VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

SANNIDHI M (1BM21CS189)

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
JUN-2023 to SEP-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 **SANNIDHI M(1BM21CS189),** who is a 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 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (22CS4PCCON)** work prescribed for the said degree.

Dr. Shyamala G
Assistant 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.
1	16/6/23	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	1
2	23/6/23	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	8
3	30/7/23	Configure default route, static route to the Router	14
4	14/7/23	Configure DHCP within a LAN and outside LAN.	18
5	21/7/23	Configure RIP routing Protocol in Routers	24
6	4/7/23	Configure OSPF routing protocol	27
7	11/08/23	Demonstrate the TTL/ Life of a Packet	30
8	21/7/23	Configure Web Server, DNS within a LAN.	33
9	4/08/23	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	36
10	18/08/23	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	40
11	11/08/23	To construct a VLAN and make PC's communicate among a VLAN	43
12	11/08/23	To construct a WLAN and make the nodes communicate wirelessly	46
13	18/08/23	Write a program for error detecting code using CRC-CCITT (16-bits).	50
14	03/09/23	Write a program for congestion control using Leaky bucket algorithm.	54
15	03/09/23	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.	58
16	03/09/23	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.	62

WEEK 1

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

Observation book:

EXPERIMENT - 1

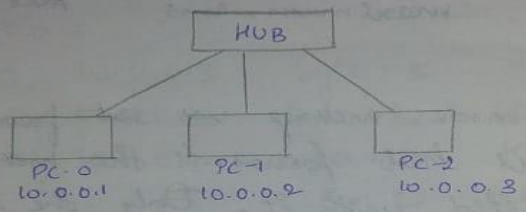
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

AIM:

To make and understand topology using hub, switch and hybrid topology using both.

Network with Hub

TOPOLOGY:



PROCEDURE:

i) Select end users, here PC and generic hub.
ii) Click on end user → config → fast ethernet → assign IP addres
iii) Send a simple PDU message from PC-0 to PC-2 in the simulation mode.
iv) In real time mode click on PC0 and under desktop, command prompt, ping another PC. Ex: ping 10.0.0.3

RESULT:

i) The simple PDU is sent from PC0 to hub.
ii) Hub sends PDU to all ports except the incoming port. The PDU is rejected by other ports except destinati port.
iii) PC2 sends an acknowledgement to hub.
iv) PC0 receives this and transfer is completed.

ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=4ms TTL=128

Reply from 10.0.0.3: bytes=32 time=4ms TTL=128

Reply from 10.0.0.3: bytes=32 time=4ms TTL=128

Reply from 10.0.0.3: bytes=32 time=4ms TTL=128

Ping statistics from 10.0.0.3

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

Approximate round trip times in milli-seconds

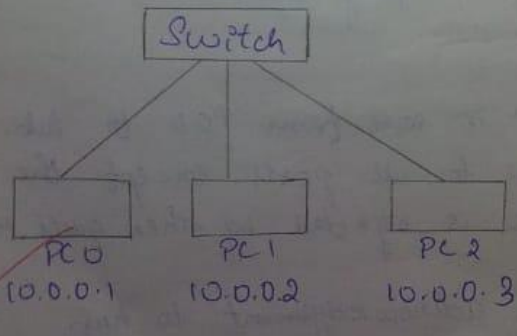
minimum = 4ms, maximum = 4ms, Average = 4ms.

OBSERVATION:

In simulation mode message was sent from source to hubs, this hub forwards the message to all devices connected with it. Only the destination device responds and receives the message whereas all othes devices reject it.

Topology with Switch.

TOPOLOGY:



PROCEDURE

Add a generic switch and connect 3 PCs to it using coppre straight-through wire.

(i) wait for PC and switch connections to be established.
(ii) Set IP address of all end devices by clicking on end users → config → fast ethernet → assign IP address.
(iv) Send a simple PDU request from PC0 to PC2 in simulation mode. In realtime mode ping PC2 from PC0 on Desktop → command prompt.

RESULT

Source - PC0
Destination - PC 2
] status successfull.

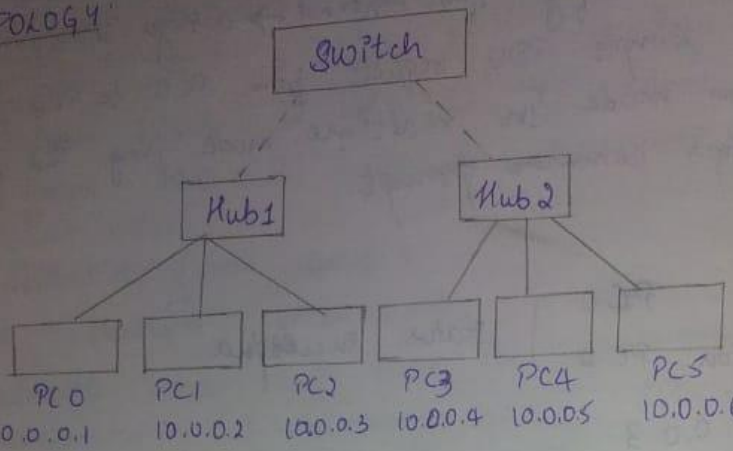
```
Ping 10.0.0.3
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL=128
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL=128
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 128

ping statistics for 10.0.0.3
packets: sent = 4 received = 4 lost = 0
Approximate round trip time (milli second)
Minimum = 0ms, maximum = 0ms, average = 0ms.
```

OBSERVATION

The simple PDU is sent from PC0 to switch. Switch sends this data only to the destination device i.e PC0. The destination device acknowledges and receives the data.

Hybrid Topology.

TOPOLOGY



PROCEDURE

Add generic hubs, generic switch and connect 3 end devices, here PC.

Connect end devices to the two different hubs and then connect two hubs to a switch using copper cross over wire

) Click on end device → config → fast ethernet → assign IP address.

) Send PDU from source device to destination and wait for simulation to finish.

In real time mode ping devices using command prompt.

RESULT:

source PC0
destination PC5
} status successfull.

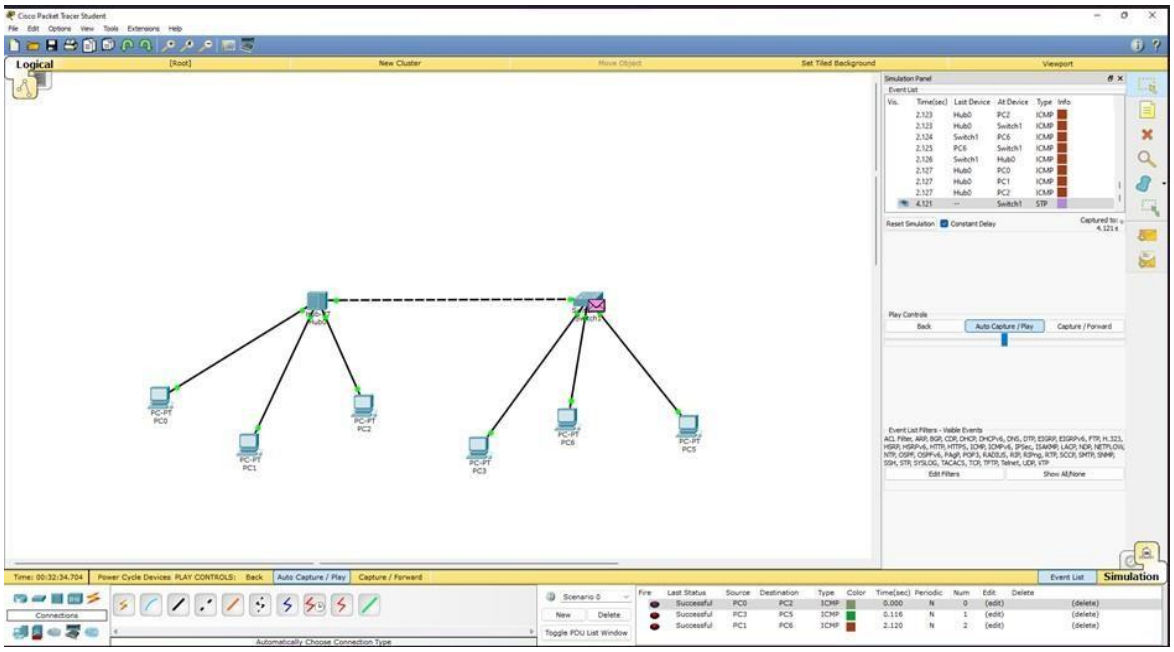
```
Ping 10.0.0.6
Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128
Reply from 10.0.0.6 byte = 32 time = 10ms TTL = 128
Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128
Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128.
Ping statistics for 10.0.0.6
packets sent = 4, received = 4, loss = 0.
```

OBSERVATIONS:

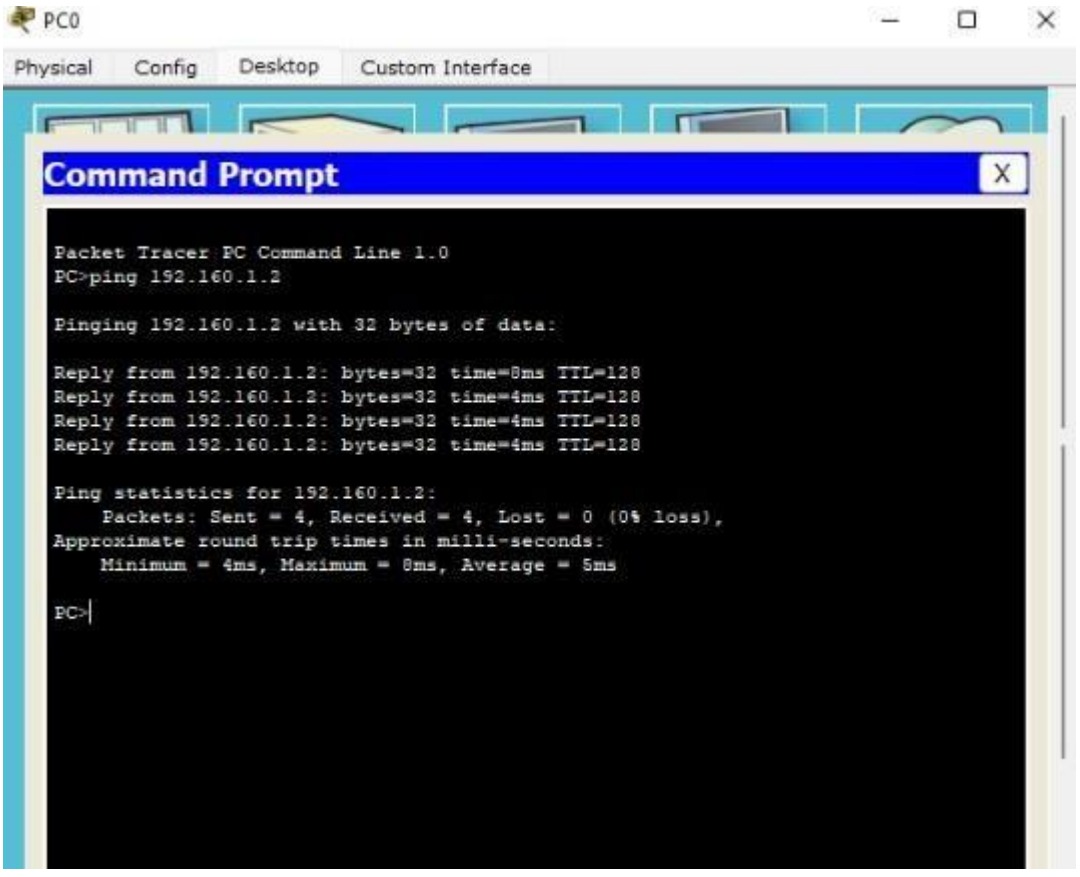
Sending message from PC0 to PC5

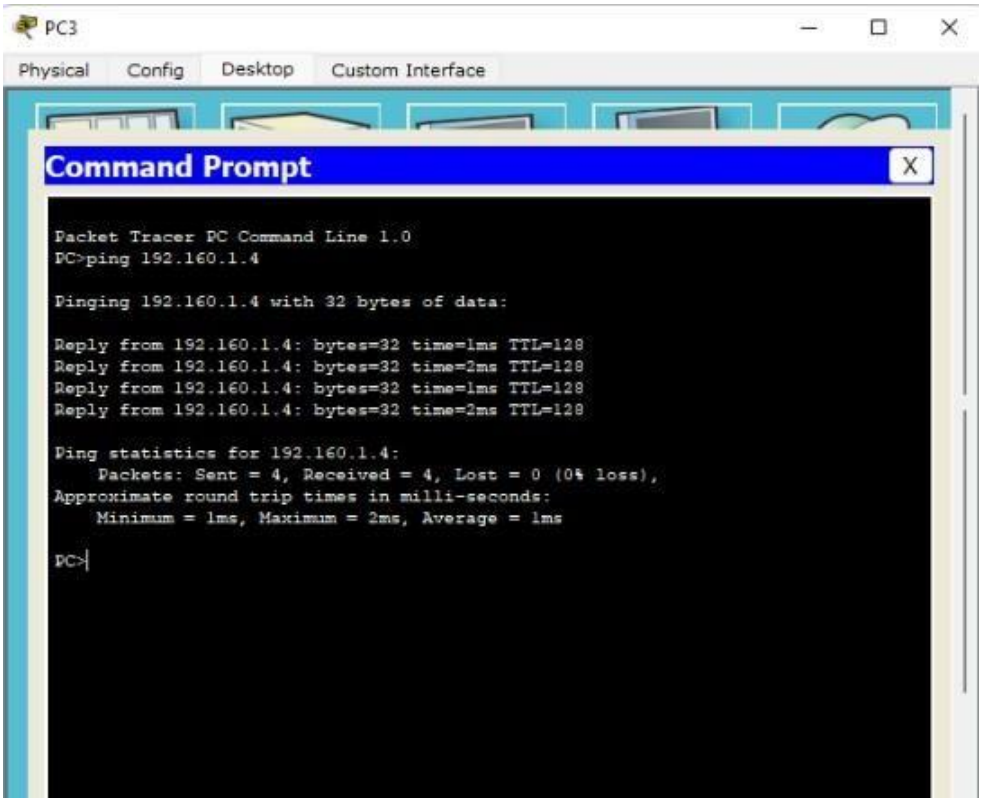
PC1 is the source node which sends the message to Hub 1. Hub 1 sends this message to other PC's connected with it i.e to PC1 & PC2. and also sends the message to switch. PC1 and PC2 rejects the message. Switch sends this message to Hub2. Hub2 sends it to all end devices. PC5 accepts the message and acknowledges where as all other PC's reject it.

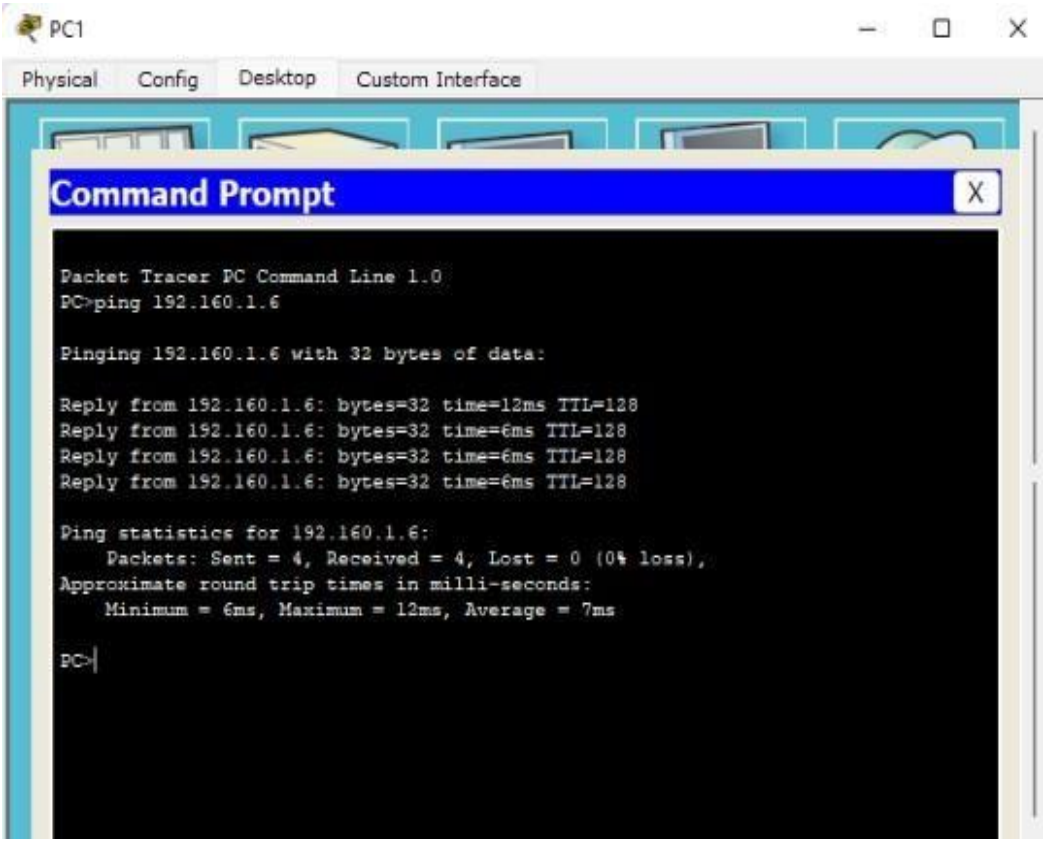
SJT
16/01/23



Output :


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.2
Pinging 192.160.1.2 with 32 bytes of data:
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=4ms TTL=128
Reply from 192.160.1.2: bytes=32 time=4ms TTL=128
Reply from 192.160.1.2: bytes=32 time=4ms TTL=128
Ping statistics for 192.160.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 0ms, Average = 5ms
PC>

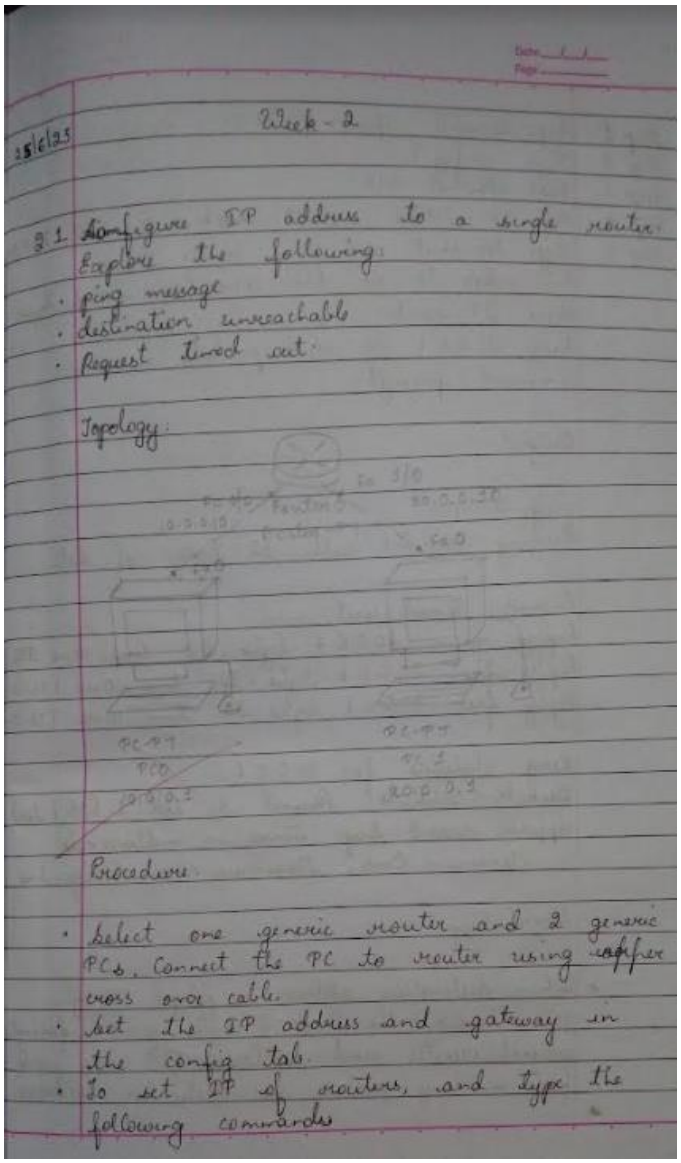

PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.4
Pinging 192.160.1.4 with 32 bytes of data:
Reply from 192.160.1.4: bytes=32 time=1ms TTL=128
Reply from 192.160.1.4: bytes=32 time=2ms TTL=128
Reply from 192.160.1.4: bytes=32 time=1ms TTL=128
Reply from 192.160.1.4: bytes=32 time=2ms TTL=128
Ping statistics for 192.160.1.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 2ms, Average = 1ms
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.6
Pinging 192.160.1.6 with 32 bytes of data:
Reply from 192.160.1.6: bytes=32 time=12ms TTL=128
Reply from 192.160.1.6: bytes=32 time=6ms TTL=128
Reply from 192.160.1.6: bytes=32 time=6ms TTL=128
Reply from 192.160.1.6: bytes=32 time=6ms TTL=128
Ping statistics for 192.160.1.6:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 12ms, Average = 7ms
PC>

WEEK 2

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

Observation book :



Week - 2

25/6/25

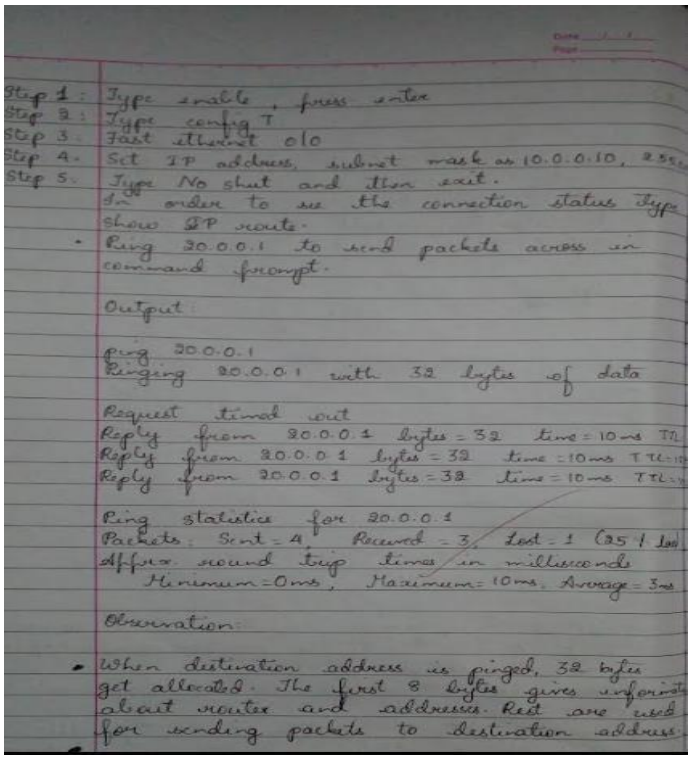
3.1 Configure IP address to a single router. Explore the following:

- ping message
- destination unreachable
- Request timed out.

Topology:

Procedure:

- Select one generic router and 2 generic PCs. Connect the PC to router using copper cross over cable.
- Set the IP address and gateway in the config tab.
- To set IP of routers, and type the following commands



Step 1 : Type enable, press enter
Step 2 : Type config T
Step 3 : Fast ethernet 0/0
Step 4 : Set IP address, subnet mask as 10.0.0.10, 255.
Step 5 : Type No shut and then exit.
In order to see the connection status Type show IP route.
Ping 30.0.0.1 to send packets across in command prompt.

Output:

Ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data

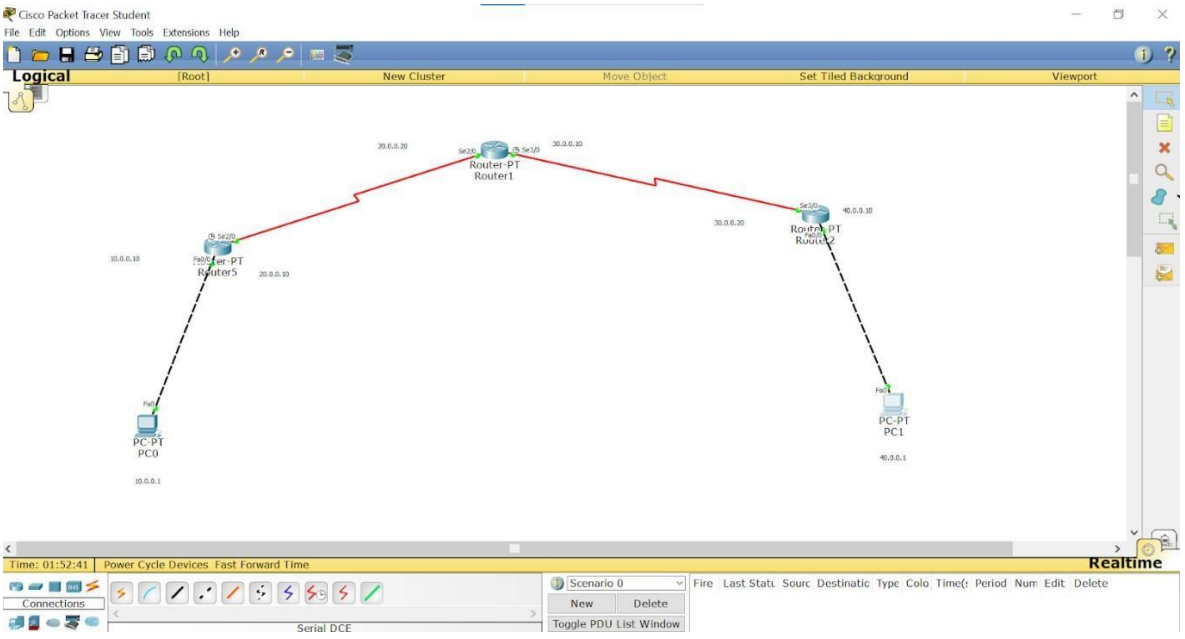
Request timed out
Reply from 20.0.0.1 bytes = 32 time = 10ms TTL
Reply from 20.0.0.1 bytes = 32 time = 10ms TTL=
Reply from 20.0.0.1 bytes = 32 time = 10ms TTL=

Ping statistics for 20.0.0.1
Packets: Sent = 4 Received = 3, Lost = 1 (25 %
Approx round trip times in milliseconds
Minimum = 0ms, Maximum = 10ms, Average = 3ms

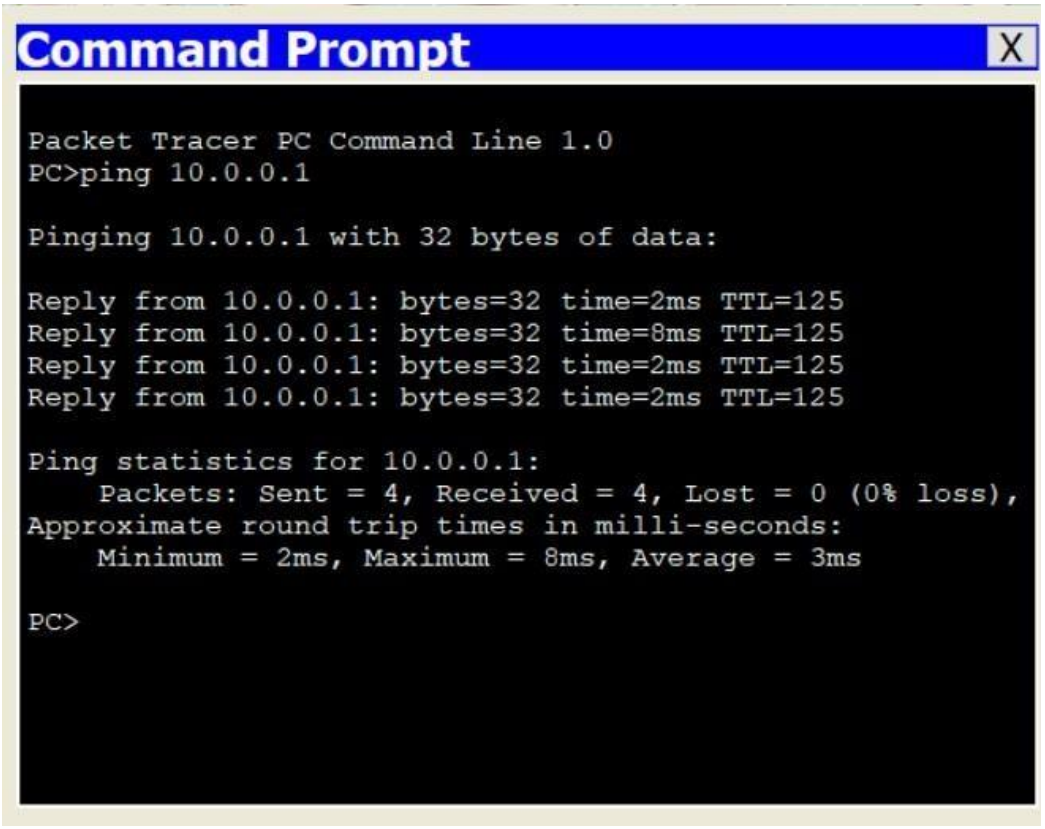
Observation:

- When destination address is pinged, 32 bytes get allocated. The first 8 bytes gives information about router and addresses. Rest are used for sending packets to destination address.

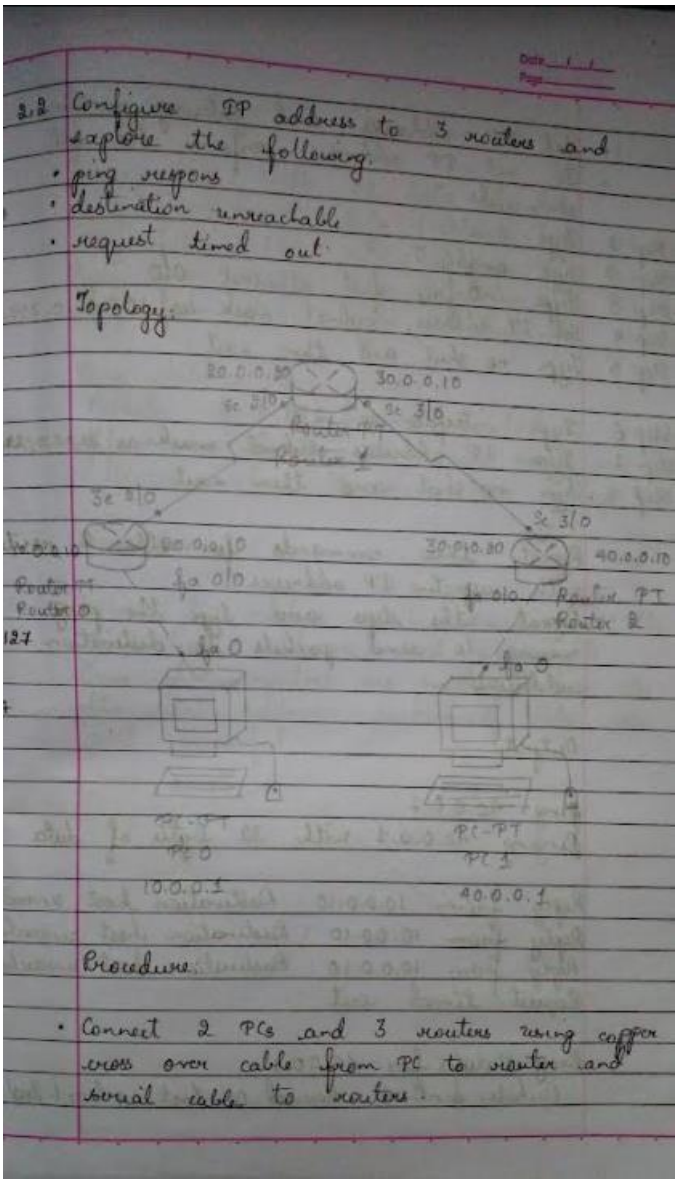
Topology :



Output :


Command Prompt
X
Packet Tracer PC Command Line 1.0
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=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 3ms
PC>

Observation book :



2.2 Configure IP address to 3 routers and explore the following

- ping respons
- destination unreachable
- request timed out

Topology:

30.0.0.10
Se 3/0
Router PT
Router 1
Se 3/0
Se 2/0
30.0.0.30
40.0.0.10
Router PT
Router 2
Fa 0/0
Router PT
Router 0
127
Fa 0
Fa 0
PC-PT
PC 0
10.0.0.1
PC-PT
PC 1
40.0.0.1

Procedure:

- Connect 2 PCs and 3 routers using copper cross over cable from PC to router and serial cable to routers.

- Set IP addresses and gateway numbers
- To set IP addresses perform the following commands.

Step 1 Type enable
Step 2 Type config T
Step 3 Type interface fast ethernet 0/0
Step 4 Set IP address, subnet mask as 10.0.0.10, 255.0.0
Step 5 Type no shut and then exit

Step 6 Type interface se 2/0
Step 7 Type IP address, subnet mask as 20.0.0.10, 255.0.0
Step 8 Type no shut and then exit

- Repeat these commands for other 2 routers with respective IP addresses.
- Repeat the steps and type the ping message to send packets to destination addresses.

Output:

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
Request timed out

Ping statistics for 40.0.0.1
Packets: Sent = 4 Received = 0 Lost = 4 (100% loss)

ping 10.0.0.1

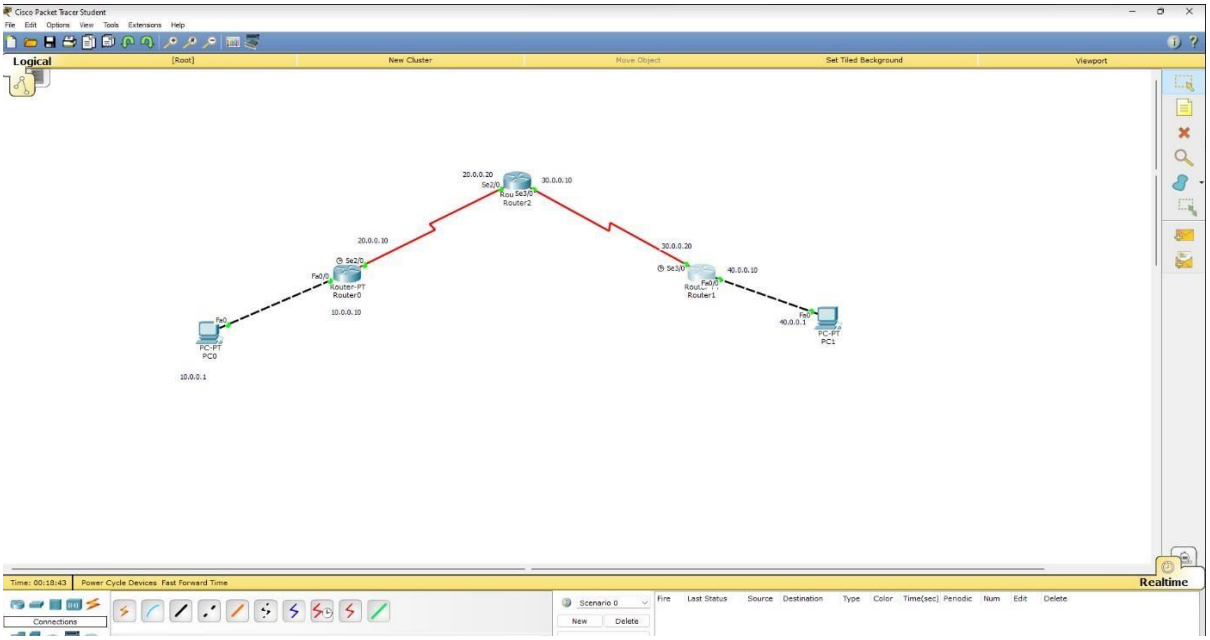
Pinging 10.0.0.1 with 32 bytes of data
Reply from 10.0.0.1 : bytes = 32 time = 2ms, TTL = 125
Reply from 10.0.0.1 : bytes = 32 time = 8ms, TTL = 125
Reply from 10.0.0.1 : bytes = 32 time = 8ms TTL = 125
Reply from 10.0.0.1 : bytes = 32 time = 8ms TTL = 125

Ping statistics for 10.0.0.1
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approx round trip times in milli seconds.
Minimum = 2ms, Maximum = 8ms, Average = 5ms

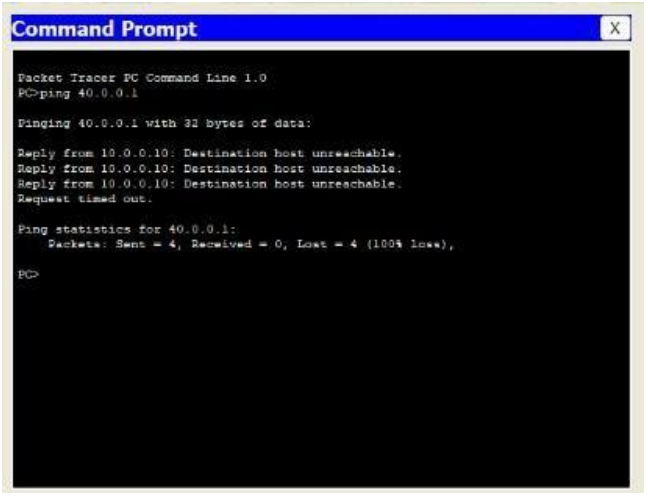
Observation:

When the routers don't know about the destination addresses, host unreachable.
Once the routers are made to hop to the next addresses, packets of data are sent successful.

Topology :



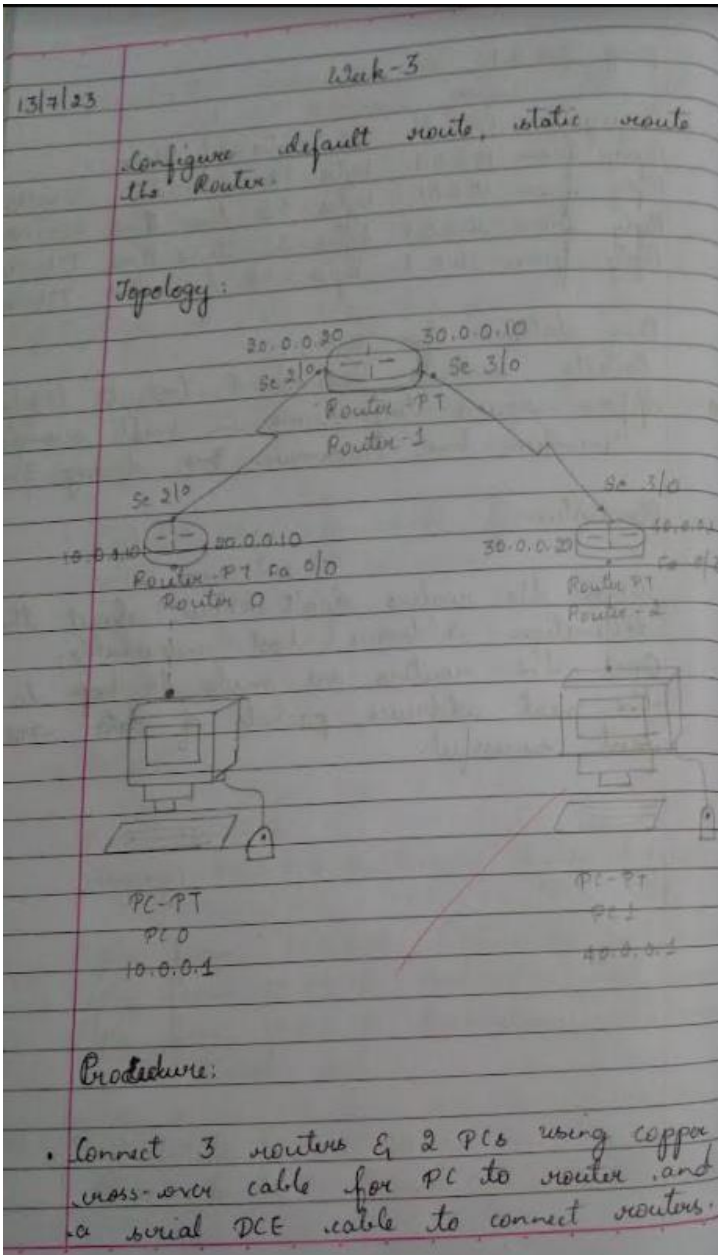
Output :

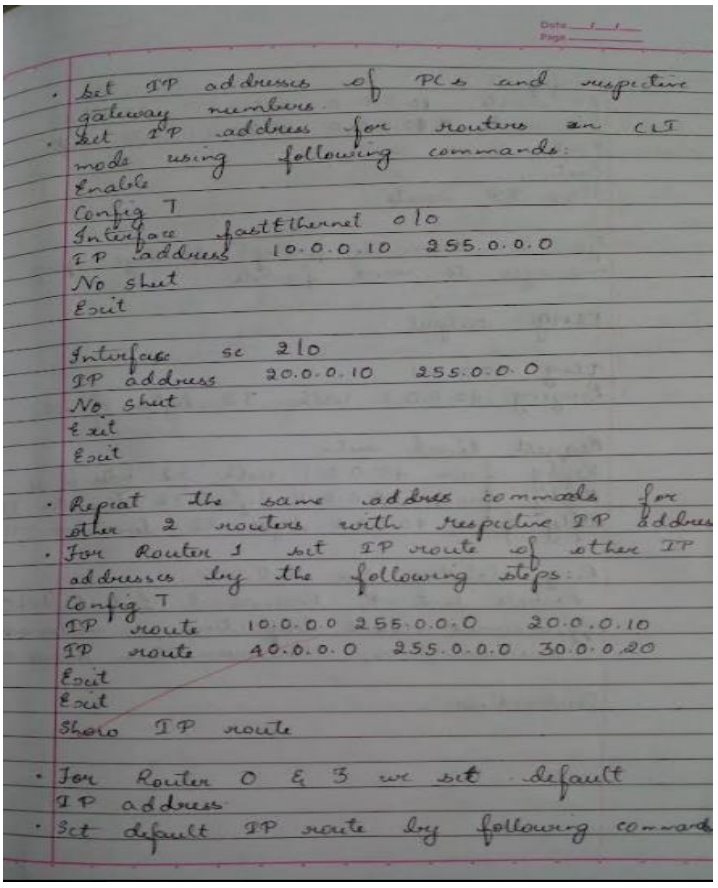


WEEK 3

Configure default route, static route to the Router

Observation book :





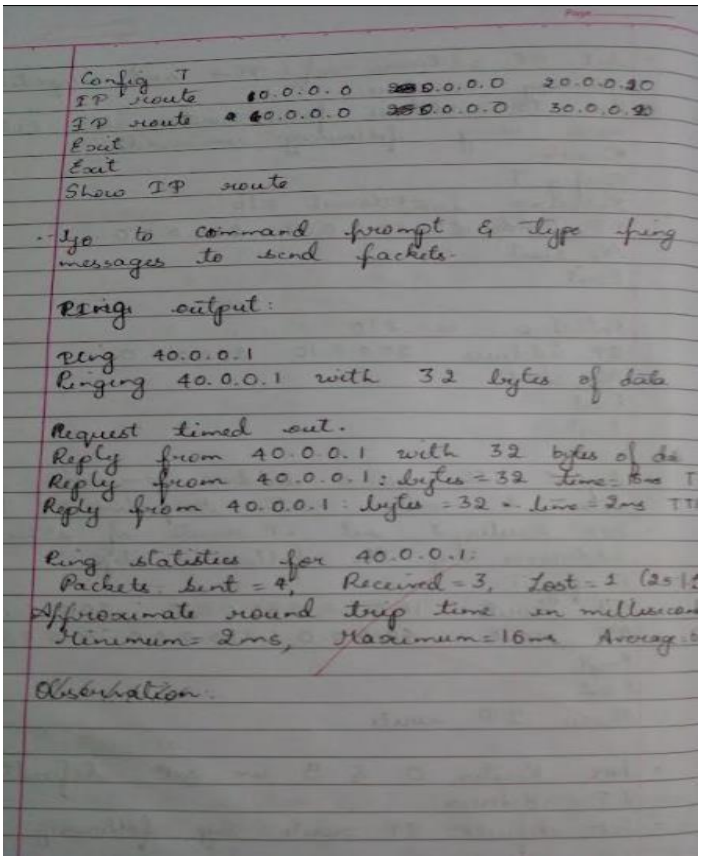
- Set IP addresses of PCs and respective gateway numbers.
- Set IP address for routers in CLI mode using following commands:

```
Enable
Config T
Interface fastEthernet 0/0
IP address 10.0.0.10 255.0.0.0
No shut
Exit

Interface se 2/0
IP address 20.0.0.10 255.0.0.0
No shut
Exit
Exit
```

- Repeat the same address commands for other 2 routers with respective IP address
- For Router 1 set IP route of other IP addresses by the following steps:

```
Config T
IP route 10.0.0.0 255.0.0.0 20.0.0.10
IP route 40.0.0.0 255.0.0.0 30.0.0.20
Exit
Exit
Show IP route
```

- For Router 0 & 3 we set default IP address
- Set default IP route by following comand



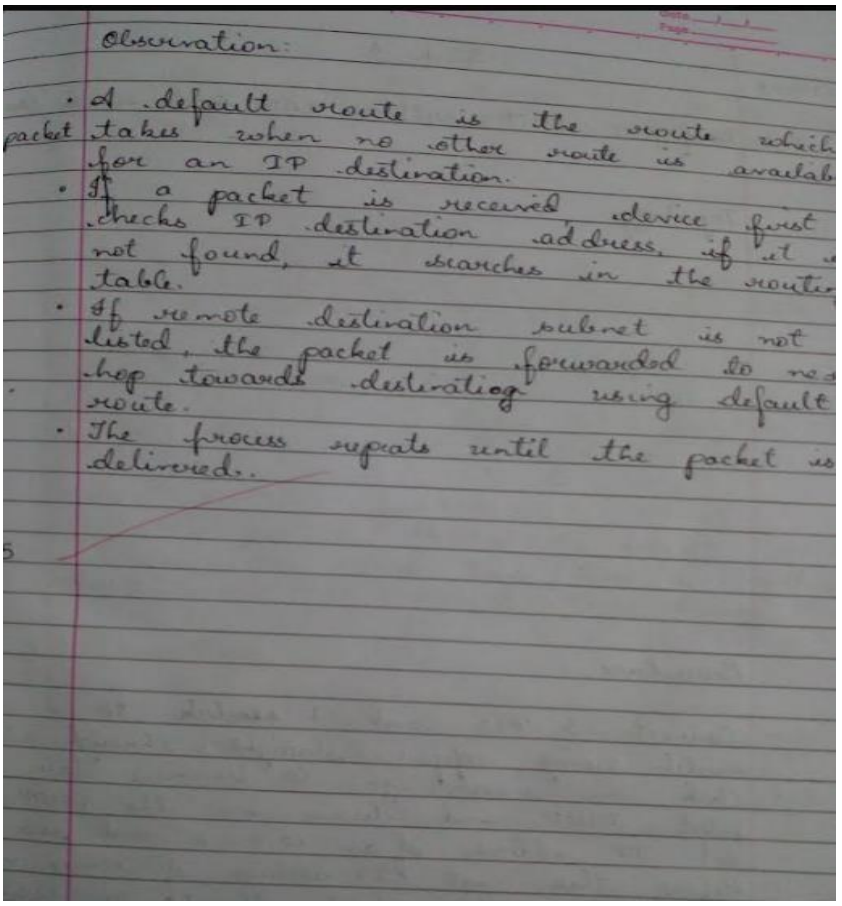
```
Config T
IP route 10.0.0.0 255.0.0.0 20.0.0.10
IP route 40.0.0.0 255.0.0.0 30.0.0.20
Exit
Exit
Show IP route
```

- Go to command prompt & type ping messages to send packets.

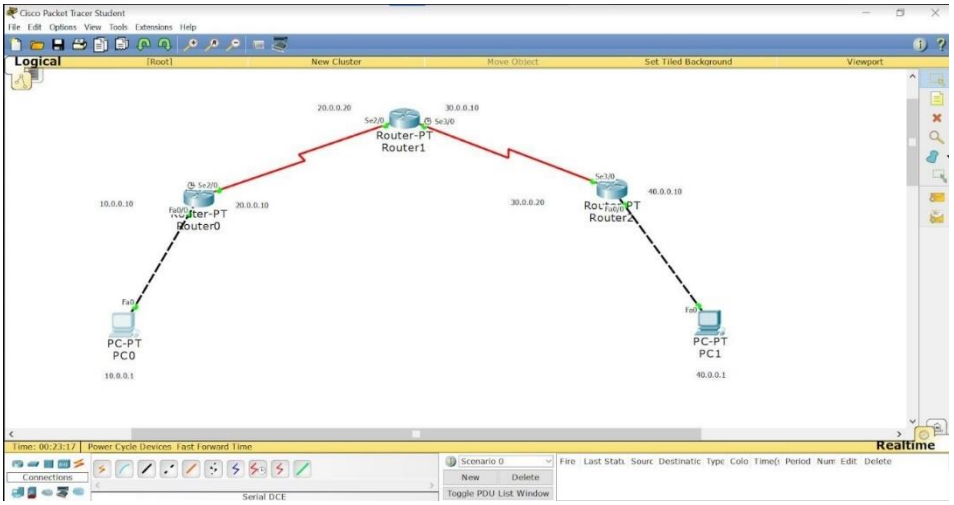
Ping output:

```
Ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data

Request timed out.
Reply from 40.0.0.1 with 32 bytes of da
Reply from 40.0.0.1: bytes=32 time=8ms T
Reply from 40.0.0.1: bytes=32 time=2ms TT

Ping statistics for 40.0.0.1:
  Packets: Sent = 4, Received = 3, Lost = 1 (25 %
Approximate round trip time in millisecon
  Minimum = 2ms, Maximum = 16ms, Average =
```

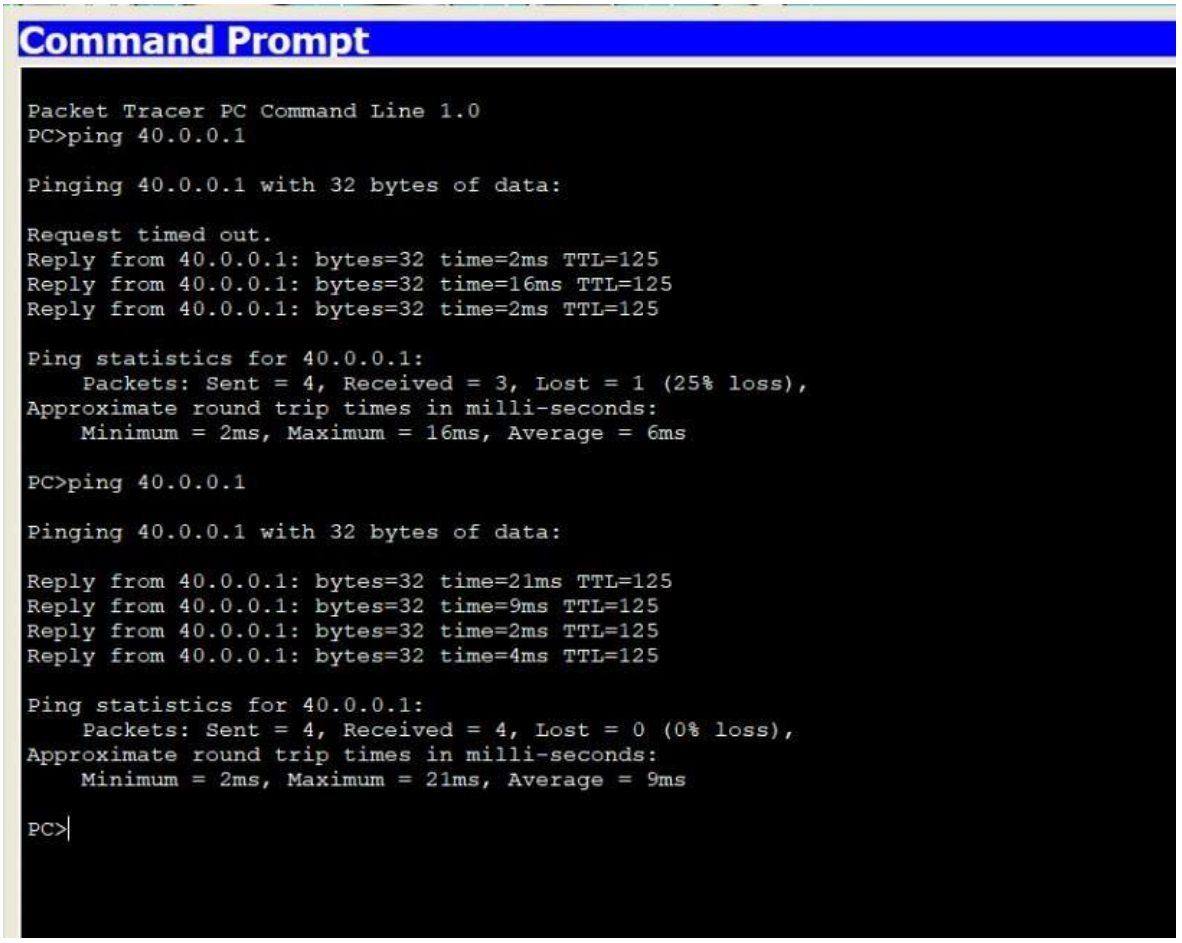
Observation:


Observation:
• A default route is the route which packet takes when no other route is available for an IP destination.
• If a packet is received device first checks IP destination address, if it is not found, it searches in the routing table.
• If remote destination subnet is not listed, the packet is forwarded to next hop towards destination using default route.
• The process repeats until the packet is delivered.

Topology :


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
10.0.0.1
40.0.0.1
Time: 00:23:17
Power Cycle Devices
Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window

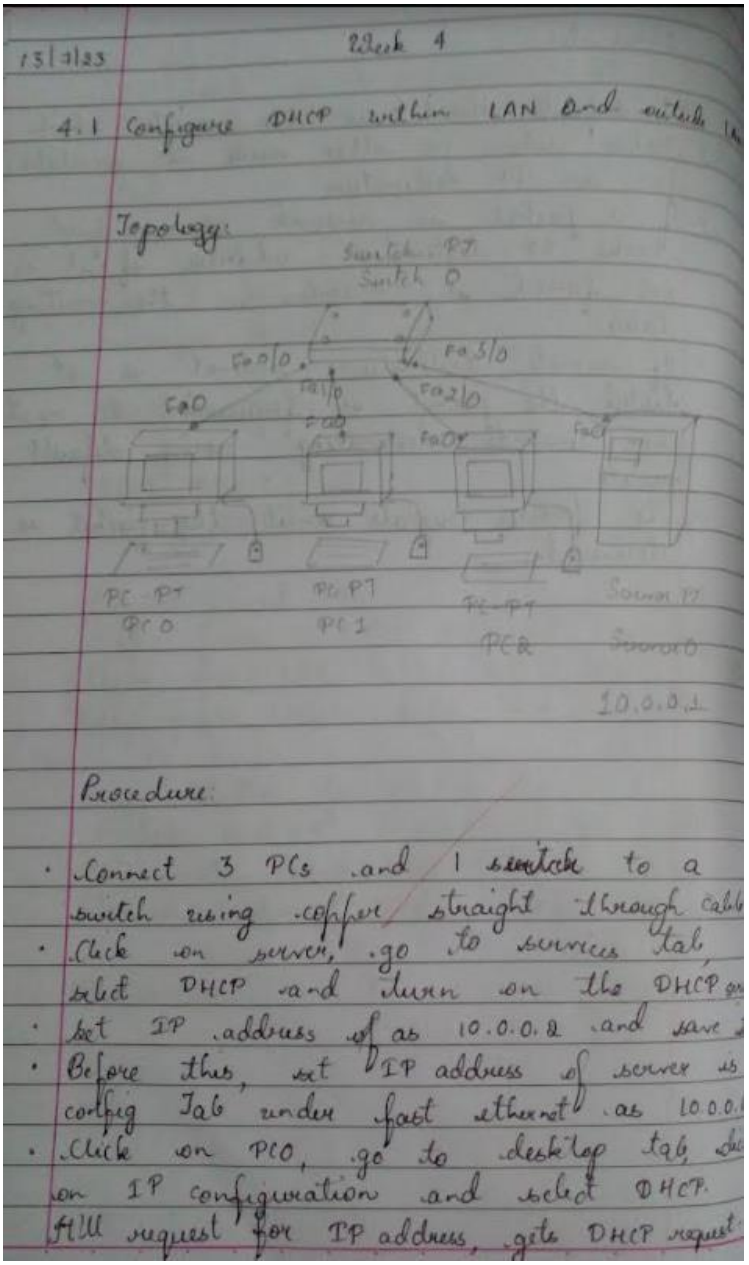
Output :

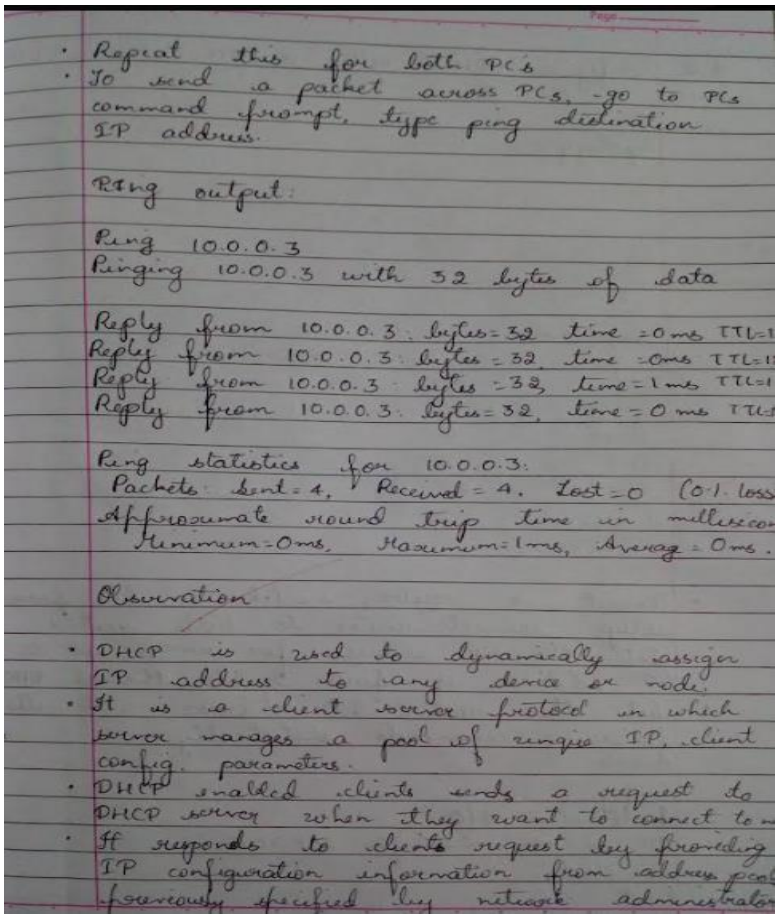

Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 16ms, Average = 6ms
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=21ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms 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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 21ms, Average = 9ms
PC>

WEEK 4

Configure DHCP within a LAN and outside LAN.

Observation book :





- Repeat this for both PCs
- To send a packet across PCs, go to PCs command prompt, type ping destination IP address.

Ping output:

Ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data

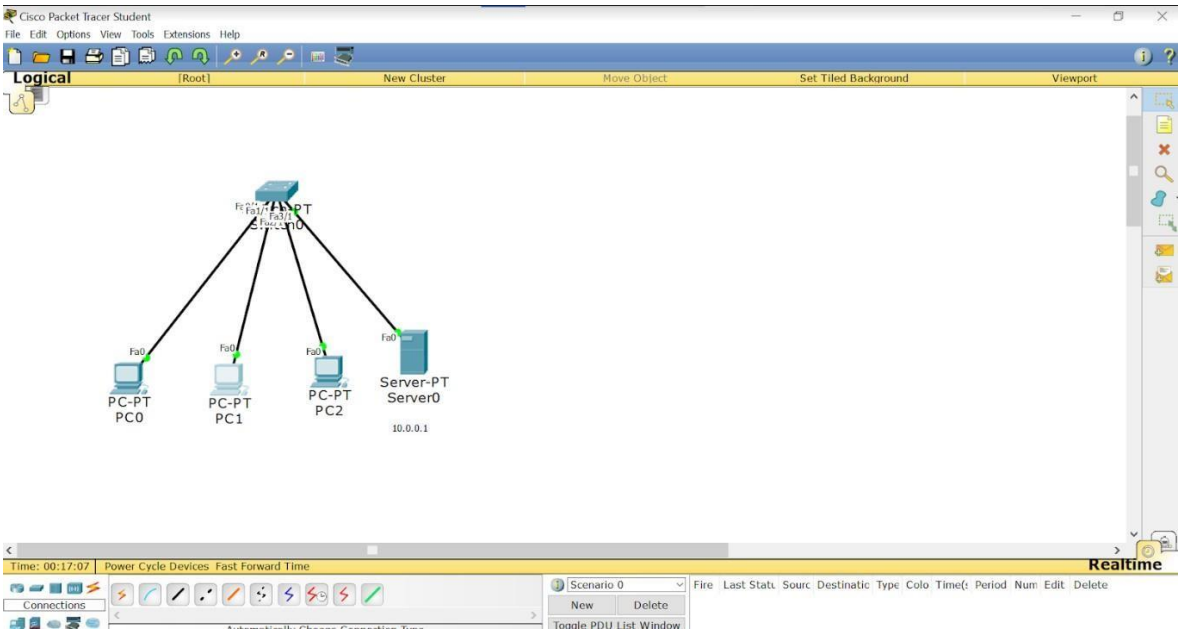
Reply from 10.0.0.3: bytes=32 time=0ms TTL=1
Reply from 10.0.0.3: bytes=32 time=0ms TTL=1
Reply from 10.0.0.3: bytes=32 time=1ms TTL=1
Reply from 10.0.0.3: bytes=32 time=0ms TTL=1

Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip time in milliseconds:
Minimum=0ms, Maximum=1ms, Average=0ms.

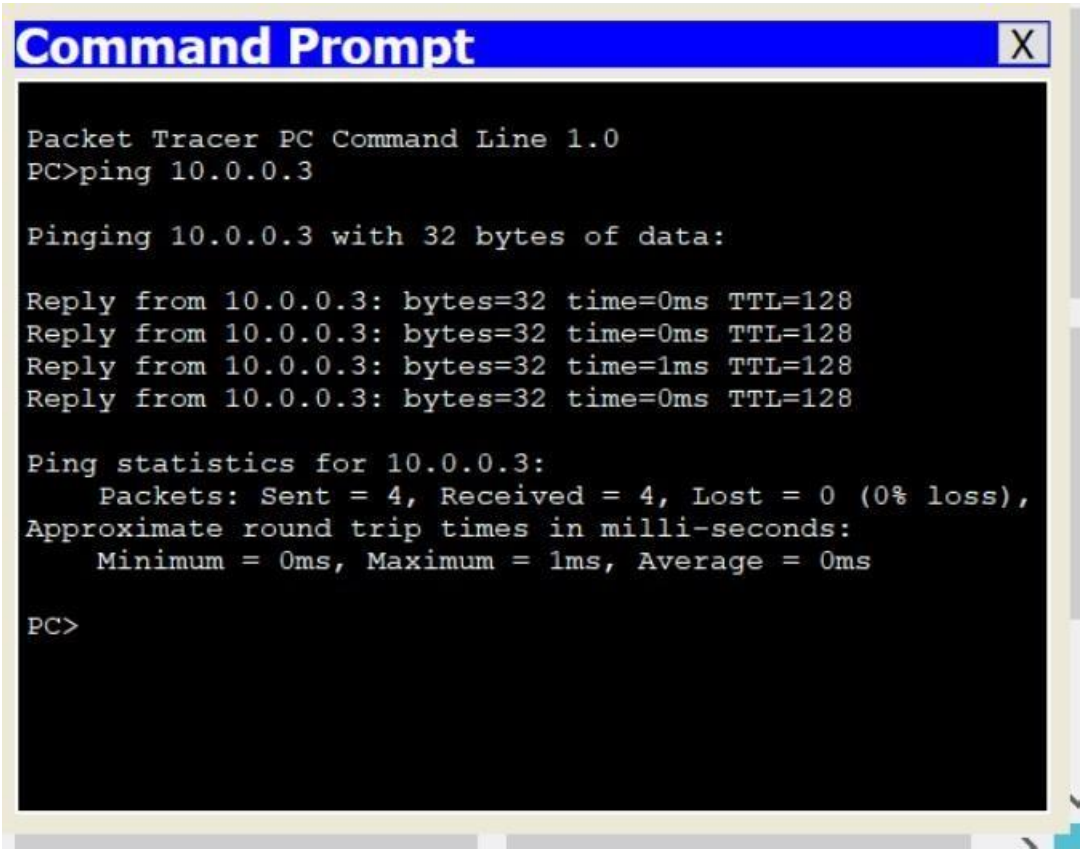
Observation:

- DHCP is used to dynamically assign IP address to any device or node.
- It is a client server protocol in which server manages a pool of unique IP, client config parameters.
- DHCP enabled clients sends a request to DHCP server when they want to connect to n
- It responds to clients request by providing IP configuration information from address pool previously specified by network administrator

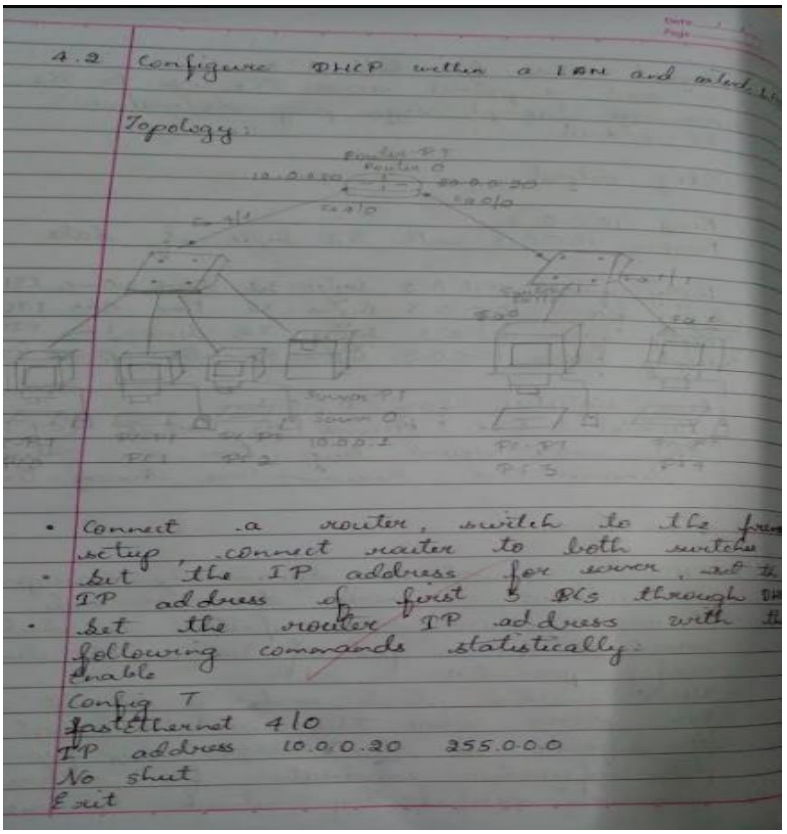
Topology :



Output :


Command Prompt
X
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=0ms 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 = 1ms, Average = 0ms
PC>

Observation book :

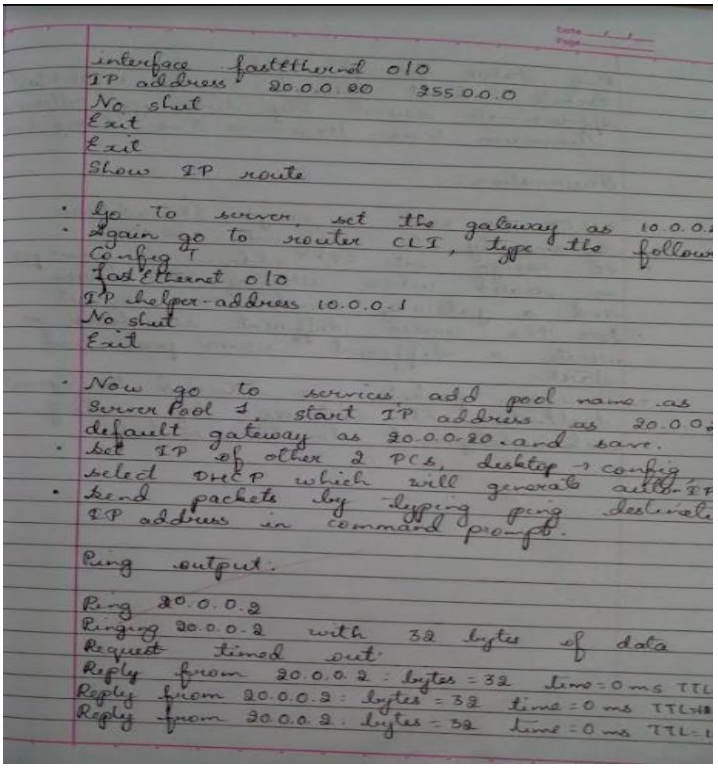


4.2 Configure DHCP within a LAN and ...

Topology :

- Connect a router, switch to the ... setup, connect router to both switches
- Set the IP address for server, ... IP address of first 3 PCs through ...
- Set the router IP address with ... following comands statistically:

Enable
Config T
fastEthernet 4/0
IP address 10.0.0.20 255.0.0.0
No shut
Exit



interface fastEthernet 0/0
IP address 20.0.0.20 255.0.0.0
No shut
Exit
Exit
Show IP route

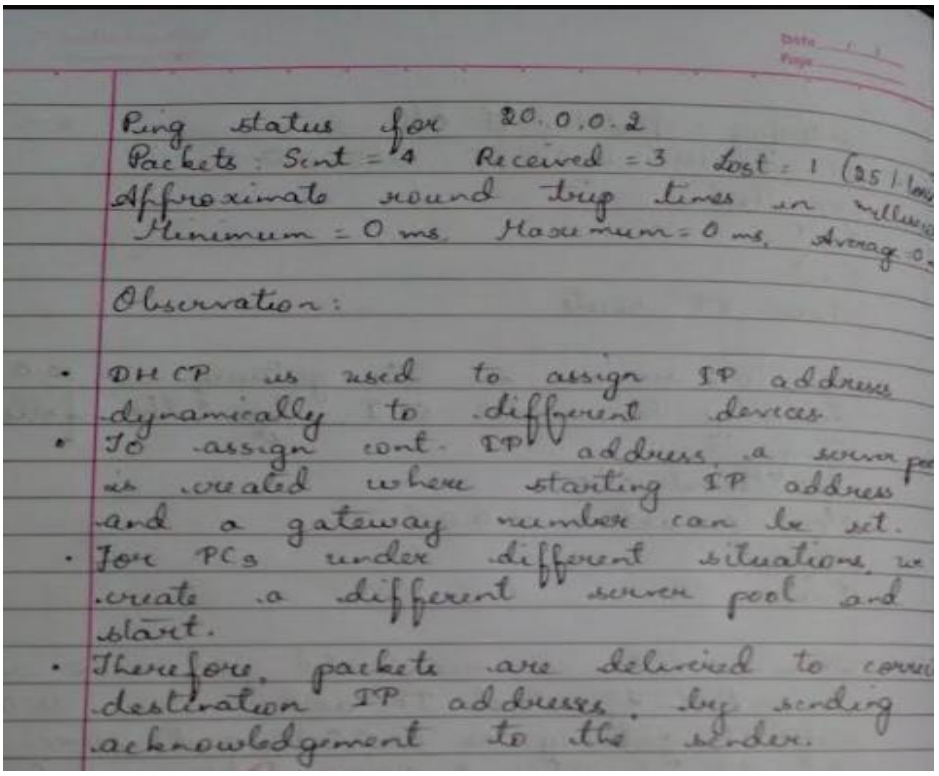
- Go to server, set the gateway as 10.0.0.
- Again go to router CLI, type the follow...

Config T
fastEthernet 0/0
IP helper-address 10.0.0.1
No shut
Exit

- Now go to services, add pool name as Server Pool 1, start IP address as 20.0.0. default gateway as 20.0.0.20 and save.
- Set IP of other 2 PCs, desktop → config, select DHCP which will generate auto IP
- Send packets by typing ping destinat... IP address in command prompt.

Ping output:

Ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data
Request timed out.
Reply from 20.0.0.2 : bytes = 32 time = 0 ms TTL
Reply from 20.0.0.2 : bytes = 32 time = 0 ms TTL=
Reply from 20.0.0.2 : bytes = 32 time = 0 ms TTL=

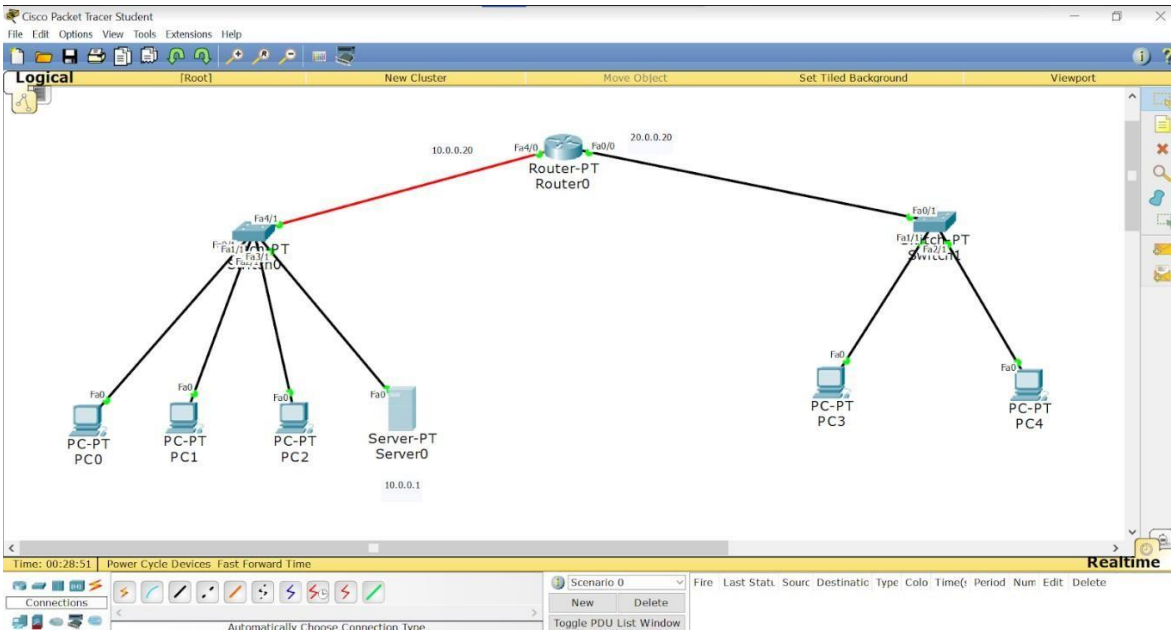


Ping status for 20.0.0.2
Packets: Sent = 4 Received = 3 Lost = 1 (25 l
Approximate round trip times in
Minimum = 0 ms, Maximum = 0 ms, Averag

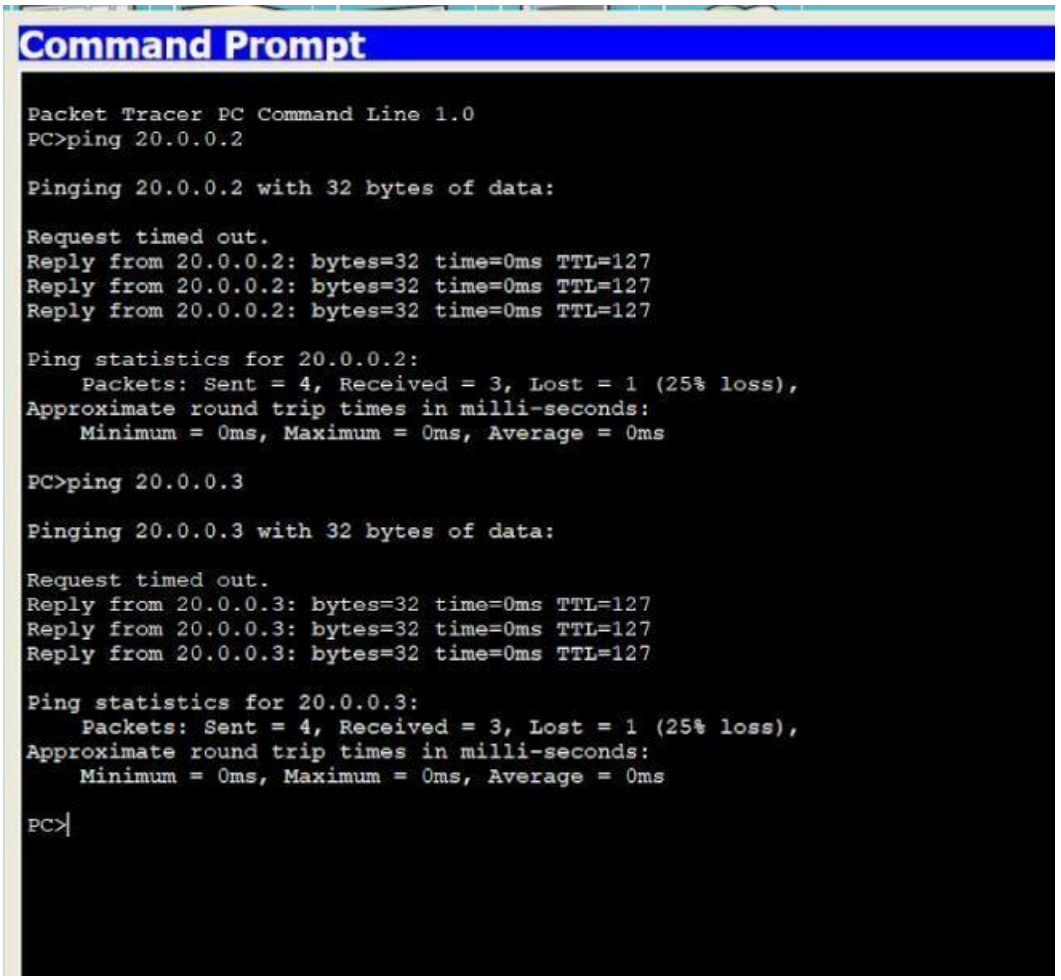
Observation:

- DHCP is used to assign IP address dynamically to different devices
- To assign cont. IP address, a serv is created where starting IP addres and a gateway number can be set.
- For PCs under different situations, w create a different server pool and start.
- Therefore, packets are delivered to corr destination IP address by sending acknowledgement to the sender.

Topology :



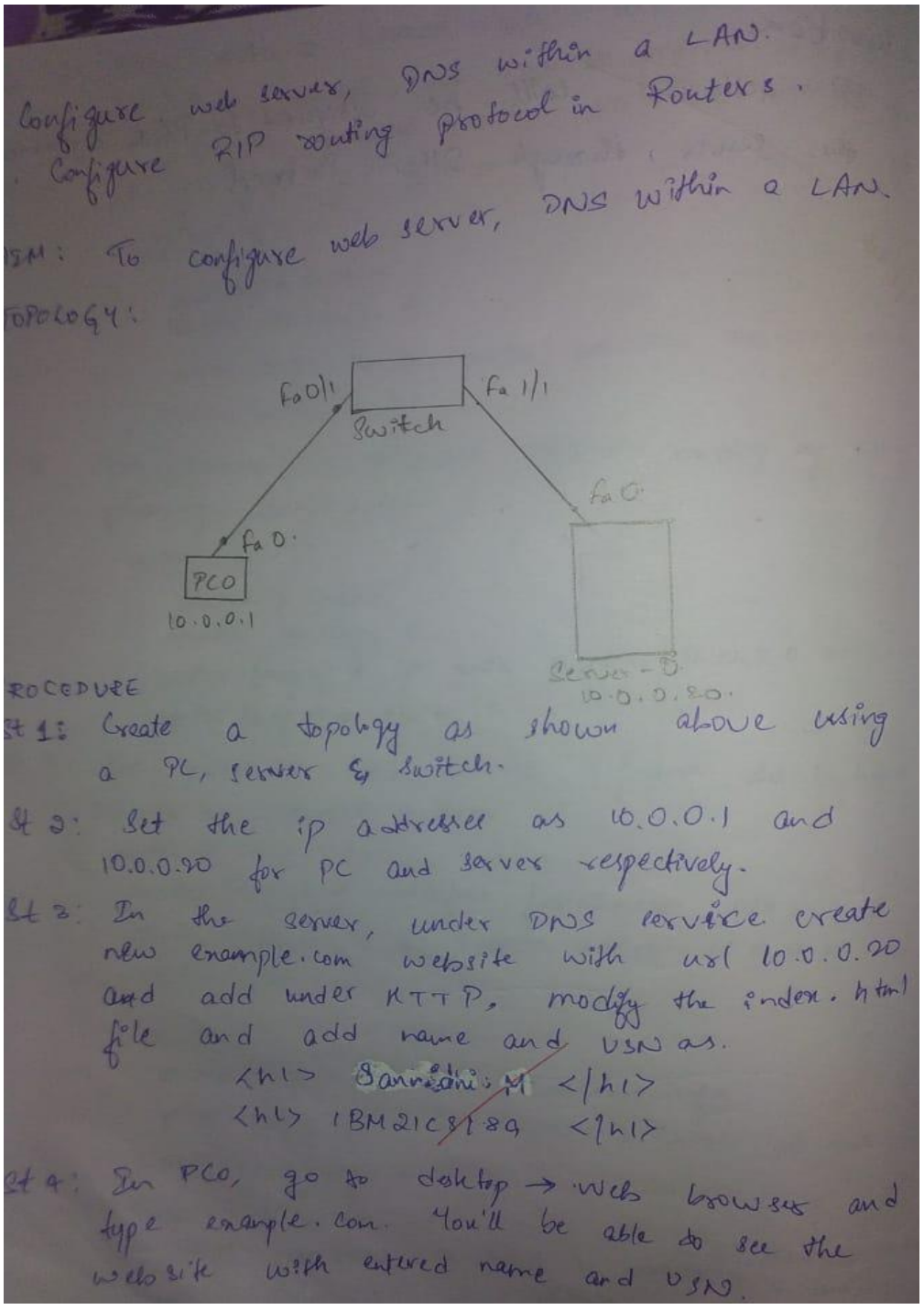
Output :


Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=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 = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

WEEK 5

Configure Web Server, DNS within a LAN.

Observation :



ESULT:

Web Browser

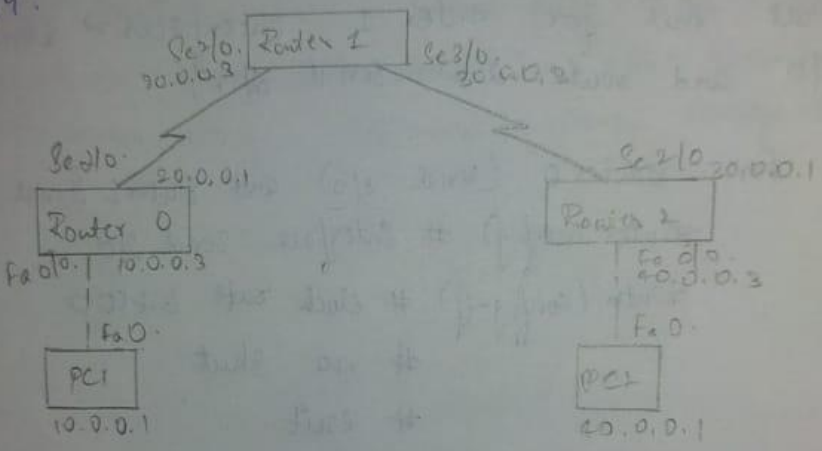
url http://example.com

cisco Packet Tracer

Sanmidhi M
1BM21CS189

IM: To configure RIP Routing protocol in Routers.

OPOLOGY:



ROCEDURE:

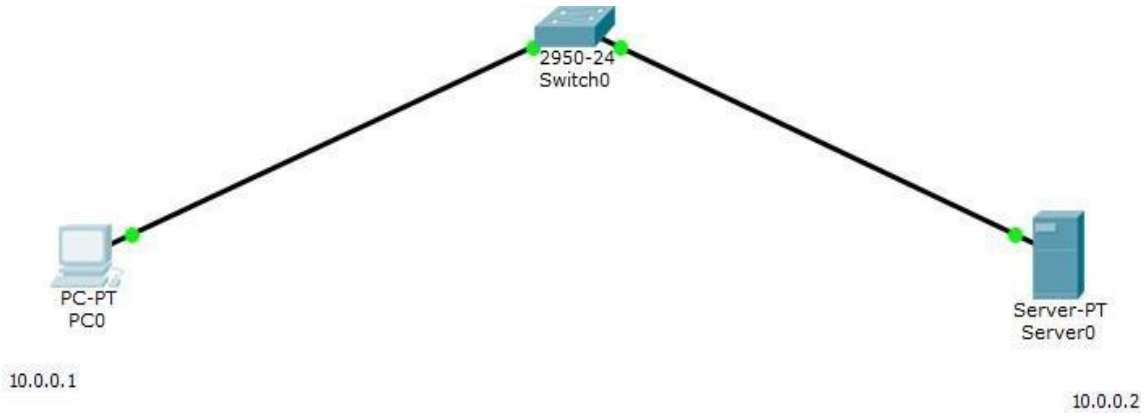
St 1: Create a topology as shown above using 2 PCs and 3 routers.

St 2: Configure the IP addresses of 2 PCs as 10.0.0.1 and 40.0.0.1 for PC1 & PC2 respectively and set the gateways as 10.0.0.3 & 40.0.0.3.

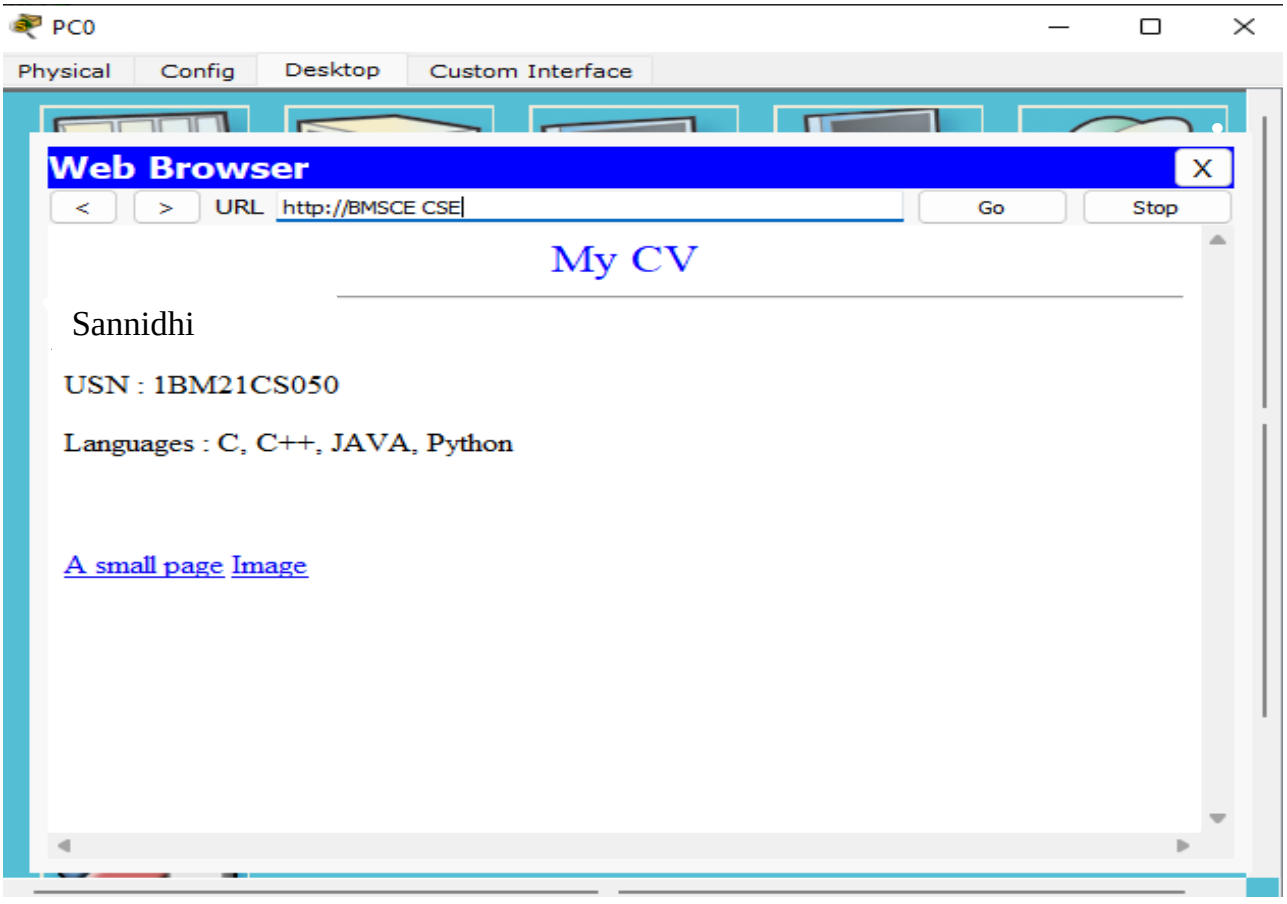
St 3: Plan the IPs to configure the routers. for Router 0

```
enable
#config t
# interface fastethernet 0/0
# ip address 10.0.0.3 255.0.0.0
# no shut
```

Topology



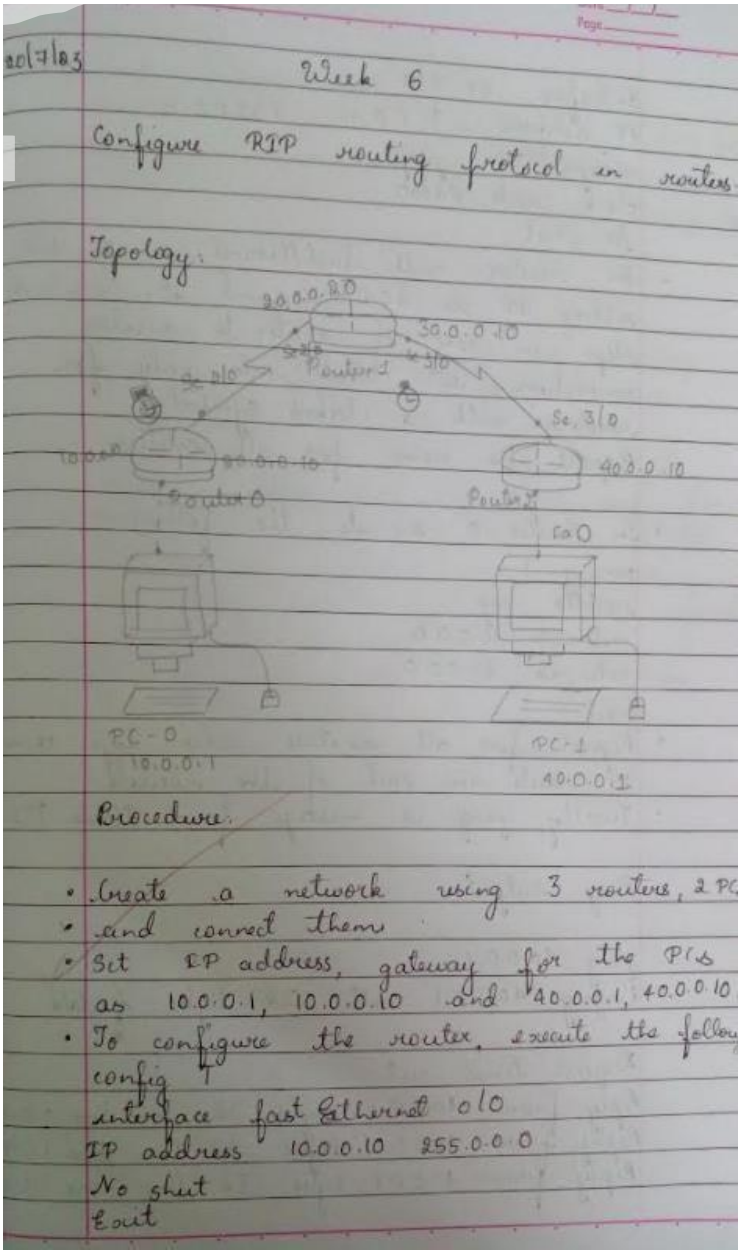
Output

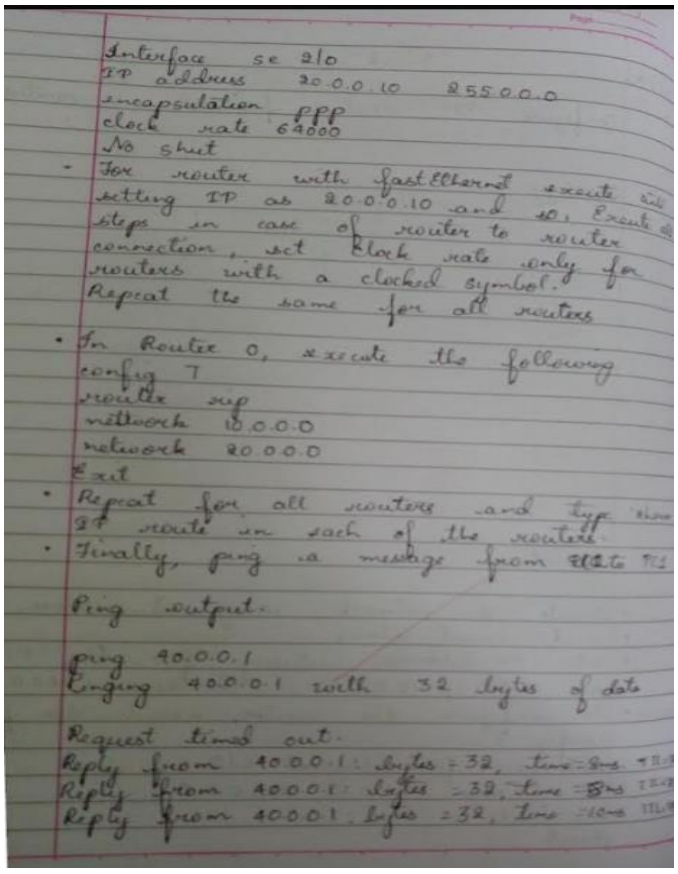


WEEK 6

Configure RIP routing Protocol in Routers

Observation :





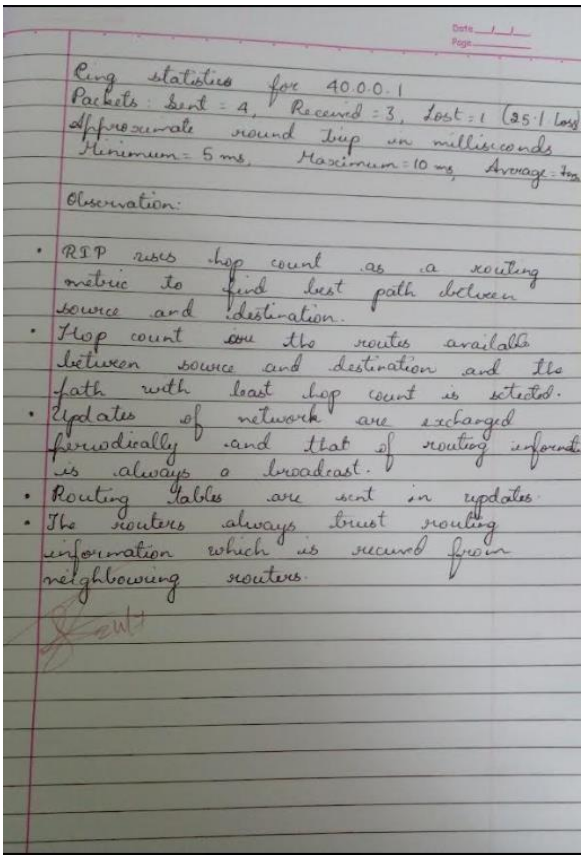
Interface se 2/0
IP address 20.0.0.10 255.0.0.0
encapsulation PPP
clock rate 64000
No shut

- For router with fastEthernet execute with setting IP as 20.0.0.10 and so, Execute the steps in case of router to router connection, set clock rate only for routers with a clocked symbol. Repeat the same for all routers
- In Router 0, execute the following
config T
router rip
network 10.0.0.0
network 20.0.0.0
Exit
- Repeat for all routers and type show IP route on each of the routers.
- Finally, ping a message from PC2 to PC1

Ping output.

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 = 8ms TTL
Reply from 40.0.0.1: bytes = 32, time = 5ms TTL
Reply from 40.0.0.1: bytes = 32, time = 10ms TTL



Date / /
Page

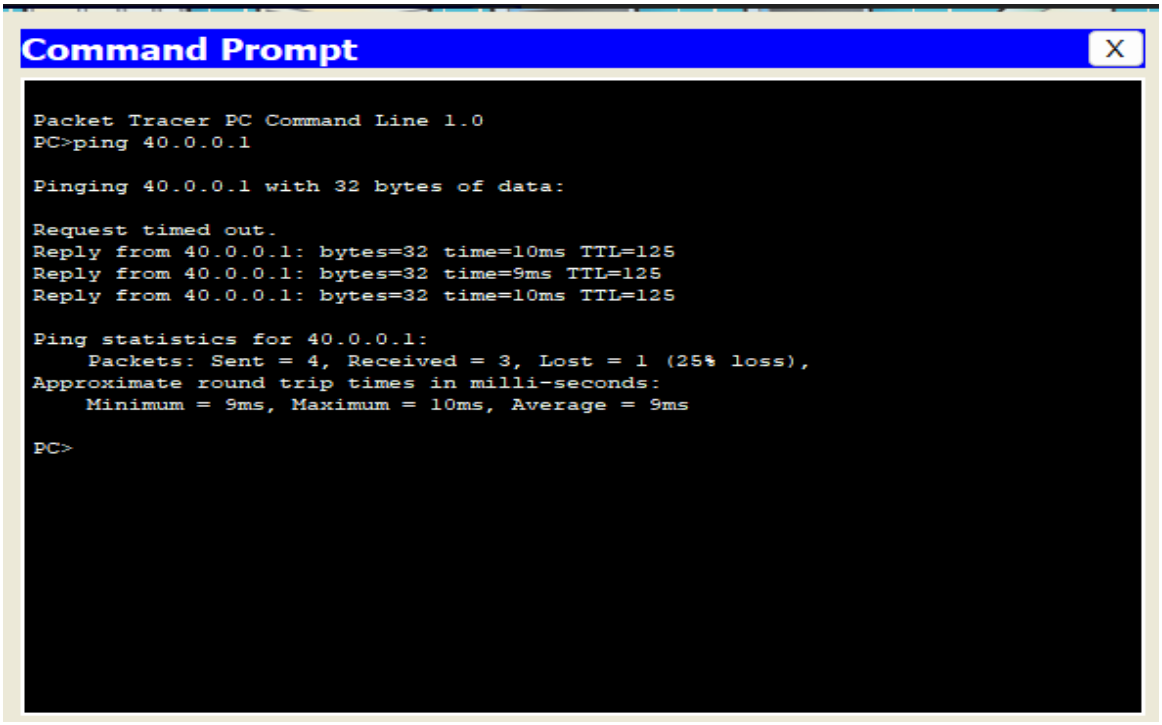
Ping statistics for 40.0.0.1
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip in milliseconds
Minimum = 5 ms, Maximum = 10 ms, Average = 7ms

Observation:

- RIP uses hop count as a routing metric to find best path between source and destination.
- Hop count is the routes available between source and destination and the path with least hop count is selected.
- Updates of network are exchanged periodically and that of routing information is always a broadcast.
- Routing Tables are sent in updates.
- The routers always trust routing information which is received from neighbouring routers.

Topology :

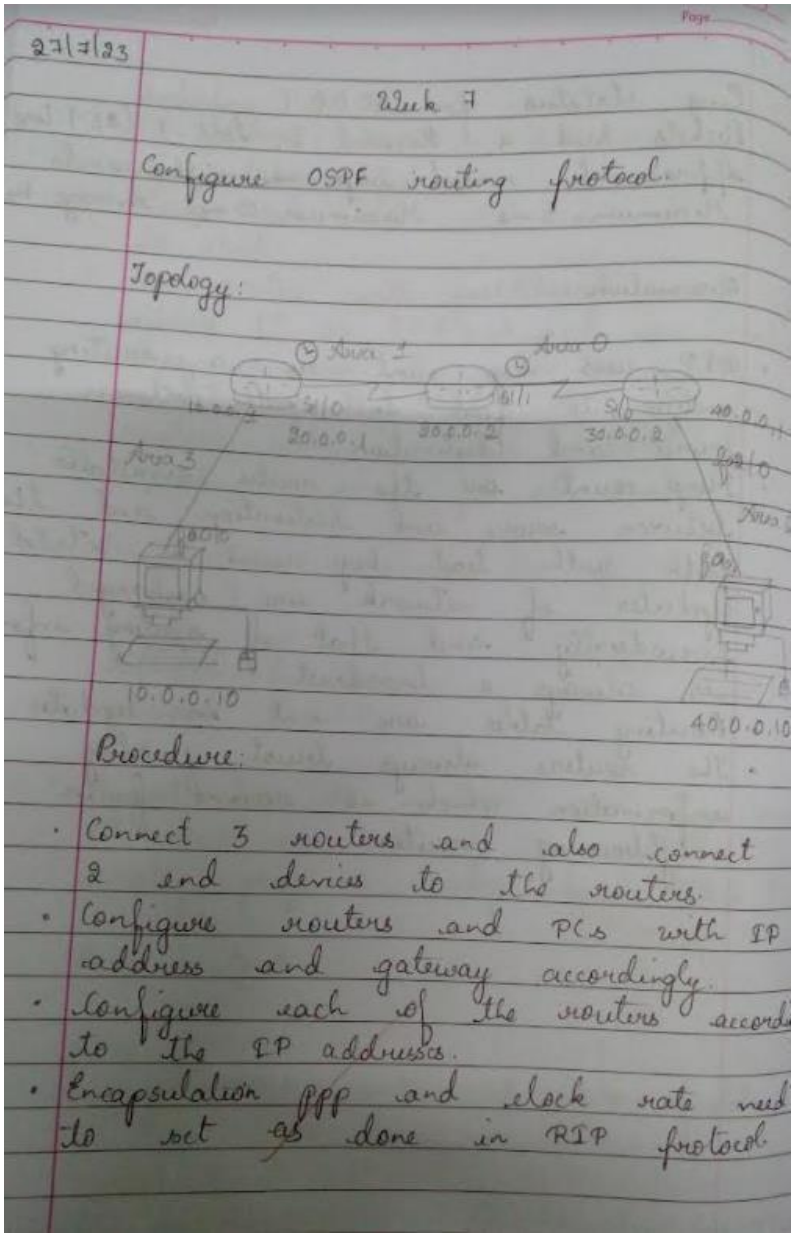
Output :

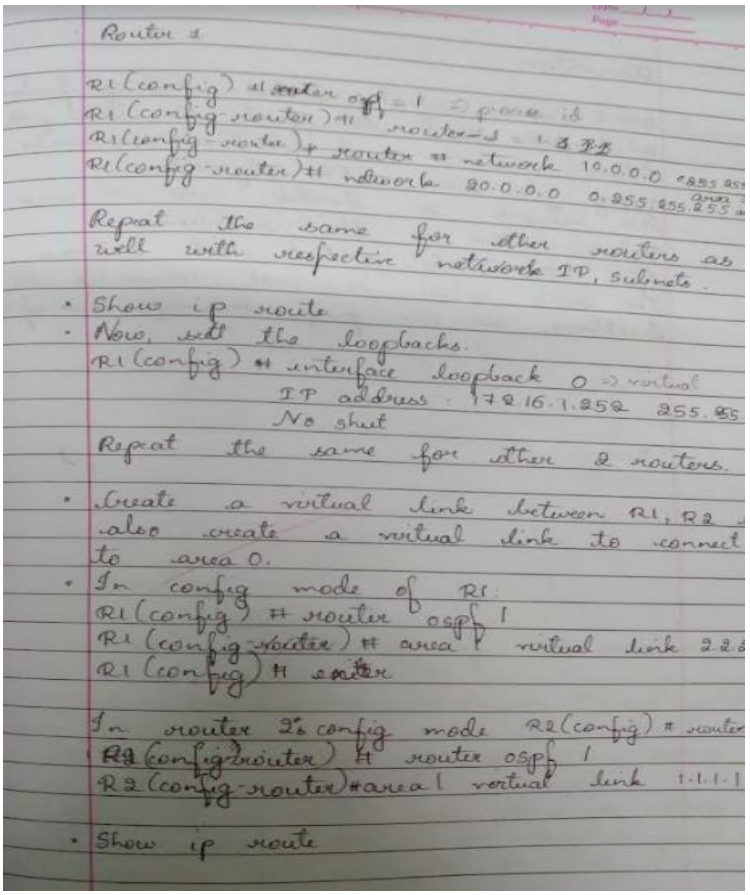


WEEK 7

Configure OSPF routing protocol

Observation :





Router 1

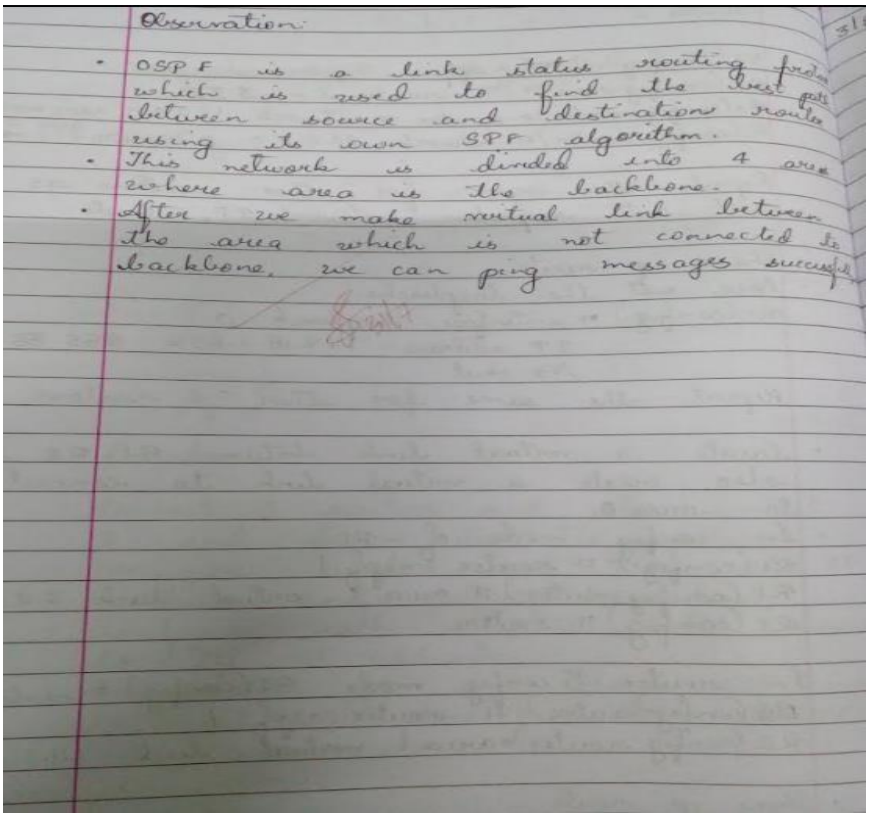
R1(config) #router ospf 1 => process id
R1(config-router)# router-id 1.1.1.1
R1(config-router)# router # network 10.0.0.0 0.255.255.255 area 0
R1(config-router)# network 20.0.0.0 0.255.255.255 area 0

Repeat the same for other routers as well with respective networks IP, Subnets.

- Show ip route
- Now, set the loopbacks.
  R1(config) # interface loopback 0 => virtual
  IP address 172.16.1.252 255.255.255.255
  No shut
  Repeat the same for other 2 routers.
- Create a virtual link between R1, R2 also create a virtual link to connect to area 0.
- In config mode of R1:
  R1(config) # router ospf 1
  R1(config-router) # area 1 virtual link 2.2.2.2
  R1(config) # router

In router 2's config mode R2(config) # router
R2(config)router) # router ospf 1
R2(config-router)#area 1 virtual link 1.1.1.1

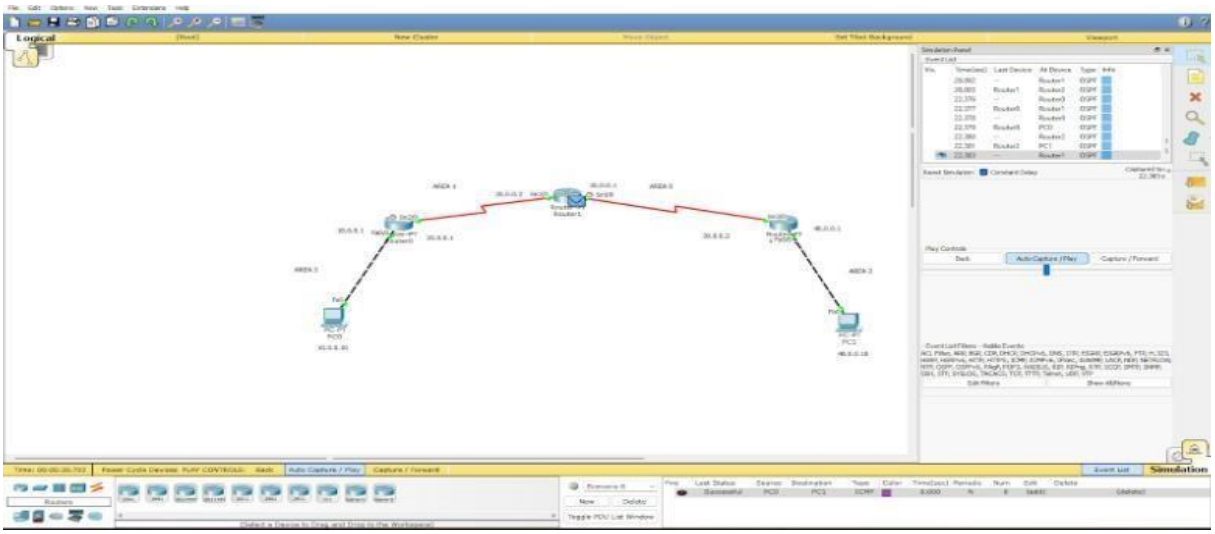
- Show ip route



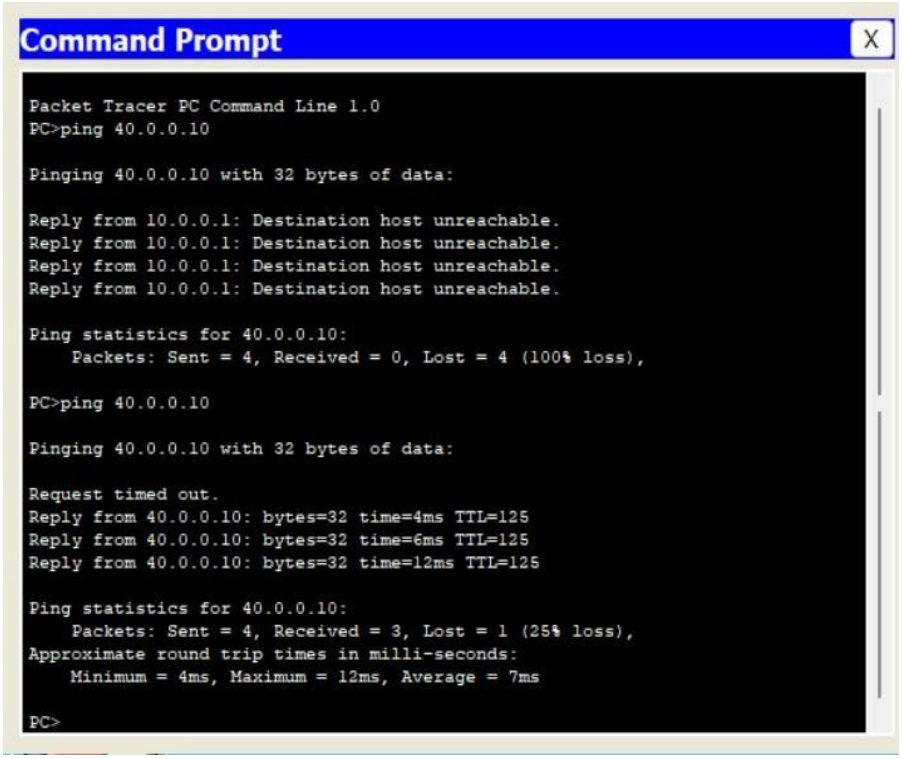
Observation:

- OSPF is a link status routing protocol which is used to find the best path between source and destination router using its own SPF algorithm.
- This network is divided into 4 areas where area is the backbone.
- After we make virtual link between the area which is not connected to backbone, we can ping messages successfully

Topology :



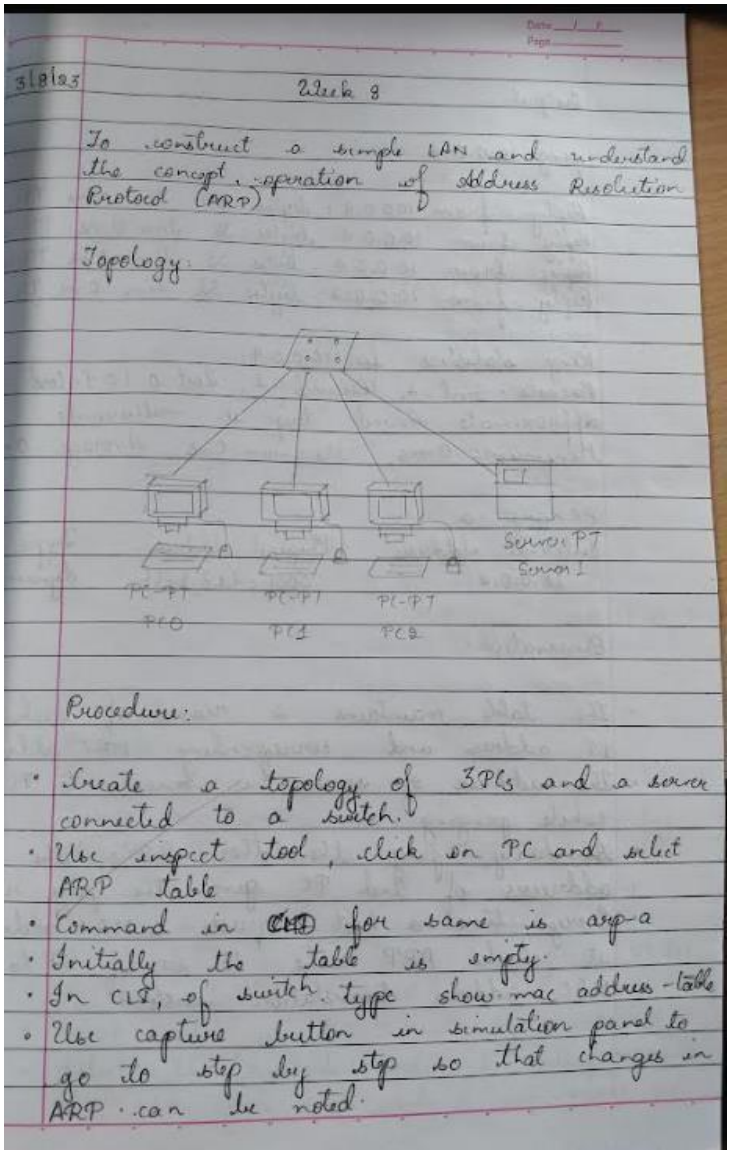
Output :



WEEK 8

Construct a simple LAN and to understand ARP

Observation :



3/8/23

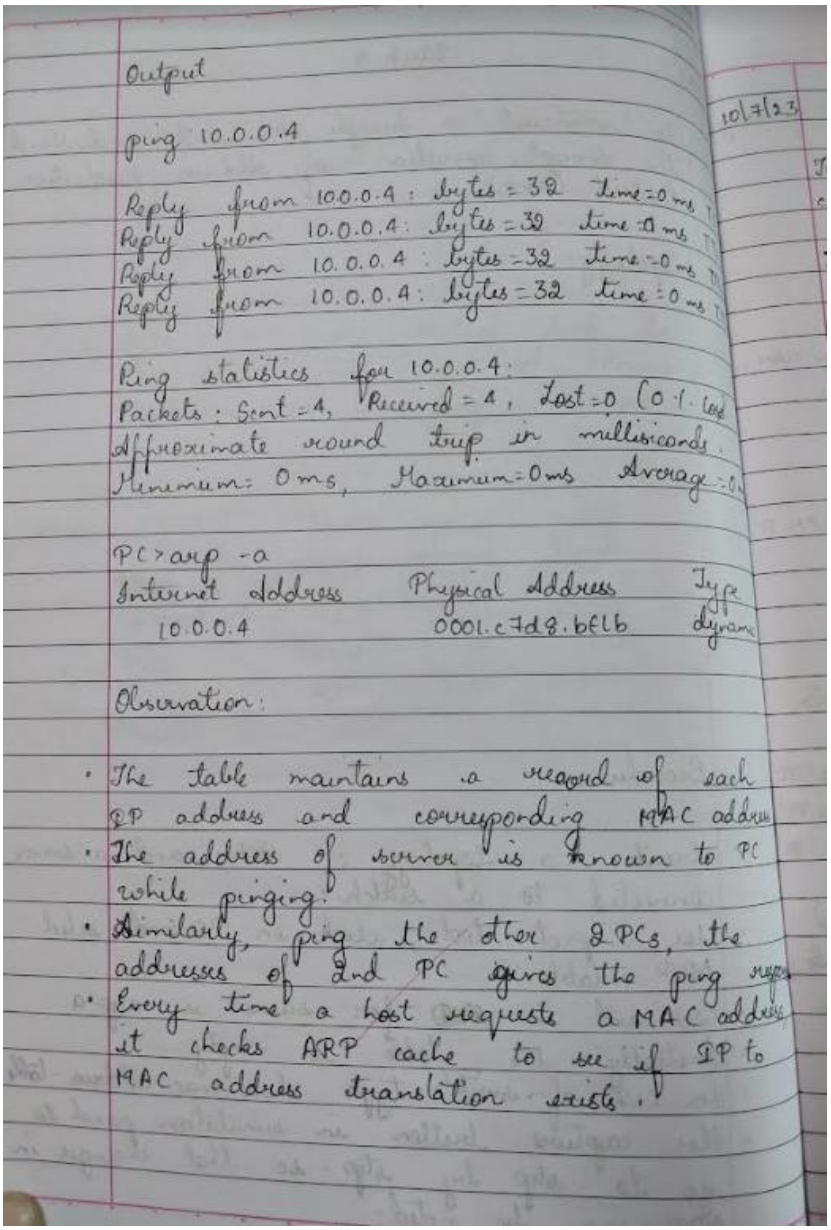
Week 8

To construct a simple LAN and understand the concept, operation of Address Resolution Protocol (ARP)

Topology:

Procedure:

- Create a topology of 3PCs and a server connected to a switch.
- Use inspect tool, click on PC and select ARP table
- Command in CLI for same is arp-a
- Initially the table is empty.
- In CLI, of switch type show mac address-table
- Use capture button in simulation panel to go to step by step so that changes in ARP can be noted.



Output

ping 10.0.0.4

Reply from 10.0.0.4 : bytes = 32 time = 0 ms
Reply from 10.0.0.4 : bytes = 32 time = 0 ms
Reply from 10.0.0.4 : bytes = 32 time = 0 ms
Reply from 10.0.0.4 : bytes = 32 time = 0 ms

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

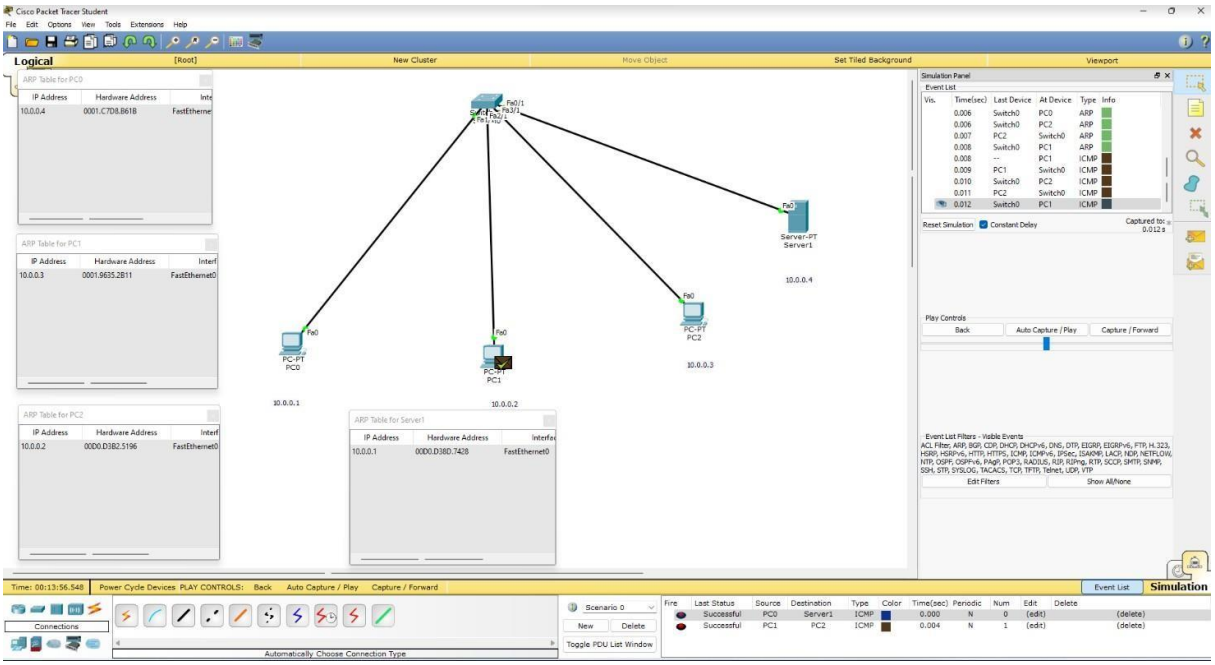
PC > arp -a

Internet Address	Physical Address	Type
10.0.0.4	0001.c7d8.bf1b	dynamic

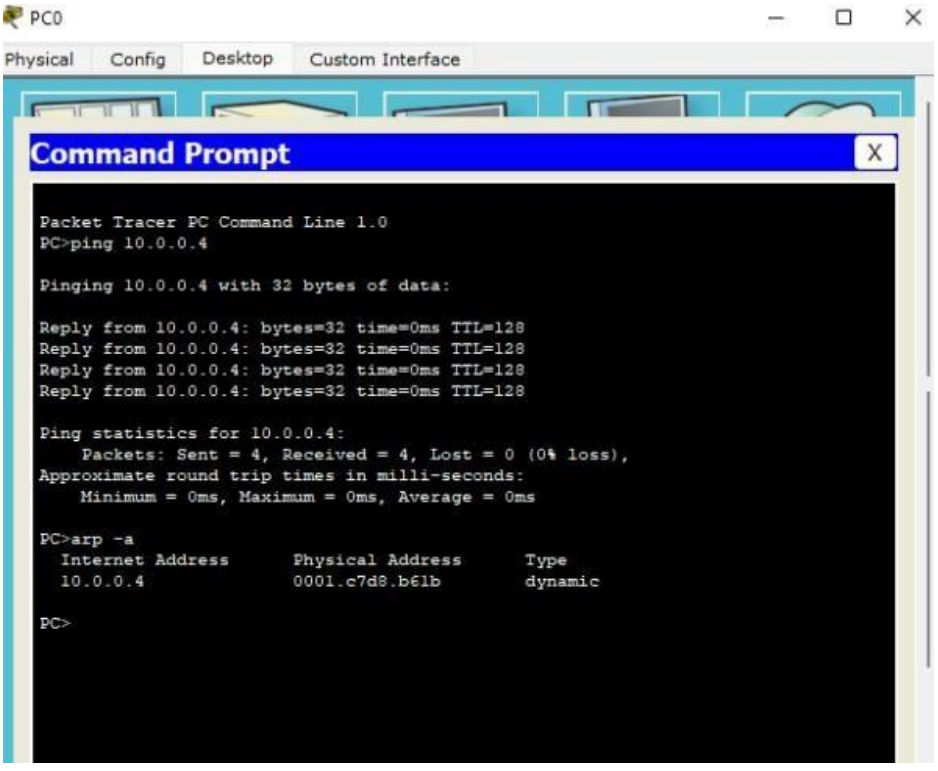
Observation :

- The table maintains a record of each IP address and corresponding MAC address
- The address of server is known to PC while pinging.
- Similarly, ping the other 2 PCs, the addresses of 2nd PC gives the ping reply
- Every time a host requests a MAC address it checks ARP cache to see if IP to MAC address translation exists.

Topology :



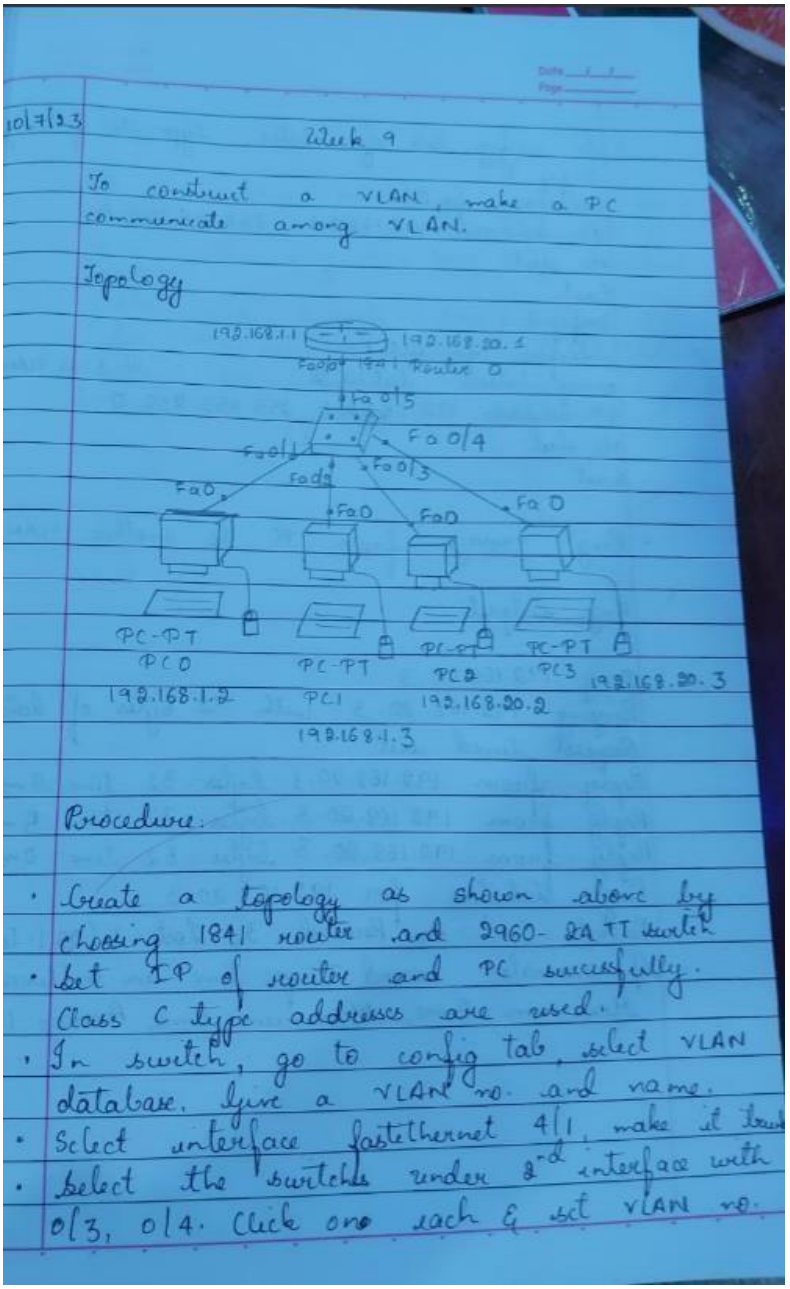
Output :

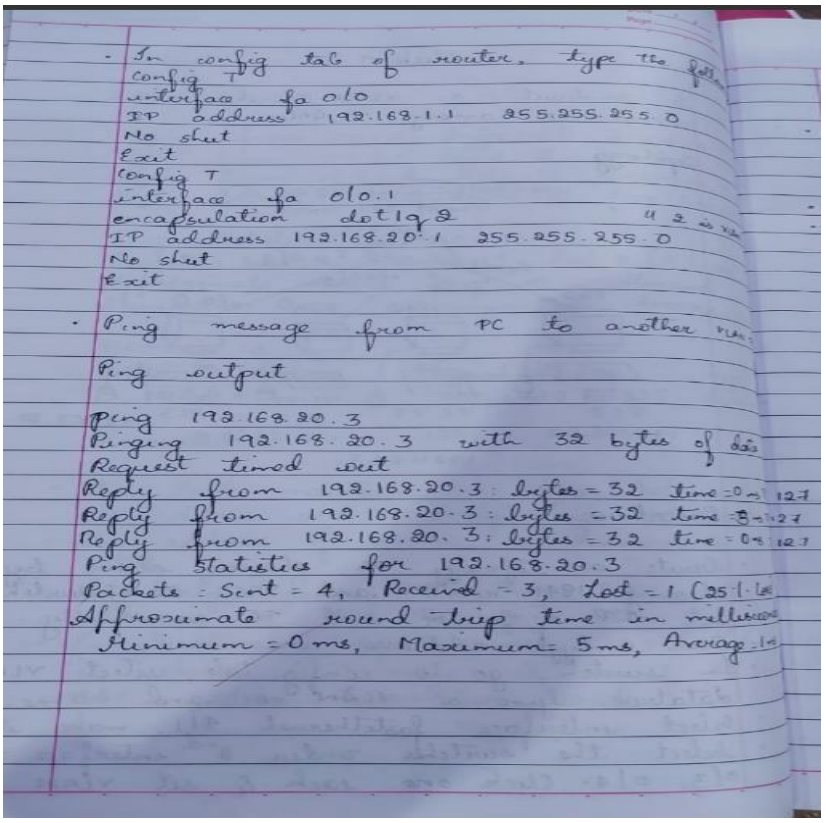


WEEK 9

To construct a VLAN and make a pc communicate among VLAN.

OBSERVATION:



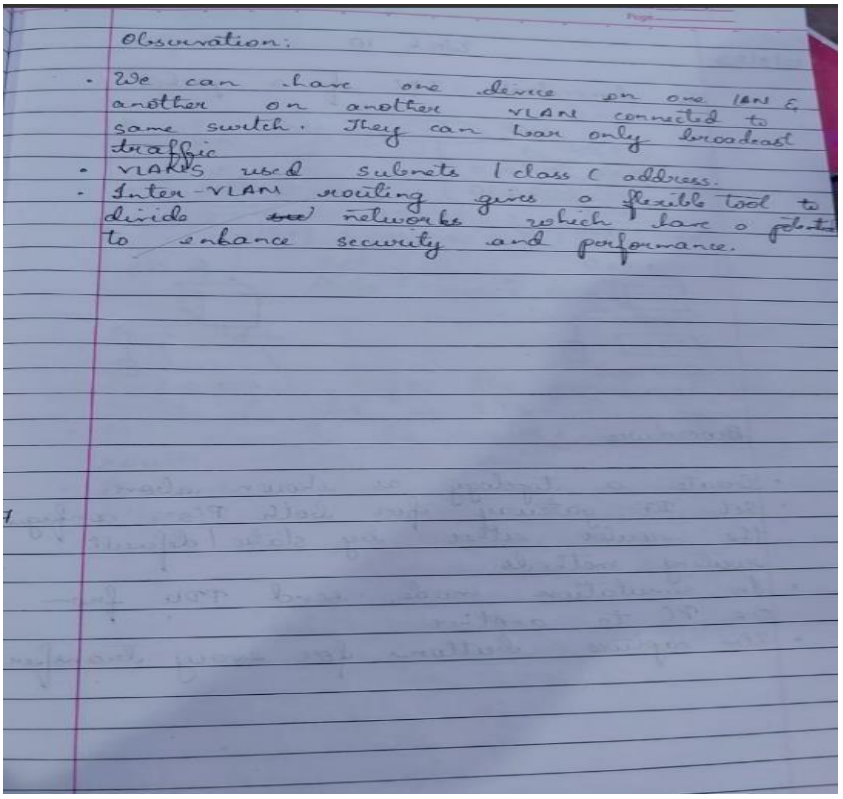


- In config tab of router, type the following
  Config T
  interface fa 0/0
  IP address 192.168.1.1 255.255.255.0
  No shut
  Exit
  Config T
  interface fa 0/0.1
  encapsulation dot1q 2 (2 is VLAN)
  IP address 192.168.20.1 255.255.255.0
  No shut
  Exit

- Ping message from PC to another VLAN:

  Ping output

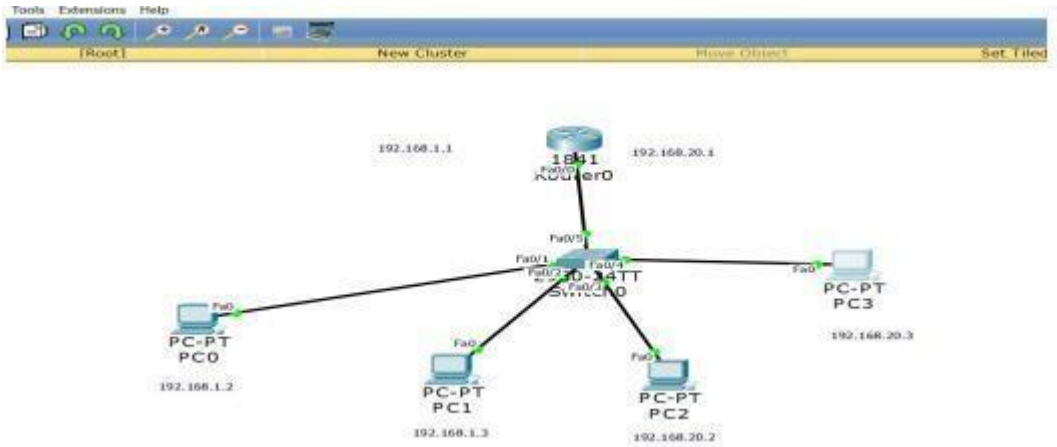
  Ping 192.168.20.3
  Pinging 192.168.20.3 with 32 bytes of data:
  Request timed out
  Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
  Reply from 192.168.20.3: bytes=32 time=8ms TTL=127
  Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
  Ping statistics for 192.168.20.3
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss)
  Approximate round trip time in milliseconds:
  Minimum = 0ms, Maximum = 5ms, Average = 1ms



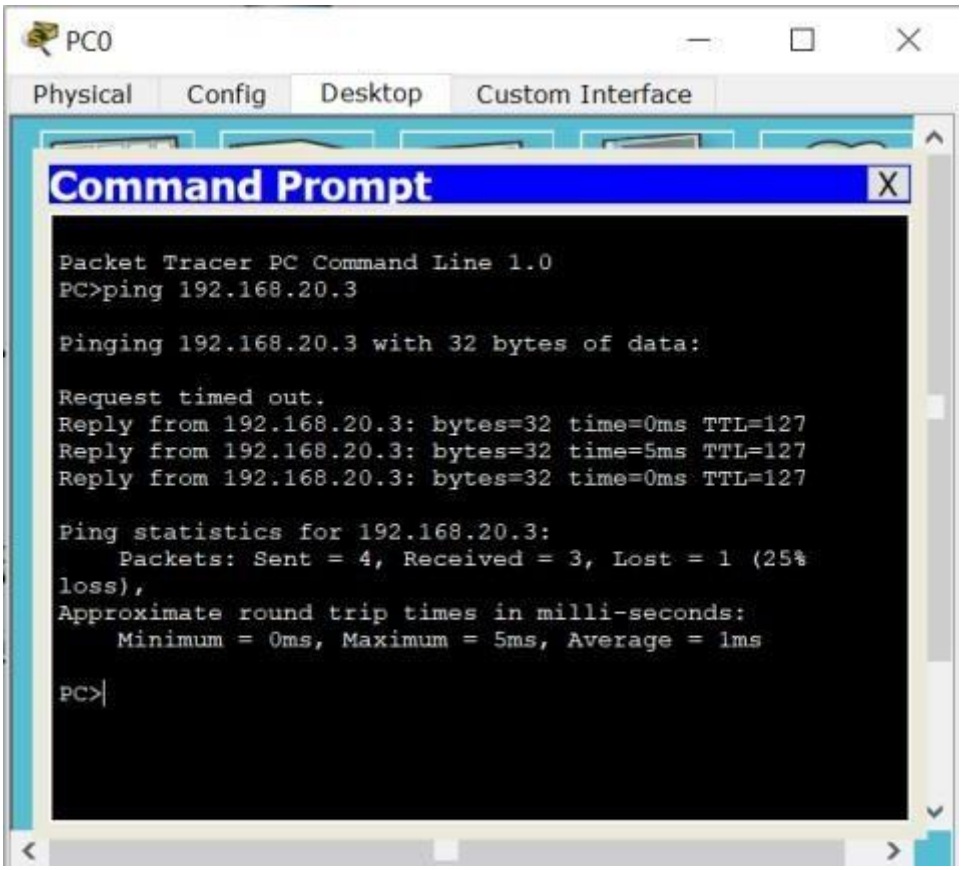
Observation:

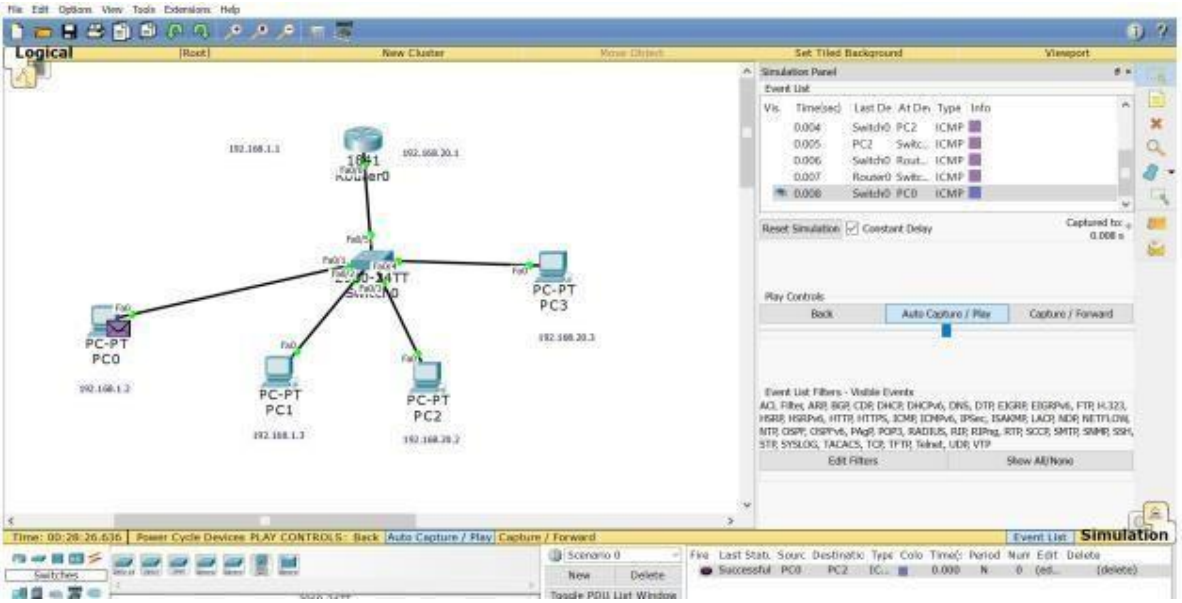
- We can have one device on one VLAN & another on another VLAN connected to same switch. They can have only broadcast traffic
- VLANs used subnets / class C address.
- Inter-VLAN routing gives a flexible tool to divide ~~see~~ networks which have a potential to enhance security and performance.

Topology :



Output :

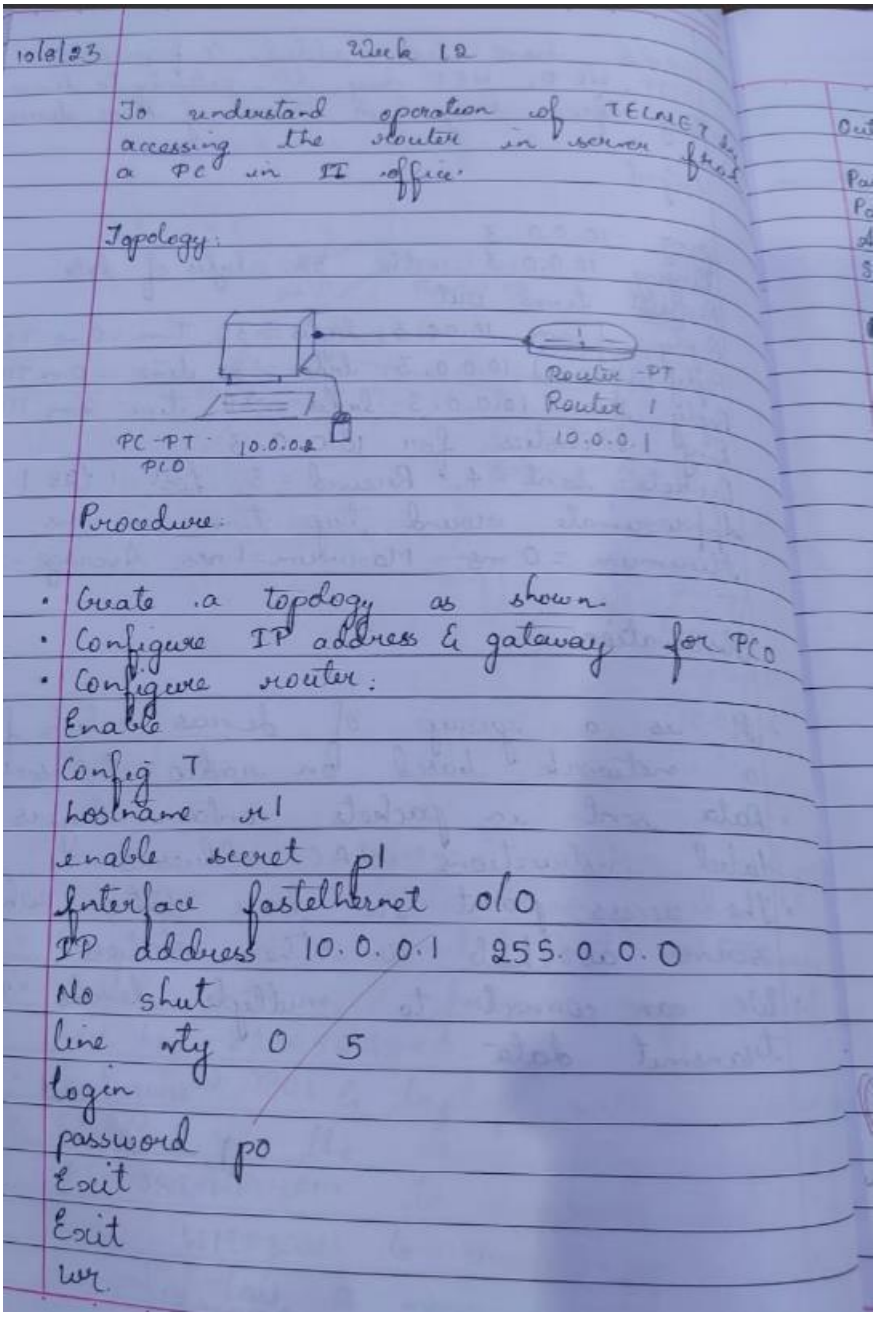

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25%
loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms
PC>

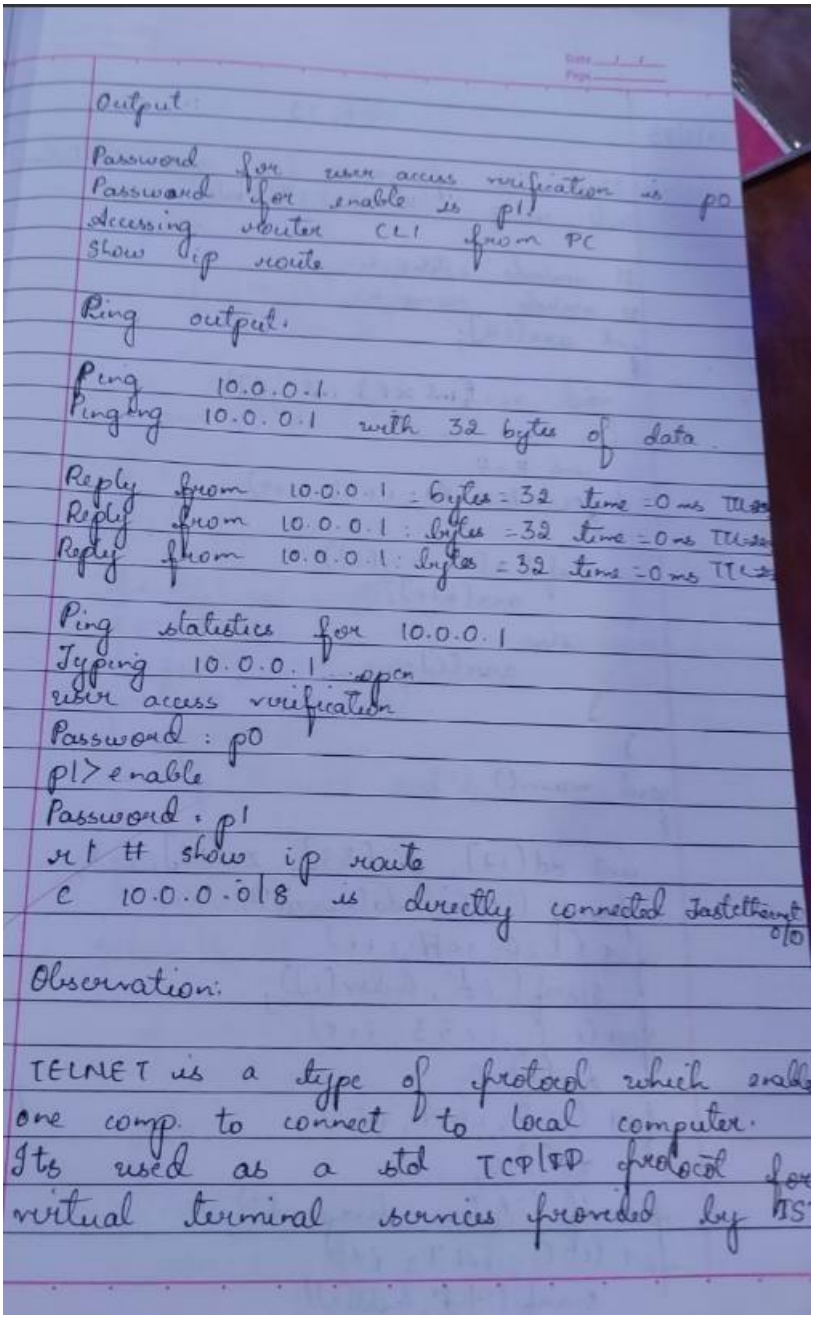

Logical
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
Simulation

Week 10

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

Observation:





Output:

Password for user access verification is p0
Password for enable is p1
Accessing router CLI from PC
show ip route

Ping output:

Ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data

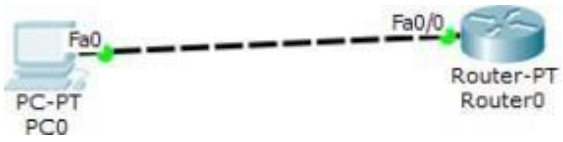
Reply from 10.0.0.1 : bytes = 32 time = 0 ms TTL=
Reply from 10.0.0.1 : bytes = 32 time = 0 ms TTL=
Reply from 10.0.0.1 : bytes = 32 time = 0 ms TTL=

Ping statistics for 10.0.0.1
Typing 10.0.0.1 ...open
user access verification
Password : p0
p1>enable
Password : p1
r1 # show ip route
C 10.0.0.0/8 is directly connected FastEthernet 0/0

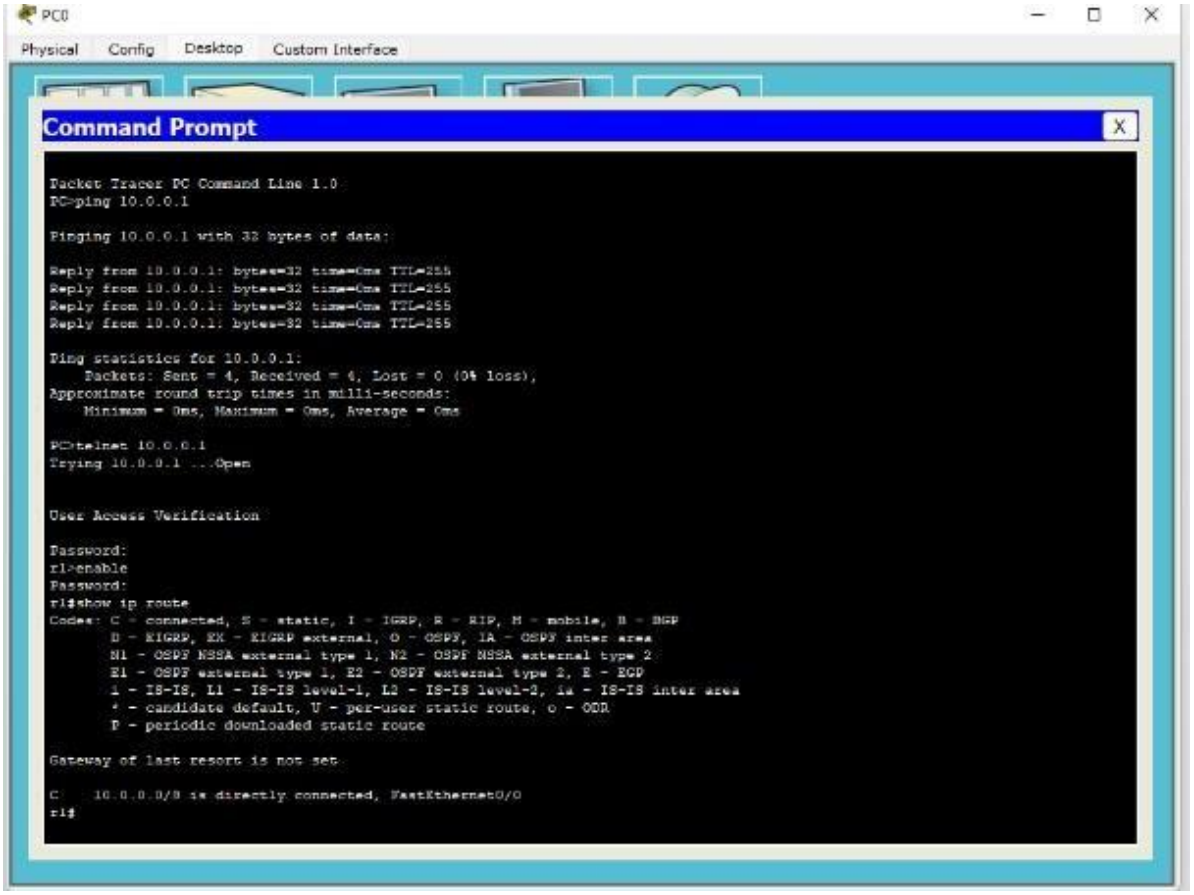
Observation:

TELNET is a type of protocol which enable one comp. to connect to local computer.
Its used as a std TCP/IP protocol for virtual terminal services provided by IS

Topology:



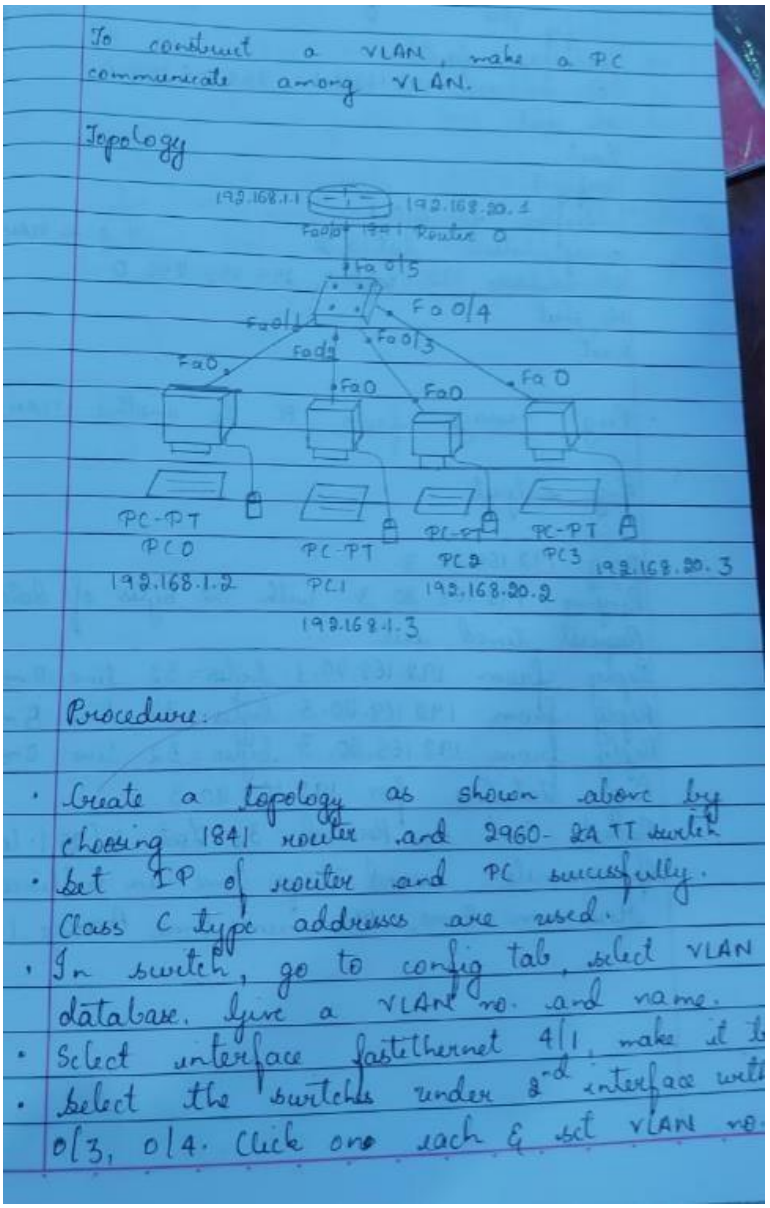
Output :

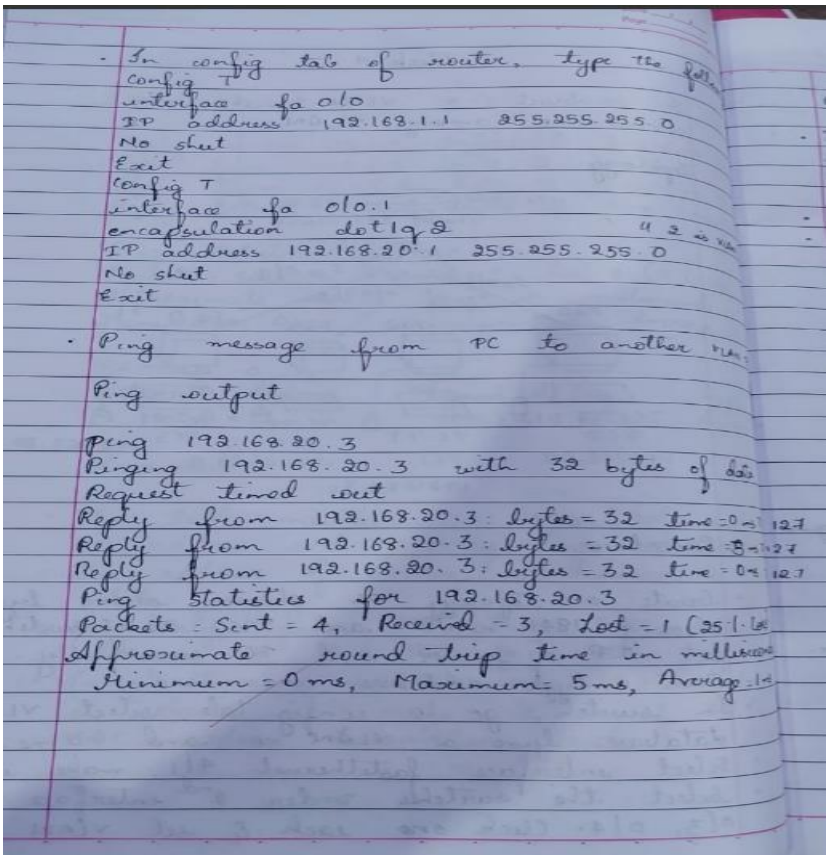

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
r1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
r1#

Week 11

To construct a VLAN and make the PC's communicate among a VLAN

Observation:



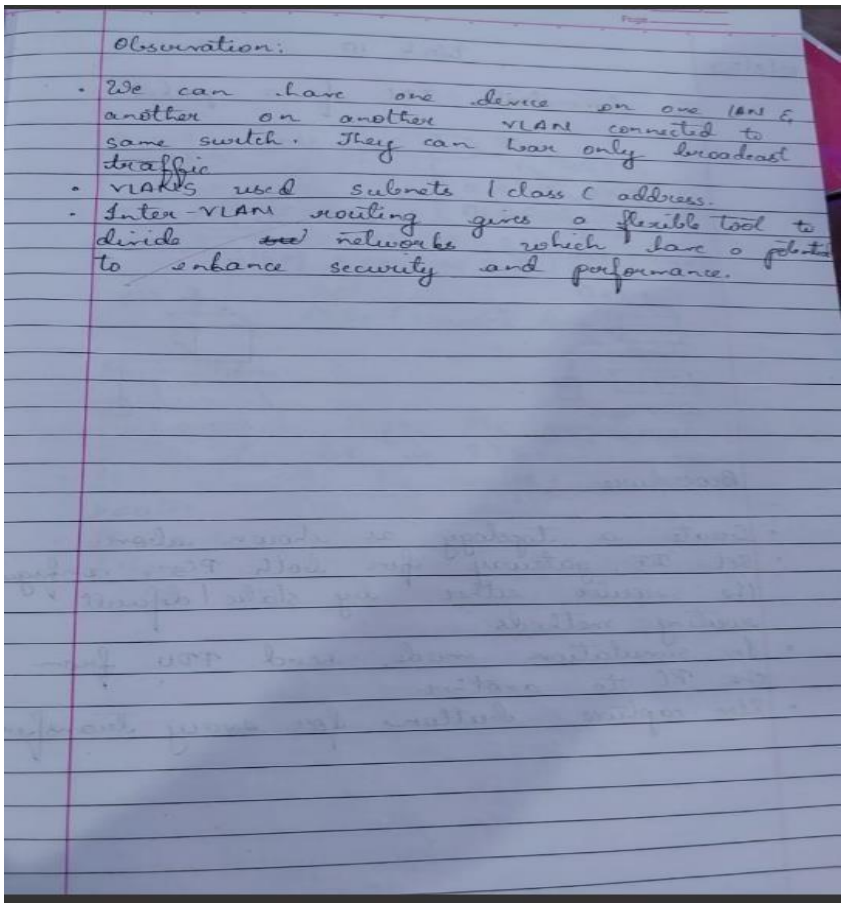


- In config tab of router, type the following
config T
interface fa 0/0
IP address 192.168.1.1 255.255.255.0
No shut
Exit
config T
interface fa 0/0.1
encapsulation dot1q 2
IP address 192.168.20.1 255.255.255.0
No shut
Exit

- Ping message from PC to another VLAN

Ping output

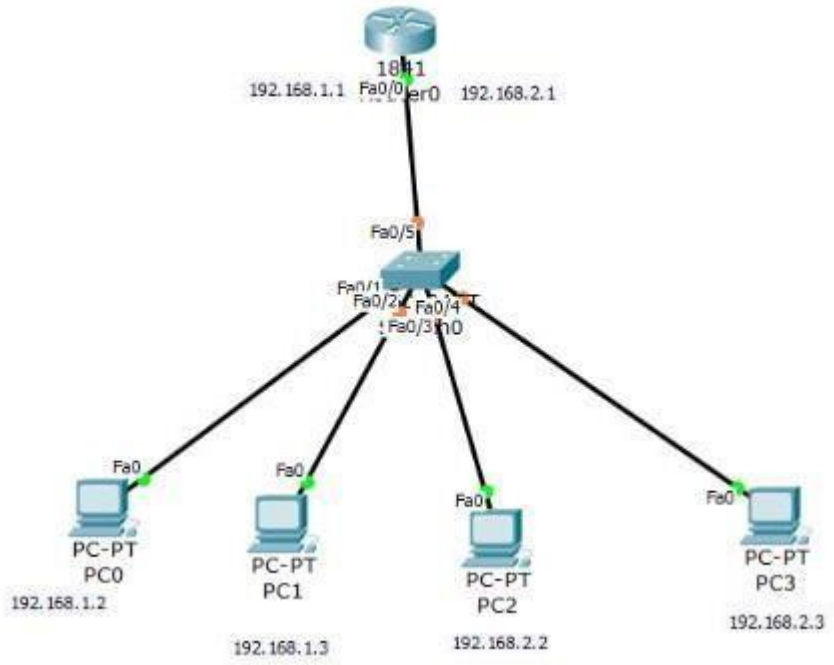
Ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data
Request timed out
Reply from 192.168.20.3: bytes = 32 time = 0ms TTL=127
Reply from 192.168.20.3: bytes = 32 time = 0ms TTL=127
Reply from 192.168.20.3: bytes = 32 time = 0ms TTL=127
Ping statistics for 192.168.20.3
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip time in milliseconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms



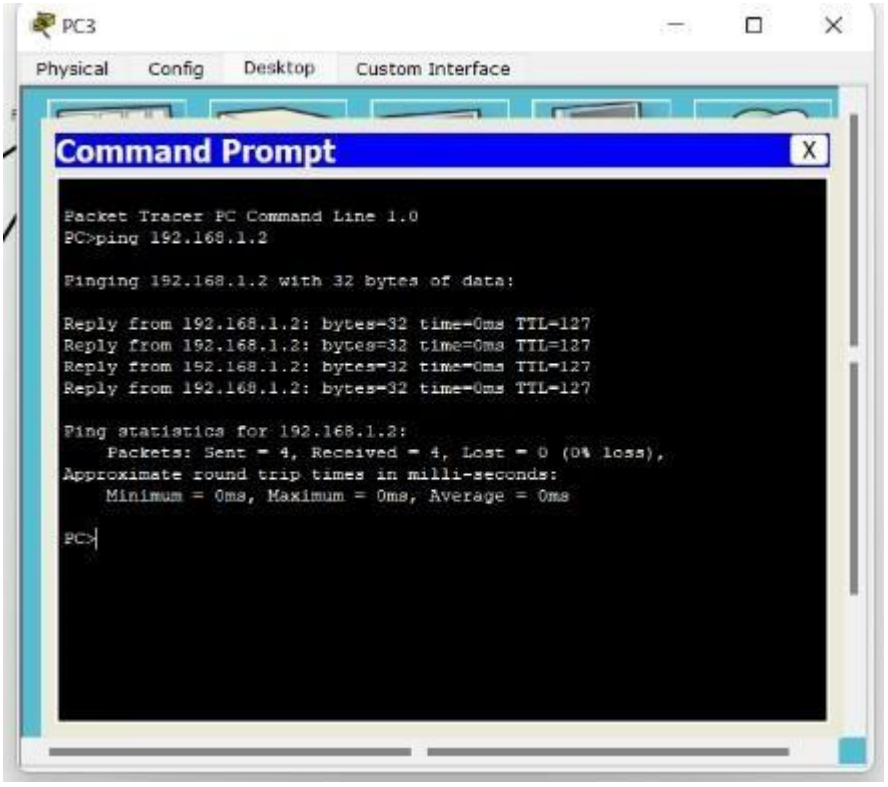
Observation:

- We can have one device on one VLAN & another on another VLAN connected to same switch. They can bear only broadcast traffic
- VLANs used subnets I class C address.
- Inter-VLAN routing gives a flexible tool to divide ~~see~~ networks which have a potential to enhance security and performance.

Topology :



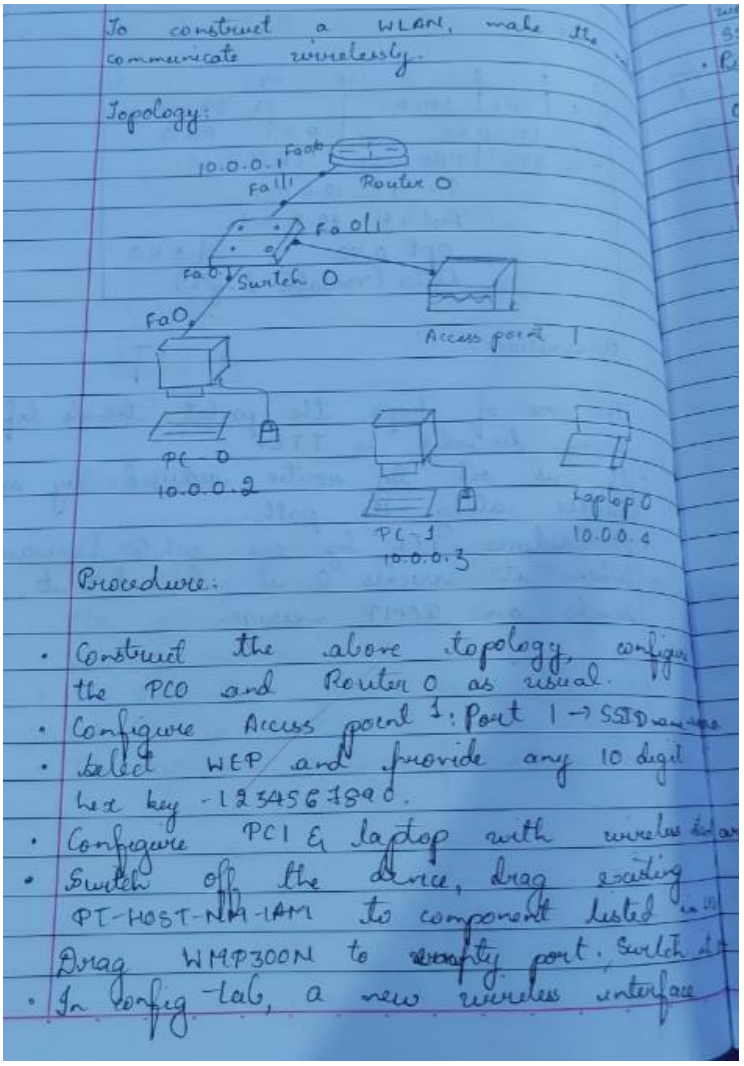
Output :

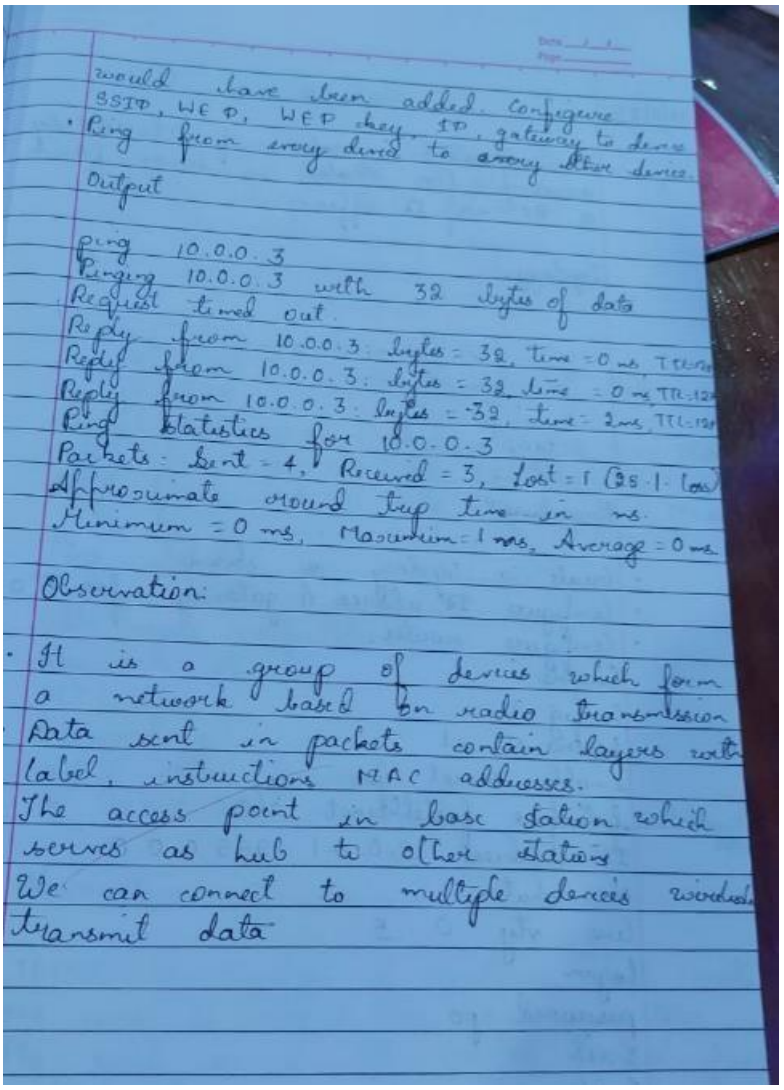


Week 12

To construct a WLAN and make the nodes communicate wirelessly.

Observation:





would have been added. Configure SSID, WEP, WEP key, IP, gateway to devices
• Ping from every host to every other devices.

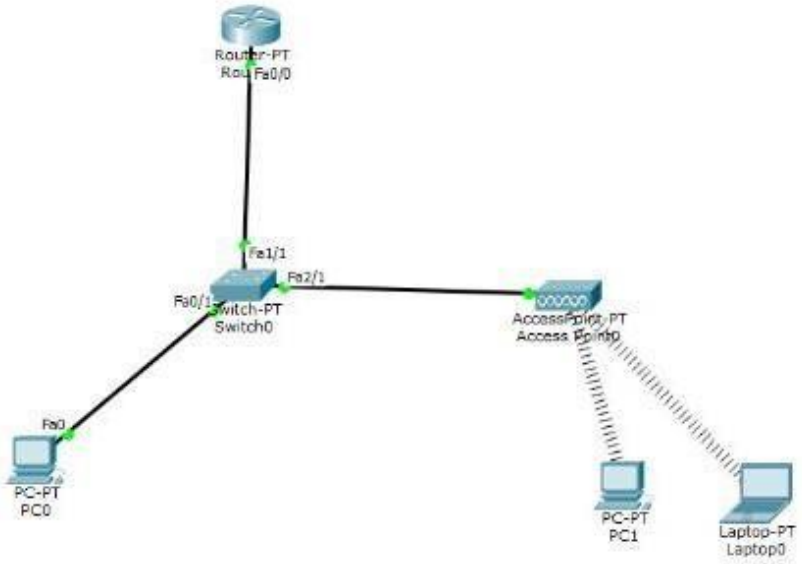
Output

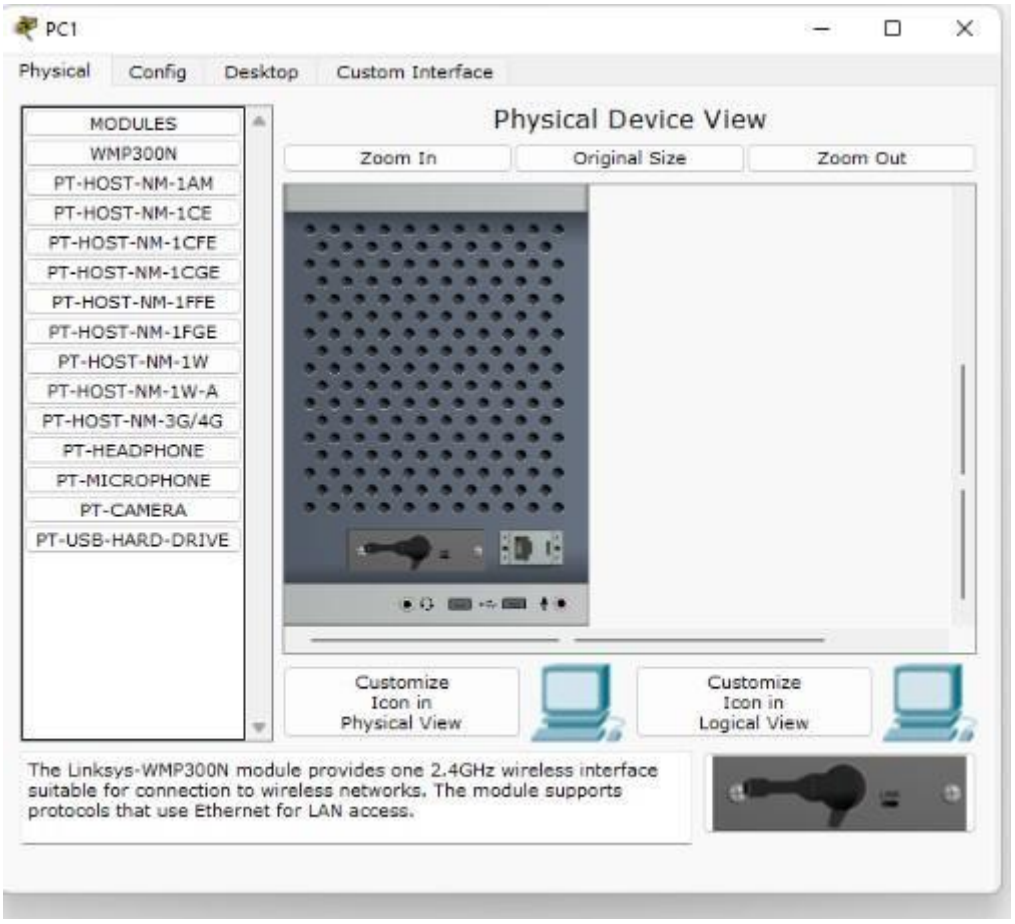
Ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data
Request timed out.
Reply from 10.0.0.3: bytes = 32, time = 0 ms, TTL=
Reply from 10.0.0.3: bytes = 32, time = 0 ms, TTL=12
Reply from 10.0.0.3: bytes = 32, time = 2ms, TTL=12
Ping statistics for 10.0.0.3
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip time in ms
Minimum = 0 ms, Maximum = 1 ms, Average = 0 ms

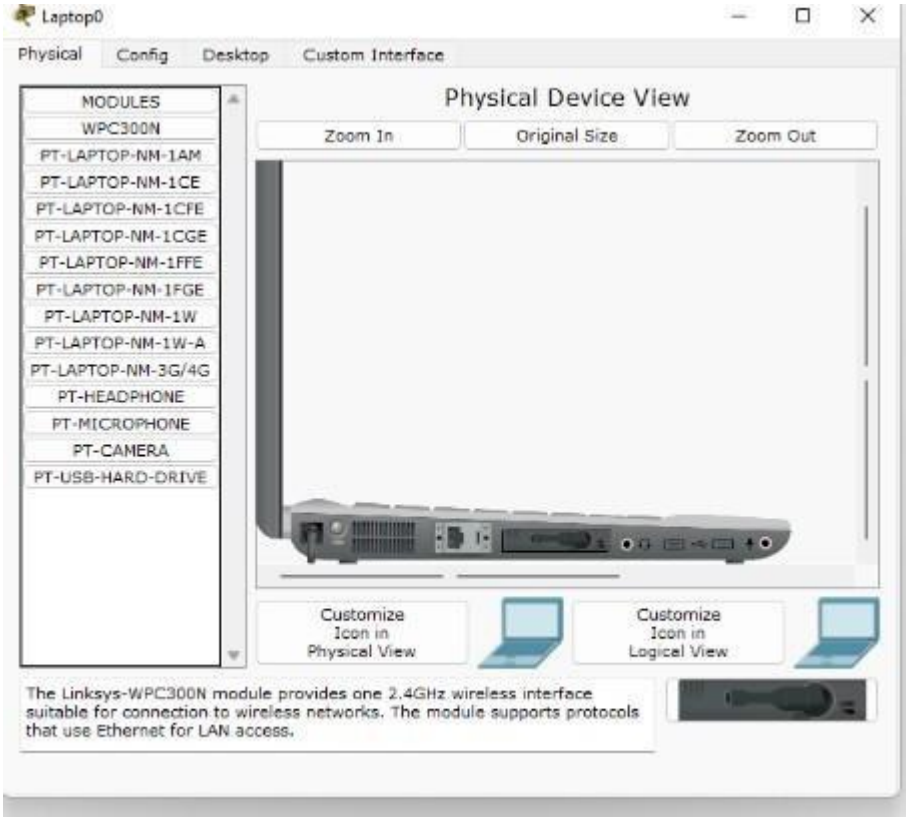
Observation:

• It is a group of devices which form a network based on radio transmission. Data sent in packets contain layers with label, instructions MAC addresses.
The access point in base station which serves as hub to other stations
We can connect to multiple devices wirelessly transmit data

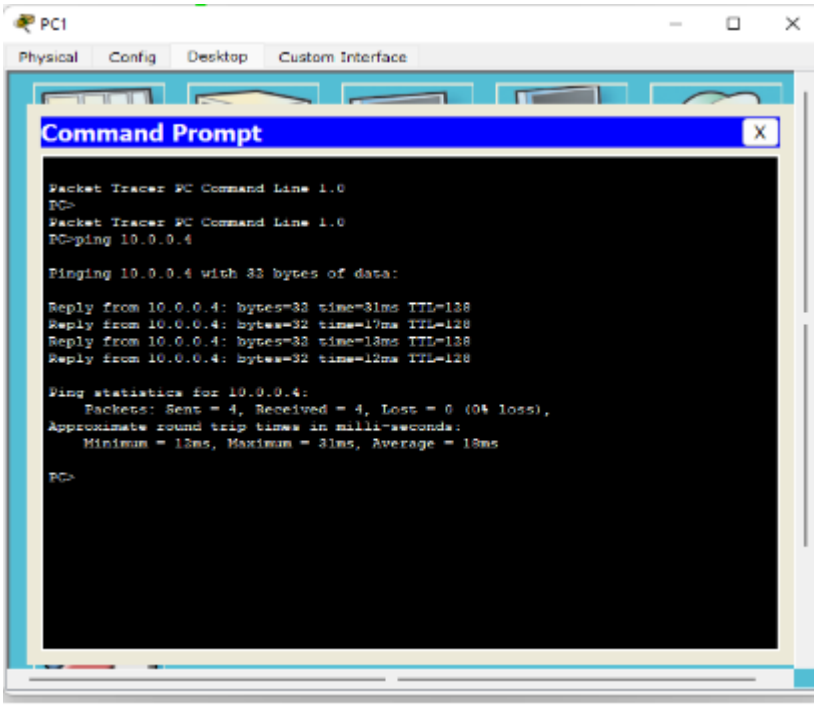
Topology :




PC1
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.


Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

Output :


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=31ms TTL=128
Reply from 10.0.0.4: bytes=32 time=17ms TTL=128
Reply from 10.0.0.4: bytes=32 time=13ms TTL=128
Reply from 10.0.0.4: bytes=32 time=12ms 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 = 12ms, Maximum = 31ms, Average = 18ms
PC>

Week 13

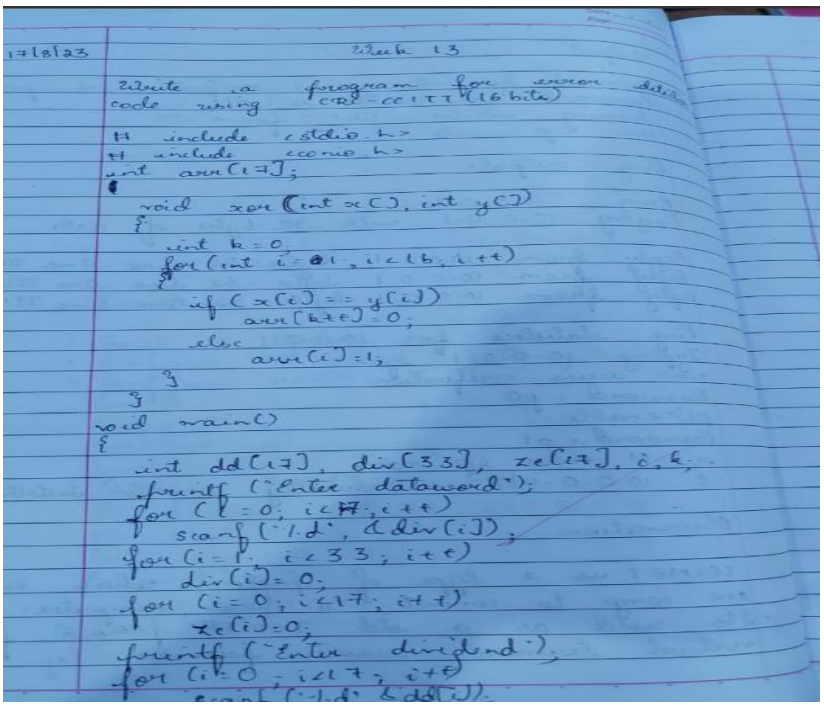
Write a program for error detecting code using CRC-CCITT (16-bits).

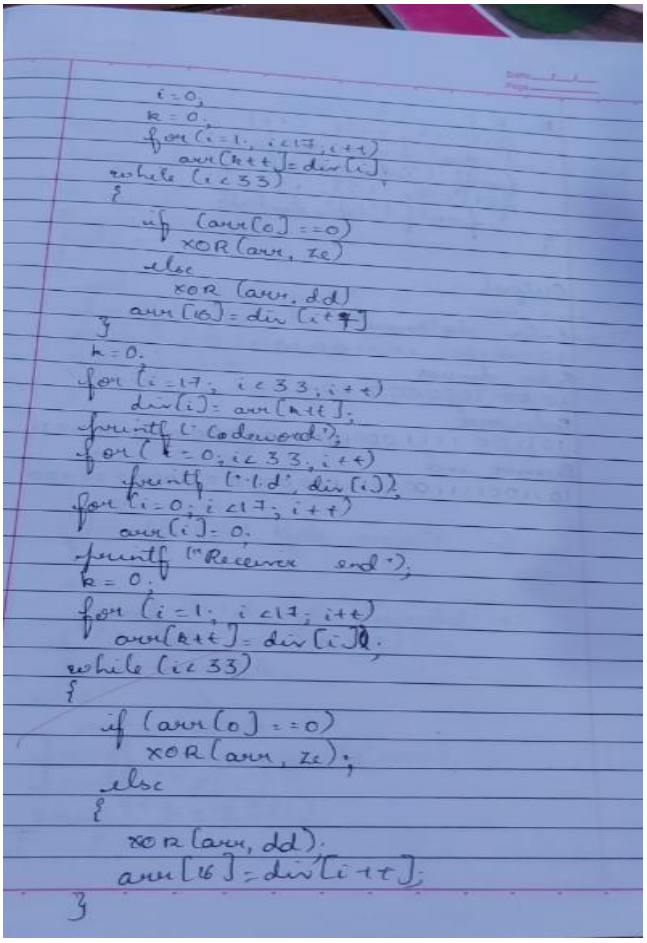
Code:

```
#include<stdio.h>
int arr[17];
void xor(int x[], int y[])
{
    int k=0;
    for(int i=1;i<16;i++)
    {
        if(x[i]==y[i])
            arr[k++]=0;
        else
            arr[i]=1;
    }
}
void main()
{
    int dd[17],div[33],ze[17],i,k;

    printf("Enter the dataword \n");
    for(i=0;i<17;i++)
        scanf("%d",&div[i]);

    for(i=i;i<33;i++)
        div[i]=0;

    for(i=0;i<17;i++)
        ze[i]=0;
    printf("Enter dividend \n");
    for(i=0;i<17;i++)
        scanf("%d",&dd[i]);
    i=0;
    k=0;
        for(i=i;i<17;i++)
            arr[k++]=div[i];
    while(i<33)
    {
        if(arr[0]==0)
            xor(arr,ze);
        else
            xor(arr,dd);

        arr[16]=div[i++];

    }
```

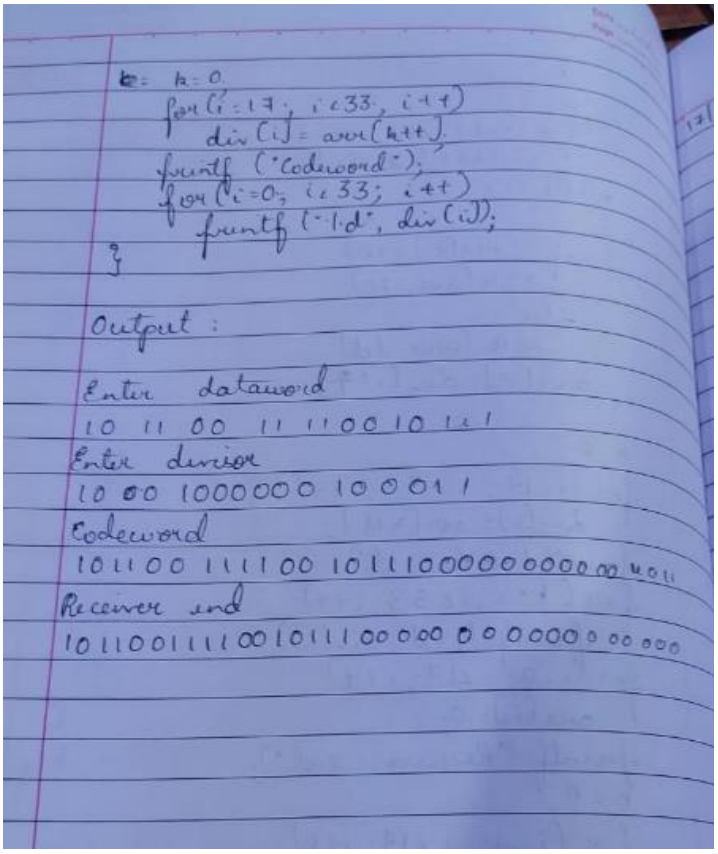
```
    k=0;
    for(i=17;i<33;i++)
        div[i]=arr[k++];
    printf("Codeword: ");
        for(i=0;i<33;i++)
            printf("%d",div[i]);
    for(i=0;i<17;i++)
        arr[i]=0;
    printf("\nAt receiver end \n");

     k=0;
        for(i=i;i<17;i++)
            arr[k++]=div[i];
    while(i<33)
    {
        if(arr[0]==0)
            xor(arr,ze);
        else
            xor(arr,dd);

        arr[16]=div[i++];

    }
    k=0;
    for(i=17;i<33;i++)
        div[i]=arr[k++];

    printf("Codeword: ");
        for(i=0;i<33;i++)
            printf("%d",div[i]);
}
```

Observation :





```
    i=0;
    k=0;
    for(i=1; i<17; i++)
      arr[k++]=div[i];
  while (i<33)
  {
    if (arr[0]==0)
      XOR(arr, zz)
    else
      XOR (arr, dd)
    arr[16]=div[i++];
  }
  k=0;
  for(i=17; i<33; i++)
    div[i]=arr[k++];
  printf("Codeword");
  for(i=0; i<33; i++)
    printf("%d", div[i]);
  for(i=0; i<17; i++)
    arr[i]=0;
  printf("Receiver end");
  k=0;
  for(i=1; i<17; i++)
    arr[k++]=div[i];
  while(i<33)
  {
    if(arr[0]==0)
      XOR(arr, zz);
    else
    {
      XOR(arr, dd);
      arr[16]=div[i++];
  }
```

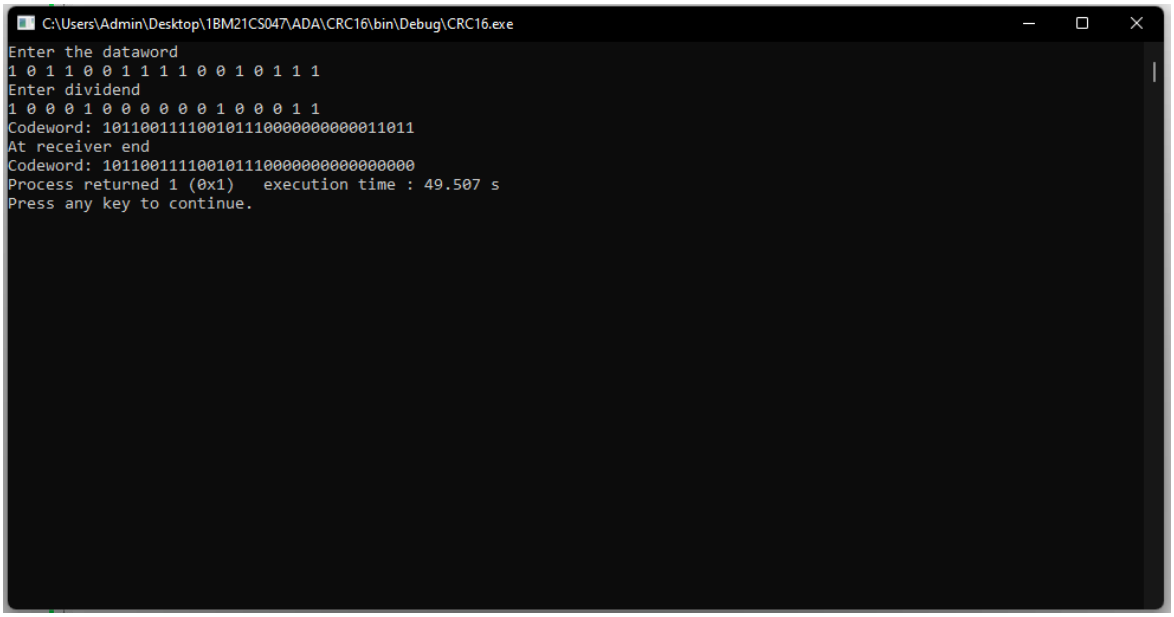


```
  k=0;
    for(i=17; i<33; i++)
      div[i]=arr[k++];
    printf("Codeword");
    for(i=0; i<33; i++)
      printf("%d", div[i]);
}
```

Output :

```
Enter dataword
10 11 00 11 1100 10 111
Enter divisor
10 00 1000000 10 001 1
Codeword
10110 1111 00 10111000000000 00 011
Receiver end
10110011110010111 00000 0 0 0000 0 00 000
```

Output :


C:\Users\Admin\Desktop\1BM21CS047\ADA\CRC16\bin\Debug\CRC16.exe
Enter the dataword
1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
Enter dividend
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
Codeword: 10110011110010111000000000011011
At receiver end
Codeword: 10110011110010111000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.

Week 14

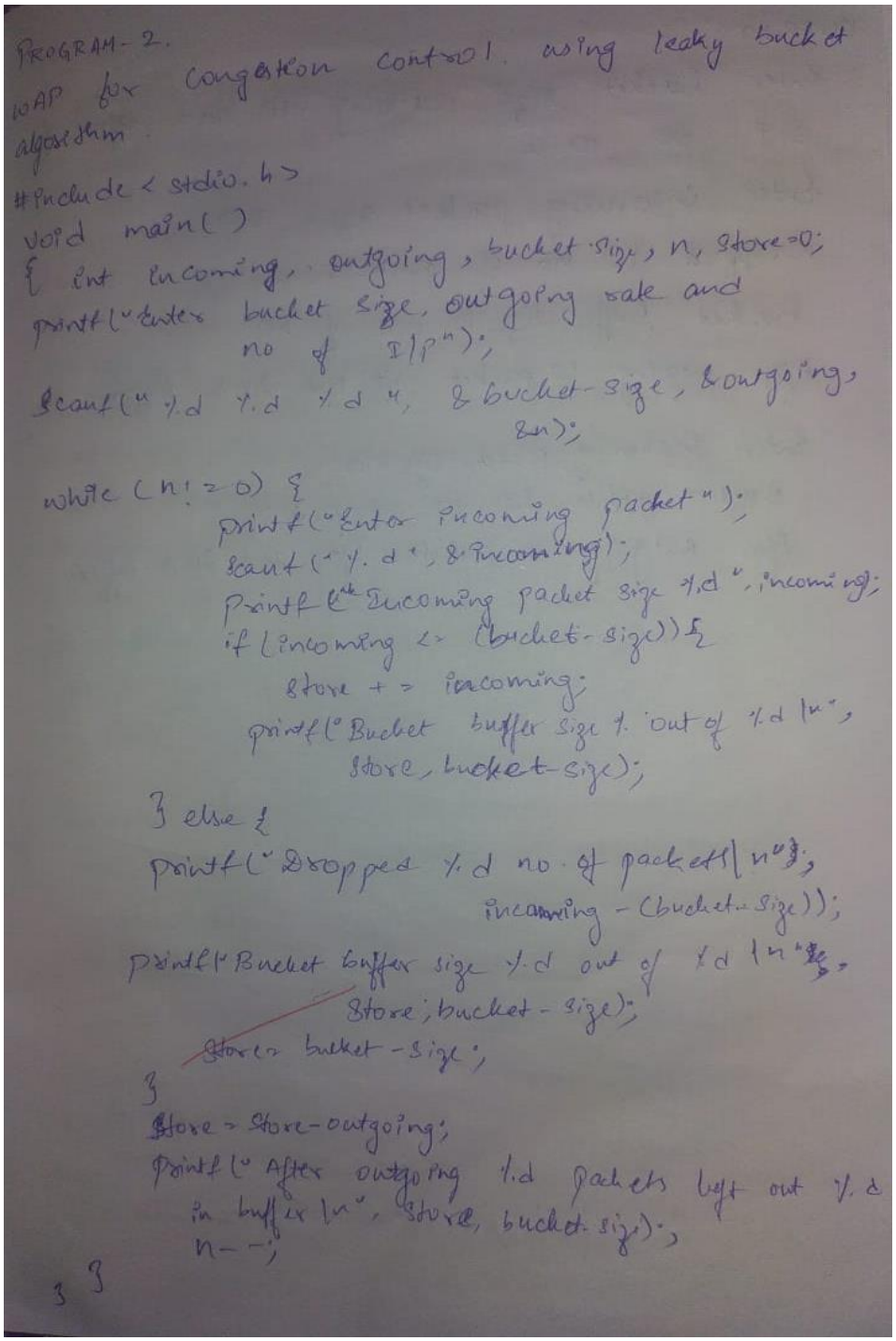
Write a program for congestion control using Leaky bucket algorithm.

Code:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function


int main()
{
   int buckets, outlets, k = 1, num, remaining;


   printf("Enter Bucket size and outstream size\n");
   scanf("%d %d", &buckets, &outlets);
   remaining = buckets;


   while (k)
   {
      num = rand() % 1000; // Generate a random number between 0 and 999
      if (num < remaining)
      {
         remaining = remaining - num;
         printf("Packet of %d bytes accepted\n", num); // Added missing variable
      }
      else
      {
         printf("Packet of %d bytes is discarded\n", num);
      }
      if (buckets - remaining > outlets)
      {
         remaining += outlets; // Fixed the calculation
      }
      else
         remaining = buckets;
      printf("Remaining bytes: %d \n", remaining);
      printf("If you want to stop input, press 0, otherwise, press 1\n");
      scanf("%d", &k);
   }


   while (remaining < buckets) // Fixed the condition
   {
      if (buckets - remaining > outlets)
      {
         remaining += outlets; // Fixed the calculation
      }
```

```
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
    return 0; // Added a return statement to indicate successful completion
}
```

Observation :



PROGRAM-2.
WAP for Congestion control using leaky bucket algorithm

```
#include<stdio.h>
void main()
{ int incoming, outgoing, bucket_size, n, store=0;
printf("Enter bucket size, outgoing rate and
        no of I/P");
scanf("%d %d %d", &bucket_size, &outgoing,
                &n);

while(n!=0) {
        printf("Enter incoming packet");
        scanf("%d", &incoming);
        printf("Incoming packet size %d", incoming);
        if(incoming <= (bucket_size)){
            store += incoming;
          printf("Bucket buffer size %d out of %d \n",
                store, bucket_size);
    } else {
    printf("Dropped %d no of packets\n",
                        incoming - (bucket_size));
    printf("Bucket buffer size %d out of %d \n",
                store; bucket_size);
        store = bucket_size;
    }
    store = store-outgoing;
    printf("After outgoing %d packets left out %d
      in buffer \n", store, bucket_size);
      n--;
}
```

OUTPUT:

Enter bucket size, outgoing rate and no. of

I/P: 20 10 2.

Enter incoming packet size: 30.

Dropped 10 no of packets.

Bucket buffer size 0 out of 20.

After outgoing 10 packet left out 20 in buffer.

Enter incoming packet size=10.

Bucket buffer size 10 out of 20.

After outgoing 10 packets left and 20 in buffer.

Output :

```
PS D:\VS Code> cd "d:\VS code\os\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
```

```
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
```

Week 15

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.

ClientTCP.py

```
from socket import *
serverName = &#39;127.0.0.1&#39;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(&quot;\nEnter file name: &quot;)
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print (&#39;\nFrom Server:\n&#39;)
print(filecontents)
clientSocket.close()
```

ServerTCP.py

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

file=open(sentence,"r")

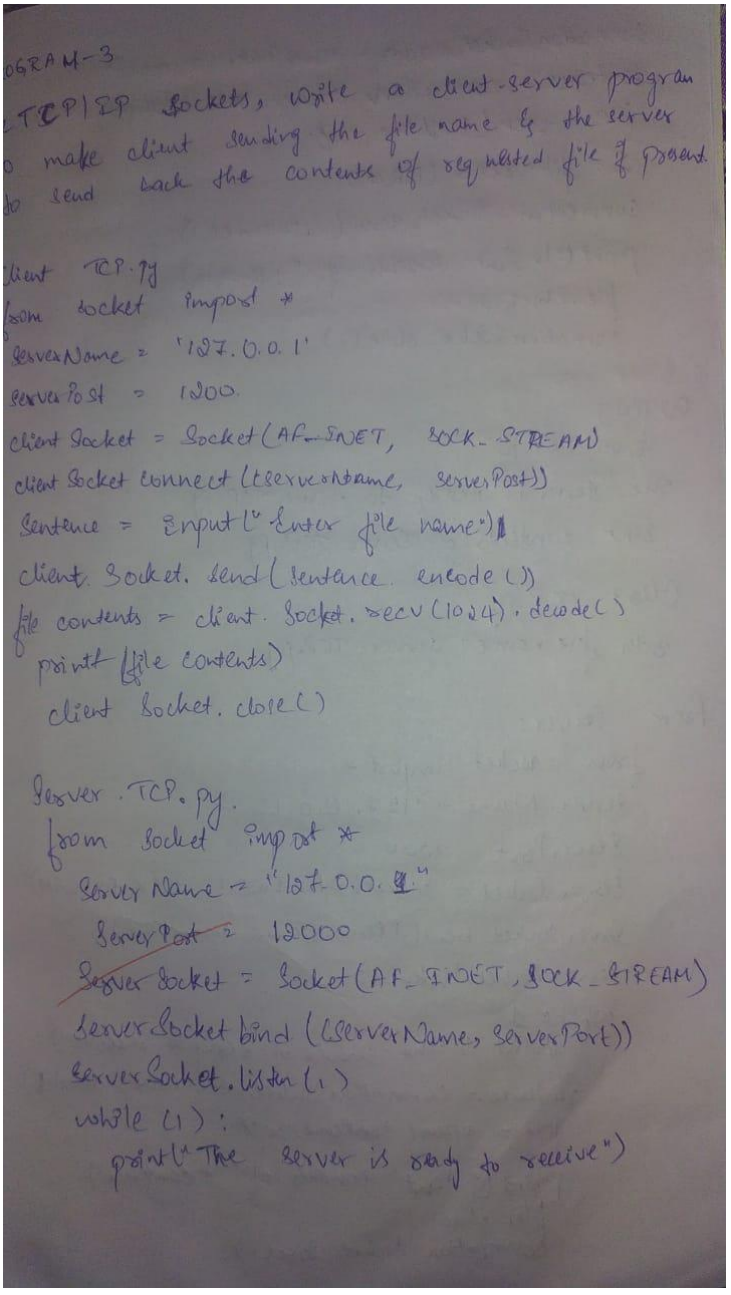
l=file.read(1024)

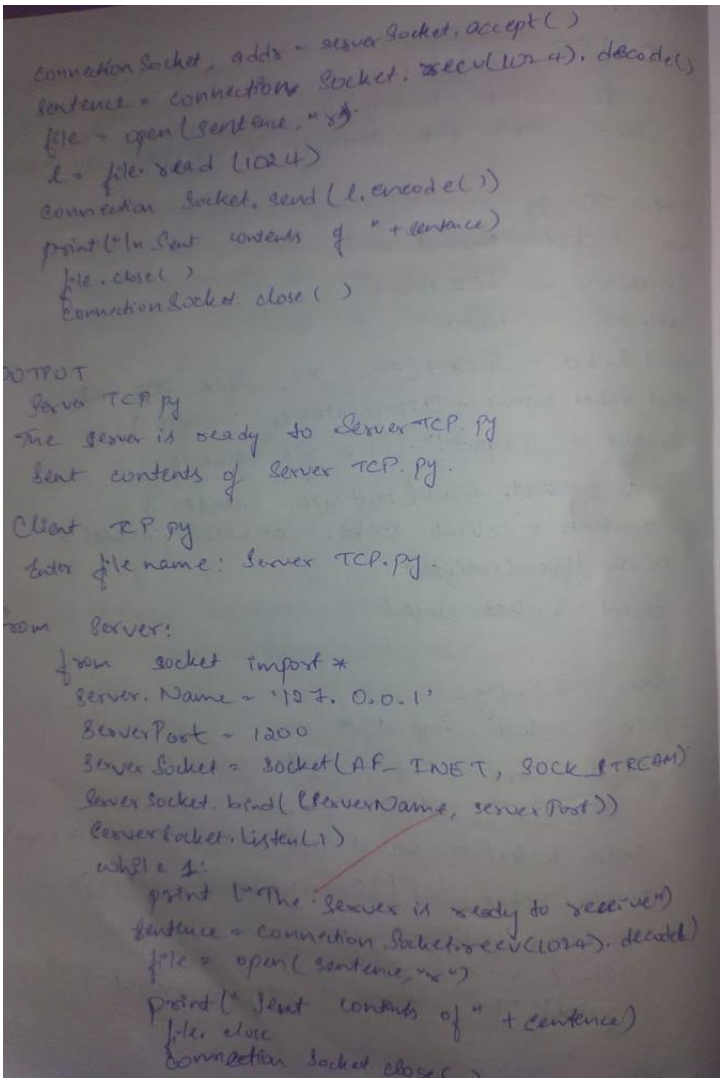
connectionSocket.send(l.encode())

print ('\nSent contents of ' + sentence)

file.close()

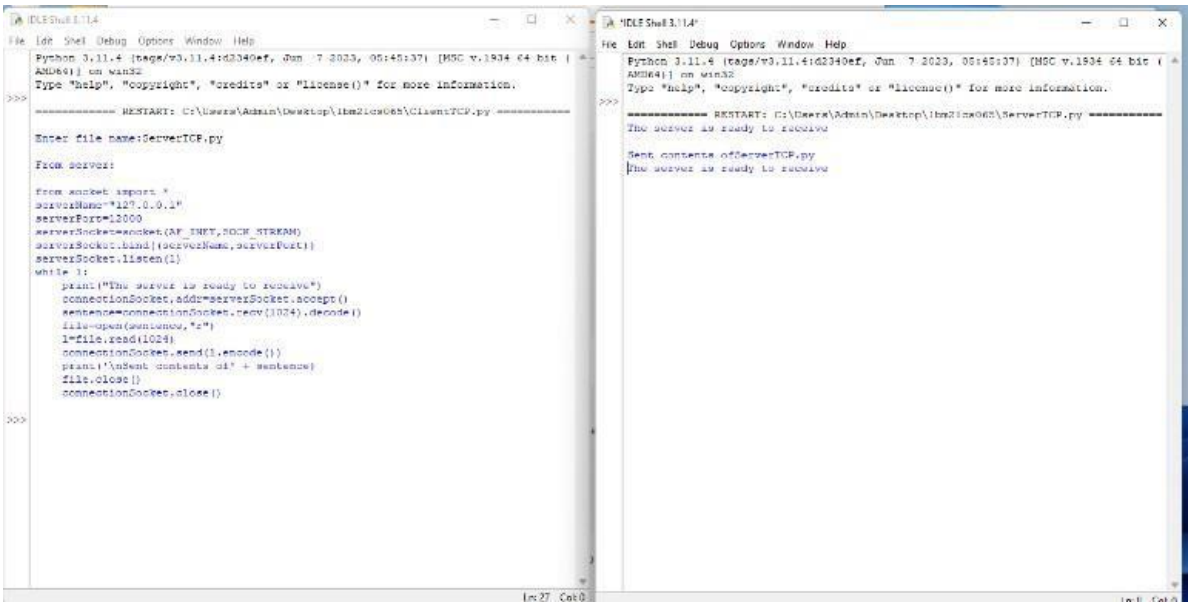
connectionSocket.close()





```
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file = open(sentence, "r")
l = file.read(1024)
connectionSocket.send(l.encode())
print('\n Sent contents of ' + sentence)
file.close()
connectionSocket.close()

OUTPUT
Server TCP.py
The server is ready to Server TCP.py
Sent contents of Server TCP.py.

Client TCP.py
Enter file name: Server TCP.py.

From Server:
from socket import *
serverName = '127.0.0.1'
serverPort = 1200
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print('The Server is ready to receive')
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    print('Sent contents of ' + sentence)
    file.close
    connectionSocket.close()
```

Output :


IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\1bm21cs065\ClientTCP.py ==========
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()
>>>
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\1bm21cs065\ServerTCP.py ==========
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive

Week 16

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.

ClientUDP.py

```
from socket import *
serverName = &amp;quot;127.0.0.1&amp;quot;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input(&amp;quot;\nEnter file name: &amp;quot;)
clientSocket.sendto(bytes(sentence,&amp;quot;utf-8&amp;quot;),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print (&amp;#39;\nReply from Server:\n&amp;#39;)
print (filecontents.decode(&amp;quot;utf-8&amp;quot;))
# for i in filecontents:
# print(str(i), end = &amp;#39;&amp;#39;)
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((&amp;quot;127.0.0.1&amp;quot;, serverPort))
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode(&amp;quot;utf-8&amp;quot;)
file=open(sentence,&amp;quot;r&amp;quot;)
```

con=file.read(2048)

serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

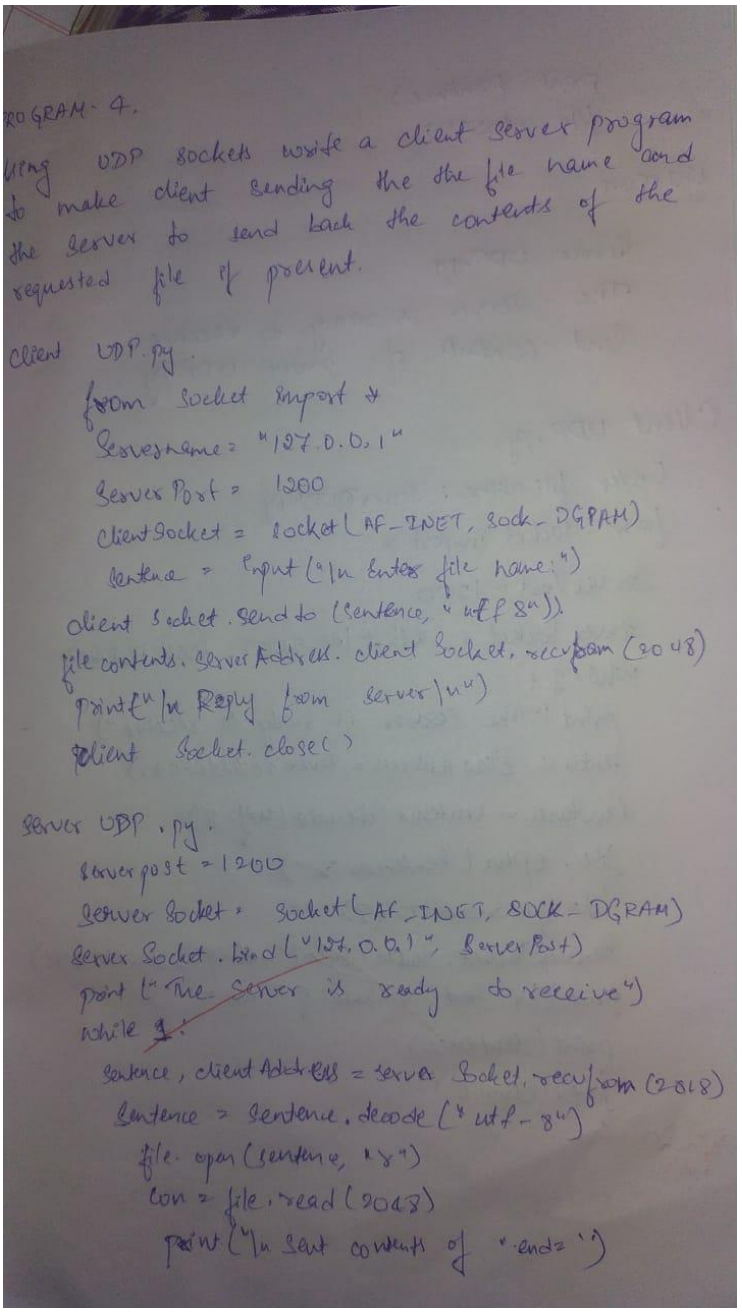
print ('\nSent contents of ', end = ' ')

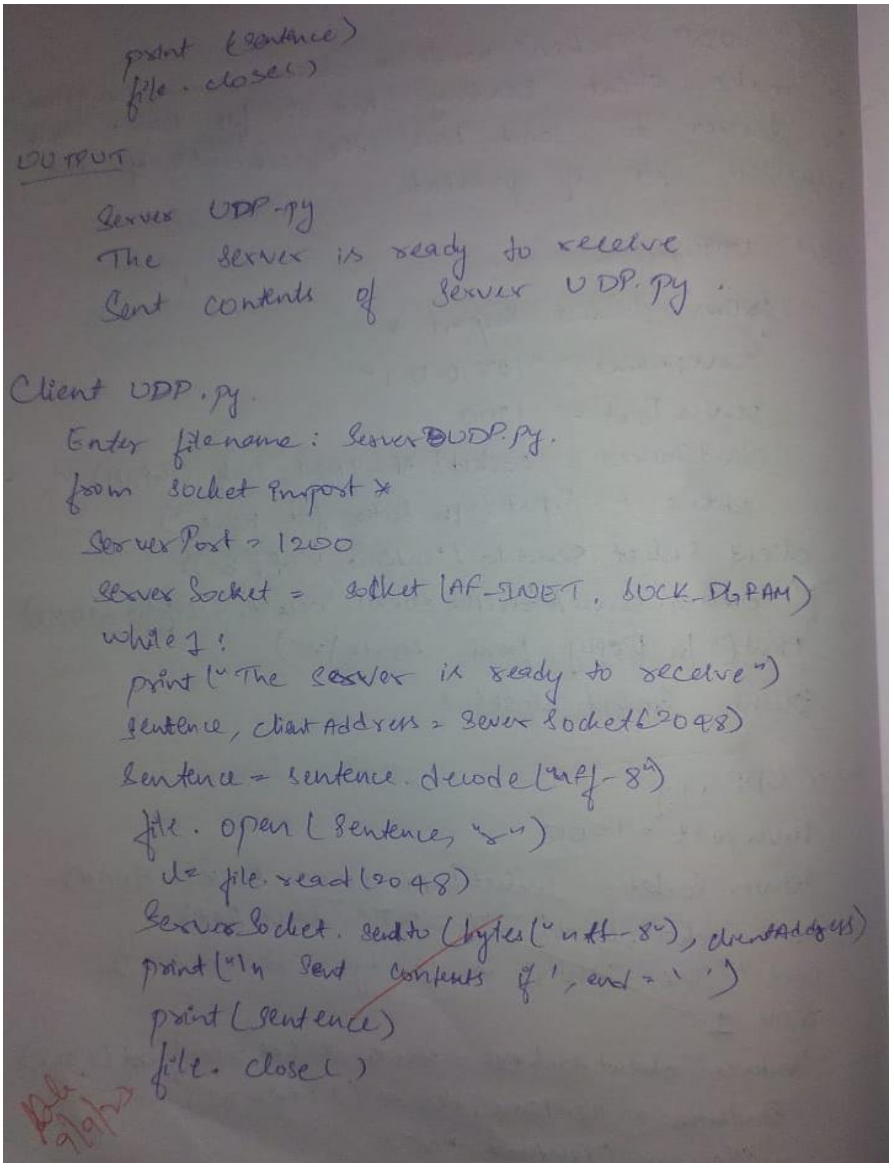
print (sentence)

for i in sentence:

print (str(i), end = '')

file.close()





print (sentence)
file.close()

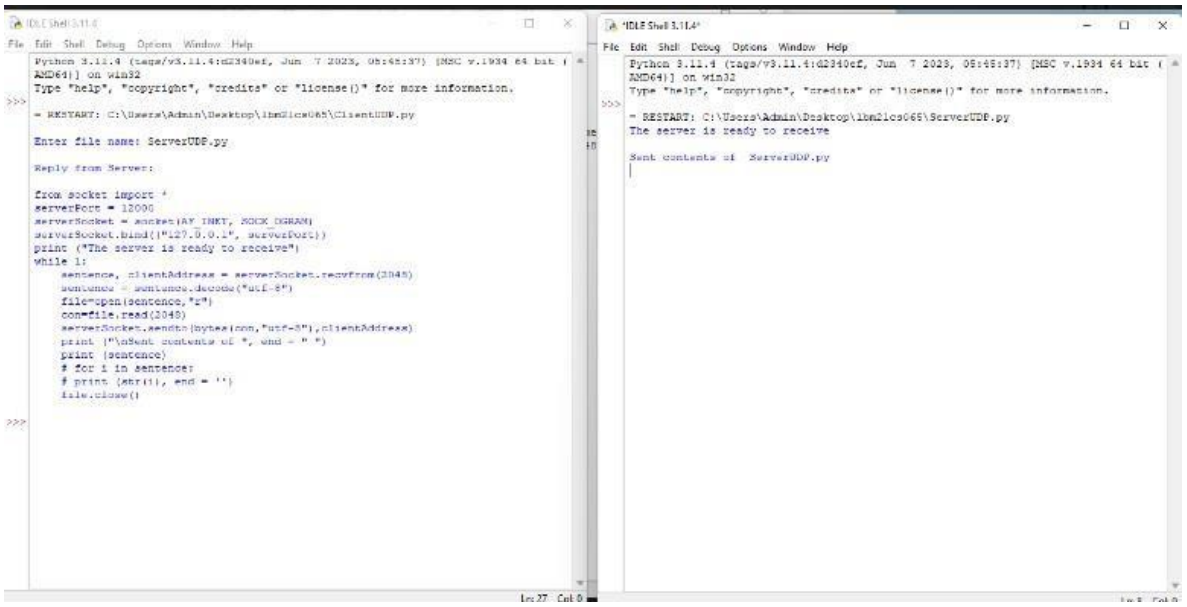
OUTPUT

Server UDP.py
The server is ready to receive
Sent contents of server UDP.py.

Client UDP.py.

Enter filename: serverUDP.py.
from socket import *
ServerPort = 12000
ServerSocket = socket (AF_INET, SOCK_DGRAM)
while 1:
print ("The server is ready to receive")
sentence, clientAddress = ServerSocket(2048)
sentence = sentence.decode("utf-8")
file.open (sentence, "r")
l= file.read(2048)
ServerSocket.sendto (bytes("utf-8"), clientAddress)
print ("\n Sent contents of ", end = ' ')
print (sentence)
file.close()

Output :


File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\Desktop\lbm21cs065\ClientUDP.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")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ("\nSent contents of ", end = " ")
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
>>>
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\Desktop\lbm21cs065\ServerUDP.py
The server is ready to receive
Sent contents of ServerUDP.py