INDEX

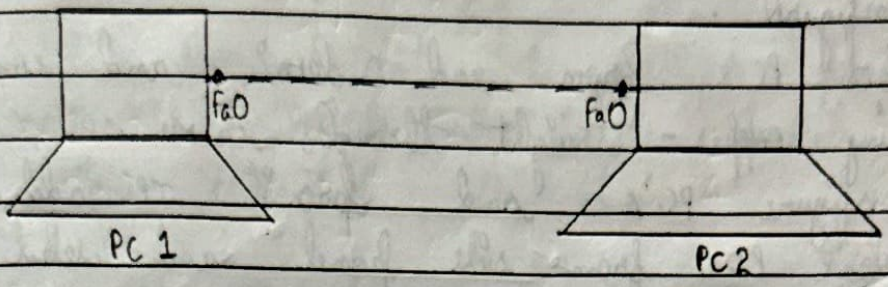
NAME: _____ STD.: _____ SEC.: _____ ROLL NO.: _____ SUB.: _____

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
		CN LAB		
1.	27-09-24	Exploring Basics	1	27/9
2.	4-10-24	LAB-2	4	4/10
3.	18-10-24	LAB-3	7.	
4.	25-10-24	LAB-4	10.	
5.	8-11-24	LAB-5	14	
6.	15-11-24	LAB-6	20.	15/11
7.	22-11-24	LAB-7	23	22/11
8.	29-11-24	LAB-8	26.	29/11
9.	20-12-24	LAB-9	31	

27-9-24

LAB - 1

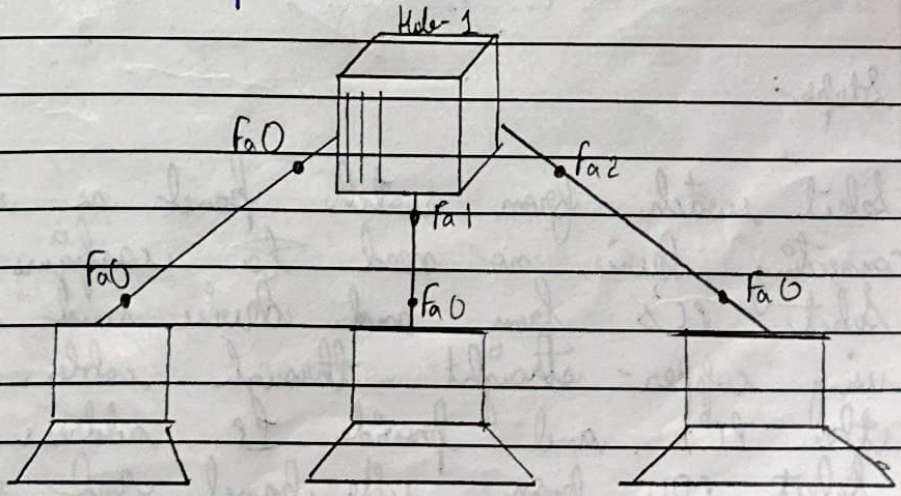
1. PC to PC:



→ Steps:

1. Select PC's from end devices and drop to the window.
2. By selecting copper cross-over wire from connection establish link b/w computers.
3. Click on computer → config → FastEthernet 0 → give IP address 10.0.0.1 the respective subnet mask will be generated. Do same for other PC as well.
4. Select PDU from side panel and select sender and receiver and observe packet tracing in simulation mode.

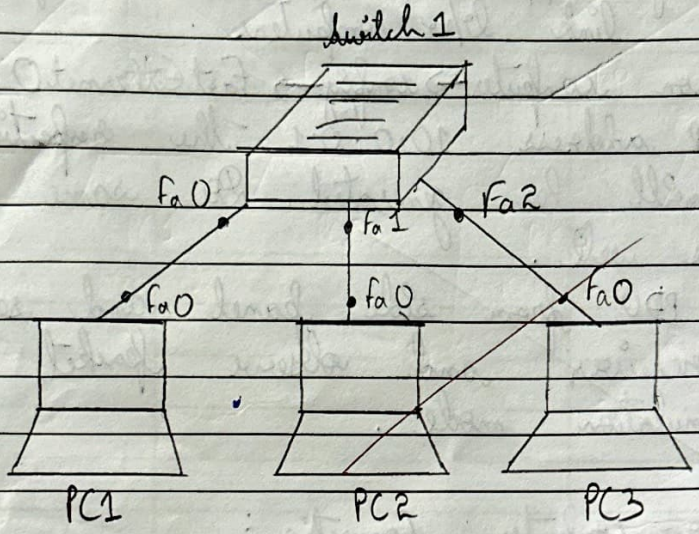
2. Hub-to-computer Connection:



Steps:

1. Select Hubs from Hubs panel as hub is connecting device therefore no need to configure.
2. Select PC's from end device and connect using copper - straight - through cable.
3. Configure PC's and provide IP address.
4. Select PDU from side panel and select sender and receiver.
5. Observe the packet transmition in simulation mode.

3. Switch - to - PC Connection.



Steps:

1. Select switch from switch panel as switch is connecting device, no need to configure.
2. Select PC's from end device and connect using copper - straight - through cable. Configure the PC's and provide IP address.
3. Select PDU from side panel and select sender and receiver.

4. Observe the packet transmission in simulator mode.

→ Observation:

1.) Hub shares data to multiple ports all together. The hub broadcasts packets to all systems connected to it.

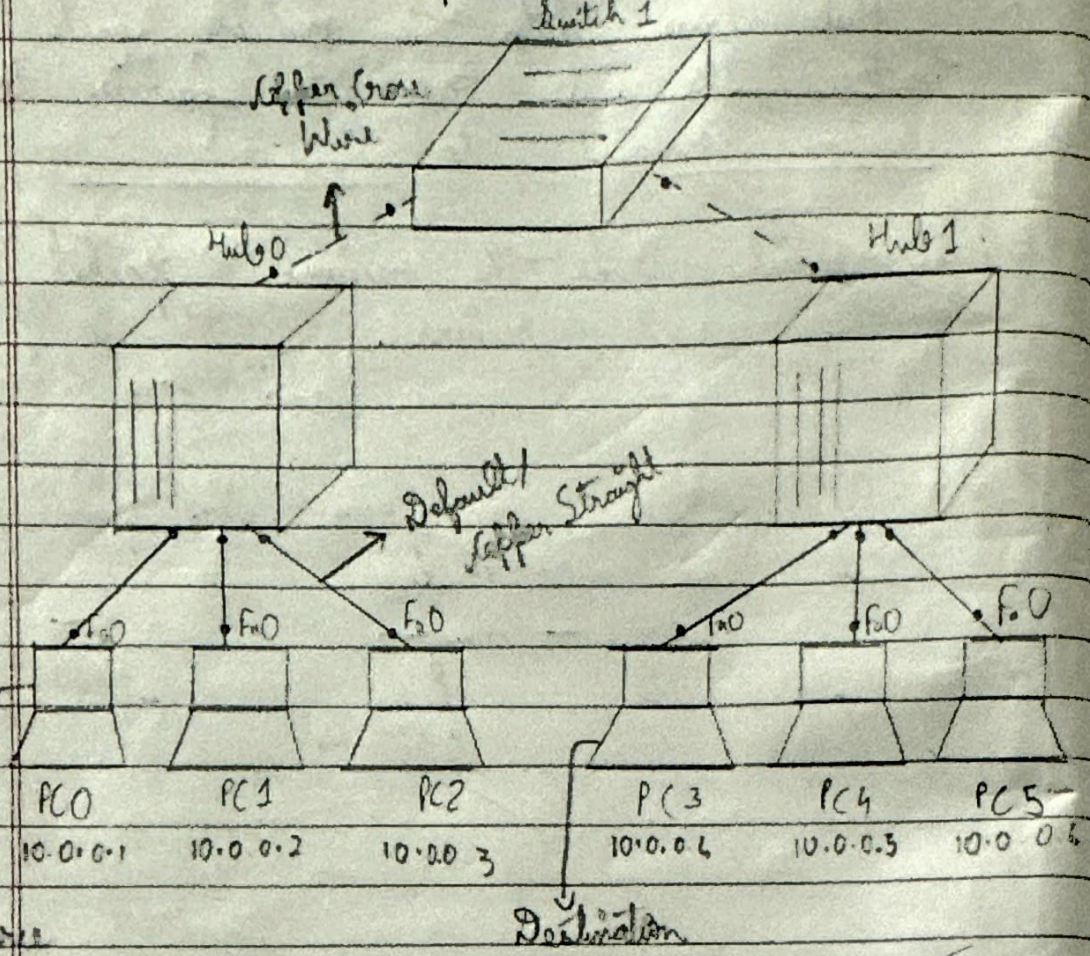
2.) The switch sends the received packet only to the selected receiver.

2719

4-10-24

LAB - 2:

1. Create a topology involving multiple hubs & a switch connecting them to a simulator to simulate a simple PDU



→ Steps:

1. Select switch from switch panel. As switch is a connecting device, no need to configure it.

2. Select 2 hubs from hubs panel. As hub is a connecting device no need to configure it.

3. Connect the switch and the hubs using copper cross wire.

4. Select 3 PC's from end device and connect configure it. Connect these 3 PC's to hub 1 through copper straight wire.

5. Select another 3 PC's from end device and configure it. Connect these 3 PC's to hub 2 through copper straight wire.

6. Select the PDU from side panel and select the sender and receiver.

→ Case 1:
Sender: PC0
Receiver: PC3

Last Status	Source	Destination	Type	Time
Successful	PC0	PC3	ICMP	6.007

→ Now sending the message using ping method.
→ Select the sender PC, then go to desktop tab and select command prompt.
→ In command prompt, type:
PC > ping 10.0.0.4
where 10.0.0.4 is the IP address of the destination PC.

Pinging 10.0.0.4 with 32 bytes of data:
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss).
Minimum = 0ms, Maximum = 5ms, Average = 1ms

→ Case 2:
Sender : PC0
Receiver : PC2

→ Select the sender PC, then go to desktop tab and choose command prompt.

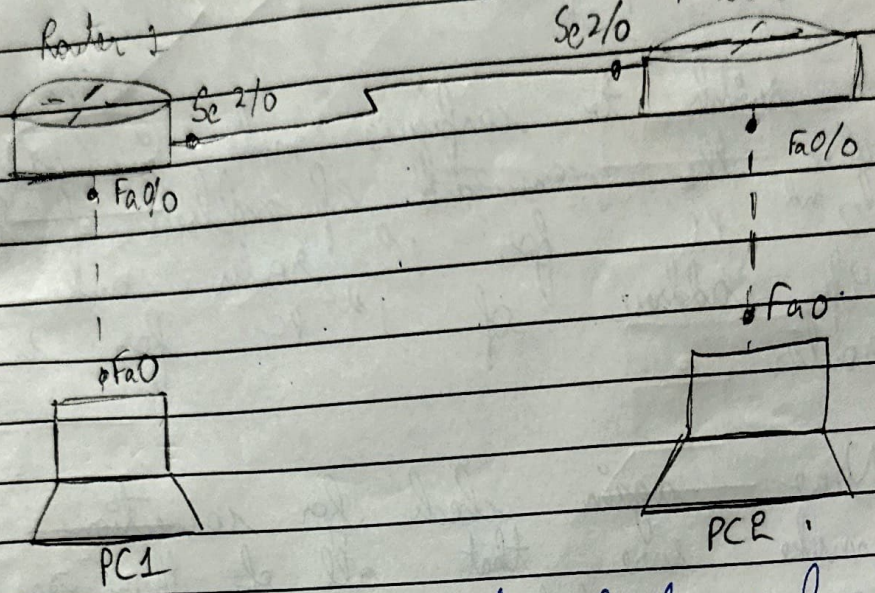
Now type.
PC > ping 10.0.0.3 → Destination PC
Pinging 10.0.0.3 with 32 bytes of data
Ping statistics for 10.0.0.3
Packets: Sent = 4 Received = 4 Lost = 0

4/10/24

18-10-24

LAB - 3:

1. Create a topology involving 2 routers & 2 PC's. Configure IP address to routers in packet tracer. Explore following messages: a.) ping messages, destination unreachable, request timed out & reply.



1. Select 2 routers from router panel and 2 PC's from end devices.

2. Configure the PC's and connect these PC's to one router each using a copper cross-over wire.

3. Connect the 2 routers using a serial-DCE wire.

4. Configure each router in the CLI tab using the following commands:

Router>> enable
config t
interface FastEthernet 0/0
ip address 20.0.0.1 255.0.0.0
no shutdown
exit.

5. Give the gateway for the PC's with the IP address of their respective routers.

6. Now check for the ip route in both the routers in ~~comm~~ CLI by going the following commands:
   show ip route.

7. Now switch to configure mode in CLI and give the ~~command~~ ip address of the 2nd PC for 1st router and ip address of 1st PC for 2nd router.

8. Now again check for connections and make sure that all of them are ~~con~~ connected to each other.

9. Now go to command prompt from the sender PC and give the following command:
   ⇒ ping 20.0.0.1
   ↳ destination PC.

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1 : bytes: 32 time = 6 ms TTL = 12
Reply from 20.0.0.1 : bytes: 32 time = 1 ms TTL = 12
Reply from 20.0.0.1 : bytes: 32 time = 5 ms TTL = 12
Reply from 20.0.0.1 : bytes: 32 time = 7 ms TTL = 12

Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in ms:
Min = 1ms, Max = 7 ms, Avg = 4 ms.

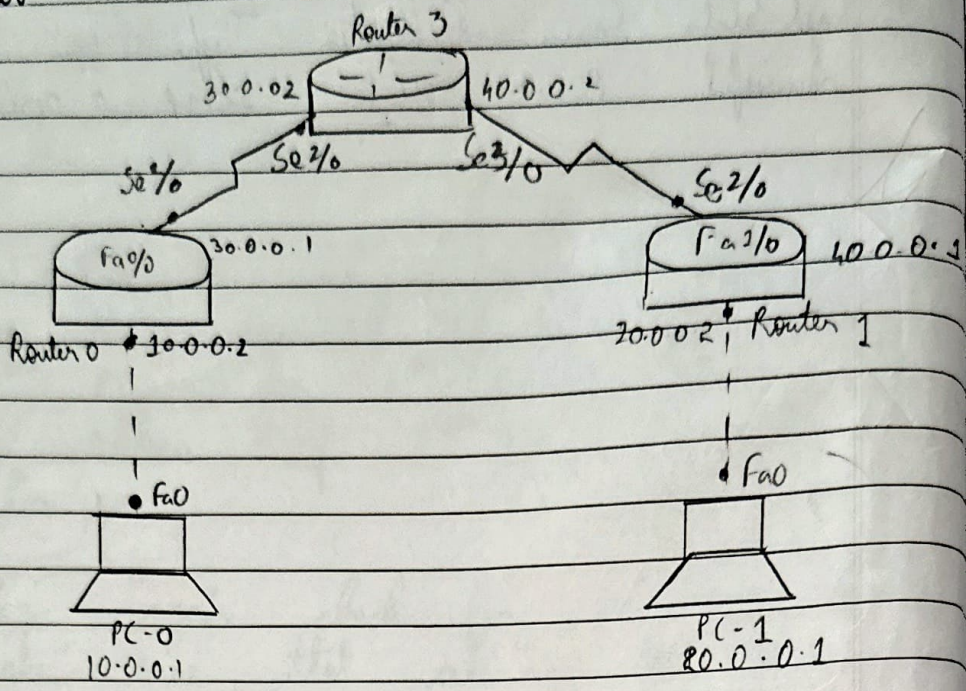
10. After this send a package from source PC to receiver PC and observe the result.

Last Status	Source	Destination	Type	Time
Successful	PC1	PC2	ICMP	0.000

25-10-24

LAB - 4:

1. Configure default route, static route to the router.



1. Select 2 PC's and 3 routers from ~~end~~ devices panel & router panel respectively.

2. Connect the routers using a Serial-DCE wire.

3. Connect the PC's to the routers using copper cross-over wire & configure them by giving IP-addresses.

4. Configure the routers in CLI:

From Router: Enable.

>> config t

~~>> ip route 10.0.0.2 255.0.0.0~~

>> interface fastethernet 0/0

>> ip address 10.0.0.2 255.0.0.0

>> no shutdown

>> exit.

Date
Page 11

5- Now give the gateway in PC of their respective routers.

6. Router 0 has to be configured with the 40.0.00 network and 20.0.0.0 network. So in CLI configure them:

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2

ip route 20.0.0.0 255.00.0 30.0.0.2

where 40.0.0.0 & 20.0.0.0 → unknown networks.
255.0.0.0 → subnet mask.
30.0.0.2 → Port through which connection is established.

7.) Router 1 has to be configured with 30.0.0.0 and 10.0.0.0 network. So in CLI.

ip route 10.0.0.0 255.0.00 40.0.0.2
ip route 30.0.0.0 255.0.0.0 40.0.0.2.

8.) Router 3 has to be configured with PC0 & PC-1. So in CLI:

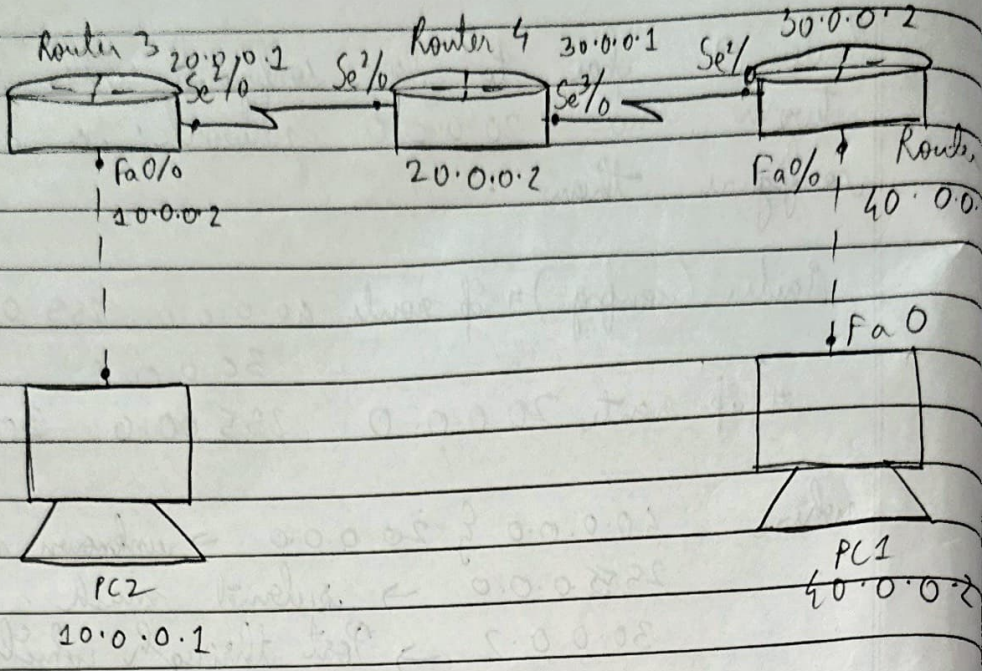
ip route: 10.0.0.0 255.0.0.0 30.0.0.1
ip route: 20.0.0.0 255.0.0.0 40.0.0.1

9.) Now send the PDU at will from PC0 to PC1.

10.) Ping in command prompt & check.

11.) If message is sent, then it is successful.

2. Default:



1. Select 3 routers (Generic) from router pool & 2 PC's from end devices.
2. Connect the PC & router with copper crossover wire & router - router with Serial - DCE cable.
3. Configure IP addresses of the PC's.
4. Configure IP addresses of the routers in CLI
5. Set the gateways of PC-1 & PC-2.
6. For proper connectivities, we need to connect the routers with unknown routers IP address & subnet mask.

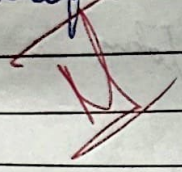
7. Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router # show ip route
C 10.0.0.0/8 is directly connected.
S* 0.0.0.0 via 20.0.0.2.

8. Similarly for router 4 & 5 as well do the same.

9. Once all configurations are completed, send message from PC2 to PC3.

10. Observe the passage.

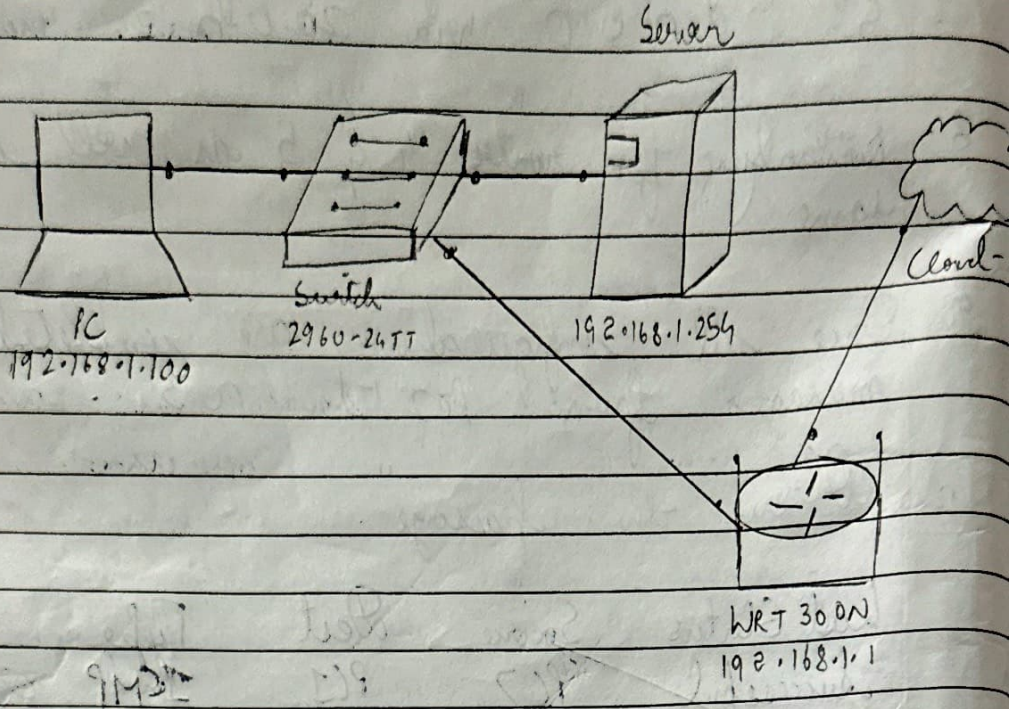
Last Status	Source	Dest	Type	Time
Successful	PC2	PC1	ICMP	0.0.1



8-11-24

LAB-5:

1. DNS:



1. We create a LAN by taking a PC, a switch and a server.

2. We will connect them using a copper straight through cable and configure them.

3. For PC give 192.168.1.200 as address and for Server give 192.168.1.254 as address. Also give these addresses as labels.

4. For server make sure port status is on in Config → FastEthernet0

5. Now ping from PC to server to check the connection by following this:
   PC → Desktop → Command prompt.
   » ping 192.168.1.254.
   This should be successful.

6. Now click one server → ~~config~~ → service → HTTP and see the HTML code.

7. Go to PC → Desktop → Web browser. Add the server IP address & click go to see the web page.

Part - 2.

1. We will add a Wireless Router and a cloud to the same ~~configuration~~ LAN.

2. Now connect the switch to the router using a straight through cable and switch to the router using a automatic choose connection type cable.

3. Now click on router → config → Display name. Change name to 192.168.1.1. Now click on LAN option on left and change IP address to 192.168.1.1.

4. Now ping from PC to computer and check if it successful.

5. Click on PC → Desktop → IP Configuration. → Add address of router as default gateway.

6. Click on server → ~~Desktop~~ Add gateway address as 192.168.1.1. Now go to service and HTTP.

7. In the index.html file edit the names to super yahoo. and save it.

8. Now go to DNS and turn it on.

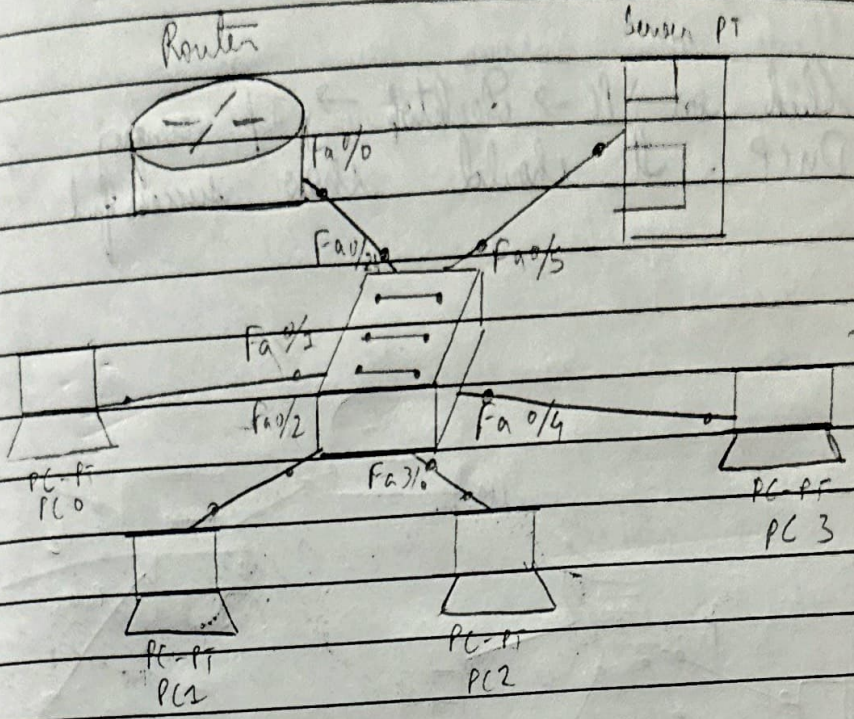
9. Under name type give superyahoo.com. Under address type: 192.168.1.254. Click add.

10. Now click on PC → IP configuration → DNS. Add server address under server type.

11. Now in web browser in PC, type superyahoo.com and the webpage is displayed.

2. DHCP:

1.



1. Create a LAN like this.

2. Configure router with ip 10.0.0.1 & subnet mask 255.0.0.0

3. Click on server → config → then assign gateway 10.0.0.1.

4. Click on Fastethernet & assign ip address and subnet mask. - 10.0.0.2 & 255.0.0.0.

5. Click on DHCP, Add in default pool give gateway as 10.0.0.1.

6. In DNS server, give server ip as 10.0.0.2.

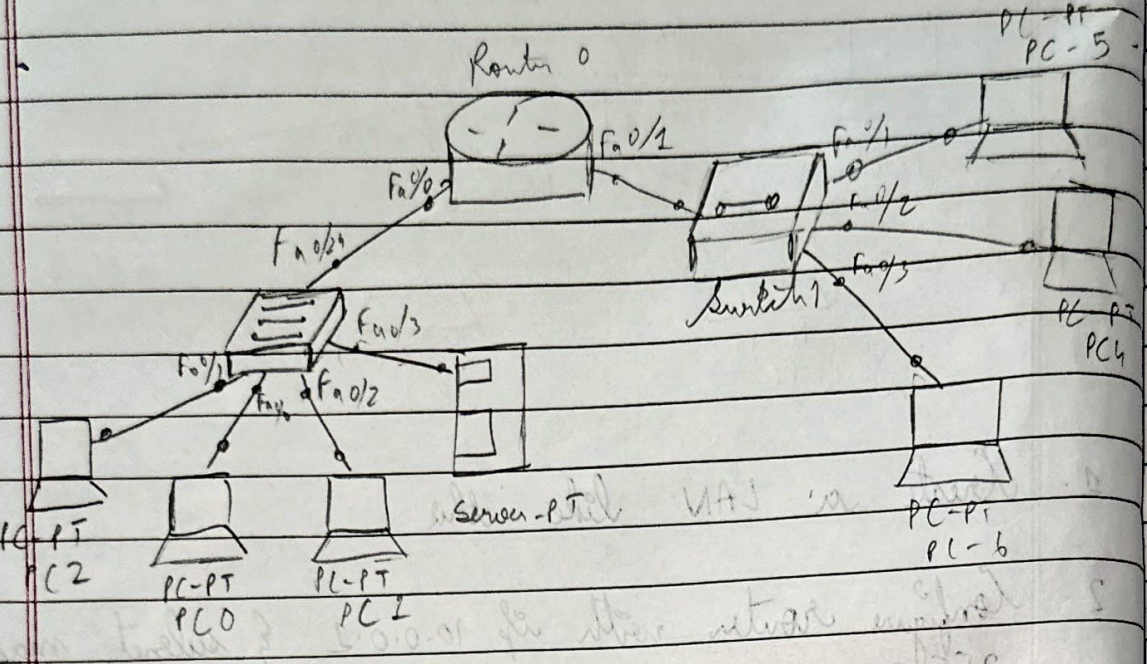
7. Edit start ip address to 10.0.0.10 & subnet mask 255.0.0.0

8. In max no. of users give 500 has pool of address.

9. Assign TFTP server ip address as 10.0.0.2 & save it.

10 Click on PC → Desktop → Ip config. & choose DHCP. It should show successful.

2.)



1. Create a topology like this.

2 Configure the router interface fastethernet 0/0 & fastethernet 0/1 with ip address 10.0.0.1 & 20.0.0.1 in config-terminal.

3 Click on Server → config → give ip address & gateway as 10.0.0.1

4. Click on fastethernet, assign ip address as 10.0.0.2 & subnet mask as 255.0.0.0. DHCP server automatically assign 10 network for default pool. Give ip address for DNS, gateway & TFTP, configure and save.

5. Click on any PC in a LAN with Server & check whether DHCP is working.

6. Click on PC → Desktop → IP config → choose DHCP to get ip from dhcp server for that PC.

7. Click on Server → Config → DHCP. Add Pool Name with 20 Network. (For PC in network without Server).
Default gateway → 20.0.0.1.
DNS Server → 10.0.0.2
Starting IP Address → 20.0.0.10
Subnet Mask → 255.0.0.0
Maximum Number of User → 100
TFTP Server 10.0.0.2
Click on add & save.

8. Go to router & give server address as 10.0.0.2 under fastethernet 0/1.

Router (config)# interface fastethernet 0/1
Router (config-if)# ip helper-address 10.0.0.2
Router (config-if)# exit

9. Click on PC → Desktop → IP config → DHCP. We get IP address from DHCP server.

15-11-24

LAB-6:

1 Write a program for error detection using CRC-CCITT (16 bits)

```
def xor_operation(dividend, divisor):
    result = []
    for i in range(len(divisor)):
        result.append('0' if dividend[i] == divisor[i] else '1')
    return ''.join(result)

def crc_encode(message, poly):
    padded_message = message + '0' * (len(poly) - 1)
    for i in range(len(message)):
        if padded_message[i] == '1':
            padded_message = padded_message[:i] +
                xor_operation(padded_message[i:i + len(poly)],
                poly) + padded_message[i + len(poly):]
    crc_remainder = padded_message[-(len(poly) - 1):]
    return message + crc_remainder

def crc_check(received_message, poly):
    for i in range(len(received_message) - len(poly) + 1):
        if received_message[i] == '1':
            received_message[i:i+len(poly)] = (
                received_message[:i] +
                xor_operation(received_message[i:i + len(poly)],
                poly) + received_message[i + len(poly):]
            )
    remainder = received_message[-(len(poly) - 1):]
    return remainder == '0' * (len(poly) - 1)
```

```
def main():
    poly = '10001000000100001'
    message = input("Enter input message in binary:")
    transmitted_message = crc_encode(message, poly)
    print(f"Transmitted Message: {transmitted_message}")
    received_message = input("Enter received message in binary:")
    if crc_check(received_message, poly):
        print("No error in data transmission")
    else:
        print("Error in data transmission has occured")

if __name__ == "__main__":
    main()
```

O/P:

Enter input message in binary: 10111
Transmitted message: 10111010001011010110
Enter received message in binary: 10111011000101101 0111
Error in data transmission has occured.

2. Leaky Bucket Algorithm:

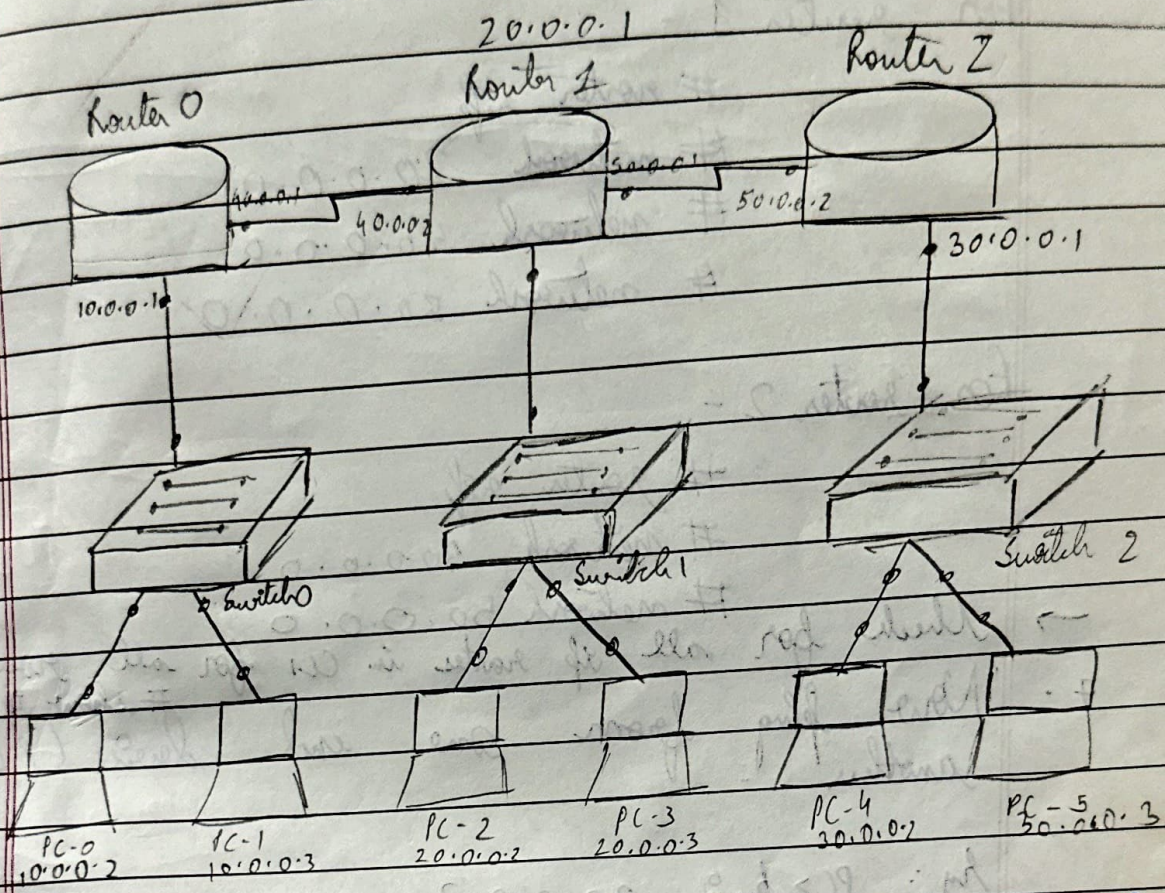
```
no_of_queries = int(input("Enter total no of queries:"))
bucket_size = int(input("Enter bucket size:"))
input_pkt_size = int(input("Enter no. of packets entering
                            bucket at a time:"))
output_pkt_size = int(input("Enter no of packets exiting
                             bucket at a time:"))
storage = 0
for i in range(no_of_queries):
    size_left = bucket_size - storage
    if input_pkt_size <= size_left:
        storage += input_pkt_size
    else:
        print(f"Packet loss = {input_pkt_size - size_left}")
    print(f"Buffer size = {storage} out of bucket size =
          {bucket_size}")
    storage -= output_pkt_size
    if storage < 0:
        storage = 0
```

O/P: Enter total no of queries: 5
Enter bucket size: 20
Enter no of packets entering bucket at a time: 6
Enter no of packets exiting bucket at a time: 4
Buffer size = 6 out of bucket size = 20
Buffer size = 8 out of bucket size = 20
Buffer size = 10 out of bucket size = 20
Buffer size = 12 out of bucket size = 20
Buffer size = 14 out of bucket size = 20

22-11-24

LAB-7:

1. RIP:



1.) Create a topology with 3 routers, 3 switches & 6 PC's.

2.) Configure the PC's with ip address.

3.) Configure the routers in CLI or Config and give the router ip address as gateway for the respective PC's.

4.) Now connect each router with each other.

5.) Now connect router to all other networks.

6. For router 0 go to CLI -
>> Router <config> # router rip
network 40.0.0.0
network 10.0.0.0

For router 1 -
router rip
network 20.0.0.0
network 40.0.0.0
network 50.0.0.0

For router 2 -
router rip
network 30.0.0.0
network 50.0.0.0

→ Check for all rip router in CLI for all routers. # show ip rou

7. Now ping from one end device (PC) to another.

Eg: PC> ping 30.0.0.2

Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

8.) Successful sending & receiving of packet.

9.) Now in simulation mode, send a packet from one PC in one network to another PC in another network. Click Auto Capture / Play.

10.) Click on the Info on left in LMS (color block) of router to router message sent and observe that in inbound

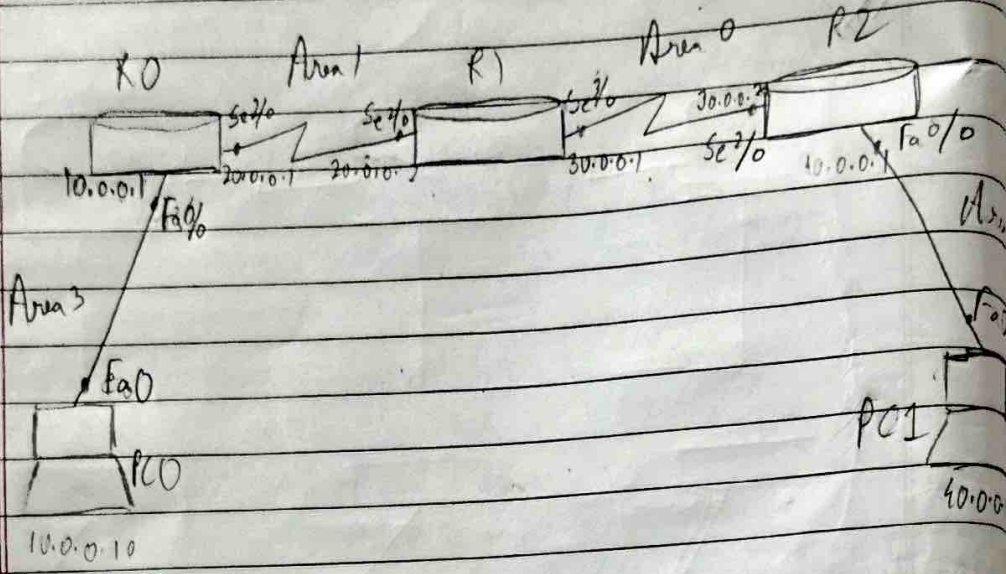
details and outbound PDU details, the TTL value has a difference of 1. i.e

Eg:- Inbound PDU - TTL = 253.
Outbound PDU - TTL = 252.

29-11-23

LAB - 8

OSPF:



1). Create a topology like this and configure ip address for PC's.

2. Configure ip address to all interface:

In Router R1,

R1(config)# interface fastethernet 0/0.
R1(config-if)# ip address 10.0.0.1 255.0.0.0
" no shutdown
" exit

R1(config)# interface serial 2/6
R1(config-if)# ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shutdown
exit.

In Router R2:

R2(config)# interface serial 2/0.
R2(config-if)# ip address 20.0.0.2 255.0.0.0.
" # encapsulation ppp.
no shutdown.
exit.

R2(config)# interface serial 3/0.
R2(config-if)# ip address 30.0.0.1 255.0.0.0.
" # encapsulation ppp.
clock rate 56000.
no shutdown.
exit.

In router R3:

R3(config)# interface serial 2/0.
R3(config-if)# ip address 30.0.0.2 255.0.0.0
encapsulation ppp.
no shutdown.
exit.

R3(config)# interface fastethernet 0/0.
R3(config-if)# ip address 40.0.0.1 255.0.0.0
no shutdown.
exit.

3. 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
" #network 10.0.0.0 0.255.255.255 area 3.
#network 20.0.0.0 0.255.255.255 area 1.
#exit.

In router R2,
R2(config)#router ospf 1.
R2(config-router)#router-id 2.2.2.2
#network 20.0.0.0 0.255.255.255 area 1
#network 30.0.0.0 0.255.255.255 area 0
#exit.

In Router 3:
R3(config)#router ospf 1.
R3(config-router)#network 30.0.0.0 0.255.255.255 area 0
#network 40.0.0.0 0.255.255.255 area 2
#exit.

Configure router-id when we configure ospf.

4. Now check routing table of R1,

Router# show ip route.

R2 knows Area 0. Network 20.0.0.0 connected to R2 from R1. So R1 learns network through this network.

R3(config)# router ospf 1, Here 1 is Process ID.

Now we configure loopback address to routers.

R1(config-if)# interface loopback 0
ip add 172.16.1.252 255.255.0.0.
no shutdown.

R2(config-if)# interface loopback 0
ip add 172.16.1.253 255.255.0.0.
no shutdown.

R3(config-if)# interface loopback 0
ip add 172.16.254 255.255.0.0.
no shutdown.

5. Now check routing table of R3, R3 doesn't know area 3, so we create virtual link b/w R1 & R2.

6. Create virtual link b/w R1, R2 by this we create a virtual link to connect area 3 to area 0.

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

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

7. R2 & R3 get updates about area 3.
Let us check.

8. Check connectivity b/w host 10.0.0.1.0 to 40.0.0.10 by going to command prompt and pinging destination PC.2

PC> ping 40.0.0.10.

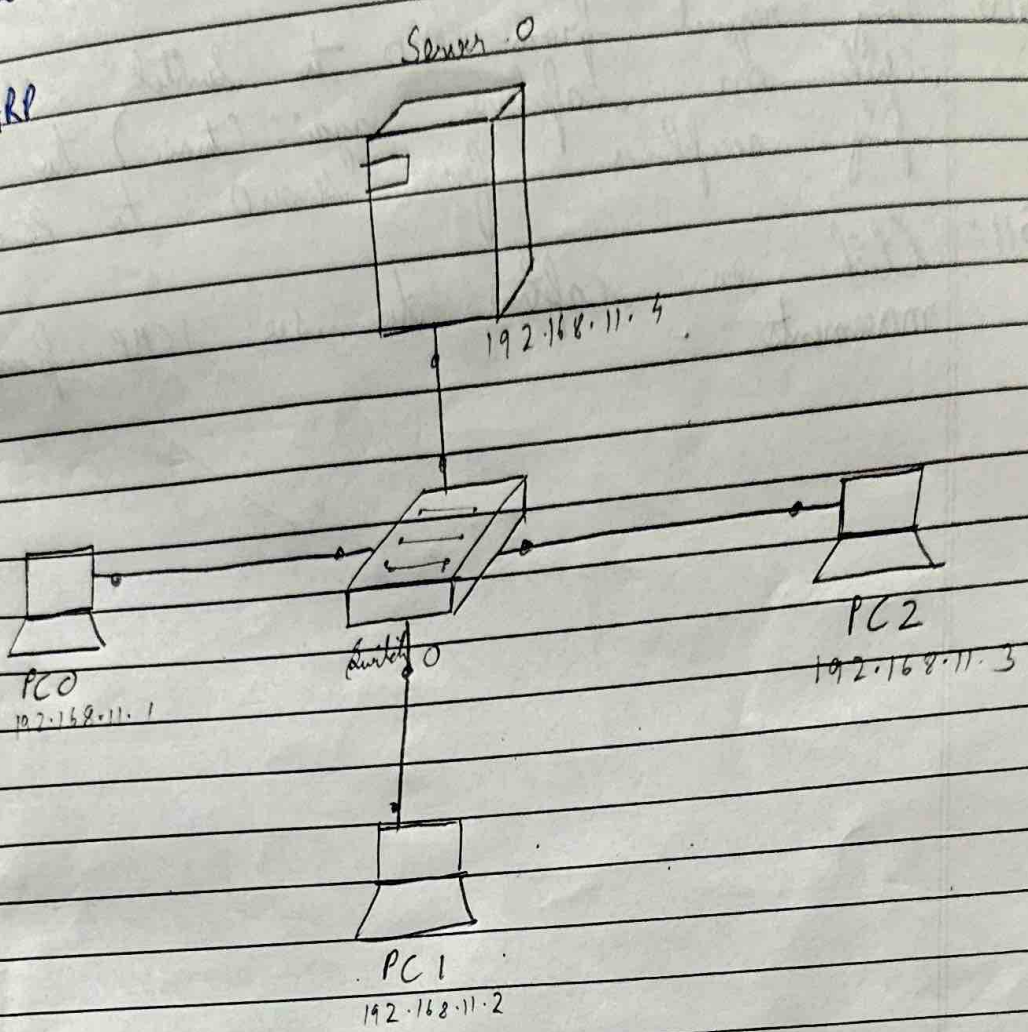
Packets: Sent=4, Received=4, Lost=0 (0% loss).

9.) Successful communication.

20-12-24

LAB-9:

1.) ARP



S1: Assign IP addresses to all PCs & Server

S2: Go to Simulation panel, click on Inspect & right click on PC0

S3: Notice that there are no entries in the ARP table.

S4: Repeat the same for server.

S5: Click on PC0 & go to command prompt.
Type >arp -a
>ping 192.168.11.4
(Ping from PC0 to Server)

S6: 2 packets are created ICMP and ARP.

S7: Hover over the packets to check the type of packet.

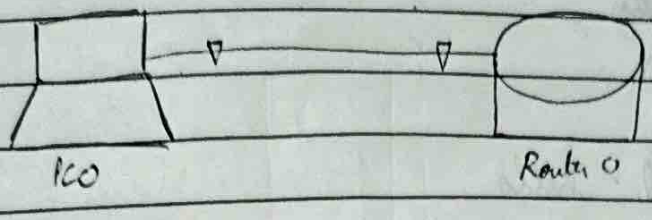
S8: Click on ARP packet.

S9. Click on capture button to start the simulation.

S10: Ping request from PC0 to Switch. Click on capture again (twice) to see ping acceptance from Server0 to PC0.

S11: Click on capture to see ICMP packet movements.

2. TELNET Protocol:



1. Create the topology and configure IP address & gateway for PC0

2) Configure the router in CLI

3) Go to PC → Command Prompt and ping.

```
Router > en
Router #
Router # config t
Router (config) #
Router (config) # hostname R1
R1 (config) # enable secret rp
R1 (config) # int g0/0/0
R1 (config-if) # ip add 192.168.1.1 255.255.255.0
R1 (config-if) # no shut.
R1 (config-line) # password login
R1 (config-line) # password tp.
R1 (config-line) # exit
R1 (config) #
R1 (config) # exit.
```

3) Go to PC → Command Prompt

```
>ping 192.168.1.1
- - - - -
>telnet 192.168.1.1
Trying 192.168.1.1 ... Open
```

User Access Verification

Password: #fuse

~~Password~~

R1>

R1>en

Password: hello

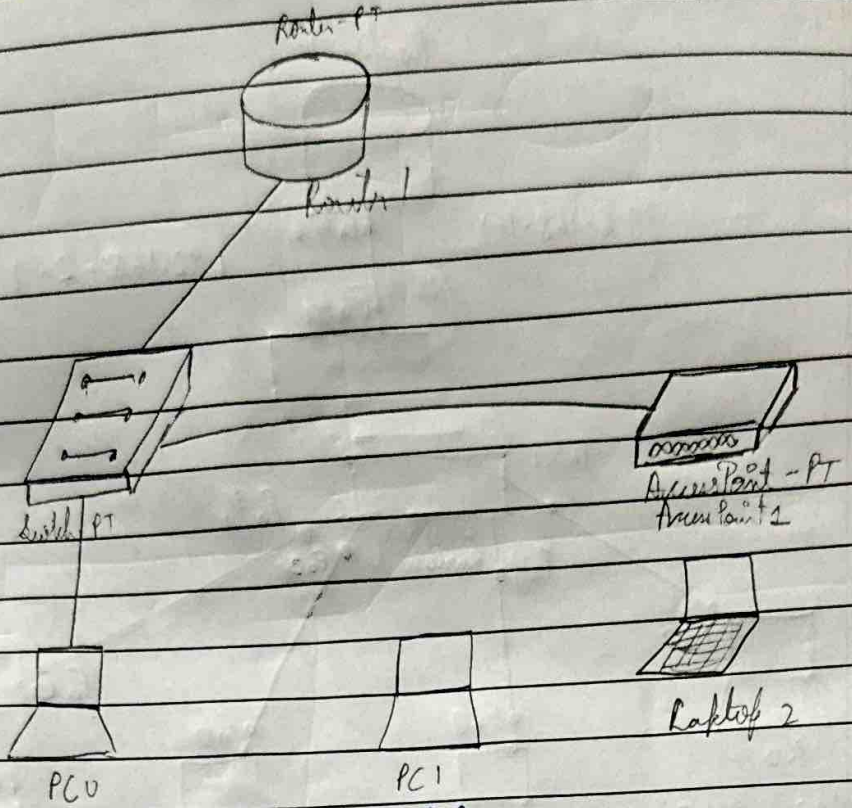
R1#

R1#..—

R1#.---

R1#--··

3. WLAN:



1.) Construct the above topology.

2.) Configure PC0 & Router1 as normally done.

3.) Configure Access Point 1 - Port1 -> SSID None - Any Name (WLAN here)

4.) Select WEP & give 10 digit hex key - 1234567890

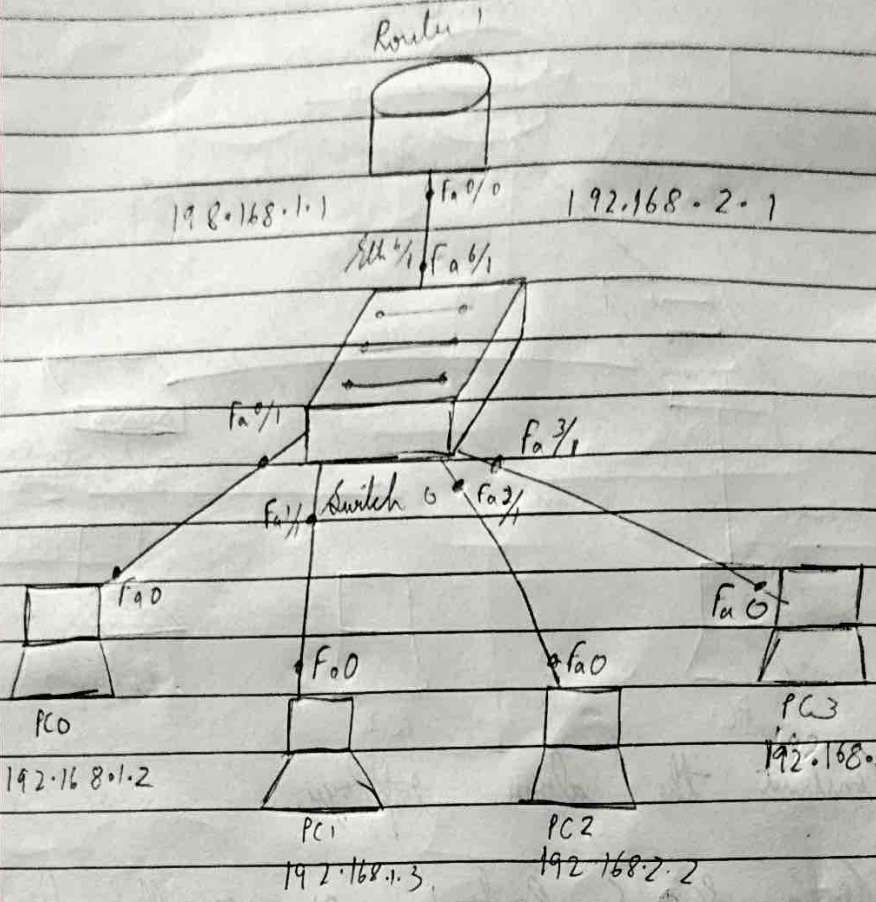
5.) Configure PC4 & Laptop with wireless standards.

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

7.) In the config tab, a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address & Gateway to the device.

8.) Ping from every device to every other device & see results.

4. Virtual LAN:



1.) Create a topology like this.

2.) In the switch, go to Config tab and select VLAN Database.

3.) Give any VLAN number (eg 2) and include any name. Click on ADD.

4.) Select the interface i.e. 6/1 near switch from router and make it the trunk.

5.) VLAN 2 ~~New~~ NEWVLAN systems have to be looked at in the interface of the switches.

6. Go to the config tab of router and select VLAN DATABASE. Enter the number and

name of vlan created. Go to CLI.

Router (vlan)#int
APPLY completed
Exiting...
Router#config t
Router(config)#interface fastEthernet 0/0.1
Router(config-subif)#
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 192.168.2.1 255.255.255.0
Router(config-subif)#no shut.
Router(config-subif)#exit.
Router(config)#exit.

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

ClientTCP.py

```
from socket import *
serverName = "127.0.0.2"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
serverSocket.listen(1)
sentence = input("\n Enter file name:")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('\n From 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('\n Sent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

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

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))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print("\n Reply 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("\n Sent contents of", end=" ")
    print(sentence)
    # for i in sentence:
    #   print(str(i), end="")
file.close()
```