

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

COMPUTER NETWORKS

Submitted by

M UDAYA RAJ (1BM20CS080)

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 **M UDAYA RAJ (1BM20CS080)**, 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 2022-2023. 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. Seema
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index

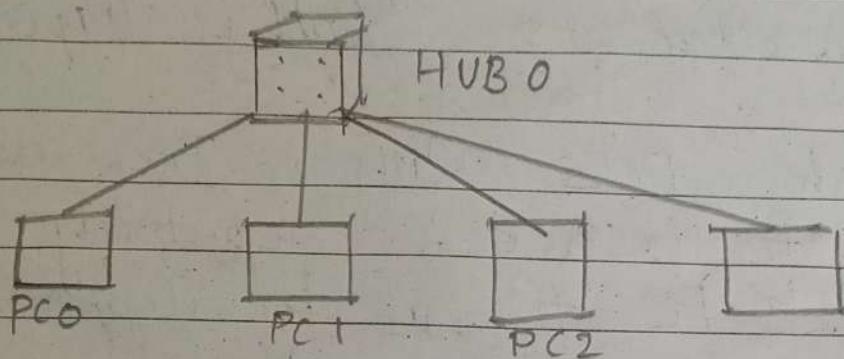
Sl. No.	Date	Experiment Title	
1	10-11-2022	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	1
2	17-11-2022	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply	9
3	24-11-2022	Configuring default route to the Router	22
4	01-12-2022	Configuring DHCP within a LAN in a packet Tracer	28
5	08-12-2022	Configuring RIP Routing Protocol in Routers	32
6	15-12-2022	Demonstration of WEB server and DNS using Packet Tracer	37
7	29-12-2022	Write a program for error detecting code using CRC-CCITT (16-bits).	43
8	12-01-2023	Write a program for distance vector algorithm to find suitable path for transmission.	50
9	12-01-2023	Implement Dijkstra's algorithm to compute the shortest path for a given topology.	54
10	05-01-2023	Write a program for congestion control using Leaky bucket algorithm.	47
11	28-01-2023	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.	58
12	29-01-2023	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.	61

10/11/22

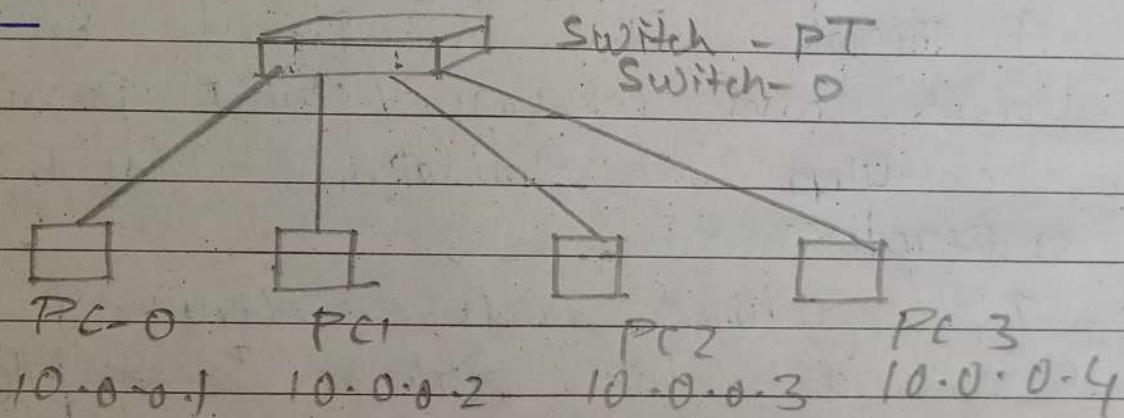
AIM :- Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.

Topology :-

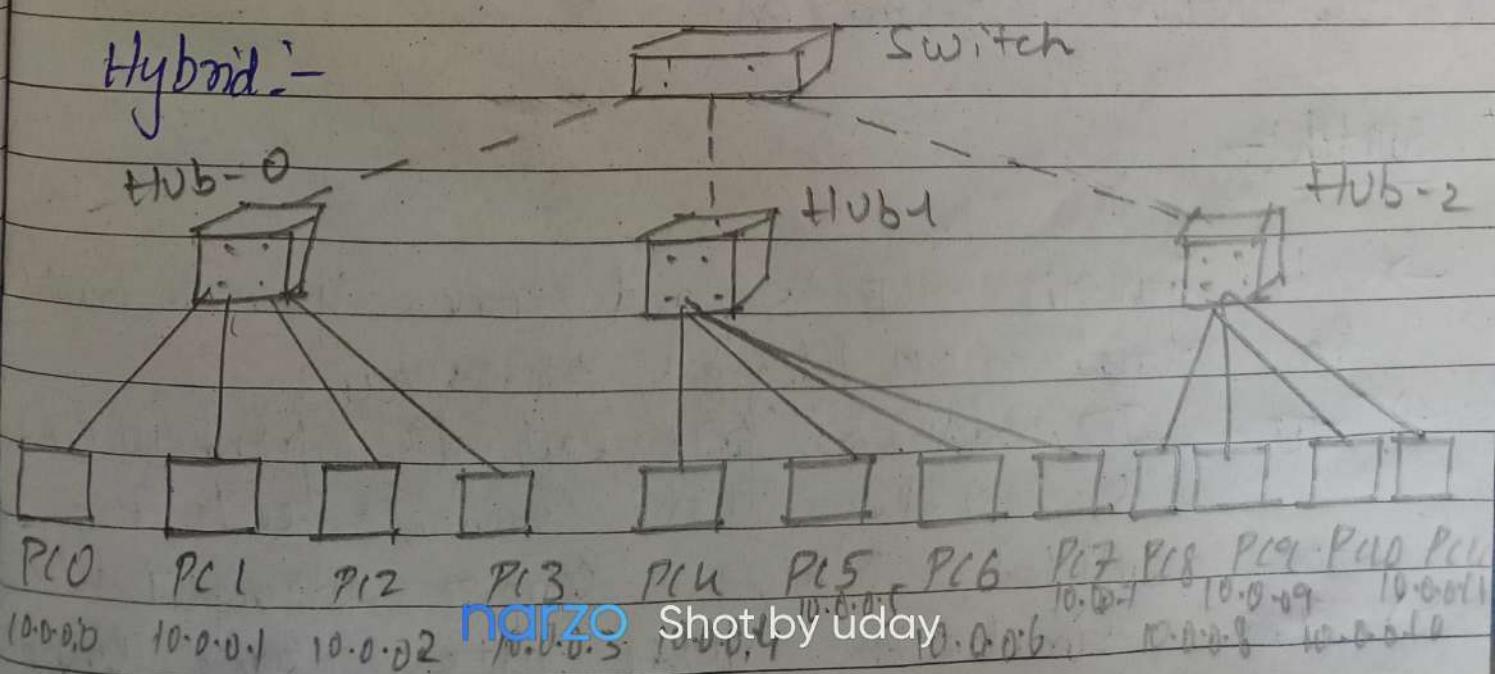
Hub :-



Switch :-



Hybrid :-



Procedure:-

Hub:-

1. place 7 generic PC's and 1 generic hub in logical workspace and all 7 PC's are connected to hub by copper straight wire.
2. set each PC's with IP address from 10.0.0.0 to 10.0.0.6 respectively and connect each PC to hub by copper straight wire.
3. A simple PDU is placed on any 2 devices and message / packet passing can be seen in simulation mode by clicking  autocapture.
4. In realtime mode a command prompt is opened for certain PC & following command is given to transfer message PING destination IP address.

Switch:-

- 4 generic PC's and one generic Switch is placed on logical workspace.
- Set IP address for each PC from 10.0.0.7 to 10.0.0.10 and connect each PC to switch using copper straight wire.
- In simulation mode after placing simple PDU to any 2 PC's click auto capture and packet transfer can be seen.

→ In real-time mode click on any PC and open command prompt and type 'ping or dest-IP' to send message.

* Hybrid:-

- 12 PC's, 3 hubs, 1-Switch all generic are placed onto logical workspace.
- 3-generic hubs are connected to switch using copper cross-over wire and 12 PC's are connected to 3 hubs, using each using copper straight wire assigning IP address for each PC from 10.0.0.0 to 10.0.0.11 respectively.
- After adding & PC's from different hubs with simple-PDV and clicking on auto Capt -ure, packet passing simulation can be seen in simulation mode.
- In realtime mode open command prompt by clicking any PC → Devices → Workstation and Prompt and type 'ping or dest-IP- address' to send packet.

Observations:-

* Hub:-

- Learning outcome:- After source sends message to hub its broadcasted to all end devices but only distribution device tends and send response back to hub for narzo Shot by uday

Source to get response.
→ hub establishes connection to end-devices
quickly and signals by green-light

Result:-

PING 10.0.0.3

PINGING 10.0.0.3 with 32 bytes of data
REPLY FROM 10.0.0.3 bytes = 3 time = 0 ms

PING STATISTICS FOR 10.0.0.3

DETAILS of how many packets sent and received.

* Switch -

Learning observation:-

- Unlike hub, switch does not give green signal immediately but takes some amount of time called learning time and the packets can be sent once green signal can be sent once green signal is generated.
- Initially switch also broadcasts for all end-devices and the next time the communication happens & message passing happens only between source and destination devices.

Result:-

PING 10.0.0.5

PINGING 10.0.0.5 with 32 bytes of data
narzo Shot by uday

PING STATISTICS FOR 10.0.0.3

"Details of how many packets sent and received".

* Hybrid:-

Learning outcome :-

→ message sent by one pc of one hub to switch is sent to destination hub which broadcast to all devices of that hub and only destined end-device sends back response to source of other hub.

Result :-

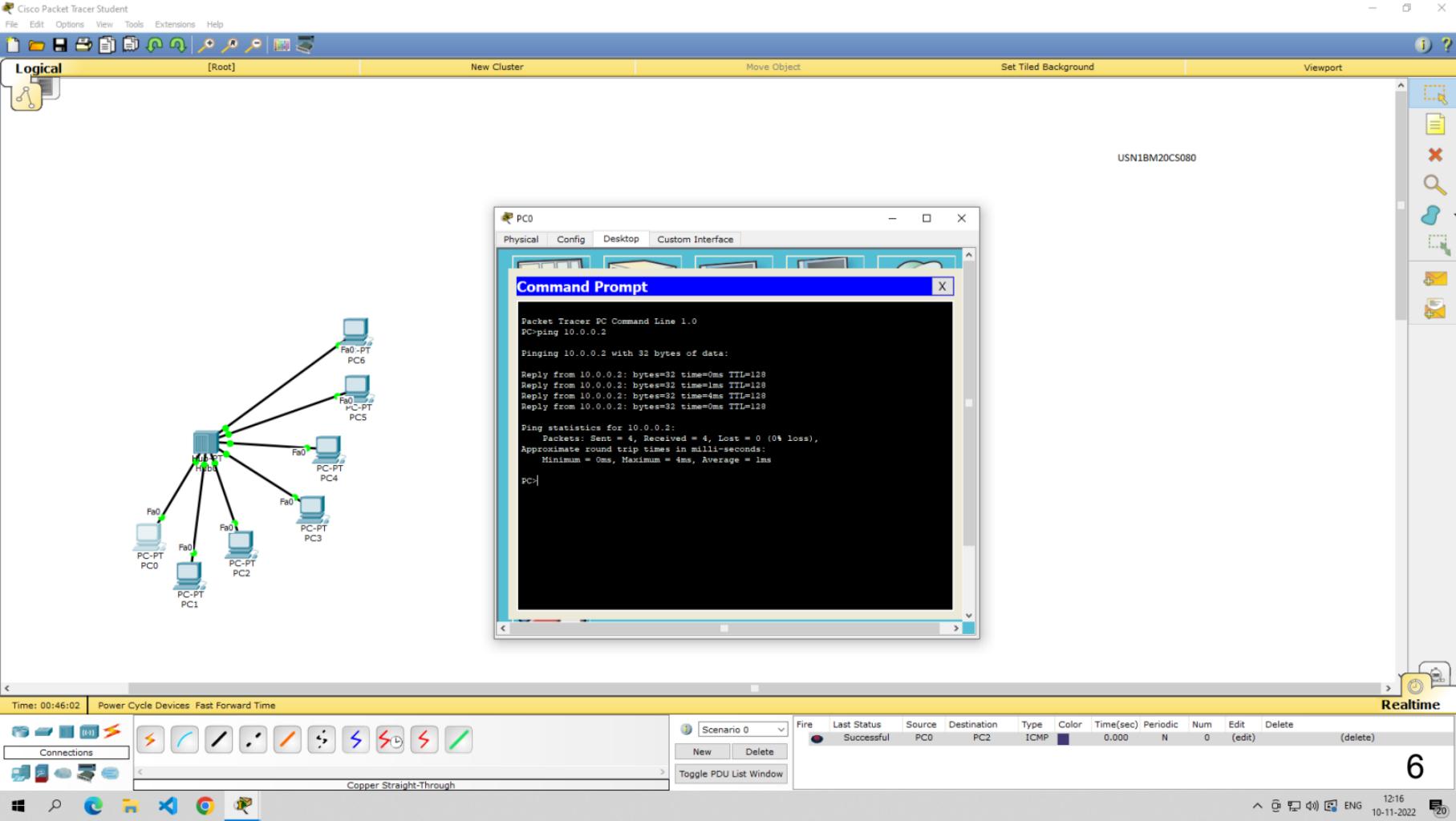
PING 10.0.0.4

PING or 10.0.0.1e with 32 bytes of data

REPLY from 10.0.0.4 bytes = 32

PING STATISTICS for 10.0.0.4

"Details of number of packets sent and received".





Logical

[Root]

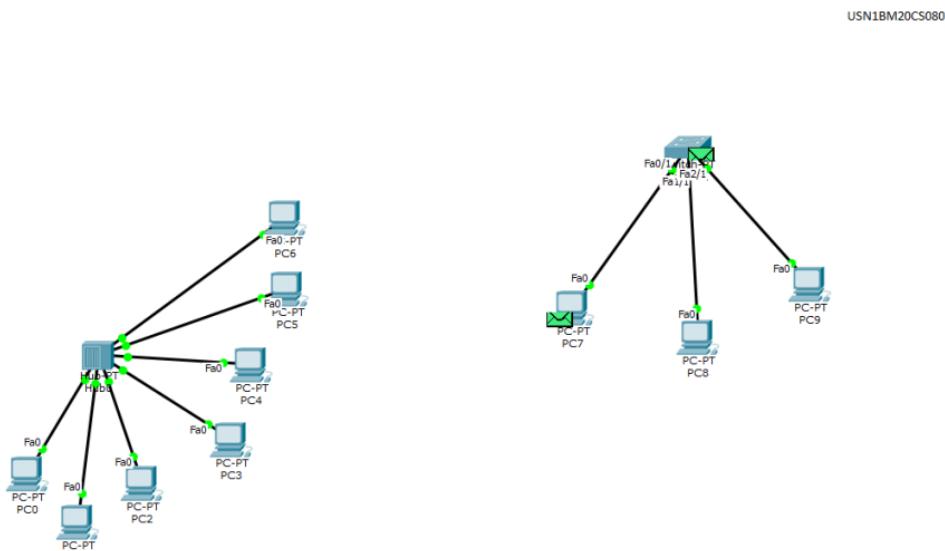
New Cluster

Move Object

Set Tiled Background

Viewport

?



USN1BM20CS080

Event List				
Vis.	Time(sec)	Last Device	At Device	Type
	0.004	Hub0	PC0	ICMP
	0.004	Hub0	PC1	ICMP
	0.004	Hub0	PC3	ICMP
	0.004	Hub0	PC4	ICMP
	0.004	Hub0	PC5	ICMP
	0.004	Hub0	PC6	ICMP
	0.004	Switch1	PC7	ARP
	0.004	--	PC7	ICMP
	0.005	PC7	Switch1	ICMP

Reset Simulation

Constant Delay

Captured to:

0.005 s

Play Controls

Back Auto Capture / Play Capture / Forward

Event List Filters - Visible Events

ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv3, PAGP, POP3, RADIUS, RIP, RIPng, RTP, SCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

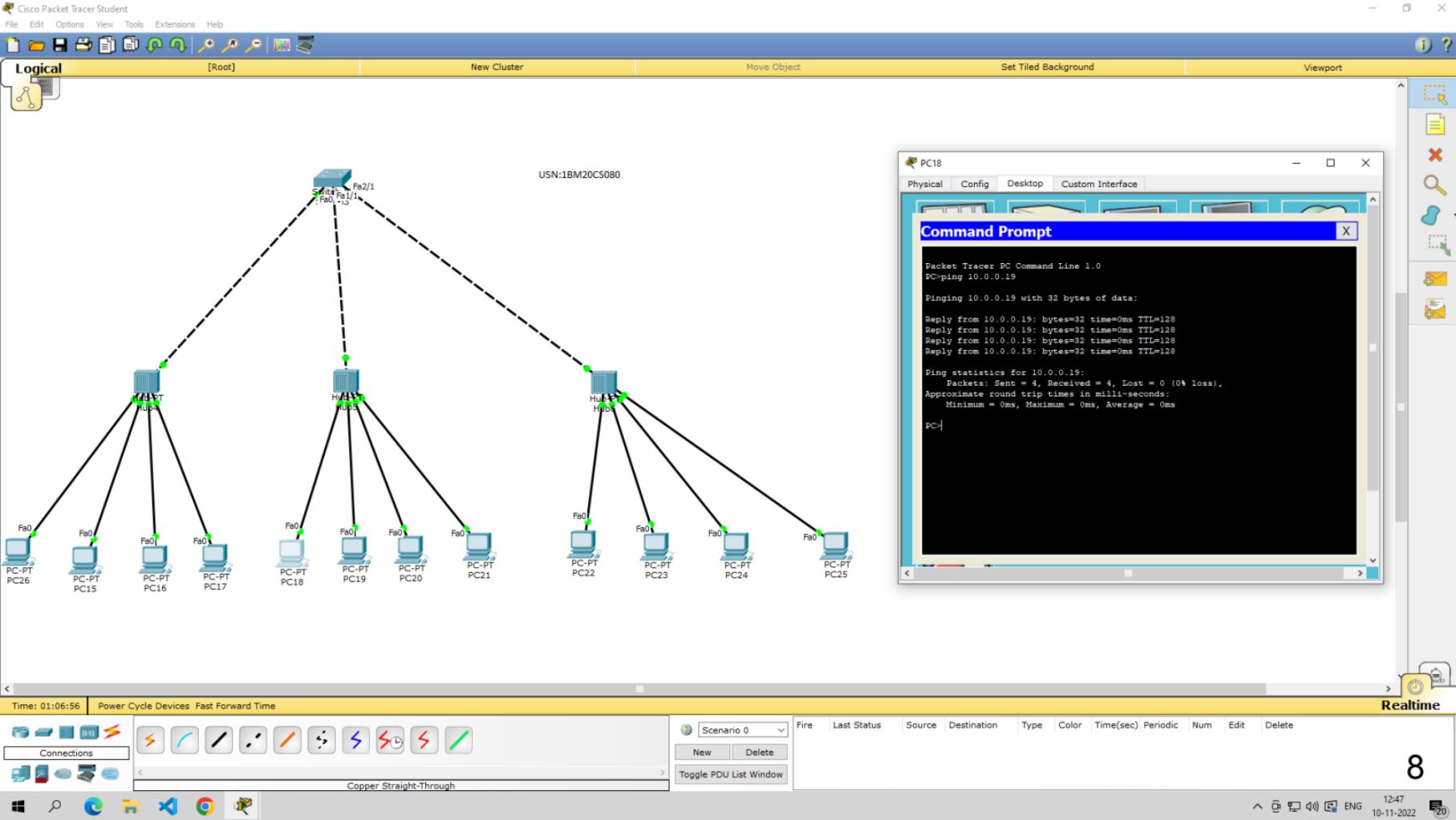
Edit Filters

Show All/None



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC2	ICMP	█	0.000	N	0	(edit)	(delete)
●	In Progress	PC7	PC8	ICMP	█	0.000	N	1	(edit)	(delete)

7





Logical

[Root]

New Cluster

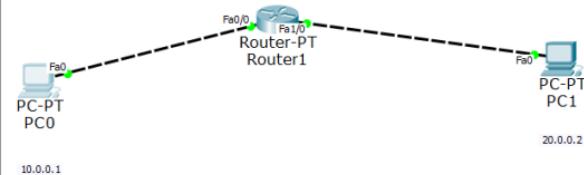
Move Object

Set Tiled Background

Viewport



1BM20CS080



10.0.0.1

PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Ping statistics for 20.0.0.2:  
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
  Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 1ms, Average = 0ms  
  
PC>ping 20.0.0.2  
  
Pinging 20.0.0.2 with 32 bytes of data:  
  
Reply from 20.0.0.2: bytes=32 time=3ms TTL=127  
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127  
  
Ping statistics for 20.0.0.2:  
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
  Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 3ms, Average = 0ms  
  
PC>
```

Time: 00:00:42 Power Cycle Devices Fast Forward Time

Realtime



Routers



Scenario 0

New Delete

Toggle PDU List Window

Fire Last Statu Sour Destinatio Type Colo Time(s) Period Num Edit Delete

(Select a Device to Drag and Drop to the Workspace)



Lab Week - 2

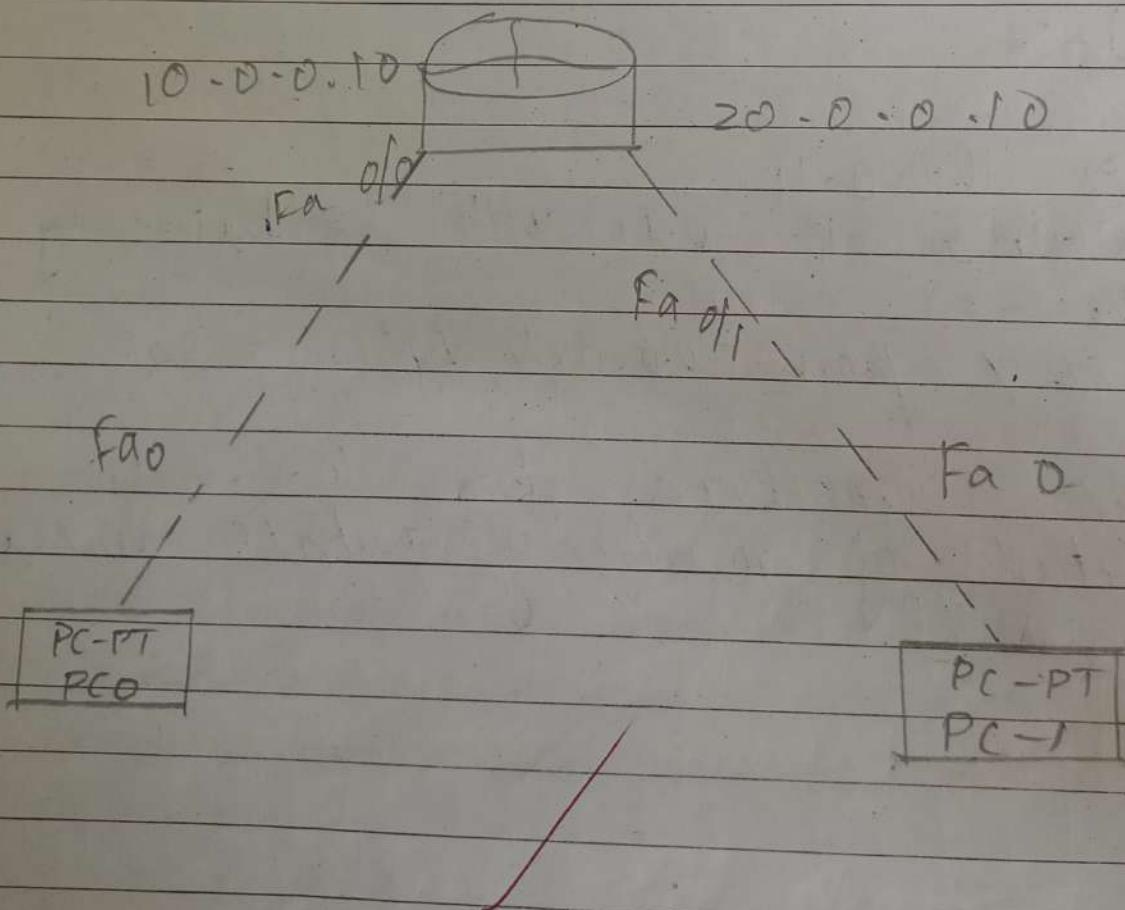
Experiment. Using router and PCs

Aim :- Configuring IP address to router
in packet tracer to explore the
following messages : ping responses
destination unreachable, Request timeout,
reply

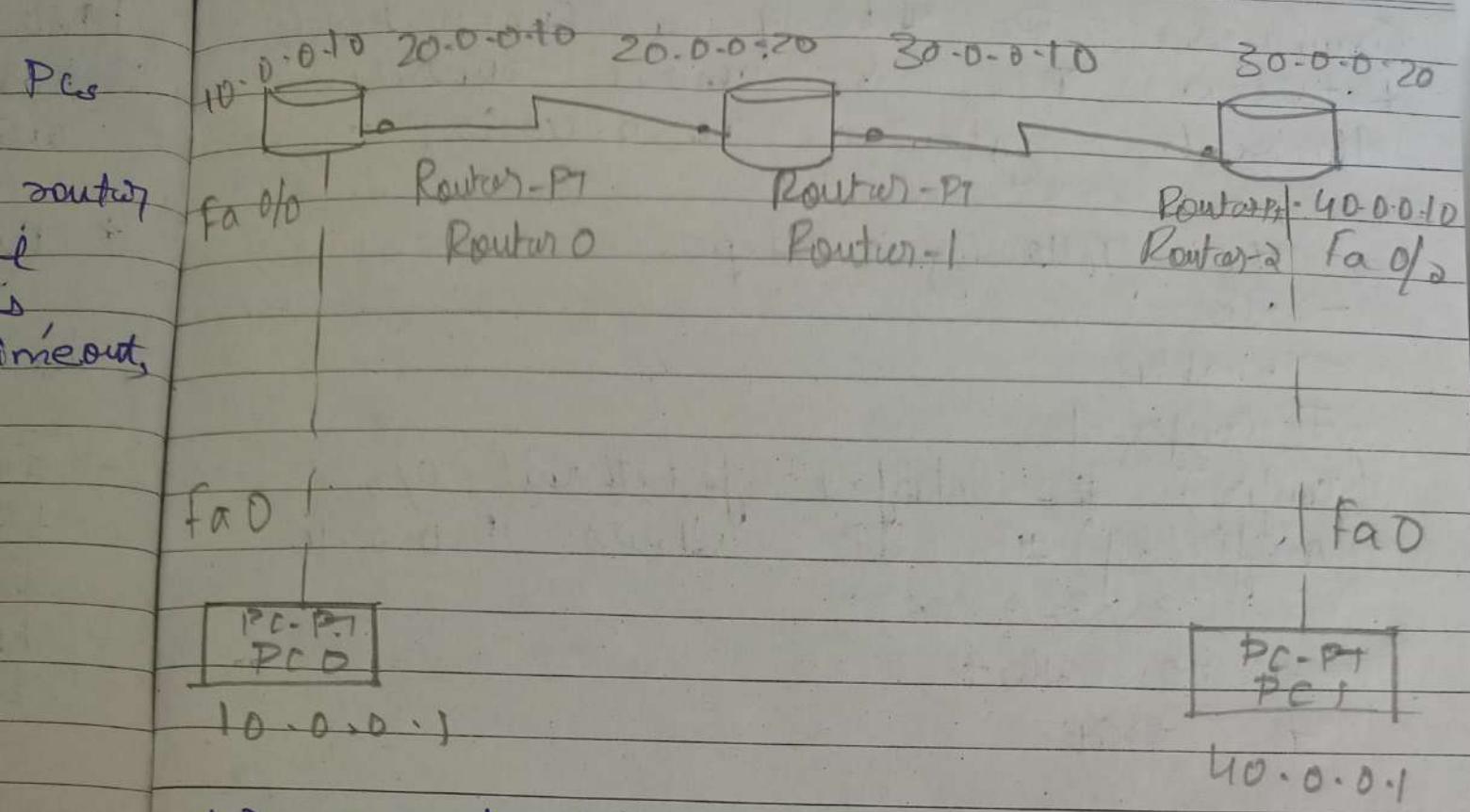
Topology :-

Using single router, two PCs

Generic Router



Using three routers & 2 PCs



→ Procedure:-

Using single router, 2 PC's

- (i) Place a generic router & 2 generic PC's in Workspase
- (ii) Connect the router & PC's Using Copper Cross over.
- (iii) Configure IP address of each PC and in the Configuration Under settings set gateways for PCs to router.
- (iv) click on the generic router & go to **Marco** Enter the following

Commands to set up a connection between PCs & generic router through gateway 10.0.0.10.

Xaser,

Do the following steps

→ NO

→ enable

config

(config) # interface fastethernet 0/0
(config-if) # IP address 10.0.0.10
255.0.0.0

no shutdown

exit

Now to setup connection between PC2 & router through gateway 20.0.0.10

interface fastethernet 1/0
ip address 20.0.0.10 255.0.0.0
no shutdown
exit

Once we enter no shut both timer the amber light between the PC & router turns green indicating that the two lines are connected.

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

Narzo Shot by Uday

Real
W
P
J
U
→

→

→

→

→

Real time mode :- Select the PC you want to send the packet from ~~the~~ PC and open its Command prompt from desktop tab. Specify the destination bar address. A response is sent from destination PC to Source PC.

Using three routers, 2 PCs.

- Place 3 generic routers & 2 generic PCs
- place a node for each device & specify the IP address.
- Connect the routers using serial DCI
- Click on - PC & then Configure tab. and Configure IP address of PCs.
- Next click on settings in Config tab - Set gateway as IP address of next router.
- IP address of PC & its gateway address should belong to same network.

for connecting routers

click on Router 0.

Go to CLI & enter the commands

→ no

→ enable

→ interface serial 2/0

→ IP address 20.0.0.10 255.0.0.0

→ no shut-

Repeat same ^{Commands} for Router 1

After this the red signal ~~on~~ changes to green indicating they are ready for communication.

for connecting two devices [PC & router]

→ Go to router.

→ Open CLI for router 0 & enter the following commands

→ no

→ enable

→ config

→ interface fastethernet 0/0

→ IP address 10.0.0.10 255.0.0.0

→ no shut-

The red light changes to green to indicate that they are ready for communication.

Configuring Router 0 of network 30-

- no
- enable
- config
- interface serial 2/0
- IP route 30.0.0.0 255.0.0.0 20.0.0.0
- exit
- show ip route.

Configuring Router 0 of network 40

- no
- enable
- config
- interface serial 2/0
- IP route 40.0.0.0 255.0.0.0 20.0.0.0
- exit
- show ip route.

Similarly repeat for router 1 & router 2

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

Real time mode:- When the PC 0 goes to its command prompt & ping the router 0 once the message has been sent successfully. Repeat this with routers 1 & 2 as well. Finally ping PC 1.

Observation:-

1 Router:-

When PC 0 pings PC 1 for the first time, we get the first packet as request time out.

Now, if we ping PC 1 again we get all 4 packets. Next reverse the pinging of PC 0 from PC 1.

3 router:

Before training the routers we get the result as destination not reachable. After training the routers, we get clear statistics.

Result

Using 1 router, 2 PCs

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
 $TTL=127$.

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

packets: sent=11, received=4, lost=0

2) Using three routers two PCs

→ Ping 40.0.0.1
pinging 40.0.0.1 with 32 bytes of data

Reply from 10.0.0.10 : destination host unreachable
Ping statistics for 40.0.0.1
packets: sent 4, received = 0, lost = 4

3) Re ping 20.0.0.10 with 32 bytes of data.

Reply from 20.0.0.10 : bytes = 32 time = 1ms TRC = 1

Reply from 20.0.0.10 : bytes = 32 time = 0ms
TRC = 285

Ping statistics for 20.0.0.10:
packets: sent 4, received = 4, lost = 0

4) Ping 30.0.0.10

Ping 30.0.0.10 with 32 bytes of data

Reply from 30.0.0.10 : bytes = 32 time = 1ms TRC = 1

Reply from 30.0.0.10 / by Uday
bytes = 32 time = 8ms TRC = 15

ping statistics for 30.0.0.10:

packets: sent = 4, received = 4, lost = 0

5) 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 = 22, time > 10ms TTL=128

Reply from 40.0.0.1: bytes = 22, time = 8ms,
 $TTL = 125$

ping statistics for 40.0.0.1:

packets: sent = 4, received = 3, lost = 1

6) Ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data

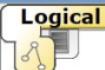
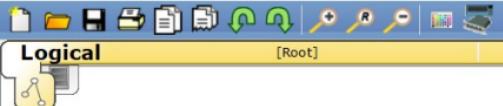
Reply from 40.0.0.1: bytes = 32 time = 2ms
 $TTL = 125$

Reply from 40.0.0.1: bytes = 32 time = 9ms

ping statistics for 40.0.0.1:

packets: sent = 4, received = 4, lost = 0

N
27/11/22



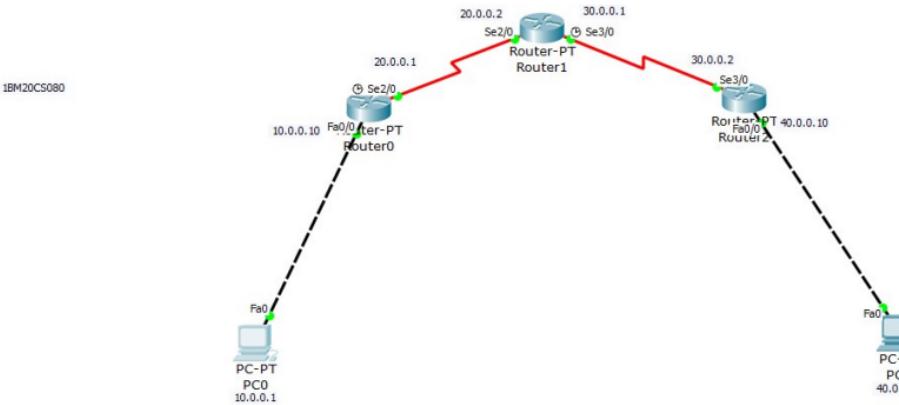
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Command Prompt

```

Request timed out.

Reply from 40.0.0.1: bytes=32 time=12ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms 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 = 4ms, Maximum = 12ms, Average = 8ms

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=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=12ms TTL=125
Reply from 40.0.0.1: bytes=32 time=17ms TTL=125
Reply from 40.0.0.1: bytes=32 time=7ms TTL=125

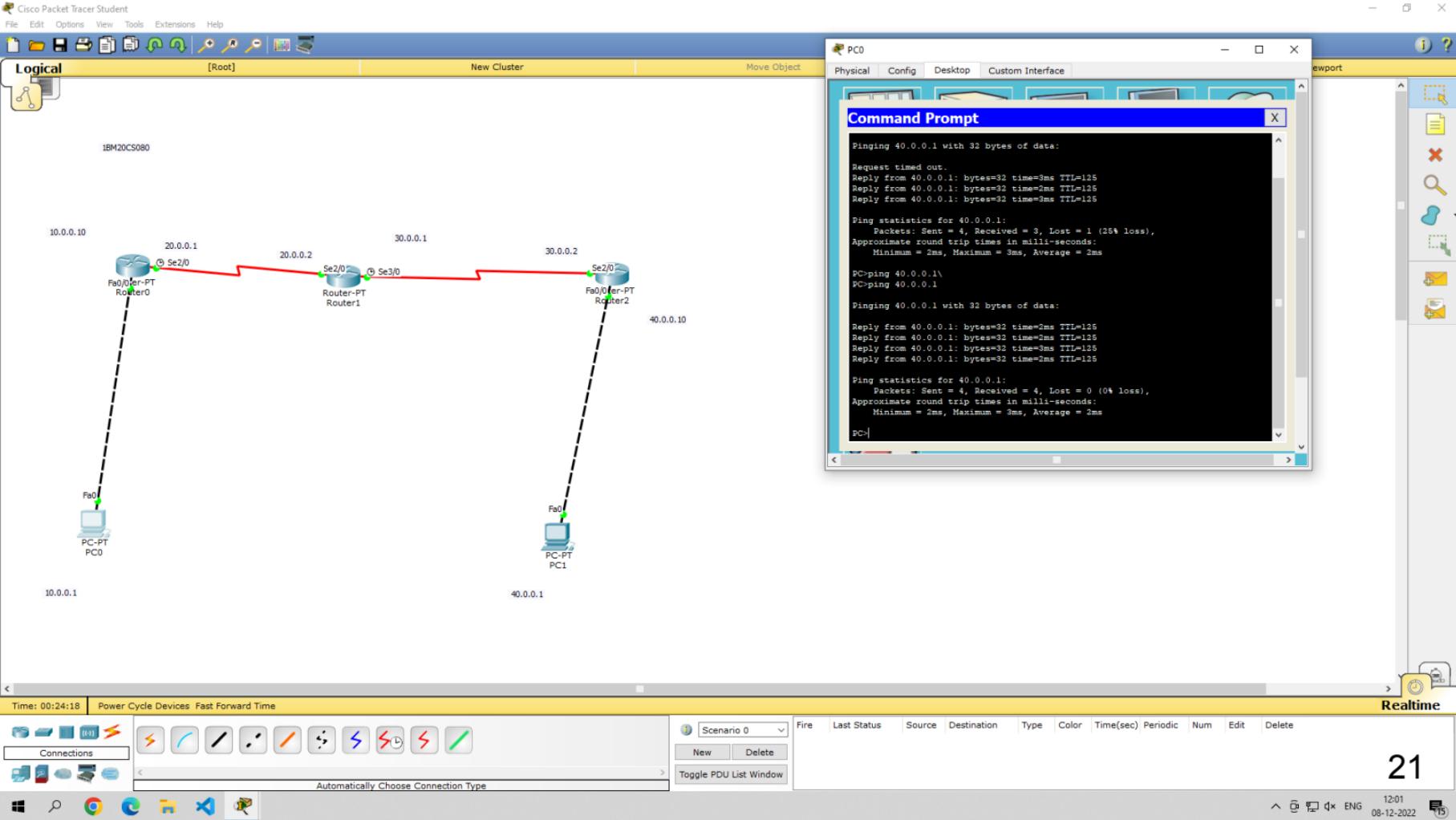
Ping statistics for 40.0.0.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 17ms, Average = 12ms

PC>
  
```



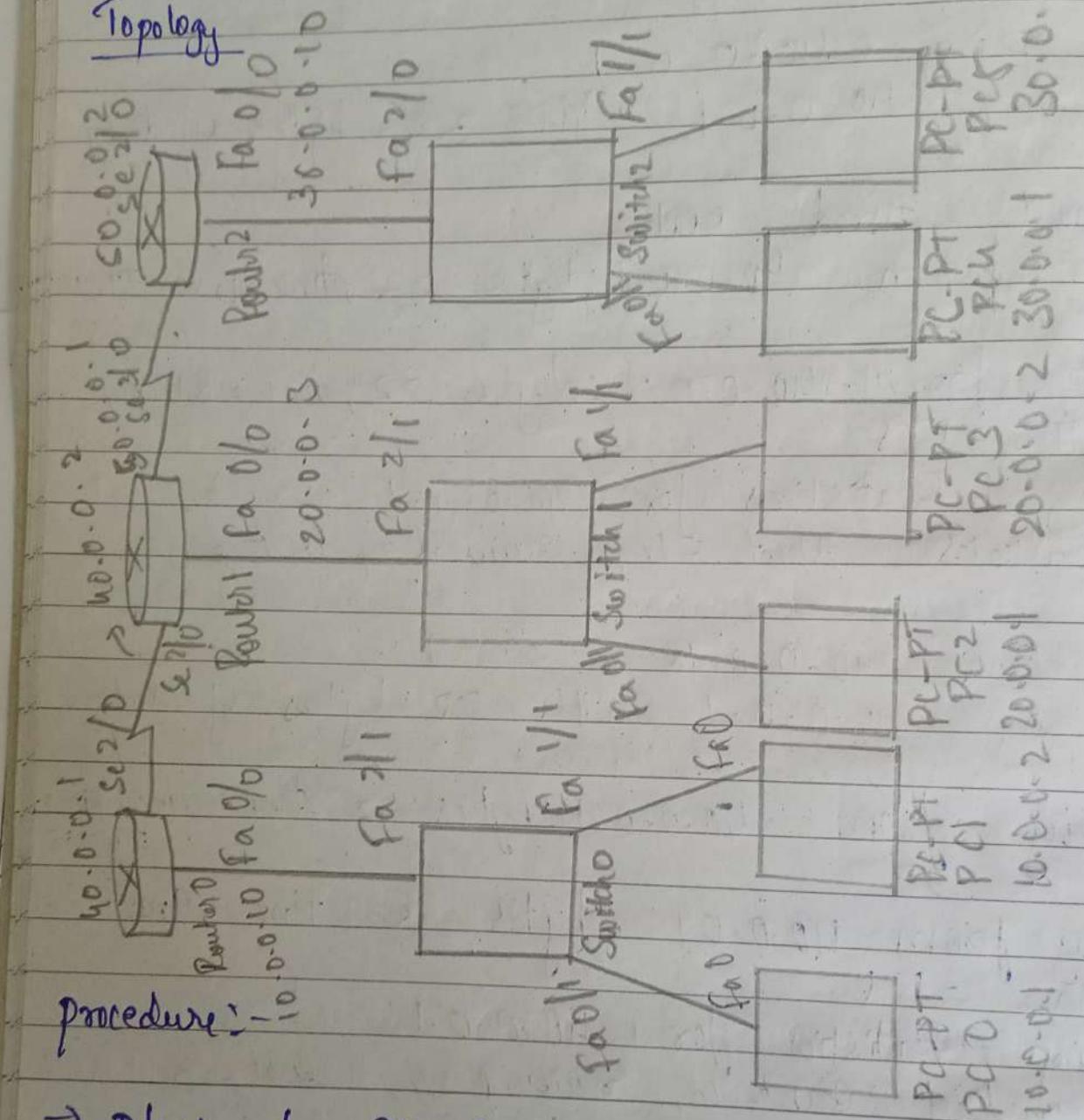
Scenario 0	New	Delete	Fire	Last Status	Source	Destination	Type	Color	Time(se)	Periodic	Num	Edit	Delete
1841 1941 2620XM 2621XW 2811 2901 2911 819 Generic Generic													

Toggle PDU List Window



Lab-3 NATZO Shot by uday
 AIM:- Configuring default route to the router

Topology



procedure:-

- place 6 generic routers , 3 switches & 6 PCs , 3 switches & 6 PCs , 3 routers and connect two PCs to each switch with copper straight through wire and each switch is connected to one router with a copper straight through wire and the 3 routers are connected among themselves by

Router serial DCE cable and the nodes are placed for all the devices & networks.

→ PC is clicked to set attributes for a PC and each PC has 3 attributes which are the IP addresses Subnet mask & the gateway & all the three are set according to the nodes placed. Thus process is done for all the 6 PC's.

→ For Router 1 the config are done in the command line interface (CLI). The IP address and Subnet mask are set for both the interface fast ethernet 0/0 as 10.0.0.10 & 255.0.0.0 and Serial 2/0 as 40.0.0.1 & 255.0.0.0 Router 2 is default router for Router 1 & this is done by the Cmd ip route 0.0.0.0 0.0.0.0 40.0.0.2

→ For Router 2 the IP address & Subnet mask are set for all 3 interfaces - fast ethernet 0/0 as 20.0.0.3 & 255.0.0.0 & serial 2/0 as 40.0.0.2 & 255.0.0.0 and serial 3/0 as 50.0.0.1 & 255.0.0.0

→ Router 2 does not have any default static routing is done for the

network 10 & 40 by the following cmd
ip route 10.0.0.0 255.255.0.0 40.0.0.1
ip route 30.0.0.0 255.0.0.0 50.0.0.2

→ Router 3 is configured in both IP interfaces with IP address & subnet mask as fast ethernet 0/0 with 30.0.0.10 & 255.0.0.0 & serial 2/0 with 50.0.1 & 255.0.0.0 - the default router for router 3 router 2 & thus set by the cmd : ip route 0.0.0.0 0.0.0.0 50.0.0.1

→ ping cmd is executed from 10.0.0.1 to 20.0.0.1 & from 10.0.0.1 to 30.0.0.2.

Observation:-

Learning Outcome:-

- one router cannot have two default routers.
- The default router for first router is the middle router because any packets which have to be delivered will go to the middle router.
- The default router for 3rd router is the middle router for the same reason.

Narzo Shot by Uday

→ the middle router does not have any default router because if one of the router is made default then there is a chance that the packets which are to be sent to the switch are sent to the router.

~~Router~~ -

Results -

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 = 4ms, TTL = 126

Reply from 20.0.0.1: bytes = 32, time = 2ms, TTL = 126

Reply from 20.0.0.1: bytes = 32, time = 6ms, TTL = 126

ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

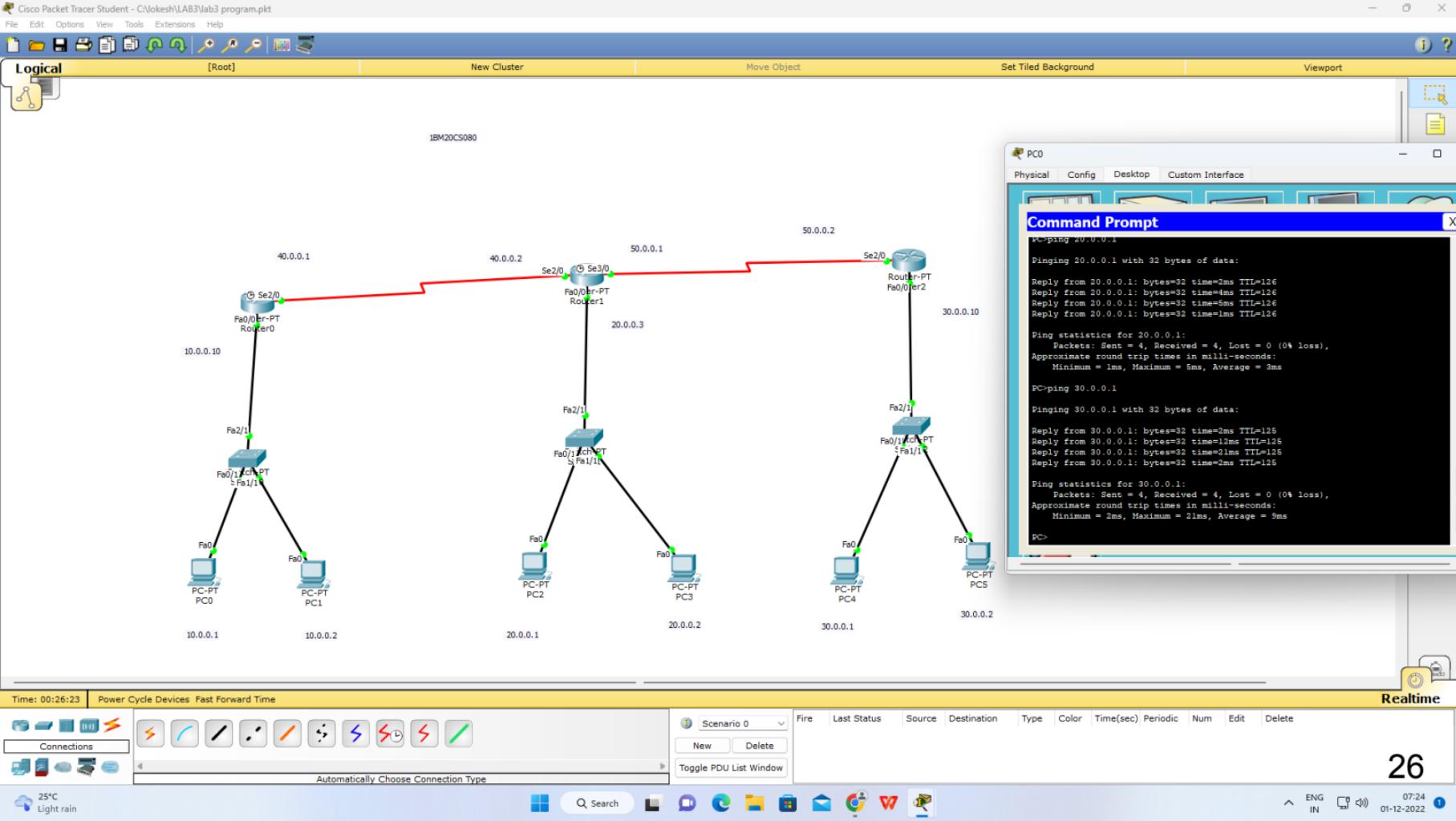
Request timed out

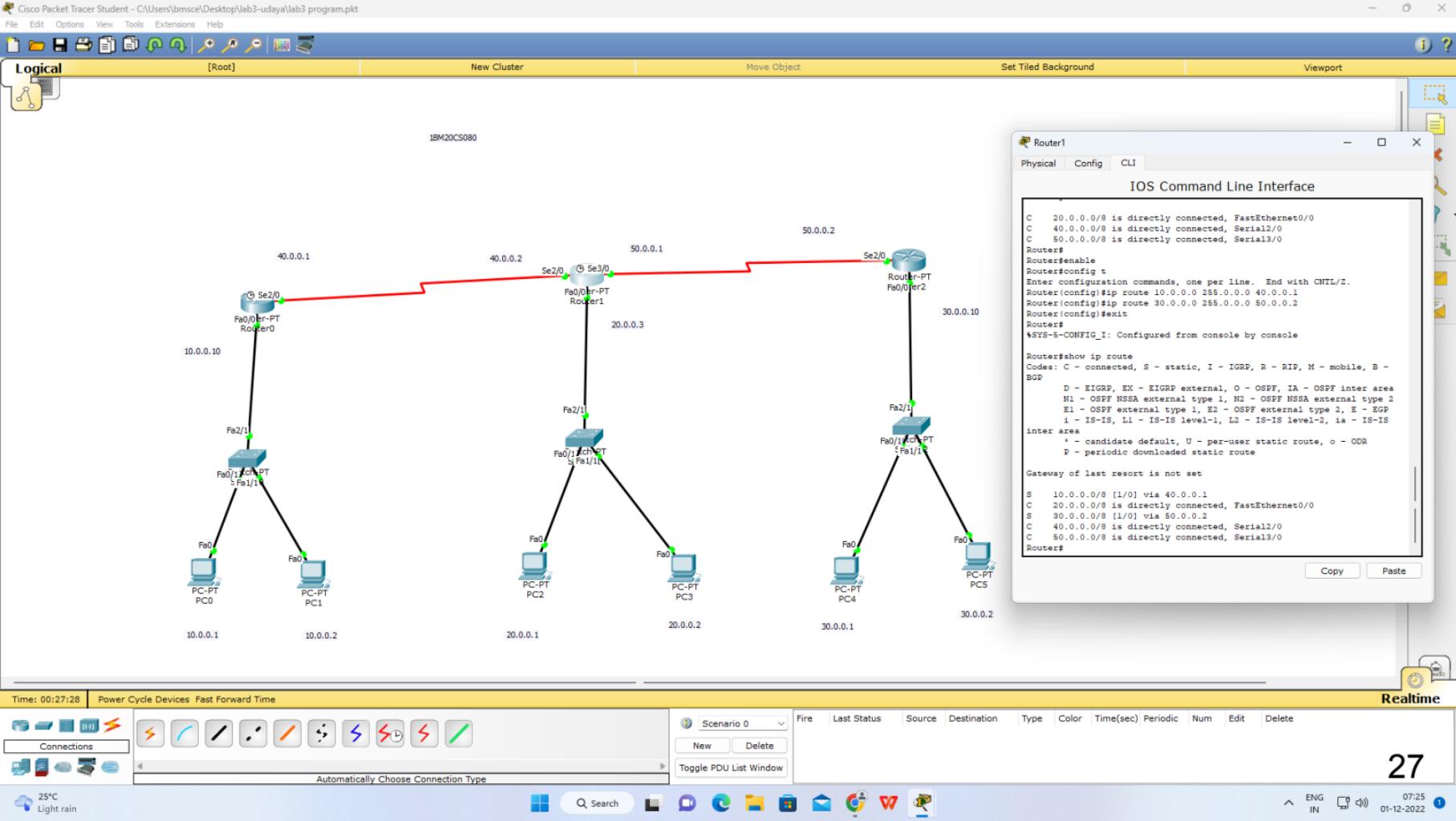
Reply from 30.0.0.2: bytes = 32, time = 4ms, TTL = 125

Reply from 30.0.0.2: bytes = 32, time = 4ms, TTL = 125

Reply from 30.0.0.2: bytes, time = 4ms, TTL = 125

Marzo Shot by Uday

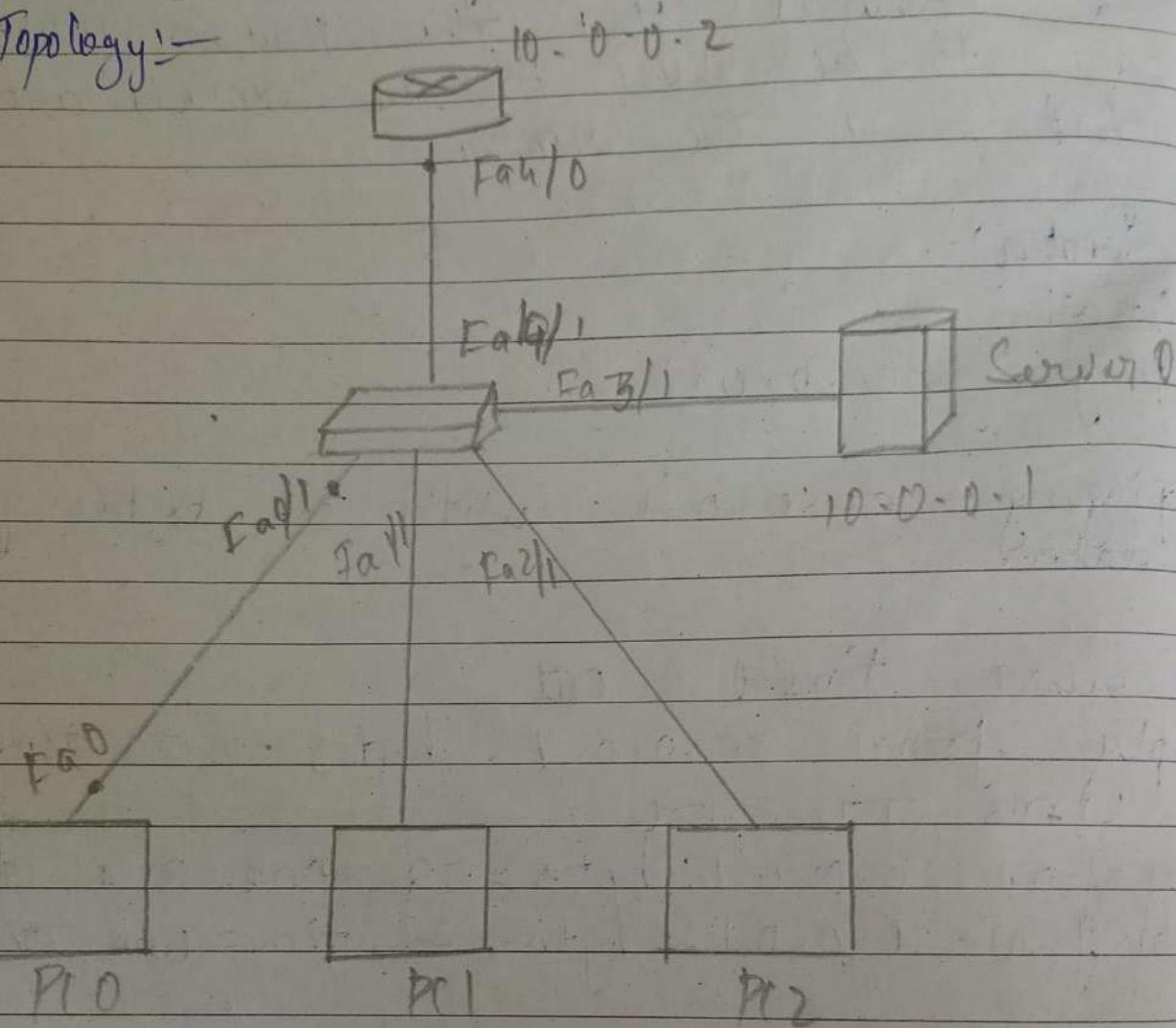




Lab - 4

AIM :- Configuring DHCP within a LAN
in a packet Tracer.

Topology :-



Procedure :-

1. Place 3 PC's, 1 switch, 1 server & one Router. Connect all devices & router to switch.
2. Set the ip address 10.0.0.1 for server in Marzo Shot by uday gateway as 10.0.0.2

- AN
3. at the Router 0 , do the following CLI Commands
 → interface fastethernet 4/0
 → ip address 10.0.0.2 255.0.0.0
 4. format the Server , go to services and enable Service
 Set the default gateway 10.0.0.2 ,
 DNS Server 10.0.0.1
 Start IP : 10 0 0 3
 & click save
 - 5 - for all the PC's go to IP Config & turn on the DHCP
 - 6 - after all the above steps ping for the messages .

Observation:-

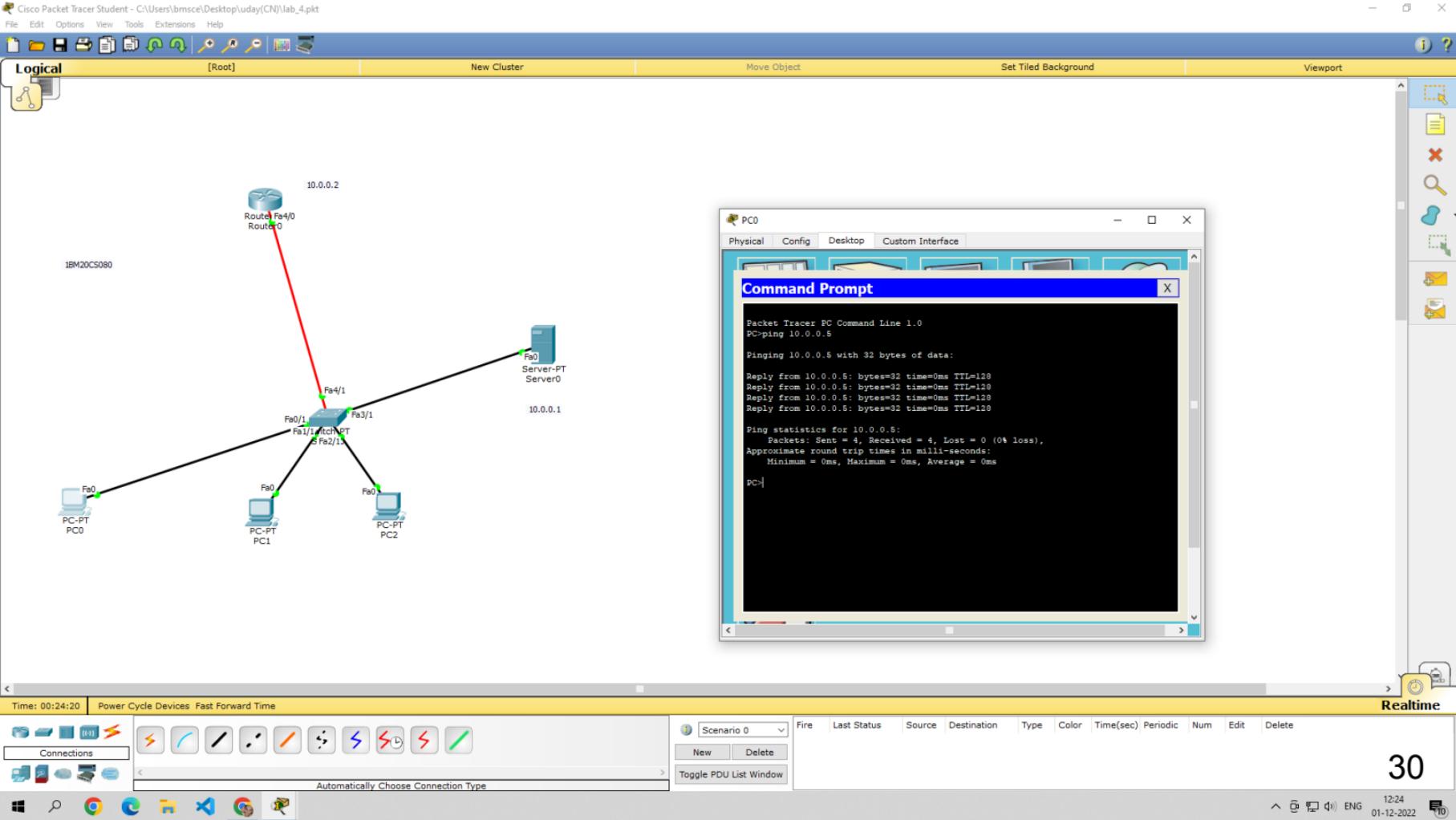
Learning outcome :- ?
 → Server ~~sits~~ provides the IP addresses for the PC's

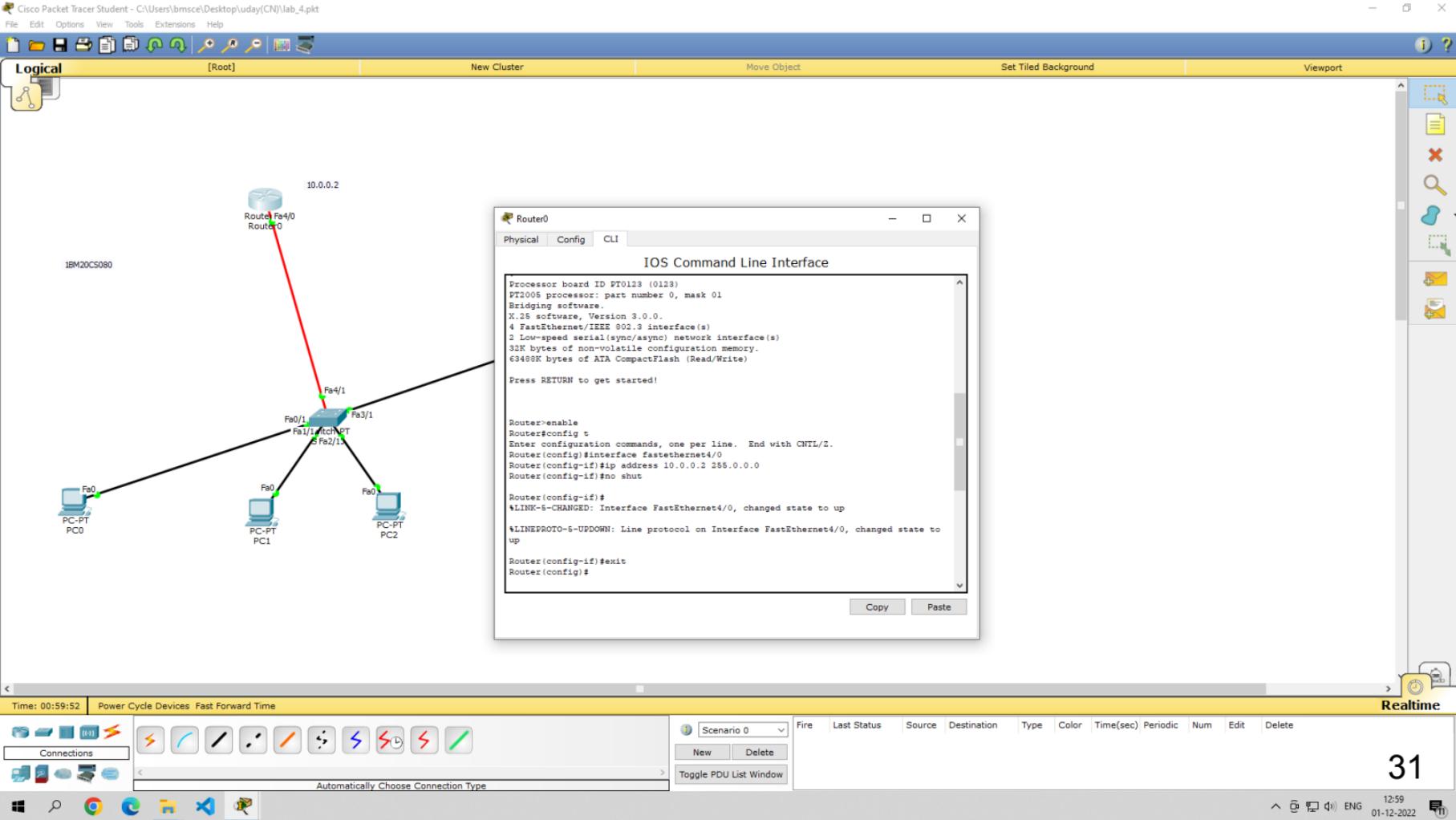
Result:-

Ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128
 Reply from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128
 Reply from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128
 Reply from narzo 5. Shot by uday N. 122 bytes = 32 time = 0ms TTL = 128





8/12/2022

Lab :- 5

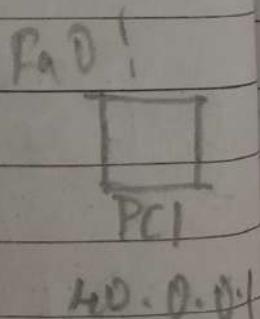
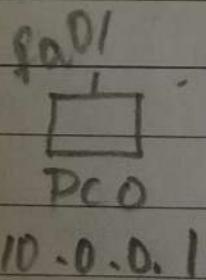
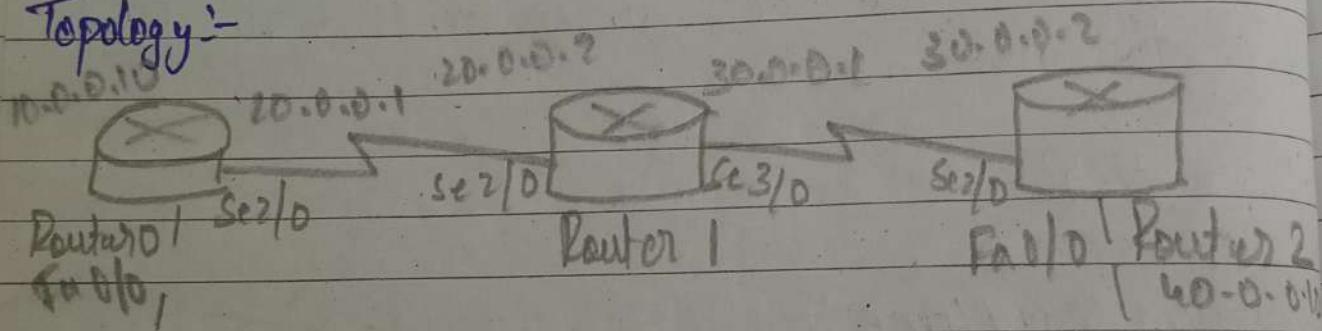
AIM :- Configuring RIP routing protocol by
routers

RIP:-

Routing information protocol that uses hop as a metric to find the best path b/w the source & the destination.

Cable?
?

Topology:-



2/2022
Q1 in

Procedure:-

1. place 2 PCs, 3 routers. ~~connect~~
- 2- set IP address for PC0 as 10.0.0.1
Set subnet gateway as 10.0.0.10
for PC1 Set IP as 10.0.0.1 gateway as 10.0.0.10
3. Now go at Router 0, at CLI mode
make connections for PC0 & Router 0
by following commands
→ interface fastethernet 0/0
→ ip address 10.0.0.10 255.0.0.0
→ no shutdown.
4. Do the above step similarly
for Router 2 to connect PC1.
5. Now for Router 0 & Router 1
connection
at Router 0:
~~Router~~ at CLI
interface serial 2/0
IP address 20.0.0.1 255.0.0.0
encapsulation PPP
clock rate 64000
no shutdown
- Similarly do for at Router 1
to make connection between
Router 0 & Router 1
6. Repeat same for Router 1 &
Router 2.

7. After all these above steps Ping for the messages.

Observations:- Using RIP routing it is easy when large number of routes are present as in case of static, IP routing is done for each & every routers

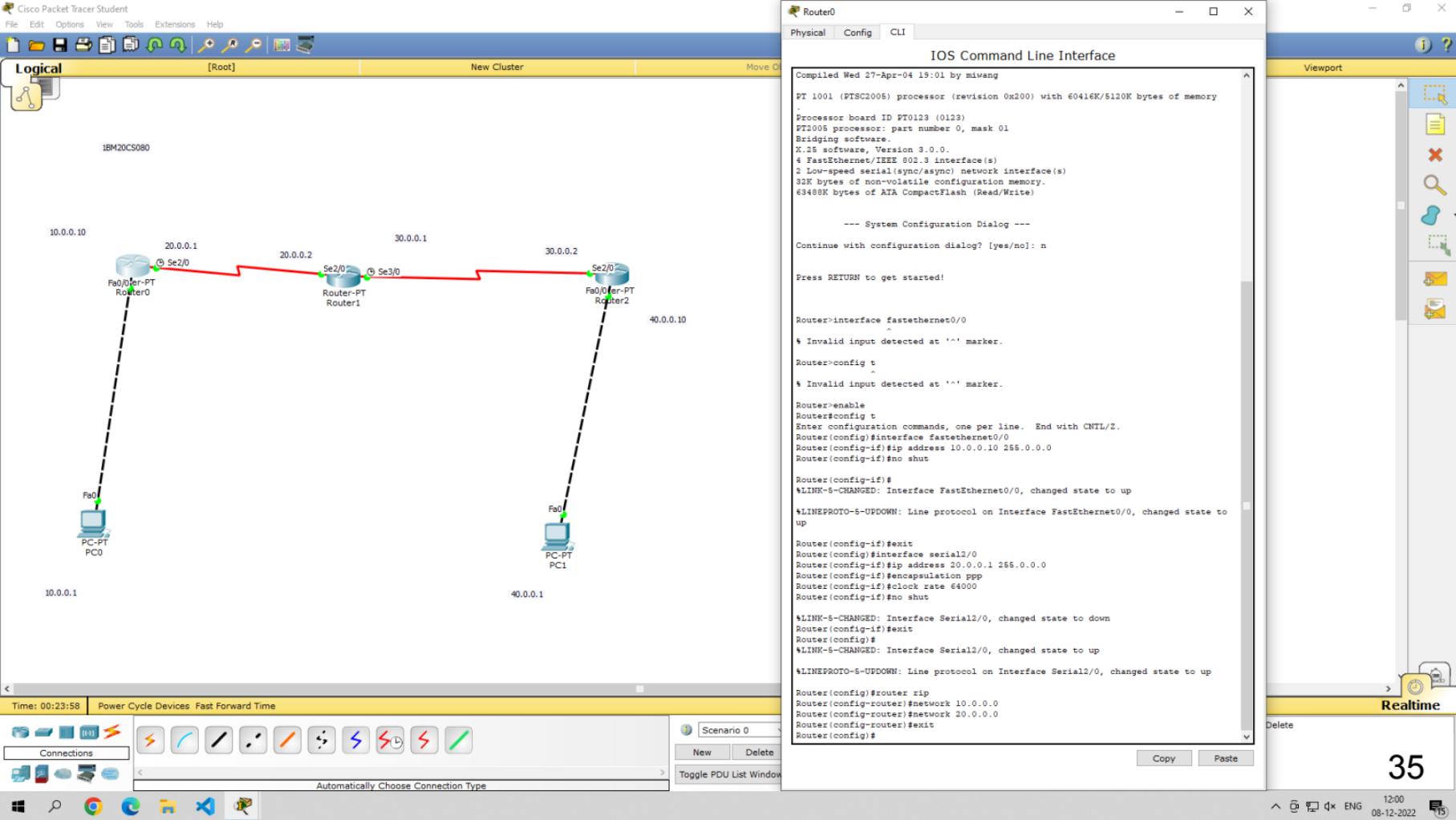
Result:- ping 40.0.0.1

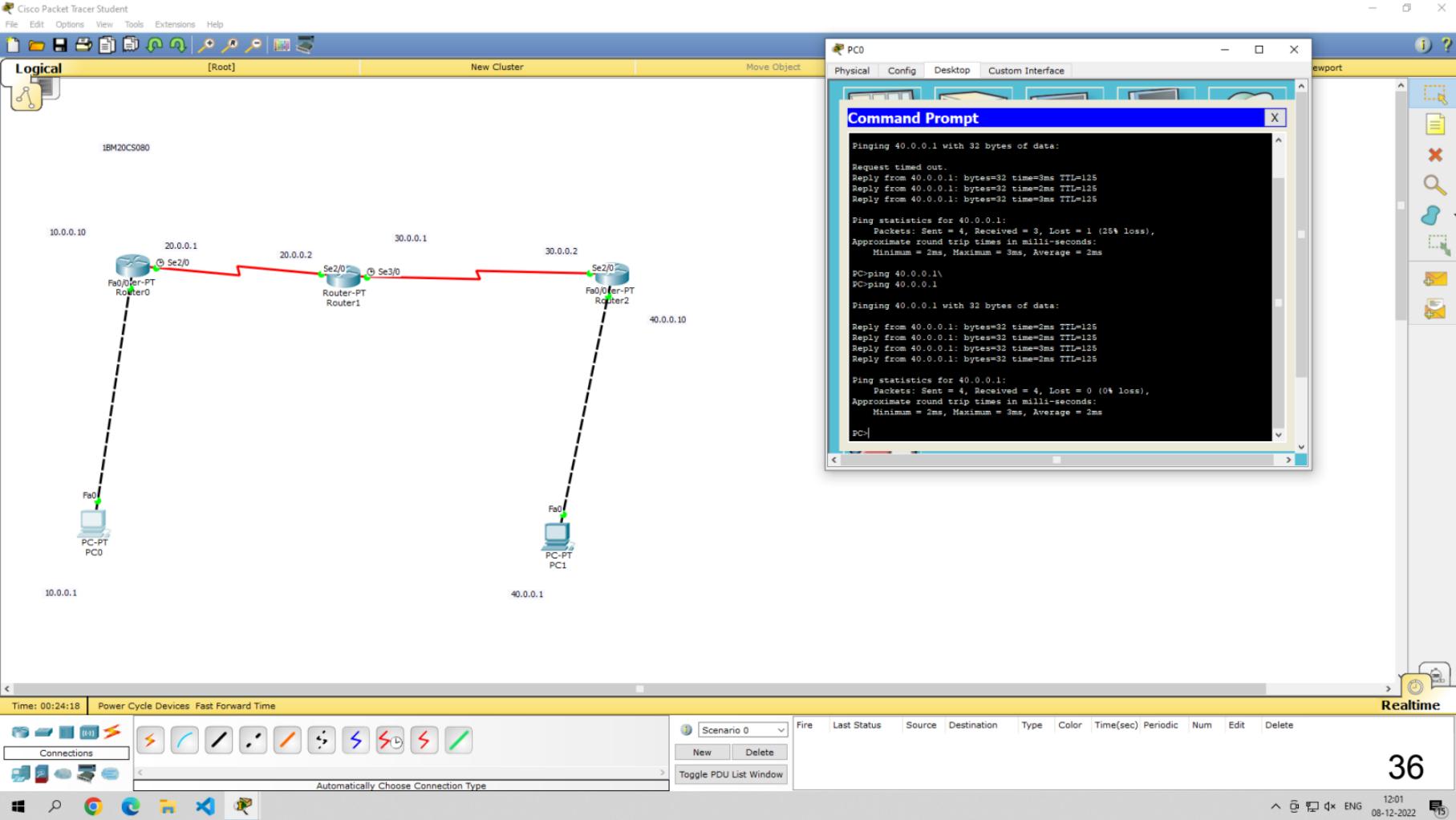
pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1: bytes=32 time=2ms TTL=11
Reply from 40.0.0.1: bytes=32 time=21ms TTL=11
Reply from 40.0.0.1: bytes=32 time=21ms TTL=11
Reply from 40.0.0.1: bytes=32 time=21ms TTL=11

Ping statistics for 40.0.0.1:

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

✓
8/12/22





Lab-6

15/12/22

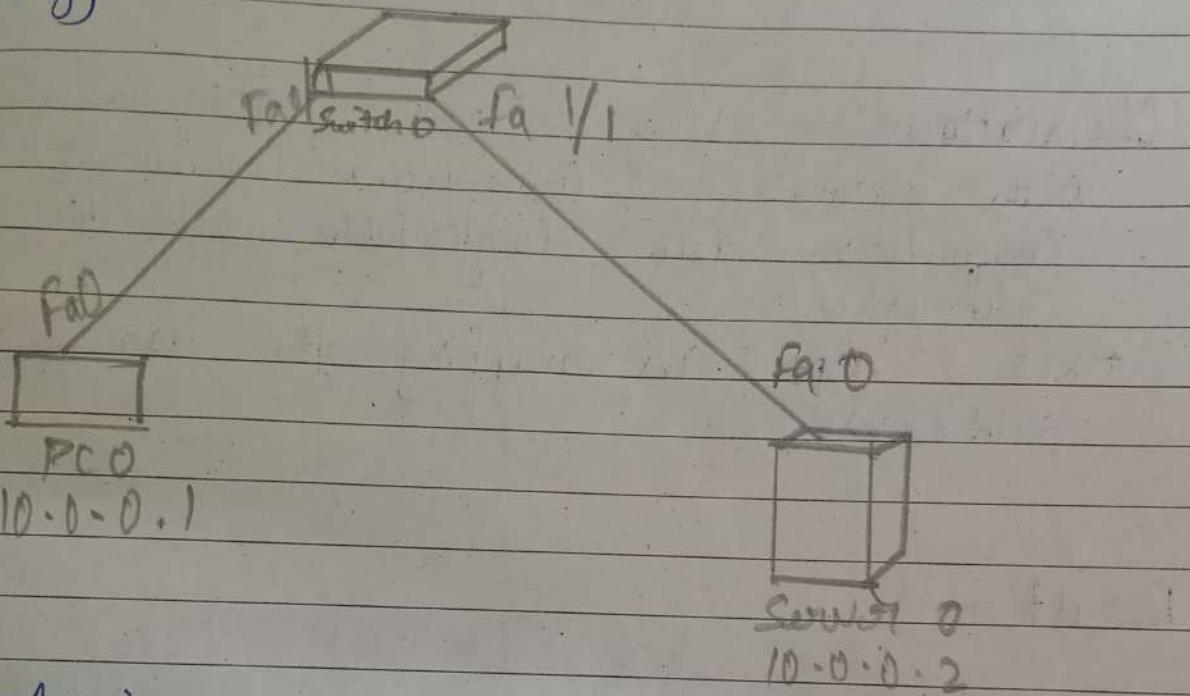
AIM:- Lab Experiment

AIM:- Demonstration of WEB Server &

DNS Using packet Tracer.

DNS:- helps to map a name with IP address as naming is comfortable than IP address.

Topology



procedure:-

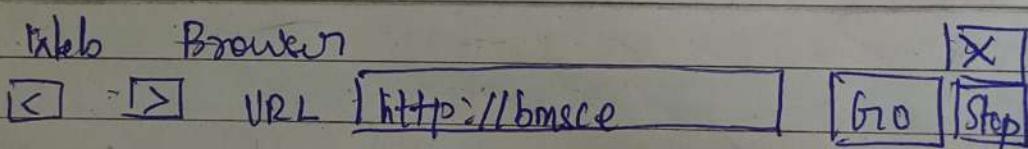
1. Set the IP addresses for PC & Server as 10.0.0.1 & 10.0.0.2.
2. At the PC invoke Web browser in the Desktop tab. & give the ip of the server & see the home page displayed.
3. Now go to Services → properties of the server at HTTP click on the edit button of index.html & change the first from default & narzo shot by uday changed output.

4. at the server → services click on the DNS → selection & enter the name & IP as seen in the figure below. click on add button.
5. check the same by giving the name instead of IP address.

Observations:- Using DNS its easy with naming conventions where computers are comfortable with IP addresses.

DNS helps in mapping the name & IP addresses.

Result:-



BMSCE

Welcome to BMSCE

Resume

Web Browser

[<] [>] URL http://bmsce/uday.html [Go] [Stop]

name

Uday

age

21

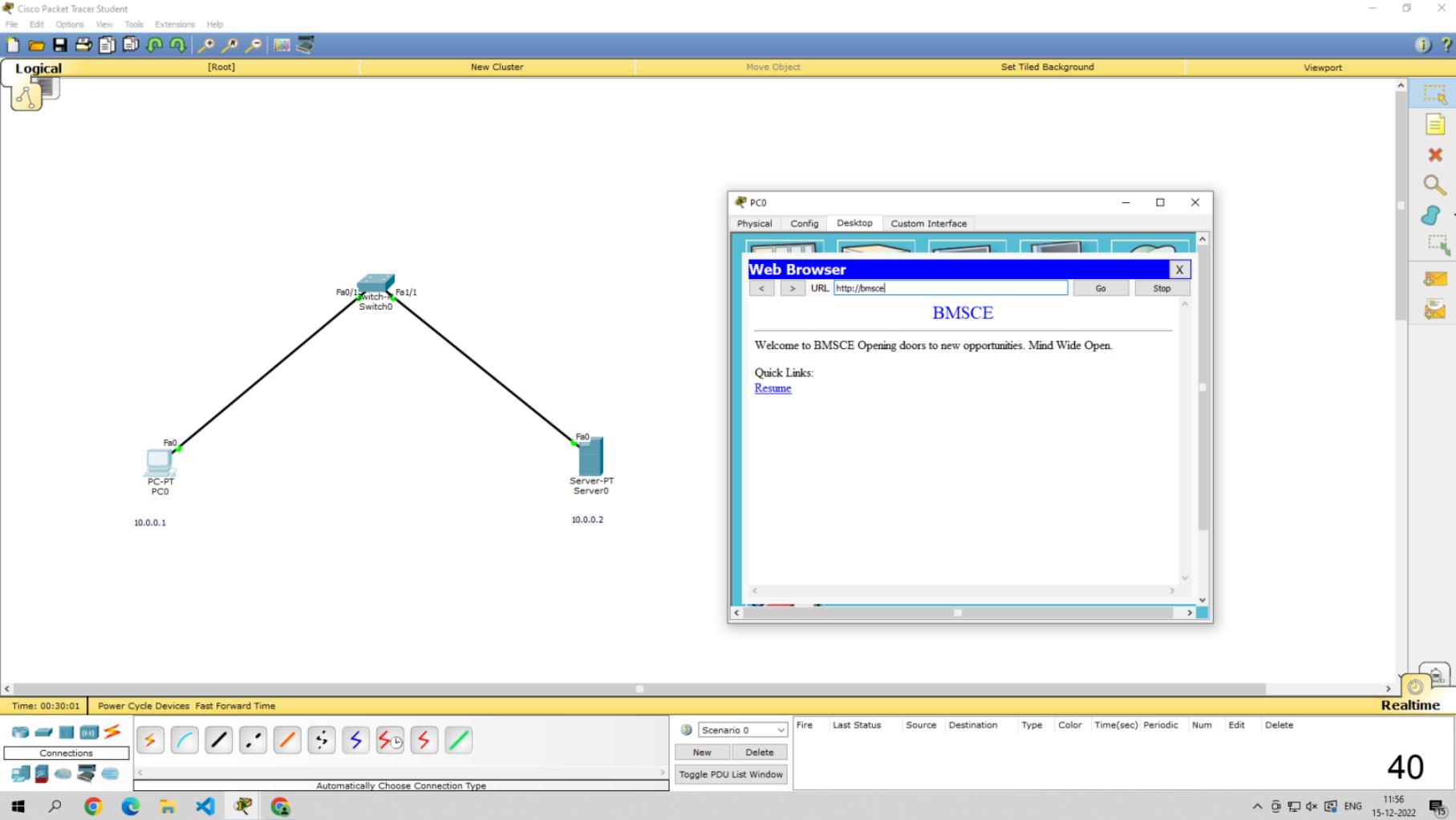
College

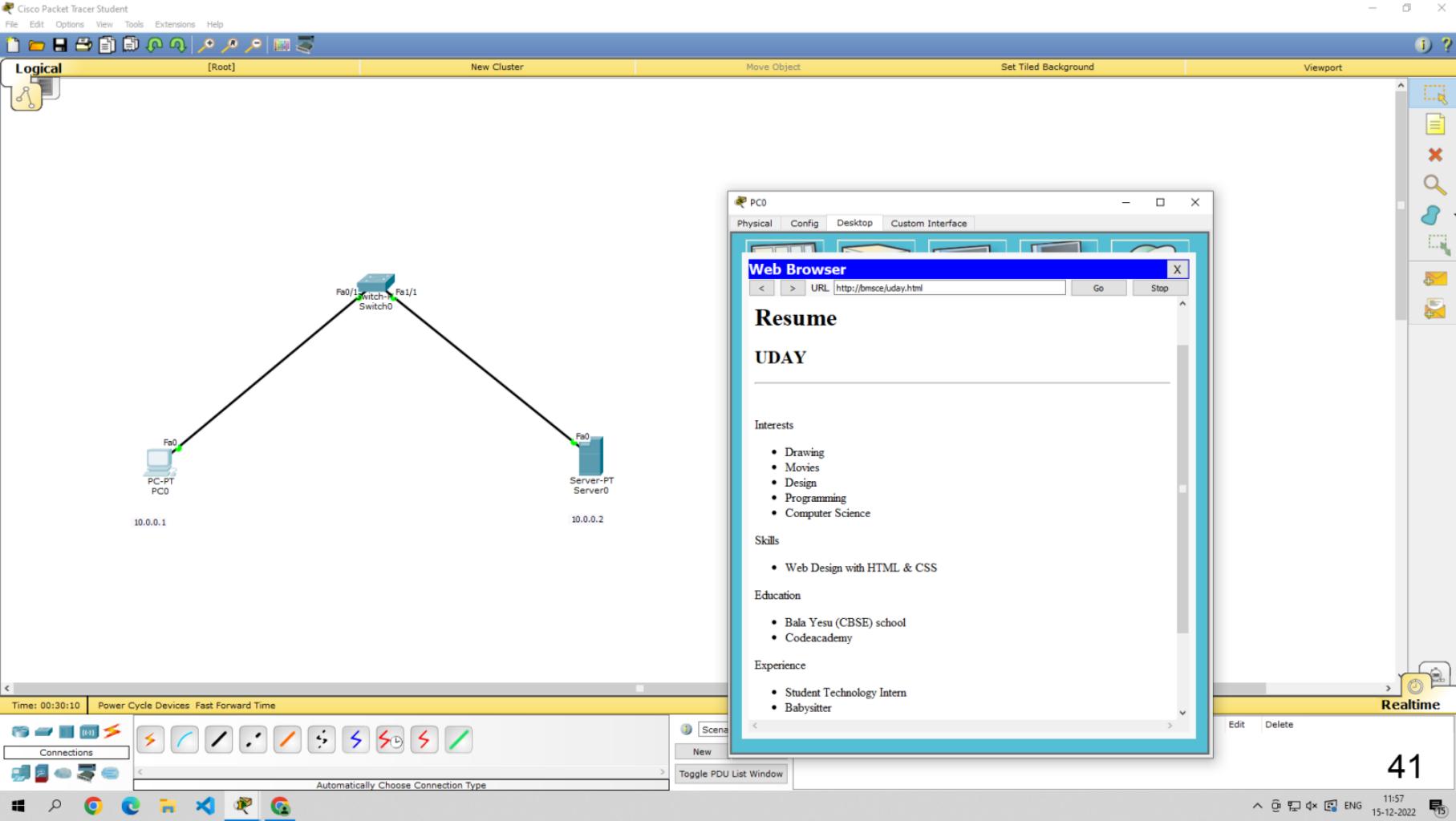
BMCE

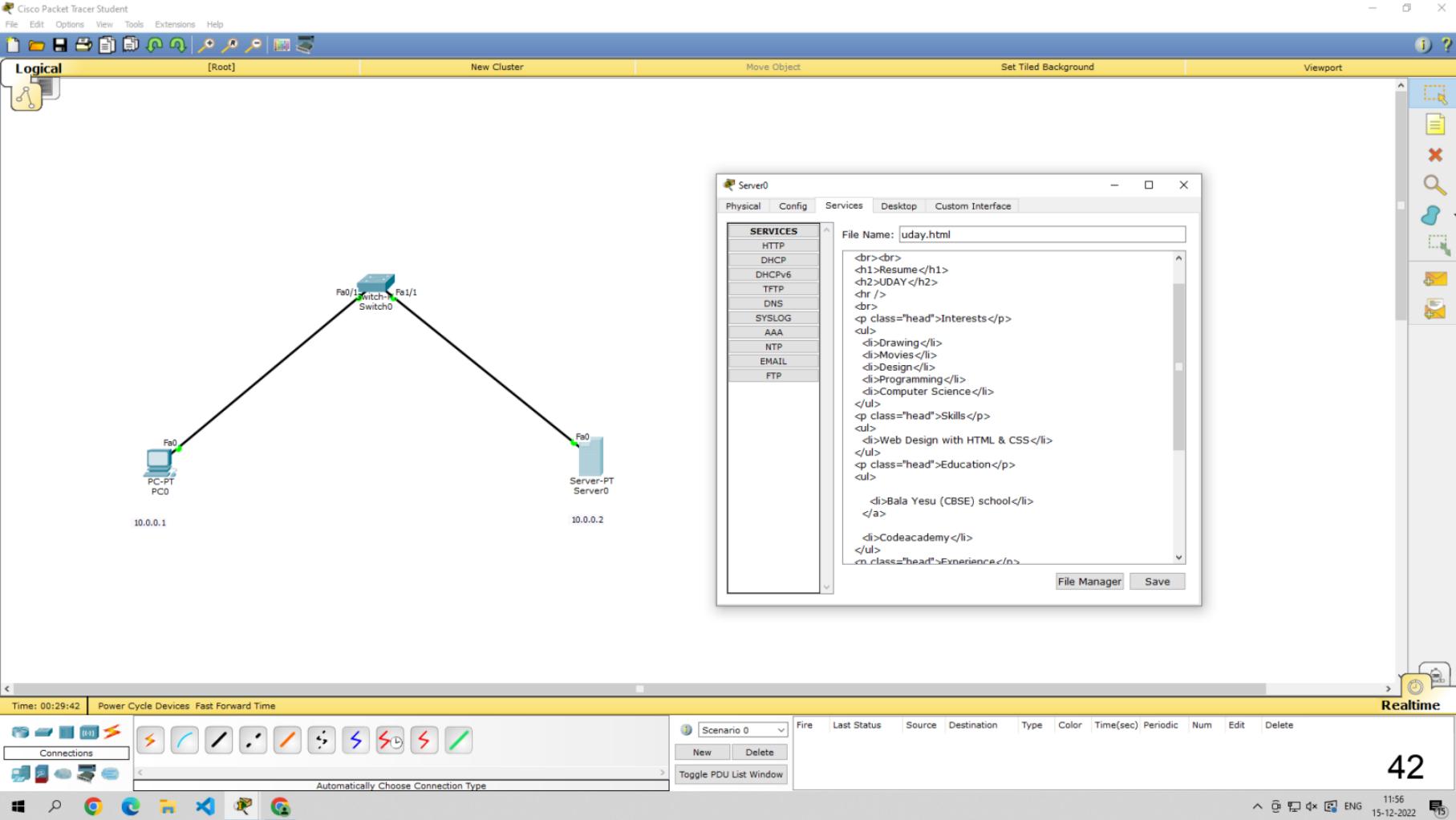
Experience

1 year

Uday
29/12/2022







Lab - 7

WAP for error detecting code using

#include < stdio.h >

```
void main() {
```

```
char input[100], key[30], temp[30];
quot[100], rem[30], key1[30];
printf ("Enter data");
gets (input);
printf ("Enter g(x) ");
gets (key);
crc (input, key);
crc (input, key);
```

}

- void crc (char input, char key) {

```
keylen = strlen(key);
msglen = strlen(input);
strcpy (key1, key);
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] = key[i[j]; 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);

```

```

steps - strcpy (temp, temp);
printf ("Quotient is ");
for (i = 0; i < msglen; i++)
    printf ("%c", quot[i]);
printf ("Rem");
for (i = 0; i < keylen; i++)
    printf ("%c", rem[i]);
getch();

```

Output:-

Enter data :- 1011010101

enter g(x) = 1010

~~quotient is = 1001000100~~

Reminder = 000

modified = 1011010101000

Checking for errors

enter data = 1010010101000

$$g(x) = 10^10$$

$$\text{Remainder} = 100$$

~~$$\frac{N(x)}{x^9 - 1} \rightarrow r^2$$~~

```
75 for (j=0;j<keylen;j++)  
76 key[j]=key1[j];  
77 for (j=keylen-1;j>0;j--) {  
78 if(temp[j]==key[j])  
79 rem[j-1]='0'; else  
80 rem[j-1]='1';  
81 }  
82 rem[keylen-1]=input[i+keylen];  
83 strcpy(temp,rem);  
84 }  
85 strcpy(rem,temp);  
86  
87 printf("%c",quot[i]);
```

input

```
/usr/include/stdio.h:577:14: note: declared here  
577 | extern char *gets (char *_s) __wur __attribute_deprecated__;  
| ^~~~  
/usr/bin/ld: /tmp/ccalni7s.o: in function `main':  
main.c:(.text+0x3a): warning: the `gets' function is dangerous and should not be used.  
Enter Data: 1011010101  
Enter G(x): 1010
```

```
Quotient is 1001000100  
Remainder is 000  
Modified data is: 1011010101000  
Remainder is at receiver side 000
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.
```

Write a program for Congestion Control using leaky bucket algorithm.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int input = 0, i = 0, bucketLimit = 400;
    int op = 1;
    printf("Bucket limit is 400\n Rate is 50 mbps\n");
    while (op) {
        printf("Enter the input\n");
        scanf("%d", &i);
        if (i == 400 || input == 400)
            input = input + i;
        else 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 0 to end\n");
        scanf("%d", &op);
    }
    return 0;
}
```

Output :-

Bucket limit is 400

Rate is 50mbps
enter the input

800

qty in bucket 250
pours 1 to add input 0 to end

300

~~qty in bucket~~

Bucket limit exceeded

pours 1 to add input 0 to end

end 1

Enter the input

50

qty in bucket 250

pours 1 to add input 0 to end

0

Now
12/1/22

```
PS C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents> cd "c:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\Leaky Bucket"
\" ; if ($?) { gcc leaky.c -o leaky } ; if ($?) { .\leaky }
Bucket limit is 400
Rate is 50mbps
enter the input
300
qty in bucket 250
press 1 to add input again 0 to end
0
PS C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\Leaky Bucket>
```

9. Bellman Ford

Implement Bellman Ford algorithm to compute the shortest path for
#include <stdio.h> a given topology
#include <stdlib.h>

```
int Bellman_Ford (int G[20][20], int V, int  
edge[20][20])
```

{

```
int i, u, v, k, distance[20], parent[20],  
flag = 1;
```

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

```
distance[i] = 1000; parent[i] = -1;  
printf ("Enter Source");
```

```
scanf ("%d", &s);
```

```
distance[s] = 0;
```

```
distance[s-1] = 0;
```

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

```
for (k = 0; k < E; k++) {
```

```
u = edge[k][0]; v = edge[k][1];
```

```
if (distance[u] + G[u][v] < distance[v])
```

```
distance[v] = distance[u] + G[u][v];
```

```
parent[v] = u;
```

```
}
```

```
for (k = 0; k < E; k++) {
```

```
u = edge[k][0]; v = edge[k][1];
```

```
if (distance[u] + G[u][v] <
```

```
distance[v])
```

```
flag = 0;
```

Outp

BE
E
E

```
if (flag)
    for (j = 0; j < V; j++)
        pointf("Vertex %d \rightarrow %d = real parent", i, j);
    cout << endl;
    return flag;
}

int main()
{
    int N, edge[20][20], G[20][20], i, j;
    i = 0;
    printf("Enter no of Vertices %d", N);
    scanf("%d", &N);
    printf("Enter graph in matrix form");
    for (i = 0; i < N; i++)
    {
        for (j = 0; j < N; j++)
        {
            scanf("%d", &G[i][j]);
            if (G[i][j] == 0)
                edge[i][j] = 0;
            else
                edge[i][j] = 1;
        }
    }

    if (Bellman_Ford(G, V, k, edge))
        printf("No negative weight cycle");
    else
        printf("Negative weight cycle exists");
    return 0;
}
```

Output:-

BELLMAN FORD.

Enter no. of Vertices = 5
Enter graph in matrix form

$$\begin{matrix} 0 & 6 & 0 & 7 & 0 \\ 0 & 0 & 5 & 8 & \rightarrow \\ 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & 9 \\ 2 & 0 & 7 & 0 & 6 \end{matrix}$$

Enter source = 1

Vertex 1 \rightarrow Cost = 0 parent = 0

Vertex 2 \rightarrow Cost = 2 parent = 3

Vertex 3 \rightarrow Cost = 4 parent = 4

Vertex 4 \rightarrow Cost = 7 parent = 1

Vertex 5 \rightarrow Cost = 2 parent = 2

W.A
 $2/2/10^4$

```
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

Outgoing Line: A B C D E
Hop Count: 1 0 99 99 99

Router Table entries for router C:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 1 99 0 1 1

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
```

10. dijkstra

Implement Dijkstra's algorithm to compute the shortest path for a given topology.

```
#include <stdio.h>
#include <conio.h>
#include <INFINITY>
#define MAX 10
```

```
void dijkstra( int G[MAX][MAX], int n,
                int startnode );
```

```
int main() {
```

```
    int G[MAX][MAX], i, j, n, u;
    printf("Enter no of vertices");
    scanf("%d", &n);
    printf("Enter the adjacency matrix");
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &G[i][j]);
    printf("Enter the starting node");
    scanf("%d", &u);
    dijkstra(G, n, u);
}
```

```
void dijkstra( int G[MAX][MAX],
                int n, int startnode )
{
```

```
    int cost[MAX][MAX], dist[10];
    pred[10];
```

int visited[MAX], count, mindistance, nextnode,
 i, j.
 for (i=0; i<n; i++)
 for (j=0; j<n; j++)
 if (grid[i][j] == 0)
 cost[i][j] = INFINITY;
 else
 cost[i][j] = grid[i][j];

for (i=0; i<n; i++)
 {
 distance[i] = cost[startnode][i];
 pred[i] = startnode;
 visited[i] = 0;
 }

distance[startnode] = 0;
 visited[startnode] = 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] <
 distance[i])
 narzo Shot by uday

distance $[i][j] = \min(\text{distance}) + \text{cost}[i][j]$
 $[i][j]$
 $\text{pred}[i][j] = \text{next node}$
 ...
 count++;
 for ($i = 0$; $i < n$; $i++$) {
 if ($i == \text{start node}$) {
 printf ("n", "Distance of node
 ", i , $d = \text{ad}["n", distance[i][j]]$);
 pointj [".\n path=" $d", i]);
 }
 do {
 $j = \text{pred}[i][j];$
 printf (" -> ", j);
 } while ($j != \text{start node}$);
 }$

Output:-

Enter the no. of vertices: 4
 Enter the adjacency matrix:
 0 1 1 1
 1 0 1 0
 1 1 0 1
 1 0 1 0

Wav 23
R2

Enter the starting node: 1
 Distance of 0 = 1
 Path = 0 \leftarrow 1

Distance of 2 = 1

Path = 0 \leftarrow 2 \leftarrow 1

narzo Shot by uday

Distance of 3 = 2

```
M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\distance vector\" ; if  
($?) { g++ dijstras.cpp -o dijstras } ; if ($?) { .\dijstras }
```

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
---	---

```
PS C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\distance ve  
ctor> █
```

II. TCP/IP

Using TCP/IP Socket Write a client server program to display contents of client.py file if present.

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("n Enter file name: ")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print("n from server(n")
print(fileContents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

while 1:

print("The server is ready to receive")

~~connectionSocket, addr = serverSocket.accept()~~
~~filename = connectionSocket.recv(1024).decode()~~
file = open(filename, "r")
d = file.read(1024)

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

Output:-

The server is ready to receive
contents of server TCP.py
the server is ready to receive

Enter filename:- serverTCP.py
from server:
from socket import *

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

✓
2/2/2025

ServerTCP.py - C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code docu... File Edit Format Run Options Window Help

```
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()
```

IDLE Shell 3.9.13 File Edit Shell Debug Options Window Help

```
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1  
929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents  
\CN\ServerTCP.py  
The server is ready to receive  
  
Sent contents of ServerTCP.py  
The server is ready to receive
```

IDLE Shell 3.9.13 File Edit Shell Debug Options Window Help

```
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()  
>>>
```

ClientTCP.py - C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\ClientT... File Edit Format Run Options Window Help

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000  
  
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

Q. UDP

Using UDP sockets, write a client server program to make client send the file name and the server to send back the content of the requested file by pressing Ctrl+C.

```

from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name")
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("Reply from server!")
print(fileContent.decode("utf-8"))
clientSocket.close()
clientSocket.close()

```

Server.py

```

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while True:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    print(sentence)
    file = open(sentence.decode("utf-8"), "r")
    l = file.read(2048)
    file.close()

```

```
serverSocket.sendto(bytess(1, "utf-8"),  
                     clientAddress)  
print('client contents of ', end = ' '))  
print('contents')  
file.close()
```

Output :-

The server is ready to receive
and contents of server.py

Enter filename:- server UDP.py

Reply from server:

from socket import *

ServerPort = 12000

serverSocket = socket(AF_INET, SOCK_DGRAM)

serverSocket.sendto(bytess(1, "utf-8"), clientAddress)
file.close()

WAP
TCP/IP

ClientUDP.py - C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\UDP\C... -

```
File Edit Format Run Options Window Help
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ("\nReply from Server:\n")
print (filecontents.decode("utf-8"))
# for i in filecontents:
# print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```

IDLE Shell 3.9.13

```
File Edit Shell Debug Options Window Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\UDP\Se
erUDP.py
The server is ready to receive
Sent contents of ServerUDP.py
```

IDLE Shell 3.9.13

```
File Edit Shell Debug Options Window Help
ntUDP.py
Enter file name: ServerUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    l=file.read(2048)
    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print ('\nSent contents of ', end = ' ')
    print (sentence)
    # for i in sentence:
    # print (str(i), end = '')
    file.close()
>>>
```

ServerUDP.py - C:\Users\M UDAYA RAJ\OneDrive\Documents\vs code documents\CN\UDP\...
File Edit Format Run Options Window Help
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
 sentence, clientAddress = serverSocket.recvfrom(2048)
 sentence = sentence.decode("utf-8")
 file=open(sentence,"r")
 l=file.read(2048)
 serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
 print ('\nSent contents of ', end = ' ')
 print (sentence)
 # for i in sentence:
 # print (str(i), end = '')
 file.close()