

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## COURSE TITLE

*Submitted by*

**JAYAM MOUNEASH (1BM20CS063)**

*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**

**October-2022 to Feb-2023**

**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 “LAB COURSE **COMPUTER NETWORKS**” carried out by **JAYAM MOUNEASH (1BM20CS063)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (20CS5PCCON)** work prescribed for the said degree.

Dr. Nandhini Vineeth  
Designation  
Department of CSE  
BMSCE, Bengaluru

**Dr. Jyothi S Nayak**  
Professor and Head  
Department of CSE  
BMSCE, Bengaluru

# Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
<b>1</b>	<b>10-11-22</b>	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	
<b>2</b>	<b>17-11-22</b>	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply	
<b>3</b>	<b>24-11-22</b>	Configuring default route to the Router	
<b>4</b>	<b>01-12-22</b>	Configuring DHCP within a LAN in a packet Tracer	
<b>5</b>	<b>08-12-22</b>	Configuring RIP Routing Protocol in Routers	
<b>6</b>	<b>15-12-22</b>	Demonstration of WEB server and DNS using Packet Tracer	
<b>7</b>	<b>29-12-22</b>	Write a program for error detecting code using CRC-CCITT (16-bits)	
<b>8</b>	<b>12-01-23</b>	Write a program for distance vector algorithm to find suitable path for transmission.	
<b>9</b>	<b>12-01-23</b>	Implement Dijkstra's algorithm to compute the shortest path for a given topology.	
<b>10</b>	<b>05-01-23</b>	Write a program for congestion control using Leaky bucket algorithm.	
<b>11</b>	<b>28-01-23</b>	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.	
<b>12</b>	<b>29-01-23</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.	

30/11/12  
Lab: week-10

## Interface overview:

Initial interface has 10 components.

- 1) Menu Bar
- 2) Main Tool Bar
- 3) Common

## Workspaces and Modes

Packet traces — 2 workspaces

- 1) Logical
- 2) Physical

- 2 modes
- 1) Realtime
  - 2) Simulation

Logical / Physical Workspace & Navigation Bar: You can toggle b/w. Physical workspace & Logical workspace.

In ~~b/w~~ Physical workspace, this bar allows you to navigate through physical locations.

In Logical workspace, this bar allows you to go back to previous level in clusters.

Workspace: This area is where you will create network and watch simulations etc.

Realtime / Simulation Bar: You can toggle b/w tabs on this bar. It has & simulation mode with relative time in realtime & simulation mode.

## Simple Test

### III) Sending

- 1) Launch packet tracer
- 2) Create a network with generic PC & generic server to the workspace.
- 3) Under connections select copper straight cable & connect both PC & server.
- 4) Now after connecting both PC & server send sample RDU from PC to server.

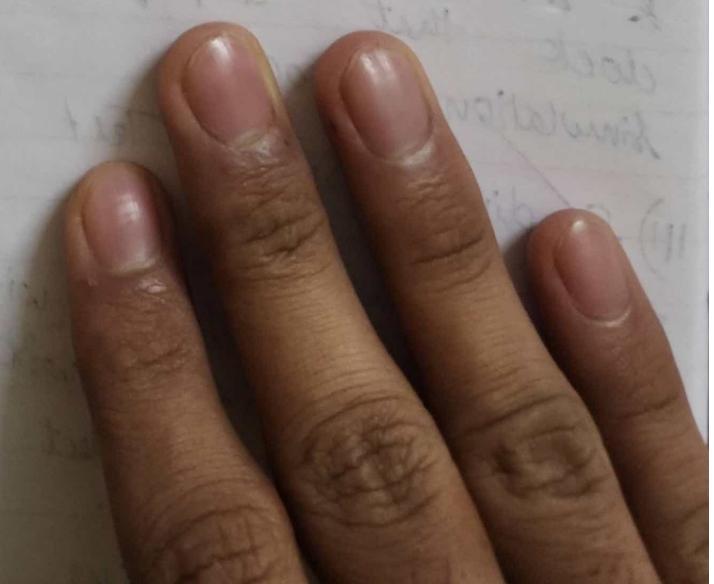
- click on Auto Capture/Play & hence we can view the animation of packet transfer.
- In real-time mode open command prompt & send ping using commands. & destination IP addresses.

Routers: A router is networking device that is used to transfer date packets between computer networks.

Switches: Switch is networking device used to segment the network into different subnetworks called lan networks.

- IPRT is address that consists of network address and host address
- Copper bus over is wire that is used to connect two wired connected devices of the same network level.
- Open C drive → prog files (x86) → CISCO packet help → default → index.htm

N  
10/11/22



10/11/22

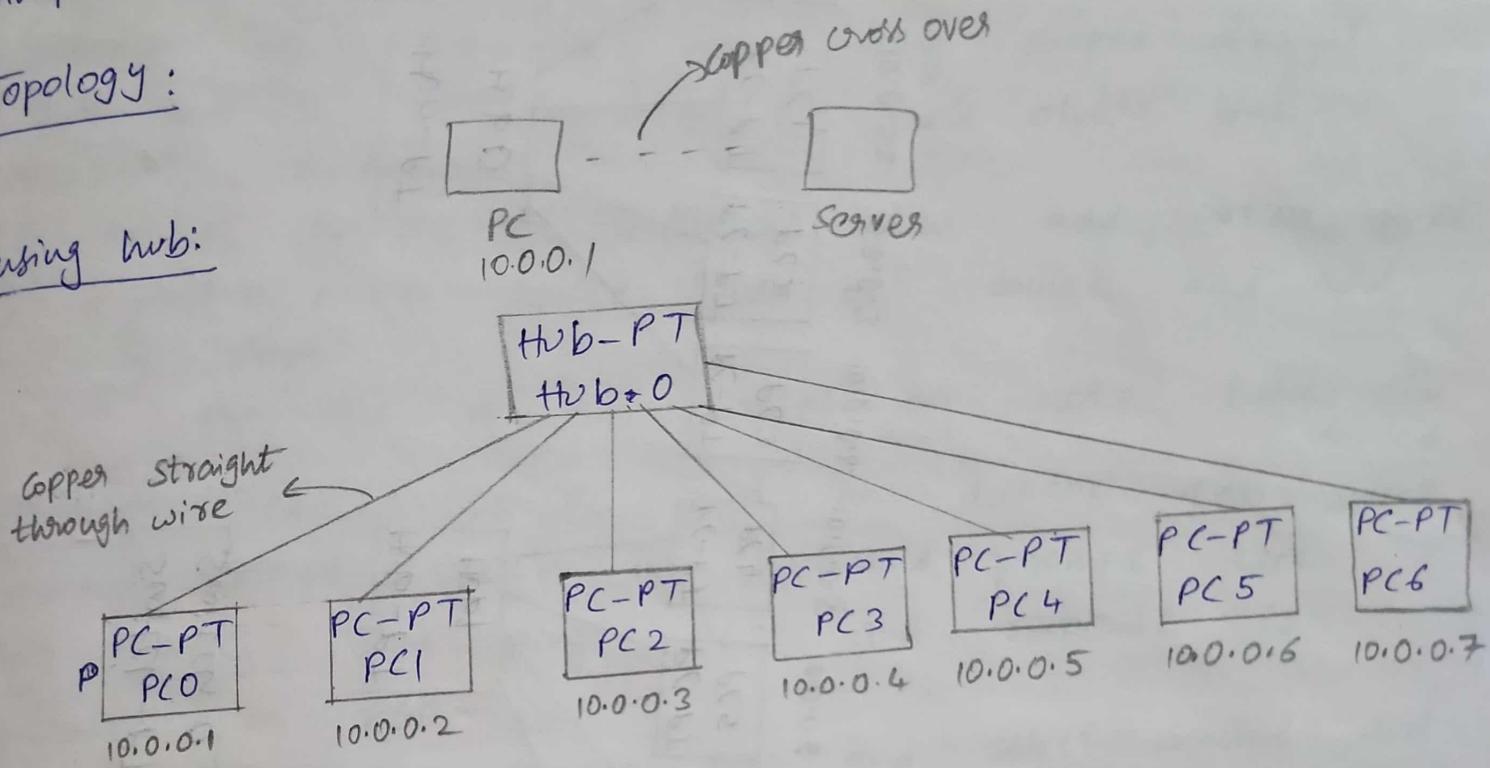
## Lab - week - 1

Aim: Creating simple PDU simple hub

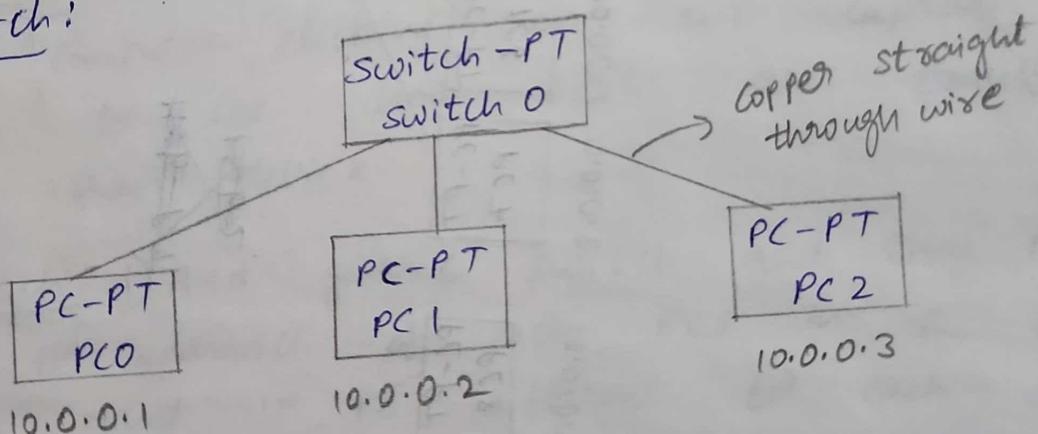
a topology and simulate sending a from source to destination using and switch as connecting domain.

Topology:

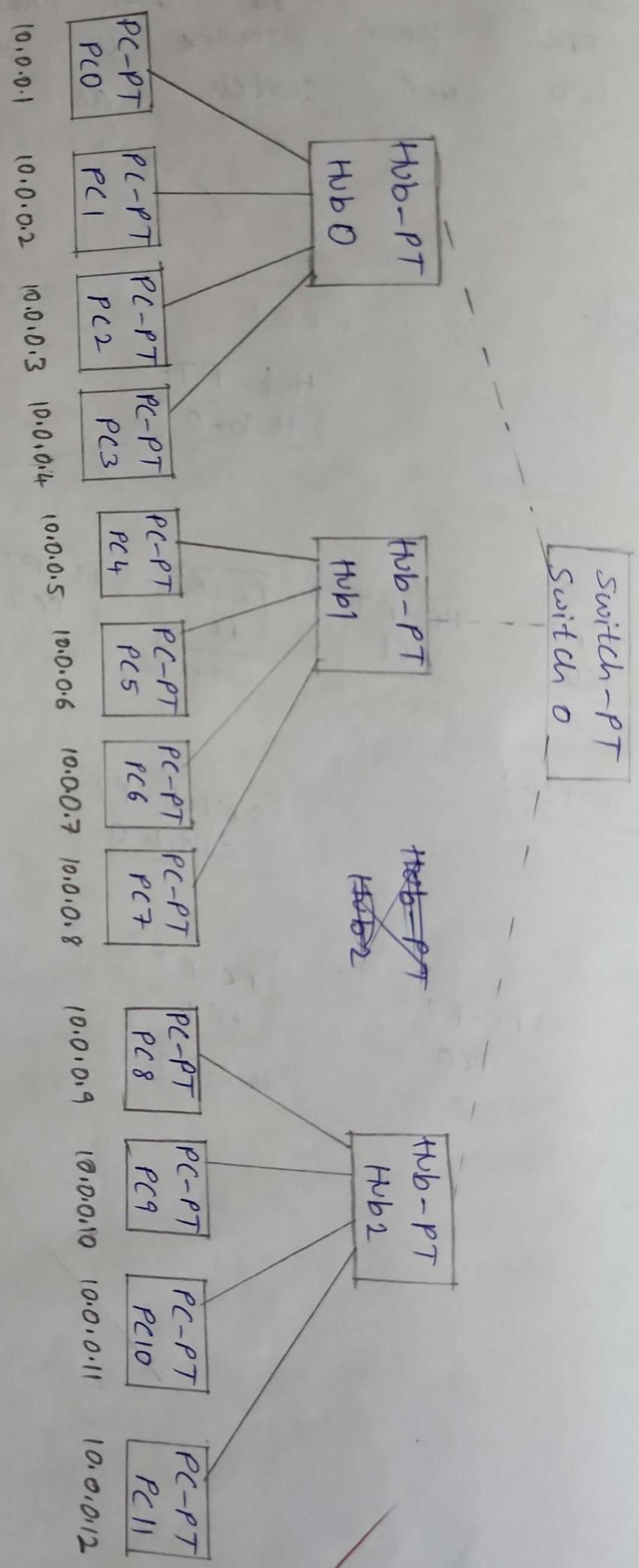
using hub:



using switch:



# Hybrid (using hubs and switches)



### Procedure:

- Using Hub:
- i) Add generic hub and seven PCs to workspace
  - ii) Configure the IP address of each PC in configuration tab. Ensure that IP is different for each device.
  - iii) Connect all PC's to hub using copper straight wire.
  - iv) Hub and PC is connected to each other's fast ethernet connection.
  - v) If no. of ports is insufficient then add extra port by clicking on device. Turn off device and add necessary ports.
  - vi) Write the IP's of all devices in note below the device.

Real time: Select source PC and in desktop tab, select command prompt option. In command prompt type ping 10.0.0.3. This pings PC2 and response is generated in PC0.

Simulation time: Select simple PDU and select source and destination computer. Clicking on auto capture option allows us to see how ports are transferred to and from device.

- Using switch:
- i) Add generic switch and then PCs to workspace.
  - ii) Configure IP addresses of each PC's in the configurations tab. Ensure that IP is different for each device.
  - iii) Connect all PCs to switch using copper straight through wire.
  - iv) If no. of ports are insufficient then add extra ports by clicking on device. Turn off device and add necessary ports.
  - v) Write IP's of all devices in note below the device.

Real time: Select source PC and in the desktop tab. Select command prompt option. In command prompt option, Ping destination PC by specifying its IP.

Simulation time: Select simple PDU and select source and destination Computer. clicking on auto Capture option allows us to see how packets are transferred.

Hybrid mode: i) Add a switch, 3 hubs and 12 PC's to workspace.  
ii) Connect three hubs to switch and 4 PC's to each of the hubs using copper cross over and copper straight through wires respectively.

iii) Configure the IP of each of the PC in configure and add a note below each PC containing IP addresses

Real time mode: Select PC you want to send packet from and open its command prompt. specify destination PC by specifying its IP address. A response is sent by destination PC to source PC.

Simulation mode: Add a simple PDU by selecting the pair of PC and click on auto capture from right panel.

Observation:

→ Hub:

Learning outcomes: i) when source sends a packet in network the hub ~~sends~~ source the packet and ends broadcast over the network, i.e., it sends data to all the end devices in network and node where it matches with the specified address accepts the packet and acknowledge it. Remaining nodes ignore the message.

ii) Comm'n. b/w hub and end devices is established through copper straight through wire as they belong to different layers.

iii) No. of ports can be added if needed by clicking on the device and adding the necessary ports.

Result: PC > ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : byte = 32 time = 0ms

Reply from 10.0.0.3 : byte = 32 time = 0ms

Reply from 10.0.0.3 : byte = 32 time = 0ms

Reply from 10.0.0.3 : byte = 32 time = 0ms

ping statistics for 10.0.0.3

packet sent = 4, received = 4, lost = 0

→ Switches:

Learning outcomes: i) when source device sends a message to the switch once a connection is established, which takes some time called learning time, the switch receives the packet. It initially broadcasts the packet to all connected devices to locate the destination. Once the destination is located the message is sent only to that device.

ii) Connection between the switch and end device is established using copper straight though as they belong to different network layers.

iii) No. of ports can be added if needed by clicking on device and adding the necessary ports.

Result:

PC > ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : bytes = 32 time = 0 ms

Reply from 10.0.0.3 : bytes = 32 time = 0 ms

Reply from 10.0.0.3 : bytes = 32 time = 0 ms

Reply from 10.0.0.3 : bytes = 32 time = 0 ms

~~Ping statistics for 10.0.0.3~~

Packet: Sent = 4, Received = 4, Lost = 0

→ Hybrid Mode:

Learning outcomes: i) switch and hub are connected through copper cross over as they belong to the same network layer but PC and hubs are connected through copper straight through as they belong to different network layers.

- ii) message from source PC to destination is sent through the hub which then sends to all its connected PCs and the switch. The switch then sends the message to all its connected PC. The destination PC acknowledges that it has received the message by sending a acknowledged back to the source PC.
- iii) The no. of ports can be added if needed by clicking on device and adding the necessary ports.

Result

PC > Ping 10.0.0.6

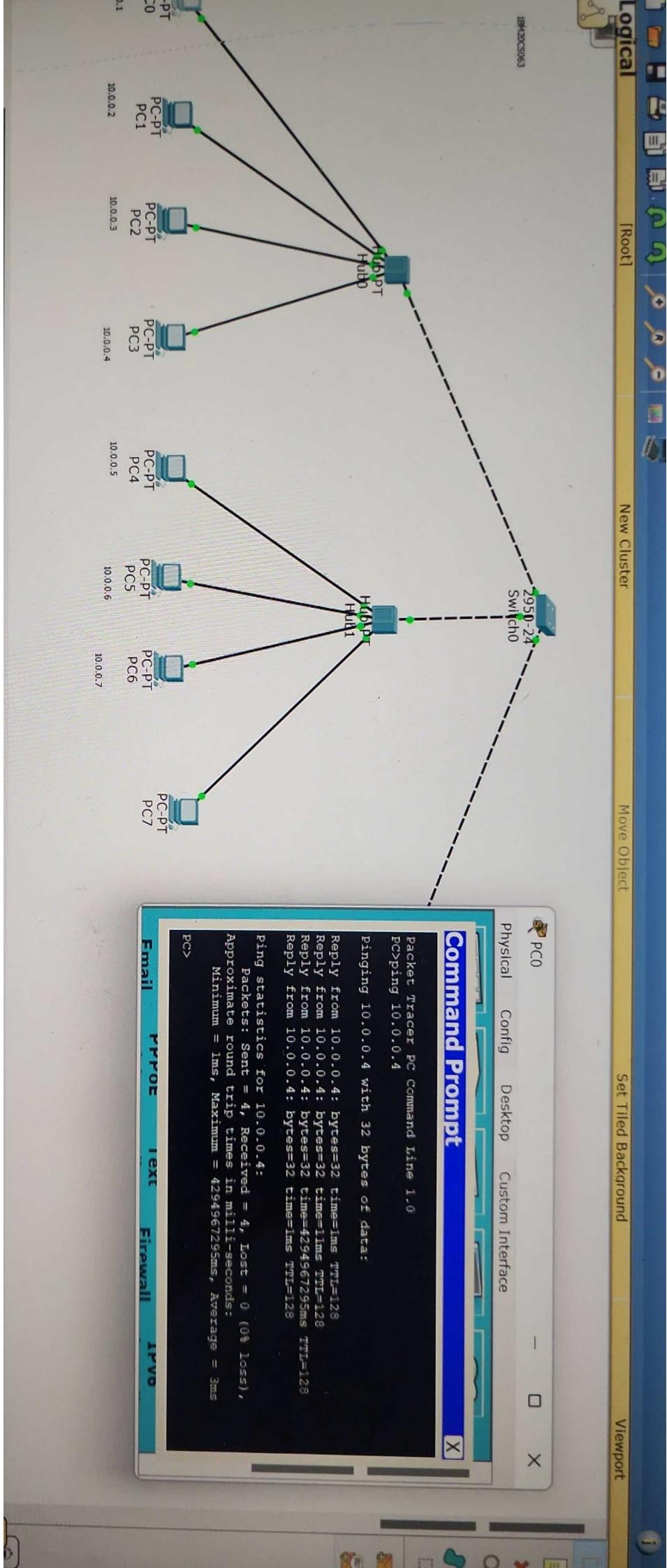
Pinging 10.0.0.6 with 32bytes of data

Reply from 10.0.0.6 bytes = 32

Ping statistics for 10.0.0.4

"Details of numbers of packets send and received".

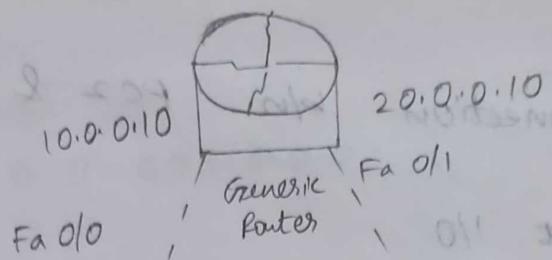
N  
17/11/22



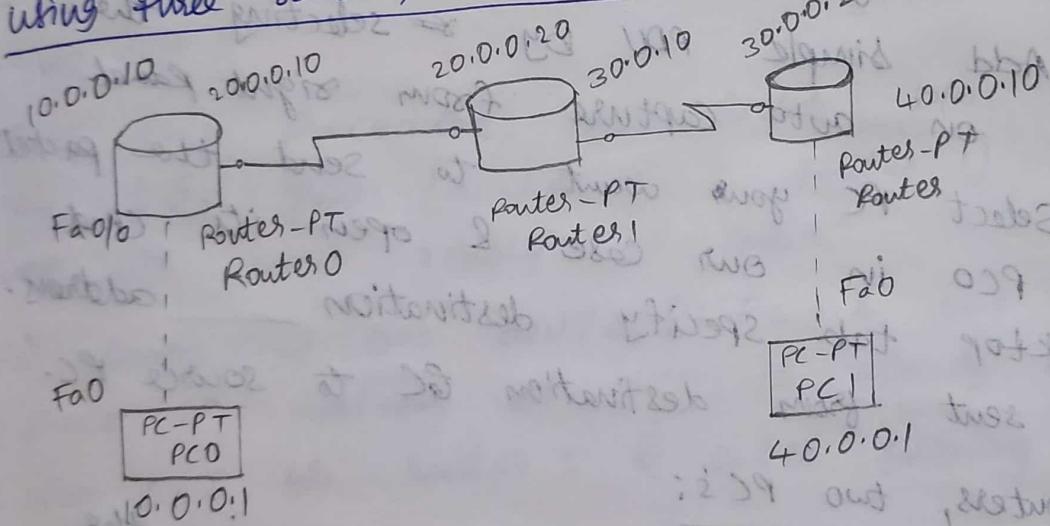
Lab: week 2

Aim: Configuring IP addresses to routers in packet tracer to explore following messages: Ping responses, destination unreachable, request timed out, reply.

Topology: using single router, 2 PC's



using three routers, two PC's:



Procedure:

- using single router, two PC's & add two generic PC's in your workspace.
- i) place a generic router
  - ii) connect router & PCs using copper cross over wire
  - iii) configure IP addresses of each PC & in the configuration tab under settings set gateways for both PCs to router
  - iv) click on generic router and go to CLI tab. enter the following commands to set up connection b/w PC (1) and generic router through gateway (10.0.0.10)

→ No  
 enable  
 # config +  
 (Config) # interface fast ethernet 0/0.  
 (Config-if) # ip address 10.0.0.10 255.0.0.0  
 # no shut  
 # exit  
 Now to set up connection b/n PC2 & router through  
 gateway 20.0.0.10  
 # interface fastethernet 1/0  
 # IP address 20.0.0.10 255.0.0.0  
 # no shut  
 # exit

Once we enter no shut both times the amber light  
 b/n the PC & router turns green indicating that  
 two devices are rarely for use.  
Simulation mode: Add simple PDU by selecting the  
 PCs and click on auto capture from right panel.  
Real time mode: Select PC you want to send the packet  
 from which is PC0 in our case & open its cmd  
 prompt from desktop tab. Specify destination \_\_\_\_\_ address.  
 A response is sent from destination PC to source PC.

→ using three routers, two PCs:

- i) Place 3 generic routers and 2 generic PCs in the workspace.
- ii) Place note for each device [PC & router] and specify IP address.
- iii) Connect routers & PC using copper cross over
- iv) Connect routers using serial \_\_\_\_\_
- v) Click on each PC, go to the configure tab. Set the IP address & subnet mask in fastethernet-
- vi) Next click on settings in config tab. Set gateway on IP address of the next router [eg 10.0.0.10]

vii) IP address of PC and its gateway address should belong to the same network.

For connecting two routers:

click on Router 0. Go to CLI and enter following commands

- no enable
- config t
- interface serial 2/0
- IP address 20.0.0.10 255.0.0.0
- no shut

click on router 1, open CLI & enter following commands.

- no enable
- config t
- interface serial 2/0
- IP address 20.0.0.20 255.0.0.0
- no shut

After this procedure, red lights b/n 2 routers will now turn green [router 0 & router 1] indicating that they are now ready for comm.

For connecting two devices [1 PC & one router]  
i) since IP address of PC is already configured, go to router.

ii) open CLI for router 0 & enter following commands.

- no enable
- config t
- interface fastethernet 0/0
- IP address 10.0.0.10 255.0.0.0
- no shut

Red light turn green, which means ready for comm.

Teaching Router 0 of network 30:

- no
- enable
- config t
- interface serial 2/0
- ip route 30.0.0.0 255.0.0.0 20.0.0.20
- exit
- show ip route

Teaching Router 0 of network 40:

- no
- enable
- config t
- interface serial 2/0
- ip route 40.0.0.0 255.0.0.0 20.0.0.20
- exit
- show ip route

Similarly, repeat for routers 1 & 2.

Simulation mode: Add simple PDU by ~~extent~~ selecting PC 2 & click on auto capture from right panel.

Realtime mode: Select PC 0 & go to its cmd prompt & ping router 0. Once message has been sent successfully repeat this with routers 1 & 2 as well. Finally, ping PC 1.

#### Observation:

#### Learning outcomes

1 Router: when PC 0 pings PC 1 for first time, we get packet as request timed out.

Now, if we ping PC 1 again from PC 0 we get all 4 packets without any loss. Now reverse. Pinging from PC 0 to PC 1 will also not lead to any loss, all packets are acknowledged.

- 3 routers: Before training routers, we get results as destination not unreachable. After training routers, we get clear statistics the result.

Result: using 1 router, two PC's.

C:\> 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 <1ms TTL=127.

Reply from 20.0.0.1: bytes=32 time <1ms TTL=127.

Reply from 20.0.0.1: bytes=32 time <1ms TTL=127.

Ping statistics for 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1: bytes=32 time <1ms TTL=127

Ping statistics for 20.0.0.1:

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

- using 3 routers, 2 PC's:

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

[3 more times]

Ping statistics for 40.0.0.1

Packets: sent = 4, received = 0, lost = 4 (100% loss)

PC> ping 10.0.0.10 with 32 bytes of data.

Reply from 10.0.0.10: bytes=32 time = 0ms TTL=255

[3 more times]

Ping statistics for 10.0.0.10

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

3) PC > Ping 20.0.0.10 with 32 bytes of data  
Reply from 20.0.0.10: bytes=32 time=1ms TTL=255  
[3 more times]

Ping statistics for 20.0.0.10:

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

4) PC > ping 30.0.0.10

Pinging 30.0.0.10 with 32 bytes of data

Reply from 30.0.0.10: bytes=32 time=1ms

[3 more times]

Ping statistics for 30.0.0.10:

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

5) 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=10ms TTL=125

" " " : " = 32, " = 13ms TTL=125  
" (and 119.0) " : " = 32, " = 8ms TTL=125

Ping statistics for 40.0.0.1:

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

6) PC > ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Reply from 40.0.0.1: bytes=32 time=24ms TTL=125

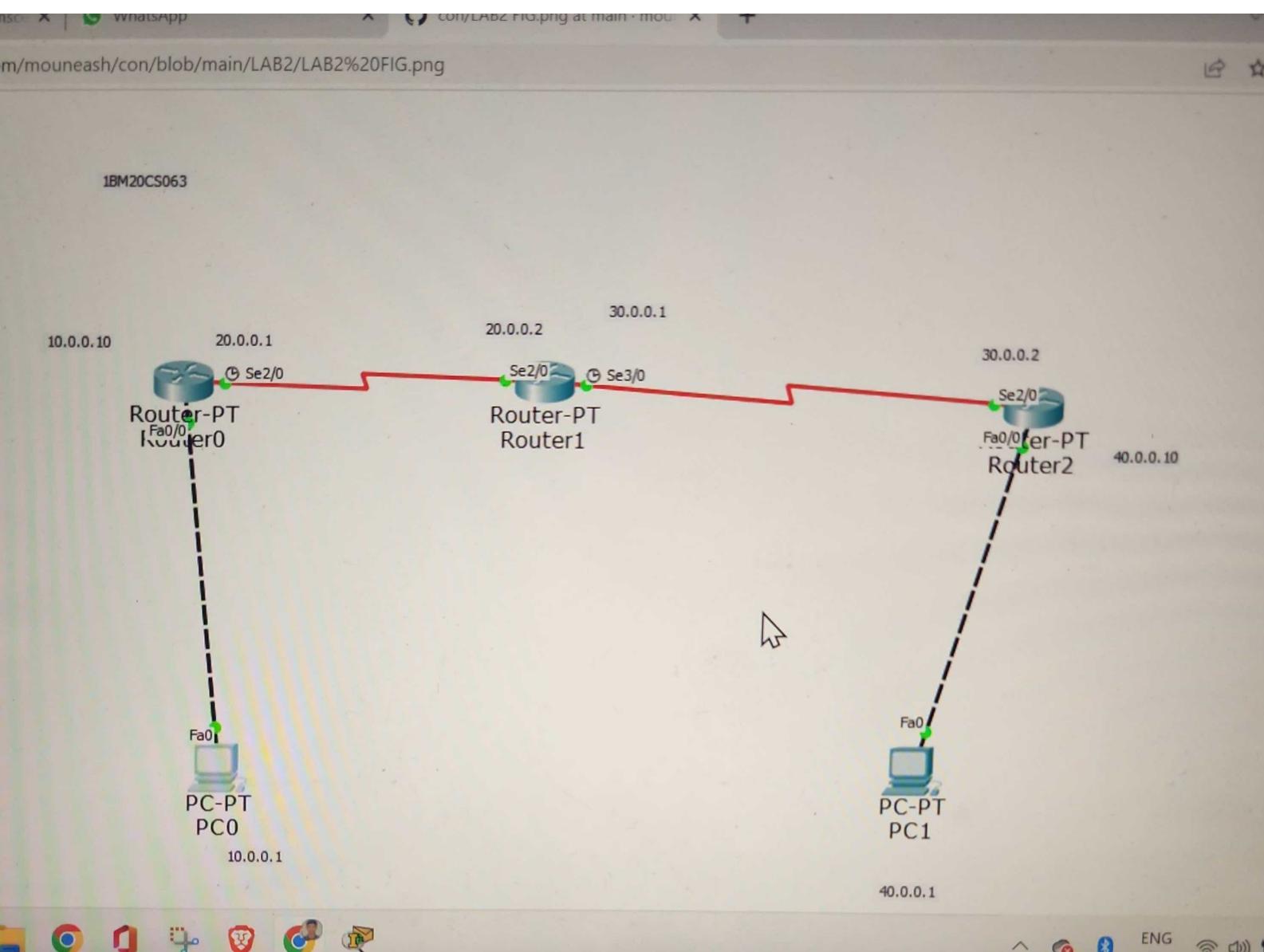
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125

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

Ping statistics for 40.0.0.1:

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

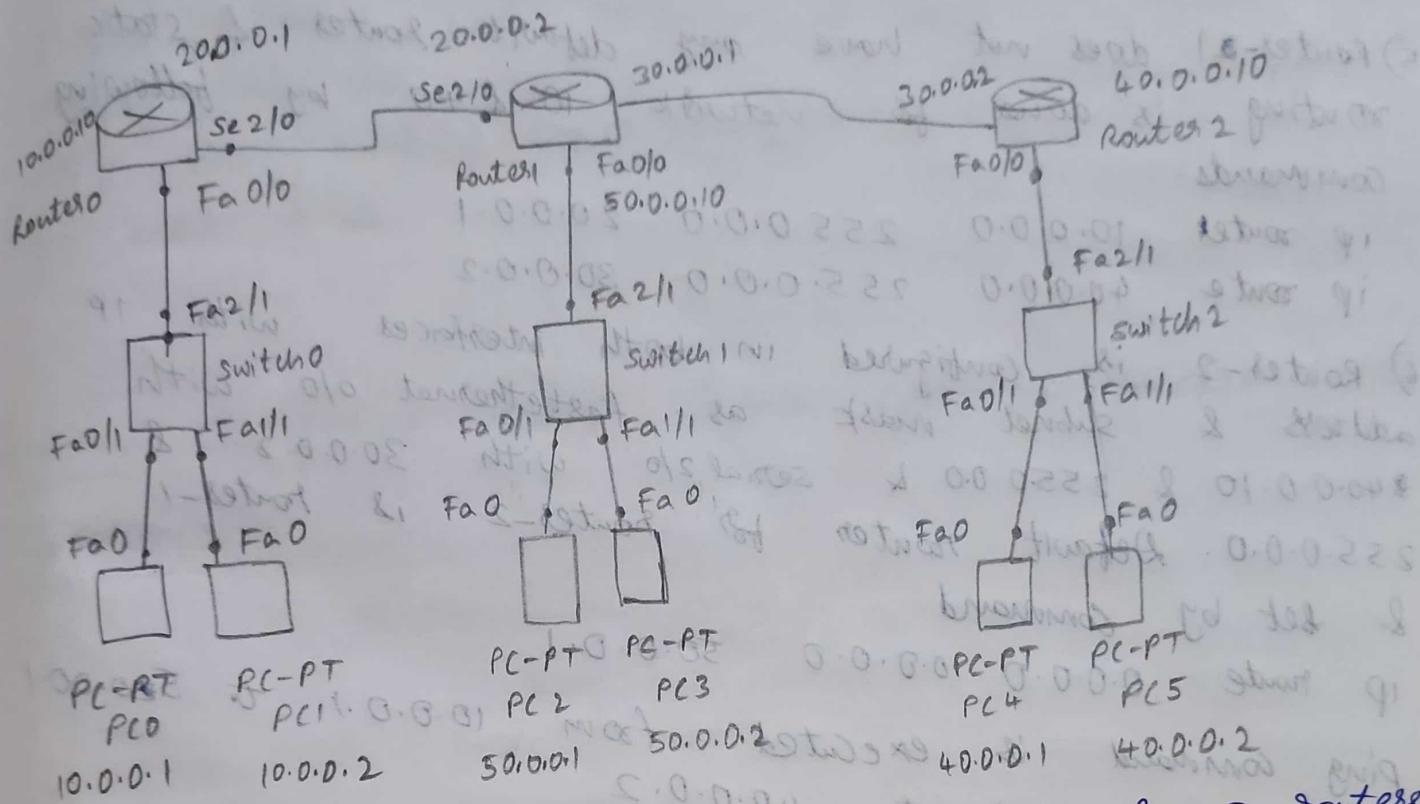
✓  
24 / 1 / 2<sup>2</sup>



25/11/22

3) Aim: Configuring default route to router

Topology:



Procedure: 1) First, Put 6 PC's, 3 switches & 3 routers

& Connect two PC's to each switch with copper straight through wire and each switch is connected to one router with copper straight through wire, & three routers are connected by serial cables and nodes are placed for all devices & networks.

2) A PC is clicked to set attributes to PC & each PC has 3 attributes which are IP address, subnet mask & gateway & all three are set according to nodes placed. This process is done for all PCs.

3) For router-0, configurations are done for both interfaces - fast ethers & subnet mask are set for both interfaces - fast ethers net0/0 as 10.0.0.10 & 255.0.0.0 & serial 2/0 as 40.0.0.1 and 255.0.0.0. Router-0 is default router for router-0 and 255.0.0.0. Router-0 is default router for router-0 & this is done by command ip route 0.0.0.0 0.0.0.0 20.0.0.2.

4) For Router-1, IP address & Subnet mask are set to all three interfaces - fastethernet 0/0 as 50.0.0.10 & 255.0.0.0 & serial 2/0 as 20.0.0.2 & 255.0.0.0 & serial 3/0 as 30.0.0.1 & 255.0.0.0.

5) Router-1 does not have any default router & static routing - is done for network 10 & 40 by following commands.

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

6) Router-2 is configured in both interfaces with IP address & subnet mask as fastethernet 0/0 with 30.0.0.10 & 255.0.0.0 & serial 2/0 with 30.0.0.2 & 255.0.0.0. Default router for Router-2 is Router-1 & set by command

ip route 0.0.0.0 0.0.0.0 50.0.0.1

Ping command is executed from 10.0.0.1 to 50.0.0.1 & from 10.0.0.1 to 40.0.0.2

#### Observations:

#### Learning Outcomes:

- One router can't have 2 default routers
- Default router for first router is middle router as any packets which have to be delivered will go to middle router.
- Default router for right router is middle router for the same reason
- Middle router doesn't have any default router as if one of the routers is made default then there is a chance that packets which are to be sent to switch may send to router.

#### Result:

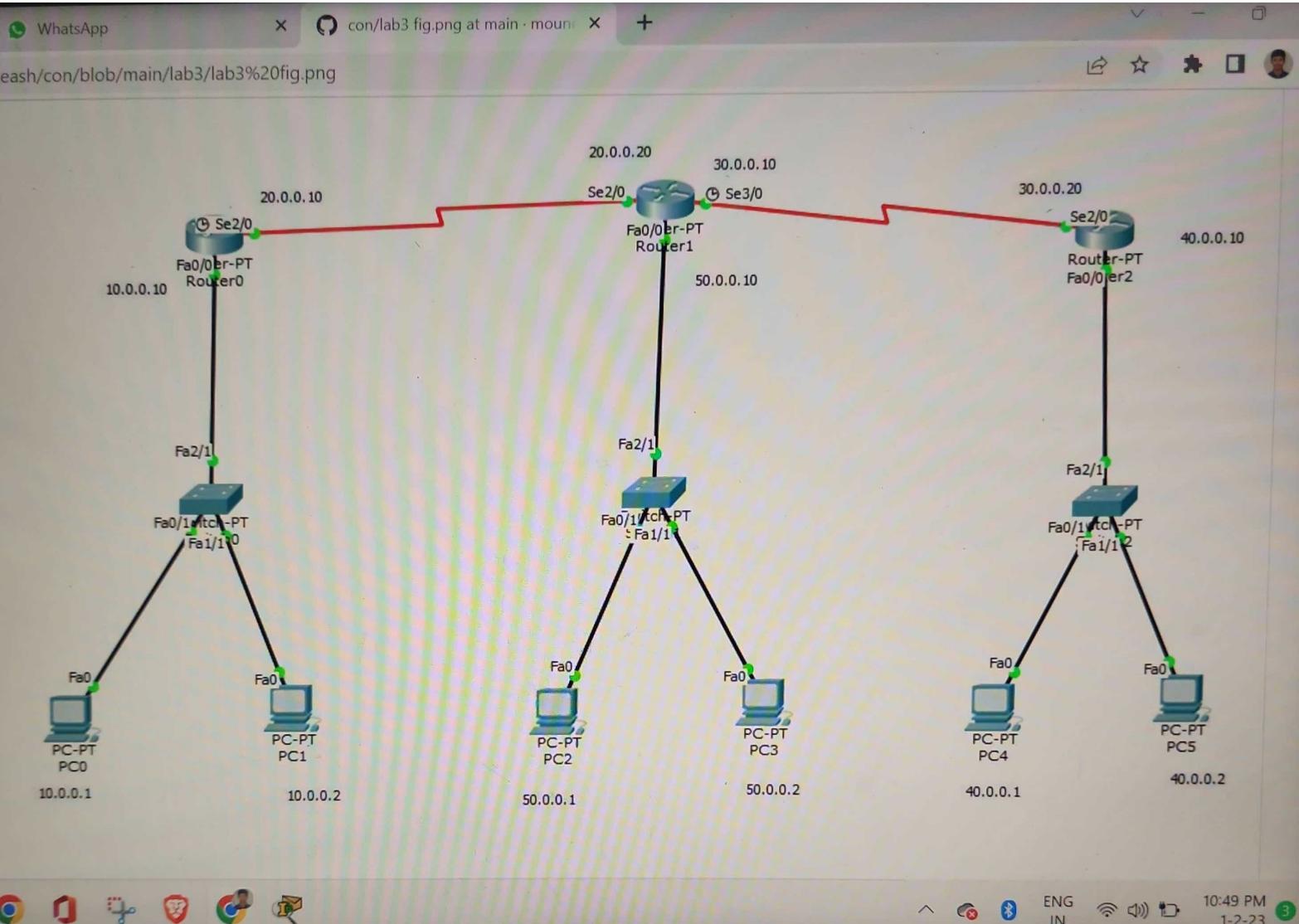
Ping to 50.0.0.1

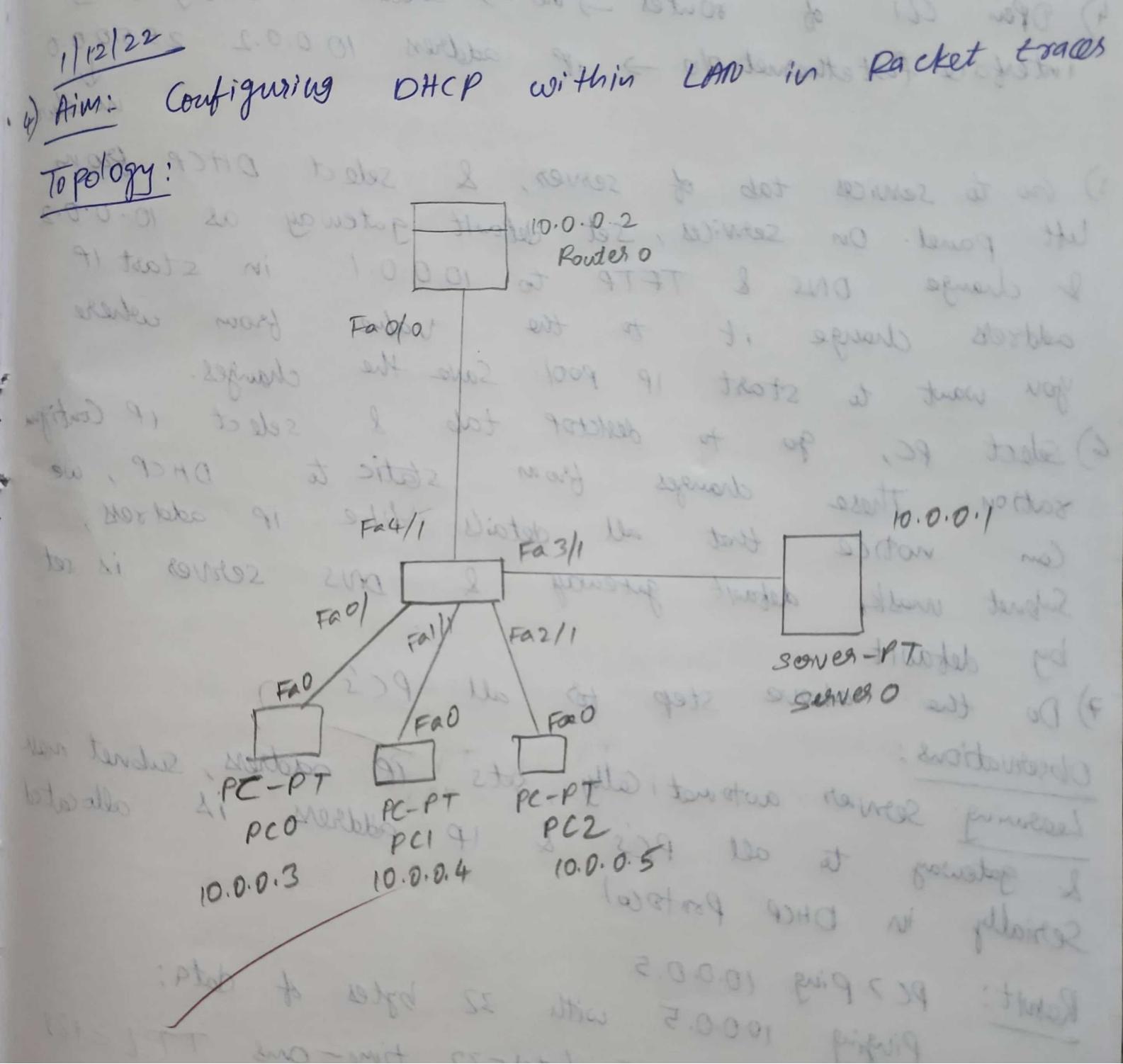
Pinging to 50.0.0.1 with 32 bytes of data

Request timed out

Reply from 50.0.0.1: bytes = 32, time = 1ms, TTL = 126  
↙ 2 more times

Ping to 40.0.0.2  
pinging 40.0.0.2 with 32 bytes of data  
Request timed out  
Reply from 40.0.0.2: bytes = 32, time = 4ms, TTL = 125  
↙ 2 more times





Procedure: 1) Place generic router, generic switch, 1 Server & 3 PCs into the work station & connect them as: Router to switch & switch to all 3 PCs & server

2) Open Server Config tab, set its IP address & subnet mask, from settings tab set its gateway  
3) For server IP address is 10.0.0.1, gateway is 10.0.0.2  
4) Open CLI of router  $\rightarrow$  no  $\rightarrow$  enable  $\rightarrow$  config t  $\rightarrow$  interface fastethernet 0/0  $\rightarrow$  ip address 10.0.0.2 255.0.0.0

5) Go to services tab of server, & select DHCP from left panel. On services, set default gateway as 10.0.0.2 & change DNS & TFTP to 10.0.0.1 in start IP address change it to the value from where you want to start IP pool. Save the changes.

6) Select PC, go to desktop tab & select IP Configuration. These changes from static to DHCP, we can notice that all details like IP address, Subnet mask, default gateway & DNS server is set by default.

7) Do the above step for all PCs?

Observations:  
Learning: Server automatically sets IP address, subnet mask & gateway to all PCs & IP address is allocated serially in DHCP protocol.

Result: PC > ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

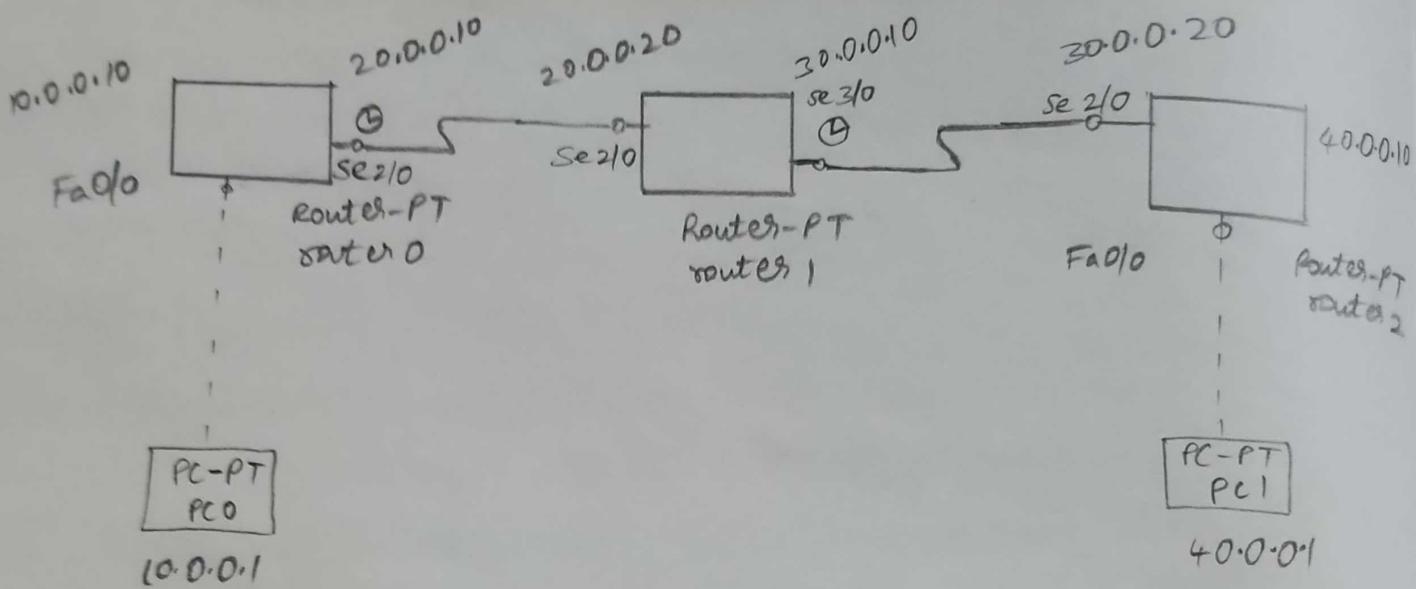
Reply from 10.0.0.5: bytes=32 time=ans TTL=128

3 ms times

N 22



8/12/22  
5. Aim: Configuring RIP routing protocol in routers.  
BKR: Topology:



- Procedure:
- 1) Place 3 generic routers & 2 generic PCs in workspace
  - 2) Connect routers & two PCs using copper cross wires
  - 3) Connect the routers using serial DCE with clock symbol
  - 4) Place notes near PCs & routers
  - 5) Select IP address of PC0 & PC1 as well as their subnet mask & default gateway.
  - 6) Go to CLI of router 0 & enter following commands

- enable
- Config t
- interface fastether net 0/0
- IP address 10.0.0.10 255.0.0.0
- no shut

Connection should show green. Repeat for PC1 & Router 2 as well.

(Connection should be established between R1 & R2)

Now, open CLI of router 0 & enter the following commands:

- enable
- config t
- interface serial 2/0
- encapsulation PPP
- clock rate 64000
- no shutdown

Router 0 has 2 serial ports

Serial 0/0 is Router 0's serial port 0

Serial 0/1 is Router 0's serial port 1

Serial 0/2 is Router 0's serial port 2

Serial 0/3 is Router 0's serial port 3

open CLI of router 1

- enable
- config t
- interface serial 2/0
- ip address 20.0.20.20 255.0.0.0
- encapsulation PPP
- no shutdown

Connection will turn green:

open CLI of router 1 again:

- enable
- config t
- interface serial 3/0
- ip address 30.0.0.10 255.0.0.0
- encapsulation PPP
- clock rate 64000
- no shutdown

Open CLI for router 2:

- enable
- config t
- interface serial 2/0
- ip address 30.0.0.20 255.0.0.0
- encapsulation PPP
- no shutdown

Now, open CLI of router 0 & enter following commands.

```
(config)# router rip  
(config-router) # network 10.0.0.0  
config-router) # network 20.0.0.0  
config-router) # exit  
show ip route
```

Similarly repeat for router 1 & router 2 with networks 20 & 30 and 30 & 40.

Simulation mode: Add simple PDU by selecting the PCs & click on auto capture from right panel.

Realtime mode: Simple PC go to command prompt and select destination address 40.0.0.1

#### Observation:

Leaving outcome: Routing information protocol is a protocol that routers use to exchange information about topology among the network.

It is used when in place of static IP routing because with help of rip protocol routing it becomes easy when large scale of routers is present.

#### Result:

PC > ping 40.0.0.1

Request timed out

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

Reply from 2 times

Ping statistics for 40.0.0.1:

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

2) PC > ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.1: bytes=32 time=19ms TTL=125

[ 3 mode times

ping statistics for 40.0.0.1:

packets: sent = 4, Received = 4, lost = 0 (0% loss)

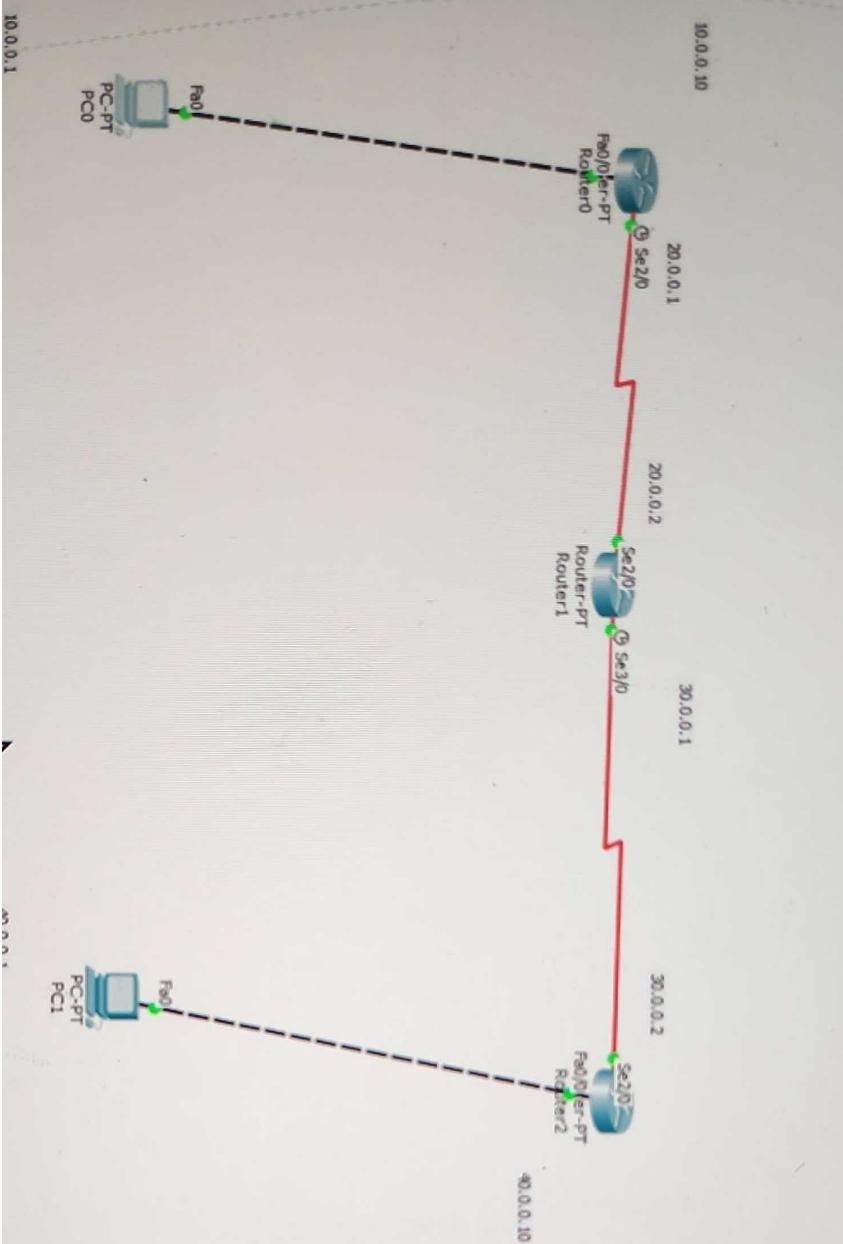
Approximate round trip times in milli-seconds:

Minimum = 2ms, maximum = 19ms, Average = 10ms





18420CS063



PC0

Physical Config Desktop Custom Interface

## Command Prompt

```
pinging 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milliseconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125
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=10ms TTL=125

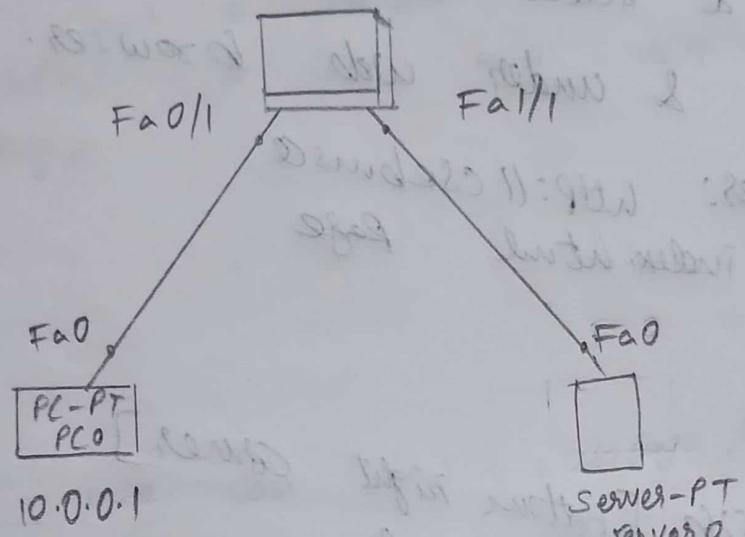
Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milliseconds:
        Minimum = 7ms, Maximum = 14ms, Average = 9ms
```

10.0.0.1

15/12/22 6) DNS

Aim: Demonstration of web server & DNS using packet tracer.

Topology:



Procedure:

- i) place generic PC, switch and server in workspace.
- ii) place note of ip address under PC as 10.0.0.1 & server as 10.0.0.2
- iii) connect PC to switch & switch to server
- iv) set ip address of PC as 10.0.0.1 & server as 10.0.0.2

- click on PC
- go to desktop tab
- open webserver & enter ip address as 10.0.0.2
- Now, click on server
- Go to server.

under http, go to index.html. click on edit & make some changes. Do overwrite it.

- Now go to PC & under web browser: →  
In the address, give ip address as 10.0.0.2.  
click on go. It displays the changes made in  
index.html
- Now go to server again  
under service, go to DNS  
configure name as: cebusce  
address: 10.0.0.2  
click on add & select & save.
- Now, go to PC & under web browser:  
In url : enter: http://cebusce  
It displays index.html page
- click on server  
HTTP  
click on new file [bottom right corner]  
Enter file name as cv.html  
<html>  
<head><h1> CSE student details </h1></head>  
<table>  
<tr>  
<th> Name </th>  
<th> Branch </th>  
<th> Class </th>

```

<th> USN </th>
<th> DOB </th>
</tr>
<tr>
    <td> Jayam Mooneash </td>
    <td> CSE </td>
    <td> 5-B </td>
    <td> IBM20CS063 </td>
    <td> 27-06-2001 </td>
</tr>

```

<table>  
 ↘ Now, go to PC & under web browser:

i) In url, enter: csebmsce

It will show a newlink in the index.html page  
 as csebmsce. & it will display the student  
 details page

Observation: we can view webpage, when we type  
 'csebmsce' in browser because 10.0.0.2 address is  
 linked to name 'csebmsce' according to DNS.  
 - Since it's difficult for users to remember IP  
 addresses, linking is required i.e., IP is linked  
 with domain name.

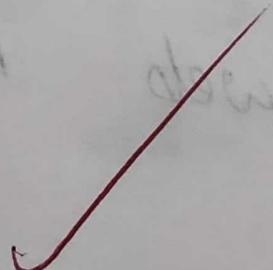
Result:

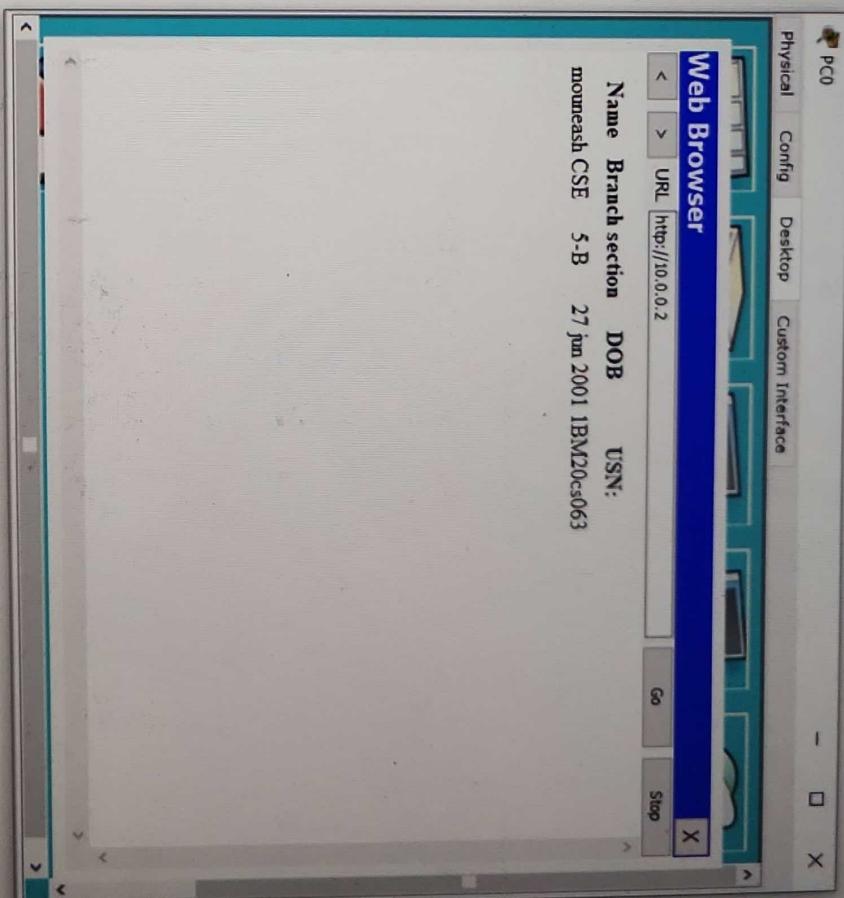
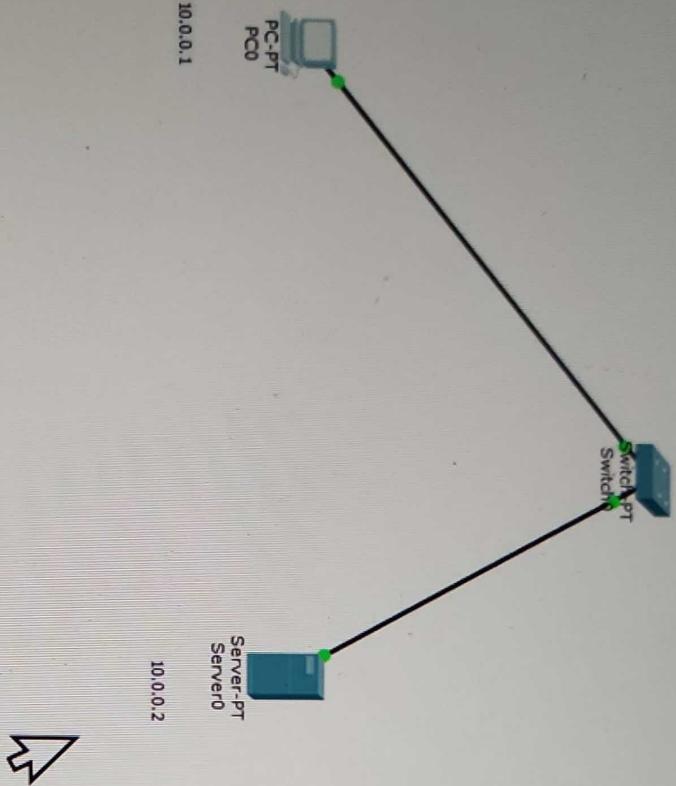
web browser  
 ↘ URL http://csebmsce  
 BMSCE packet tracer  
 welcome to BMSCE packet tracer  
 Quick links

Qwall  
 29-12-2022

<http://110.8.8.105/cv.html>

<u>NAME</u>	KRIPAKAR
NAME	JAYAM MOUNESH
AGE	20
DOB	27/06/2001
College	BMSCE
Branch	CSE
Year	3rd





Cycle - 2

① WAP for error detecting code using CRC-CCJ  
TT(16 bits)

```

#include <stdio.h>
#include <string.h>
#define N strlen(g)

char t[28], cs[28], g[10];
int are, c;

void xorfunction() {
    for (c = 1; c < N; c++)
        cs[c] = ((cs[c] == g[c]) ? '0' : '1');
}

void ccc() {
    for (e = 0; e < N; e++)
        cs[e] = t[e];

    do {
        if (cs[0] == '1')
            xorfunction();

        for (c = 0; c < N - 1; c++)
            cs[c] = cs[c + 1];

        cs[c] = t[e++];
    } while (e <= a + N - 1);
}

int main() {
    printf("In Enter data: ");
    scanf("%s", t);
    printf("In Enter generating Polynomial: "); X
}

```

```
#include <stdio.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>

void main(){
    int i, j, keylen, msglen;
    char input[100], key[30], temp[30], quot[100], rem[30],
        key1[30];

    printf("Enter data: ");
    gets(input);
    printf("Enter k(x): ");
    gets(key);
    keylen = strlen(key);
    msglen = strlen(input);
    strcpy(key1, key);
    for(i=0; i<keylen-1; i++){
        input[msglen+i] = '0'
    }
    for(i=0; i<keylen; i++)
        temp[i] = input[i];
    for(i=0; i<msglen; i++)
        quot[i] = temp[i];
    if(quot[i] == '0')
        for(j=0; j<keylen; j++)
            key[j] = '0';
    else
        for(j=0; j<keylen; j++)
            key[j] = key1[j];
}
```

```
for (j = keylen - 1; j > 0; j--) {  
    if (temp[j] == key[j])  
        rem[j-1] = '0'; else  
        rem[j-1] = '1';
```

```
}
```

```
rem[keylen - 1] = input[i + keylen];
```

```
strcpy(temp, rem);
```

```
}
```

```
strcpy(rem, temp);
```

```
printf("In Quotient is ");
```

```
for (i = 0; i < msglen; i++)
```

```
printf("%c", quot[i]);
```

```
printf("In Remainder is ");
```

```
for (i = 0; i < keylen - 1; i++)
```

```
printf("%c", rem[i]);
```

```
printf("In Modified data is ");
```

```
for (i = 0; i < msglen; i++)
```

```
printf("%c", input[i]);
```

```
for (i = 0; i < keylen - 1; i++)
```

```
printf("%c", rem[i]);
```

```
strcat(input, rem);
```

```
keylen = strlen(key);
```

```
msglen = strlen(input);
```

```
strcpy(key1, key);
```

```
for (i = 0; i < keylen - 1; i++)
```

```
input[msglen + i] = '0';
```

```
for (i = 0; i < keylen; i++)
```

```
temp[i] = input[i];
```

```
for (i = 0; i < msglen; i++) {
```

```
quot[i] = temp[0];
```

```

if (quot[i] == '0')
for (j=0; j < keylen; j++)
    key[j] = '0';
else
    for (j=0; j < keylen; j++)
        key[j] = temp key1[j];
    for (j = keylen - 1; j > 0; j--) '
        if (temp[j-1] == key[j])
            temp[j-1] = '0'
        else
            temp[j-1] = '1';
}
temp[keylen-1] = input[i+keylen];
strcpy(temp, temp);
printf("In remainder is at receiver side ");
for (i=0; i < keylen-1; i++)
    printf("%c", temp[i]);
getch();
}

```

O/P:

Ex-1: Enter data: 1011010101

$DW$

Enter  $g(x)$ : 1010  $\rightarrow$  CRC-16 only

Quotient : 1001000100

Remainder: 000

modified: 1011010101000

Checking for error

Ex-2: Enter data: 1011010101000

$G(x)$ : 1010

Remainder: 000

$D$

$(x^8 + x^{12} + x^5 + 1) \text{ mod } x^{16}$

$x^{16} + x^{12} + x^5 + 1$

$\text{Fixed}$

$(++i; i < n) \geq 1$

$i^8 = [i + \text{new}(i)]$

$R = 0$

$R \neq 0$  Error detected

$(++i; n) \geq 1$

$[0] \text{ result} = [ ]$

Case - 3) Enter input : 1011010101  
Transmitted message - 101101010111110110111010  
Enter received message:  
111111110000000011111111  
Error in data transmission has occurred

✓  
5/1/23

Enter data to be transmitted: 1011010101

Enter the Generating polynomial: 1010

Data padded with n-1 zeros : 1011010101000

CRC or Check value is : 000

Final data to be sent : 1011010101000

Enter the received data: 1011010101001

Data received: 1011010101001

Error detected

5/01/23

2) Leaky Bucket algm  
program for congestion control using leaky bucket algm.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int input = 0, i = 0, bucket_limit = 400, op = 1;
    printf("Bucket limit is 400\n");
    printf("Output rate is 50 mpps\n");
    while(op){
        printf("Enter input: ");
        scanf("%d", &i);
        if(i <= 400 && input <= 400){
            input += i;
            input -= 50;
        }
    }
}
```

```

if ( input <= 400 )
    printf ("qty in bucket : %d\n", input);
}
else if ( input > 400 )
    printf ("Bucket limit Exceeded\n");
else
    printf ("Bucket limit exceeded\n");
    printf ("Press 1 to add input in 0 to end");
    scanf ("%d", &OP);
}
return 0;
}

```

O/P :-

Bucket limit is 400

Rate is 50mbps

Enter input

300

qty in bucket 250

Press 1 to add input, 0 to end

1

300

Bucket limit exceeded

Press 1 to add input 0 to end

1

Enter input

50

qty in bucket 250

12/1/23 Press 1 to add, input 0 to end

Bucket limit is 400  
Rate is 50mbps  
enter the input  
200  
qty in bucket 150  
press 1 to add input again 0 to end  
1  
enter the input  
200  
qty in bucket 300  
press 1 to add input again 0 to end  
1  
enter the input  
500  
Bucket limit Exceeded  
press 1 to add input again 0 to end  
^A

3) 12/01/23 Bellmann Ford algm: program for distance vector algm to find suitable path for transmission

```

#include <iostream>
#include <stdlib.h>
int BellmanFord( int G[20][20], int V, int E, int edge
[20][20] ) {
    int i, u, v, k, distance[20], parent[20], s, flag=1;
    for (i=0; i < V; i++)
        distance[i] = 1000;
    parent[i] = -1;
    cout ("Enter source : ");
    scanf ("%d", &s);
    if (distance[s-1] == 0)
        for (i=0; i < V-1; i++)
            for (k=0; k < E; k++)
                if (distance[u] + G[u][v] < distance[v])
                    distance[v] = distance[u] + G[u][v];
                parent[v] = u;
    }
    for (k=0; k < E; k++)
        if (distance[u] + G[u][v] < distance[v])
            distance[v] = distance[u] + G[u][v];
            parent[v] = u;
    flag = 0;
}
if (flag)
    for (i=0; i < V; i++)
        cout ("vertex " + i + ", distance[i], parent[i]+1);
return flag;

```

EE/10

\* graphs & their operations for warshall : graphs not nullable

```

int main(){
    int v, edge[20][2], G[20][20], i, j, k=0;
    pf("Enter no. of vertices:");
    scanf("%d", &v);
    pf("Enter graph in matrix form: \n");
    for (i=0; i<v; i++)
        for (j=0; j<v; j++) {
            if (scanf("%d", &G[i][j]) != 0)
                edge[k][0] = i, edge[k][1] = j;
            k++;
        }
    if (Bellman_ford(G, v, k, edge))
        printf("InNo negative weight cycle\n");
    else
        printf("In Negative weight cycle exists\n");
    return 0;
}

```

Enter the number the routers(<10): 5

Enter 1 if the corresponding router is adjacent to routerA else enter 99:  
B C D E

Enter matrix:1 1 99 99

Enter 1 if the corresponding router is adjacent to routerB else enter 99:  
A C D E

Enter matrix:1 99 99 99

Enter 1 if the corresponding router is adjacent to routerC else enter 99:  
A B D E

Enter matrix:1 99 1 1

Enter 1 if the corresponding router is adjacent to routerD else enter 99:  
A B C E

Enter matrix:99 99 1 99

Enter 1 if the corresponding router is adjacent to routerE else enter 99:  
A B C D

Enter matrix:99 99 1 99

Router Table entries for router A:-

Destination Router: A B C D E

Outgoing Line: A B C D E

Hop Count: 0 1 1 99 99

Router Table entries for router B:-

Destination Router: A B C D E

Outgoing Line: A B C D E

Hop Count: 1 0 99 99 99

**Router Table entries for router D:-**

**Destination Router: A B C D E**

**Outgoing Line: A B C D E**

**Hop Count: 99 99 1 0 99**

**Router Table entries for router E:-**

**Destination Router: A B C D E**

**Outgoing Line: A B C D E**

**Hop Count: 99 99 1 99 0**

Q) Implement Dijkstras algm to find shortest Path to given topology.

Dijkstras :

```
#include <stdio.h>
#include <conio.h>
#define INFINITY 999
#define MAX 10
void dijkstra(int G[MAX][MAX], int n, int startnode);
int node() {
    int G[MAX][MAX], i, j, n, u;
    printf("Enter no. of vertices");
    scanf("%d", &n);
    printf("In Enter adjacency matrix : ");
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            scanf("%d", &G[i][j]);
        }
    }
    printf("Enter starting node:");
    scanf("%d", &u);
    dijkstra(G, n, u);
    return 0;
}
```

```
Void dijkstra (int G[MAX][MAX], int n, int startnode) {
    int cost[MAX][MAX], distance[MAX], pred[MAX];
    int visited[MAX], count, min distance, nextnode, i, j;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (G[i][j]==0)
                cost[i][j] = INFINITY;
            else
                cost[i][j] = G[i][j];
    for (i=0; i<n; i++) {
        distance[i] = cost[startnode][i];
        pred[i] = startnode;
        visited[i] = 0;
    }
```

```

distance [start node] = 0;
visited [start node] = 1;
Count = 1;

while (Count < n-1) {
    mindistance = INFINITY;
    for (i=0; i<n; i++) {
        if (distance[i] < mindistance && visited[i]) {
            mindistance = distance[i];
            nextnode = i;
        }
    }
    visited [nextnode] = 1;
    for (i=0; i<n; i++) {
        if (!visited[i]) {
            if (mindistance[nextnode[i]] < distance[i]) {
                distance[i] = mindistance + cost [nextnode][i];
                pred [i] = nextnode;
            }
        }
    }
    Count++;
}

for (i=0; i<n; i++) {
    if (i != start node) {
        printf("In Distance of node %d, d = %.d", i, distance[i]);
        j=i;
        do {
            j = pred[i];
            printf(" %.d", j);
        } while (j != startnode);
    }
}

```

O/P : Enter no. of vertices : 4

Enter adjacency matrix

	0	1	2	3
0	0	1	1	0
1	1	0	1	0
2	1	1	0	1
3	1	0	1	0

Enter starting node : 1

Distance of 0 = 1

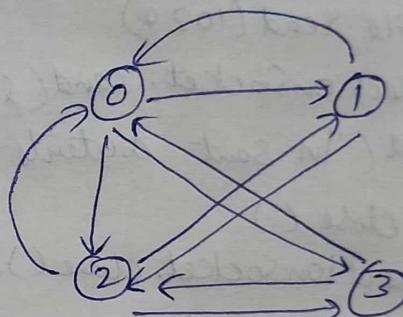
Path = 0 ← 1

Distance of 2 = 1

Path = 2 ← 1

Distance of 3 = 2

Path = 3 ← 0 ← 1



ii) Using TCP/IP sockets, write client server program to make client sending filename and server to send back contents of requested file if present

client.py  
from socket import \*  
servername = '127.0.0.1'  
serverport = 12000

```
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((servername, serverport))
sentence = input("Enter filename: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('From Server: ')
print(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(1)
```

```
while 1:
```

```
    print("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("In sent contents of " + sentence)
```

```
    file.close()
```

```
    connectionSocket.close()
```

O/P :-

Enter filename: serverTCP.py

From Server:

```
connectionSocket, addr = serverSocket.accept()
```

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

```
file = open(sentence, "r")
```

```
l = file.read(1024)
```

```
connectionSocket.send(l.encode())
```

```
print("In sent contents of " + sentence)
```

```
file.close()
```

```
connectionSocket.close()
```

server is ready to receive

sent contents of serverTCP.py

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

```
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"), (ServerName, ServerPort))
```

```
fileContent, serverAddress = clientSocket.recvfrom(2048)
print('In Reply from Server: \n')
print(fileContent.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("Server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
    print('In Sent Contents of ', end=' ')
    print(sentence)
    file.close()
```

Off: Enter file name: ServerUdp.py #client op

Reply from Server :

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("Server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
    print('In Sent Contents of ', end=' ')
    print(sentence)
    file.close()
```