VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

COMPUTER NETWORKS LAB

Submitted by

PRANNAV D (1BM21CS046)

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 NETWORKS LAB" carried out by **PRANNAV D (1BM21CS046),** 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 NETWORKS - (22CS4PCCON)** work prescribed for the said degree.

M Lakshmi Neelima
Assistant Professor
Department of CSE
BMSCE, Bengaluru

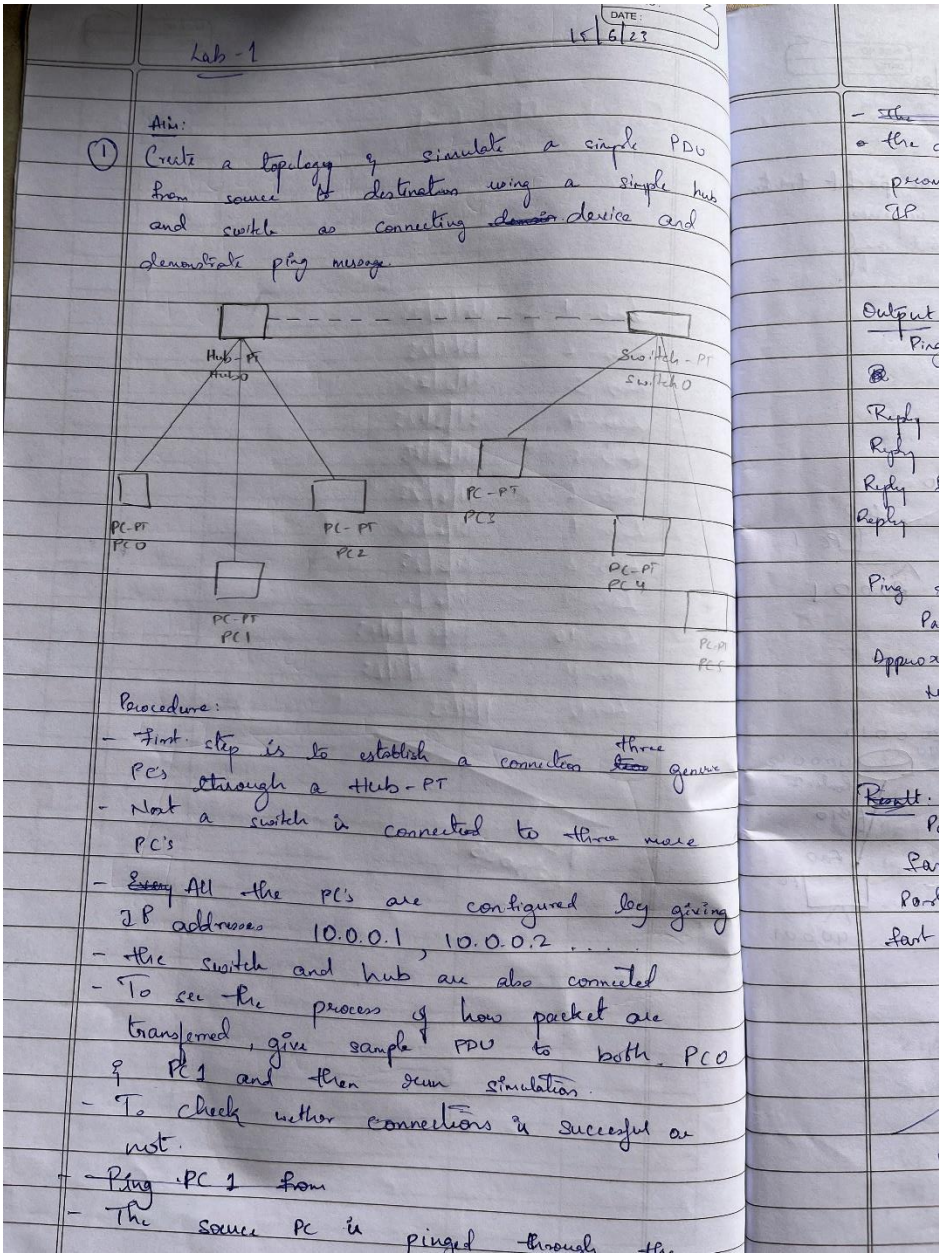
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

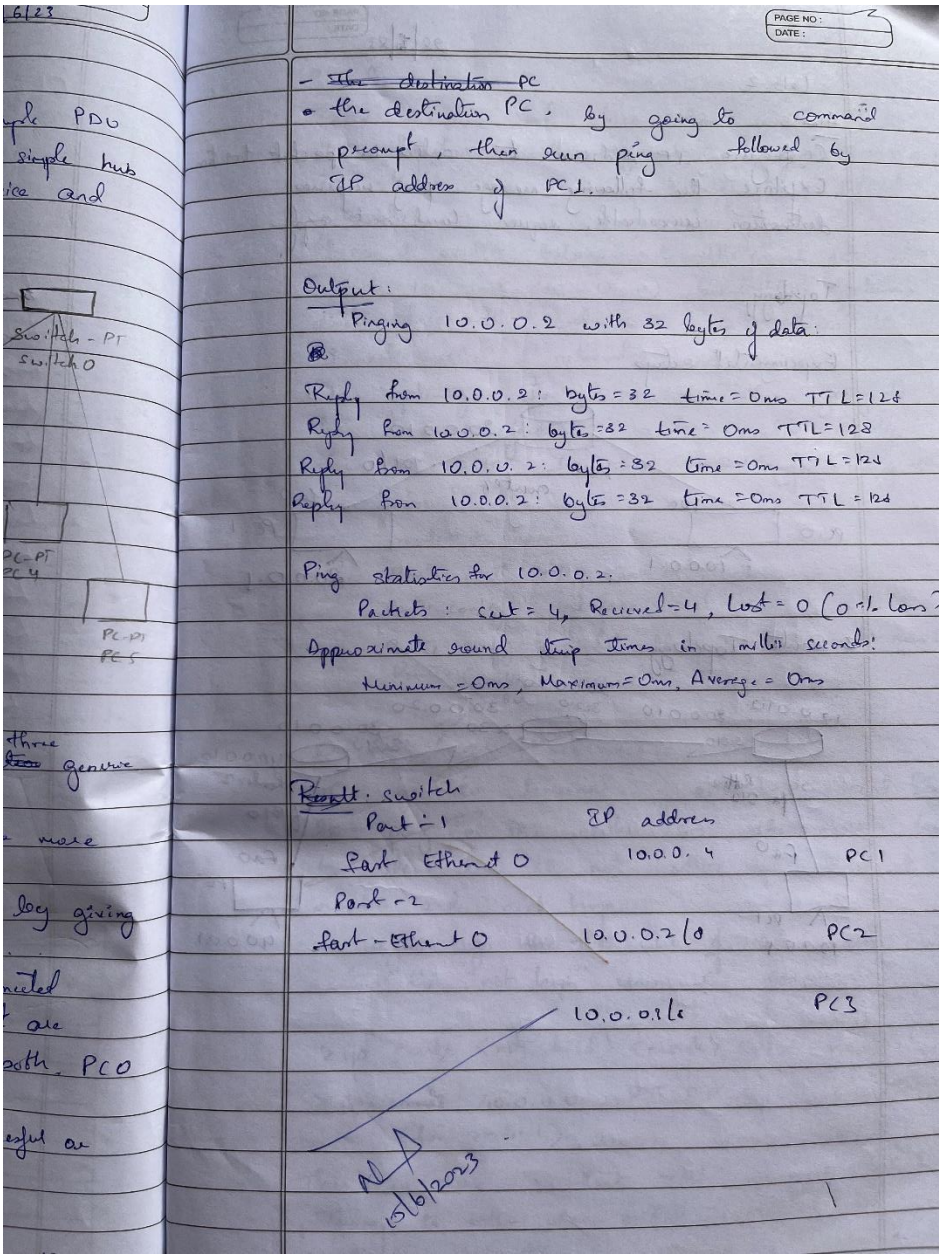
Index

LAB 1

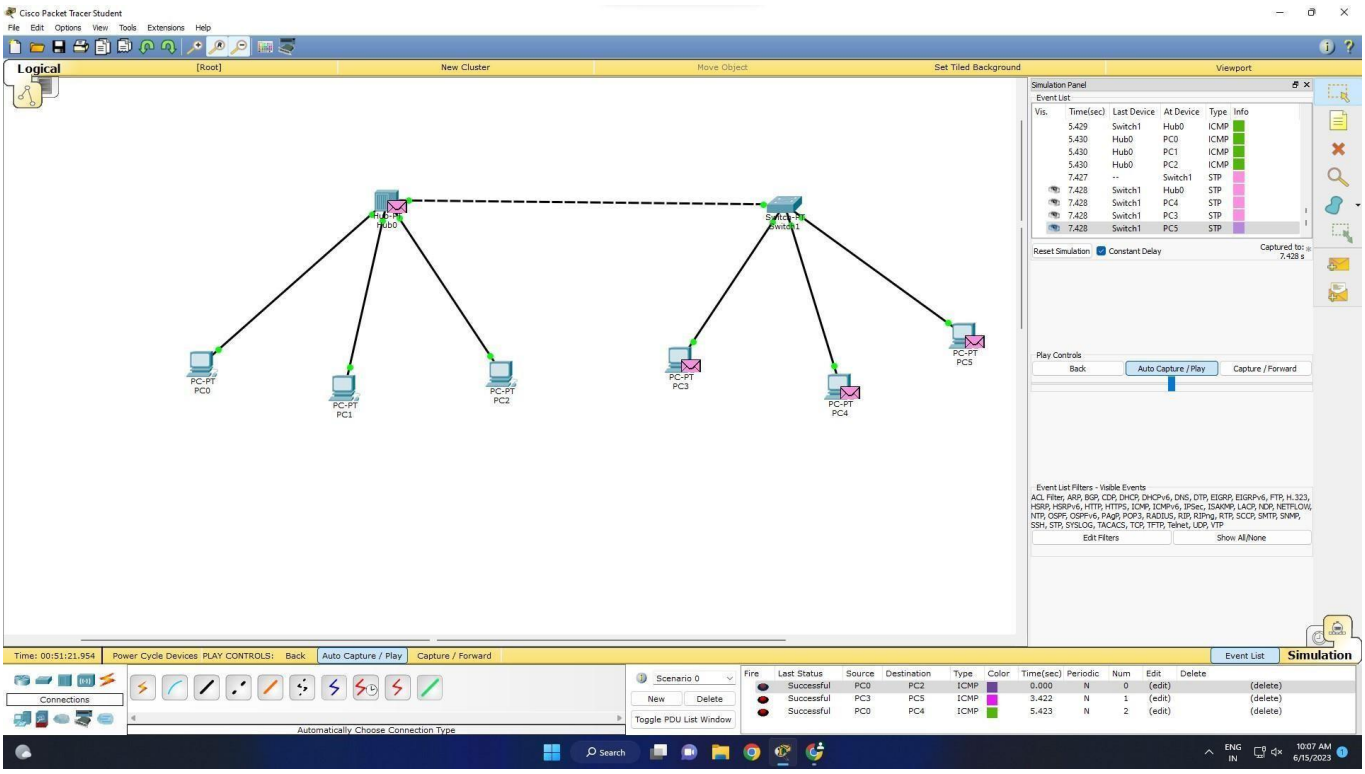
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

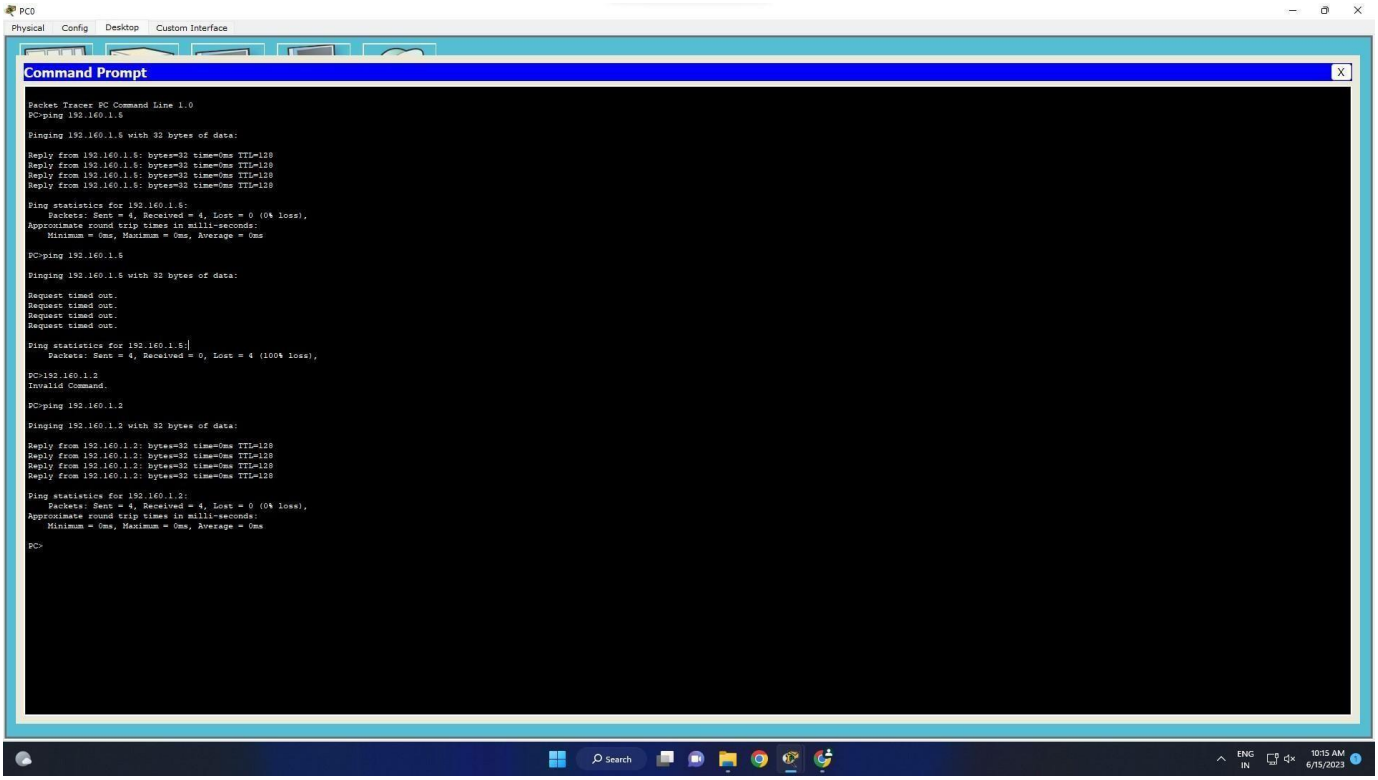
OBSERVATION:

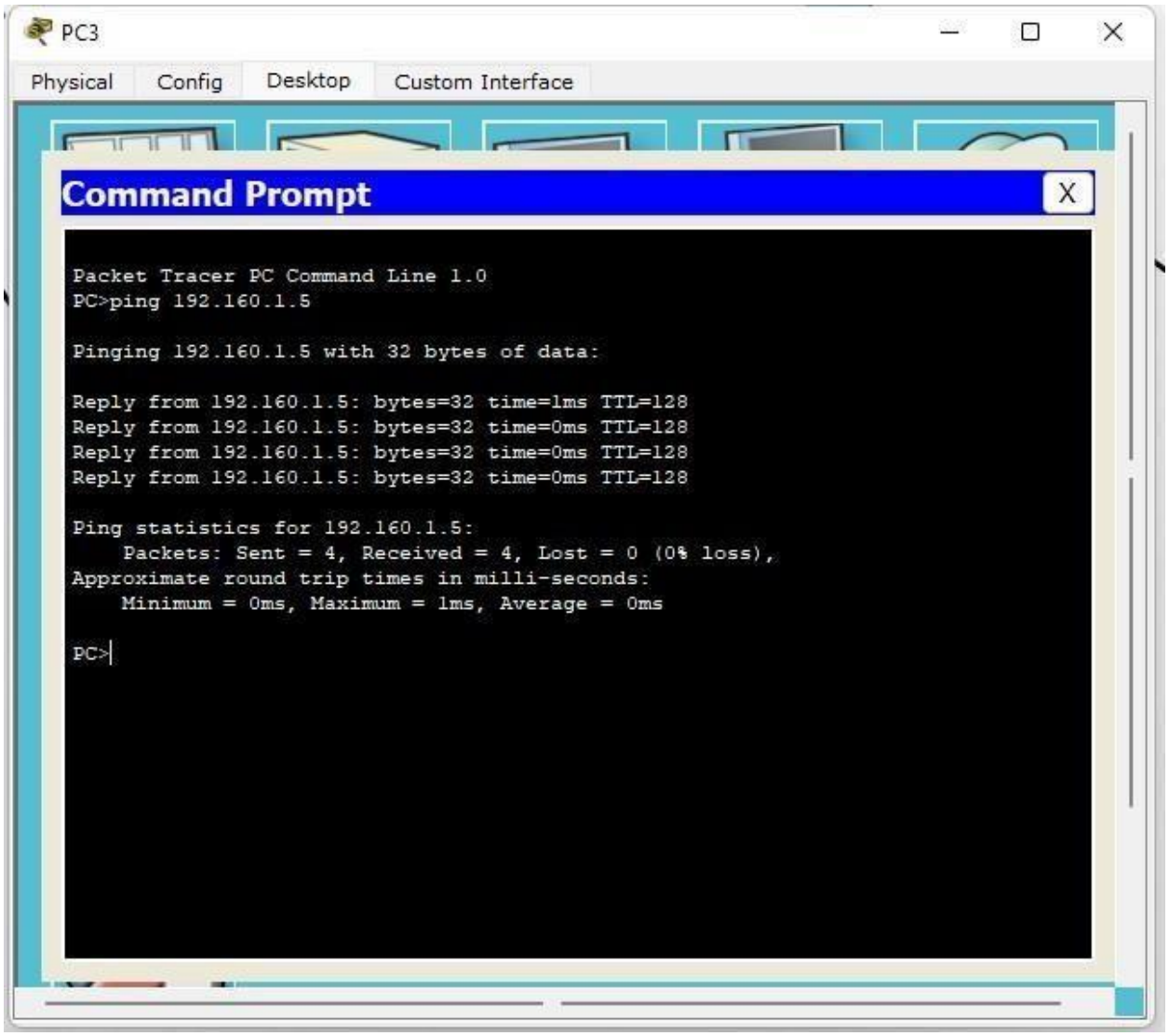



PAGE NO :
DATE :
- the destination PC
• the destination PC, by going to command prompt, then run ping followed by IP address of PC1.
Output:
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms 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
Port : switch
Port=1 IP address
Fast Ethernet 0 10.0.0.4 PC1
Port-2
Fast-Ethernet 0 10.0.0.2 PC2
10.0.0.3 PC3
15/6/2023

OUTPUT:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub-PT Hub0
Switch-PT Switch1
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
5.429 Switch1 Hub0 ICMP
5.430 Hub0 PC0 ICMP
5.430 Hub0 PC1 ICMP
5.430 Hub0 PC2 ICMP
7.427 -- Switch1 STP
7.428 Switch1 Hub0 STP
7.428 Switch1 PC4 STP
7.428 Switch1 PC3 STP
7.428 Switch1 PC5 STP
Reset Simulation
Constant Delay
Captured to: 7.428 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:51:21.954
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
Successful PC0 PC2 ICMP 0.000 N 0 (edit) (delete)
Successful PC3 PC5 ICMP 3.422 N 1 (edit) (delete)
Successful PC0 PC4 ICMP 5.423 N 2 (edit) (delete)
Search
ENG IN
10:07 AM 6/15/2023

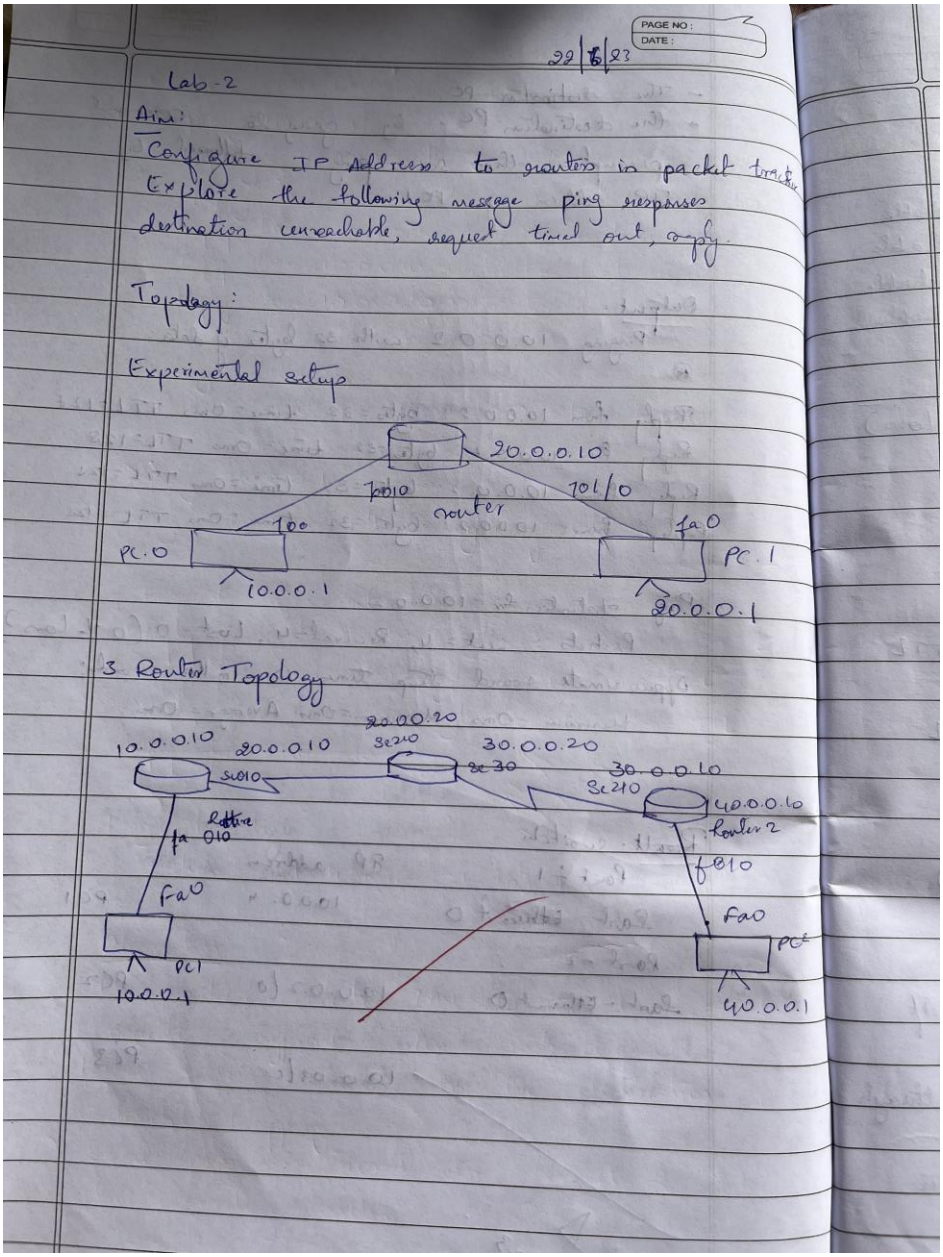

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
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 = 0ms, Average = 0ms
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.160.1.5:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>192.160.1.2
Invalid Command.
PC>ping 192.160.1.2
Pinging 192.160.1.2 with 32 bytes of data:
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
Search
ENG IN
10:15 AM 6/15/2023


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:



DATE :

Result

(i) > 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: Recieved = 0: loss = 4 (100: loss)

> ping 40.0.0.1

pinging 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

~~Reply from 40.0.0.10 by~~

NLA 13/9/2023

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 ids. Packets are forwarded to the destination through network hopping.

Serial pass are used to connect 2 router the connecting cables.

PAGE NO :
DATE :

Procedure:

Take 2 PC and place them as shown in the topology, 2 different IP address (10.0.0.1 and 40.0.0.1) as they belong to 2 different networks.

Place 2 routers belonging to these 2 networks (10.0.0.10 and 40.0.0.10) using their gateways and place the 3rd router in between to connect the 2 networks.

Set gateways for 2 PCs and then going to the CLI interface for each router specify the IP router for file to transfer using the commands

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

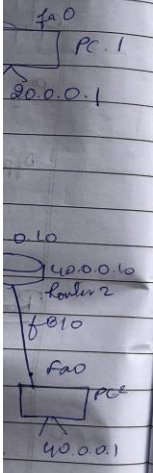
Then go to terminal of either PC and try to ping to the either one, the message fails to deliver due to not setting up network static routes and next hop.

We again go for the CLI of each router and setup the "net hop" using the commands

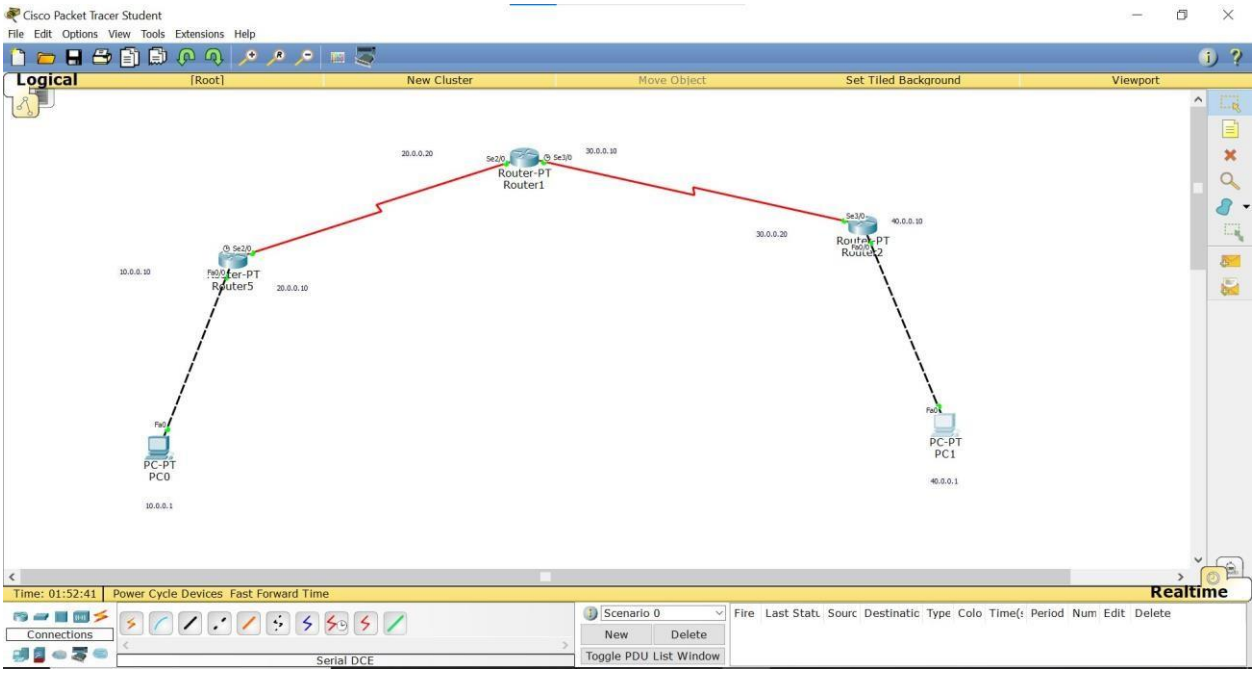
> ip route <network id> <mask> <next-hop>

> ip route 40.0.0.0 255.0.0.0 20.0.0.20
(for router 1)

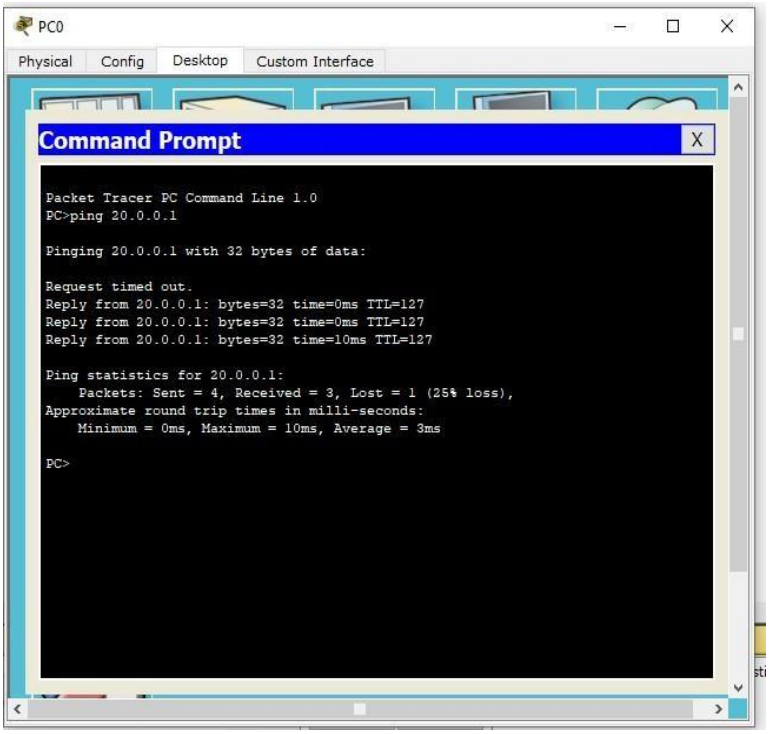
This is done so that router recognises which pathway to take when packet is required for particular destination

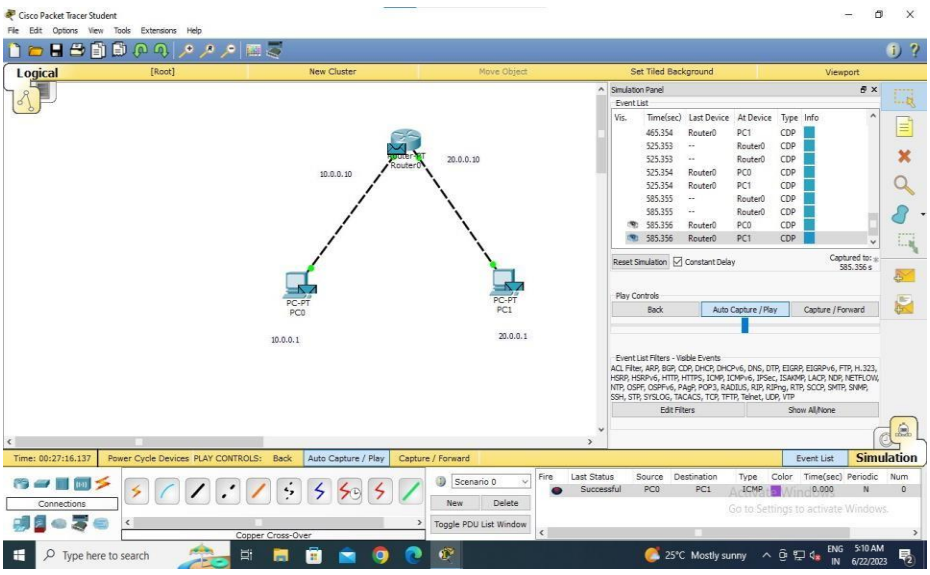


TOPOLOGY:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router5
Router2
PC-PT
PC0
PC-PT
PC1
20.0.0.20
30.0.0.10
30.0.0.20
40.0.0.10
10.0.0.10
20.0.0.10
10.0.0.1
40.0.0.1
Time: 01:52:41
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Statu
Sourc
Destinatic
Type
Colo
Time(s
Period
Num
Edit
Delete

OUTPUT:


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>


Cisco Packet Tracer Student
Logical
10.0.0.10
20.0.0.10
PC-PT PC0
PC-PT PC1
10.0.0.1
20.0.0.1
Simulation

LAB 3

Configure default route, static route to the Router.

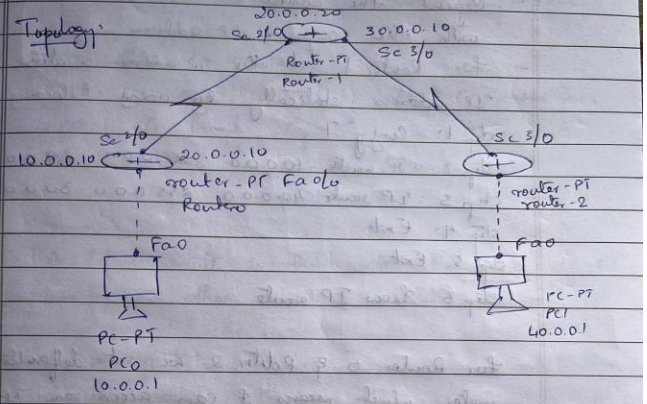
OBSERVATION:

PAGE NO:
DATE: 13/7/23

Lab 3

Aim: Configure default route, static route to three router

Topology:



Procedure:

- Connect 3 routers and 2 PC's using copper cross-over cable for PC to router and a serial DCE cable to connect router to router.
- Set the IP address of both PC's & respective gateway
- For all 3 routers set the respective IP address in CLI mode by using the commands.

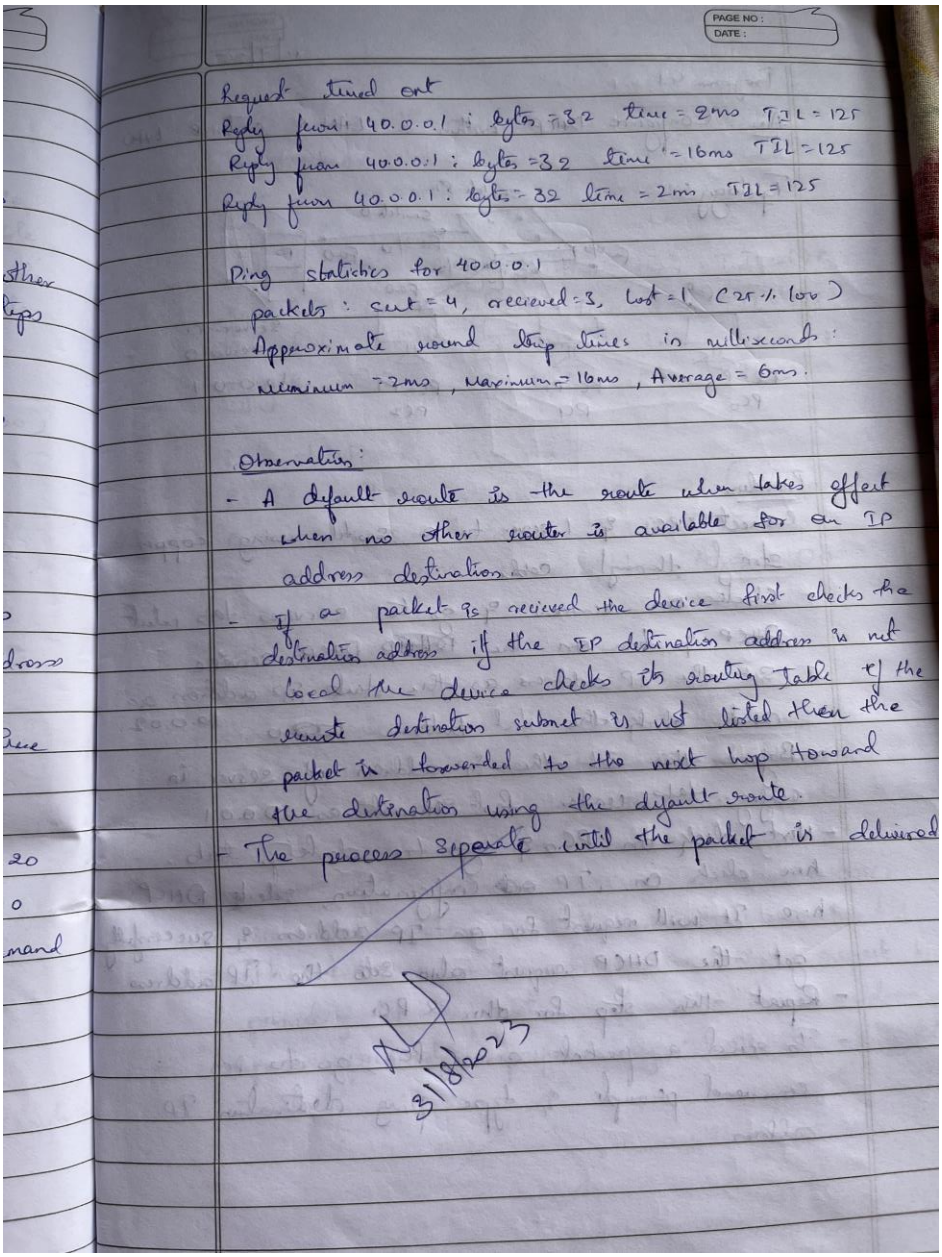
Step 1: Enable
Step 2: Config T
Step 3: interface Fast Ethernet 0/0
Step 4: IP address 10.0.0.10 255.0.0.0
Step 5: No shut
Step 6: Exit
Step 7: Interface Se 2/0.
Step 8: IP address 20.0.0.10 255.0.0.0
Step 9: No shut

step 10: Exit
step 11: Exit

- Repeat the commands for other 2 routers with the respective IP address
- for Router 1, set the IP ~~address~~ router if other IP address, statically by using following steps
  step 1: Config T
  step 2: IP route 10.0.0.0 255.0.0.0 20.0.0.10
  step 3: IP route 40.0.0.0 255.0.0.0 30.0.0.20
  step 4: Exit
  step 5: Exit
  step 6: Show IP route

- for Router 0 & Router 2 we set default IP router which means it can access any IP address with any subnet mask.
- Set the default IP route by following these commands
  step 1: Config T
  step 2: IP route 0.0.0.0 0.0.0.0 20.0.0.20
  step 3: IP route 0.0.0.0 0.0.0.0 30.0.0.10
- step 2 is given for router 0 & step 3 command for Router 1
- Go to PC's command prompt & type ping message to send packet across

Ping output:
Packet tracer PC command line 1.0
PC > ping 40.0.0.1
pinging 40.



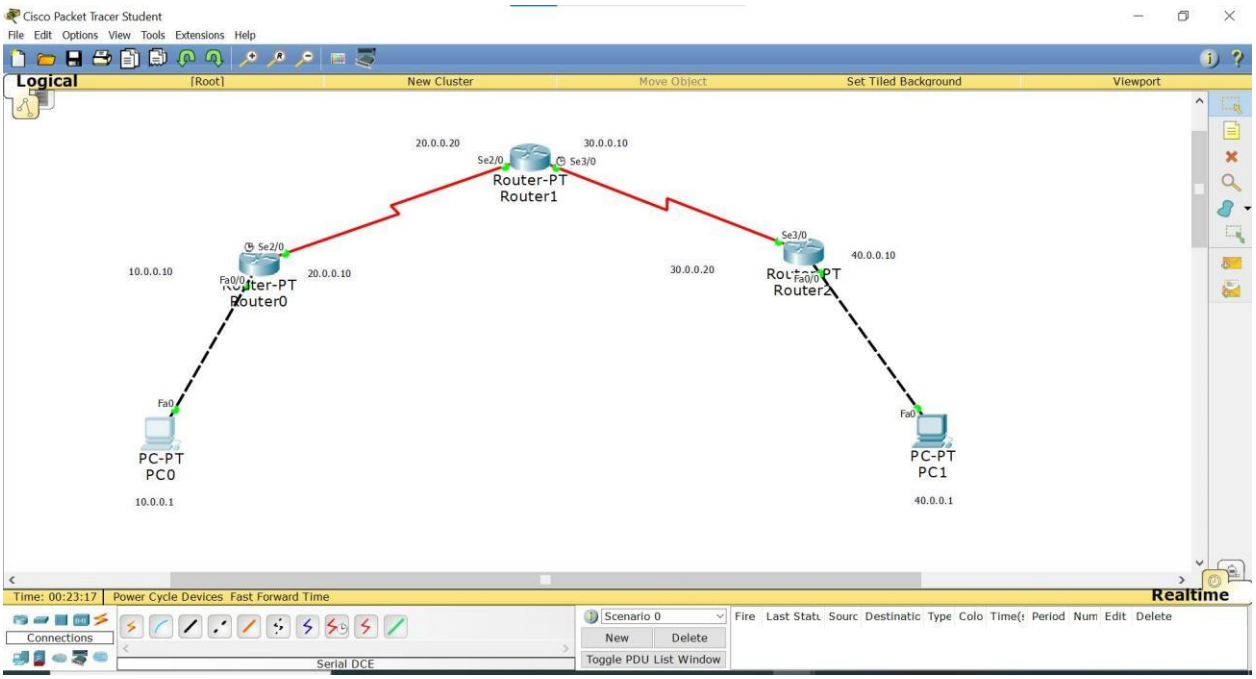
Request timed out
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 16ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 125

Ping statistics for 40.0.0.1
packets : sent = 4, received = 3, lost = 1 (25% loss)
Approximate round trip times in milliseconds :
Minimum = 2ms, Maximum = 16ms, Average = 6ms.

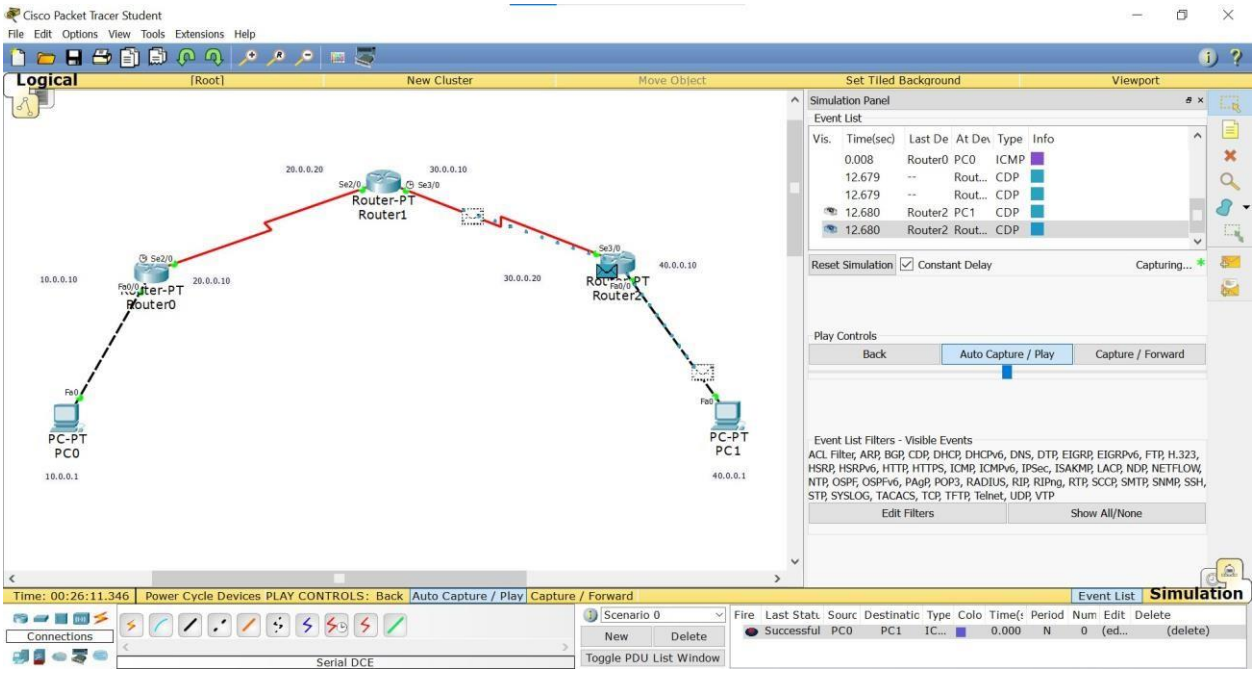
Observations :
- A default route is the route which takes effect when no other router is available for an IP address destination.
- If a packet is received the device first checks the destination address if the IP destination address is not local the device checks its routing table if the remote destination subnet is not listed then the packet is forwarded to the next hop toward the destination using the default route.
- The process repeats until the packet is delivered

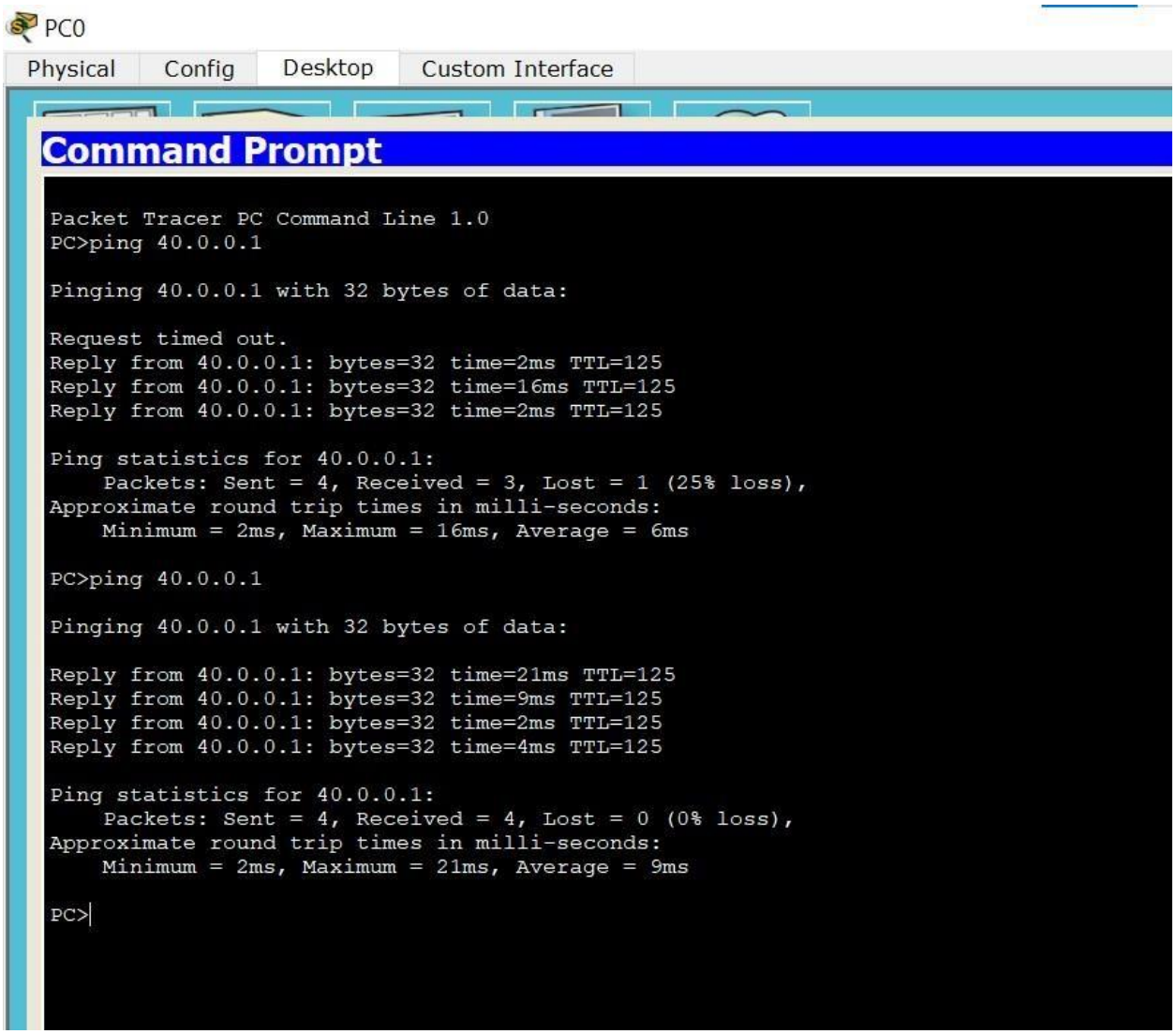
31/10/23

TOPOLOGY:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
10.0.0.10
20.0.0.10
30.0.0.20
40.0.0.10
Fa0/0
Router0
Router2
Fa0
PC-PT
PC0
PC1
10.0.0.1
40.0.0.1
Time: 00:23:17
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Period
Num
Edit
Delete

OUTPUT:

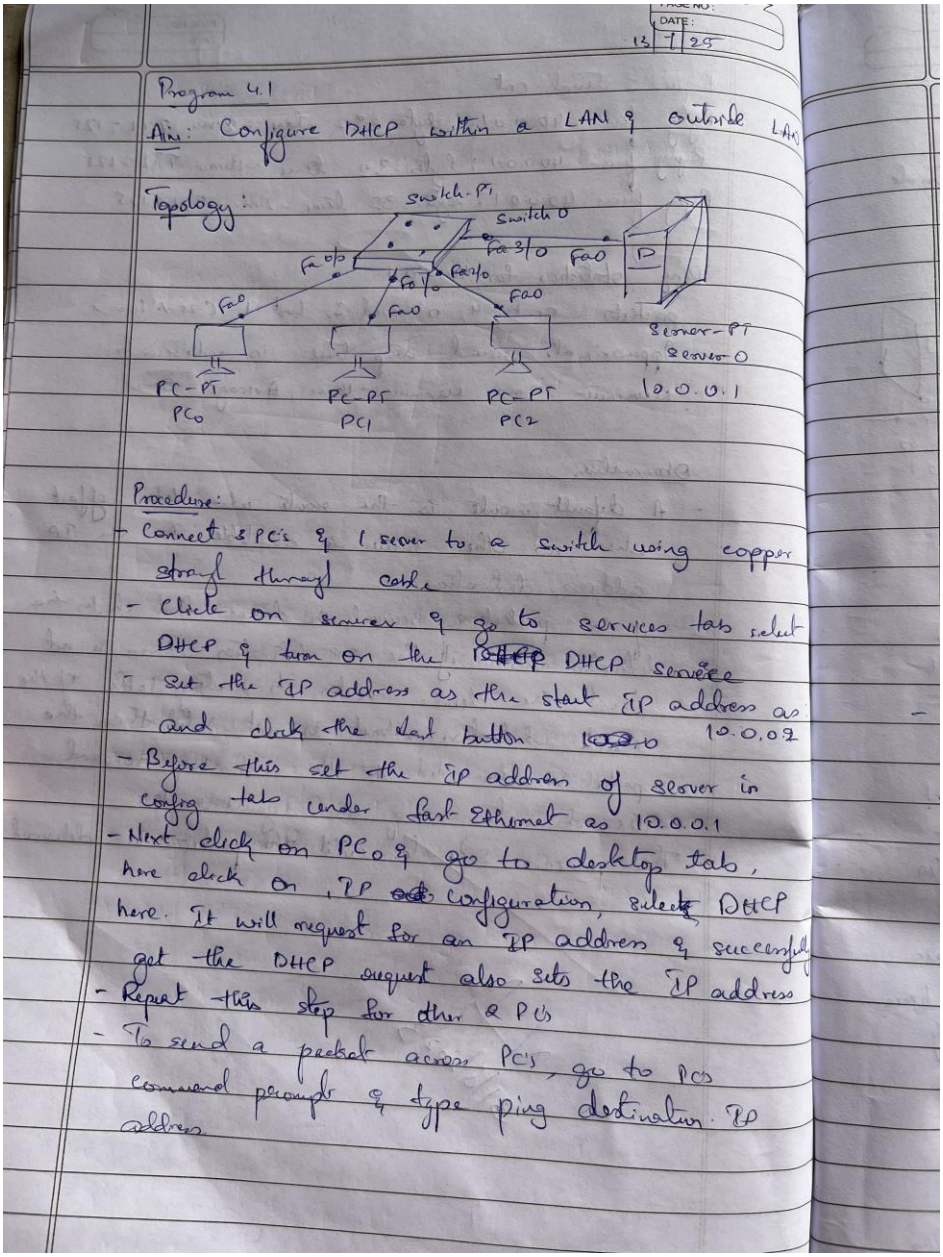

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.008
Router0
PC0
ICMP
12.679
Router1
CDP
12.680
Router2
PC1
Reset Simulation
Constant Delay
Capturing...
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: 00:26:11.346
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Period
Num
Edit
Delete
Successful
PC0
PC1
0.000
(edit)
(delete)


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>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=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 16ms, Average = 6ms
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=21ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 21ms, Average = 9ms
PC>

LAB 4

Configure DHCP within a LAN and outside LAN.

OBSERVATION:
TOPOLOGY:



PAGE NO:
DATE:

ping output:

Packet tracer PC command line 1.0

PC> Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data.

Reply from 10.0.0.3 bytes=32 time=0ms TTL=128
Reply from 10.0.0.1 bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 bytes=32 time=1ms 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 milliseconds:

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

Observation:

- DHCP is used to dynamically assign an IP address to any device or node.
- It is a client server protocol in which servers manage a pool of unique IP addresses & also about client configuration parameters.
- DHCP-enabled clients sends a request to DHCP server when they want to connect to a network.
- The DHCP server responds to the client request by providing IP configuration information from address pools, previously specified by a network administration.

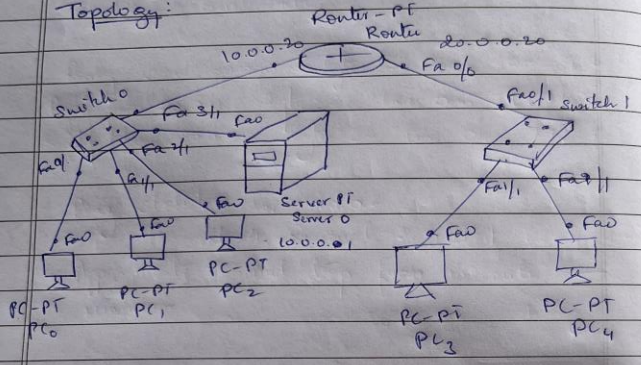
3/8

DATE: 13-7-23

Program 4.2

Aim: Configure DHCP within a LAN & outside LAN

Topology:



Procedure:

* Add a router a switch & 2 PCs to 4.1 program network & connect the router to both switches.
* Set the server IP address of server and with the help of server set the first 3 PCs IP address through DHCP
* Now set the router IP address with the following commands statistically.

Step 1: No

Step 2: Enable

Step 3: Config T

Step 4: Enterface fast Ethernet 4/0

Step 5: IP address 10.0.0.20 255.0.00

Step 4: No Shut

Step 5: Exit

PAGE NO :
DATE :

& outside

switch 1

Fa0/1

Fa0

C-PT
PC4

to 4.1

to

and

PAGE NO :
DATE :

Step 8: Interface fast Ethernet 0/0
Step 9: IP address 20.0.0.20 255.0.0.0
Step 10: No shut
Step 11: Exit
Step 12: Exit
Step 13: show IP ~~address~~ route.

* Go to server & set the gateway as 10.0.0.20
* Again go to router CLI & follow these commands.
  Step 14: Config T
  Step 15: interface fast Ethernet 0/0
  Step 16: IP helper-address 10.0.0.1
  Step 17: No shut
  Step 18: Exit
* Now go to server services & add one more ~~pool~~ as server pool 1, start IP address as 20.0.0.2 & default gateway as 20.0.0.20. Then click add & save.
* Now set the other 2 pc's IP address by going to Desktop → IP config & select DHCP which will automatically generate the IP address.
* Now the network is complete & can send packets from any PC to other by typing ping destination IP address in their respective command prompts.

8/8/2023

Ping output:

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)

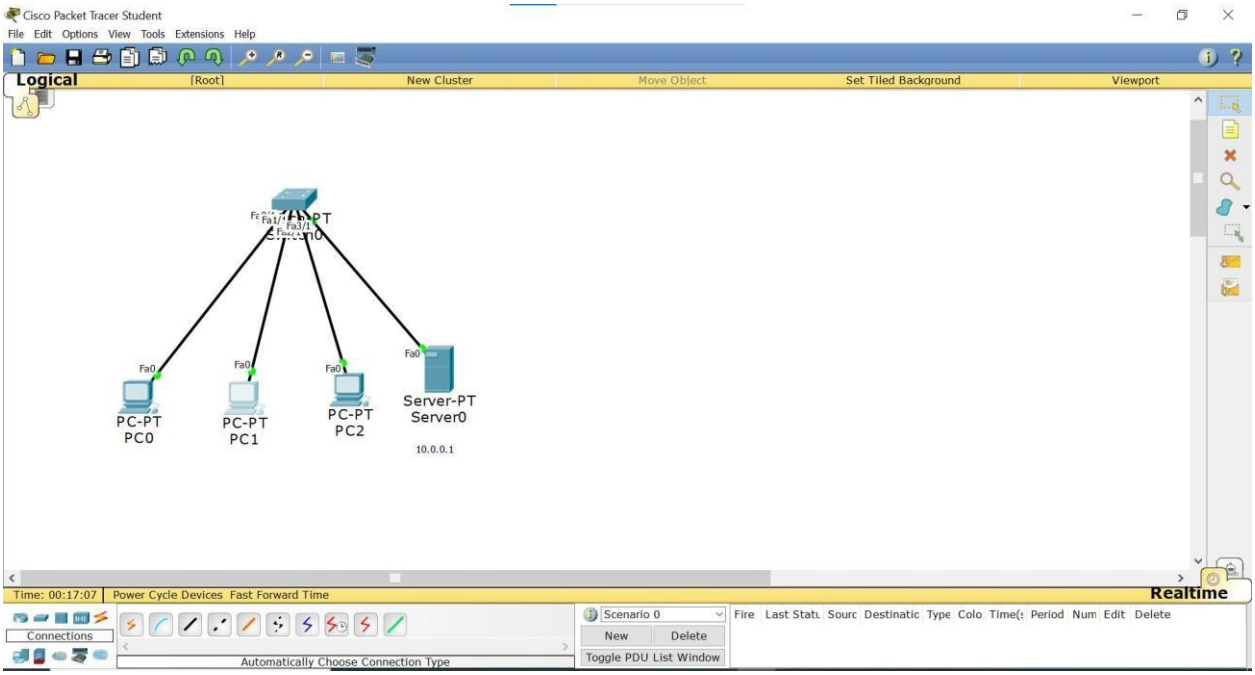
Approximate round trip from in milliseconds

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

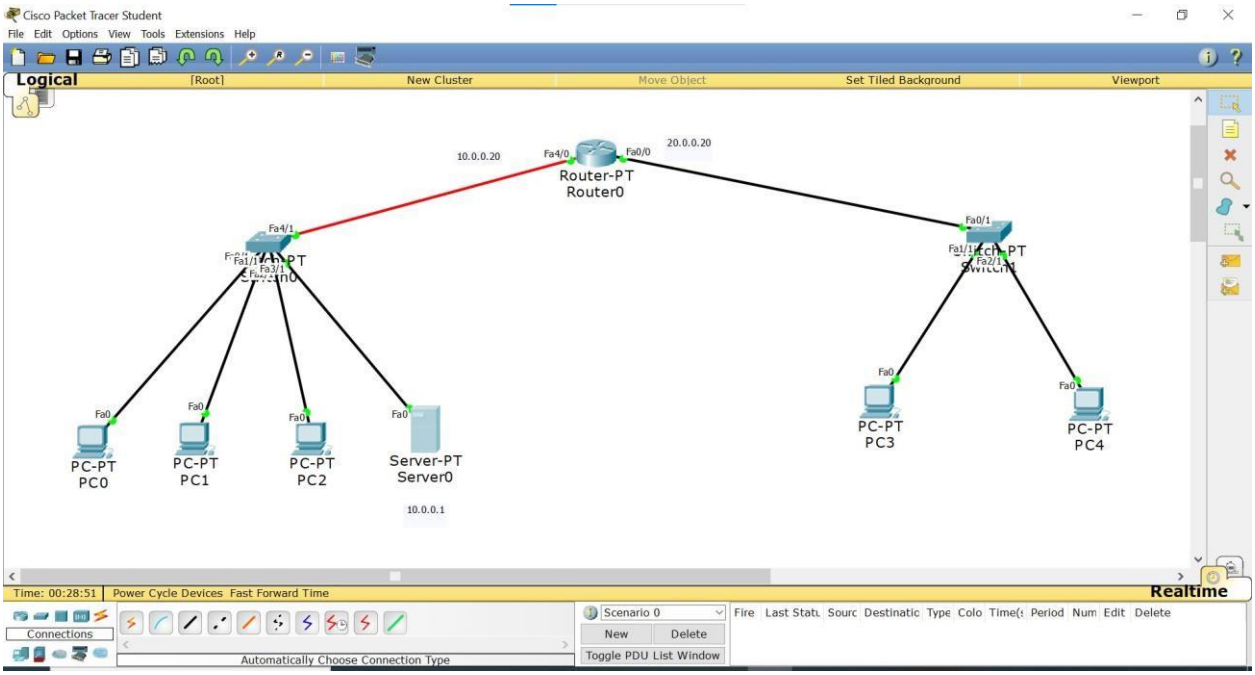
Observation

* DHCP is used to assign IP address dynamically to different devices.
* To assign continuous IP address we create a server pool where we assign the starting IP address & a default gateway number.
- for PCs under different switches we create a different server pool again & start.
- This tasks are of delivering the packets to correct destination IP address & also send back the packets into original device.

PROGRAM 4.1:

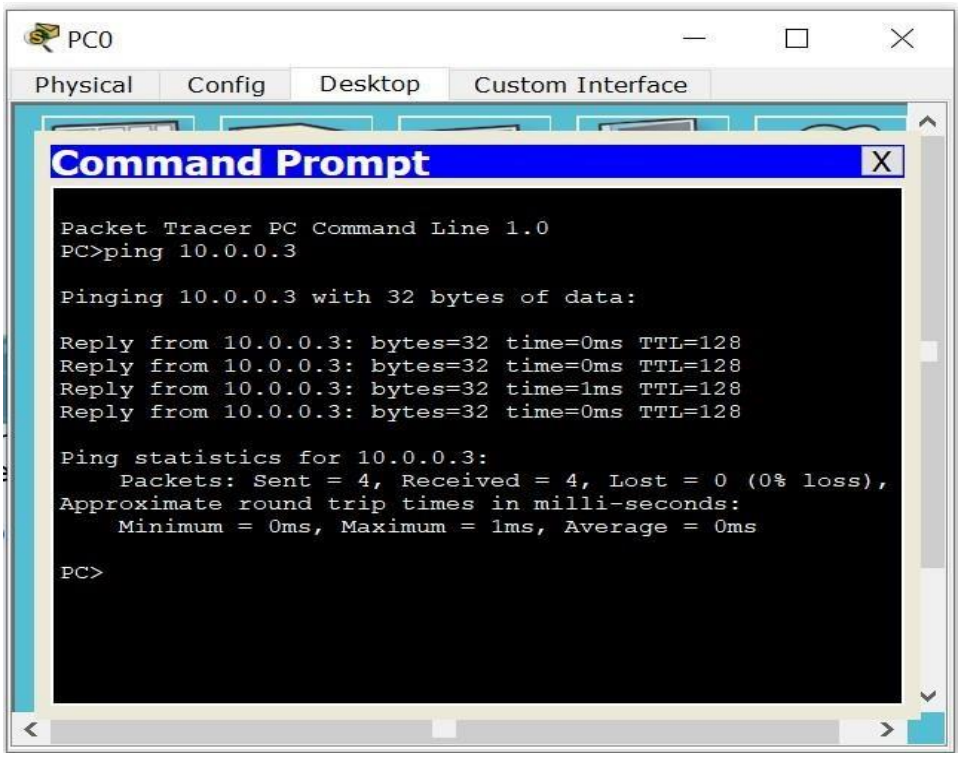


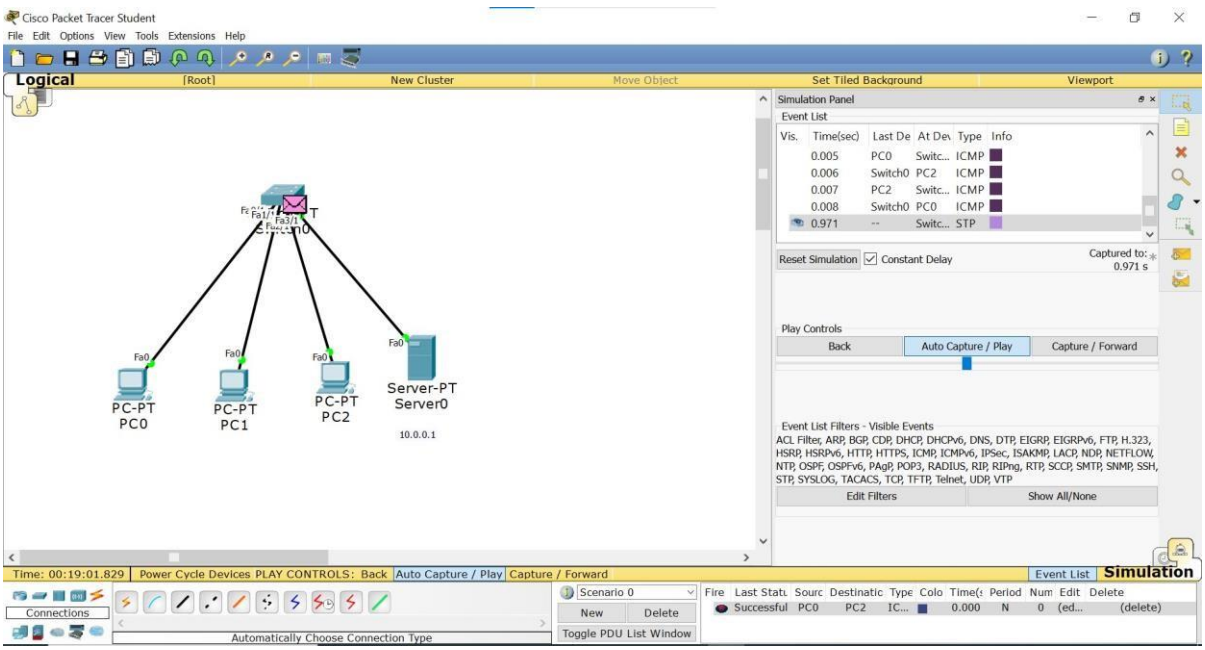
PROGRAM 4.2:



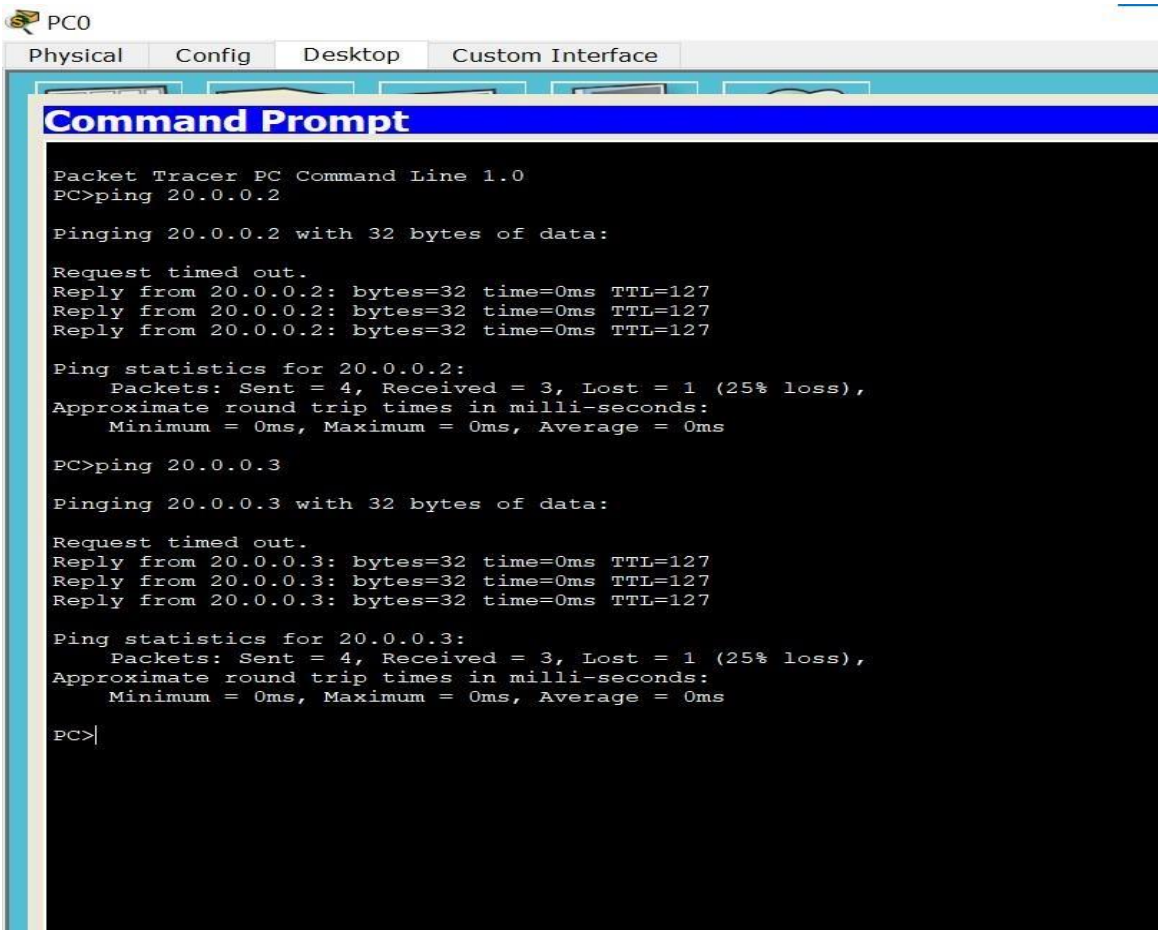
OUTPUT:

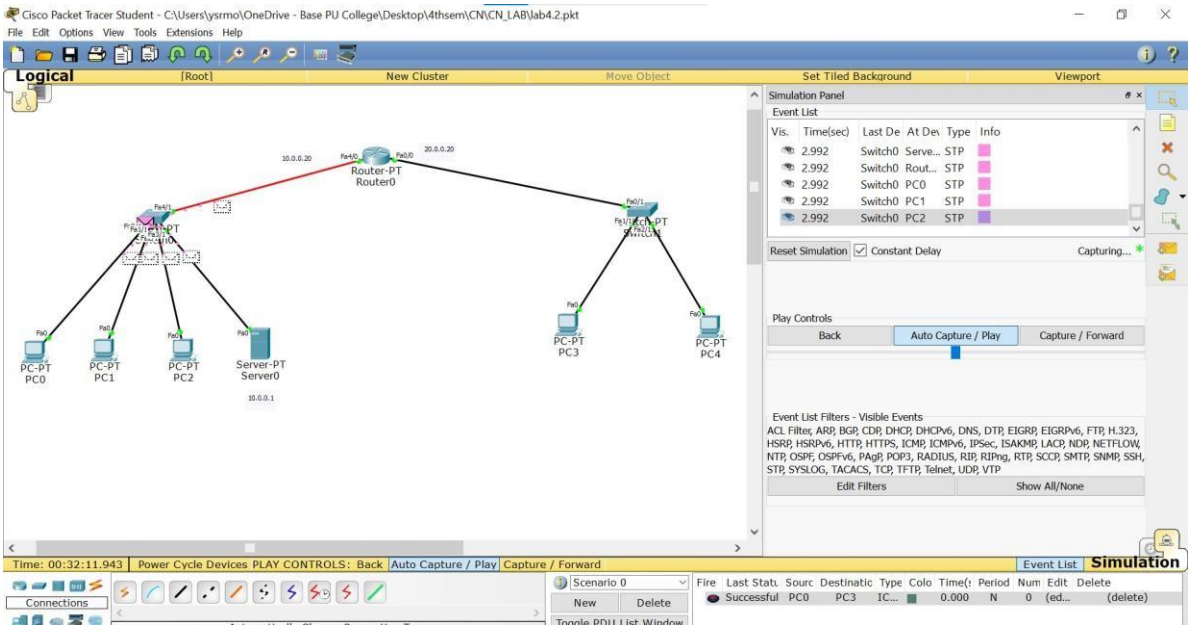
PROGRAM 4.1:





PROGRAM 4.2:

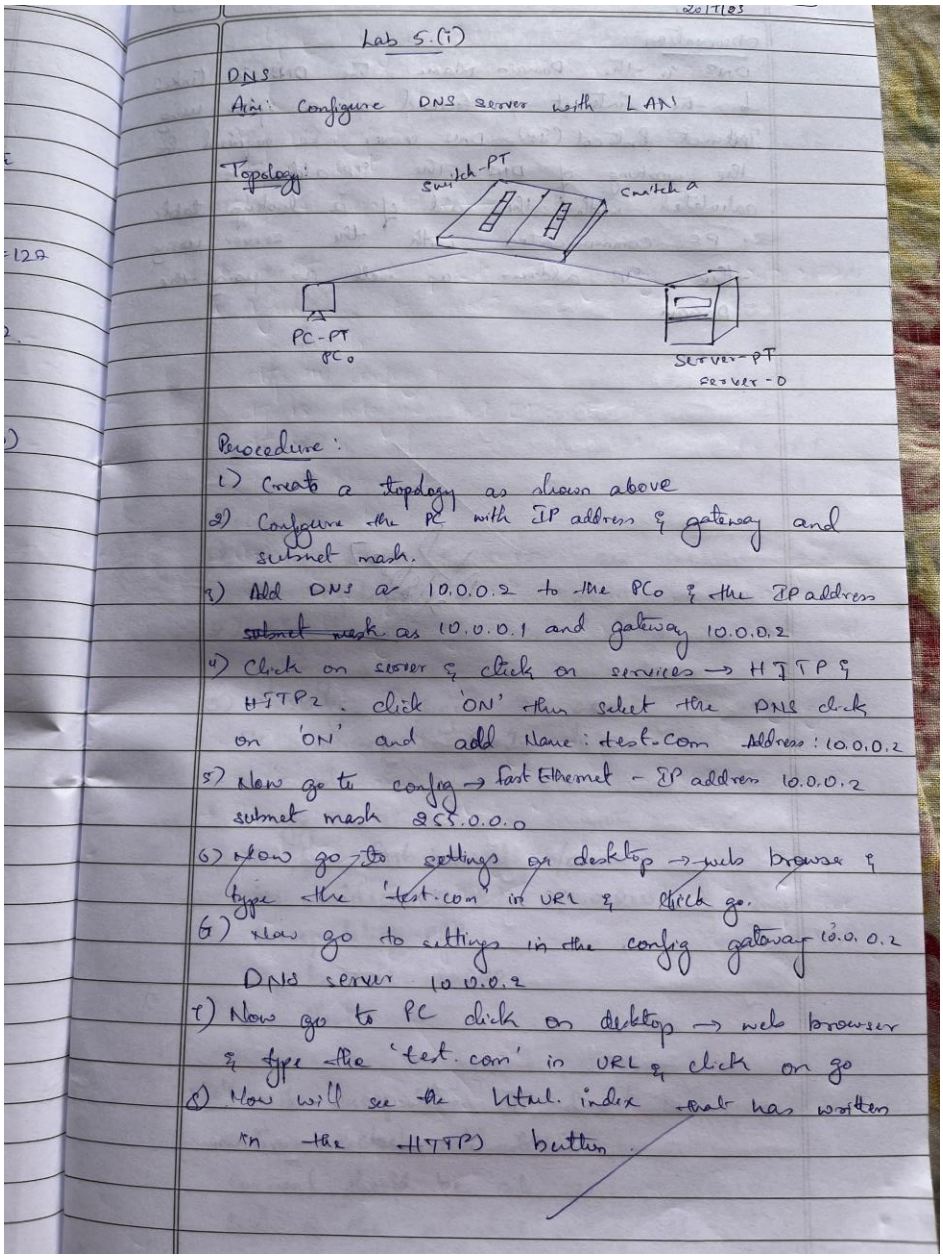

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms 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>


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab4.2.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT Router0
PC-PT PC0
PC-PT PC1
PC-PT PC2
Server-PT Server0
PC-PT PC3
PC-PT PC4
Simulation Panel
Event List
Reset Simulation
Constant Delay
Capturing...
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:32:11.943
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Scenario 0
New
Delete

WEEK5

Configure Web Server, DNS within a LAN.

OBSERVATION:



Lab 5.(i)

DNS

Aim: Configure DNS server with LAN

Topology:

Procedure:

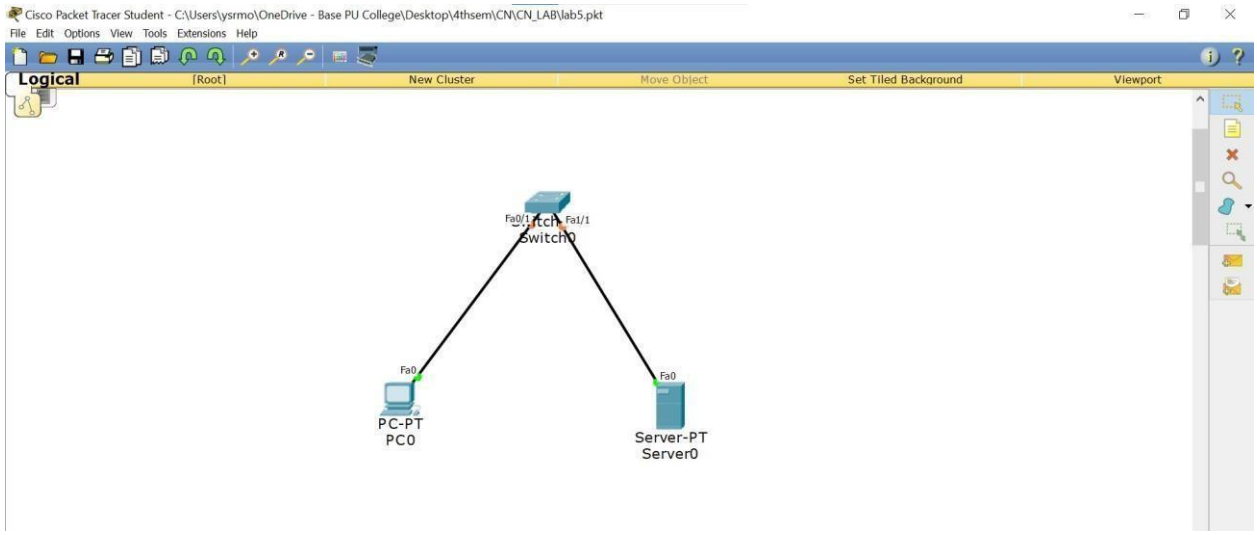
1) Create a topology as shown above
2) Configure the PC with IP address & gateway and subnet mask.
3) Add DNS as 10.0.0.2 to the PC0 & the IP address subnet mask as 10.0.0.1 and gateway 10.0.0.2
4) Click on server & click on services → HTTP & HTTP2. click 'ON' then select the DNS click on 'ON' and add Name: test.com Address: 10.0.0.2
5) Now go to config → Fast Ethernet - IP address 10.0.0.2 subnet mask 255.0.0.0
6) Now go to settings on desktop → web browser & type the 'test.com' in URL & click go.
6) Now go to settings in the config gateway 10.0.0.2 DNS server 10.0.0.2
7) Now go to PC click on desktop → web browser & type the 'test.com' in URL & click on go
8) Now will see the html index that has written in the HTTP) button.

Observation:

DNS is the Domain Name system. DNS is linked to the internet & focused on a system using Internet Protocol (IP). DNS server are required for the 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.

31/8/2022

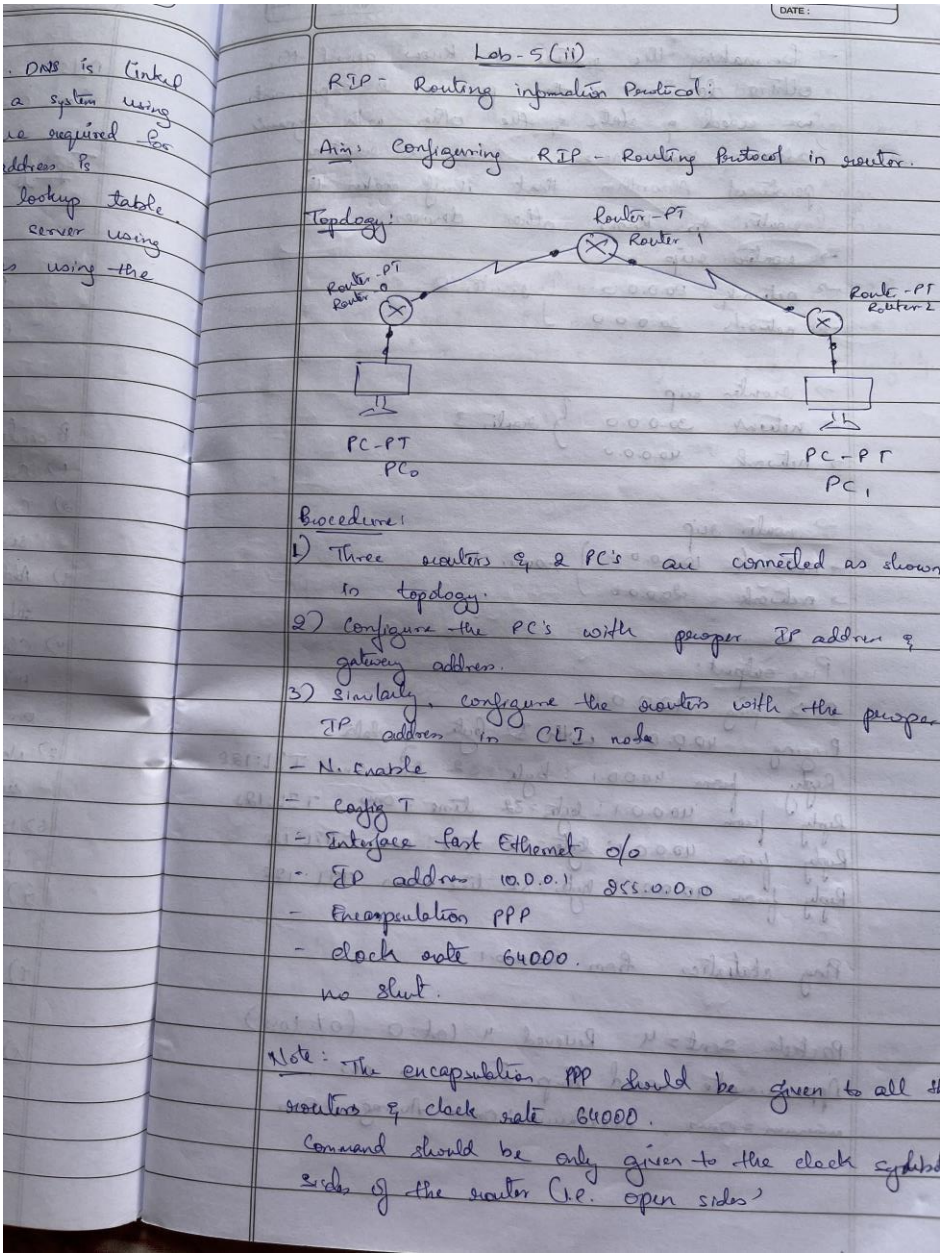
TOPOLOGY:


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab5.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/1
Fa1/1
Switch0
Fa0
PC-PT
PC0
Fa0
Server-PT
Server0

LAB 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



Lab-5(ii)

RIP - Routing information Protocol:

Aim: Configuring RIP - Routing Protocol in router.

Topology:

Procedure:

1) Three routers & 2 PC's are connected as shown in topology.
2) Configure the PC's with proper IP address & gateway address.
3) Similarly, configure the routers with the proper IP address in CLI mode.
- N. Enable
- config T
- Interface fast Ethernet 0/0
- IP address 10.0.0.1 255.0.0.0
- Encapsulation PPP
- clock rate 64000.
no shut.

Note: The encapsulation PPP should be given to all routers & clock rate 64000.
Command should be only given to the clock symbol sides of the router (i.e. open sides)

PAGE NO:
DATE:

→ for making the routers to know about the others devices in the previous 2 experiments we used 1 static & 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
→ network 30.0.0.0
} router 2

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

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

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 for 40.0.0.1

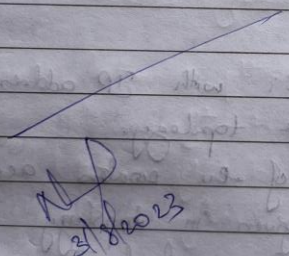
Packets sent = 4 Recieved 4 lost = 0 (0% loss)
Approximate round trip times in ms
minimum = 0ms, maximum : 0ms Average = 0ms

DATE :

Observation:

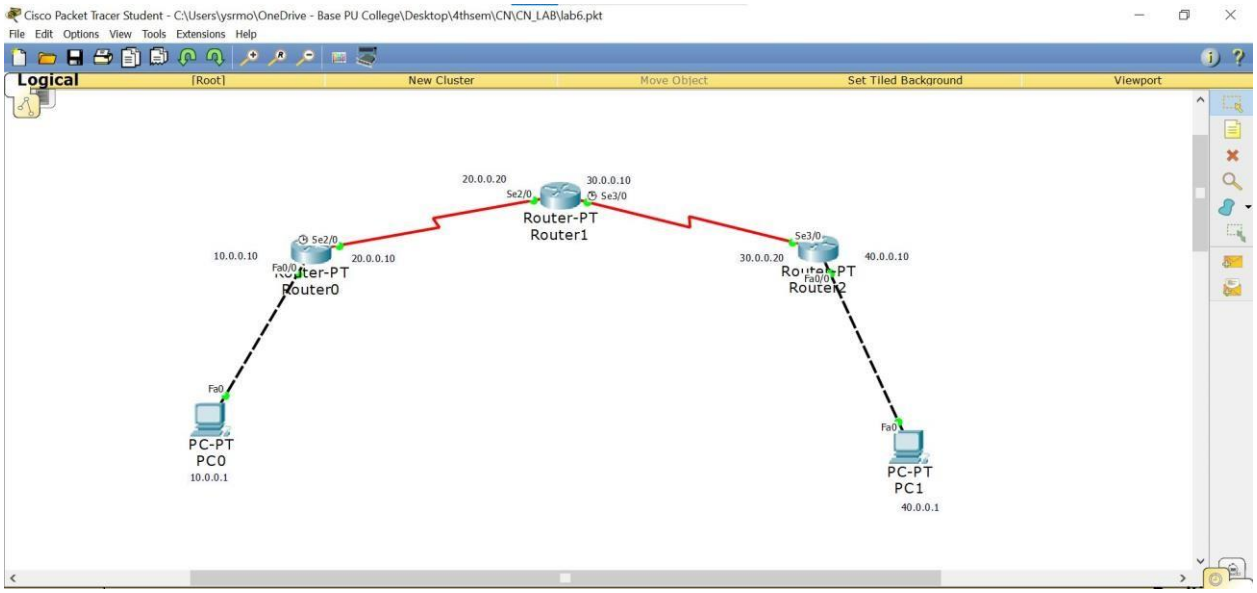
RIP is the Routing Information Protocol is a distance vector protocol that uses hop count as its primary metric. RIP defines how routers should share information when moving traffic among an interconnected group of local area network.

→ The RIP protocol here, used to connected the routers to one other & PC's using RIP protocol & message is pinged successfully.

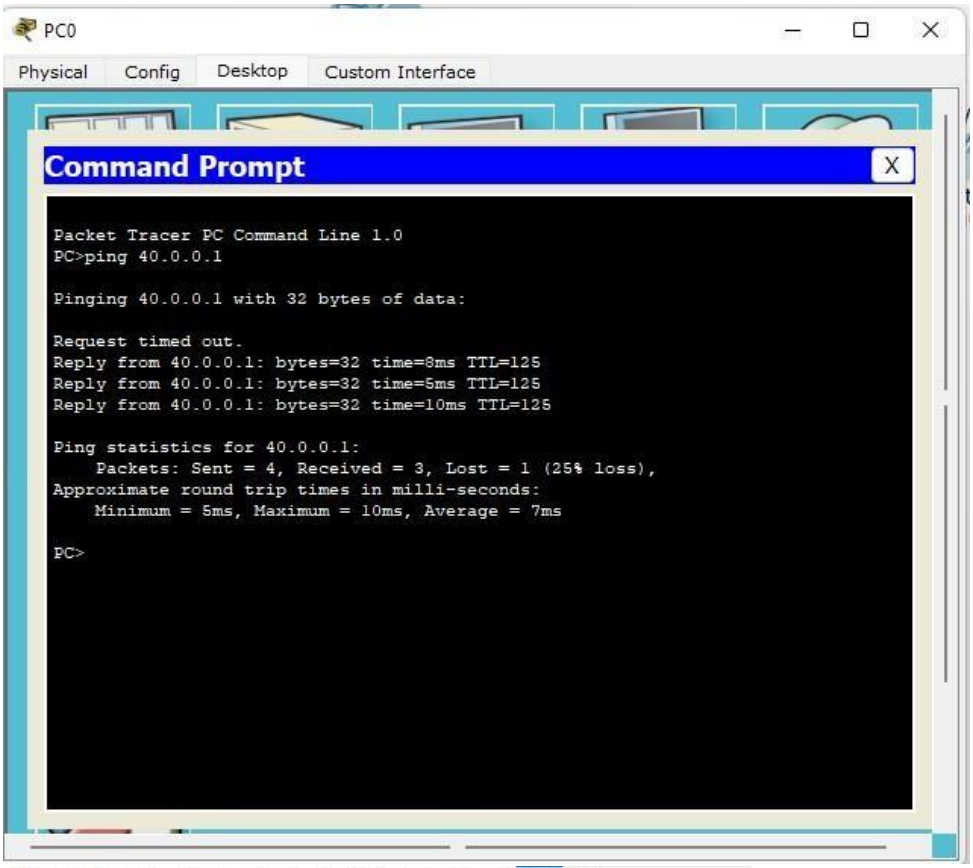


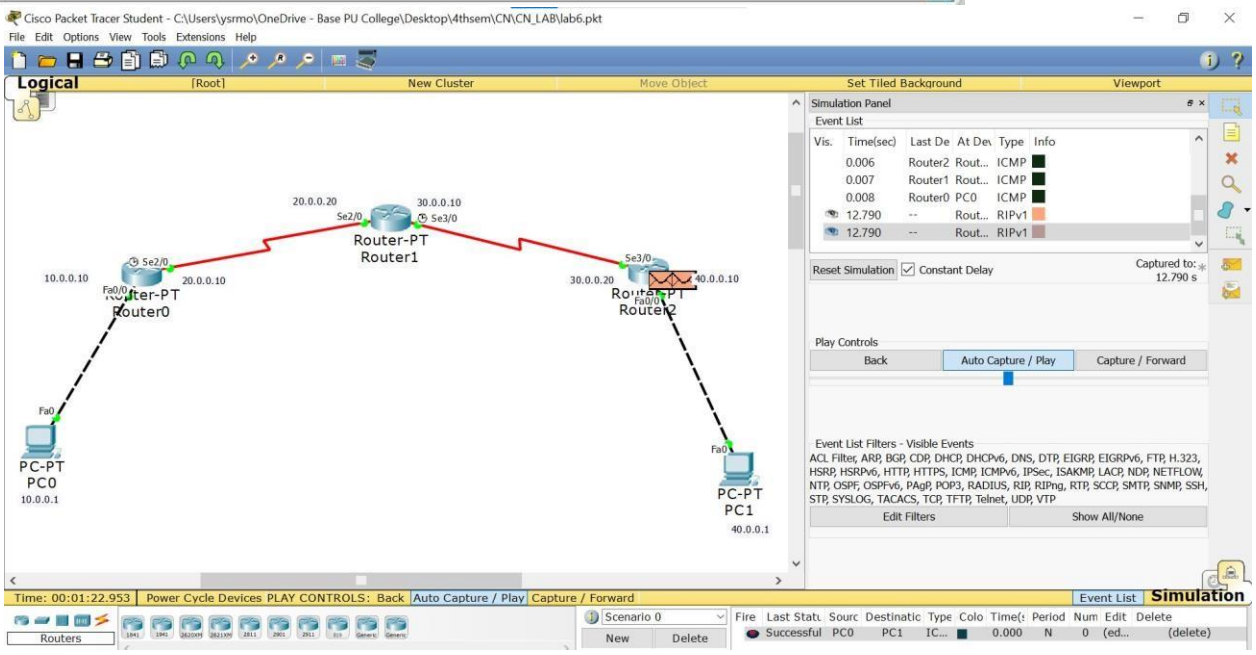
31/8/2023

TOPOLOGY:



OUTPUT:

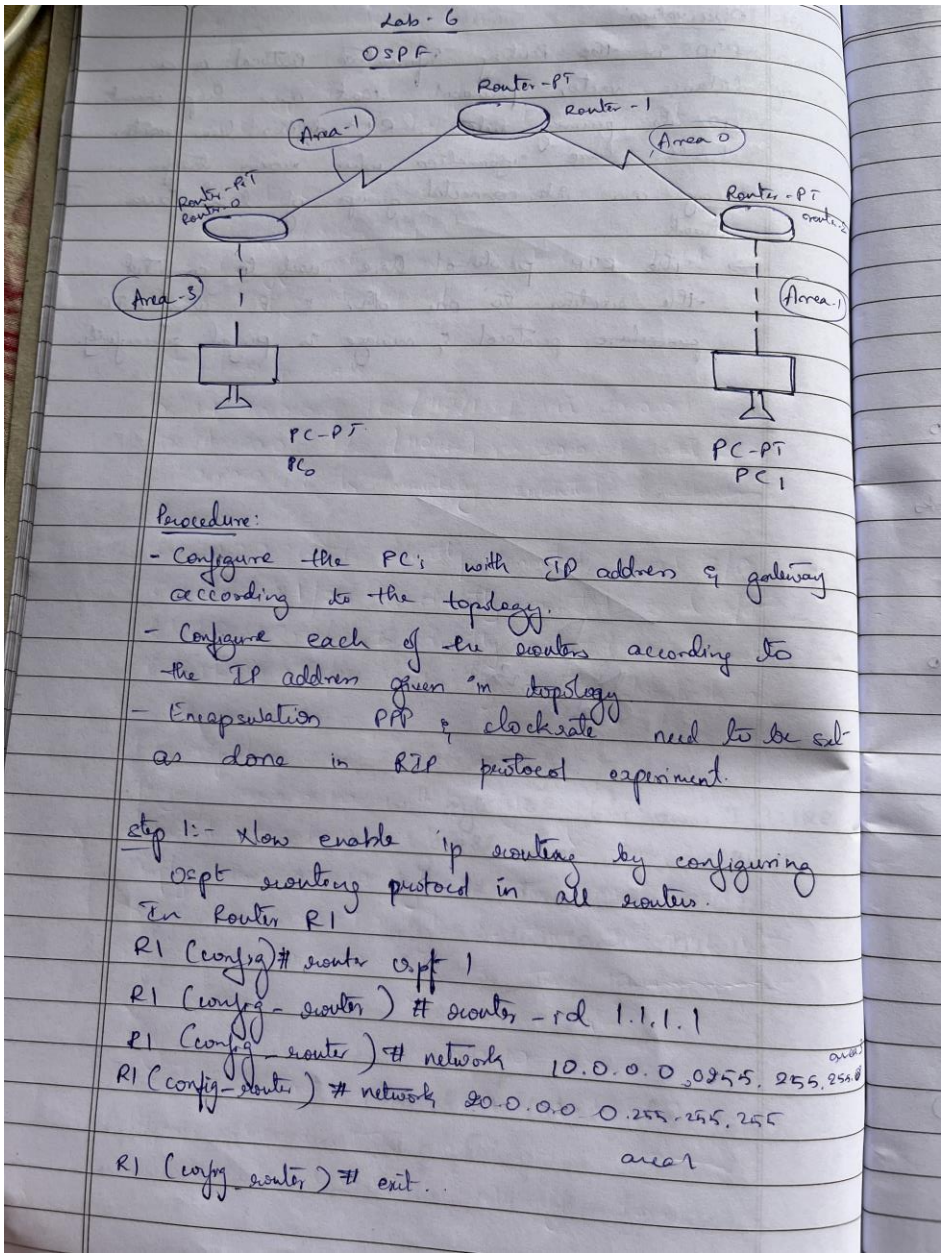

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=5ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 10ms, Average = 7ms
PC>


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
10.0.0.1
PC-PT
PC1
40.0.0.1
20.0.0.20
30.0.0.10
10.0.0.10
20.0.0.10
30.0.0.20
40.0.0.10
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:01:22.953
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Routers
Scenario 0
New
Delete

LAB 7

Configure OSPF routing protocol.

OBSERVATION:



Procedure:

- Configure the PC's with IP address & gateway according to the topology.
- Configure each of the routers according to the IP address given in topology
- Encapsulation PPP & clockrate need to be set as done in RIP protocol experiment.

Step 1:- Now enable ip routing by configuring ospf routing protocol in all routers.

In Router R1

R1 (config)# router ospf 1

R1 (config-router)# router-id 1.1.1.1

R1 (config-router)# network 10.0.0.0 0.255.255.255 area 1

R1 (config-router)# network 20.0.0.0 0.255.255.255 area 1

R1 (config-router)# exit.

In router R2

R2 (config)# router ospf 1

R2 (config-router)# router-id 2.2.2.2

R2 (config-router)# network 20.0.0.0 0.255.255.255 area 0

R2 (config-router)# network 30.0.0.0 0.255.255.255 area 0

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 area 0

R3 (config-router)# network 40.0.0.0 0.255.255.255 area 2

R3 (config-router)# exit

Step 4: loopback in serial interface

In router R1

R1 (config-if)# interface loopback 0

R1 (config-if)# ip address 172.16.1.252 255.255.0.0

R1 (config-if)# no shut

In router R2 in serial interface

R2 (config-if)# interface loopback 0

R2 (config-if)# ip address 172.16.1.253 255.255.0.0

R2 (config-if)# no shut

In router R3

R3 (config-if)# interface loopback 0

R3 (config-if)# ip address 172.16.1.254 255.255.0.0

R3 (config-if)# no shut

31/8/2023

Step 3: virtual link

In router R1

R1 (config)# router ospf 1

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

In router R2:

R2 (config)# router ospf 1

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

R2 (config-router)# exit

→ show ip route

O IA 10.0.0.0 18 [110/129] via 10.0.0.1 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 fast ethernet 0/0

C. 172.16.0.0/16 is directly connected, loopback 0.

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

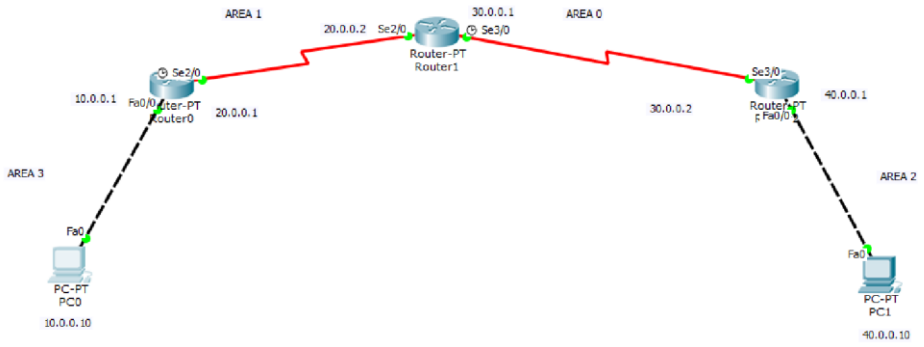
pinging statistics for 40.0.0.10

packets: sent=4, received=3, lost=1 (25% loss)

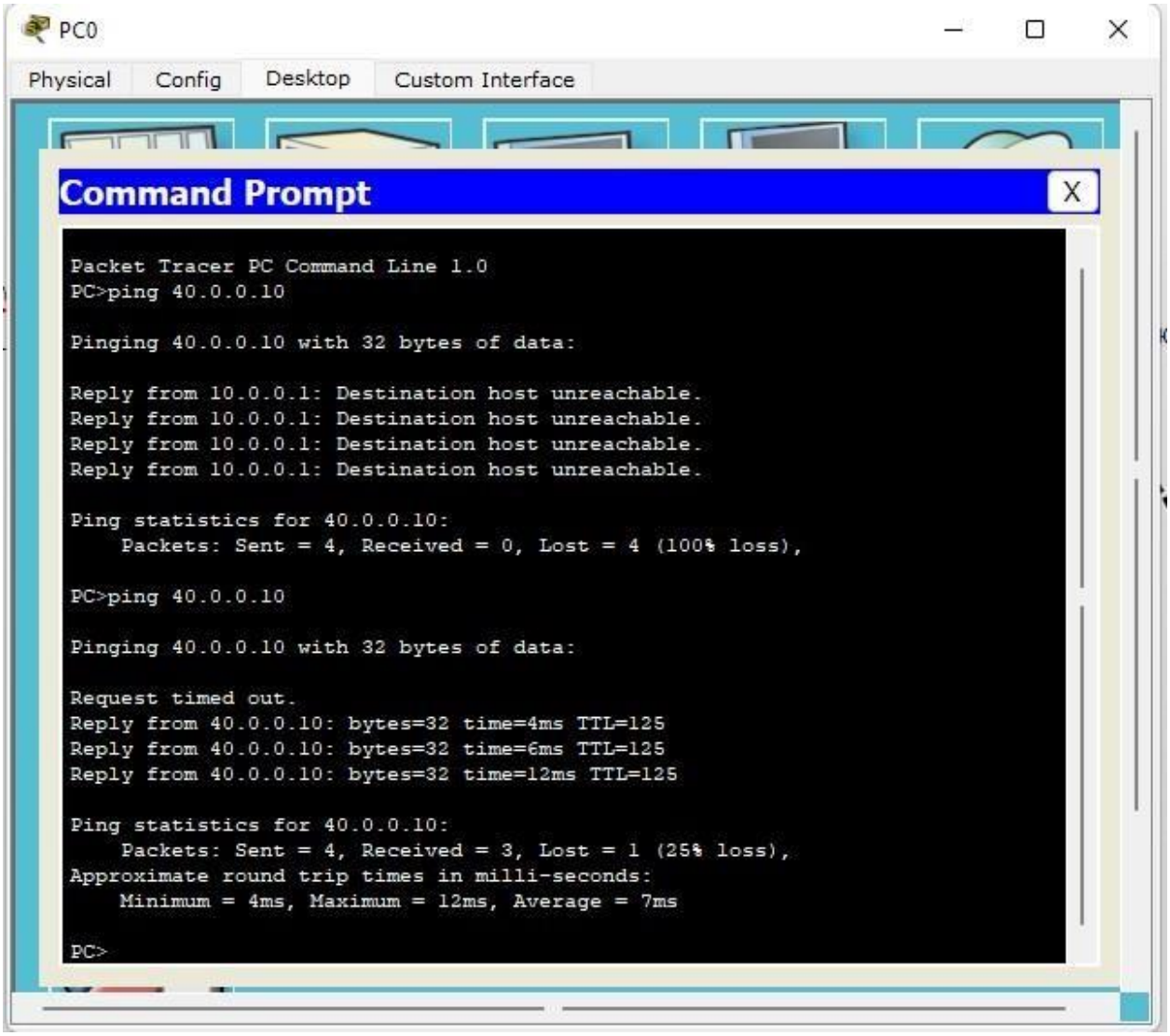
Approx round trip in ms:

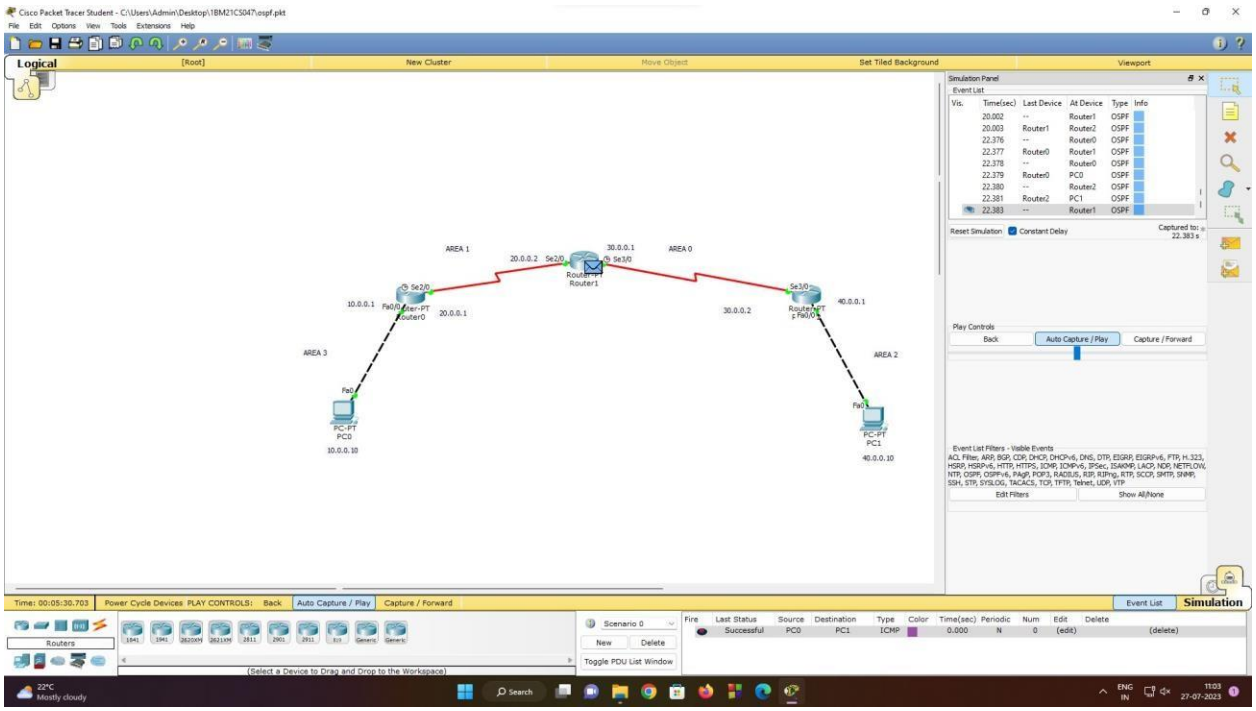
min=2ms, Max=10ms, Average=7ms

TOPOLOGY:



OUTPUT:

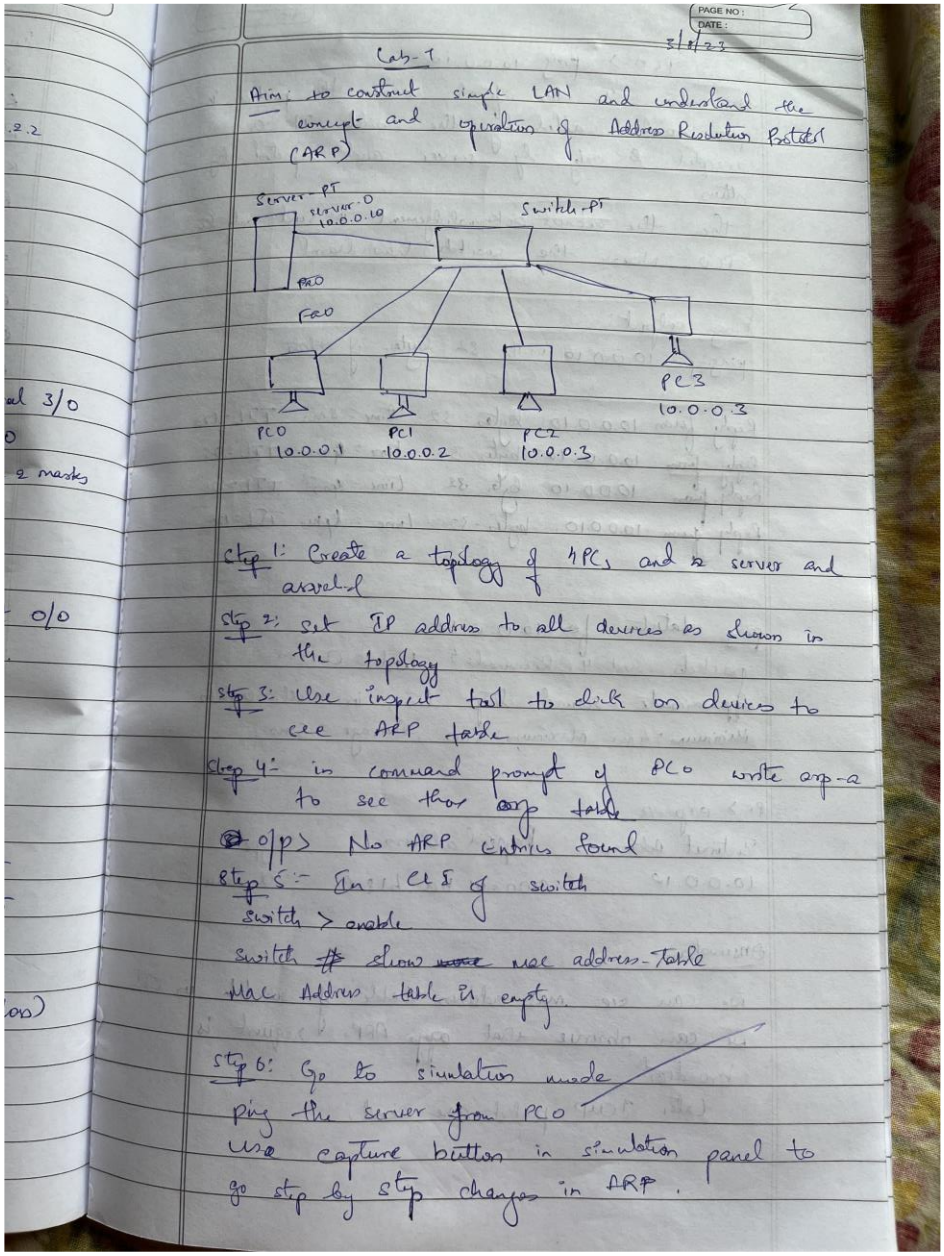


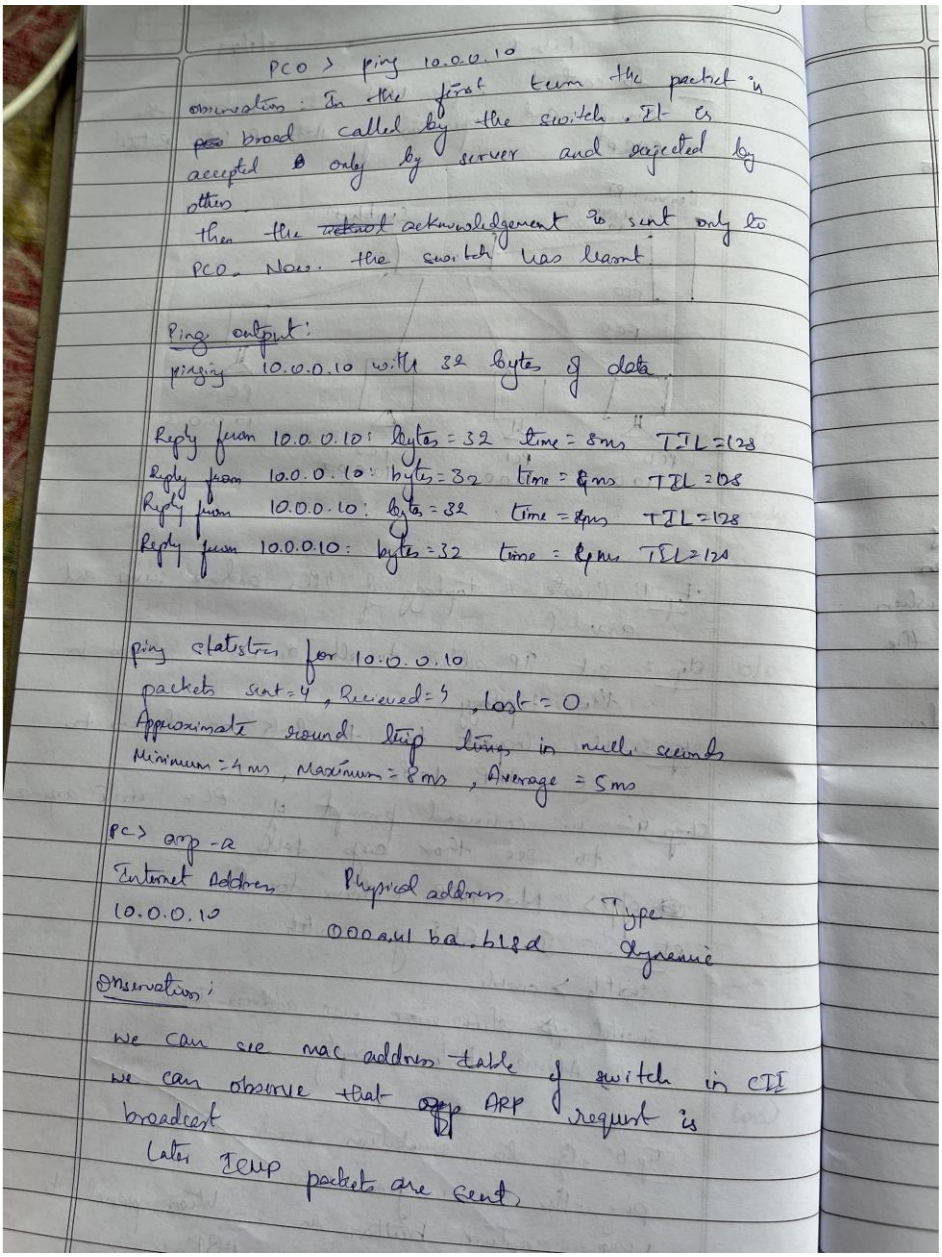

Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 0
AREA 3
AREA 2
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
Router-PT
Router0
10.0.0.1
Fa0/0
20.0.0.1
30.0.0.2
Router-PT
Fa0/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
Fa0
PC-PT
PC1
40.0.0.10
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
20.002 -- Router1 OSPF
20.003 Router1 Router2 OSPF
22.376 -- Router0 OSPF
22.377 Router0 Router1 OSPF
22.378 -- Router0 OSPF
22.379 Router0 PC0 OSPF
22.380 -- Router2 OSPF
22.381 Router2 PC1 OSPF
22.383 -- Router1 OSPF
Reset Simulation
Constant Delay
Captured to: 22.383 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:05:30.703
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Routers
(Select a Device to Drag and Drop to the Workspace)
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
Successful PC0 PC1 ICMP 0.000 N 0 (edit) (delete)
22°C
Mostly cloudy
Search
ENG
IN
11:03
27-07-2023

LAB 8

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

OBSERVATION:





PC0 > ping 10.0.0.10

Observation: In the first turn the packet is broad called by the switch. It is accepted only by server and rejected by others.
then the ~~reboot~~ acknowledgement is sent only to PC0. Now the switch has learnt.

Ping output:
pinging 10.0.0.10 with 32 bytes of data.

Reply from 10.0.0.10: bytes=32 time=8ms TTL=128
Reply from 10.0.0.10: bytes=32 time=6ms TTL=128
Reply from 10.0.0.10: bytes=32 time=8ms TTL=128
Reply from 10.0.0.10: bytes=32 time=6ms TTL=128

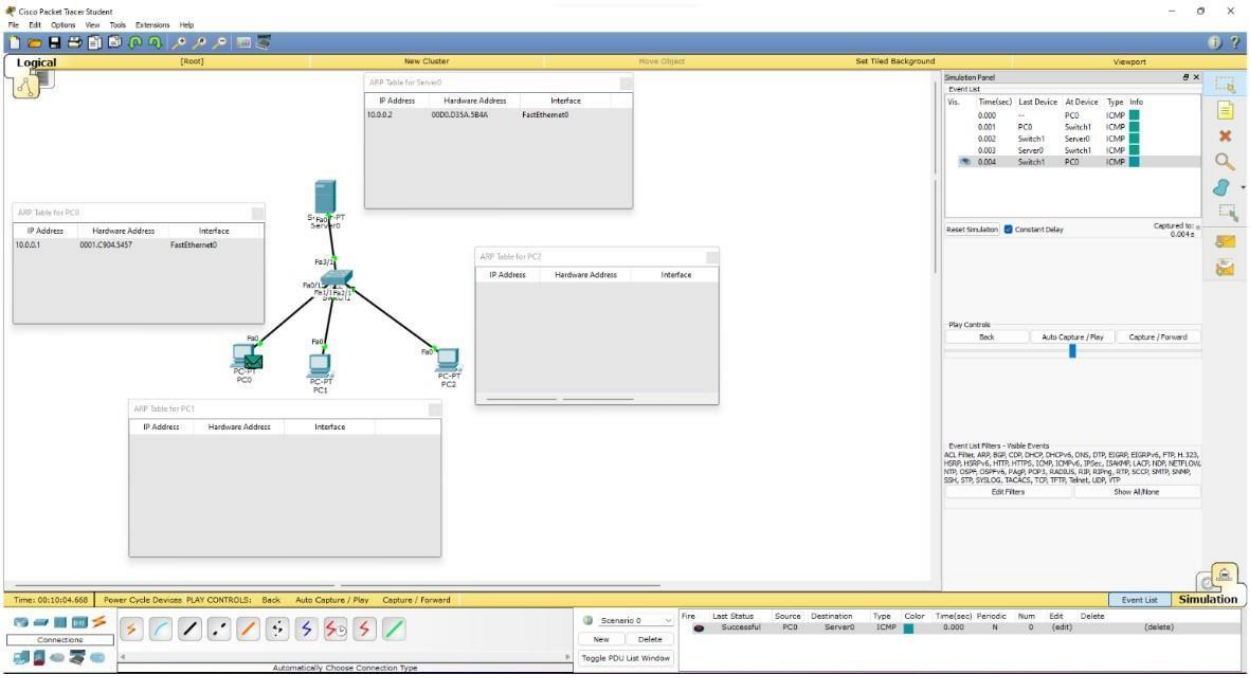
ping statistics for 10.0.0.10
packets sent=4, Received=4, Lost=0.
Approximate round trip time in milli seconds
Minimum=4ms, Maximum=8ms, Average=5ms

PC> arp -a
Internet Address Physical address Type
10.0.0.10 0000.aa1 ba.b1ed dynamic

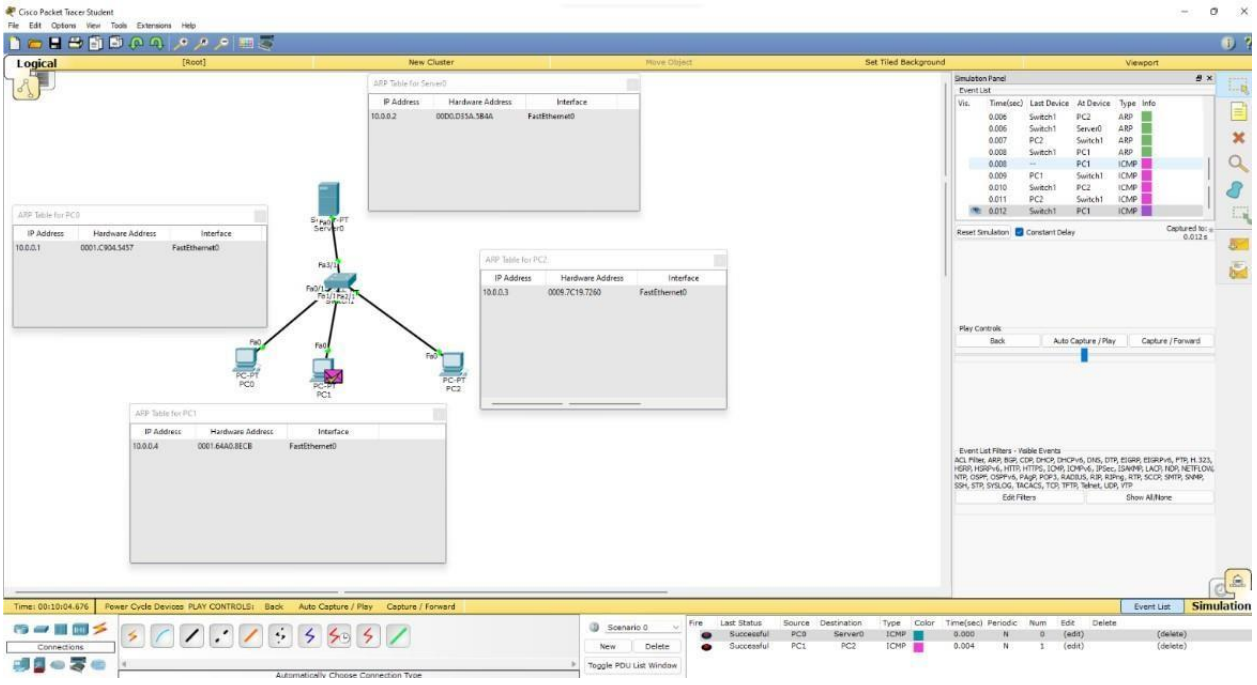
Observation:

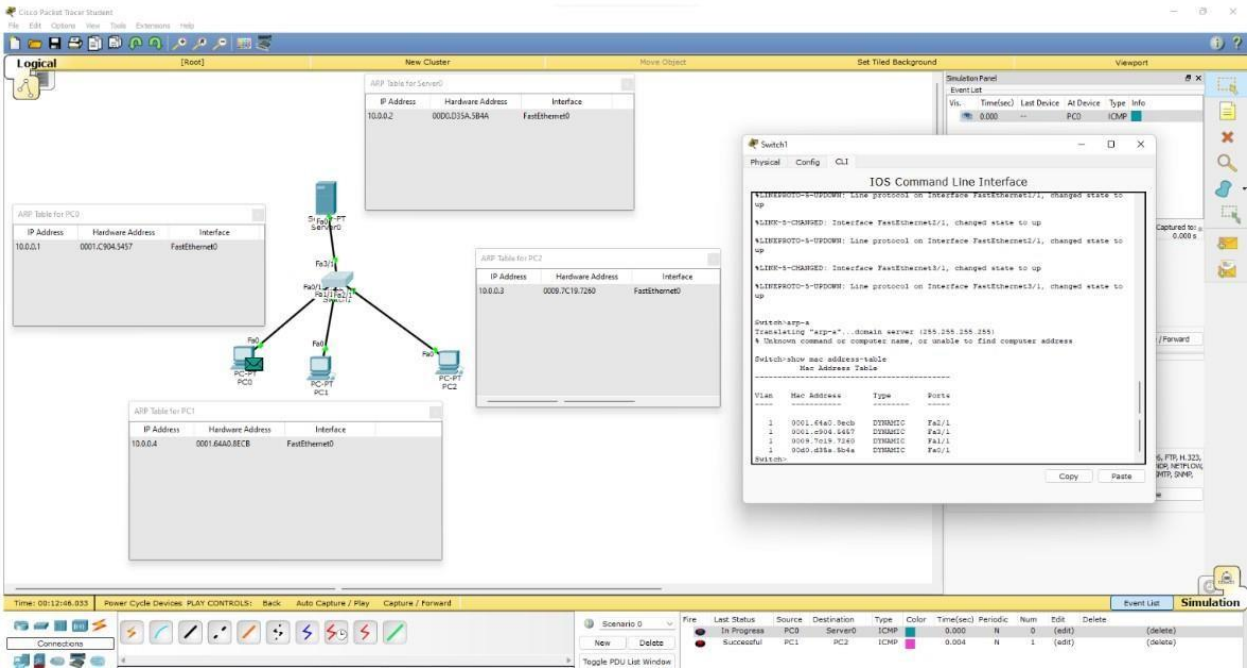
We can see mac address table of switch in CLI
We can observe that ARP request is broadcast
Later ICMP packets are sent.

TOPOLOGY:



OUTPUT:

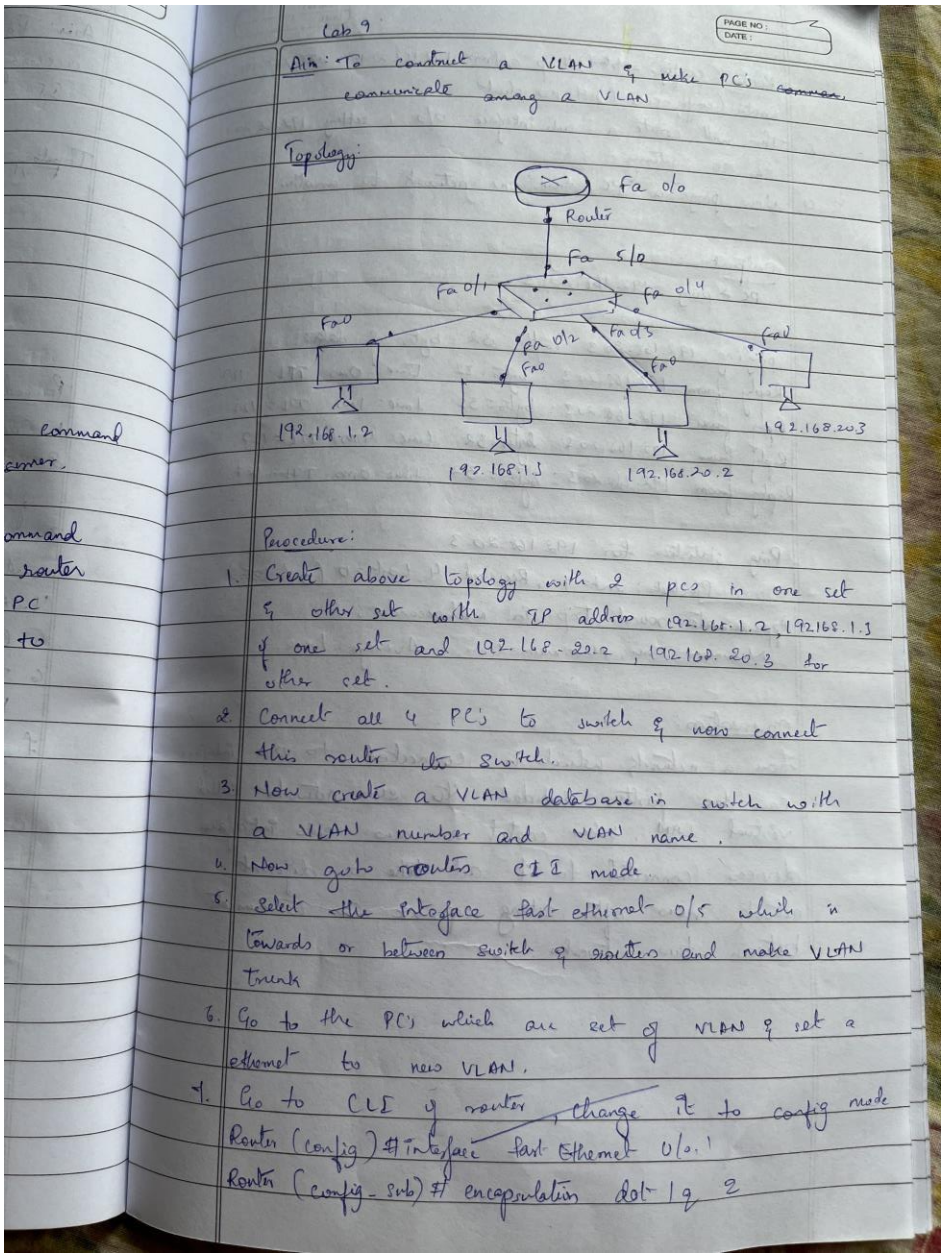



Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
ARP Table for Server0
IP Address Hardware Address Interface
10.0.0.2 00D0.D35A.5B4A FastEthernet0
ARP Table for PC0
IP Address Hardware Address Interface
10.0.0.1 0001.C904.5457 FastEthernet0
ARP Table for PC2
IP Address Hardware Address Interface
10.0.0.3 0009.7C19.7260 FastEthernet0
ARP Table for PC1
IP Address Hardware Address Interface
10.0.0.4 0001.64A0.8ECB FastEthernet0
Switch1
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet1/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up
Switch>arp -a
Translating "arp-a"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0001.64a0.8ecb DYNAMIC Fa2/1
1 0001.c904.5457 DYNAMIC Fa3/1
1 0009.7c19.7260 DYNAMIC Fa1/1
1 00d0.d35a.5b4a DYNAMIC Fa0/1
Switch>
Copy
Paste
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- PC0 ICMP
Time: 00:12:06.033
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress PC0 Server0 ICMP 0.000 N 0 (edit) (delete)
Successful PC1 PC2 ICMP 0.004 N 1 (edit) (delete)

LAB 9

To construct a VLAN and make a pc communicate among VLAN.

OBSERVATION:



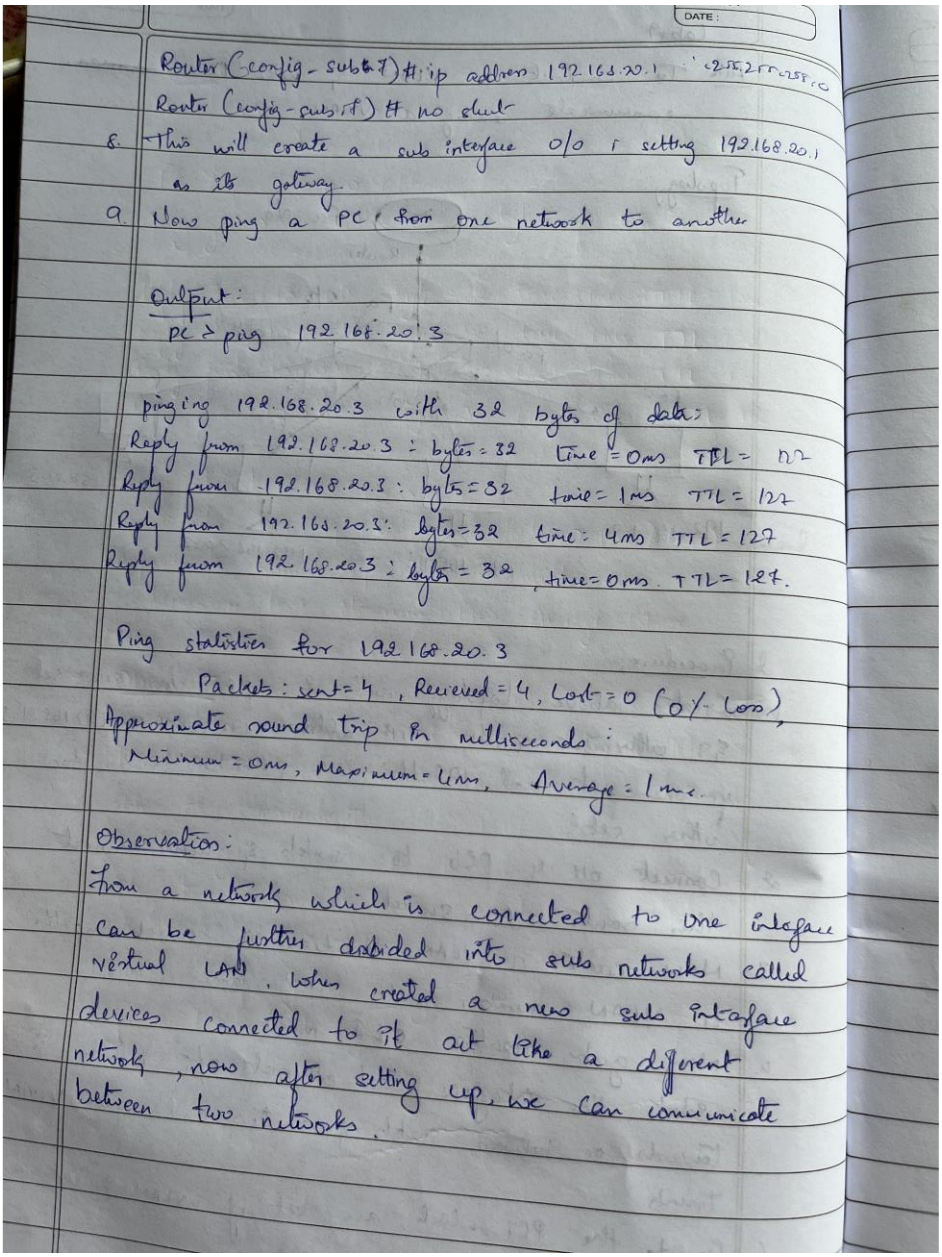
Lab 9

Aim: To construct a VLAN & make PC's communicate among a VLAN

Topology:

Procedure:

1. Create above topology with 2 pcs in one set & other set with IP address 192.168.1.2, 192.168.1.3 of one set and 192.168.20.2, 192.168.20.3 for other set.
2. Connect all 4 PC's to switch & now connect this router to switch.
3. Now create a VLAN database in switch with a VLAN number and VLAN name.
4. Now go to routers CLI mode.
5. Select the interface fast ethernet 0/5 which is towards or between switch & routers and make VLAN trunk
6. Go to the PC's which are set of VLAN & set a ethernet to new VLAN.
7. Go to CLI of router, change it to config mode
   Router (config) # interface fast Ethernet 0/0.1
   Router (config-sub) # encapsulation dot1q 2



Router (config-subif) # ip address 192.168.20.1 255.255.255.0
Router (config-subif) # no shut
8. This will create a sub interface 0/0.1 setting 192.168.20.1 as its gateway.
9. Now ping a PC from one network to another

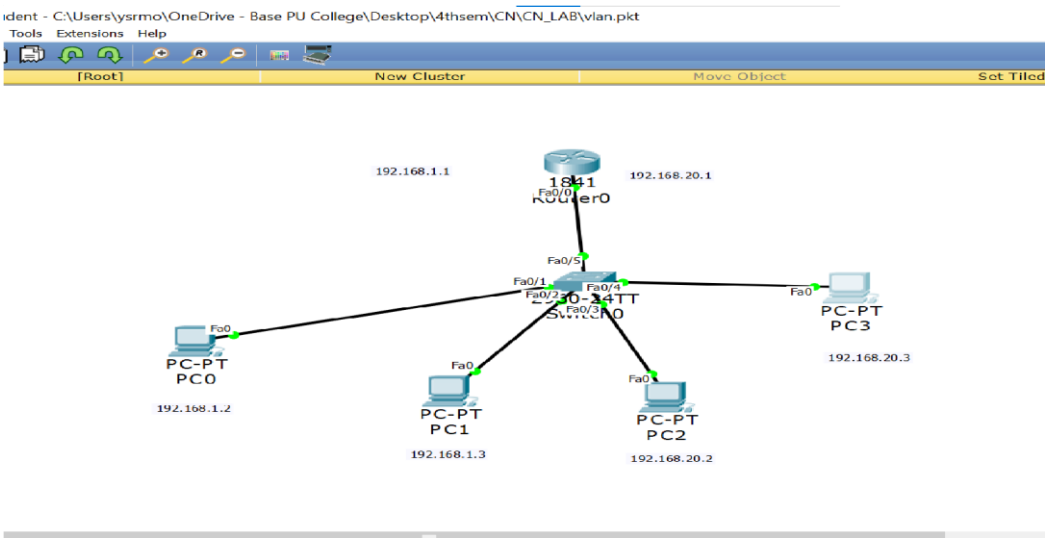
Output:
PC > ping 192.168.20.3

pinging 192.168.20.3 with 32 bytes of data:
Reply from 192.168.20.3 : bytes = 32 time = 0ms TTL = 127
Reply from 192.168.20.3 : bytes = 32 time = 1ms TTL = 127
Reply from 192.168.20.3 : bytes = 32 time = 4ms TTL = 127
Reply from 192.168.20.3 : bytes = 32 time = 0ms TTL = 127.

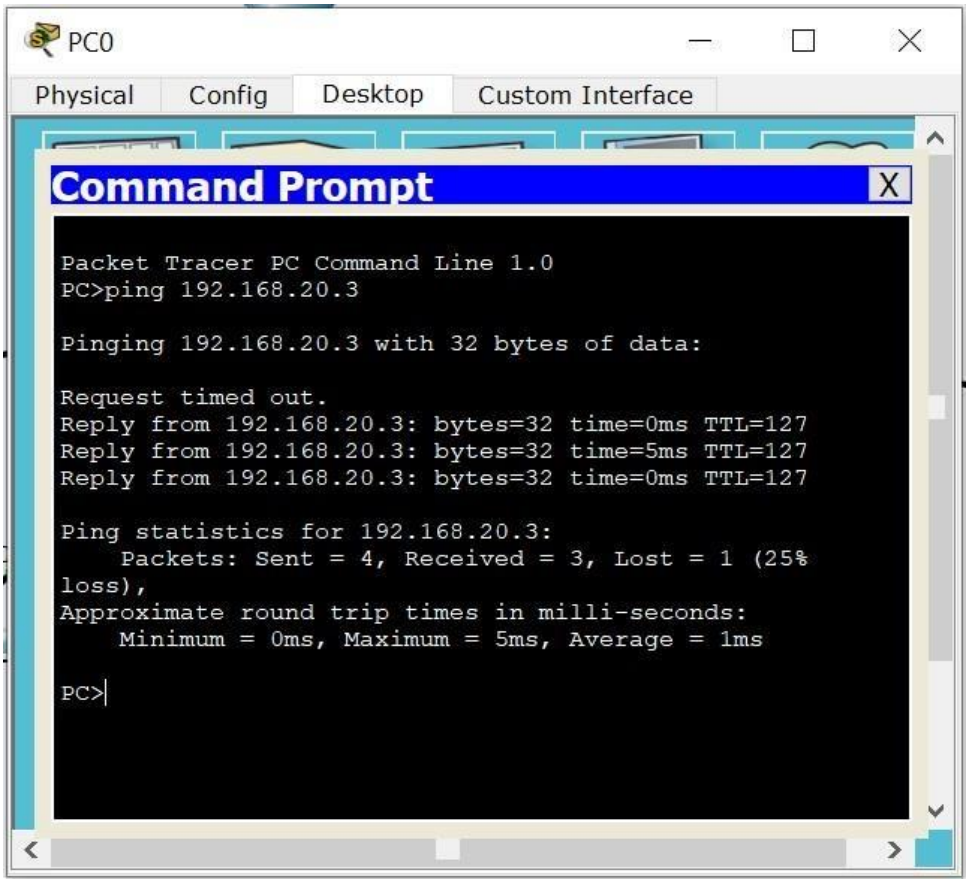
Ping statistics for 192.168.20.3
Packets : sent = 4, Received = 4, Lost = 0 (0% Loss),
Approximate round trip in milliseconds :
Minimum = 0ms, Maximum = 4ms, Average = 1ms.

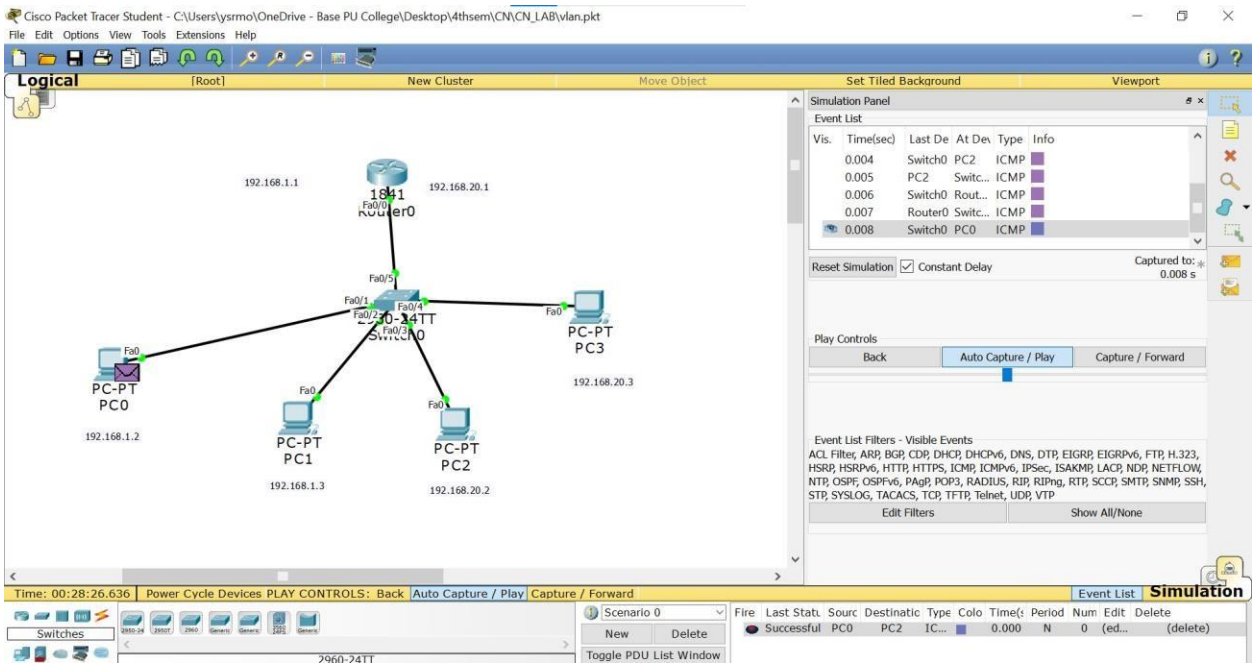
Observation:
From a network which is connected to one interface can be further divided into sub networks called virtual LAN. When created a new sub interface devices connected to it act like a different network, now after setting up, we can communicate between two networks.

TOPOLOGY:


ident - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
Tools Extensions Help
[Root]
New Cluster
Move Object
Set Tiled
192.168.1.1
1841
Router0
Fa0/0
192.168.20.1
Fa0/5
Fa0/1
Fa0/2
Fa0/4
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

OUTPUT:

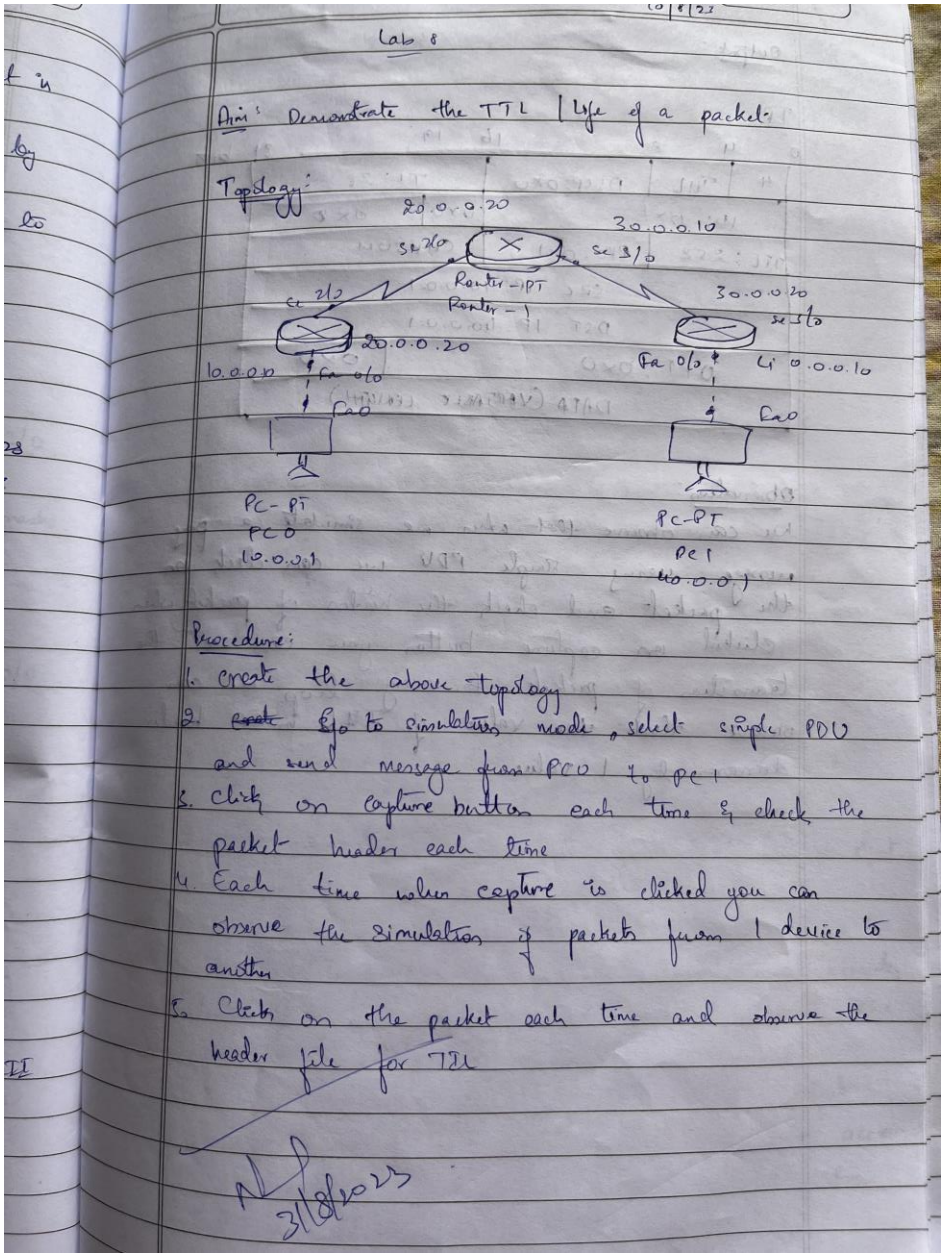

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25%
loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms
PC>


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
192.168.1.1
1841
Router0
192.168.20.1
Fa0/5
Fa0/1
Fa0/2
Fa0/3
Fa0/4
2960-24TT
Switch0
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
2960-24TT
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Stat Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC2 IC... 0.000 N 0 (ed... (delete)

LAB 10

Demonstrate the TTL/ Life of a Packet.

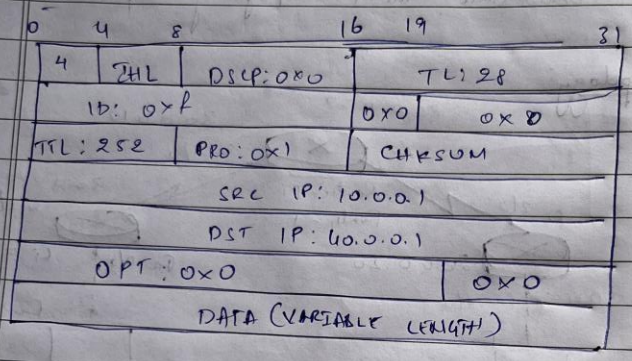
OBSERVATION:



PAGE NO :
DATE :

Output :

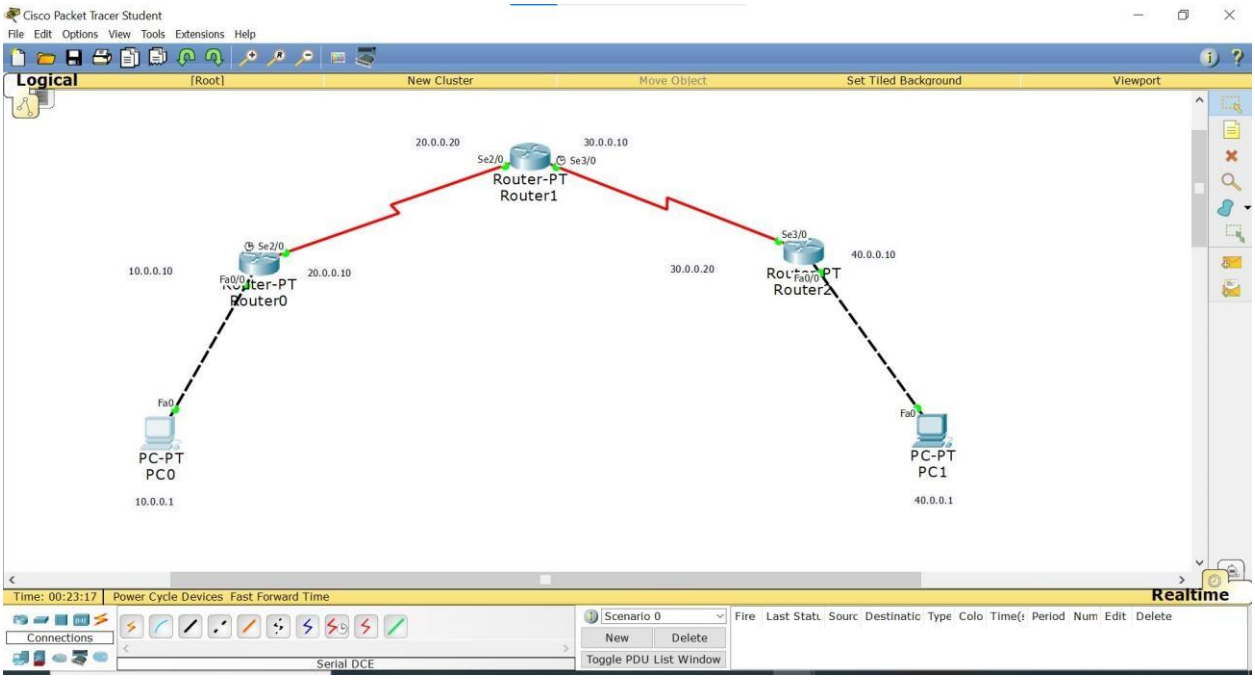
IP



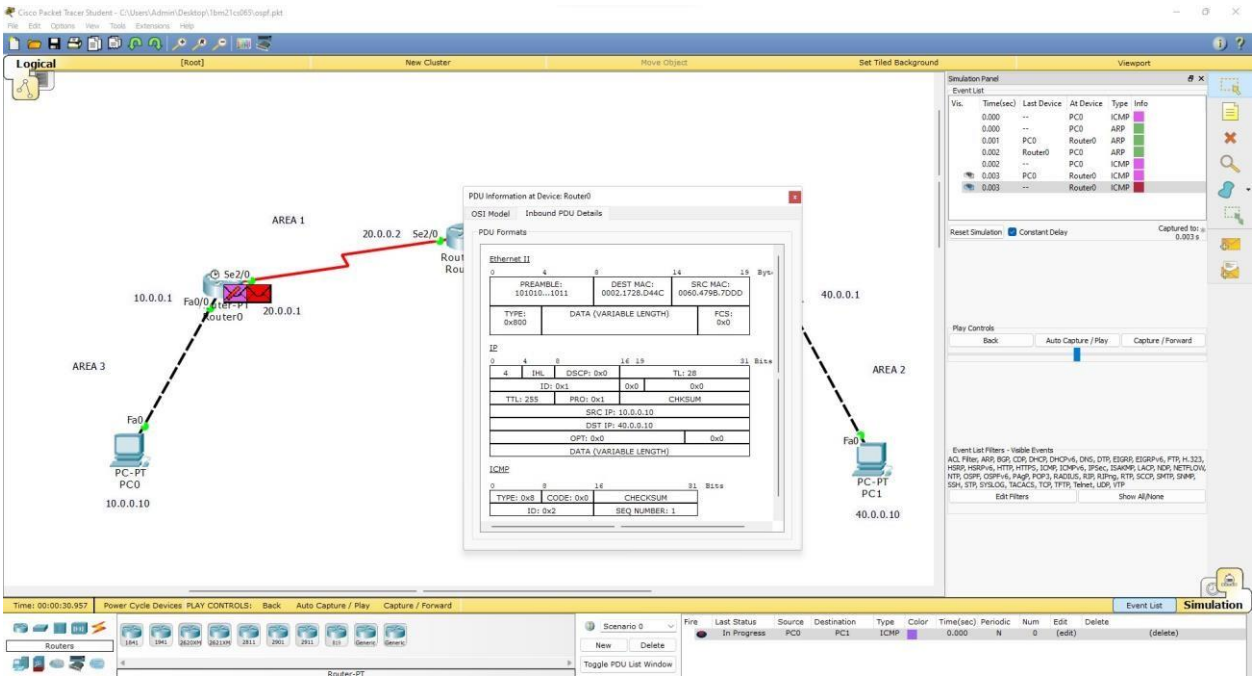
Observation :

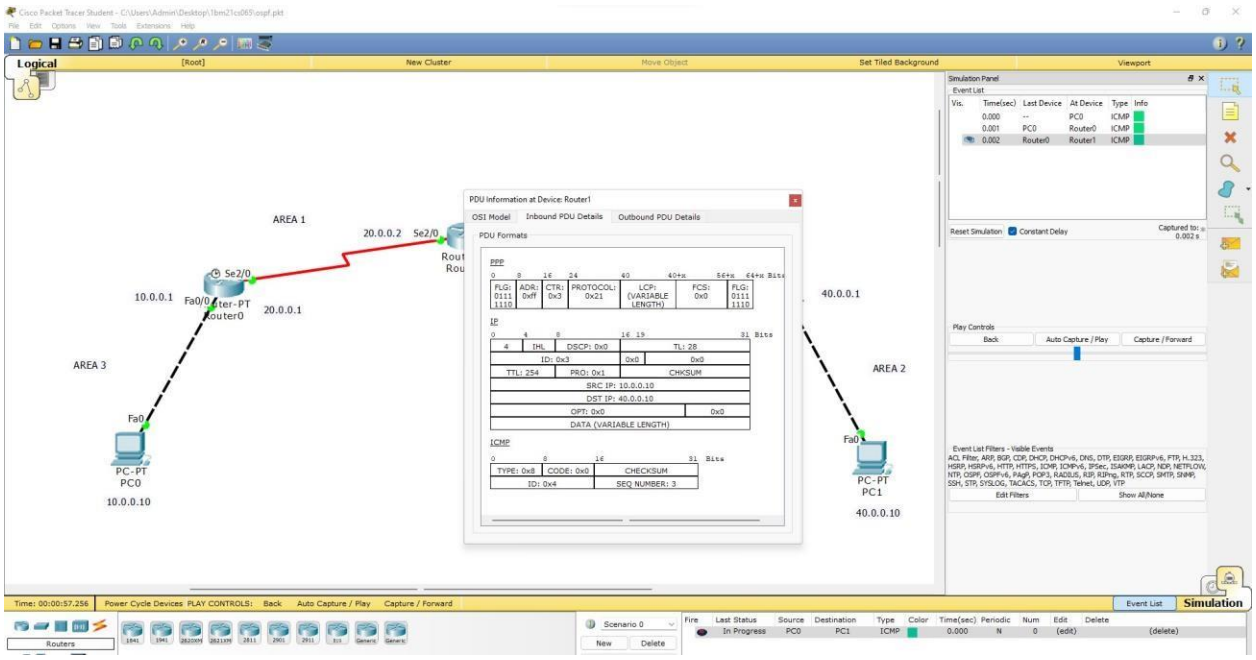
We can observe that when we simulate a ping message using simple PDU we can click on the packet and check the headers of packet when clicked on capture button you can see the transition of packets. For every hop we can observe that the value of TTL in IP header decreases by 1 value.

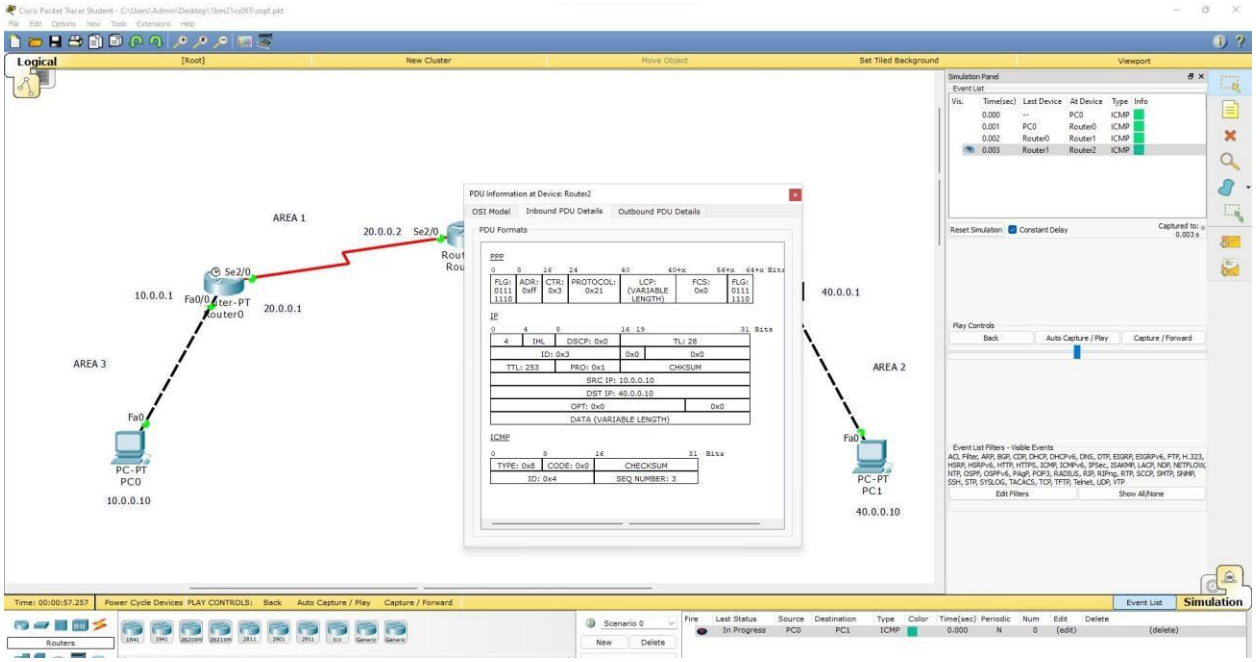
TOPOLOGY:

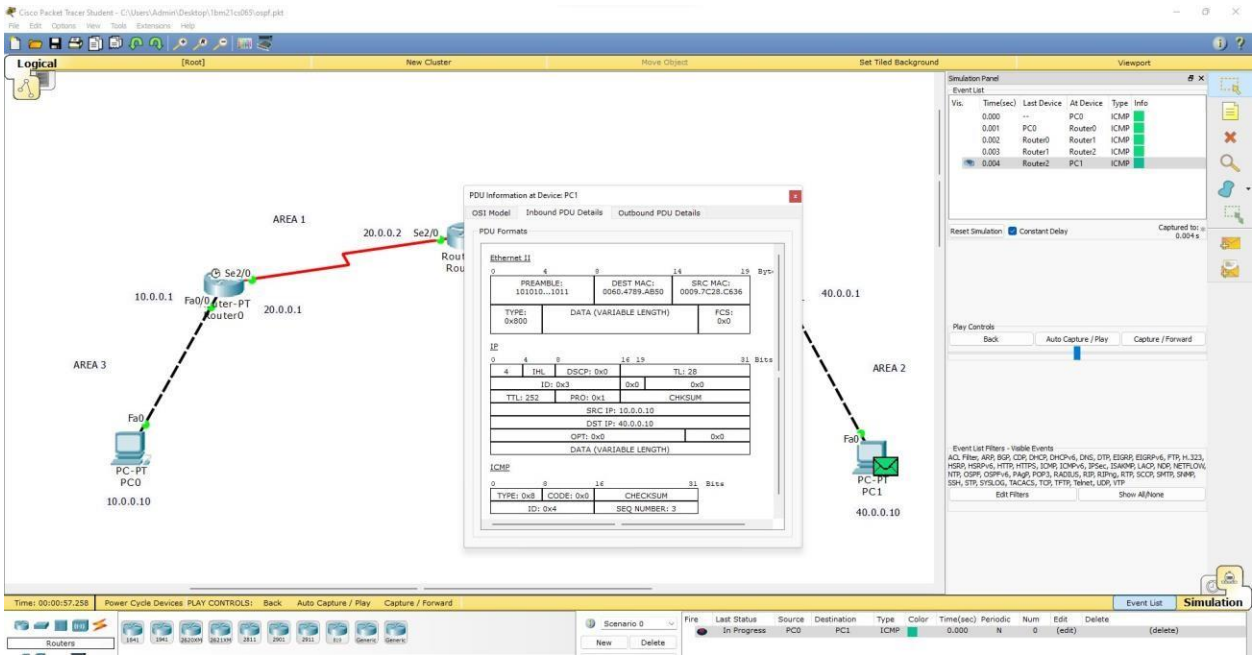


OUTPUT:




Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
40.0.0.1
AREA 3
AREA 2
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
PPP
IP
ICMP
TTL: 254
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
DATA (VARIABLE LENGTH)
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Simulation

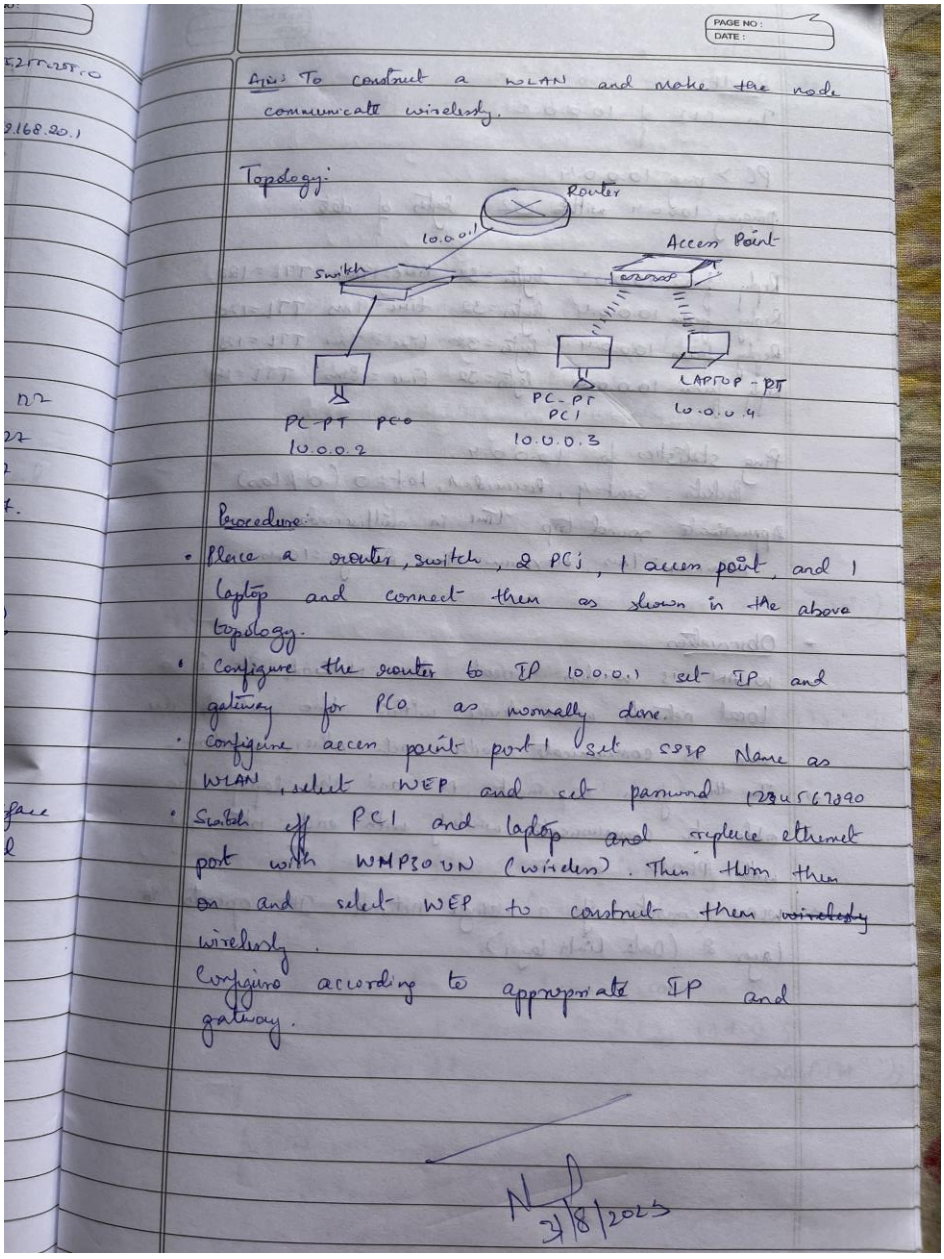

Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
40.0.0.1
AREA 3
AREA 2
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
TTL: 253
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
DATA (VARIABLE LENGTH)
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Simulation


Logical
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0060.4789.AB50
SRC MAC: 0009.7C28.C636
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
OPT: 0x0
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:00:57.258
Power Cycle Devices
PLAY CONTROLS:
Routers
Scenario 0
New
Delete
In Progress
ICMP
Simulation

LAB 11

To construct a WLAN and make the nodes communicate wirelessly

OBSERVATION:



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

Topology:

Procedure:

- Place a router, switch, 2 PCs, 1 access point, and 1 laptop and connect them as shown in the above topology.
- Configure the router to IP 10.0.0.1 set IP and gateway for PC0 as normally done.
- Configure access point port 1 set SSID Name as WLAN, select WEP and set password 1234567890
- Switch off PC1 and laptop and replace ethernet port with WMP300N (wireless). Then turn them on and select WEP to construct them ~~wirelessly~~ wirelessly.
- Configure according to appropriate IP and gateway.

3/8/2023

PAGE NO :
DATE :

Result:

In CLI of 10.0.0.2

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 = 19ms TTL = 128
Reply from 10.0.0.4 bytes = 32 time = 9 ms TTL = 128
Reply from 10.0.0.4 bytes = 32 time = 7 ms TTL = 128
Reply from 10.0.0.4 bytes = 32 time = 8 ms TTL = 128

Ping statistics for 10.0.0.4
Packets: sent = 4, Received = 4, Lost = 0 (0 % loss)
Approximate round-trip time in milliseconds
Minimum = 7ms, Maximum = 19ms, Average = 10 ms.

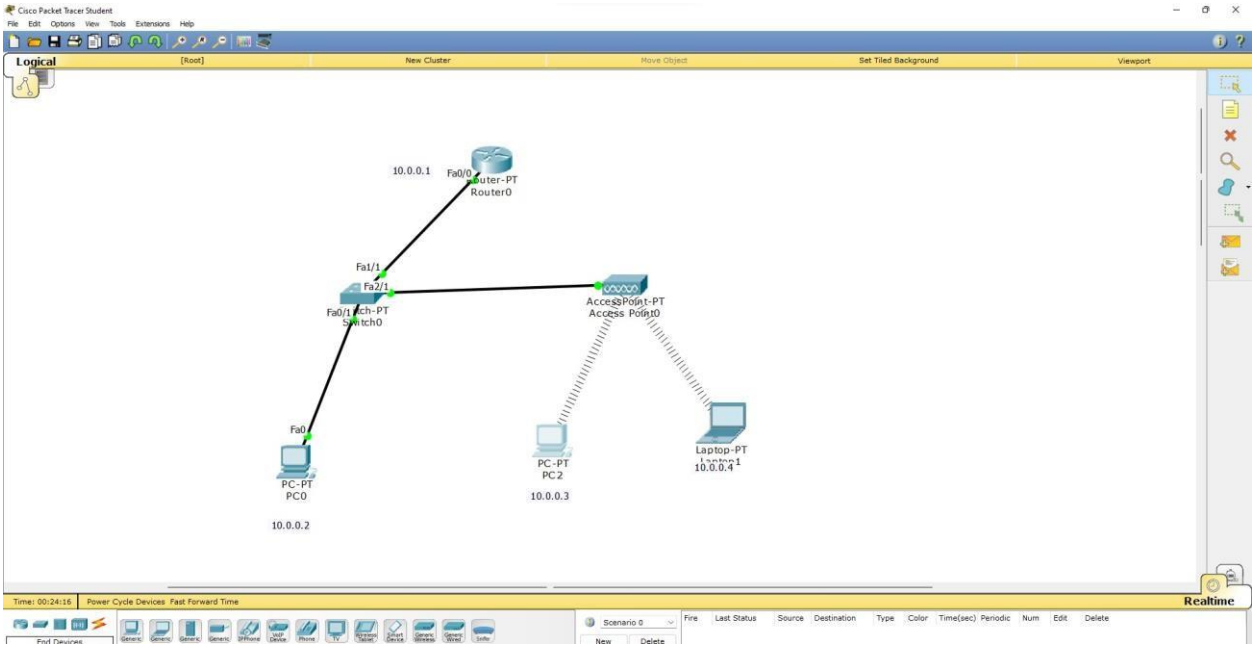
- Observation:

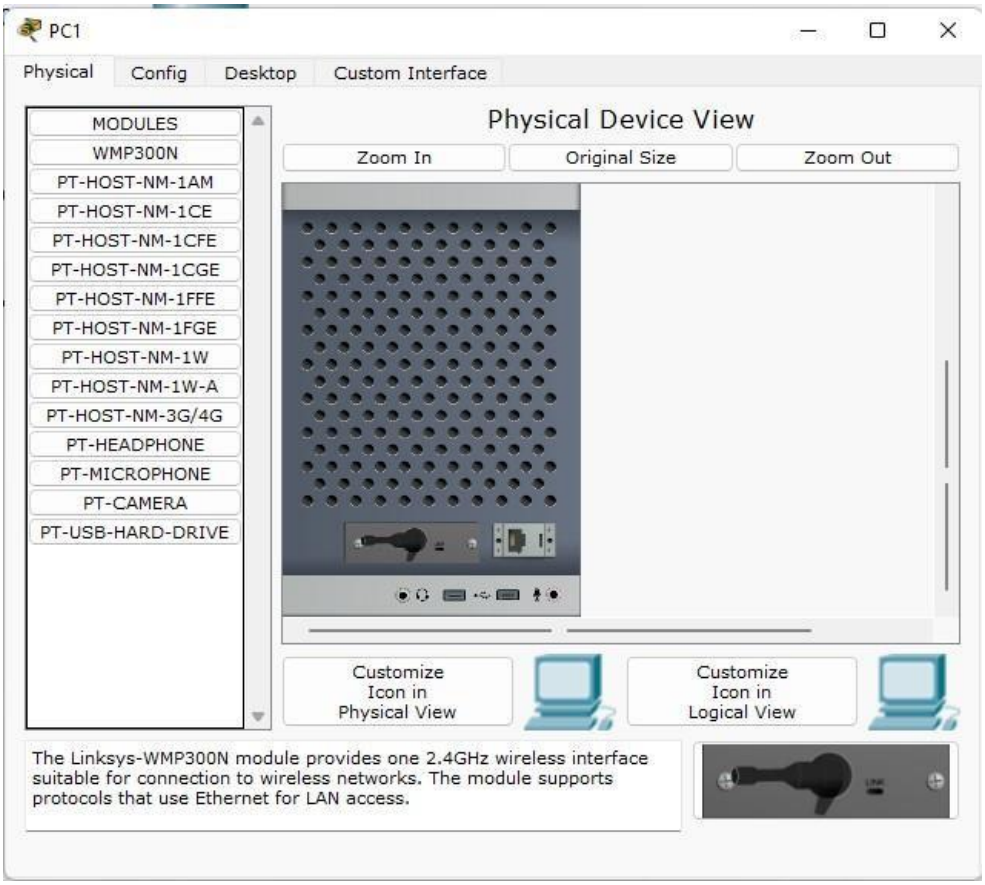
→ WLAN is wireless Local Area Network. It is a local network where device within the network are able to communicate with each other wirelessly.

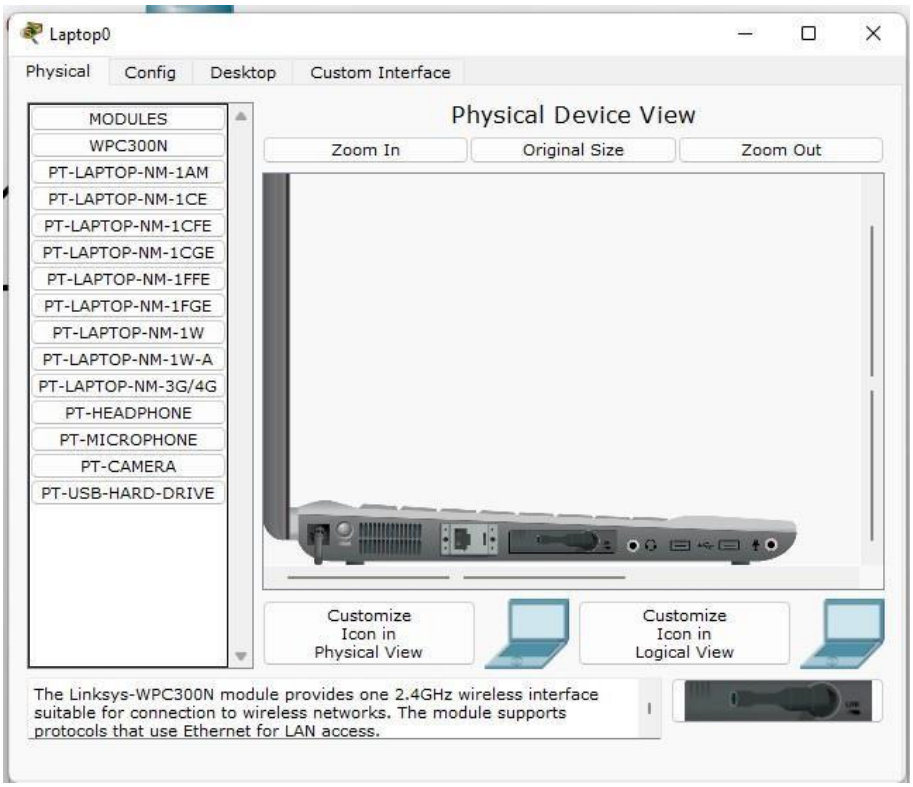
→ In the given experiment, PC1 and the laptop are able to communicate wirelessly with each other and with PC0.

→ WLAN consists of a single network. It operator in Layer 2 (Data Link layer).

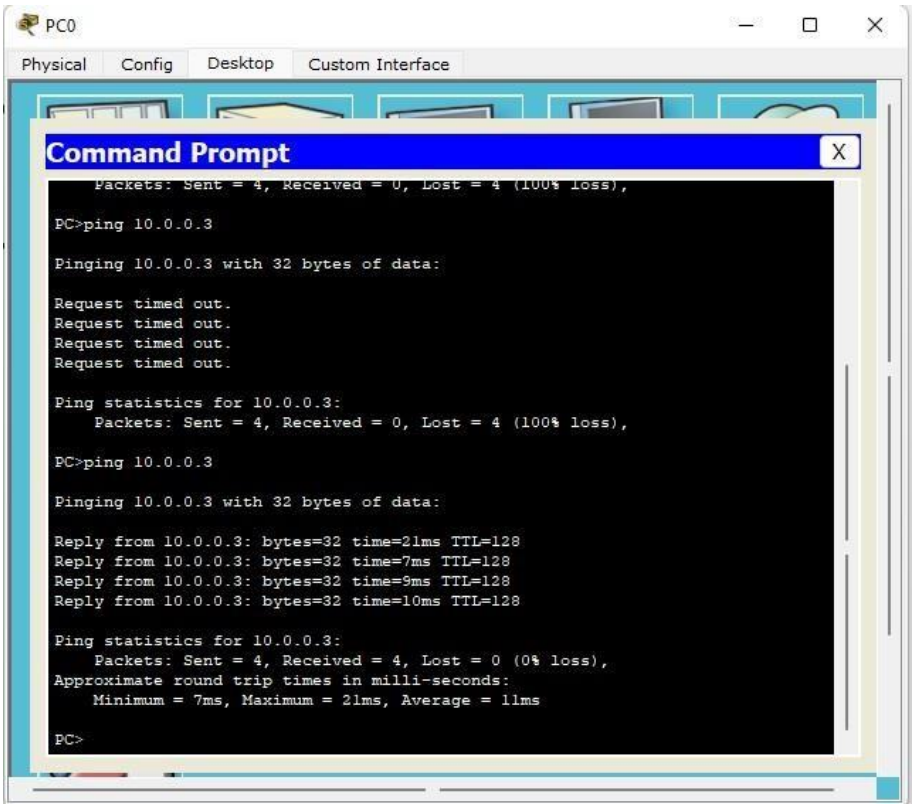
TOPOLOGY:


Logical
10.0.0.1
Fa0/0
Router-PT
Router0
Fa1/1
Fa2/1
Fa0/1
Switch0
Access Point-PT
Access Point0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC2
10.0.0.3
Laptop-PT
10.0.0.4
Realtime


PC1
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-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-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.


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.

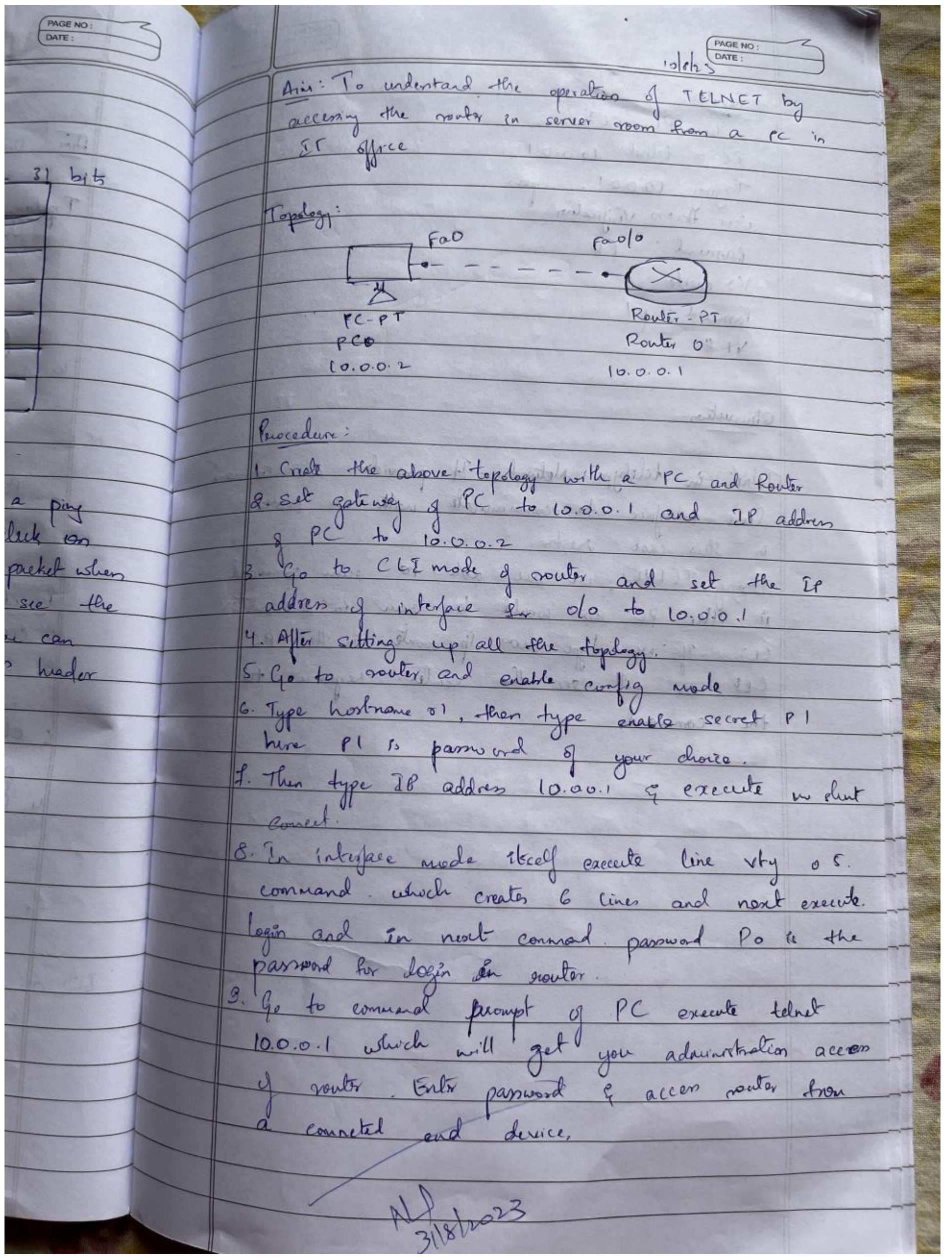
OUTPUT:


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>

LAB 12

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

OBSERVATION:



PAGE NO :
DATE : 10/8/23

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

Topology:

Fa0 — Fa0/0
PC-PT
PC0
10.0.0.2
Router-PT
Router 0
10.0.0.1

Procedure:

1. Create the above topology with a PC and Router
2. Set gateway of PC to 10.0.0.1 and IP address of PC to 10.0.0.2
3. Go to CLI mode of router and set the IP address of interface for 0/0 to 10.0.0.1
4. After setting up all the topology.
5. Go to router, and enable config mode
6. Type hostname r1, then type enable secret P1 here P1 is password of your choice.
7. Then type IP address 10.0.0.1 & execute no shut connect.
8. In interface mode itself execute line vty 0 5. command which creates 6 lines and next execute. login and in next command password P0 is the password for login in router.
9. Go to command prompt of PC execute telnet 10.0.0.1 which will get you administration access of router. Enter password & access router from a connected end device,

AL 3/8/2023

DATE :

→ Result:

In PC CLI

PC > telnet 10.0.0.1

Trying 10.0.0.1 ... open

User Access Verification

Password : PO

V1> enable

Password : p1

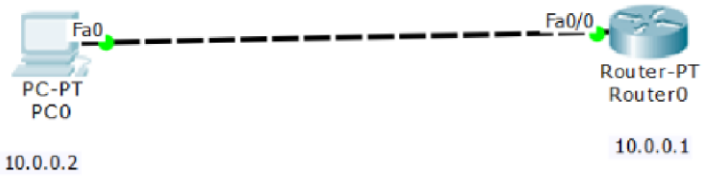
V1 #

Observation:

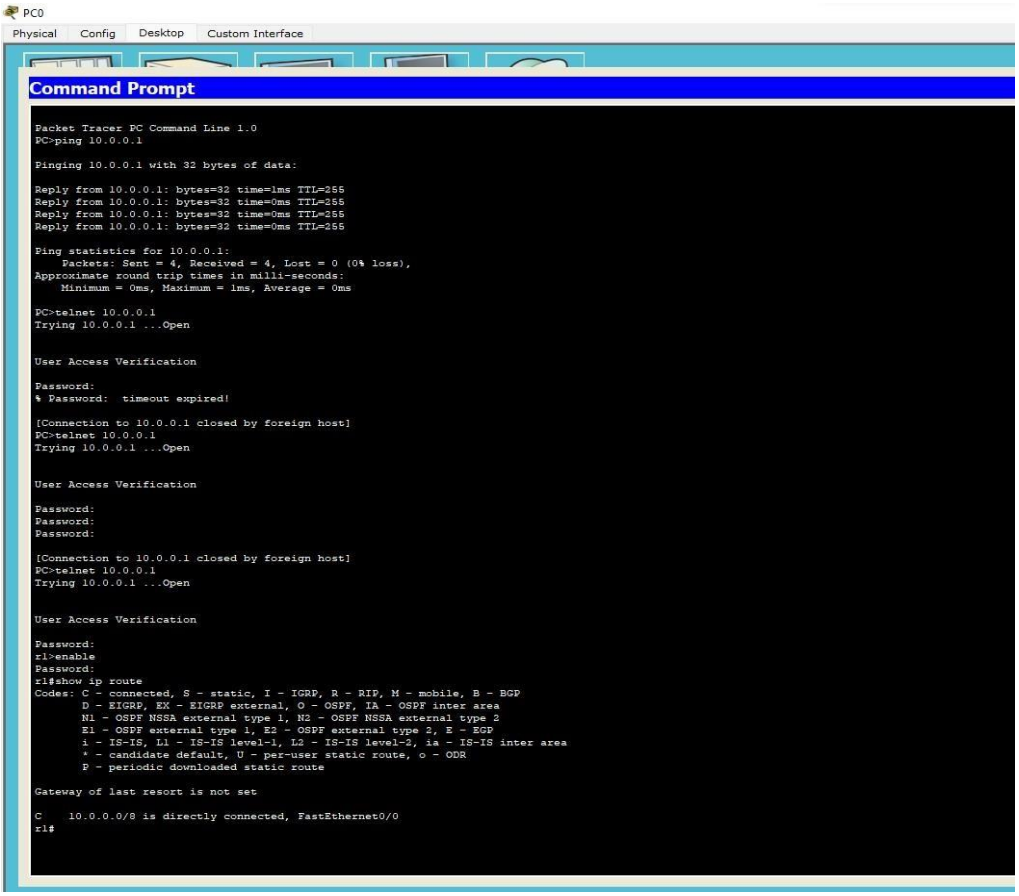
Telnet: in teletype Network. It provides a command line interface to communicate with a server, in this case router.

Using TELNET, we are able to run command in the PC as would be run in a router CLI. If we type show ip route in the PC CLI, we will see the routers response to the command.

TOPOLOGY:



OUTPUT:



LAB 13

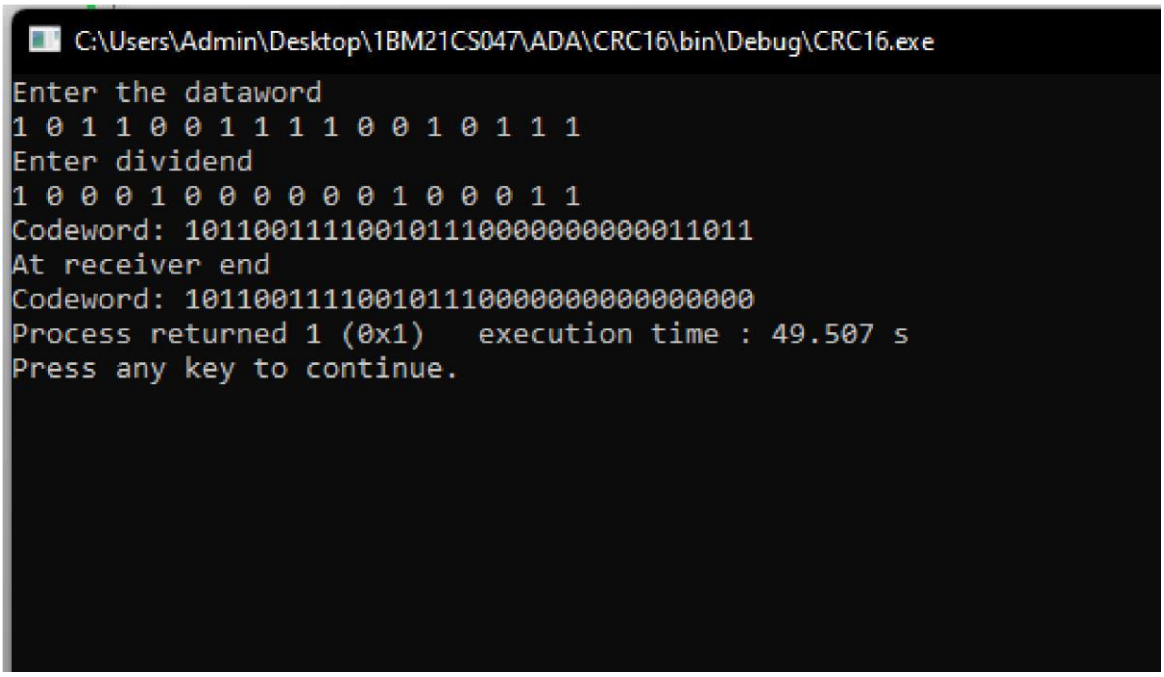
Write a program for error detecting code using CRC- CCITT (16bits).

CODE:

```
#include<stdio.h> int arr[17];

void xor(int x[], int y[])
{
    int k=0;
    for(int i=1;i<16;i++)
    { if(x[i]==y[i]) arr[k++]=0;
        else arr[i]=1;
    }
}

void main()
{ int dd[17],div[33],ze[17],i,k;

    printf("Enter the dataword \n");
    for(i=0;i<17;i++) scanf("%d",&div[i]);
    for(i=i;i<33;i++) div[i]=0;

    for(i=0;i<17;i++) ze[i]=0;
    printf("Enter dividend \n"); for(i=0;i<17;i++)
        scanf("%d",&dd[i]);
    i=0; k=0;
        for(i=i;i<17;i++)
            arr[k++]=div[i];
    while(i<33)
    { if(arr[0]==0) xor(arr,ze);
        else
```

```
            xor(arr,dd); arr[16]=div[i++];

    }
    k=0; for(i=17;i<33;i++) div[i]=arr[k++];
    printf("Codeword: "); for(i=0;i<33;i++)
    printf("%d",div[i]);

    for(i=0;i<17;i++) arr[i]=0; printf("\nAt receiver end

    \n");

     k=0;

        for(i=i;i<17;i++)

            arr[k++]=div[i];

    while(i<33)
    { if(arr[0]==0) xor(arr,ze); else xor(arr,dd);
        arr[16]=div[i++];

    }
    k=0;
    for(i=17;i<33;i++) div[i]=arr[k++]; printf("Codeword: ");

    for(i=0;i<33;i++) printf("%d",div[i]);

}
```

OUTPUT:



OBSERVATION:

Lab 10:

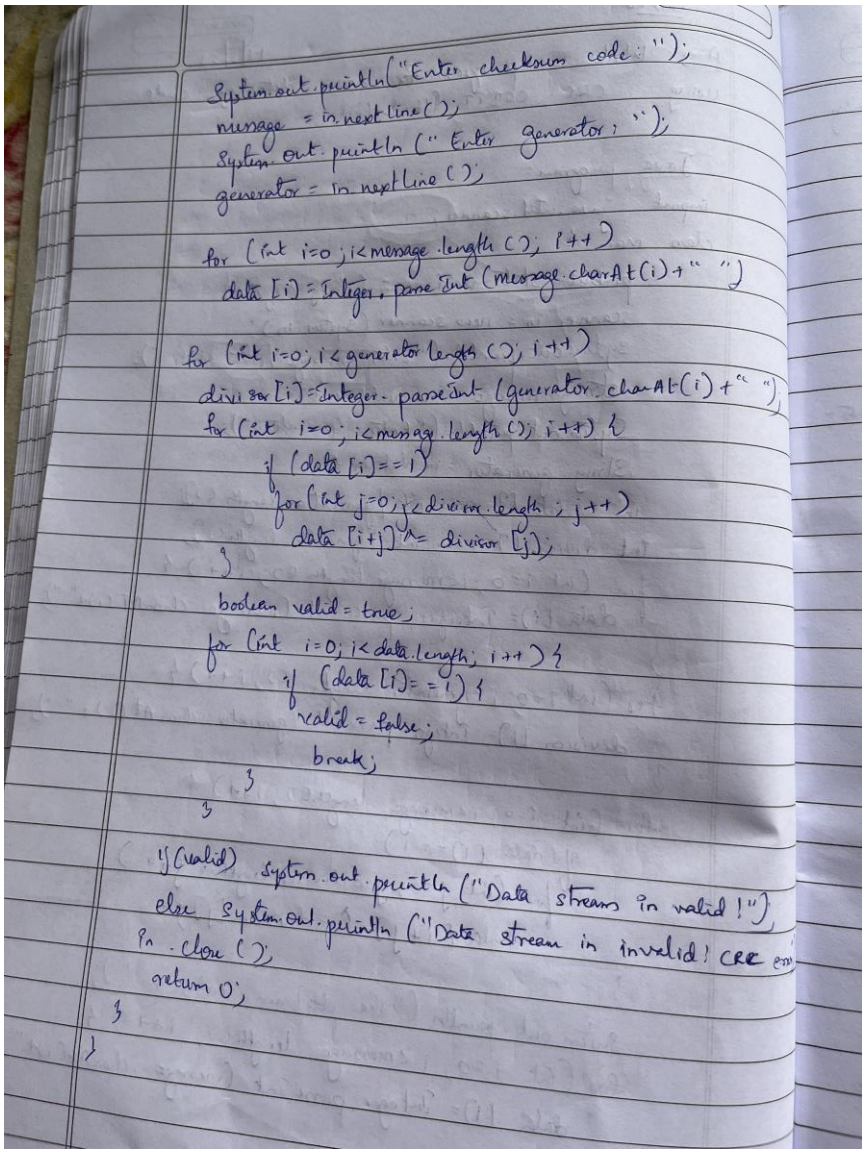
DATE: 1-1-8-23

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

Java program:

```
import java.util.scanner;
class main {
    public static void main (String args[]) {
    scanner in = new scanner (System.in);
    system.out.println("Enter message /data bits:");
    String message = in.nextLine();
    System.out.println("Enter Generator:");
    String generator = in.nextLine();
   int data[] = new int [generation.length()];
   int divisor[] = new int [generator.length()];
   for (int i=0; i<message length(); i++) {
    data[i] = Interger.parseInt (message.charAT(i)+"");
   }
   for (int i=0; i<generator.length(); i++) {
     division[i] = Integer.parseInt (generator.charAt(i)+"");
   }
   for (int i=0; i<message.length(); i++) {
       if (data[i]==1)
          for (int j=0; j<division.length; j++)
             data[i+j] ^= division[j];
   }
    System.out.println("The checksum code is");
    for (int i=0; i<message length(); i++) {
       data[i] = Integer.parseInt (message.charAt(i)+"");
   }
   for(int i=0; i<data.length; i++)
      system.out.println(data[i]);
   System.out.println();
```

31/8/2023



```
System.out.println("Enter checksum code: ");
message = in.nextLine();
System.out.println("Enter generator: ");
generator = in.nextLine();

for (int i=0; i<message.length(); i++)
 data[i] = Integer.parseInt(message.charAt(i)+"")

for (int i=0; i<generator.length(); i++)
divisor[i]=Integer.parseInt(generator.charAt(i)+"");
for (int i=0; i<message.length(); i++) {
    if (data[i]==1)
     for(int j=0; j<divisor.length; j++)
      data[i+j]^= divisor[j];
}
boolean valid = true;
for (int i=0; i<data.length; i++) {
    if (data[i]==1) {
     valid = false;
      break;
    }
}

if(valid) System.out.println("Data stream is valid!");
else System.out.println("Data stream is invalid! CRC error");
in.close();
return 0;
}
}
```

PAGE NO :
DATE :

Output:

- Enter message/data bits: 1110000
  Enter generator: 1001
  The checksum code is: 1110000111
  Enter checksum code: 1110000111
  Enter generator: 1001
  Data stream is valid

- Enter message/data bits: 1110000
  Enter generator: 1001
  The checksum code is: 1110000111
  Enter checksum code: 1110000110 // error
  Enter generator: 1001
  Data stream is not valid! CRC error.

3/8/2023

Lab 14

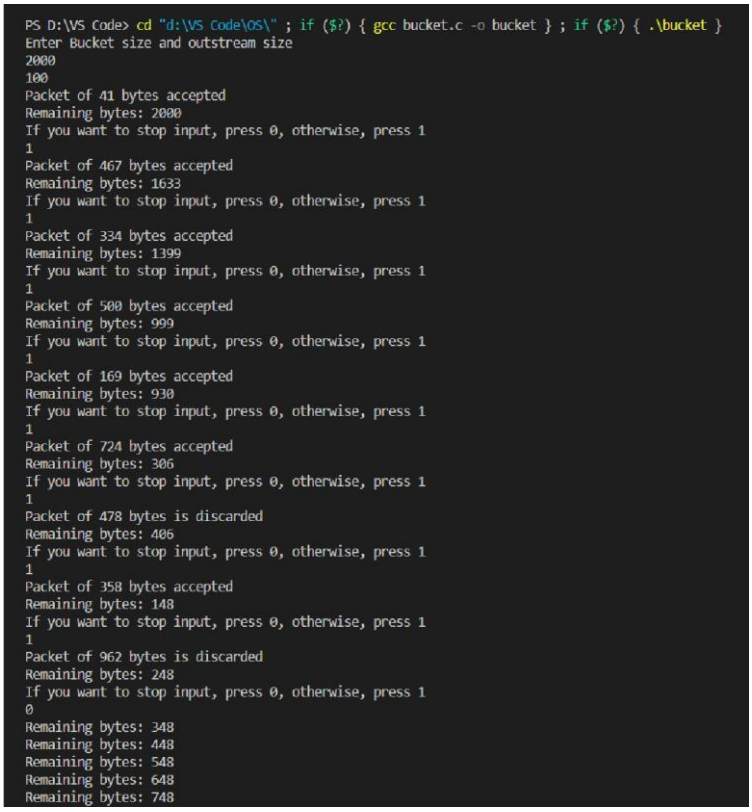
Write a program for congestion control using Leaky bucket algorithm.

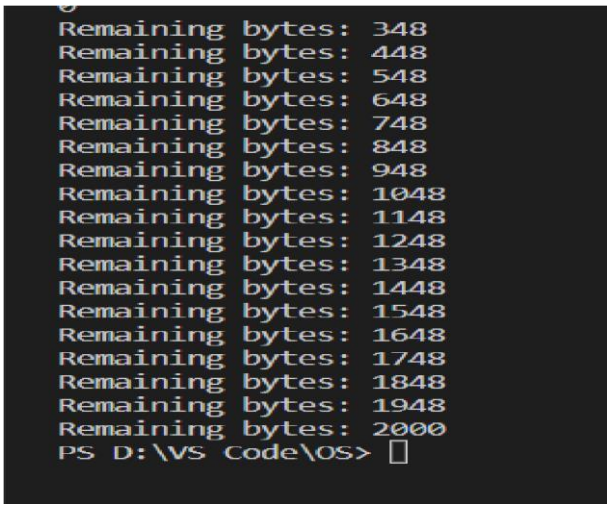
CODE:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function int main()
{
    int buckets, outlets, k = 1, num, remaining;
    printf("Enter Bucket size and outstream size\n"); scanf("%d %d", &buckets, &outlets);
    remaining = buckets; while (k)
    {
        num = rand() % 1000; // Generate a random number between 0 and 999 if (num < remaining)
        {
                                    remaining = remaining - num; printf("Packet of %d bytes
            accepted\n", num); // Added missing
variable
        }
        else
        {
            printf("Packet of %d bytes is discarded\n", num);
        }
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else remaining = buckets; printf("Remaining
        bytes: %d \n", remaining); printf("If you want to stop input, press 0, otherwise, press
        1\n"); scanf("%d", &k);
    }
    while (remaining < buckets) // Fixed the condition
    {
        if (buckets - remaining > outlets)
```

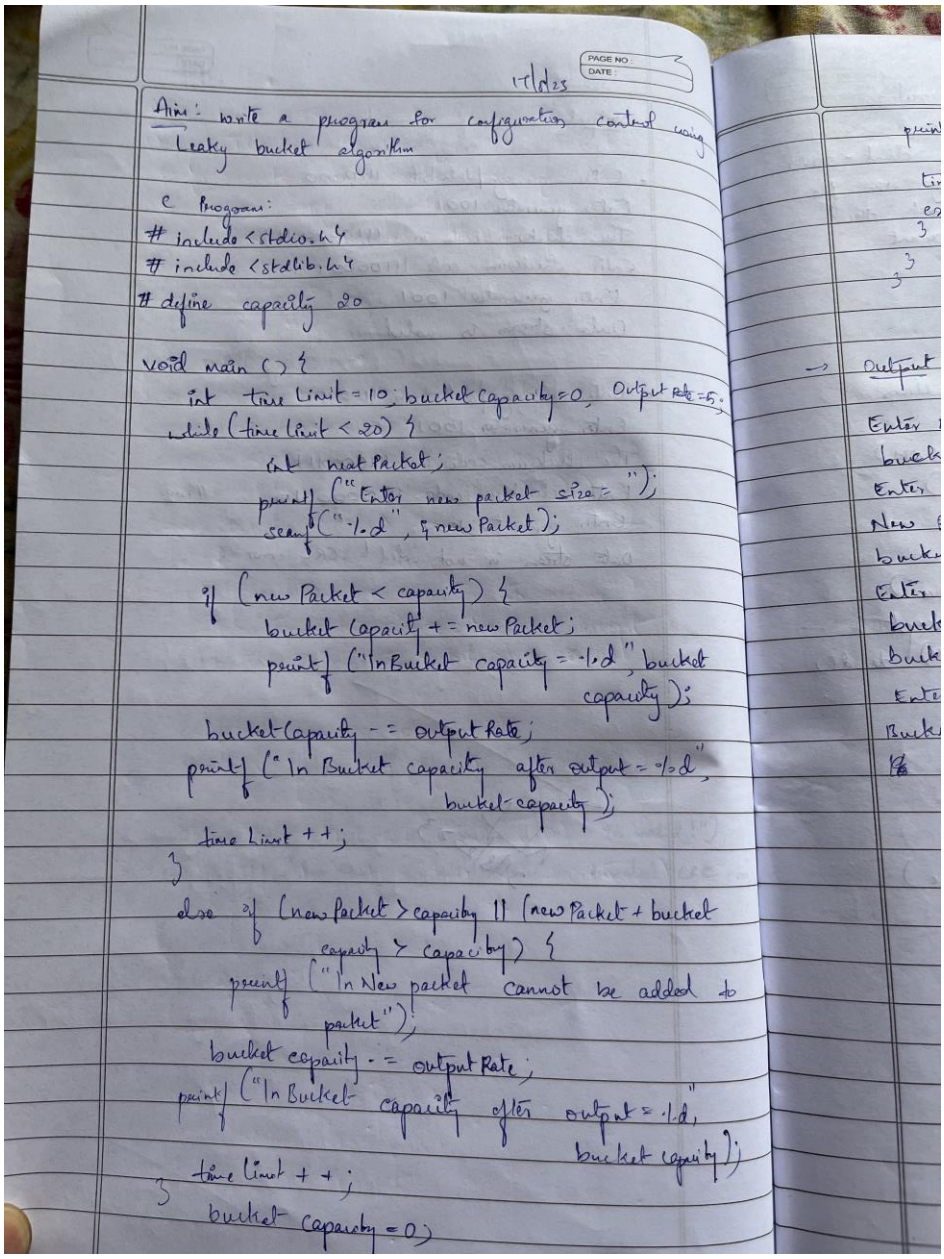
```
        {
            remaining += outlets; // Fixed the calculation
        }
        else remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
                                    return 0; // Added a return statement to indicate successful completion
}
```

OUTPUT:




Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>

OBSERVATION:



17/1/23

Aim: write a program for congestion control using Leaky bucket algorithm

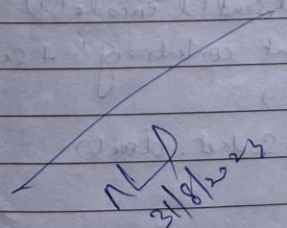
C Program:

```
#include <stdio.h>
#include <stdlib.h>
#define capacity 20

void main () {
    int timeLimit = 10, bucketCapacity = 0, outputRate = 5;
    while (timeLimit < 20) {
        int newPacket;
        printf ("Enter new packet size = ");
        scanf ("%d", &newPacket);

        if (newPacket < capacity) {
            bucketCapacity += newPacket;
            printf ("\nBucket capacity = %d", bucket
                                                capacity);
        bucketCapacity -= outputRate;
        printf ("\n Bucket capacity after output = %d",
                                    bucketCapacity);
        timeLimit++;
    }
    else if (newPacket > capacity || (newPacket + bucket
                capacity > capacity)) {
        printf ("\n New packet cannot be added to
                                    packet");
        bucketCapacity -= outputRate;
        printf ("\n Bucket capacity after output = %d",
                                    bucketCapacity);
    timeLimit++;
    bucketCapacity = 0;
```

```
        printf ("\n Bucket capacity after output =>%d",
                                    bucket capacity);
        timelimit ++;
        exit (0);
    }
}
```

→ Output:

Enter New packet size = 15
bucket capacity : 10
Enter new packet size = 27
New packet cannot be added to bucket
bucket capacity after output = 5
Enter new packet size = 0
bucket capacity : 5
Bucket capacity after output : 0
Enter new packet size : 0
Bucket capacity after output = -5



WEEK15

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.

CODE:

ClientTCP.py from socket import * serverName =

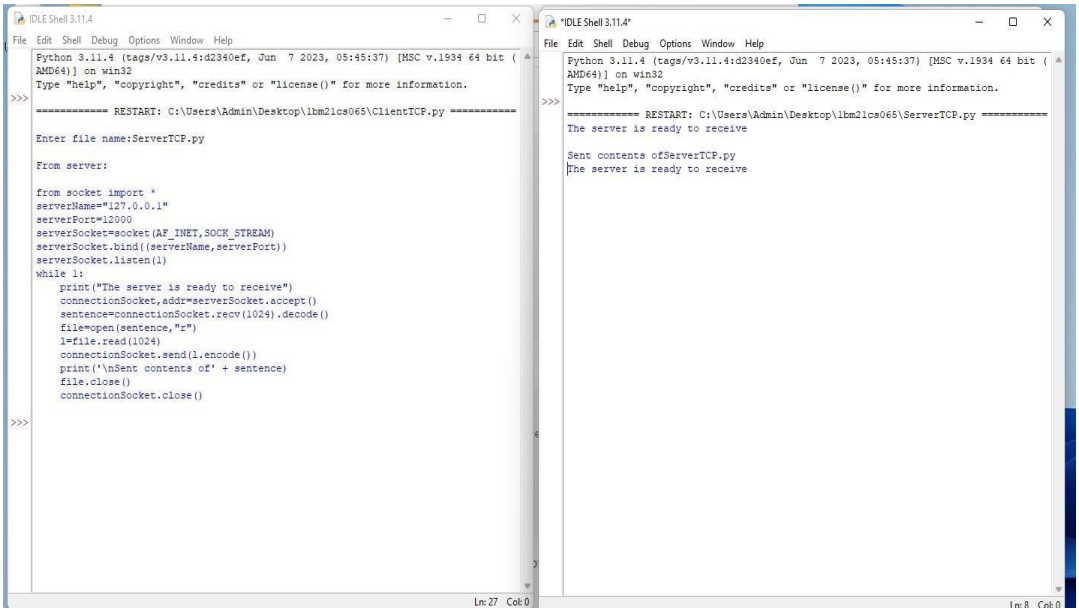
“127.0.0.1” serverPort = 12000 clientSocket = socket(AF_INET, SOCK_STREAM) clientSocket.connect((serverName,serverPort)) sentence = input(“\nEnter file name: “) clientSocket.send(sentence.encode()) filecontents = clientSocket.recv(1024).decode() print (“\nFrom Server:\n”) print(filecontents) clientSocket.close()

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”)

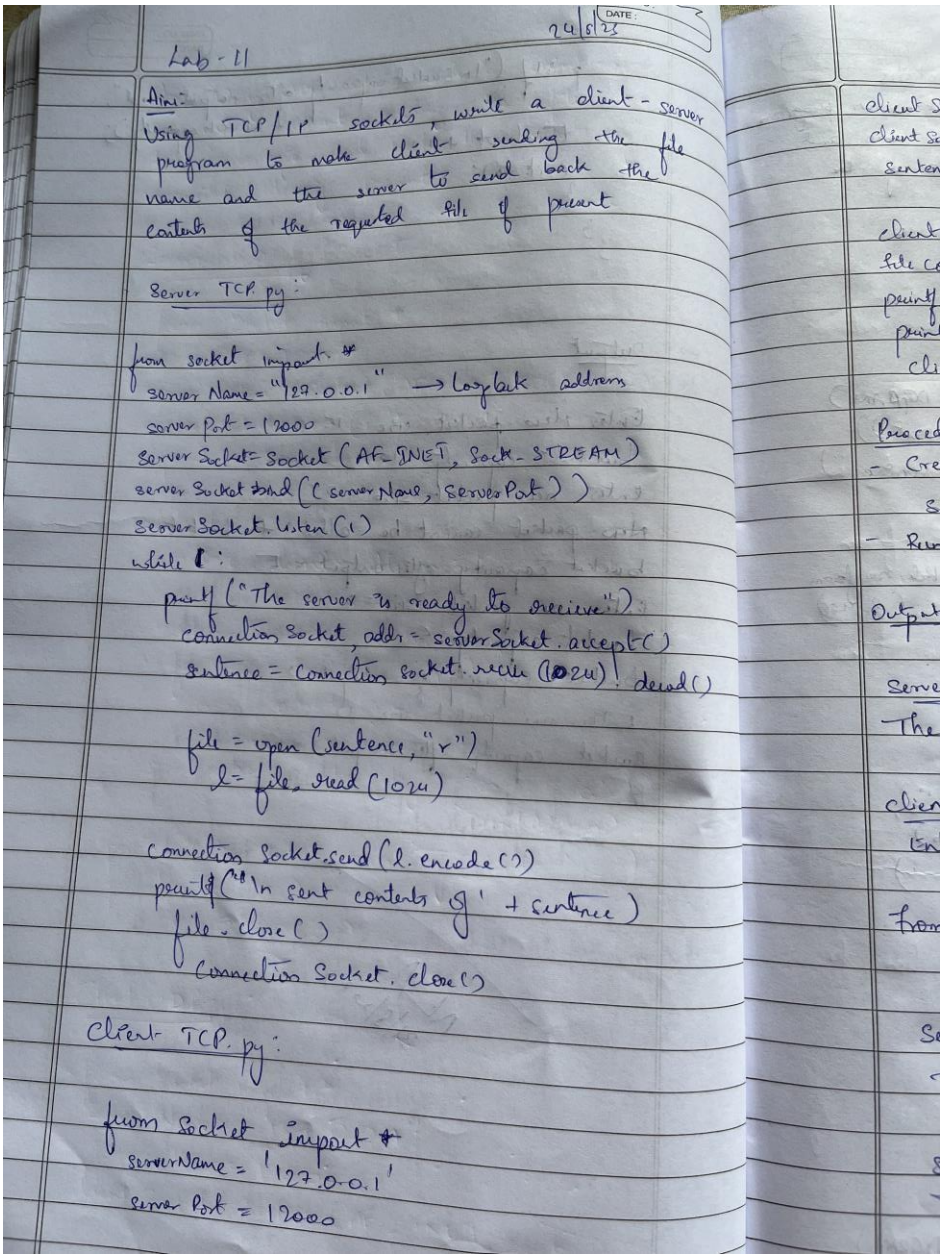
connectionSocket, addr = serverSocket.accept() sentence = connectionSocket.recv(1024).decode() file=open(sentence,”r”) l=file.read(1024) connectionSocket.send(l.encode()) print (“\nSent contents of “ + sentence)

file.close() connectionSocket.close()

OUTPUT:


IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\1bm21cs065\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: 27 Col: 0
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\1bm21cs065\ServerTCP.py ==========
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive
Ln: 8 Col: 0

OBSERVATION:



Lab - 11

DATE: 24/5/23

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

Server TCP.py:

```
from socket import *
serverName = "127.0.0.1"  → Loopback address
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file = open(sentence, "r")
    l = file.read(1024)

connectionSocket.send(l.encode())
print("\nSent contents of " + sentence)
file.close()
connectionSocket.close()
```

Client TCP.py:

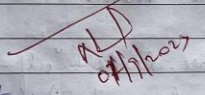
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
```

```
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()
```

Procedure:

- Create 2 IDLE instance and write client and server files
- Run server first and then the client.

Output:



Server Instance:

The server is ready to receive

client Instance:-

Enter file name : serverTCP.py

From server:

the contents of serverTCP.py is displayed here

Server Instance:-

The server is ready to receive

Sent contents of server TCP.py

The server is ready to receive.

LAB 16

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

CODE:

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,”utf8”),(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:

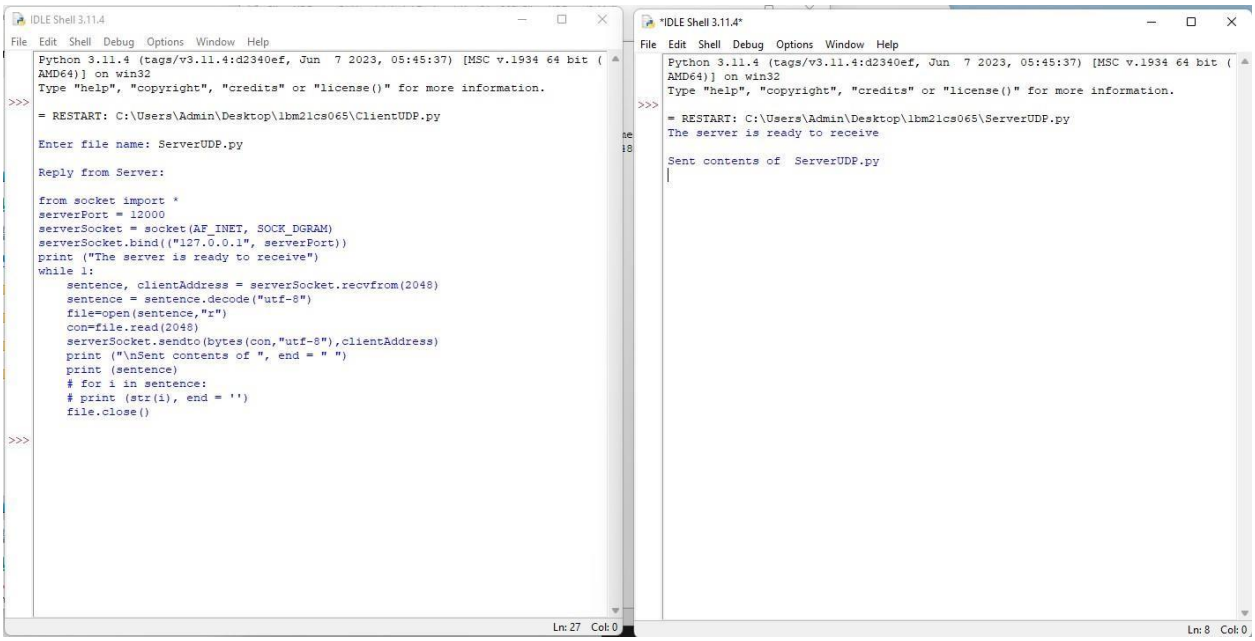
sentence, clientAddress = serverSocket.recvfrom(2048) sentence

= 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:



OBSERVATION:

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

Server UDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverport))
print("The server is ready to receive")

while 1:
    sentence, clientAddress = serverSocket.recvfrom
                                (2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)

    serverSocket.sendto(bytes(con, "utf-8"),
                        clientAddress)
print('\n sent contents of ', end = ' ')
print(sentence)
    file.close()
```

Client UDP.py:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
```

sentence = input ("\n Enter file name = ")

client:
client Socket = socket (AF_INET, sock_DGRAM)

sentence = input ("\n Enter file name = ")

client:
client socket.sendto (bytes (sentence, "utf-8"),
(serverName, serverPort))
file contents, serverAddress = clientSocket.recvfrom
(2048)
print ("\n Reply from server: \n")
print (fileContents.decode ("utf-8"))

clientSocket.close()
clientSocket.close()

01/09/2023

Output:

Server Instance:
The server is ready to recieve

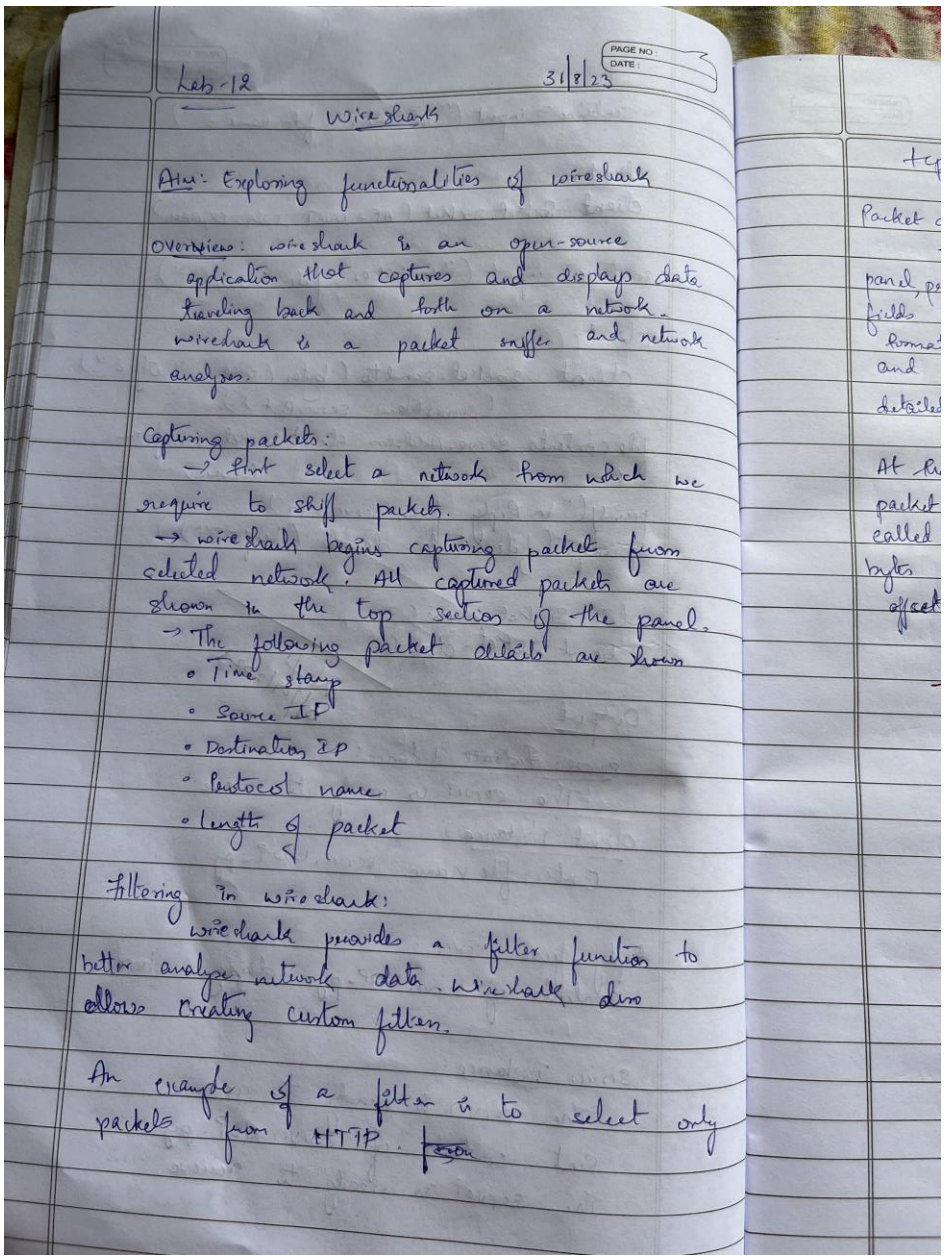
client Instance:
Enter file name: serverUDP.py

Reply from server:
contents of serverUDP.py displayed here.

Server Instance:
The server is ready to recieve.
Sent contents of serverUDP.py
The server is ready to recieve.

LAB 17

Tool Exploration -Wireshark OBSERVATION:



Lab-12 31/8/23

Wireshark

Aim: Exploring functionalities of wireshark

Overview: wireshark is an open-source application that captures and displays data travelling back and forth on a network. wireshark is a packet sniffer and network analyser.

Capturing packets:
→ First select a network from which we require to sniff packets.
→ wireshark begins capturing packet from selected network. All captured packets are shown in the top section of the panel.
→ The following packet details are shown
- Time stamp
- Source IP
- Destination IP
- Protocol name
- Length of packet

Filtering in wireshark:
wireshark provides a filter function to better analyse network data. wireshark also allows creating custom filters.

An example of a filter is to select only packets from HTTP.

PAGE NO :
DATE :

tcp.port == 80 || udp.port == 80

Packet details:

The middle section of the panel, detail panel, presents the protocols and protocol fields of the selected packet in a collapsible format. We can apply additional filters and right click on protocol for a detailed view.

At the bottom panel, raw data of the selected packet is seen in hexadecimal format. It is called as hex dump. It contains 16 hexadecimal bytes and 16 ASCII bytes alongside the data offset.

KL
01/09/2023