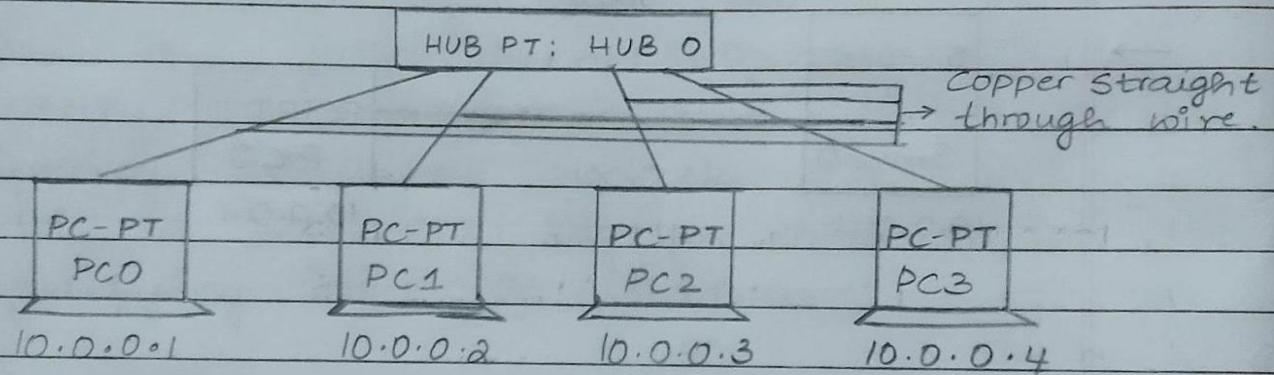


## LAB-1

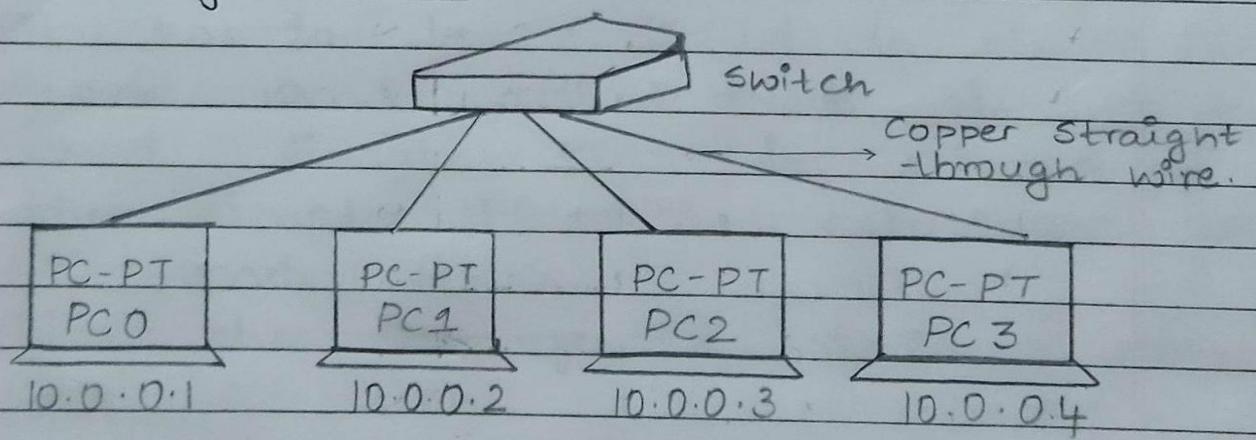
Aim :- Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices

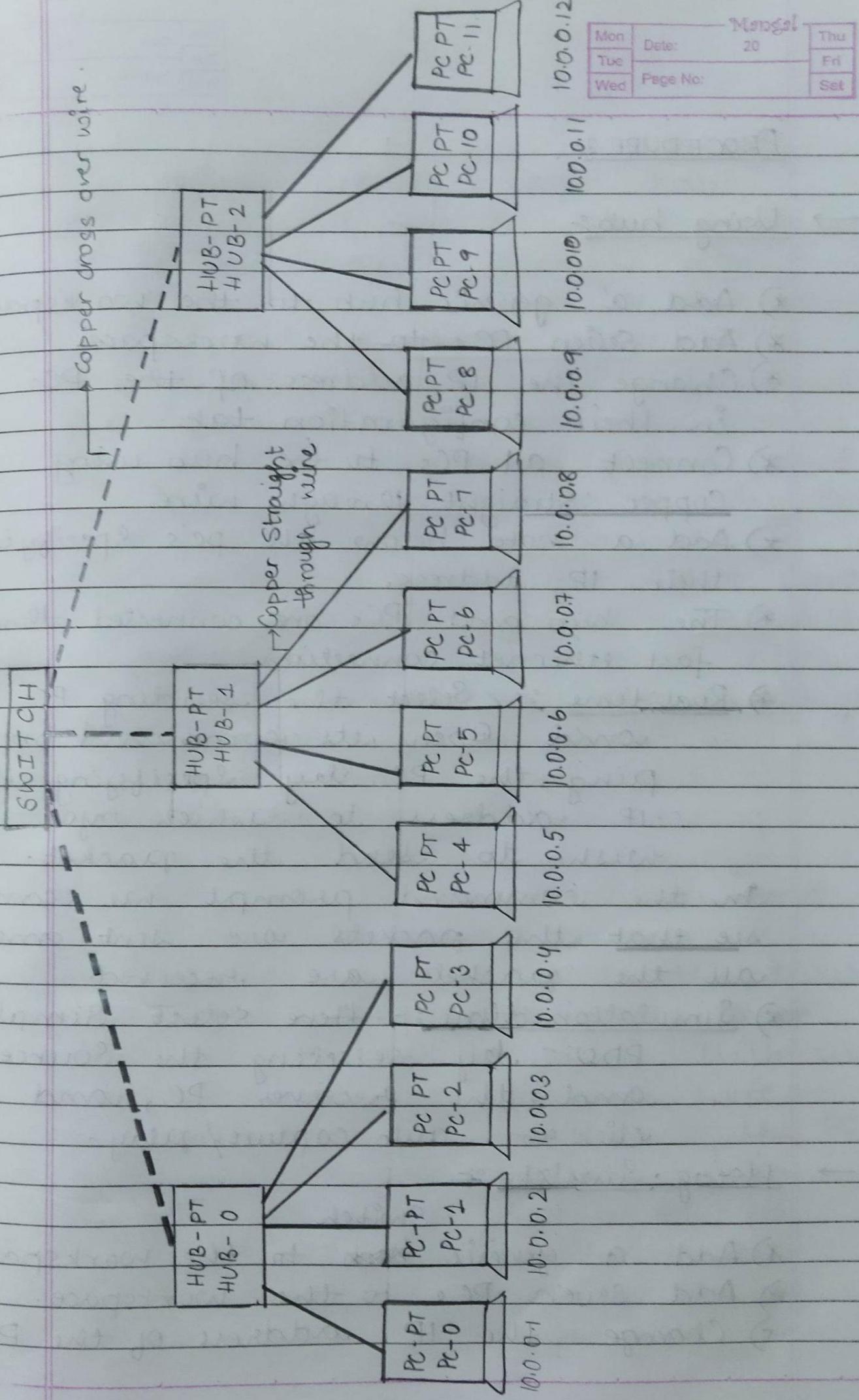
### TOPOLOGY :-

#### \* Using Hub :-



#### \* using Switch :-





## PROCEDURE :-

### → Using hub :-

- \* Add a generic hub to the workspace.
- \* Add Seven PCs to the workspace.
- \* Change the IP address of the PCs in their configuration tab.
- \* Connect all PCs to the hub using Copper Straight through wire
- \* Add a note below all pc's specifying their IP address.
- \* The hub and PCs are connected through fast ethernet connection
- \* Real time :- Select the sending PC and Open its command prompt by specifying the PC by specifying its IP address to which you wish to send the packet.  
In the command prompt we can see that the packets are sent and all the packets are received.
- \* Simulation time :- Here select Simple PDU by selecting the Source and the receiver PC, and click on auto capture/ play -

### → Using Switch :-

Switch

- \* Add a generic ~~hub~~ to the workspace
- \* Add Seven PCs to the workspace.
- \* Change the IP address of the PCs

in their configuration tab

- \* ) Connect all PCs to the hub using copper Straight through wire.
- \* ) Add a note below all PC specifying their IP address.
- \* ) Real time :- Select the PC from which you desire to send the packet. Open its command prompt and Ping the destination PC by specifying its IP address.
- \* ) Simulation time :- Here select simple PDU and the pair of PC between which you desire and click on auto capture / play from the right panel.

→ Hybrid mode :-

- \* ) Add a switch, 3 hubs and 12 PCs to the workspace.
- \* ) Connect all 3 hubs to the switch using cross over wire ~~cross~~.
- \* ) Connect 4 PCs to each hub using cross straight through wire.
- \* ) Update the IP addresses of the PCs from their configuration tab.
- \* ) Add the note below the PCs specifying their IP addresses.
- \* ) Real time mode :- Select the PC you want to send the packet from, and open its

command prompt, specify the destination PC by specifying the IP address, few packets are sent all of which are received by the receiving PC.

\* Simulation mode:- Add a Simple PDU by selecting the pair of PC and click on autocapture from right most panel.

### OBSERVATION :-

#### HUB :-

#### Learning Outcome :-

- When a source node sends a packet in the network, the hub receives the packet and sends broadcast over the network, i.e., it sends data to all the end devices in the network and the node whose IP matches with the specified address accepts the packet and acknowledges it, rest node just ignores the message.
- The connection between the hub and the end devices is established using ~~crossover wire~~ Copper straight through wire as they belong to different layers of networking.

Mon	Date:	Mangal 20	Thu
Tue			Fri
Wed	Page No:		Sat

### Result :-

PC > ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: byte=32, time=0ms

Reply from 10.0.0.5: byte=32, time=0ms

Reply from 10.0.0.5: byte=32, time=0ms

Reply from 10.0.0.5: byte=32, time=0ms.

Ping Statistics for 10.0.0.5:

Packets: Sent = 4, received = 4, loss = 0.

### SWITCH :- Learning Outcome :-

→ When a Source ~~node~~ device sends the message to the Switch once the connection is established, which takes some time known as learning time.

After the Switch gets the packet, it initially broadcasts the packet to all end <sup>devices</sup> ~~users~~ to locate the destination. Once the destination is located then the message is sent only to it further.

→ The connection between the switch and ~~end devices~~ - end devices is established using copper straight wire as they belong to different network layers.

### Result :-

PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=1ms

Reply from 10.0.0.1: bytes=32 time=3ms

Reply from 10.0.0.1: bytes=32 time=0ms

Reply from 10.0.0.1: bytes=32 time=0ms

Mon	Date:	M. M. Gal
Tue		20
Wed	Page No.:	

Ping statistics for 10.0.0.1

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

### HYBRID NETWORK :-

#### Learning Outcomes :-

- The switch and hub are connected through copper cross over as they belong to the same layer but hub and PCs are connected through copper straight through as they belong to different network layer.
- The message from the source PC is sent to the switch that sends the message to the hub having destination PC, but the hub broadcasts the message to all its end devices and only the referred / destination PC sends back the acknowledgement.

#### Results :-

Ping 10.0.0.7

Pinging 10.0.0.7 with 32 bytes of data:

Reply from 10.0.0.7: bytes=32 time=1ms TTL=128

Reply from 10.0.0.7: bytes=32 time=0ms TTL=128

Reply from 10.0.0.7: bytes=32 time=0ms TTL=128

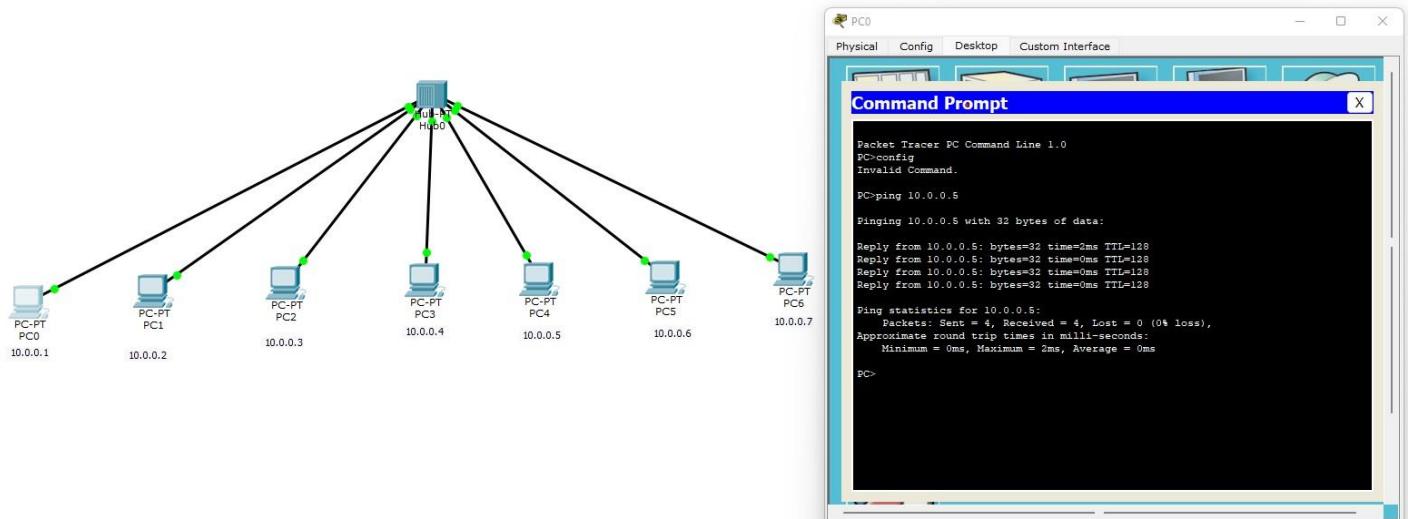
Reply from 10.0.0.7: bytes=32 time=0ms TTL=128

Ping Statistics for 10.0.0.7:

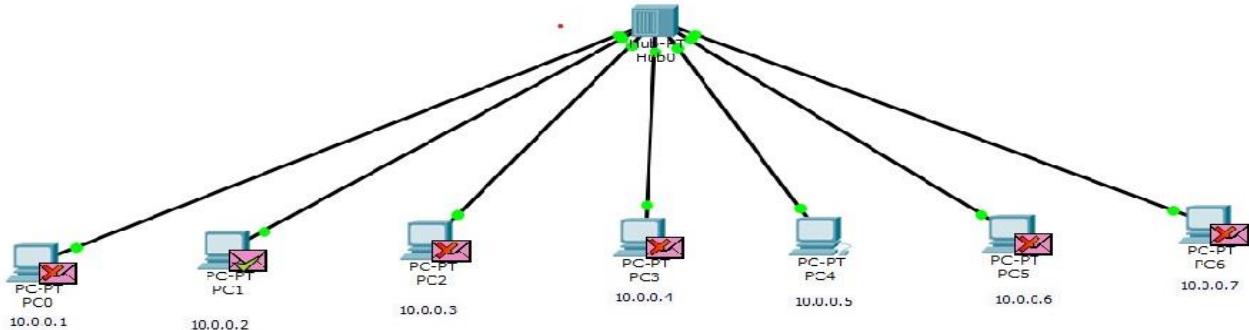
Packet: sent = 4, Received = 4, Lost = 0 (0% loss)

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

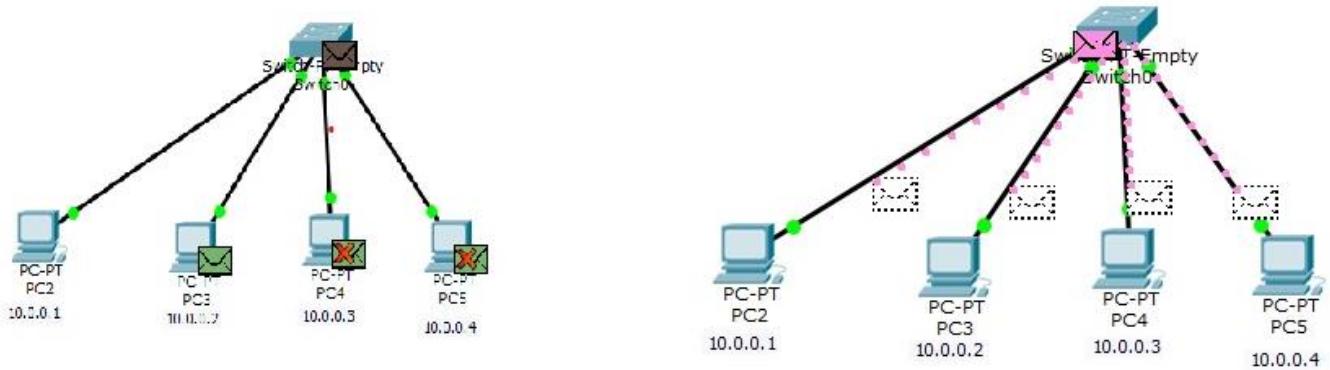
## HUBS----REAL TIME



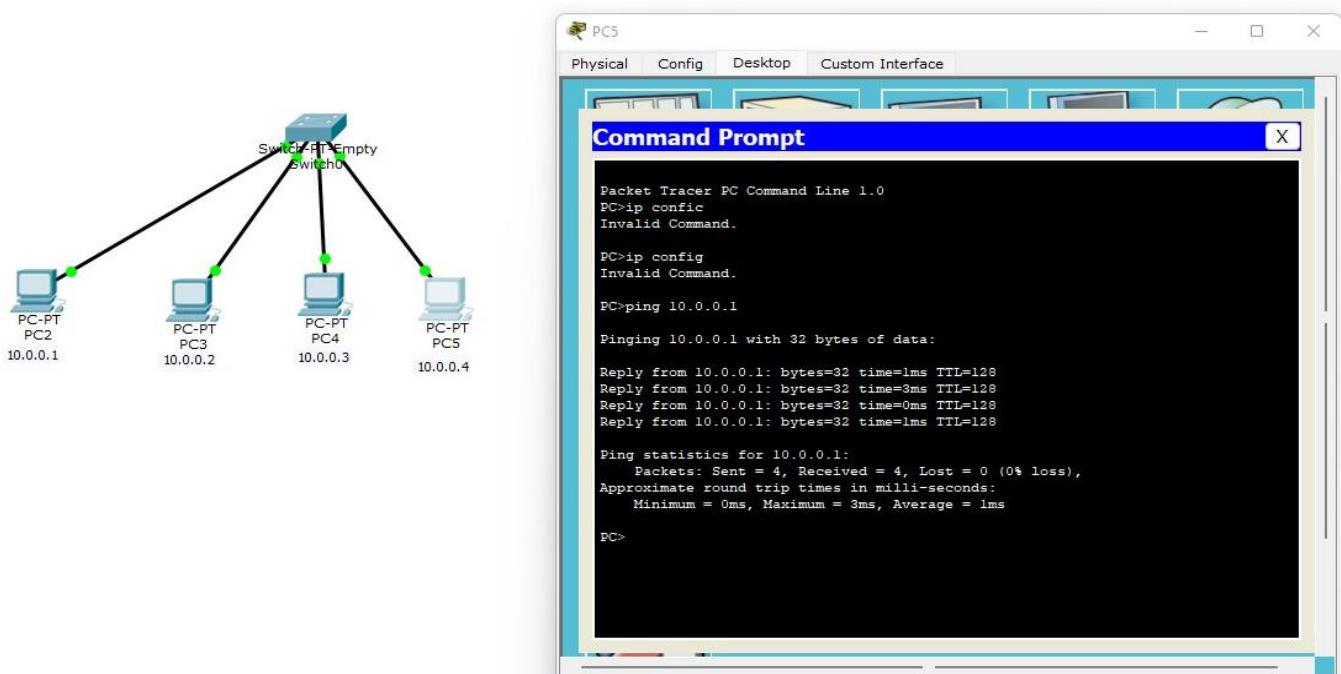
## HUBS---- STIMULATION



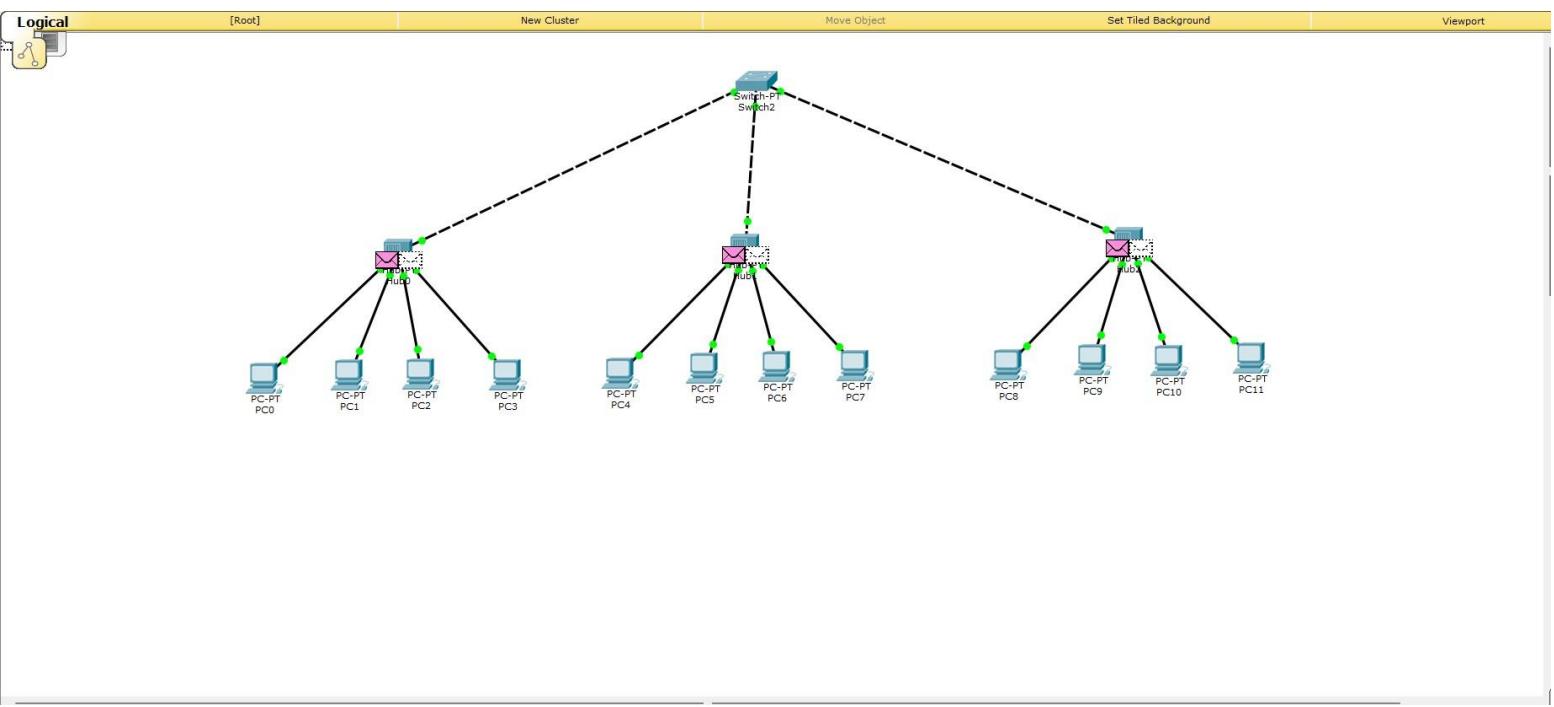
## SWITCHES-----REAL TIME



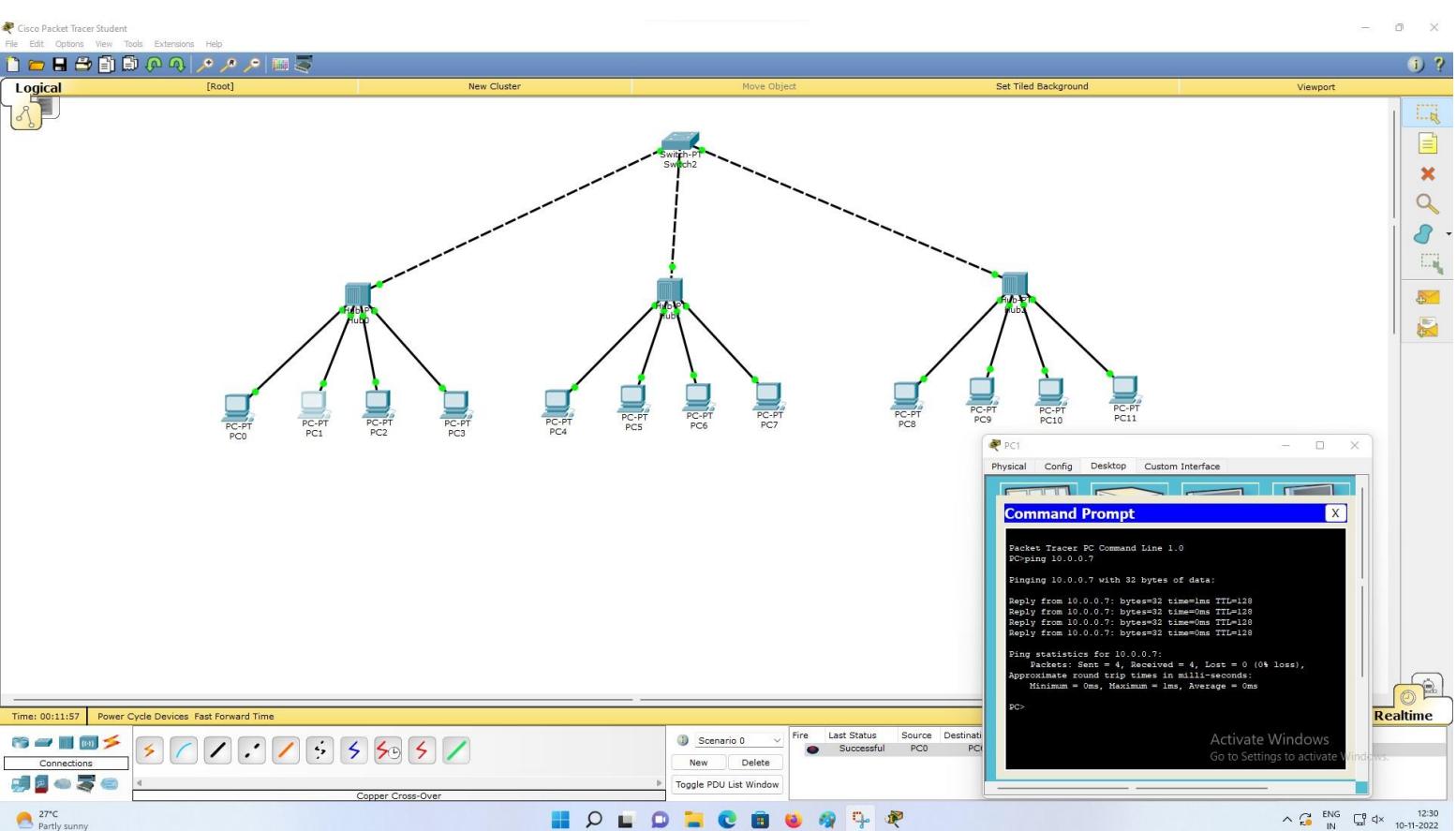
## SWITCHES-----STIMULATION



## HYBRID NETWORK-----REAL TIME



## HYBRID NETWORK-----STIMULATION

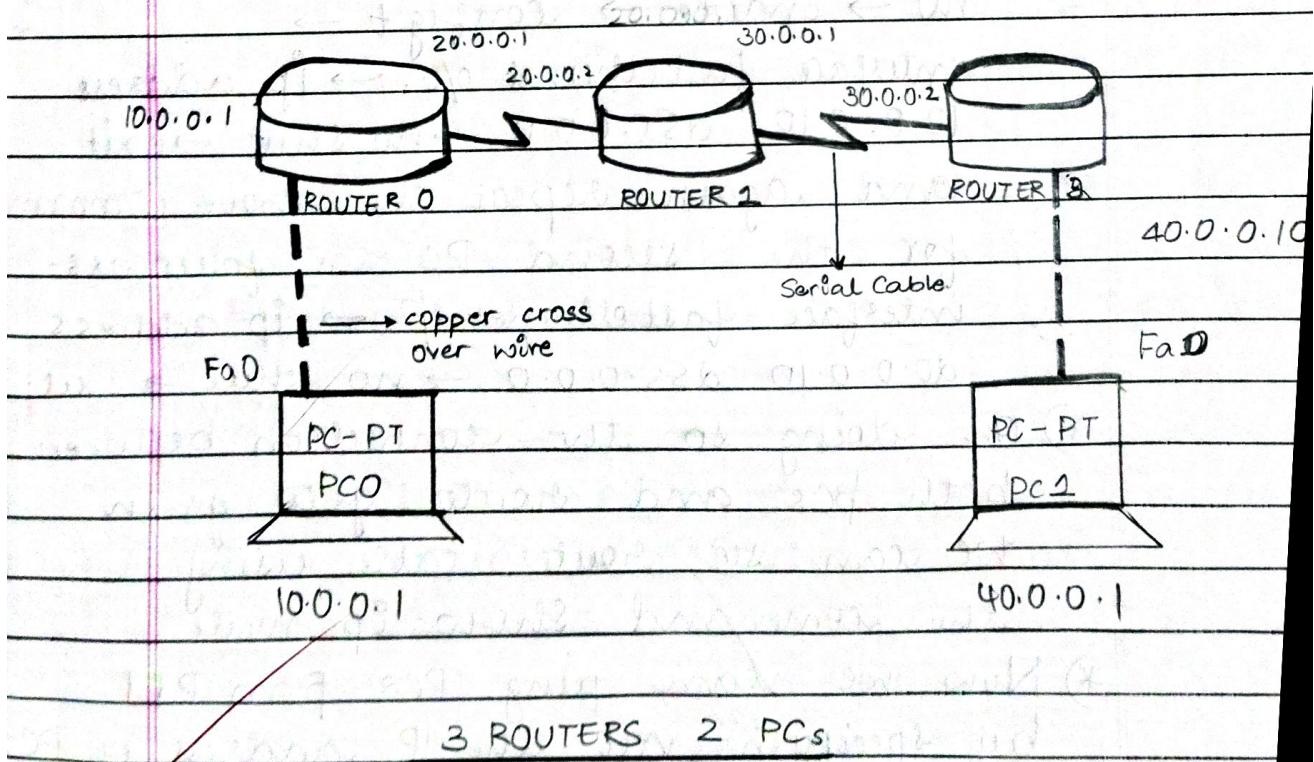
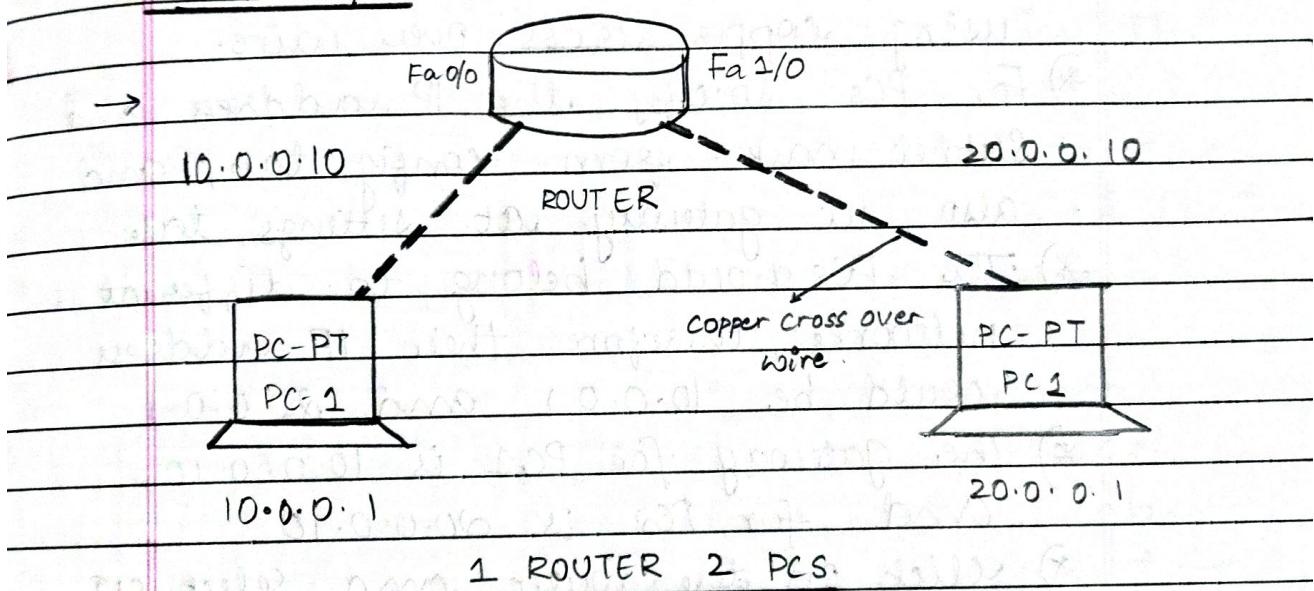


Mon	Date:	17/11/2022	Thu
Tue			Fri
Wed	Page No:		Sat

## LAB-3

AIM - Configuring IP address to routers in Packet tracer. Explore the following messages: Ping responses, Destination unreachable, Request time out reply.

### TOPOLOGY :-



Mon	Date:	20	Thur
Tue			Fri
Wed	Page No.:		Sat

## PROCEDURE:-

- using Single router and 2 PCs.
- \* Add one generic router and 2 PCs into the workstation
- \* Connect both the PCs to the router using copper cross over wire.
- \* For PCs Specify the IP address and Subnet mask from config tab, and also set gateway at settings tab.
- \* The PCs should belong to different networks, therefore their IP address could be. 10.0.0.1 and 20.0.0.1
- \* The gateway for PC1 is 10.0.0.10, And for PC2 is 20.0.0.10
- \* Click on the router and Select CLI  
`no → enable → config →`  
`interface fastethernet 0/0. → ip address`  
`10.0.0.10 255.0.0.0 → no shut → exit`  
 and Again repeat the above command for the second PC as follows:-  
`interface fastethernet 1/0 → ip address`  
`20.0.0.10 255.0.0.0 → no shut → exit`
- \* On doing so the connection between both PCs and router gets green we can see router table using the command Show ip route.
- \* Now we can ping PC2 from PC1 by specifying all the IP address of PC in the command prompt of PC1.

Mon	Date:	20	Thur
Tue			Fri
Wed	Page No:		Sat

→ Using 2 PC and 3 routers.

- \* Place three generic routers and 3 PCs into the workstation. connect PC1 to router R1 using copper cross over wire. Routers R1, R2 and R3 are connected to each other using serial DCE cable. All connections in this network are initially shown red.
- \* Routers ~~are~~ are connected to each other through serial ports and Routers and PCs are connected through fast ether net.
- \* The IP address, subnet mask and gateway is ~~selected~~ set for each PC in their config and settings tab.
- \* The CLI of router R1 is opened → no → enable → config t → interface fast ethernet 0/0 → ip address 10.0.0.10 255.0.0.0 → no shut → exit.  
By doing this we establish the connection b/w the PC1 and Router R1.
- \* For connection b/w Router R2 and R1.  
Open CLI of R1 → config t → interface serial 0/0 → ip address 10.0.0.10 255.0.0.0  
no shut → exit ⇒ connection established
- \* For connection b/w Router R2 and R3  
~~Open CLI of R2 → config t → interface serial 0/0 → ip address 10.0.0.2 255.0.0.0  
→ no shut → exit~~  
config t → interface serial 3/0 → ip address 30.0.0.1 255.0.0.0 → no shut → exit

Mon	Date:	20	Mon 20
Tue			Tue
Wed	Page No.:		Sat

\* For connection b/w R3 and PC2 :-

Open CLI → no → enable → config t →

interface serial 0/0 → ip address

30.0.0.2 255.0.0.0 → no shut

Config t → interface fastethernet 0/0

→ ip address 40.0.0.10 255.0.0.0 →

no shut → exit

\* Now all connections are established.

\* Ping PC1 from PC0 ⇒ we get a reply as  
Destination unreachable.

\* When the router 20.0.0.2 is pinged  
by PC0 the reply is given as request  
timed out

### TRAINING :-

→ Router R1.

ip route 30.0.0.0 255.0.0.0 20.0.0.2

ip route 40.0.0.0 255.0.0.0 20.0.0.2

→ Router R2

ip route 10.0.0.0 255.0.0.0 20.0.0.1

ip route 40.0.0.0 255.0.0.0 30.0.0.2

→ Router R3

ip route 10.0.0.0 255.0.0.0 30.0.0.1

ip route 20.0.0.0 255.0.0.0 30.0.0.1

\* Now we can ping PC1 from PC0  
and all replies are seen.

\* We can check the routing table using  
the 'show ip route' command in  
the CLI of a particular router.

Mon	Date:	20	Thur
Tue			Fri
Wed	Page No:		Sat

## OBSERVATION

### → 1 Router:-

when PCO pings PC1 for first time we get:-  
the first packet as request timed out.

#### result :-

ping 20.0.0.1  
pinging 20.0.0.1 with 32 bytes of data  
request timed out

reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

reply from 20.0.0.1 : bytes = 32 time 0 ms TTL:127

#### ping statistics :-

packet sent 4 , received 3 lost = 1 (25%).

Now if we ping PC1 again from PCO we get all 4 packets without any loss.

Now reverse pinging of PCO from PC1 will also not lead to any loss, all packets are acknowledged.

### → 3 Routers:-

~~Before training Routers, if we ping PC1 from PCO we get as follows:-~~

ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data:

reply from 40.0.0.1: Destination host unreachable

reply from 40.0.0.1: Destination host unreachable

reply from 40.0.0.1: Destination host unreachable

Mon	Date:	20	Mon
Tue			Tue
Wed	Page No.:		Sat

If we ping PC1 to any router we get the reply as request timed out. and all the 4 packets sent are not received with (100% loss).

After training:-

ping 40.0.0.1

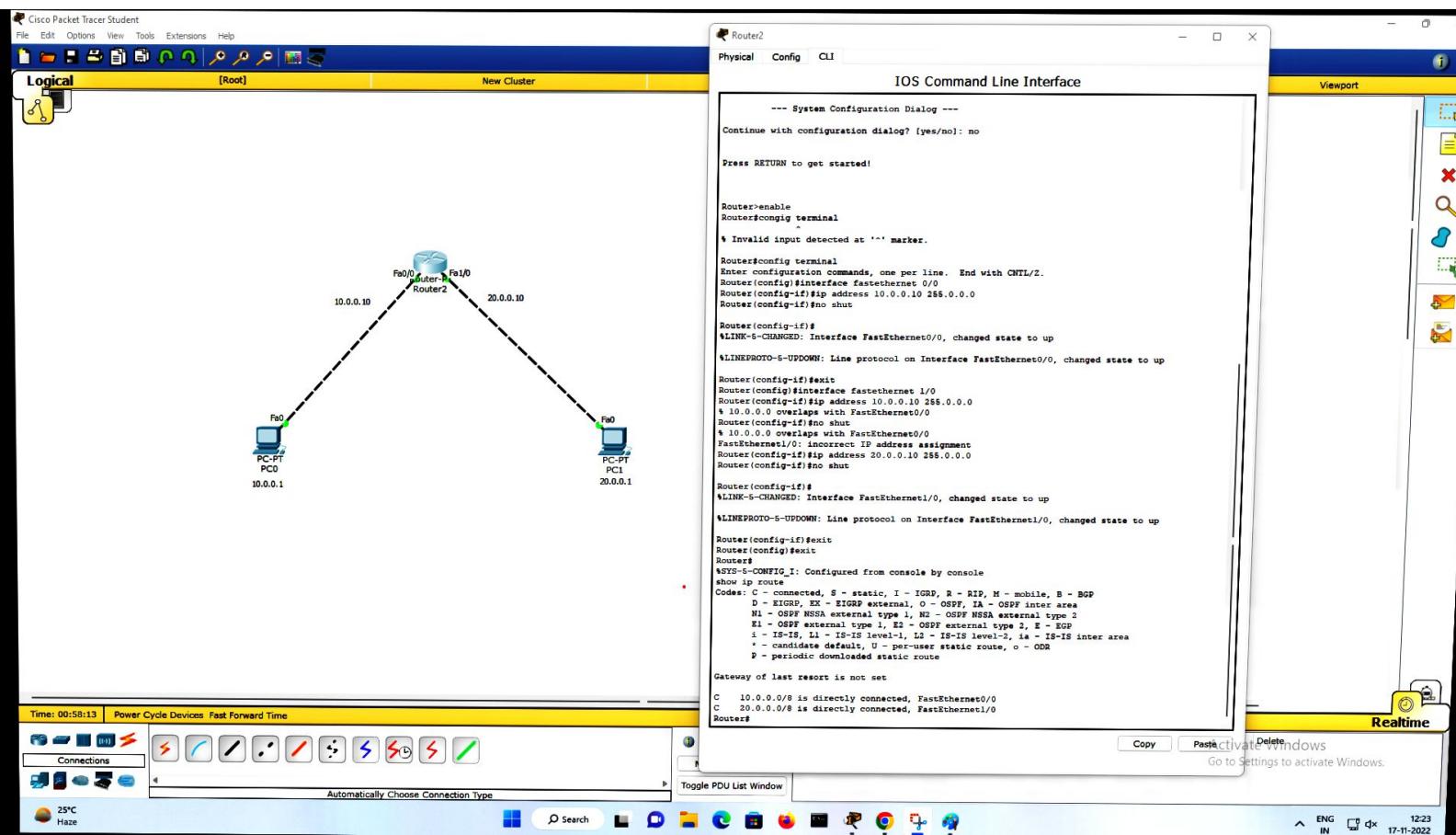
Pinging 40.0.0.1 with 32 bytes of data.  
request timed out

reply from 40.0.0.1 : bytes=32 Time=2ms TU=125

Pinging statics:-

packets sent = 4 , received = 3 loss 1 (25% loss).

✓  
24/11/22



PC

Physical Config Desktop Custom Interface

## Command Prompt

X

```
PC ping 20.0.0.1
```

```
Pinging 20.0.0.1 with 32 bytes of data:
```

```
Request timed out.
```

```
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
```

```
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
```

```
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
```

```
Ping statistics for 20.0.0.1:
```

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

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
PC ping 20.0.0.1
```

```
Pinging 20.0.0.1 with 32 bytes of data:
```

```
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
```

```
Ping statistics for 20.0.0.1:
```

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

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
PC>
```

Packet Tracer PC Command Line 1.0  
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 10.0.0.10: Destination host unreachable.  
Reply from 10.0.0.10: Destination host unreachable.  
Request timed out.  
Reply from 10.0.0.10: Destination host unreachable.

Ping statistics for 40.0.0.1:

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:

Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.

Ping statistics for 20.0.0.2:

Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.  
Reply from 40.0.0.1: bytes=32 time=7ms TTL=125  
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125  
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125

Ping statistics for 40.0.0.1:

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

Approximate round trip times in milli-seconds:

Minimum = 7ms, Maximum = 9ms, Average = 8ms

PC>

Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

Router(config-if)#interface serial2/0
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, cinterface serial2/0
Router(config-if)#exit
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

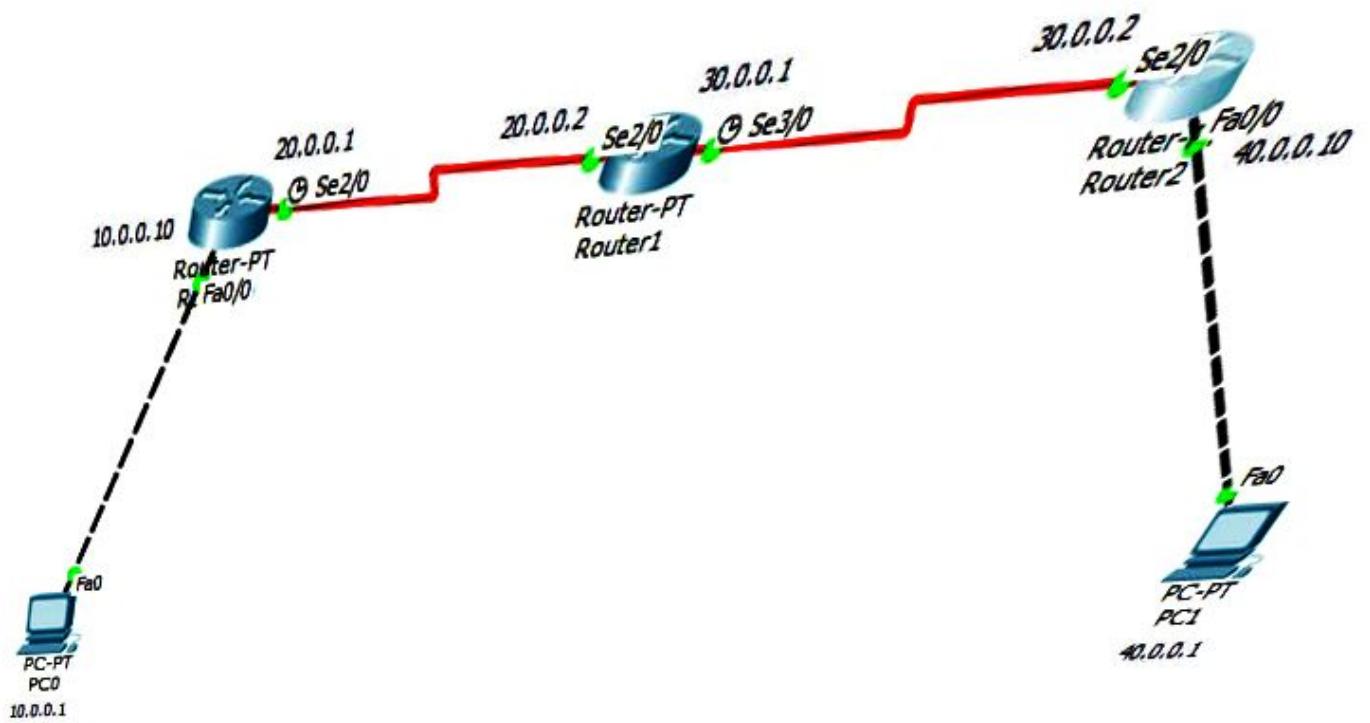
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#exit
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

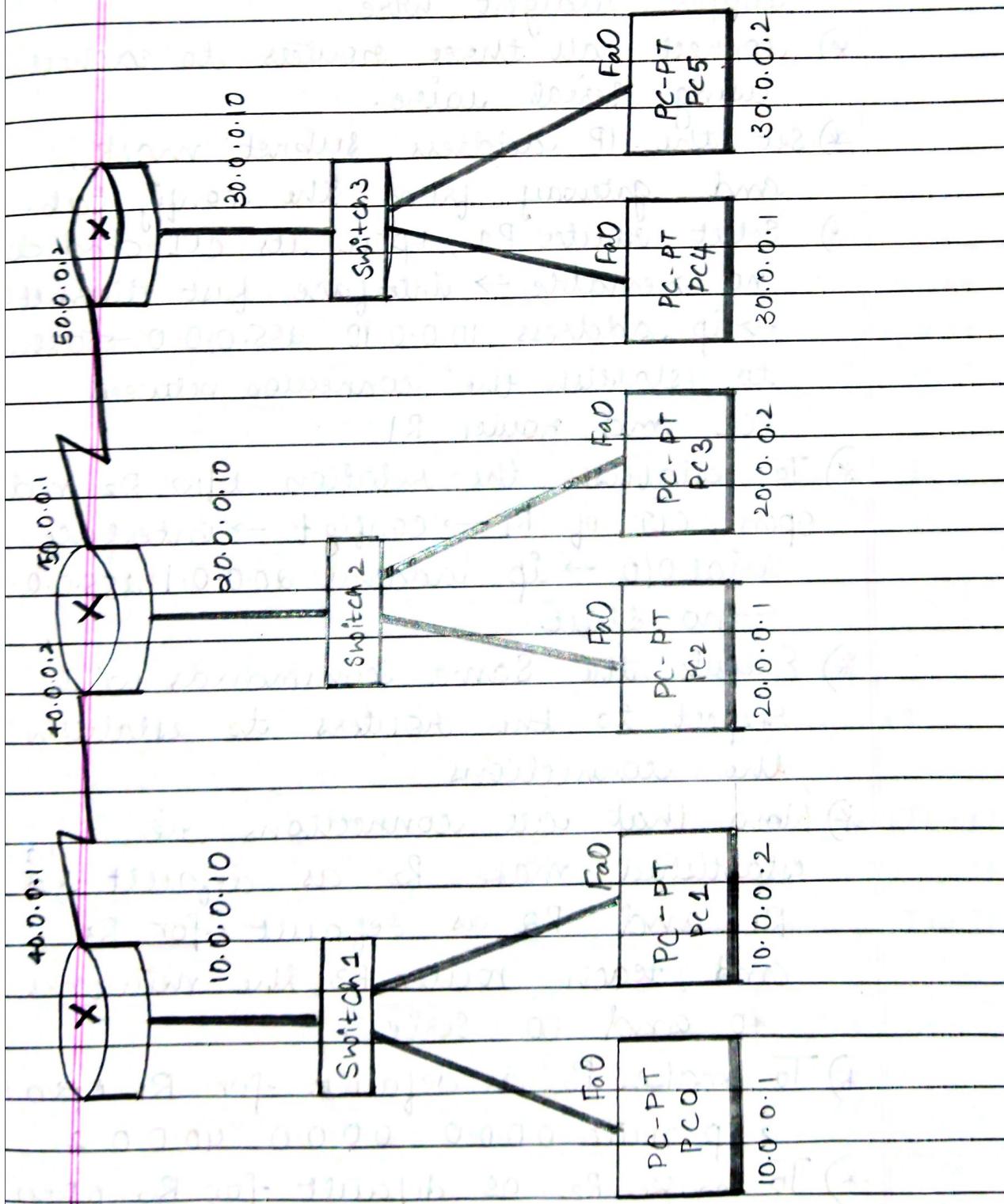
S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
Router#
```



## LAB - 4

AIM :- Configuring default route to the routers.

### TOPOLOGY :-



Mon	Date:	20	Thur
Tue			Fri
Wed	Page No:		Sat

## PROCEDURE:-

- \* Place 3 routers, 3 switches and 6 PCs into the workstation.
- \* Connect 2 PCs to each switch and each switch to each router using copper straight wire.
- \* connect all three routers to each other using serial wire.
- \* Set the IP address, subnet mask, and gateway from the config tab.
- \* Select router R1, open its CLI and do
  - no → enable → interface fast ethernet 0/0
  - ip address 10.0.0.10 255.0.0.0 → no shut
  - to establish the connection between PC1 and router R1
- \* To establish the relation b/w R2 and R1
  - open CLI of R1 → config t → interface serial 0/0 → ip address 20.0.0.1 255.0.0.0
  - no shut.
- \* Execute the same commands with respect to the routers to establish the connections
- \* Now that all connections are established make R2 as default for R1 and R3 as default for R3 and teach router R3 the networks 40 and 50 series.
- \* To make R2 as default for R1 open CLI
  - ip route 0.0.0.0 0.0.0.0 40.0.0.2
- \* To make R2 as default for R3 open

Mon	Date:	20	Thu
Tue			Fri
Wed	Page No:		Sat

CLI of R3 → ip route 0.0.0.0 0.0.0.0 50.0.0.1

\* To teach about 10 and 30 network to R2

→ open CLI of router R2 → config t →

ip route 10.0.0.0 255.0.0.0 40.0.0.1

ip route 30.0.0.0 255.0.0.0 50.0.0.2.

\* Ping one PC of a network from another PC of different network.

### OBSERVATION:-

Learning :- In this network router R2 does not have a default router, because R1 and R3 cannot become a default router simultaneously and if any one of R1 and R2 is default then the packets that are supposed to enter router R2 ~~can~~ go to R3 | R1 as they are default.

### RESULT :- Command prompt of PC0

> ping 20.0.0.2

pinging 20.0.0.2 with 32 bytes of data:

request timed out

reply from 20.0.0.2: bytes=32 time=5ms TTL=126

reply from 20.0.0.2: bytes=32 time=5ms TTL=126

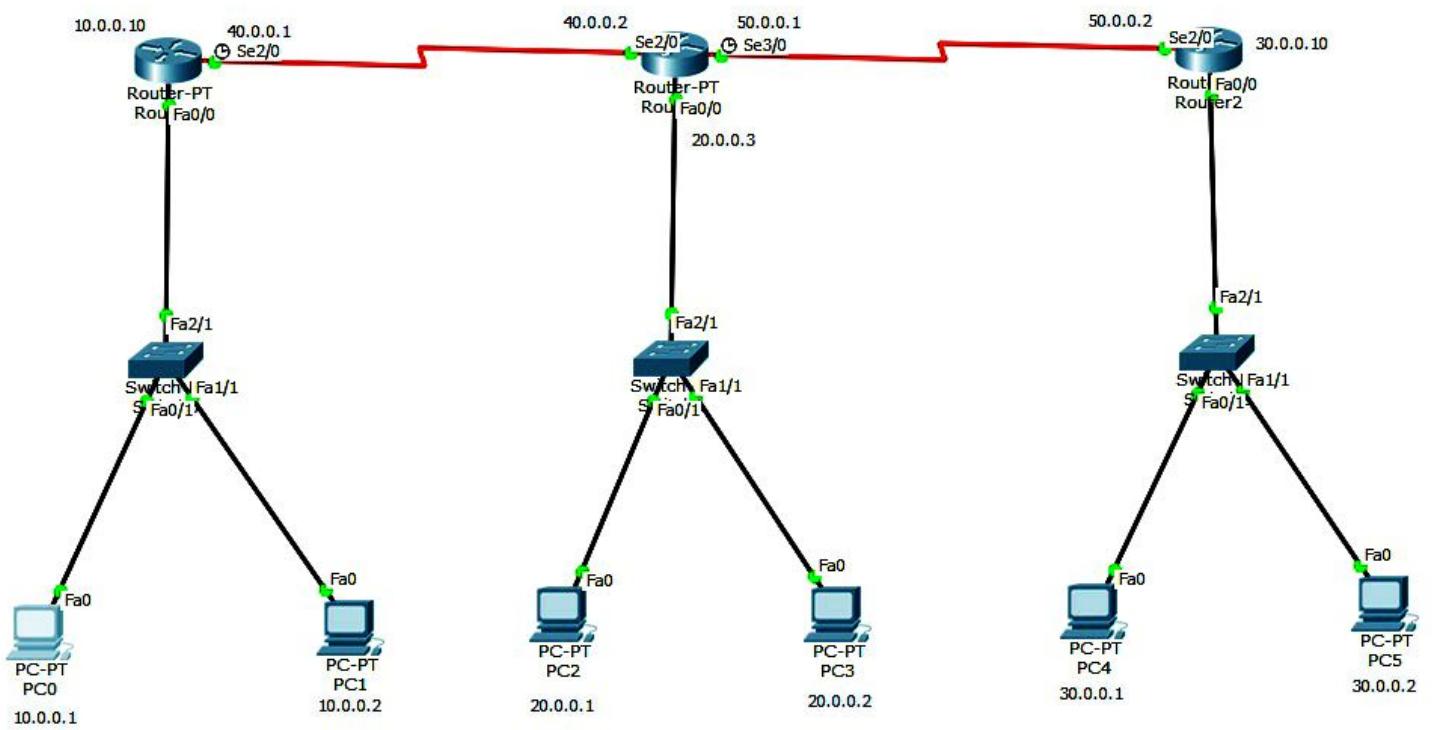
reply from 20.0.0.2: bytes=32 time=5ms TTL=126

ping statistics for 20.0.0.2:

packets: sent = 4, Received = 3, lost (25% loss),

Aprox round trip time in null second

minimum=2ms, maximum=11ms Average=7ms.



**Router0**

Physical Config CLI

### IOS Command Line Interface

```

Press RETURN to get started!

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router>ip route 0.0.0.0 0.0.0.0 40.0.0.2
^
% Invalid input detected at `'' marker.

Router>ip route 0.0.0.0 0.0.0.0 40.0.0.2
^
% Invalid input detected at `'' marker.

Router>enable
Router>config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 40.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

```

**Router1**

Physical Config CLI

### IOS Command Line Interface

```

Press RETURN to get started!

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router>enable
Router>config t
Translating "couter"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address

Router>config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 80.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

```

**Router2**

Physical Config CLI

### IOS Command Line Interface

```

Press RETURN to get started!

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router>enable
Router>config t
Translating "configt"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address

Router>enable
Router>config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 60.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

```

## Command Prompt

```
Packet Tracer PC Command Line 1.0
```

```
PC>ping 30.0.0.2
```

```
Pinging 30.0.0.2 with 32 bytes of data:
```

```
Request timed out.
```

```
Reply from 30.0.0.2: bytes=32 time=5ms TTL=126
```

```
Reply from 30.0.0.2: bytes=32 time=3ms TTL=126
```

```
Reply from 30.0.0.2: bytes=32 time=6ms TTL=126
```

```
Ping statistics for 30.0.0.2:
```

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

```
Approximate round trip times in milli-seconds:
```

```
        Minimum = 3ms, Maximum = 6ms, Average = 4ms
```

```
PC>ping 30.0.0.1
```

```
Pinging 30.0.0.1 with 32 bytes of data:
```

```
Request timed out.
```

```
Reply from 30.0.0.1: bytes=32 time=3ms TTL=126
```

```
Reply from 30.0.0.1: bytes=32 time=10ms TTL=126
```

```
Reply from 30.0.0.1: bytes=32 time=11ms TTL=126
```

```
Ping statistics for 30.0.0.1:
```

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

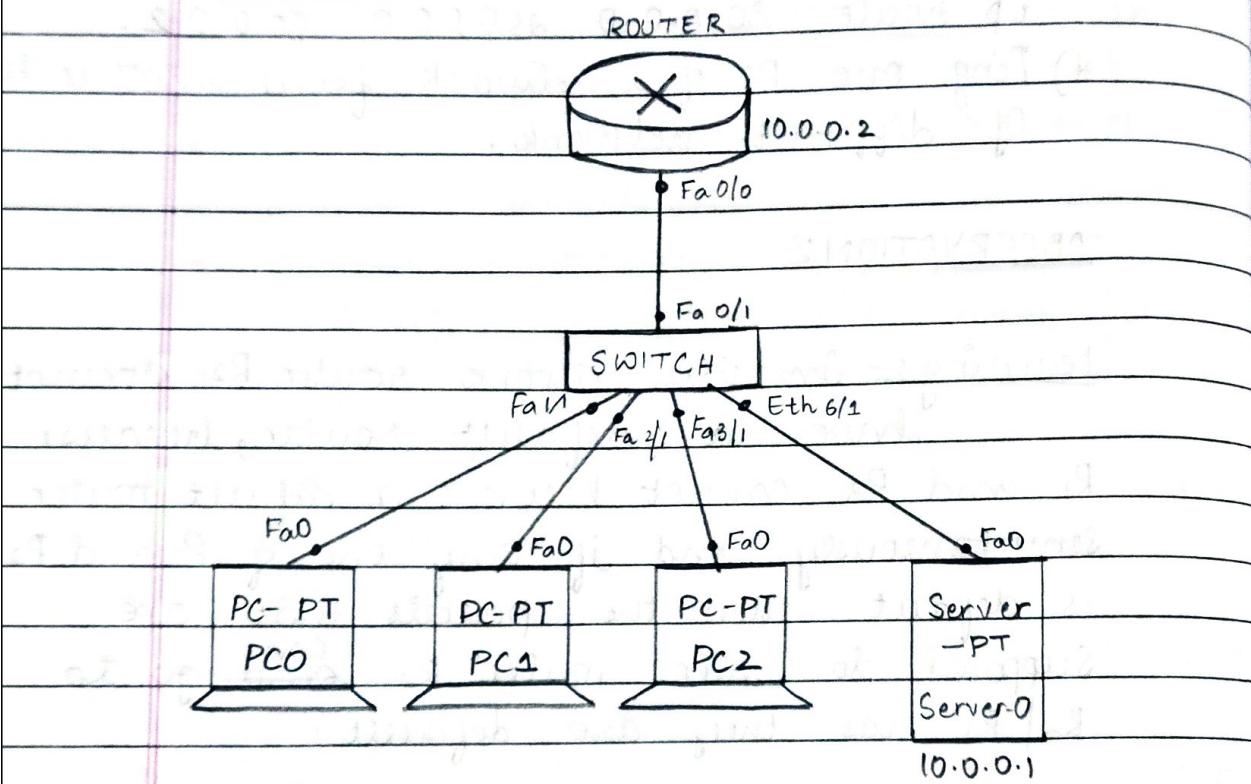
```
Approximate round trip times in milli-seconds:
```

```
        Minimum = 3ms, Maximum = 11ms, Average = 7ms
```

## LAB - 4

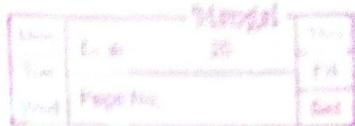
AIM: Configuring DHCP within a LAN in a packet tracer.

### TOPOLOGY:-



### PROCEDURE:-

- \* Place a generic router, generic switch, and 3 PCs into the workstation and connect them as:- Router to switch and switch to all 3 PCs and Server.
- \* Open Server Config tab, set its IP address and Subnet mask, from Settings tab set its gateway.
- \* For Server IP address is 10.0.0.1, gateway is 10.0.0.2.



- \* Open the CLI of router → no → enable → config → interface fastethernet 0/0 → ip address 10.0.0.2 255.0.0.0 → no shut
- \* Go to the Services tab of server, and select DHCP from the left panel. On the Services, Set default gateway as 10.0.0.2 and change DNS and TFTP to 10.0.0.1 In start IP address change it to the value from where you want to start the IP pool. Save the changes.
- \* Select the PC, go to the Desktop tab and Select ~~config IP address~~ IP configuration. There change from static to DHCP. we can notice that all the details like ip address, Subnet mask, Default Gateway and DNS Server is set by Default.
- \* Do the above step for all the PCs.

### OBSERVATIONS:-

### LEARNING:-

The server automatically sets the IP address, Subnet mask, and gateway to all the PCs and IP address is allocated serially in DHCP protocol.

Mon	08:00	20	Monday
Tue			
Wed			

## RESULT :-

> ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data

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

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

Approximate round trip times in milli-second:

Minimum = 0ms, Maximum = 0ms, Average = 0ms.

Router0

Physical Config CLI

## IOS Command Line Interface

```
--- System Configuration Dialog ---
```

```
Continue with configuration dialog? [yes/no]: no
```

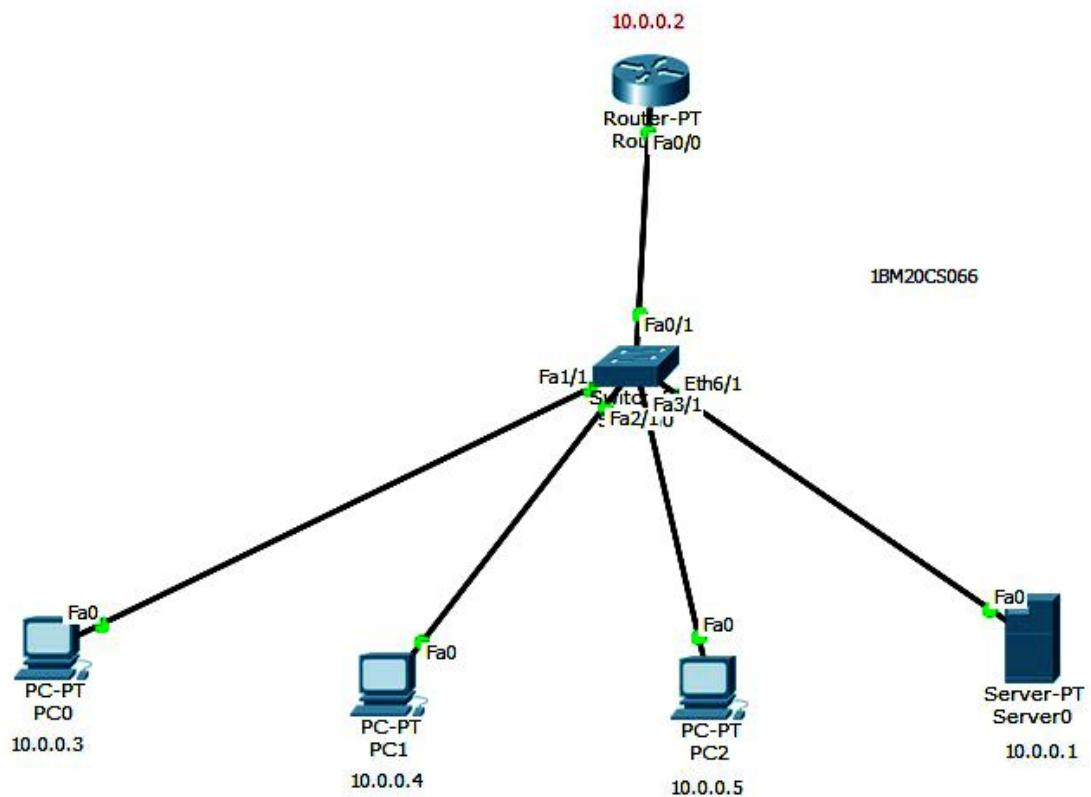
```
Press RETURN to get started!
```

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip maddress 10.0.0.2 255.0.0.0
^
% Invalid input detected at '^' marker.

Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to
up
```



X

## Command Prompt

```
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=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 = 0ms, Average = 0ms

PC>
```

Server0

Physical Config Services Desktop Custom Interface

**SERVICES**

- HTTP
- DHCP
- DCHPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP

**DHCP**

Interface: FastEthernet0 Service:  On  Off

Pool Name: serverPool

Default Gateway: 10.0.0.2

DNS Server: 10.0.0.1

Start IP Address : 10 0 0 3

Subnet Mask: 255 0 0 0

Maximum number of Users : 512

TFTP Server: 10.0.0.1

Add Save Remove

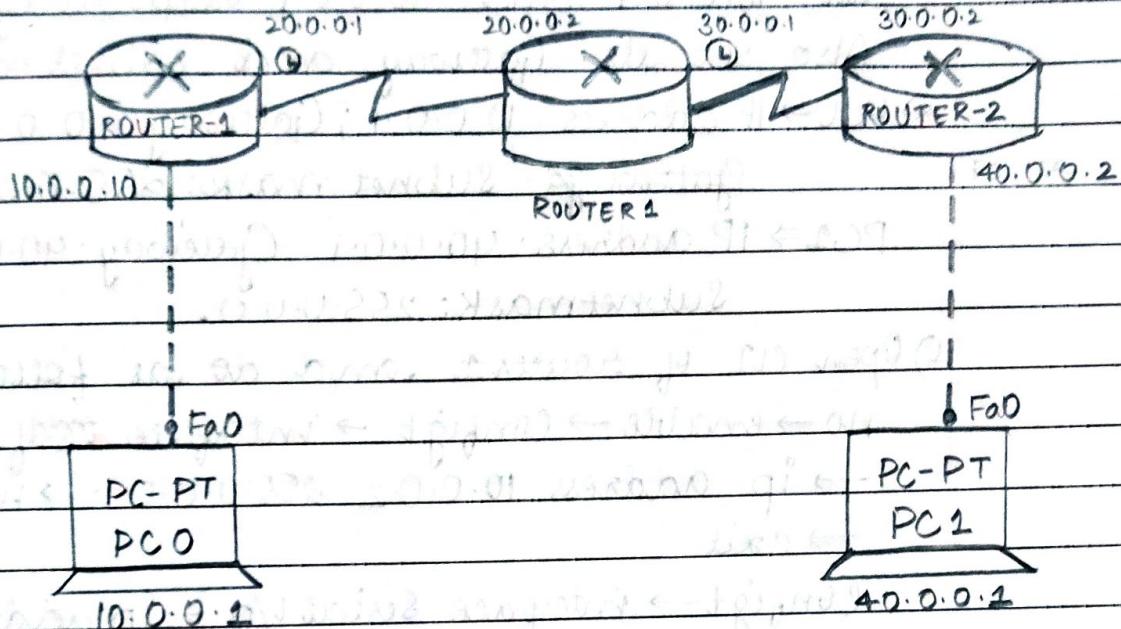
Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP
server...	10.0.0.2	10.0.0.1	10.0.0.3	255.0.0.0	512	10.0.0.1

Mon	Date:	8/12/2022	Mongol
Tue			Thu
Wed		Page No:	Fri
			Sat

## LAB-5

AIM:- Configuring RIP routing protocol in Router

### TOPOLOGY:-



### SERIAL DCE.

We can enable clocking on the DCE side to bring up the line protocol. You can tell which end of the connection is the DCE side by the small "clock" icon next to the port.

RIP:- It is a protocol that routers can use to exchange network topology information. RIP uses a distance vector algorithm to decide which path to put a packet onto to get to its destination.

## PROCEDURE :-

\* ) Place two PCs and 3 routers in the workspace.

\* ) For both the PCs sets its IP address as 10.0.0.1 and 40.0.0.1 respectively.

Also set its Gateway and Subnet mask.

\* ) PC0  $\Rightarrow$  IP address : 10.0.0.1 ; Gateway : 10.0.0.2

~~Gateway~~ - Subnet mask: 255.0.0.0

PC1  $\Rightarrow$  IP address: 40.0.0.1 Gateway: 40.0.0.2

Subnetmask: 255.0.0.0.

\* ) Open CLI of router1 and do as follows:-

no  $\rightarrow$  enable  $\rightarrow$  config t  $\rightarrow$  interface ~~fastethernet~~ fastethernet

$\rightarrow$  ip address 10.0.0.2 255.0.0.0  $\rightarrow$  no shut

$\rightarrow$  exit

config t  $\rightarrow$  interface serial 0/0  $\rightarrow$  ip address

20.0.0.1 255.0.0.0  $\rightarrow$  encapsulation PPP  $\rightarrow$

clock rate 64000  $\rightarrow$  no shut.

\* ) Open CLI of router2 and do as follow.

no  $\rightarrow$  enable  $\rightarrow$  config t  $\rightarrow$  interface serial 2/0  $\rightarrow$

ip address 20.0.0.2  $\rightarrow$  encapsulation PPP  $\rightarrow$  no shut

exit  $\rightarrow$  exit

Config t  $\rightarrow$  interface serial 3/0  $\rightarrow$  ip address

30.0.0.1 255.0.0.0  $\rightarrow$  encapsulation PPP

clockrate 64000  $\rightarrow$  no shut.

Now the connection between Router 1 and Router2.

\* ) Open CLI of router3 and do as follows:-

enable  $\rightarrow$  config t  $\rightarrow$  interface fastethernet 0/0

ip address 40.0.0.2 255.0.0.0  $\rightarrow$  no shut.

exit  $\rightarrow$  exit  $\rightarrow$  config

Interface Serial 2/0  $\rightarrow$  ip address 30.0.0.2

255.0.0.0  $\rightarrow$  encapsulation PPP  $\rightarrow$  no shut.

Mon	Date:	Mangal 20	Thu
Tue			Fri
Wed	Page No.:		Sat

Now the connections are all established.

~~\*)~~ \*) Now specify the RIP protocol.

\*) Now open the CLI of router 1:-

router rip → network 10.0.0.0 → network  
20.0.0.0 → exit → exit.

\*) Open the CLI of router 2:-

router rip → network 20.0.0.0 → network  
30.0.0.0 → ~~exit~~.

\*) Ping PC<sub>1</sub> to PC<sub>2</sub>.

### OBSERVATION:-

#### RESULT:-

PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out

Reply from 40.0.0.1: byte = 32 time = 6ms TTL = 125

Reply from 40.0.0.1: byte = 32 time = 9ms TTL = 125

Reply from 40.0.0.1: byte = 32 time = 7ms TTL = 125

Ping statistics for 40.0.0.0

Packets: Sent = 4, received = 3, lost = 1 (25% loss).

### LEARNING:-

when RIP protocol is used we do not have to do static routing for all the routers i.e., we have to teach all the routers by providing with the next hop in dynamic routing (RIP protocol) we just have to specify the networks known by the router.

```
Continue with configuration dialog? [yes/no]: no
```

```
Press RETURN to get started!
```

```
Router>enable
Router#interface fastethernet0/0
^
% Invalid input detected at '^' marker.

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#configt
Translating "configt"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 20.0.0.2
% Incomplete command.
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 20.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shut'
^
% Invalid input detected at '^' marker.

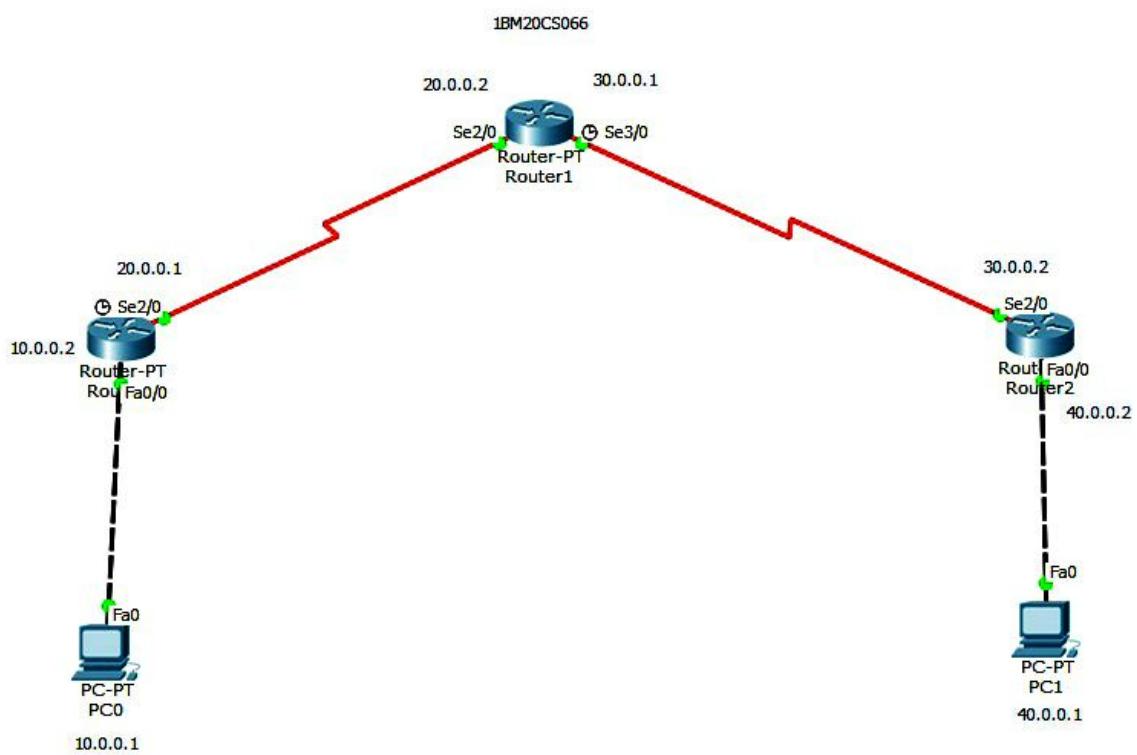
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#exit
Router(config)#

```





PC0



Physical Config Desktop Custom Interface

## Command Prompt

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

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=6ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=7ms TTL=125

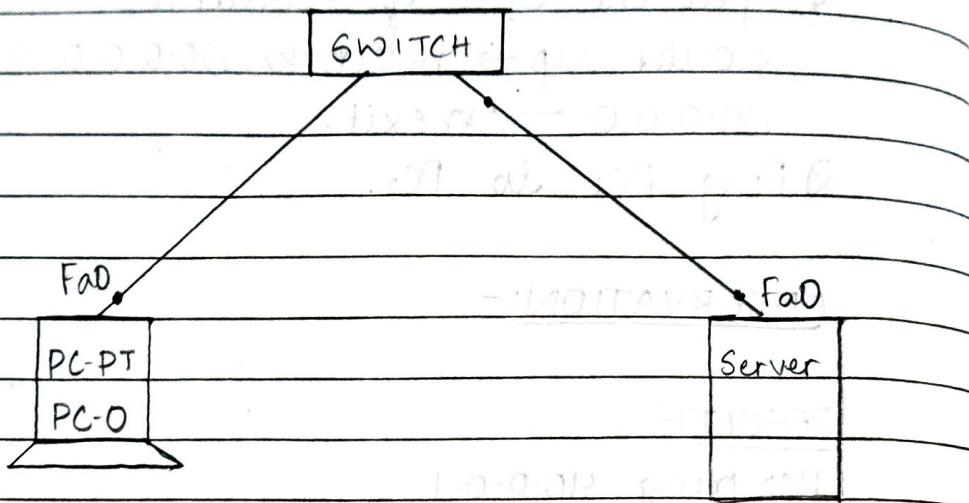
Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 9ms, Average = 7ms

PC>
```

## LAB-6

AIM: Demonstration of WEB3 server and DNS using packet tracer.

TOPOLOGY:-



PROCEDURE:-

- \* Place a PC, switch and Server on the workspace and connect them.
- \* Set the IP address and Subnet mask of the PC as 10.0.0.1 and 255.0.0.0 respectively.
- \* Set the IP address and Subnet mask of Server as 10.0.0.2 and 255.0.0.0.
- \* Open PC → Desktop → web browser → give IP address of Server (10.0.0.2).
- \* Open Server → Services → HTTP → the HTTP windows open. Click on the edit option of index.html and change the contents → Save.
- \* Now the web browser page of PC is also modified.

Mon	Date	20	Page No.	Mosol
Tue				Fri
Wed				Sat

## DOMAIN NAMING SYSTEM:-

- \* To activate DNS, Open Server → Services → DNS → On → Enter the name of the resource record and IP address of the server and click on Add.
- \* Now the name and IP address is ~~change~~ fixed.
- \* Now create your own html file :-
- \* ~~Now open PC~~ Now open PC → web browser → and specify the URL → GO.
- \* The contents of the HTML file created is shown.

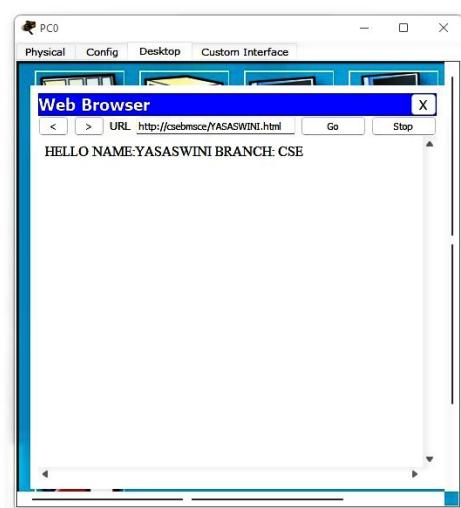
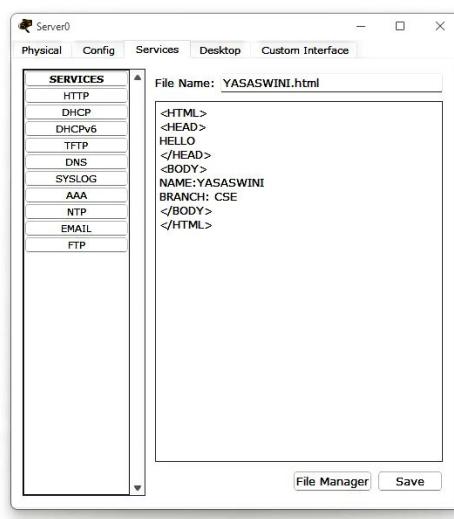
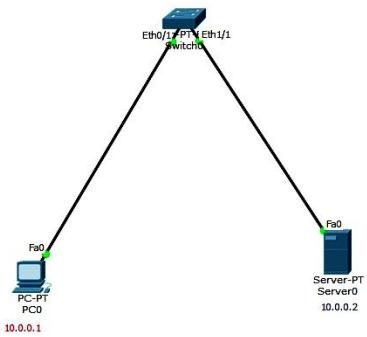
### RESULT:-

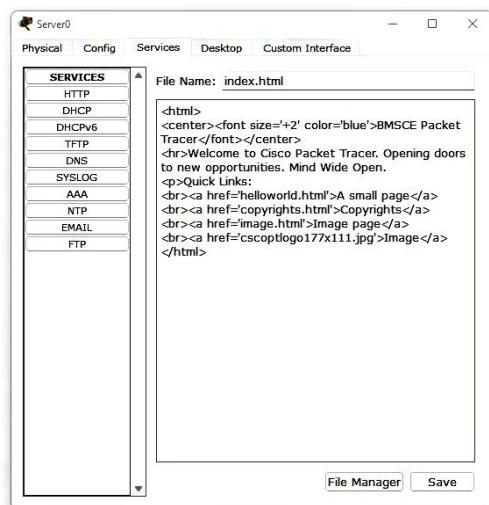
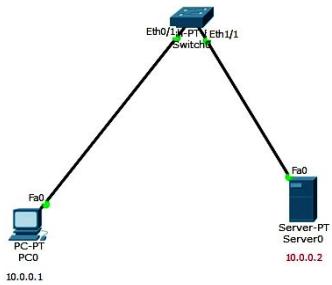
```
HTML file created :- <html>
    <head> Hello </head>
    <body>
        Name: yasaswini <br>
        branch: CSE
    </body>
</html>
```

output:- Hello.

Name: yasaswini  
branch: CSE

LEARNING:- DNS helps us map name with our IP address. We are comfortable with naming conventions such as bmsce.ac.in whereas computer is comfortable with IP address. Hence DNS helps in the mapping of the name and the IP.





## DJIKSTRAS ALGORITHM

```

#include <bits/stdc++.h>
#include <limits.h>
#include <stdio.h>
using namespace std;
#define V 4
int mindist(int dist[], bool SptSet[])
{
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++)
        if (SptSet[v] == false && dist[v] <= min)
            min = dist[v], min_index = v;
    return min_index;
}
void printSolutions(int dist[])
{
    printf("Vertex");
    for (int i = 0; i < V; i++)
        printf("\n%d %d", i, dist[i]);
}
void dijkstra(int graph[V][V], int src)
{
    int dist[V];
    bool SptSet[V];
    for (int i = 0; i < V; i++)
        dist[i] = INT_MAX, SptSet[i] = false;
    dist[src] = 0;
    for (int count = 0; count < V - 1; count++) {
        int u = mindist(dist, SptSet);
        SptSet[u] = true;
        for (int v = 0; v < V; v++)
            if (!SptSet[v] && graph[u][v] >= 0 && dist[u] + graph[u][v] < dist[v])
                dist[v] = dist[u] + graph[u][v];
    }
}

```

Month	Today	29	Days
Year	Page No.	29	Page
Week	Page No.	29	Page

```

if (!sp[s][v] && graph[u][v] && dist[v] != INT_MAX && dist[u]+graph[u][v]<dist[v])
    dist[v]=dist[u]+graph[u][v];
}

printf("dist");
}

int main()
{
    int graph[v][v];
    cout << "Enter the graph" << endl;
    for(int i=0;i<v;i++)
    {
        for(j=0;j<v;j++)
            cin>>graph[i][j];
    }
    dijkstra(graph,0);
}

```

Output:- Enter the graph.

```

0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0

```

vertex	Distance from Source
0	0
1	8
2	2
3	5

Graph with 4 vertices and 8 edges  
 0 1 2 3  
 0 8 2 5  
 1 0 6 8  
 2 6 0 0  
 3 8 0 0

Enter no. of vertices:4

Enter the adjacency matrix:

0 5 9999 9999

2 0 4 9999

9999 9999 0 6

4 7 5 0

Enter the starting node:0

Distance of node1=5

Path=1<-0

Distance of node2=9

Path=2<-1<-0

Distance of node3=15

Path=3<-2<-1<-0

Mon	Tues	Wed	Thurs

## DISTANCE VECTOR:-

```

#include <iostream>
using namespace std;
#define MAX_ID
int n;
class Router {
    char adj-new[MAX], adj-old[MAX];
    int table-new[MAX], table-old[MAX];
public:
    for(int i=0; i<MAX; i++)
        table-old[i] = table-new[i] = 99;
    }
    void copy() {
        for (int i=0; i<n; i++) {
            adj-new = adj-old[i];
            table-old[i] = table-new[i];
        }
    }
    int equal() {
        for (int i=0; i<n; i++)
            if (table-old[i] != table-new[i] || adj-new[i] != adj-old[i])
                return 0;
        return 1;
    }
    void input(int j) {
        cout << "Enter " << char('A'+j) << " else enter 99: ";
        for (int i=0; i<n; i++)
            if (i!=j) cout << (char)('A'+i) << " ";
        cout << " Enter matrix ";
        for (i=0; i<n; i++)
    }
}

```

```

if (i == j)
    table_new[i] = 0;
else
    cin >> table_new[i];
    adj_new[i] = (char) ('A' + i);
}
cout << endl;
}

void display() {
    cout << "Destination Router";
    for (int i = 0; i < n; i++)
        cout << " Outgoing line: ";
    for (int i = 0; i < n; i++)
        cout << " Hop Count: ";
    for (int i = 0; i < n; i++)
}

void build(int j) {
    for (int i = 0; i < n; i++)
        for (int k = 0; (i != j) && (k < n); k++)
            if (table.old[i] != 99)
                if ((table_new[i] + table_new[k] < table[k]))
                    table_new[k] = table_new[i] + table_new[k];
                    adj_new[k] = (char) ('A' + i);
}
cout << endl;
}

int g[MAX];
void build_table() {
    int i = 0, j = 0;
    while (i != n) {
        g[i] = copy();
        g[i] = build(i);
    }
}

```

Mon	Tues	20	Thurs
Fri	Sat		Sun
Mon	Project Hoc		

}

```
for (i=0; i<n; i++)
if (!sr[i].equal ()) {
    j = 1;
    break;
}
```

}

```
int main () {
```

```
cout << "Enter no. of routers"
```

```
cin >> n;
```

```
for (int i=0; i<n; i++)
```

```
sr[i].input(i);
```

```
build_table();
```

```
for (int i=0; i<n; i++) {
```

```
cout << "Entries are : " << (char) ('A' + i);
```

```
sr[i].display();
```

```
cout << endl;
```

J

J

Output: No. of routers :- 5

Enter if router is next to A : B C D E

matrix 1 = 1 1 99 99

Enter routers B : A C D E A B C E

Enter matrix : 99 99 1 99

Enter router E : A B C D

matrix :- 99 99 1 99.

routing table for A :-

Destination : A B C D E

Outgoing : A B C D E

Hop Count : 0 1 1 99 99.

Enter no. of vertices:4

Enter the adjacency matrix:

0 5 9999 9999

2 0 4 9999

9999 9999 0 6

4 7 5 0

Enter the starting node:0

Distance of node1=5

Path=1<-0

Distance of node2=9

Path=2<-1<-0

Distance of node3=15

Path=3<-2<-1<-0

## UDP

### Client :-

```

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"))
fileContents, ServerAddress = ClientSocket .
recvfrom (2048)
print (Reply from Server)
print (fileContents.decode ("utf-8"))
ClientSocket . close()
ClientSocket . close()

```

### Server :-

```

from socket import *
ServerPort = 12000
ServerSocket = socket (AF_INET, SOCK_DGRAM)
ServerSocket . bind (("127.0.0.1", ServerPort))
print ("Ready to receive")
while 1:
    Sentence, = ServerSocket . recvfrom (2048)
    Sentence = Sentence.decode ("utf-8")
    file = open (Sentence, "r")
    l = file . read (2048)
    ServerSocket . sendto (l, "utf-8"), ClientAdd)
    print (Contents sent) print (sentence)
    file . close()

```

File	Data	20	Time
File	File	File	File
File	File	File	File

Output :- ~~Server.py~~

Server:-

The Server is ready to receive.

Sent Contents of Server.py.

The Server is ready to receive.

Client Information

```
Enter file name: ServerTCP.py
```

```
From Server:
```

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

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

```
>>> I
```

## UDP

### Client :-

```

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"))
fileContents, ServerAddress = ClientSocket .
recvfrom (2048)
print (Reply from Server)
print (fileContents.decode ("utf-8"))
ClientSocket . close()
ClientSocket . close()

```

### Server :-

```

from socket import *
ServerPort = 12000
ServerSocket = socket (AF_INET, SOCK_DGRAM)
ServerSocket . bind (("127.0.0.1", ServerPort))
print ("Ready to receive")
while 1:
    Sentence, = ServerSocket . recvfrom (2048)
    Sentence = Sentence.decode ("utf-8")
    file = open (Sentence, "r")
    l = file . read (2048)
    ServerSocket . sendto (l, "utf-8"), ClientAdd)
    print (Contents sent) print (sentence)
    file . close()

```