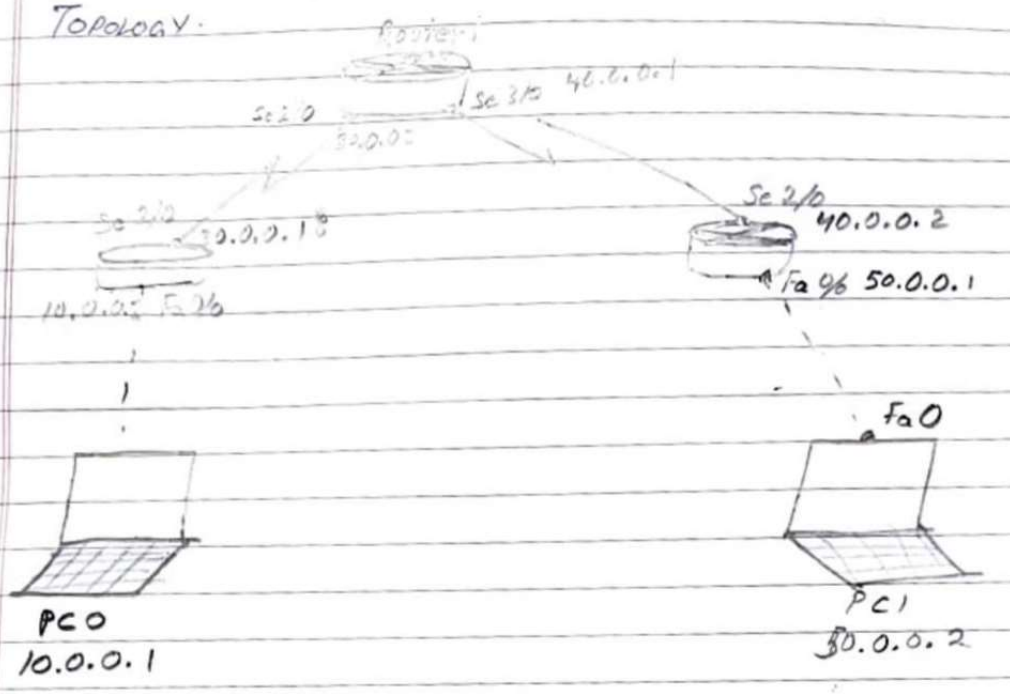
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=21ms TTL=125
Reply from 40.0.0.2: bytes=32 time=24ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 24ms, Average = 15ms

C:\>
```

7. Demonstrate the TTL/ Life of a Packet

classmate
Date
Page

AIM: Demonstrate the TTL/Life of a Packet.

TOPOLOGY:



PROCEDURE:

① Create a topology as shown above with 2PCs & 3 routers.

② Config the IP address of PC0 & PC1 as 10.0.0.1 & 50.0.0.2 resp.

③ Config. the IP address of routers using following cmds

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

④ Configure the routers using default/static routing.

⑤ In simulation mode, send a simple PDU from 1 PC to another.

⑥ Use capture button to capture every transfer.

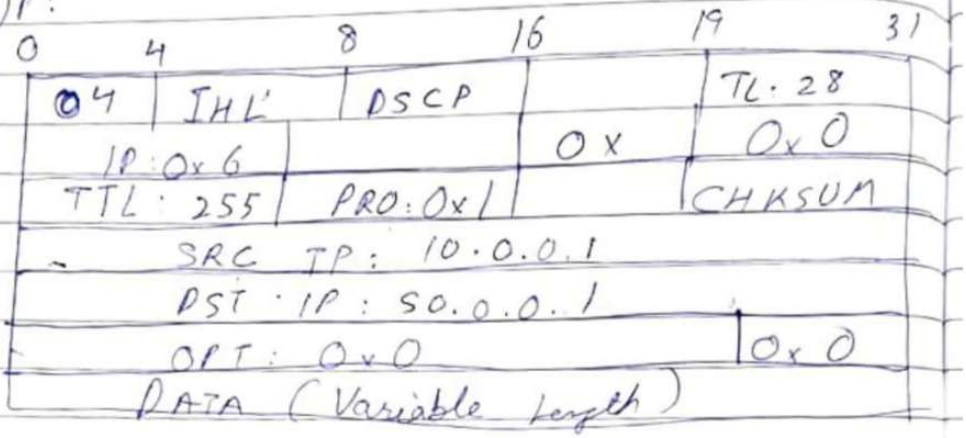
⑦ Click on PDU during every transfer to see the

inbound & outbound PDU details.

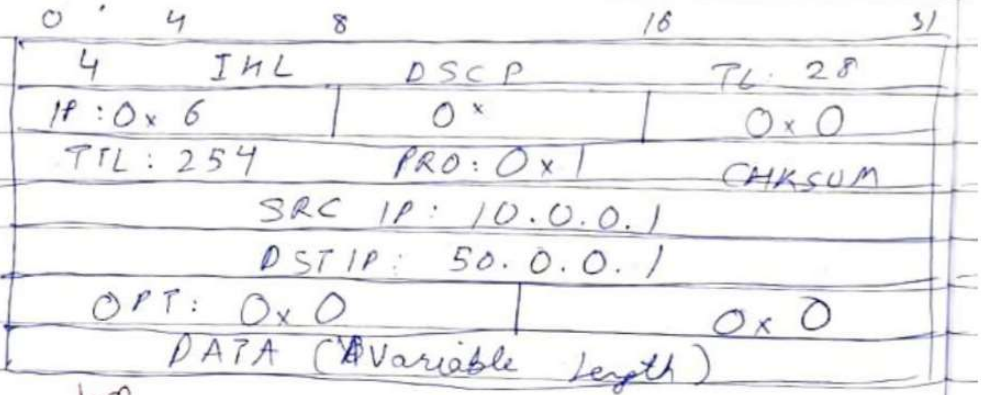
OUTPUT :-

IP :

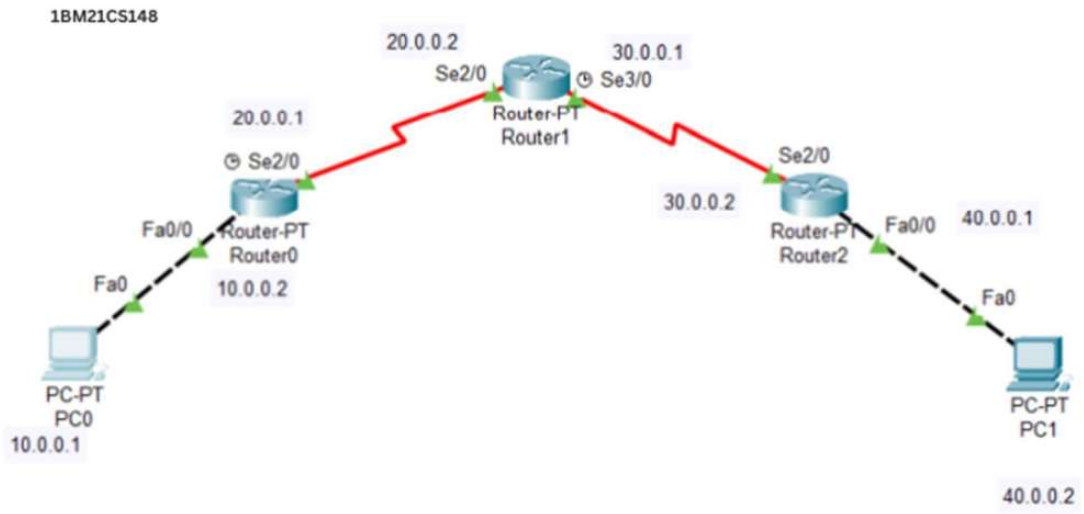
1)



8/10



Observation ?



PDU Information at Device: Router2

OSI Model | Inbound PDU Details | Outbound PDU Details

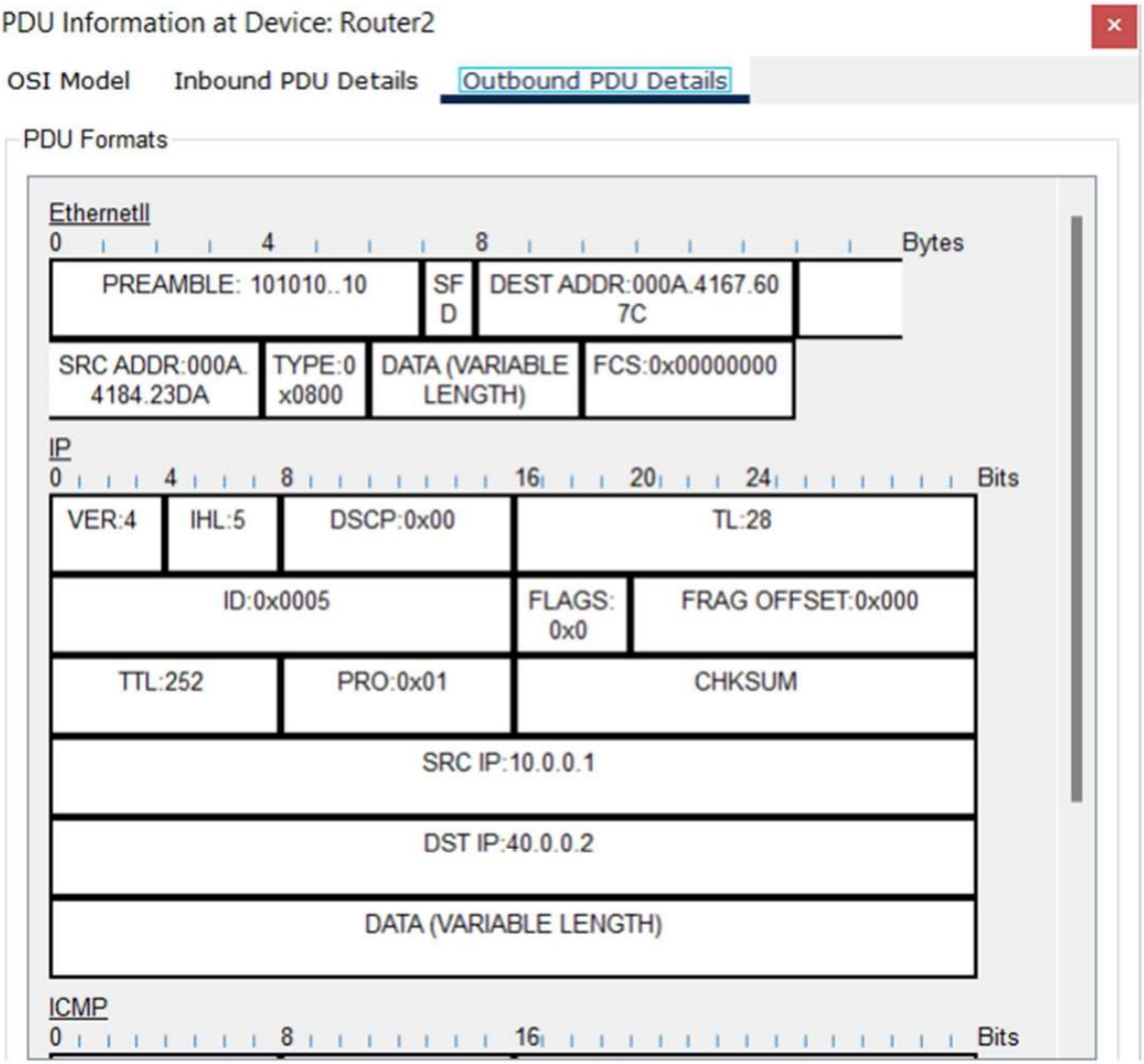
PDU Formats

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)						

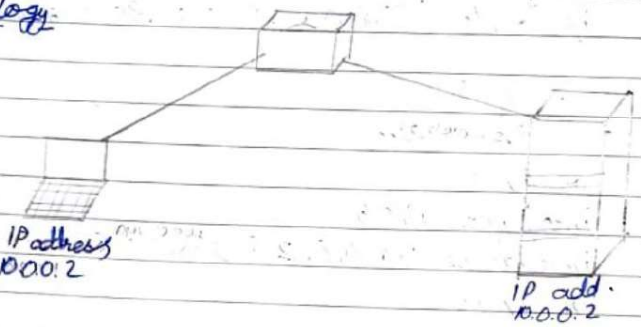

PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
EthernetII
0
4
8
Bytes
PREAMBLE: 101010..10
SFD
DEST ADDR:000A.4167.607C
SRC ADDR:000A.4184.23DA
TYPE:0x0800
DATA (VARIABLE LENGTH)
FCS:0x00000000
IP
0
4
8
16
20
24
Bits
VER:4
IHL:5
DSCP:0x00
TL:28
ID:0x0005
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

8. Configure Web Server, DNS within a LAN

WEEK-6

Q. Create a topology (1 PC, 1 Switch, 1 Server)

① Topology:



STEPS

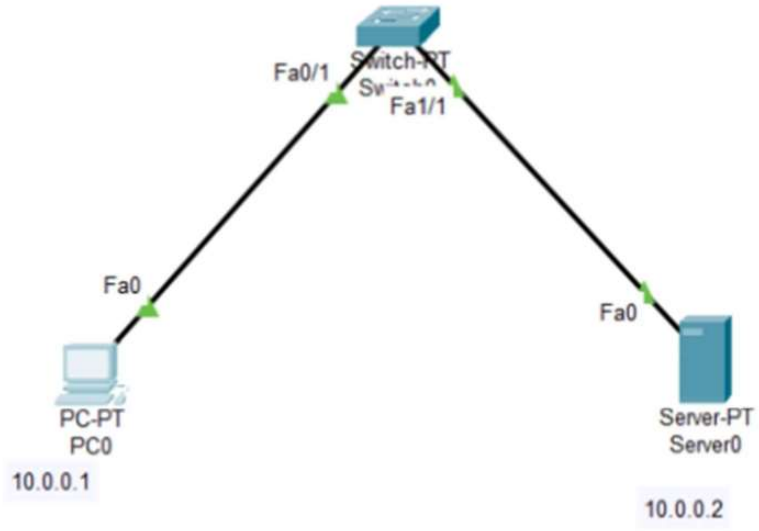
① Make the above topology & connect them as shown
② Config the PC & Server IP address'
③ Turn the DNS service 'On' in the server & add a website name with the address as IP address of the server.
④ Go to HTTP services in server & edit the 'index.html' page. Add relevant info & save.
⑤ Go to web browser in the PC & enter the URL as the name of the website you set in the server DNS.

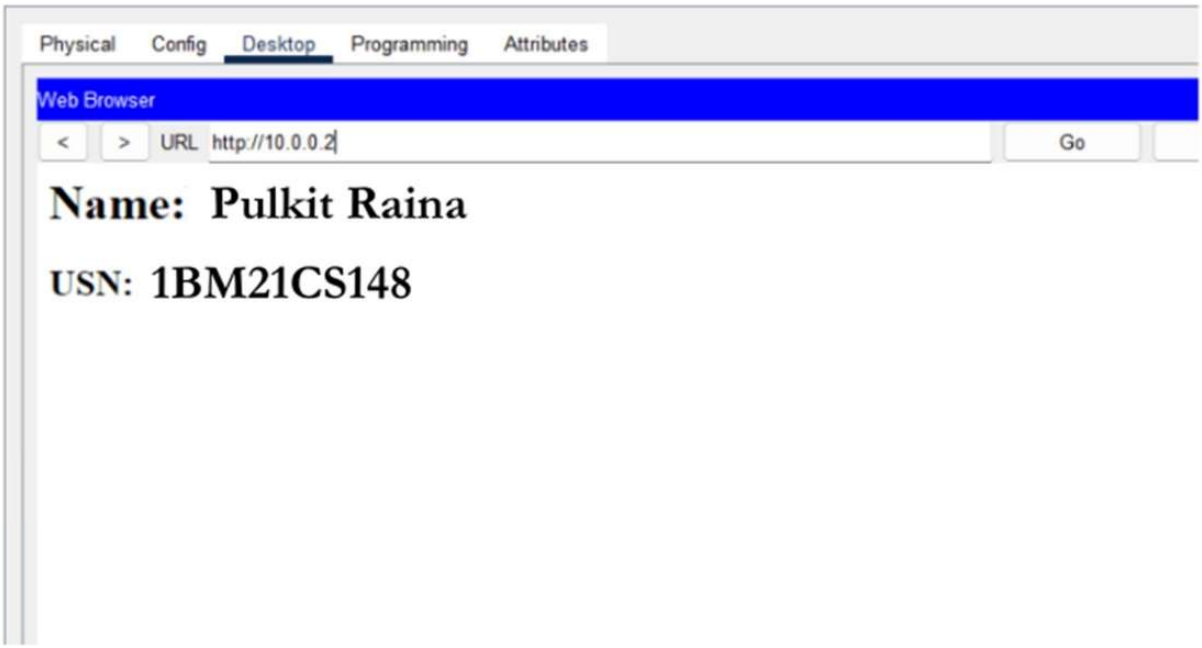
OUTPUT:

In the web browser in PC we can now enter the name of the server & we are shown the index.html page with the edits made.

Observation:
We can enter name of a website instead of its IP address.

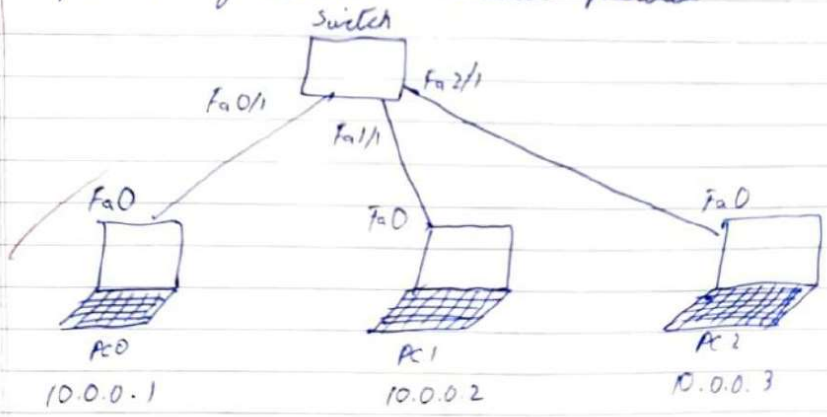
1BM21CS148


Switch-PT
Fa0/1
Fa1/1
Fa0
Fa0
PC-PT
PC0
10.0.0.1
Server-PT
Server0
10.0.0.2


Physical
Config
Desktop
Programming
Attributes
Web Browser
URL
http://10.0.0.2
Go
Name: Pulkit Raina
USN: 1BM21CS148

9. To construct simple LAN and understand the concept and operation of Address Resolution Protocol

Q. Construct a LAN & understand the concept & operation of address resolution protocol.



A) Configure IPs of all PCs

PC0 (cmd)
- arp -a
- ping 10.0.0.2
- arp -a
- arp -d

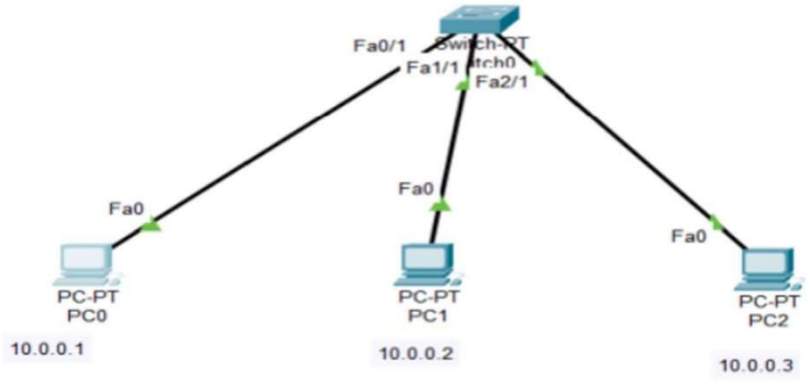
→ Output
* arp -a
No ARP entries found
* ping 10.0.0.2
Reply from 10.0.0.2 byte = 32 time=0ms TTL=128
" " " " " " " TTL = "
" " " " " " " TTL = "
" " " " " " " TTL = "
0

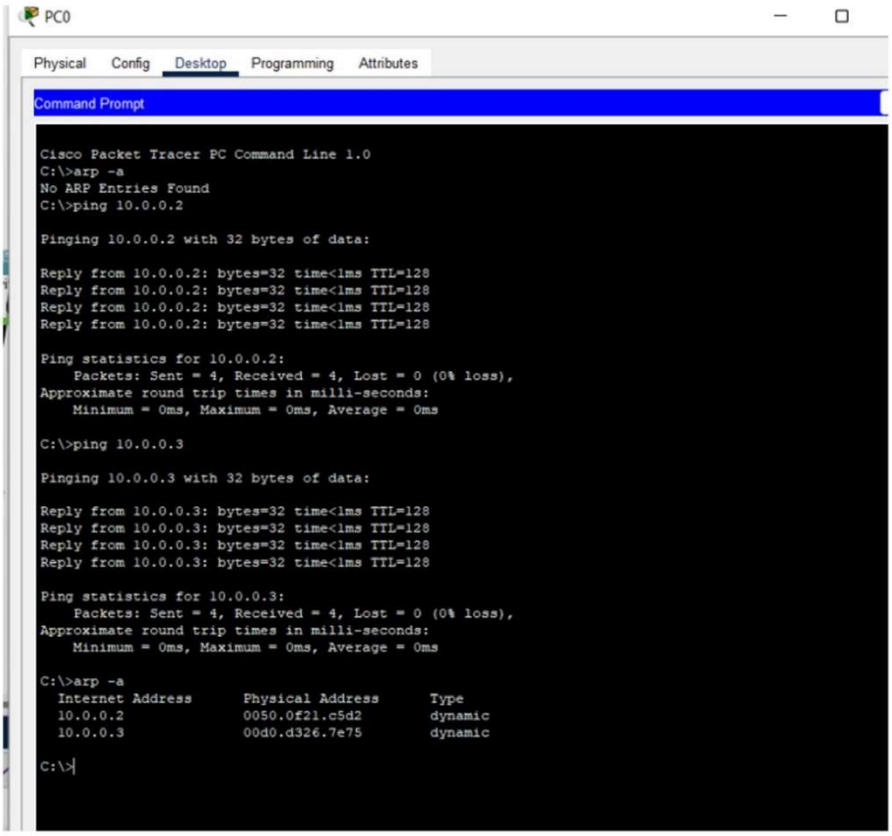
All topics not completed

* arp -a
Internet address physical address types
10.0.0.2 00.02.16[illegible].7820 dynamic

8/10
22/8/23

1BM21CS148


Fa0/1
Switch-PT
Fa1/1
Fa2/1
Fa0
Fa0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
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=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
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:\>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms 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
C:\>arp -a
Internet Address
Physical Address
Type
10.0.0.2
0050.0f21.c5d2
dynamic
10.0.0.3
00d0.d326.7e75
dynamic
C:\>

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

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

Topology:

Procedure:

1. Create the topology
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 cmds to create hostname, password.

```
Router > enable
Router # config t
Router(config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # interface fa 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 = 1ms TTL =
Reply from 10.0.0.2: bytes = 32 time = 2ms TTL =
Reply from 10.0.0.2: bytes = 32 time = 1ms TTL =
Reply from 10.0.0.2: bytes = 32 time = 1ms TTL =

ping statistics for 10.0.0.2:
packets: Sent = 4, received = 4, lost = 0 (0% loss)
Approximate round trip time in ms:
Minimum = 0, maximum = , Average =

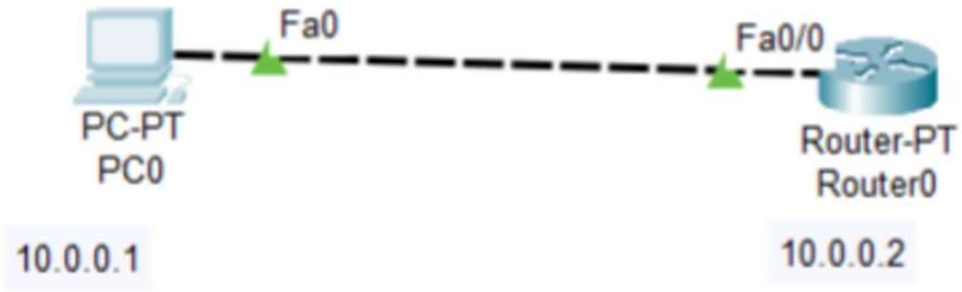
PC > telnet 10.0.0.2
trying 10.0.0.2 ... open
User Access Verification
Password: p0
R1 > enable
password: p1
R1 # show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile
Gateway of last resort is not set.

C 10.0.0.0/8 is directly connected, Fa 0/0

8/10

Observation: The admin in PC can run router commands in PC & get router.

1BM21CS148



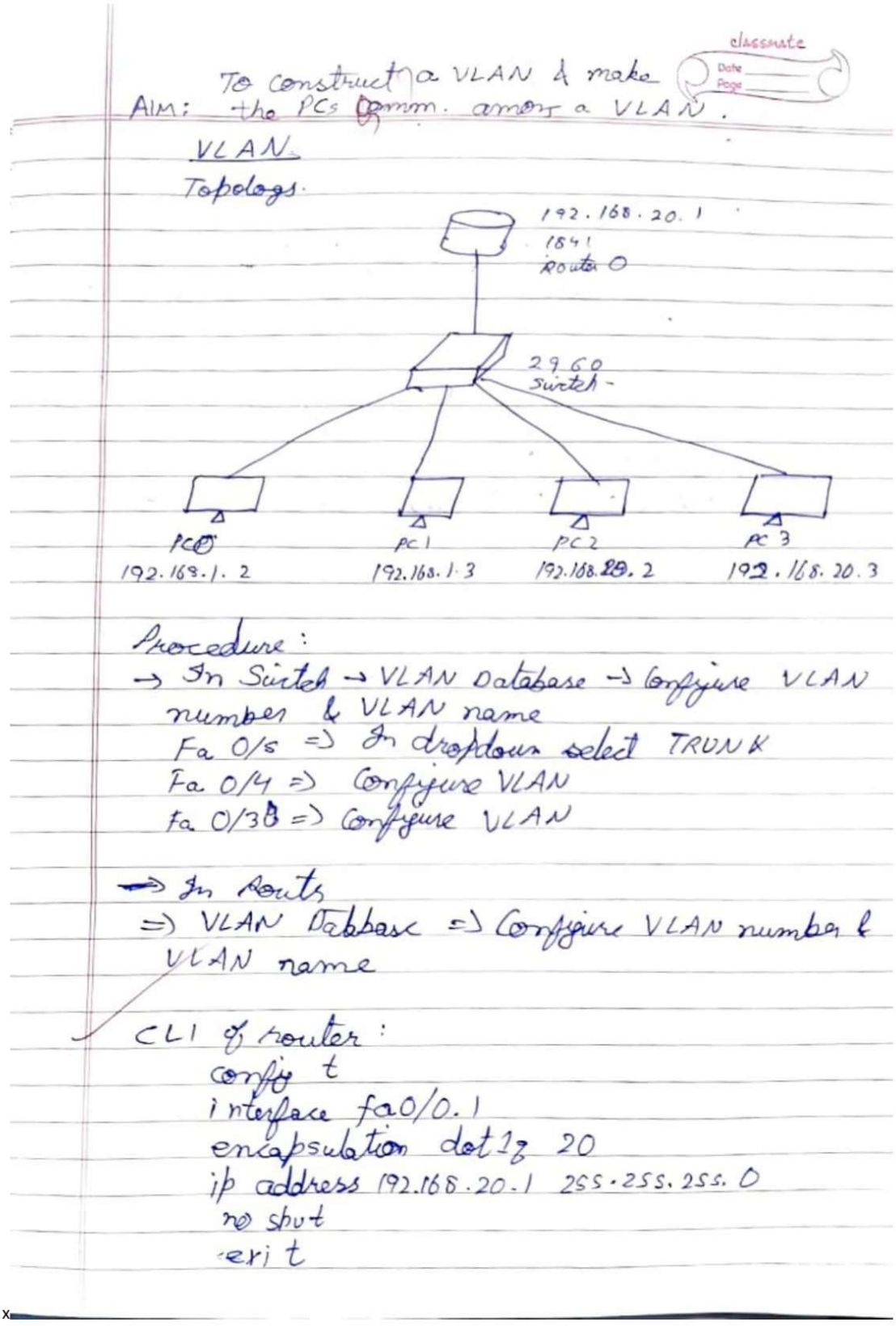
PC0

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#
```

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



Output:

Ping 192.168.20..3

Pinging 192.168.20.3 with 32 bytes of data:

Reply from 192.168.20.3 bytes=32 time=0ms TTL=127

" " " " " " " " "

" " " " " " " " "

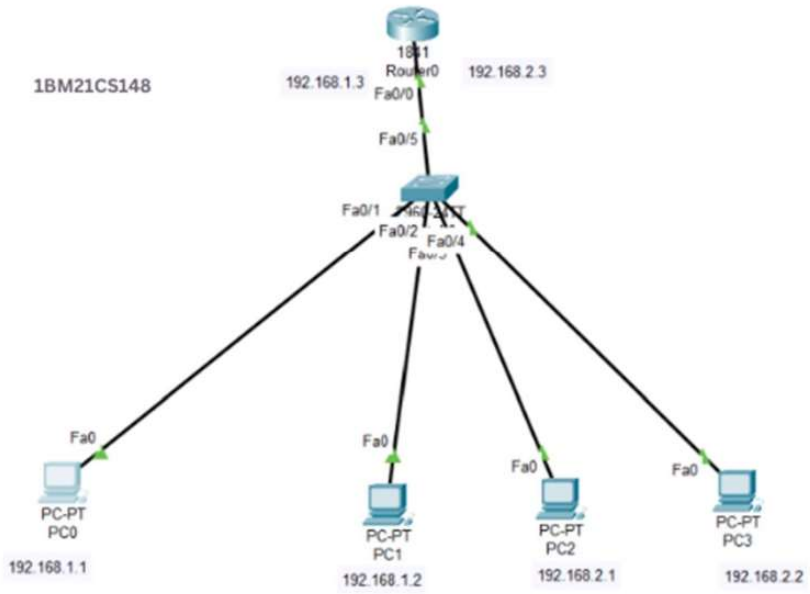
" " " " " " " " "

Ping statistics for 192.168.20.3

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

8/10

22/8/23



PC0

Physical Config Desktop Programming Attributes

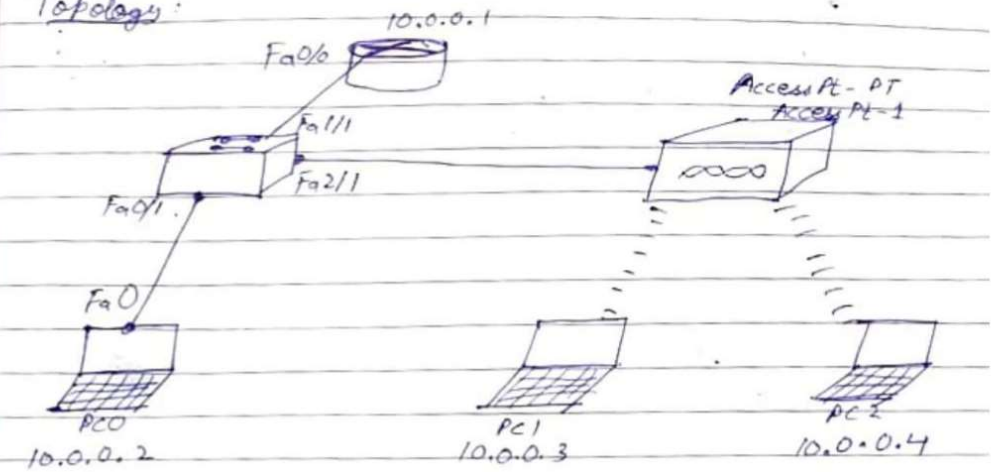
Command Prompt

```
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

C:\>
```

12. To construct a WLAN and make the nodes communicate wirelessly

To construct a WLAN & make the nodes comm. wirelessly

Topology:



Procedure:

① Create the topology as shown above with PC, switch & router & Access Pt connected directly (wired conn.)

② Configure PC0 & router as normally done.

③ Config. the Accesspt.1, Go to port1. & Given SSID name - any name

④ Select WEP & give any 10 digit hex key - eg: 1234567890

⑤ Config PC4 & Laptop with wireless standards

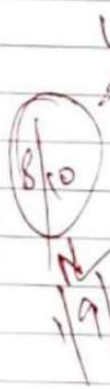
⑥ Switch off the device, Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to empty port. Switch on the device.

⑦ In the config tab, a new wireless interface would have been added. Now config. SSID, WEP, WEP Key, IP address & gateway (as normally done) to the device.

OBSERVATION:

Wireless connection is established with the PCs & the Laptop.

OUTPUT:



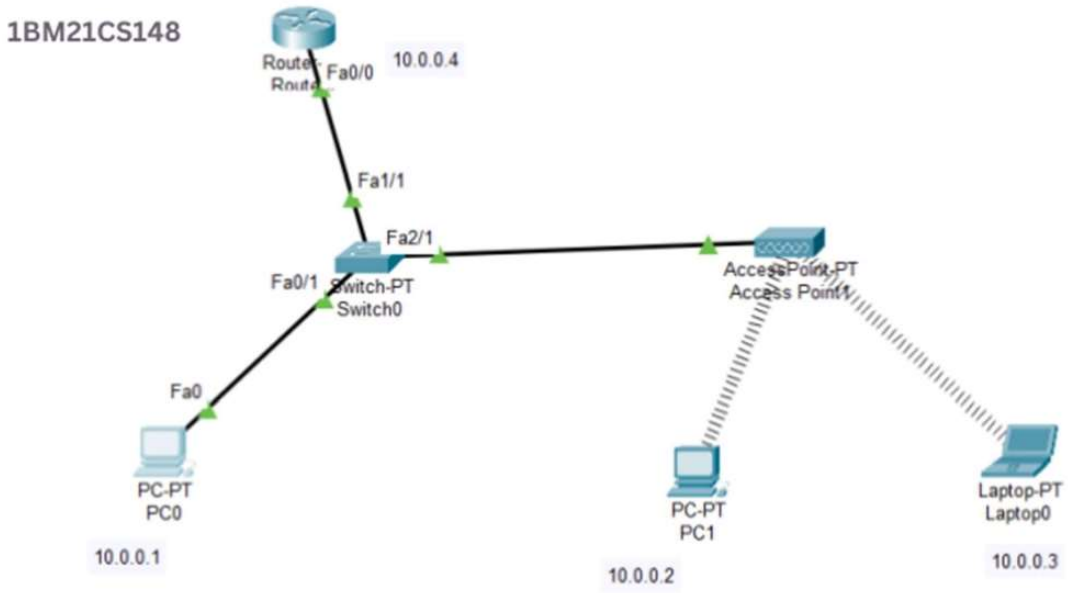
ping 10.0.0.3

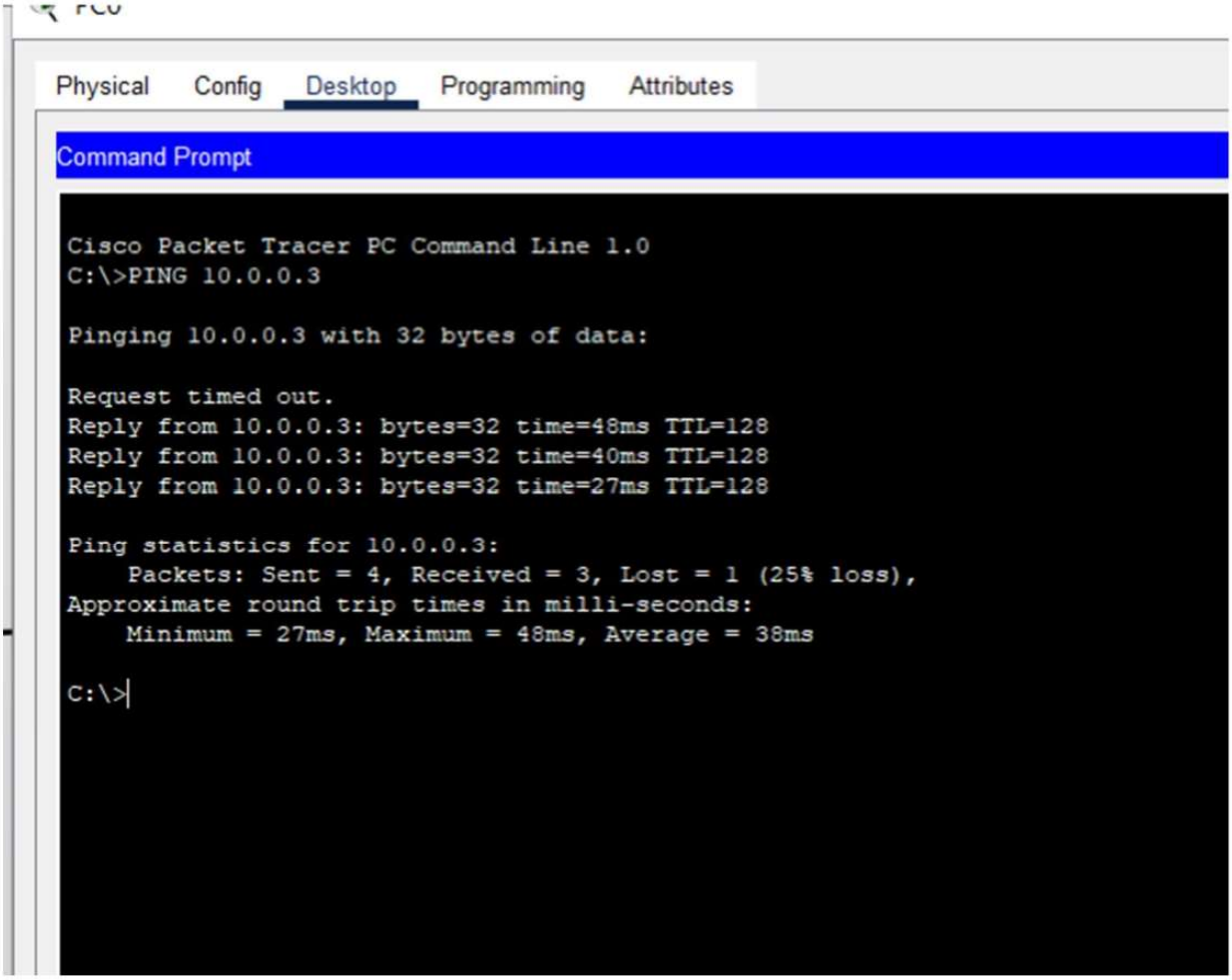
Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3 with 32 bytes of time = 15ms TTL = 128
Reply from 10.0.0.3 with 32 bytes time = 15ms TTL = 128
Reply from 10.0.0.3 with 32 bytes time = 15ms TTL = 128
Reply from 10.0.0.3 with 32 bytes time = 15ms TTL = 128

Ping statistics for 10.0.0.3

Packets sent = 4, received = 4, lost = 0 (0% loss)


1BM21CS148
Router
Fa0/0
10.0.0.4
Fa1/1
Fa2/1
Fa0/1
Switch-PT
Switch0
Access Point-PT
Access Point1
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
Laptop-PT
Laptop0
10.0.0.3


Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>PING 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 10.0.0.3: bytes=32 time=48ms TTL=128
Reply from 10.0.0.3: bytes=32 time=40ms TTL=128
Reply from 10.0.0.3: bytes=32 time=27ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 27ms, Maximum = 48ms, Average = 38ms
C:\>

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

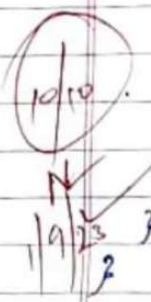
CYCLE-2.

Write a prog. to detect error using CRC-CCITT(16 bit

```
#include<stdio.h>
#include<conio.h>

char m[50],g[50],r[50],q[50],temp[50];

void crc(int n){
  int i,j;
  for(i=0;i<n;i++)
     temp[i]=m[i];
  for(i=0;i<n-16;i++){
     if(r[0]=='1'){
        g[i]='1';
        cdram();
     }
     else{
        g[i]='0';
        shift();
     }
     r[16]=m[17+i];
     r[17]='\0';
     for(j=0;j<=17;j++)
        temp[j]=r[j];
  }
  g[n-16]='\0';
}

void cdram(){
  int i,j;
  for(i=1;i<=16;i++)
     r[i-1]=((int)temp[i]-48)^((int)g[i]-48)+48;
}
```

```
void shift(){
    int i;
    for(i=1; i<=16; i++)
        r[i-1] = r[i];
}

void crctrans(int n){
    int i, k = 0;
    for(i=n-16, i<n; i++){
        m[i] = ((int) m[i]-48) ^ ((int)r[k+1]-48)+48;
    }
    m[i] = '\0';
}

void main()
    int n, j=0;
    char ch; int flag = 0;

    printf("Enter the frame bits: ");
    while((ch == getc(stdin)) != '\n'){
        m[j++] = ch;
        n = i;
    for (i=0; i<16; i++)
        m[n++] = '0';
        m[n] = '\0';

    printf("Msg after appending 16 zeroes: %s", m);

    for(i=0; i<16; i++)
        g[i] = '0';
        g[0] = g[4] = g[11] = g[16] = '1'; g[17] = '\0';

    printf("In 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 check");
crc(n);
printf("\n last Remainder: %s", s);

for(i=0; i<16; i++){
  if(n[i] != '0')
      flag = 1;
  else
      continue;
  if(flag == 1)
      printf("Error during transmissi!");
  else
      print("Received frame is correct!");
}
```



OUTPUT:

Enter frame bits: 1011

Msg after appending 16 zeroes: 1011 0000 0000 0000 0000

Generator: 1000 1000 0001 0000 1

Quotient: 1011

Transmitted frame: 1011 1011 0001 0110 1011

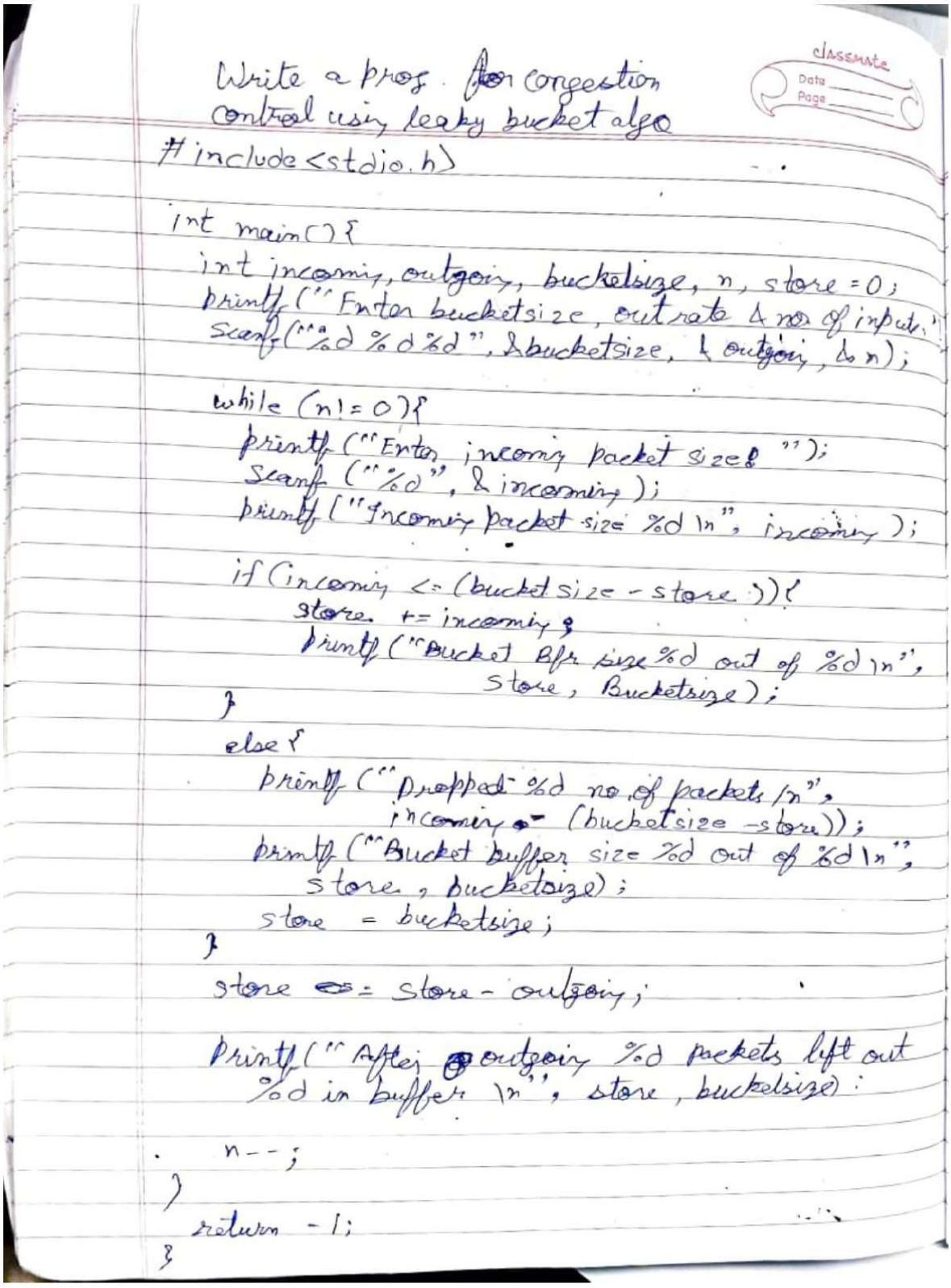
Enter transmitted frame: 1011 1011 0001 0110 0111

last remainder: 0000 0000 0000 0000

Received frame is correct.

```
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
```

14. Write a program for congestion control using Leaky bucket algorithm



Write a prog. for congestion control using leaky bucket algo

```
#include<stdio.h>

int main(){
  int incoming, outgoing, bucketsize, n, store = 0;
  printf("Enter bucketsize, out rate & no of inputs: ");
  scanf("%d %d %d", &bucketsize, &outgoing, &n);

  while (n != 0){
    printf("Enter incoming packet size: ");
    scanf("%d", &incoming);
    printf("Incoming packet size %d \n", incoming);

    if (incoming <= (bucketsize - store)){
      store += incoming;
      printf("Bucket Bfr size %d out of %d \n",
                store, Bucketsize);
    }
    else{
      printf("Dropped %d no of packets \n",
                incoming - (bucketsize - store));
      printf("Bucket buffer size %d out of %d \n",
        store, bucketsize);
      store = bucketsize;
    }
    store = store - outgoing;

    Printf("After outgoing %d packets left out
      %d in buffer \n", store, bucketsize);

    n--;
  }
  return -1;
}
```

OUTPUT.

Enter bucket size, out rate & no. of input: 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

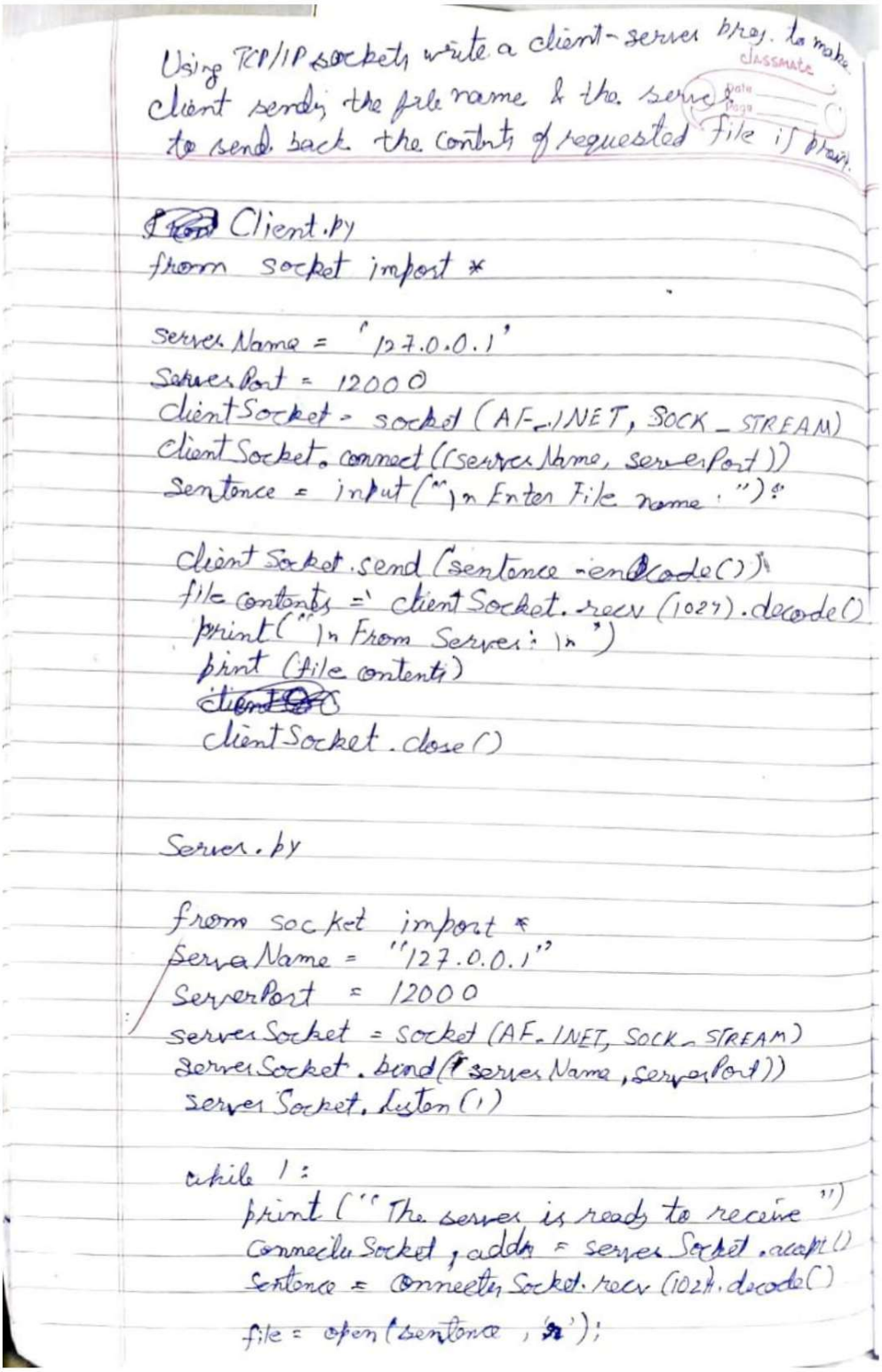
Enter the incoming packet size: 10
Incoming packet size 10
Bucket buffer size 10 out of 20
After outgoing 10 packets left out of 20 in buffers.

9/10

5. Case

```
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
```

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



Using TCP/IP sockets write a client-server prog. to make client sends the file name & the server to send back the contents of requested file if present.

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()
```

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())
print("\n Sent contents of " + sentence).
file.close()
connectionSocket.close()
```

bind() → bind() method binds a server to a specific IP & port so that it can listen to incoming requests on that IP & port.

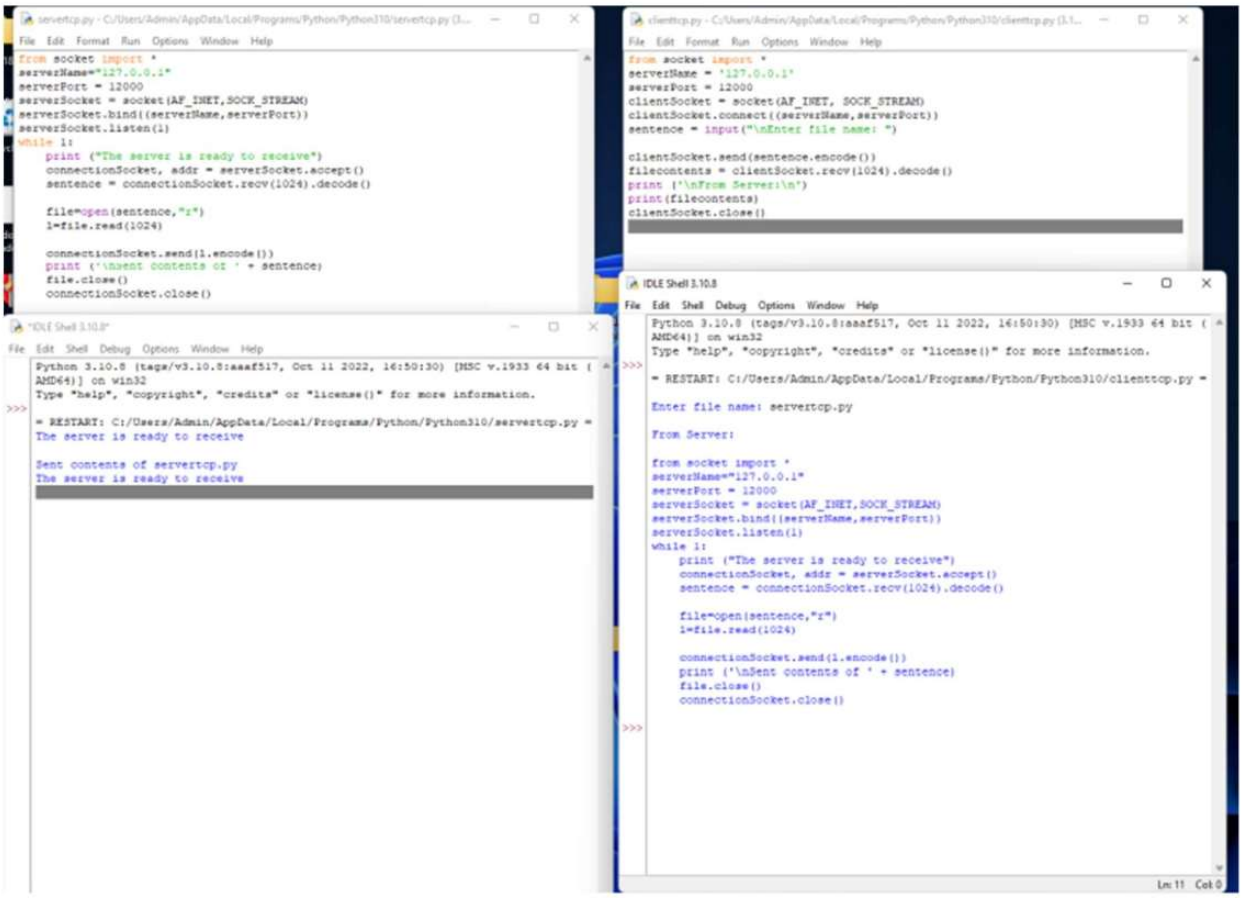
accept() => initiate a connection with client & the close method closes the connection with client.

Client Output:

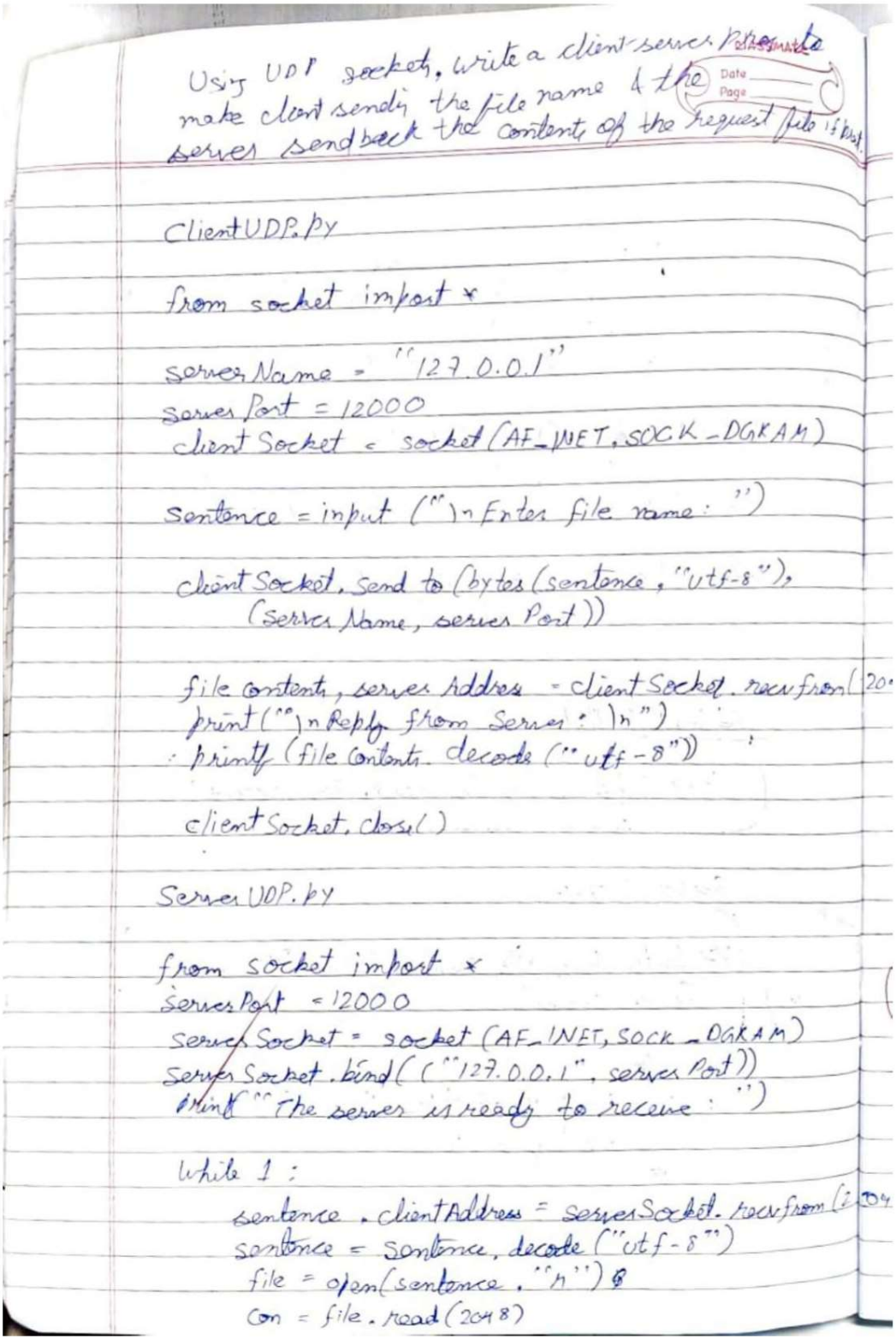
```
>>> The server is ready to receive
```

Server Output:

```
Enter the file name: ServerTCP.py
from server!

/*
    Contents of the file will be
    displayed here
*/
```


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

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

ClientUDP.py

```
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(20
print("\nReply from Server: \n")
print(fileContents.decode("utf-8"))

clientSocket.close()
```

ServerUDP.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(2
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
```

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

print("\n Sent Content of ", end = ' ')
print(sentence)

file.close()
```

The server is ready to receive

Sent content of server UDP.py
The server is ready to receive

2048)

Server UDP output

Enter the file name: ServerUDP.py

reply from server:

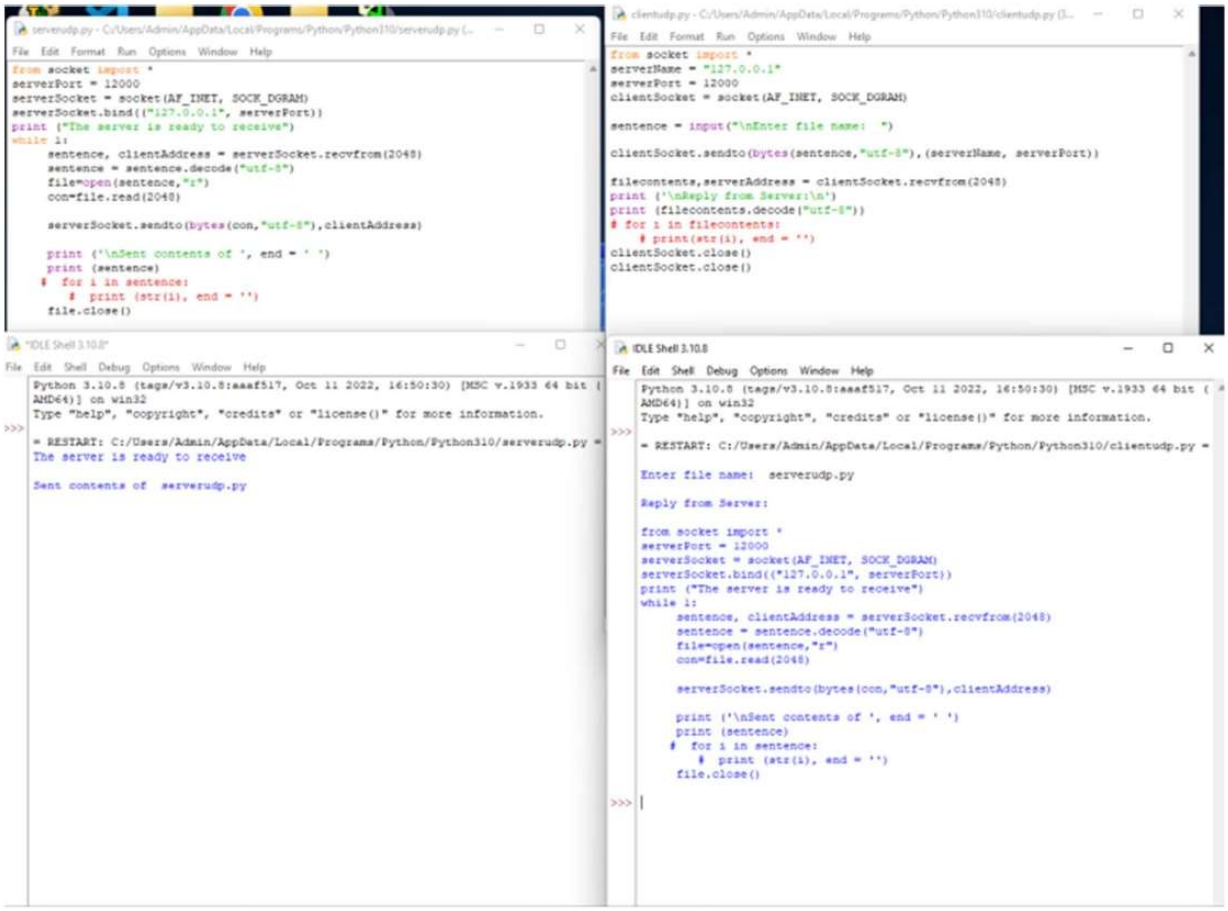
```
/*
   Contents of the file are
   displayed here
*/
```

10/10

9/9/22

048)

17.


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()
>>>