VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Computer Network

Submitted by

GAGAN D A (1BM21CS063)

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
JUN-2023 to SEP-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 Network" carried out by **GAGAN D A (1BM21CS063),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Network - (22CS4PCCON)** work prescribed for the said degree.

Dr.Latha N R
Assistant Professor
Department of CSE
BMSCE, Bengaluru

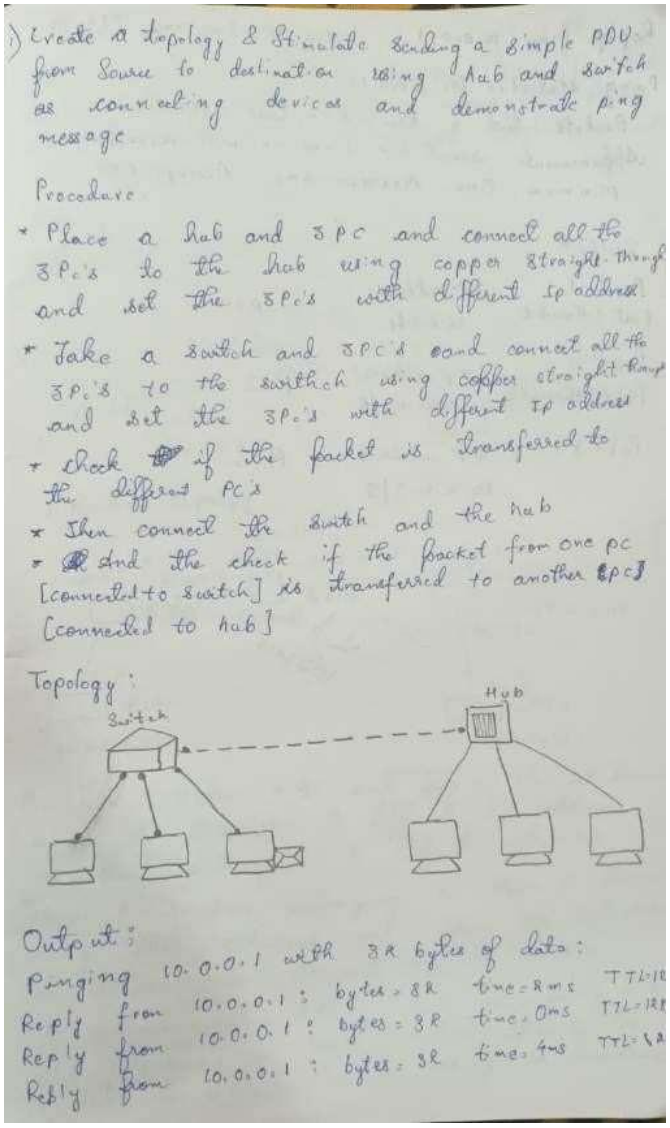
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

LAB-1

Create a topology and simulate sending a simple PDU from sourceto destination using hub and switch as connecting devices and



1) Create a topology & Stimulate sending a simple PDU from source to destination using Hub and switch as connecting devices and demonstrate ping message

Procedure

* Place a hub and 3 PC and connect all the 3 PC's to the hub using copper straight through and set the 3 PC's with different IP address
* Take a switch and 3 PC's and connect all the 3 PC's to the switch using copper straight through and set the 3 PC's with different IP address
* Check if the packet is transferred to the different PC's
* Then connect the switch and the hub
* And the check if the packet from one pc [connected to switch] is transferred to another pc [connected to hub]

Topology:

Output:

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1 : bytes = 32 time = 8ms TTL=128

Reply from 10.0.0.1 : bytes = 32 time = 0ms TTL=128

Reply from 10.0.0.1 : bytes = 32 time = 4ms TTL=128

Reply from 10.0.0.1 : bytes = 32 time 4ms TTL = 128

Ping statistics for 10.0.0.1 :

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

Approximate round trip times in milli seconds :

Minimum = 0ms, Maximum = 4ms, Average = 1ms

Switch

Port-1 — IP address 10.0.0.4 — PC1

Fast Ethernet 0

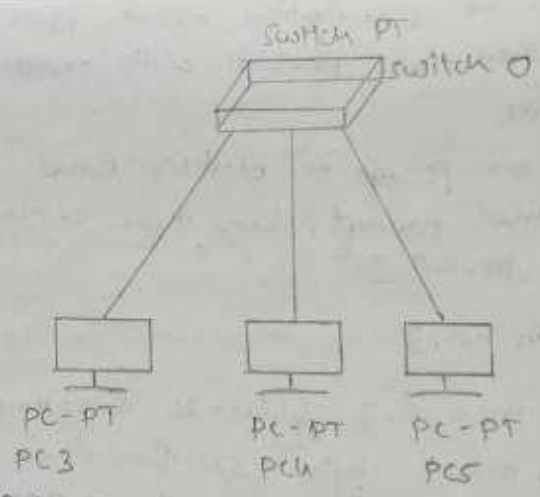
Port-2 — IP address 10.0.0.2/8 — PC2

Fast Ethernet 0

Port-3 — IP address 10.0.0.3/8 — PC3

15/6/2023

Switch



procedure:-

Step1 :- Select switch and 3PC's.

Step2 :- Set IP address for all the pc's
10.0.0.4, 10.0.0.8, 10.0.0.6
Pc → catalog → fast ethernet → Ip address.

Step 3 :- Connect Pc's to the switch by selecting copper straight through.

Step 4 :- Add simple PDU.
Select source and destination.

Step 5 :- Go to simulation mode and click on Auto capture / play.

Step 6 :- Click on PC → Desktop → command prompt

ping message

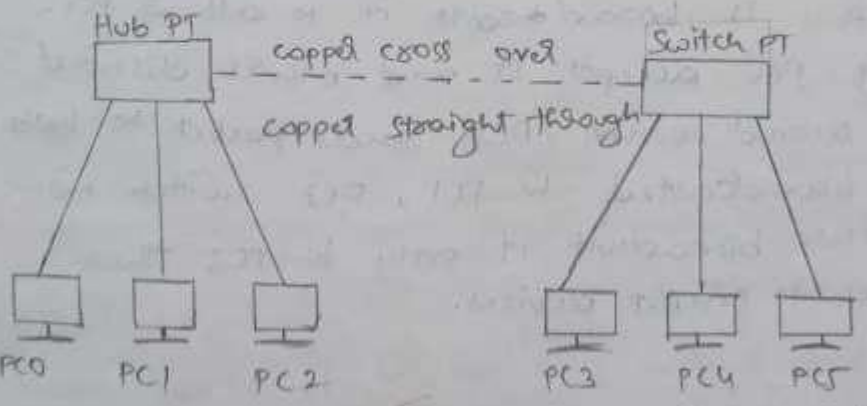
```
PC> ping 10.0.0.6
Pinging 10.0.0.6 with 32 bytes of data.
Reply from 10.0.0.6 bytes=32 time=1ms TTL=128
Reply from 10.0.0.6 bytes=32 time=1ms TTL=128
Reply from 10.0.0.6 bytes=32 time=1ms TTL=128
Reply from 10.0.0.6 bytes=32 time=1ms TTL=128
```

Ping Statistics for 10.0.0.6
packets sent = 4 Received = 4 Lost = 0 (0% loss)
Approximate round trip times in milli-seconds
minimum = 4ms Maximum = 4ms Average = 4ms

Observation:-

When the first time the packet is sent the switch will distribute the packet with all the devices. Once it learns about the IP address it will only send packet to the destination and send acknowledgement to the source.

Switch-Hub Connection:-



Step 1:- Previously drawn hub-topology and switches topology are connected through copper cross over. In hub port 3 is used in switch port ethernet 3/1 is used.

Step 2:-Add simple PDC from PC0 to PC3

ping 10.0.0.4.

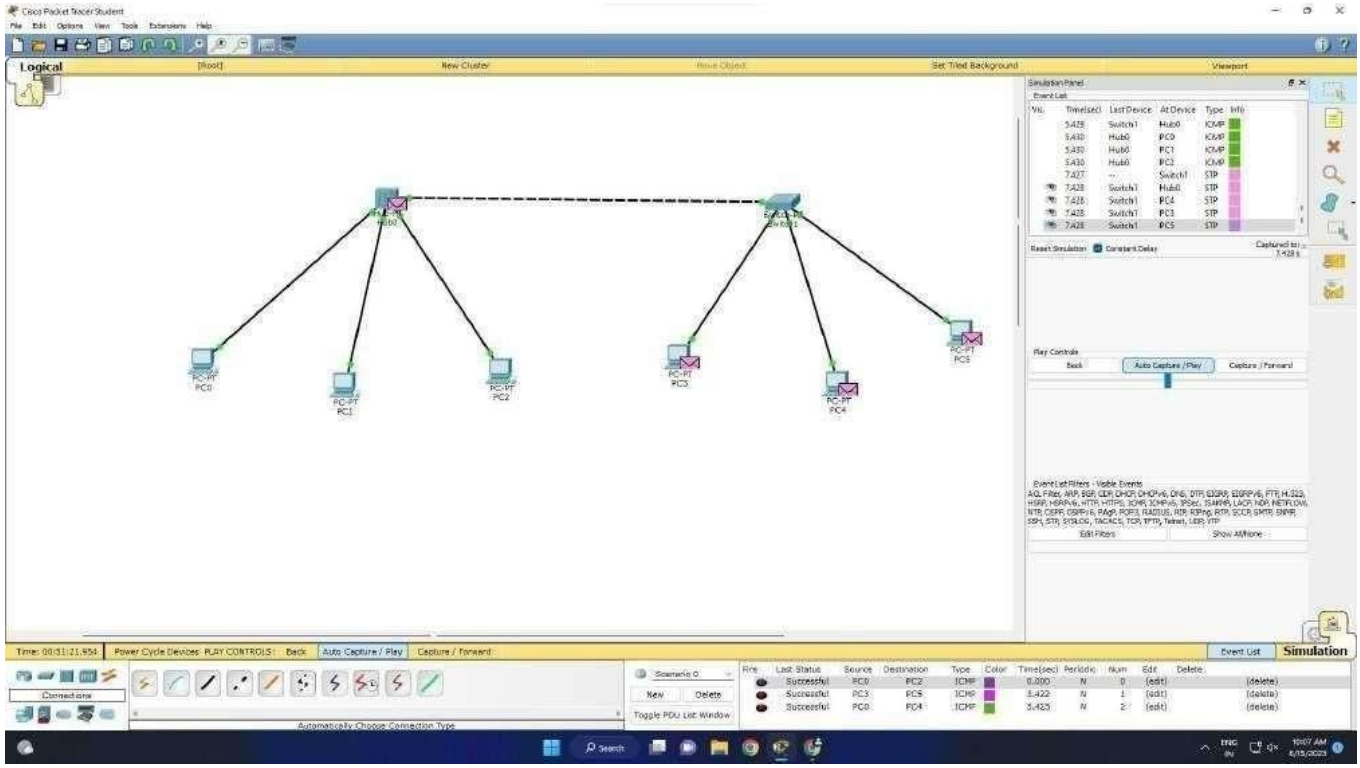
Pinging 10.0.0.4 with 32 bytes of data
Reply from 10.0.0.4 : bytes=32 time=1ms TTL=128
Reply from 10.0.0.4 : bytes=32 time=1ms TTL=128

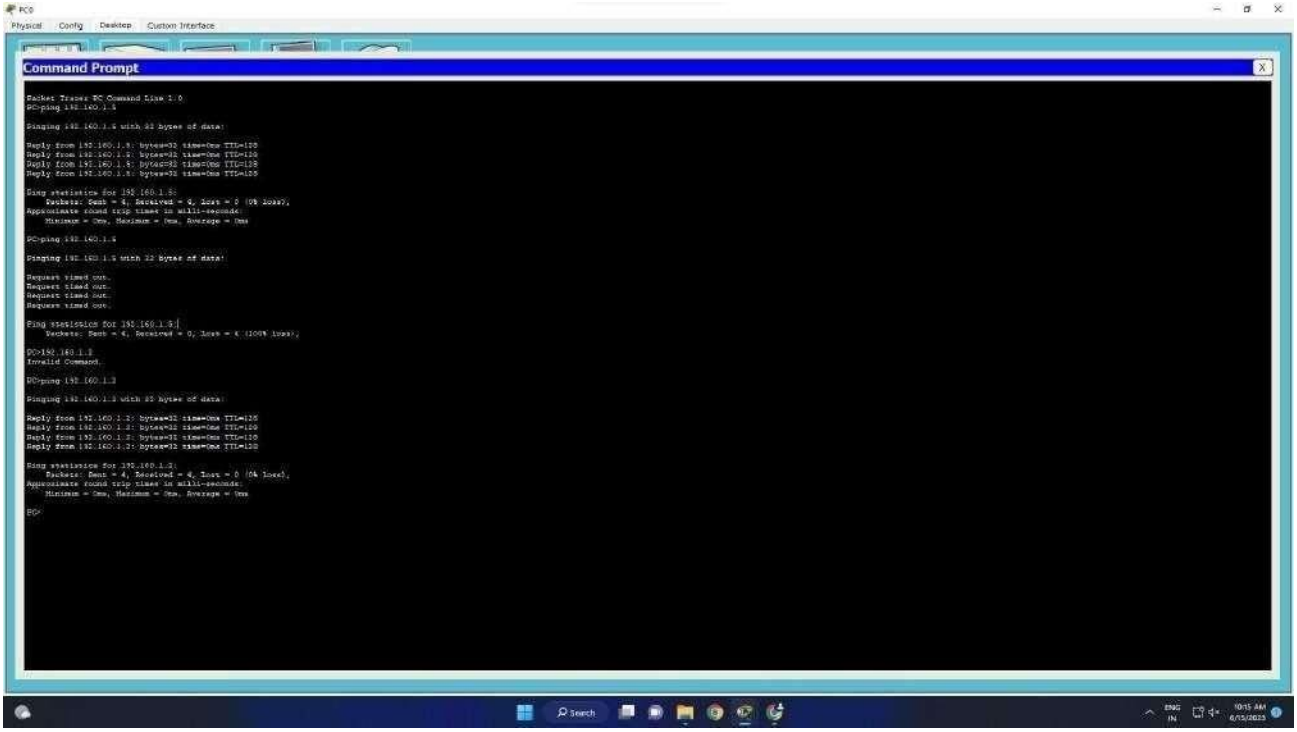
Reply from 10.0.0.4 : bytes = 32 time = 1ms TTL = 128
Reply from 10.0.0.4 : bytes = 32 time = 1ms TTL = 128
Ping satisfies for 10.0.0.4.
Packets : sent = 4 Received = 4 Lost = 0 (0% loss)
Approximate round trip times in milli seconds
Minimum = 4ms Maximum = 4ms Average = 4ms

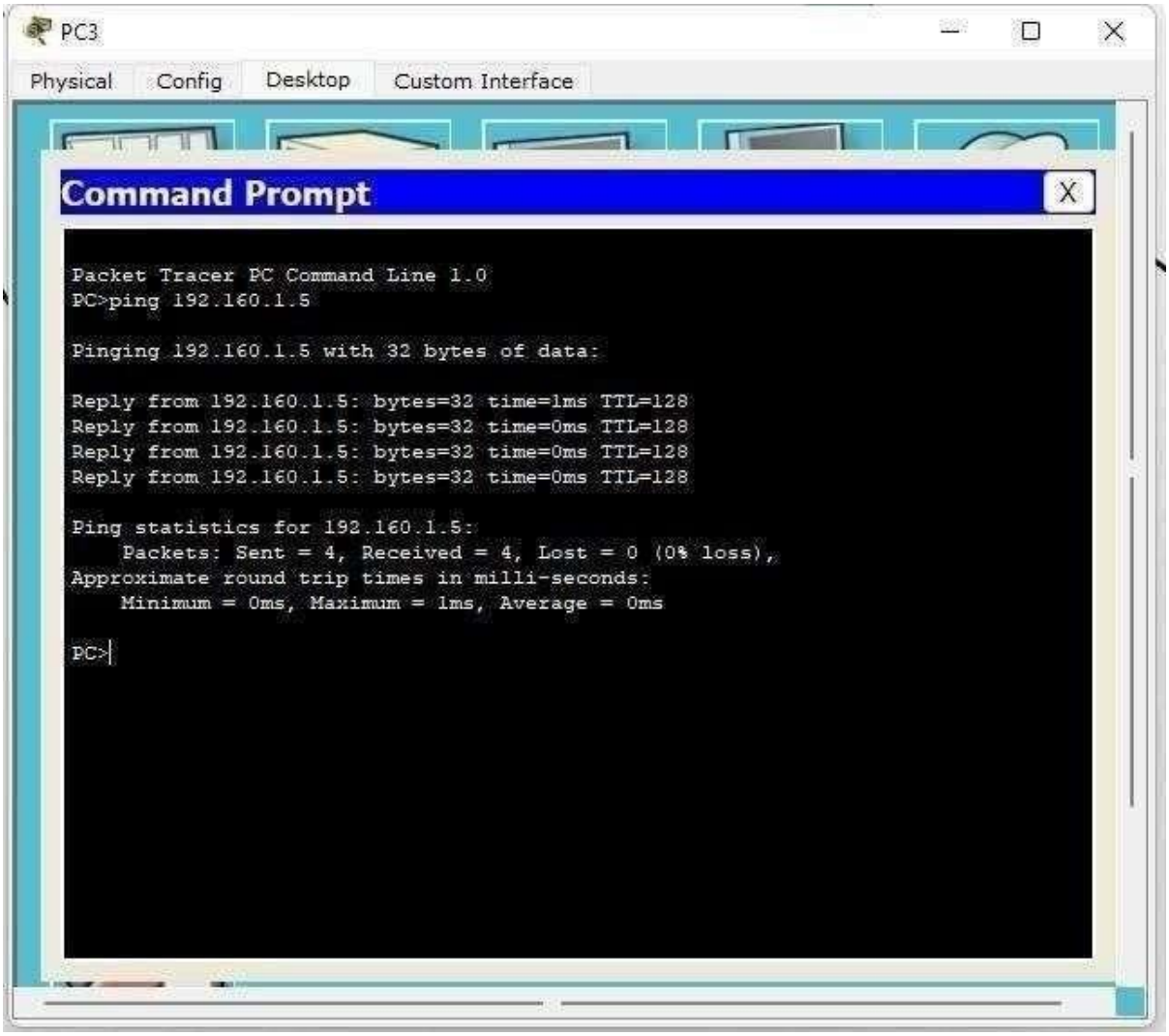
Observation:-

In simulation mode PC0 sends packet to hub sends it to PC1, PC2 & switch board cast it to PC3, PC4 and PC5.
PC1, PC2, PC4 and PC5 discards them, PC3 accepts and sends acknowledgement to hub through switch.
Hub is broad-casts it to all 3 PCs.
Only PC0 accepts it and others discard
In second round PC0 sends packet to hub & broadcasted to PC1, PC2 switch now switch broadcast it only to PC3, Thus switch is smart devices.

OUTPUT:



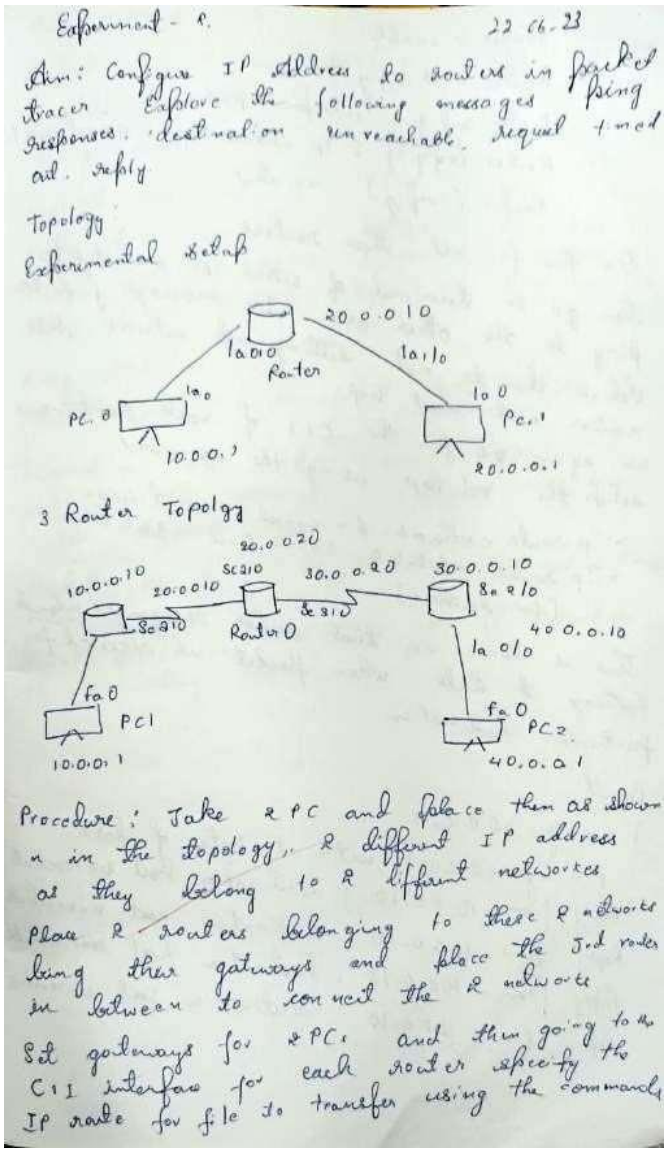



PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=1ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

LAB 2

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

OBSERVATION:



router > enable
Router > # config terminal
Router config : interface <port>
Router config-if : Ip address <ip> <subnet mask>
Router (config) : no shut

Do this for all three routers.

Then go to terminal of either PC and try to ping to the other one. The message fails to deliver due to not setting up network static router and next hop.

We again go for the CLI of each router and setup the "nxt hop" using the command.

> ip route <network-id> <mask> <next-hop>
> ip route 40.0.0.0 255.0.0.0 20.0.0.20
(for router 1)

This is done so that router recognizes which pathway to take when packet is received for particular destination

Result :

:) > ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data
Reply from 10.0.0.10 : Destination host unreachable
Reply from 10.0.0.10 : Destination host unreachable
Reply from 10.0.0.10 : Destination host unreachable
Reply from 10.0.0.10 : Destination host unreachable

Ping statistics
Packets sent = 4 ; Received = 0 ; loss = 4 (100% loss

:) > ping 40.0.0.1

Ping 40.0.0.1 with 32 bytes of data

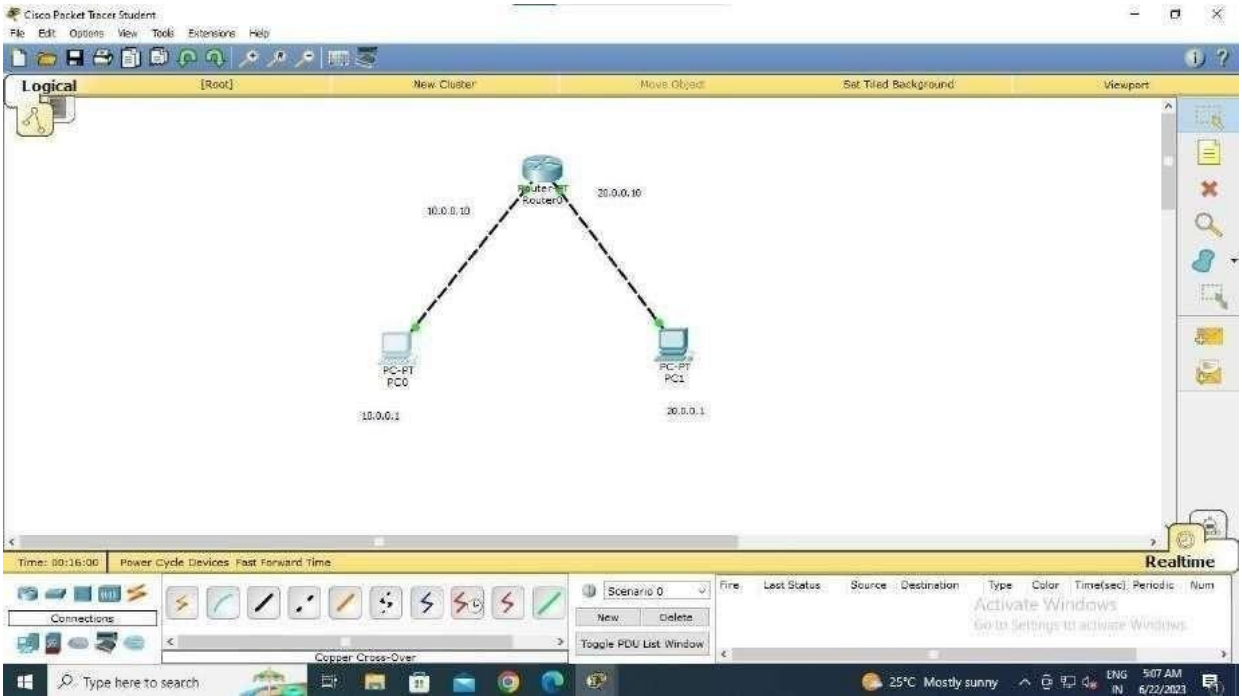
Request timed out

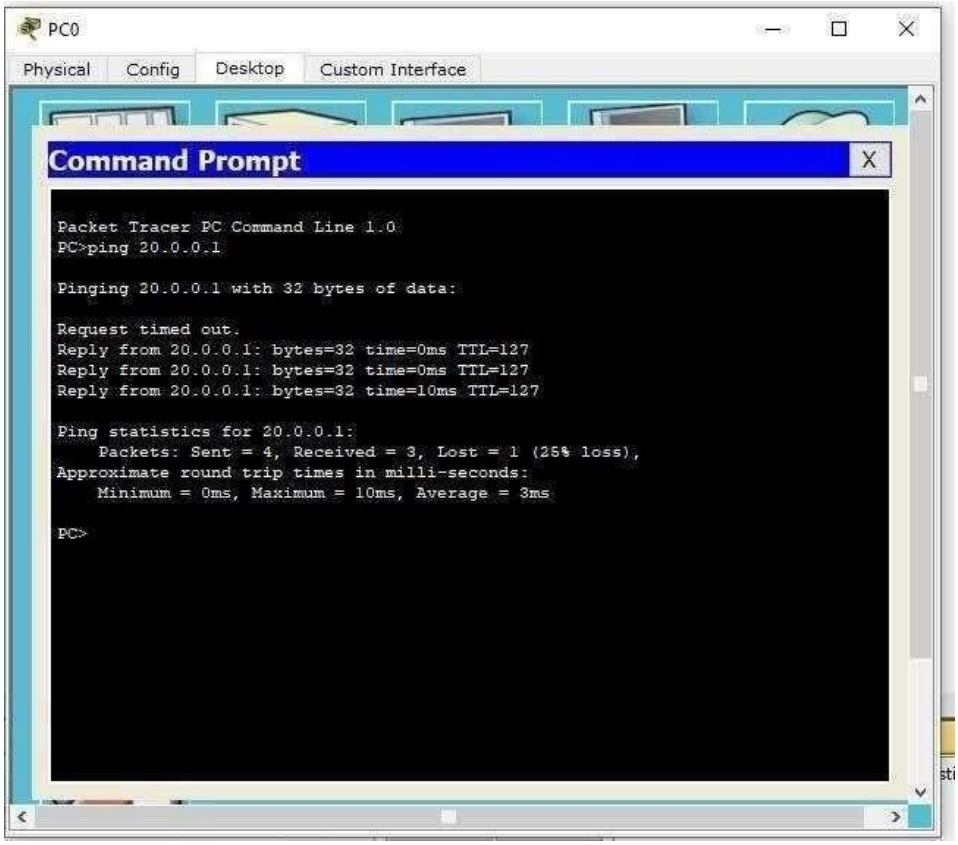
Reply from 40.0.0.10 bytes = 32 time = 2ms TTL
Reply from 40.0.0.10 bytes = 32 time = 2ms TTL
Reply from 40.0.0.10 bytes = 32 time = 2ms TTL

Ping statistics
Packets sent = 4 ; Recieved = 1, loss = 1 (25% loss)

Observation: The router connects LAN to the internet It connects different networks with different loss.
Packets are forwarded to the destination through network hopping.
Serial ports are used to connect 2 routers the connecting cable

Topology


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router0
10.0.0.10
20.0.0.10
PC-PT PC0
PC-PT PC1
10.0.0.1
20.0.0.1
Time: 00:16:00
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Copper Cross-Over
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Activate Windows
Type here to search
25°C Mostly sunny
ENG IN
5:07 AM 6/22/2023


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>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=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=10ms TTL=127
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 3ms
PC>

Default IP address LAB - 3

Topology Similar to that of Lab-2 that contains 3 router & PC's connected as shown in that

Procedure:

→ Select PC's and router and configure them with suitable IP address

→ Make connections to all the devices using suitable connection links

→ for the routers to link between pc's and other router. use CL2 mode and start typing the commands

→ Now enter Enable → Enter

→ Config T enter

→ Interface fastethernet 0/0

→ IP address 10.0.0.10 255.0.0.0

→ Now shut

Repeat this step similarly to all the routers In order to make default path

→ Type in config T

→ Ip route Destination Subnet Mask Intermediate device.

→ IP router 0.0.0.0 0.0.0.0 20.00.10.

Similarly perform this to all the routers

→ Show IP route enter

10.0.0.0/8 is directly connected to fast ethernet 0/0

C 20.0.0.0/8 is directly connected serial 2/0

0.0.0.0/0 [1/0] via 20.0.0.10

Similarly all the routers are connected

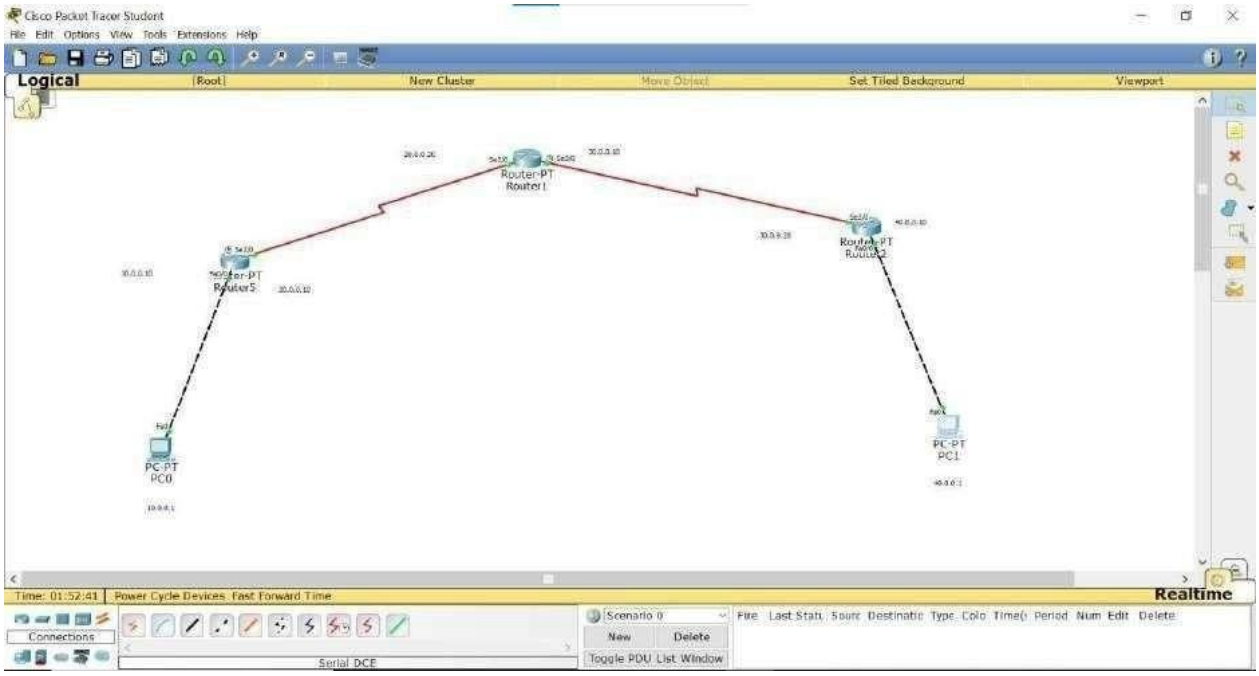
Observation:

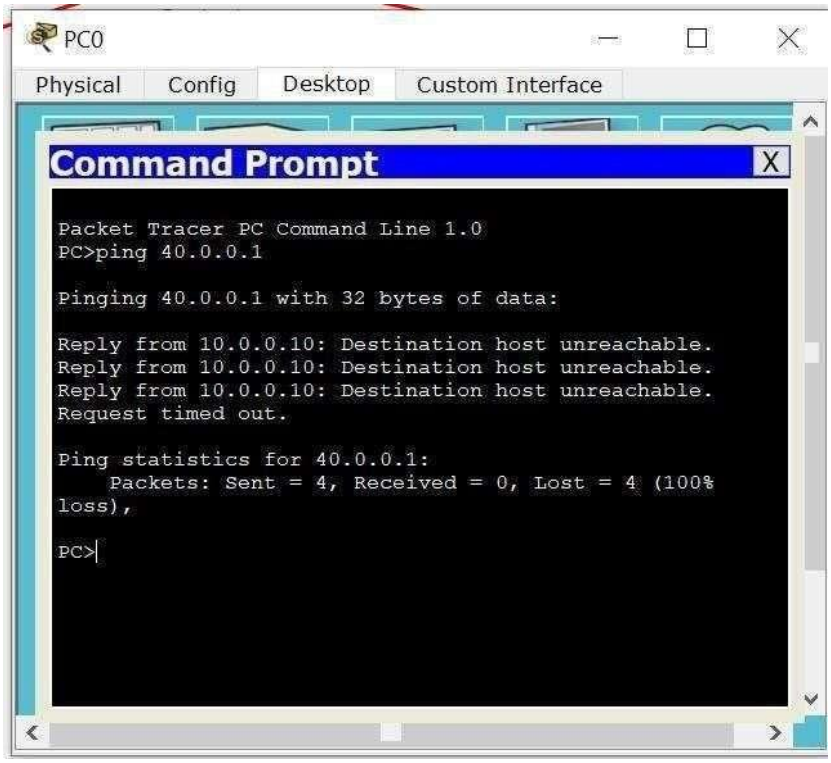
In the previous one, we have given the IP address to all the routers with destination subnet and intermediate IP address of those particular device. But here in this experiment we use a default IP address ie 0.0.0.0 and subnet Mask 0.0.0.0 so that it can create a pass through channel to all the packets that are send will be transferred by the intermediate device. This is generally used in large no of device connection

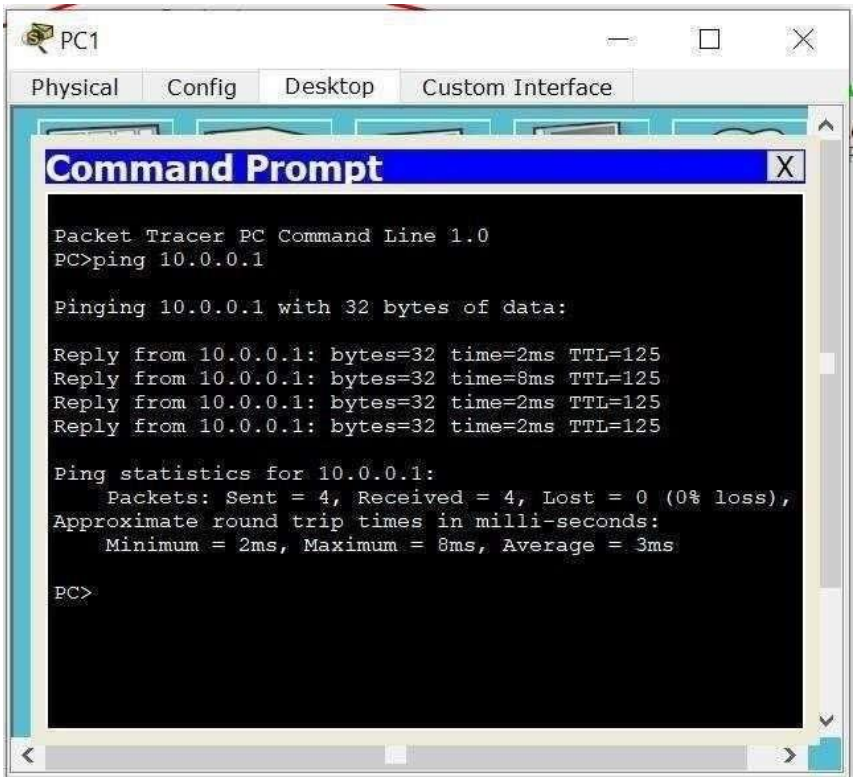
Topology:

17/8

Topology:

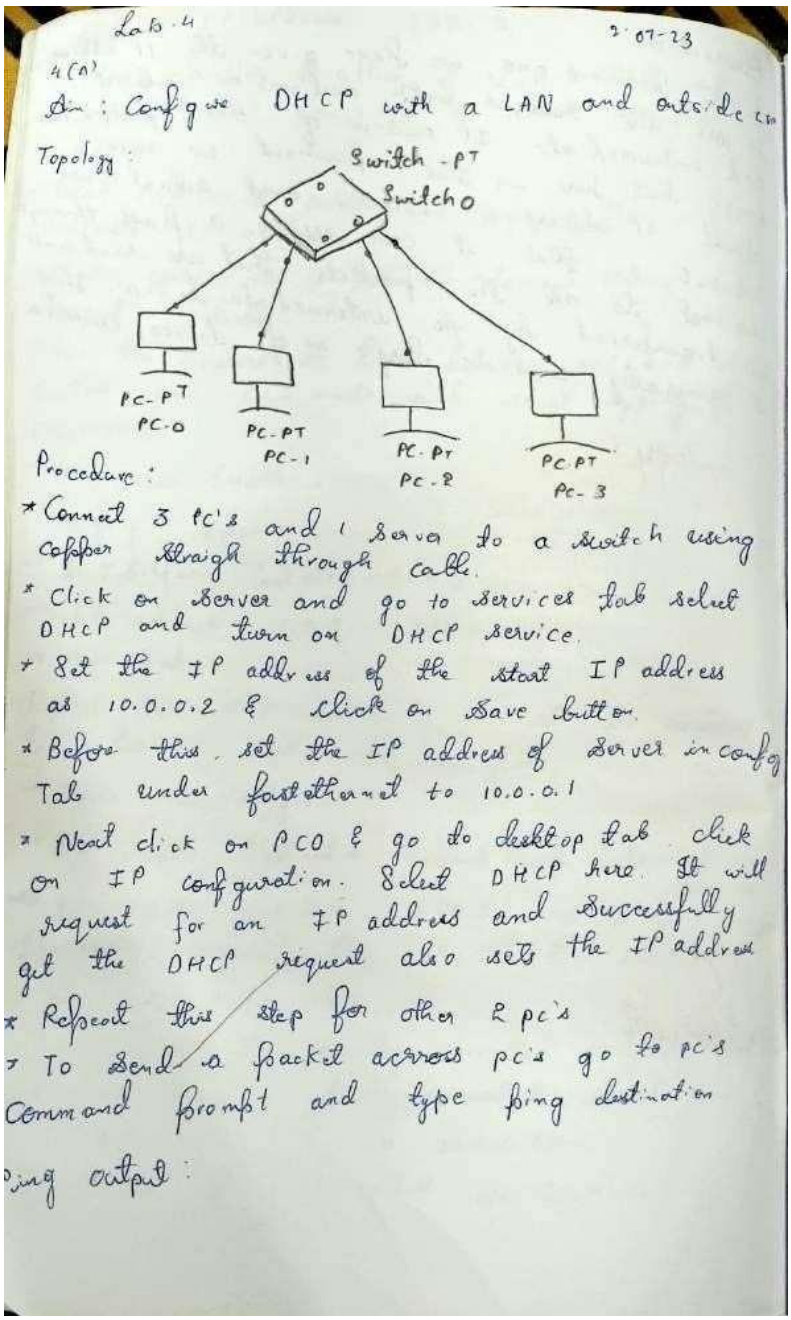

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router5
Router-PT
Router3
PC-PT
PC0
PC-PT
PC1
Time: 01:52:41
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Stat Source Destinatio Type Colo Time(s Period Num Edit Delete


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100%
loss),
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 3ms
PC>

LAB 4

Configure DHCP within a LAN and outside LAN.OBSERVATION:



Packet tracer PC command line 10

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=1ms TTL=128

ping statistics from 10.0.0.3

packets sent = 4 Received 4. Lost = 0 (0% loss)

Approximate round trip times in ms

minimum = 0ms, Maximum = 1ms Average = 0ms.

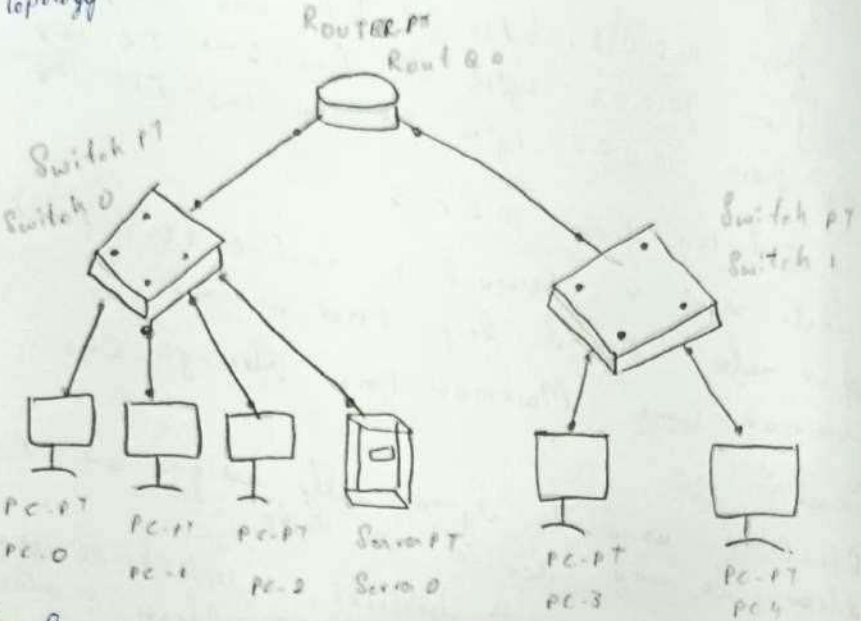
Observation:

* DHCP is used to dynamically assigns an IP address to any device or node
* It is client-server protocol in which servers manage a pool of unique IP address & also about client configuration parameters.
* DHCP enabled clients sends a request to DHCP server when they want to connect to a network.

4 (A)

Aim : Configure DHCP with a LAN and outside LAN.

Topology



Procedure :

Add a router, a switch and 5 pc's to 4 (A) program network & connect the router to both switches

→ Set the server IP address of server & with the help of server set the first 3 PC's IP address through DHCP

→ click on Server

→ go to desktop → IP configuration

→ Add IP address, Subnet mask and gateway

IP address 10.0.0.1

Subnet Mask 255.0.0.0

Gateway 10.0.0.20

Step 3 : Configure the router

→ click on router go to CL & 2

nable
uter # config t.
uter (config) # fastethernet 0/0
outer (config) # ip address 10.0.0.20 255.0.0.0
Router (config if) # no shut
Router (config if) # exit
Router (config) # interface fastethernet 1/0
Router (config if) # ip address 20.0.0.20 250.0.0.0
Router (config if) # exit
exit

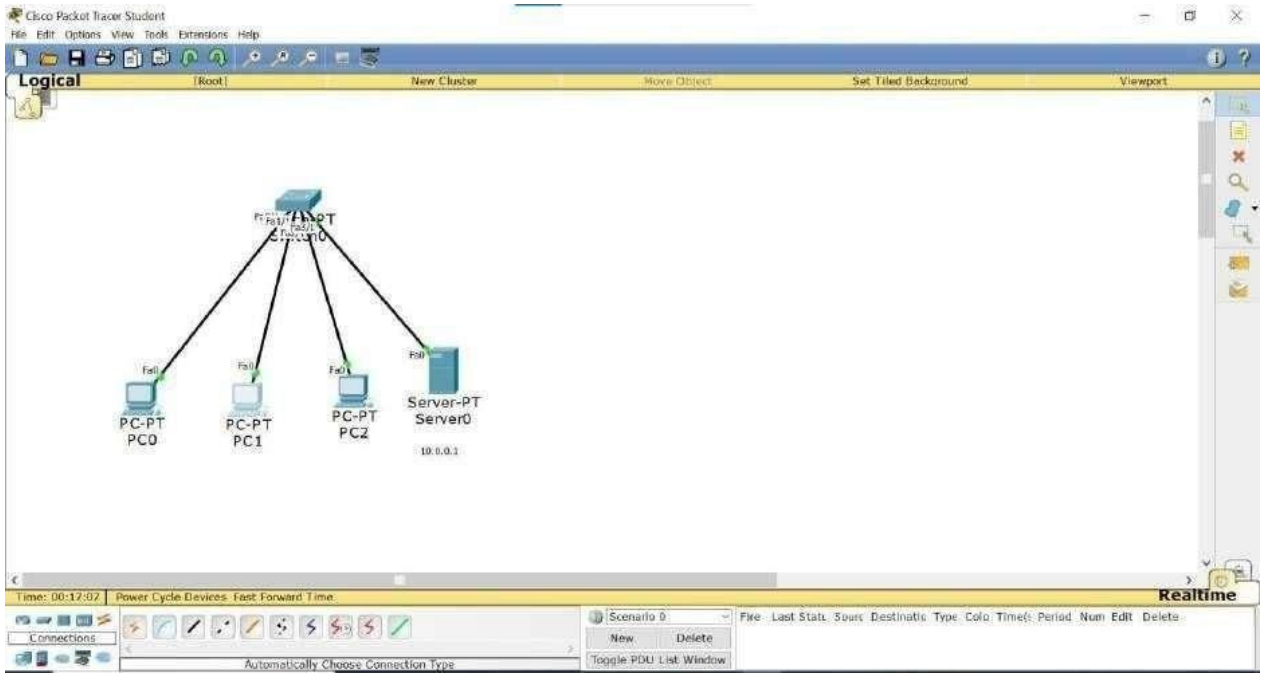
Routing table
Router → show ip route
10.0.0.0/8 is directly connected.

Step 4: Go to Server
→ Select services then go to DHCP
→ set service on
→ set start IP address from the server

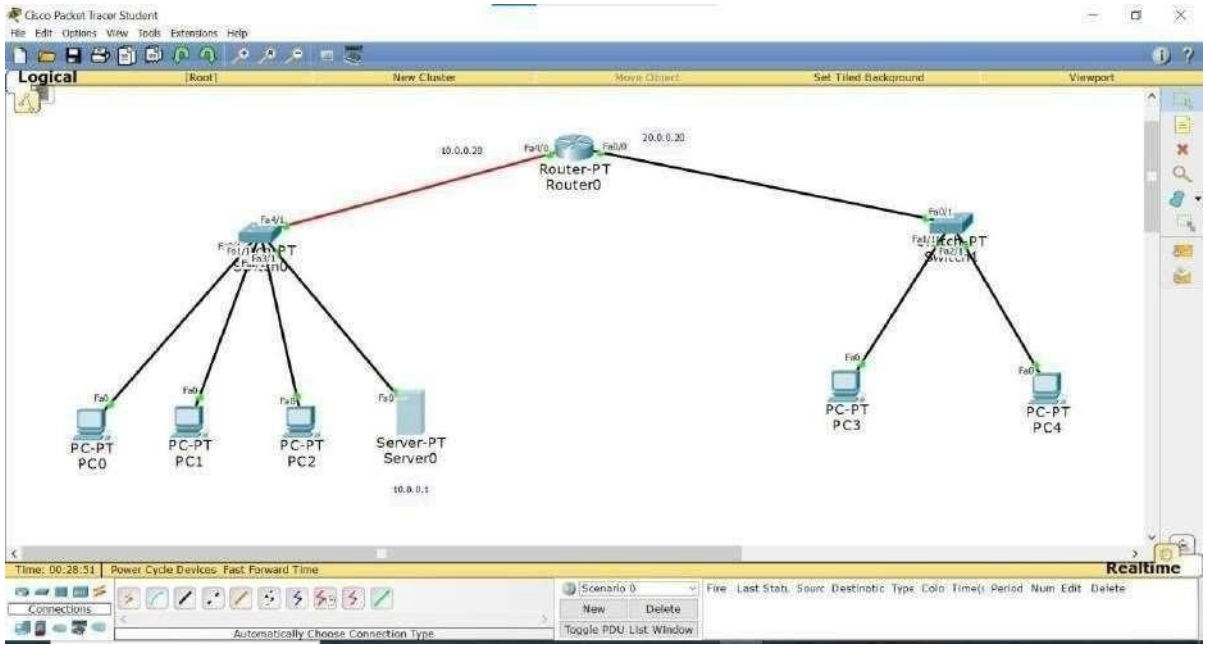
Step 5: Then configure the pc's
→ Select a pc then desktop → go to IP configuration
select DHCP
→ Repeat the same procedure for all other pc's

Observation:
→ DHCP is used to design IP address dynamically to different devices.
→ To assign continous IP address we create a server pool where we assign the starting IP address and a default gateway number.
for pc's under different switches we create a different server pool again & start

PROGRAM 4.1:

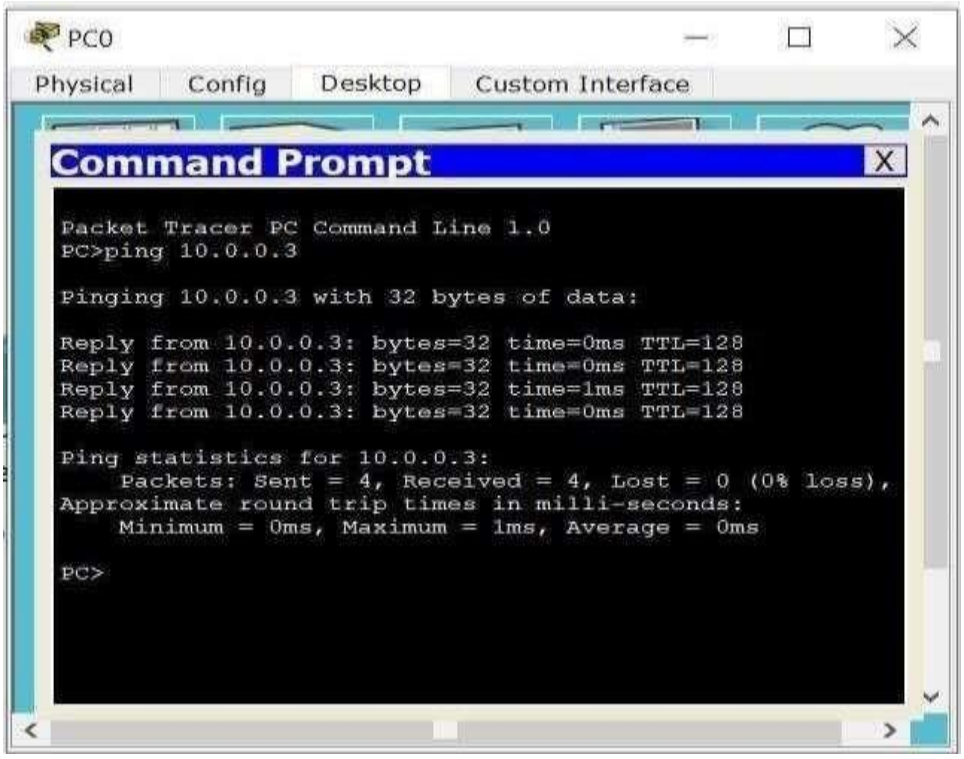


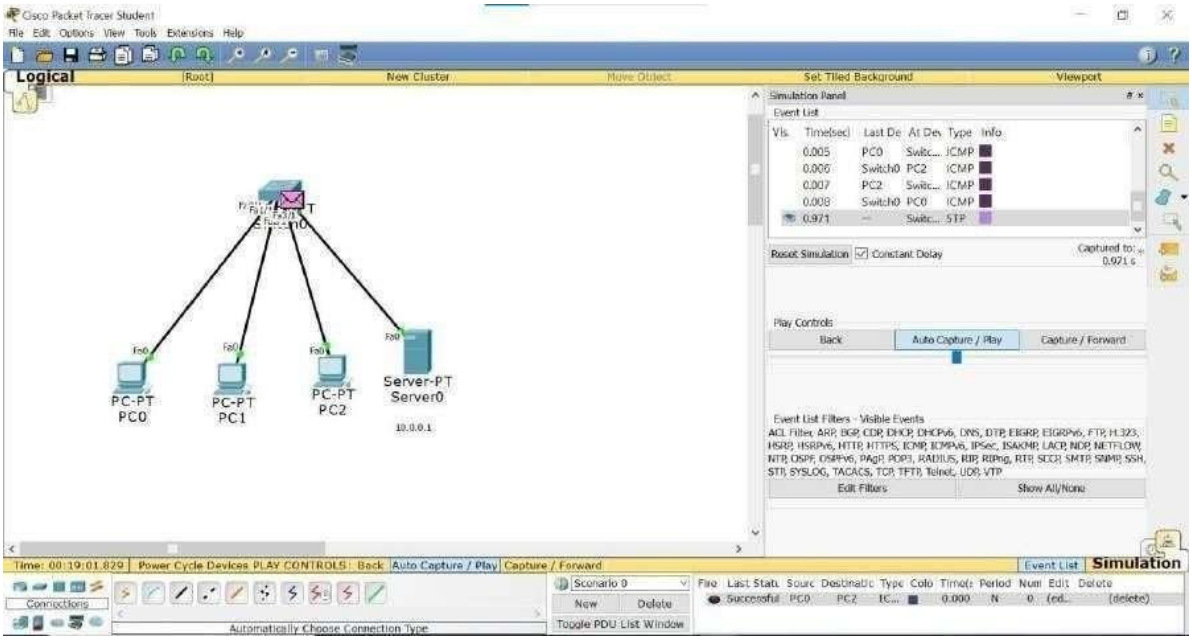
PROGRAM 4.2:



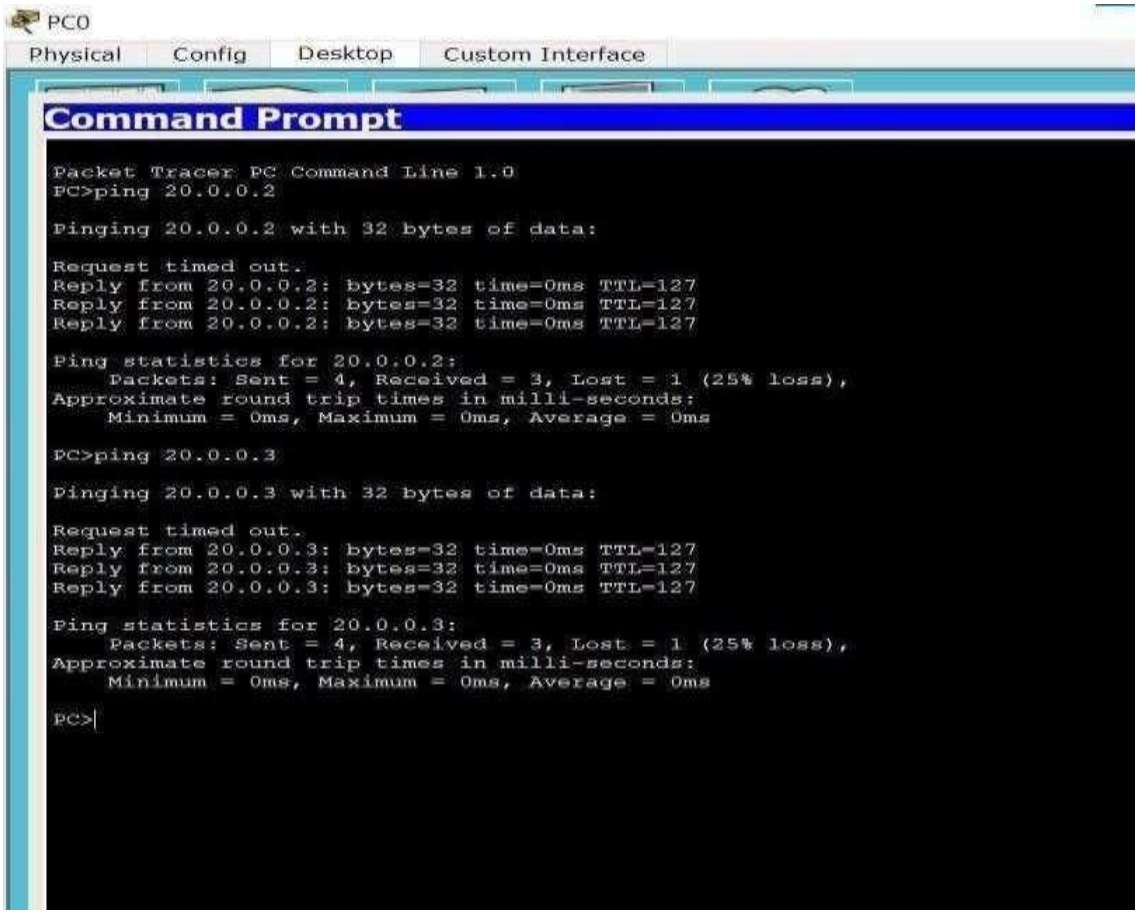
OUTPUT:

PROGRAM 4.1:





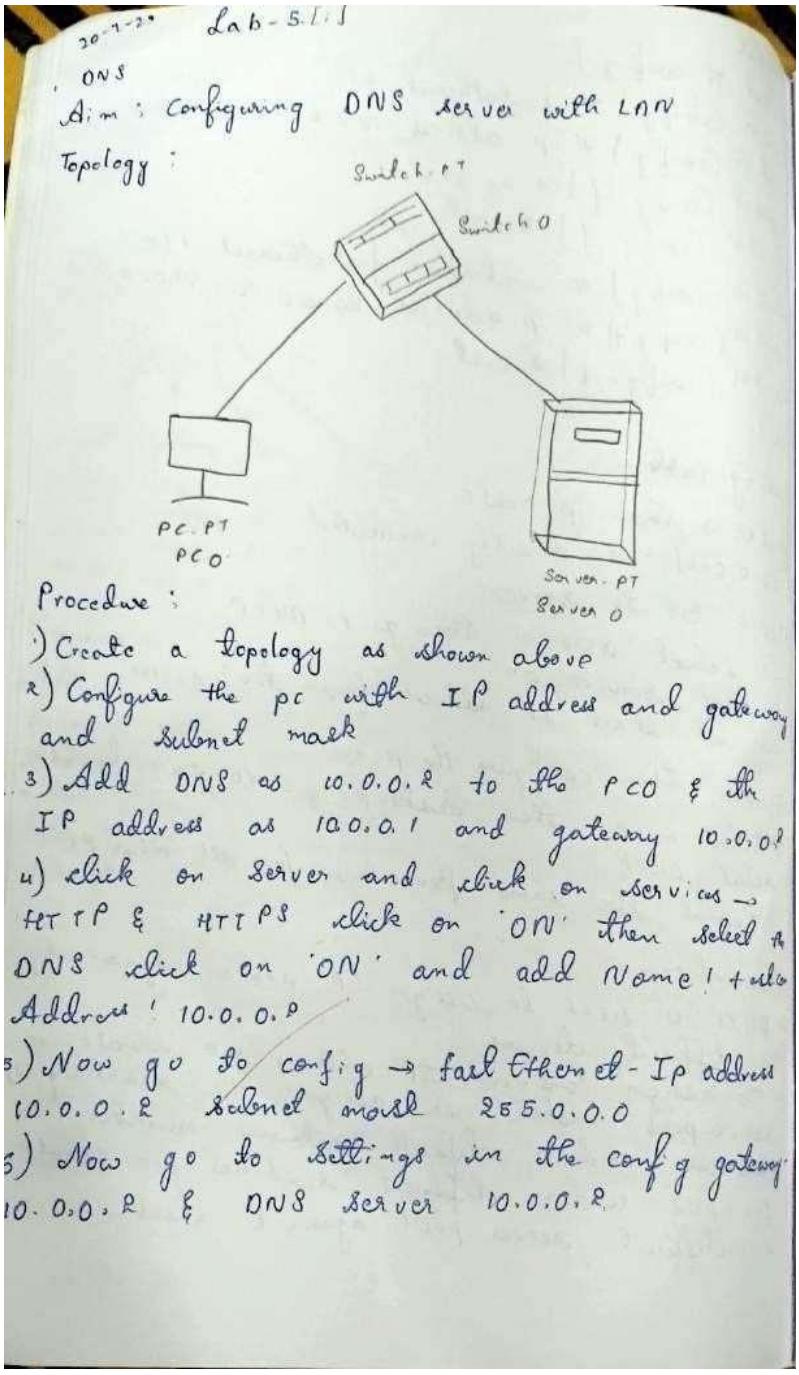
PROGRAM 4.2:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

LAB 5

Configure Web Server, DNS within a LAN.

OBSERVATION:



7) Now go to PC click on desktop → web browser and type the 'test.com' in URL and click on go

8) You will see the html index that has written the HTTP ballon.

Observation:

DNS is the Domain Name System. DNS is linked to the internet and focused on operating system using Internet protocol. DNS servers are required for working of DNS. The IP address is calculated with the aid of a lookup table.

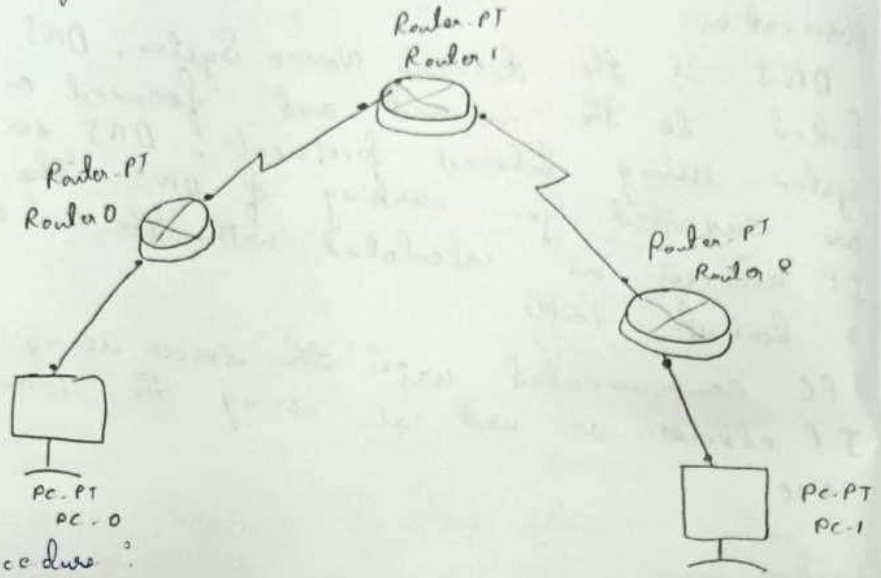
→ PC communicated with the server using the IP address as well as using the domain name.

Lab 5[ii]

RIP Routing Protocol

Aim: Configuring RIP Routing Protocol in Routers

Topology:



Procedure:

1) Three routers and 2 PC's are connected as show in topology

2) Configure the PC's with propor IP address and gateway address

3) Similarly, configure the Router's with the propo IP address in CLI mode.

→ N. Enable
→ Config T
→ Interface fastethernet 0/0
→ IP address 10.0.0.1 255.0.0
→ encapsulation ppp
→ clockrate 64000
no shut

Note: The encapsulation ppp should be done in all the routers and clock rate 64000 command should be only given to the clock symbols sides of the router

→ for making the routers to know about the other devices, in the previous 2 experiments we used static and the other with dynamic address but here we use a routing protocol algorithm that itself makes the router to know other devices

→ router rip
→ network 20.0.0.0 } router 2
→ network 30.0.0.0

→ router rip
→ network 30.0.0.0 } router 3
→ network 40.0.0.0

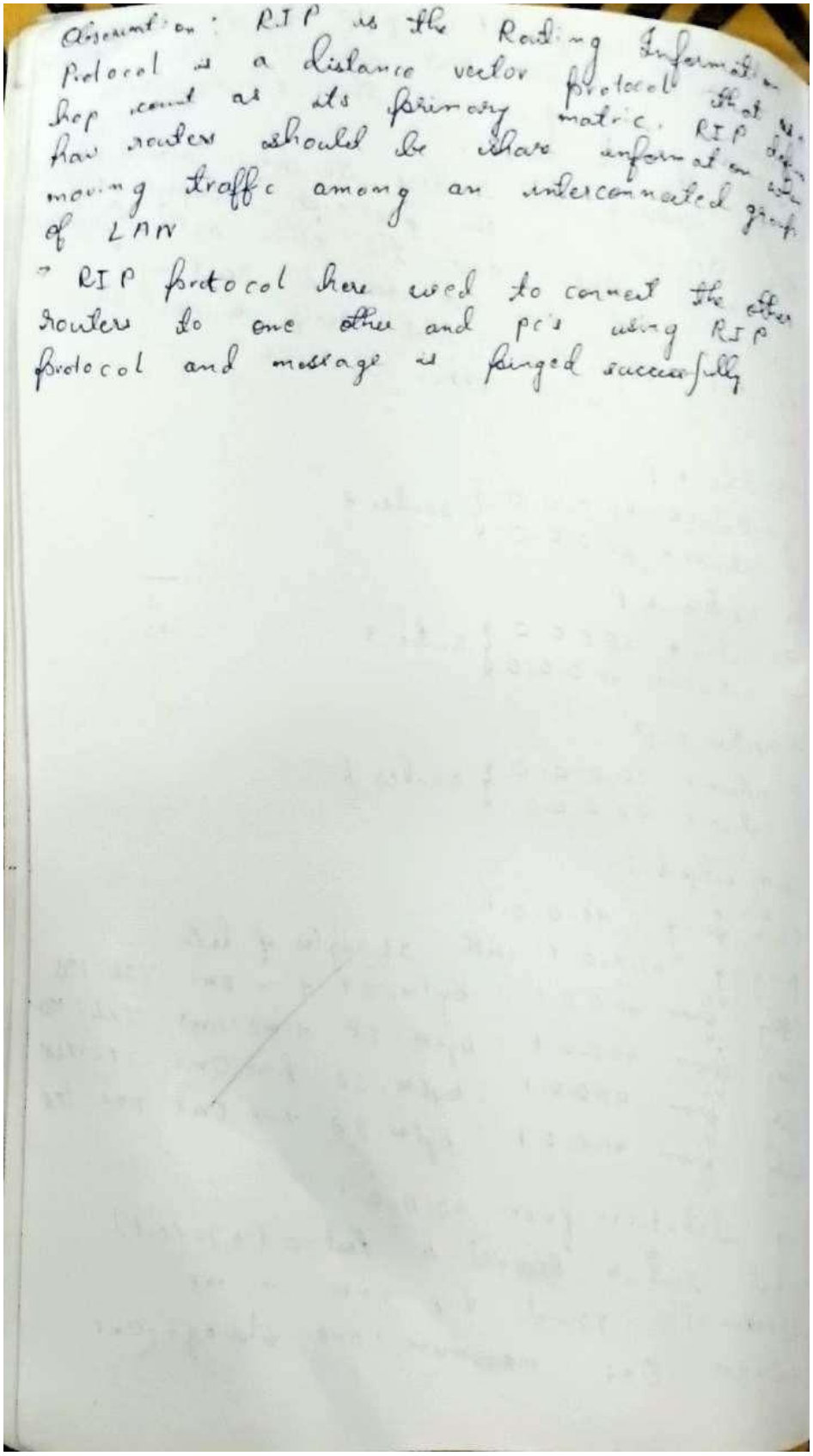
→ router rip
→ network 10.0.0.0 } router 1
→ network 20.0.0.0

Ping output:

PC> ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1 : bytes=32 time:0ms TTL:128
Reply from 40.0.0.1 : bytes=32 time:0ms TTL:128
Reply from 40.0.0.1 : bytes=32 time:0ms TTL:128
Reply from 40.0.0.1 : bytes=32 time:0ms TTL:128

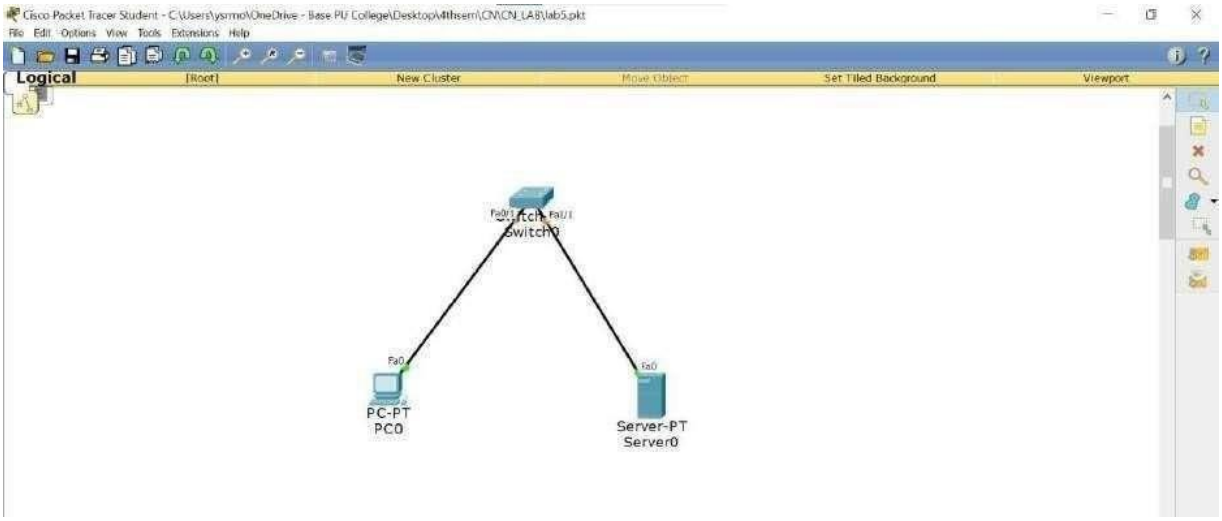
ping statistics from 40.0.0.1
packets sent 4 Received 4 Lost=0 (0% Lost)
Approximate round trip times in ms
minimum: 0ms, maximum: 0ms Average: 0ms



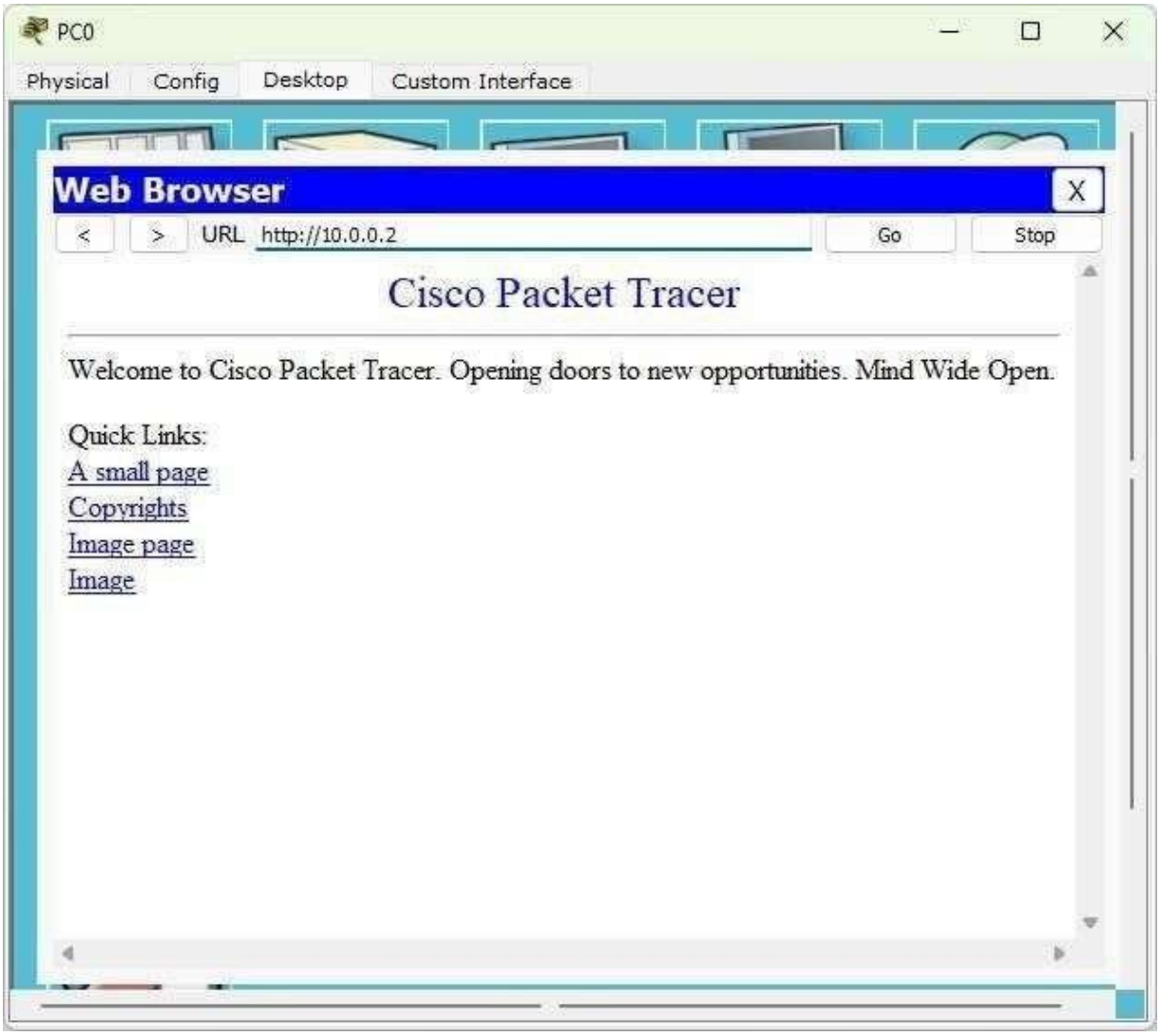
Observation: RIP is the Routing Information Protocol is a distance vector protocol that uses hop count as its primary matric. RIP determines how routers should share information when moving traffic among an interconnected group of LANs

- RIP protocol here used to connect the other routers to one other and pcs using RIP protocol and message is pinged successfully

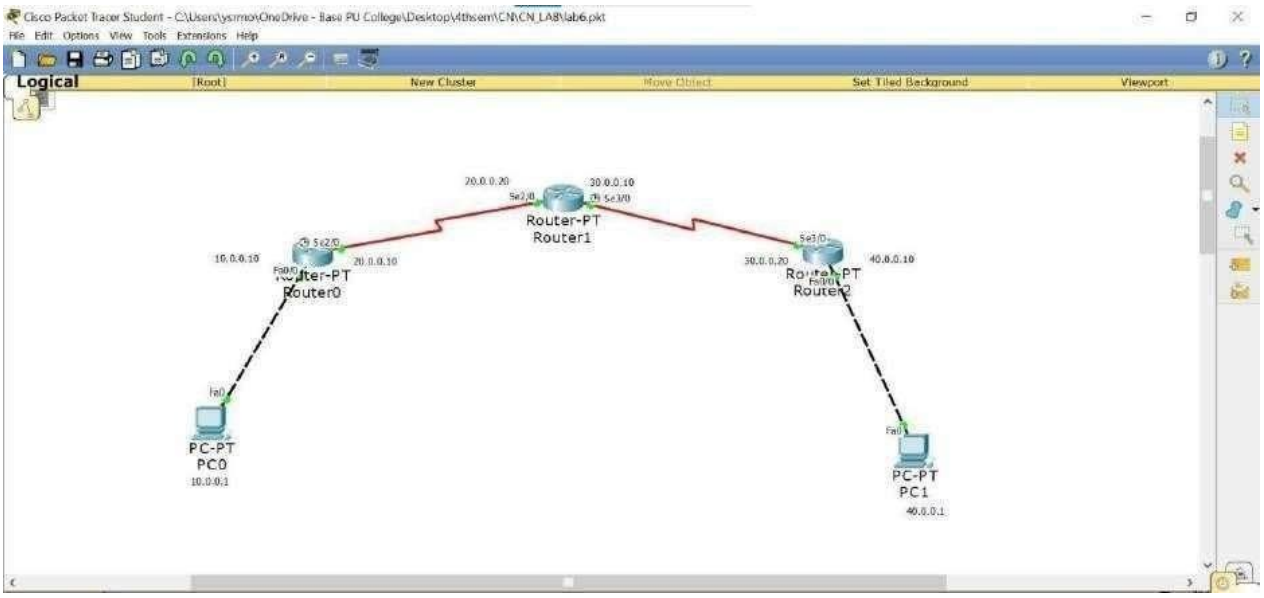
TOPOLOGY:


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Switch0
Fa0/1
Fa0/1
Fa0
Fa0
PC-PT
PC0
Server-PT
Server0

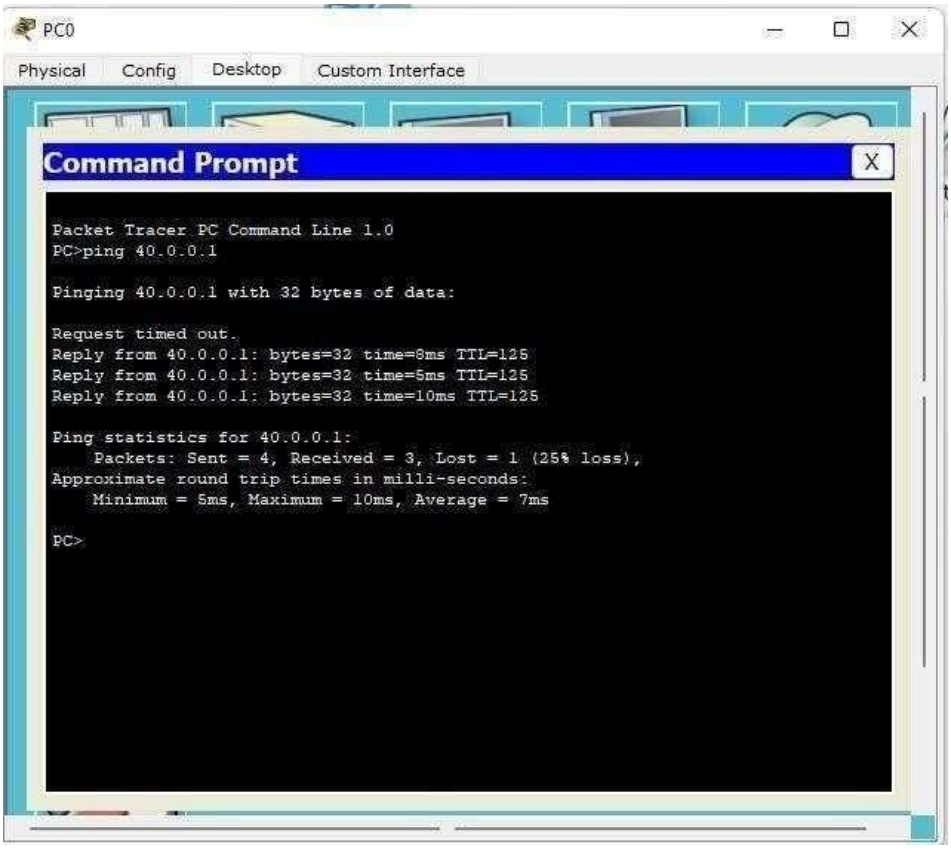
OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL
http://10.0.0.2
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image

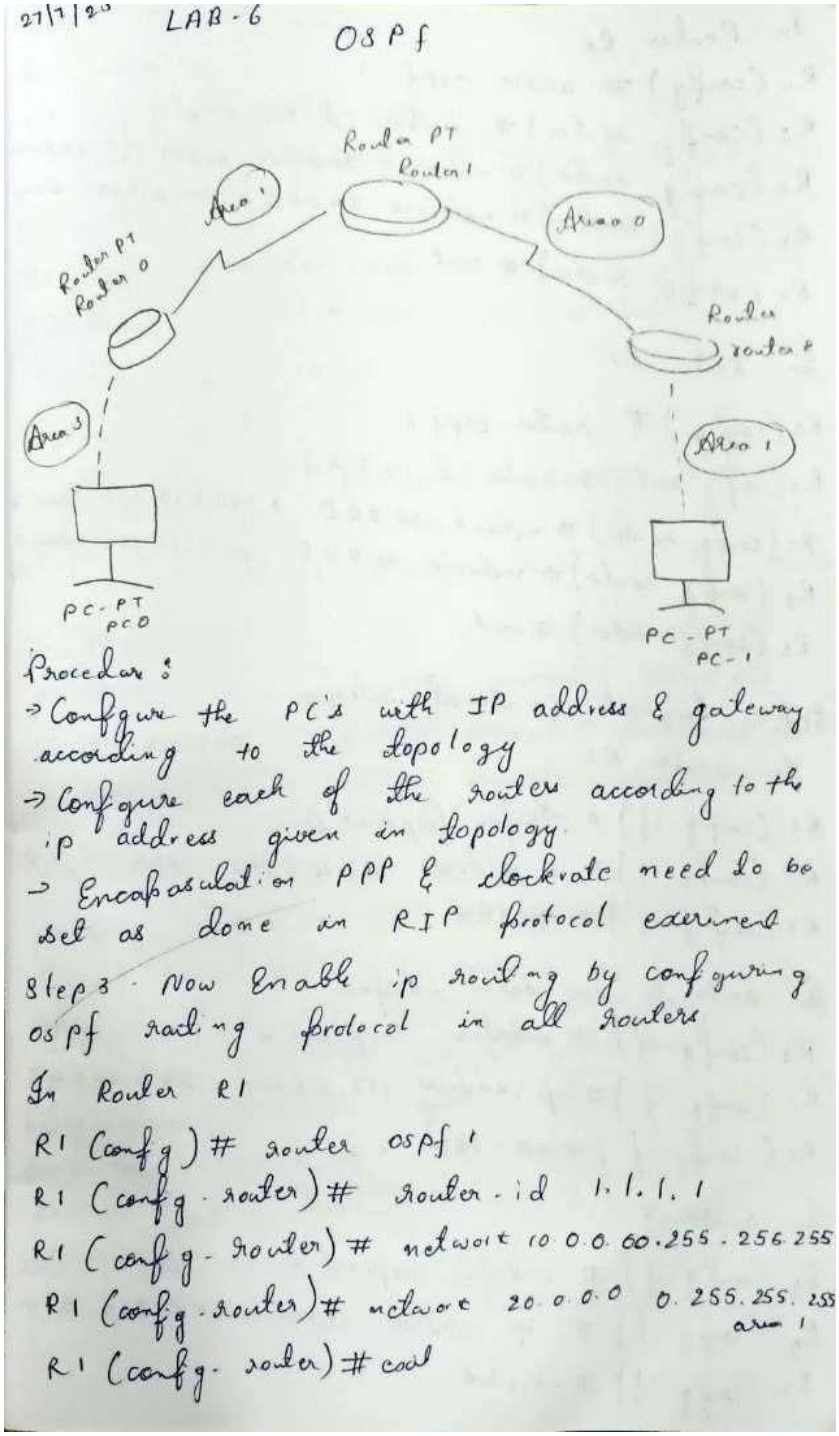
TOPOLOGY: RIP Protocol



OUTPUT:



Configure OSPF routing protocol. Lab-6



In Router R2.

R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 200.0.0.0 0.255.255.25
R2(config-router)# network 3000.0.0 255.255.255.
R2(config-router)# exit

In Router R3

R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 30.0.0.0 0.255.255.255 a
R3(config-router)# network 40.0.0.0 0.255.255.255 a
R3(config-router)# exit

Step 4: loopback in serial interface

In router R1

R1(config-if)# interface loopback 0
R1(config-if)# ip address 172.16.1.252 255.255.0
R1(config-if)# no shut

In router 2 in serial interface

R2(config-if)# interface loopback 0
R2(config-if)# ip address 172.16.1.253 255.255.0
R2(config-if)# no shut

In router 3

R3(config-if)# interface loopback 0
R3(config-if)# ip address 172.16.1.254 255.255.
R3(config-if)# no shut

Step 5 - Virtual link.

In router R1

R1 (config) # router ospf 1

R1 (config-router) # area 1 virtual link 2.2.2.2

R1 (config-router) # end

In router R2

R2 (config) # router ospf 1

R2 (config-router) # area 1 virtual-link 1.1.1.1

R2 (config-router) # end

-> show ip route

O IA 10.0.0.0/8 [110/129] via 30.0.0.1 serial 3/0

O IA 20.0.0.0/8 [110/128] via 30.0.0.1 serial 3/0

30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 30.0.0.0/8 is directly connected, serial 3/0

C 30.0.0.1/32 is directly connected, serial 3/0

C 40.0.0.0/8 is directly connected fastethernet 0/0

C 172.16.0.0/16 is directly connected, loopback

Ping output

Pinging 40.0.0.10 with 32 bytes of data

Request timed out

Reply from 40.0.0.10 bytes=32 time=2ms TTL=125

Reply from 40.0.0.10 bytes=32 time=9ms TTL=125

Reply from 40.0.0.10 bytes=32 time=10ms TTL=125

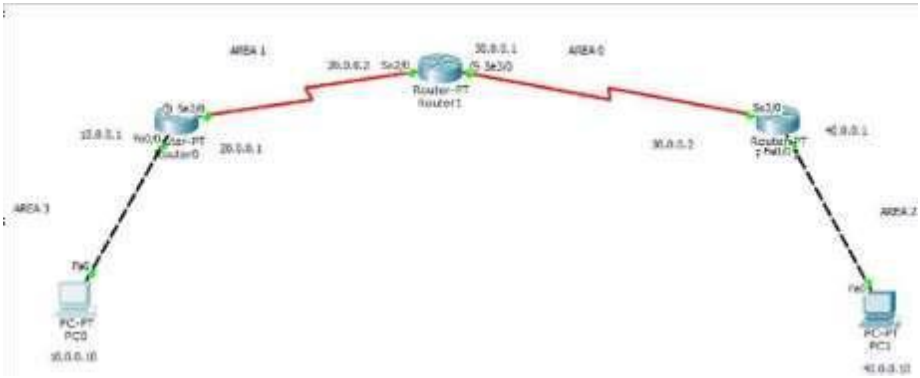
Pinging statistics for 40.0.0.10

Packets: sent=4, received=3, lost=1

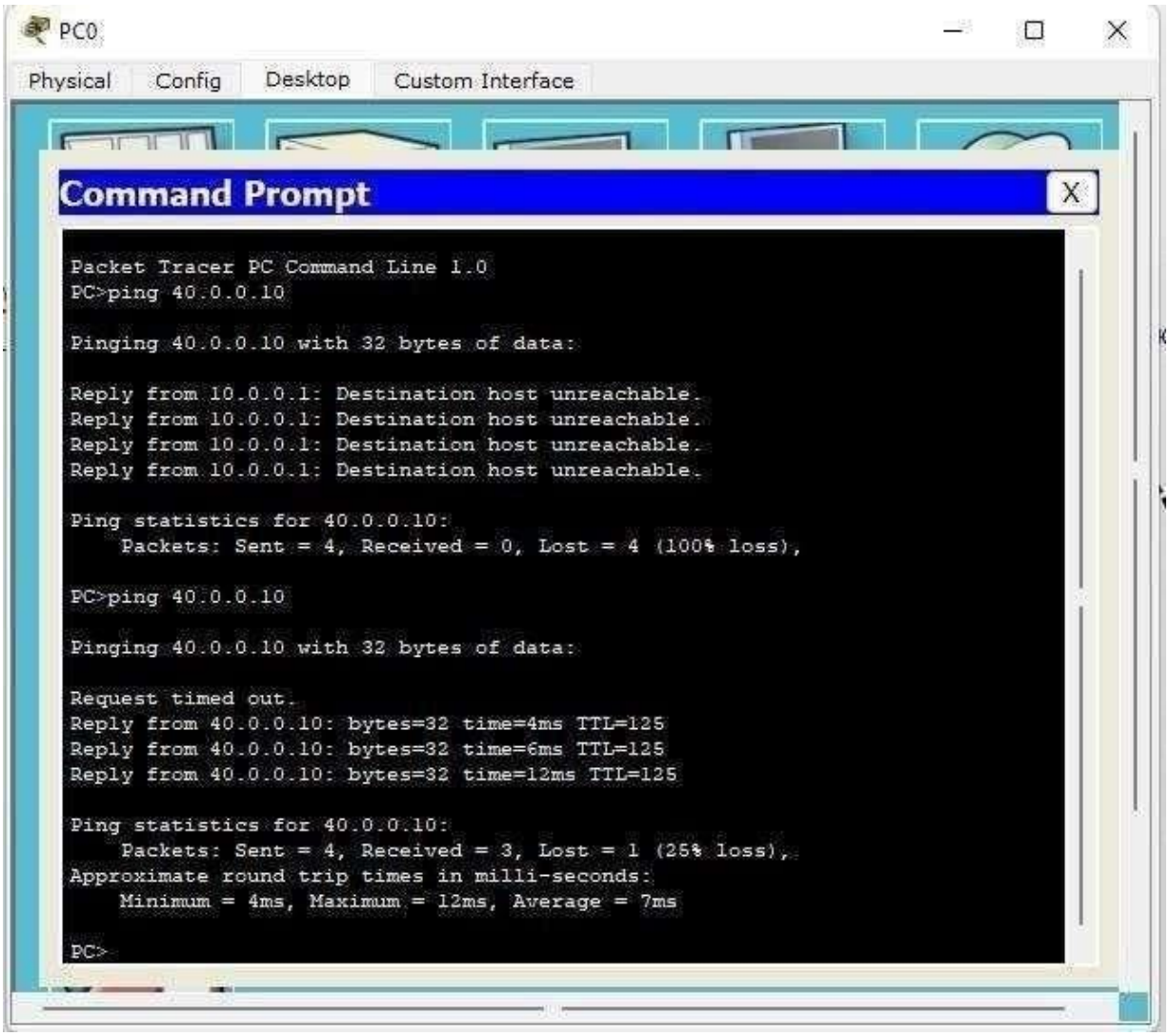
Approx round trip in ms:

min=2ms, max=10ms, Average=7ms

TOPOLOGY:

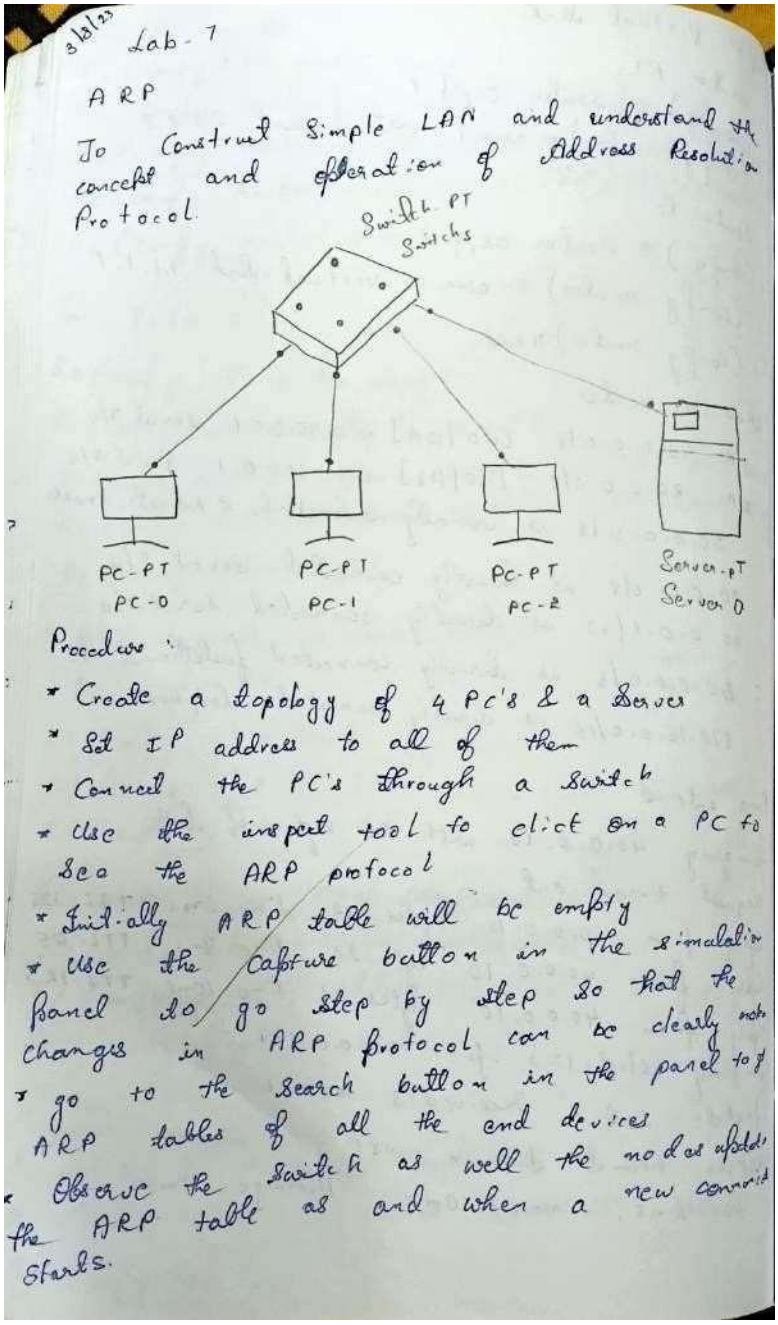


OUTPUT:



LAB 7

To construct a simple LAN and understand the concept and operationof Address Resolution Protocol (ARP).



Ping output

PC > arp -a

PC > ping 10.0.0.2

PC > Ping 10.0.0.03

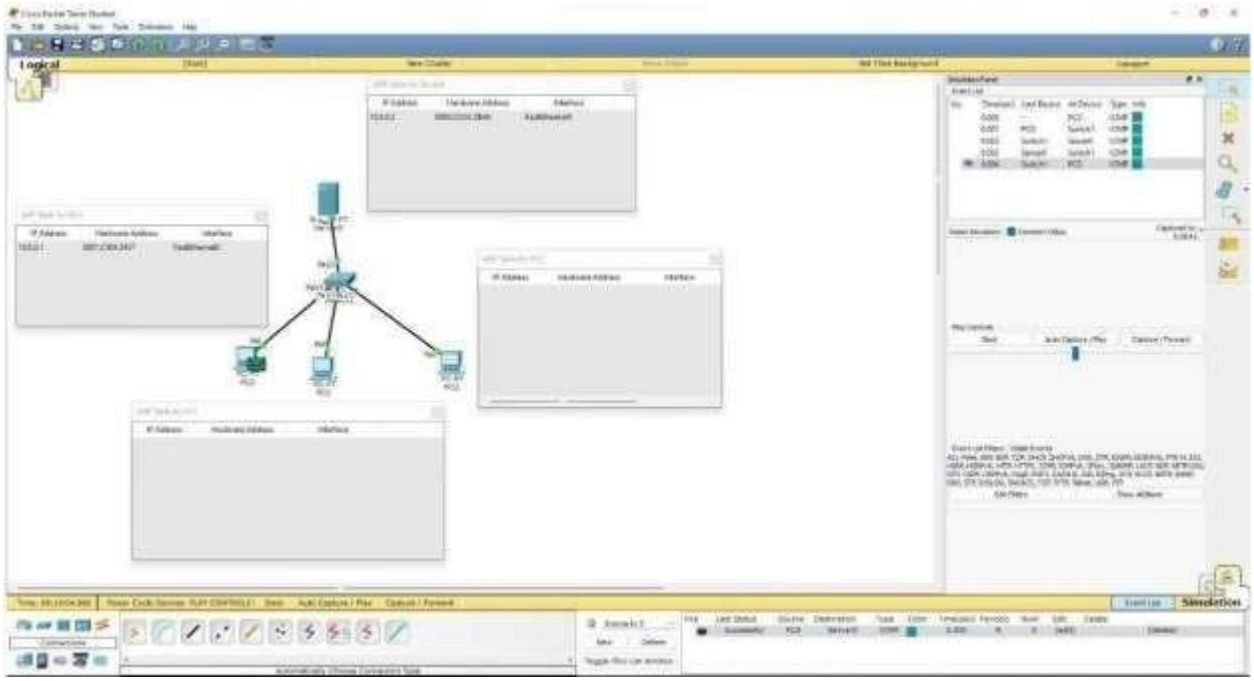
PC > ping 10.0.04.

Internet address	Physical address	Type
10.0.0.2	00cd.bd15.3a18	dynamic
10.0.0.3	0000.0657.b7e8	dynamic
10.0.0.4	00d0.bc57.aa38	dynamic

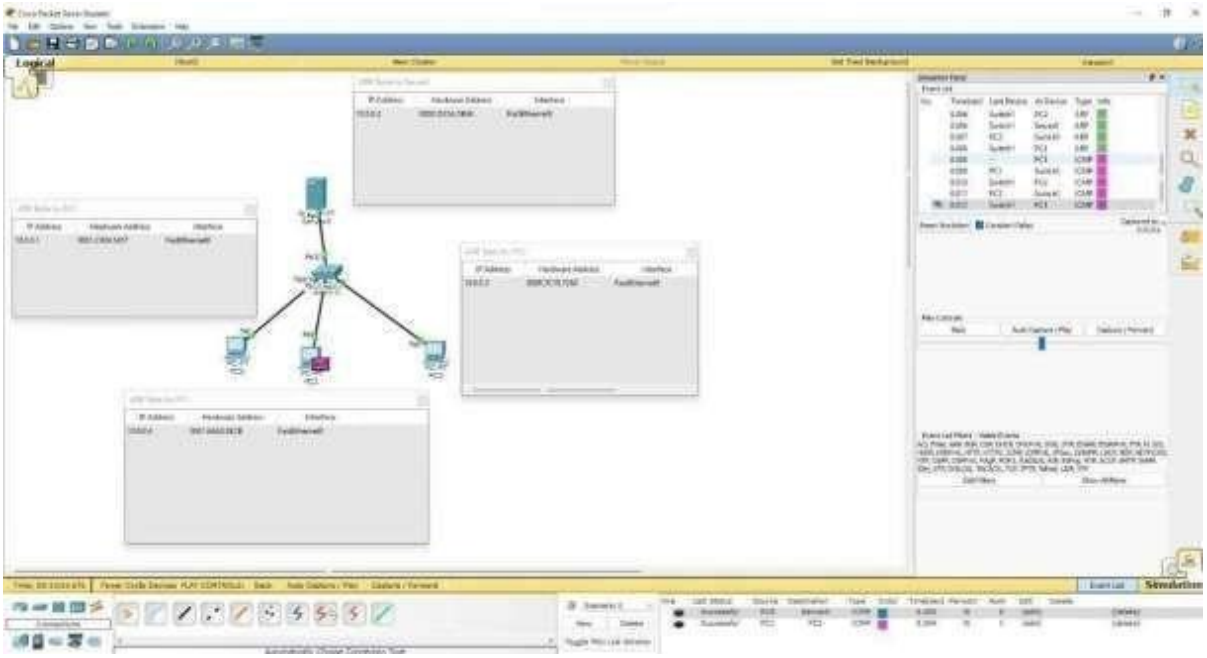
Observation

We observe the changes when we capture every time when we do in all the PC's ARP tables will have values empty at the starting & values will be formed whenever we ping each PC in the ping

TOPOLOGY:



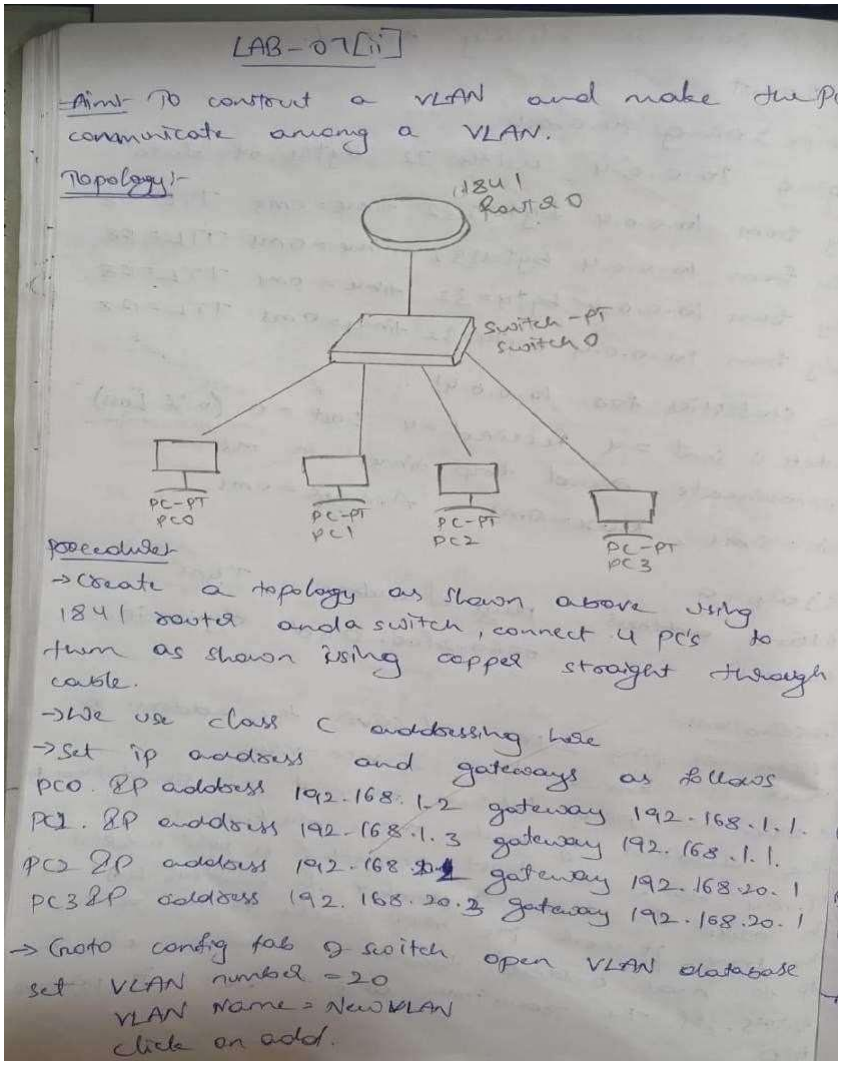
OUTPUT:



LAB 8

To construct a VLAN and make a pc communicate among VLAN

Observation:



→ In switch go to fastethernet 5/0 and connect it to router and configure it.
select Trunk and choose 20, New VLAN.
→ For f0/3 and f0/4 select 10 : NewLAN and keep access as it is.
→ open config table in router, goto VLAN database Add VLAN no. 10
→ In router, go to CLI mode.

```
fa 0/0
Router(config)# ip address 192.168.1.1 255.255.255.0
Router(config)# no shut.

Router(config)# interface fastethernet 0/0.1
Router(config-subif)# encapsulation dot1q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
Router(config-subif)# no shut
Router(config)# exit
```

Ping output:-

```
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=0ms TTL=127
Reply from 192.168.20.2 bytes=32 time=2ms TTL=127
Reply from 192.168.20.2 bytes=32 time=1ms TTL=127
Ping statistics for 192.168.20.2
 packets sent=4 Recieved=3 lost=1 (25% loss)
Approximate round trip in ms:
 min=0ms Max=2ms Average=0ms.
```

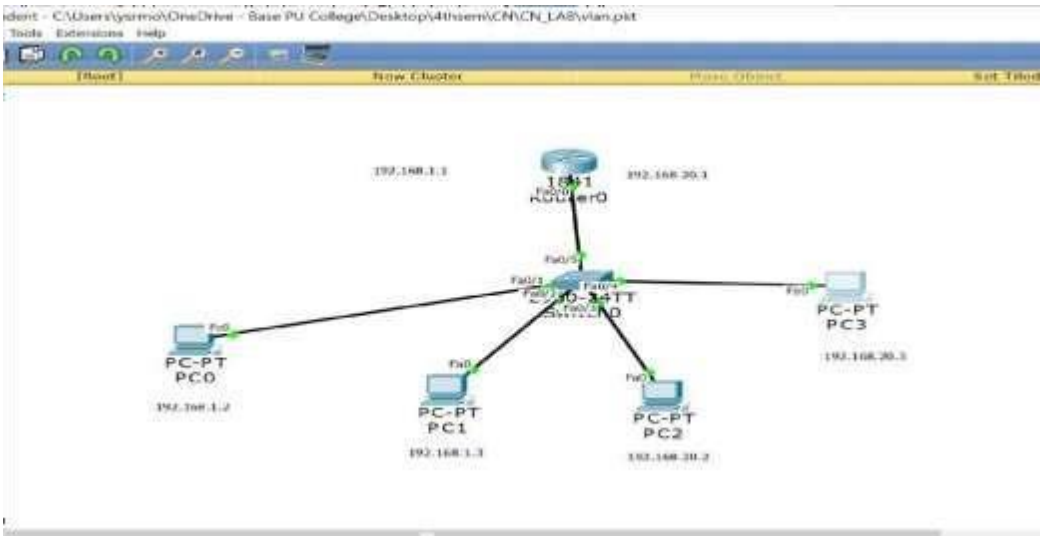
Observation:-

→ We can observe that after VLAN is configured we can successfully ping PC2 (192.168.20.2) from PC0 (192.168.1.2)

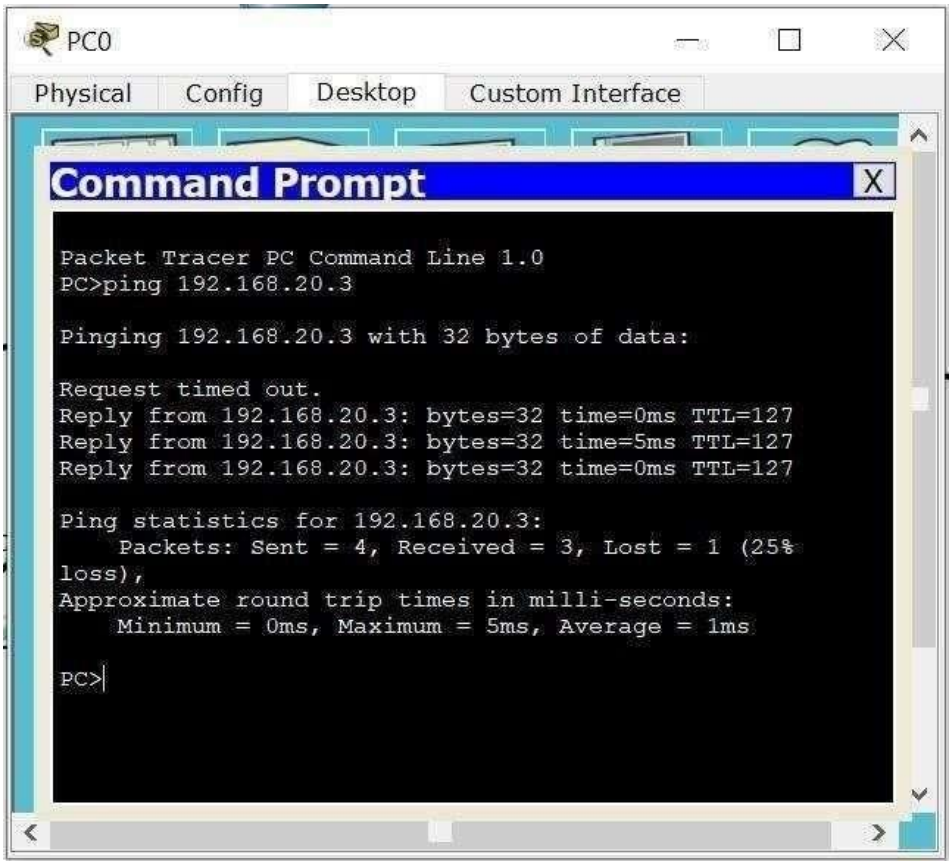
*PC2 and PC3 are grouped together and communication among them is done via VLAN.

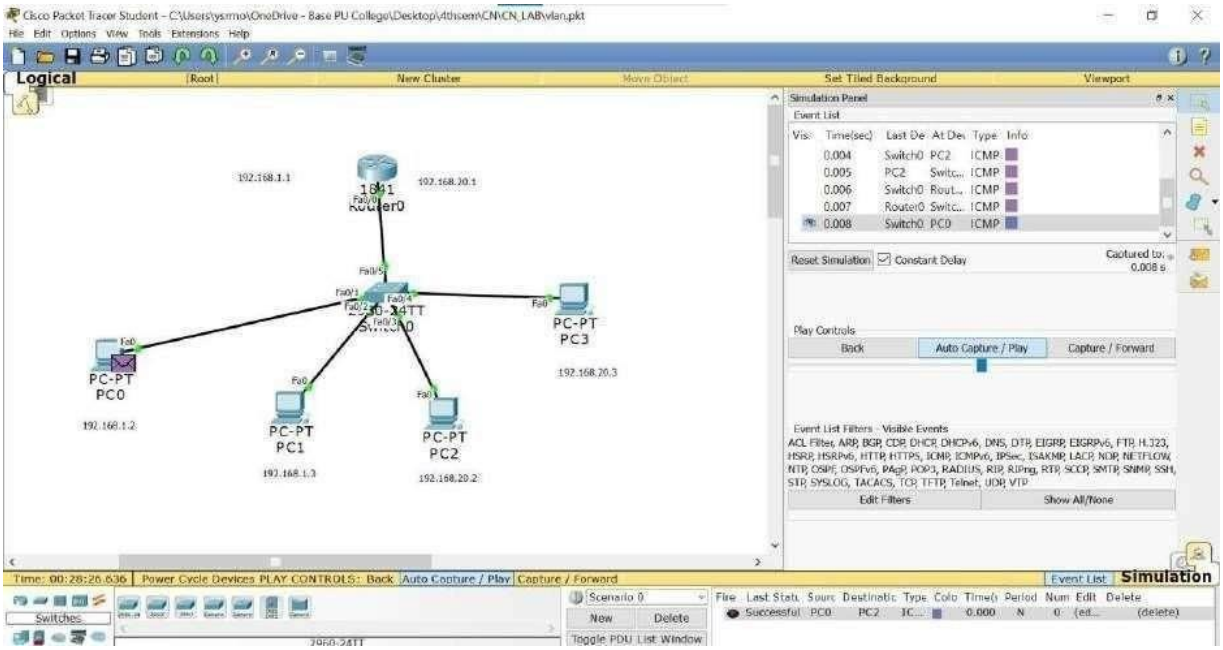
192.168.20.1 is a sub interface of f0/1 router

TOPOLOGY:



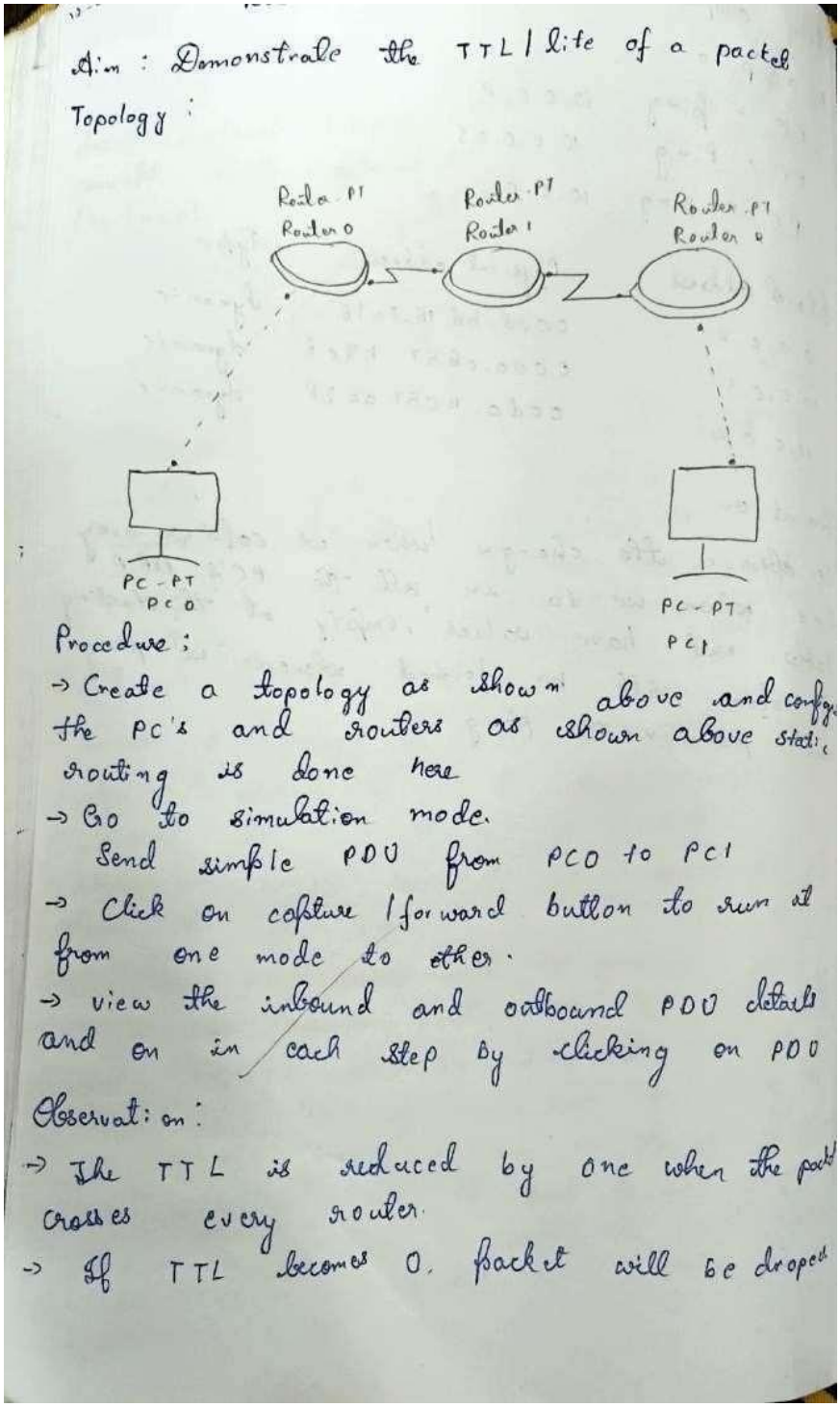
OUTPUT:




Cisco Packet Tracer Student – C:\Users\ys.rmo\OneDrive - Base PU College\Desktop\vtu\sem\CN\VCN_LAB\vlan.pkt
File Edit Options View Tools Extensions Help
Logical
Root
New Cluster
Move Object
Set Tiled Background
Viewport
192.168.1.1
192.168.20.1
1841
Router0
Fa0/5
Fa0/1
Fa0/4
Fa0/2
Fa0/3
Fa0
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.1.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3
Simulation Panel
Event List
Vis. Time(sec) Last De... At De... Type Info
0.004 Switch0 PC2 ICMP
0.005 PC2 Switc... ICMP
0.006 Switch0 Rout... ICMP
0.007 Router0 Switc... ICMP
0.008 Switch0 PC0 ICMP
Reset Simulation
Constant Delay
Captured to:
0.008 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 00:28:26.636
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Switches
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Stat... Sourc... Destinatio... Type Colo... Time(... Period Num Edit Delete
Successful PC0 PC2 IC... 0.000 N 0 (ed... (delete)

LAB 9

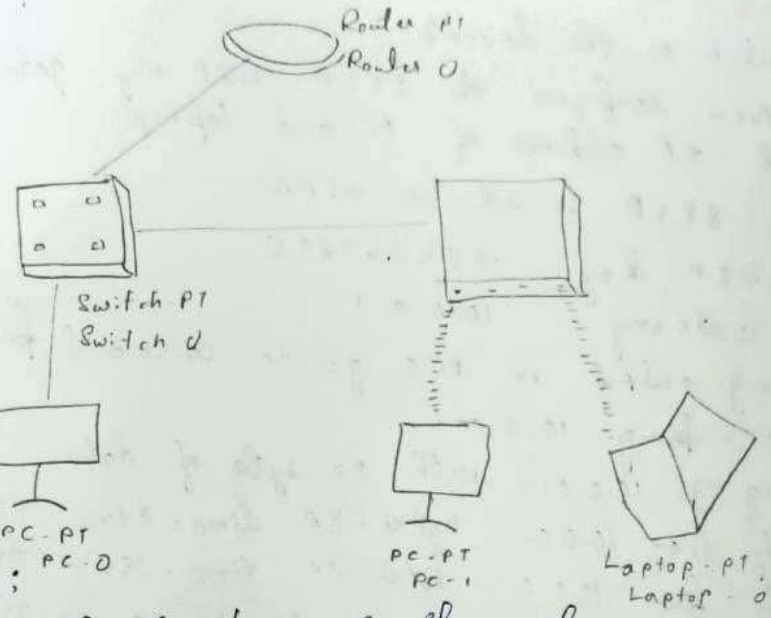
Demonstrate the TTL/ Life of a Packet



Lab - 08 [ii]

Aim : To construct a WLAN and make the nodes communicate wirelessly

Topology :



Procedure ;

→ Create the topology as shown above.

→ Configure the PC's as normally we do configure PC 3

IP address : 0.0.0.2

→ Configure router 0

Set the IP address as 10.0.0.1 for fa 0/0

→ Configure the access point - PT in config of port 1

→ SSID - WLAN

Select WEP and give any 10 digit number as password 1234567890

→ To configure pc 4 and laptop with wireless standard

Switch off the device

- Drag the existing PT-MOST-NM-1AM to the place it on its mentioned name
- Drag the WMP300N wireless interface to empty port
Switch on the device
- Now configure the SSID, WEP key, gateway and IP address of PC and laptop

The SSID is set to WLAN.

WEP key : 1234567890

Gateway : 10.0.0.1

ping output in PC0 go to command prompt

PC > ping 10.0.0.4

pinging 10.0.0.4 with 32 bytes of data

Reply from 10.0.0.4 bytes: 32 time = 24ms TTL = 1

Reply from 10.0.0.4 bytes: 32 time = 15ms TTL = 1

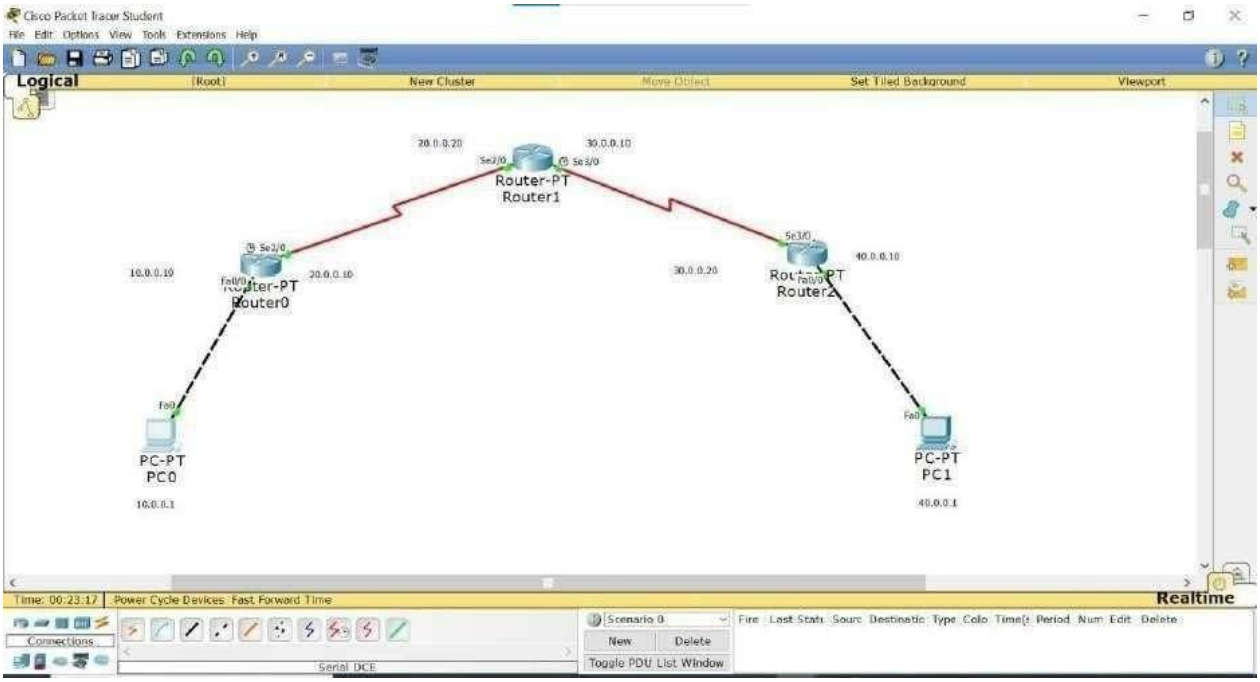
Reply from 10.0.0.4 bytes: 32 time = 5ms TTL = 1

Reply from 10.0.0.4 bytes: 32 time = 12ms TTL = 1

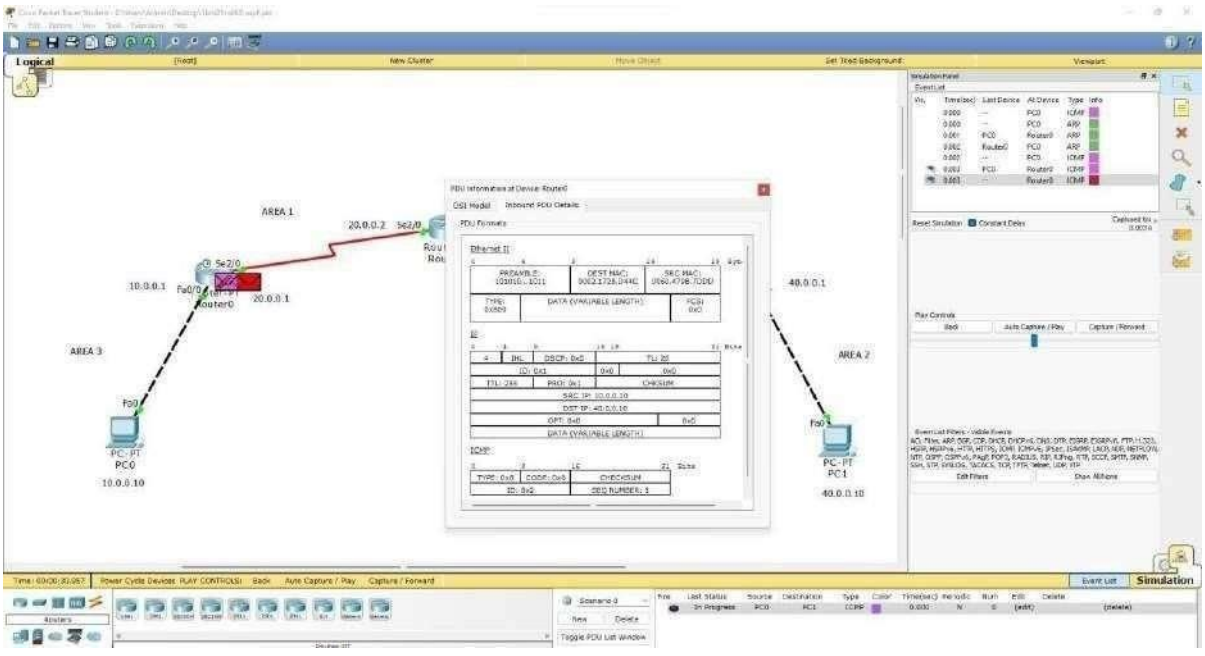
- Ping statistics for 10.0.0.4
packets sent = 4, Received = 4 Lost 0
- Approximate round trip time in ms
min = 5ms max = 24ms average = 15

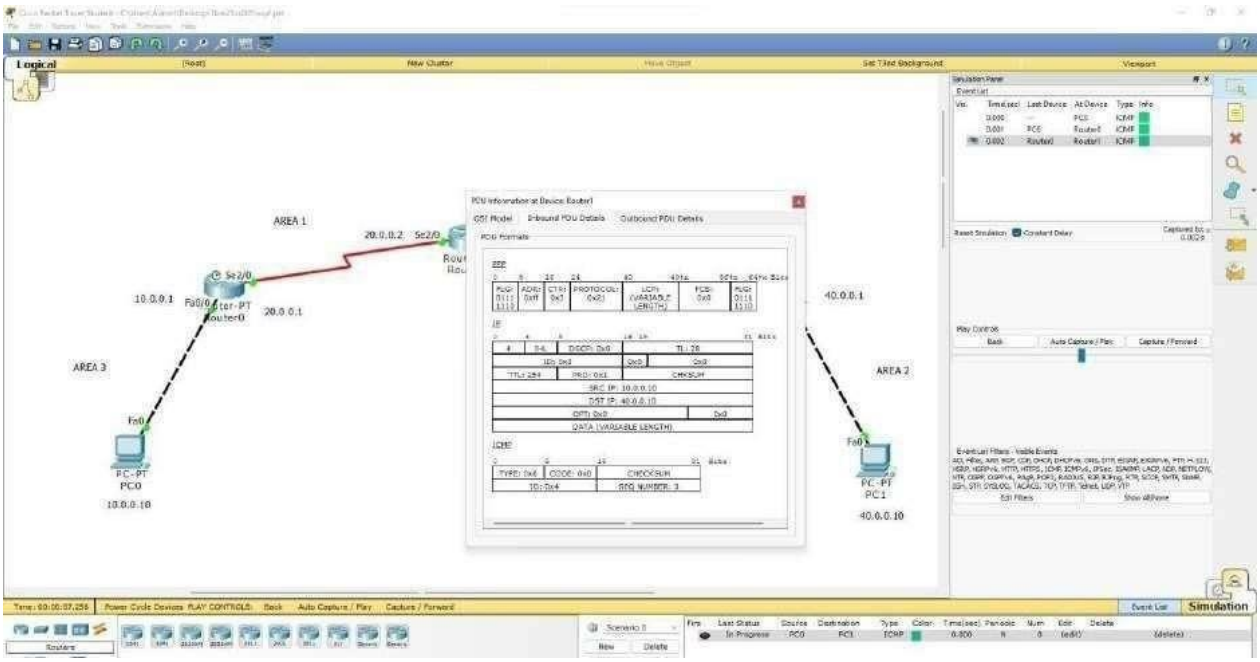
- Observation : We can ping each and every device to the other device. So we can observe the wireless connection is done successfully.
- When connection is established there is we can observe stripped lines connecting access points and end devices.

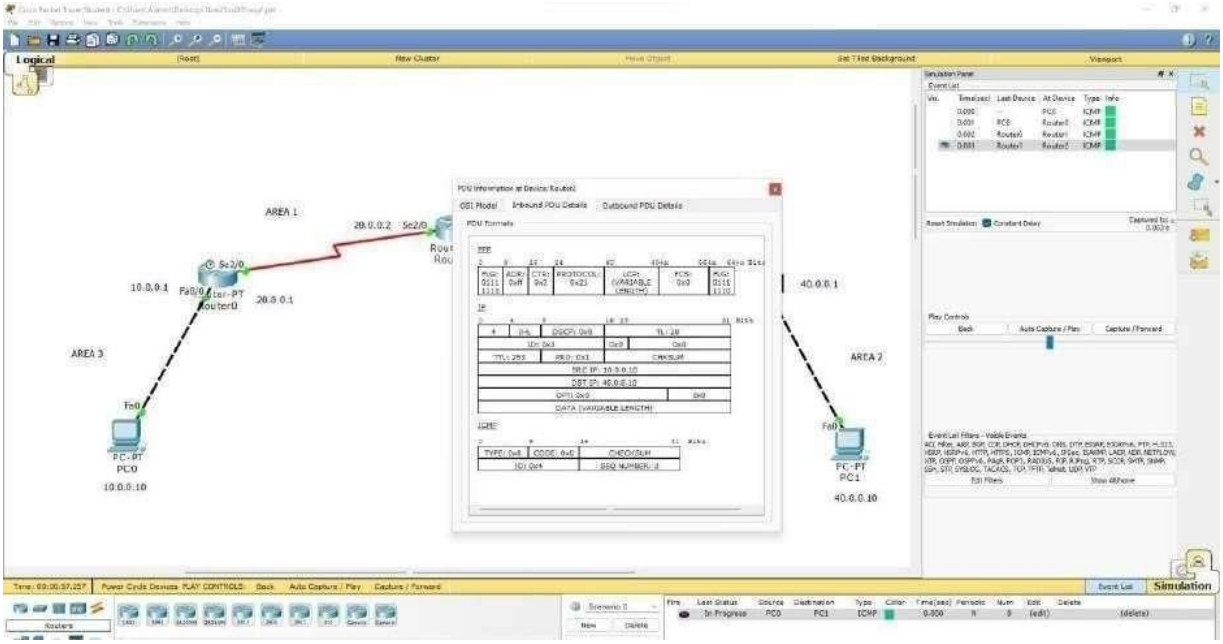
TOPOLOGY:

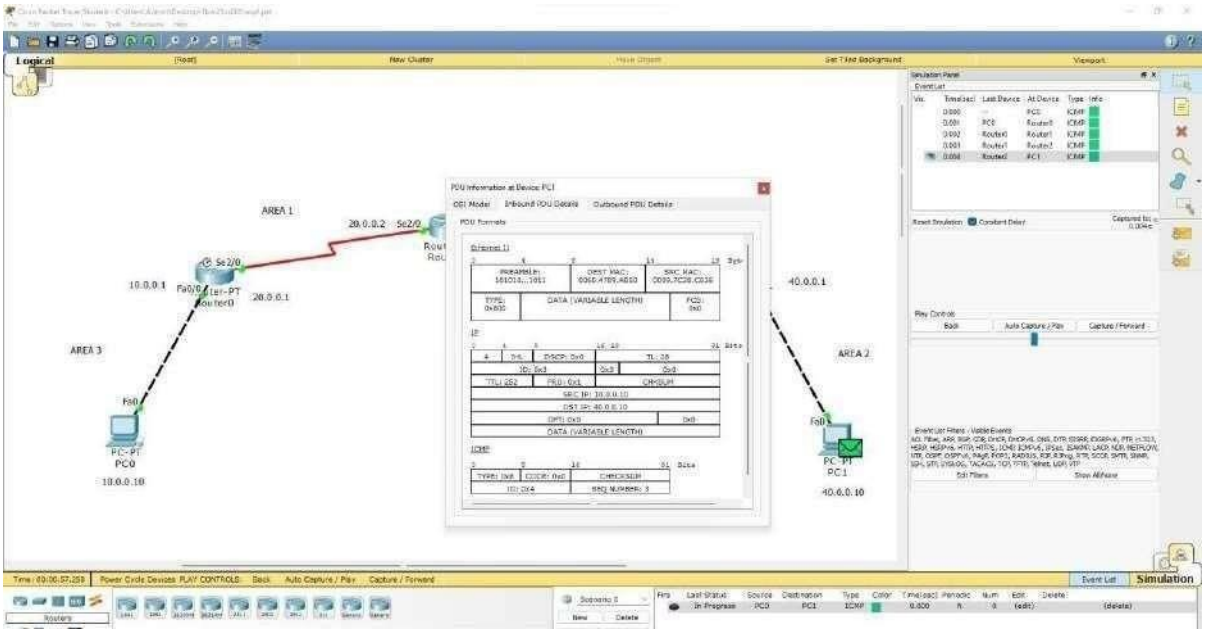


OUTPUT:

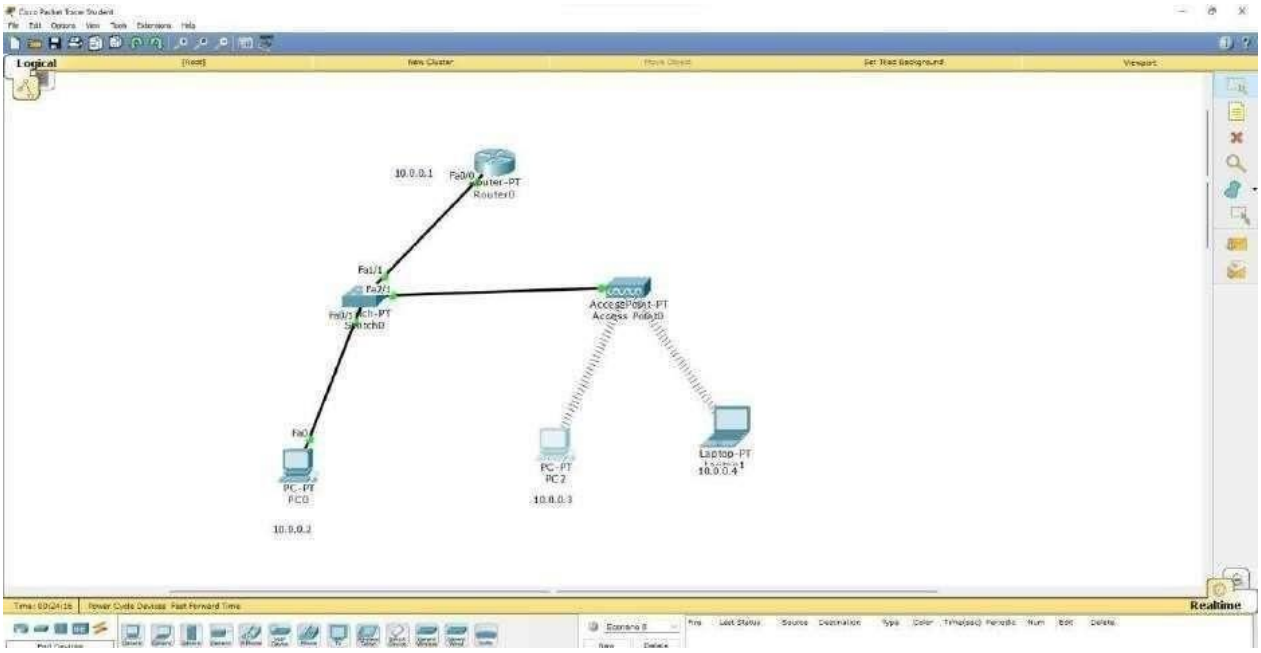


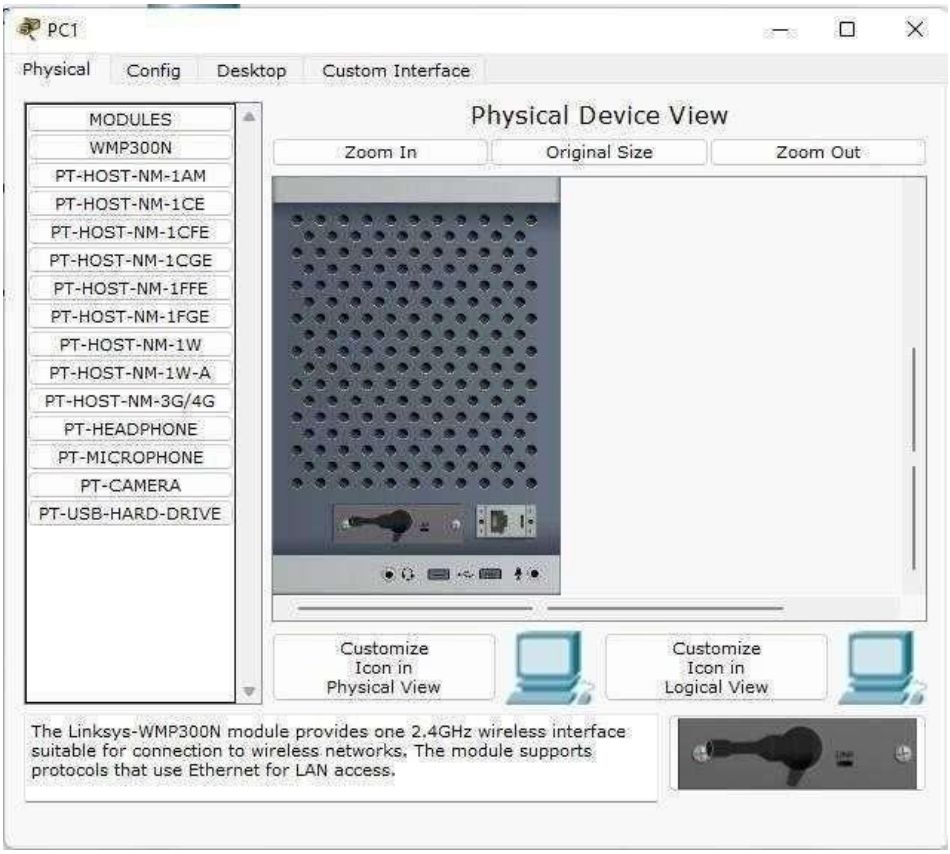

AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
AREA 3
AREA 2
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
DATA (VARIABLE LENGTH)
Logical
Simulation


AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
AREA 3
AREA 2
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Logical
Simulation


Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
AREA 3
Fa0
PC-PT
PC0
10.0.0.10
40.0.0.1
AREA 2
Fa0
PC-PT
PC1
40.0.0.10
Simulation

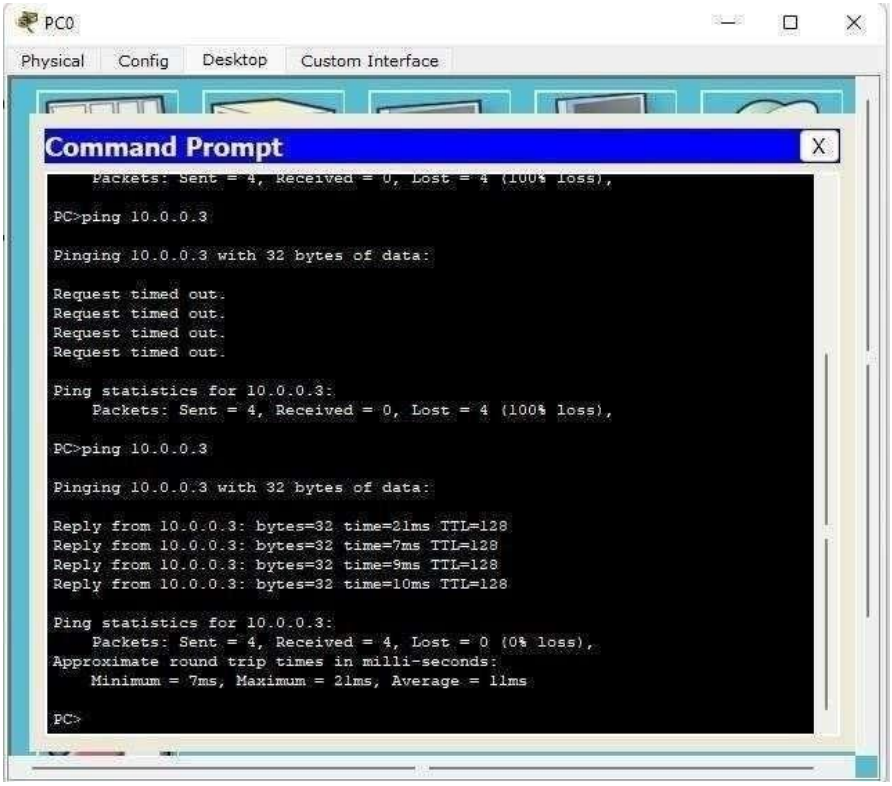
TOPOLOGY: WLAN






Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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=21ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=10ms 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 = 7ms, Maximum = 21ms, Average = 11ms
PC>

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

Lab 08 [::]

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

Topology :

Router PT
Router 0

PC-PT
PC0

Procedure :

→ Create the topology as shown above.
→ Connect the devices using copper cross-one
→ Configure the PC
IP address : 10.0.0.2
Gateway : 10.0.0.1
→ Go to CLI mode in Router 0

```
Router > en
Router # config t
Router (config) # hostname r1
r1 (config) # enable secret pass 1.
r1 (config) # interface fa 0/0
r1 (config) # ip address 10.0.0.1 255.0.0.0
r1 (config-if) # no shut
r1 (config-if) # line vty 0 5
r1 (config-if) # login
```

r1 (config-line)# password p0
r1 (config-line)# exit

#ping output in PC 0

We can successfully ping 10.0.0.1 from PC0

PC > telnet 10.0.0.1
Trying 10.0.0.1 open
User Access Verification
Password :

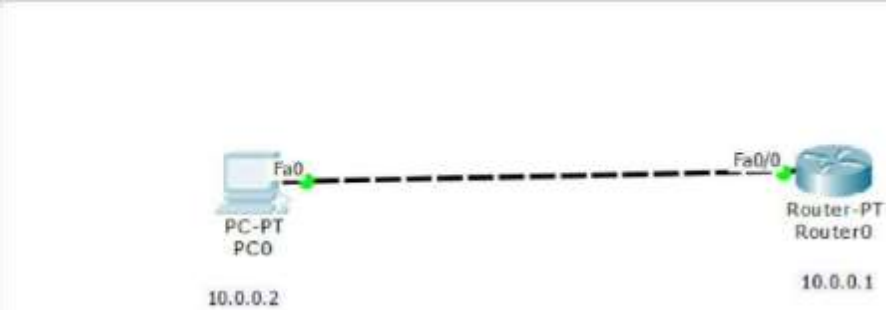
r1 > en
Password : p1
r1 # show ip route
C 10.0.0.0/8 is directly connected. Fa 0/0

Observations :

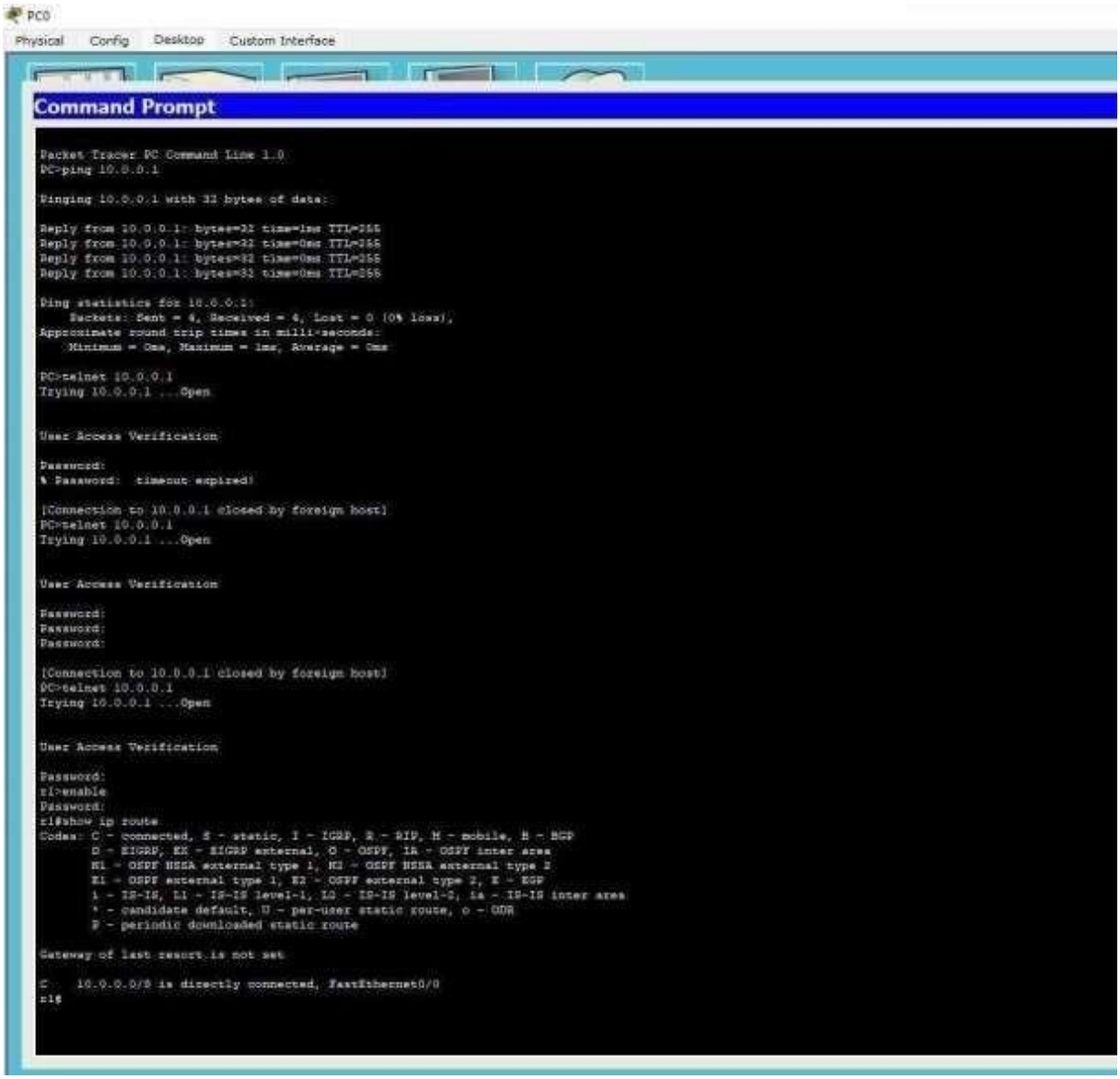
→ We can observe that the admin in PC is a run commands as run in router CLI and the result from the PC

→ So with the helps of TELNET, we can access the router in server room from a PC.

TOPOLOGY:



OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
% Password: timeout expired!
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
Password:
Password:
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
r1#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
r1#

LAB 13 Program 1

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

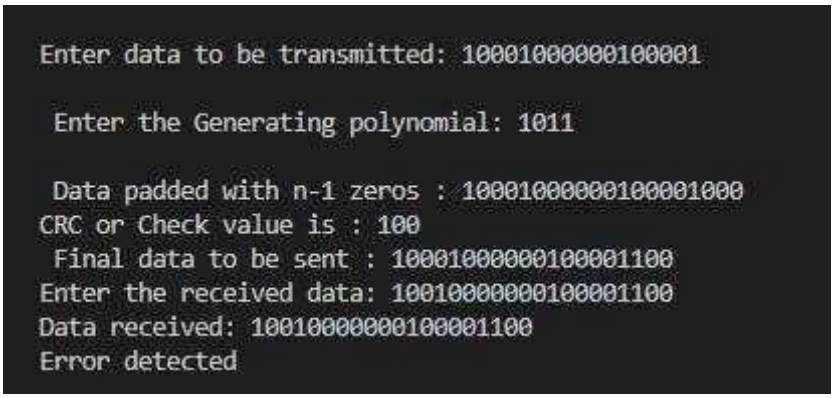
CODE:

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

#define N strlen(gen_poly) char data[28]; char check_value[28]; char
gen_poly[10]; int data_length,i,j;

void XOR(){

for(j = 1;j < N; j++) check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');}

void receiver(){
   printf("Enter the received data: "); scanf("%s",
   data); printf("Data received: %s", data); crc();
   for(i=0;(i<N-1) && (check_value[i]!='1');i++);
      if(i<N-1)
         printf("\nError detected\n\n");
      else
         printf("\nNo error detected\n\n");
}

void crc(){
   for(i=0;i<N;i++)
      check_value[i]=data[i];
   do{
      if(check_value[0]=='1')
         XOR();
      for(j=0;j<N-1;j++)
         check_value[j]=check_value[j+1];
      check_value[j]=data[i++];
```

```
    }while(i<=data_length+N-1);
}

int main()
{ printf("\nEnter data to be transmitted: ");
    scanf("%s",data); printf("\n Enter the
    Generating polynomial: ");
    scanf("%s",gen_poly);
    data_length=strlen(data);
    for(i=data_length;i<data_length+N-1;i++)
        data[i]='0';
    printf("\n Data padded with n-1 zeros : %s",data);
    crc();
    printf("\nCRC or Check value is : %s",check_value);
    for(i=data_length;i<data_length+N-1;i++)
    data[i]=check_value[i-data_length];
    printf("\n Final data to be sent : %s",data);
    receiver();
        return 0;
}
```

OUTPUT:

```
Enter data to be transmitted: 10001000000100001

 Enter the Generating polynomial: 1011

 Data padded with n-1 zeros : 10001000000100001000
CRC or Check value is : 100
 Final data to be sent : 10001000000100001100
Enter the received data: 10001000000100001100
Data received: 10001000000100001100
No error detected
```



Program 2

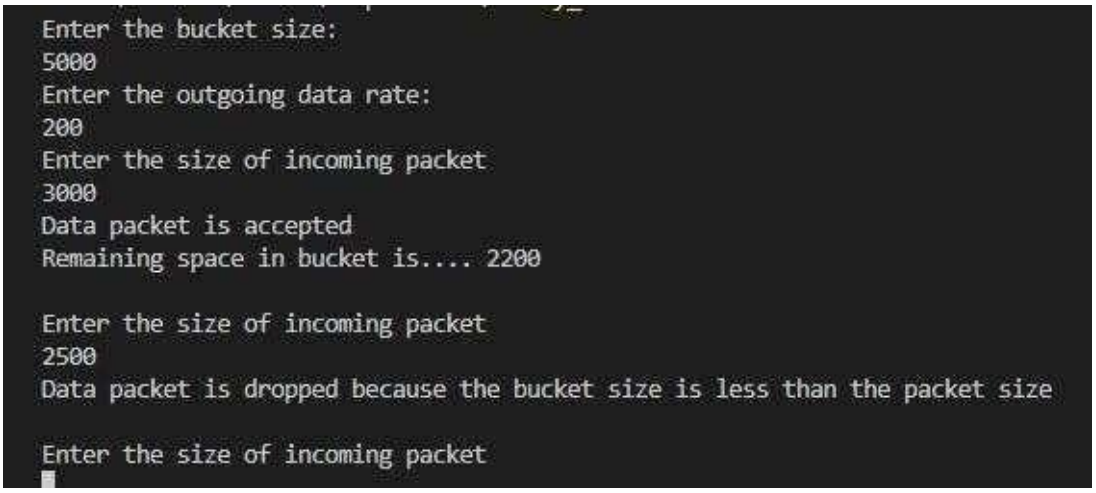
Write a program for congestion control using Leaky bucket algorithm.

CODE:

```
#include<stdio.h>
void main()
{ int b_size,d_rate,in_d_rate,rem_b_size;
   printf("Enter the bucket size:\n");
   scanf("%d",&b_size);
   rem_b_size=b_size; printf("Enter the
   outgoing data rate:\n");
   scanf("%d",&d_rate); while(1) {
   printf("Enter the size of incoming packet\n");
   scanf("%d",&in_d_rate);
   if(in_d_rate<=b_size)
   { if(in_d_rate<=rem_b_size)
      { rem_b_size=rem_b_size-in_d_rate;
      rem_b_size=rem_b_size+d_rate; printf("Data packet is
      accepted\n"); printf("Remaining space in bucket is....
      %d\n",rem_b_size); printf("\n");
      } else{ printf("Data packet is dropped because the bucket size is less than
      the packet
size\n");
      printf("\n");
   }
   }
   }
```

}

OUTPUT:


Enter the bucket size:
5000
Enter the outgoing data rate:
200
Enter the size of incoming packet
3000
Data packet is accepted
Remaining space in bucket is.... 2200
Enter the size of incoming packet
2500
Data packet is dropped because the bucket size is less than the packet size
Enter the size of incoming packet

Lab - 9 [J]

Write a program for error detecting code using CRC

```
#include <stdio.h>
#include <string.h>
#define N strlen(Poly)
char data[30];
char check_value[30];
char Poly[10];
int data_length, i, j;
Void xor
{
  for (j=1; j<N; j++)
   check value [j] = (check value [j] == poly [j])?0:1;
}
Void receiver()
{
  printf ("Enter the received data :");
  Scanf ("%s", data);
  printf ("Data received : %s", data);
  crc();
  for (i=0; i<N-1 && (check_value[i] != '1'); i++)
  if (N-1)
  printf ("Error detected");
  else
  printf ("No error detected");
}
Void crc()
{
  for (i=0; i<N; i++)
   check value [i] = data [i];
do {
```

```
    if (check value[0] == '1')
      XOR();
    for (j=0; j<N-1; j++)
      check_value[j] = check value[j+1]
      check_value[j] = data[j+1];
    } while (i<= data.length+N+1);
}
int main()
{
    printf ("Enter data to be transmitted: ");
    scanf ("%s", &data);
    printf (" Enter the divisor polynomial: ");
    scanf ("%s", &poly);
    data.length = strelen (data);
    for (i= data.length ; i<datalength+N-1 ;i++)
        data[i]=0;
    printf ("Data padded with n-1 zeros : %s ) data
        crc();
    printf (" CRC value is %s ) check_value);
    for (i= datalength ; i <data.length+N-1 ; i++)
      data[i] = check value[i-data.length];
    printf (" final dataword to be sent : %s da
    receiver ();
    return 0;
}
```

Output:

Enter data to be transmitted : 101010

Enter the divisor polynomial : 1011

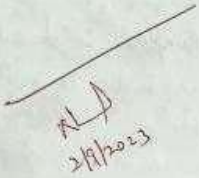
Data padded with zeros : 101010000

CRC value is : 001

Final code word to be sent : 101010001

Enter the recived data : 100010000

Error detected.



27/12/2023

Lab. 09 [ii]

Write a program for Congestion Control using leaky bucket algorithm

```
#include <stdio.h>
int main()
{
 int incoming, outgoing, buck_size, n, store=0;
 printf("Enter the bucket size: ");
 scanf("%d", &outgoing);
 printf("Enter the no of inputs: ");
 scanf("%d", &n);
 while(n!=0)
 {
  printf("Enter the incoming bucket size: ");
  scanf("%d", &incoming);
  if(incoming <= (buck_size-store))
  {
   store = incoming;
   printf("Bucket Buffer size %d out of %d\n",
          bucket size);
  }
  else
  {
   printf("Dropped %d no of packets\n");
   incoming = (bucket_size - store);
   printf("Bucket Buffer Size %d of %d\n", store,
          buck_size);
   store = buck_size;
  }
```

```
        store = store - outgoing;
        printf("After outgoing %d packets left out of
          %d in buffer\n", store, buck_size);
    }
}
```

Output:

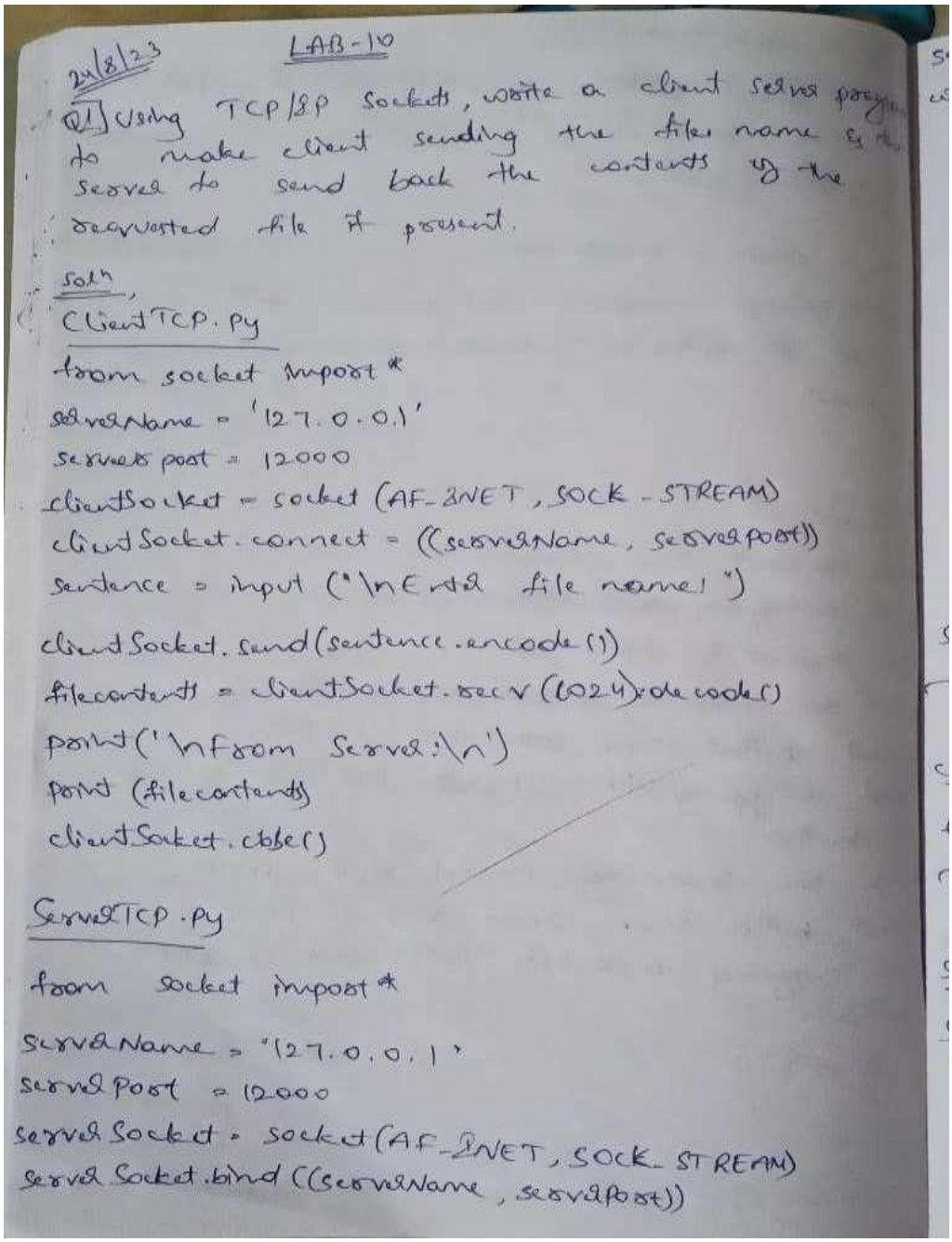
Enter bucket size : 5000
Enter outgoing rate : 2000
Enter number of ~~packets~~ inputs : 2
Enter the incoming packet size : 3000
Bucket Buffer size 3000 out of 5000
After outgoing 1000 packets left out of 5000 in buffer
Enter the incoming packet size : 1000
Bucket buffer size 2000 out of 5000
After outgoing 0 packets left out of 5000 in buffer.

NLP
21/9/2023

LAB 14 Program 1

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.

Observation:



24/8/23 LAB-10

Q1] Using TCP/IP sockets, write a client server program to make client sending the file name & the server to send back the contents of the requested file if present.

Soln

ClientTCP.py

```
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()
```

ServerTCP.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('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Output

Server Side:

The server is ready to receive

Client Side:

Enter file name: serverTCP.py

The contents of file serverTCP is displayed here.

Server Side:

Sent contents of serverTCP.py.

SOLUTION:

ClientTCP.py 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()

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

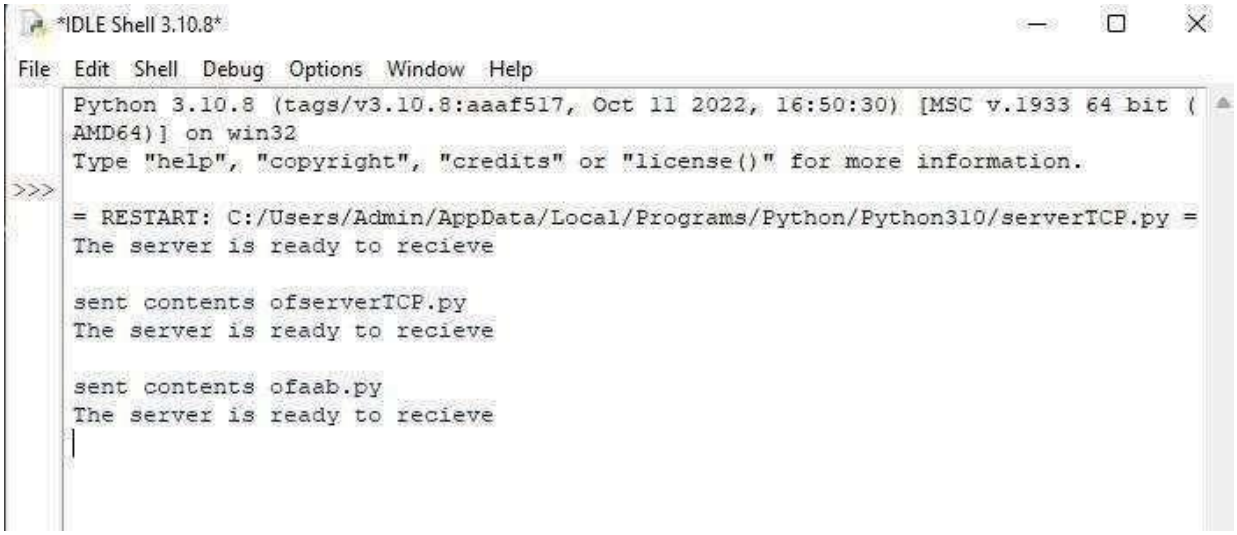
OUTPUT:

Client:

```
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 recieve")
    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()
```

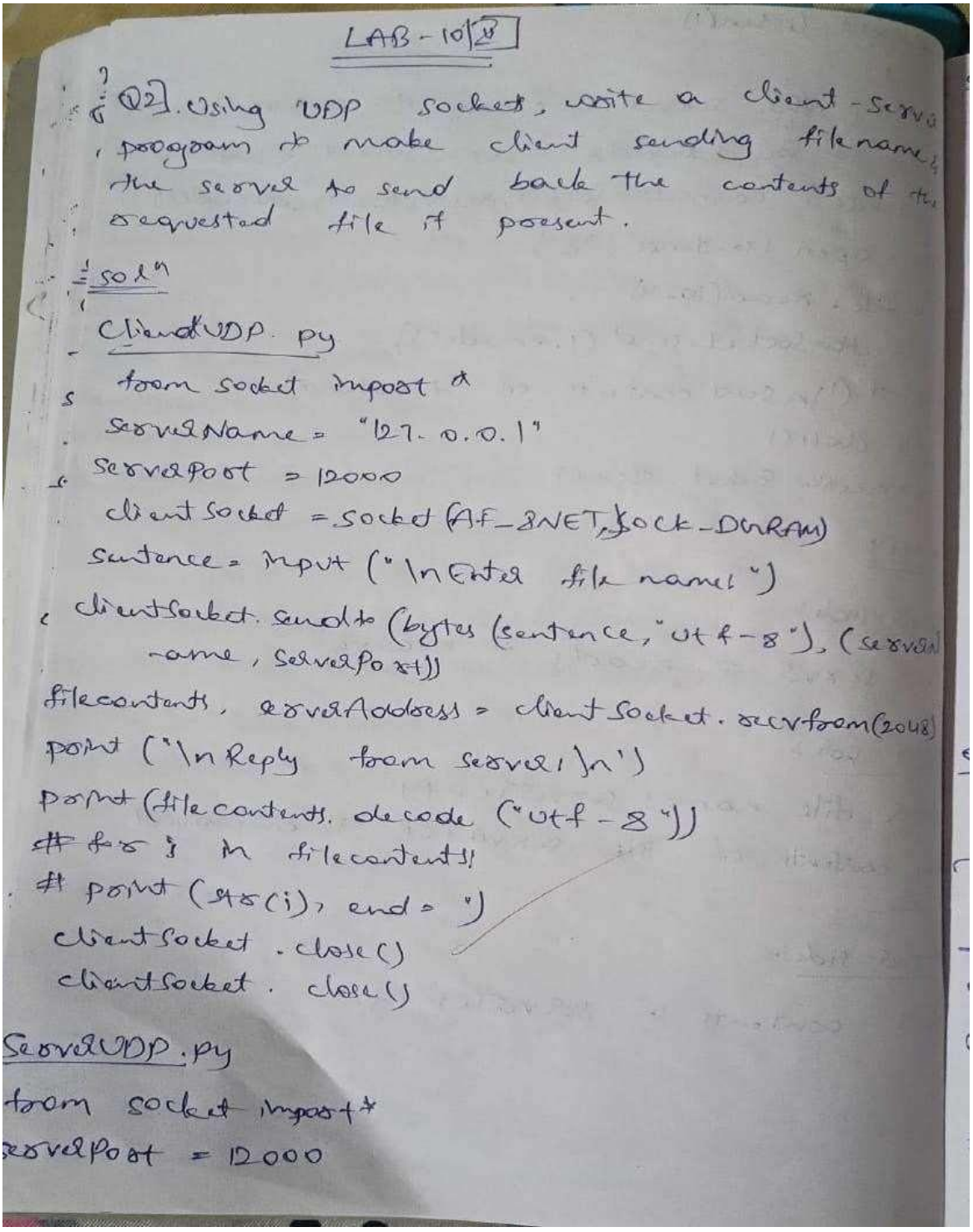
Server:

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter file name:aab.py

From Server:

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.
class Node:
    def __init__(self,data):
        self.data=data
        self.left=None
        self.right=None
        self.height=1

class AVL Tree:
    def getHeight(self,root):
        if not root:
            return 0
        return root.height

    def getBalance(self,root):
        if not root:
            return 0
        return self.getHeight(root.left)-self.getHeight(root.right)

    def rightRotate(self,z):
        y=z.left
        T3=y.right

        y.right=z
        z.left=T3

        z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
        y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

        return y

    def insert(self,root,data):
        if not root:
            return Node(data)
        if data < root.data
            root.left=self.insert(root.left,data)
        else:
            root.right=se
>>>
```


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 recieve
sent contents ofserverTCP.py
The server is ready to recieve
sent contents ofaab.py
The server is ready to recieve

Program 2

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.



LAB - 10

Q2. Using UDP sockets, write a client-server program to make client sending filename, the server to send back the contents of the requested file if present.

Soln

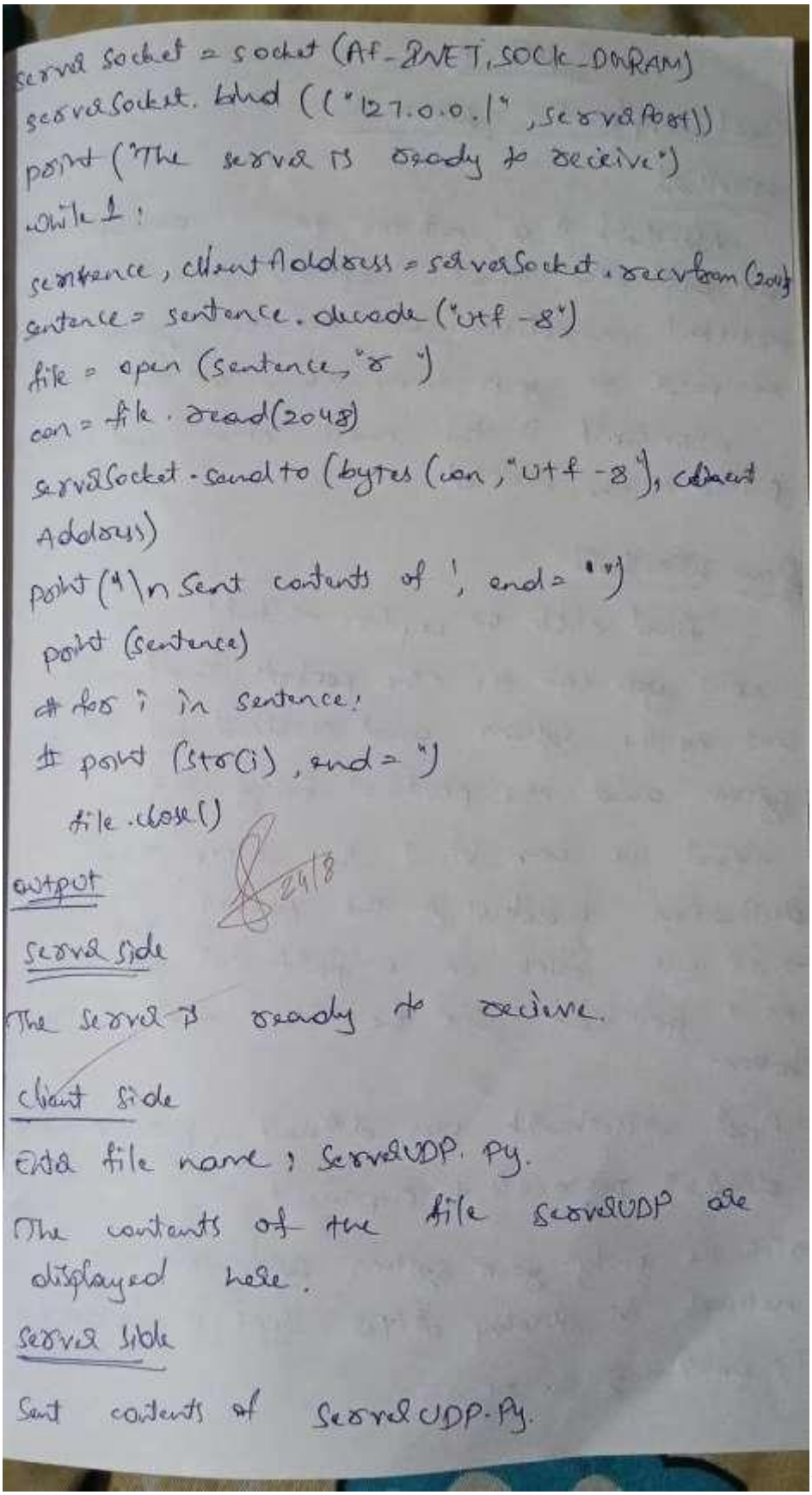
ClientUDP.py

```
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(2048)
print("\nReply from server:\n")
print(fileContents.decode("utf-8"))
# for i in fileContents:
# print(str(i), end = "")
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
```

Observation:



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

output

Server side

The server is ready to receive.

client side

Enter file name : serverUDP.py.

The contents of the file serverUDP are displayed here.

Server side

Sent contents of serverUDP.py.

SOLUTION:

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(2048)
print ('\nReply from Server:\n') print
(filecontents.decode("utf-8")) # for i in filecontents:
    # print(str(i), end = '')
clientSocket.close()
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(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()
```

OUTPUT:

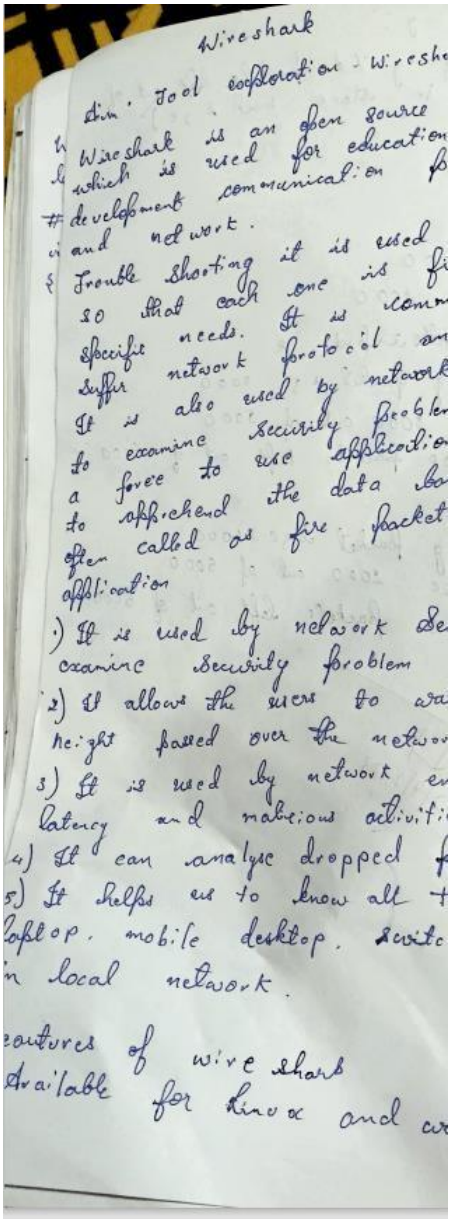
Client:

```
= 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()

>>>
```

Server:

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =
The server is ready to receive

Sent contents of  serverUDP.py
```

Lab 15
Observation



→ Capture wire packet data from a network interface
→ Open files containing packet data captured with tcp dump / win Dump, wire shark and many other packet capture programs
→ Display packets with very detailed protocol & information.
→ Save packet dates captured.
→ Filter packets on many criteria
→ Create various statistics.

21/9/2023