VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COURSE TITLE

Submitted by

Aryan Madhan Pillai (1BM21CS033)

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 "LAB COURSE **TITLE**" carried out by Aryan Madhan Pillai **(1BM21CS033),** 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 **Course Title - (Course code)** work prescribed for the said degree.

Name of the Lab-Incharge
Dr. Shyamala G
Associate Prof.
Department of CSE
BMSCE, Bengaluru

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

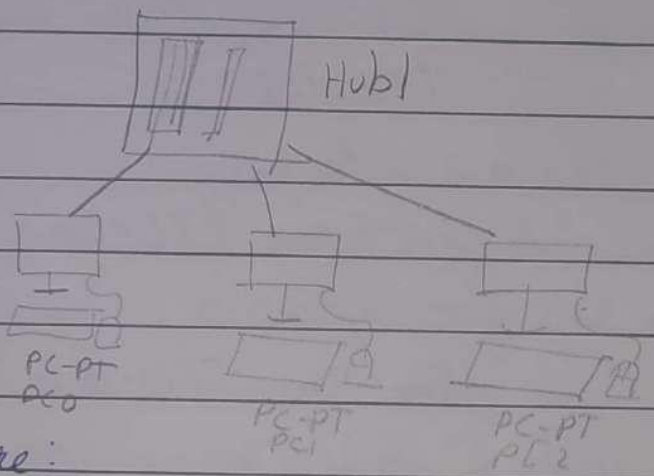
Index

Date: 15/6/23

EXPERIMENT 1

Aim: Create a topology, here simulate sending a simple PDU from source to destination using a simple hub and virtual as connecting

Topology: Hub to PC



Procedure:

1) Select hub and three PC's
2) Connect the hub to the individual PC using copper straight through wires
3) Assign the IP address to the PC's 10.0.0.1, 10.0.0.2, 10.0.0.3.
4) Select the packet and select the source and destination PC

Observation: In simulation mode

- PC sends packet to hub and hub sends it to both PC1 and PC2.
- PC1 discards the message whereas PC2 accepts it

- PC2 sends the acknowledgement packet to the hub
- Hub again sends it to PC0 and PC1
- PC1 discards and PC0 accepts it

Output

Reply from 10.0.0.2 bytes = 32
time = 2ms
TTL = 128

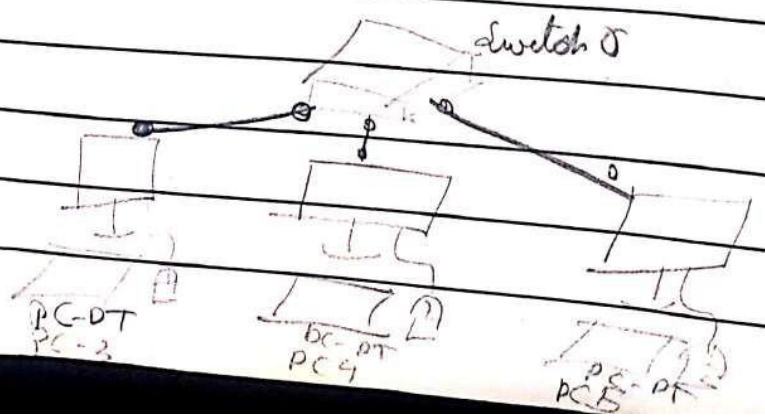
Reply from 10.0.0.2 bytes = 32
time = 0ms TTL = 128

Reply from 10.0.0.2 bytes = 32
time = 3ms TTL = 128

Reply from 10.0.0.2 bytes = 32
time = 0ms TTL = 128

Ping statistics for 10.0.0.2
Packet: Sent = 4, Received = 4, 6
lost = 0 (0% loss)
Approximate round trip time in milli seconds
minimum = 0ms, Maximum = 3ms
Average = 2ms

Topology: Switch to PC



Date :

Procedure:

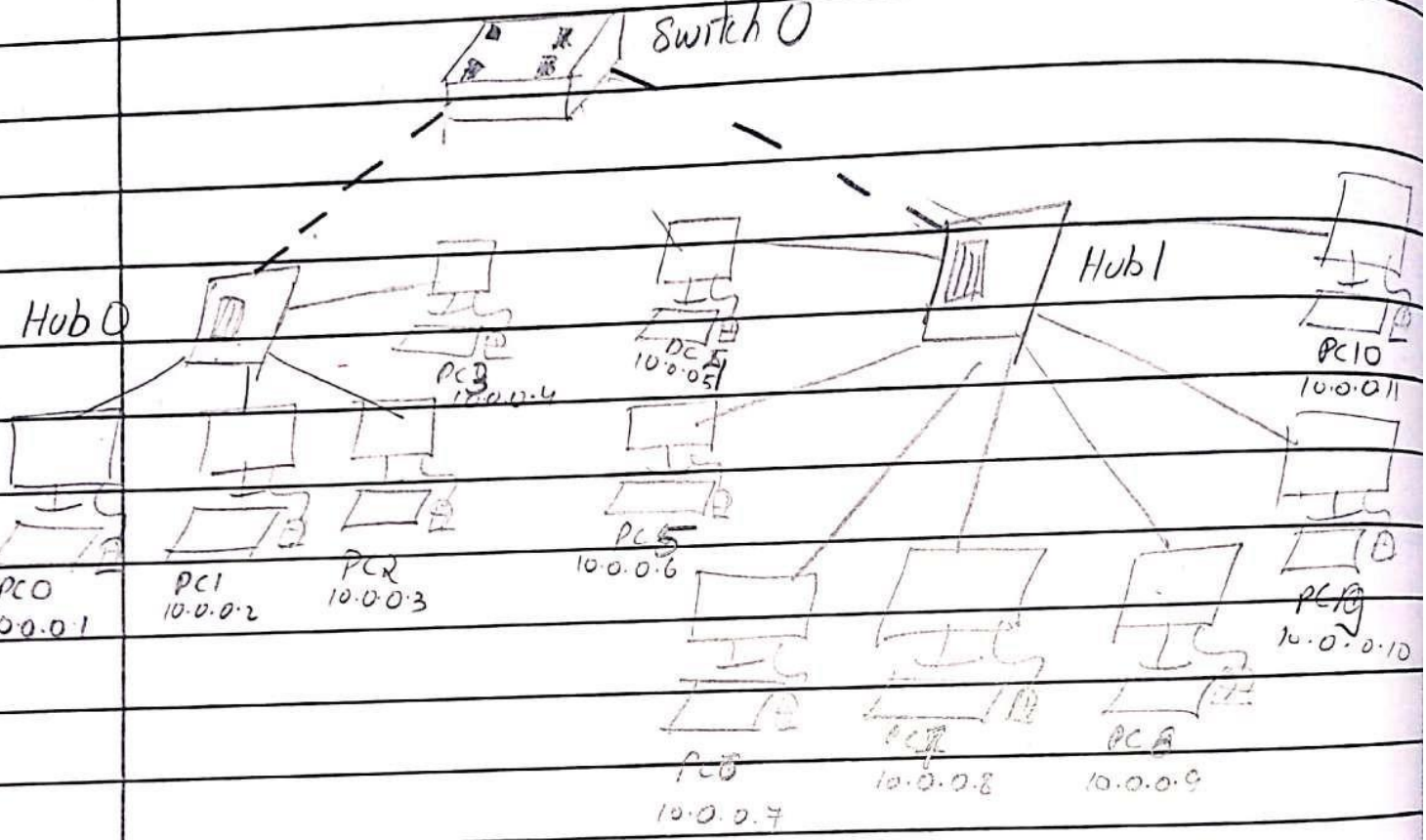
1) Select a switch and 3 PC's
2) Connect the switch to the individual PC's using a copper straight through
3) Assign the IP addresses to the PC's - 10.0.0.4, 10.0.0.5, 10.0.0.6 respectively
4) Select the PDU and the source and destination PC.

Output

Reply from 10.0.0.5 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.5 bytes = 32 time = 0 ms TTL = 128
Reply from 10.0.0.5 bytes = 32 time = 3ms TTL = 128
Reply from 10.0.0.5 bytes = 32 time = 3 ms TTL = 128
Ping Statistics for 10.0.0.5
Packets: Sent = 4, Received = 4 lost = 0 (0% loss)
Approximate round trip times in milli seconds
Minimum = 0ms Maximum = 3ms, Average = 2ms

Observations: In simulation mode
- PC3 sends packets to switch and it sends to both PC4 and PC5 in first round.
- PC4 rejects and PC5 accepts and sends acknowledgement packet to both PC3 and PC5.
PC4 discards it PC3 accepts it.
Now because of switch PC3 sends packets only to PC5.

Topology :- Hybrid - Hub and Switch



Procedure

- 11 PC's 2 generic Hub's and 1 switch were placed in the workspace
- 4 PC's were connected to Hub 0 via copper straight through cables. Remaining 7 PC's were connected to hub 1 via copper straight through cable.
- All PC's were assigned IP's (10.0.0.1 to 10.0.0.11)
- The 2 hubs were connected to the switch via copper cross over cables which are used to connect device on the same land.

Date : 1

Result

PC1 command line :

PC> ping 10.0.0.8

Pinging 10.0.0.8 with 32 bytes of data :

Reply from 10.0.0.8 bytes = 32 time = 8ms TTL = 128
Reply from 10.0.0.8 bytes = 32 time = 20ms TTL = 128
Reply from 10.0.0.8 bytes = 32 time = 20ms TTL = 128
Reply from 10.0.0.8 bytes = 32 time = 20ms TTL = 128

Ping statistics for 10.0.0.8
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip time in milli seconds
Minimum = 0ms Maximum = 8ms, Average = 2ms

Observations :

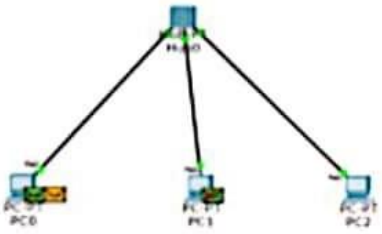
Hub : A hub recieves data and broadcasts it to all devices connected to it.

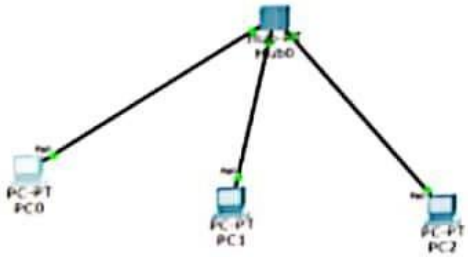
Switch : A switch prevents traffic between 2 devices from being shared with other devices connected to it. It sends message only to the reciever.

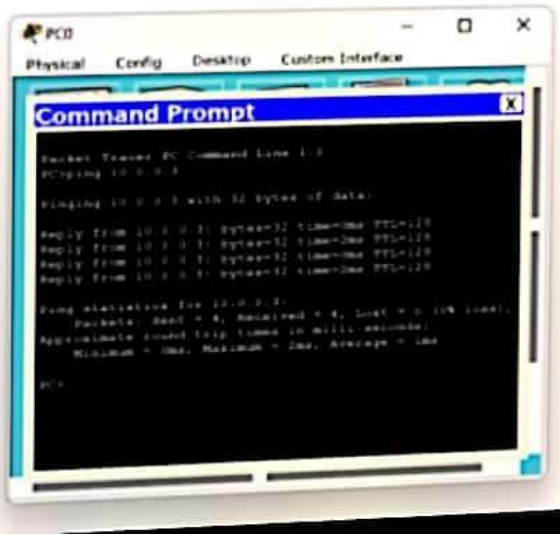
22/6/22

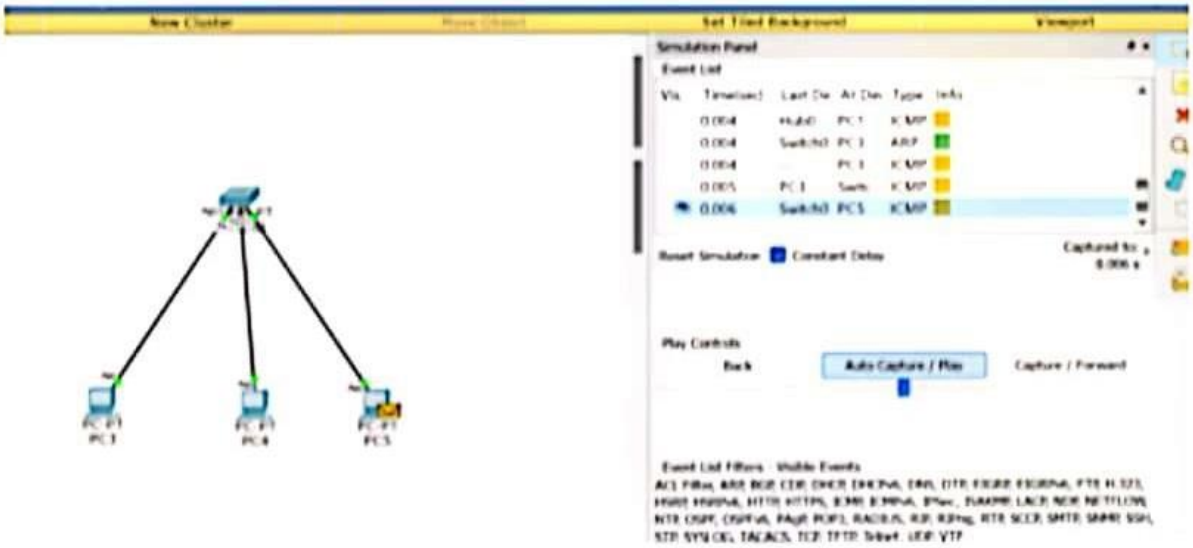
OUTPUT :

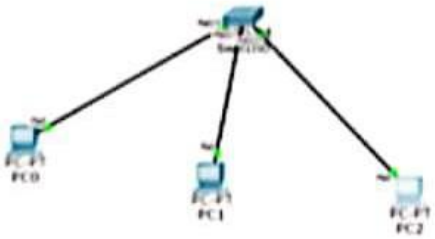
1.Hub and PCs

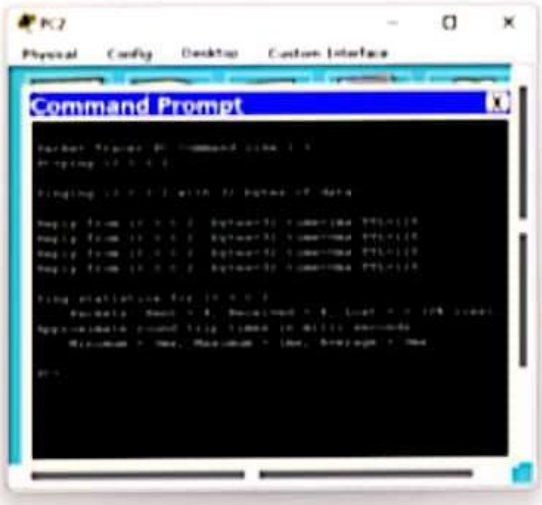


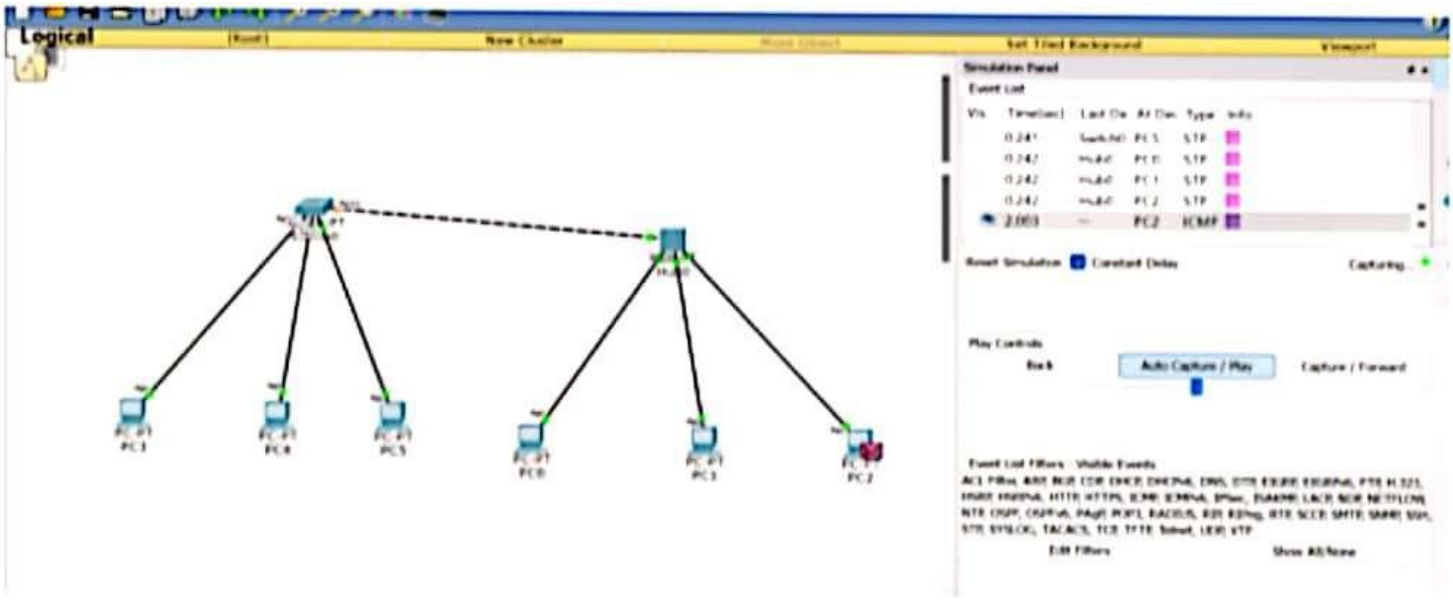


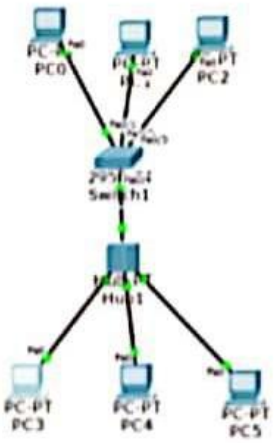


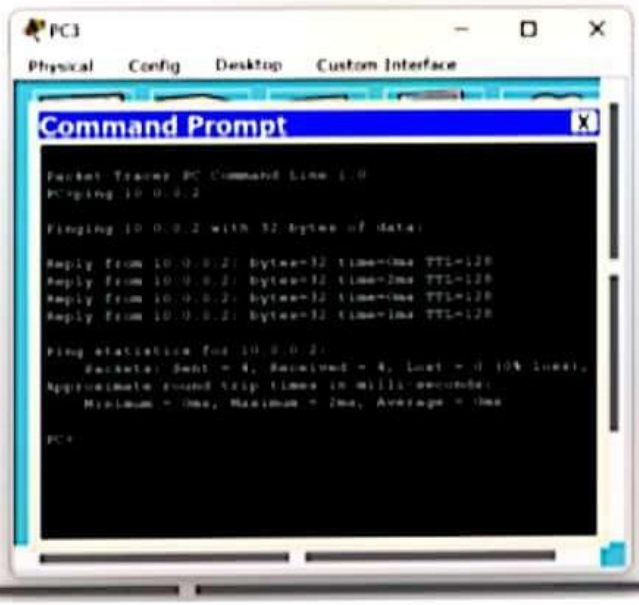












Date : 24/6/23

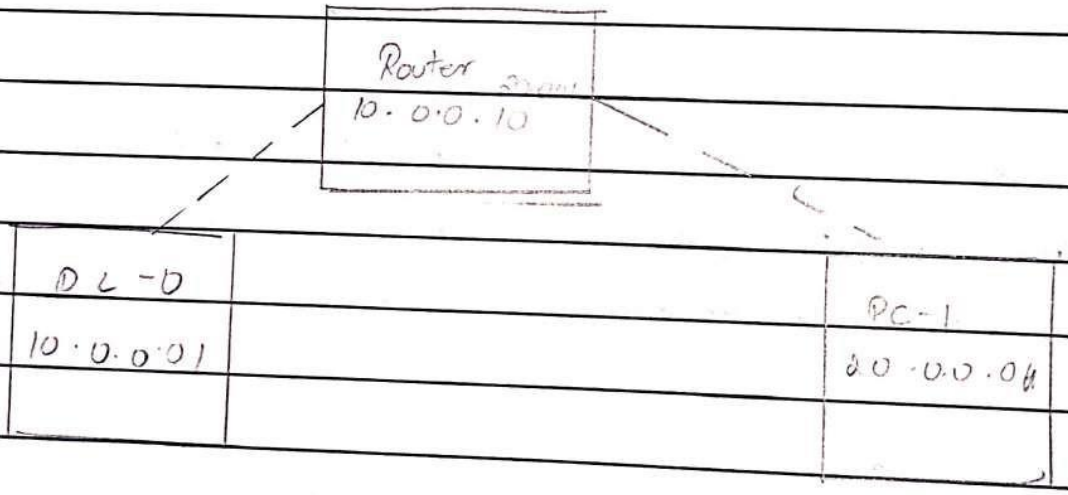
Exp - 2

Network Connection Using Single Router

Aim:

Configuring IP address to routers, explore ping responses, destination unreachable, request timed out and reply.

Topology:



Procedure:

1. Connect two end devices to a router through copper cross-over cable
2. Assign IP address to end devices.
3. Configure getaways in router through CLI using the following commands:

Date :

ⓐ Enable.
ⓑ config t
ⓒ interface <port>
ⓓ ip address <ip address> <subnet mask>
ⓔ no shut
ⓕ exit

④ Set the respective gateways in the end devices
⑤ Ping from one end user to another.

Result :

Pinging 20.0.0.1 with 32 bytes of data :
Request timed out
Reply from 20.0.0.1 bytes = 32 time = 1ms TTL = 127
Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127

Ping statistics from 20.0.0.1 :
Packets : Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip times in milli seconds :
Minimum = 0 ms, Maximum = 1 ms, Average = 0 ms

Observation :

Router is a device used to connect multiple networks. Router is capable of transforming packets

from one network to another. End device sends data packet to router. The destination IP address is noted by the router. The packet is redirected towards the concerned network by the router.

Eg: For Router 0 CLI

```
Root>enable
Router # config t
Router (config)# interface fastethernet 0/0
Router (config-if)# ip address 10.0.0.10 255.0.0.0
Router (config-if)# no shut
exit
Router (config-if)# interface fastethernet 1/0
Rater (config-if)# ip address 20.0.0.10 255.0.0.0
Routes (config-if)# no shut
exit
```

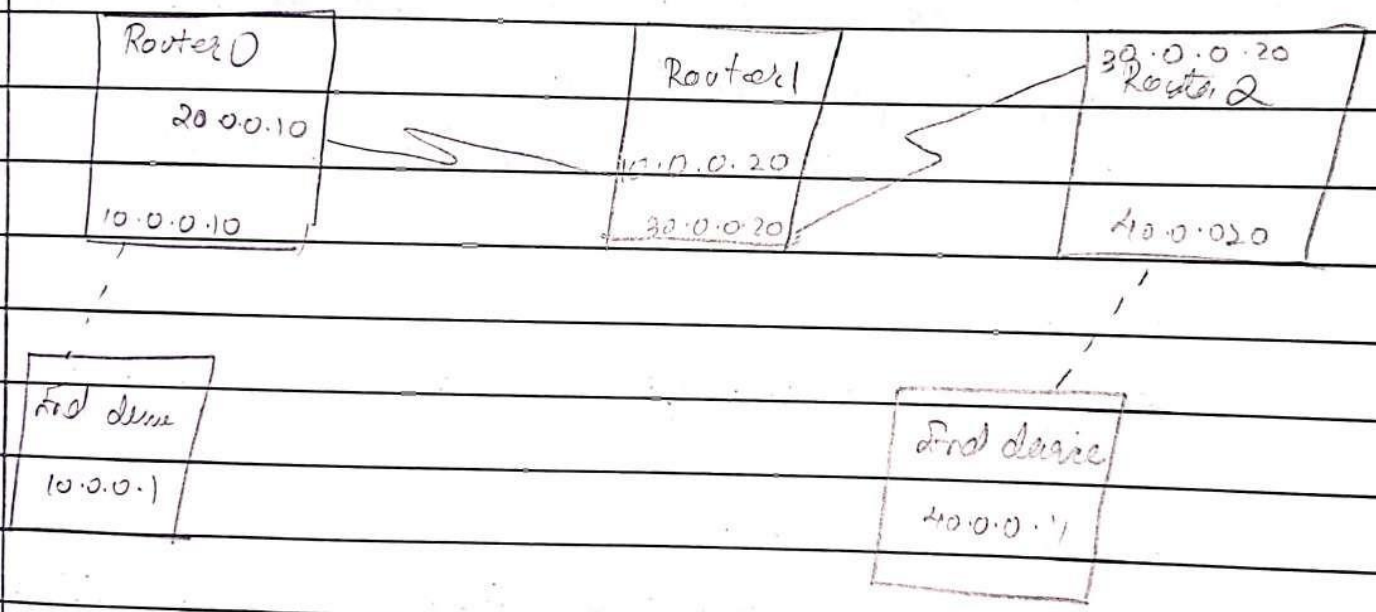
EXP - 2

Network with multiple routers

Aim:

Configuring IP address of multiple routers, exploring ping responses, destination unreachable, request timed out and reply.

Topology:



Procedure:

① Add two end devices and three routers to work space.

② Connect ~~routers~~ to through serial DTE cable and end devices to routers through copper cross-over cable.

③ Assign IP addresses to end devices and gateways

④ Configure gateways through CLI using following

commands

(a) Enable

(b) config t

(c) interface <port>

(d) ip address <ip address> <subnet mask>

(e) no shut

(f) exit

(5) Using command ip route <destination ip> <routing ip> set path for each router

(6) Ping from one end devices to another.

Result:

Pinging 40.0.0.1 : with 32 bytes of data :

Request timed out :

Reply from 40.0.0.1 : bytes = 32 time = 12ms TTL = 127

Reply from 40.0.0.1 : bytes = 32 time = 12ms TTL = 127

Reply from 40.0.0.1 : bytes = 32 time = 14ms TTL = 127

Ping statistics from 40.0.0.1:

Packets : Sent = 4, Received = 3, Lost = 1 (25% loss)

Approximate round trip times in milli seconds

Minimum = 12ms Maximum = 14 ms Average = 8 ms

Date :

Configure the routers by opening CLI

Router 0 :

Router > enable
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.10 255.0.0.0
Router (config-if) # no shut
exit
Router (config) # interface serial 2/0
Router (config) # ip address 20.0.0.10 255.0.0.0
Router (config-if) # no shut
exit
exit

Router 1 :

Router > enable
Router # config t
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.20 255.0.0.0
Router (config-if) # no shut
exit
Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.20 255.0.0.0
Router (config-if) # no shut
exit
Router (config) # exit

Observation

Destination host Unreachable

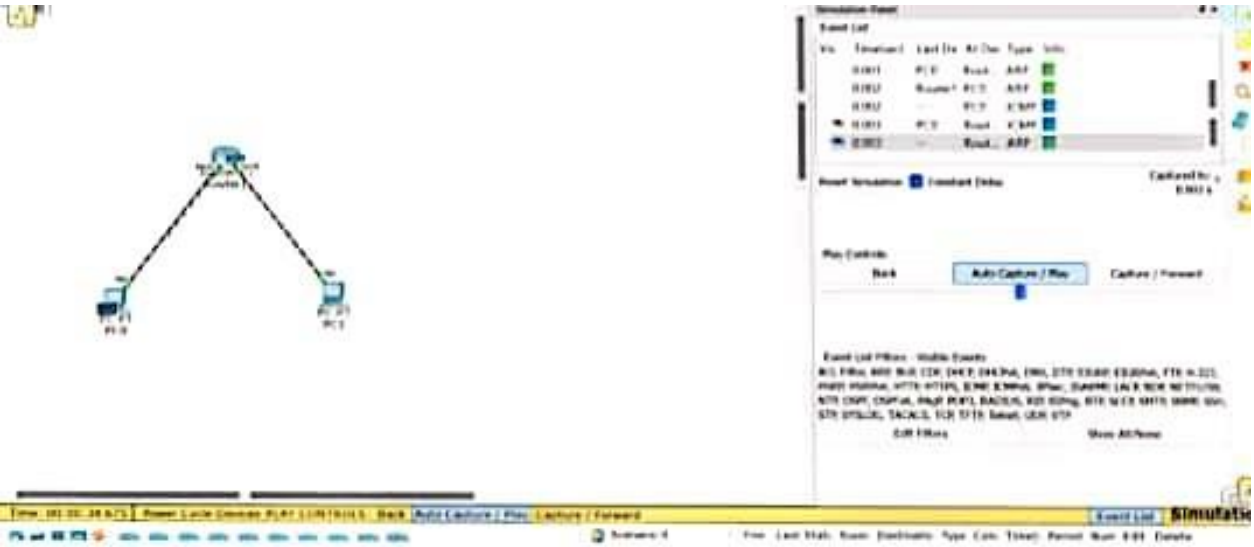
For each router, we need to define a route for packets to be moved to different networks. Unless route is defined, packet will not reach destination. Following result is obtained when gateway isn't set.

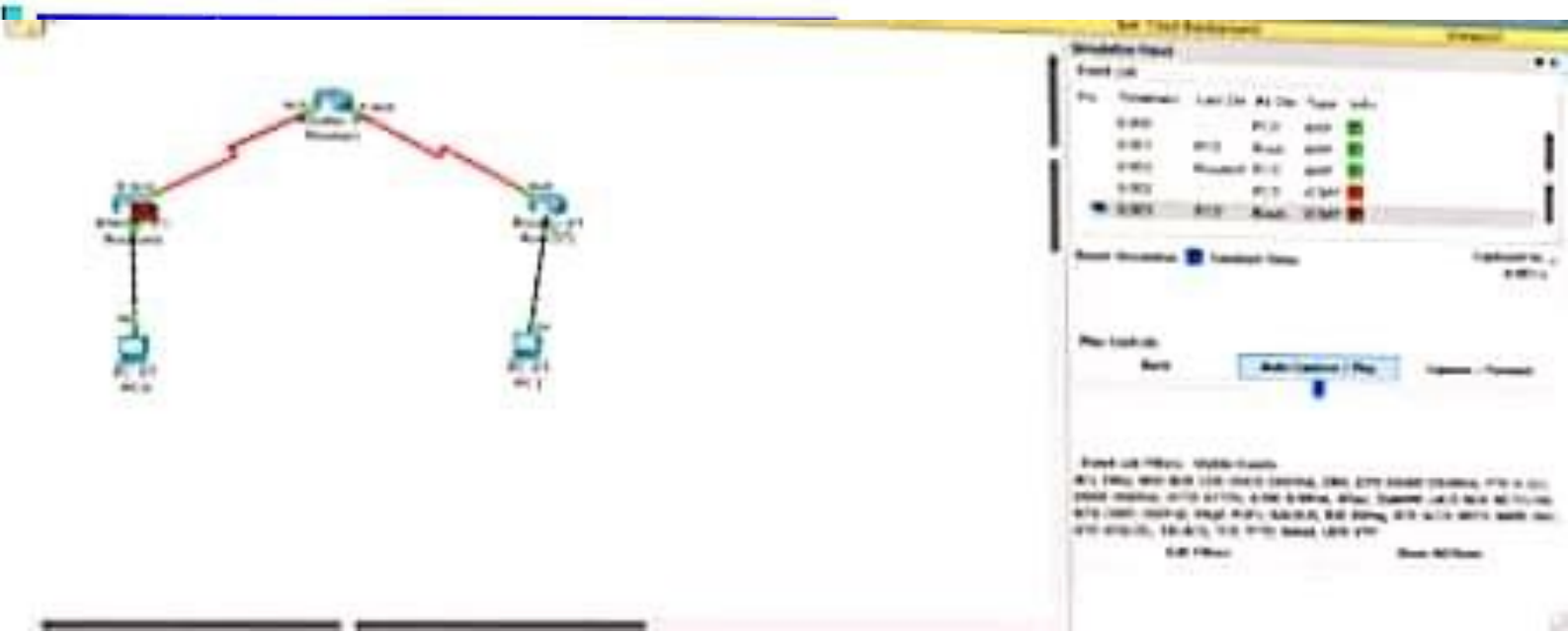
Reply from 10.0.0.10 : Destination host unreachable
Reply from 10.0.0.10 : Destination host unreachable
Reply from 10.0.0.10 : Destination host unreachable
Reply from 10.0.0.10 : Destination host unreachable

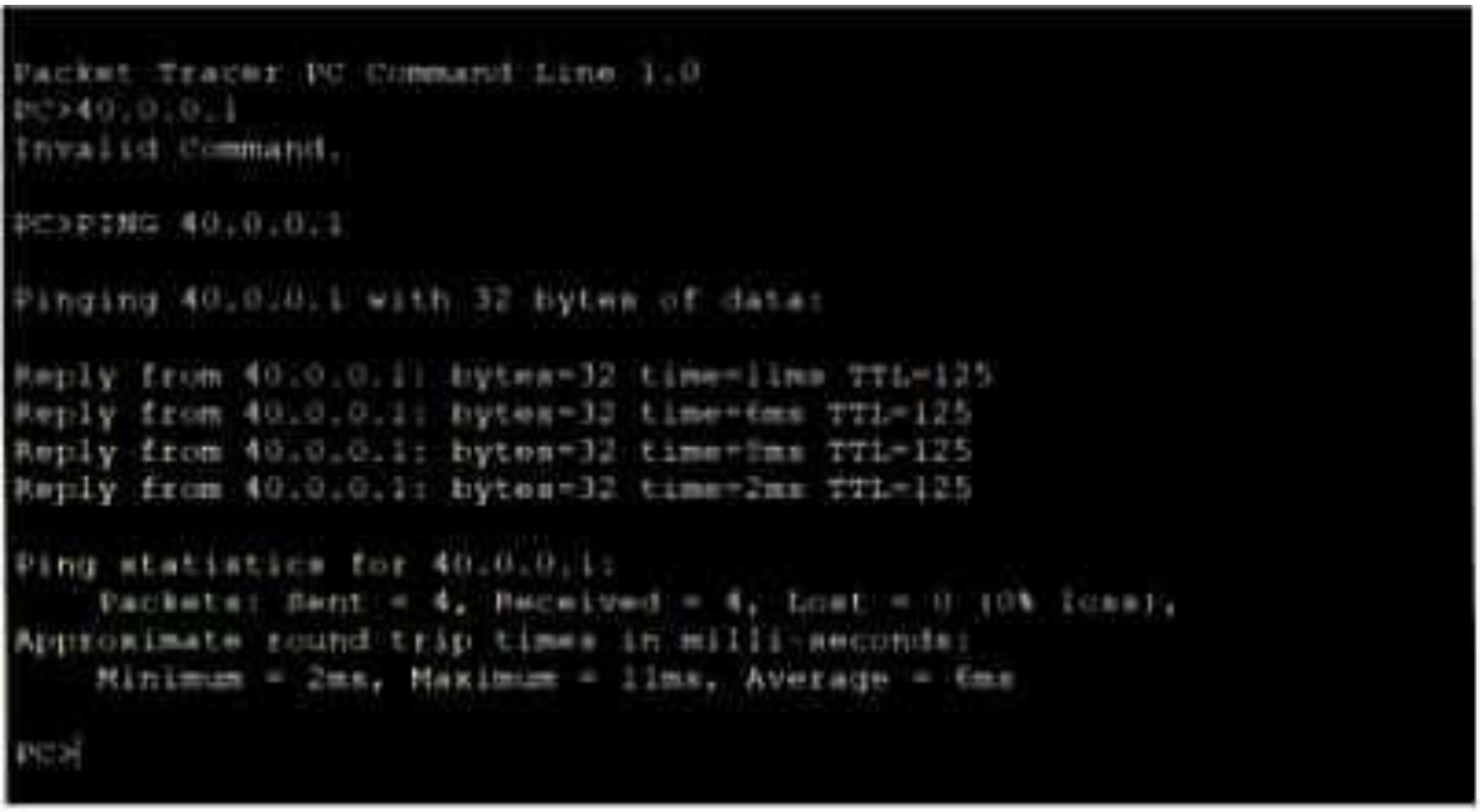
The above message signifies gateway 10.0.0.10 does not know where to redirect the packet to.

Request timed out :

On successful transmission from source to destn an acknowledgement is sent from the destination host to source host & in the form of ICMP packets. If the acknowledgement ICMP message does not reach source a 'Request timed out' message is shown. It may be due to packet loss, physical issue in transmission or incorrect gateway assignment





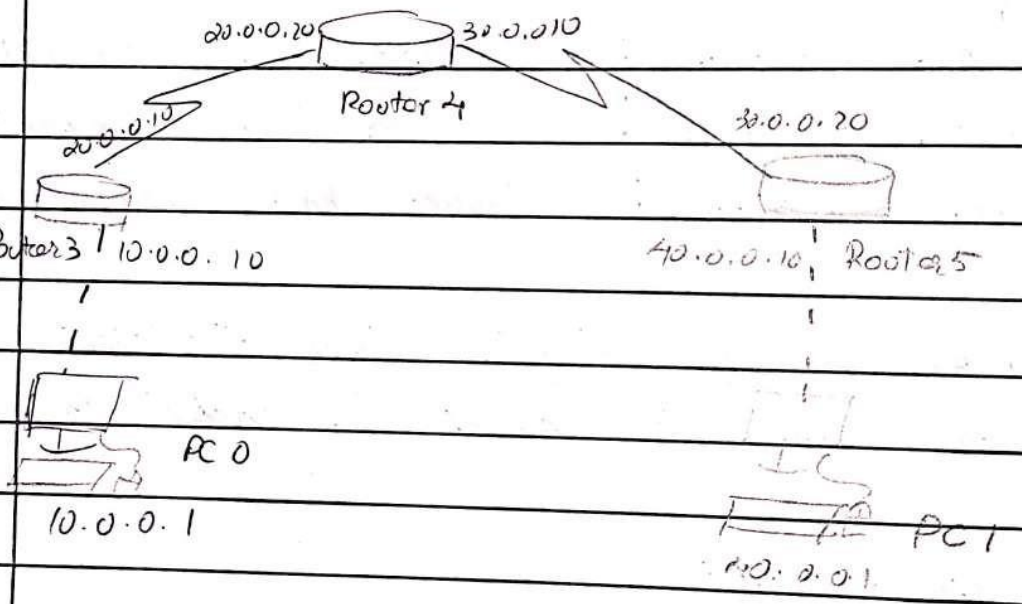


Configuration of default route and static route.

Aim:

Configure default route and static route to the router.

Topology



Procedure:

1. Add two end devices and three routers to the workspace
2. Connect routers through serial DTE cable and end devices through copper cross over cable
3. Assign IP addresses ~~and~~ to the end devices

   End device 1 : 10.0.0.1

   End device 2 : 40.0.0.1

④ Configures gateways through the following CLI command
ⓐ Enable
ⓑ config t
ⓒ interface <port>
ⓓ ip address <ip address> <subnet mask>
ⓔ no shut
ⓕ exit

⑤ Using command line interface:
ip route 0.0.0.0 0.0.0.0 0.0.0.0 <destination ip>

⑥ Ping from one end device to another
Ex: Ip route 0.0.0.0 0.0.0.0 0.0.0.0 40.0.0.1
Result
Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1 : bytes=32 time=2ms TTL=125
Reply from 40.0.0.1 : bytes=32 time=9ms TTL=125
Reply from 40.0.0.1 : bytes=32 time=2 ms TTL=125
Reply from 40.0.0.1 : bytes=32 time = 10 ms TTL=125

Ping statistics for 40.0.0.1:
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip time in milliseconds:
Minimum = 2ms, Maximum = 10ms, Average = 5 ms.

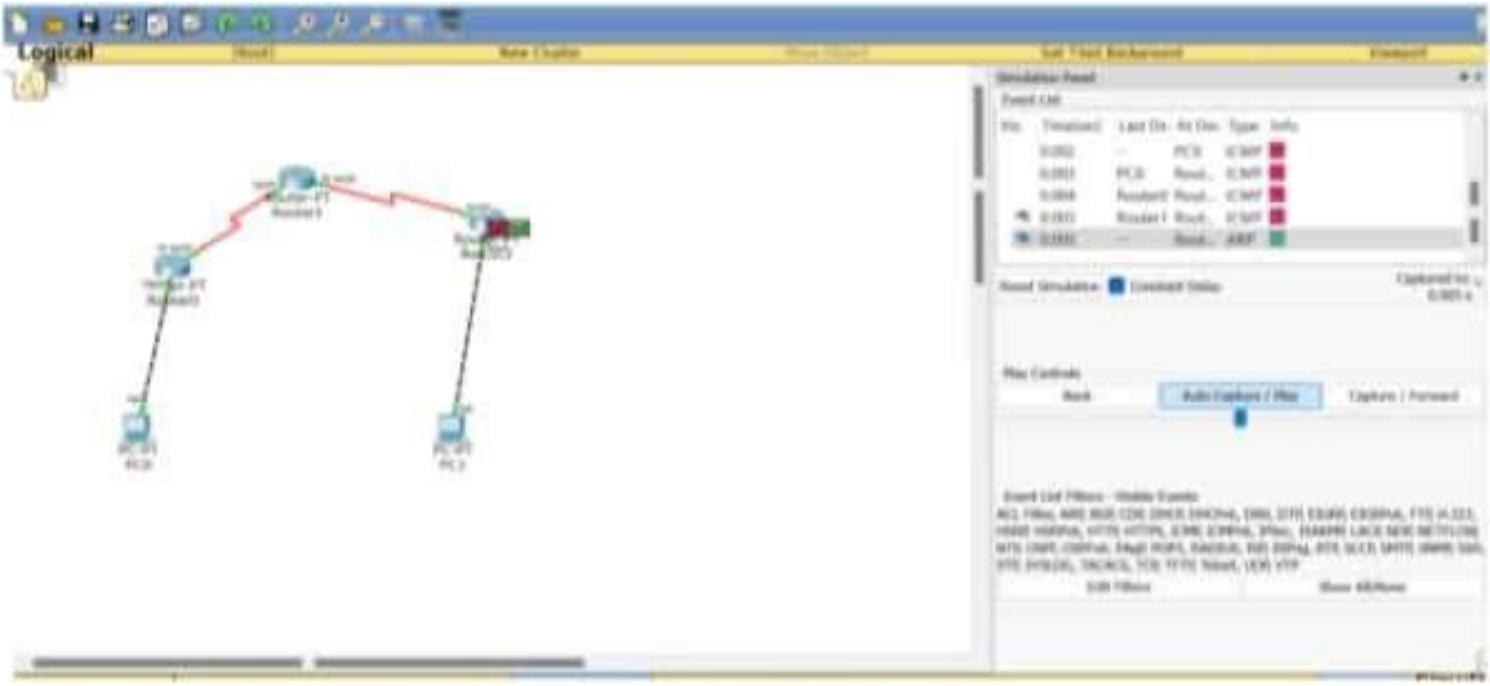
Observation

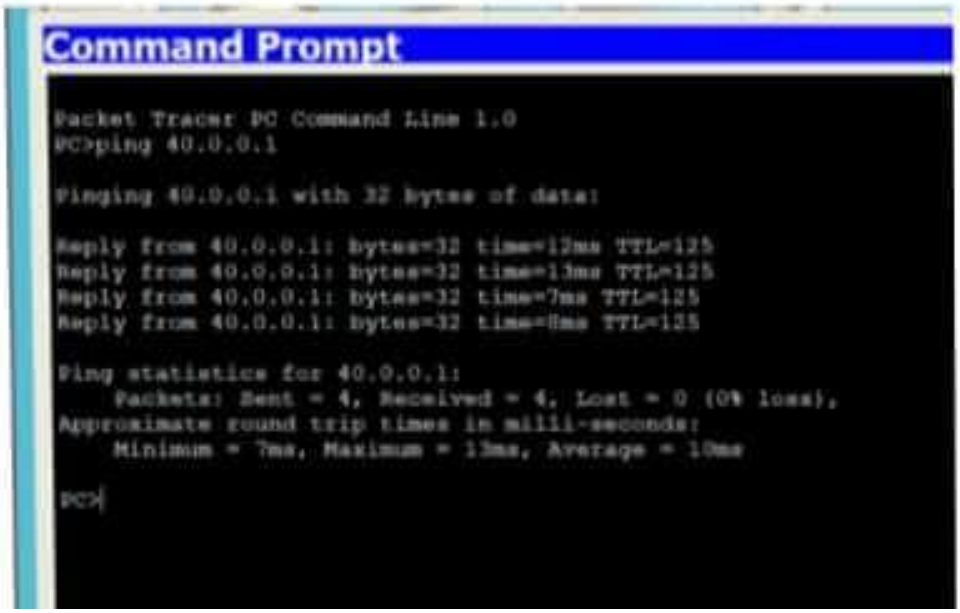
→ Routers are seen to connect 2 different networks together.

→ If a router has only one path to traverse Default Routing to send packets of any destination, to its adjacent device. Router 0 & Router 1

End devices send the packet to the routers which then redirect it to the appropriate destination.

29/7/23





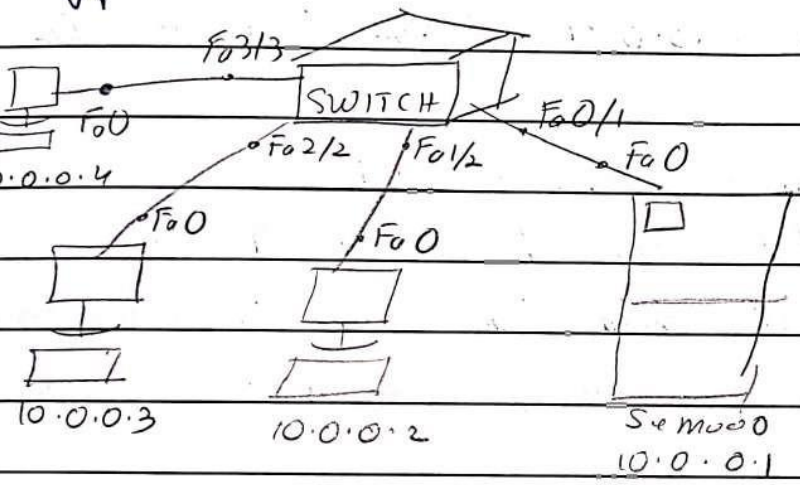
Date : 13/7/23

Lab 4

Aim: Configure DHCP within a LAN & outside LAN

① Within a LAN

Topology



Procedure

- Create a LAN network (10.0.0.0) by selecting 3 PC's & server and connect them to a switch.
- Set the sever's IP address to 10.0.0.1 & set the default gateway to 10.0.0.20
- Set the server to DHCP mode (services → DHCP → Service ON)
- Put down the gateway & the start IP address (10.0.0.2)
- Change all the other PC's IP configuration to DHCP.

Date :

Result

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 =1ms TTL=128

Reply from 10.0.0.4 bytes =32 time-0 ms TTL=128

Reply from 10.0.0.4 bytes=32 time-0ms TTL =128

Reply from 10.0.0.4 bytes=32 Time=0 ms TTL =128

Ping statistics for 10.0.0.4

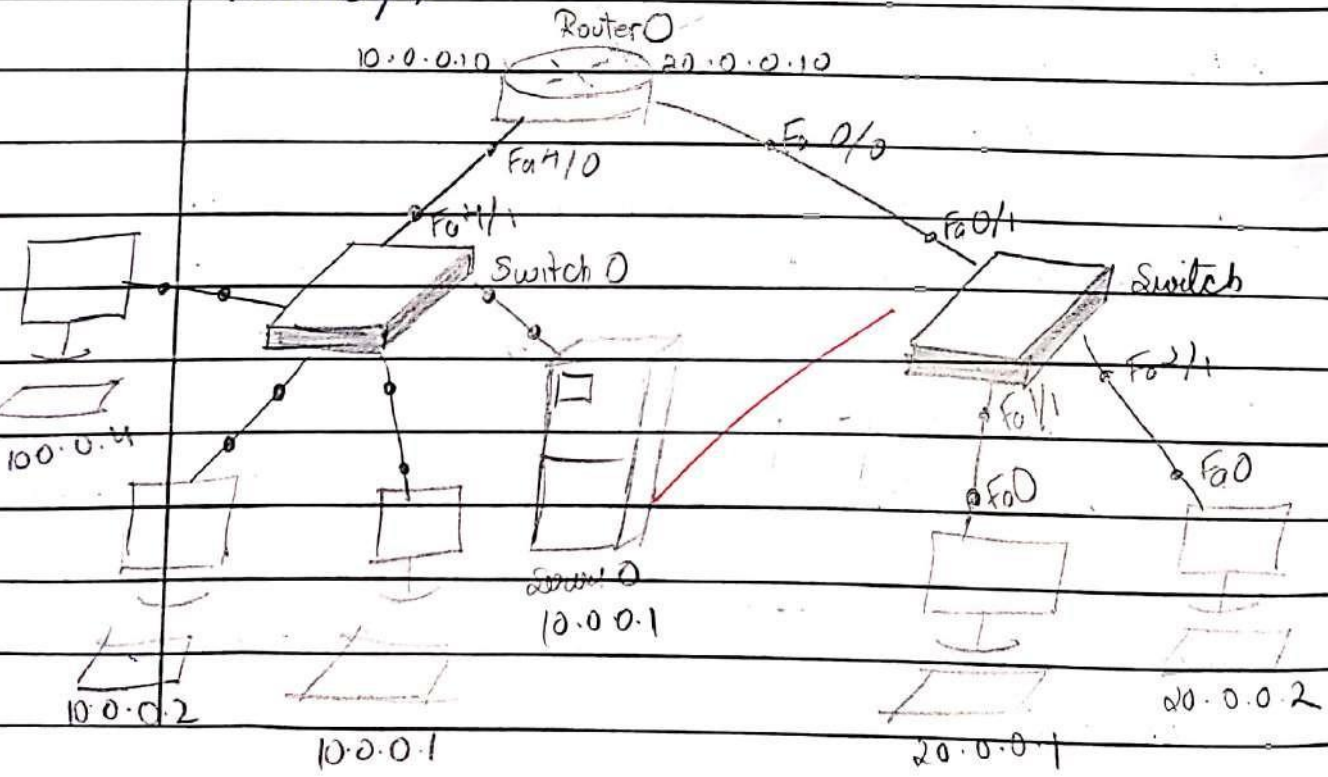
Packets : sent = 4, Recieved = 4, Lost =0 (0% loss)

Approximate round trip times in millisecond

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

② Outside of LAN

Topology :



Date :

Procedure :

- Follow the same steps as in case of inside LAN, by creating a 10.0.0.0 network with the server IP address - 10.0.0.1 and the gateway 10.0.0.10
- Create another network with 2PC's and a switch and connect the 2networks using a Router.
- Configure the router to connect the 2 networks through the gateway
  - > enable
  - > config t
  - > interface fa 4/0
  - > ip address 10.0.0.20 255.0.0.0
  - > no shut
  - > exit
  - > interface fa 0/0
  - > ip address 20.0.0.20 255.0.0.0
  - > no shut
  - > exit
- Go to the services of server 0 and set another DHCP Pool gateway (default) to 20.0.0.10 and then in the CLI commands of Router set the server as a ip-address-helper.

The following are the 2 pools

Pool Name	Default Gateway	DNS server address	Start IP	Subnet Mask	Max. User
ServerPool	10.0.0.10	0.0.0.0	10.0.0.2	255.0.0.0	512
ServerPool	20.0.0.10	0.0.0.0	20.0.0.1	255.0.0.0	512

> config t
> ~~fg~~ interface fa 0/0
> ip helper address <server ip-address>
> no shut
exit

Ping from 10.0.0.2 to 20.0.0.2

Result

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=1ms TTL=127

Ping statistics for 20.0.0.2
Packets: Sent = 4 Received = 3 Lost = 1 (25% Loss)
Approx time (in Milliseconds)
Minimum = 0ms, Maximum = 1ms, Average = 1ms

Date :

Observation:

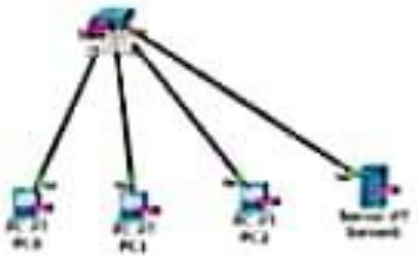
- The DHCP helps manage allocation of IP address to end users.
The device wanting to access a network gets an IP address allocated dynamically to it by the user.

The allocated IP address is taken back, when shutdown.

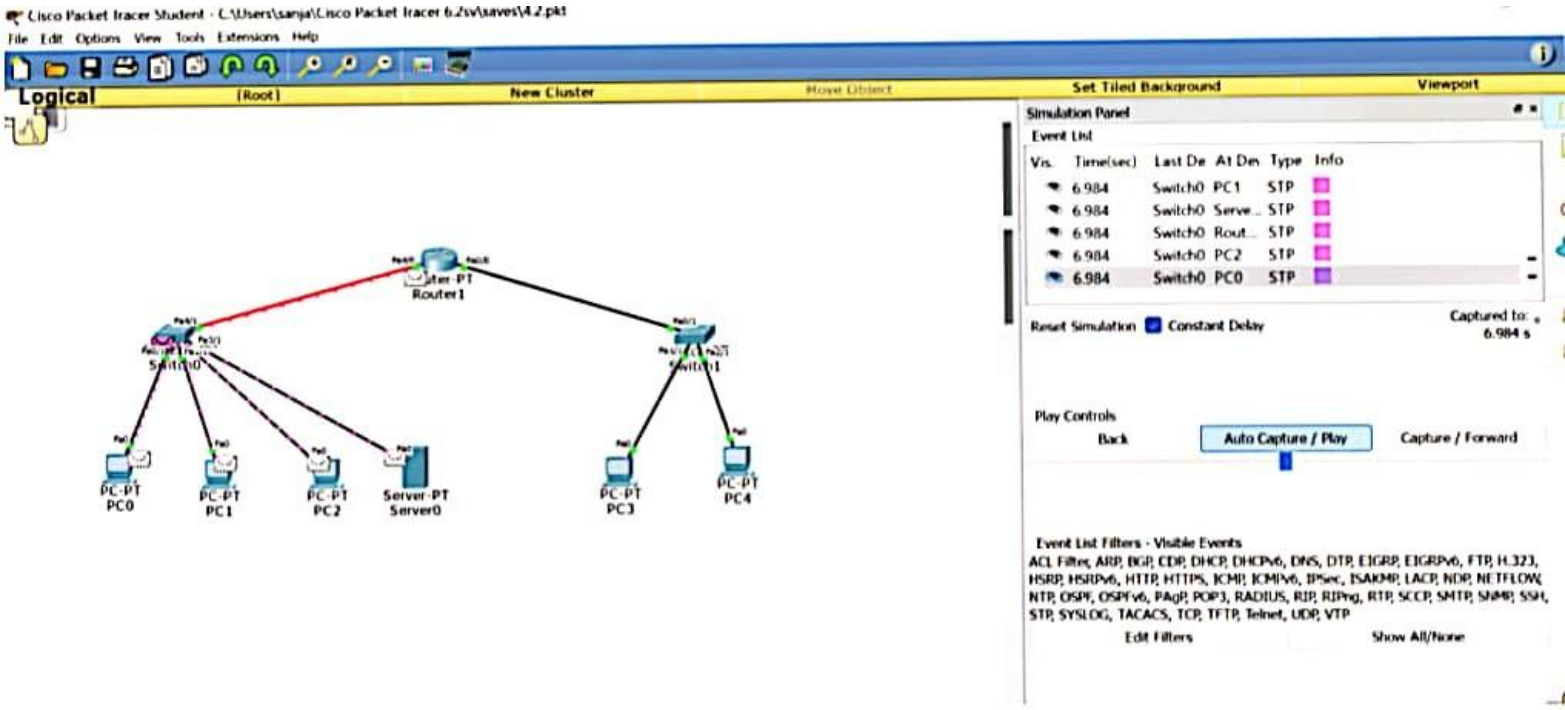
- If the requesting device is outside the LAN (case 2), then the servers ip address must be assigned to the router as the "address helper" so that it can automatically, configure that device's IP address

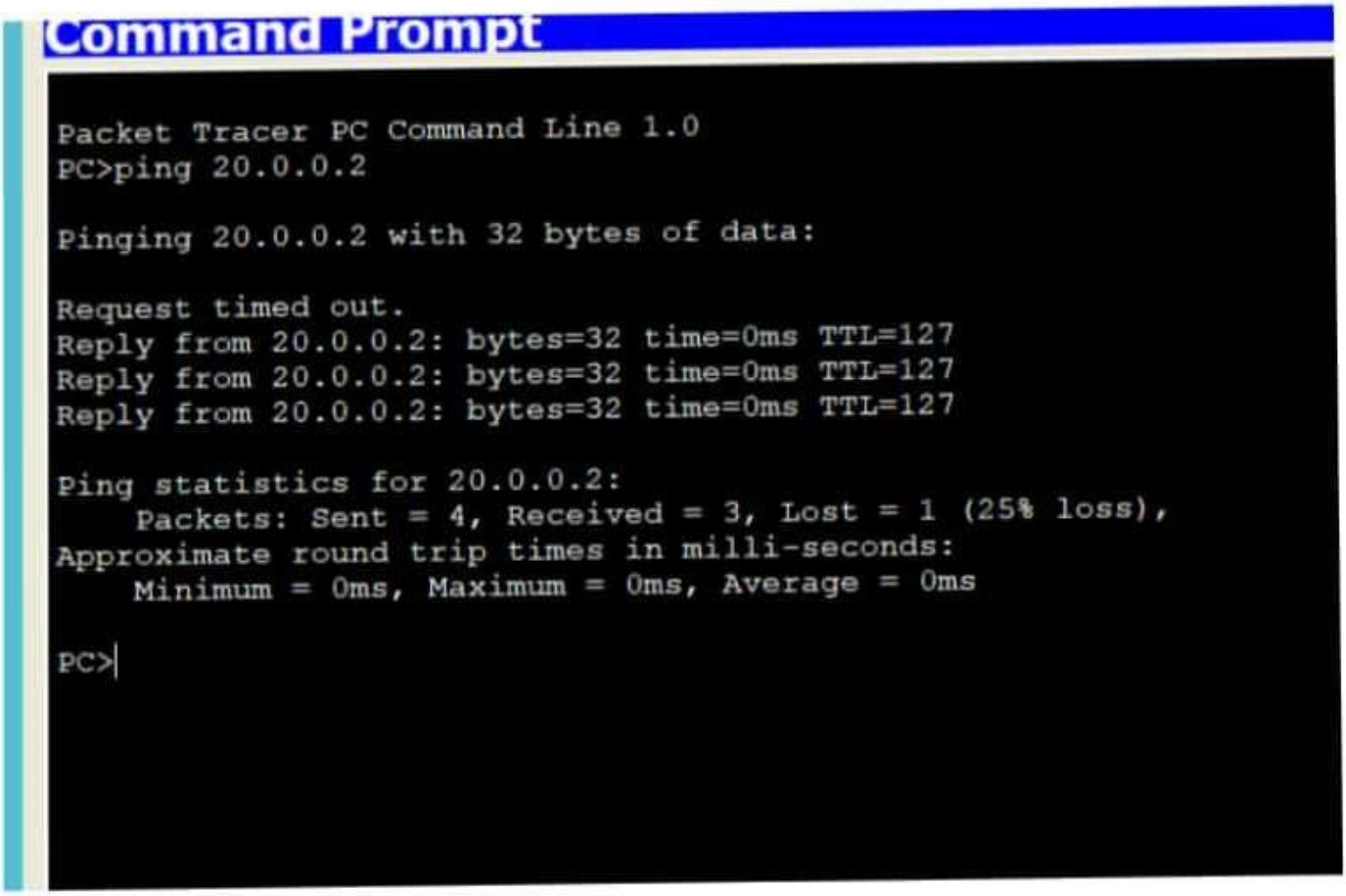
- When playing outside the current network at first it shows "Request timed out" as the router takes time to find the correct.

- We assign another server Pool in the Server's services so that the server knows the gateway to target and the starting ip address to assign to the devices of a different network.



```
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=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

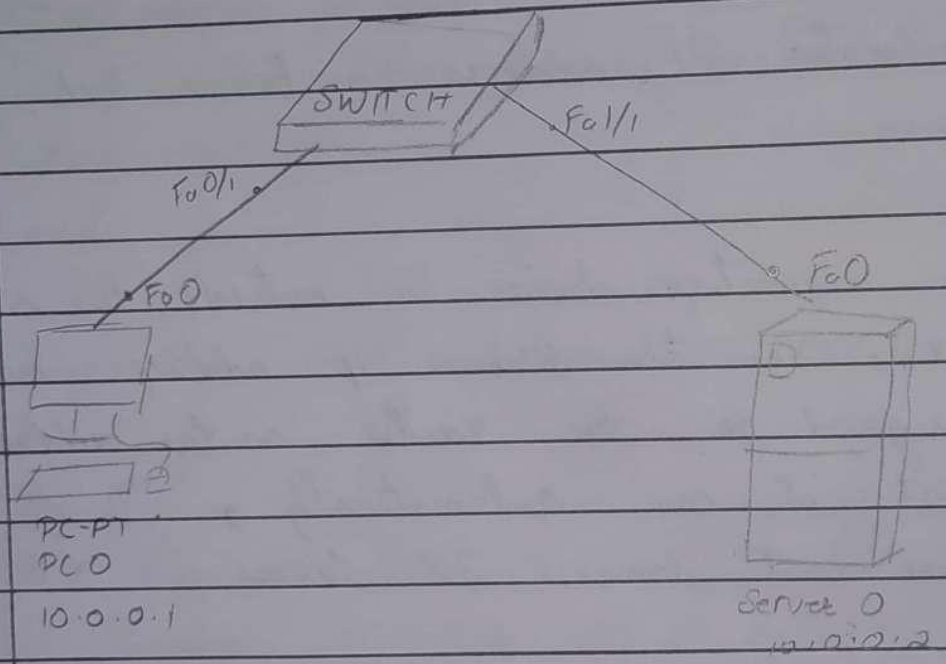




Exp 5

Aim: Configure Web Server, DNS within a LAN

TOPOLOGY



PROCEDURE:

1. Create a lan network by selecting a PC, server and connect them to a switch
2. Set the PC's IP address to 10.0.0.1 and the server's IP address to 10.0.0.2
3. In the services page of the server edit the index.html and make appropriate changes as required in Html format. The content should be details of the resume and

then save it.

7. Go to DNS services Page and name it Aryan.com and then add the address as 10.0.0.2
8. Go to Desktop in the PC and select web browser and type Aryan.com in the URL and click GO, the resume appears.

Result:

RESUME

NAME: Aryan

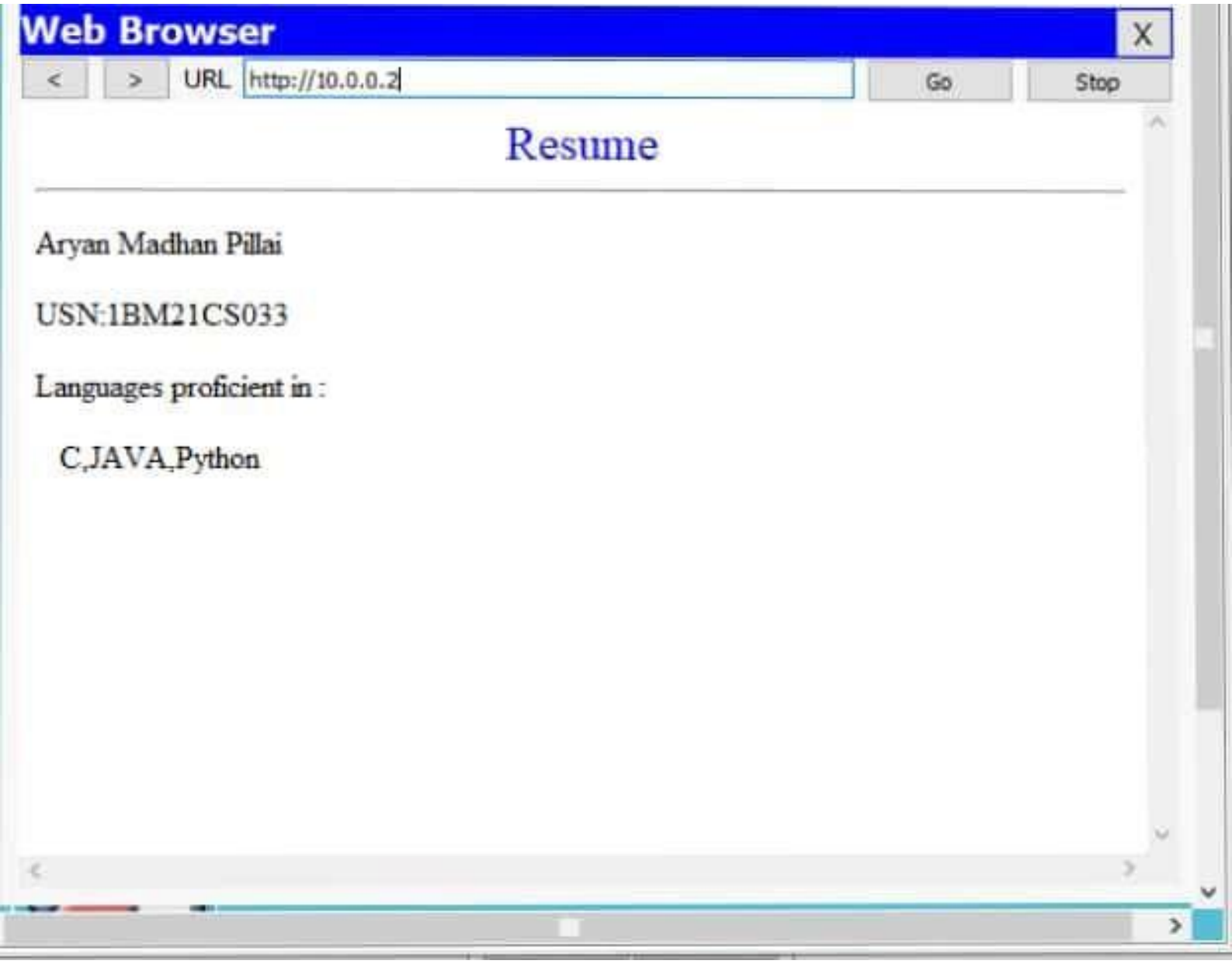
CONTACT: 8888888888

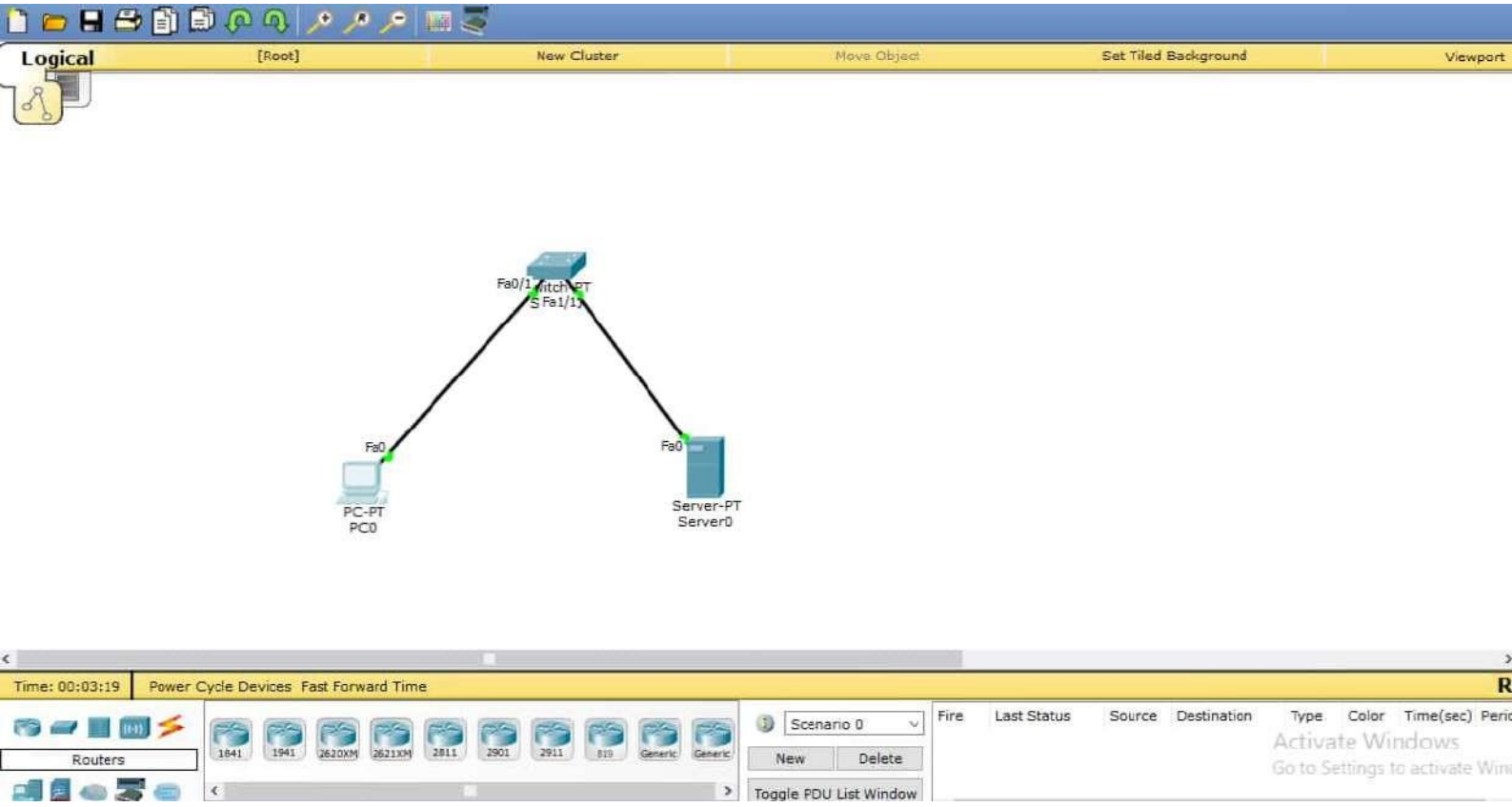
EMAIL: org@gmail.com

IMAGE

Observation:

On entering the domain name on web browser, we obtain stored web page. Domain name services translates domain names to their respective IP addresses, enabling system communication even when user does not know the IP address.

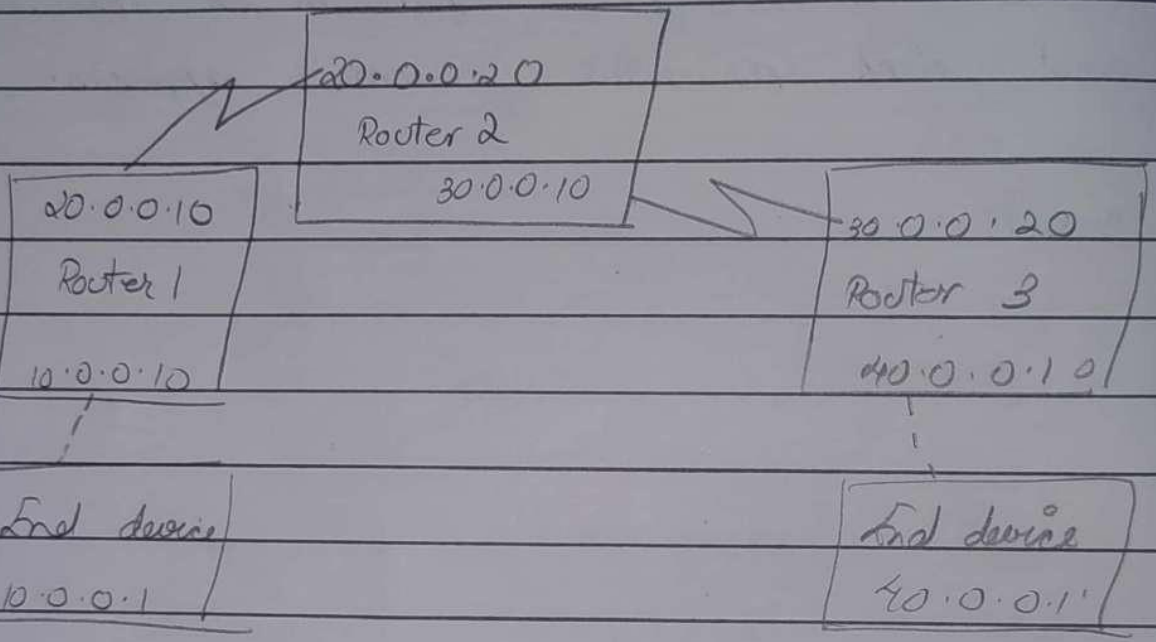




Exp-6

Aim: Configure RIP routing protocol in routers.

Topology:



Procedure:

① Connect 3 routers and 2 end devices

② Design IP addresses to both end devices. Design IP addresses to all routers. Use the following commands

Router > enable
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ipaddress 100.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit.

Date :

④ Set gateways to end devices
End device 1 : 10.0.0.10
End device 2 : 10.0.0.10

⑤ Assign routers to all routers and in the configure mode use the following commands.
Router (config)# router rip
Router (config-router)# network <network address>
Router (config-router)# network <network address>

For router 1,
Router (config)# router rip
Router (config-router)# network 10.0.0.0
Router (config-router)# network 20.0.0.0

⑥ Ping end devices to test connection

Result :

> ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data :

Request timed out.
Reply from 10.0.0.1 : bytes = 32 time = 5ms TTL = 125
Reply from 10.0.0.1 bytes = 32 time = 5 ms TTL = 125
Reply from 10.0.0.1 bytes = 32 time = 21 ms TTL = 125

Ping statistics for 10.0.0.1 :
Packets : Sent = 4, Recieved = 3 lost = 1 (25% loss)

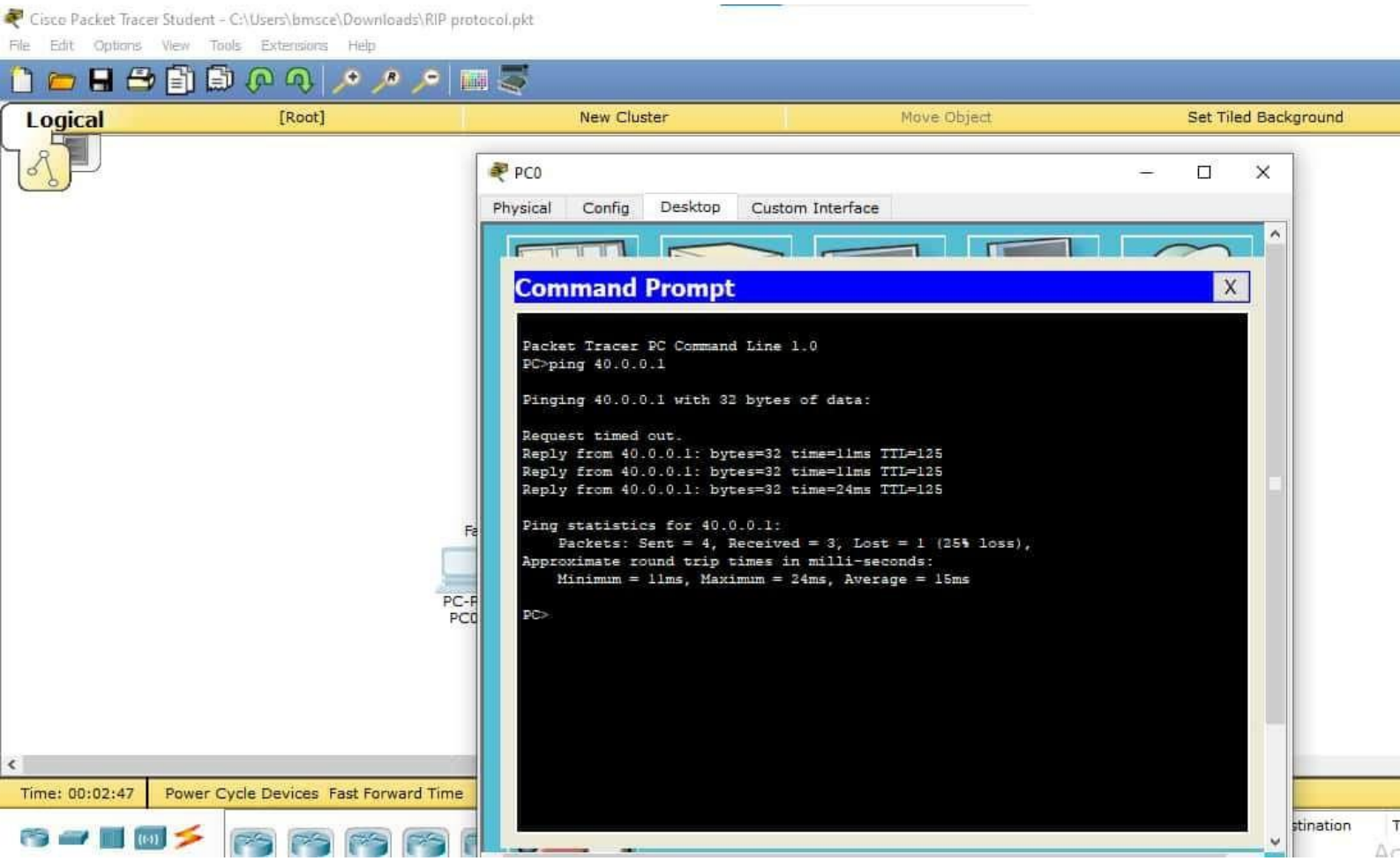
Date :

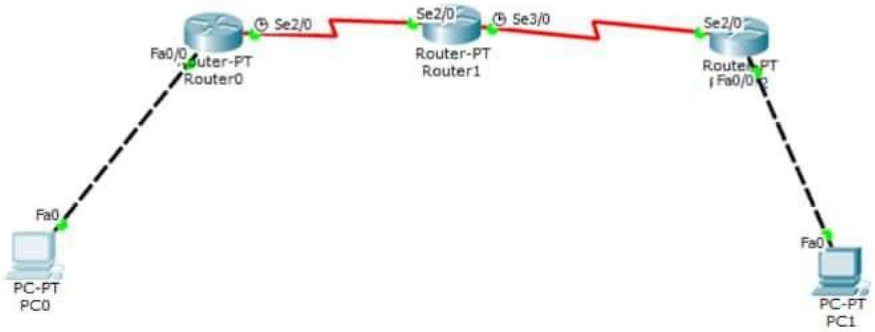
Approximate round trip times in milli-seconds

Minimum = 2ms, Maximum = 21ms, Average = 10ms

Observation:

Routing Information Protocol is a dynamic routing protocol that uses hop count as routing metric to find best path between source and destination networks. Hop count is the number of routers occuring in between source and destination networks



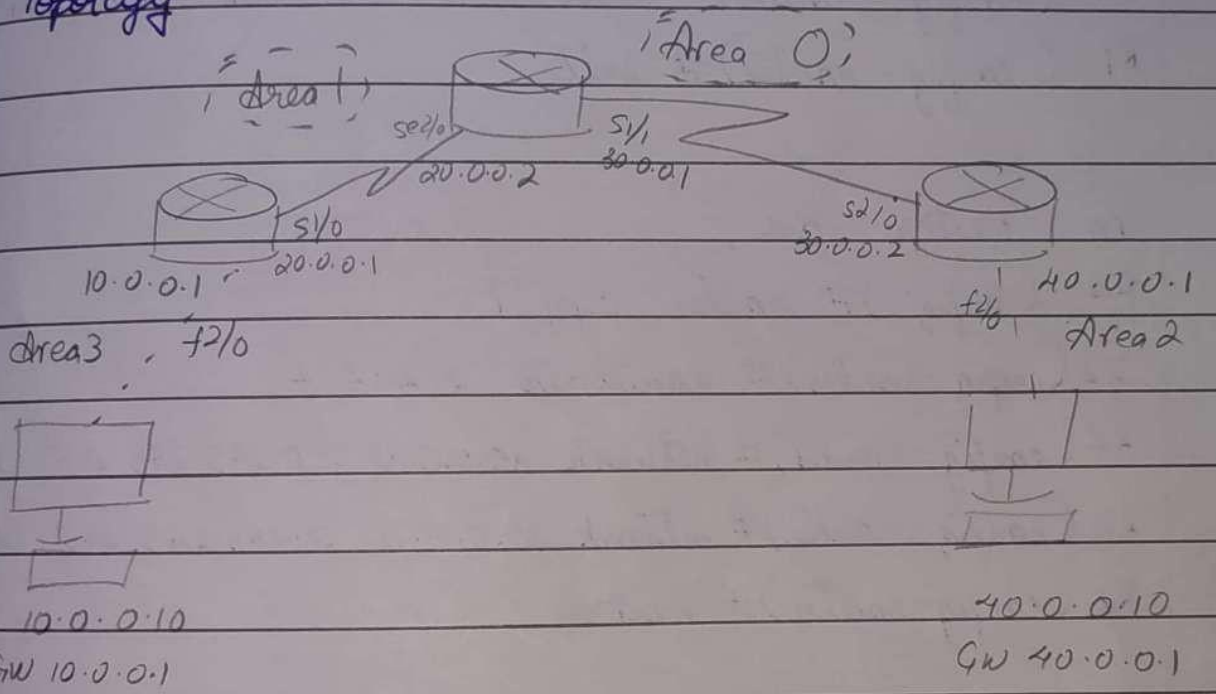


Date :

Expt

Aim: Configure OSPF routing protocal

Topology:



Procedure

1. Configure the PC's with IP address and gateway according to the topology
2. Configure each of the routers according to IP addresses in the topology
3. Encapsulation PPP and clock rate need to bee set as done in RIP protocal exp.
4. Set the router OSPF using the following commands:-

In Router 1 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
R1(config-router)# network 20.0.0.0 0.255.255.255 area 1
R1(config-router)# exit
```

In Raster 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 1
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
```

* Set the interface loopback

R1(config-if)# interface loopback 0
R1(config-if)# ip address 172.16.1.252 255.255.255.0
R1(config-if)# no shutdown

R2(config-if)# interface loopback 0
R2(config-if)# ip address 172.16.1.253 255.255.255.0
R2(config-if)# no shutdown

R3(config-if)# interface loopback 0
R3(config-if)# ip address 172.16.1.254 255.255.255.0
R3(config-if)# no shutdown

* 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.

* After this show IP route, it should show all 4

Result :

PC > ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data:
Request timed out
Reply from 40.0.0.10 bytes=32 time=10ms TTL=125
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=7ms TTL=125
Ping statistics for 40.0.0.10 :
Packets : Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip times in milli-seconds
Minimum = 2ms, Maximum = 10ms, Average = 7ms

Router 1 :

show ip route

O IA 10.0.0.0/8 [110/65] 20.0.0.1 00:00:11, serial 2/0
20.0.0.0/8 is variably subnetted, 2 subnets 2 masks
C 20.0.0.0/8 is directly connected, serial 12/0
C 20.0.0.1/32 is directly connected, serial 12/0
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 30.0.0.2/32 is directly connected, serial 3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2,
C 172.16.0.0/16 is directly connected, Loopback 0

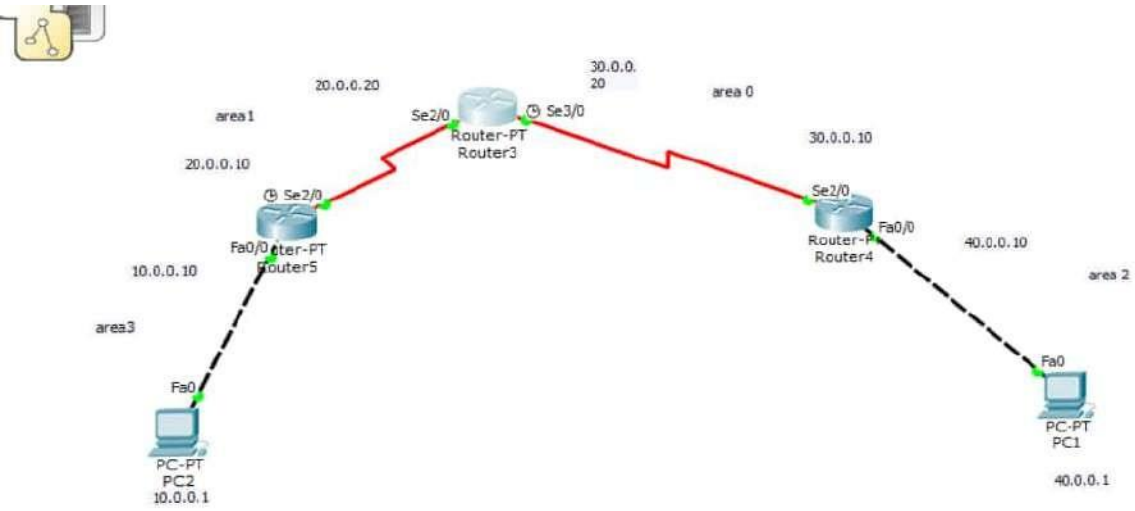
Date :

Observation :

- ~~RIP~~ ~~or~~ ~~Routing~~ ?

* OSPF is Open Shortest Path First. It is a protocol which finds the best routing path between source and destination router. It uses its own shortest path algorithm.
* Networks are divided into areas. Backbone (area 0) form core of the OSPF network. Other Network are connected to the backbone

31/3/22

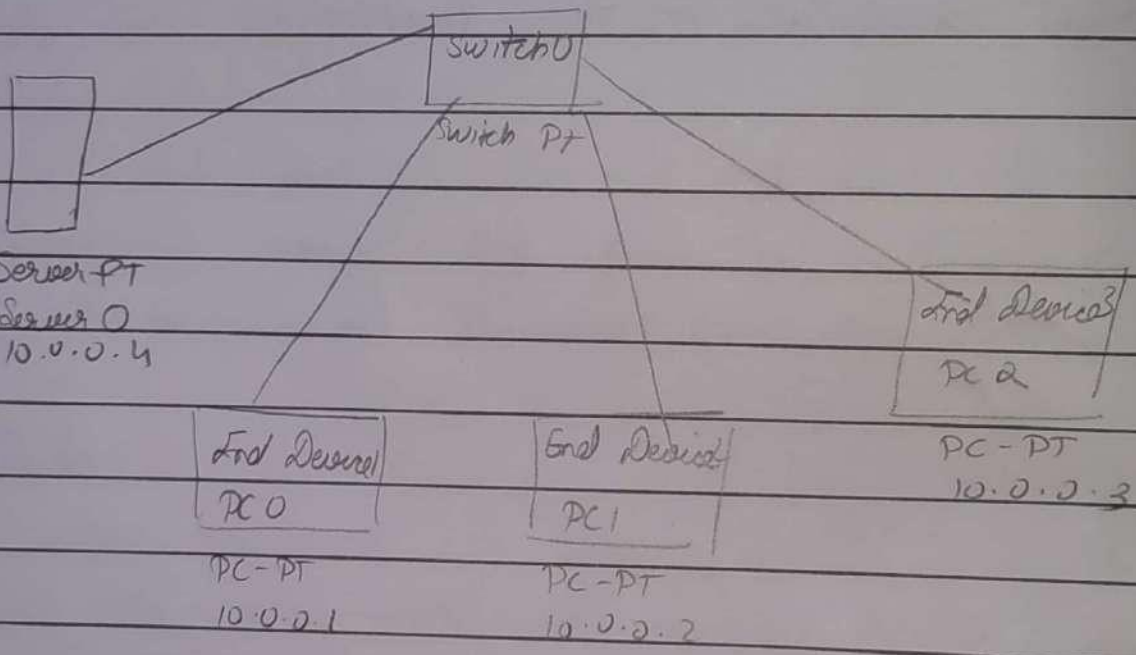


```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=20ms TTL=253

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 20ms, Average = 6ms

PC>
```

Exp - 8

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

Topology:



Procedure:

- Create a topology of 4 PC's and a server
- Assign IP addresses to all.
- Connect them through a switch.
- Use the inspect tool to click on a PC to see the ARP Table (arp command in command prompt - a)

- Initially the arp table is empty
- In CLI of switch → show mac address table can be given on an every transaction to see how the switch learns from transactions and build the adress table.
- Use the capture button to see the process step by step.

Result:

PC> arp -a

Internet Address	Physical Address	Type
10.0.0.2	000d.bd15.3a18	dynamic
10.0.0.3	00e0.8b7.b7e8	dynamic
10.0.0.4	0080.bc97.9a38	dynamic

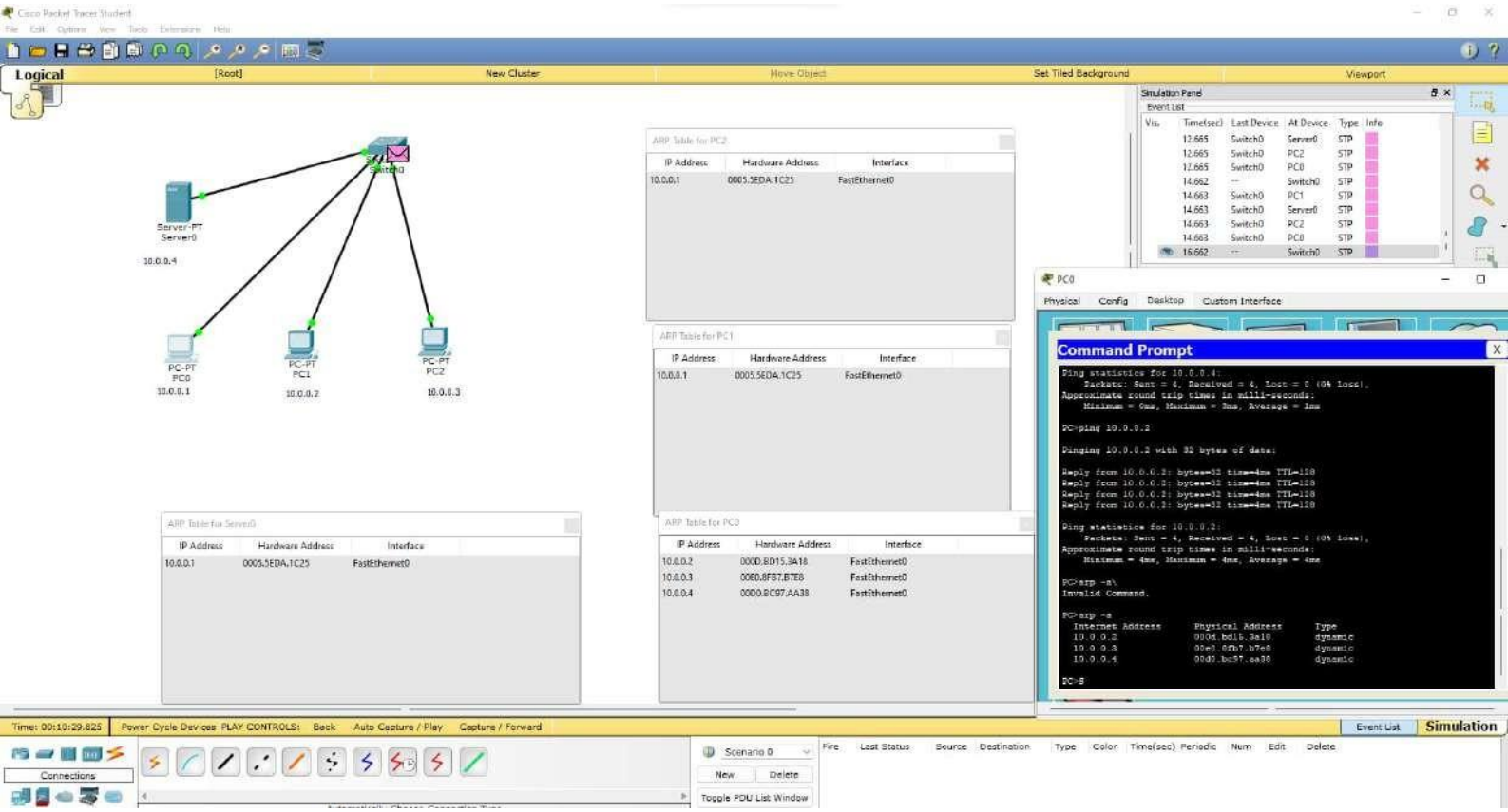
Mac Table for Switch 0

VLAN	Mac Address	Port
1	0005.5EDA.1C25	FastEthernet 0/1
1	000D.BD15.3A18	FastEthernet 2/1
1	0080.BC97.AA38	FastEthernet 1/1
1	00E0.8FB7.B7E8	FastEthernet 3/1

Date :

Observation:

- ARP is a protocol that uses the IP address to find out the physical address / MAC address.

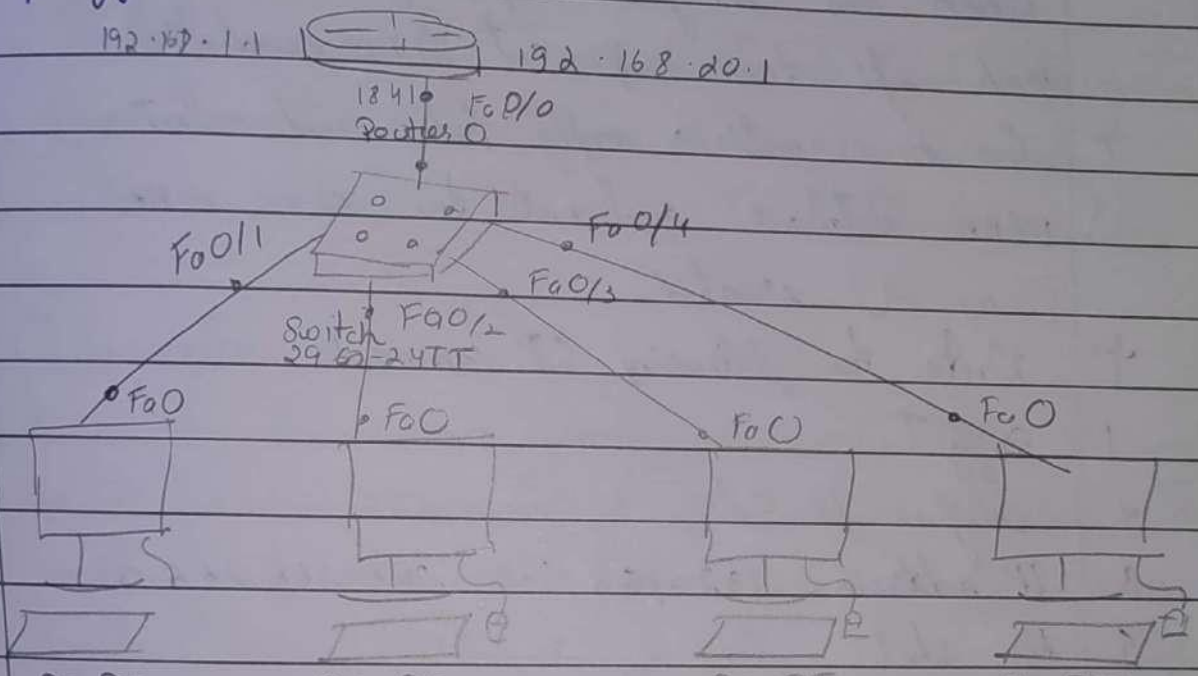


Date :

LAB 9

Aim - To construct a VLAN and make a PC communicate among VLAN

Topology



PROCEDURE :

- Create a topology as shown choose 1841 router and 2960-24TT switch
- Set the IP address of the router and 4PC's respectively, we use class C type addresses also set gateways
- In switch, go config tab and select VLAN database

Give any VLAN no like 2 and name as VLAN

- Select the interface fast ethernet 4/1 and make it trunk
- Next select the switches under 2nd interface which has interface 0/3 & 0/4. click on each of them and set VLAN number 2
- Go to router → config tab and select VLAN database and enter the name VLAN & no 2 created.
- Write the following CP commands

1. Config T
2. interface fa 0/0
3. IP address 192.168.1.1 255.255.255.0
4. No shut
5. Exit
6. Config T
7. interface fa 0/0.1
8. encapsulation dot1q 2
9. ip address 192.168.20.7 255.255.255.0
10. No shut
11. Exit

- Ping mess

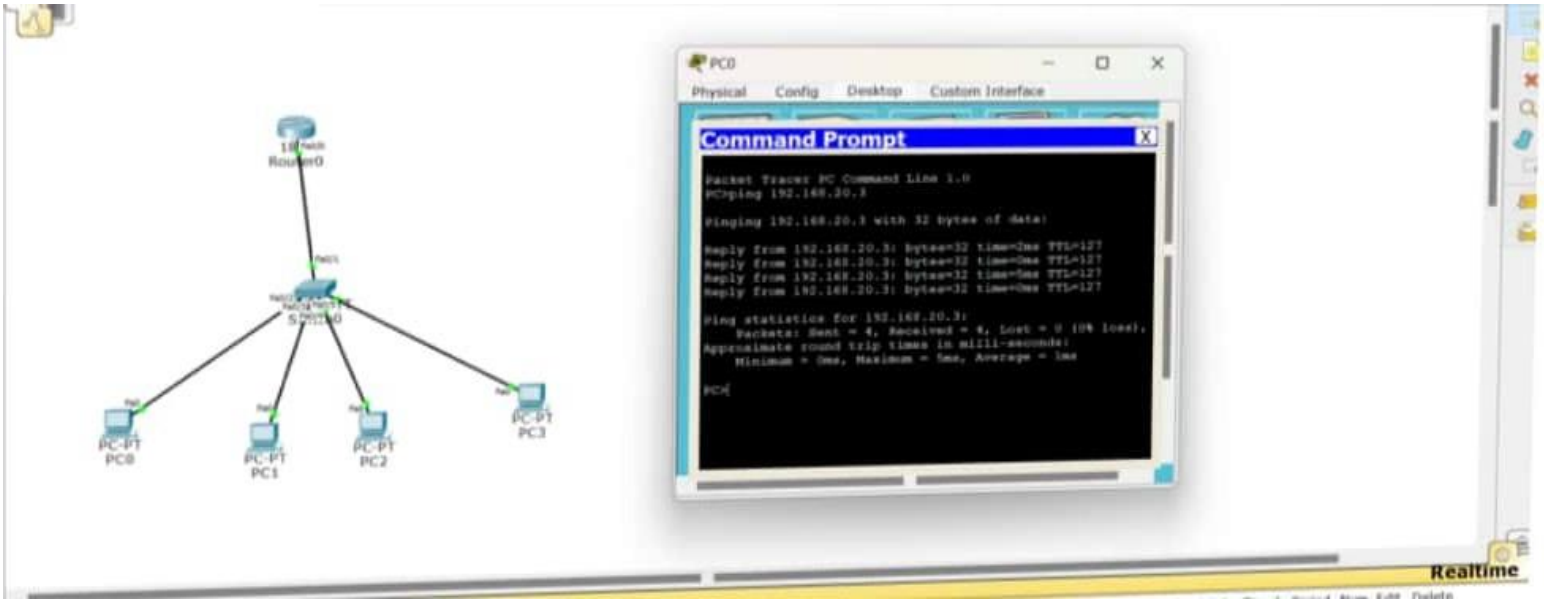
Packet Tracer PC command line 1.0
PC> Ping 192.168.20.3

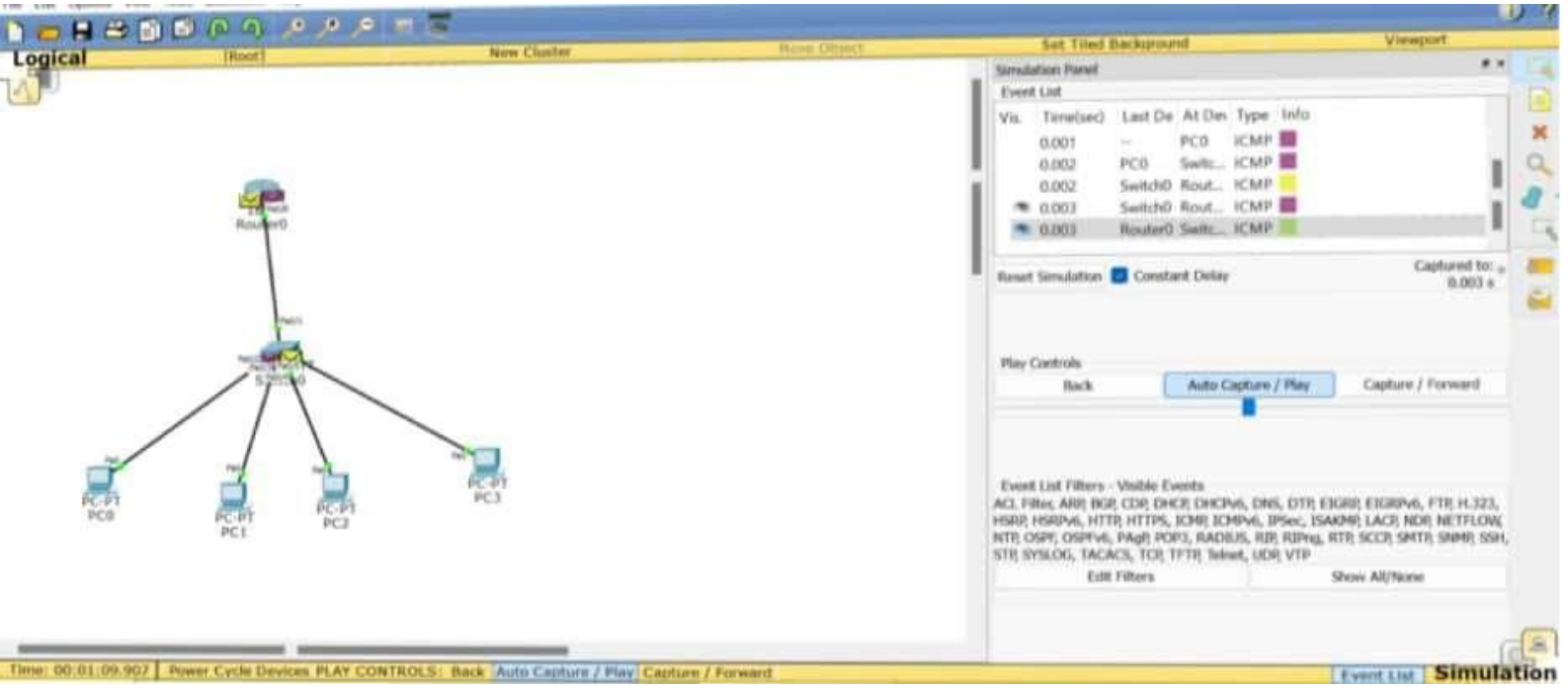
Date :

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 milliseconds
Minimum=0ms Maximum=5ms, Average=1ms

Observation:

- We can have one device one one VLAN & another on a VLAN connected to the same switch. They will only hear other broadcast traffic from within their VLANs as if they were connected to 2 switches
- Inter VLAN routing gives a flexible tool to logically subdivide their networks that has potential to enhance security & performance.



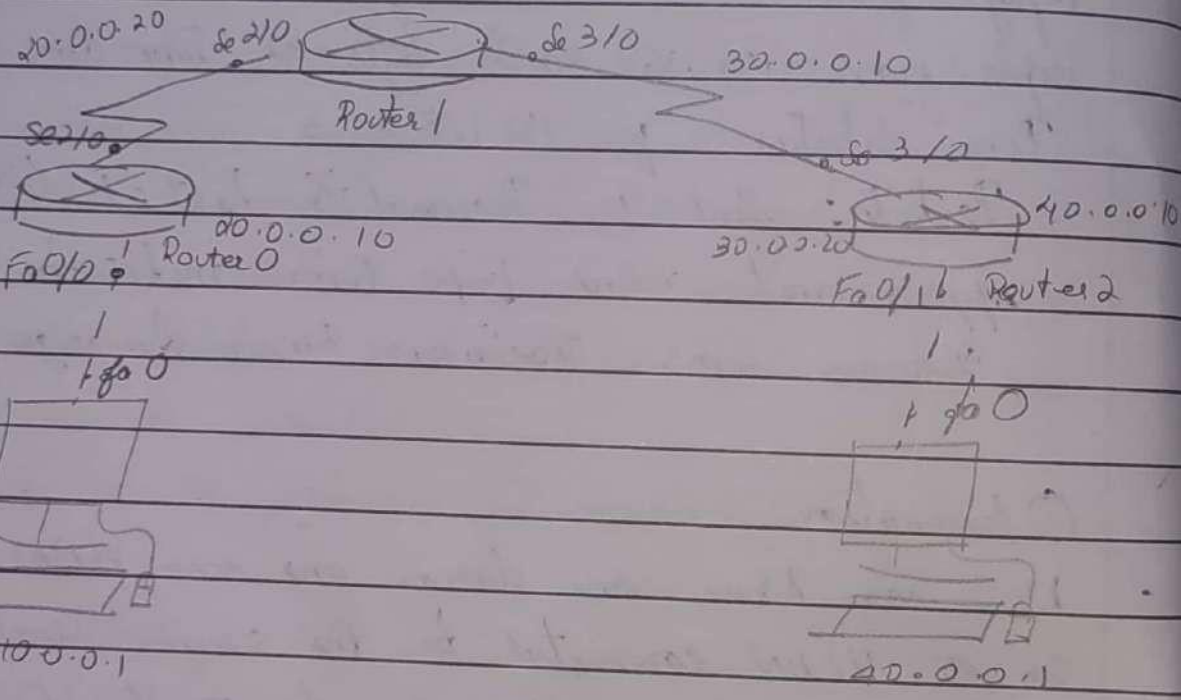


Date :

LAB 10

Aim:

Demonstrate TTL / life of a packet



Procedure:

- Create a topology as shown above with two PC's and 3 routers.
- Set the IP address and gateway for both PC's.
- Configure the routers either static / default routing way.
- In simulation mode send a simple PDU from one PC to another.
- Use capture button to capture every transfer
- Click on the PDU during every transfer to see the

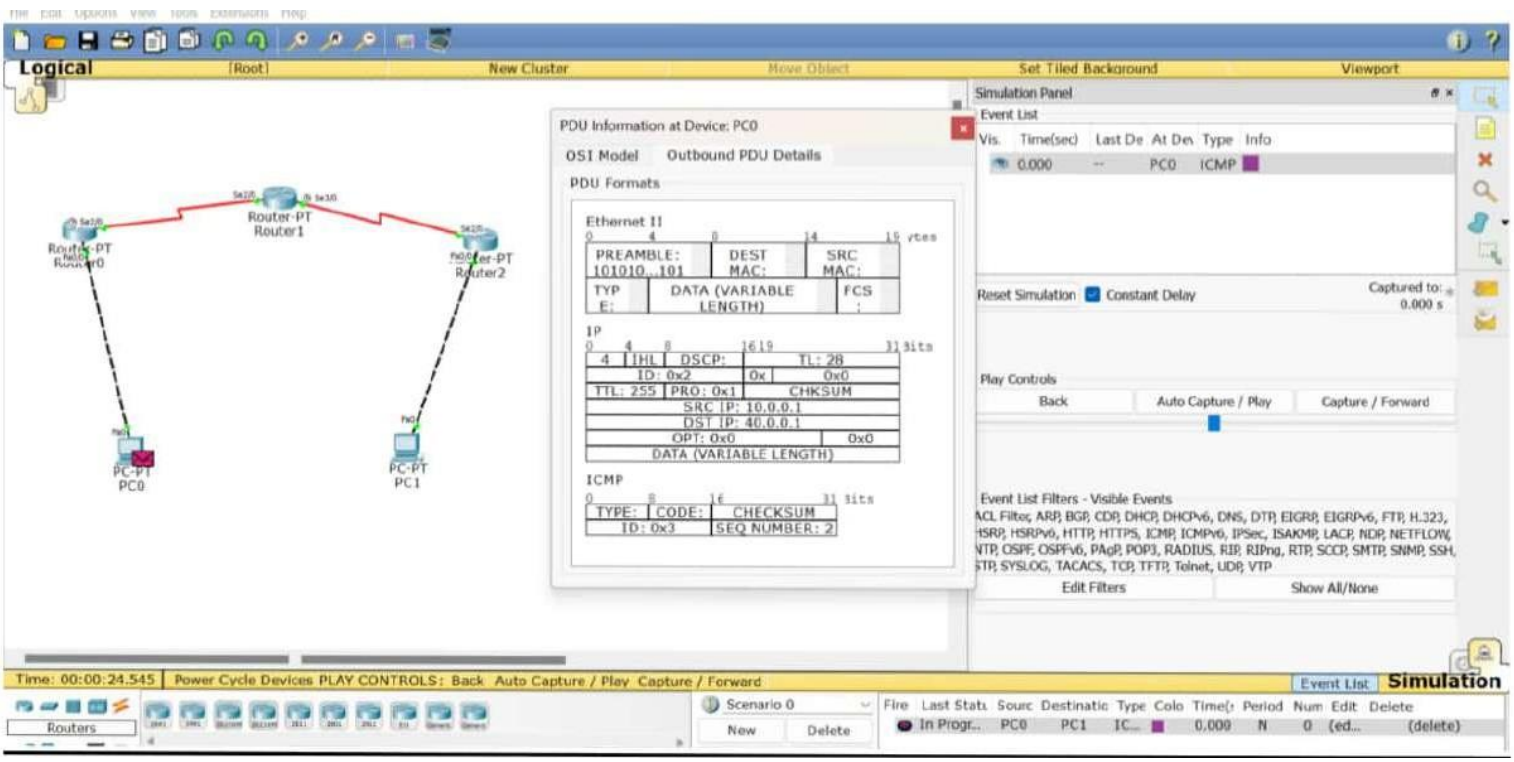
inbound & outbound PDU details.

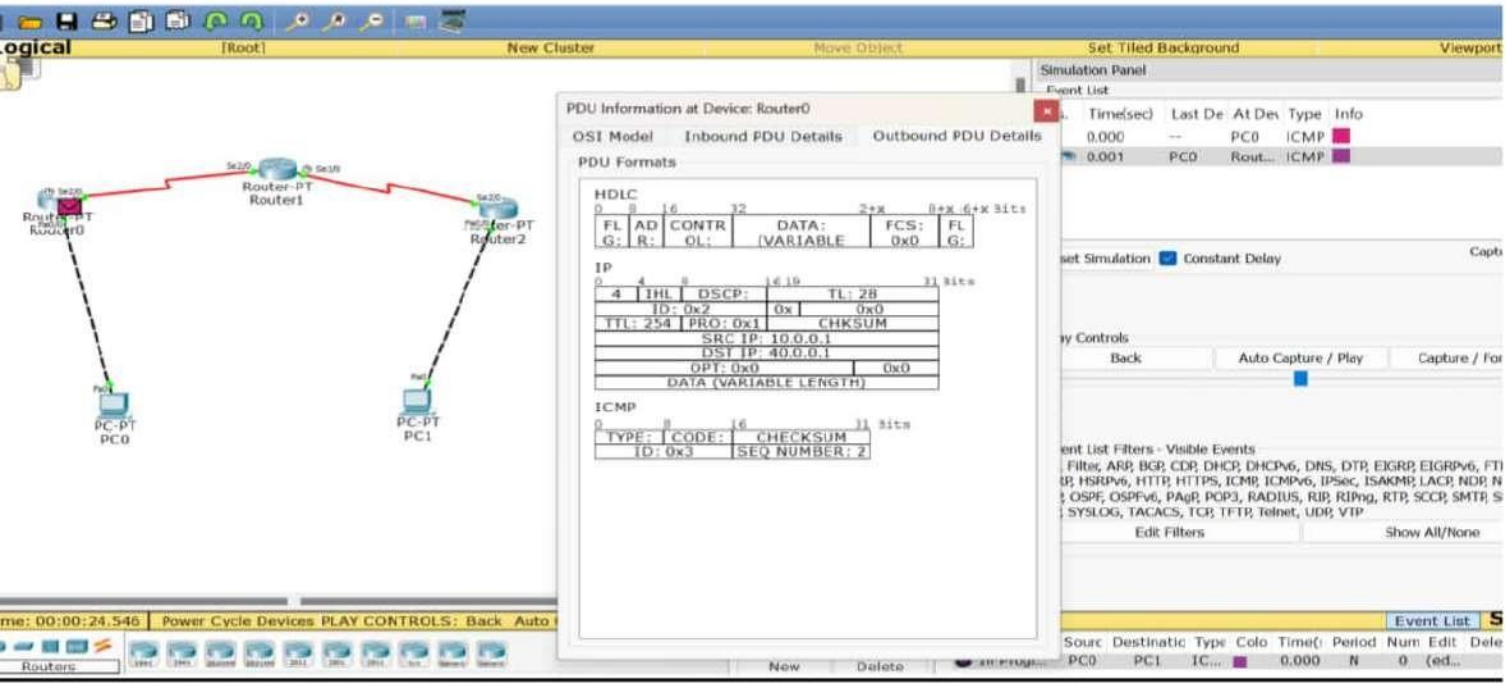
OUTPUT:

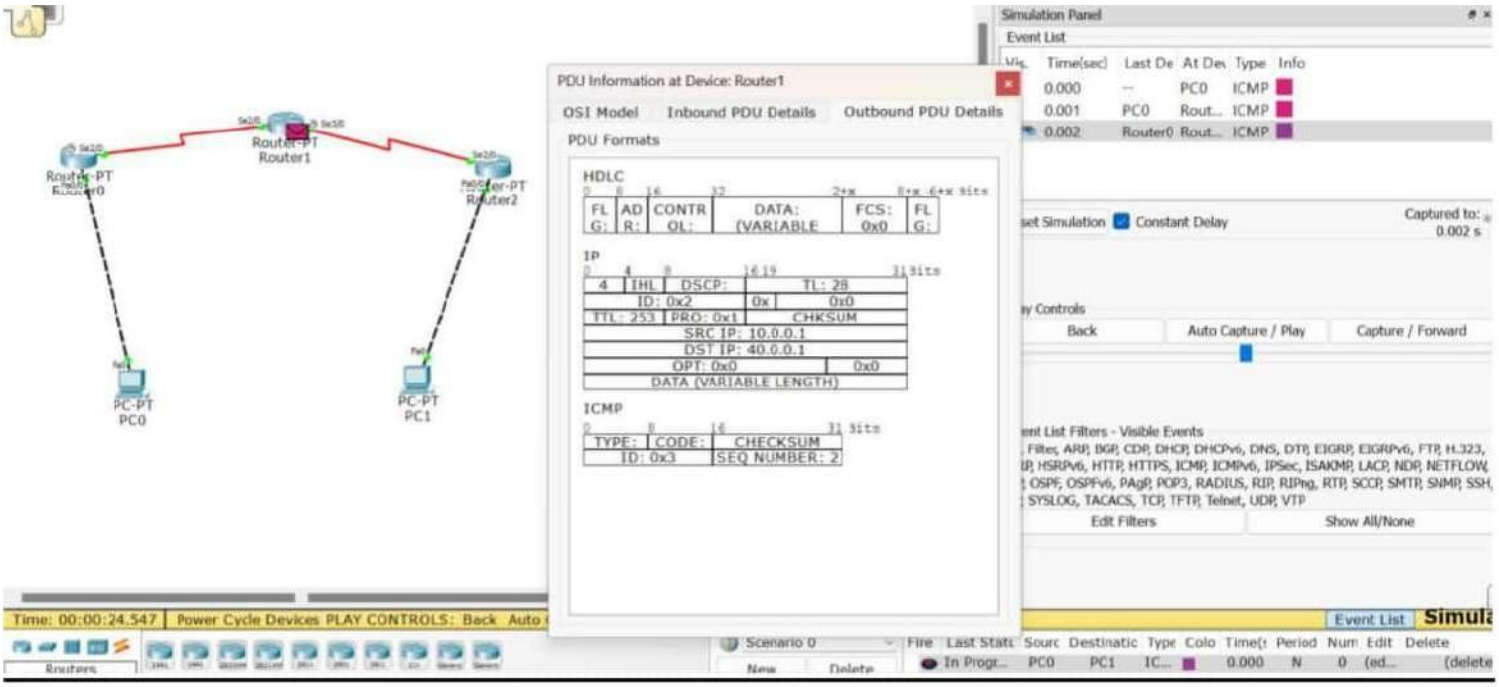
0	4	8	16	19	31
V4	HL	DSCP	TL: 28		
ID 0x6			0x	0x0	
TTL: 255		PRO: 0x1	CHKSUM		
SRC IP: 10.0.0.1					
DST IP 40.0.0.1					
OPT 0x0				0x0	
DATA					

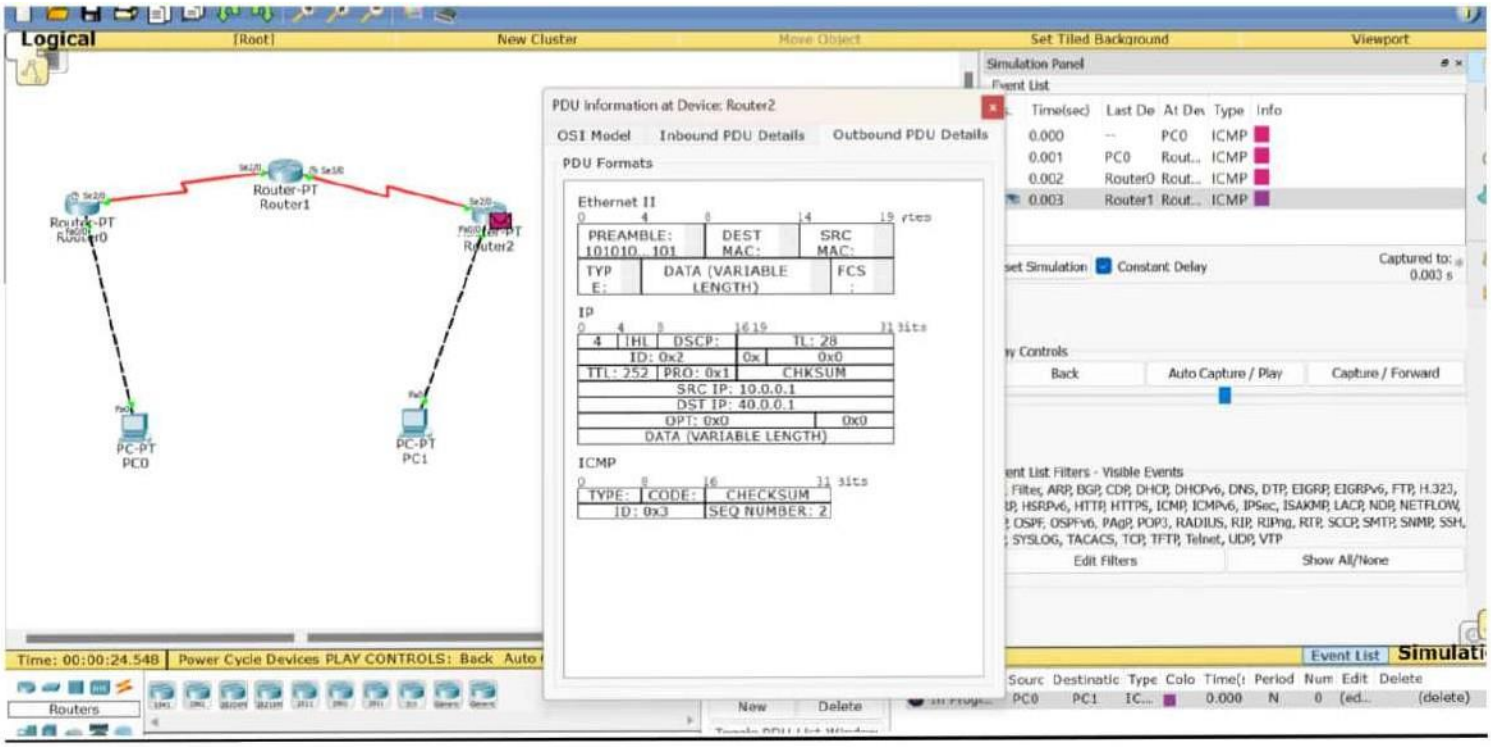
OBSERVATION:

* The no. of hops the packet travel before being discarded as TTL
* Datagrams TTL field is set by the sender & reduced by each router along the path to its destination.
* The router reduces TTL value by one while forwarding the packets.
* When the TTL value is 0, the router discards it & sends an ICMP message.







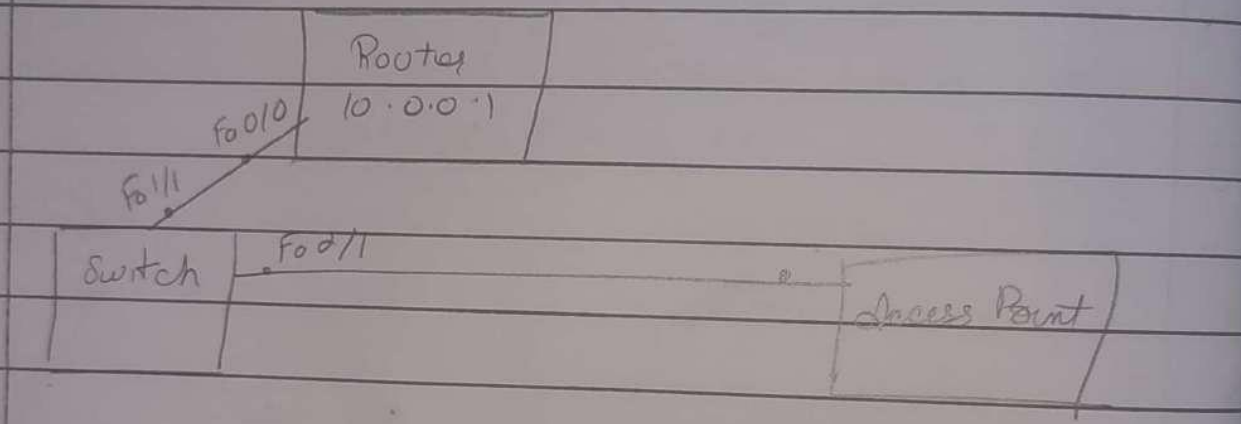


LAB 11

Aim:

To construct a WLAN and make the nodes communicate wirelessly.

Topology



Procedure:

1. Construct the above topology.
2. Configure PC & router 0 as normally done.
3. Configure access point 1 - Port 1 → SSID Name - WLAN
4. Select WEP & give any 10 digit hex key - 1234567890
5. Configure PC1 & laptop with wireless standards.

* Switch off the device. Drag the existing PT-HOST-NM-1DM to the component listed in LHS. Drag WMP300N wireless interface to the empty port. Switch on the device.
* In the config tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and gateway to the device.
* Ping from every device to every other device.

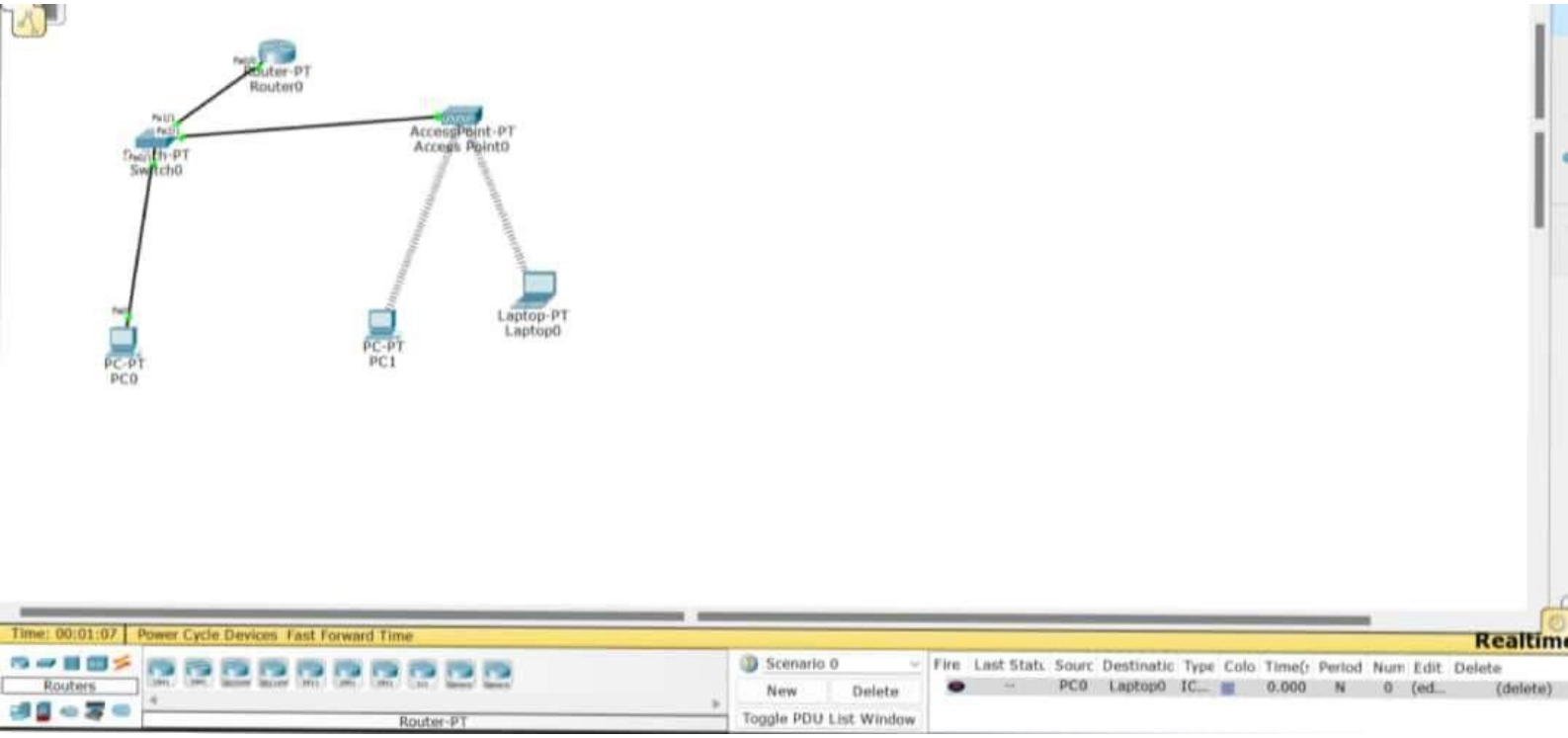
PING OUTPUT

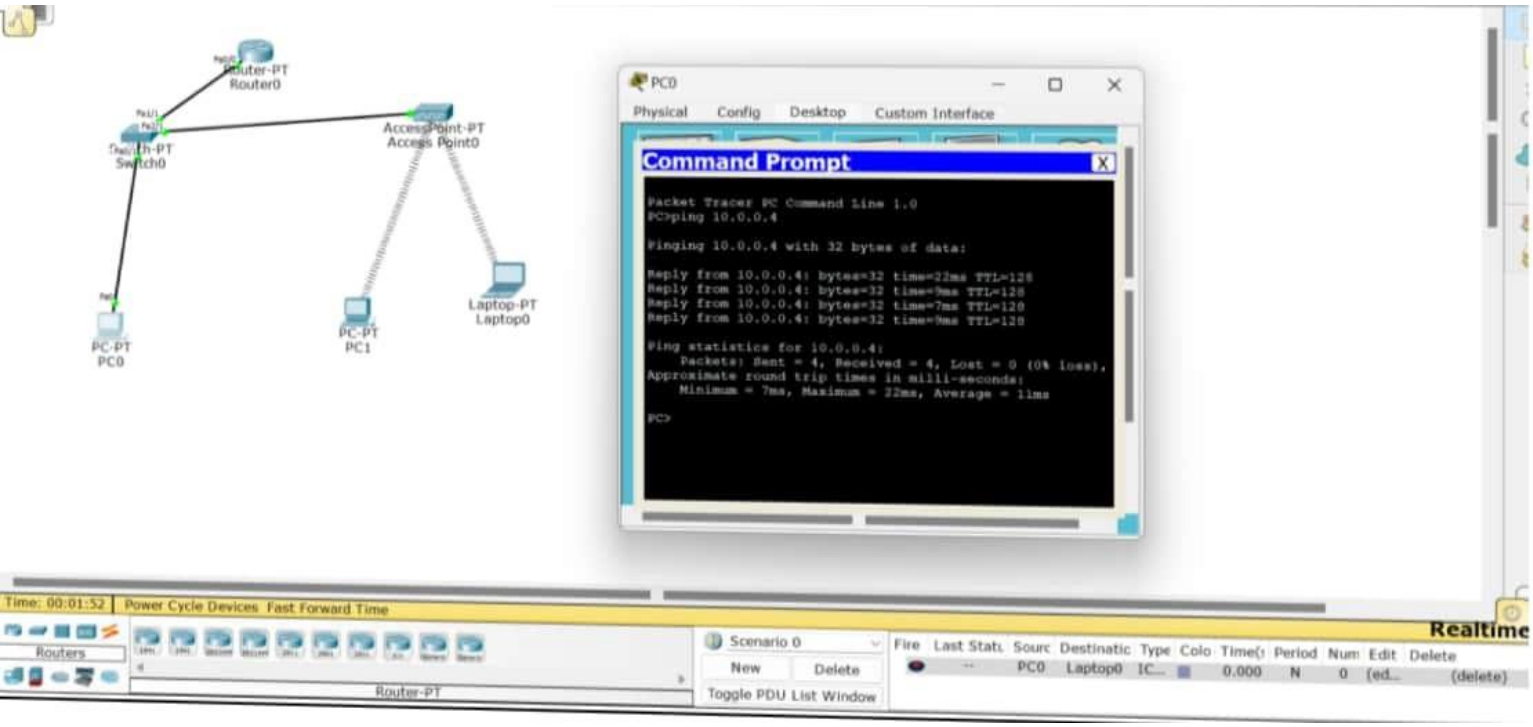
Packet Tracer PC command line 1.0
PC> ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=2ms TTL=127
Ping statistics for 10.0.0.3
Packets: Sent = 4, Received = 3, Lost = 1 (25% Loss)
Approximate Round trip times in milliseconds
Minimum = 0ms, Maximum = 1ms, Average = 0ms.

Date :

Observation

1. A WLAN is a group of devices that form a network based on radio transmission
2. Data sent in packets contain layers with labels and instructions MAC address to endpoints for routing.
3. With one access point we can connect to multiple devices wirelessly & transmit data.





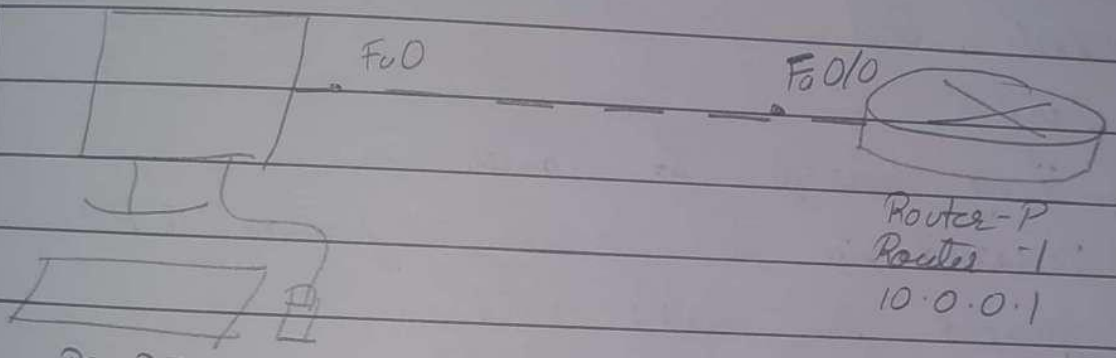
Date :

LAB - 12

Aim

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

TOPOLOGY:



Procedure:

1. Create a topology as shown above
2. Config the IP address & gateway for PC0
3. Config the router by executing the following commands.

```
enable
config T
hostname r1
```

4) enable secret P1
3) interface fastEthernet 0/0
6) ip address 10.0.0.1 255.0.0.0
7) No shut
8) line vty 05
9) login
10) password P0
11) Exit ; Exit
12) wr

Ping message to router
Password for User access Verification is P0
Password for enable is P1
Accessing router CLI from PC
Show IP route

Ping output :-
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=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0 ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0 ms TTL=255

Date :

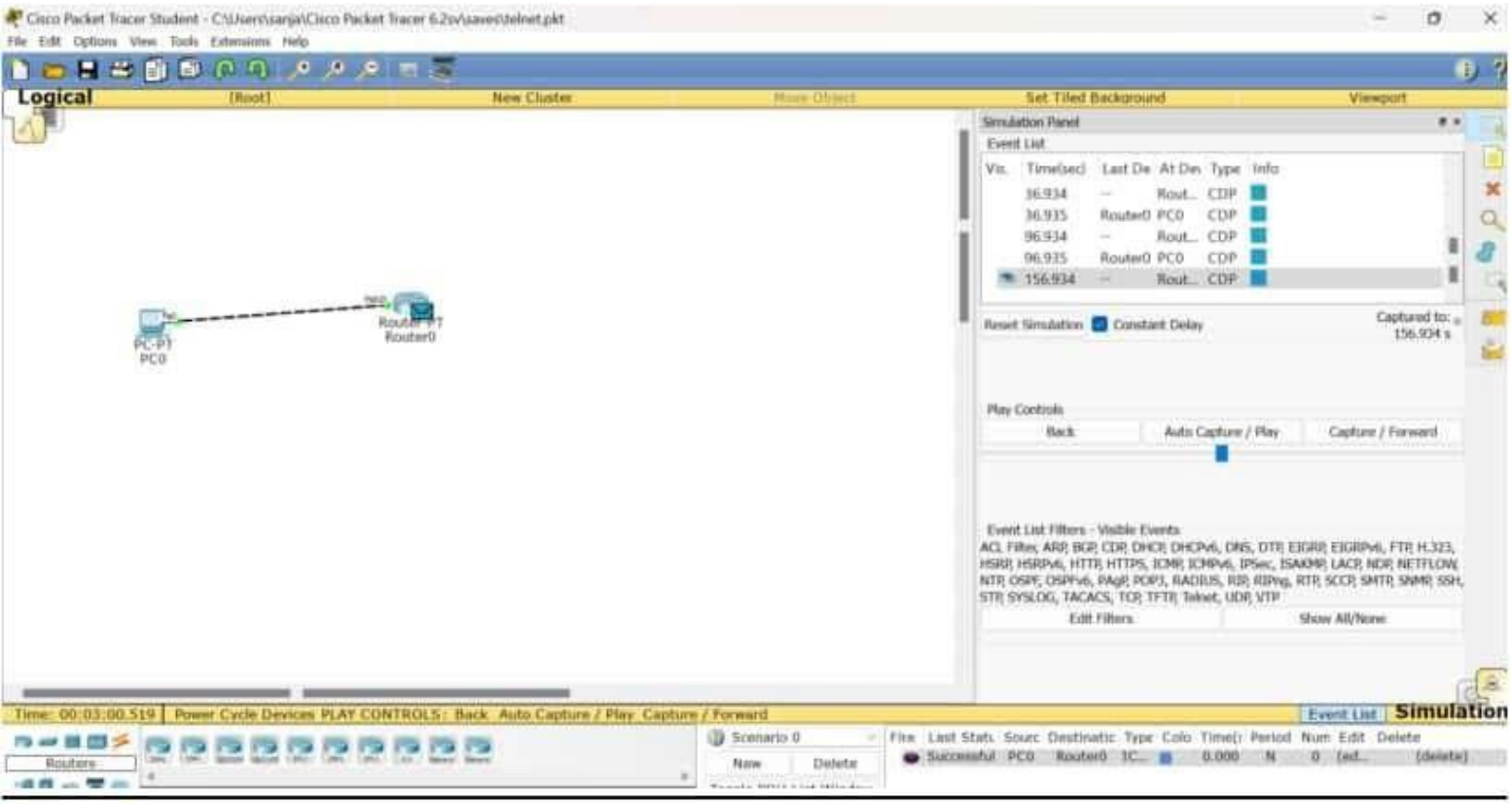
Packets : Sent = 4 Recieved = 4 Lost = 0 (0% Loss)
Approximate round trip times in milliseconds
Minimum = 0ms, Maximum = 0ms, Average = 0ms

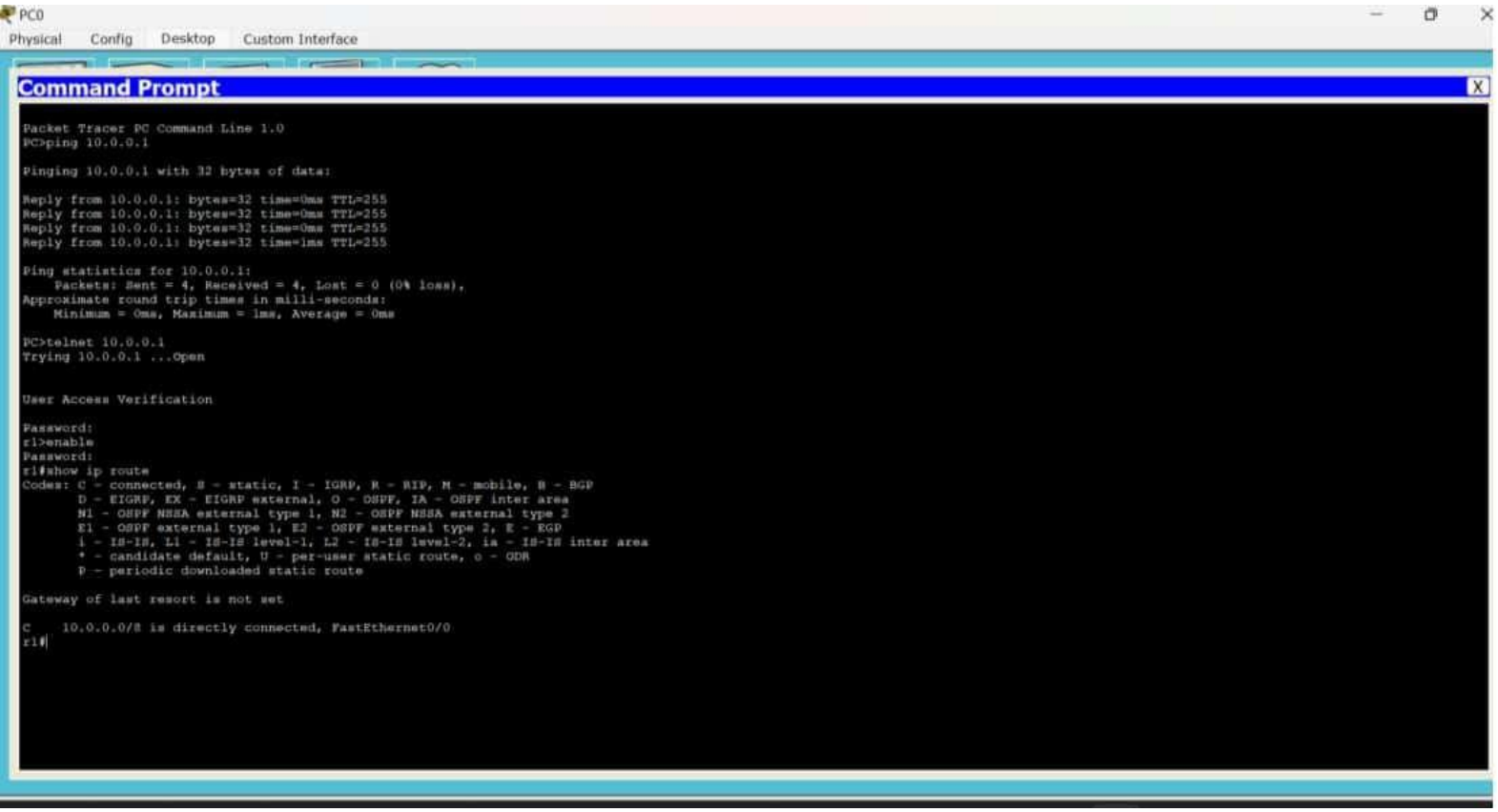
PC> telnet 10.0.0.1
Typing 10.0.0.1 ... open
User access verification
Password : PO
> enable
Password : P1
11 # show ip route

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

OBSERVATION

* TELNET stands for Teletype Network. It is a type of protocol that enables one computer to connect to the local computer.
* It is used as a standard TCP/IP protocol for virtual terminal service provided by ISO
* During TELNET operation, whatever is being performed on the remote computer will be displayed by the local computer. TELNET operates on a client / server principle.





EXPERIMENT 14

Aim: Write a program for congestion control using leaky bucket algorithm

```
#include <stdio.h>
#include <conio.h>
void main()
{
int bucket_size, dr;
printf("Enter bucket size and data rate \n");
scanf("%d", &bucket_size);
scanf("%d", &dr);
int emp = bucket_size;
while(1){
int ch, ps;
printf("Enter the packet size ");
scanf("%d", &ps);
if(ps <= bucket_size)
{
if(ps <= emp)
{ printf("Packet of size %d transmitted", ps);
}
else
{ printf("Packet dropped"); }
```

```
else
{
    printf("Packet Dropped");
}
printf("Continue ? 1 or 0 ? : ");
scanf("%d", &ch);
if(ch == 0)
    break;
}
}
```

OUTPUT :

Enter bucket size and data rate
5000
200
Enter the packet size
6000
packet dropped
Do you Continue ? 1 or 0 ? : 1
Enter the packet size :
3000
packet of size 3000 transmitted
Continue ? 1 or 0 : 1
Enter the packet size :
2000
Packet of size 3000 transmitted.

```
Enter the packet size = 3000
The Packet of size 3000 is added and in the bucket

Enter 1 to Continue or 0 to Stop: 1

Enter the packet size = 2000
The Packet of size 2000 is added and in the bucket

Enter 1 to Continue or 0 to Stop: 1

Enter the packet size = 1500
The Packet of size 1500 is dropped due to lack of space in the bucket

Enter 1 to Continue or 0 to Stop: 0
PS D:\BMSCE\Academics\Semester IV\Computer networks\Lab\Leaky bucket>
```

Date :

EXPERIMENT - 15

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.

Client.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()
```

Server.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept();
    sentence = connectionSocket.recv(1024).decode();
    file = open(sentence, 'r')
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents " + sentence + " to " + addr)
    file.close()
    connectionSocket.close()
```

OUTPUT:

Client

enter file name : Server.py

Reply from server

Contents of Server.py

Server

Sent contents of : Server.py

ServerTCP.py - C:/Users/sanja/OneDrive/Documents/ServerTCP.py (3.9.13)

File Edit Format Run Options Window Help

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

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

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

ClientTCP.py - C:/Users/sanja/OneDrive/Documents/ClientTCP.py (3.9.13)

File Edit Format Run Options Window Help

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name:")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ("\nFrom Server:\n")
print(filecontents)
clientSocket.close()
```

Date :

EXPERIMENT 16

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

ClientUDP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = Socket (AF_INET, SOCK_DGRAM)
sentence = input ("\n Enter file name");
clientSocket.sendto (bytes (sentence, "utf-8"), (serverName, serverPort))
file contents, serverAddress = clientSocket.recvfrom (2048)
print ("\n Reply from Server");
print ( file contents.decode ("UTF-8"))
for i in file contents:
        print (str (i), end = '');
clientSocket.done()
clientSocket.done()
```

→ 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)
for i in sentence:
    print(str(i), end='')
file.done()
```

OUTPUT:

Server Side

Server ready to receive
Sent contents of server.UDP.py
The server is ready to receive

client Side

Enter file name: server.UDP.py
Reply from serv:
from socket import *
:

Code of server.UDP.py written

ClientUDP.py - C:/Users/sanja/OneDrive/Documents/ClientUDP.py (3.9.13)

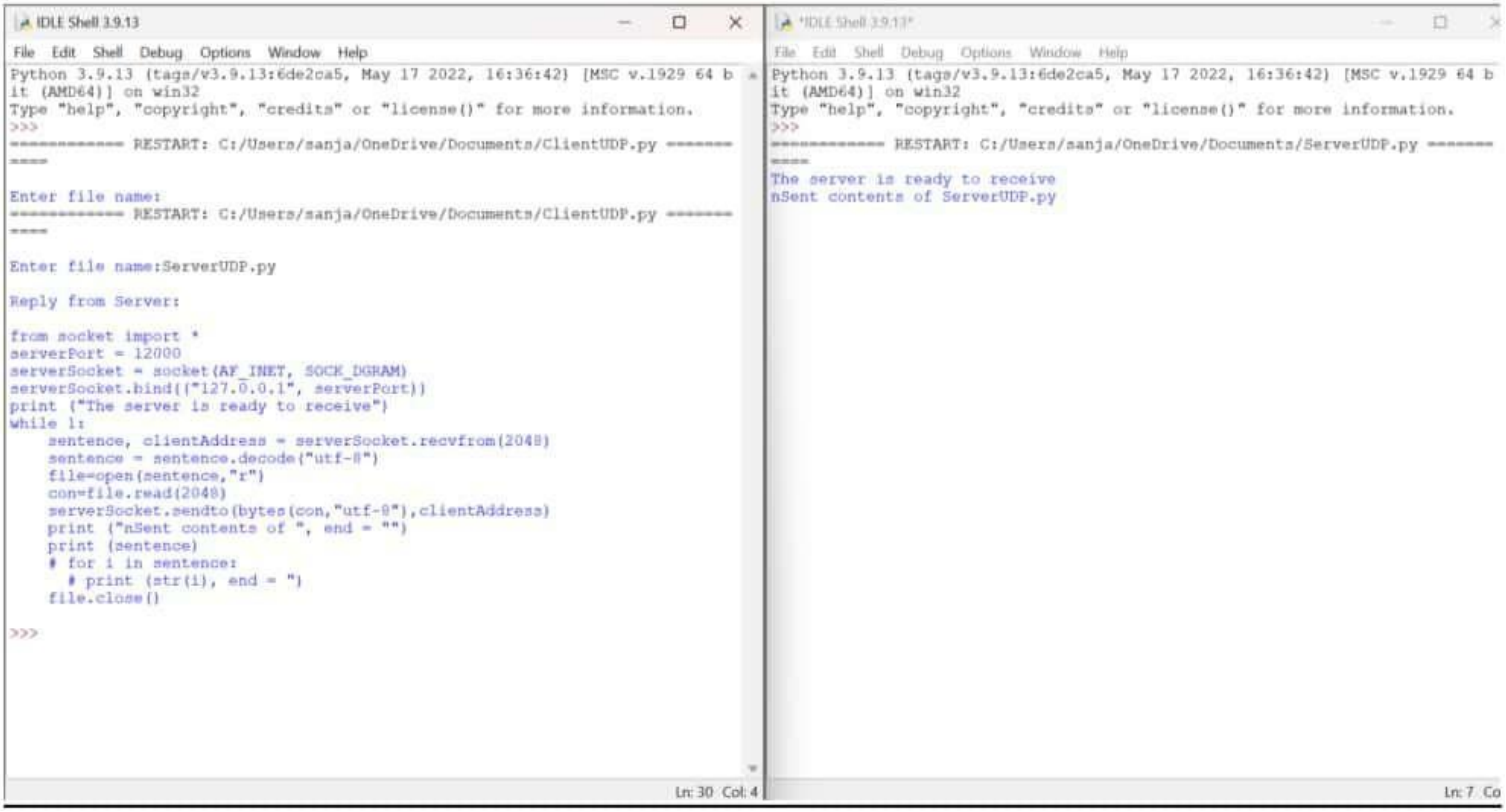
File Edit Format Run Options Window Help

```
from socket import *
serverName = "127.0.0.1"



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

ServerUDP.py - C:/Users/sanja/OneDrive/Documents/ServerUDP.py (3.9.13)

File Edit Format Run Options Window Help

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



Wireshark Tutorial

Wireshark is an open source packet analyzer, which is used for education, analysis, software development and network troubleshooting.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer.

Functionality

(1) Packet capture and filtering

Primary function lies in capturing network packets from various interfaces. Its flexible option enable users to capture specific types of traffic based on protocols, source/destination address.

(2) Real time analysis

Wireshark's real time feature aids in detecting sudden traffic spikes and unauthorized network usage.

(3) Protocol analysis

It decrypts encrypted protocol offering insights into source communication methods.

(4) Packet reconstruction

(5) Statistical information

(6) Colour coded visualisation:
To indicate various aspects such as error.

Interface of wireshark

(1) Menu bar and options displayed, Capture and file menu

(2) Packet listing window - Determine the packet flows/ captured packets in the traffic.

(3) Packet header - detailed window, contains info about the components of the packet.

(4) Bottom window called as packet contents wi which contains contents in ASCII and hexadeci format.

(5) Filter field at the top, which helps filtering packets based on any component according to your requirements.

Procedure:

→ select ethernet
→ Filter TCP or any required protocol
→ Click on it, new window opens
→ Drop down: Transmission control protocol, Src Port 62148, DST port: 443, Seq: 2, Ack: 65, Len: 0
→ This is available in the new window in the left split of screen

→ Clicking on dropdown highlights its counterpart in right split side of screen.

→ In CMD, type > ip config.

Result: Windows IP configuration
Ethernet adapter ethernet
connection-specific DNS suffix
Lin-local IPV8 address ... Fe80::be78:1609:ed25
IPV4 address 10.129.2.58
subnet mask ... 255.255.0.0
Default Gateway ... 10.129.0.11