VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

AFIFAH KHAN (1BM20CS195)

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 **AFIFAH KHAN (1BM20CS195),** 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. Nandini Vineeth
Asisstant Professor
Department of CSE
BMSCE, Bengaluru

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

`

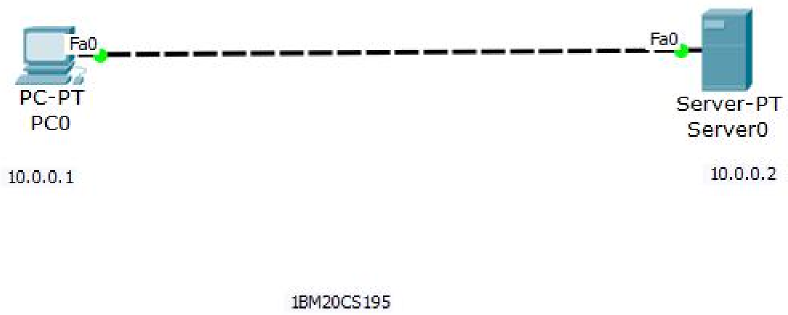
Index

Sl. No.	Date	Experiment Title	Page No.
		Cycle - 1	
0.	1-11-22	My First CN Lab.	4-7
1.	7-11-22	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	8-14
2.	14-11-22	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply	15-23
3.	19-11-22	Configuring default route to the Router	24-31
4.	19-11-22	Configuring DHCP within a LAN in a packet Tracer	32-36
5.	5-12-22	Configuring RIP Routing Protocol in Routers	37-43
6.	12-12-22	Demonstration of WEB server and DNS using Packet Tracer	44-49
		Cycle - 2	
7.	19-12-22	Write a program for error detecting code using CRC-CCITT (16-bits).	50-52
8.	9-1-23	Write a program for distance vector algorithm to find suitable path for transmission.	53-55
9.	26-12-22	Implement Dijkstra’s algorithm to compute the shortest path for a given topology.	56-59
10.	2-1-23	Write a program for congestion control using Leaky bucket algorithm.	60-61
11.	17-1-23	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.	62-63
12.	17-1-23	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.	64-66

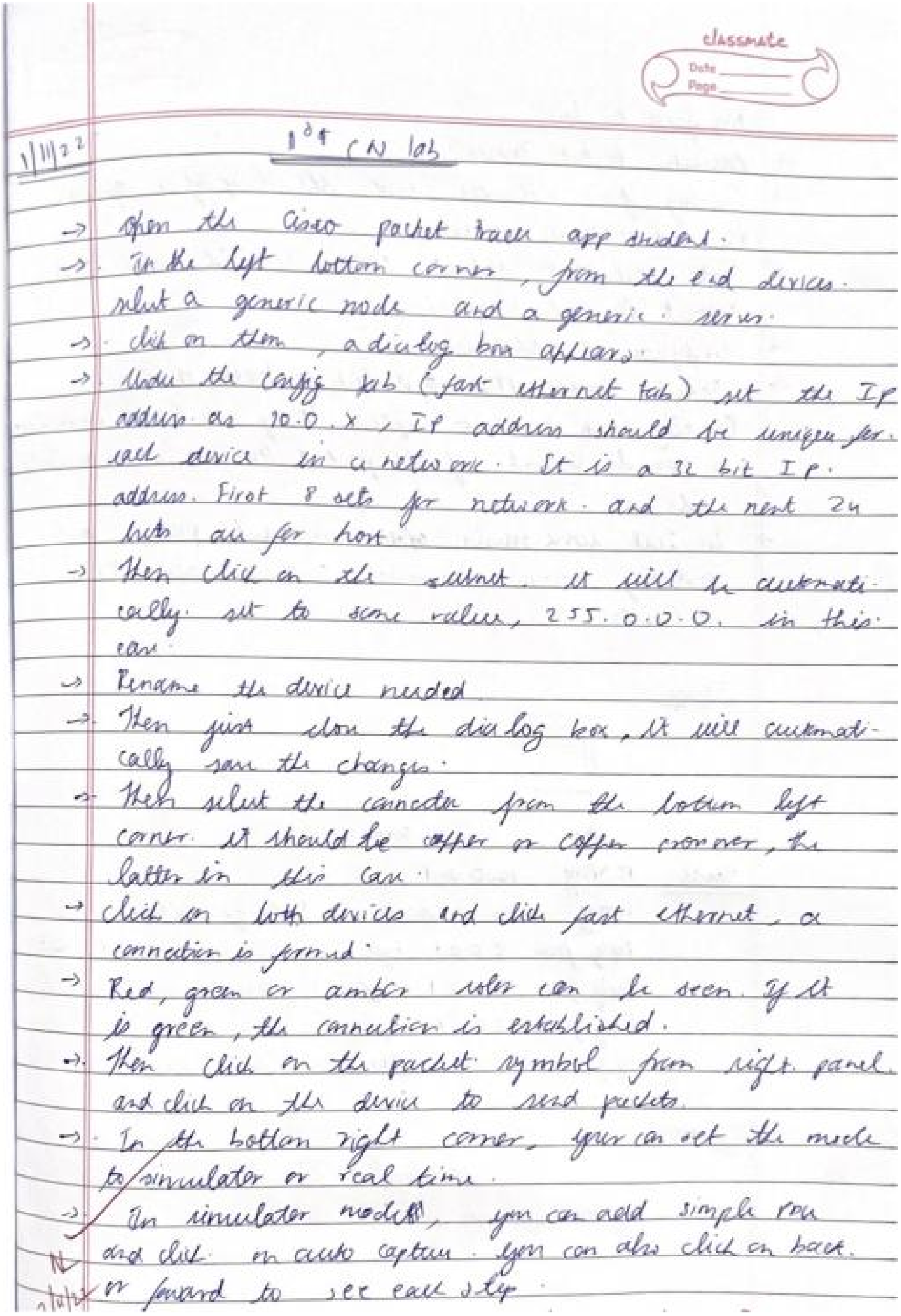
Experiment - 0

Aim : To understand the working of Cisco Packet Tracer and simulate sending simple PDU from source to destination.

Topology :



Procedure :



classmate

Date ________

Page ________

1/11/22

1st CN lab

-> Open the Cisco packet tracer app student.

-> In the left bottom corner, from the end devices. select a generic node and a generic server.

-> click on them, a dialog box appears.

-> Under the config tab (fast ethernet tab) set the IP address as 10.0.x, IP address should be unique for each device in a network. It is a 32 bit IP. address. First 8 bits for network. and the next 24 bits are for host

-> Then click on the subnet. it will be automatically set to some value, 255.0.0.0. in this case

-> Rename the device needed

-> Then just close the dialog box, it will automatically save the changes.

-> Then select the connector from the bottom left corner. it should be copper or copper crossover, the latter in this case.

-> click on both devices and click fast ethernet, a connection is formed.

-> Red, green or amber color can be seen. If it is green, the connection is established.

-> Then click on the packet symbol from right panel. and click on the device to send packets.

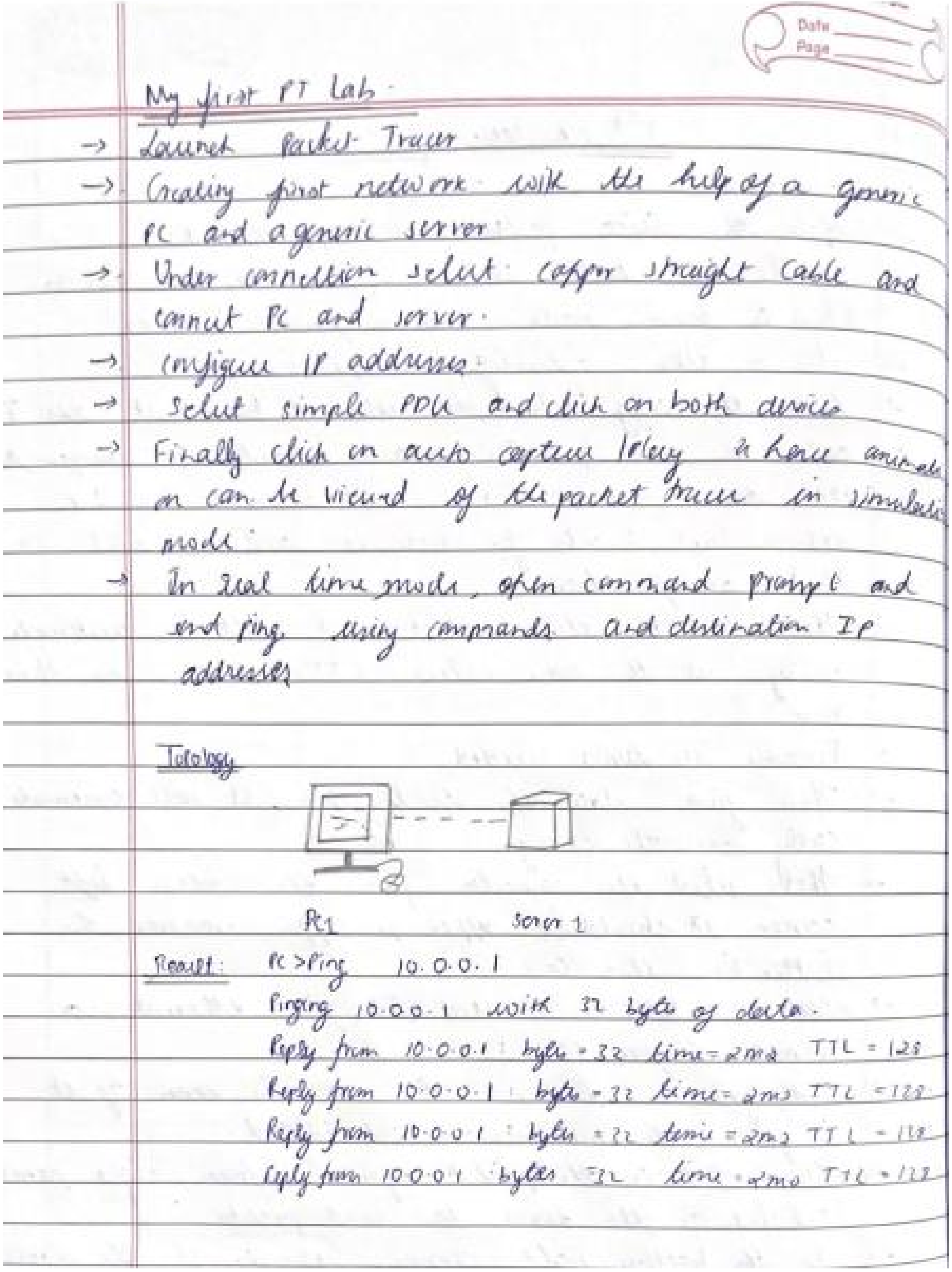
-> In the bottom right corner, you can set the mode to simulator or real time.

-> In simulator mode, you can add simple pdu and click on auto capture. you can also click on back. or forward to see each step.

N

2/11/22

5



Date
Page

My first PT Lab.

- → Launch Packet Tracer
- → Creating first network with the help of a generic PC and a generic server
- → Under connection select copper straight cable and connect PC and server.
- → configure IP addresses.
- → Select simple PDU and click on both devices
- → Finally click on auto capture /play & hence animation can be viewed of the packet trace in simulation mode.
- → In real time mode, open command prompt and send ping using commands and destination IP addresses

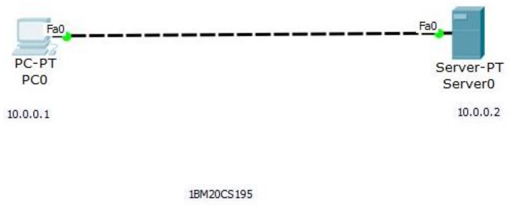
Topology

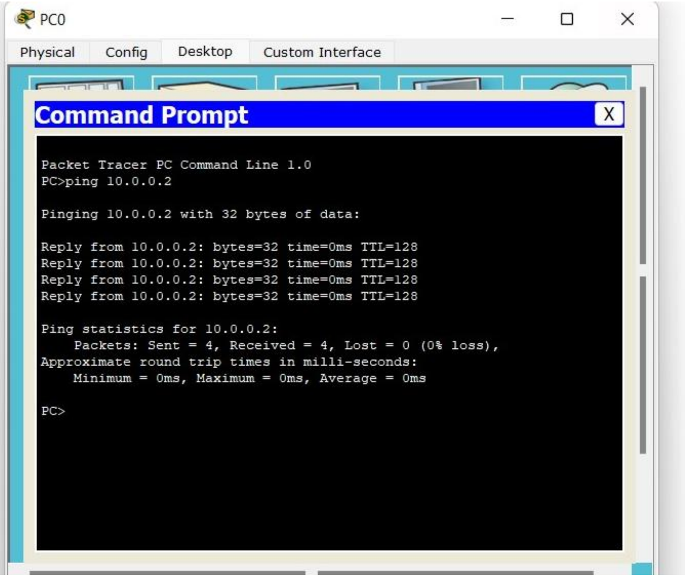
PC1 Server 1

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=2ms TTL=128
Reply from 10.0.0.1 bytes=32 time=2ms TTL=128
Reply from 10.0.0.1 bytes=32 time=2ms TTL=128
Reply from 10.0.0.1 bytes=32 time=2ms TTL=128
```

Snapshot of Output :



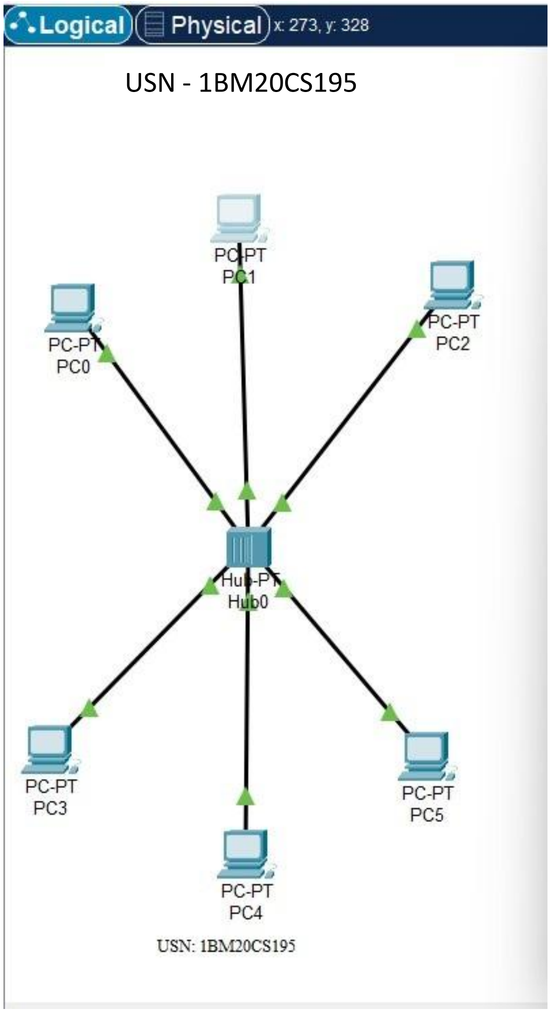


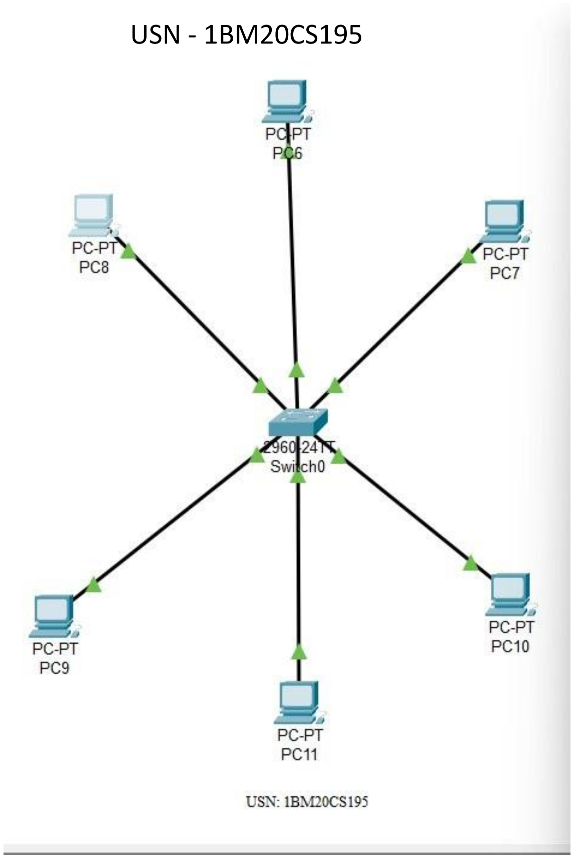
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

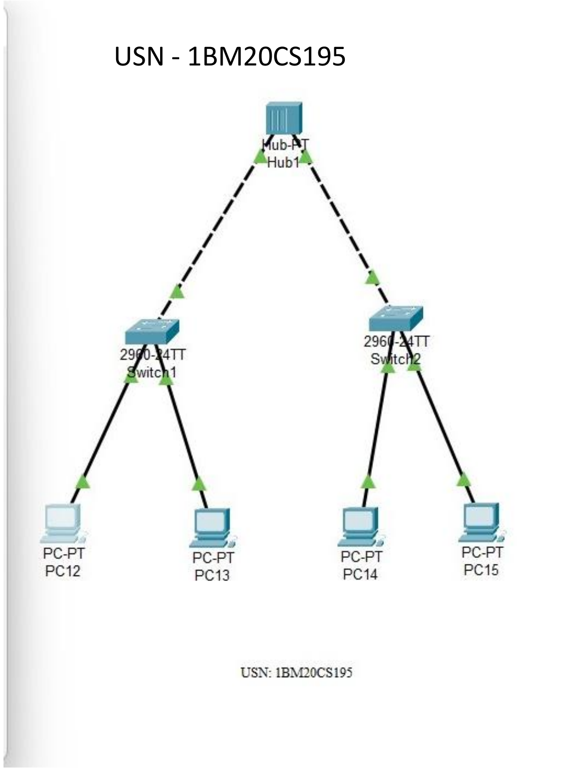
Experiment - 1

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

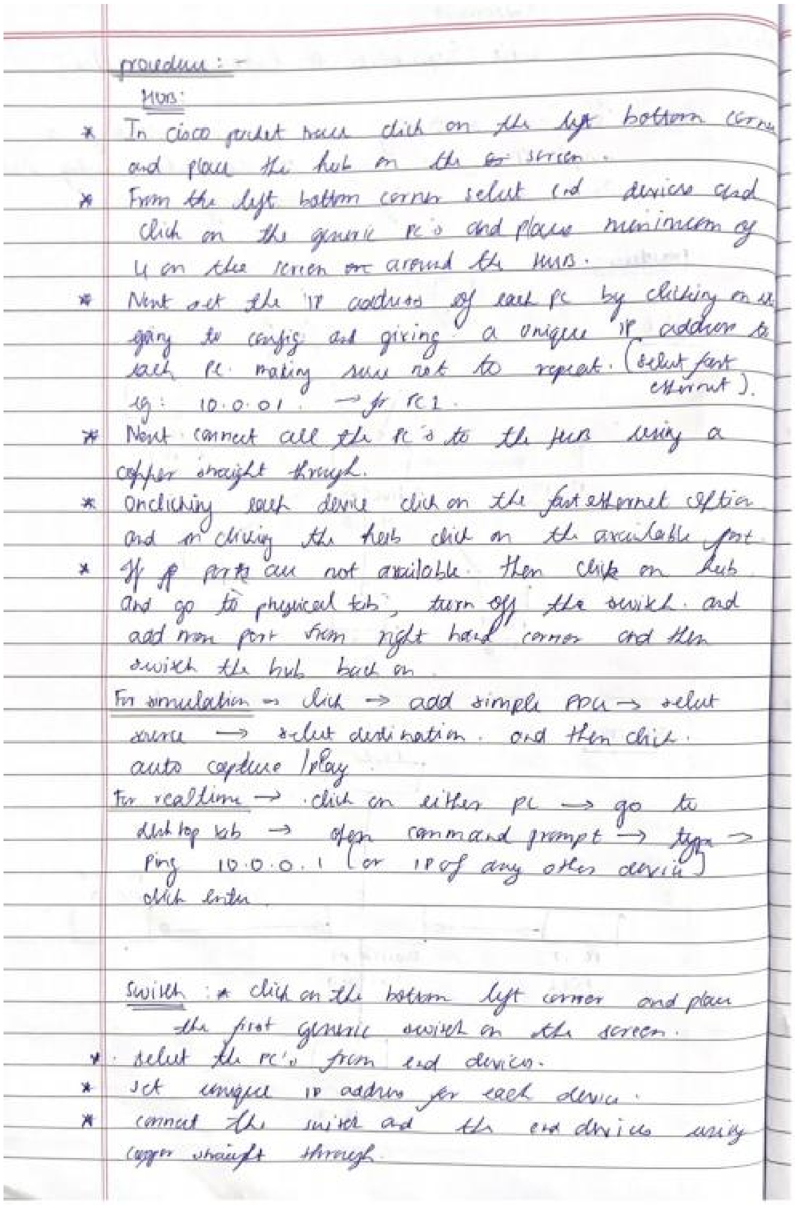
Topology :




USN - 1BM20CS195
PC-PT
PC6
PC-PT
PC8
PC-PT
PC7
2960-24TT
Switch0
PC-PT
PC9
PC-PT
PC10
PC-PT
PC11
USN: 1BM20CS195


USN - 1BM20CS195
Hub-PT
Hub1
2960-24TT
Switch1
2960-24TT
Switch2
PC-PT
PC12
PC-PT
PC13
PC-PT
PC14
PC-PT
PC15
USN: 1BM20CS195

Procedure :



procedure :-

HUB:

* In cisco packet tracer click on the left bottom corner and place the hub on the screen.

* From the left bottom corner select end devices and click on the generic pc's and place minimum of 4 on the screen ~~on~~ around the HUB.

* Next set the IP address of each pc by clicking on it going to config and giving a unique IP address to each pc making sure not to repeat. (select fast ethernet).
eg: 10.0.01 → for PC1.

* Next connect all the pc's to the hub using a copper straight through.

* Onclicking each device click on the fast ethernet option and on clicking the hub click on the available port

* If ports are not available then click on hub and go to physical tab, turn off the switch and add more port from right hand corner and then switch the hub back on

For simulation → click → add simple PDU → select source → select destination and then click auto capture / play

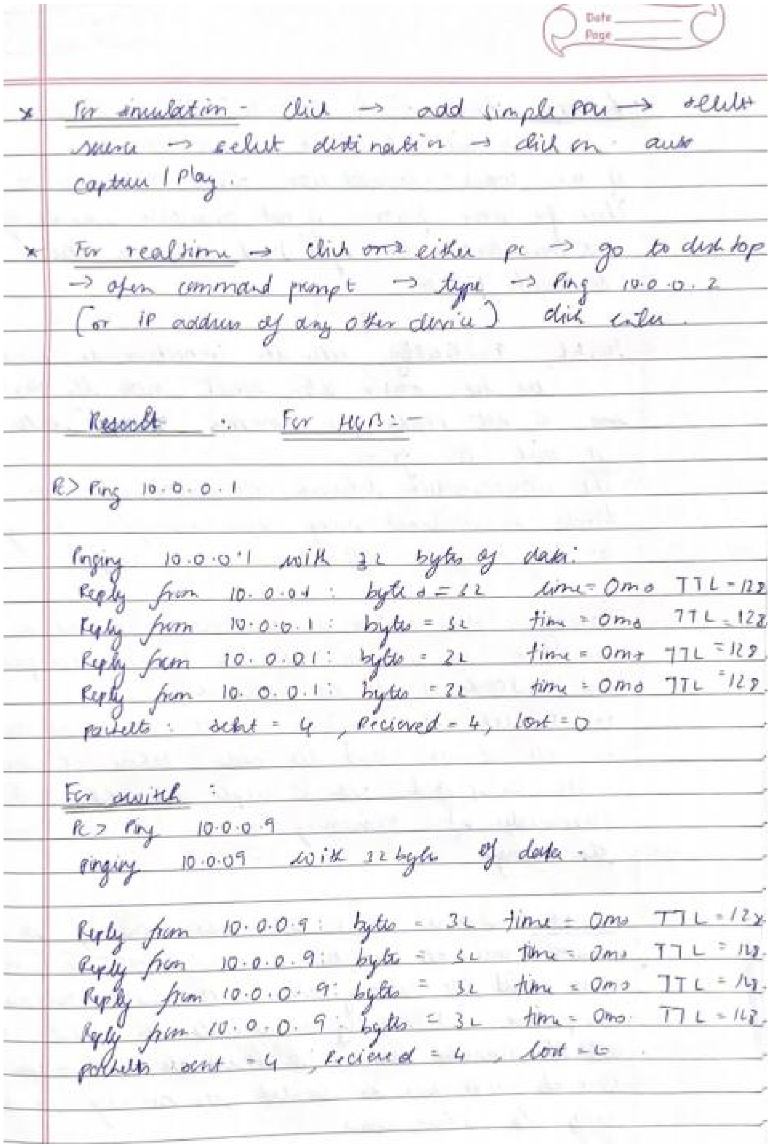
For realtime → click on either pc → go to desktop tab → open command prompt → type → ping 10.0.0.1 (or IP of any other device) click enter.

Switch : * click on the bottom left corner and place the first generic switch on the screen.

* select the pc's from end devices.

* set unique IP address for each device.

* connect the switch and the end devices using copper straight through



- For simulation - click → add simple PDU → select source → select destination → click on auto capture / Play.

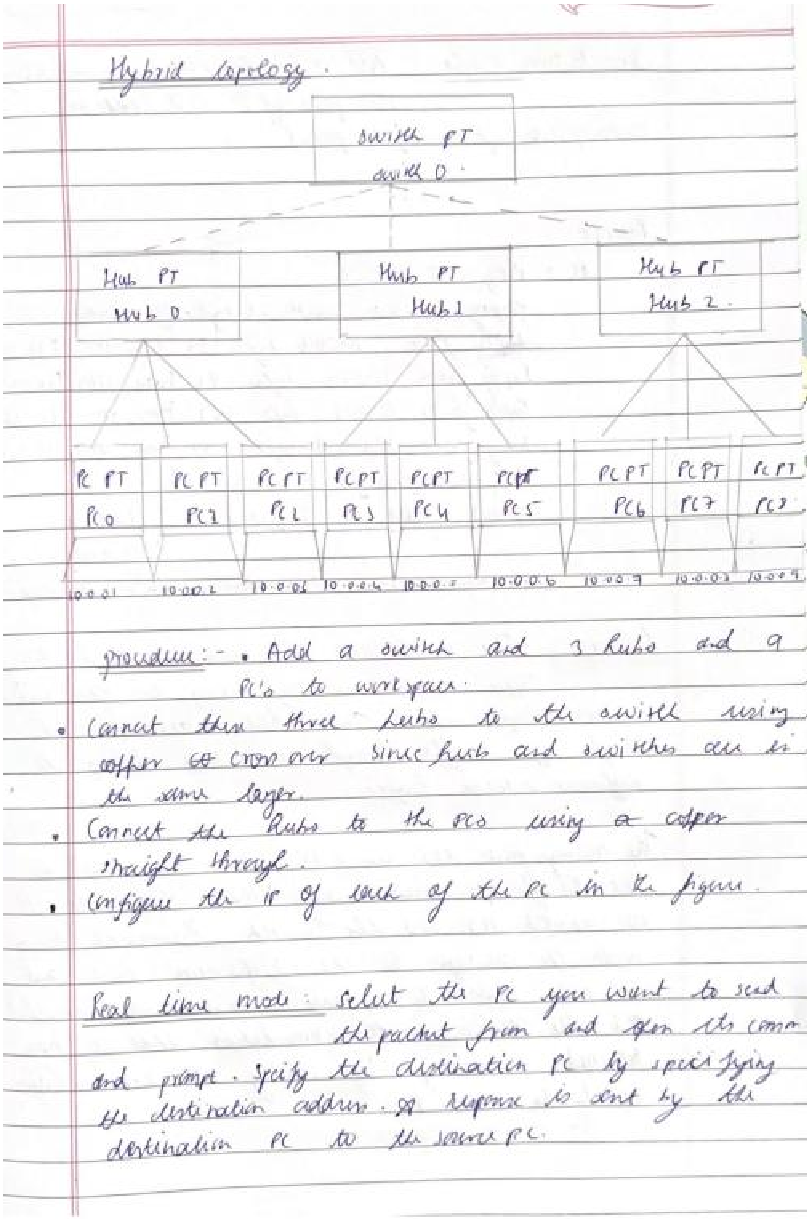
- For realtime → click on either pc → go to desktop → open command prompt → type → Ping 10.0.0.2 (or IP address of any other device) click enter.

Result : For HUB :-

```
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 = 0ms TTL = 128
 Reply from 10.0.0.1 : bytes = 32   time = 0ms TTL = 128
 Reply from 10.0.01 : bytes = 32    time = 0ms TTL = 128
 Reply from 10.0.0.1 : bytes = 32   time = 0ms TTL = 128
 packets : sent = 4 , Recieved = 4, lost = 0
```

For switch :

```
PC> Ping 10.0.0.9
 Pinging 10.0.09 with 32 bytes of data.

 Reply from 10.0.0.9: bytes = 32 time = 0ms TTL = 128
 Reply from 10.0.0.9: bytes = 32 time = 0ms TTL = 128
 Reply from 10.0.0.9: bytes = 32 time = 0ms TTL = 128
 Reply from 10.0.0.9: bytes = 32 time = 0ms TTL = 128
 packets sent = 4 , Recieved = 4 , lost = 0
```



switch PT
switch 0
Hub PT
Hub 0
Hub PT
Hub 1
Hub PT
Hub 2
PC PT
PC0
PC PT
PC1
PC PT
PC2
PC PT
PC3
PC PT
PC4
PC PT
PC5
PC PT
PC6
PC PT
PC7
PC PT
PC8
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
10.0.0.8
10.0.0.9

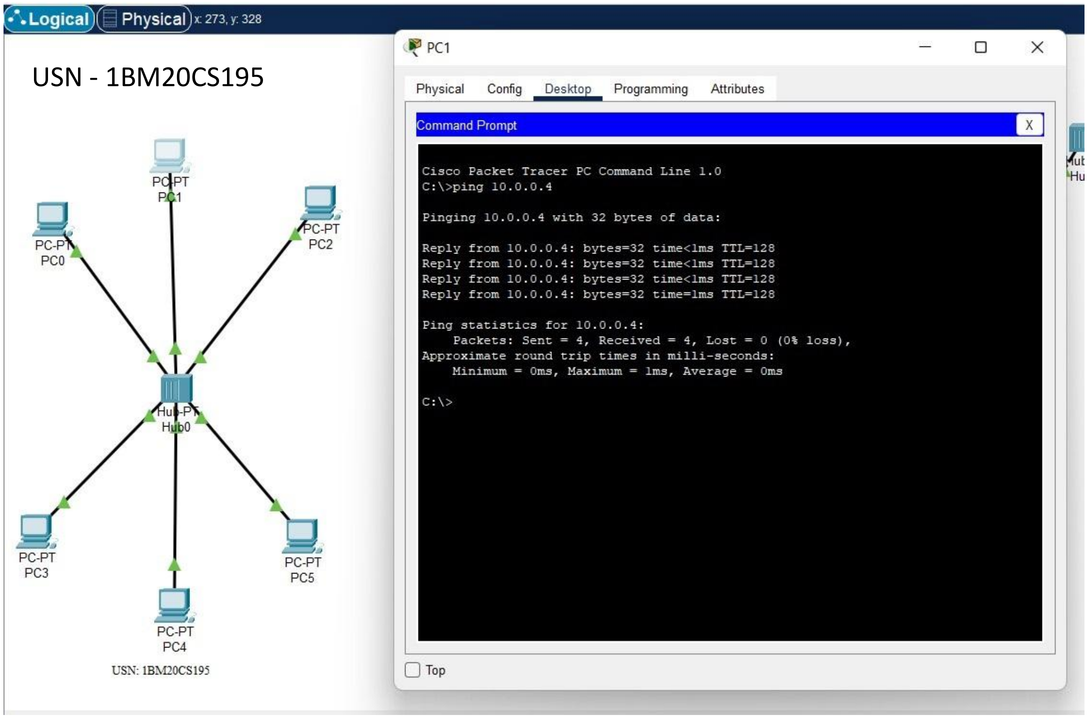
Hybrid topology.

procedure:- • Add a switch and 3 hubs and 9 PC's to workspace.

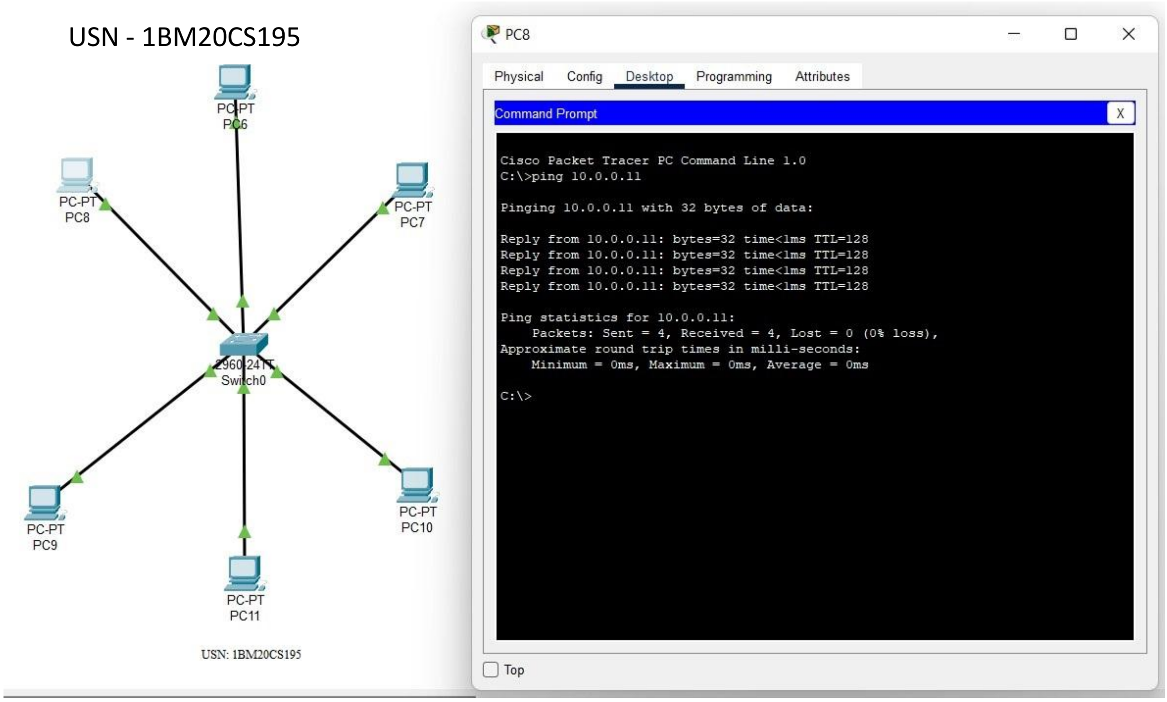
- Connect these three hubs to the switch using copper ~~co~~ cross over since hub and switches are in the same layer.
- Connect the hubs to the PCs using ~~a~~ copper straight through.
- Configure the IP of each of the PC in the figure.

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 destination address. ~~A~~ Response is sent by the destination PC to the source PC.

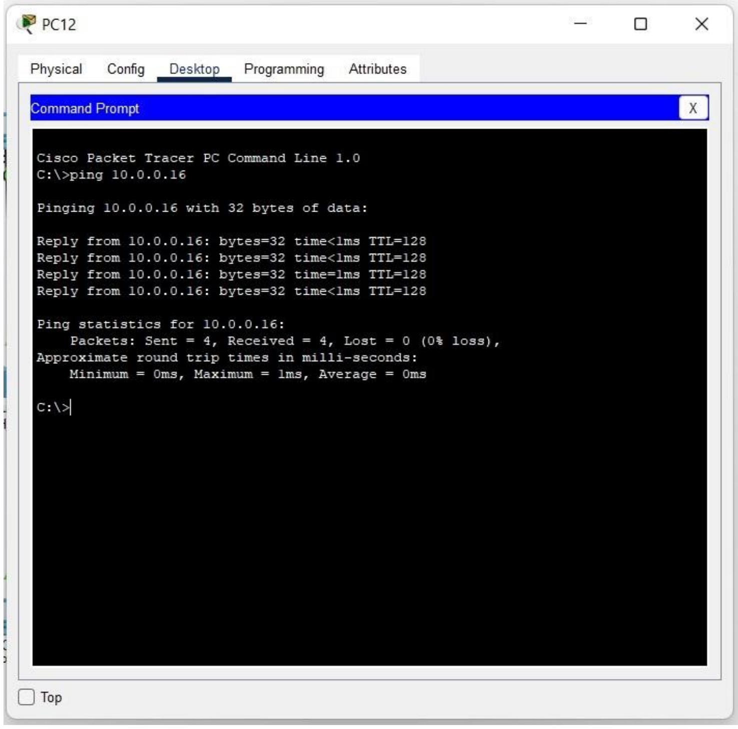
Snapshot of Output :



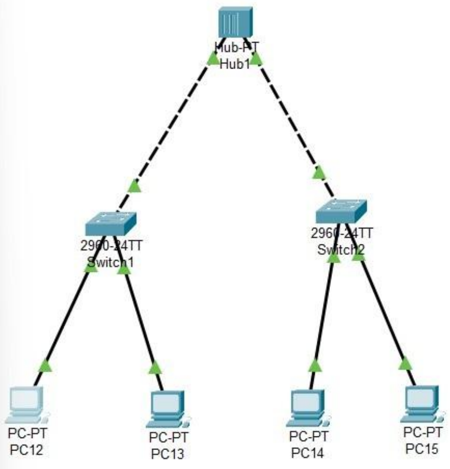
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data:

Reply from 10.0.0.11: bytes=32 time<1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128
Reply from 10.0.0.11: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.16

Pinging 10.0.0.16 with 32 bytes of data:

Reply from 10.0.0.16: bytes=32 time<1ms TTL=128
Reply from 10.0.0.16: bytes=32 time<1ms TTL=128
Reply from 10.0.0.16: bytes=32 time=1ms TTL=128
Reply from 10.0.0.16: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.16:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

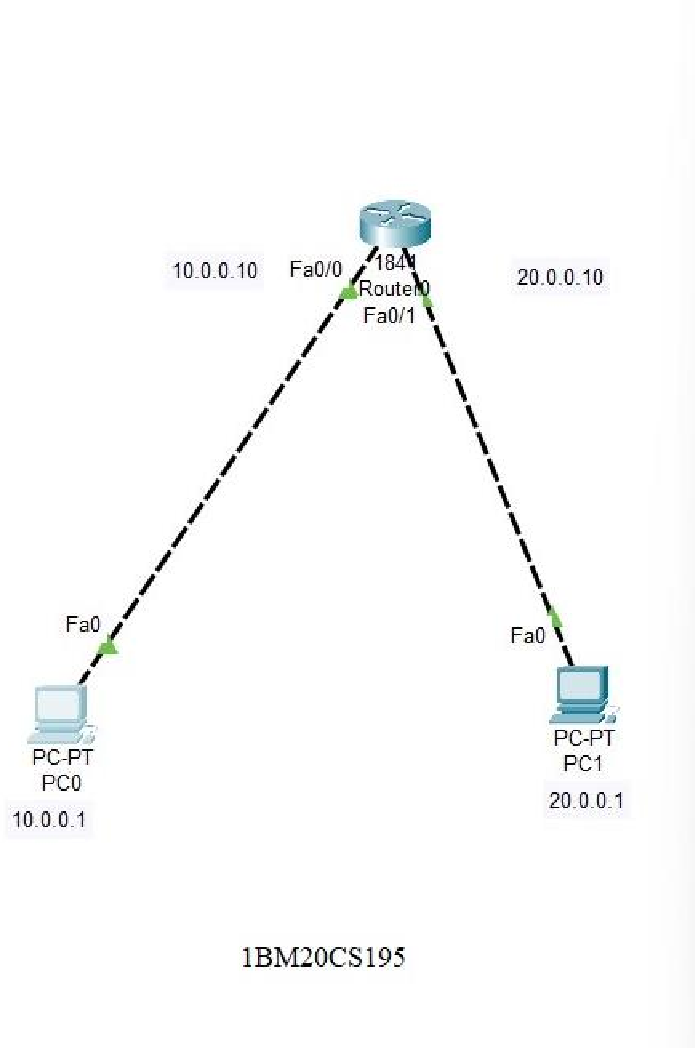


USN: 1BM20CS195

Experiment - 2

Aim : Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.

Topology :



Procedure :

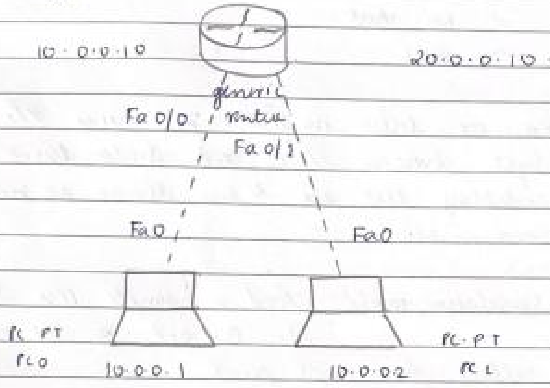
Experiment 2:

14/11/24

Title: Experiment using Routers and PCs.

Aim: Configuring IP address to routers in Packet tracer & Explore the following messages: Ping Responses, destination unreachable, Request timed out, Reply.

Topology:



Procedure: ⇒ Place a generic router and two generic PCs in your workspace.

⇒ Connect the router and PCs using copper cross over

⇒ Configure IP address of each PC and

⇒ in the configuration tab under settings set gateway for both PCs to the sole router

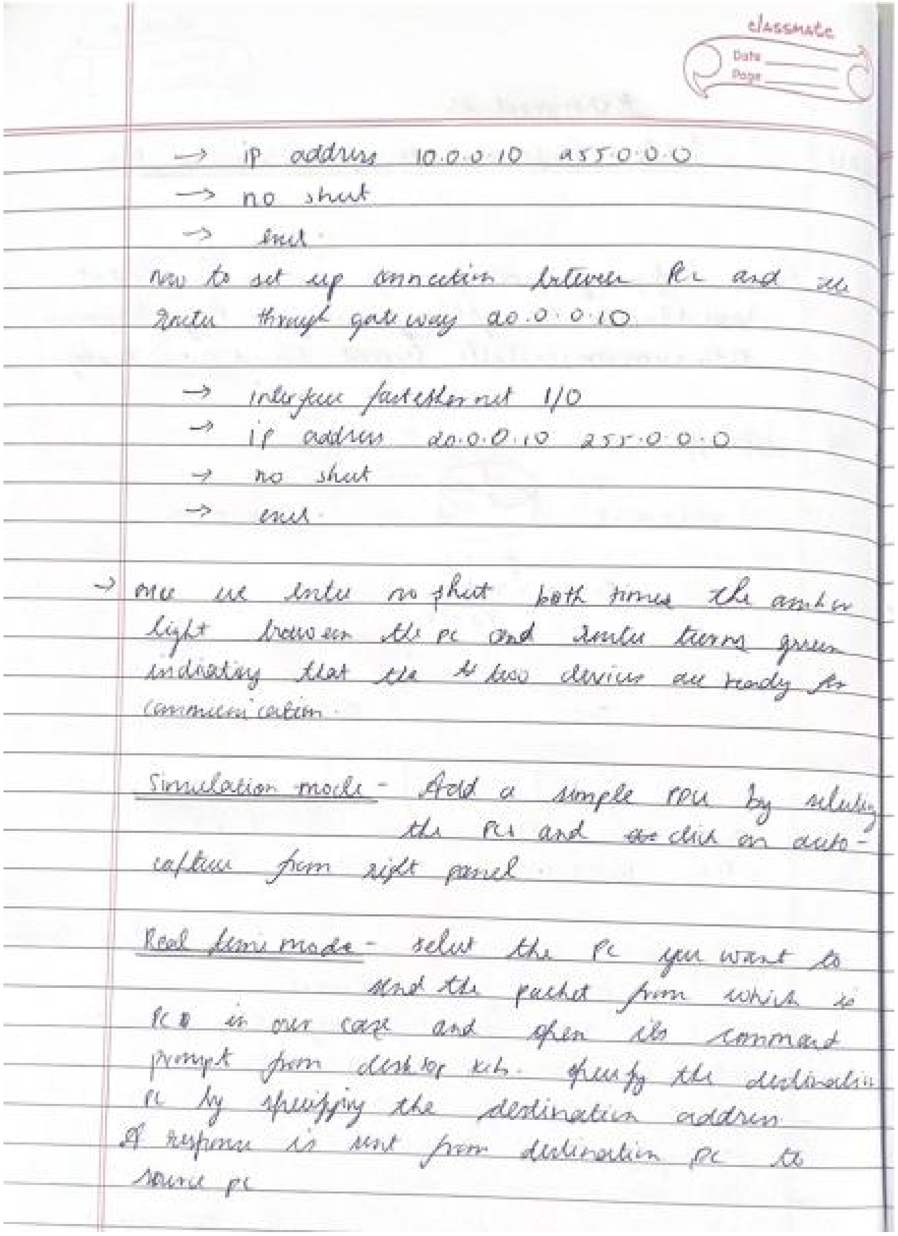
⇒ Click on the generic router and go the CLI tab. enter the following commands to set up a connection between PCs and generic router through gateway 10.0.0.10.

→ NO

→ enable

→ config t

→ interface fastethernet 0/0



→ ip address 10.0.0.10 255.0.0.0
→ no shut
→ exit.

now to set up connection between PC and the router through gateway 20.0.0.10

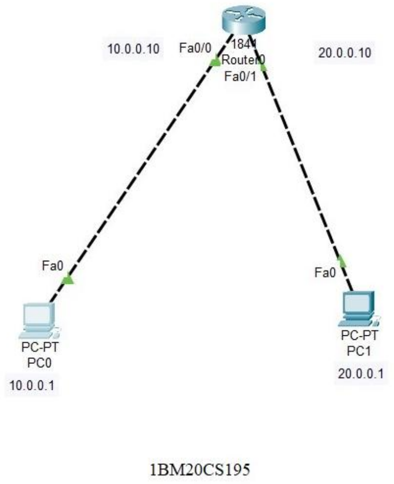
→ interface fastethernet 1/0
→ ip address 20.0.0.10 255.0.0.0
→ no shut
→ exit.

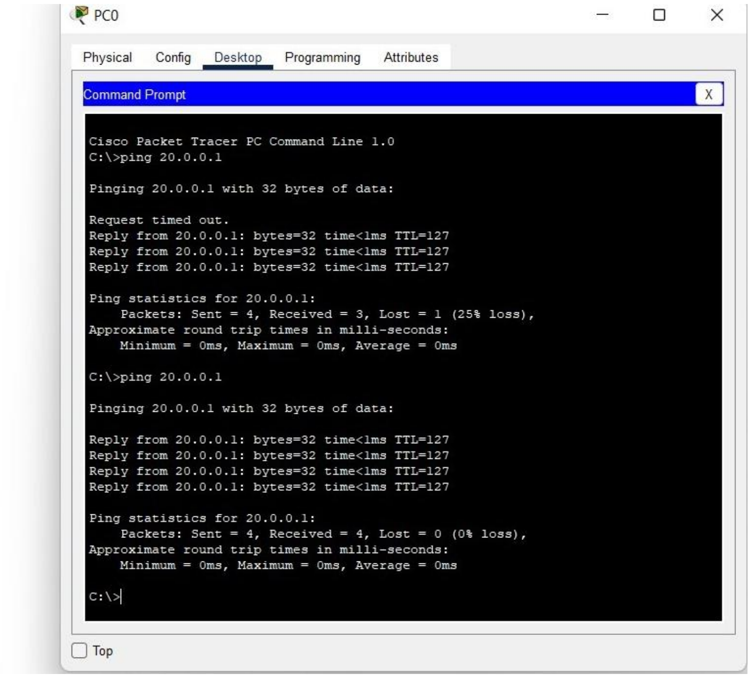
→ once we enter no shut both times the amber light between the pc and router turns green indicating that the ~~a~~ two devices are ready for communication.

Simulation mode - Add a simple PDU by selecting the PC and ~~so~~ click on auto-capture from right panel

Real time mode - select the PC you want to send the packet from which is PC0 in our case and open its command prompt from desktop tab. Specify the destination PC by specifying the destination address
A response is sent from destination PC to source PC

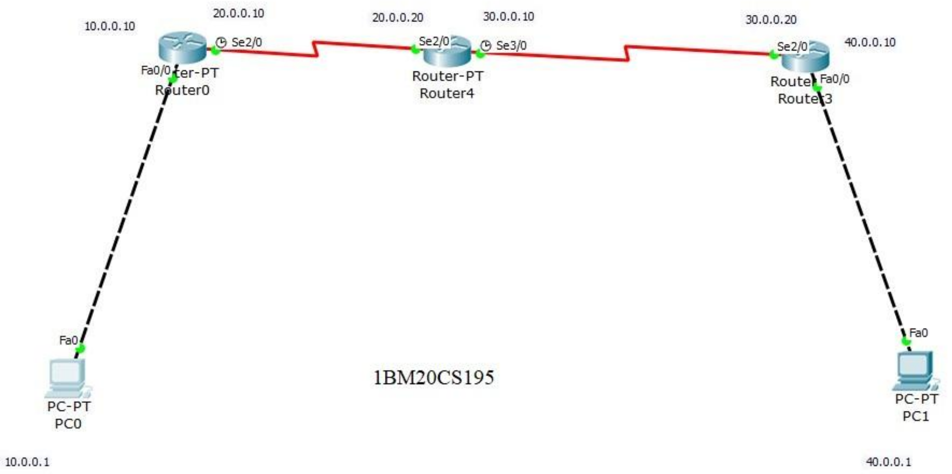
Snapshot of Output :





```
Cisco Packet Tracer PC Command Line 1.0
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:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 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 = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

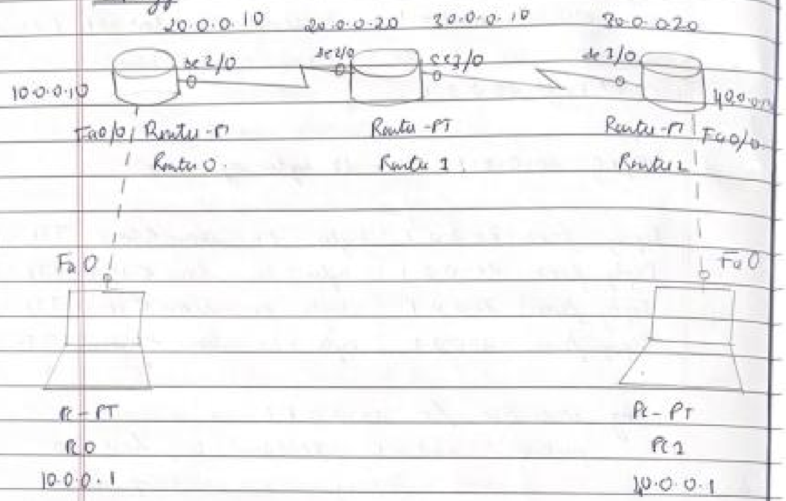
Topology :



Procedure :

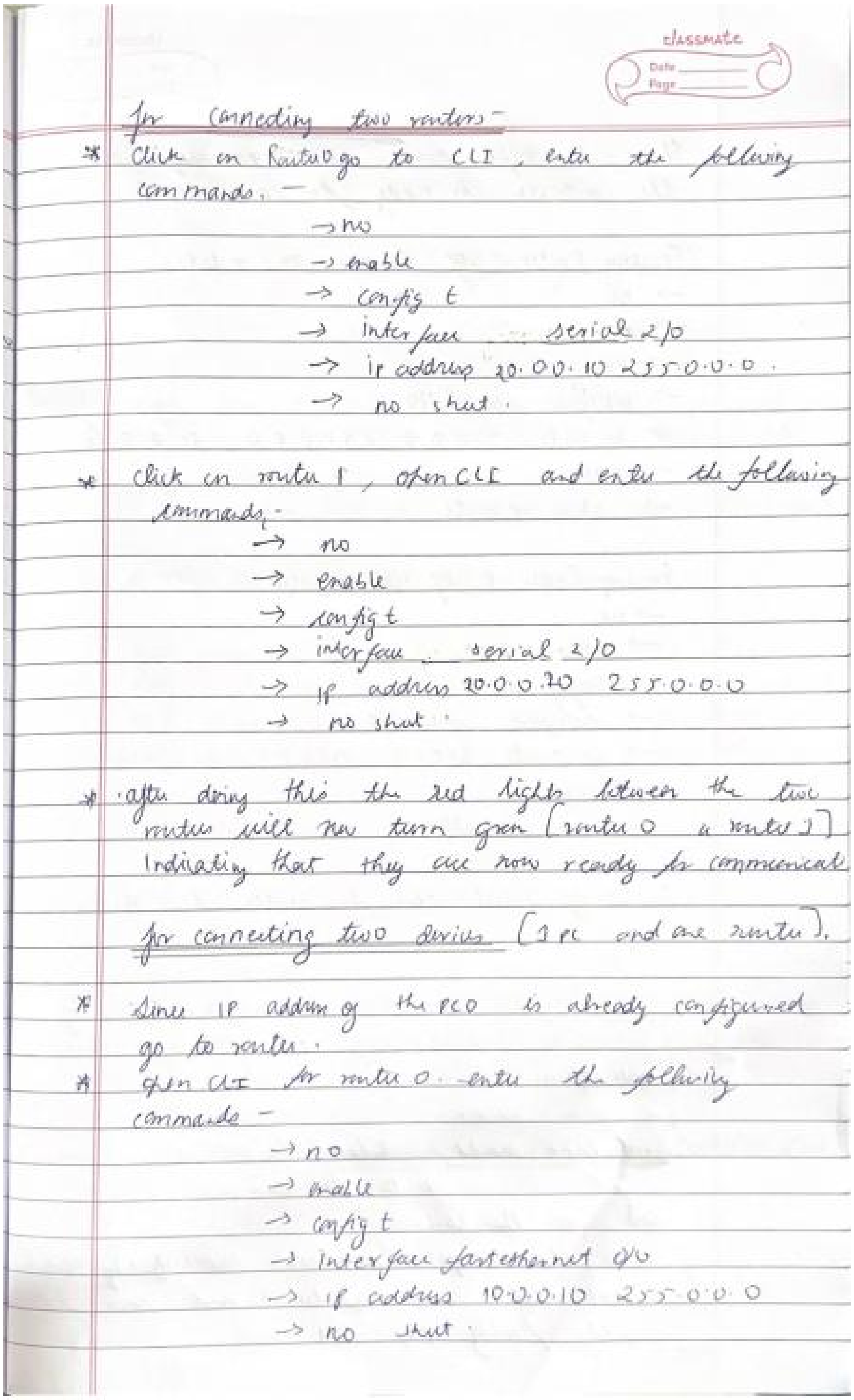
with multiple routers and multiple PCs

Topology:



Procedure: * Place 3 generic routers and 2 generic PC's in the workspace.

* Place a note for each device (PC and router) and specify the IP address.
* Connect the router and PC using copper crossover.
* Connect the routers using serial DCE
* Click on each PC, go to the configure tab set the IP address and subnet mask in fast ethernet 0.
* Next click on settings in the config tab set the gateway as the IP address of the next router. (eg: 10.0.0.10).
* IP address of PC and its gateway address should belong to the same network.



for connecting two routers -

* Click on Router0 go to CLI, enter the 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 and enter the following commands:-

```
→ no
→ enable
→ config t
→ interface serial 2/0
→ ip address 20.0.0.20 255.0.0.0
→ no shut.
```

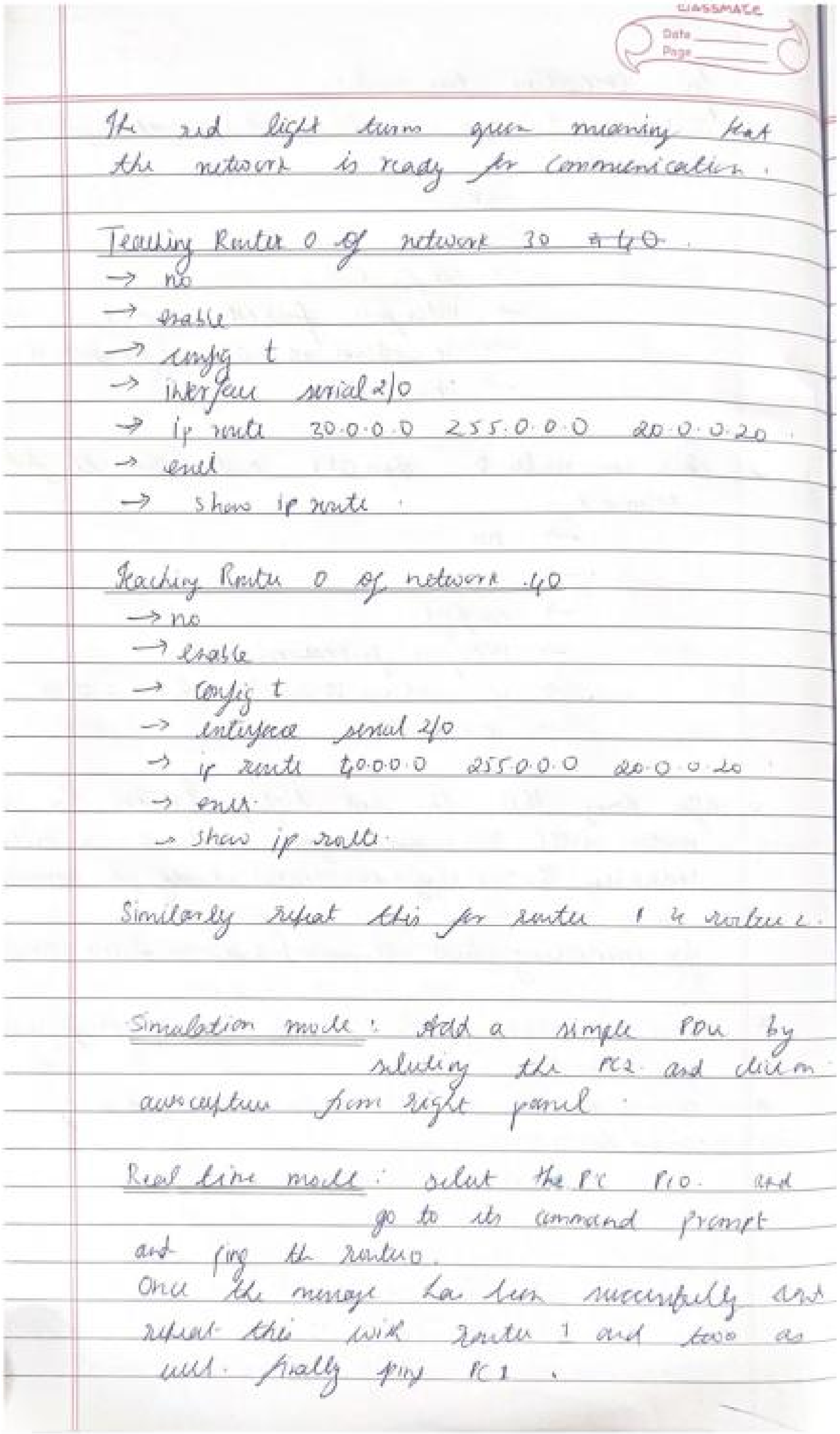
* after doing this the red lights between the two routers will now turn green [router 0 & router 1] Indicating that they are now ready for commonical

for connecting two devices (1 pc and one router).

* Since IP address of the PC0 is already configured go to router.

* open CLI for router 0. enter the following commands -

```
→ no
→ enable
→ config t
→ interface fastethernet 0/0
→ ip address 10.0.0.10 255.0.0.0
→ no shut.
```



The red light turns green meaning that the network is ready for communication.

Teaching Router 0 of network 30 & 40.

```
→ no
→ enable
→ config t
→ interface serial2/0
→ ip route 30.0.0.0 255.0.0.0 20.0.0.20
→ end
→ show ip route
```

Teaching Router 0 of network 40

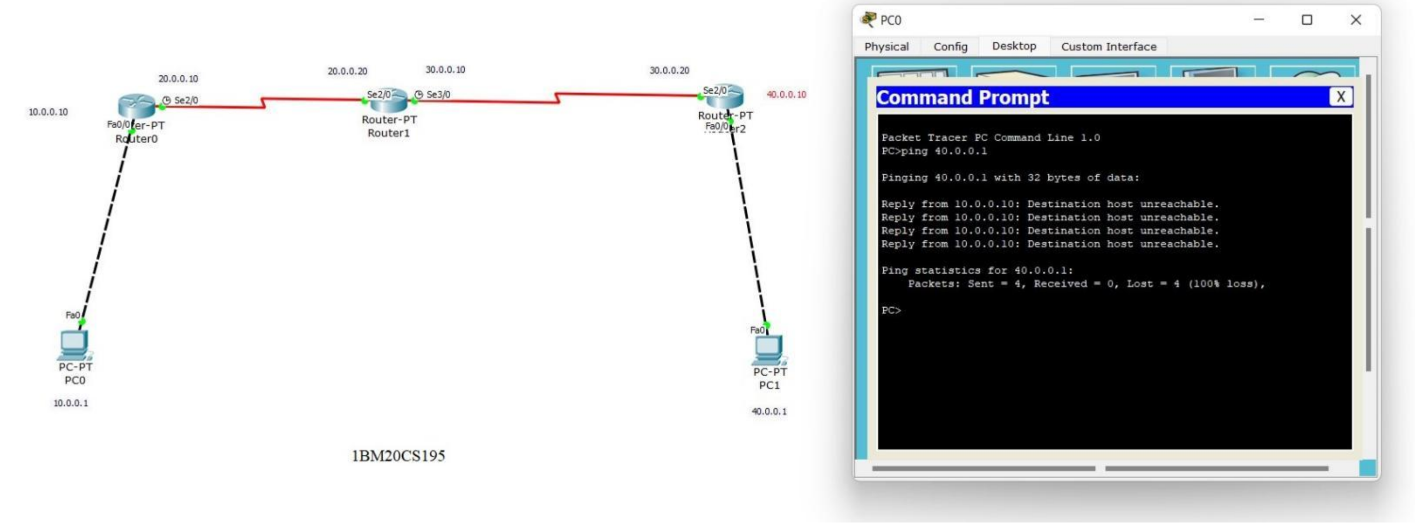
```
→ no
→ enable
→ config t
→ interface serial 2/0
→ ip route 40.0.0.0 255.0.0.0 20.0.0.20
→ end
→ show ip route
```

Similarly repeat this for router 1 & router 2.

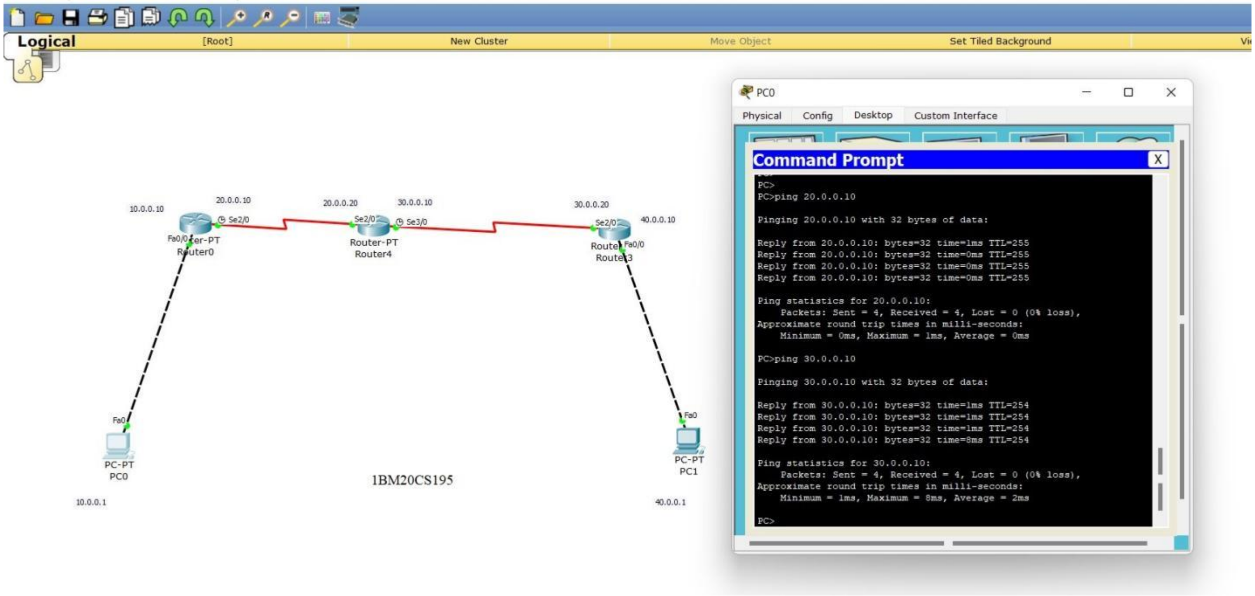
Simulation mode: Add a simple PDU by selecting the PC2 and click on autocapture from right panel.

Real time mode: select the PC PC0 and go to its command prompt and ping the router0.
Once the message has been successfully sent repeat this with router 1 and two as well. finally ping PC1.

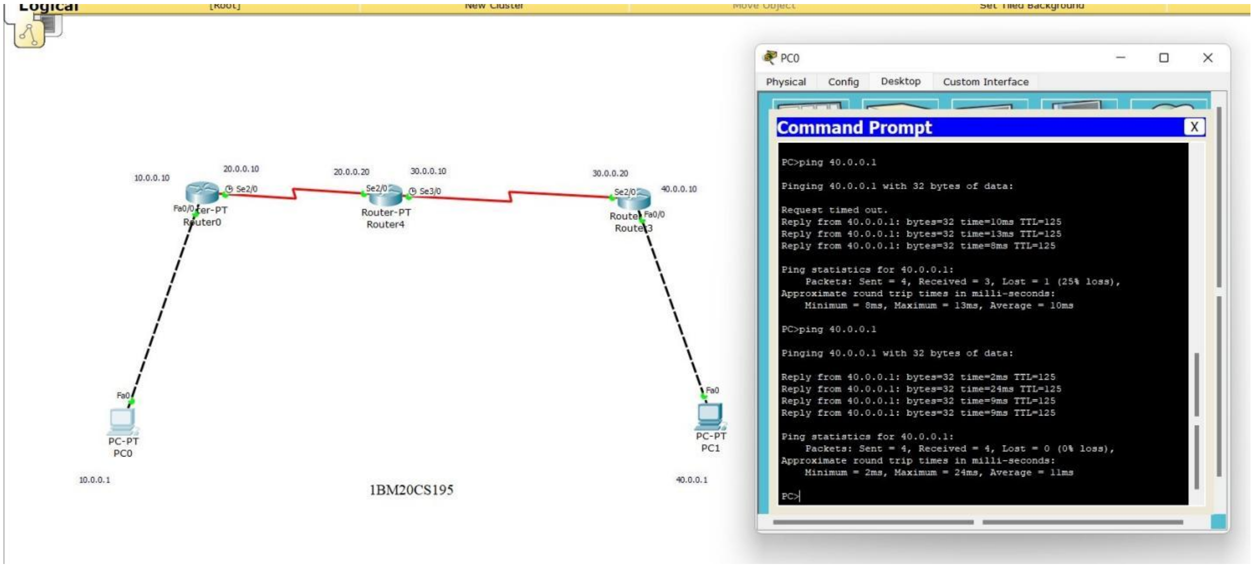
Snapshot of Output :



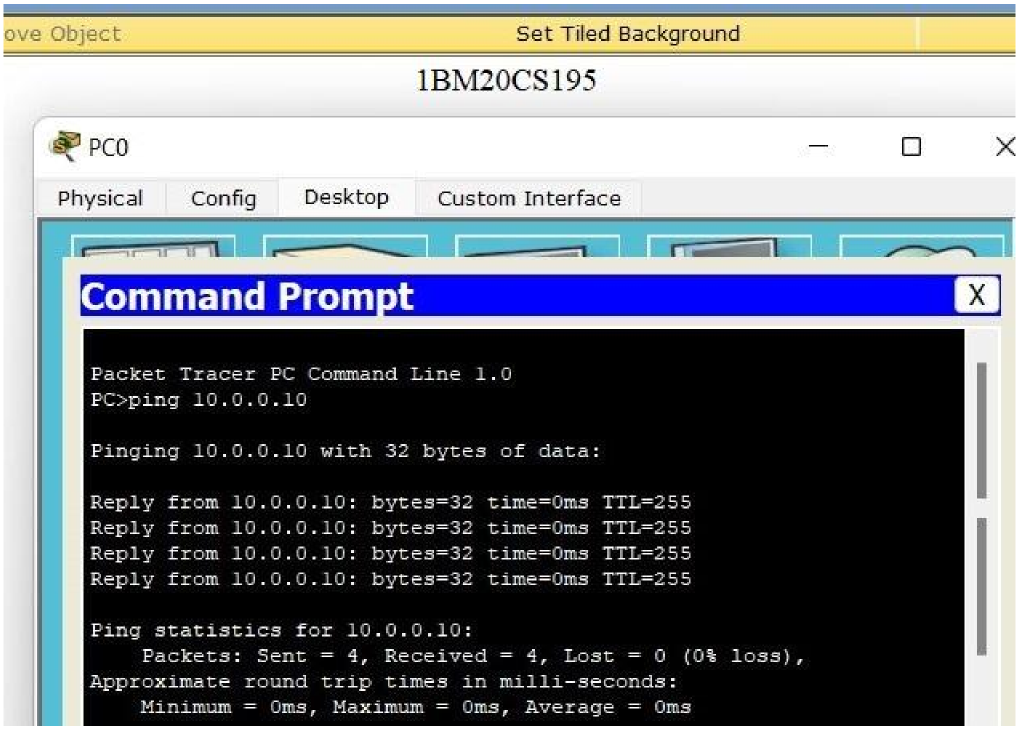
```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

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.
Reply from 10.0.0.10: Destination host unreachable.
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>
```



```
PC>
PC>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Reply from 20.0.0.10: bytes=32 time=1ms TTL=255
Reply from 20.0.0.10: bytes=32 time=0ms TTL=255
Reply from 20.0.0.10: bytes=32 time=0ms TTL=255
Reply from 20.0.0.10: bytes=32 time=0ms TTL=255

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

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 TTL=254
Reply from 30.0.0.10: bytes=32 time=1ms TTL=254
Reply from 30.0.0.10: bytes=32 time=1ms TTL=254
Reply from 30.0.0.10: bytes=32 time=8ms TTL=254

Ping statistics for 30.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 8ms, Average = 2ms

PC>
```



```
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
Reply from 40.0.0.1: bytes=32 time=13ms 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 = 8ms, Maximum = 13ms, Average = 10ms

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),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 24ms, Average = 11ms

PC>
```

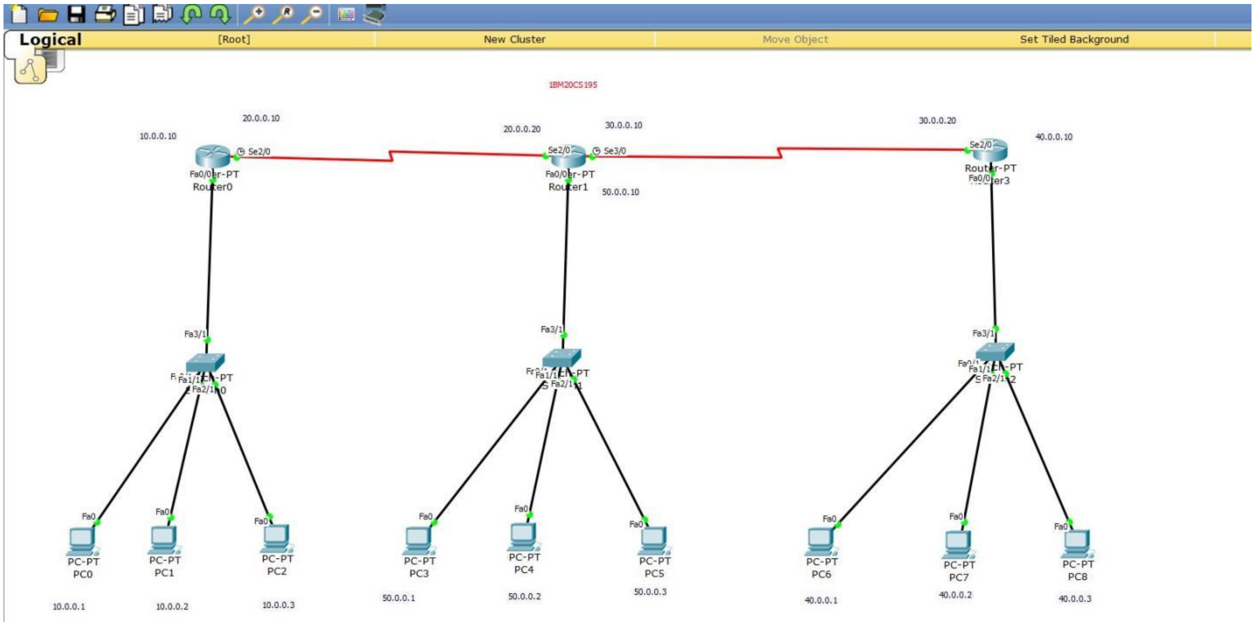


```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

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

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

EXPERIMENT - 3

Aim : Configuring default route to the Router

Topology :



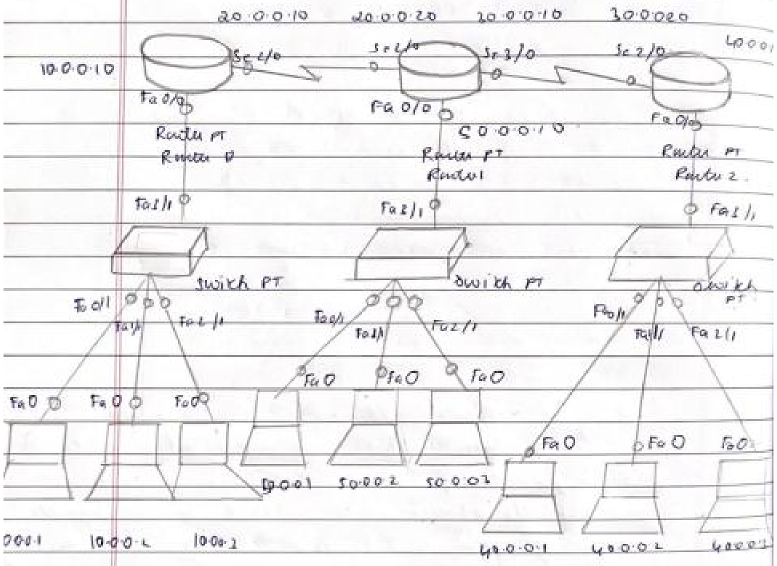
Procedure :

Experiment 3

28/11/22 Configuring default route to the router via switches.

Aim: To configure default route to a router via switches using minimum commands.

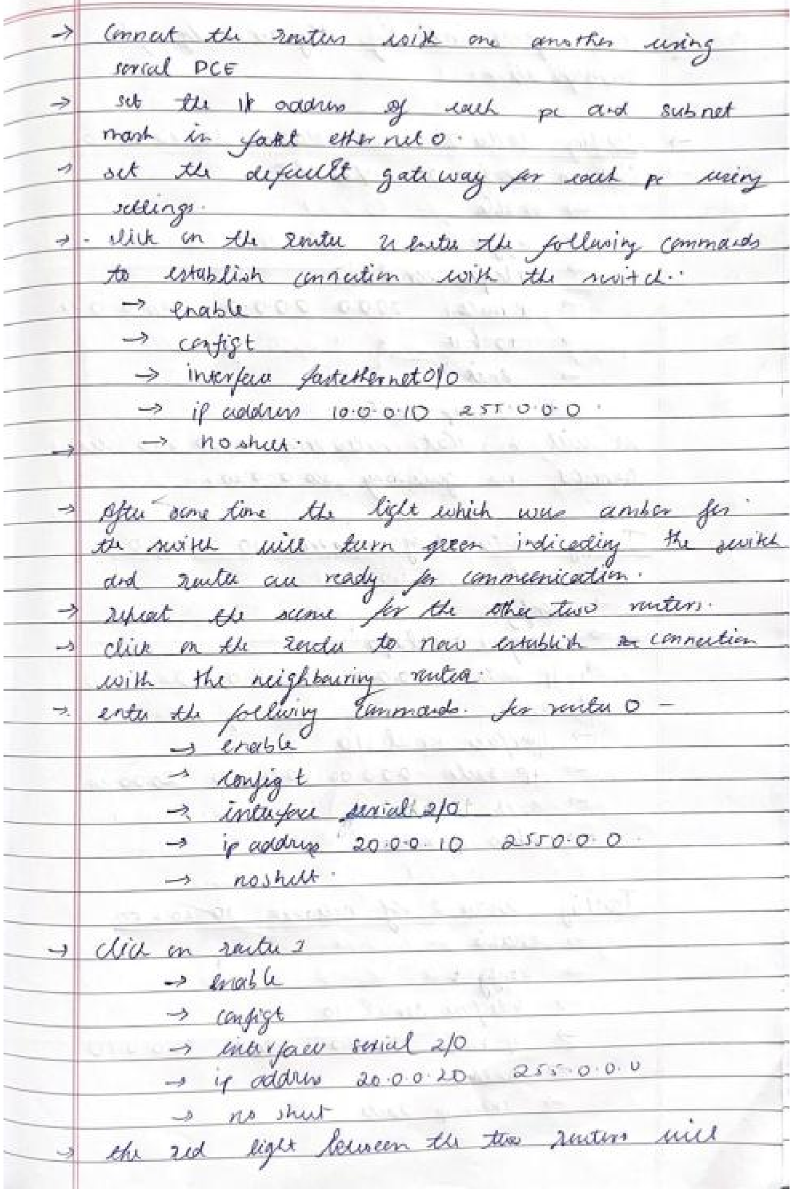
Topology:



procedure: → place 3 generic routers, 3 generic switches & 9 generic PCs in the work space.

→ connect the PCs to the switches using copper straight through

→ connect the switches to routers also using copper straight through



→ Connect the routers with one another using serial DCE

→ set the ip address of each pc and subnet mask in fast ethernet 0.

→ set the default gateway for each pc using settings.

→ click on the router & enter the following commands to establish connection with the switch.

```
→ enable
→ configt
→ interface fastethernet0/0
→ ip address 10·0·0·10 255·0·0·0
→ noshut.
```

→ After some time the light which was amber for the switch will turn green indicating the switch and router are ready for communication.

→ repeat the same for the other two routers.

→ click on the router to now establish ~~a~~ connection with the neighbouring router.

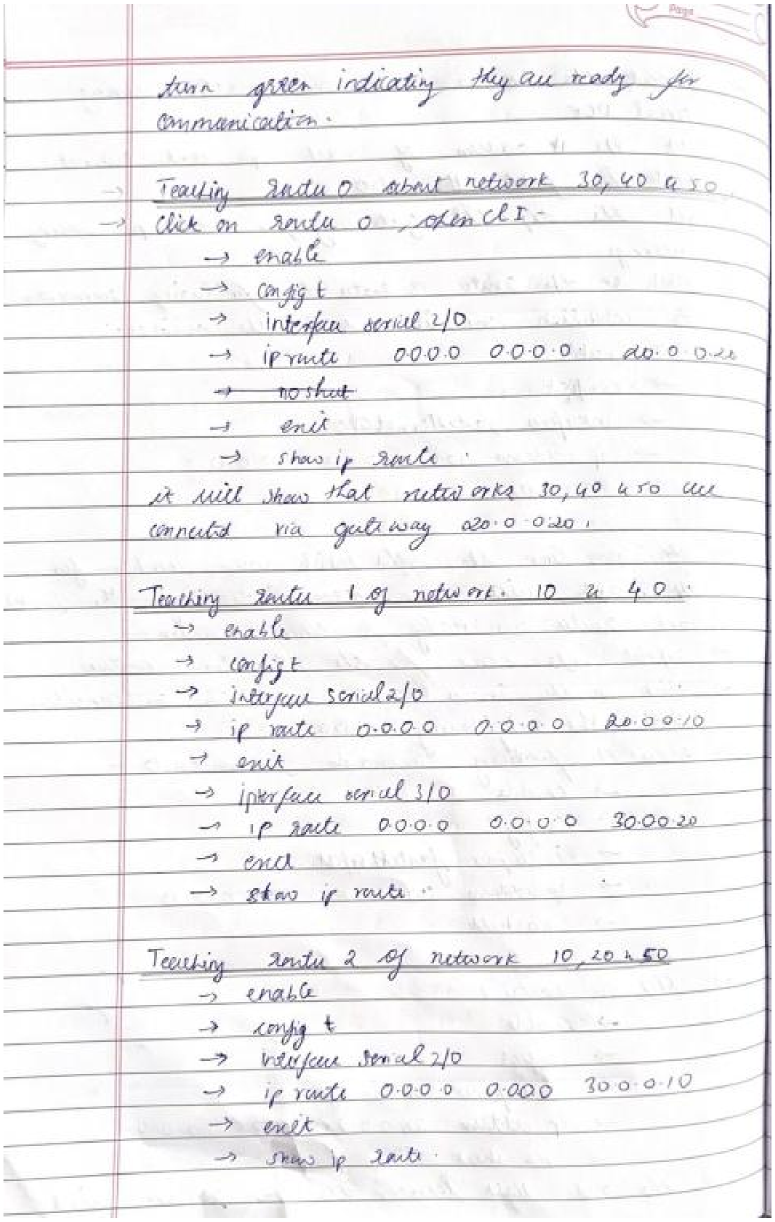
→ enter the following commands for router 0 –

```
→ enable
→ config t
→ interface serial 2/0
→ ip address 20·0·0·10 255·0·0·0
→ noshut.
```

→ click on router 1

```
→ enable
→ configt
→ interface serial 2/0
→ ip address 20·0·0·20 255·0·0·0
→ no shut
```

→ the red light between the two routers will



turn green indicating they are ready for communication.

→ Teaching router 0 about network 30, 40 & 50

→ Click on router 0, open CLI.

→ enable
→ config t
→ interface serial 2/0
→ ip route 0.0.0.0 0.0.0.0 20.0.0.20
~~→ no shut~~
→ exit
→ show ip route.

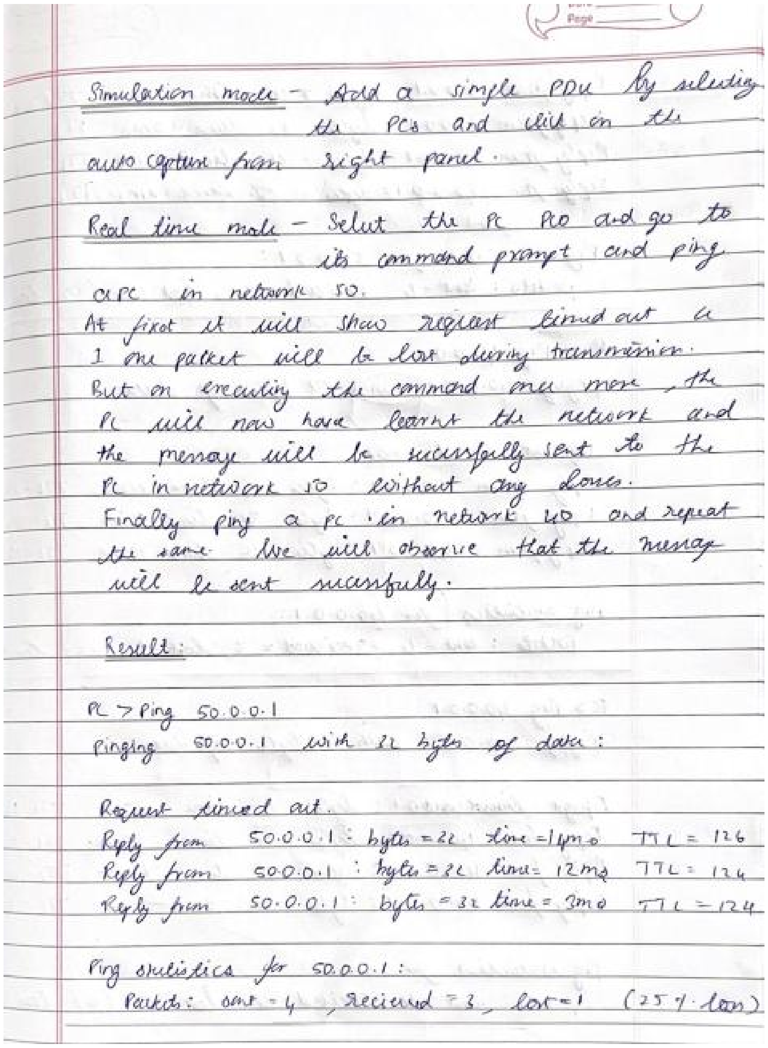
it will show that networks 30, 40 & 50 are connected via gateway 20.0.0.20.

Teaching router 1 of network. 10 & 40.

→ enable
→ config t
→ interface serial 2/0
→ ip route 0.0.0.0 0.0.0.0 20.0.0.10
→ exit
→ interface serial 3/0
→ ip route 0.0.0.0 0.0.0.0 30.0.0.20
→ exit
→ show ip route.

Teaching router 2 of network 10, 20 & 50

→ enable
→ config t
→ interface serial 2/0
→ ip route 0.0.0.0 0.0.0.0 30.0.0.10
→ exit
→ show ip route.



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

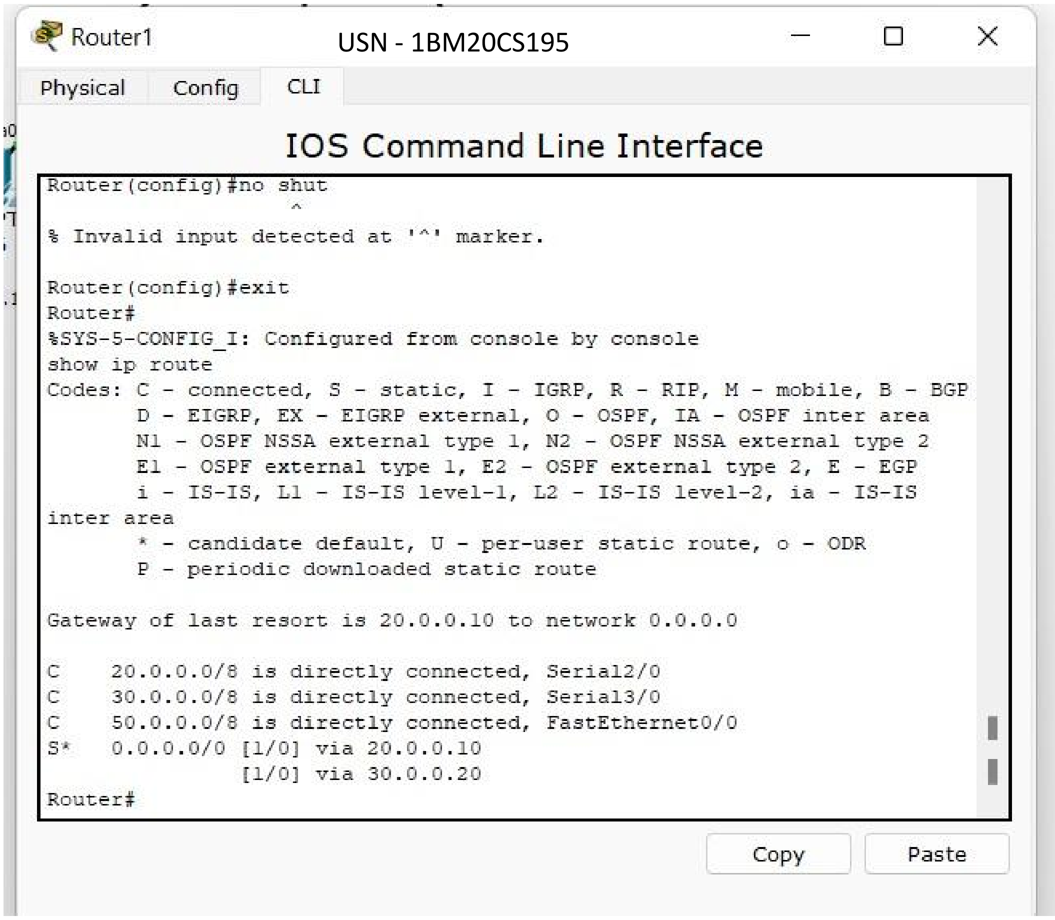
Real time mode - Select the PC PC0 and go to its command prompt and ping a PC in network 50.

At first it will show request timed out & 1 one packet will be lost during transmission. But on executing the command once more, the PC will now have learnt the network and the message will be successfully sent to the PC in network 50 without any losses. Finally ping a PC in network 40 and repeat the same. We will observe that the message will be sent successfully.

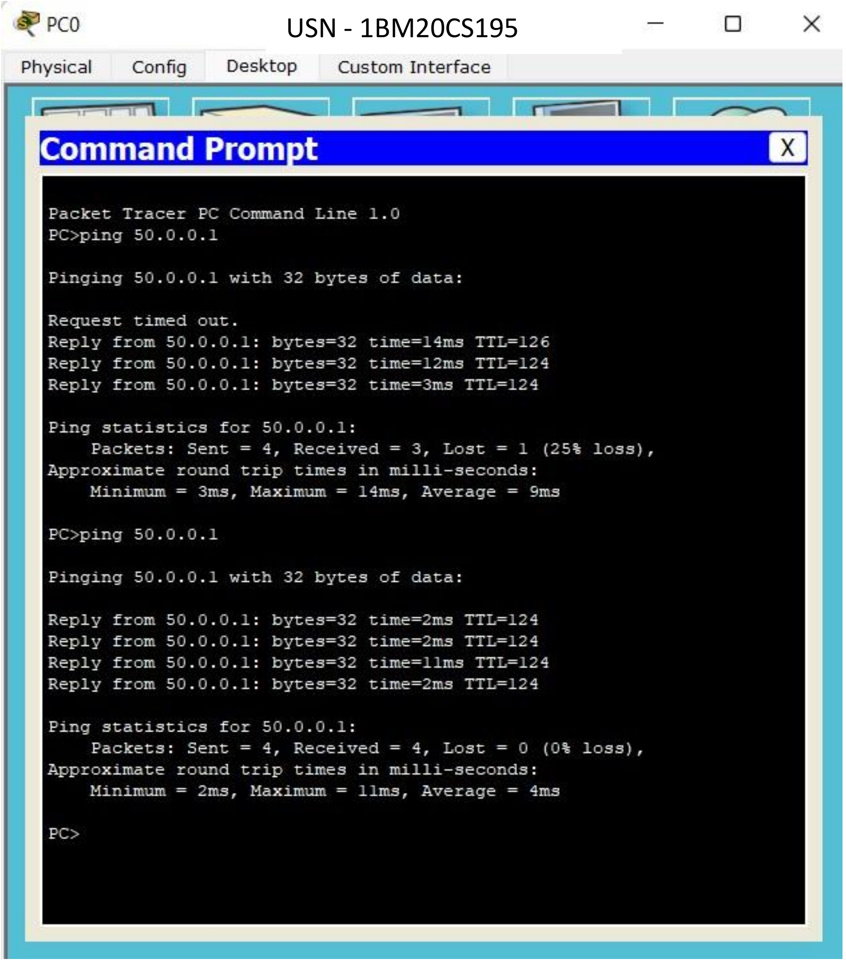
Result:

```
PC > Ping 50.0.0.1
Pinging 50.0.0.1 with 32 bytes of data:

Request timed out.
Reply from   50.0.0.1 : bytes = 32  time = 14ms   TTL = 126
Reply from   50.0.0.1 : bytes = 32  time = 12ms   TTL = 124
Reply from   50.0.0.1 : bytes = 32  time = 3ms    TTL = 124

Ping statistics for 50.0.0.1:
  Packets: sent = 4 , Recieved = 3, lost = 1   (25% loss)
```

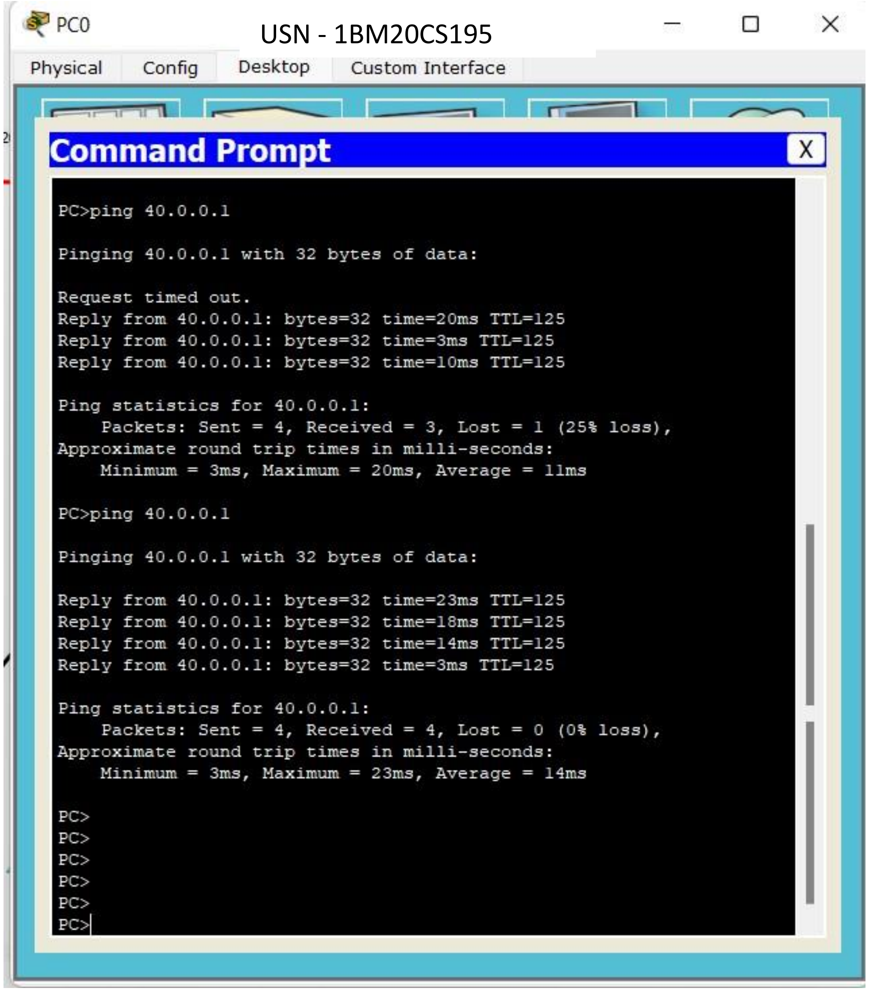
Snapshot of Output :



```
Router(config)#no shut
                   ^
% Invalid input detected at '^' marker.

Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
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 20.0.0.10 to network 0.0.0.0

C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
C    50.0.0.0/8 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 20.0.0.10
               [1/0] via 30.0.0.20
Router#
```



```
Packet Tracer PC Command Line 1.0
PC>ping 50.0.0.1

Pinging 50.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 50.0.0.1: bytes=32 time=14ms TTL=126
Reply from 50.0.0.1: bytes=32 time=12ms TTL=124
Reply from 50.0.0.1: bytes=32 time=3ms TTL=124

Ping statistics for 50.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 14ms, Average = 9ms

PC>ping 50.0.0.1

Pinging 50.0.0.1 with 32 bytes of data:

Reply from 50.0.0.1: bytes=32 time=2ms TTL=124
Reply from 50.0.0.1: bytes=32 time=2ms TTL=124
Reply from 50.0.0.1: bytes=32 time=11ms TTL=124
Reply from 50.0.0.1: bytes=32 time=2ms TTL=124

Ping statistics for 50.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 11ms, Average = 4ms

PC>
```

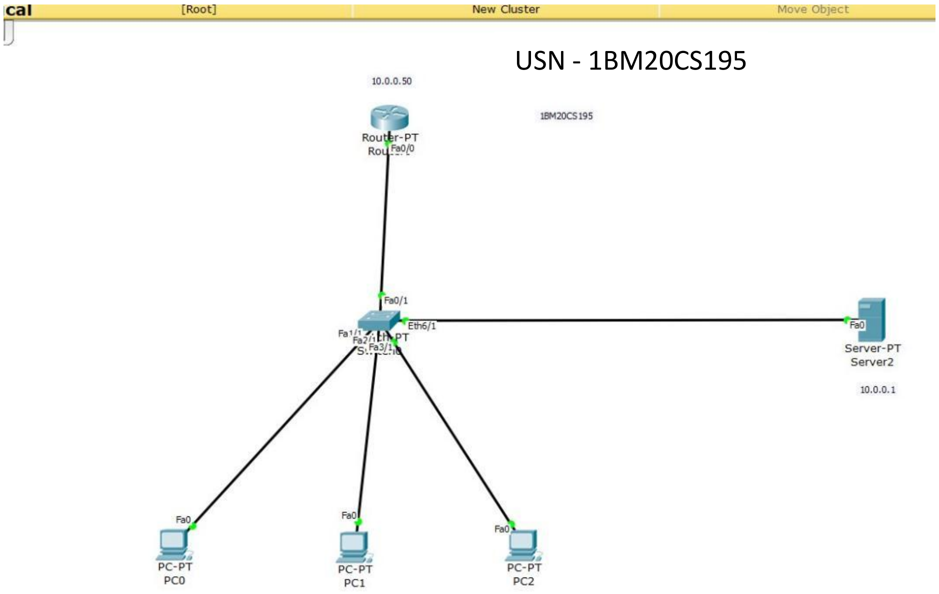


```
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=20ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms 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 = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 20ms, Average = 11ms

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=23ms TTL=125
Reply from 40.0.0.1: bytes=32 time=18ms TTL=125
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms 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 = 3ms, Maximum = 23ms, Average = 14ms

PC>
PC>
PC>
PC>
PC>
PC>
```

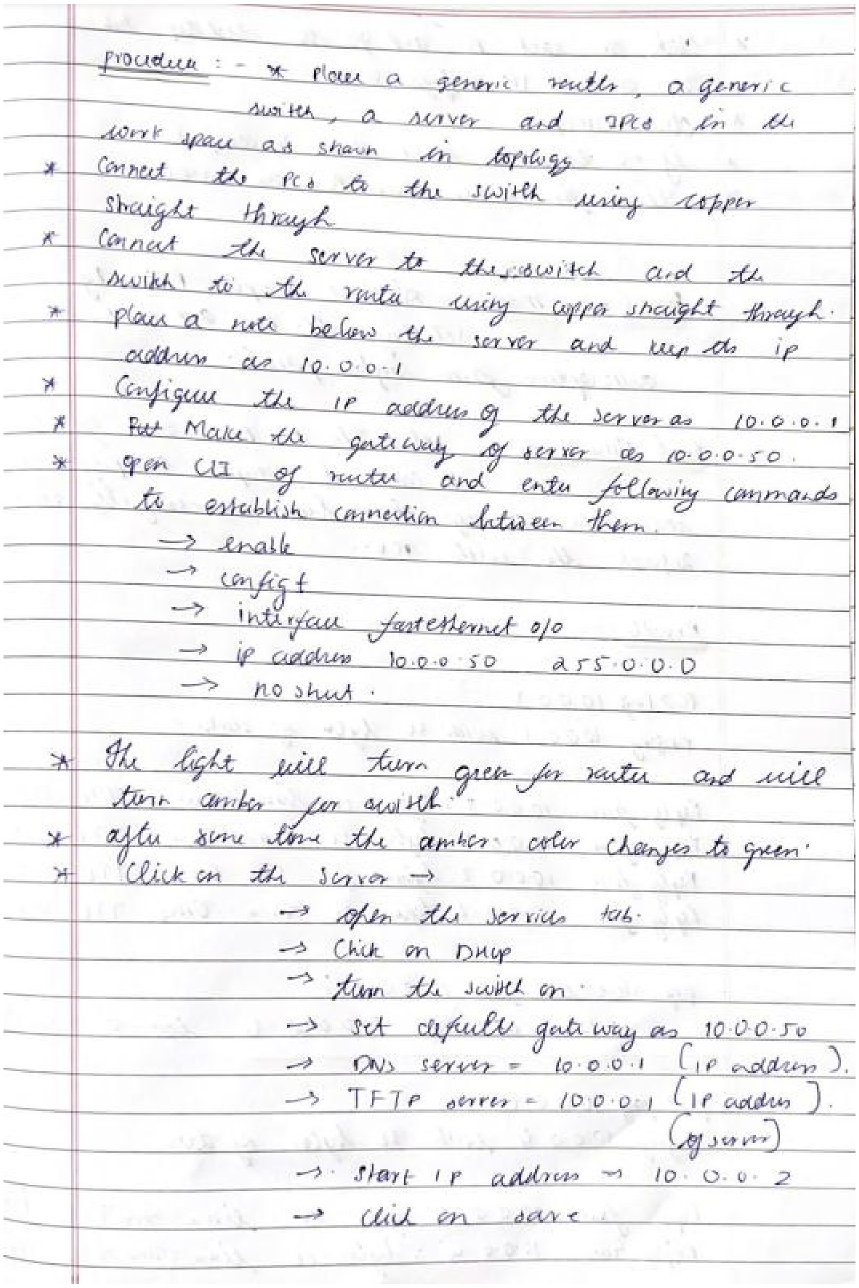
EXPERIMENT - 4

Aim : Configuring DHCP within a LAN in a packet Tracer

Topology :



Procedure :



Procedure :- * Place a generic router, a generic switch, a server and 2PCs in the work space as shown in topology

* Connect the PCs to the switch using copper straight through

* Connect the server to the switch and the switch to the router using copper straight through.

* place a note below the server and keep its ip address as 10.0.0.1

* Configure the IP address of the server as 10.0.0.1

* Put Make the gateway of server as 10.0.0.50.

* open CLI of router and enter following commands to establish connection between them.

→ enable

→ config t

→ interface fastethernet 0/0

→ ip address 10.0.0.50 255.0.0.0

→ no shut.

* The light will turn green for router and will turn amber for switch.

* after some time the amber color changes to green.

* Click on the server →

→ open the services tab.

→ Click on DHCP

→ turn the switch on.

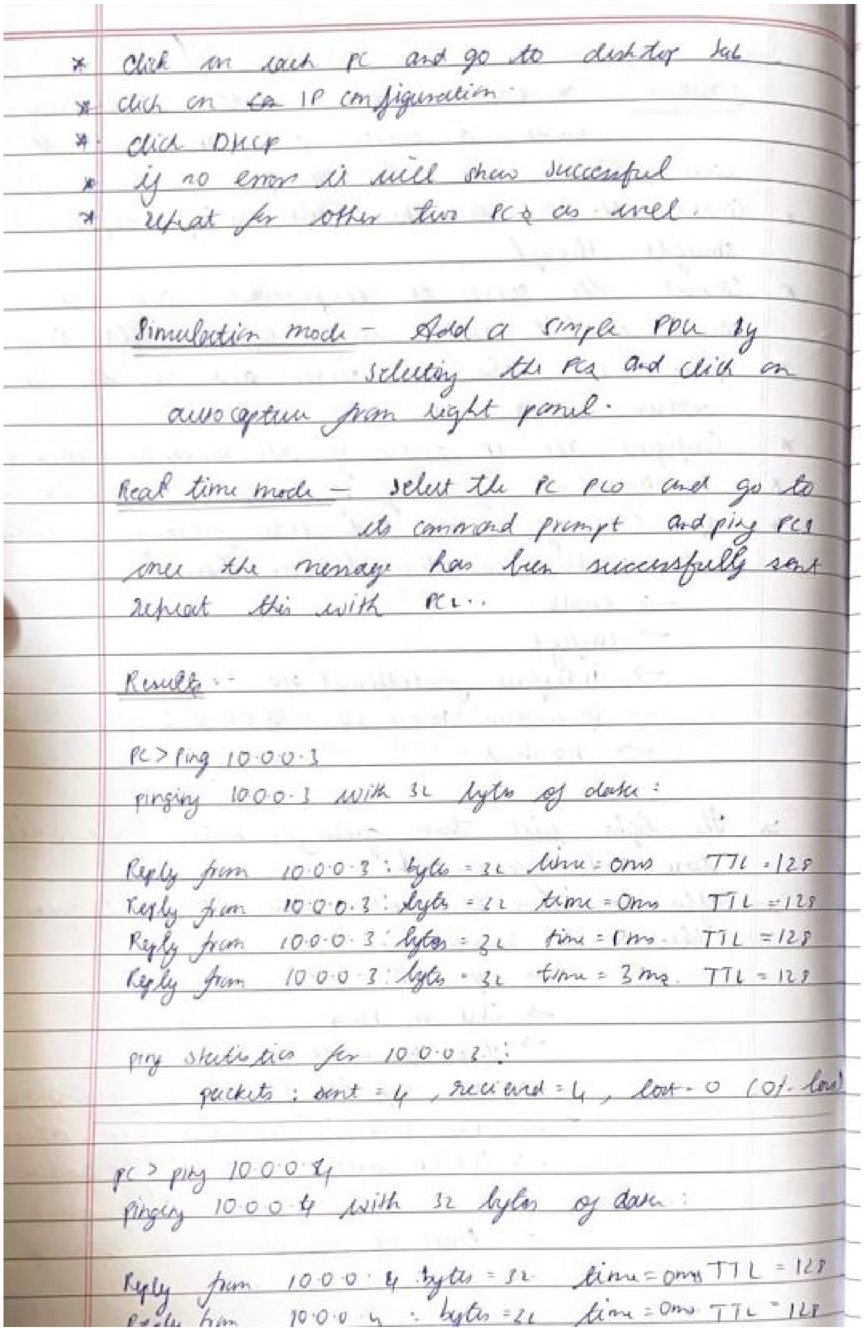
→ set default gateway as 10.0.0.50

→ DNS server = 10.0.0.1 (ip address).

→ TFTP server = 10.0.0.1 (ip address). (of server)

→ start ip address → 10.0.0.2

→ click on save



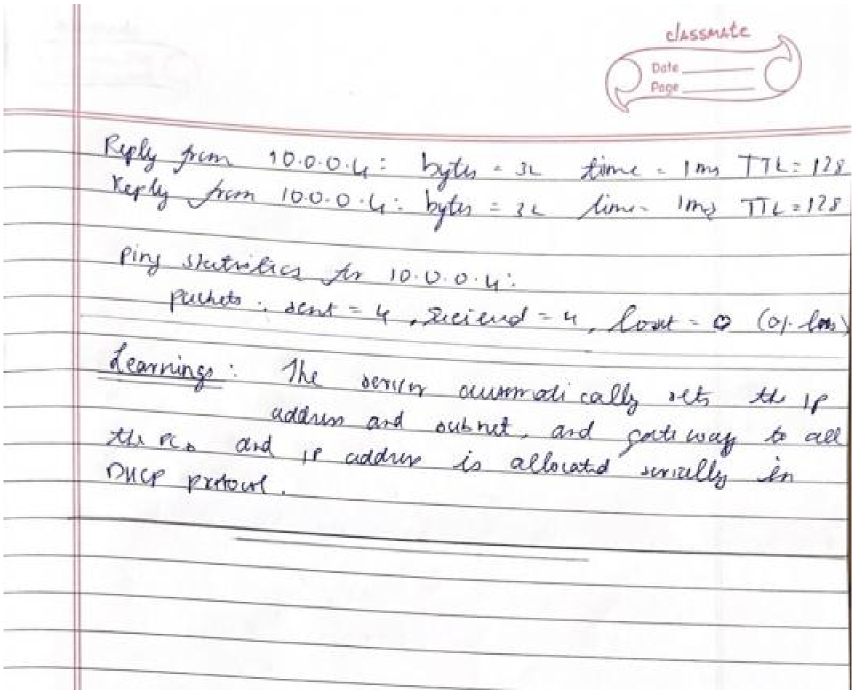
* Click on each PC and go to desktop tab
* click on ~~ca~~ IP configuration.
* click DHCP
* if no error it will show successful
* repeat for other two PCs as well.

<u>Simulation mode</u> – Add a simple PDU by selecting the PCs and click on auto capture from right panel.

<u>Real time mode</u> – select the PC PC0 and go to its command prompt and ping PC1 once the message has been successfully sent repeat this with PC2..

<u>Results</u> :-

```
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 = 0ms   TTL = 128
Reply from 10.0.0.3 : bytes = 32  time = 0ms   TTL = 128
Reply from 10.0.0.3 : bytes = 32  time = 1ms.  TTL = 128
Reply from 10.0.0.3 : bytes = 32  time = 3ms.  TTL = 128

ping statistics for 10.0.0.3 :
     packets : sent = 4 , recieved = 4 , lost = 0  (0% loss)

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
Reply from 10.0.0.4 : bytes = 32  time = 0ms. TTL = 128
```

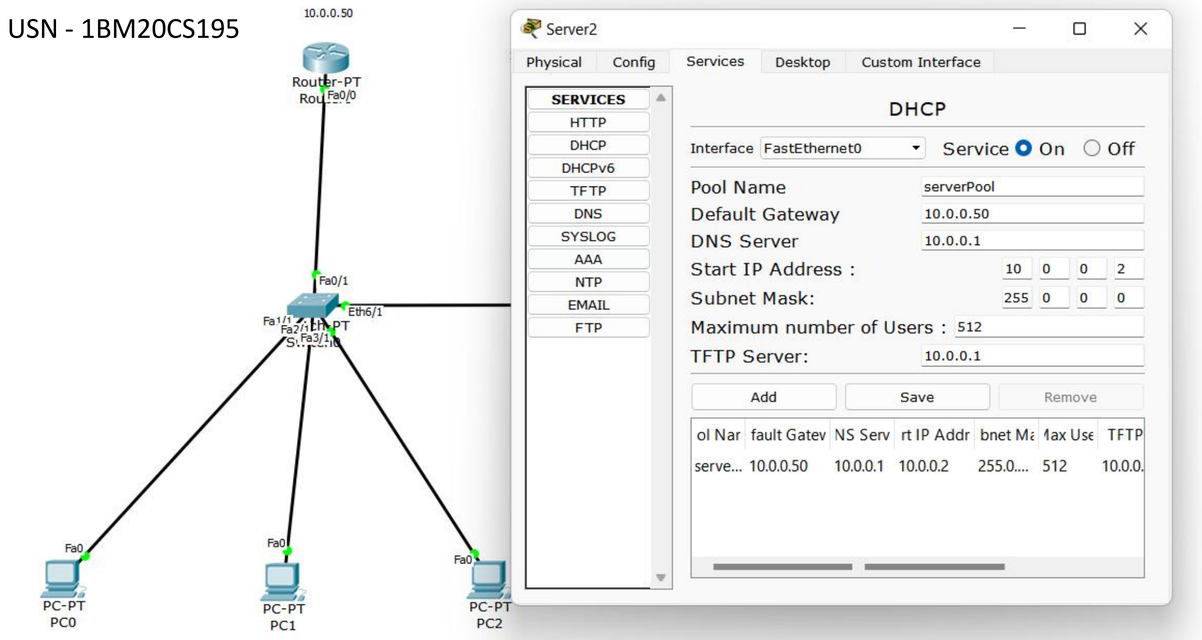


Reply from 10.0.0.4: bytes = 32 time = 1ms TTL=128
Reply from 10.0.0.4: bytes = 32 time = 1ms TTL=128

Ping statistics for 10.0.0.4:
packets: sent = 4, received = 4, lost = 0 (0% loss)

Learnings: The server automatically sets the ip address and subnet and gateway to all the PCs and ip address is allocated serially in DHCP protocol.

Snapshot of Output :



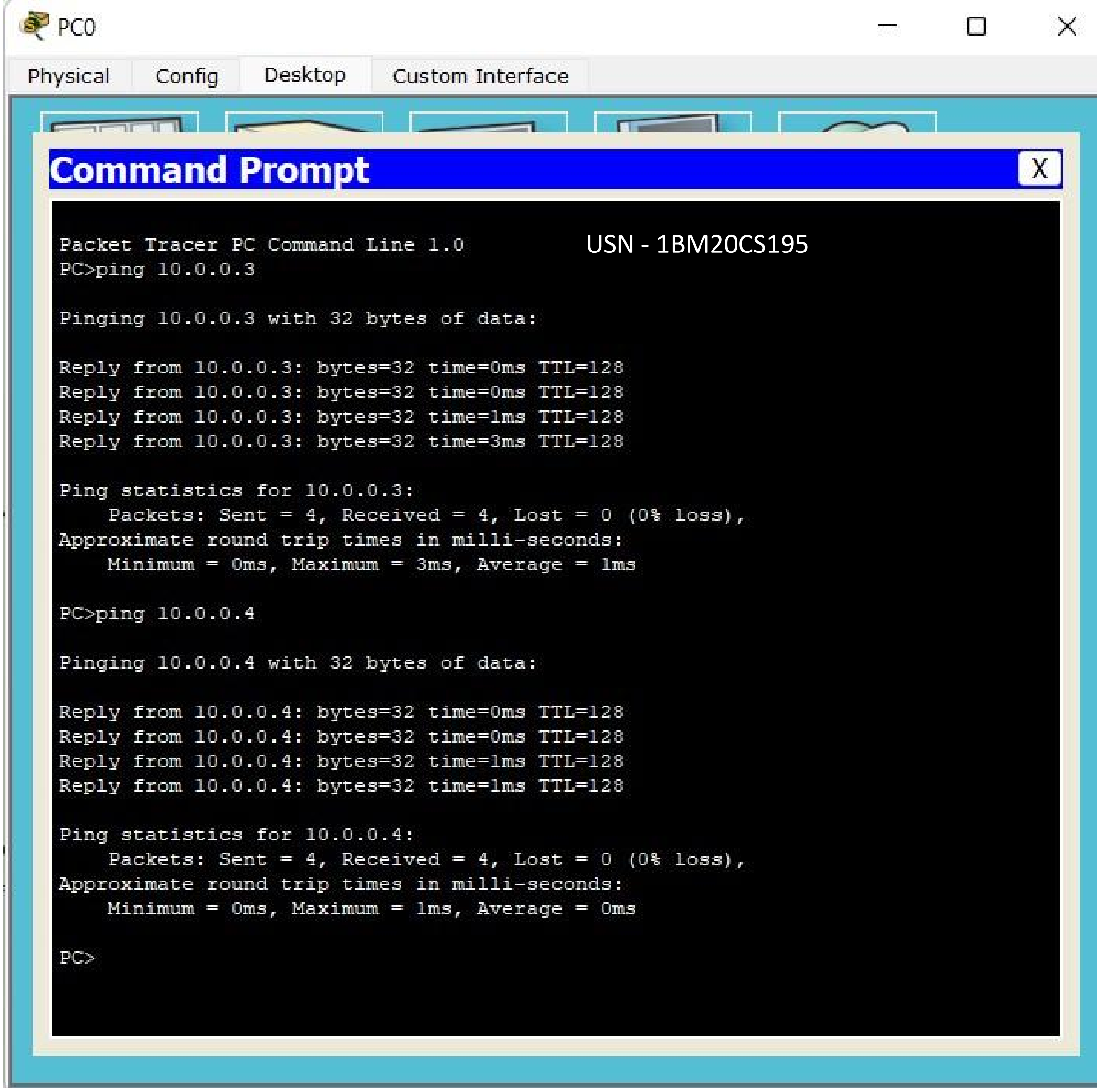
USN - 1BM20CS195

DHCP

Interface FastEthernet0 Service ● On ○ Off

Pool Name serverPool
Default Gateway 10.0.0.50
DNS Server 10.0.0.1
Start IP Address : 10 0 0 2
Subnet Mask: 255 0 0 0
Maximum number of Users : 512
TFTP Server: 10.0.0.1

ol Nar	fault Gatev	NS Serv	rt IP Addr	bnet Ma	1ax Use	TFTP
serve...	10.0.0.50	10.0.0.1	10.0.0.2	255.0....	512	10.0.0.



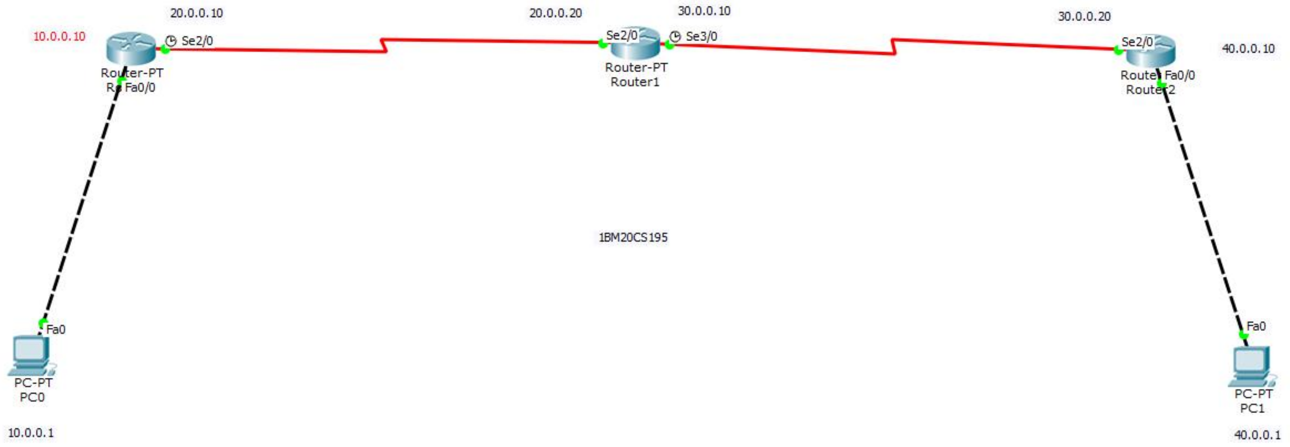
USN - 1BM20CS195

```
Packet Tracer PC Command Line 1.0
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=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 1ms

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
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

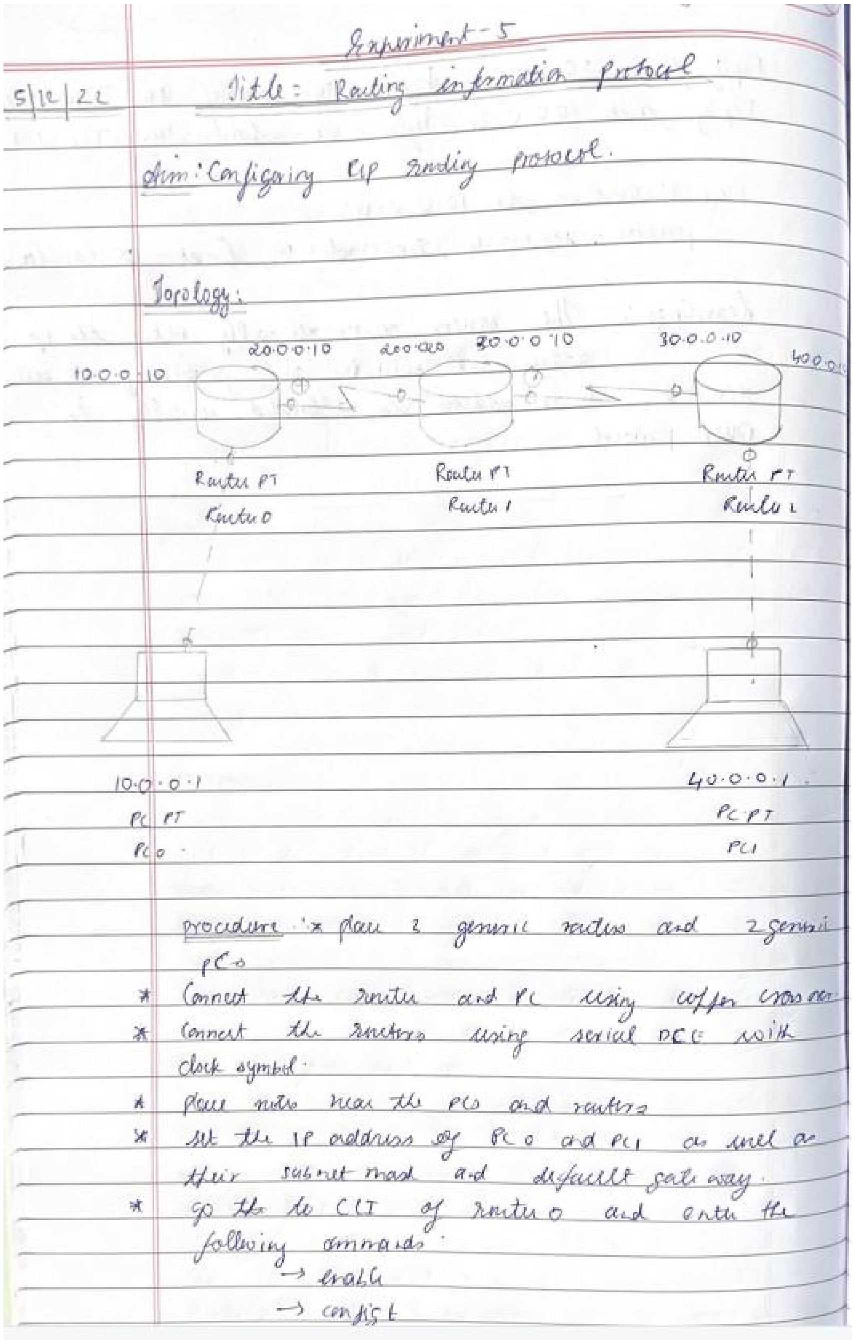
EXPERIMENT - 5

Aim : Configuring RIP Routing Protocol in Routers

Topology :



Procedure :



Experiment - 5

5/12/22 Title: Routing information protocol

Aim: Configuring RIP routing protocol.

Topology:

10.0.0.10
20.0.0.10
20.0.0.20
30.0.0.10
30.0.0.10
40.0.0.10
0
0
0
0
0
0
0
Router PT
Router 0
Router PT
Router 1
Router PT
Router 2
10.0.0.1
PC PT
PC 0
40.0.0.1
PC PT
PC1

procedure: * place 3 generic routers and 2 generic PCs

* Connect the router and PC using copper cross over

* Connect the routers using serial DCE with clock symbol.

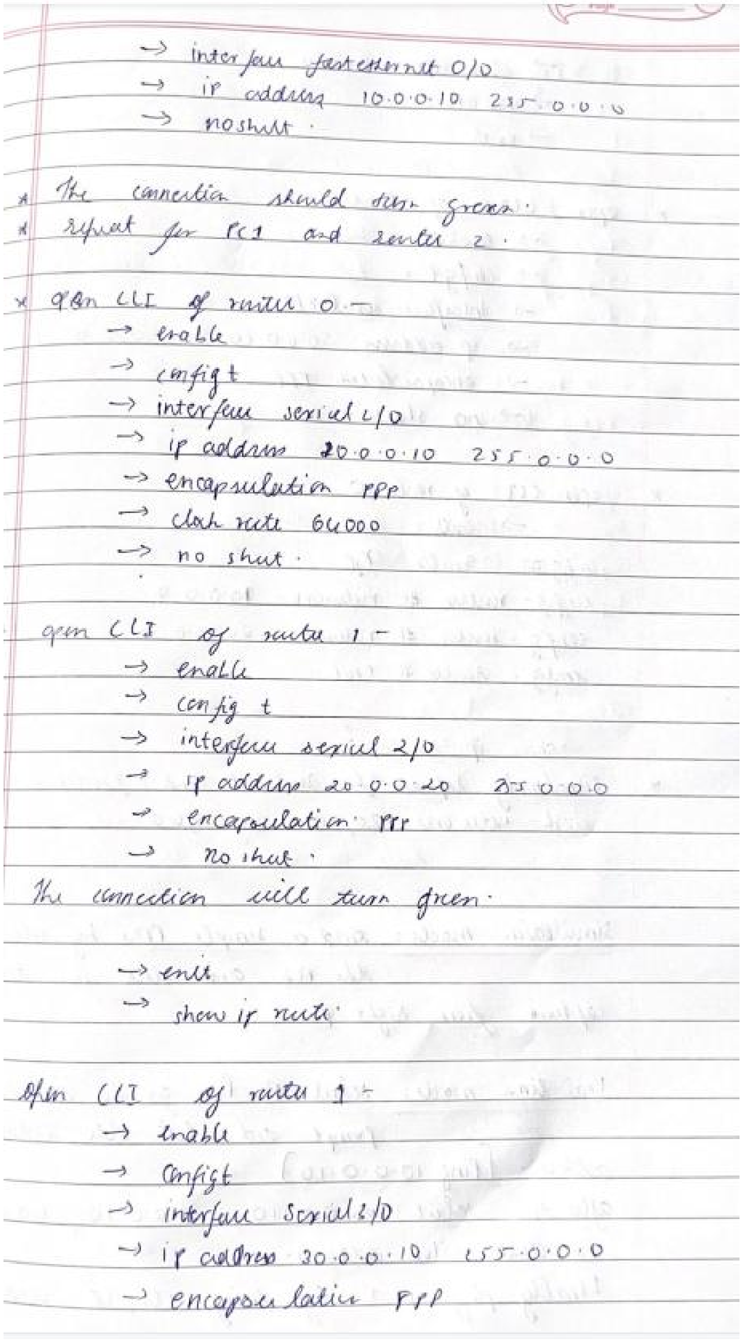
* place notes near the PCs and routers

* set the IP address of PC 0 and PC1 as well as their subnet mask and default gate way.

* go the to CLI of router 0 and enter the following commands:

→ enable

→ config t



→ interface fastethernet 0/0
→ ip address 10.0.0.10 255.0.0.0
→ noshut.

* The connection should turn green.
* repeat for PC1 and router 2.

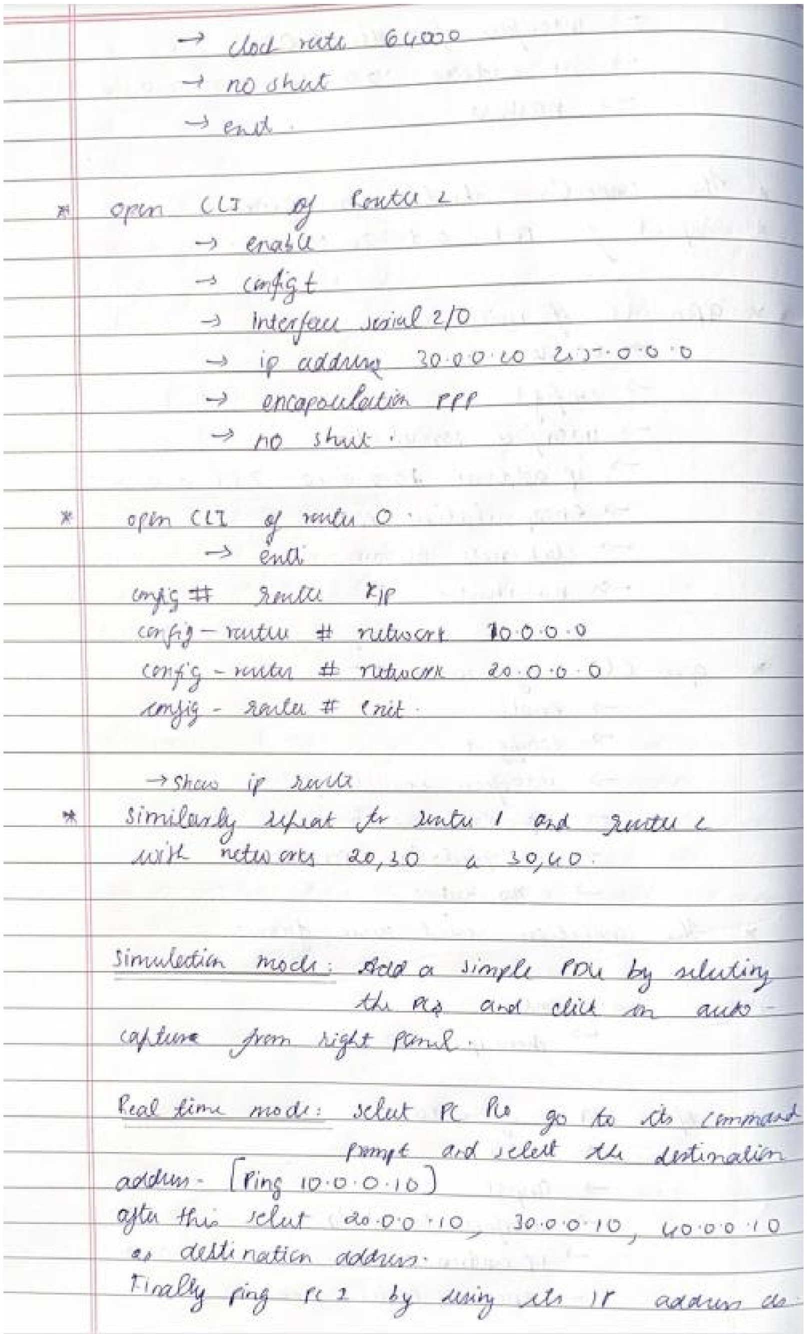
* Open CLI of router 0 -
→ enable
→ config t
→ interface serial 2/0
→ ip address 20.0.0.10 255.0.0.0
→ encapsulation ppp
→ clock rate 64000
→ no shut.

open CLI of router 1 -
→ enable
→ config t
→ interface serial 2/0
→ ip address 20.0.0.20 255.0.0.0
→ encapsulation ppp
→ no shut.

The connection will turn green.

→ exit
→ show ip route

Open CLI of router 1 -
→ enable
→ config t
→ interface serial 2/0
→ ip address 30.0.0.10 255.0.0.0
→ encapsulation ppp



→ clock rate 64000
→ no shut
→ end.

* open CLI of Router 2
→ enable
→ config t
→ interface serial 2/0
→ ip address 30.0.0.20 255.0.0.0
→ encapsulation PPP
→ no shut.

* open CLI of router 0.
→ end
config # router rip
config-router # network 10.0.0.0
config-router # network 20.0.0.0
config-router # exit.

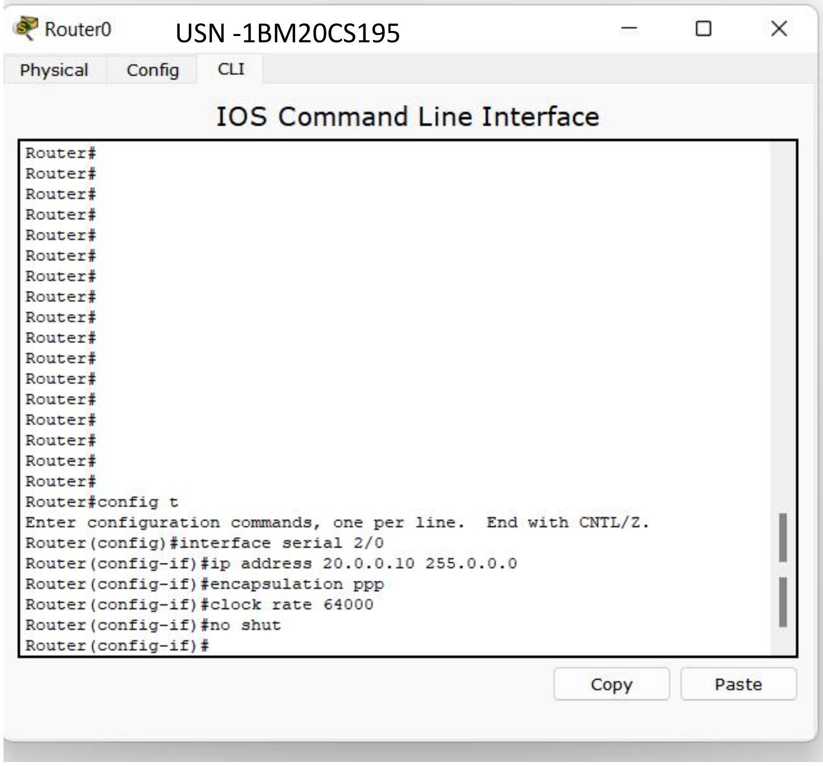
→ show ip route

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

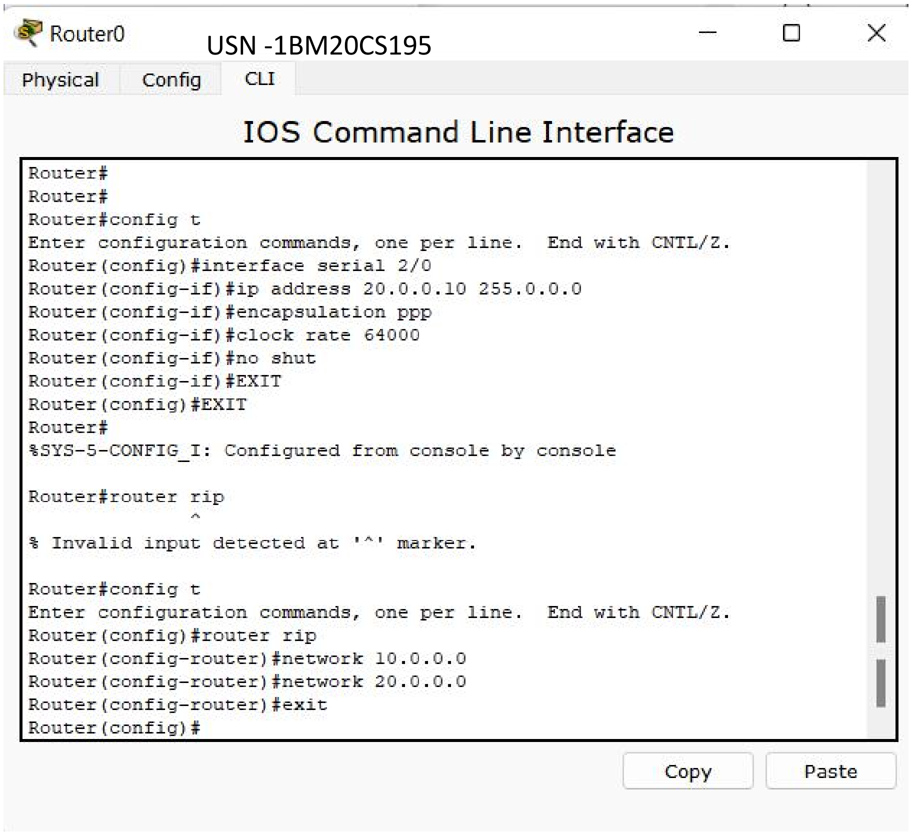
Simulation mode: Add a simple PDU by selecting the PCs and click on auto capture from right panel.

Real time mode: select PC P0 go to its command prompt and select the destination address. [Ping 10.0.0.10)
after this select 20.0.0.10, 30.0.0.10, 40.0.0.10 as destination address.
Finally ping PC 2 by using its IP address as

Snapshot of Output :

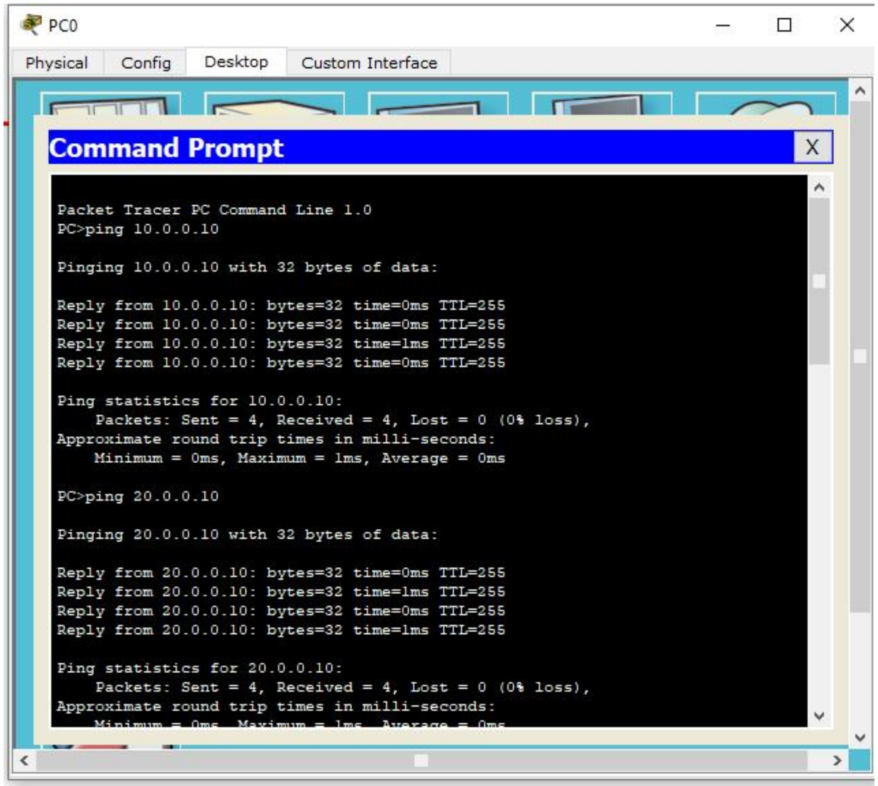


```
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 2/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shut
Router(config-if)#
```

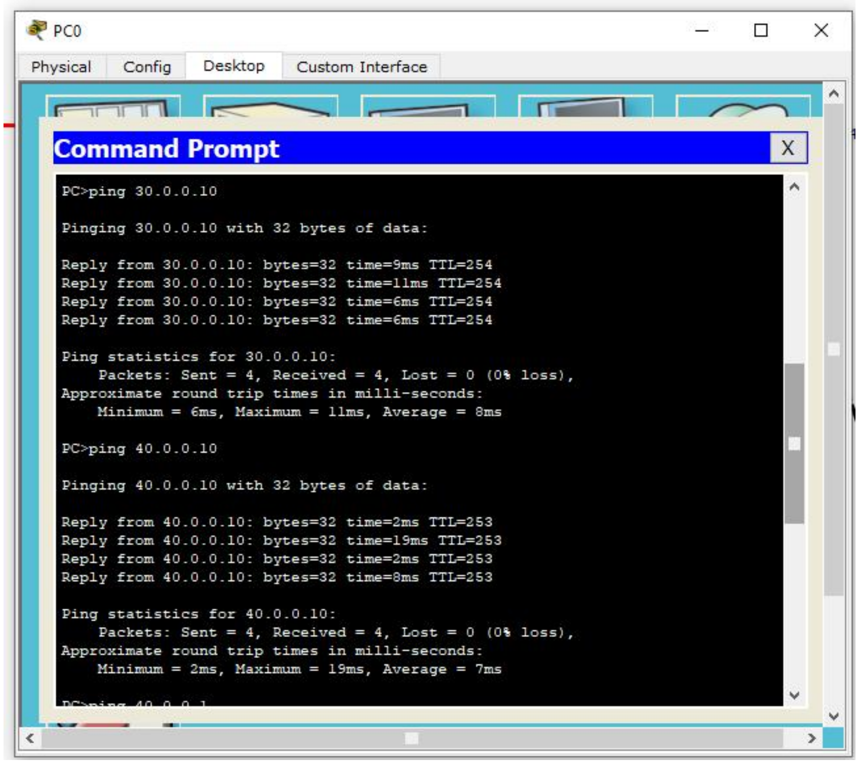


USN -1BM20CS195

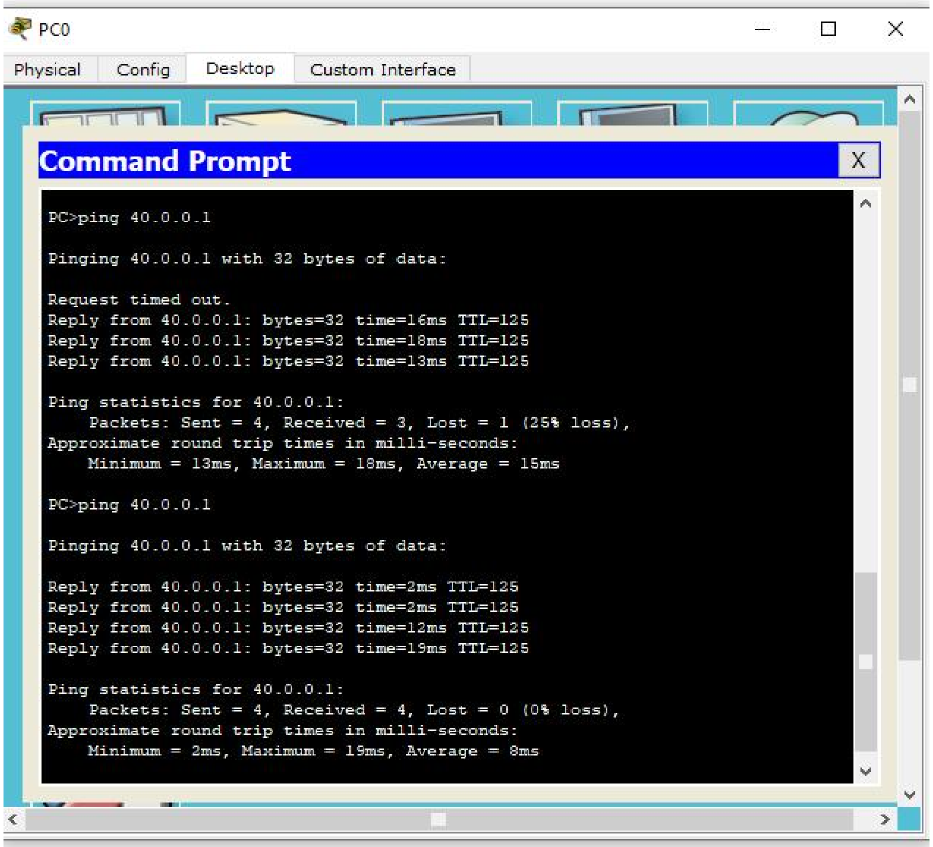
```
Router#
Router#
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 2/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shut
Router(config-if)#EXIT
Router(config)#EXIT
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#router rip
              ^
% Invalid input detected at '^' marker.

Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
```



```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
Reply from 10.0.0.10: bytes=32 time=1ms TTL=255
Reply from 10.0.0.10: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Reply from 20.0.0.10: bytes=32 time=0ms TTL=255
Reply from 20.0.0.10: bytes=32 time=1ms TTL=255
Reply from 20.0.0.10: bytes=32 time=0ms TTL=255
Reply from 20.0.0.10: bytes=32 time=1ms TTL=255

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```



```
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=9ms TTL=254
Reply from 30.0.0.10: bytes=32 time=11ms TTL=254
Reply from 30.0.0.10: bytes=32 time=6ms TTL=254
Reply from 30.0.0.10: bytes=32 time=6ms TTL=254

Ping statistics for 30.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 11ms, Average = 8ms

PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=2ms TTL=253
Reply from 40.0.0.10: bytes=32 time=19ms TTL=253
Reply from 40.0.0.10: bytes=32 time=2ms TTL=253
Reply from 40.0.0.10: bytes=32 time=8ms TTL=253

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 19ms, Average = 7ms

PC>ping 40.0.0.1
```

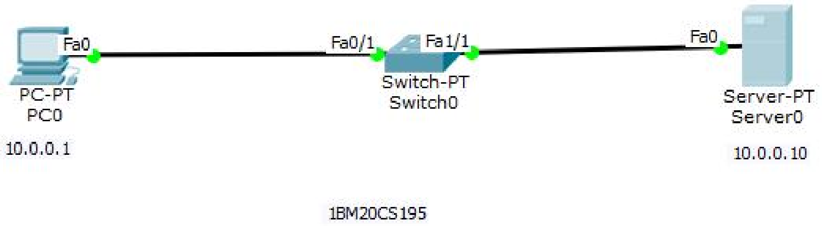


```
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=18ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms 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 = 13ms, Maximum = 18ms, Average = 15ms

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=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=12ms TTL=125
Reply from 40.0.0.1: bytes=32 time=19ms 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 = 2ms, Maximum = 19ms, Average = 8ms
```

EXPERIMENT - 6

Aim : Demonstration of WEB server and DNS using Packet Tracer

Topology :



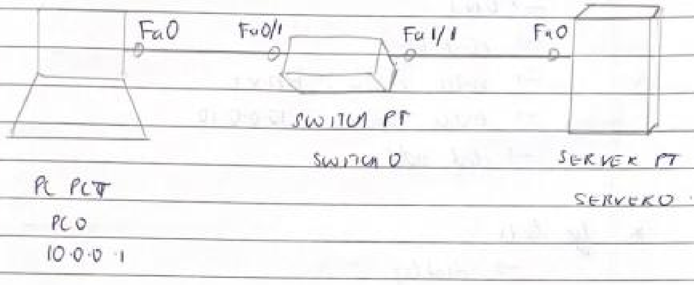
Procedure :

Experiment - 6

12/12/24 **Title: Domain Name service.**

Aim: Demonstration of webserver and DNS using Packet tracer.

Topology:



Procedure: * place a PC a switch and a server in the work space.

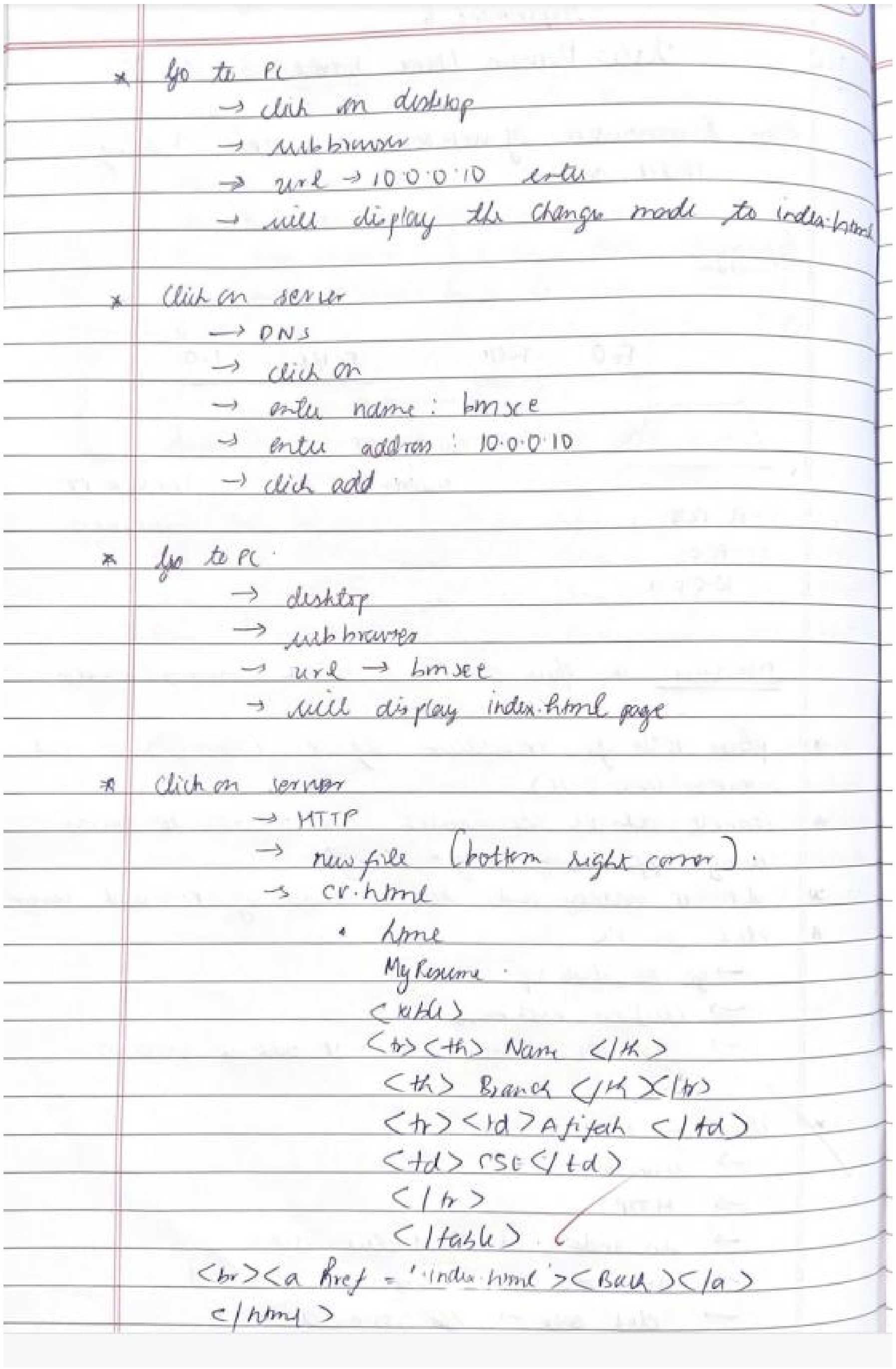
* place notes for IP address of PC (10.0.0.1) and server (10.0.0.10)

* Connect the PC to switch and switch to server using copper ~~Cro~~ straight through.

* Set IP address and subnet mask of PC and server

* click on PC
- → go to desk top
- → click on web browser
- → in URL enter the IP address 10.0.0.10

* click on server
- → services
- → HTTP
- → in index.html → click edit
- → make one two changes in the text
- → click save → yes over write



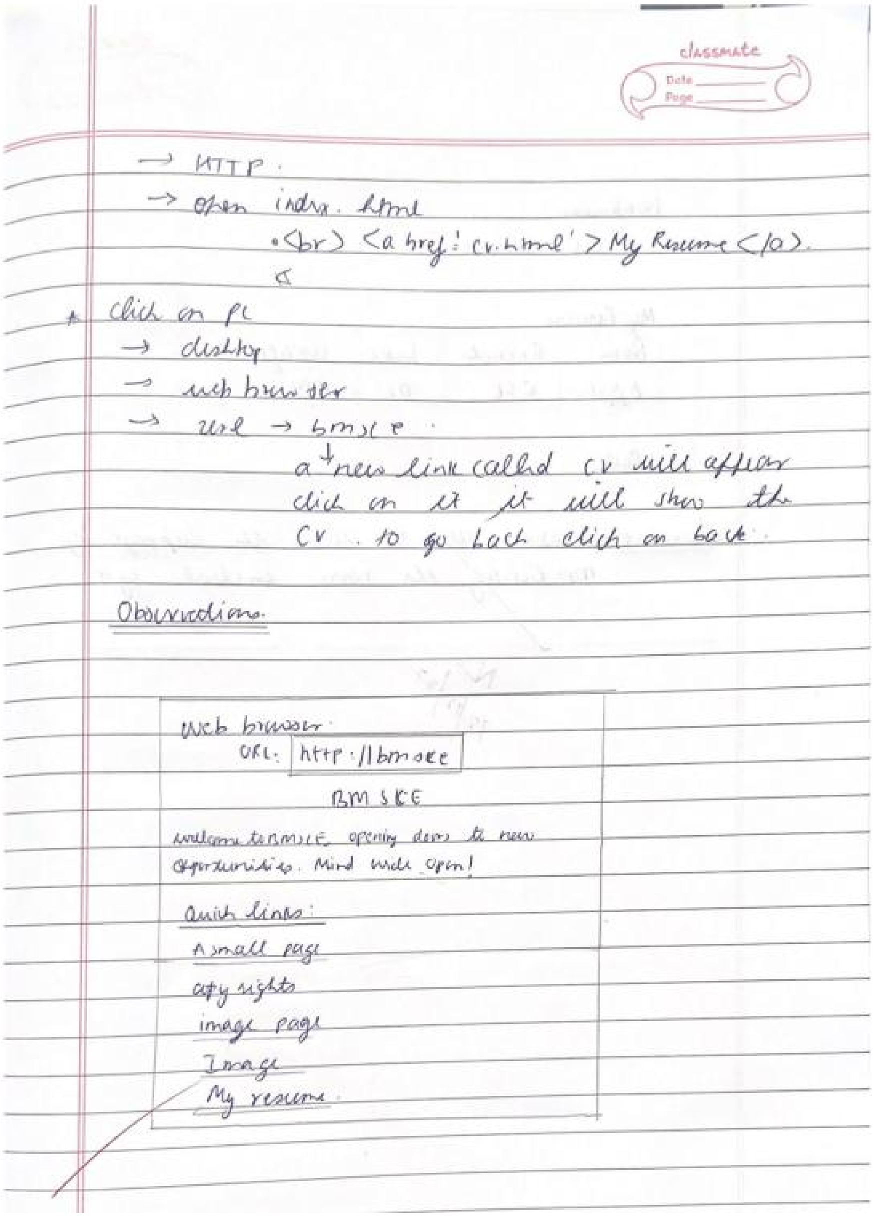
* Go to PC
→ click on desktop
→ webbrowser
→ url → 10.0.0.10 enter
→ will display the change made to index.html

* click on server
→ DNS
→ click on
→ enter name : bmsce
→ enter address : 10.0.0.10
→ click add

* Go to PC
→ desktop
→ webbrowser
→ url → bmsce
→ will display index.html page

* click on server
→ HTTP
→ new file (bottom right corner).
→ cv.html

```
• html
  MyResume
  <table>
  <tr><th> Name </th>
  <th> Branch </th></tr>
  <tr><td>Afifah </td>
  <td> CSE</td>
  </tr>
  </table>.
<br><a href = ''index.html'><Back></a>
</html>
```



→ HTTP.

→ open index. html

•
 <a href='cv.html'> My Resume </a>.

* click on pc

→ desktop

→ web browser

→ url → bmsce.

a new link called cv will appear click on it it will show the cv. to go back click on back.

Observations.

web browser

URL: http://bmsce

BMSCE

welcome to BMSCE, opening doors to new opportunities. Mind wide open!

Quick links:

A small page

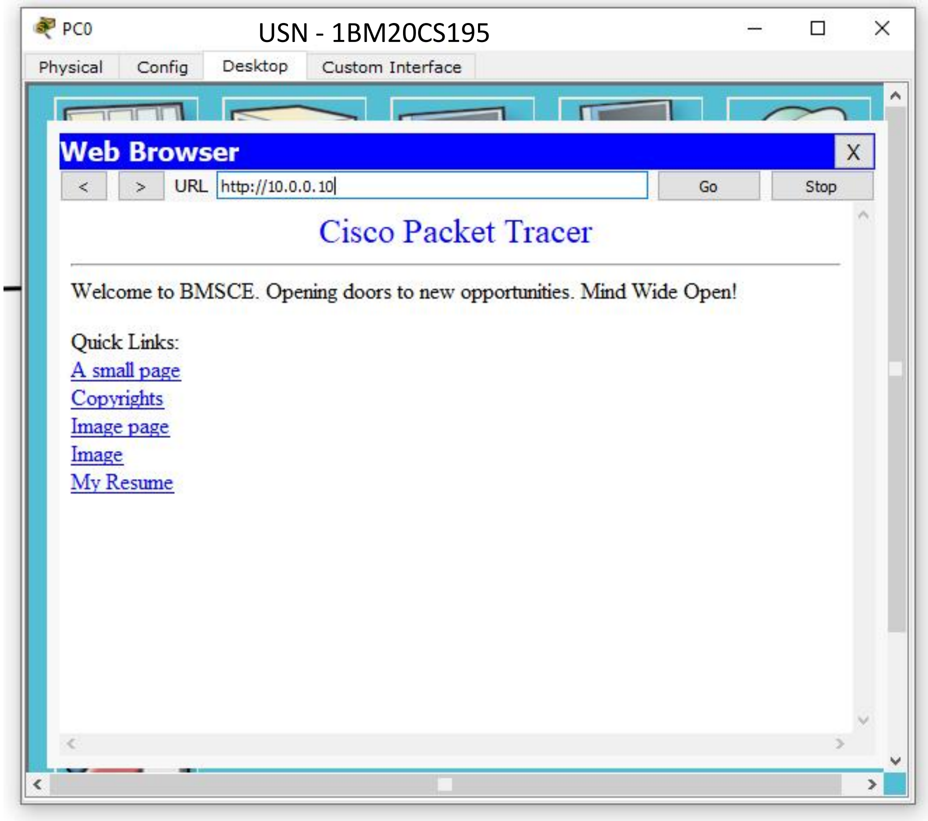
copy rights

image page

Image

My resume.

Snapshot of Output :



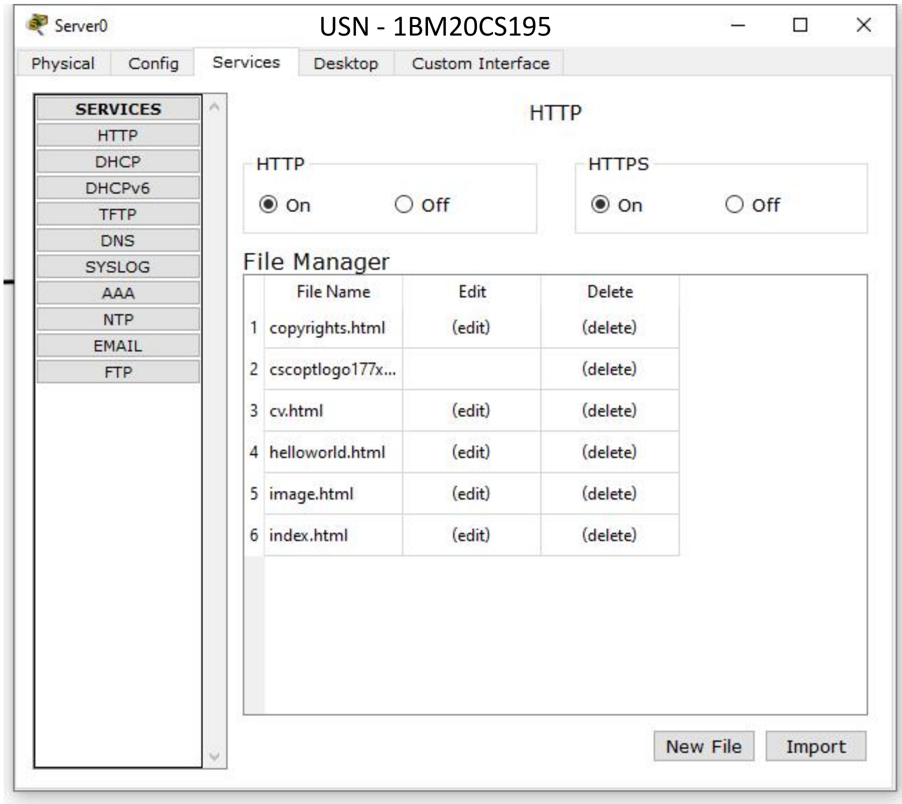
USN - 1BM20CS195

URL http://10.0.0.10

Cisco Packet Tracer

Welcome to BMSCE. Opening doors to new opportunities. Mind Wide Open!

Quick Links:
A small page
Copyrights
Image page
Image
My Resume



USN - 1BM20CS195

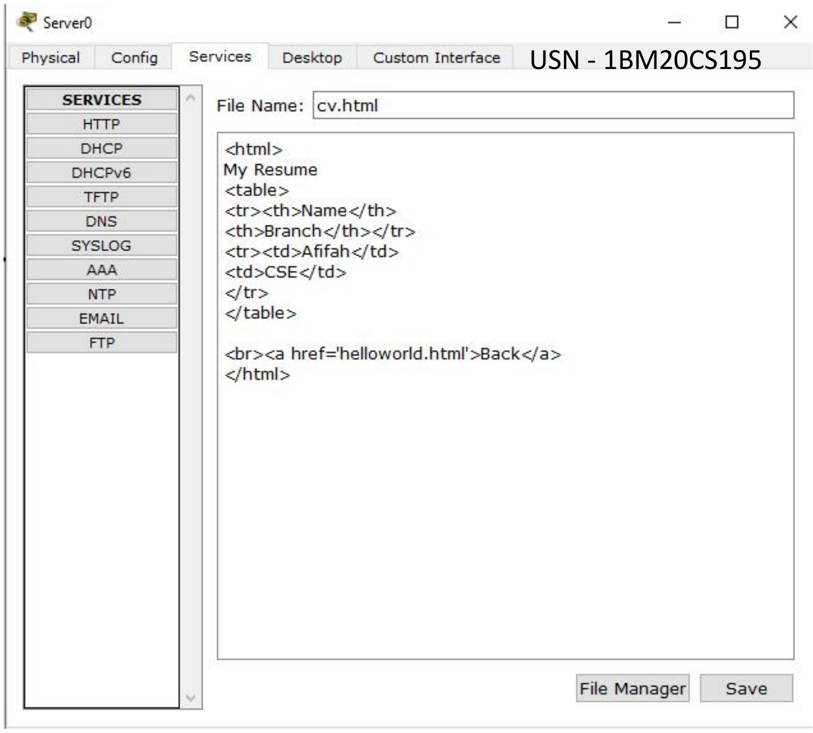
HTTP

HTTP: ● On ○ Off

HTTPS: ● On ○ Off

File Manager

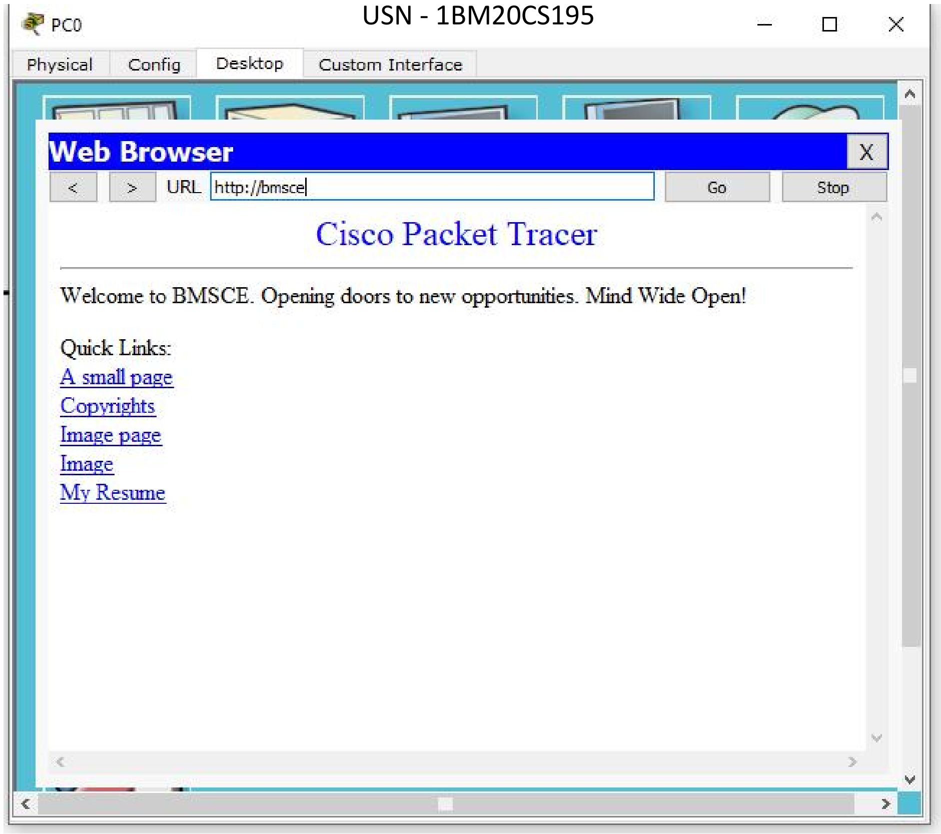
	File Name	Edit	Delete
1	copyrights.html	(edit)	(delete)
2	cscoptlogo177x...		(delete)
3	cv.html	(edit)	(delete)
4	helloworld.html	(edit)	(delete)
5	image.html	(edit)	(delete)
6	index.html	(edit)	(delete)



USN - 1BM20CS195

File Name: cv.html

```
<html>
My Resume
<table>
<tr><th>Name</th>
<th>Branch</th></tr>
<tr><td>Afifah</td>
<td>CSE</td>
</tr>
</table>

<br><a href='helloworld.html'>Back</a>
</html>
```



USN - 1BM20CS195

URL http://bmsce

Cisco Packet Tracer

Welcome to BMSCE. Opening doors to new opportunities. Mind Wide Open!

Quick Links:
A small page
Copyrights
Image page
Image
My Resume

EXPERIMENT - 7

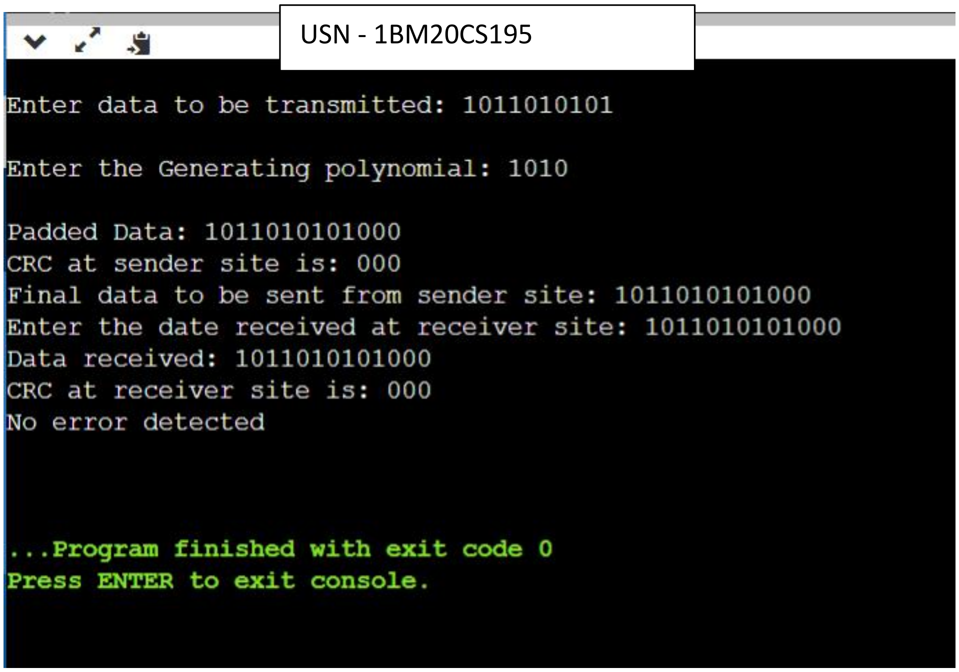
Program : Write a program for error detecting code using CRC-CCITT (16-bits).

Code :

```
#include<stdio.h>

#include<string.h>

#define N strlen(gen_poly)

char data[30];

char check_value[30];

char gen_poly[10];

int data_length,i,j;

void XOR(){

  for(j = 1;j < N; j++)

  check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');

}


void crc(){

  for(i=0;i<N;i++)

    check_value[i]=data[i];

  do{

    if(check_value[0]=='1')

      XOR();


    for(j=0;j<N-1;j++)

      check_value[j]=check_value[j+1];


    check_value[j]=data[i++];
    }while(i<=data_length+N-1);
}
void receiver(){
    printf("Enter the date received at receiver site: ");
    scanf("%s", data);
    printf("Data received: %s", data);
    crc();
    for(i=0;(i<N-1) && (check_value[i]!='1');i++);
        if(i<N-1){
            printf("\nCRC at receiver site is: %s",check_value);
            printf("\nError detected!\n\n");
        }

        else{
            printf("\nCRC at receiver site is: %s",check_value);
             printf("\nNo error detected\n\n");
        }
}
int main(){
    printf("\nEnter data to be transmitted: ");
    scanf("%s",data);
    printf("\nEnter the Generating polynomial: ");
    scanf("%s",gen_poly);

    data_length=strlen(data);

    for(i=data_length;i<data_length+N-1;i++)
        data[i]='0';
```

```
   printf("\nPadded Data: %s",data);
   crc();


   printf("\nCRC at sender site is: %s",check_value);


   for(i=data_length;i<data_length+N-1;i++)
       data[i]=check_value[i-data_length];
  printf("\nFinal data to be sent from sender site: %s\n",data);
  receiver();
   return 0;
}
```

Output :



USN - 1BM20CS195

```
Enter data to be transmitted: 1011010101

Enter the Generating polynomial: 1010

Padded Data: 1011010101000
CRC at sender site is: 000
Final data to be sent from sender site: 1011010101000
Enter the date received at receiver site: 1011010101000
Data received: 1011010101000
CRC at receiver site is: 000
No error detected



...Program finished with exit code 0
Press ENTER to exit console.
```

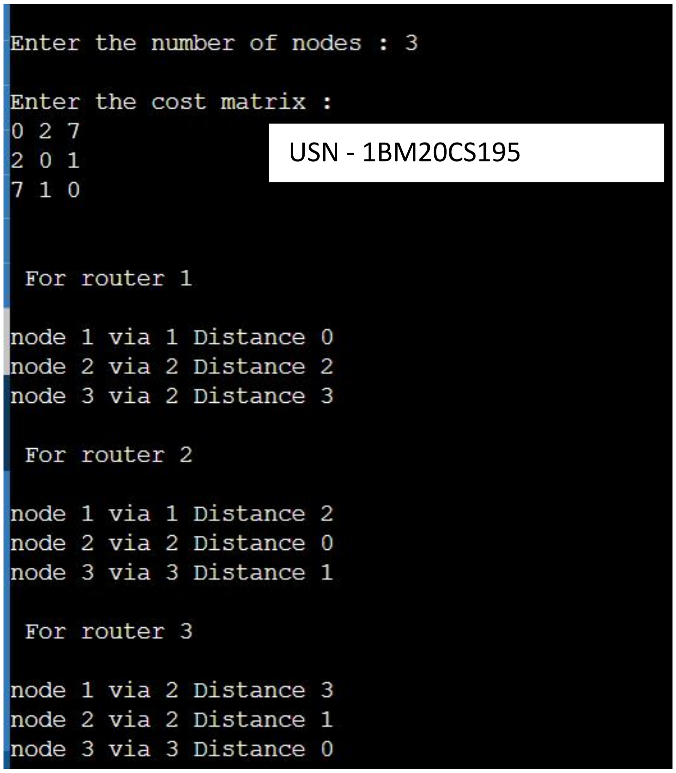
EXPERIMENT - 8

Program : Write a program for distance vector algorithm to find suitable path for transmission.

Code :

```
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];
            rt[i].from[j]=j;
        }
    }
      do
      {
         count=0;
         for(i=0;i<nodes;i++)

         for(j=0;j<nodes;j++)
         for(k=0;k<nodes;k++)
            if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
            {
               rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
               rt[i].from[j]=k;
               count++;
            }
      }while(count!=0);
      for(i=0;i<nodes;i++)
      {
         printf("\n\n For router %d\n",i+1);
         for(j=0;j<nodes;j++)
         {
            printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
         }
      }
   printf("\n\n");

}
```

Output :



```
Enter the number of nodes : 3

Enter the cost matrix :
0 2 7
2 0 1
7 1 0


 For router 1

node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 2 Distance 3

 For router 2

node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1

 For router 3

node 1 via 2 Distance 3
node 2 via 2 Distance 1
node 3 via 3 Distance 0
```

USN - 1BM20CS195

EXPERIMENT - 9

Program : Implement Dijkstra’s algorithm to compute the shortest path for a given topology.

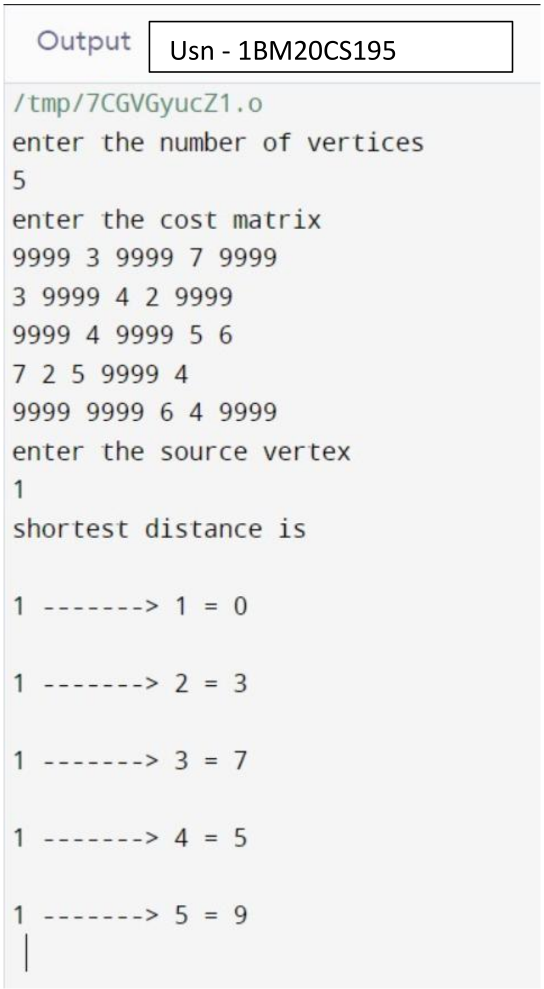
Code :

```
#include<stdio.h>
#include<conio.h>
int c[10][10],n,src;
void dijkistra();
int main()
{
   printf("\nenter the number of vertices\n");
   scanf("%d",&n);
   printf("\nenter the cost matrix \n");
   for(int i=1;i<=n;i++)
   {
      for(int j=1;j<=n;j++)
      {
         scanf("%d",&c[i][j]);
      }
   }
   printf("\nenter the source vertex\n");
   scanf("%d",&src);
   dijkistra();
   return 1;
}

void dijkistra()
{
```

```
int dist[10],vis[10],j,count,min,u;
for(j=1;j<=n;j++)
{
   dist[j]=c[src][j];
}
for(j=1;j<=n;j++)
{
   vis[j]=0;
}
dist[src]=0;
vis[src]=1;
count=1;
while(count!=n)
{
   min=9999;
   for(j=1;j<=n;j++)
   {
      if(dist[j]<min && vis[j]!=1)
      {
         min=dist[j];
         u=j;
      }
   }
   vis[u]=1;
   count++;
   for(j=1;j<=n;j++)
   {
      if(min+c[u][j]<dist[j] && vis[j]!=1)
```

```
            {
                dist[j]=min+c[u][j];
            }
        }
    }
    printf("\n shortest distance is \n");
    for(j=1;j<=n;j++)
    {
        printf("\n%d -------> %d = %d \n ",src,j,dist[j]);
    }
}
```

Output :



Output

Usn - 1BM20CS195

```
/tmp/7CGVGyucZ1.o
enter the number of vertices
5
enter the cost matrix
9999 3 9999 7 9999
3 9999 4 2 9999
9999 4 9999 5 6
7 2 5 9999 4
9999 9999 6 4 9999
enter the source vertex
1
shortest distance is

1 -------> 1 = 0

1 -------> 2 = 3

1 -------> 3 = 7

1 -------> 4 = 5

1 -------> 5 = 9
```

EXPERIMENT - 10

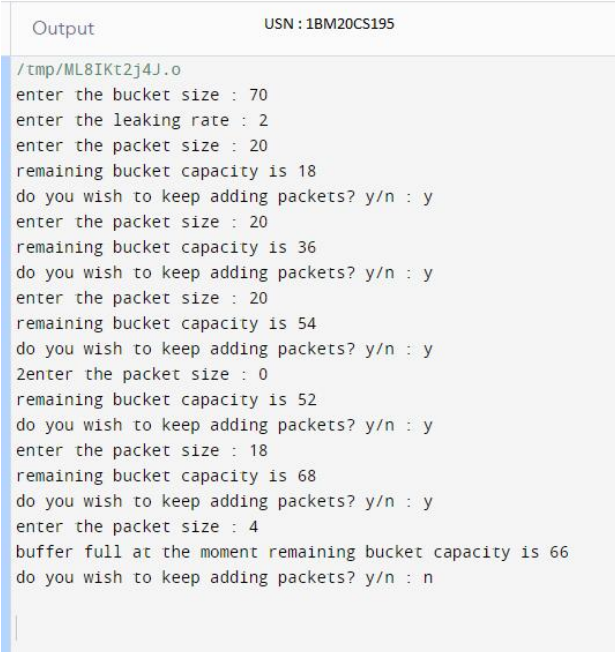
Program : Write a program for congestion control using Leaky bucket algorithm.

Code :

```
#include <stdio.h>
int main() {
 int packet=0,bsize=0,rate=0;
 int capacity=0;
 char ans='y';


  printf("enter the bucket capacity: ");
  scanf("%d",&capacity);
  printf("enter the leaking rate: ");
  scanf("%d",&rate);
  while(ans=='y')
  {
      printf("\nenter the packet size: ");
      scanf("%d",&packet);


       if((bsize+packet) > capacity)
      {
         printf("\nbuffer full at the moment ");
      }


      else if((bsize+packet) <= capacity)
      {
         bsize+=packet;
```

```
        }
        bsize-=rate;
        printf("\nremaining bucket capacity is %d",bsize);
        printf("\ndo you wish to keep adding packets? y/n: ");
        scanf("%s",&ans);

}
return 0;
}
```

Output :



USN : 1BM20CS195

```
/tmp/ML8IKt2j4J.o
enter the bucket size : 70
enter the leaking rate : 2
enter the packet size : 20
remaining bucket capacity is 18
do you wish to keep adding packets? y/n : y
enter the packet size : 20
remaining bucket capacity is 36
do you wish to keep adding packets? y/n : y
enter the packet size : 20
remaining bucket capacity is 54
do you wish to keep adding packets? y/n : y
2enter the packet size : 0
remaining bucket capacity is 52
do you wish to keep adding packets? y/n : y
enter the packet size : 18
remaining bucket capacity is 68
do you wish to keep adding packets? y/n : y
enter the packet size : 4
buffer full at the moment remaining bucket capacity is 66
do you wish to keep adding packets? y/n : n
```

EXPERIMENT - 11

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

Code :

client.py

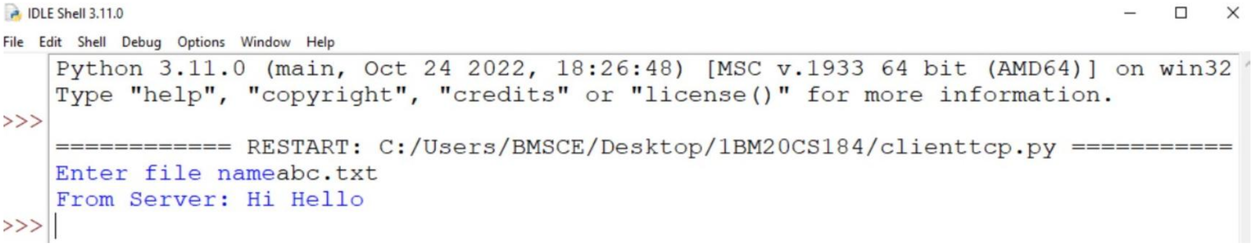
```
from socket import *
serverName='DESKTOP-9CJQB77'
serverPort=12530
clientSocket=socket(AF_INET,SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence=input("Enter file name")
clientSocket.send(sentence.encode())
filecontents=clientSocket.recv(1024).decode()
print('From Server:',filecontents)
clientSocket.close()
```

server.py

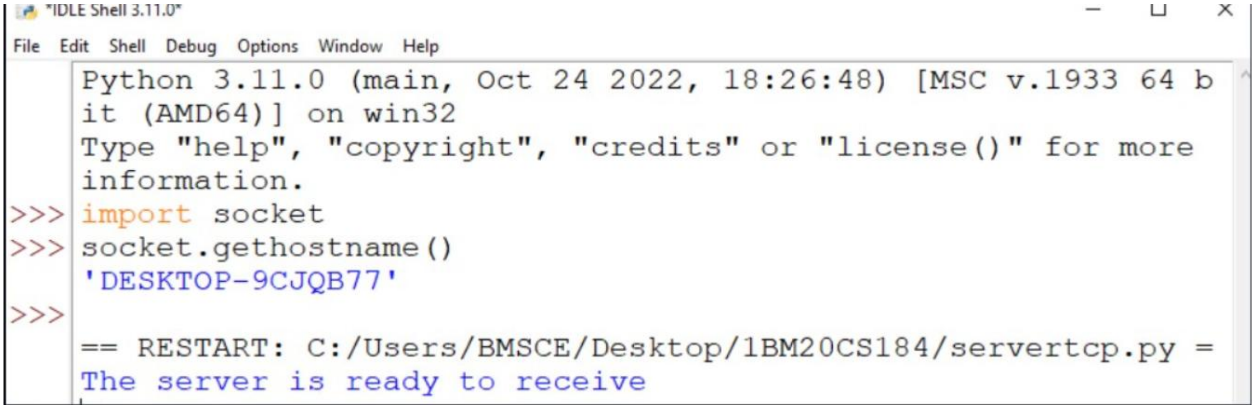
```
from socket import *
serverName='DESKTOP-9CJQB77'
serverPort=12530
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
print("The server is ready to receive")
```

```
while(1):
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    file.close()
    connectionSocket.close()
```

Output :



```
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: C:/Users/BMSCE/Desktop/1BM20CS184/clienttcp.py ===========
Enter file nameabc.txt
From Server: Hi Hello
>>>
```



```
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> import socket
>>> socket.gethostname()
'DESKTOP-9CJQB77'
>>>
== RESTART: C:/Users/BMSCE/Desktop/1BM20CS184/servertcp.py =
The server is ready to receive
```

]

EXPERIMENT - 12

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

Code :

clientudp.py

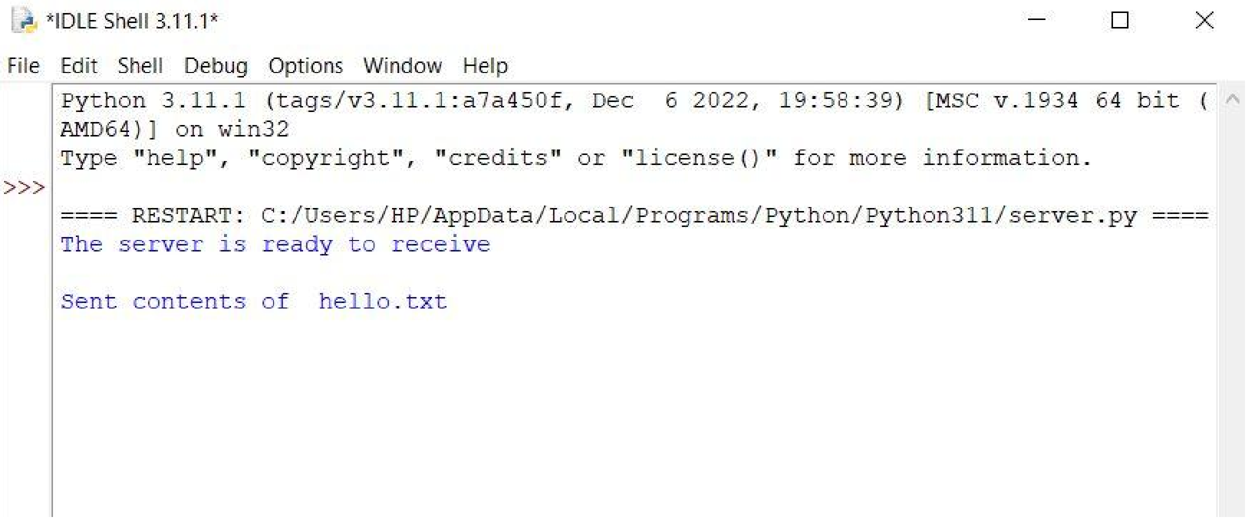
```
from socket import *
        serverName = "127.0.0.1"
        serverPort = 12000
        clientSocket = socket(AF_INET, SOCK_DGRAM)
        sentence = input("\nEnter file name: ")
        clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
        filecontents,serverAddress = clientSocket.recvfrom(2048)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

serverudp.py

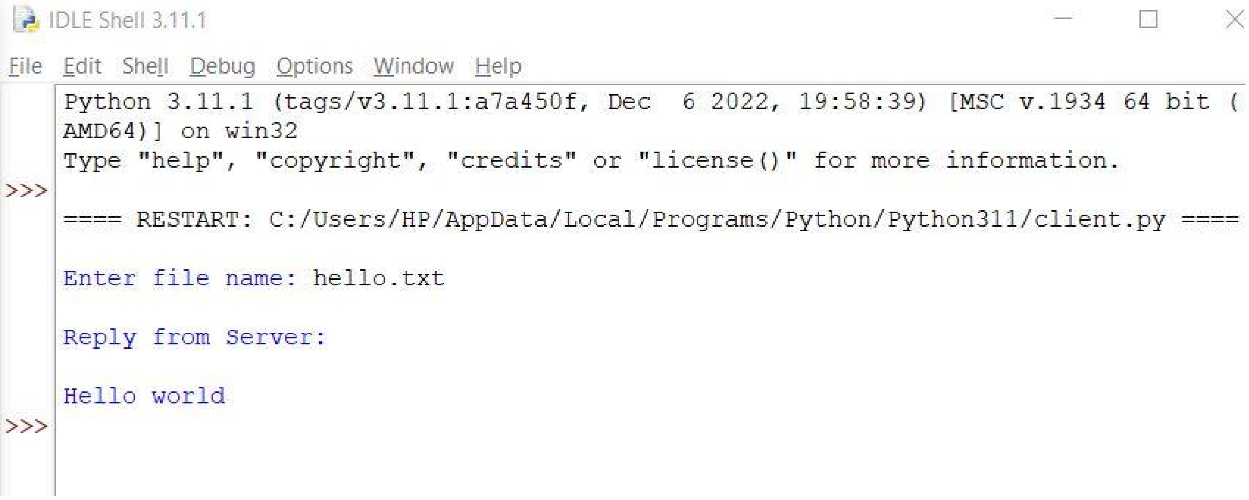
```
from socket import *
serverPort = 12000
```

```
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 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()
```

Output :



```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/HP/AppData/Local/Programs/Python/Python311/server.py ====
The server is ready to receive

Sent contents of  hello.txt
```



```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/HP/AppData/Local/Programs/Python/Python311/client.py ====

Enter file name: hello.txt

Reply from Server:

Hello world
>>>
```