VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

Ayush Girish Gaonkar(1BM22CS063)

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

Sep 2024-Jan 2025

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **Ayush Girish Gaonkar(1BM22CS063),** 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 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks Lab - (23CS5PCCON)** work prescribed for the said degree.

Dr. Nandhini Vineeth	**Dr. Kavitha Sooda**
Associate Professor, Department of CSE, BMSCE, Bengaluru	Professor and Head, Department of CSE BMSCE, Bengaluru

Index

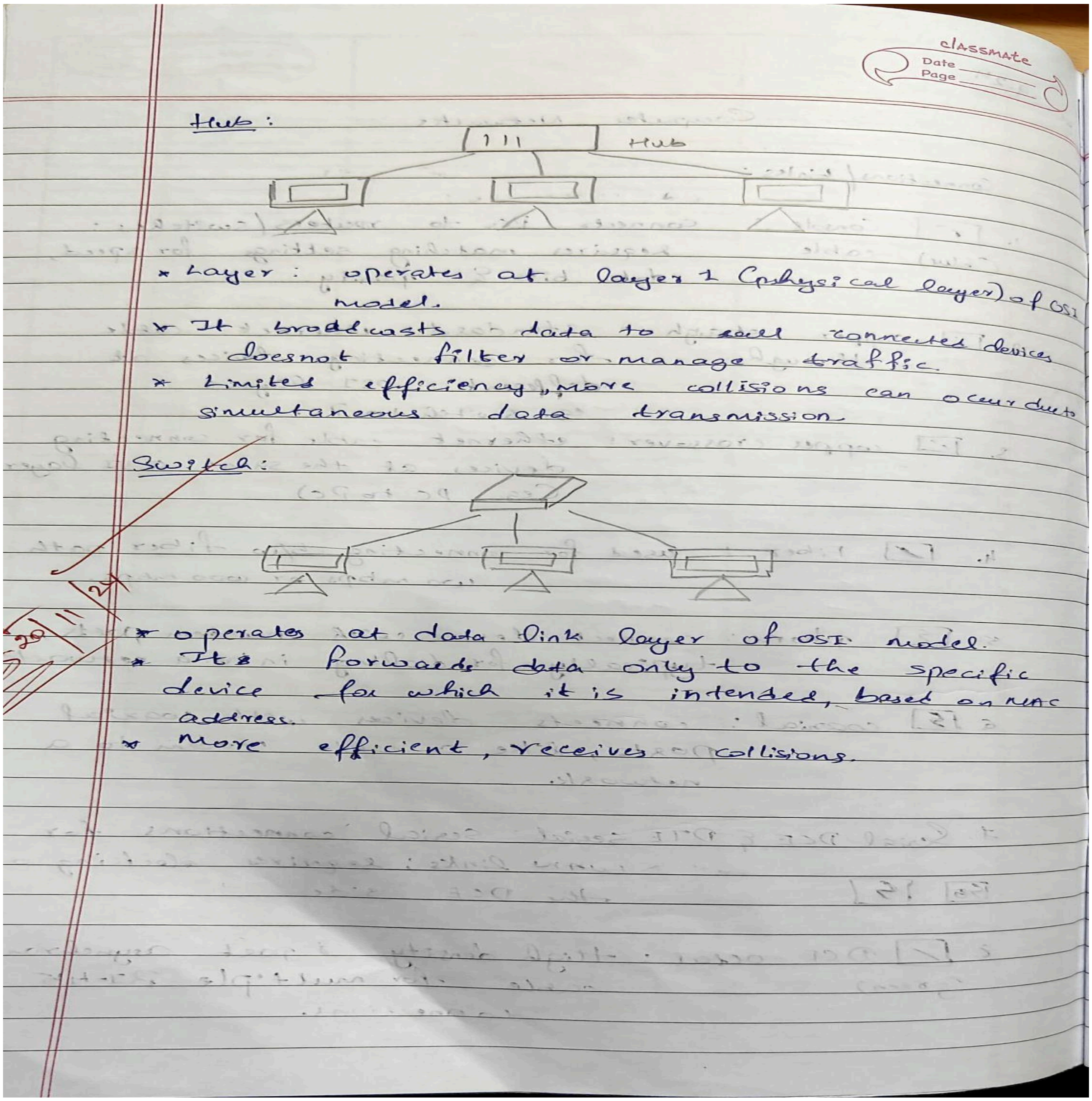
Sl. No.	Date	Experiment Title	Page No.
1	09-10-24	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.	1-4
2	09-10-24	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	5-8
3	16-10-24	Configure default route, static route to the Router (Part 1).	9-11
4	23-10-24	Configure default route, static route to the Router (Part 2).	12-15
5	13-11-24	Configure DHCP within a LAN and outside LAN.	16-20
6	20-11-24	Configure RIP routing Protocol in Routers .	21-24
7	20-11-24	Demonstrate the TTL/ Life of a Packet.	25-27
8	27-11-24	Configure OSPF routing protocol.	28-30
9	18-12-24	Configure Web Server, DNS within a LAN.	31-32
10	18-12-24	To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	33-35
11	18-12-24	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	36-38
12	18-12-24	To construct a VLAN and make the PC’s communicate among a VLAN.	39-42
13	18-12-24	To construct a WLAN and make the nodes communicate wirelessly.	43-46
14	18-12-24	Write a program for error detecting code using CRC-CCITT (16-bits).	47-48
15	18-12-24	Write a program for congestion control using Leaky bucket algorithm.	49-50
16	18-12-24	Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	51-51
17	18-12-24	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.	52-53
18	18-12-24	WIRESHARK	54-55

Github Link:
https://github.com/ayush00070/CN_LAB

Program 1

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

Topology , Procedure and Observation:



Hub:

Hub

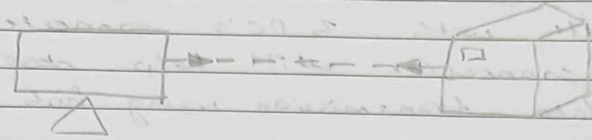
* Layer: operates at layer 1 (physical layer) of OSI model.
* It broadcasts data to all connected devices, doesnot filter or manage traffic.
* Limited efficiency, more collisions can occur due to simultaneous data transmission.

Switch:

* operates at data link layer of OSI model.
* It forwards data only to the specific device for which it is intended, based on MAC address.
* More efficient, reduces collisions.

Experiment

1. PC to Server



PC - PT
PC 0
10.0.0.1

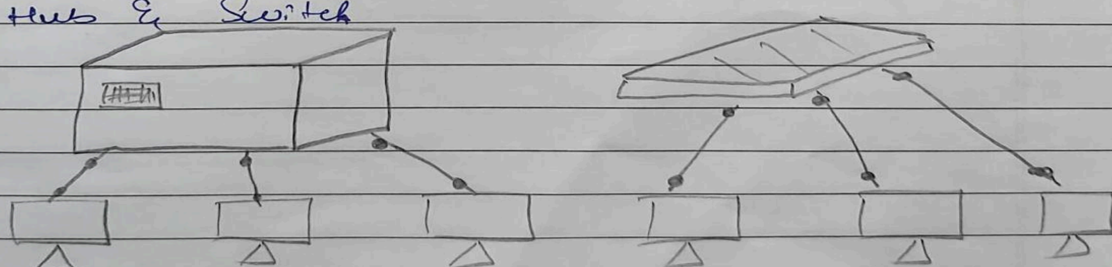
Server - PT
Server 0
10.0.0.2

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

Topology: A PC is connected to server using a crossover ethernet cable.
IP address of PC - 10.0.0.1, server - 10.0.0.2

Observation: Direct communication allows PC to communicate with server, which is typical in small networks for tasks such as file sharing, service requests or testing server responses to client queries

2. Hub & Switch



PC-PT	PC-PT	PC-PT	PC-PT	PC-PT	PC-PT
PC0	PC1	PC2	PC3	PC4	PC5
10.0.0.6	10.0.0.7	10.0.0.8	10.0.0.11	10.0.0.12	10.0.0.1

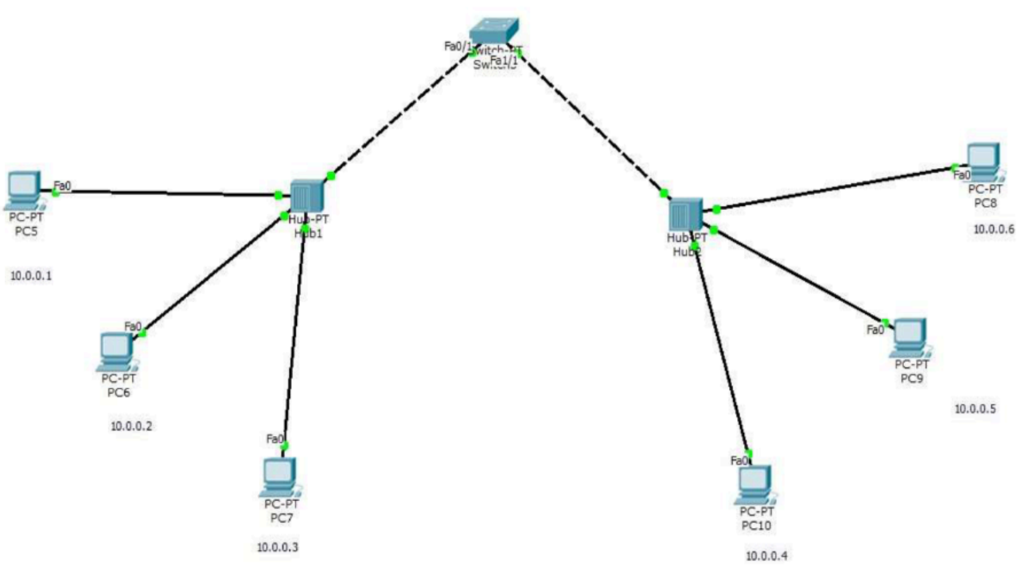
Aim - To create simple network consisting of 3 PCs & connected to a central hub & another network with 3 PC's connected to a switch, this connection will help observe the behaviour of data transmission using hub & switch devices.

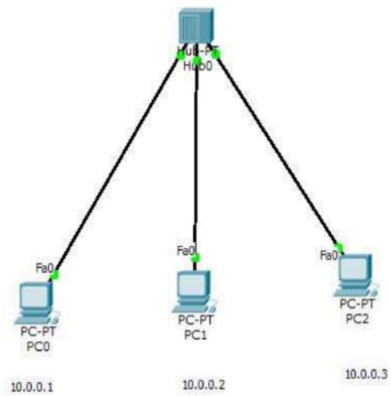
Topology: 3 PCs are connected to a hub & switch using straight through ethernet cables.

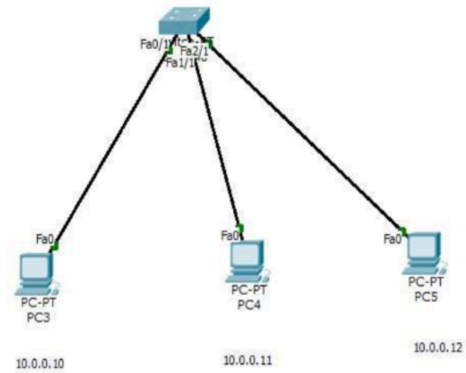
Observation: Hub broadcasts packets to all devices which may cause unnecessary traffic.

Switch forwards packets only to appropriate device by learning MAC address, making it more efficient in reducing traffic.

Screen Shots:




Hub-PT
Hub0
Fa0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
10.0.0.1
10.0.0.2
10.0.0.3


Fa0/1
Fa2/1
Fa1/1
Fa0
Fa0
Fa0
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
10.0.0.10
10.0.0.11
10.0.0.12

Program 2

Aim: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

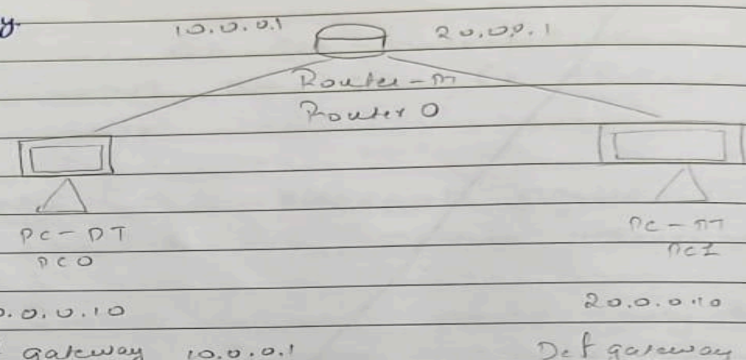
Topology , Procedure and Observation:

Lab-2

09/10/2024

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

Topology.



1. PC0: connected to router's interface fa0/0 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

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

classmate
Date ______
Page 7

Procedure: (1) Two PCs (PC0 & PC1) are assigned with IP 10.0.0.10 & 20.0.0.10 & gateway 10.0.0.1 & 20.0.0.1 respectively.

(2) Open CLI in router & enter the following
Router > enable
Router # config t
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
B
Router (config) # interface fastethernet 1/0
Router (config-if) # ip address 10.0.0.1
Router (config-if), no shutdown
exit

(3) Ping another system or interface from the command prompt of PC0 or PC1 using command
>ping 10.0.0.10

Observation
Command prompt gives output
Pinging 10.0.0.10 with 32 bytes of Data

ping statistics for 20.0.0.10
Packets ~~Sp~~ Sent = 4, Received = 4, lost = 0
(0% loss)
Approximate round trip times in milli-seconds:
Minimum = 0 ms, maximum = 0 ms, Average = 0 ms

~~The following~~
Router > enable
Router # show ip route

Codes: C - connected

~~gateway of~~

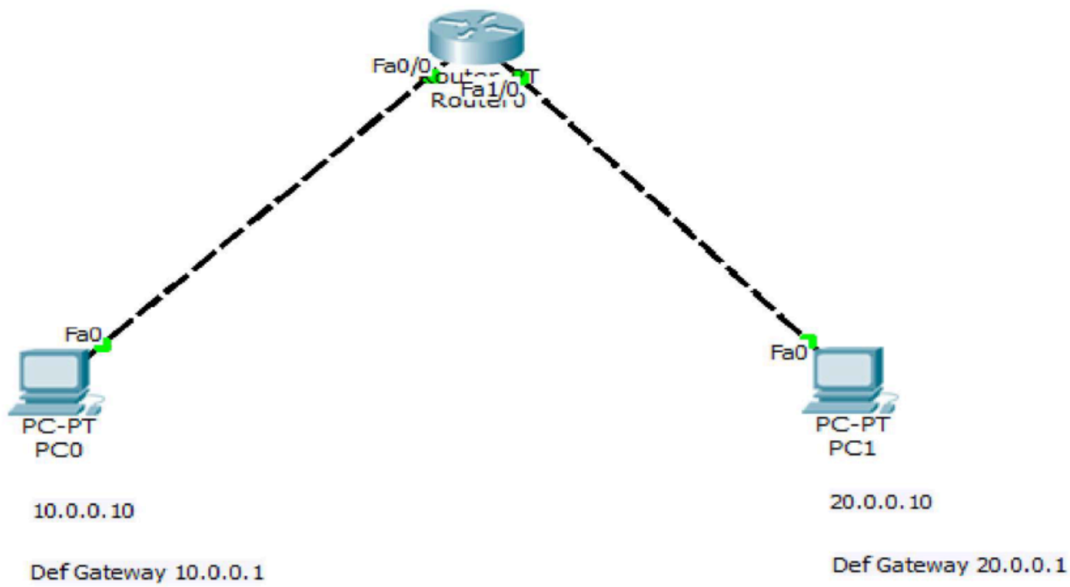
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected, FastEthernet 1/0

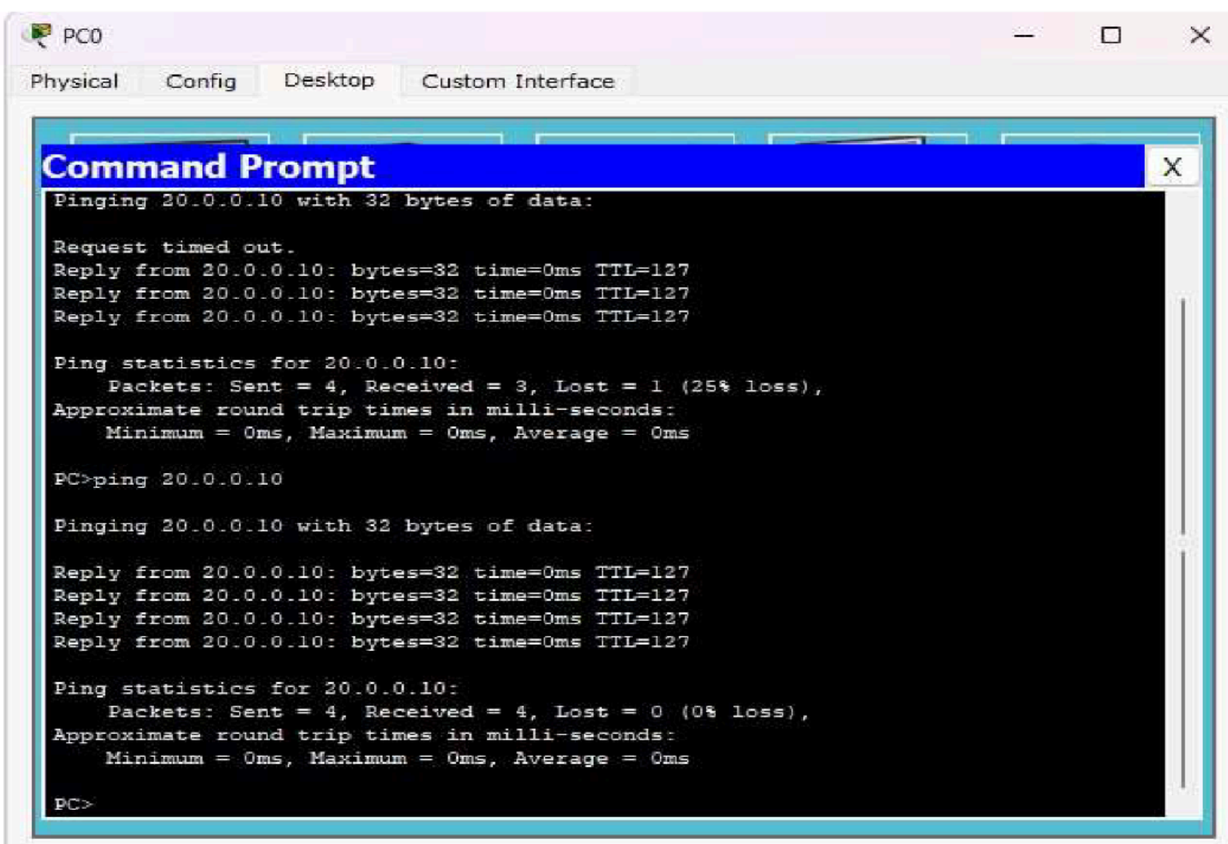
Procedure:-

(1) Two PC are selected, PC0 & PC1
(2) A generic Router is selected.
(3) The two PC's and router are connected using copper cross over.
(4) It is labelled with the gateway & IP address.
(5) Procedure is executed in real time mode in order to do PINGS
(6) After the Router is configured CLI the connections from red to green indicating that the router is configured & connection is established.
(7) Then you PING PC0 to PC1 & PC1 to PC0.

20/11/24

Screen Shots:





Program 3

Aim: Configure default route, static route to the Router(Part 1).

Topology , Procedure and Observation:

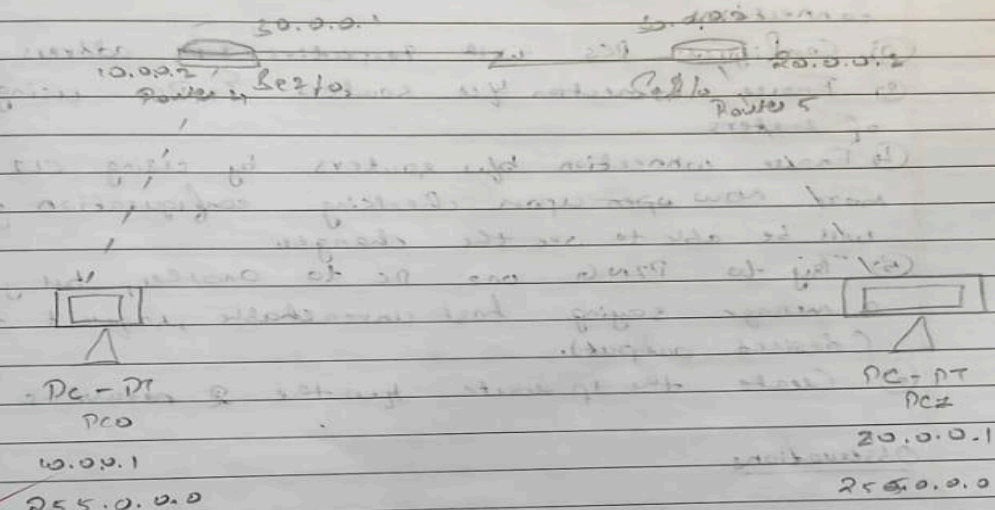
6/10/24

classmate
Date
Page 9

LAB-3

Aim: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

Topology.



Routers:-

R4: interface fa 0/0 connected PC0
interface Se 2/0 connected to R4
IP address :- 30.0.0.1

R5: interface fa 0/0 connected to PC1
interface Se 2/0 connected to R5
IP address: 30.0.0.2

End Devices

PC0 :- interface fa0 connected to R4
configured gateway 10.0.0.2
IP address :- 20.0.0.1

PC1 :- Interface fa0 connected to R5
configured gateway :- 20.0.0.2
IP address :- 20.0.0.1

Procedure :

(1) Place 2 Routers & 2 PCs connect the PC with Routers using crossover connecting with fastethernet & connect the routers to each ~~to~~ other using serial connections.

(2) Configure PCs with respective IP address & gateway.

(3) Ensure connection b/w routers & PCs using the CLI of routers

(4) Ensure connection b/w routers by using CLI once more Now ~~open~~ upon checking configuration you all will be able to see the changes.

(5) Try to PING one PC to another, but you get a message saying host unreachable, request timed out (desired output).

(6) Create the ip route b/w the 2 networks

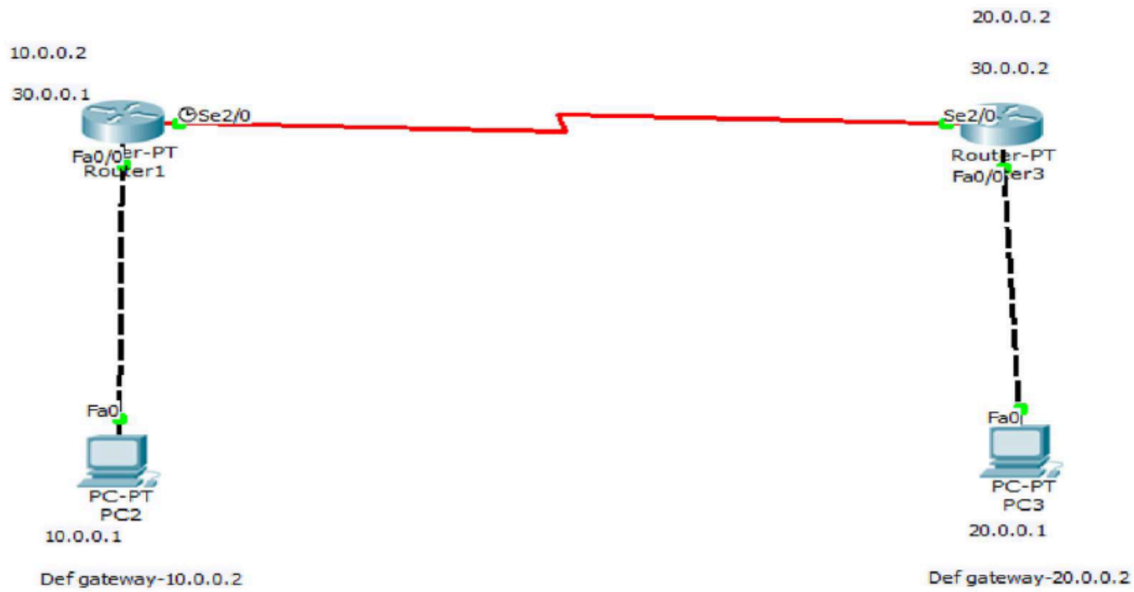
Observations

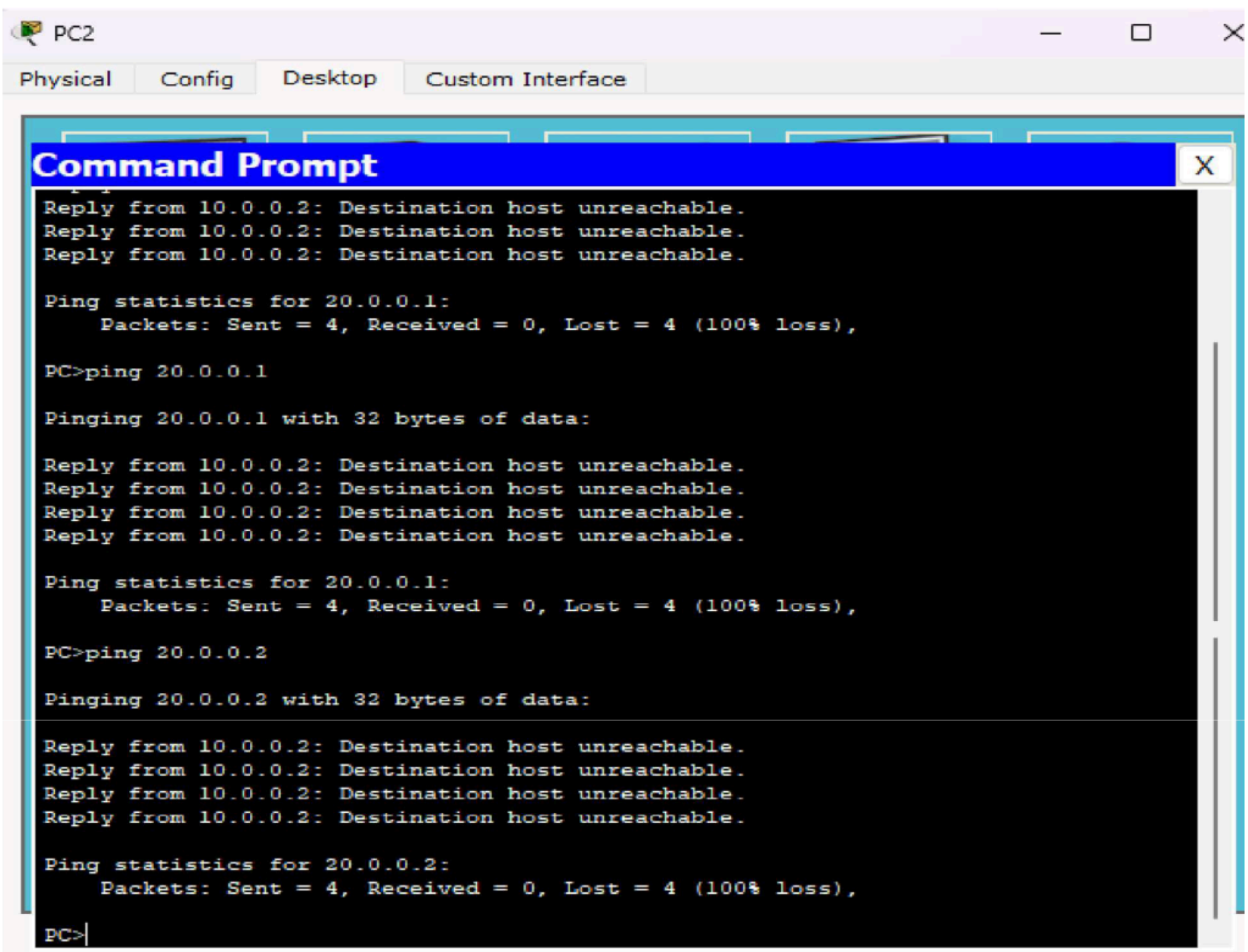
(1) There are 2 routers which are configured & ~~conectt~~ connected to the nodes.
However the nodes aren't able to communicate with each other.

C 10.0.0.0/8 is directly connected, FastEthernet0/0
S 20.0.0.0/8 [1/0] via 30.0.0.2
C 30.0.0.0/8 is directly connected, Serial2/0

(2) After establishing ip route b/w the 2 networks ping b/w the 2 PCs is working.

Screen Shots:






```
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```

Program 4

Aim: Configure default route, static route to the Router(Part 2).

Topology , Procedure and Observation:

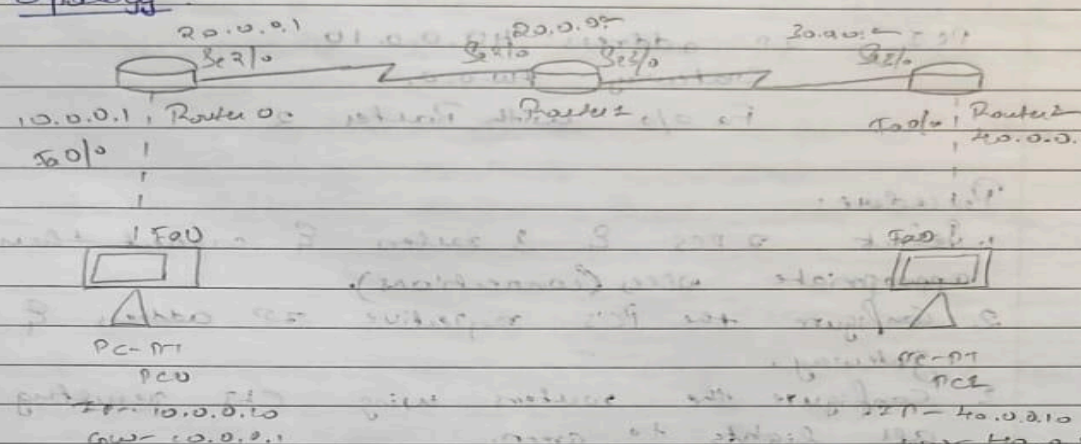
Date:

Lab – 4

Static Routing & Default Routing

Aim: Configure default route, static route to the Router.

Topology :



~~1. PC0 : connected to Router's interface fa 0/0 using a cross-over cable.~~
~~IP : 10.0.0.10~~
~~default gateway : 10.0.0.1~~
~~2. PC1 : connected to the Router's interface fa 0/0 using a cross over cable.~~
~~IP address : 20.0.0.10~~
~~default gateway : 20.0.0.1~~

Router 1 : IP address : 20.0.0.1
Fa 0/0 with PC0
Se 2/0 with Router1

Router -1 : IP address : 20.0.0.2
Se 2/0 with Router 1
Se 3/0 with Router 2

Router 2 : IP address 30.0.0.2
Se 3/0 with Router 2
Fa 0/0 with PC1

End devices:
PC0 : IP address: 10.0.0.10
Gateway : 10.0.0.1
Fa 0/0 with Router 0

PC1 : IP address: 40.0.0.10
Gateway : 40.0.0.1
Fa 0/0 with Router 2

Procedure:
1. Select 2 PCs & 3 routers & connect them using appropriate wires (connections).
2. Configure the PC's respective IP address & gateway.
3. Configure the routers using CLI resulting in all lights to green.
4. Obtain IP route from PC0 to the router 0 & router 1. Likewise with PC1 to router 1 & router 2.
5. Obtain default routers ~~fro~~ for Router 0 & Router 2.
6. Ping PC1 from PC0 & PC0 from PC1.

Observation:
* All connections (fast ethernet & serial) have turned green.
* IP route before set up:
20.0.0.0/8 is directly connected, serial 2/0
30.0.0.0/8 is directly connected, serial 3/0

show ip route after setup:
10.0.0.0/8 [1/0] via 20.0.0.1

20.0.0.0/8 is directly connected, serial 2/0
30.0.0.0/8 is directly connected, serial 3/0
40.0.0.0/8 ~~is directly co~~ via 20.0.0.2

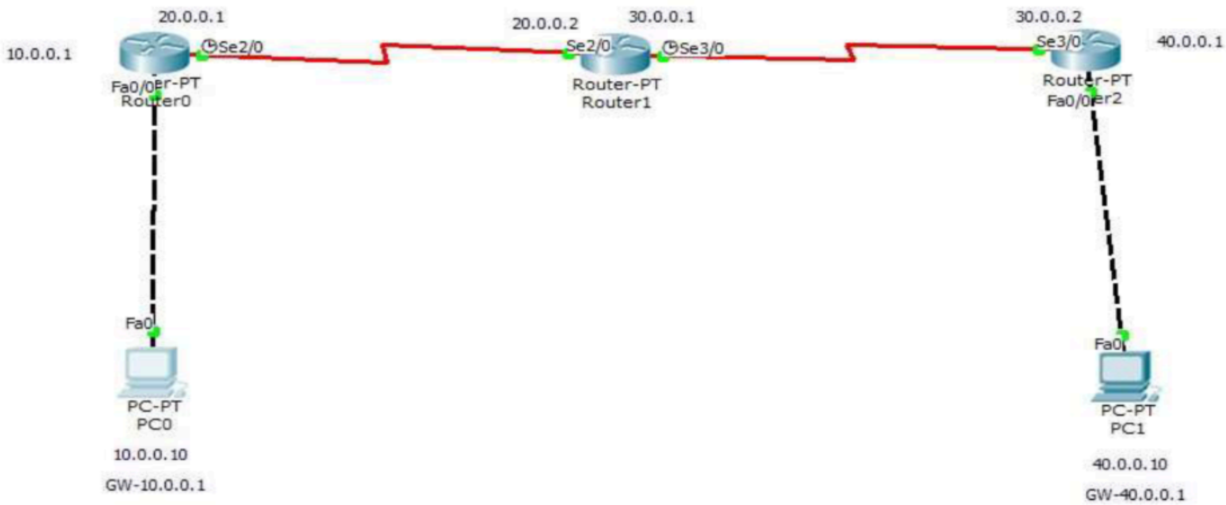
* Ping from one PC to another is successful,
* So the middle router (router 1) is set-up with 2 next-hops.
* Default Router :- to transfer when no other route is available.
* Static Route : define Route with assigned destination.

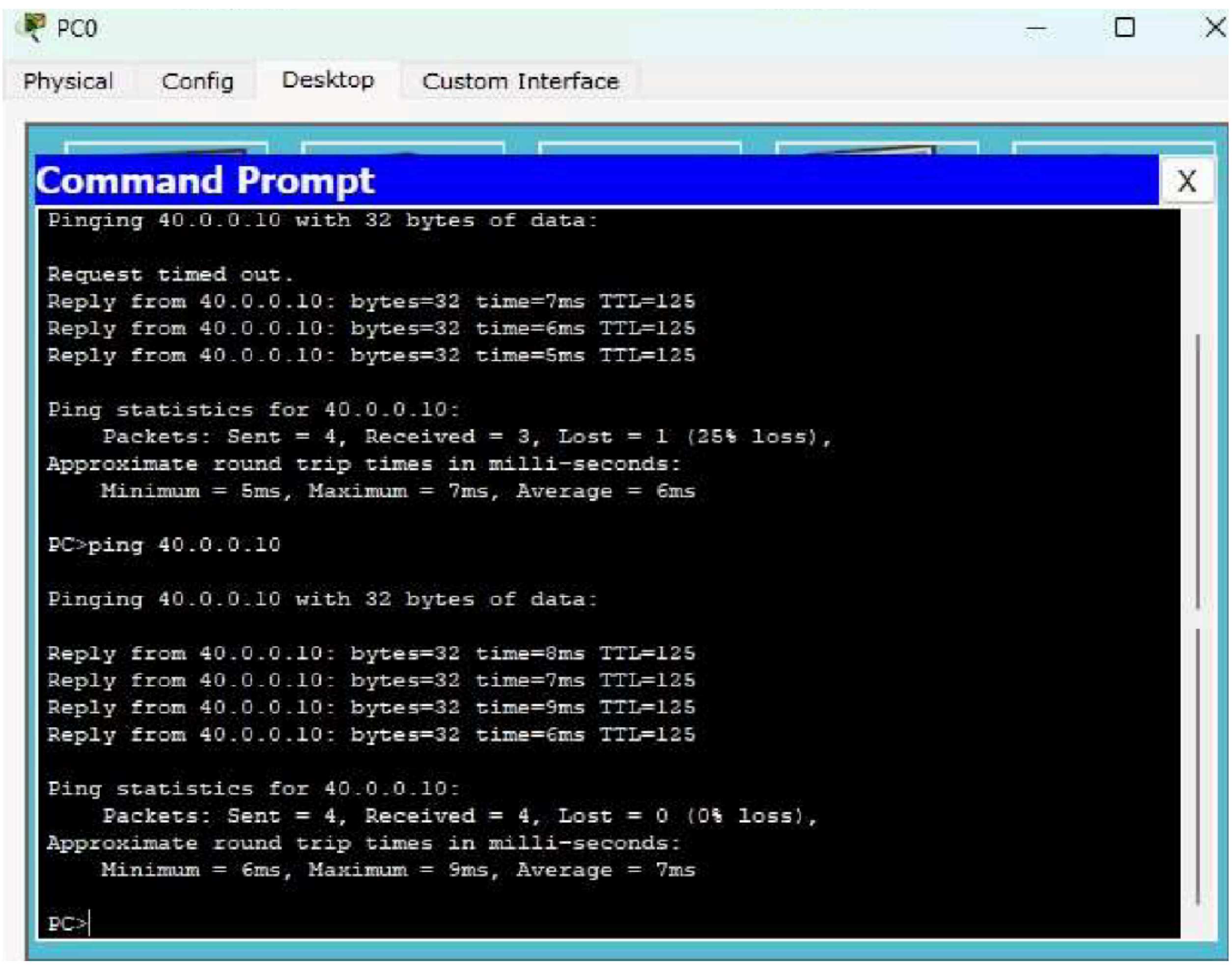
* IP route for Router 0
10.0.0.0/8 is directly connected, FastEthernet 0/0
20.0.0.0/8 is directly connected, serial 2/0
0.0.0.0/0 [1/0] via 20.0.0.2

* IP route for Router1
C 20.0.0.0/8 is directly connected, serial 3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
C 0.0.0.0/0 [1/0] via 30.0.0.1

20/11/24

Screen Shots:






```
Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms 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 = 5ms, Maximum = 7ms, Average = 6ms

PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 9ms, Average = 7ms

PC>
```

Program 5

Aim: Configure DHCP within a LAN and outside LAN.

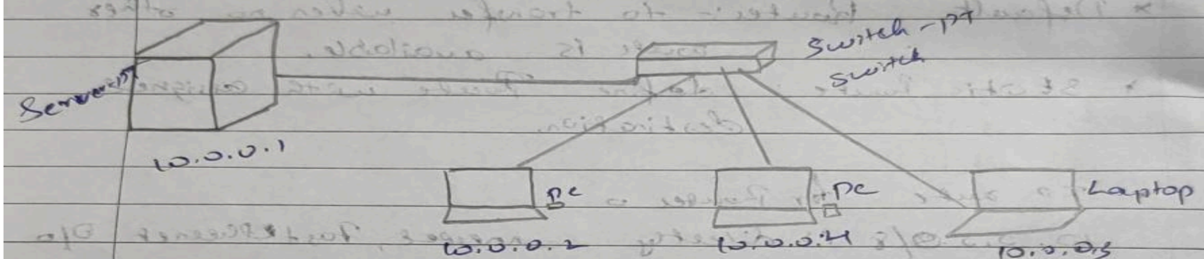
Topology , Procedure and Observation:

Lab-5

DHCP

Aim: Design a DHCP within LAN & outside LAN.

Topology:-



Procedure:-

1) Place a switch, a server, two PCs & one laptop.
2) Connect using automatically choose connection type.
3) Click Server - Desktop → IP configuration - set IP address as 10.0.0.1 & default gateway as 10.0.0.0
4) Click on PC → Desktop → IP configuration → select DHCP → : automatic / dynamic IP address is assigned.
5) Click server → config → DHCP →
   Pool Name: Swith one
   Default Gateway: 10.0.0.0
   State IP address: 10.0.0.3
   Max user: 100

Observation :-

① The IP address were allocated dynamically to the end devices.

② Ping from one end device to another device was successful.

-> 10.0.0.2

Ping 10.0.0.3

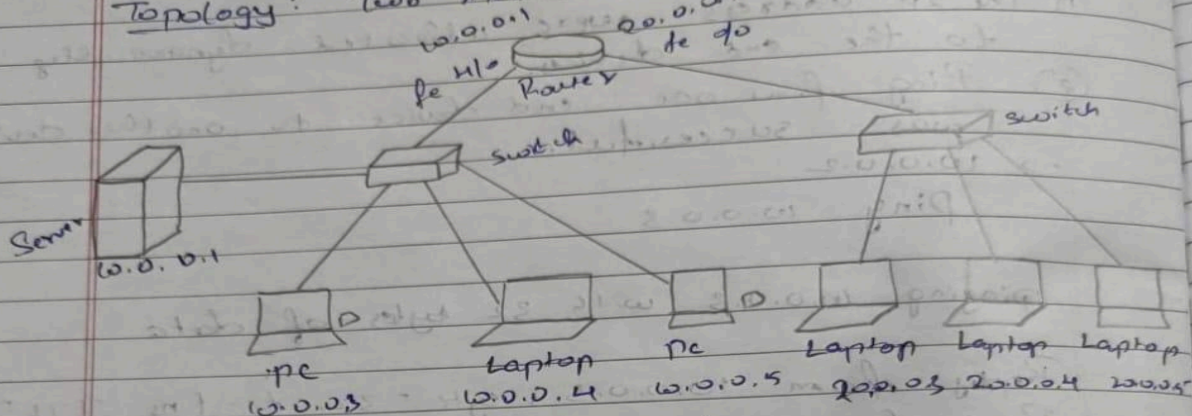
Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : byte = 32 t = 1 ms TTL = 128
Reply from 10.0.0.3 : byte = 32 t = 0 ms TTL = 128
Reply from 10.0.0.3 : byte = 32 t = 0 ms TTL = 128
Reply from 10.0.0.3 : byte = 32 t = 7 ms TTL = 128

Ping statistics for 10.0.0.3
Packets : Sent = 4, Received = 4, lost = 0 (0% loss)
Approx ~~trip~~ round trip times in milliseconds :
min :- 0 ms Max = 7 ms avg = 2 ms

— This eliminates the need for network administrator to manually assign IP addresses to each device.

Topology: Two networks



Procedure:-

(1) Retain the first network.
(2) Add another network with a switch & 3 end devices.
(3) Add a router to connect the 2 networks.
(4) Edit the IP configuration of server.
 - change IP of server to 10.0.0.2
(5) Edit switch one Root:
 Default Gateway : 10.0.0.1
 state IP: 10.0.0.3
(6) Add another Root: Switch Two!
 Default Gateway: 20.0.0.1
 state IP : 20.0.0.3
(7) Configure the router:
 Router > enable
 Router # config terminal
 Route (config) # interface fastethernet 4/0
 Route (config-if) # ip address 10.0.0.1 255.0.0.0
 Route (config-if) # ip helper-address 10.0.0.2

17

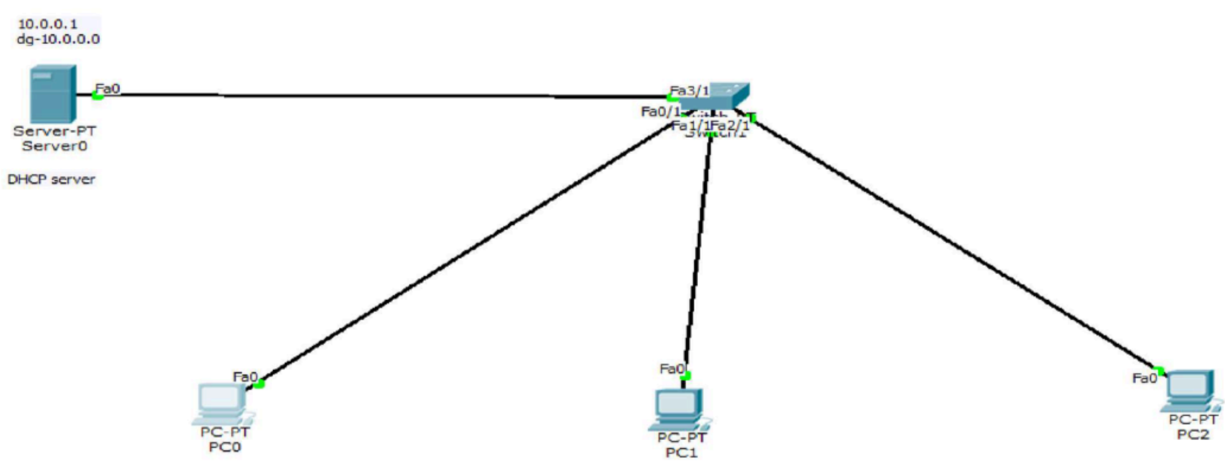
(8) Similarly configure the router network:
Router(config)# interface fastethernet 4/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# ip helper-address 10.0.0.2

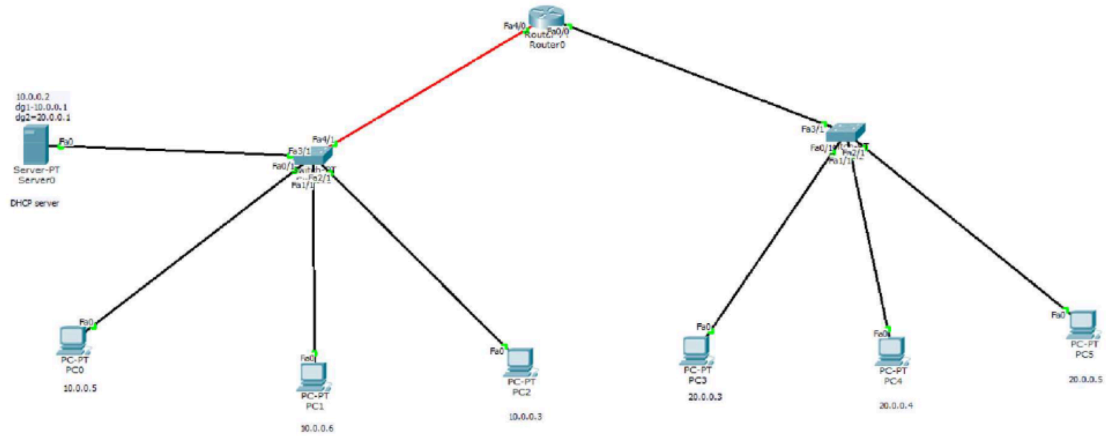
(9) Now again reconfigure the end devices of both networks using DHCP.

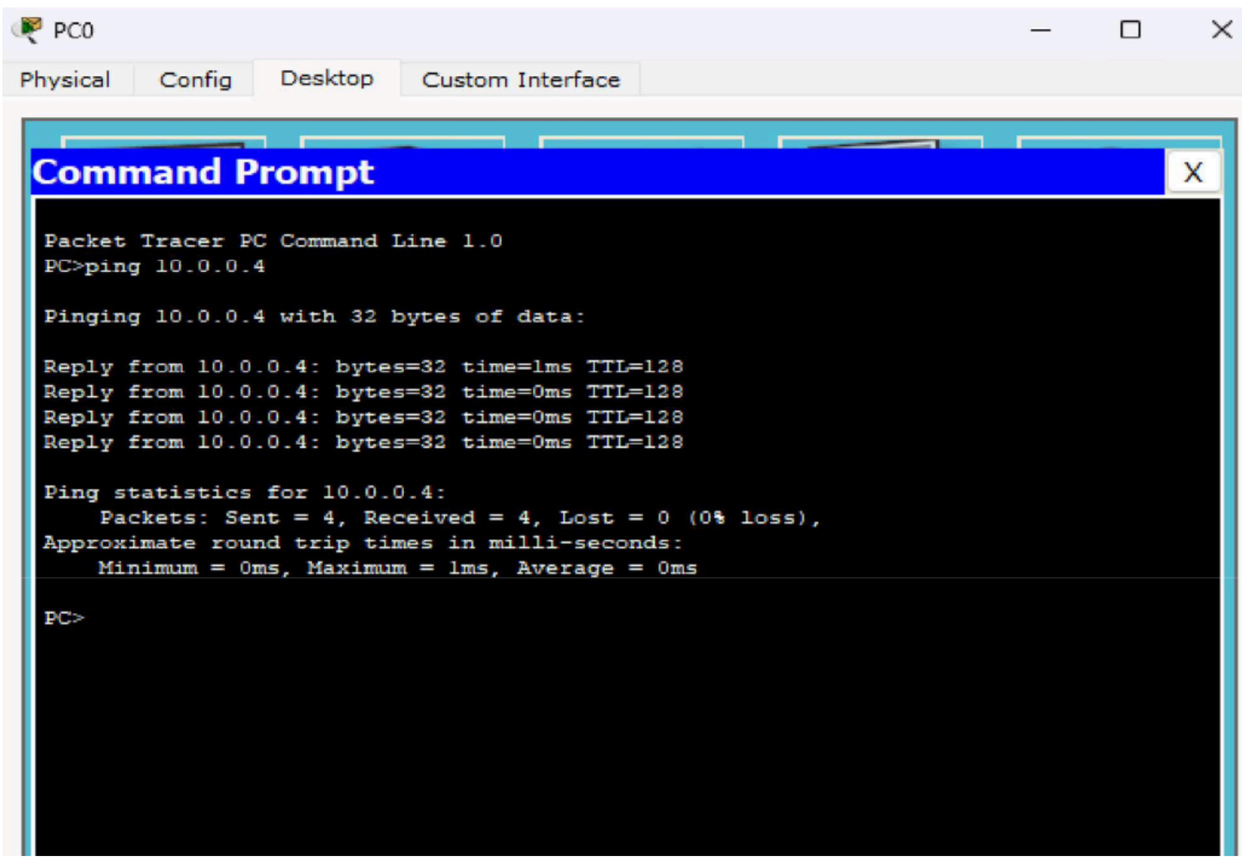
Observations:

(1) IP addresses are allocated dynamically.
(2) Data was ~~not~~ sent successfully ~~among~~ among PCs when pinged.
(3) This eliminates the need of network administrator to manually assign IP addresses to each device.

Screen Shots:




Fa4/0
Fa0/0
Router0
10.0.0.2
dg1-10.0.0.1
dg2=20.0.0.1
Fa0
Server-PT
Server0
DHCP server
Fa4/1
Fa3/1
Fa0/1
Fa2/1
Fa1/1
Fa3/1
Fa0/1
Fa1/1
Fa0
PC-PT
PC0
10.0.0.5
Fa0
PC-PT
PC1
Fa0
PC-PT
PC2
10.0.0.3
Fa0
PC-PT
PC3
20.0.0.3
Fa0
PC-PT
PC4
20.0.0.4
Fa0
PC-PT
PC5
20.0.0.5


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
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=0ms 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>

Program 6

Aim: Configure RIP routing Protocol in Routers .

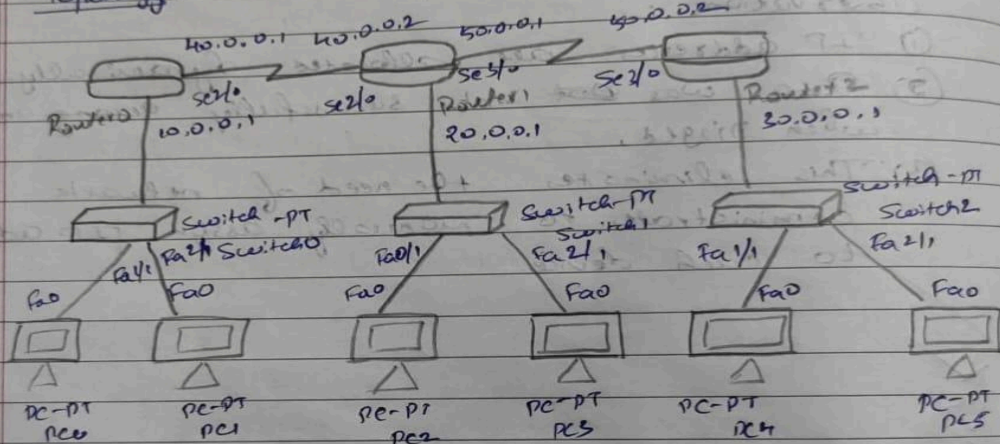
Topology , Procedure and Observation:

Lab – 6

Routing Information Protocol in Routers

Aim:- To obtain connection in the network of 3 routers, 3 switches & 6 PC's using RIP.

Topology:-



Routers:

Router 0: IP address : 10.0.0.1
Fa0/0 : with switches (10.0.0.1)
~~fa~~ se2/0 : with router 1 (40.0.0.1)

Router 1: IP address : 20.0.0.1
Fa ~~1~~/0 : with switch (20.0.0.1)
Se 2/0 : with router 0 (40.0.0.2)
Se 3/0 : with router 2 (50.0.0.1)

Route 2: IP address : 30.0.0.1
Fa 0/0 : with switch (30.0.0.1)
Se 3/0 : with router 1 (~~2~~ 50.0.0.2)

Switches:

Switch 0 : Fa 0/1 : Router 0
Fa 1/1 : PC0
Fa 2/1 : PC1

Switch 1 : Fa 1/1 : Router 1
Fa 0/1 : PC2
Fa 2/1 : PC3

Switch : Fa 0/1 : Router 2
Fa 1/1 : PC4
Fa 2/1 : PC5

~~PCs~~ PC's

PC0 : ip address : 10.0.0.2
Fa 0 : switch 0

PC1 : ip address : 10.0.0.3
Fa0 : switch 0

PC2 : ip address : 20.0.0.2
Fa0 : switch 1

PC3 : ip address : ~~30~~ 20.0.0.3
Fa0 : switch 1

PC4 : ip address : 30.0.0.2
Fa0 : switch 2

PC5 : ip address : 30.0.0.3
Fa0 : switch 2

Procedure:

1. Connect 3 Routers, 3 switches & 6 PCs using appropriate connections.
2. Configure end devices i.e PC's ip addresses & gateway.
3. Configure the interface of routers untill all connections turn green.
4. For each router, configure the rip routers & add connected networks.

5. Check with show ip route command & ping command.
6. 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

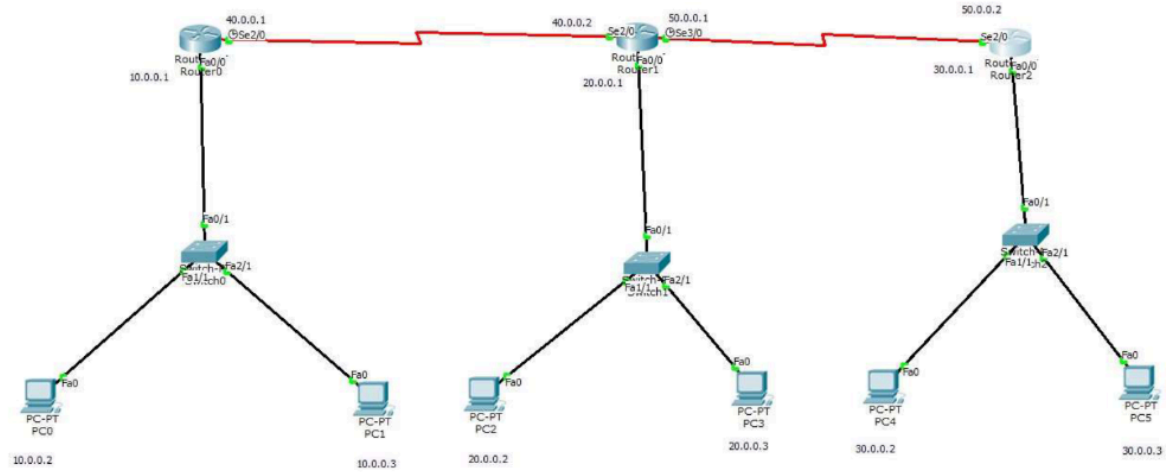
In Router 1
(config) # router rip
(config-router) # network 40.0.0.0
(config-router) # network 50.0.0.0
(config-router) # network 70.0.0.0

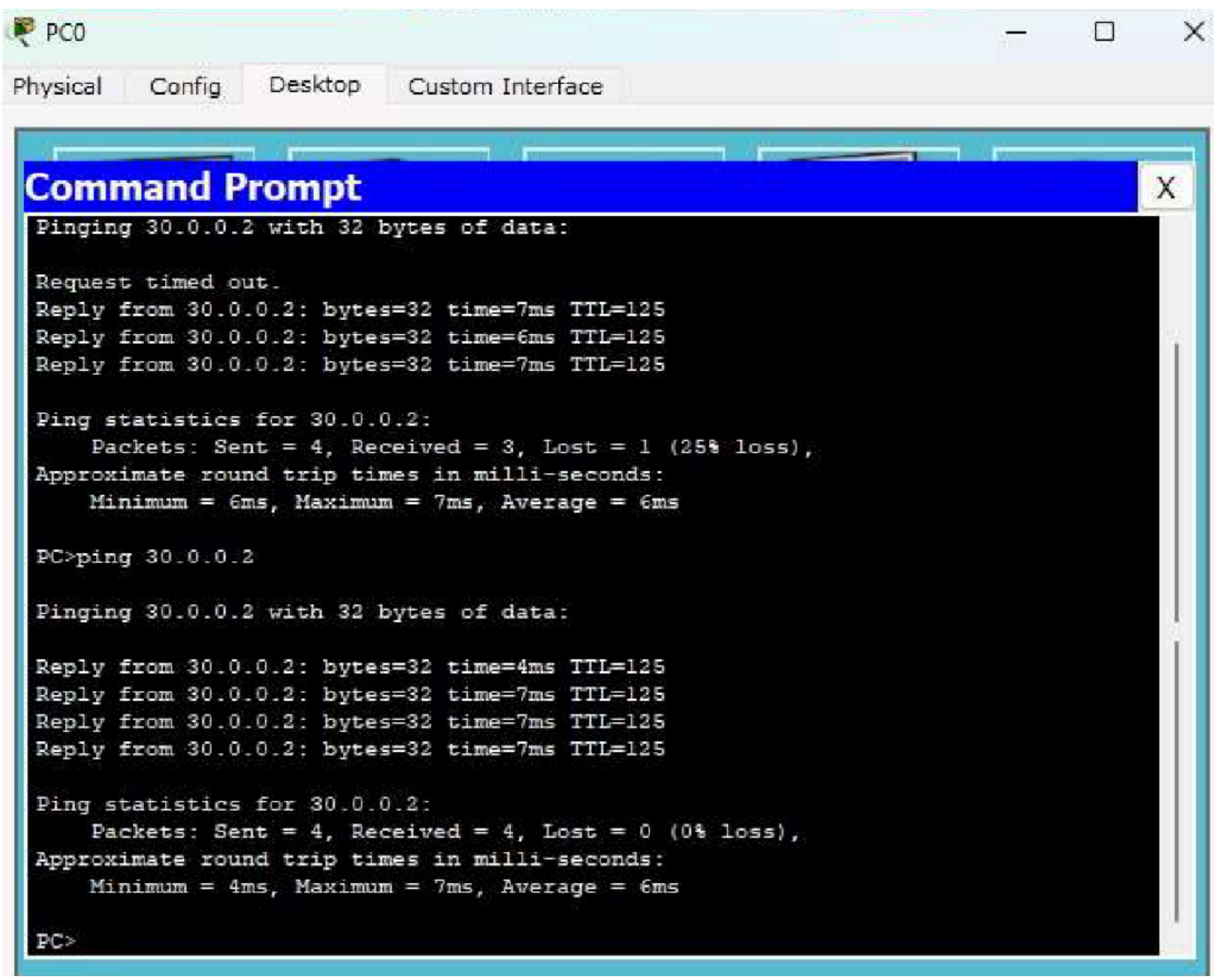
In Router 2
(config) router rip
(config-router) # network 50.0.0.0
(config-router) # network 30.0.0.0

Observation
→ Before doing RIP when ping it was timed out.
→ After apply RIP when 30.0.0.3 was pinged from 10.0.0.1 it 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

Screen Shots:





Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:

Demonstrate the TTL / Life of a Packet

Aim :- Demonstrate TTL / Life of a packet.

TTL means times 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 PC4

→ Auto capture the event list then, observe the TTL of each router in PDU information.

Observation

when the packet passes Router 0.

inbound TTL = ~~25ms~~ 255ms

outbound TTL = ~~35ms~~ 254ms

When the packet passes from Router 0 to Routers

inbound TTL = 254ms

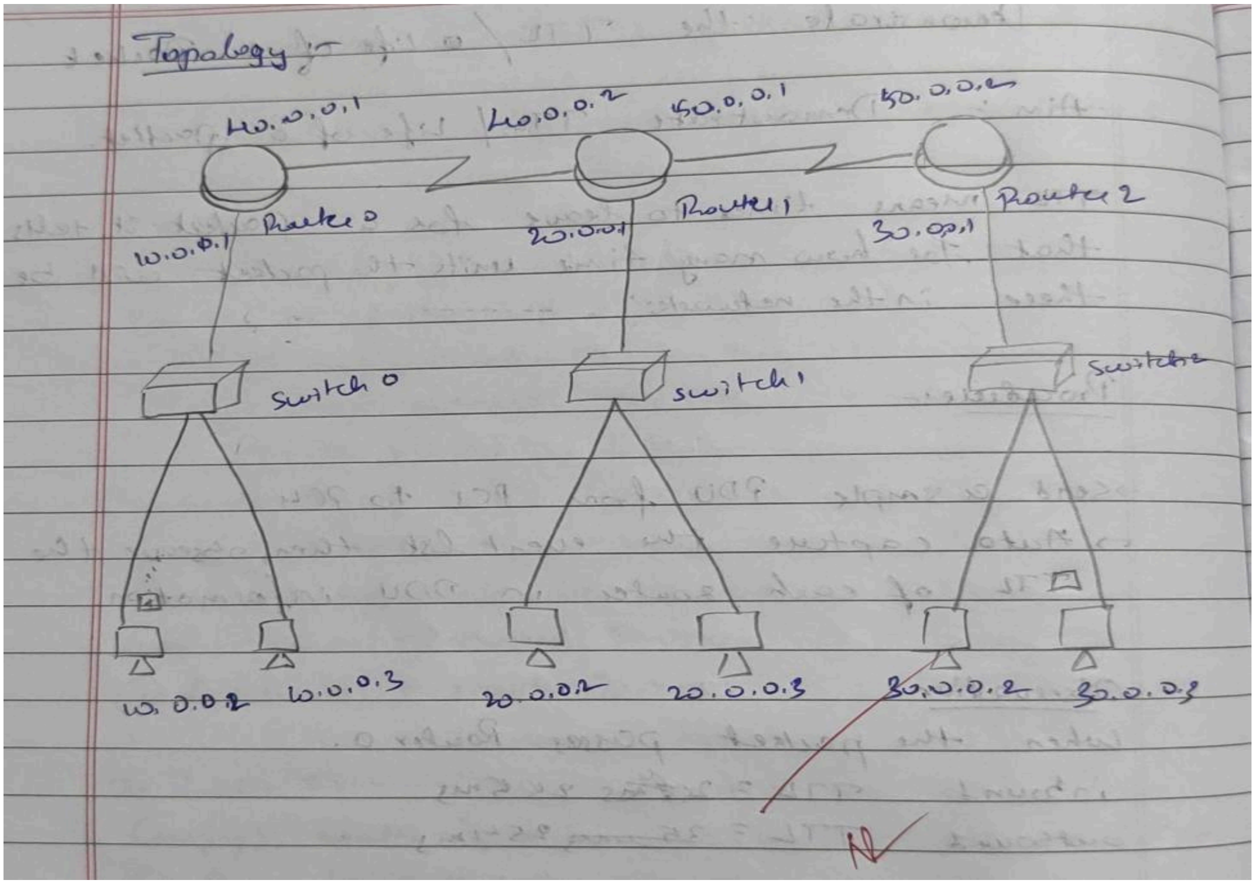
outbound TTL = 253ms

when the packet passes across Router

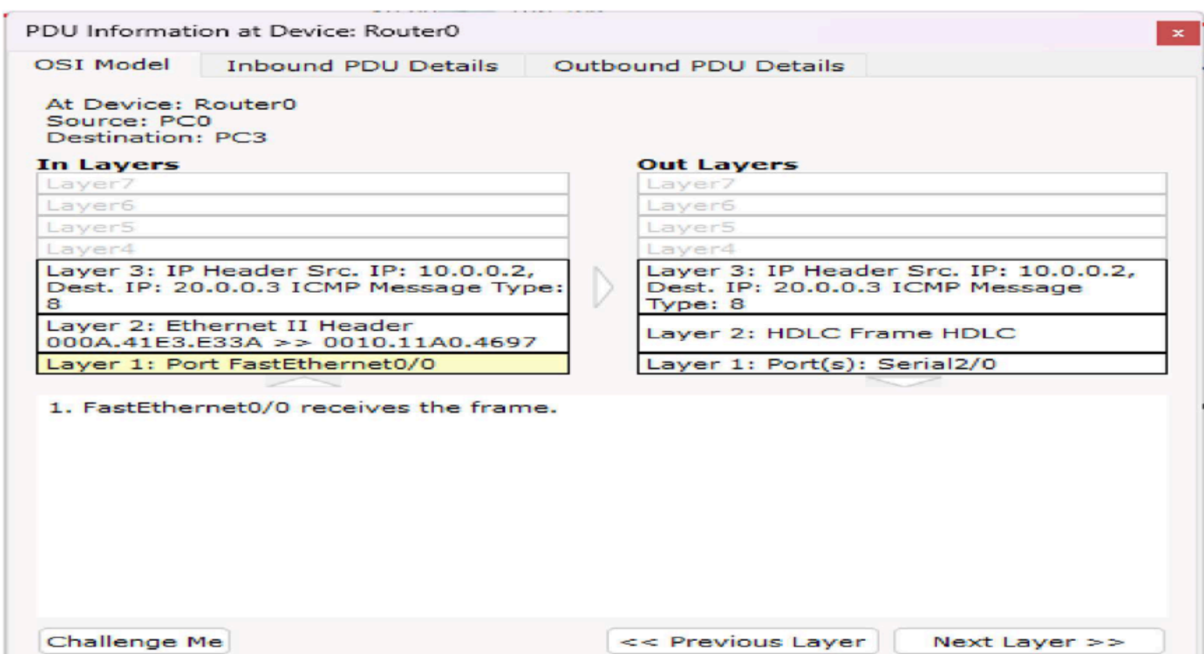
inbound TTL = 253 ms

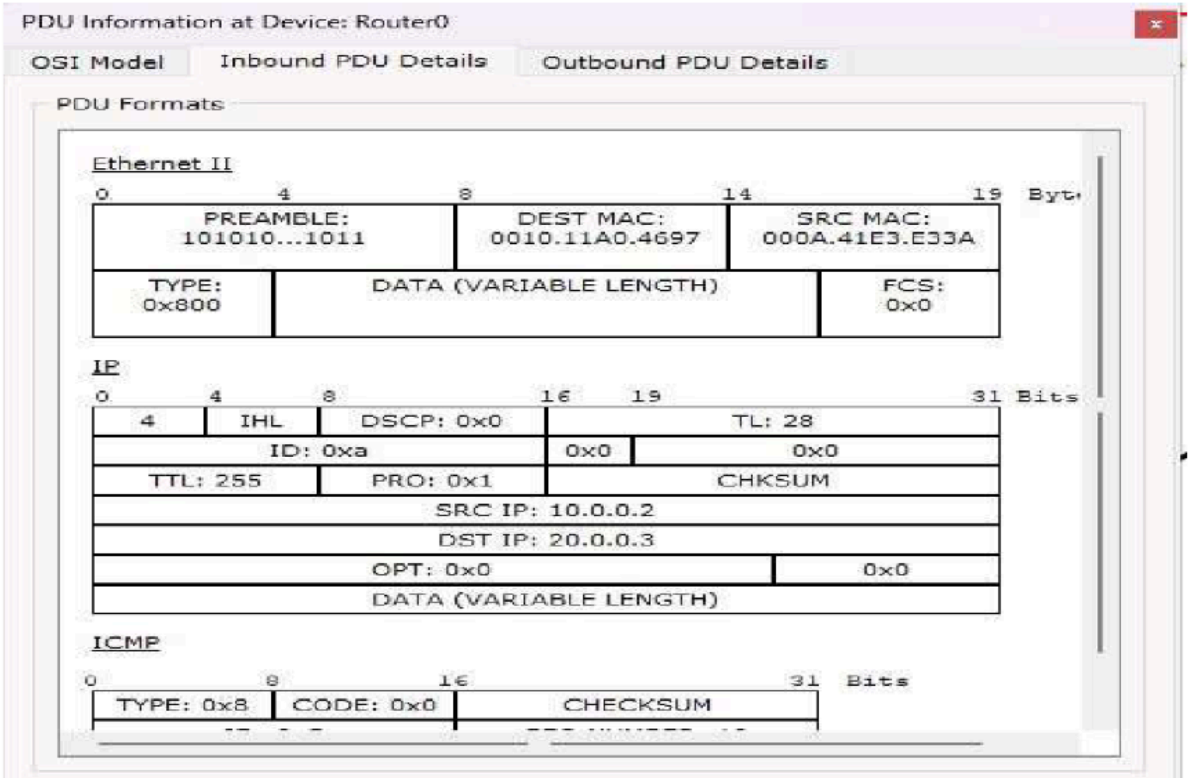
outbound TTL = 252 ms

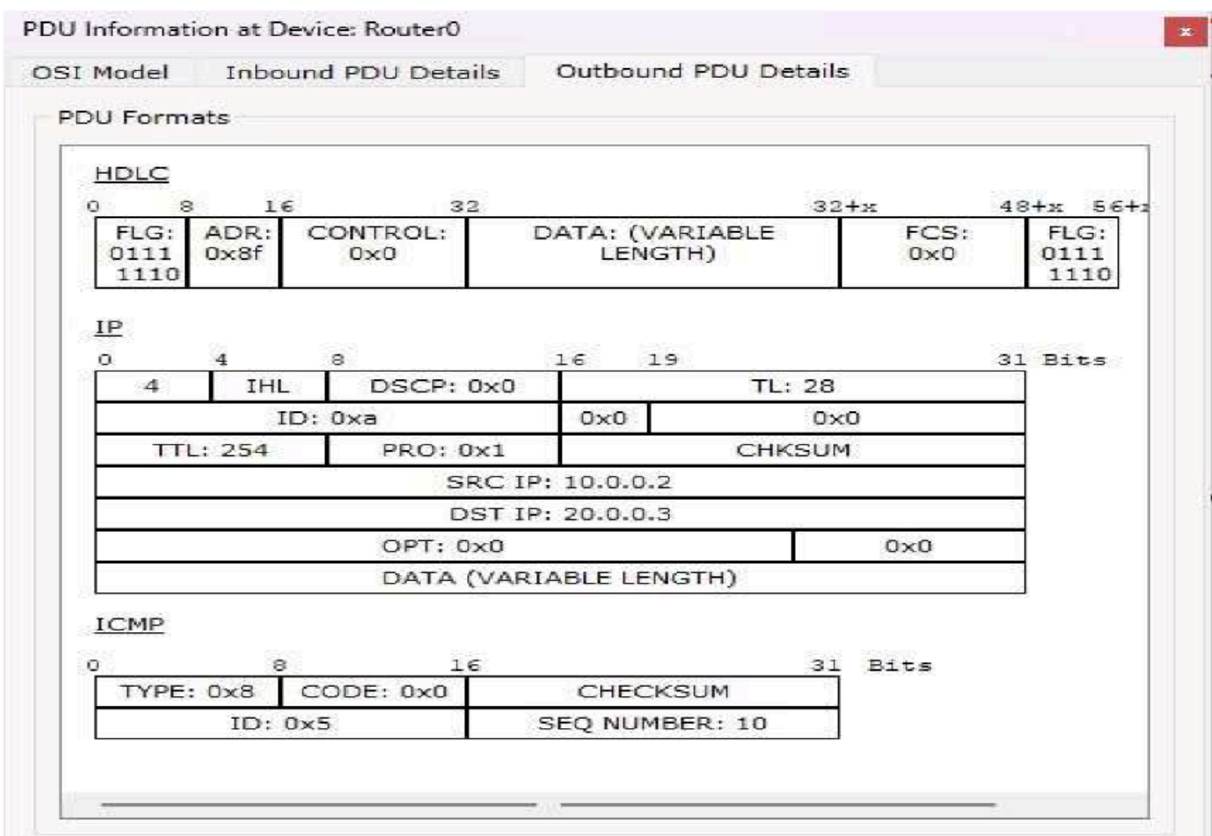
Hence we can conclude there will be decrement in TTL for 1ms when it passes ~~Ro~~ across Router.



Screen Shots:




PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 10

Program 8

Aim: Configure OSPF routing protocol .

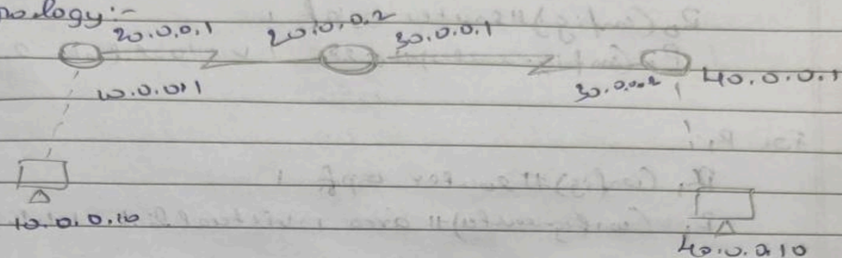
Topology , Procedure and Observation:

23

Configure OSPF routing protocol.

Aim: To demonstrate OSPF routing protocol with several routers & networks.

Topology:-



Procedure:-

→ configure all devices as given above in the ~~company~~ topology.

→ Enter the following commands for all 3 routers.

→ Router (config-if) # .

→ To all points with clock symbol
clock route 64000

→ Enter OSPF commands

Router (config) # router OSPF

\# router -id 1.1.1.1

Router (config-router) # network n.0.0.0 0.255.255.255

Router (config-router)

do the above for all 3 routers.

→ Enter loopback commands.

Router (config) # interface loopback ?

Router (config-if) # ip add 192.16.1.252. 255.255.0.0

\# no shut

Do similarly for all the 3 routers.

→ Virtual link set up between router 0 & router 1.

For R_0

R_0(config)#router ospf 1
R_1(config-router)#area 1 virtual link 2.2.2.2.

For R_1

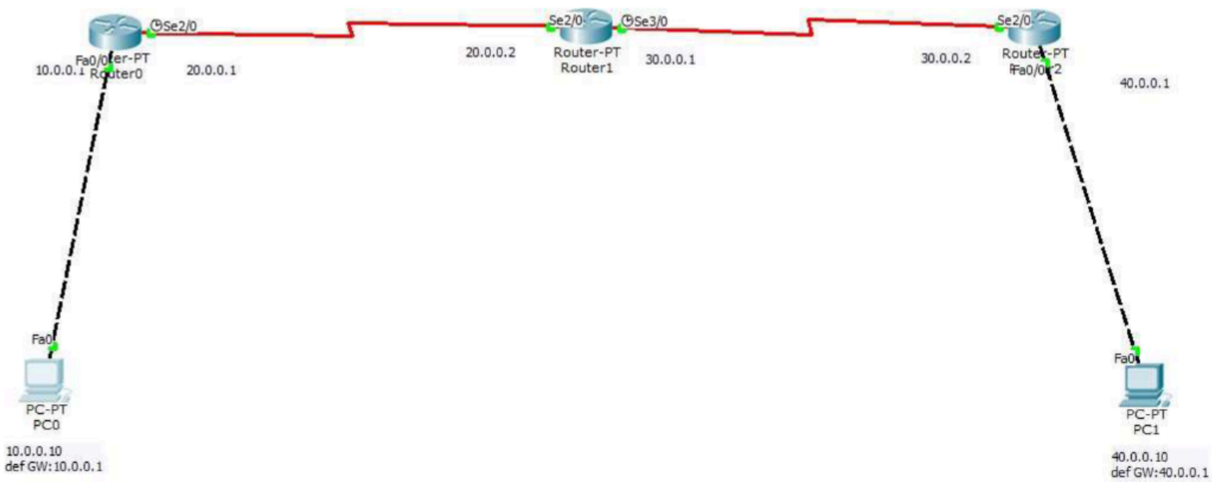
R_1(config)#router ospf 1
R_1(config-router)# area 1 virtual link 1.1.1.1.

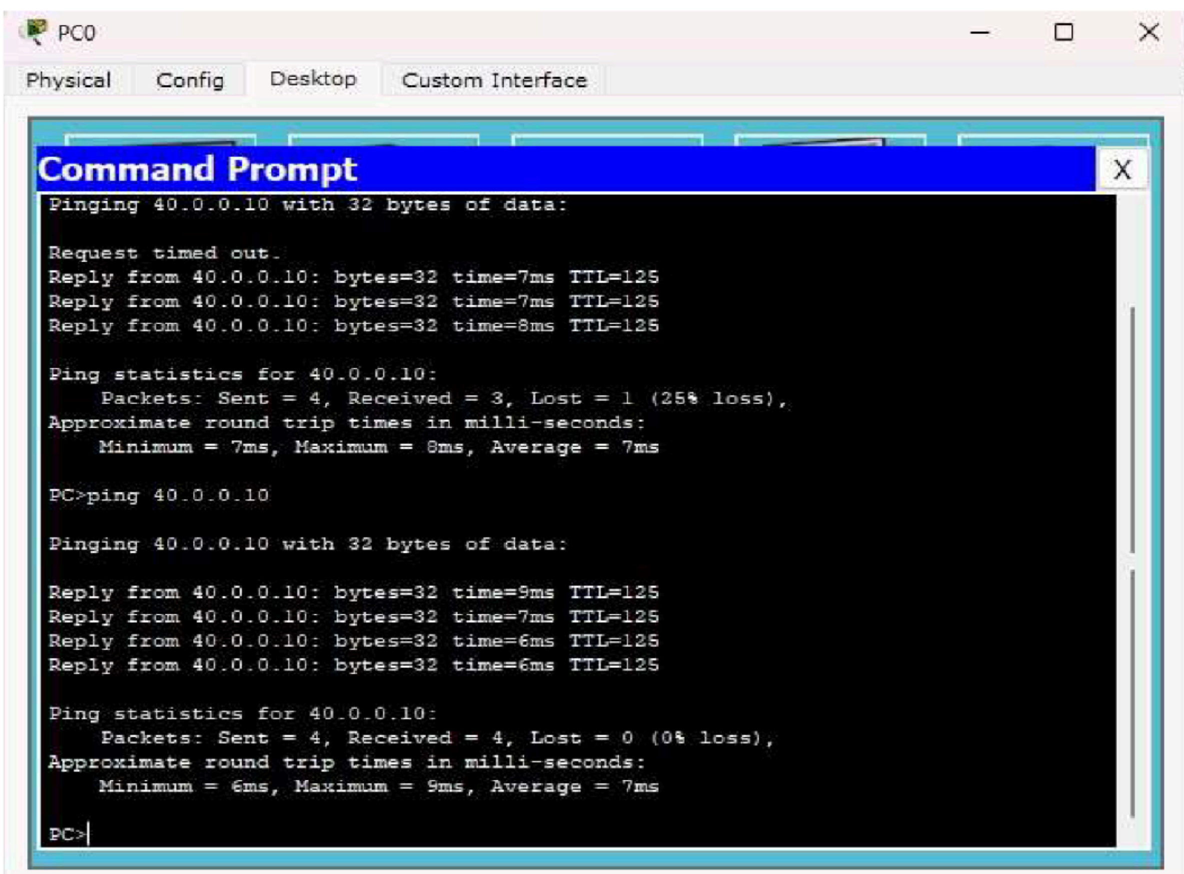
Observation:-

Now all the 3 router know each other.
on showing show ip route in R_3

O	IA	10.0.0.0/8	via	30.0.0.1	Serial 2/0
O	IA	20.0.0.0/8	via	30.0.0.1	Serial 2/0
C		30.0.0.0/8	is directly connected serial 2/0		
C		30.0.0.0/3	is directly connected serial 2/0		
C		40.0.0.0/8	is directly connected fast ethernet.		
C		172.16.0.0/16	is directly connected, loopback 0		

Screen Shots:





Program 9

Aim: Configure Web Server, DNS within a LAN.

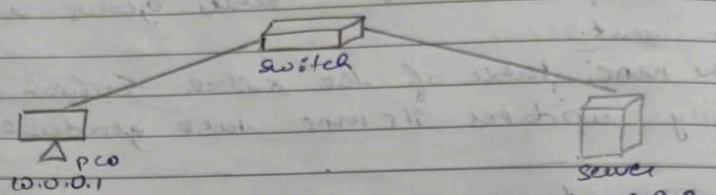
Topology , Procedure and Observation:

18/12/2024 Lab No 11

DNS

Aim:- Configure Web server, DNS within a LAN

Topology:-



Procedure :-

① Set up the LAN as per the topology mentioned above & configure the devices.

② Go to server → Services → DNS :
Name : bmsce [Domain name]
Address: 10.0.0.2
Add the mapping of domain name to 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

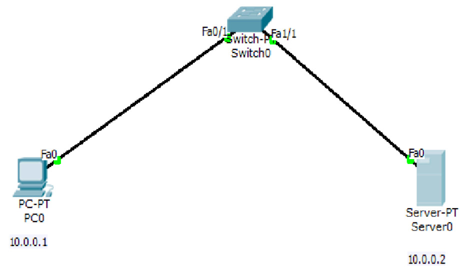
Observations:-

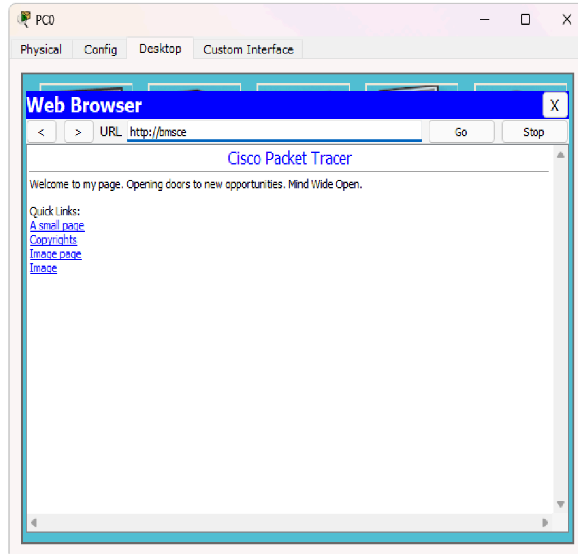
① The webpages hosted by the server were visible on the browser.

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

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.

Screen Shots:





Program 10

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

Topology , Procedure and Observation:

Date
Page

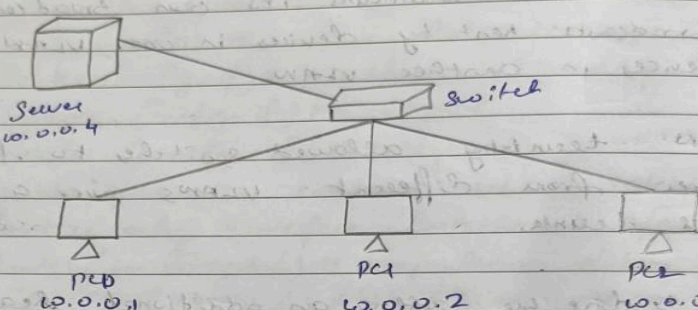
18/12/2024

Lab No 10

ARP

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

Topology:-



Procedure:

1. Create the topology as shown above.
2. Configure the PC's & the server
3. Click on Inspect Mode (Q), then click on the end devices & 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.

29

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

(4) The server and the PC update their ARP tables matching 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

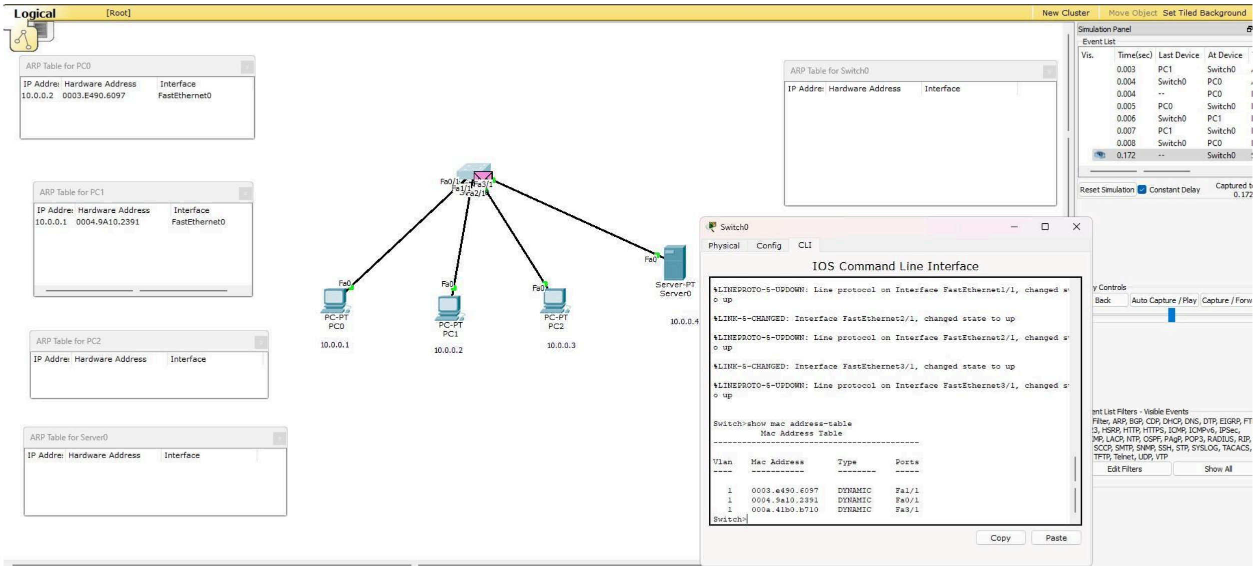
ARP Table for 10.0.0.4 :

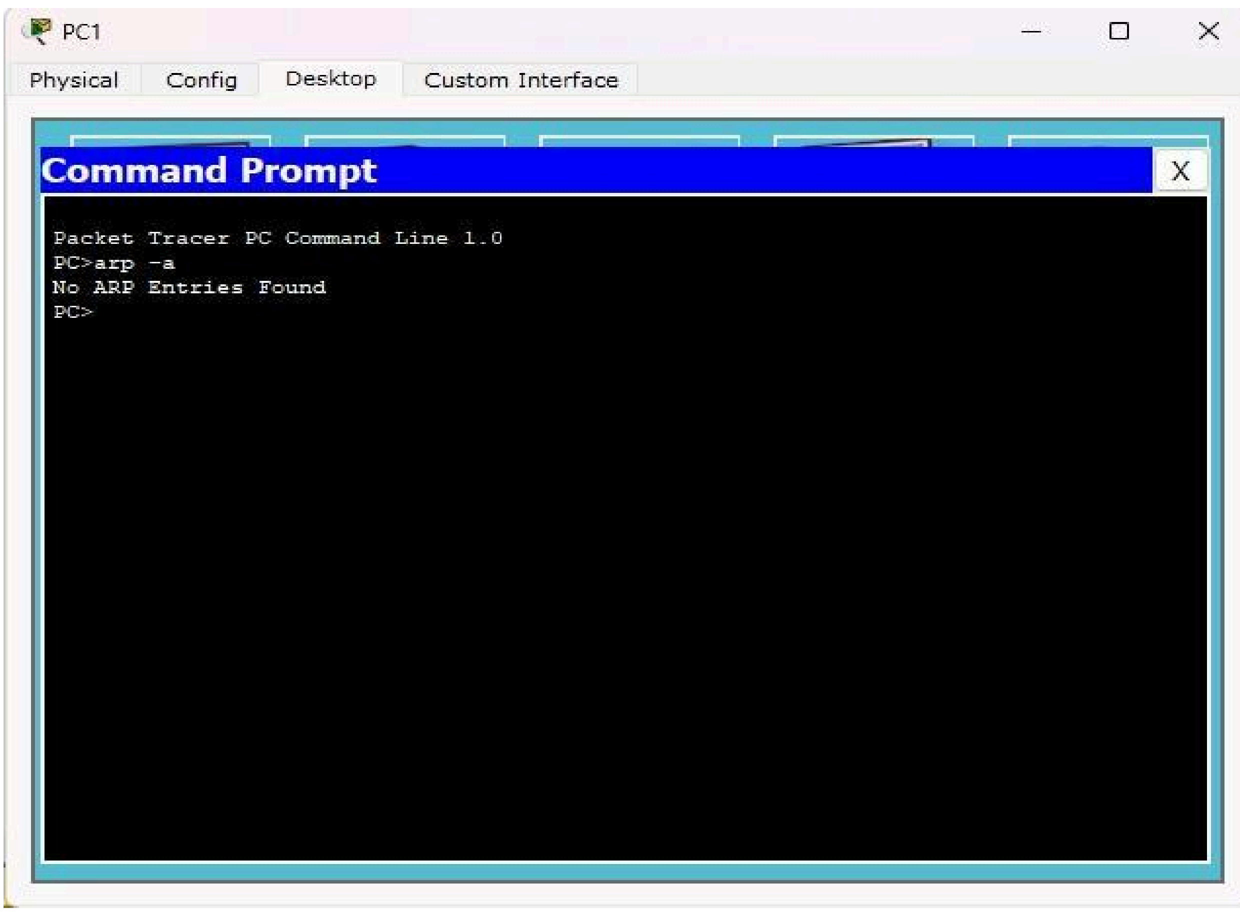
IP address	Hardware address	Interface
10.0.0.3	0001.C7A8.47E5	FastEthernet0

(7) Similarly other ARP ~~ca~~ tables are updated.

3/1/25

Screen Shots:





Program 11

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

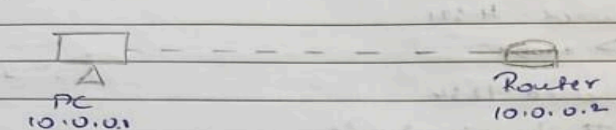
Topology , Procedure and Observation:

18/12/24 Lab No 12 31

TELNET

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

Topology:-



Procedure:

1. Create the topology as given above & configure the devices
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 described on line 194, until 'password' is set.

R1 (config-line)# password 4321 → user access verification password

R1 (config-line)# exit

R1 (config)# exit

R1 # wr

Building configuration...

[OK]

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

③ In PC : Command Prompt:
- First try Pinging to see if 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, FastEthernet 0/0

R1 #

Observations:-

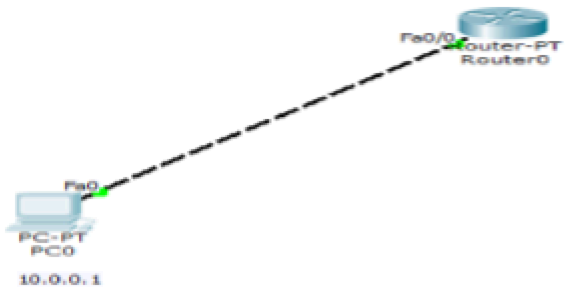
① The admin in PC is able to run commands as run in router CLI & see the results from PC.

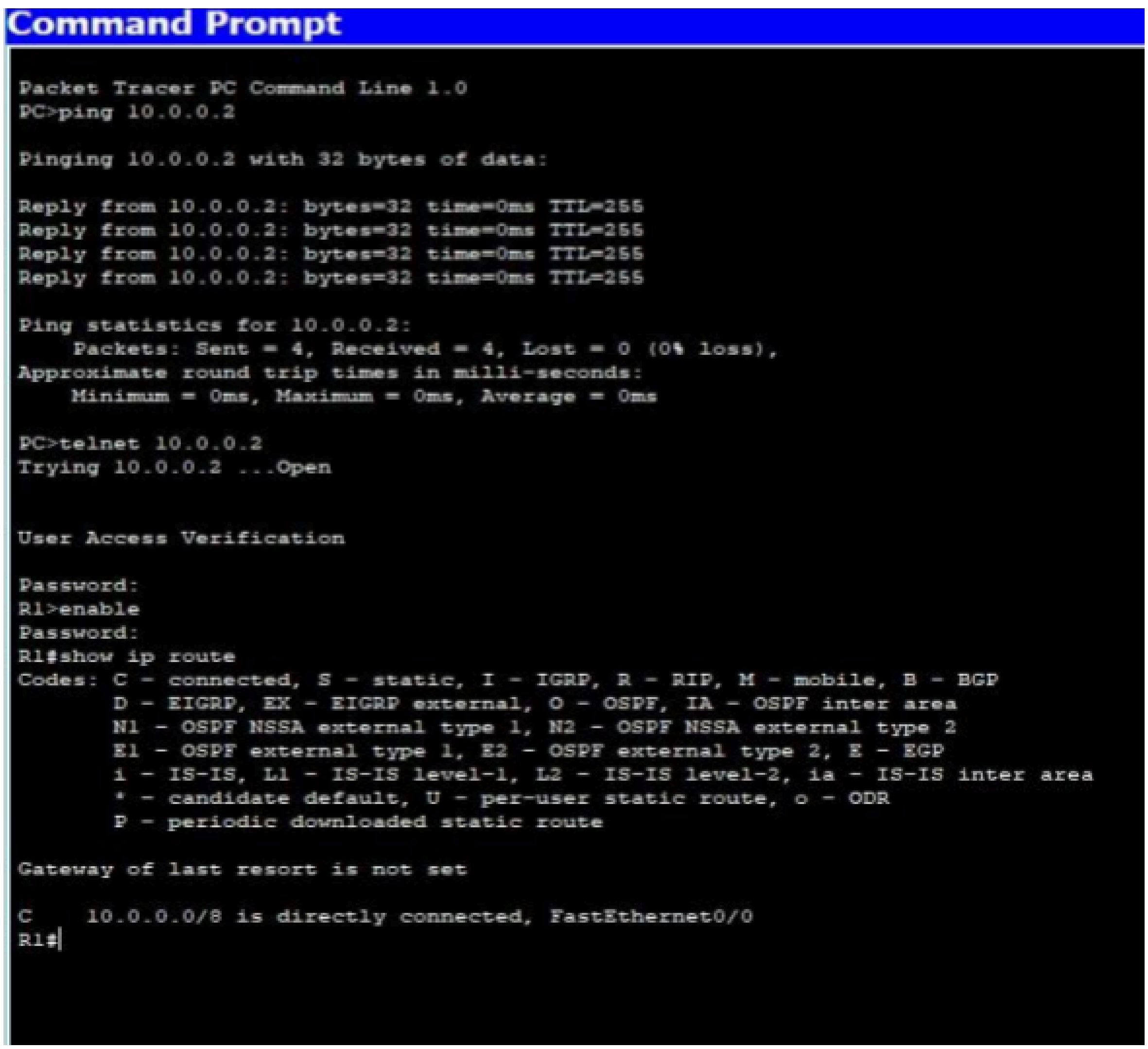
② Telnet allows users to establish a remote session with another device like router, over a TCP/IP network.

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

3/7/25

Screen Shots:





Program 12

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

Topology , Procedure and Observation:

Date
Page

11/12/24

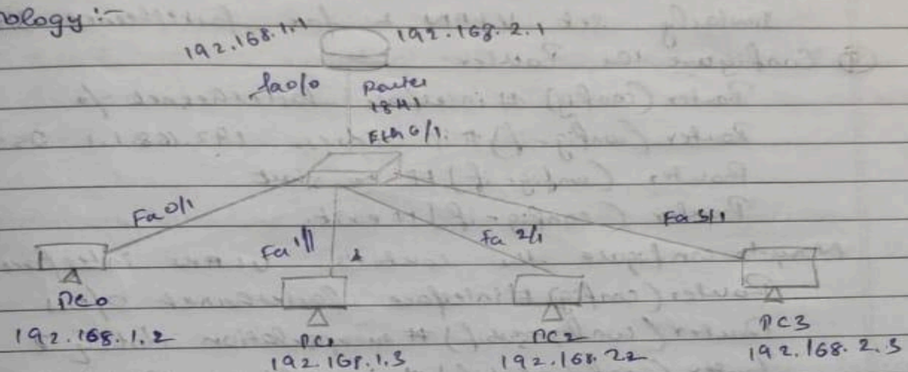
Lab No 9

VLAN

28

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

Topology :-



Procedure :-

1. Place a 1841 Router, a switch & PC's
2. Connect the four PCs to the switch via fastethernet.
3. Since only 4 fastethernet ports are available in the switch, we have to add an ethernet port.
4. Switch off the power button of 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 &
   - Select VLAN Database
   - Give VLAN number say 2
   - Give VLAN name say 'cseise'
   - Add it to the Database
6. Select the switch :
   - Go to config
   - Go to Ethernet 6/1 i.e connected to Router.
   - Make it the trunk.

(7) Config the PCs as shown in the topology.
(8) Select Switch:
- Go to config
- Go to FastEthernet 2/1
- set VLAN number as 2 i.e 'CSE'
- similarly set VLAN 2 for fastethernet 3/1 interface.
(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 192.168.2.1 255.255.255.0

Router (config-subif) # no shut
Router (config-subif) # exit
(10) Ping devices within the same VLAN & to devices of different ~~VA~~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 ←

27

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.

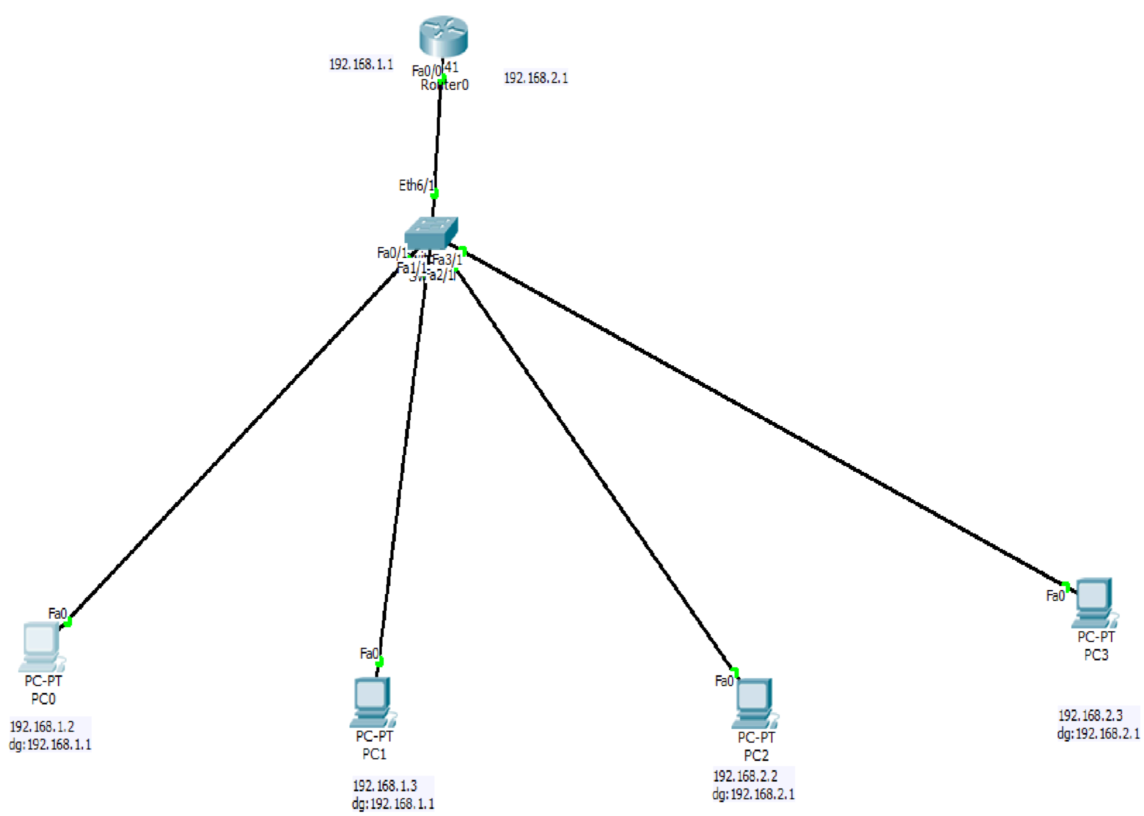
— Broadcasts sent by devices in one VLAN do not reach devices in another VLAN.

⑤ VLAN trunking allowed switches to forward frames from different VLAN's over a single links called trunk.

— This is done by adding an additional header information called tag to the ethernet frame- VLAN tagging.

3/1/25

Screen Shots:



Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

Program 13

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

Topology , Procedure and Observation:

Date
Page 33

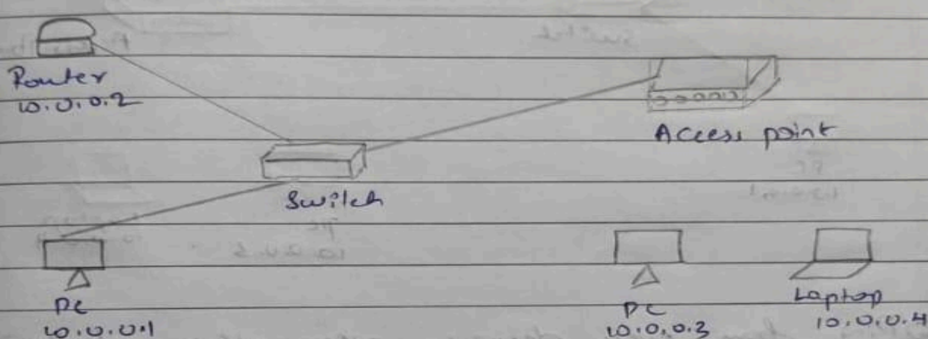
18/12/24

Lab No 13

WLAN

Aim:- To construct a wireless LAN & make the nodes communicate wirelessly.

Initial Topology:-



Procedure:-

① Create the topology as given above & configure the devices.

② Configure Access Point:

Check Access Point → config → Port 1:

SSID: bmsce

Select ⦿ WEP

Set key: 1234567890

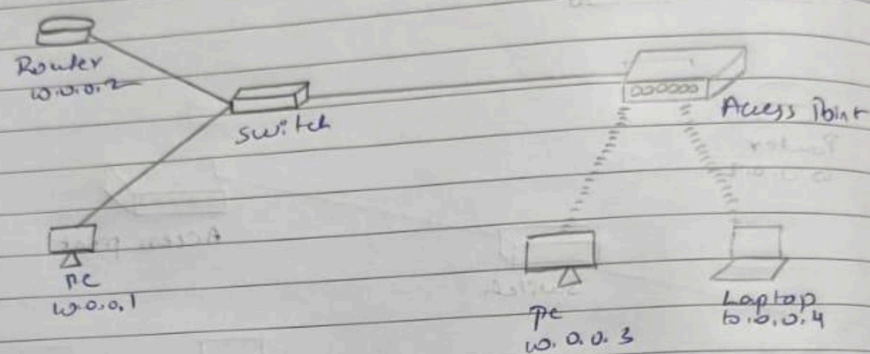
③ Configure PC & laptop with wireless standards:

- Switch off device
- Drag the existing PT-HOST-NM-1AM 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 address & 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 b/w wired & wireless devices.
- SSID broadcasting: announces the wireless networks name (SSID) to allow devices to ~~...~~ connect using WEP, WPA or WPA2

③ WMP300N wireless interface:
- wireless network adapter that enables devices to communicate with access point using wireless signals.

④ Pinging: ~~10.0~~ 10.0.0.1 to 10.0.0.3:
10.0.0.1 -> switch -> Accesspoint -> 10.0.0.3
- This is after the ARP tables are updated after

35

broadcasting.

(5) Pinging : 10.0.0.3 to 10.0.0.1 :-

10.0.0.3 -> Access Point -> switch -> 10.0.0.1

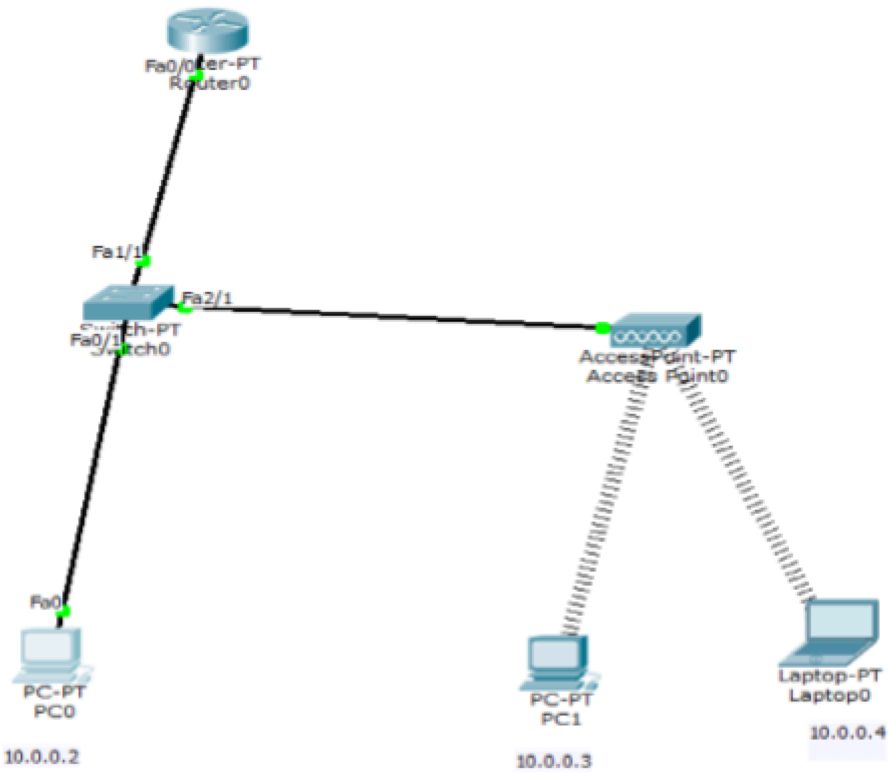
(6) Pinging : 10.0.0.3 to 10.0.0.4 :-

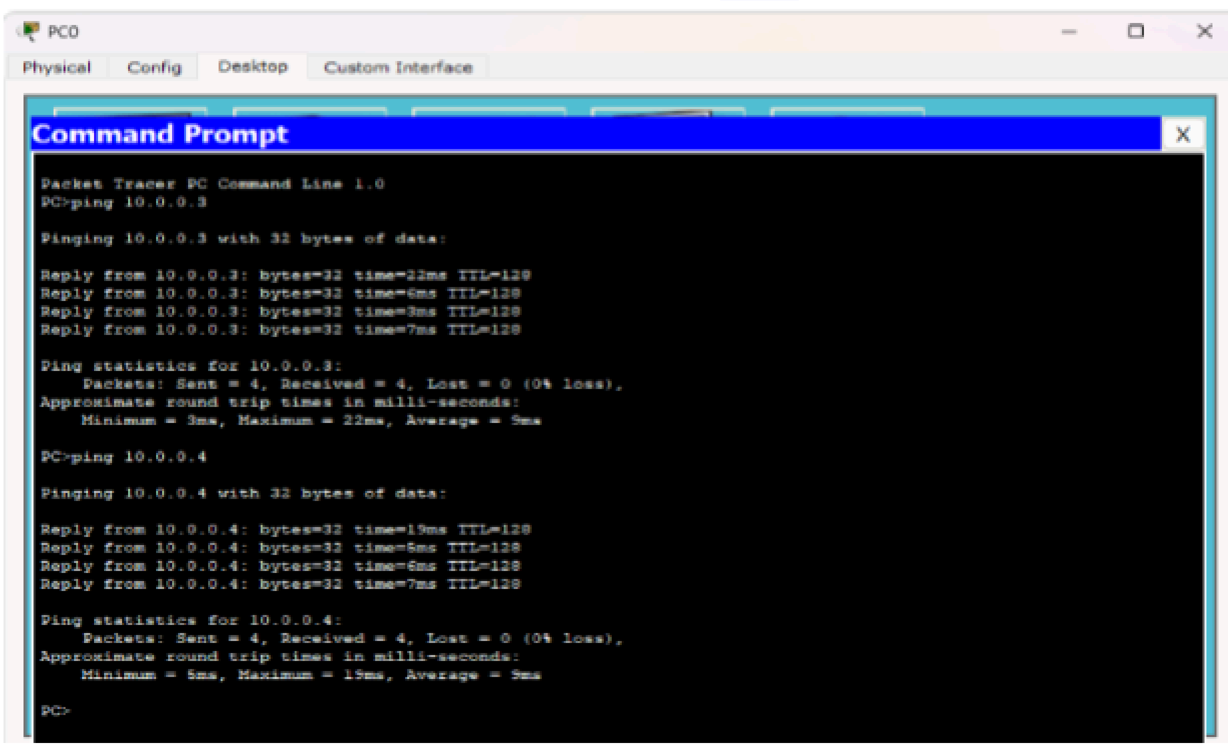
10.0.0.3 -> Accesspoint -> 10.0.0.4

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

N
3/1/25

Screen Shots:





PART-B

Program 14

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

Code :

```
def crc(data, gen_poly):
    """
    Calculates the CRC checksum for the given data and generator polynomial.

    Args:
        data: The input data as a binary string.
        gen_poly: The generator polynomial as a binary string.

    Returns:
        The CRC checksum as a binary string.
    """

    padded_data = data + '0' * (len(gen_poly) - 1)
    check_value = padded_data[:len(gen_poly)]

    for _ in range(len(data)):
        if check_value[0] == '1':
            check_value = xor(check_value, gen_poly)
        check_value = check_value[1:] + (padded_data[len(gen_poly) + _] if len(gen_poly) + _ <
len(padded_data) else '0')

    return check_value[1:]


def xor(a, b):
    """
    Performs bitwise XOR operation on two binary strings.

    Args:
        a: The first binary string.
        b: The second binary string.

    Returns:
        The result of XOR operation as a binary string.
    """
    return ''.join('0' if x == y else '1' for x, y in zip(a, b))
```

```
if __name__ == "__main__":
    data = input("Enter data: ")
    gen_poly = input("Enter generator polynomial: ")

    # Calculate CRC
    crc_value = crc(data, gen_poly)
    print("CRC:", crc_value)

    # Transmit data with CRC
    transmitted_data = data + crc_value
    print("Transmitted Data:", transmitted_data)

    # Receive data
    received_data = input("Enter received data: ")

    # Check for errors
    remainder = crc(received_data, gen_poly)
    print("No Error" if remainder == '0' * (len(gen_poly) - 1) else "Error detected")
```

Output

```
Enter data: 1100110
Enter generator polynomial: 1101
CRC: 100
Transmitted Data: 1100110100
Enter received data: 1100110100
No Error

=== Code Execution Successful ===
```

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code :

```
import time
import random

def leaky_bucket(packets, bucket_size, output_rate):
    remaining = 0  # Bytes remaining in the bucket

    for packet in packets:
        if packet > bucket_size:
            print(f"Packet of size {packet} bytes exceeds bucket capacity ({bucket_size} bytes) -
REJECTED")
        elif packet + remaining > bucket_size:
            print(f"Bucket capacity exceeded with packet size {packet} bytes - REJECTED")
        else:
            remaining += packet
            print(f"\nPacket of size {packet} bytes added to bucket")
            print(f"Bytes in bucket: {remaining}")
            # Transmit data from the bucket
            while remaining > 0:
                time.sleep(1)  # Simulate time for transmission
                if remaining <= output_rate:
                    print(f"Transmitting {remaining} bytes")
                    remaining = 0
                else:
                    print(f"Transmitting {output_rate} bytes")
                    remaining -= output_rate
                print(f"Bytes remaining in bucket: {remaining}")

if __name__ == "__main__":
    # Generate random packet sizes
    packets = [random.randint(1, 100) for _ in range(5)]
    print(f"Generated packets: {packets}")

    # Get bucket size and output rate from the user
    bucket_size = int(input("Enter bucket size: "))
    output_rate = int(input("Enter output rate: "))

    # Simulate leaky bucket
    leaky_bucket(packets, bucket_size, output_rate)
```

Output

Clear

```
Generated packets: [80, 63, 57, 12, 69]
Enter bucket size: 60
Enter output rate: 30
Packet of size 80 bytes exceeds bucket capacity (60 bytes) - REJECTED
Packet of size 63 bytes exceeds bucket capacity (60 bytes) - REJECTED

Packet of size 57 bytes added to bucket
Bytes in bucket: 57
Transmitting 30 bytes
Bytes remaining in bucket: 27
Transmitting 27 bytes
Bytes remaining in bucket: 0

Packet of size 12 bytes added to bucket
Bytes in bucket: 12
Transmitting 12 bytes
Bytes remaining in bucket: 0
Packet of size 69 bytes exceeds bucket capacity (60 bytes) - REJECTED

=== Code Execution Successful ===
```

Program 16

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

Code and Output:

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

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

```
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()
```

Program 17

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

Code and Output:

```
from socket import *

# Set up server
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))

print("The server is ready to receive")

# Listen for incoming connections
while True:
    sentence, clientAddress = serverSocket.recvfrom(2048)

    try:
        # Open and read the requested file
        with open(sentence.decode(), "r") as file:
            fileContents = file.read(2048)

        # Send the file contents to the client
        serverSocket.sendto(fileContents.encode(), clientAddress)
        print("Sent back to client:", fileContents)

    except FileNotFoundError:
        errorMessage = "File not found!"
        serverSocket.sendto(errorMessage.encode(), clientAddress)
        print("File not found error sent to client.")
```

```
from socket import *

# Server details
serverName = "127.0.0.1"
serverPort = 12000

# Create UDP socket
clientSocket = socket(AF_INET, SOCK_DGRAM)

# Input file name
sentence = input("Enter file name: ")

# Send the file name to the server
clientSocket.sendto(sentence.encode(), (serverName, serverPort))

# Receive file contents from the server
filecontents, serverAddress = clientSocket.recvfrom(2048)
print('From Server:', filecontents.decode())

# Close the socket
clientSocket.close()
```

WIRESHARK

Wireshark

It is a powerful used network protocol analyzer. It allows you to capture and inspect data packets travelling over a network in realtime, making it a crucial tool for studying computer networks, troubleshooting n/w issues and understanding protocols.

Features:

1. Packet capture: captures live n/w traffic from various interfaces (eg. WiFi)
2. Protocol analysis: supports 100's of protocols like TCP, UDP.
3. Filtering: isolate specific packets.
4. Visualization: display pkt details with hierarchical layers

Use cases:

1. Network Troubleshooting:
   - Diagnosing slow network speeds.
   - Identifying bottle necks or misconfiguration.
2. Security analysis:
   - detecting malicious traffic or intrusion.
3. Protocol Study:
   - understanding pkt structure & communication flow.

Common Filters:

- http : show only HTTP traffic
- tcp. port == 80 : show traffic on TCP port 80
- ip. addr == 192.168.1.1 : show pkt to or from a single specific IP address.
- UDP : show only UDP traffic.