INDEX

Name Dipesh Sah

Standard 5 Section 'B' Roll No. 092

Subject Computer Networks Lab.

Sl. No.	Date	Title	Page No.	Teacher Sign / Remarks
01.	25-Sep-24	PC to server, Hubs & Switches	01-07	
02.	09-Oct-24	Single Router.	08-10	
03.	16-Oct-24	Dual Routers.	11-15	
04.	23-Oct-24	Default Route ; Static Route	16-17	
05.	13-Nov-24	DHCP within a LAN & outside LAN	18-21	
06.	20-Nov-24	Routing Information Protocol in Router	22-23	
07.	20-Nov	Demonstration of TTL of a Packet	24-25	
08.	27-Nov	OSPF Routing Protocol.	26-30	
09.	18-Dec	Construction of VLAN	31-34	
10.	18-Dec	Concept & Operation of ARP	35-36	
11.	18-Dec	Configure Webserver & DNS	37-38	
12.	18-Dec	Operation of TELNET	39-41	
13.	18-Dec	Construction of WLAN	42-44	
14.	01-Jan	Leaky Bucket Algorithm	45-47	
15.	01-Jan	Error Detection using CRC	48-49	
16.	01-Jan	TCP/IP Sockets	50-51	
17.	01-Jan	UDP sockets	52-53	
18.	02-Jan	Tool Exploration - Wireshark	54-55	

- Cisco Packet Tracer

This initial interface contains ten components. Those ten components are :-

1. Menu Bar :- This bar provides the file, edit options, view, tools, extensions and Helps menus. You will find basic commands such as open, save, save as, print and preferences in these menus. You will also be able to access the Activity wizard from the Extensions menu.

2. Main Tool Bar :- This bar provides short-cut icons to the file and edit menu commands. This bar also provides buttons for copy, paste, undo, redo, zoom, the drawing palette and the custom devices dialog. On the right, you will also find the network Information button, which can be used to enter a description for the current network.

3. Common Tools Bar :- This bar provides access to these commonly used workspace tools: select, move, layout, place note, delete, inspect, resize shape, and simple PDUs add complex PDU.

4. Logical / Physical Workspace and Navigation bar :-
You can toggle between the physical workspace & logical workspace with the tabs on this bar. In logical workspace, this bar allows you to go back to a previous level in a cluster, create a new cluster and many more. In physical workspace, this bar allows you to navigate through physical locations, create a new city and many more.

5. Workspace :- This area is where you will create your network, watch ~~simult~~ simulations and view many kinds of info and statistics.

6. Realtime / Simulation Bar :- You can toggle between realtime mode & simulation mode with the tabs on this bar.

7. Network component box :- This box is where you choose devices and connections to put into the workspace

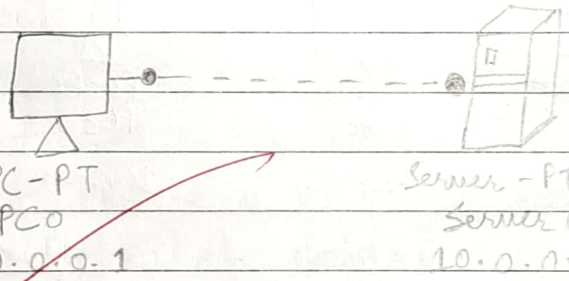
8. Device - type selection Box :- This box contains the type of devices & connection available in packet tracer.

9. Device - specific selection Box :- This box is where you choose specifically which devices you want to put in your network & which connections to make.

10. User Created Packet Window :- This window manages the packets you put in the network during simulation scenarios.

Experiment - 1

1. PC to Server.

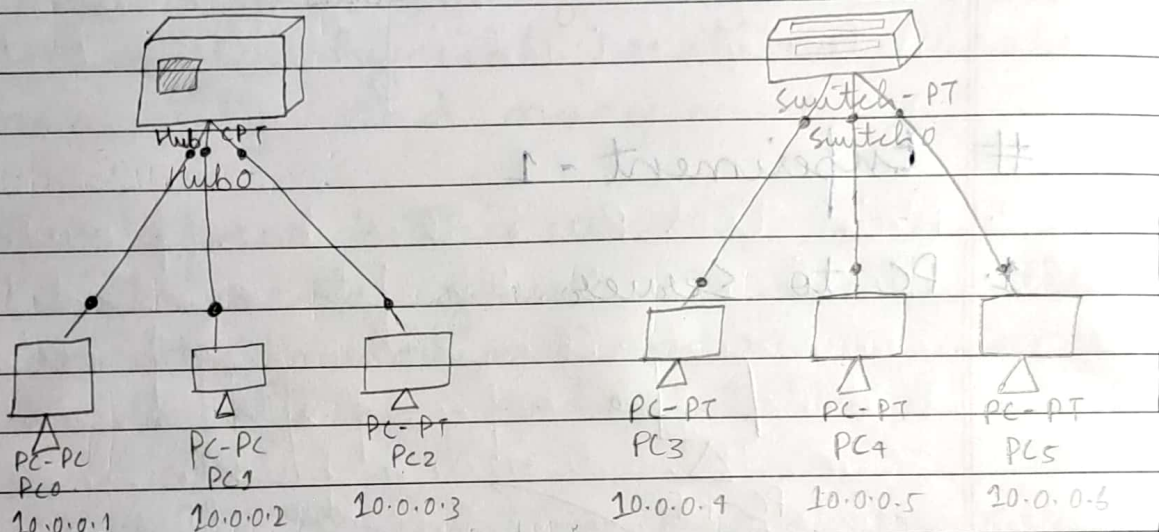


- Aim :- To set up a point - to-point network between a PC & a server, facilitating direct communication to observe data exchange.

- Topology :- A PC (PC0) is connected to a server (server 0) using a cross-over ethernet cable.
IP address of PC0 : 10.0.0.1
IP address of server 0 : 10.0.0.2

- Observation :- The direct connection allows PC0 to communicate with server 0, which is typical in small networks for tasks such as file sharing, service requests or testing server responses to client ~~app~~ queries.

2. Hub & Switch.



- Aim :- To create a simple network that consists of three PCs connected to a central hub and another network with three PCs connected to a switch. This configuration will help observe the behaviour of data transmission using hub and switch devices.

- Topology :

i) Hub Network :- Three PCs (PC0, PC1, PC2) are connected to a hub (Hub0) using straight through Ethernet cables.

IP Addresses :-
PC0 = 10.0.0.1, PC1 = 10.0.0.2, PC2 = 10.0.0.2

ii.) Switch Network :- Three PCs (PC3, PC4, PC5) are connected to a switch (switch 0) using straight - through ethernet cables.
IP Addresses :-
PC3 = 10.0.0.4, PC4 = 10.0.0.5, PC5 = 10.0.0.6

Procedure :-

1.) Add 1 hub, 1 switch and 6 PCs (PC0), (PC1, PC2) for the hub : (PC3, PC4, PC5) for the switch to the CISCO Packet Tracer workspace.

2. Use copper straight - through cables to connect PC0, PC1, PC2 to hub0. Similarly connect PC3, PC4, PC5 to switch0 using same type of cables.

3. Assign IP addresses to each PC and obtain subnet mask.

4. Switch to simulation mode to observe data traffic behaviour when packets are sent between the devices.

5. In the hub network, observe how the hub broadcasts packets to all the devices, causing potential traffic overload.

In the switch network, observe how the switch forwards packets only to the intended recipient, reducing unnecessary traffic.

6. The hub broadcasts data to all the connected devices leading to more network congestion, while the switch efficiently sends data only to the correct device, optimizing performance.

Observation:

1. The hub broadcasts packets to all devices, which may cause unnecessary traffic.

2. The switch forwards packets only to the appropriate device by learning MAC addresses, making it more efficient in reducing traffic.

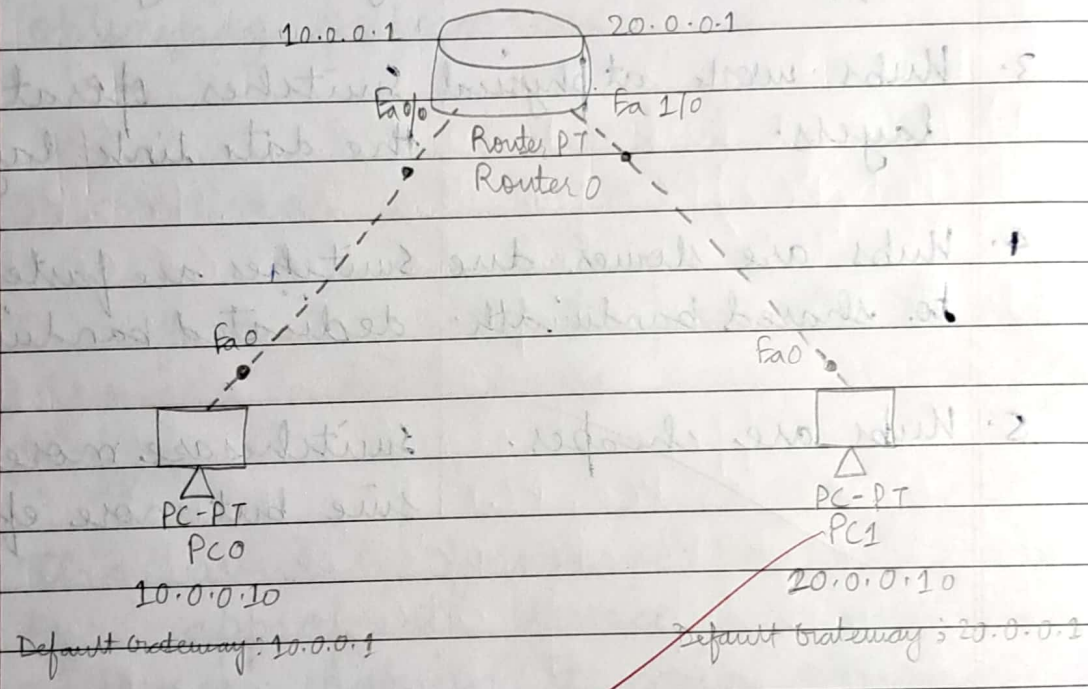
Difference between Hubs & Switches.

S.No	Hubs	Switches.
1.	Hub broadcasts data to all devices.	Switches send it only to the destination.
2.	Hubs create more traffic.	Switches reduce traffic by directing data.
3.	Hubs work at physical layers.	Switches operate at the data link layer.
4.	Hubs are slower due to shared bandwidth.	Switches are faster with dedicated bandwidth.
5.	Hubs are cheaper.	Switches are more expensive but more efficient.

Lab :- 2

Aim :- To create a network consisting of 2 PCs connected to a router. This connection will help observe the behaviour of data transmission using router.

⇒ Topology :-



1. PC0 :- connected to router's interface Fa 0/0 using a cross-over scale.
   IP address :- 10.0.0.10
   Default Gateway :- 10.0.0.1

2. PC1 :- connected to the router's interface Fa 1/0 using a cross-over scale.
   IP address :- 20.0.0.10
   Default Gateway :- 20.0.0.1

~~Lab: 2~~

3. Router :-
   Interface Fa 0/0 connected to PC0
   Interface Fa 1/0 connected to PC1
   IP address of Fa 0/0 : 10.0.0.1
   IP address of Fa 1/0 : 20.0.0.1

⇒ Procedure :- Two PCs (PC0 and PC1) are assigned with IP address 10.0.0.10 & 20.0.0.20 & gateway 10.0.0.1 & 20.0.0.1. respectively.

② Open CLI in router & enter the following :-

```
Router > enable
Router # config terminal
Router (config)# interface fastethernet 0/0
Router (config-if)# ip address 10.0.0.1 255.0.0.0
Router (config-if)# no shutdown.
exit

Router (config)# interface fastethernet 1/0
Router (config-if)# ip address 20.0.0.1 255.0.0.0
Router (config-if)# no shutdown.
exit
```

③ Ping another system or interface from the command prompt of PC0 or PC1 using command prompt.

```
> ping 20.0.0.10.
```

⇒ Observation :-

Command prompt gives output

Pinging 20.0.0.10 with 32 bytes of data:

Reply from 20.0.0.10 : bytes = 32 time = 0 ms TTL = 127
Reply from 20.0.0.10 : bytes = 32 time = 0 ms TTL = 127
Reply from 20.0.0.10 : bytes = 32 time = 0 ms TTL = 127
Reply from 20.0.0.10 : bytes = 32 time = 0 ms TTL = 127

Ping statistics for 20.0.0.10 :

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

Approximate round trip times in milli-seconds :

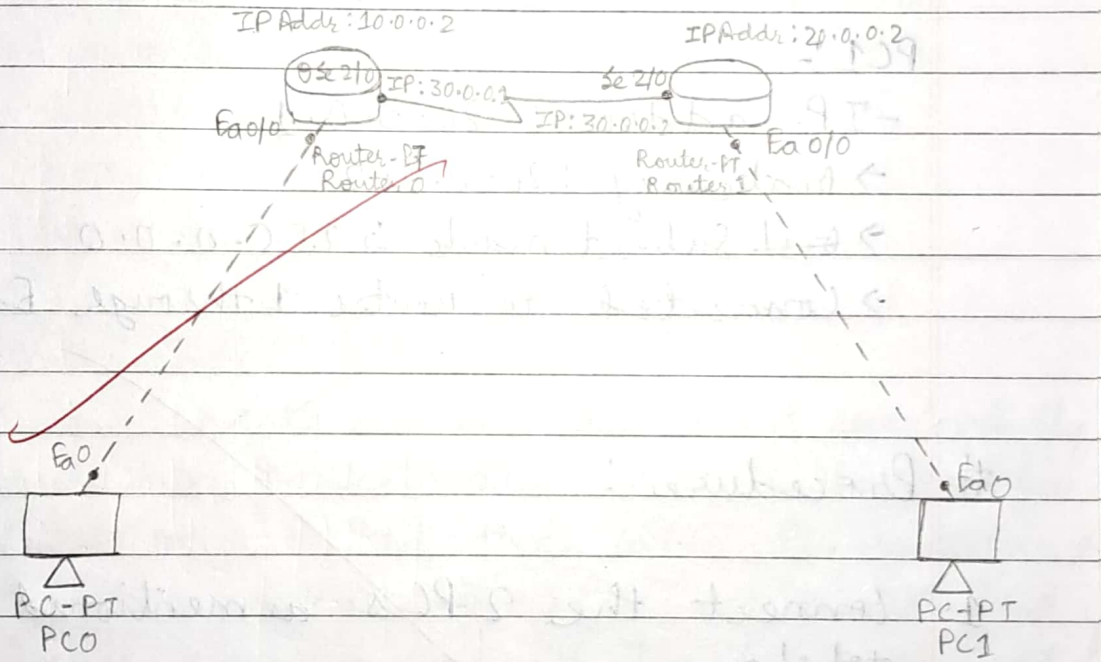
Minimum = 0 ms, Maximum = 2 ms,

Average = 0 ms.

Lab: 32(a)

Aim: Configure IP Address to routers in Cisco Packet Tracer.
Explore the following messages:
(i) ping responses. (ii) Destination unreachable
(iii) request timed out & reply.

Topology:-



Routers:
Router 0: Interface Fa 2/0 connected to PC 0 : 10.0.0.2
Interface se 2/0 connected to Router 1: 30.0.0.1
connected to Router 1 & PC0.

Router 1: Interface Se 2/0 : 30.0.0.2
Interface Fa 1/0 : 20.0.0.2
Connected to PC1 via Fa 0/0.

End Devices :
PC0 :
- IP address : 10.0.0.1
→ Gateway : 10.0.0.2
→ Subnet mask : 255.0.0.0
→ connected to Router 0 through Fa 0/0.

PC1 :-
- IP address : 20.0.0.1
→ Gateway : 20.0.0.2
→ Subnet mask : 255.0.0.0
→ Connected to Router 1 through Fa 0/0.

Procedure :

1. Connect the 2 PCs as mentioned in the topology.

2. Initialize the ip addresses of each device as shown in the figure.

3. Open CLI in the router 1 and execute the following commands :-
> enable
> config terminal
> interface fastethernet 0/0
> ip address 10.0.0.2 255.0.0.0

> no shut
> exit

3. Repeat the steps to connect to the other PC to Router.

4. Open CLI again. Run the following commands to connect the routers together:
> interface serial 2/0
> ip address 30.0.0.1 255.0.0.0
> no shut
> exit

5. Repeat the same for others

Observation:
Two routers are connected successfully but the packets are unable to transfer from one PC to the other PC of different router. When pinged, we encountered issues saying request timed out or host unreachable.

Output:
> show ip route;

C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 30.0.0.0/8 is directly connected, Serial 2/0.

Experiment 2(b) continued from 2(a):

Configure default route, static route to the router.

Aim: Successful transmission of packets from one PC to other of different routers.

Topology :- Two PCs are connected to two different routers using copper cross over wires and those two routers are connected to each other ~~one~~ using serial DCE.

Procedure:

1. After successful configuration of the two routers. Open CLI of one router and follow the commands to configure static routing.

> ip route 20.0.0.0 255.0.0.0 30.0.0.2

2. Open the CLI of other router.
> ip route 10.0.0.0 255.0.0.0 30.0.0.1

Observation :-

After configuration of static routing, the two PC's of two different router networks are able to transfer / transmit the packets using the ping command.

Output :-

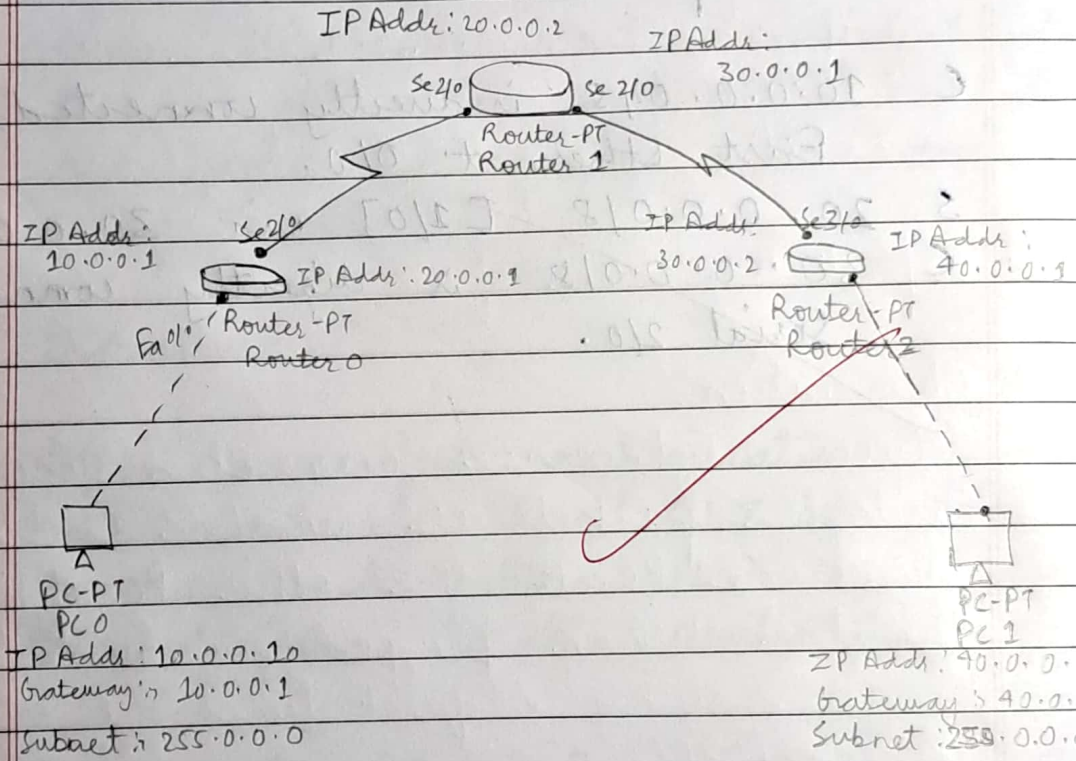
> show ip route;

C 10.0.0.0/8 is directly connected, FastEthernet 0/0.
S 20.0.0.0/8 [1/0] via 30.0.0.2
C 30.0.0.0/8 is directly connected, Serial 2/0.

Lab-03 :

Ques:) Configure default route, static route to the routers.

- Aim :- To apply & demonstrate static & default routing.
- Topology :-



Procedure :

1. Configure all the systems and routes with respective IP and Gateways with the help of the above topology.
2. Strictly route networks 10.0.0.0 & 40.0.0.0 in routers using CLI.

> ip route :> 10.0.0.0 255.0.0.0 20.0.0.1
> ip route 40.0.0.0 255.0.0.0 30.0.0.2

3 Execute commands on router 0 & router 2 as such to implement default routing.

<R0> ip route 0.0.0.0 0.0.0.0 20.0.0.2
<R2> ip route 0.0.0.0 0.0.0.0 30.0.0.1

Observation :

Upon executing "show ip route" following observations are made with successful pinging on end devices with no connection errors.

R0 :>
C 10.0.0.0/8 is directly connected, Fastethernet 0/0
C 20.0.0.0/8 is directly connected, serial 2/0
S* 0.0.0.0 [1/0] via 20.0.0.2

R1 :>
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, serial 2/0
C 30.0.0.0/8 is directly connected, serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

R2 :
C 30.0.0.0/8 is directly connected, serial 3/0
C 40.0.0.0/8 is directly connected : Fast ethernet 0/0
S* 0.0.0.0/8 [1/0] via 30.0.0.1

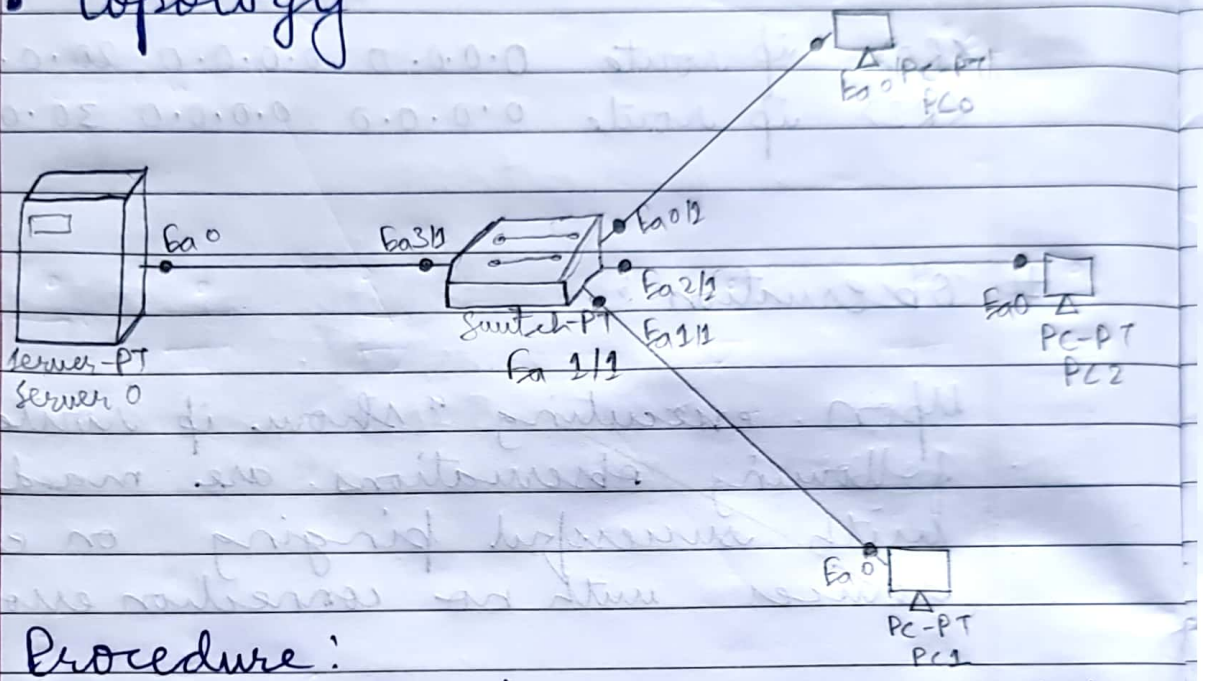
Lab-04

DHCP.

Aim :- Configure DHCP within a LAN and outside LAN.

① within the LAN.

- Topology



- Procedure:

1. Choose a generic server, a switch and 2 PCs with a laptop and connect them to switch using auto cable.

2. Click on server → Desktop → IP config → Static. Set IP Address as 10.0.0.1 and default gateway 10.0.0.0

3. Again configure the server PT by:
   config → Services → DHCP
   service → ON, pool name; switch1.
   Default gateway: 10.0.0.0
   max. no. of users = 100
   start IP → 10.0.0.3 then click 'Add'

4) For each PC → go to config → IP config change static to DHCP.

Observation :

- IP address was allocated dynamically.
- Data was sent successfully among PC's when pinged.

Output :

PC> ping 10.0.0.4

pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4 ; byte = 32 time = 0ms TTL = 28

" " " "

" " " "

Ping statistics for 10.0.0.4

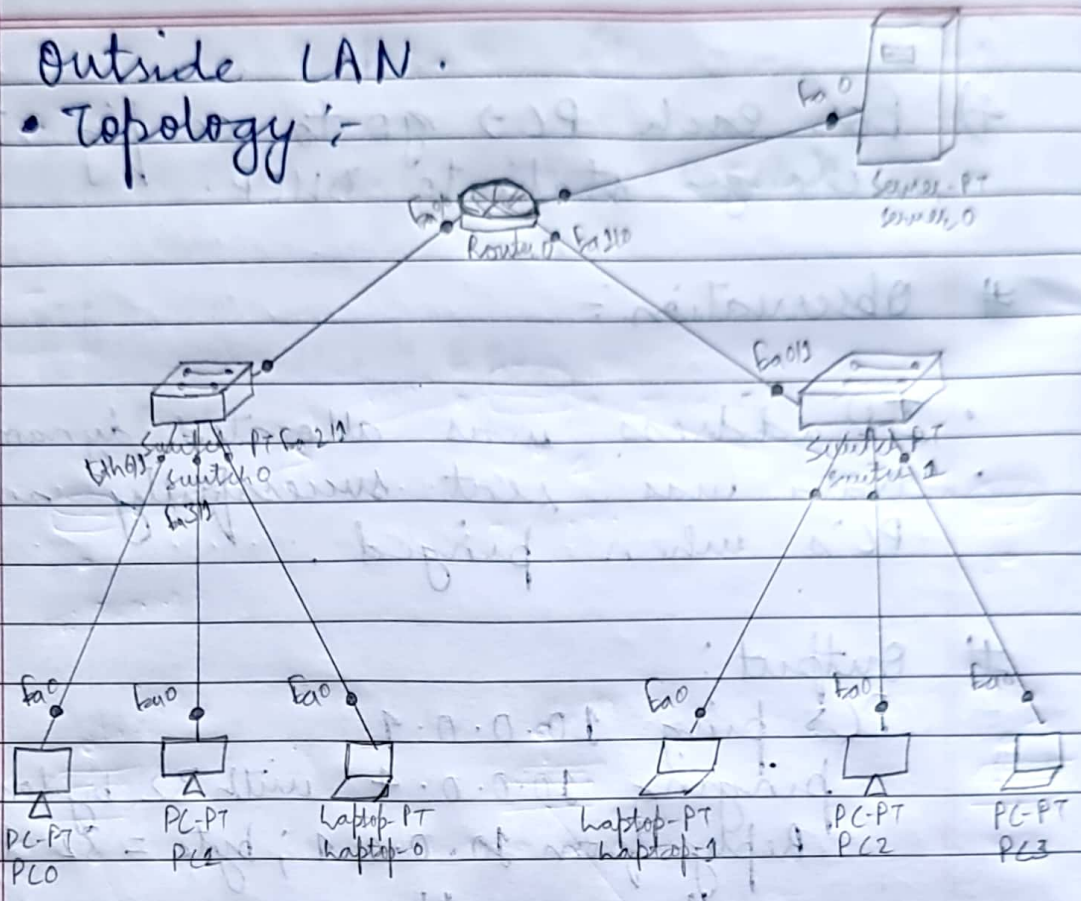
Packets sent = 4, received = 4, lost = 0 (0% loss)

appropriate round trip time in milliseconds.

Minimum = 0ms, Maximum = 0 ms,

⑪ Outside LAN.

• Topology :-



Procedure :

① For the existing server, with switch & server, a laptop and 2 PCs, add a router and add second network switch.

2. In server IP config → static → IP addr: 10.0.0.2
   def gateway: 10.0.0.1

3. Inserver 2 → config → server → DHCP → modify the existing switch 1 → def gateway: 10.0.0.1
   in some : 10.0.0.0.

4. In server → DHCP → for switch 2,
   Poolname : switchtwo
   def gateway : 20.0.0.1
   Start IP : 20.0.0.3

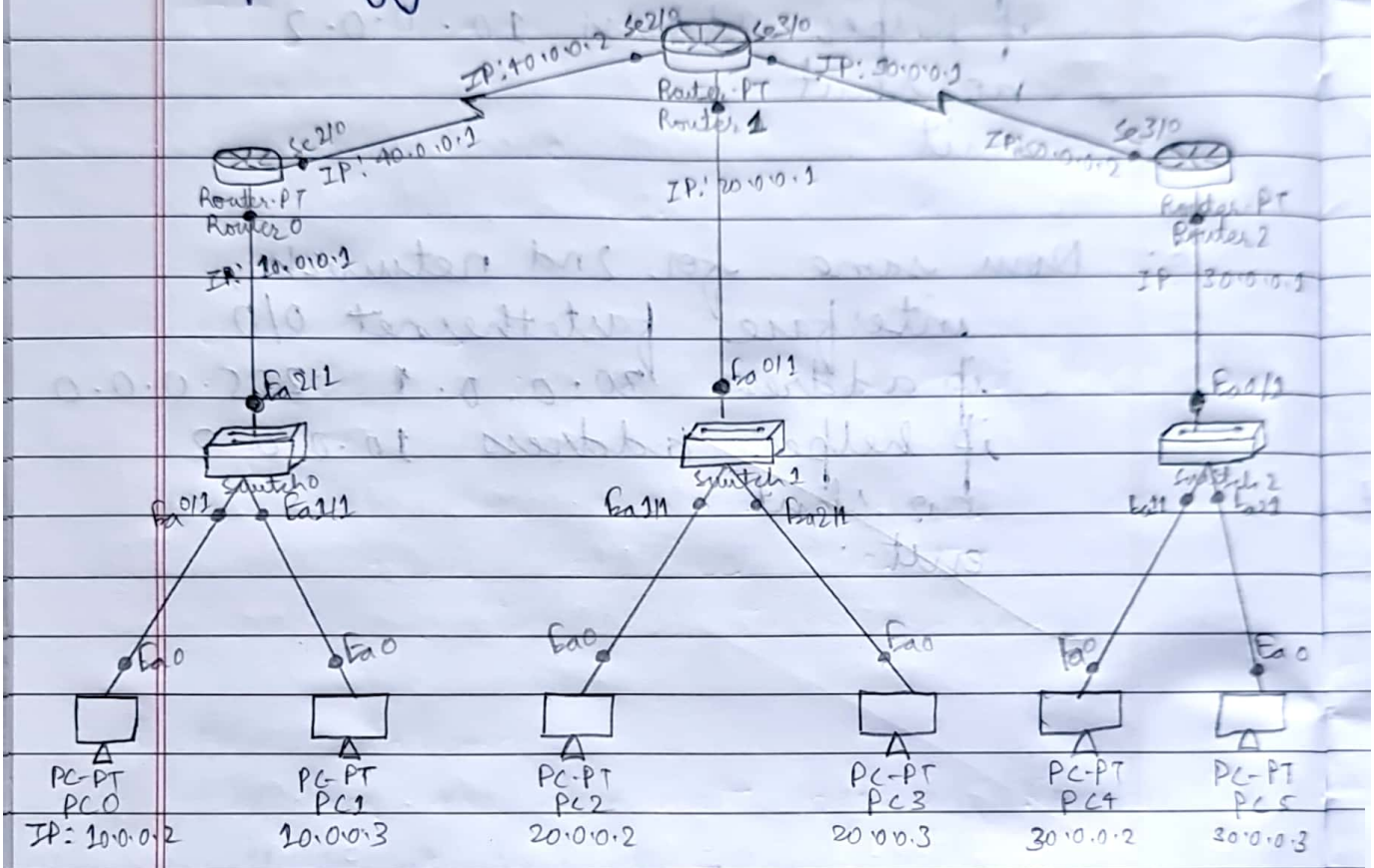
5. Do the router configuration :
enable,
config terminal
interface fastethernet 4/0
ip address 10.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shut
exit

6. Now same for 2nd network,
interface fastethernet 0/0
ip address 20.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shut
exit.

Lab :- 6 (Experiment-5(a))

- Aim: configure routing information protocol in routers (R.I.P).

- Topology :-



Procedure :

1. Place 3 routers (generic), 3 generic switches and 6 PCs.
2. Connect the routers to the corresponding switches, Then connect 2 PCs to one switch.
3. Configure the end devices & define the gateways.
4. Configure the routers using CLI and check again for green lights for all the connections.

5) Configure routing information protocol to 3 routers.

In Router - 0;
(config) # router rip
(config - router) # network 10.0.0.0
(config - router) # network 40.0.0.0

In Router - 1:
(config) # router rip
(config - router) # network 40.0.0.0
(config - router) # network 50.0.0.0
(config - router) # network 20.0.0.0;

In Router - 2:
(config) # router rip
(config - router) # network 50.0.0.0
(config - router) # network 30.0.0.0

Observation:
- Before doing RIP when pinging it showed timed out.
- After applying RIP when 30.0.0.3 was pinged from 10.0.0.1 was pinged successfully.

show ip route for router 2:
R 10.0.0.0/8 [120/2] via 50.0.0.1, serial 2/0
R 20.0.0.0/8 [120/1] via 50.0.0.1, serial 2/0
R 50.0.0.0/8 is directly connected, serial 2/0.

Experiment : 6(b)

- Aim:- Demonstrate TTL / Life of a Packet.

TTL means 'time to leave' for a packet. It tells that for how many time units the packet will be there in the network.

- Procedure:-
  - Send a simple PDU from PC1 to PC3.
  - Click on Auto capture the event list then observe the TTF of each router in PDU Information.

- Observation:-

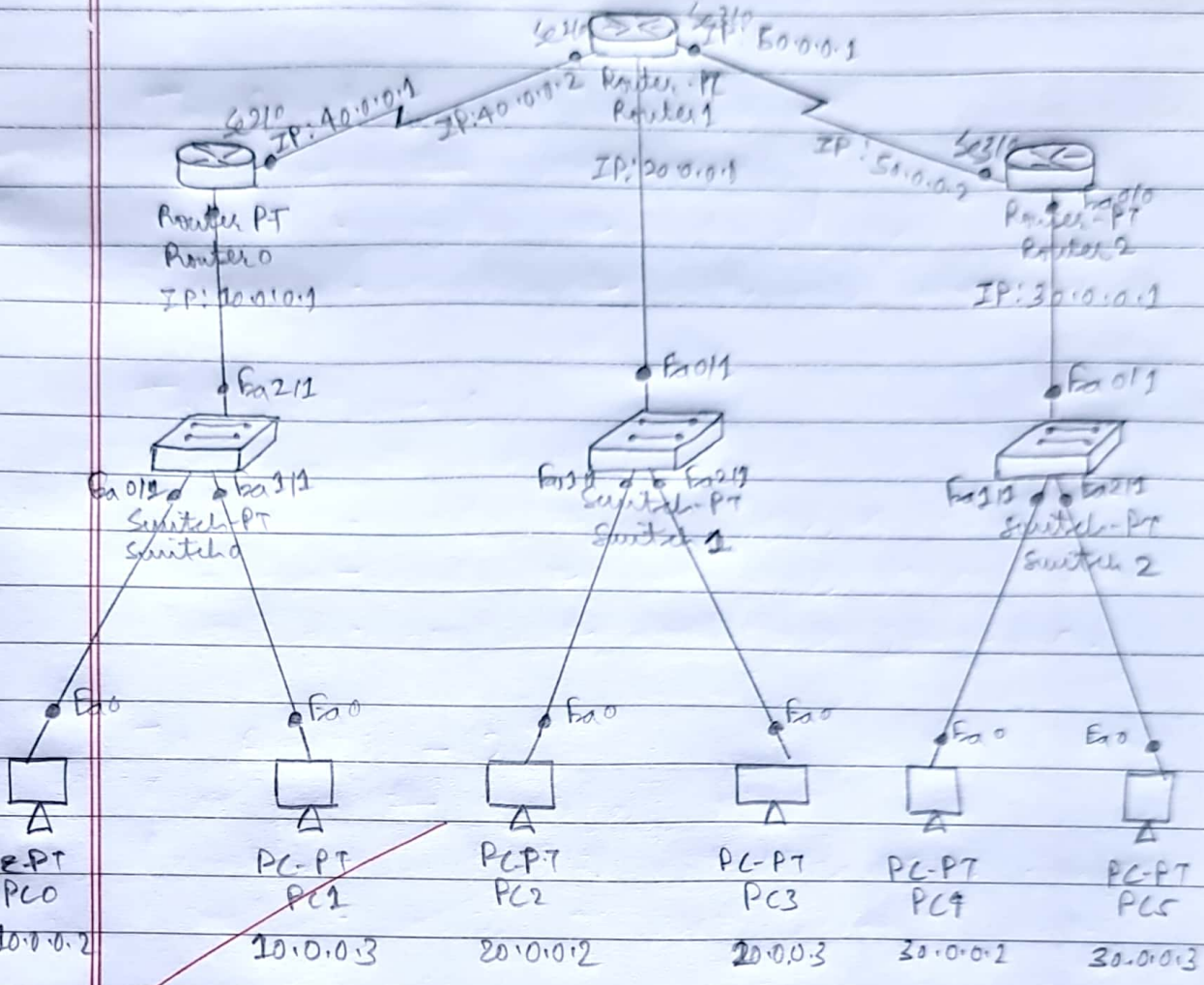
(i) When the packet passes Router 0,
inbound - TTL = 255 ms
outbound - TTL = 254 ms.

(ii) When the packet passes to Router 1,
inbound - TTL = 254 ms.
outbound - TTL = 253 ms.

(iii) When the packet passes across Router 2,
inbound - TTL = 253 ms.
outbound - TTL = 252 ms.

Thus, we conclude that there will be a decrement in TTL for 1 ms's when it passes across a Router.

- Topology :-

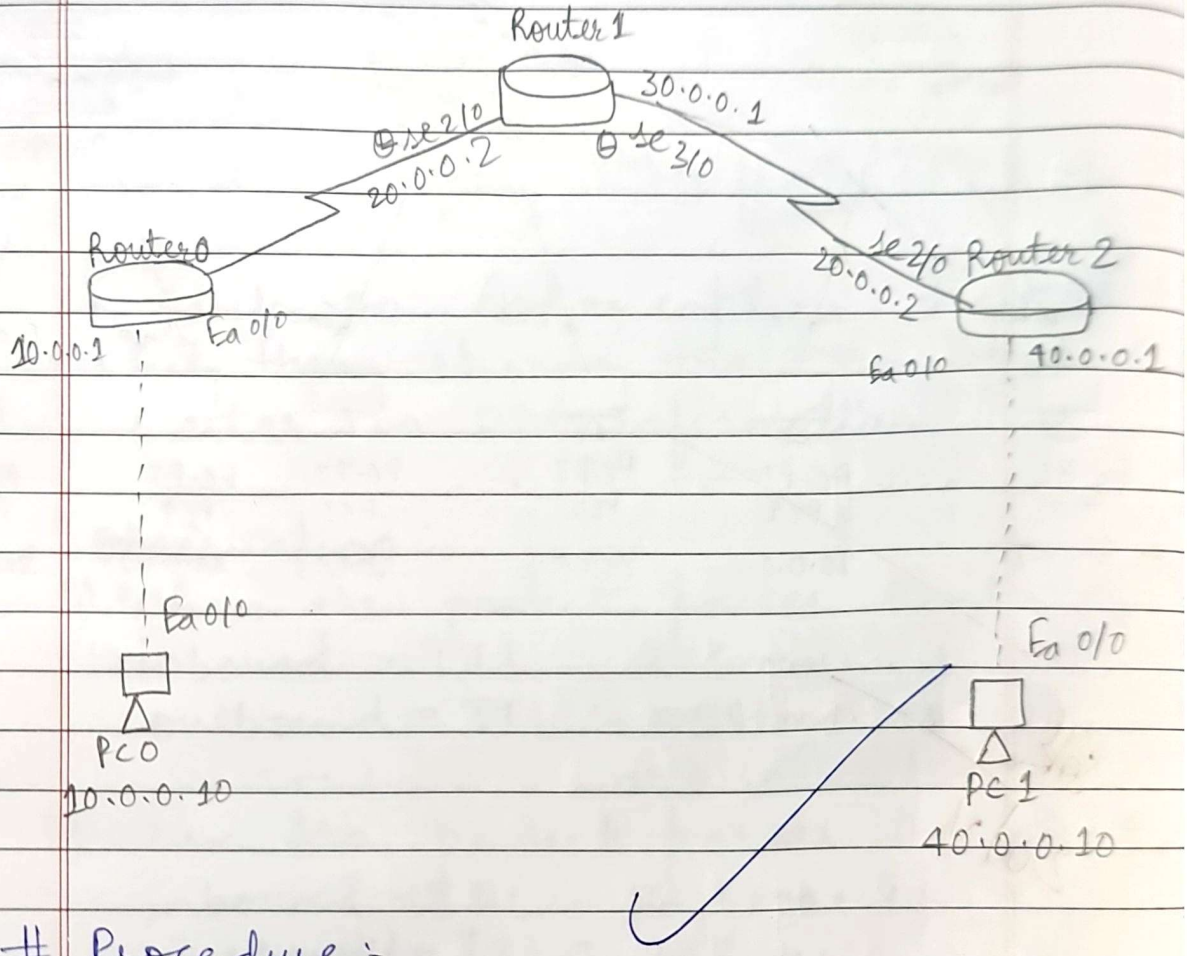

Router-PT
Router 0
IP: 10.0.0.1
IP: 40.0.0.1
IP: 40.0.0.2
Router-PT
Router 1
IP: 20.0.0.1
IP: 50.0.0.1
IP: 50.0.0.2
Router-PT
Router 2
IP: 30.0.0.1
Fa2/1
Fa0/1
Fa0/2
Fa1/1
Switch-PT
Switch 0
Switch-PT
Switch 1
Switch-PT
Switch 2
PC-PT
PC0
IP: 10.0.0.2
PC-PT
PC1
10.0.0.3
PC-PT
PC2
20.0.0.2
PC-PT
PC3
20.0.0.3
PC-PT
PC4
30.0.0.2
PC-PT
PC5
30.0.0.3

Experiment - 7

Question ⇒ OSPF routing protocol configuration.

Aim ⇒ To configure OSPF routing protocol.

Topology ⇒



Procedure :-

1) Connect the devices in the same manner as shown above.
Click on end devices → config → settings → set the default gateway (IP address of it's router) → then click on fast ethernet () & set the IP address of the end device & subnet mask.

Click on Router:
for Router 0 → CLI
(setting up fast ethernet)
R0(config)# interface fastethernet 0/0
R0(config-if)# ip address 10.0.0.1 255.0.0.0
R0(config-if)# no shutdown.
R0(config-if)# exit

(setting up serial connection)
R0(config)# interface serial 2/0
R0(config)# ip address 20.0.0.1 255.0.0.0
R0(config-if)# encapsulation PPP
R0(config-if)# clock rate 64000
R0(config-if)# no shut down
R0(config-if)# exit

Similarly, we set up the IP's of R1 and R2 while the setup of fast ethernet remains the same, the setting up of serial connections has 2 extra lines (encapsulation PPP, clock rate 64000).
Clock rate 64000 must only be written if the serially connected port shows a Ⓛ symbol.

Thus, we write the clock rate command for R0 serial 2/0, R1 serial 3/0.

After this step, all the connections must have been turned to green.

2.) To enable IP routing by configuring OSPF routing protocol in all routes.

Router R0 → CLI

```
R0(config)# router ospf 1
R0(config-router)# router-id 1.1.1.1
R0(config-router)# network 10.0.0.0

R0(config-router)# network 20.0.0.0

R0(config-router)# exit.
```

Similarly do the same for R1 and R2 and specify the area numbers and ip addresses with subnet masks in CLI.

3.) Once the setting up of networking area is done, we configure loopback address to the routers:-

```
R0(config-if)# interface loopback 0
R0(config-if)# ip add 172.16.1.252 255.255.0.0
R0(config-if)# no shutdown

R1(config-if)# interface loopback 0
R1(config-if)# ip add 172.16.1.253 255.255.0.0
R1(config-if)# no shutdown

R2(config-if)# interface loopback 0
R2(config-if)# ip add 172.16.1.254 255.255.0.0
R2(config-if)# no shutdown.
```

4) On checking routing table of R2 using show ip route, we can see that R2 doesn't know about area 3.

Gateway of last resort is not set.
O IA 20.0.0.0/8 [110/128] via 30.0.0.1 serial 1/0.
C 40.0.0.0/8 is directly connected, fastethernet 0/0
C 30.0.0.0/8 is directly connected, serial 2/0.

Since, R2 doesn't know about area 3, we have to create a virtual link between R0 and R1.

5. Creating virtual link between R1, R0
In Router R0,
R0(config)# router ospf 1
R0(config-router)# area 1 virtual link 2.2.2.2
R0(config-router)# exit.

In Router R1
R1(config)# router ospf 1
R1(config-router)# area 1 virtual link 1.1.1.1
R1(config-router)# exit.

6) Now, check routing table of R2.
Once all these steps are completed, the messages can be pinged across the devices.

Observations :

In Router R2 :
Router # show ip route

O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:57:25, serial 2/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0
O IA 10.0.0.0/8 [110/129] via 30.0.0.1 00:57:25, serial 2/0
C 30.0.0.0/8 is directly connected, serial 2/0
C 172.16.0.0/16 is directly connected, Loopback0.

Similarly the output is shown for Router 0 and 1.

Ping output :
(from PC0 to PC1)
PC0 → command prompt
C:\>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data

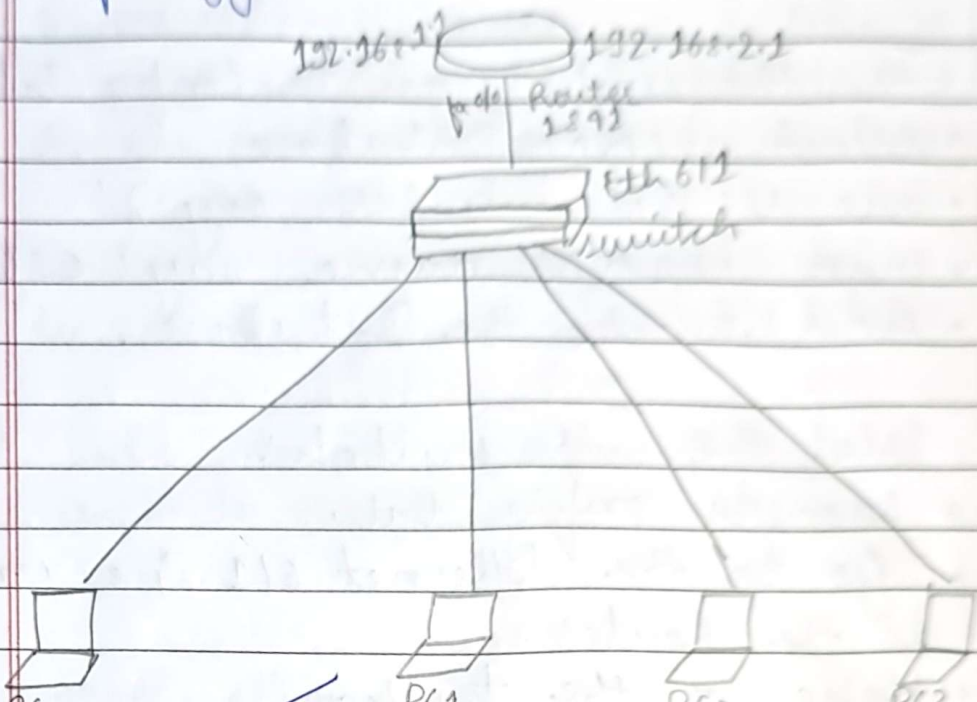
Request timed out
Reply from 40.0.0.10 : bytes = 32, time = 21 ms TTL = 125
Reply from 40.0.0.10 : bytes = 32, time = 2 ms TTL = 125
Reply from 40.0.0.10 : bytes = 32, time = 28 ms TTL = 128
Pinging statistics for 40.0.0.10
Packets : sent = 4, Recieved = 3, Loss = 1 (25% loss)

Bafna Gold
Date: 18 Dec Page: 31

Experiment - 8
VLAN

Aim :- To construct a VLAN and make the PC's communicate among a VLAN.

Topology :



Procedure :

1) Place a 1841 Router, a switch and 4 PCs.
2) Connect the four PCs to the switch via fast ethernet.
3) Since only 4 fastethernet ports are available in the switch, we have to add on ethernet port.

4) To add on ethernet port:
- Switch off the Power button of the switch.
- Add the ethernet port to the switch
- Switch on the power button
- Connect the router to the switch via ethernet 6/1.

5) In the switch, go to Config Tab and:
- select VLAN Database
- Give VLAN number say '2'
- Give VLAN name say 'CSEISE'.
- Add it to the Database.

6) Select the the switch:
- Go to config tab.
- Go to the Ethernet 6/1 i.e. connected to the Router.
- Make it the trunk.

7) Configure the PCs as shown in the topology.

8) Select switch:
- Go to config tab.
- Go to FastEthernet 2/1
- Set VLAN number as '2' i.e. 'CSEISE'
- Similarly set VLAN 2 for fastEthernet 3/1 interface too.

9) Configure the Router:
Router(config)# interface FastEthernet 0/0

Router (config-if)# ip address 192.168.1.1 255.255.255.0
Router (config-if)# no shut
Router (config-if)# exit.

Now, to configure the router's VLAN interface:
Router (config)# interface fastethernet 0/0.1
Router (config-subif)# encapsulation dot1q 2
Router (config-subif)# ip address 192.168.3.1 255.255.255.0
Router (config-subif)# no shut.
Router (config-subif)# exit.

10) Ping devices within the same VLAN and to the devices of different VLAN.

Observations:

1) When devices are pinged within same VLAN:
- Pinging 192.168.1.3 from 192.168.1.2
  - the data packet doesn't go to the router.
  - the switch forwards the packet without the need of the router.

2) When a device pings a device of another VLAN:
- Pinging 192.168.2.3 from 192.168.1.2

- The data packet's journey is as follows:

192.168.1.2 → switch → Router

192.168.2.3 ← Switch ←

3.) VLANs divide a single switch into multiple logical switches.
- Devices in one VLAN cannot directly communicate with devices in another VLAN without a Router.

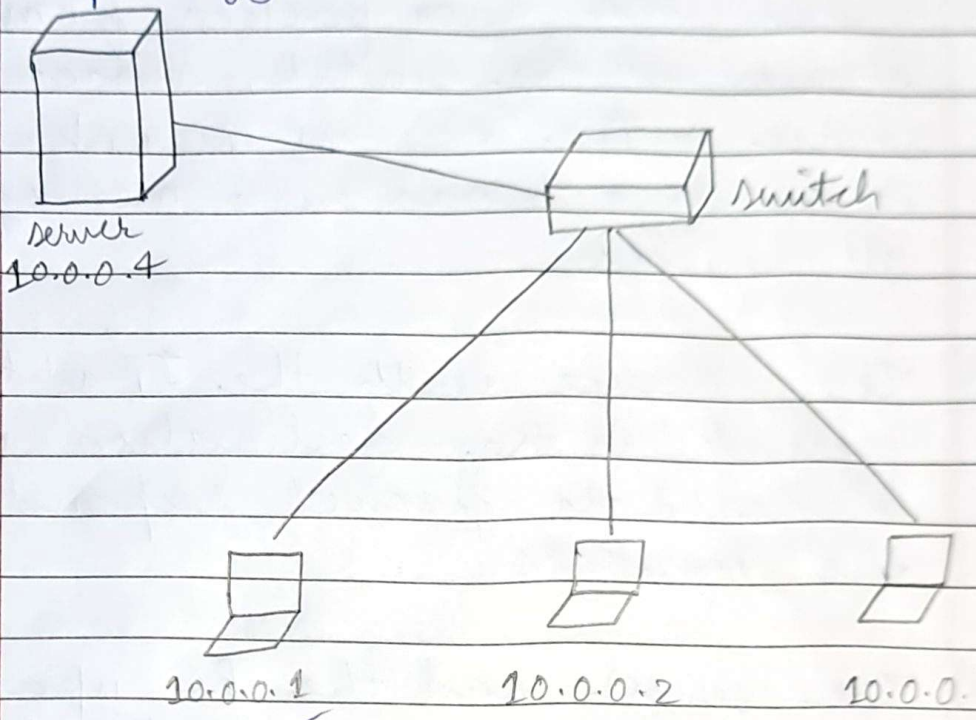
4.) Traffic Isolation:
- Each VLAN maintains its own broadcast domain.
- broadcasts sent by devices in one VLAN do not reach devices in another VLAN.

5.) VLAN trunking allowed switches to forward frames from different VLANs over a switch link called the trunk.
- This is done by adding an additional header information called tag to the ethernet frame - VLAN tagging.

Experiment - 9
ARP

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

Topology :-



Procedure :

1.) Create the topology as shown above.
2.) Configure the PC's and the server.
3.) Click on the Inspect Mode (Q), then click on the end devices and open ARP tables.
4.) Send a data packet from any end device say server to other end device say 10.0.0.3 PC.

5.) Open simulation mode to capture each step of data transfer.

Observations :-

1.) The ARP tables of all end devices are initially empty.

2.) When the data packet from server arrives at the switch, since the source MAC address is unknown, it sends a broadcast message to all devices.

3.) The device with the IP address present in the destination address of the data packet responds to the messages.

4.) The server and the PC updates their ARP tables matching the IP addresses to MAC addresses.

5.) Over time, the ARP tables grows as data packets are sent.

6.) The MAC table of the switch which was initially empty updates its MAC table gradually too.

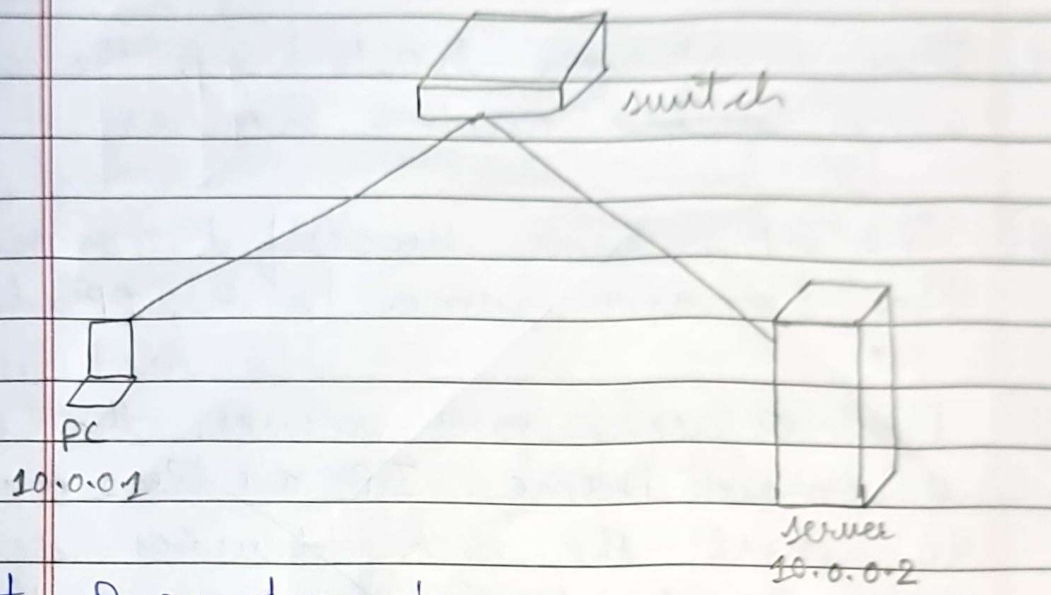
ARP Table for 10.0.0.4

IP Address	Hardware Address	Interface
10.0.0.3	0001.C726.97E5	FastEthernet0

Experiment - 10
DNS

Aim :- Configure Web server, DNS within a LAN.

Topology :-



Procedure :-

1.) Set up the LAN as per the topology mentioned above and configure the devices accordingly.

2.) Go to Server ⇒ Services ⇒ DNS:
Name : bmsce [Domain name]
Address : 10.0.0.2
Add the mapping of domain name to the address.

3.) Go to PC → Config → Global → Setting → DNS Server : 10.0.0.2

[the server that provides the DNS mapping].

4.) Go to PC → Desktop → Web Browser
Type the URL: http://bmsce

Observations :

1.) The webpages hosted by the server were visible on the browser.

2.) The DNS was successful in mapping the domain name to the IP address.

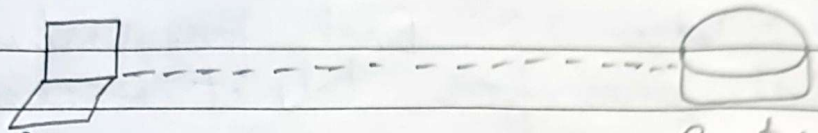
3.) DNS server is a server that contains a Domain Name : IP address mapping to which the end devices send requests to map the Name to IP addresses.

Experiment - 11

TELNET

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

Topology :-



Procedure :-

1.) Create the topology as given above and configure the devices accordingly

2.) Commands in Router :-

Router > enable

Router # config terminal

Router (config)# hostname R1

R1 (config)# enable secret 1234
(enable password)

R1 (config)# interface fastethernet 0/0

R1 (config-if)# ip address 10.0.0.2 255.0.0.0

R1 (config-if)# no shut.

R1 (config-if)# line vty 0 3

R1 (config-line)# login.

% login disabled on line 194, until 'password' is set.
R1(config-line)# password 4321
R1(config-line)# exit ↓ user access verification password
R1(config)# exit
R1# wr → to perform write operation
Building configuration.
[OK]

Note: vty 0 3: First for virtual terminal lines for telnet access.

3.) In PC: command prompt:
- First try pinging to see if the devices are connected.

PC> telnet 10.0.0.2
Trying 10.0.0.2 open

User Access Verification:

Password: 4321
Password: 4321
R1> enable
Password: 1234

R1# show ip route.

C: 10.0.0.0/8 is directly connected, Fast, Ethernet 0/0.
R1#

Observations :

1.) The admin in PC is able to run commands as run in router CLI and see the results from PC.

2.) Telnet allows users to establish a remote session with another device like router, over a TCP/IP network.

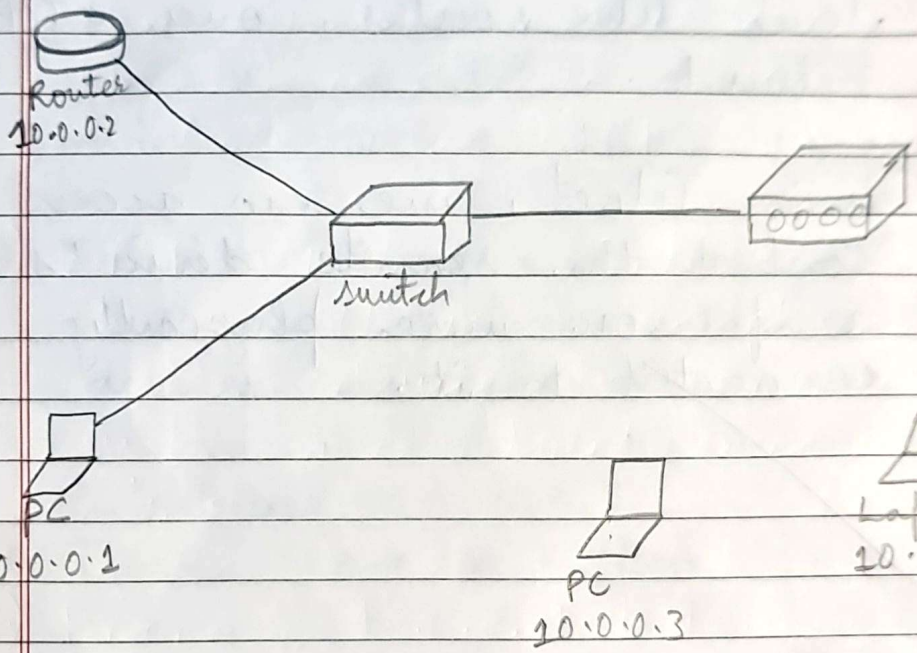
3.) Using Telnet, we can access and control the remote device's CLI as if you were physically connected to it.

10-Dec-2023

Experiment - 12
WLAN

Aim: To construct a Wireless LAN and make the nodes communicates wirelessly.

Initial Topology:



Procedure:

1) Create the topology as given above and configure the devices.

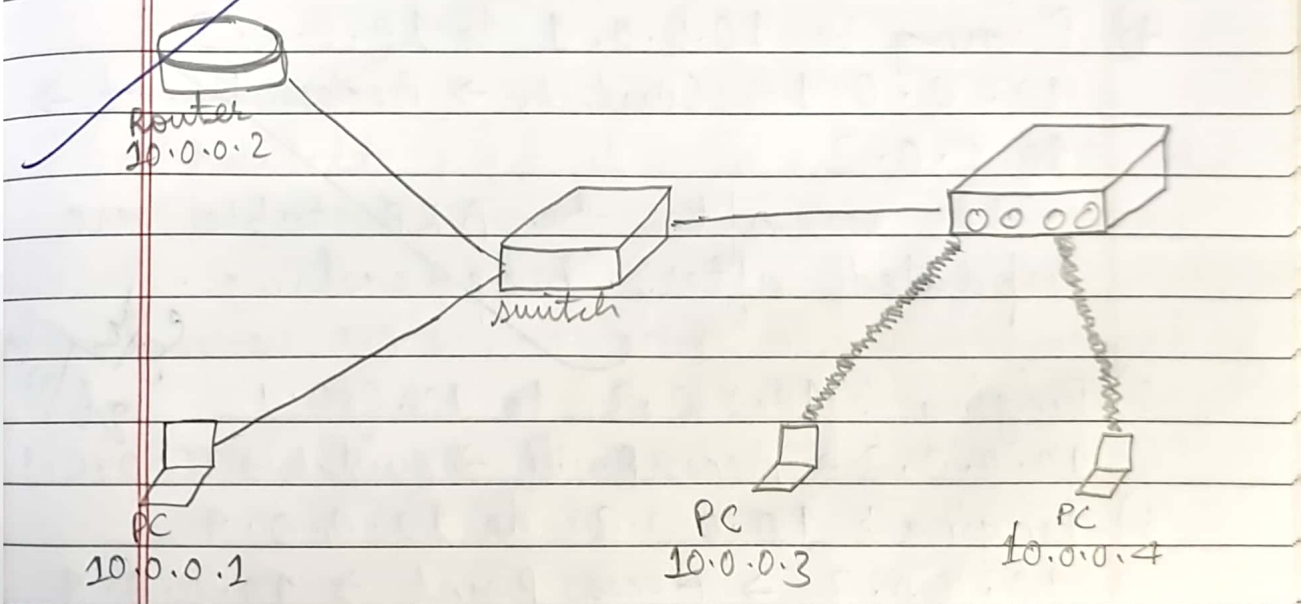
2) Configure Access Point:
Click Access Point → config → Port1:
SSID : bmsce
Select ① WEP
Set key : 1234567890

3.) Configure PC and laptops with Wireless standards:
- Switch off device.
- Drag the existing PT-HOST-NM-LAN to the component listed in the LHS of Physical.
- Drag WMP300N Wireless interface to the empty port.
- Switch on the device.

4.) In the config tab, a new wireless interface was added.

5.) Configure the device by entering SSID, WEP, WEP key, IP addresses and Gateway.

Topology after Wireless Configuration:



6.) Ping from every device to every other device to check for connection.

Observations :

1.) We were able to ping from every device to every other device.

2.) Access Point :
Creates bridge between wired and wireless devices.
- SSID Broadcasting : announces the wireless network's name (SSID) to allow devices to connect using WEP, WPA or WPA2.

3.) WMP300N Wireless Interface :
- Wireless Networks adapter that enables devices to communicate with access Point using wireless signals.

4.) Pinging :- 10.0.0.1 to 10.0.0.3
10.0.0.1 → Switch → Access Point → 10.0.0.3
• This is after the ARP tables are updated after broadcasting.

5.) Pinging : 10.0.0.3 to 10.0.0.1 :
10.0.0.3 → Access Point → switch → 10.0.0.1

6.) Pinging : 10.0.0.3 to 10.0.0.4
10.0.0.3 → Access Point → 10.0.0.4

7.) Every device is now connected to every other device in the WLAN.

Cycle-2

Experiment -13
Leaky Bucket Algorithm.

In the network layer, before the network can make quality of service guarantees, it must know what traffic is being guaranteed, one of the main causes of congestion is that traffic is often

There are two types of traffic shaping :-
1.) Leaky Bucket
2.) Token Bucket.

For Example, Let n = 1000
packet = 200 700 500 450 400 200
since n > size of the packet at the head of ~~the~~ queue i.e. n > 200.

Therefore, n = 1000 - 200 = 800
packet size of 200 is sent into the networks: 200 700 500 450 400

Now again, n > size of the packet of the head of the queue i.e: n > 400
Therefore, n = 800 - 400 = 400

Code:

```
#include <stdio.h>
int main(){
    int incoming, outgoing, bucket_size,
    n, store = 0;
    printf("Enter the bucket size,
    outgoing rate, and the no. of
    stores: ");
    scanf("%d, %d %d", &bucket_size,
    &outgoing, &n);

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

    if incoming in (bucket_size - store){
        store += incoming;
        printf("Bucket buffer size %d
        out of %d \n", store, buck_size);
    } else{
        printf("Dropped %d no of packets
        \n", incoming - (buck_size - store));
        printf("Bucket Buffer size %d
        out of %d \n", store, buck_size);
        store = buck_size;
    }
    store = store - outgoing;
    printf("After outgoing %d bytes
    left out of %d in buffer \n", store,
```

```
        buck-size);
        m--;
      }
}
```

Output:

Enter bucket size, outgoing rate and no. of inputs : 100 20 3
Enter the incoming packet size: 30
Incoming packet size 30

Bucket suffix size 30 out of 100
After outgoing 10 bytes left out of 100 in buffer.

Enter the incoming packet size : 50
Incoming packet size : 50
Bucket suffix size 60 out of 100
After outgoing 40 bytes left out of 100 in buffer.

Enter the incoming packet size: 80
Incoming packet size : 80
Dropped 20 no. of packets.

Bucket Buffer size 40 out of 100
After outgoing 80 bytes left out of 100 in buffer.

01-Jan-2025

Experiment -14

AIM: Implementation of CRC -

Code:

```
def XOR (a, b):
    result = []
    for i in range (1, len(b)):
        if a[i] == b[i]:
            result.append ('0');
        else:
            result.append ('1');

    return ''.join ('result')

def moddiv (dividend, divisor):
    pick = len (divisor)
    temp = dividend [0: pick]
    while pick < len (dividend):
        if temp[0] == '1':
            temp = XOR (divisor, temp)+
            dividend [pick]
        else:
            temp = XOR ('0' * pick, temp)
            + dividend [pick]
        pick += 1

        if temp[0] == '1':
            temp = XOR (divisor, temp)
        else:
            temp = XOR ('0'*pick, temp)
```

```
    checkword = temp
    return checkword

def encodeData(data, key):
    l_key = len(key)
    append_data = data + '0'*(l_key-1)
    remainder = mod2div(append_data, key)
    codeword = data + remainder
    print("Remainder ", remainder)
    print("Encode Data (Data + Remainder) :",
          codeword)

data = "100100"
key = "1101"
encodeData(data, key)
```

Output :

```
Sender site : ---
Remainder : 001

Encode Data (Data + Remainder) : 100100001

Receiver side :
correct message received.
```

01-Jan-2025

Experiment - 15

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

⇒ Code

ServerTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000

serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)

while 1:
    print("Server ready to receive")
    connectionSocket, addr = serverSocket.accept()

sentence = connectionSocket.recv(1024).decode()

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

```
connectionSocket.send(l.encode())
print('\n Sent contents of' + sentence)
file.close()
connectionSocket.close()
```

ClientTCP.py

```
from socket import *

serverName = '127.0.0.1'
serverPort = 14000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,
  serverPort))

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

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).
  decode()

print('\n From Server: \n')
print(filecontents)
clientSocket.close()
```

⇒ Output:

```
The server is ready to receive.
Send contents of ServerTCP.py.
Reply from server:
```

01-Jan-2025

Experiment - 16

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.

⇒ Code:

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()
```

clientUdp.py

```
from socket import *

serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input("\nEnter file name: ")

clientSocket.sendto(bytes(sentence, "utf-8"),
(serverName, serverPort))

filecontents, serverAddress = clientSocket.
recvfrom(2048)
print("\nReply from Server:\n")
print(filecontents.decode("utf-8"))

clientSocket.close()
```

Output :-

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

} server side

```
Enter file name: serverUDP.py
Reply from server:
```

} client side

02-Jan-2025

Experiment - 17

Tool Exploration - Wireshark.

Wireshark is a powerful and widely used network protocol analyzer. It allows us to capture and inspect data packets travelling over networks in real-time, making it a crucial tool for studying computer networks, troubleshooting network issues and understanding the protocols.

Key Features:

1.) Packet capture: captures live network traffic from various interfaces (ex: ethernet, wi-fi)

2.) Protocol Analysis: supports hundreds of protocols (ex: TCP, UDP, HTTP, FTP)

3.) Filtering: Offers powerful filters to isolate specific packets or traffic types.

4.) Visualization: Displays packets details with hierarchial layers (ethernet, IP, TCP/UDP).

Use Cases of Wireshark.

1.) Network Troubleshooting.
Diagnosing slow network speeds
Identifying netw & misconfigurations.

2.) Security Analysis:
Detecting malicious traffic / intrusions

3.) Protocol study.
Understanding packet structures and communication flows.

common filters:

1.) http: shows only http traffic.

2.) tcp port == 8: shows traffic on TCP port 80

3.) ip addr == 192.168.1.1: shows packets to and from specific IP.

4.) UDP: shows only UDP traffic.