VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Networks (23CS5PCCON)

Submitted by

BHANODAY KURMA

(1BM22CS066)

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 Academic Year 2024 - 25 (odd)

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 (23CS5PCCON)" carried out by **Bhanoday Kurma(1BM22CS066),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Laboratory report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

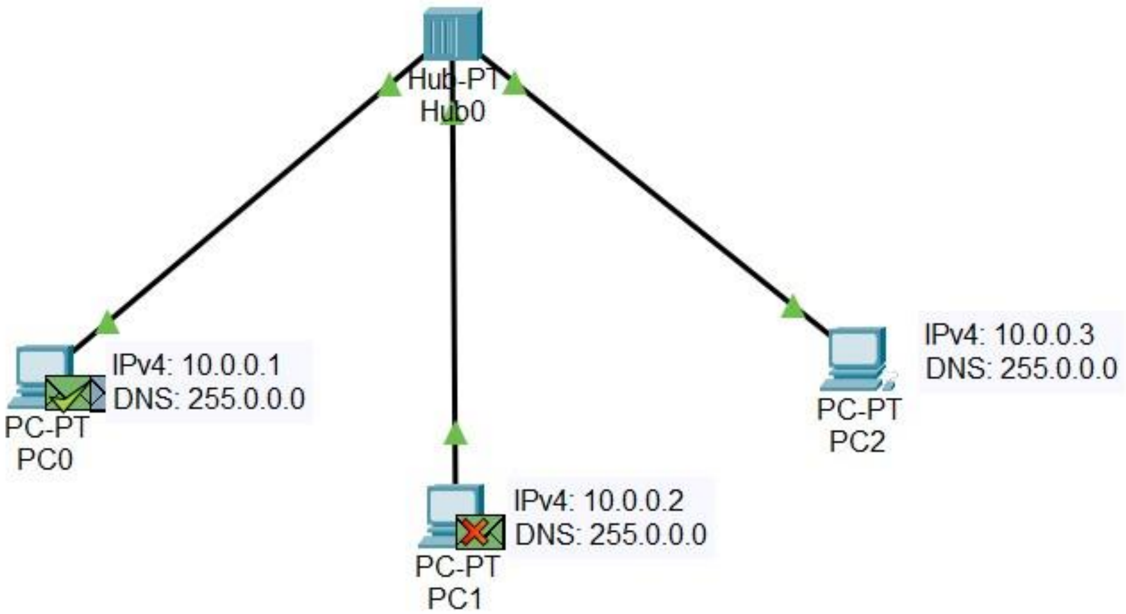
Dr. Nandhini Vineeth
Associate Professor
Department of CSE, BMSCE

Dr. Kavitha Sooda
Professor & HOD
Department of CSE, BMSCE

INDEX

LABORATORY PROGRAM – 1

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



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC2	ICMP		0.000	N	0	(edit)	

```
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 9ms, Average = 2ms
```

Week-1

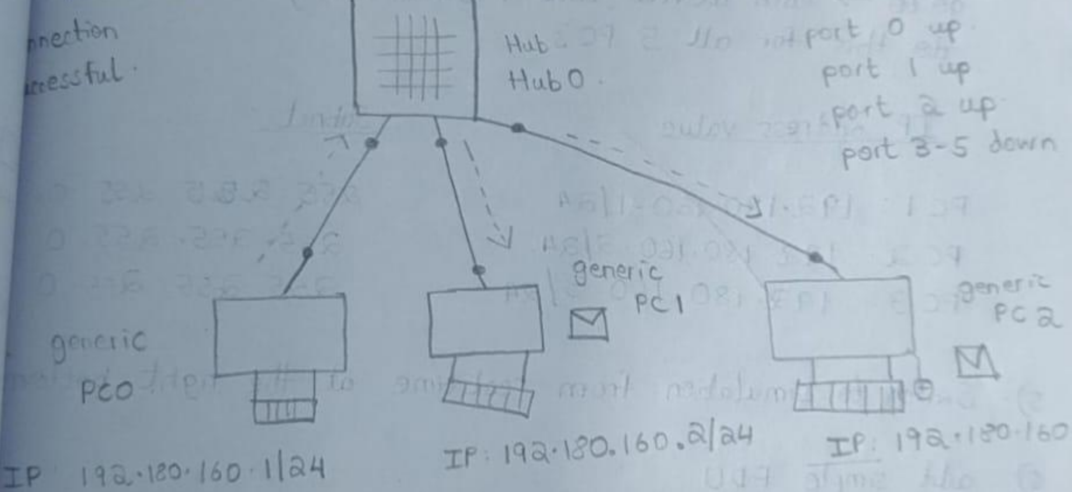
Switch & Hub.

~~Created a network by selecting end devices. Add a generic server & PC to the workspace, and used a copper cross over cable.~~

~~Sent a simple text message in realtime~~

Aim: To demonstrate the transmission for simple PDU b/w 2 devices connected using a hub & switch

Create a topology & simulate sending a simple PDU from source to destination using hub & switch as connecting devices & demonstrate ping msg



Create a topology & simulate sending a simple PDU from source to destination using hub and switche as connecting devicer and demonstrate ping message.

Observation:

Setup:

1) Open cisco packet tracer: Launch the application

2) Add devices from icon menu → Select end devices → add 3 PC's to workspace

3) Connect Devices:

Use copper straight wire through cable to connect each PC to the Hub, establish the conection should turn red to green.

4) IP Config:

go to → each device click → select config → then fast ethernet do this for all 3 PC's.

IP address value	Subnet
PC1: 192.180.160.1/24	255.255.255.0
PC2: 192.180.160.2/24.	255.255.255.0
PC3: 192.180.160.3/24	255.255.255.0

5) Switch to simulation from real time at the right bottom.

6) add simple PDU.

Click on the source PC0 & destination PC2.
this will create an ICMP packet from PC1 to PC2

7) Run the simulation
click autoplay / capture.

7) Result:

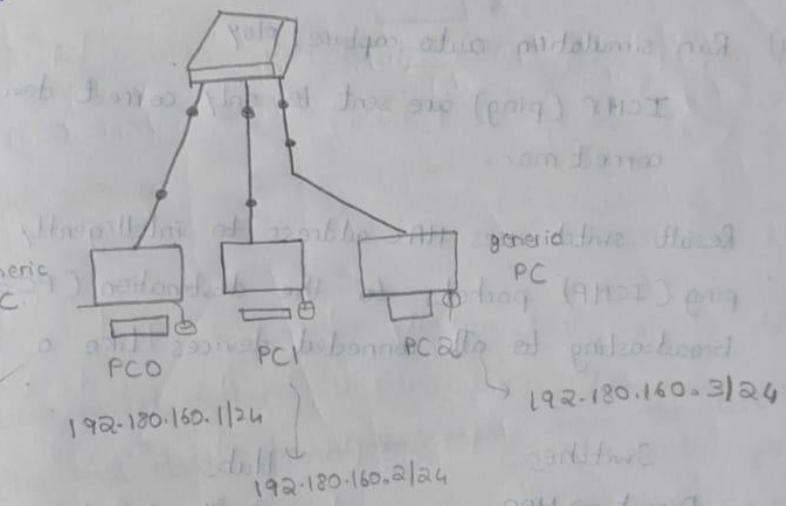
→ From PC0 to p2 & p1 the PDU (protocol data unit).

→ return msg/ PDU is sent from PC2 to PC0 that is receiver

→ But only PC2 will accept the PDU & hence X mark on P1

since it is hub broadcast happens to all connected to a hub. only intended PC will be received.

* Connecting PC to switch:



Observation:

1) Open cisco packet tracer

2) Add device

switch → network devices

Network devices - (generic switch).

end devices

generic PC.

3) Connect the 3 PCD12 to switch parts using copper straight through.

4) Setup devices by providing ip's

goto config → fast ethernet → input i/p.

IP address : 192.180.160.1

IP address : 192.180.160.2

IP address : 192.180.160.3

5) Switch to simulation mode, choose visible event/protocol.

6) Add PDU to source and destination to PC2.

7) Run simulation auto capture/play.

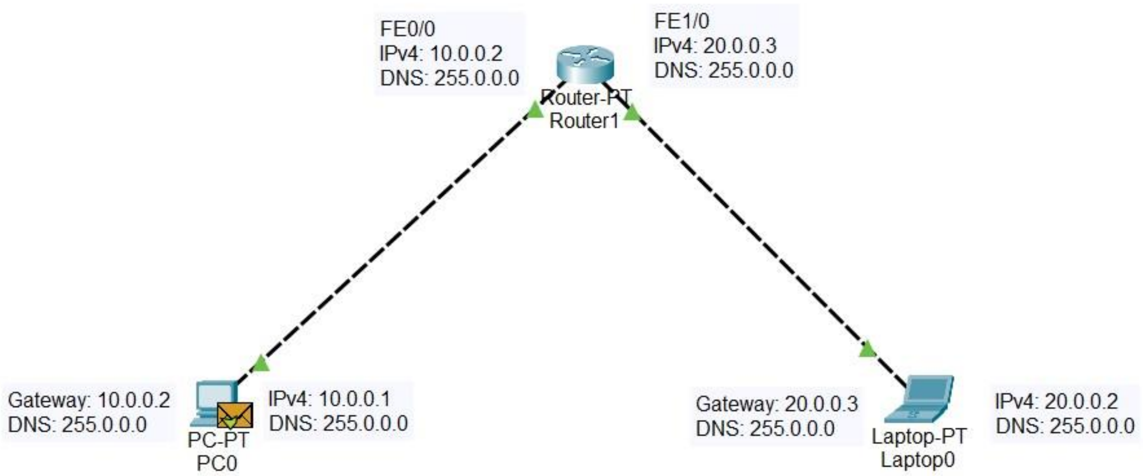
ICMP (ping) are sent to only correct device with correct mac.

8). Result switch uses MAC address to intelligently forward the ping (ICMP) packets to the destination (PC2) rather than broadcasting to all connected devices like a hub would.

Switches.	Hubs.
• Direct → MAC.	Broadcast model.
• Intelligent.	Not intelligent.
• Works in full duplex	Works in half duplex
• ~~Only a~~ Multiple devices can send data at same time	only one device can send data at a time.

LABORATORY PROGRAM – 2

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



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	In Progress	PC0	Laptop0	ICMP		0.000	N	1	(edit)	
	In Progress	PC0	Laptop0	ICMP		0.000	N	2	(edit)	

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255
Reply from 20.0.0.3: bytes=32 time<1ms TTL=255

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

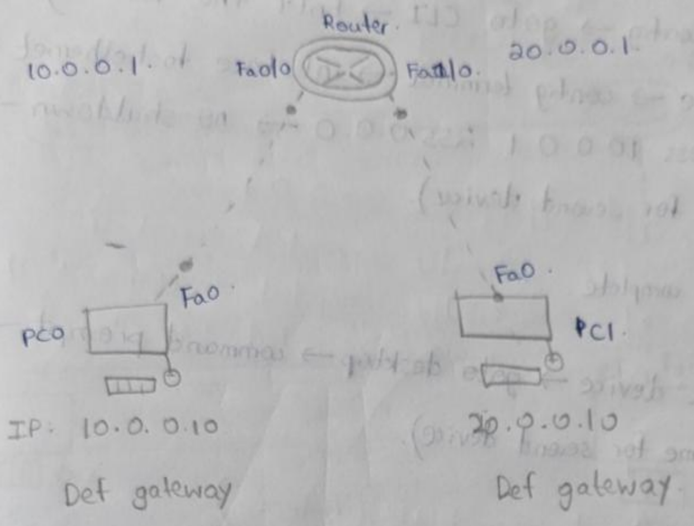
9|10|24

Lab - 2

Router

Config IP address to routers in packet tracker, explore the messages. ping responses, destination, unreachable, request timed out, reply.

Aim: To demonstrate the connection b/w two end devices in two different networks using the router.



Procedure:

1) Open cisco packet traker
2). Add the end devices and the router from the left hand icon menu
3). Connect the device: use a copper crossover cable to connect each PC to the router.
4) IP Config on each device → go to config → fast ethernet enter IP:

PC0 : 10. 0.0.10 255. 0.0.0

PC1 : 20.0.0.10 255. 0. 0. 0.

Change gateway:

PC0 : 10.0.0.1

PC1 : 20.0.0.1

5). Route Config → goto CLI → Enter the following command → Enable → config terminal → interface fastethernet 0/0 → ip address 10.0.0.1 255.0.0.0 → no shutdown → exit (Repeat for second device).

6) Setup is complete.

7) Select a device → goto desktop → command prompt → ping (same for second device).

Observation:

We observe that initially when ping request is sent we get a time out error (Router locates the second device) but after some time we get 3 replies & the message that packet have been sent & received & displays the approx time that it takes to complete all this.

Routing table shows the following:

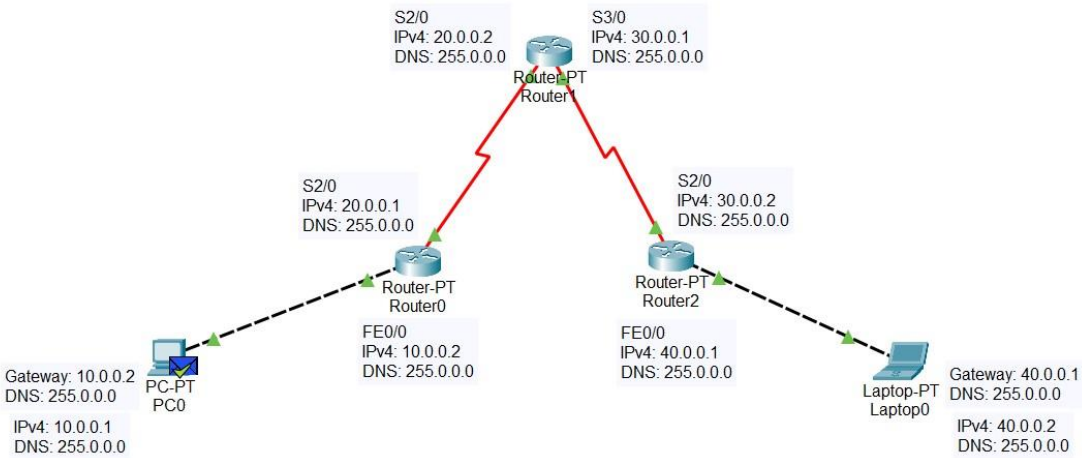
Connected C : 10.0.0.0/8 is directly connected, FastEthernet 0/0.

Connected : 20.0.0.0/8 is directly connected, FastEthernet 1/0.

20/11/24

LABORATORY PROGRAM – 3

Configure static route to the Router.



SHOW IP ROUTE

```
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
S    40.0.0.0/8 [1/0] via 20.0.0.2
```

Figure 3.1: Router0

```
S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
```

Figure 3.2: Router1

```
S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
```

Figure 3.3: Router3.3

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

23/10/24

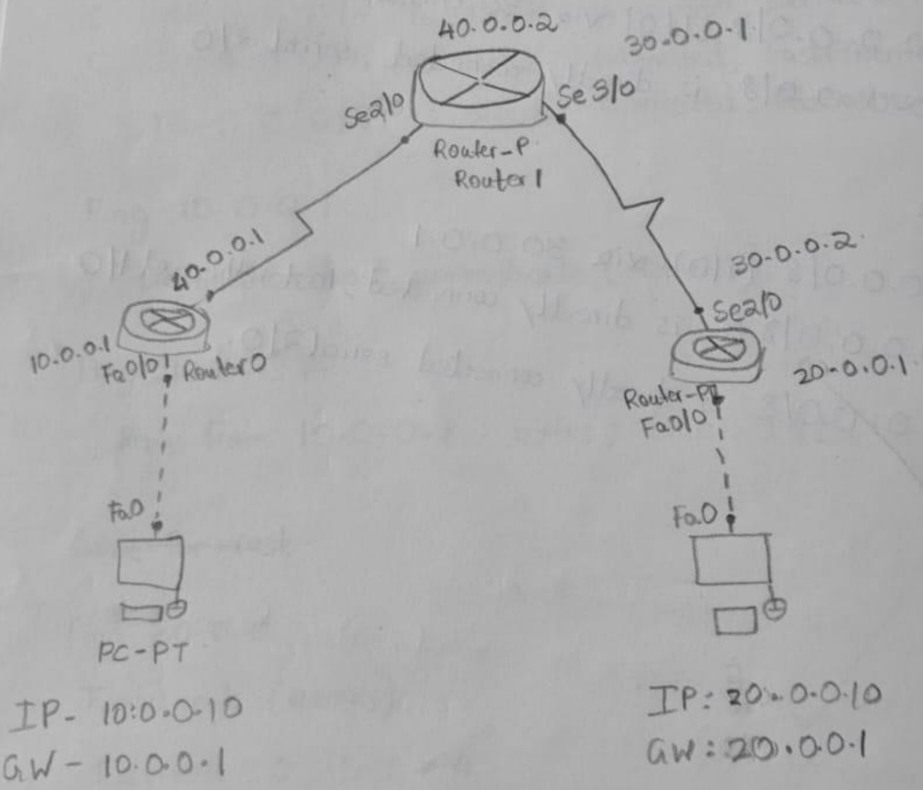
Lab 4

Q) Configure default, static router to router:

o/p seen

Aim: To demonstrate static routering & default routing using 3 routers & 2 PC's.

Topology:



Procedure:

1) Open Cisco packet Tracker
2) Add 2 end devices and 3 routers
3) Connect PC0 to Router 0 and PC1 to Router 2 using copper cross-over
4) Connect Router 0 to Router 1 using Serial DCE 2/0 and Router 1 to Router 2 using Serial DCE 3/0
5) IP Config on device → ip address
   PC0 : 10.0.0.10 255.0.0.0
   PC1 : 20.0.0.10 255.0.0.0

def gateway → PC0 : 10.0.0.1
PC1 : 20.0.0.1

6) Router 0 Config → CLI

enable
config terminal
interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
no shutdown
exit

Repeat the same for Router 2 →

interface fastethernet 1/0
ip address 20.0.0.1 255.0.0.0
no shutdown
exit

⑦. To connect b/w routers

Router 0 → CLI

enable
config terminal
interface serial 2/0
ip address 40.0.0.1 255.0.0.0
exit

Router 1 → CLI
interface Serial 2/0
ip address 40.0.0.2 255.0.0.0

interface serial 3/0
ip address 30.0.0.1 255.0.0.0
exit.

Router 2 → CLI

enable
config terminal
interface serial 3/0
ip address 30.0.0.2 255.0.0.0
exit

⑧. Static Router on Router 1 :

enable
config terminal
ip route 10.0.0.1 255.0.0.0 40.0.0.1
ip route 20.0.0.1 255.0.0.0 30.0.0.2

⑨ Default Route on Router 0 & Router 2 →

enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.2

enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1

⑩ Select PC0 → Desktop → CMD → ping msg to PC's and other routers

Observation:

After all the connections are done when we ping msg from PC0 to PC1 and other routers For PC1 (Destination host unreachable) for Router 2 and Router 1 (Request Timed out) only for router 0 (Packets : Sent=4, Received=4, Lost=0) as it is directly connected through copper cross over.

Ping: 20.0.0.10
Destination host unreachable (x 4)
Received = 0, Lost = 4

Ping 30.0.0.2
Request timed out (x4)
Received = 0 lost = 4

Ping 20.0.0.10.
Sent = 4, received = 4, lost = 0.

ping 30.0.0.2
Sent = 4, Received = 4, lost = 0.

ping 30.0.0.1
sent = 4, Received = 4, lost = 0.

Rooting has been observed as follows: (Router 0)

C: 10.0.0.0/8 is directly connected, fastethernet 0/0
C: 40.0.0.0/8 is directly connected serial 2/0
S* 0.0.0.0/0 [1/0] via 40.0.0.2.

Rooting has been observed: (Router 2)

S: 10.0.0.0/8 [1/0] via 40.0.0.1
S: 20.0.0.0/8 [1/0] via 50.0.0.2
C: 30.0.0.0/8 is directly connected serial 3/0.
C: 40.0.0.0/8 is directly connected serial 2/0.

~~PC0:~~
~~Ping: 40.0.0.10.~~
~~Reply from 40.0.0.10, bytes = 32, time = 6ms, TTL = 126~~

~~PC1:~~
~~Ping 10.0.0.10.~~
~~Reply from 10.0.0.10, bytes = 32, time~~

Routing has been observed as follows for (Router 2):

C: 20.0.0.0/8 is directly connected, fastethernet1/0.

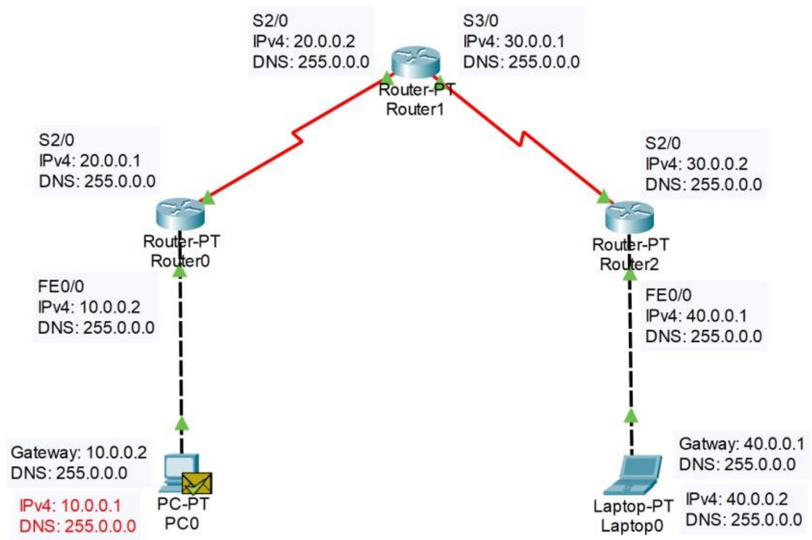
C: 30.0.0.0/8 is directly connected, serial3/0

S: 0.0.0.0/0 [1/0] via 30.0.0.1.

N
23/10/24

LABORATORY PROGRAM – 4(A)

Configure default route, static route to the Router.



SHOW IP ROUTE

```
Gateway of last resort is 20.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.2
```

Figure 4.1: Router0

```
S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
```

Figure4.2: Router1

```
Gateway of last resort is 30.0.0.1 to network 0.0.0.0

C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 30.0.0.1
```

Figure 4.3: Router2

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=34ms TTL=125
Reply from 40.0.0.2: bytes=32 time=33ms TTL=125
Reply from 40.0.0.2: bytes=32 time=30ms TTL=125
Reply from 40.0.0.2: bytes=32 time=33ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 30ms, Maximum = 34ms, Average = 32ms
```

23/10/24

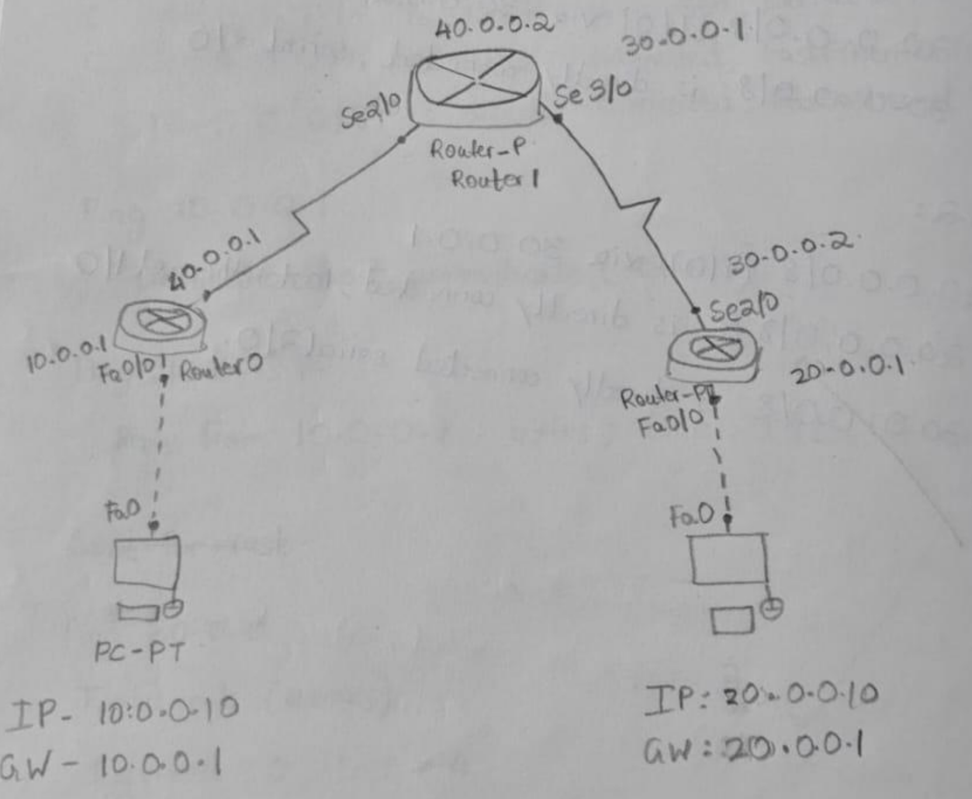
Lab 4

Q) Configure default, static router to router:

o/p seen

Aim: To demonstrate static routing & default routing using 3 routers & 2 PC's.

Topology:



Procedure:

1) Open Cisco packet Tracker
2) Add 2 end devices and 3 routers
3) Connect PC0 to Router 0 and PC1 to Router 2 using copper cross-over
4) Connect Router0 to Router 1 using serial DCE 2/0, and Router 1 to Router 2 using Serial DCE 3/0
5). IP Config on device → ip address
PC0: 10.0.0.10 255.0.0.0.
PC1: 20.0.0.10 255.0.0.0

def gateway → PC0: 10.0.0.1
PC1: 20.0.0.1

6) Router 0 Config → CLI

enable
config terminal
interface fastethernet 0/0.
ip address 10.0.0.1 255.0.0.0
no shutdown
exit

Repeat the same for Router 2 →.

interface fastethernet 1/0
ip address 20.0.0.1 255.0.0.0
no shutdown
exit.

⑦. To connect b/w routers

Router 0 → CLI

enable
config terminal
interface ~~t~~ serial 2/0
ip address 40.0.0.1 255.0.0.0
exit

Router 1 → CLI

interface serial 2/0
ip address 40.0.0.2 255.0.0.0

interface serial 3/0
ip address 30.0.0.1 255.0.0.0
exit

Router 2 → CLI

enable
config terminal
interface serial 3/0
ip address 30.0.0.2 255.0.0.0
exit

⑧. Static Router on Router 1 :

enable
config terminal
ip route 10.0.0.1 255.0.0.0 40.0.0.1
ip route 20.0.0.1 255.0.0.0 30.0.0.2

⑨ Default Route on Router 0 & Router 2 →

enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.2

enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1

⑩ Select PC0 → Desktop → CMD → ping msg to PC's and other routers.

Observation:

After all the connections are done when we ping msg from PC0 to PC1 and other routers For PC1 (Destination host unreachable) for Router 2 and Router 1 (Request Timed out) only for router 0 (Packets: Sent=4, Received=4, Lost=0) as it is directly connected through copper cross over.

Ping: 20.0.0.10
Destination host unreachable (x 4)
Received = 0, Lost = 4

Ping 30.0.0.2
Request timed out (x4)
Received = 0 lost = 4

Ping 20.0.0.10.
Sent = 4, received = 4, lost = 0.

ping 30.0.0.2
Sent = 4, Received = 4, lost = 0.

ping 30.0.0.1
sent = 4, Received = 4, lost = 0.

Rooting has been observed as follows: (Router 0)

C: 10.0.0.0/8 is directly connected, fastethernet 0/0
C: 40.0.0.0/8 is directly connected serial 2/0
S* 0.0.0.0/0 [1/0] via 40.0.0.2.

Rooting has been observed: (Router 1)

S: 10.0.0.0/8 [1/0] via 40.0.0.1
S: 20.0.0.0/8 [1/0] via 50.0.0.02
C: 30.0.0.0/8 is directly connected serial 3/0.
C: 40.0.0.0/8 is directly connected serial 2/0.

PC0:

~~Ping: 40.0.0.10.~~
~~Reply from 40.0.0.10, bytes = 32, time = 6ms, TTL = 126~~

PC1:

~~Ping 10.0.0.10.~~
~~Reply from 10.0.0.10, bytes = 32, time~~

Rooting has been observed as follows for (Router 2):

C: 20.0.0.0/8 is directly connected, fastbethernet1/0.

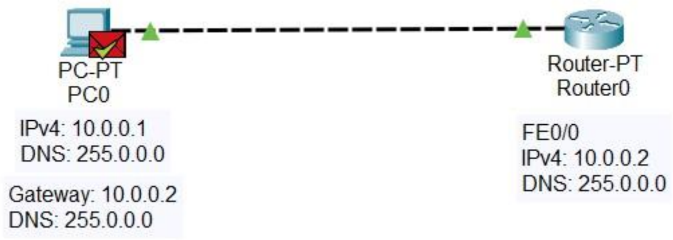
C: 30.0.0.0/8 is directly connected, serial3/0

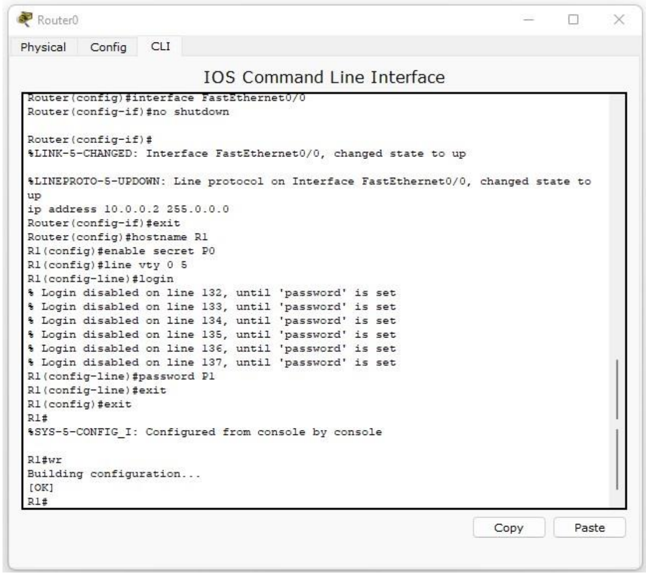
S: 0.0.0010 [I/0] via 30.0.0.1.

N 23/10/24

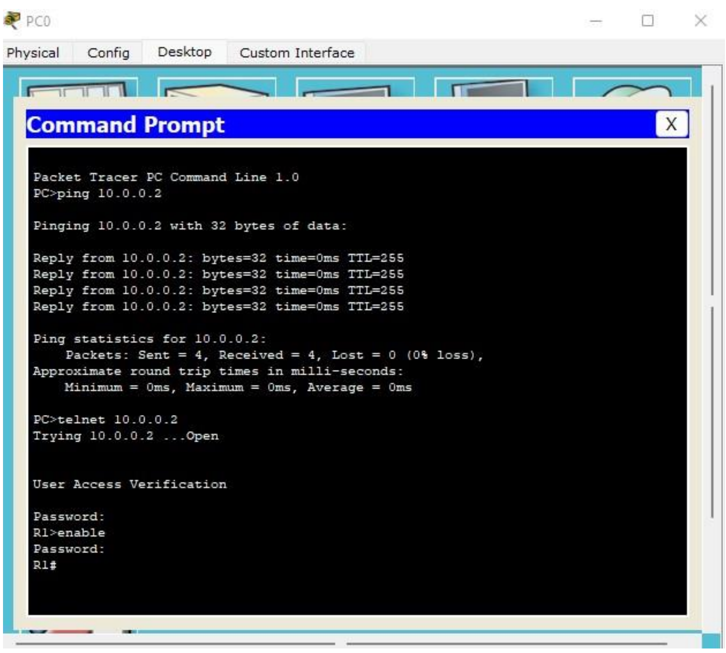
LABORATORY PROGRAM – 5

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





Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Router0	ICMP		0.000	N	0	(edit)	

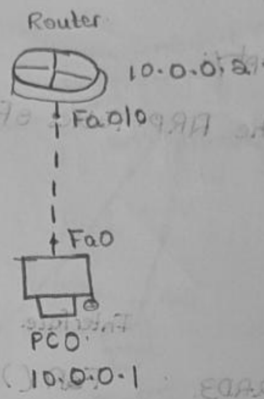


12|12|24

Lab- 10

Q). To understand the operation of TELNET by accessing the route in server room from a PC in IT office:

Topology:



Procedure:

1). Open Cisco Packet Tracer.

2). Setup the devices as shown in figure.

3). Assign IP address to PC's.

4). Setup the router in CLI, open CLI.

Enter → Enable → config terminal → hostname R1 → enable secret <password> → interface Fa0/0 → IP address 10.0.0.2 255.0.0.0 → No shutdown → link vty 0 3 → login → password <password2> → exit → exit → wr (To save changes in router) [done]

5). Go to CMD in PC0 & ping 10.0.0.2.

6). After 1st ping now. type Telnet 10.0.0.2

Result:

[Ping].

Pinging 10.0.0.2 with. 32 bytes of data.

Reply from 10.0.0.2 with. byles 32 time = 0ms., TTL = 255

:

:

Ping statistics:

[Telnet].

Trying 10.0.0.2 ..... open.

User access. verification.

<Enter Password 1>

R1 > enable.

Password: <Password 2>.

R1 # show IP route.

Gateway of last ~~word~~ route is not set

C: 10.0.0.0/8 is directly connected, Fa0/0.

R1 #.

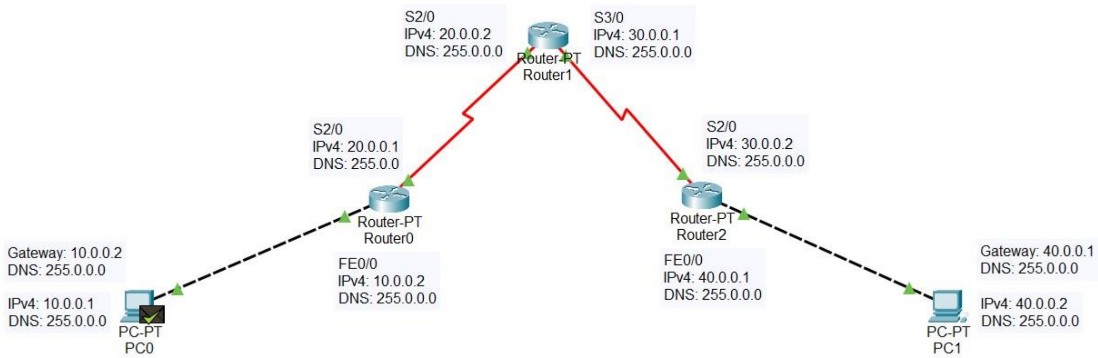
Observation:

Telnet is a test based protocol that enables remote communica over TCP/IP networks. It allows the execution of commands on a remote device, often used for initial setup or manageme In the experiment above we see that all configs & commands executed via Telnet mirrored those done directly on the rou but from PC interface instead.. Disadvantage is that Telnet lacks encryption making it less secure compared to SSH.

N 26/12/24

LABORATORY PROGRAM – 6

Demonstrate the TTL/ Life of a Packet.



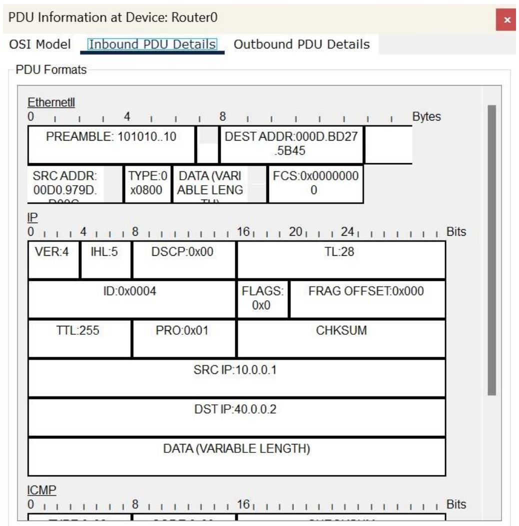


Figure 6.1: Inbound PDU, Router0

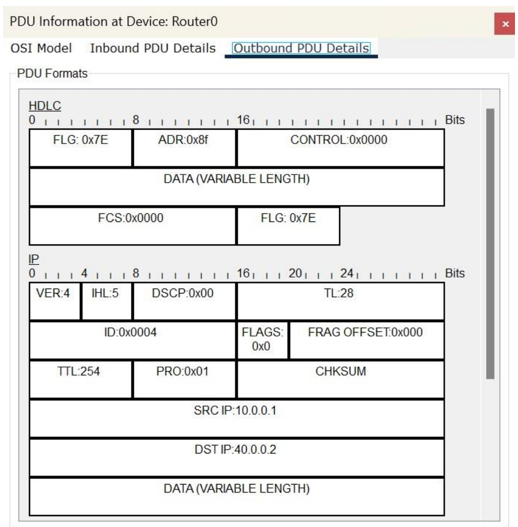


Figure 6.2: Outbound PDU, Router0

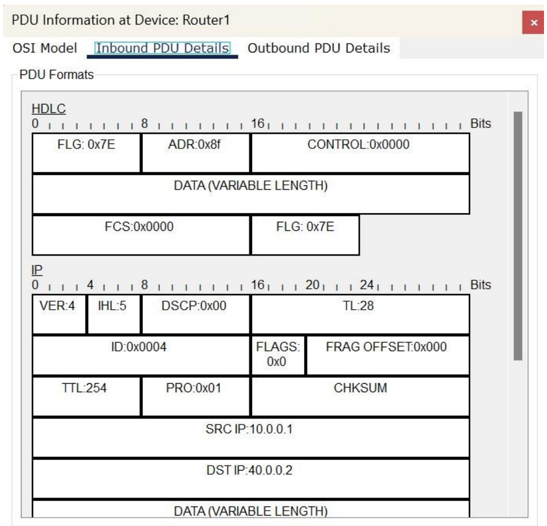


Figure 6.3: Inbound PDU, Router1

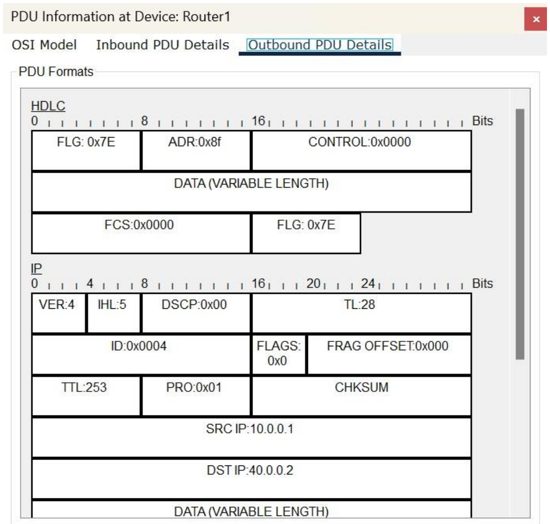


Figure 6.4: Outbound PDU, Router1

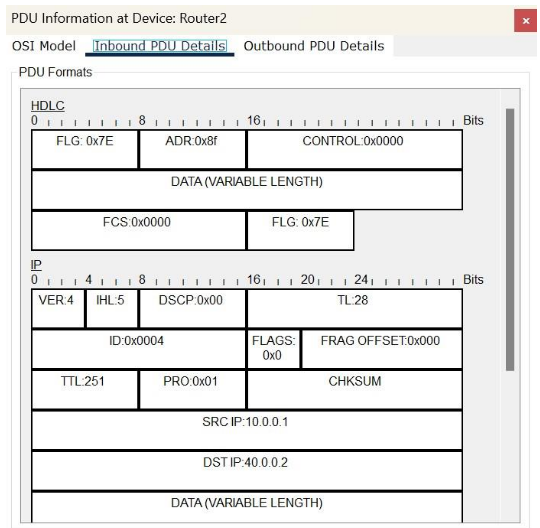


Figure 6.5: Inbound PDU, Router2

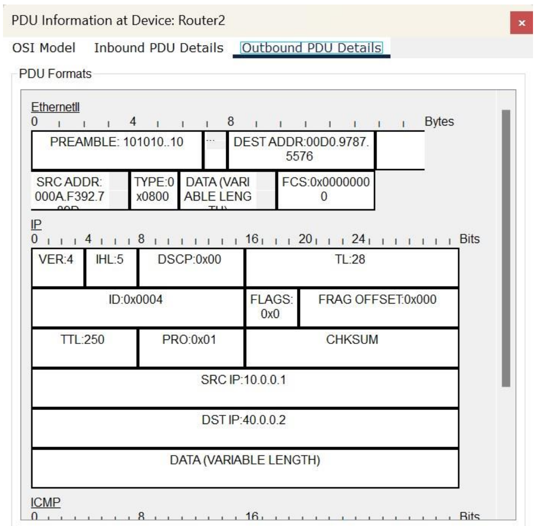


Figure 6.6: Outbound PDU, Router2

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC1	ICMP	■	0.000	N	0	(edit)	

```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=72ms TTL=123
Reply from 40.0.0.2: bytes=32 time=53ms TTL=123
Reply from 40.0.0.2: bytes=32 time=55ms TTL=123
Reply from 40.0.0.2: bytes=32 time=69ms TTL=123

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 53ms, Maximum = 72ms, Average = 62ms
```

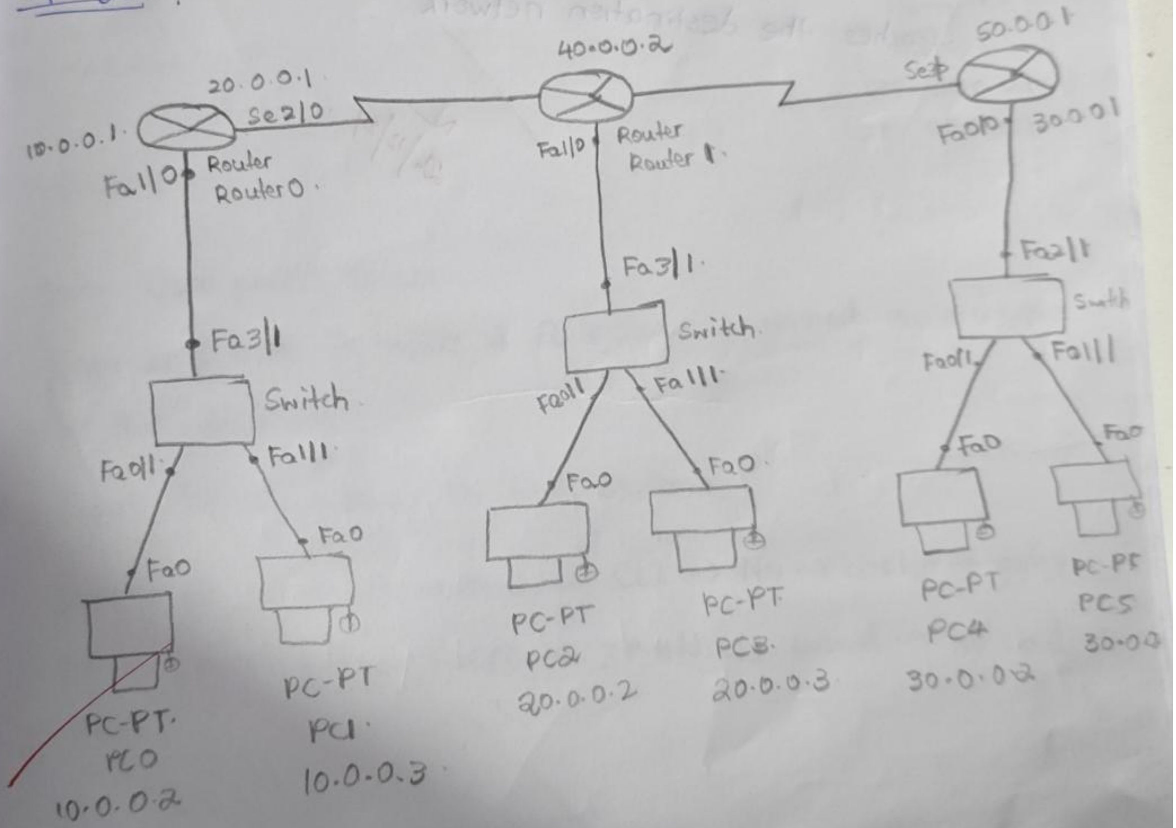
20/11/24

Exp-6

Demonstrate TTL / Life of a packet

Aim: To show how TTL changes when a simple PDU is transmitted from one system to another in different networks.

Topology:



* Procedure.

1). Cisco packet Tracer.
2) Arrange devices according to the topology.
3). Configure the routers according to gateways.
4). Configure the RIP in routers for all networks
5) Go to simulation mode
6) Select PDU & drag it to source PC. & destination PC
7) Click play on the simulation & capture all important data.

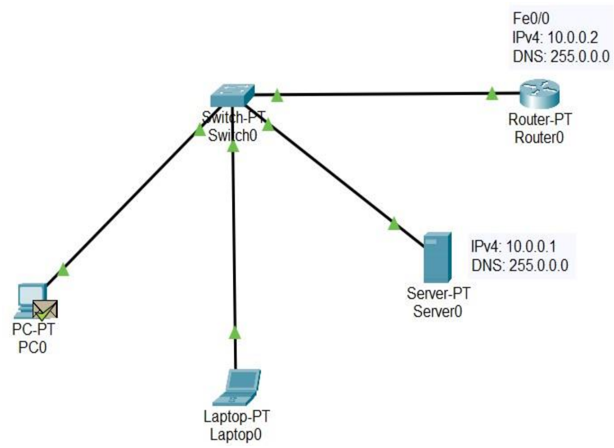
Observation:

We observe that after each network shift the TTL of the packet decreases by 1 from initial 255 till it reaches the destination network.

N 26/12/24

LABORATORY PROGRAM – 7(A)

To Configure IP addresses of the host using DHCP server within a LAN.



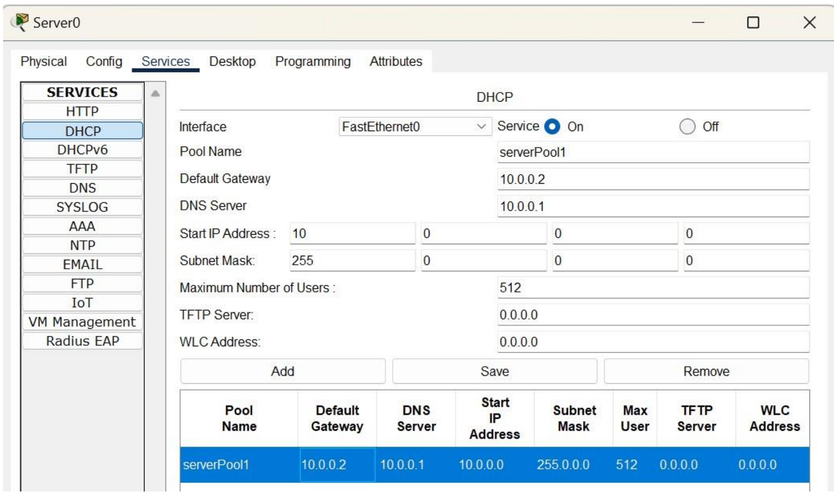


Figure 7.1: DHCP Service, Server0

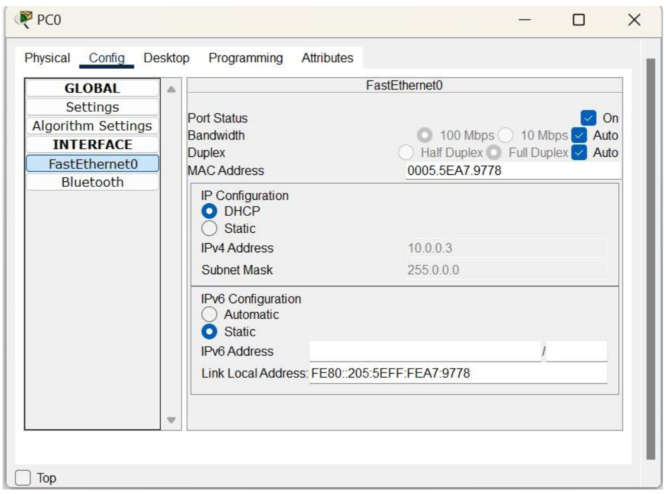


Figure 7.2: DHCP Service, PC0

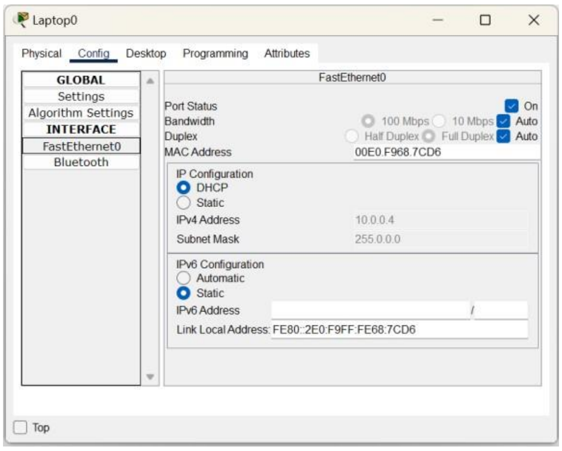
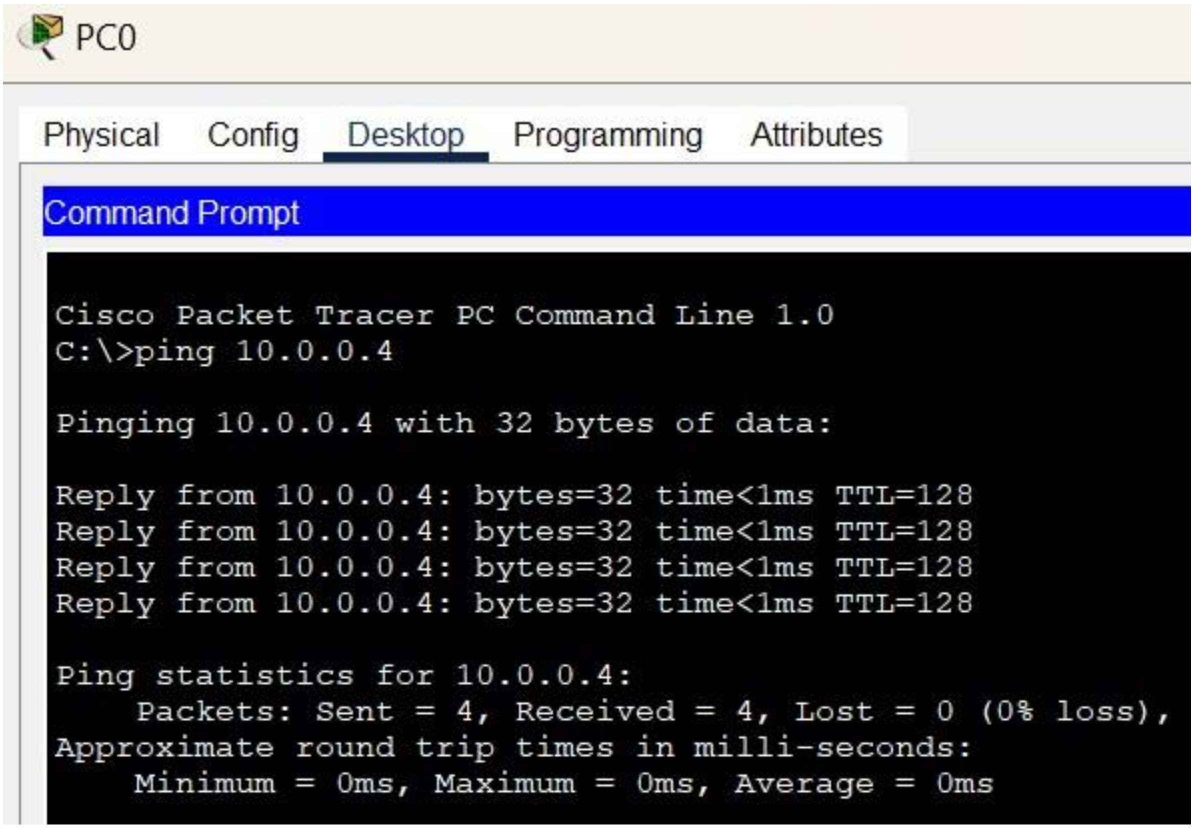


Figure 7.3: DHCP Service, Laptop0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	

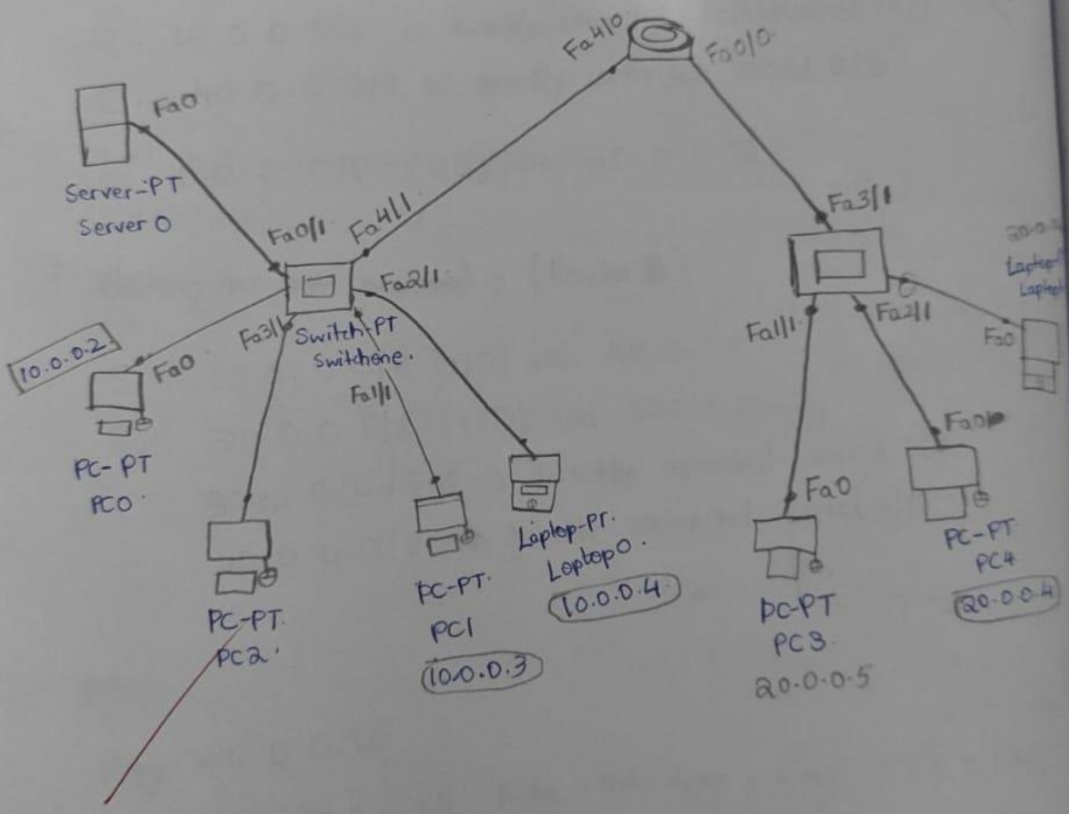


13/11/24 Lab-5

*Q) Configure DHCP within a LAN and outside a LAN:

Aim: To use server to give systems IP addresses through DHCP within 2 different networks.

Topology:



Procedure: (Within LAN)

1) Open cisco packet tracer.

2) Drag and drop the devices (end devices and servers).

3). Connect them using switch (copper straight through).

4). In server → Desktop → IP Config → set IP 10.0.0.2. & Gateway 10.0.0.1. & subnet.

5). In config → services → DHCP → create. Pool name., set gateway. 10.0.0.1., start IP 10.0.0.3. & Max n.o of users to 100 and add.

6). Goto each end device → desktop → IP config → click DHCP. → IP will be assigned.

Result:

In end device → cmd.

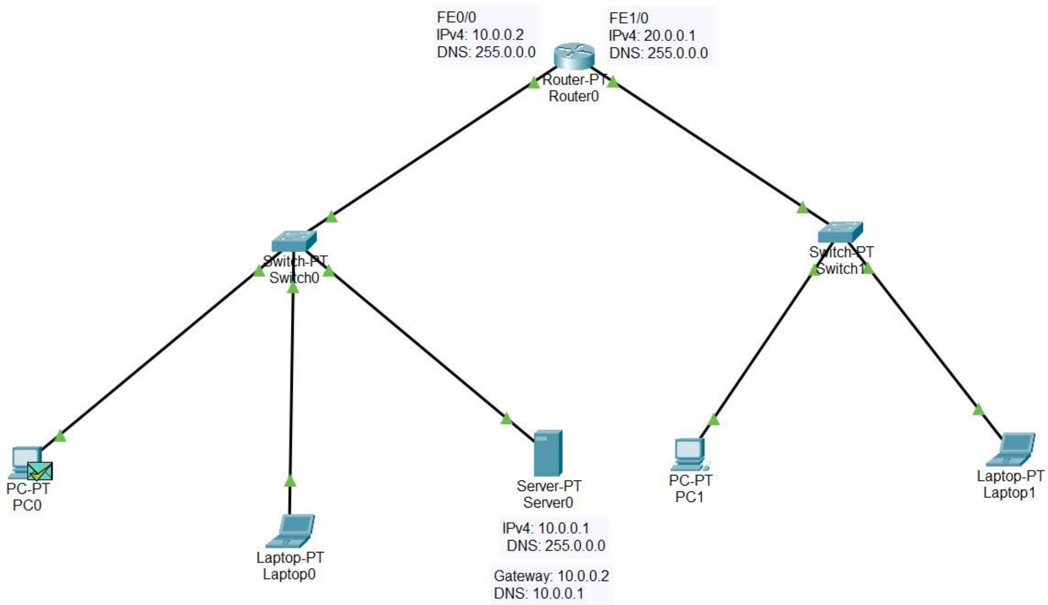
→ ping 10.0.0.4.

Pinging 10.0.0.4 with 32 bytes of data.

Reply from 10.0.0.4. bytes = 32., time = 0, TL= 120.

LABORATORY PROGRAM – 7(B)

To Configure IP addresses of the host using DHCP server outside a LAN.



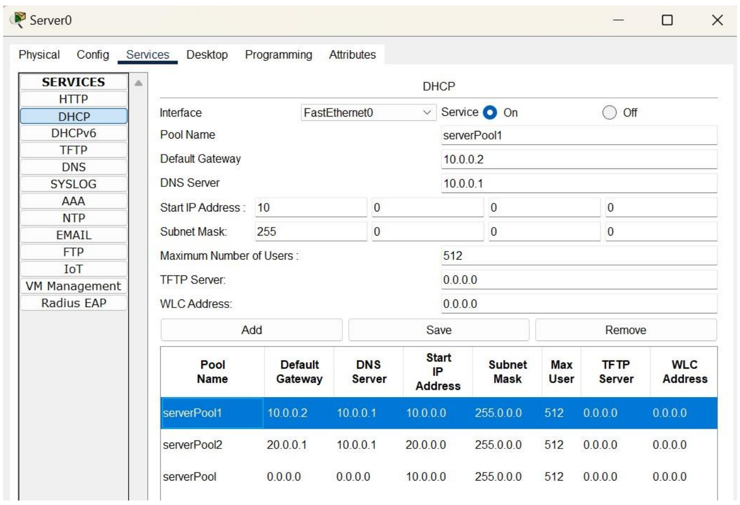


Figure 7.1.1: DHCP Service, Server0

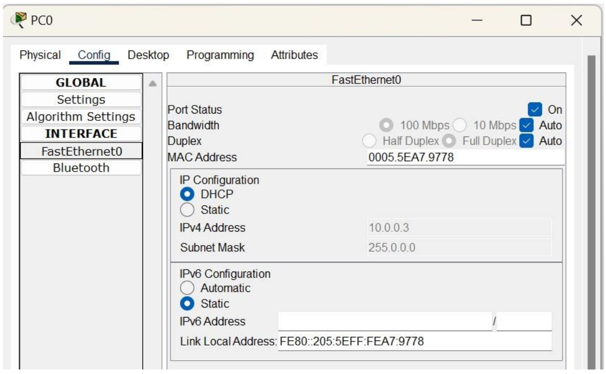


Figure7.2.2: DHCP Service, PC0

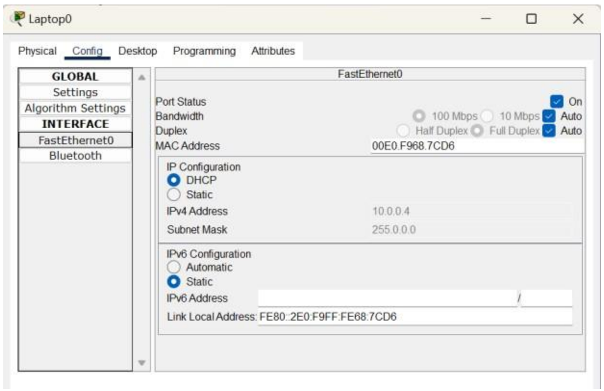


Figure 7.2.3: DHCP Service, Laptop0

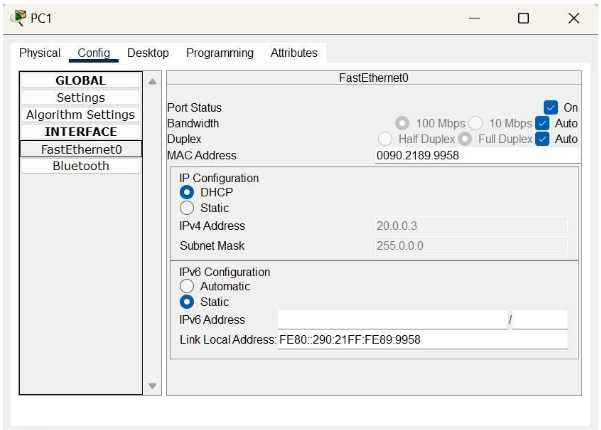


Figure 7.2.4: DHCP Service, PC1

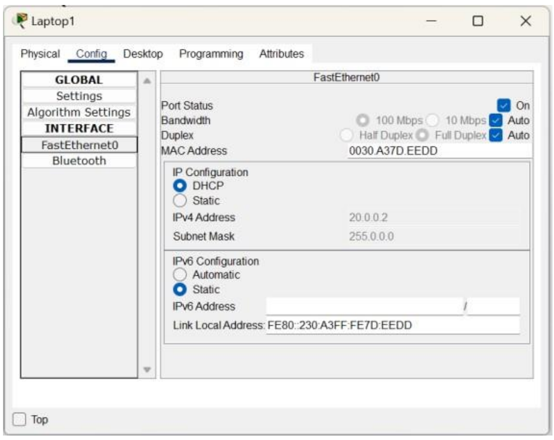


Figure 7.2.5: DHCP Service, Laptop1

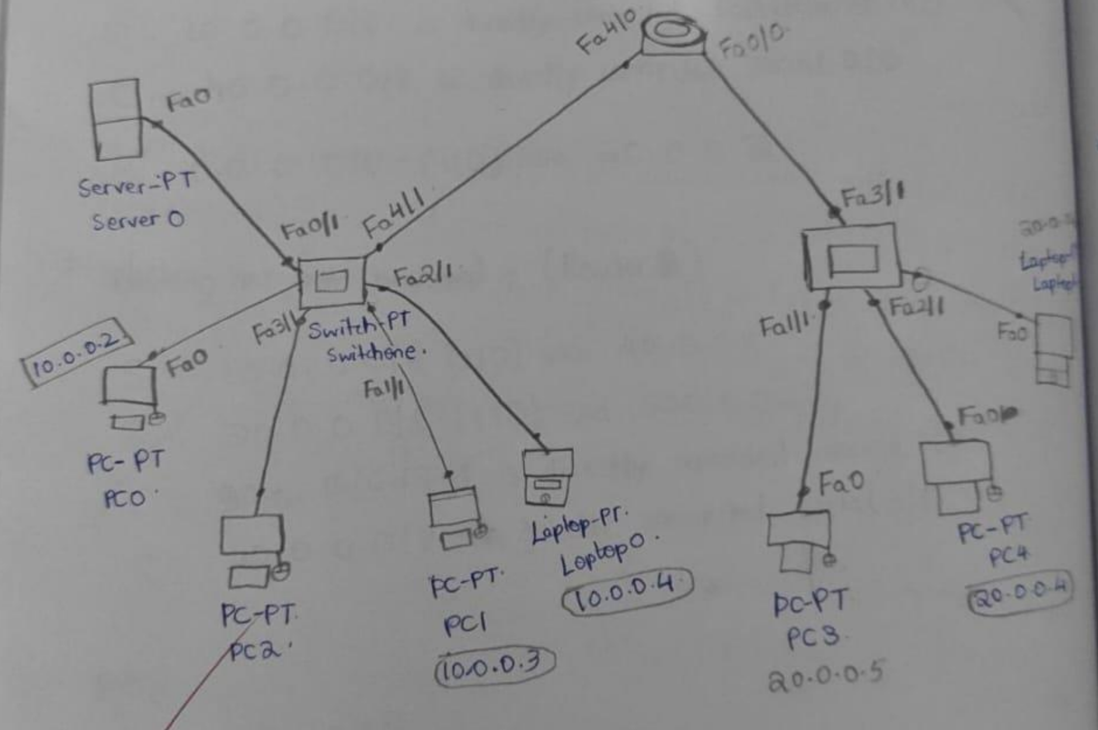
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	
	Successful	PC1	Laptop1	ICMP		0.004	N	1	(edit)	

13/11/24 Lab-5

* Q) Configure DHCP within a LAN and outside a LAN:

Aim: To use server to give systems IP addresses through DHCP within 2 different networks.

Topology:



* Procedure(Outside LAN):

1). In previous setup, add router, extra switch and few end devices.

2). Join switch & end devices through the same method as previous one.

3). Connect both switches to router.

4). In router → CLI → No → enable → config terminal.
→ interface Fa 4/0 → IP address 0.0.0.1. 255.0.0.0.
→ ip helper-address 10.0.0.2 → No shut.

5) Same for the second switch (change IP address to 20.0.0.1).

6). Go to each system → desktop → IP config → DHCP mode.
→ IP will be assigned (switch two).

Result:

Router.

Show IP route.

C. 10.0.0.0/8 is directly connected Fa4/0.
C 20.0.0.0/8 is directly connected Fa0/0.

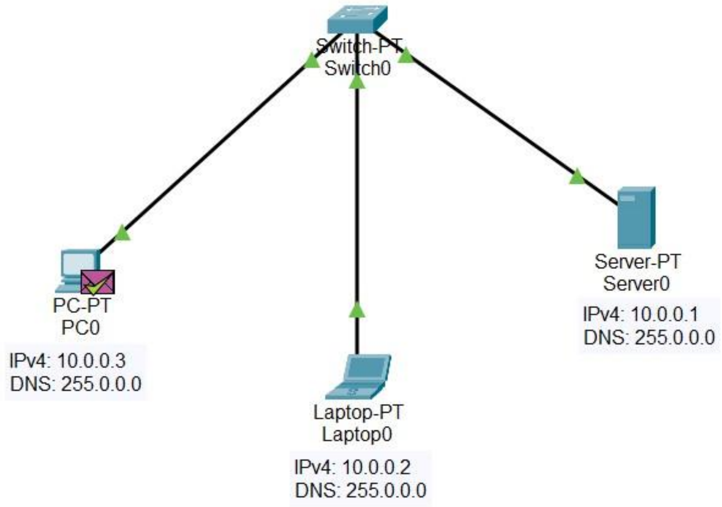
Observation:

We can use server to assign the IP address to any device connected to the server within or outside LAN. through DHCP.

N 13/11/24

LABORATORY PROGRAM – 8

To Configure DNS server to demonstrate the mapping of IP addresses and Domain names.



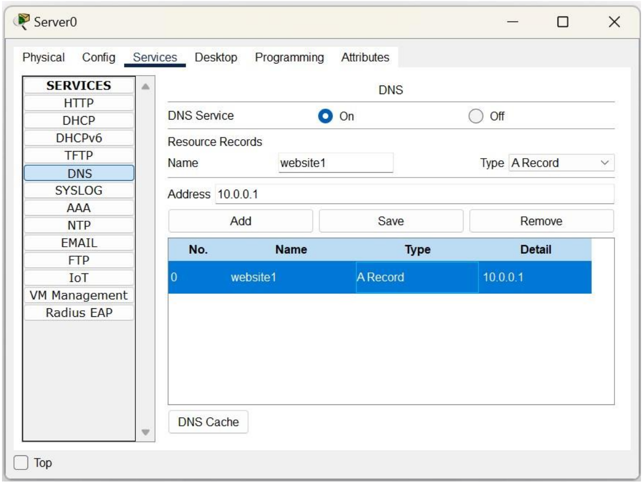
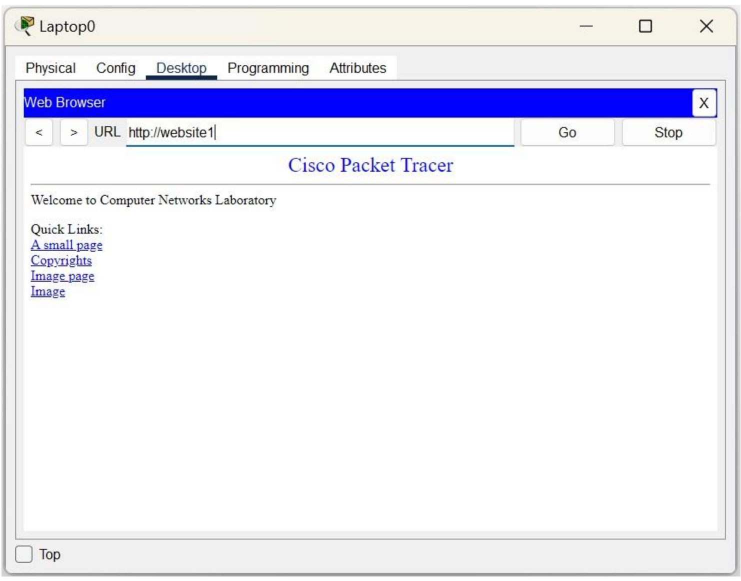


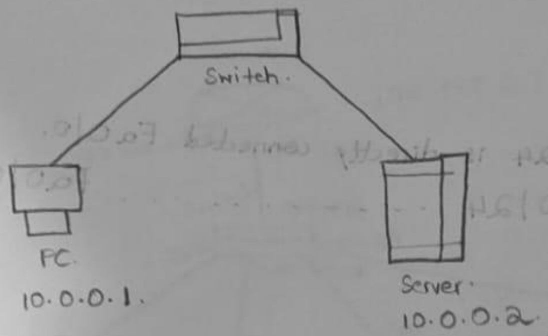
Figure 8.1: DNS Service, Server0



18|12|24 Lab-12

Q). Configure web-server, DNS within LAN.

Topology:



Procedure:

1). Set-up the LAN as per topology & configure the devices.

2). Go to ~~to~~ server → services → DNS.

Name: ~~Diddy~~.bmsce.

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 DNS mapping)

4). Go to PC → Desktop → Web Browser.

Type the url: http://bmsce.

Observation:

1) The webpage hosted by the server visible on the browser

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

3). DNS server is a server that contains a domain name: IP address mapping to which the end device send requests to map the name of IP address.

Welcome to CN Lab:

Quick Link:

A small page:

Copy right:

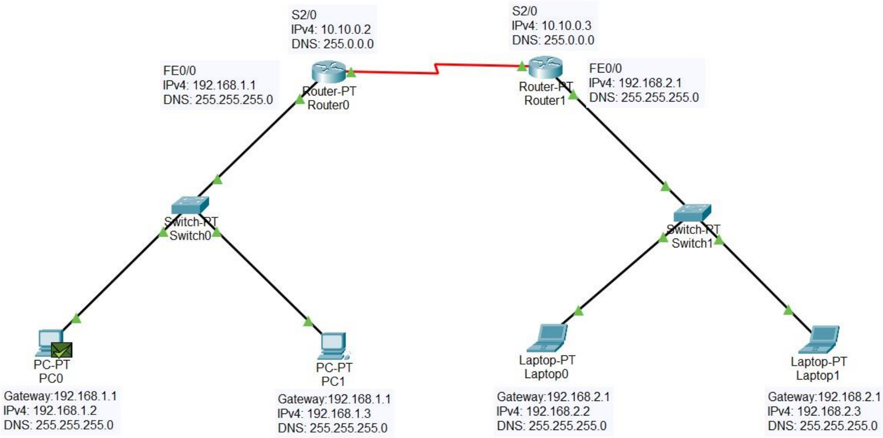
image page:

image:

26/12/24

LABORATORY PROGRAM – 9

To Configure RIP routing protocol in Routers.



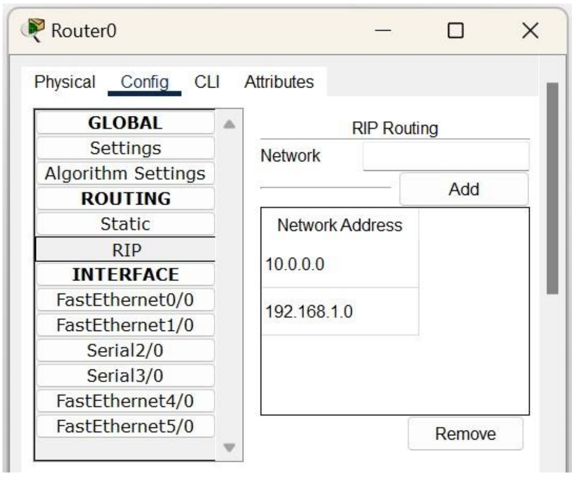


Figure 9.1: RIP, Router0

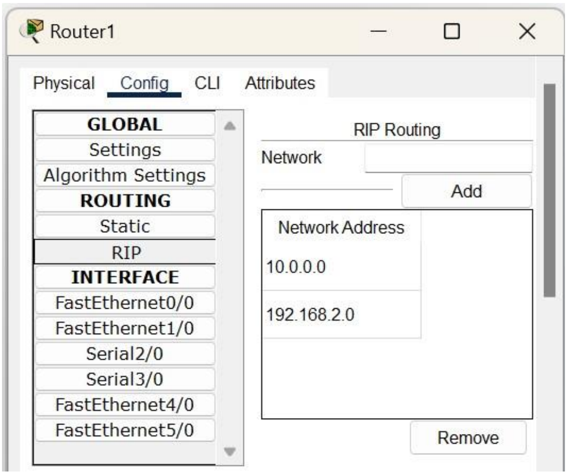


Figure 9.2: RIP, Router

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop1	ICMP		0.000	N	0	(edit)	

```
C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=18ms TTL=126
Reply from 192.168.2.3: bytes=32 time=14ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 18ms, Average = 8ms
```

20/11/24.

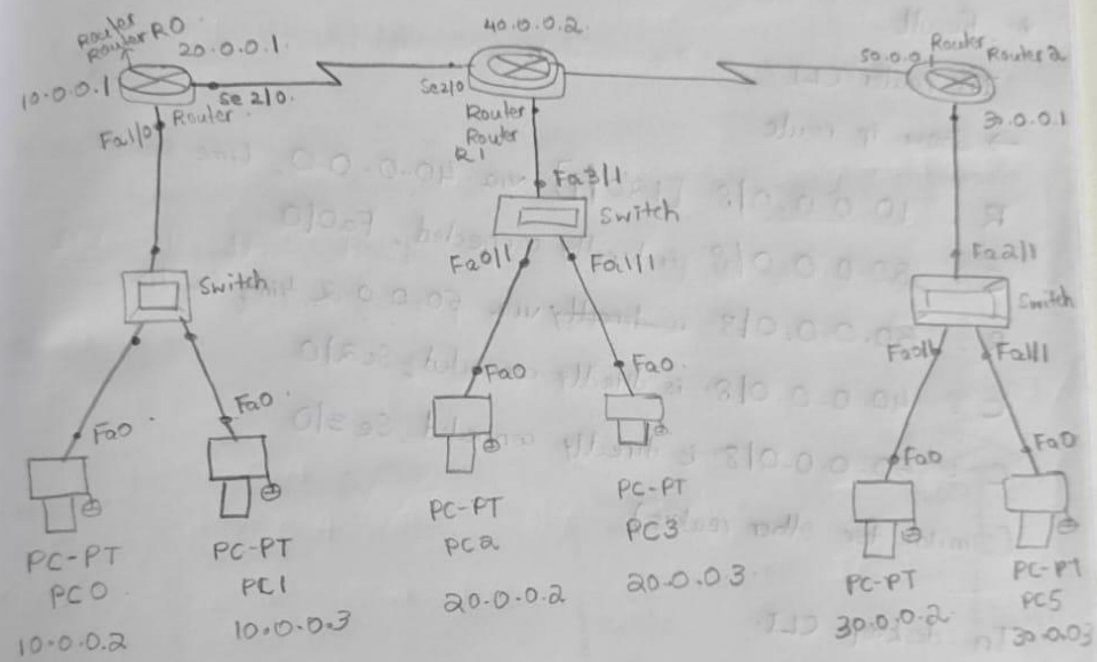
Lab-5

(RIP)

* Configure routing information protocol in Routers:

Aim: To show that RIP helps with inter-network communication using multiple routers interconnected to each other.

Topology:



* Procedure:

1) Open Cisco packet tracer
2) Drag & drop all routers, switches and end devices.
3) Connect them to each other.
4) Set IP & gateway for the devices.

5) Setup router → CLI → Enable → config terminal → interface serial/ Fastethernet → IP & subnet → no shut → exit.

6). Now in router CLI → config terminal → router rip → Network IP (eg 20.0.0.0) → exit (for all routers)

7). Ping from PC0 to PC2 & PC4.

* Result:

In router CLI

→ Show ip route.

R : 10.0.0.0/8 [120/1] via 40.0.0.0, time, Se2/0

C : 20.0.0.0/8 indiredly connected, Fa0/0.

R : 30.0.0.0/8 ~~is directly~~ via 50.0.0.2, time, Se3/0.

C : 40.0.0.0/8 is diredly connected, Se2/0.

C : 50.0.0.0/8 is directly connected, Se 3/0

(Similar for other routers).

In desktop CLI.

From PC1.

Ping 20.0.0.2

Request Timed out

Reply from 20.0.0.2 ; byte 32, time 2lms, TTL-125

" "

Ping statistics (Similar for others).

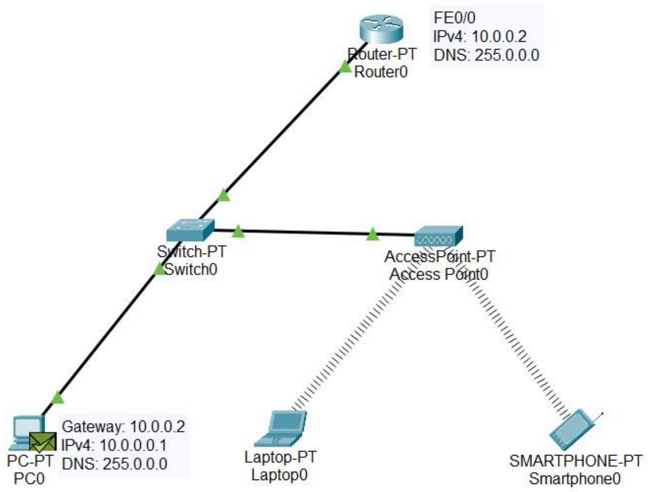
Observation:

We observe that using Router information protocol, we can connect multiple networks which can enable communication b/w system is different networks. Router talk & share their routing tables with each other which enables the connection

26/12

LABORATORY PROGRAM – 10

To demonstrate communication between two devices using a wireless LAN.



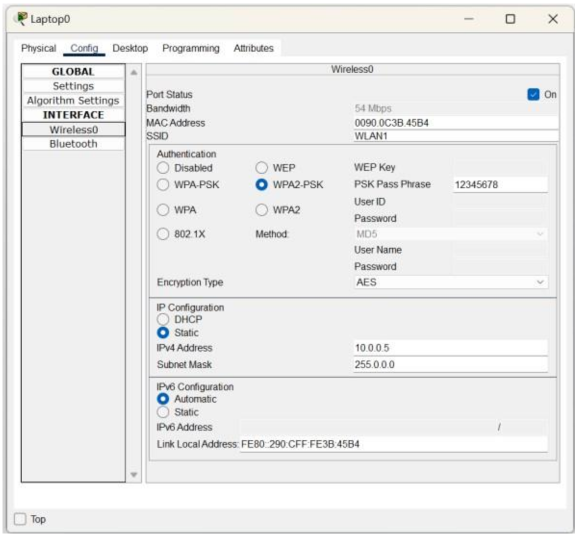


Figure 10.1: Laptop0, Wireless0

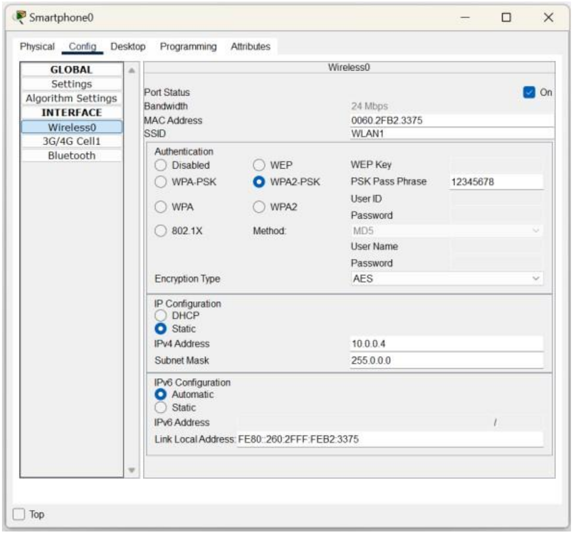
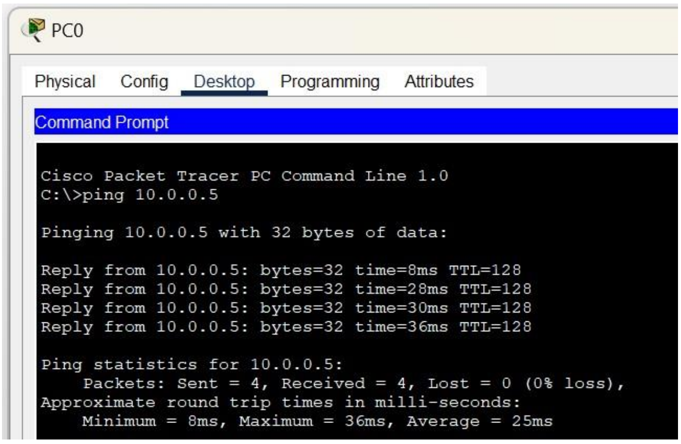


Figure 10.2: Smartphone0, Wireless0

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Laptop0	ICMP		0.000	N	0	(edit)	



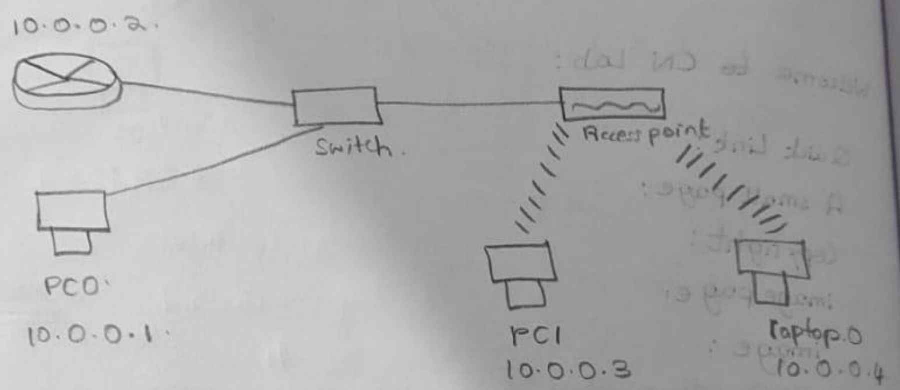
18|12|24.

Exp-12

Q). To construct a WLAN & make the nodes communicate wirelessly.

Aim: To show WLAN is effective for wireless communications.

Topology:



Procedure:

1) Open cisco packet Tracer.
2). Construct the above topology.
3). Configure access point, Port 1 → SSID name - any nam
SSID → code select WEP = 10digits (0123456789).
4). Configure PC1 & laptop with wireless standards.
5). Switch off the device, drag the existing PT-host NM-1AM to the component listed in LHS. • Drag the WMP 300 wireless interface to the empty ~~slot~~ port. Switch on the device
6). In config tab a new wireless interface would have been added. Now configure SSID → code, select WEP → 10 dig WEP Key, IP address & gateway to the device.

7) Ping from either devices.
8) Setup PC0, router as normally done).

Result:

PC0:
ping 10.0.0.3.
pinging 10.0.0.3 with 32 bytes of data
Reply from 10.0.0.3, bytes = 32 time = 19ms TTL = 128.
:
:
:

Ping statictics:

Same for other device:

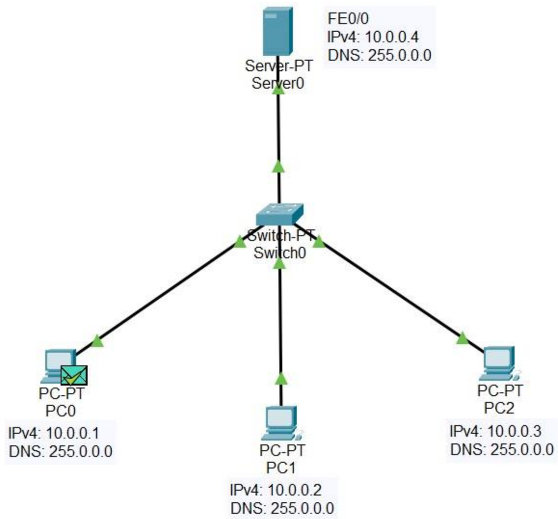
Observation:

The experiment demonstrates the creation of a wireless network using an access point configured with an SSID, WEP encryption & a 10 digit key. Devices like PC's and laptops were configured with wireless adapters, IP addresses & gateways to enable communication. - The success of ping tests b/w devices verify the setup, highlighting the simplicity & efficiency of WLAN connections for wireless communications.

26/12/24

LABORATORY PROGRAM – 11

To demonstrate the working of Address Resolution Protocol (ARP) within a LAN for communication.



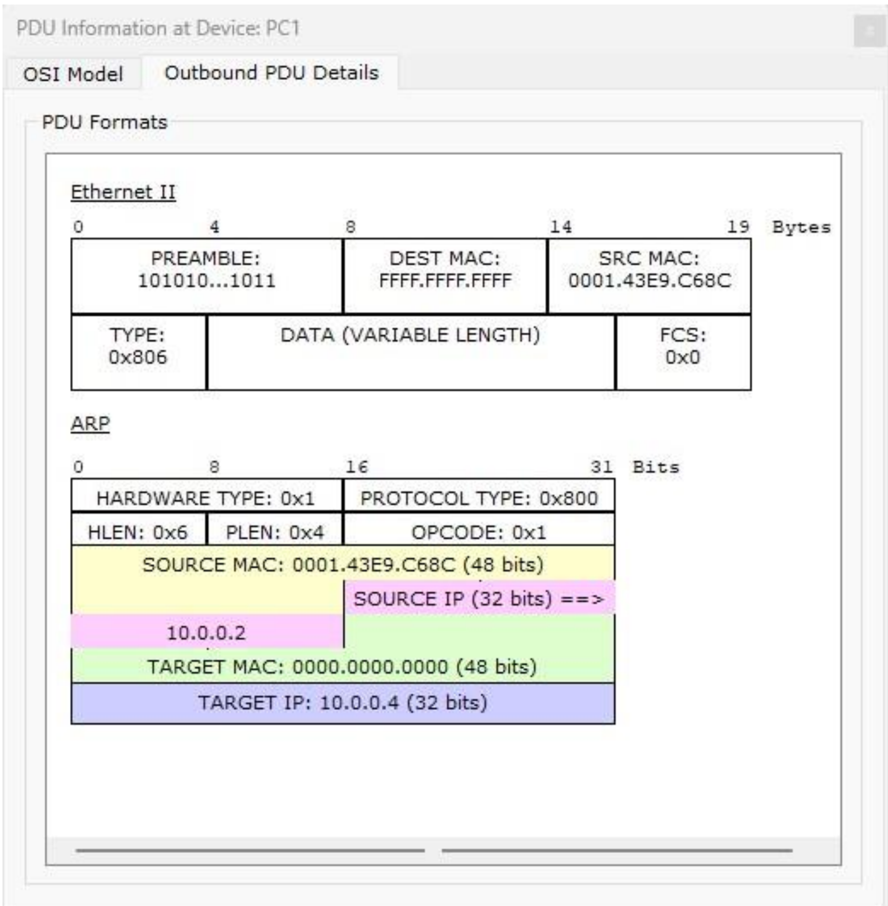


Figure 11.1: Inbound ARP, PC1

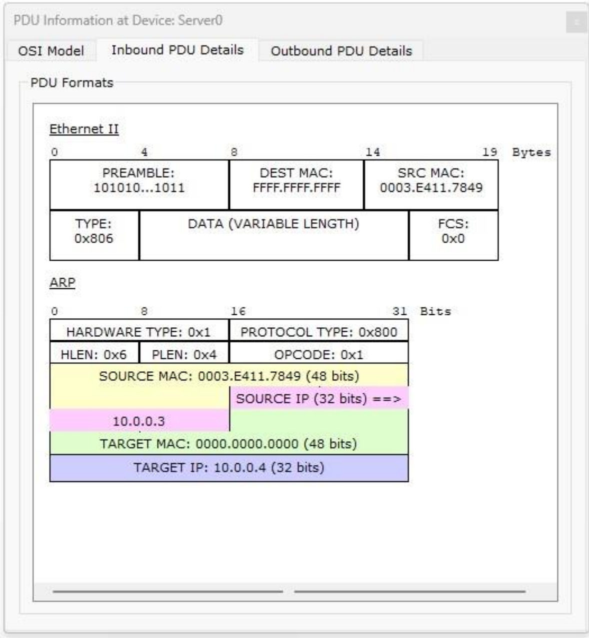


Figure 11.2: Inbound ARP, Server0

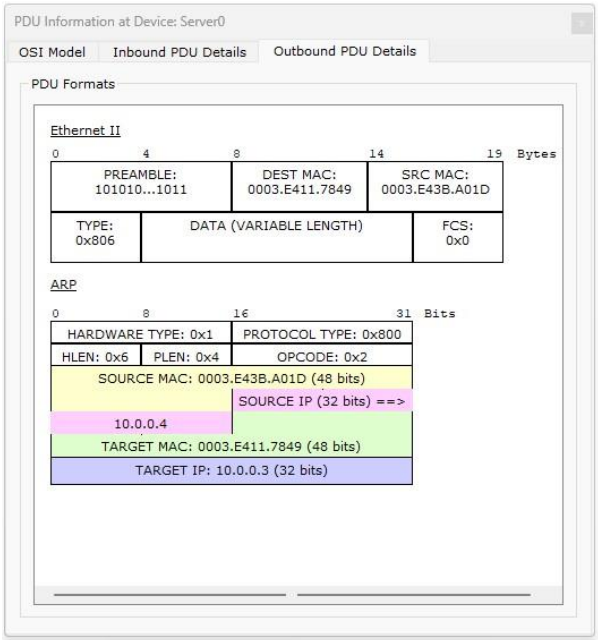


Figure 11.3: Outbound ARP, Server0

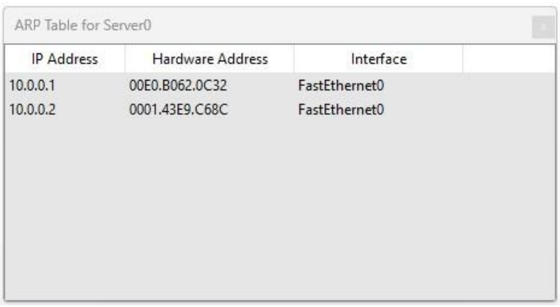


Figure 11.4: ARP Table, Server0

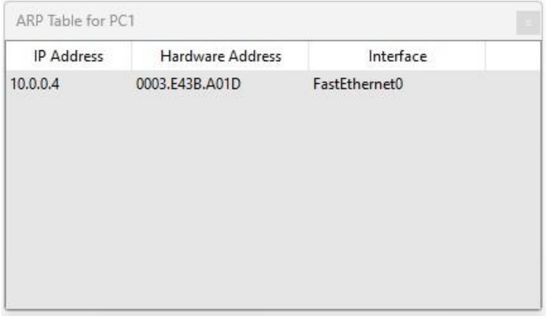
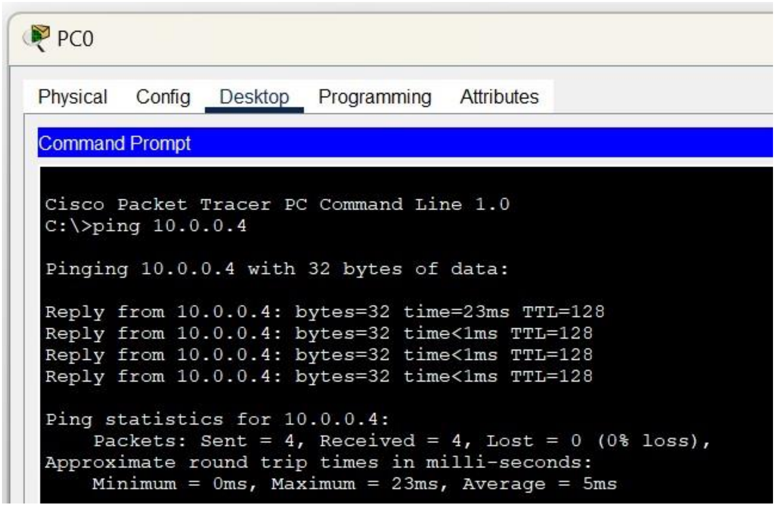


Figure 11.5: ARP Table, PC1

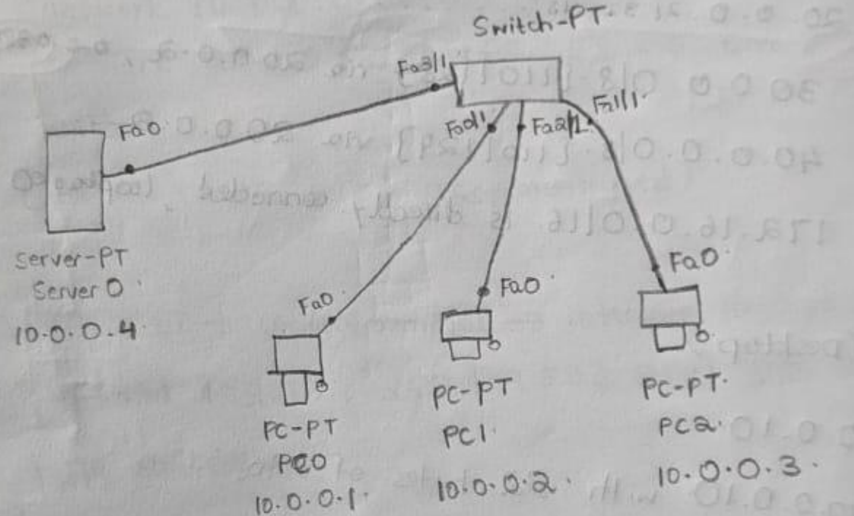


Lab - ~~8~~ 9

18/12/24

Q) To ~~contribute~~ construct simple LAN & understand the concept and operation of Address resolution protocol:

Topology:



Procedure:

1). Open Cisco packet tracer.
2). Set up devices as shown in figure.
3). Configure the IP address for the devices as shown.
4). Switch to simulation mode.
5). Direct a simple PDU from a destination to source.
6). Start simulation and observe.
7). Take inspect tool & open ARP table for all devices.

Result:

In switch CLI.

→ Show mac address-Table.

VLAN	MAC Address	Type	Ports
1	0001.42b5.6e08.	Dynamic	Fa3/1
1	0001.43ee.2ad3.	Dynamic	Fa2/1
1	0002.4a35.d3c2	Dynamic	Fa1/1
1	00d0.d35b.384b.	Dynamic.	Fa0/1.

Initially. ARP tables empty:

After simulation, begins the ARP. Tables of source & destination change.

PC1. Source

IP.	Hardware.	Interface.
10.0.0.3.	0001.43ee.2AD3	FA().

PC2: Destination

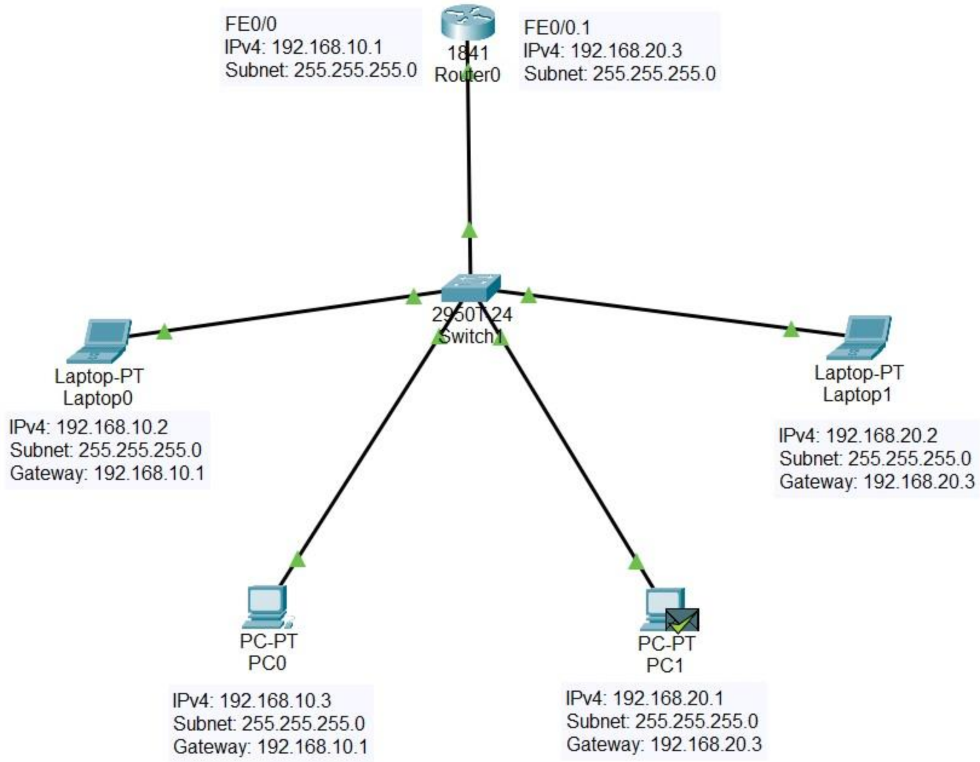
IP.	Hardware.	Interface.
10.0.0.2	0002.4A35.D3C2	FA().

Observation:

Initially, the ARP tables of all the devices are empty because no communication has occured, and no Mac-IP mapping is cached. When one device attempts to communicate with another, it sends an ARP request to determine the MAC address corresponding to the IP address of the targeted device. The targeted device responds ~~to IP address~~ with ARP reply, updating ARP tables on both ends. The switch builds its MAC address table by mapping MAC address to ports based on receiving frames.

LABORATORY PROGRAM – 12

To create a VLAN on top of the physical LAN and enable communication between physical LAN and virtual LAN.



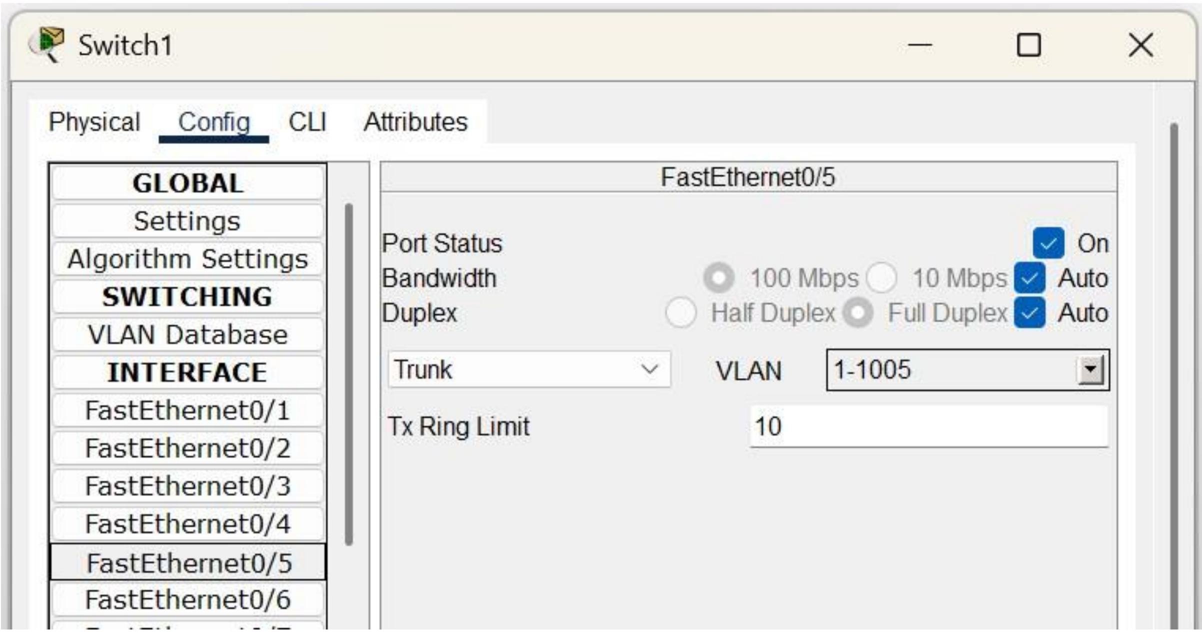


Figure 12.1: FE0/5 Switchport Trunk

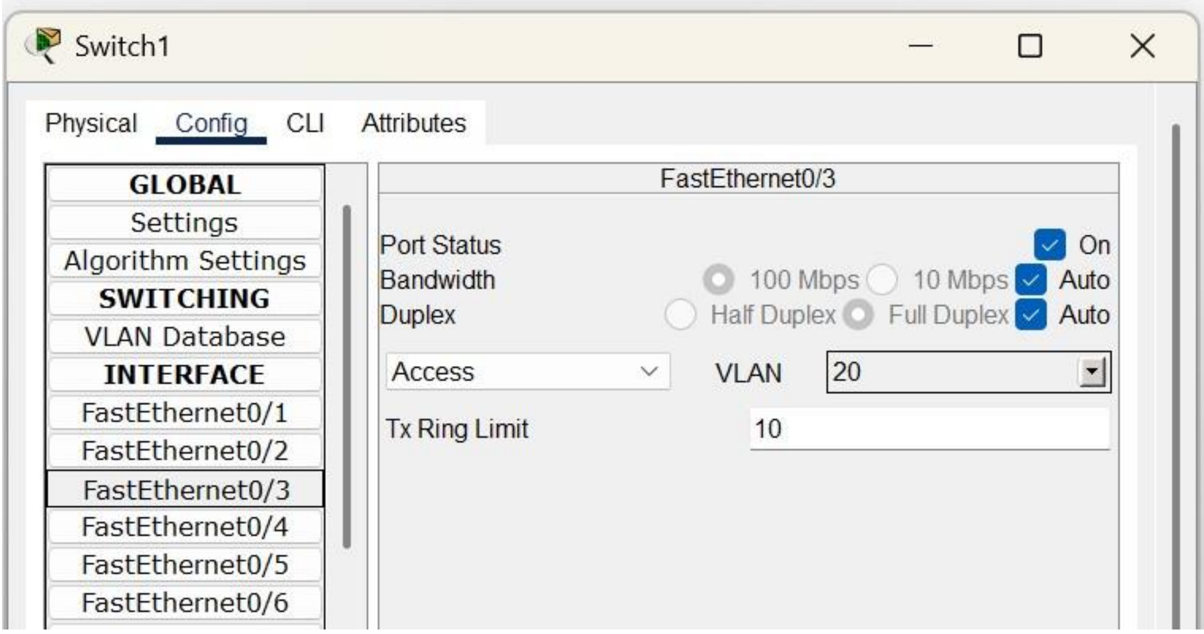


Figure 12.2: FE0/3 Switchport Access

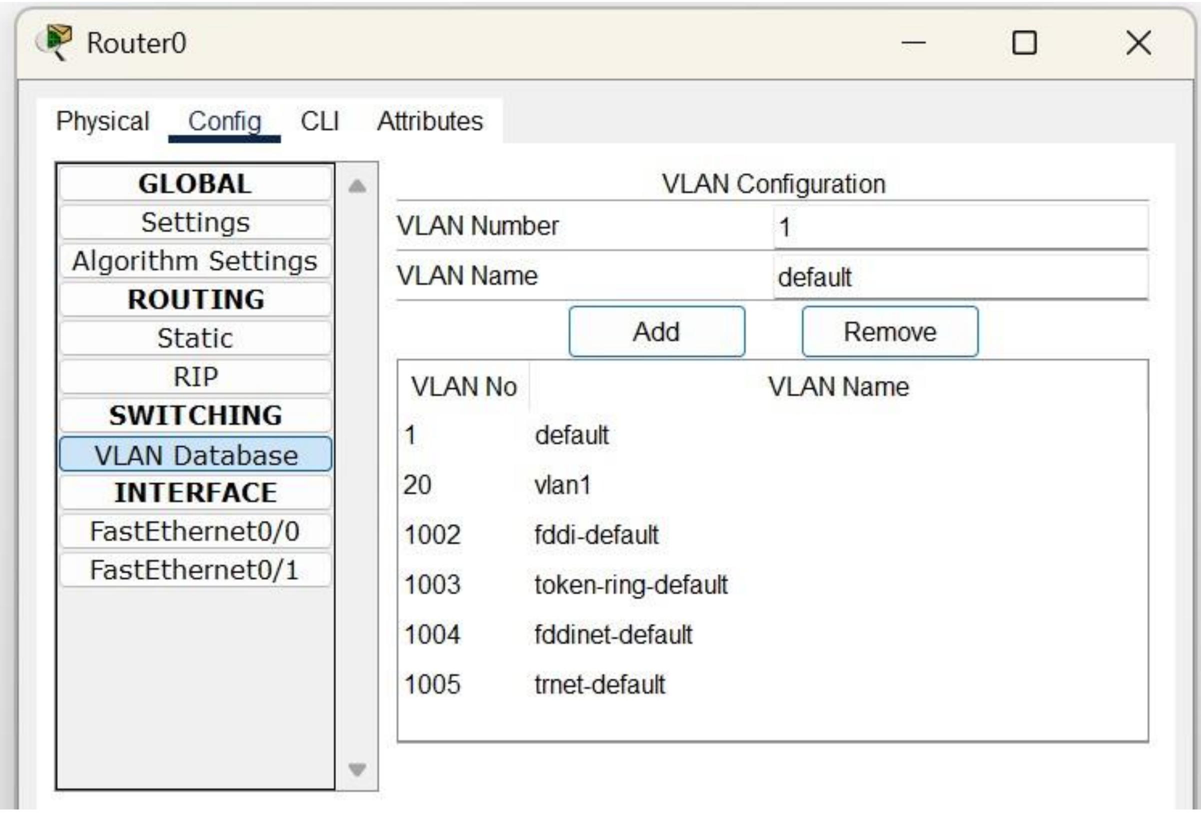


Figure 12.3: Router0 VLAN Database

```
Router(config)#interface FastEthernet0/0.1
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.3 255.255.255.0
Router(config-subif)#no shutdown
```

Figure 2: Router0, FE0/0.1

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	Router0	ICMP		0.000	N	0	(edit)	

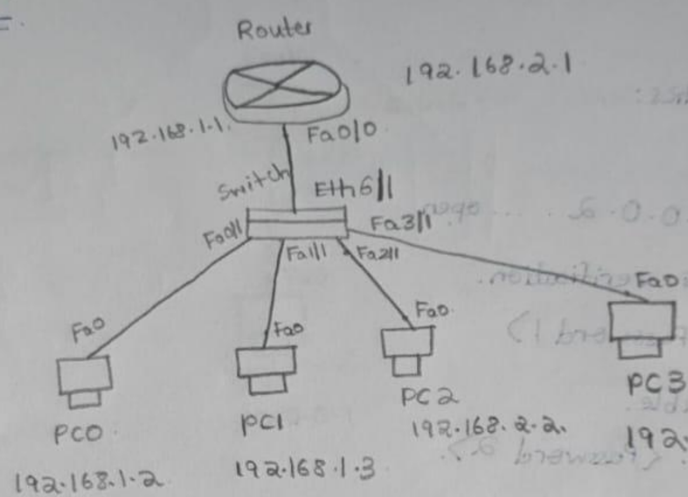
```
C:\>ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data:

Reply from 192.168.20.3: bytes=32 time=2ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.20.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

18|12|24

Lab-11

Q). To construct a VLAN & make PC's communicate among a VLAN.

Topology:



Procedure:

1). Open Cisco packet Tracer.

2). Setup the connections b/w. the devices as shown. (chose 184

3). Set IP & gateway address to all PC's

4). Setup router for one gateway

enable → config terminal → interface Fa0/0 → IP addre

192.168.1.1 255.255.255.0 → no shut → exit

5). Goto: switch config → select VLAN database.

create a new VLAN (VLAN no 2, VLAN name: Diddy.)

add.

6) Go to interface ethernet 6/1.

7). VLAN database → VLAN 2, name: Diddy → exit.

8). Config terminal → interface Fa0/0.1 → encapsulation dot1q 2 → ip address 192.168.2.1 255.255.255.0 → no shut → exit → exit.

9). Enter show IP route.

10). Ping from one device to another.

Result:

Show IP route:

C: 192.168.1.0/24 is directly connected Fa0/0.

C: 192.168.2.0/24 . . . . Fa0/0.1.

PC0.

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 = 2ms TTL = 127.

:

Ping stats.

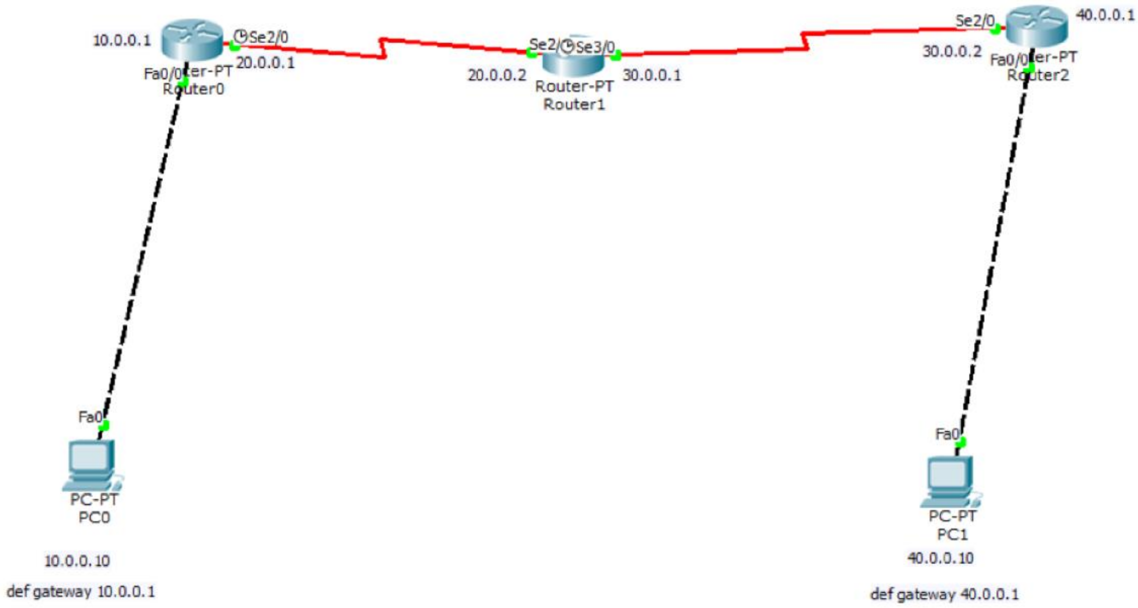
Observation:

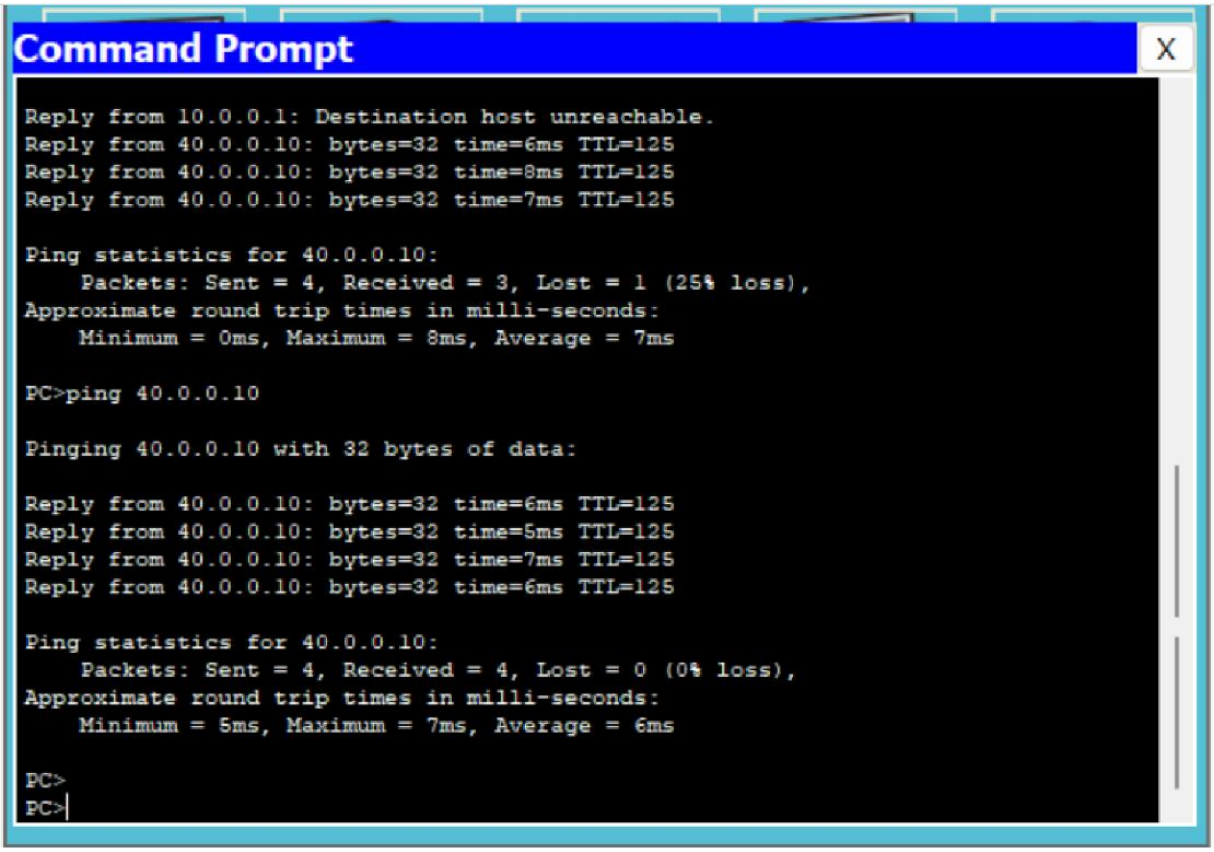
The VLAN experiment involves ~~creativity~~ creating & configuring VLAN to segment a network, assigning IP's to device for seamle[ss] intra VLAN communication & using dot1q encapsulation for int[er] VLAN connectivity to communicate through a single trunk link. This experiment highlights the importance of VLAN's in optimizing & managing modern network effectively.

26/12/24

LABORATORY 10

Configure OSPF routing protocol





IOS Command Line Interface

```
Router>
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:00:11, Serial2/0
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:21:19, Serial2/0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial2/0
C       30.0.0.1/32 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
C    172.16.0.0/16 is directly connected, Loopback0
Router>
00:51:41: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FU
LL, Loading Done
```

Copy Paste

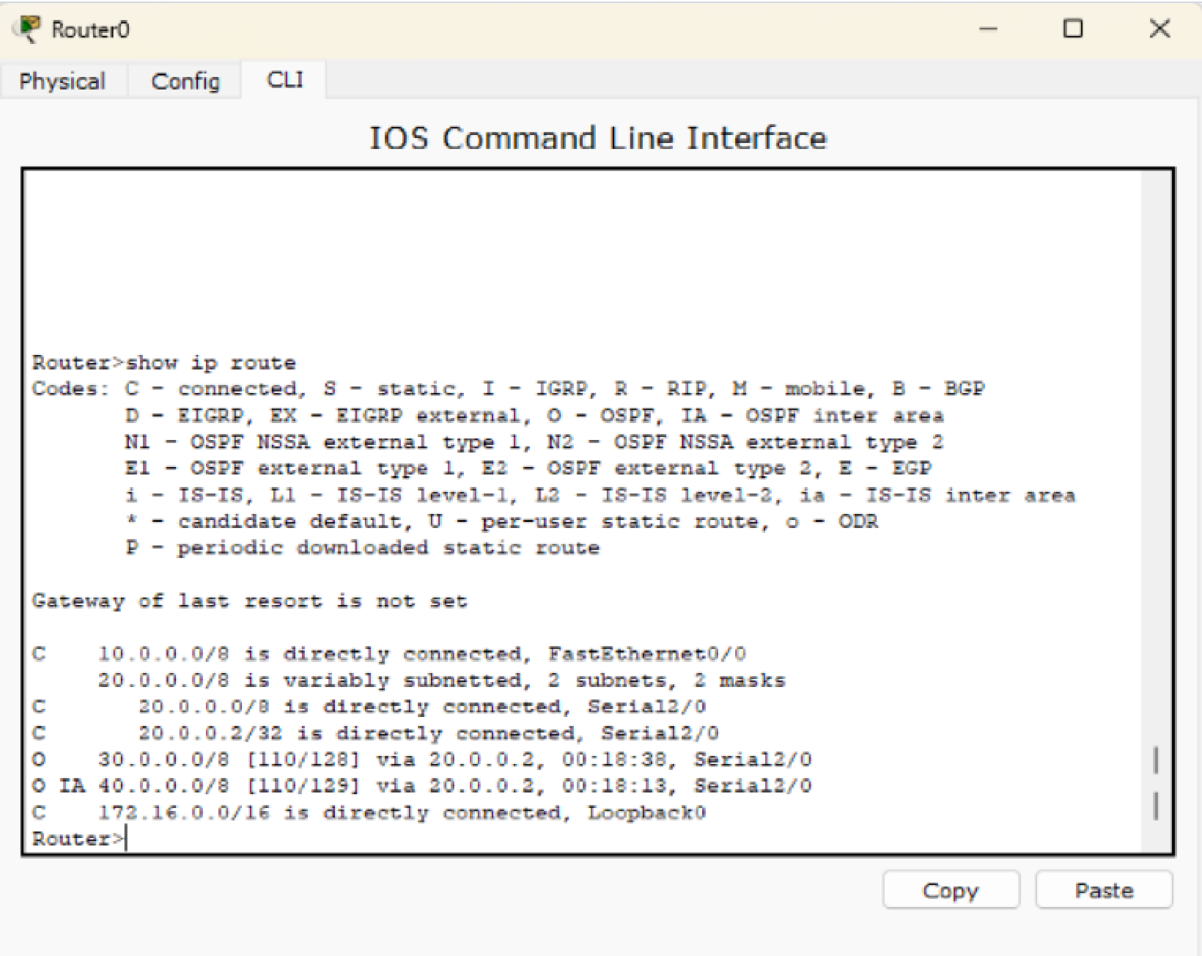
Router1

Physical Config CLI

IOS Command Line Interface

```
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.0.0.0/8 is directly connected, Serial2/0
C       20.0.0.1/32 is directly connected, Serial2/0
     30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial3/0
C       30.0.0.2/32 is directly connected, Serial3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:02:33, Serial3/0
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
```

Copy Paste

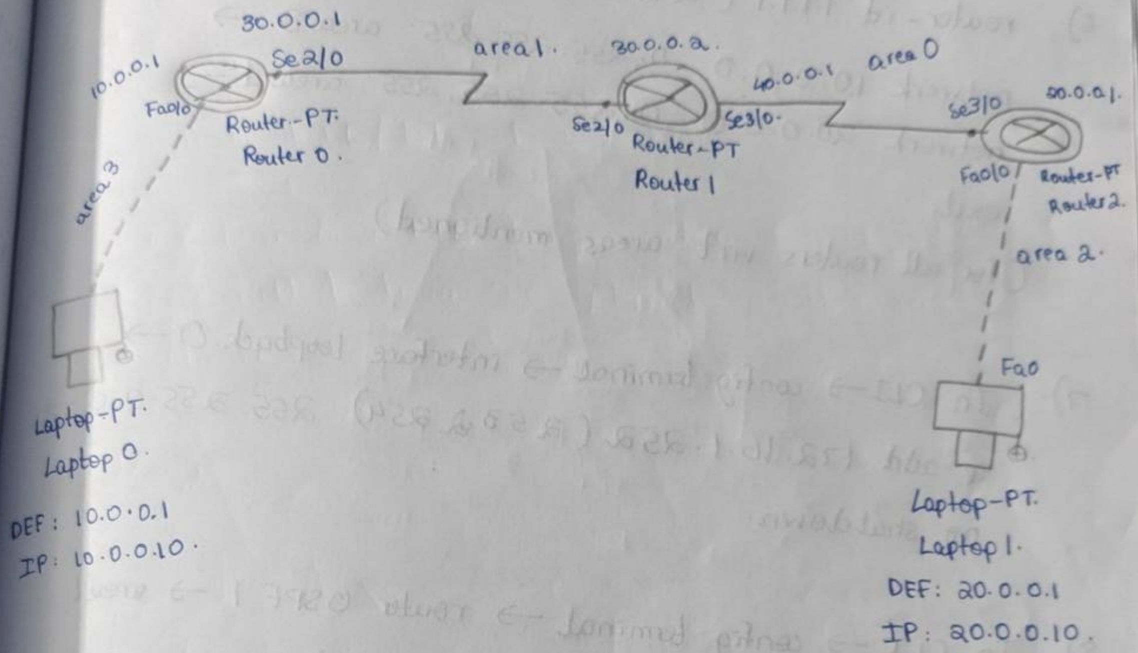

Router0
Physical
Config
CLI
IOS Command Line Interface
Router>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C   10.0.0.0/8 is directly connected, FastEthernet0/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   20.0.0.0/8 is directly connected, Serial2/0
C   20.0.0.2/32 is directly connected, Serial2/0
O   30.0.0.0/8 [110/128] via 20.0.0.2, 00:18:38, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:18:13, Serial2/0
C   172.16.0.0/16 is directly connected, Loopback0
Router>
Copy
Paste

27/11/24.

Lab-7

Q). Configure OSPF routing protocol:

Topology:



* Procedure:

1). Open Cisco packet tracer.
2) Drag and drop 3 routers & 2 systems & connect accordingly to the diagram.
3). Input IP & gateway to both systems.
4) Initialize router connections in CLI → No → enable → config terminal → interface Fa0/0 → IP address subnet → No shut → exit.

5). While doing for router to router → config terminal → interface Se2/0 → IP address subnet → encapsulation ppp → clock rate 64000. → no shutdown.

only for connection with clock system.

6) router-id 1.1.1.1 (2 & 3 for router 2, 3). →
network 10.0.0.0 0.255.255.255 area 3 →.
network 20.0.0.0 0.255.255.255 area 1 →
exit

(for all routers wrt areas mentioned).

7). In CLI → config terminal → interface loopback 0 →.
ip add 172.16.1.252 (253 & 254) 255.255.0.0 →
no shutdown.

8). In CLI → config terminal → router OSPF 1 → area 1.
virtual link 2.2.2.2 → exit (R1).

(R2).
area 1 virtual-link 1.1.1.1 → exit.

9). show ip route.

10). Ping from one system to another

Result:

Show IP route

R1.

C : 10.0.0.0/8 is directly connected, Fa0/0.

20.0.0.0/8 is variately subnetted 2 subnets, 2 masks.

C. 20.0.0.0/8 is directly connected Se2/0.

C : 20.0.0.2/32 is . . . .

O : 30.0.0.0/8 [110/128] via 20.0.0.2, 0.0.08.05 Se

O : 40.0.0.0/8 [110/129] via 20.0.0.2 . . . .

C : 172.16.0.0/16 is directly connected, loopback0.

In CMD (Desktop):

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 = 5 ms TTL

"

"

26/12/24

Ping stats

Observation: We observe how OSPF is configured & operat a network. Key observations include establishments of neighb relationship b/w routers, exchange of link-state ads & form routing table based on shortest path determined using dijikstr It demonstrates the dynamic nature of OSPF, allowing r to adapt to network changes & ensuring efficient & loop routing in IP network. Additionally we can monitor, how area, authentication & metric calculation influence the routin

LABORATORY PROGRAM – 13

Write a program for error detecting code using CRC-CCITT (8-bits).

Code

```
def xor(dividend, divisor):
    """Perform XOR operation between dividend and divisor."""
result = ''     for i in range(1, len(divisor)):         result += '0' if
dividend[i] == divisor[i] else '1'     return result

def crc(data, gen_poly):
    """Compute the CRC check value using CRC-CCITT (8-bit)."""
data_length = len(data)
    gen_length = len(gen_poly)

    # Append n-1 zeros to the data
padded_data = data + '0' * (gen_length - 1)
    check_value = padded_data[:gen_length]

    for i in range(data_length):
if check_value[0] == '1':
            # XOR operation if the first bit is 1
check_value = xor(check_value, gen_poly)         else:
            # Retain original check value if first bit is 0
check_value = check_value[1:]

        # Shift left and add the next data bit         if i +
gen_length < len(padded_data):             check_value
+= padded_data[i + gen_length]

    return check_value[1:]  # Remove the leading bit

def receiver(data, gen_poly):
    """Simulate the receiver side to check for errors."""     print("\n---------------------------
-")
    print("Data received:", data)

    # Perform CRC computation on received data
remainder = crc(data, gen_poly)

    # Check if the remainder is all zeros
if '1' in remainder:
        print("Error detected")
else:
```

```
        print("No error detected")

if __name__ == "__main__":
    # Input data and generator polynomial     data =
input("Enter data to be transmitted: ")     gen_poly =
input("Enter the Generating polynomial: ")

    # Compute CRC check value     check_value = crc(data, gen_poly)
print("\n----------------------------------------")     print("Data padded
with n-1 zeros:", data + '0' * (len(gen_poly) - 1))     print("CRC or
Check value is:", check_value)

    # Append check value to data for transmission
transmitted_data = data + check_value     print("Final
data to be sent:", transmitted_data)     print("----------
------------------------------\n")

    # Simulate the receiver side
    received_data = input("Enter the received data: ")
receiver(received_data, gen_poly)
```

Output

```
Enter data to be transmitted: 1001100
Enter the Generating polynomial: 100001011

----------------------------------------
Data padded with n-1 zeros: 100110000000000
CRC or Check value is: 0100010
Final data to be sent: 10011000100010
----------------------------------------

Enter the received data: 10011000100011

-----------------------------
Data received: 10011000100011
Error detected
```

Cycle 2

* Experiment 13 :

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

Code:

```
def crc_circuit(data: bytes, polynomial: int = 0x1021,
                init-crc: int = 0xFFFF) → int:

    crc : init_crc
    for byte in data:
        crc ^= (byte << 8).
        for_ in range(8):
            if crc & 0X8000:
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1.
        crc &= 0XFFFF

    return crc.

def encode_data_with_crc(data: bytes) → bytes:

    crc = crc_ccitt(data)
    crc_bytes = crc.to_bytes(2, byteordet = 'big').

    return data + crc bytes.
```

```
def verify_data_with_crc(data_with_crc: bytes) -> bool:
    data, received_crc = data_with_crc[:-2], data_with_crc[-2:]
    computed_crc = crc_ccitt(data)
    return computed_crc == int.from_bytes(received_crc,
                                          byteorder='big')

def main():
    message = "Hello World"
    data = message.encode('utf-8')
    computed_crc = crc_ccitt(data)
    data_with_crc = encode_data_with_crc(data)
    print(f"Data: {message}")
    print(f"Computed CRC_CCITT: 0x{computed_crc:04x}")
    is_valid = verify_data_with_crc(data_with_crc)
    if is_void:
        print("Data received correctly with no errors")
    else:
        print("Data received with errors")

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

O/P:

```
Data: "Hello World"
Computed CRC_CCITT: 0x882A
Data Received correctly with no errors.
```

N
3/1/25

LABORATORY PROGRAM – 14

Write a program for congestion control using Leaky bucket algorithm.

Code

```
# Getting user inputs
storage = int(input("Enter initial packets in the bucket: ")) no_of_queries = int(input("Enter
total no. of times bucket content is checked: ")) bucket_size = int(input("Enter total no. of
packets that can be accommodated in the bucket:
"))
input_pkt_size = int(input("Enter no. of packets that enters the bucket at a time: "))
output_pkt_size = int(input("Enter no. of packets that exits the bucket at a time: "))

for i in range(no_of_queries):  # space left
size_left = bucket_size - storage     if
input_pkt_size <= size_left:
        # update storage         storage +=
input_pkt_size     else:         print("Packet
loss =", input_pkt_size)

    print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

    # as packets are sent out into the network, the size of the storage decreases
storage -= output_pkt_size
```

Output

```
Enter initial packets in the bucket: 0
Enter total no. of times bucket content is checked: 4
Enter total no. of packets that can be accommodated in the bucket: 10
Enter no. of packets that enters the bucket at a time: 4
Enter no. of packets that exits the bucket at a time: 1
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```

* Experiment 14:

Q). Write a program for congestion control using leaky bucket algorithm.

Code:

```
#include <stdio.h>
int main() {
    int incoming, outgoing, buck_size, n, store = 0;
    printf(" Enter bucketsize, outgoing, rate & n.o of i/p");
    scanf("%d %d %d", & buck_size, & outgoing, &n);
    while(n!=0) {
        printf(" Enter the incoming packet size: ");
        scanf("%d", &incoming);
        printf("Incoming packet size %d\n", incoming);
        if( incoming <= (buck_size -store)) {
            store += incoming;
            printf(" Bucket buffersize %d out of %d\n",
                   store, buck_size);
        } else {
            printf(" Dropped %d n.o of packets\n",
                   incoming - (buck_size -store));
            printf(" Bucket buffer size %d out of
                   %d\n", store, buck_size);

            store = bucksize;

    }
```

```
store = store - outgoing;
printf("After outgoing %d bytes left out of %d in buffer
       \n", store, buck_size);
n--;
}
}
```

O/P : Enter bucket-size, outgoing rate & n.o of i/p : 10

Enter incoming packet size : 5

Incoming packet size : 5

Bucket buffer size 5 out of 10

After outgoing 2 bytes left out of 10 in buffer

Enter the incoming packet size : 15

Incoming packet size : 5

Bucket buffer size 1 out of 10

After outgoing 4 bytes left out of 10 in buffer

Enter the incoming packet size : 7

Incoming packet size : 7

Dropped buffer 4 out of 10

After outgoing 7 bytes left out of 10 in buffer

N 3/1/25

LABORATORY PROGRAM – 15(A)

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

Code: Client.py

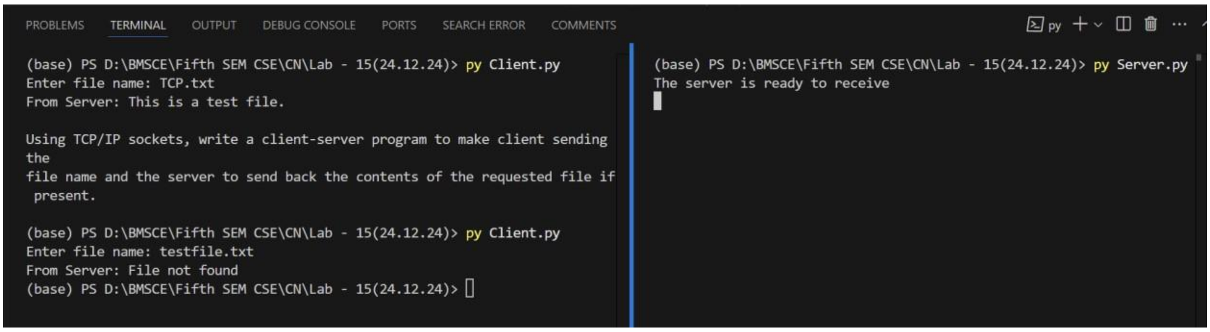
```
from socket import * serverName = "127.0.0.1"  # Server
address (localhost) serverPort = 12000  # Port number
where the server listens

# Create TCP socket
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))  # Connect to server

# Ask user for file name to request
sentence = input("Enter file name: ")

# Send file name to server
clientSocket.send(sentence.encode())

# Receive file contents from server filecontents
= clientSocket.recv(1024).decode()
print('From Server:', filecontents)

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

Code: Server.py

```
from socket import *
serverName = "127.0.0.1"  # Server address (localhost) serverPort
= 12000  # Port number to listen on

# Create TCP socket
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))  # Bind socket to the address and port
serverSocket.listen(1)  # Listen for 1 connection print("The server is ready to
receive")

while True:
   # Accept a connection
   connectionSocket, addr = serverSocket.accept()

   # Receive the file name from the client
```

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

    # Try opening the file
try:
        file = open(sentence, "r")  # Open file in read mode
        fileContents = file.read(1024)  # Read file content (up to 1024 bytes)
connectionSocket.send(fileContents.encode())  # Send file contents to client         file.close()
except FileNotFoundError:
        # Send error message if file not found
        connectionSocket.send("File not found".encode())

    # Close the connection
    connectionSocket.close()
```

Output



* Experiment 15:

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

Code:

client.py:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000 ↳ clientSocket = Socket (AF_INET, Sock_STREAM)
clientSocket . connect (( serverName, servePort)).
sentence = input ("Enter file name:")
clientSocket = send (Sentence. encode()).
file.contents = clientSocket . recv (1024) . decode().
print ("From server :", filecontents).
clientSocket . close().
```

server.py:

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

while 1 :
connection socket, addr = server Socket . accept()
sentence = connection Socket . recv (1024) . decode()
file = open (sentence, 'r')
data = file.read (1024)
connection socked . send (data.encode())

file.close ().
connection socket . close () .

O/P : server.py - The server ready to receive!
client.py - Enter file name : example.txt
From server : Hello, world!

N
3/1/25

LABORATORY PROGRAM – 15(B)

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: ClientUDP.py

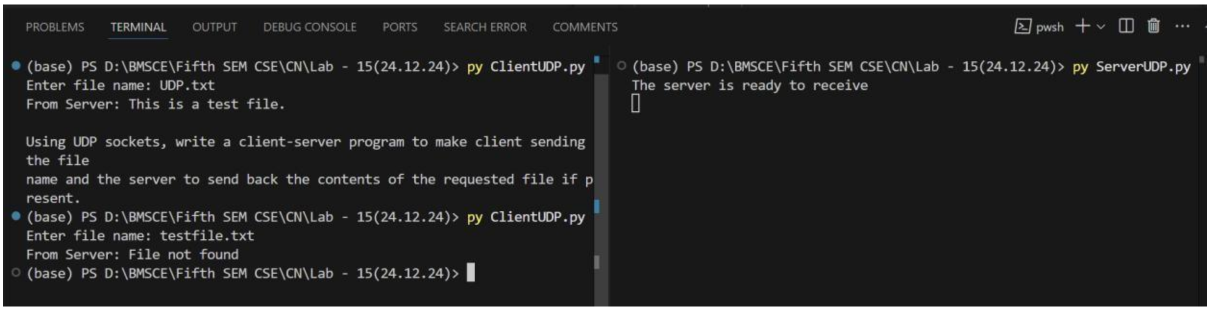
```
from socket import *
serverName = "127.0.0.1"  # Server address (localhost) serverPort
= 12000  # Port number where the server listens

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

# Ask user for file name to request
sentence = input("Enter file name: ")

# Send the file name to the server using UDP clientSocket.sendto(sentence.encode("utf-8"),
(serverName, serverPort))

# Receive file contents from the server
fileContents, serverAddress = clientSocket.recvfrom(2048)

# Print the file contents received from the server print("From
Server:", fileContents.decode())

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

Code: ServerUDP.py

```
from socket import *
serverPort = 12000  # Port number to listen on

# Create UDP socket
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))  # Bind the socket to the server address and port

print("The server is ready to receive")

while True:
   # Receive file name from the client
   sentence, clientAddress = serverSocket.recvfrom(2048)

   # Try opening the file
try:
      file = open(sentence.decode(), "r")  # Open file in read mode        fileContents =
file.read(2048)  # Read file content (up to 2048 bytes)
```

serverSocket.sendto(fileContents.encode("utf-8"), clientAddress) # Send file contents to client   file.close()   except FileNotFoundError:
        # Send error message if file not found
        serverSocket.sendto("File not found".encode("utf-8"), clientAddress)

Output



* Experiment 15:

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

Code:

Client.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000.
clientSocket = socket(AF_INET, SOCK_DGRAM).
Sentence = input("Enter file name : ").
clientSocket.sendto (bytes (sentence, "utf-8"), (204
print("From server :", filecontents ).
clientsocket.close().
```

Server.py

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

while 1:
    sentence, clientAddr = serverSocket.recvfrom(2048)
    file = open(sentence, "r").
    data = file.read(2048).

    serverSocket.sendto( bytes (data, "utf-8"), clientA
    print ("sent back to client ", data)
file.close().
```

server.py — The server is ready to receive
sent back to client : Hello world!

client.py — Enter file name: "Example.txt
From server: Hello, world!

3/1/25