VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

AVANI KAMATH (1BM21CS036)

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 **AVANI KAMATH(1BM21CS036),** 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 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.

Name of the Lab-Incharge
M Lakshmi Neelima
Designation
Assistant Professor
Department of CSE
BMSCE, Bengaluru

`

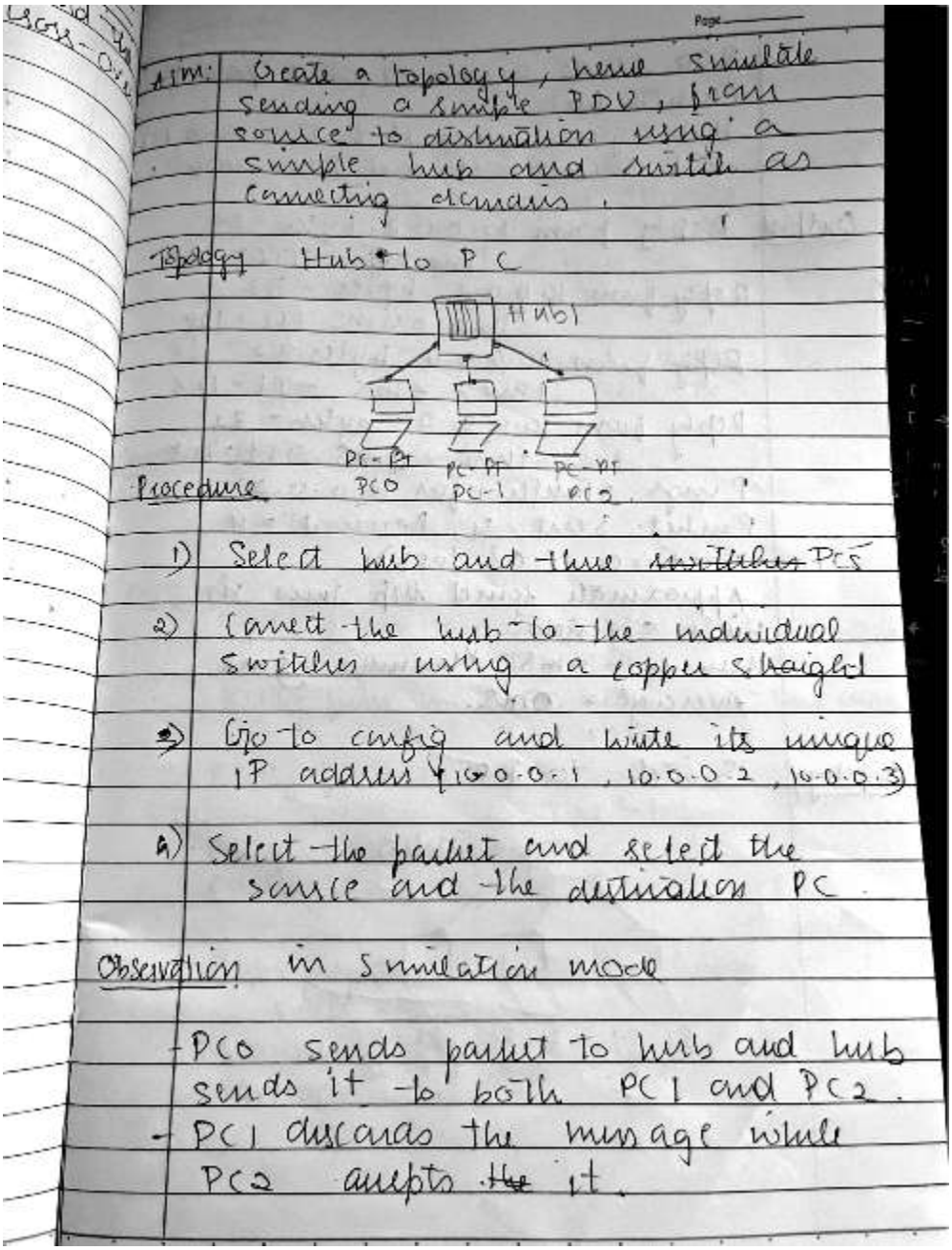
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Experiment No 1

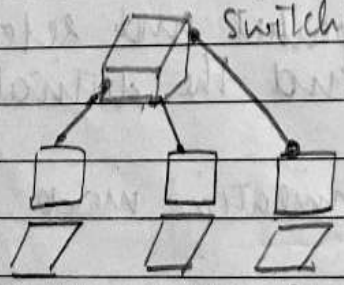
Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

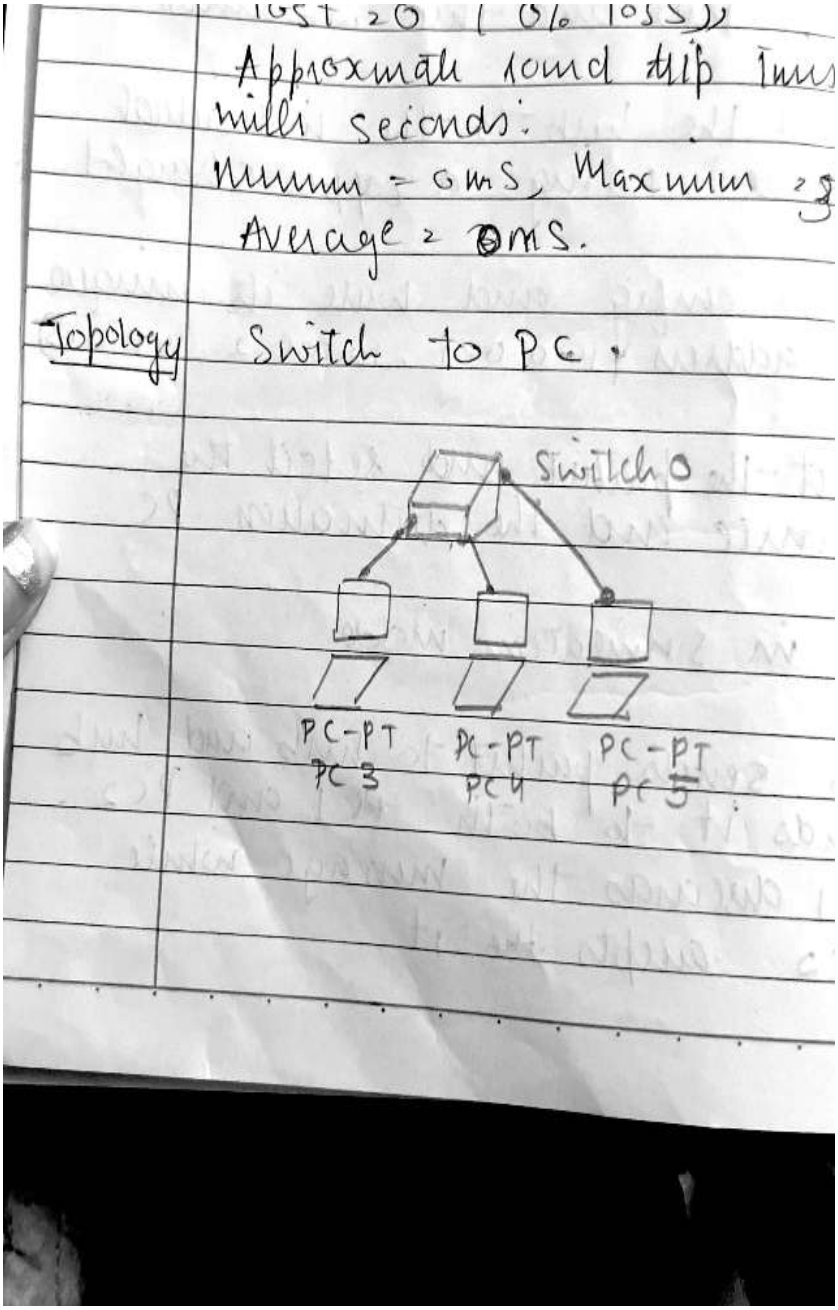
Observation:



Reply from 10.0.0.2: bytes = 32
time = 5ms TTL = 12
Reply from 10.0.0.2: bytes = 32
Time = 0 ms TTL = 128
Reply from 10.0.0.2: bytes = 32
Time = 3 ms TTL = 128
Reply from 10.0.0.2: bytes = 32
time = 0 ms TTL = 128
Pings statistics for 10.0.0.2:
Packet: Sent = 4, Received = 4,
lost = 0 (0% loss),
Approximate round trip times in
milli seconds:
Minimum = 0 ms, Maximum = 3 ms,
Average = 0 ms.

Switch to PC.




Test 20 (0% loss))
Approximate round trip in
milli seconds.
Minimum = 0ms, Maximum = 1
Average = 0ms.
Topology Switch to PC.
Switch 0
PC-PT
PC 3
PC-PT
PC 4
PC-PT
PC 5

Date / /
Page

Procedure

1) Select a switch and 3 PCs
2) ~~Fi~~
2) Connect the switch to the individual PCs using a copper straight.
3) Select config and give its unique IP address (10.0.0.4, 10.0.0.5, 10.0.0.6)
4) Select the PDU packet and select the Source and the destination PC.

~~Observations in Simulation mode~~

- ~~PC 3 sends packet~~

Output

Reply from 10.0.0.5: bytes = 32 Time = 0ms TTL = 128
Reply from 10.0.0.5: bytes = 32 Time = 0ms TTL = 128
Reply from 10.0.0.5 bytes = 32 Time = 3ms TTL = 128

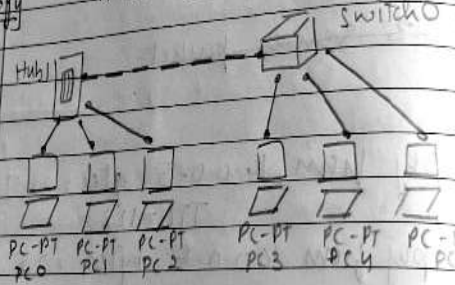
Ping statistics for 10.0.0.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip Time in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 0ms.

Observation in simulation

- PC 3 sends packet to switch and it sends to both PC 4 & PC 5 in first round.
- PC 4 reject PC 5 accepts and sends acknowledgement packet to both PC 3 & PC 5.

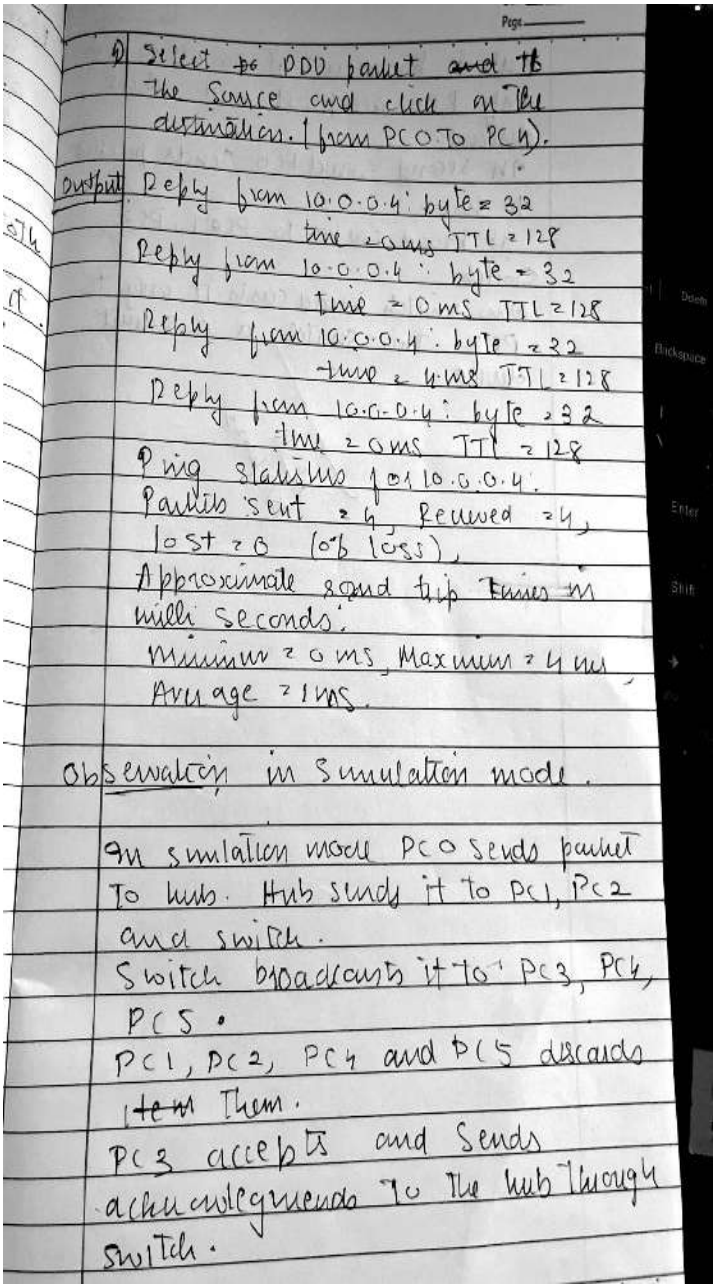
Now when PC 3 sends packet it cascades it PC 3 accepts & PC 4 sends packet it sends only to PC 5.

Topology - Hub & switch and PC



Procedure

1) Connect the hub and the switch using a copper cross-over.
2) Connect the 3 PC each to hub and switch and give IP address (10.0.0.1, 10.0.0.2 ... 10.0.0.6)
3) Select a source PC from the one that is connected to the hub. Select a destination PC that is connected to the switch.

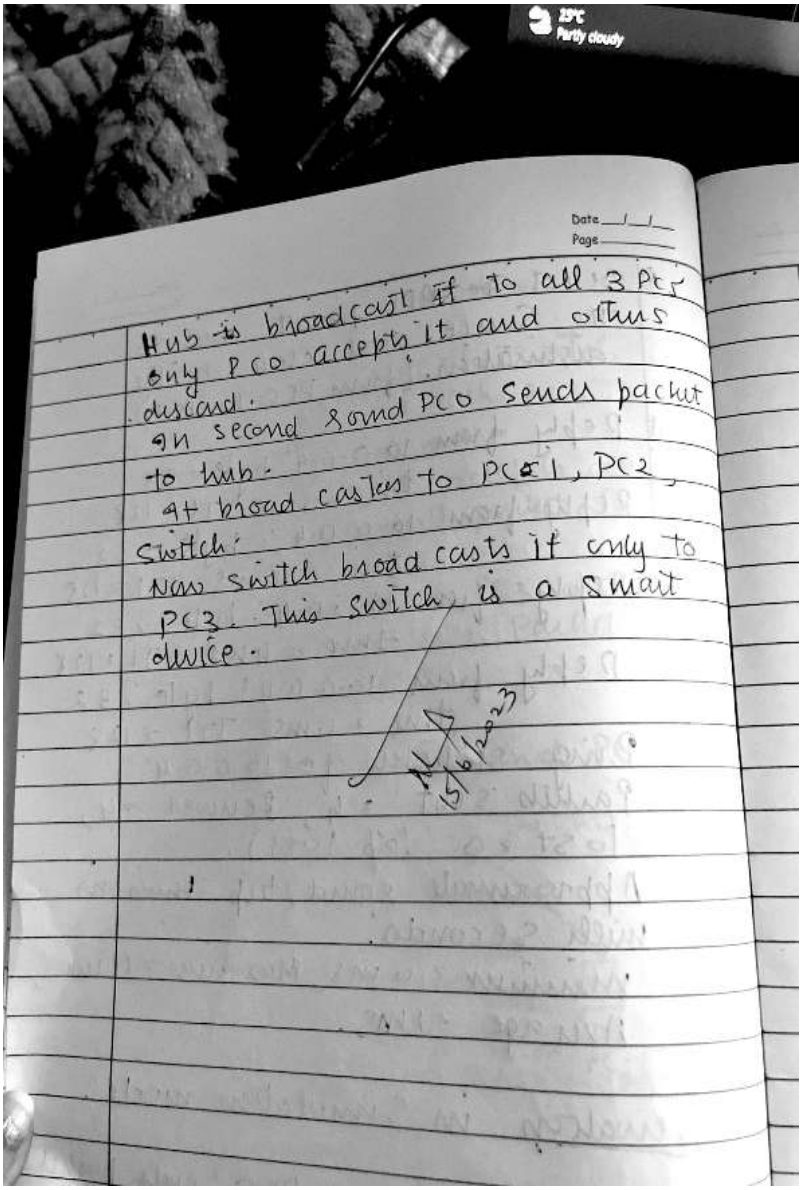


2) Select to PDU packet and to the Source and click on the destination. (from PC0 To PC4).

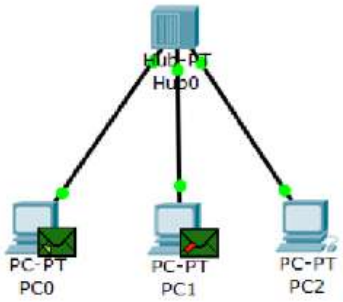
Output Reply from 10.0.0.4: byte = 32
time = 0ms TTL = 128
Reply from 10.0.0.4: byte = 32
time = 0ms TTL = 128
Reply from 10.0.0.4: byte = 32
time = 4ms TTL = 128
Reply from 10.0.0.4: byte = 32
time = 0ms TTL = 128
Ping statistics for 10.0.0.4:
Packets sent = 4, Recieved = 4,
lost = 0 (0% loss),
Approximate round trip times in milli seconds:
Minimum = 0ms, Maximum = 4ms
Average = 1ms.

Observation in Simulation mode.

In simulation mode PC0 sends packet To hub. Hub sends it to PC1, PC2 and switch.
Switch broadcasts it to PC3, PC4, PC5.
PC1, PC2, PC4 and PC5 discards them.
PC3 accepts and sends acknowledgements To the hub through switch.

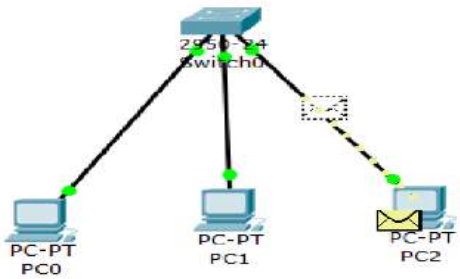

25°C
Partly cloudy
Date
Page
Hub is broadcast if to all 3 PCs only PC0 accepts it and others discard.
In second round PC0 sends packet to hub.
It broad casts to PC1, PC2, switch.
Now switch broad casts it only to PC3. This switch is a smart device.
15/6/2023

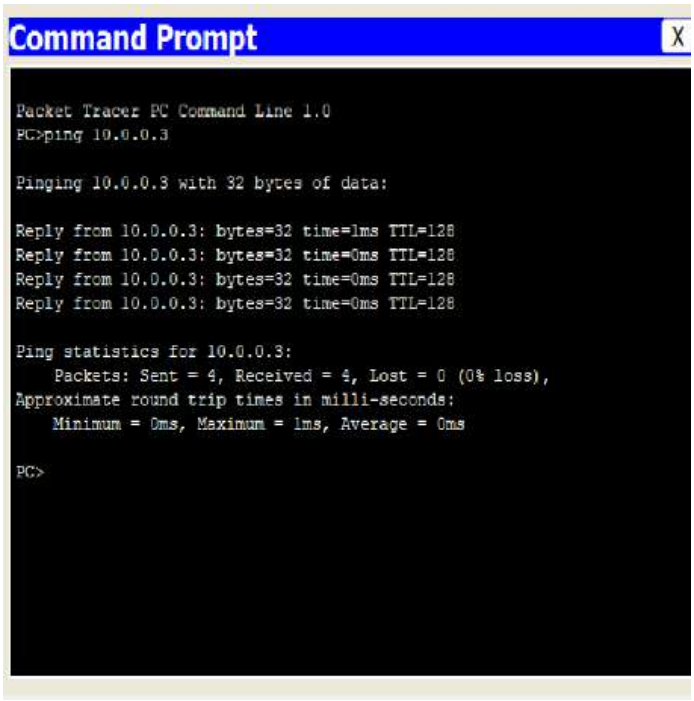
Output:
hub and pc



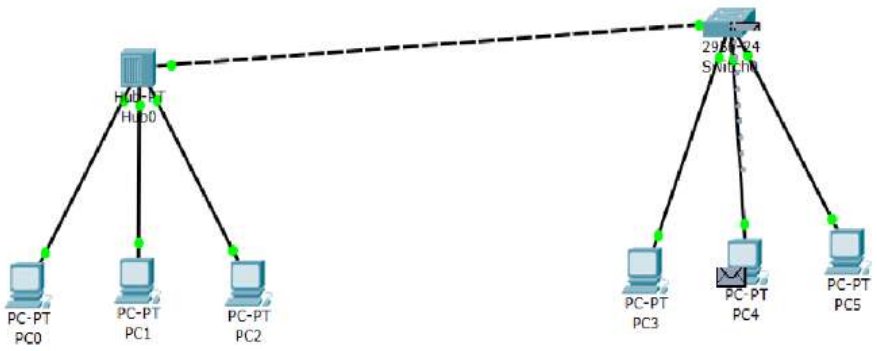

```
PC>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=1ms TTL=128
Reply from 10.0.0.5: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

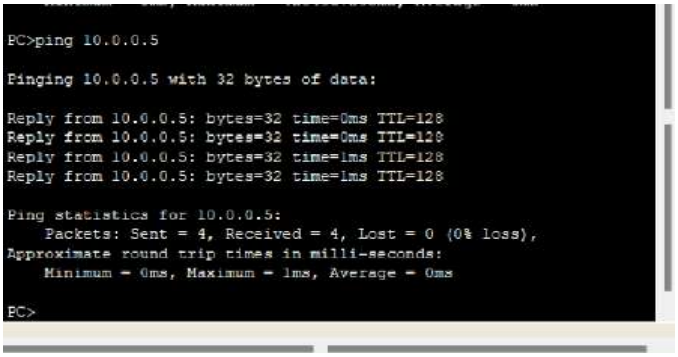
switch and pc





hub,switch and pc





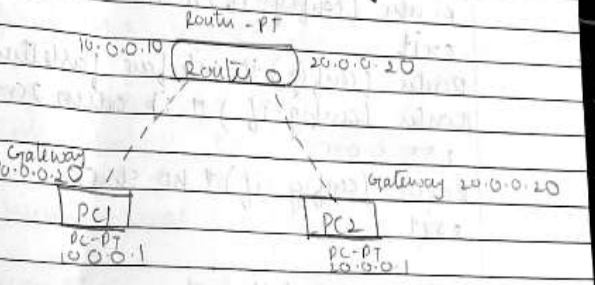
EXPERIMENT 2

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

Observation:

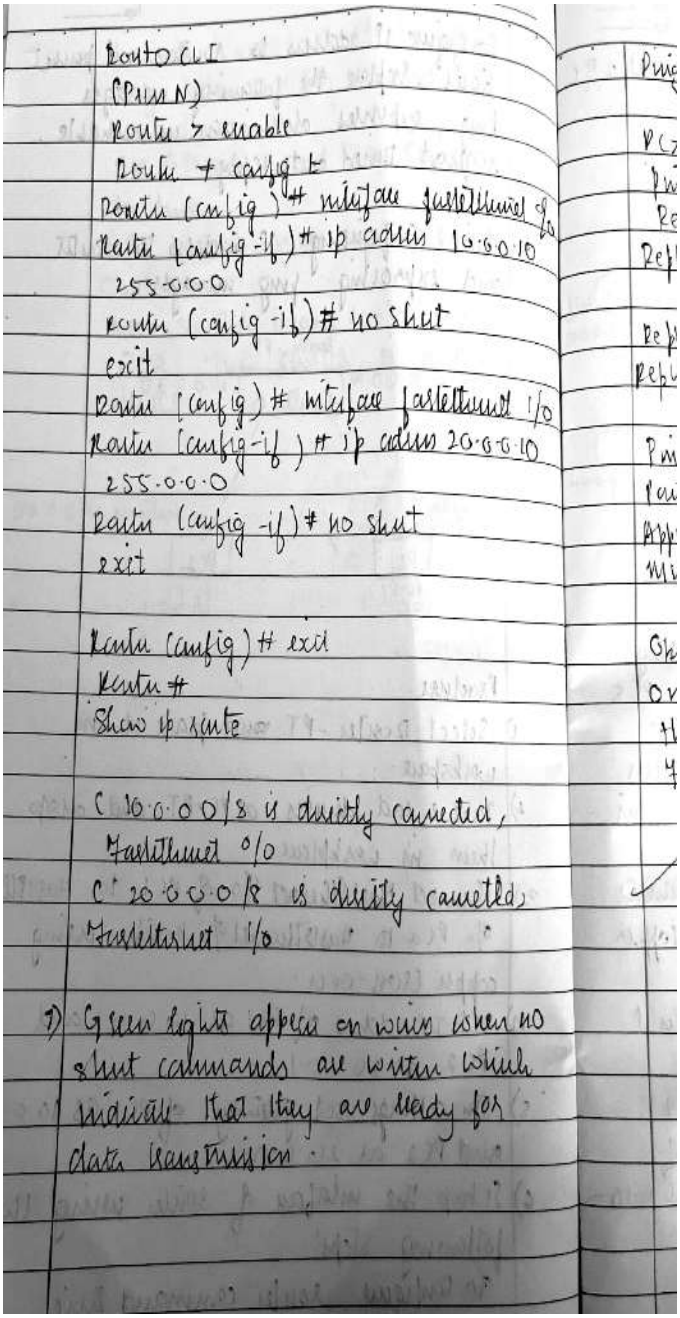
Configure IP address to routers in packet tracer. Explore the following messages: Ping responses, destination unreachable, request timed out, reply.

2a Aim: Configuring IP address to routers and exploring ping messages.



Procedure

1) Select Router-PT and place it in workspace.
2) Take 2 end devices as PC-PT and drop them in workplace.
3) Connect FastEthernet 0/0 of PC1 to FastEthernet 0/0 PC2 to FastEthernet 0/0 router using copper cross-over.
4) Set IP address of PC1 as 10.0.0.1 and PC2 as 20.0.0.1
5) In settings set gateway of PC1 as 10.0.0.10 and PC2 as 20.0.0.1
6) Setup the interface of router using the following steps:
   To configure router command line interface (CLI) is used.



Router CLI
(Press N)
Router > enable
Router # config t
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.10
255.0.0.0
Router (config-if) # no shut
exit
Router (config) # interface fastEthernet 1/0
Router (config-if) # ip address 20.0.0.10
255.0.0.0
Router (config-if) # no shut
exit

Router (config) # exit
Router #
Show ip route

C 10.0.0.0/8 is directly connected,
FastEthernet 0/0
C 20.0.0.0/8 is directly connected,
FastEthernet 1/0

1) Green lights appear on wires when no shut commands are written which indicate that they are ready for data transmission.

Date / /
Page

Ping output in PC0:-

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data.
Request Timed out.
Reply from 20.0.0.1: bytes = 32 Time = 0ms, TTL = 127
Reply from 20.0.0.1: bytes = 32 Time = 0ms TTL = 127
Reply from 20.0.0.1: bytes = 32 Time = 0ms TTL = 127

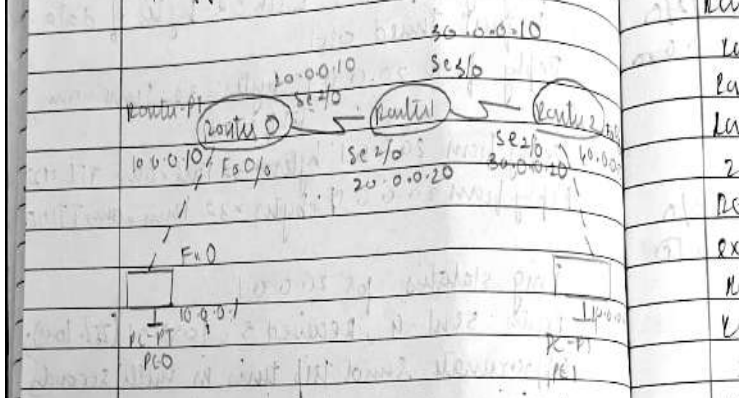
Ping statistics for 20.0.0.1
Packets: Sent = 4, Received 3: Loss = 1 (75% loss),
Approximate round trip times in milli seconds
Minimum = 0ms, Maximum = 1ms, Average

Observation
On pinging in PC0 for the first time there is a 25% loss.
From next ping there are no losses.

13/7/2023

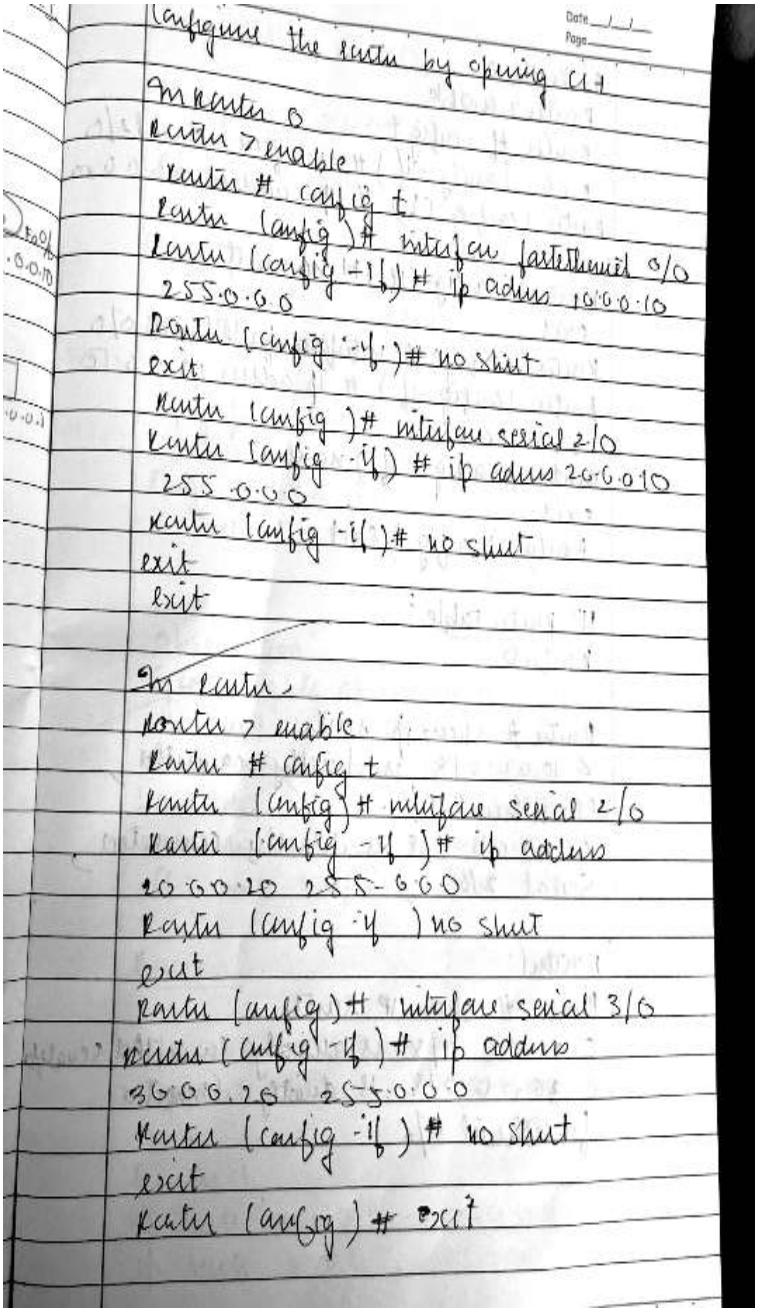
2b) Aim: configure using 3 routers and 2 PC's

Topology:



Procedure:

1) The network is started by selecting end devices PC0 to PC1 i.e generic PCs and placing them in work space.
2) Select 3 Router-PT and place them as Router 0, Router 1 and Router 2 in workspace.
3) PC0 & PC1 are connected to Router 0 and Router 2 respectively using copper cross over.
4) Connect Router 0 to router 1, Router 1 to Router 2.
5) Set up IP address of PC0 to 10.0.0.1 PC to 20.0.0.1. Set gateway of PC0 as 10.0.0.20 PC1 as 30.0.0.10



Configure the router by opening CLI

On Router 0

Router > enable

Router # config t

Router (config)# interface fastethernet 0/0

Router (config-if)# ip addres 10.0.0.10 255.0.0.0

Router (config-if)# no shut

exit

Router (config)# interface serial 2/0

Router (config-if)# ip addres 20.0.0.10 255.0.0.0

Router (config-if)# no shut

exit

exit

On Router 1

Router > enable

Router # config t

Router (config)# interface serial 2/0

Router (config-if)# ip address 20.00.20 255.0.0.0

Router (config-if) no shut

exit

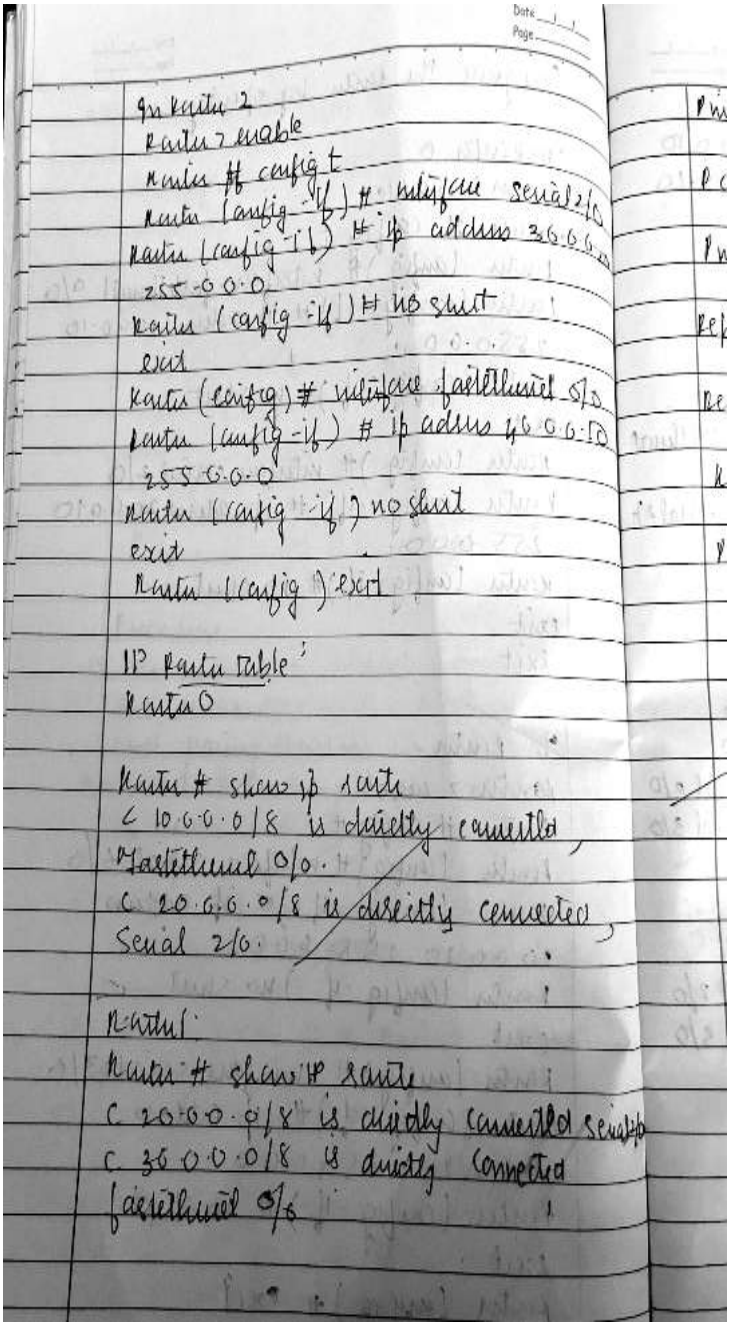
Router (config)# interface serial 3/0

Router (config-if)# ip addres 30.0.0.20 255.0.0.0

Router (config-if)# no shut

exit

Router (config)# exit

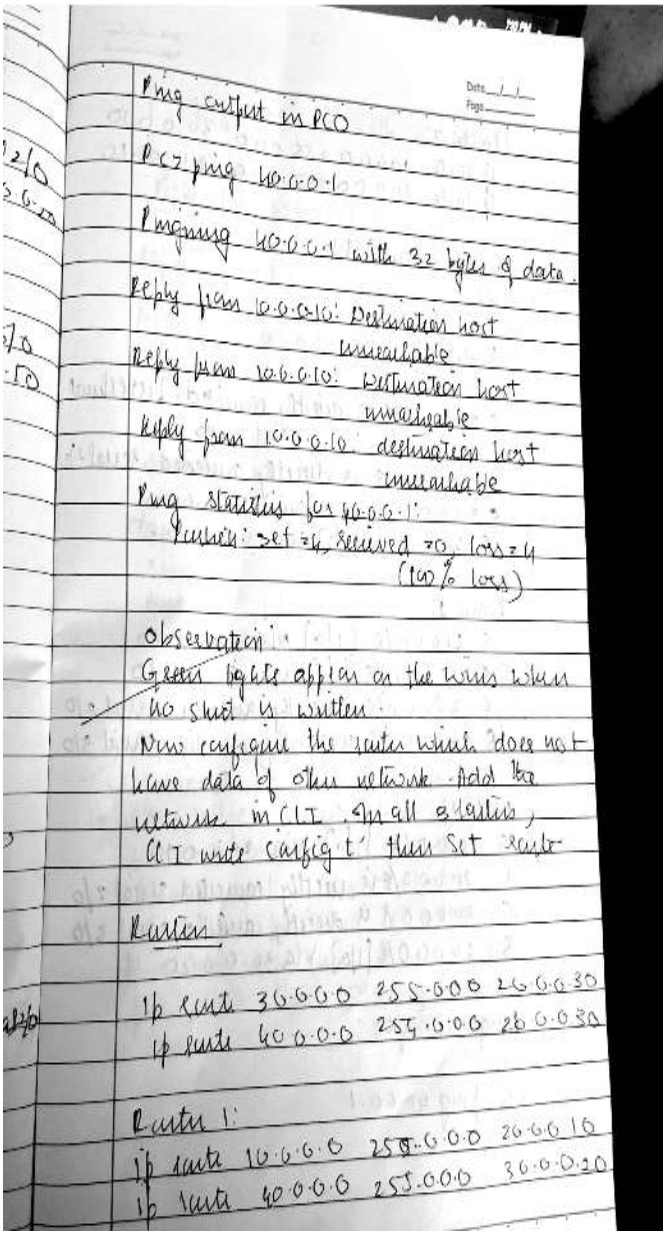


On Router 2
Router > enable
Router # config t
Router (config-if) # interface serial 2/0
Router (config-if) # ip address 30.0.0.2
255.0.0.0
Router (config-if) # no shut
exit
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 40.0.0.10
255.0.0.0
Router (config-if) no shut
exit
Router (config) exit

IP Route Table:
Router 0

Router # show ip route
C 10.0.0.0/8 is directly connected,
FastEthernet 0/0.
C 20.0.0.0/8 is directly connected,
Serial 2/0.

Router1:
Router # show ip route
C 20.0.0.0/8 is directly connected serial
C 30.0.0.0/8 is directly connected
fastEthernet 0/0



Ping output in PC0

PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 20.0.0.10: Destination host unreachable

Reply from 20.0.0.10: Destination host unreachable

Reply from 20.0.0.10: destination host unreachable

Ping statistics for 40.0.0.1:

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

Observation:

Green lights appear on the wires when no shut is written.

Now configure the router which does not have data of other network. Add the network in CLI. Install 3 routers, CLI write config t' then set route.

Routers:

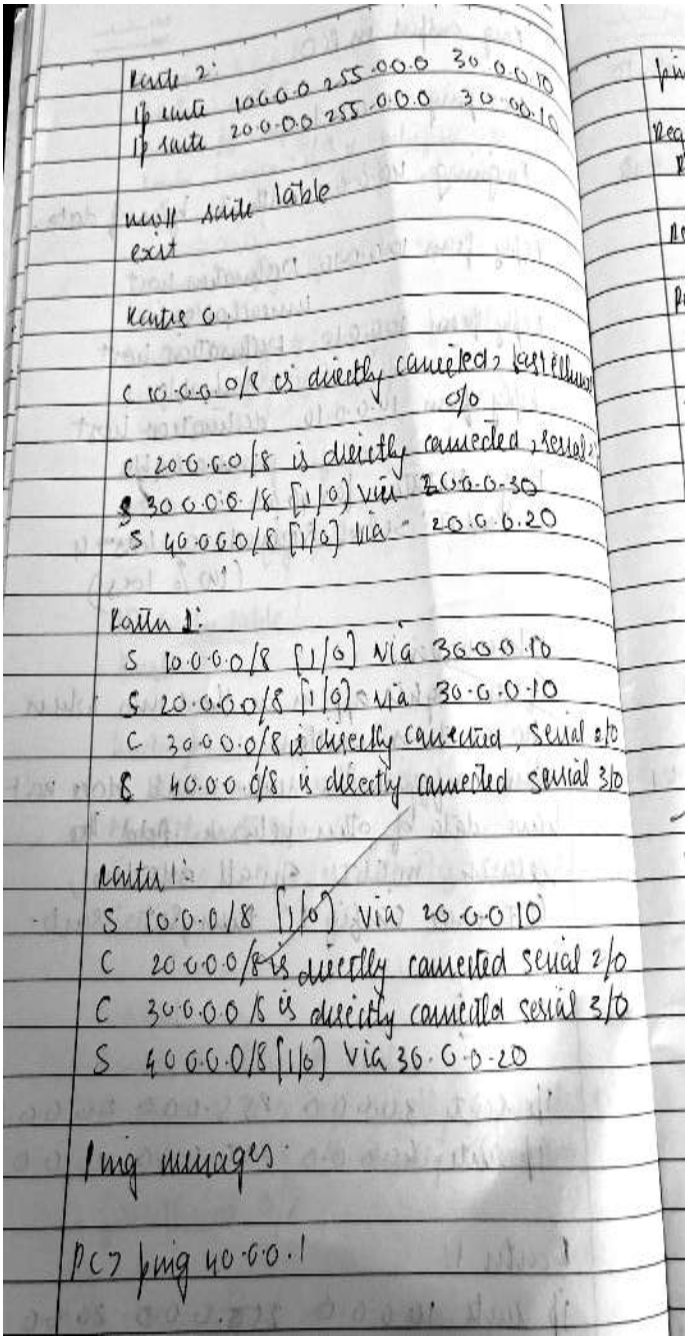
ip route 30.0.0.0 255.0.0.0 20.0.0.30

ip route 40.0.0.0 255.0.0.0 20.0.0.30

Router 1:

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



Router 2:
ip route 10.0.0.0 255.0.0.0 30.0.0.10
ip route 20.0.0.0 255.0.0.0 30.0.0.10

show ip route table
exit

Router 0:

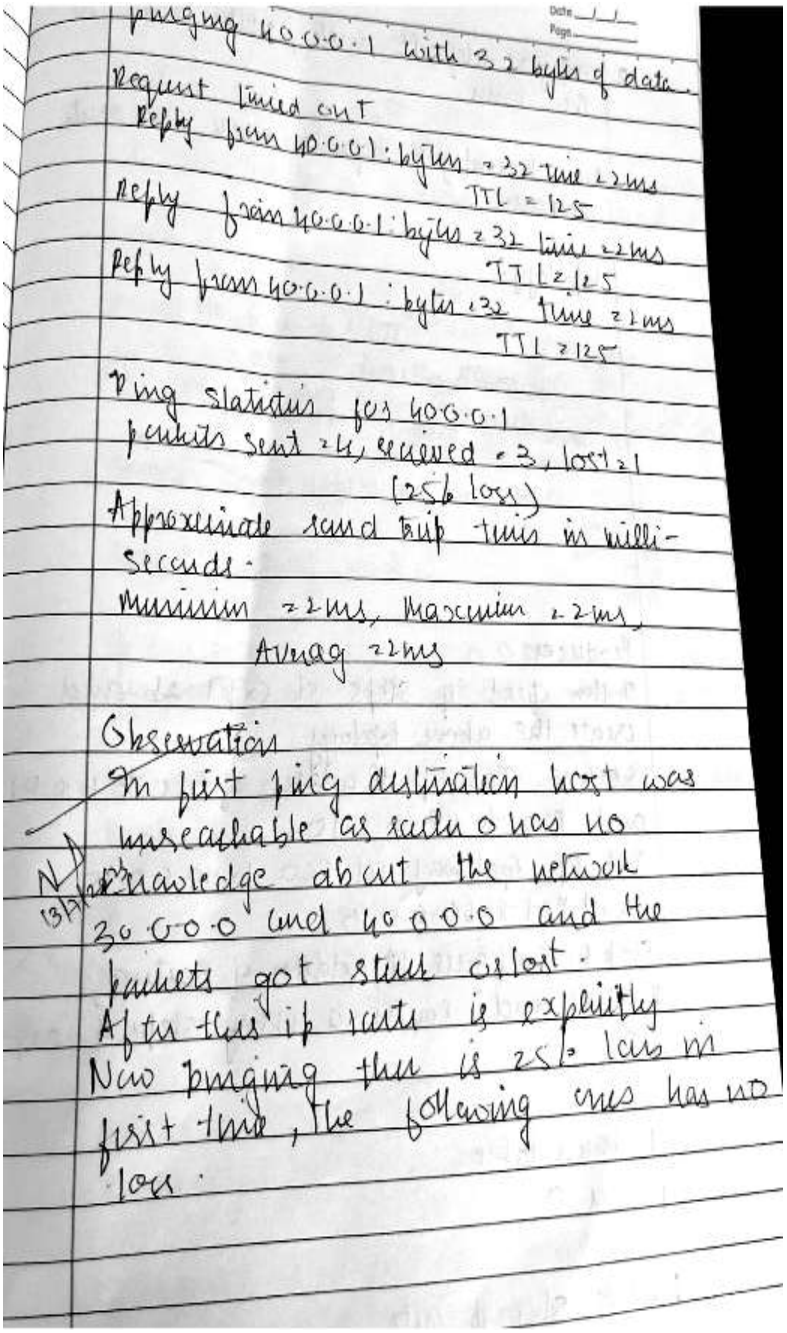
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected, Serial 2/0
S 30.0.0.0/8 [1/0] via 20.0.0.30
S 40.0.0.0/8 [1/0] via 20.0.0.20

Router 1:
S 10.0.0.0/8 [1/0] via 30.0.0.10
S 20.0.0.0/8 [1/0] via 30.0.0.10
C 30.0.0.0/8 is directly connected, Serial 3/0
S 40.0.0.0/8 is directly connected serial 3/0

Router 1:
S 10.0.0.0/8 [1/0] via 20.0.0.10
C 20.0.0.0/8 is directly connected serial 2/0
C 30.0.0.0/8 is directly connected serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.20

Ping messages:

PC7 ping 40.0.0.1

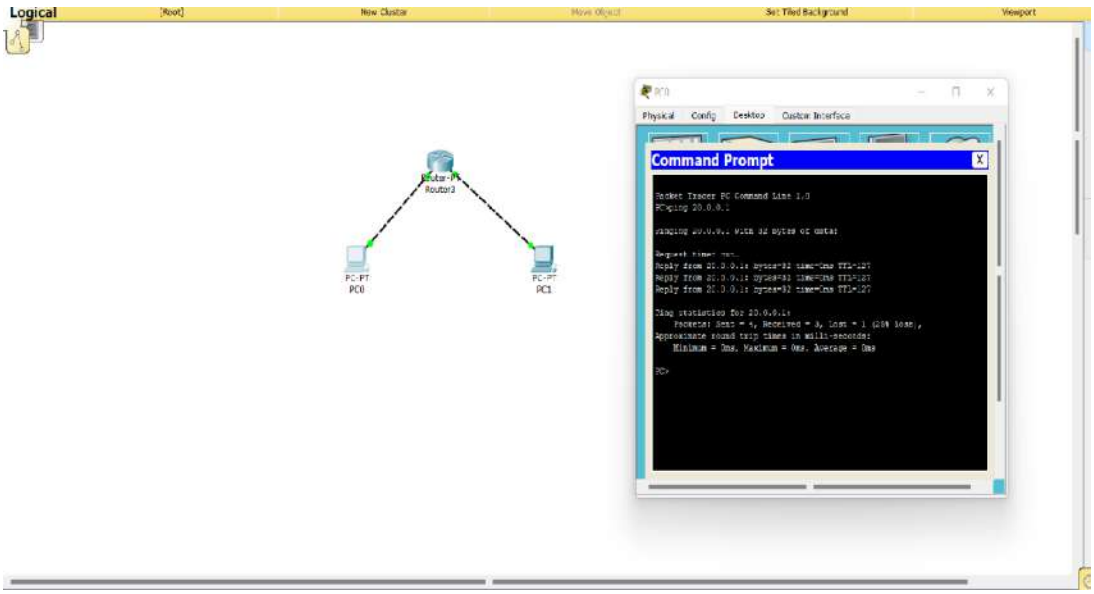


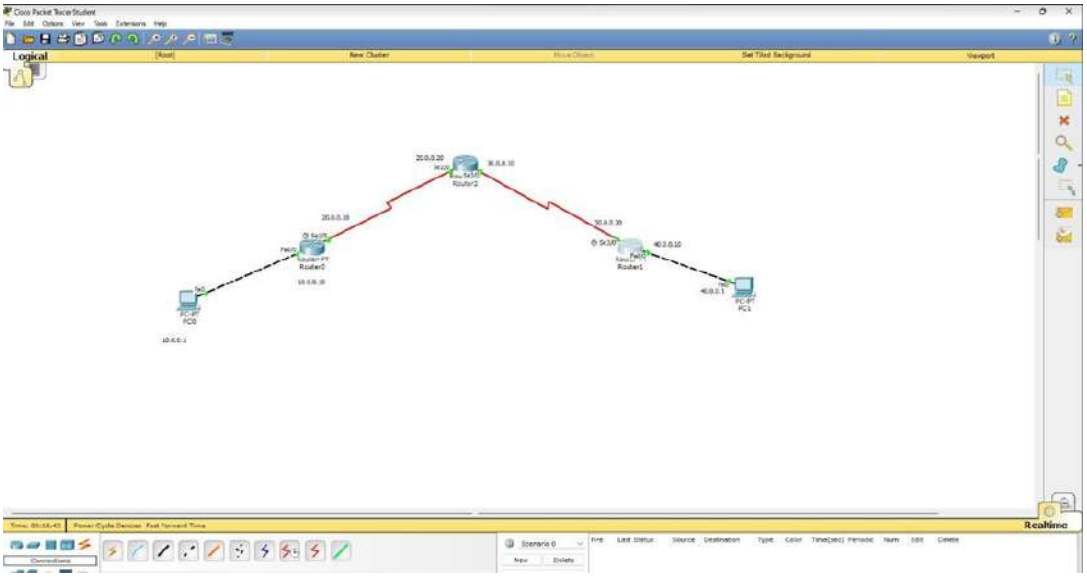
pinging 40.0.0.1 with 32 bytes of data.
Request timed out
Reply from 40.0.0.1: bytes = 32 time = 2ms
TTL = 125
Reply from 40.0.0.1: bytes = 32 time = 2ms
TTL = 125
Reply from 40.0.0.1: bytes = 32 time = 2ms
TTL = 125

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 = 2ms,
Average = 2ms

Observation
In first ping destination host was unreachable as router 0 has no knowledge about the network 30.0.0.0 and 40.0.0.0 and the packets got stuck or lost.
After this it starts is explicitly
Now pinging then is 25% loss in first time, the following ones has no loss.

Output


Logical
Physical
Config
Desktop
Custom Interface
Command Prompt
Router3
PC-PT
PC0
PC-PT
PC1


Logical
Router2
Router0
Router1
PC-PT
PC0
PC-PT
PC1
Realtime

EXPERIMENT 3

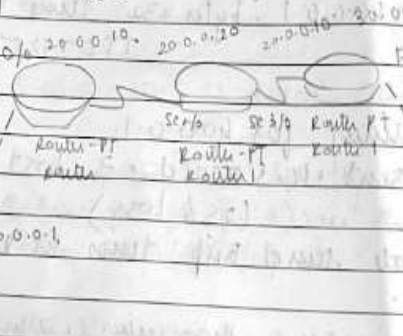
Configure default route, static route to the Router

Observation

2 Configure default route, static route to the router

Aim: to configure default route, static route to Router

Topology:



Procedure:

Follow first 4 steps of expt 2b and create the above topology.

Step 5: Set up IP address of PC0 to 10.0.0.1 and PC2 to 40.0.0.10

Set up Gateway at PC0 to 10.0.0.10 and PC1 to 40.0.0.10

Step 6: Configure IP address of Router0, Router1 and router2 using steps of expt 2B.

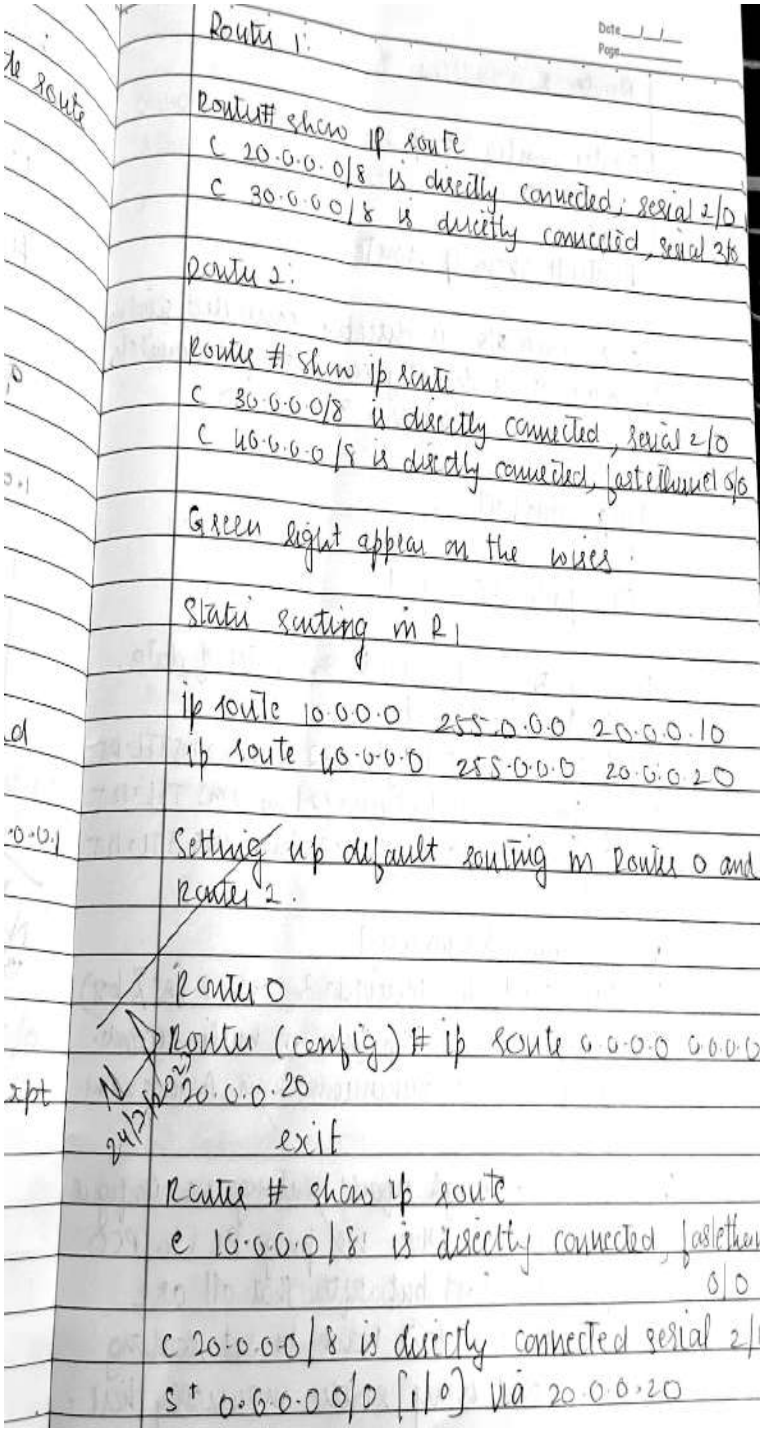
IP route table

Router 0

Router# show ip route

C 10.0.0.0/8 is directly connected to fastethernet

C 20.0.0.0/8 is directly connected serial 2/0



Router 1:

Router# show ip route
C 20.0.0.0/8 is directly connected, serial 2/0
C 30.0.0.0/8 is directly connected, serial 3/0

Router 2:

Router # show ip route
C 30.0.0.0/8 is directly connected, serial 2/0
C 40.0.0.0/8 is directly connected, fastethernet 0/0

Green light appear on the wires.

Static routing in R1

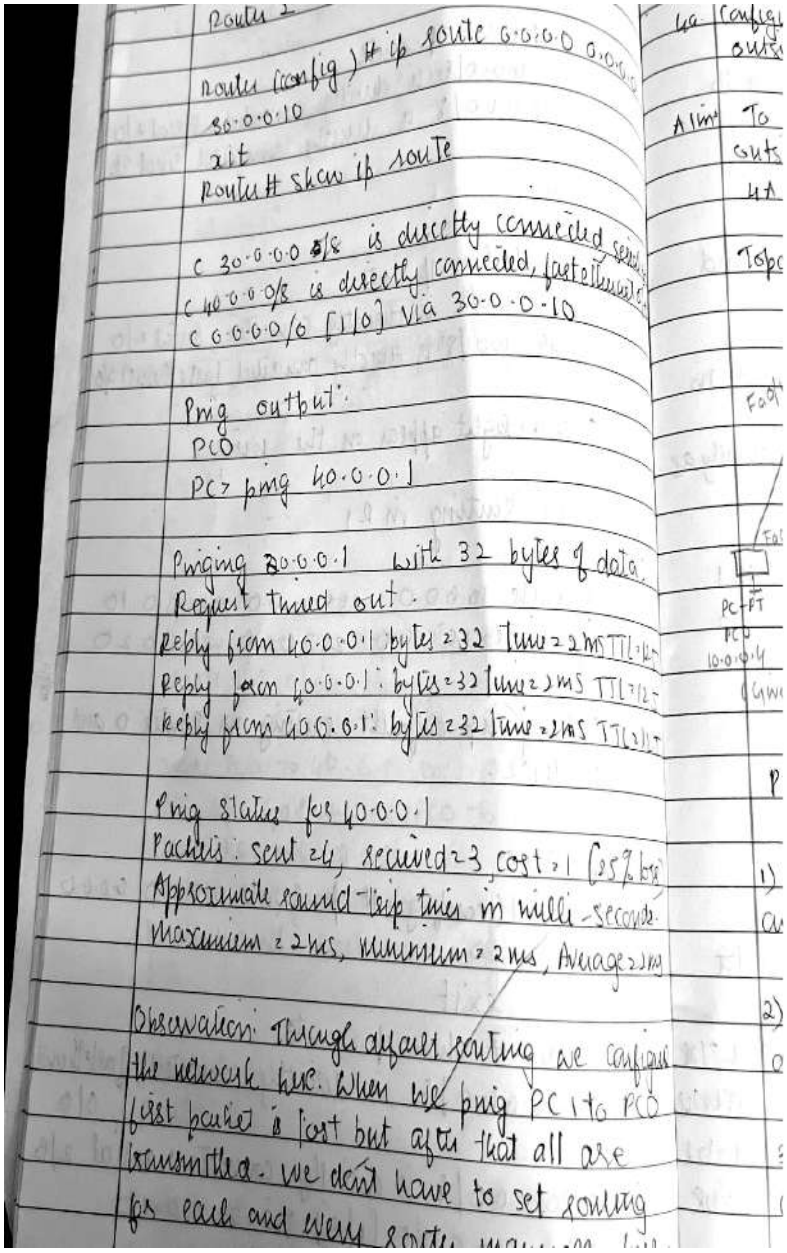
ip route 10.0.0.0 255.0.0.0 20.0.0.10
ip route 40.0.0.0 255.0.0.0 20.0.0.20

Setting up default routing in Router 0 and router 2.

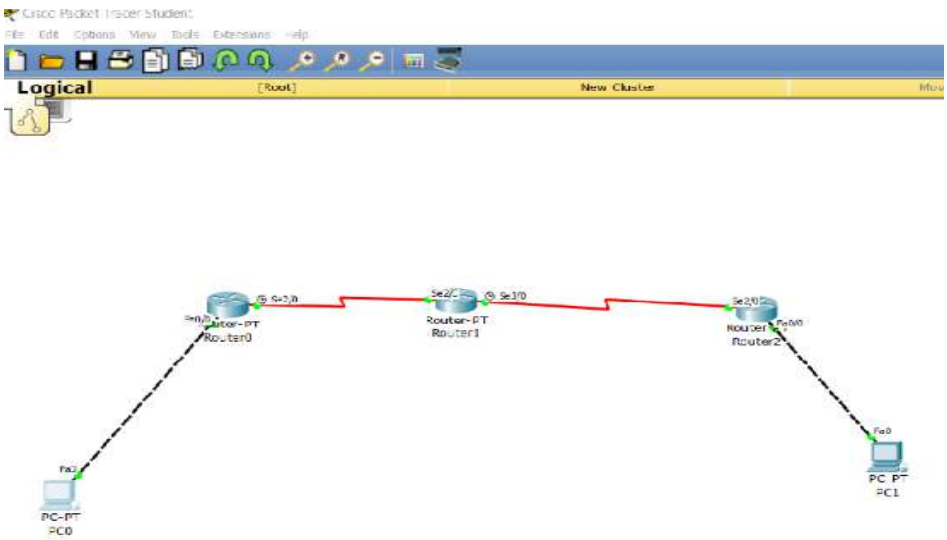
Router 0

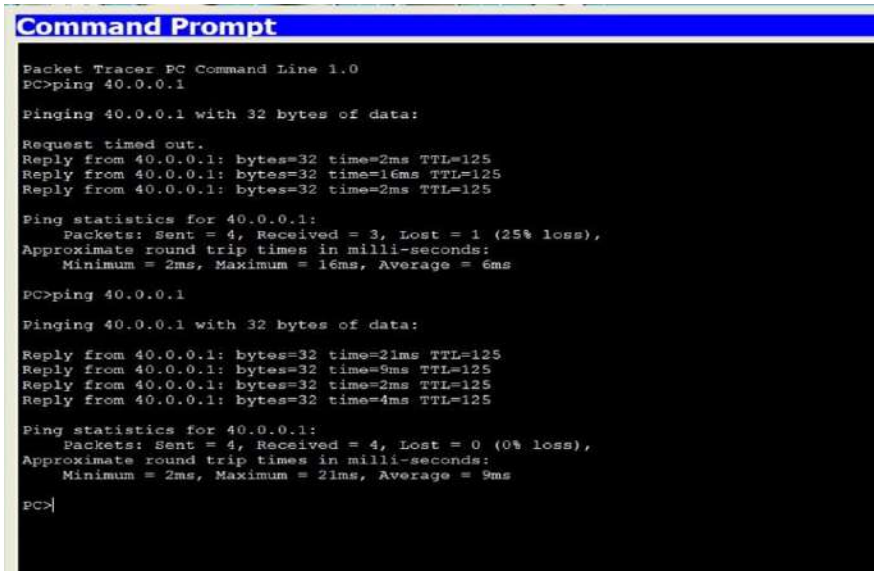
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.20
exit

Router # show ip route
C 10.0.0.0/8 is directly connected, fastethernet 0/0
C 20.0.0.0/8 is directly connected serial 2/
S* 0.0.0.0/0 [1/0] via 20.0.0.20


Router
Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.10
exit
Router# show ip route
C 30.0.0.0/8 is directly connected, se
C 40.0.0.0/8 is directly connected, fastEthernet
C 0.0.0.0/0 [1/0] via 30.0.0.10
Ping output:
PC0
PC> ping 40.0.0.1
Pinging 30.0.0.1 with 32 bytes of data.
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=12
Reply from 40.0.0.1: bytes=32 time=2ms TTL=12
Reply from 40.0.0.1: bytes=32 time=2ms TTL=12
Ping statistics for 40.0.0.1
Packets: sent=4, received=3, cost=1 (25% loss)
Approximate round trip time in milli-seconds:
Maximum=2ms, minimum=2ms, Average=2ms
Observation: Through default routing we configure the network here. When we ping PC1 to PC0 first packet is lost but after that all are transmitted. we don't have to set routing for each and every
PC-FT
PC0
10.0.0.4

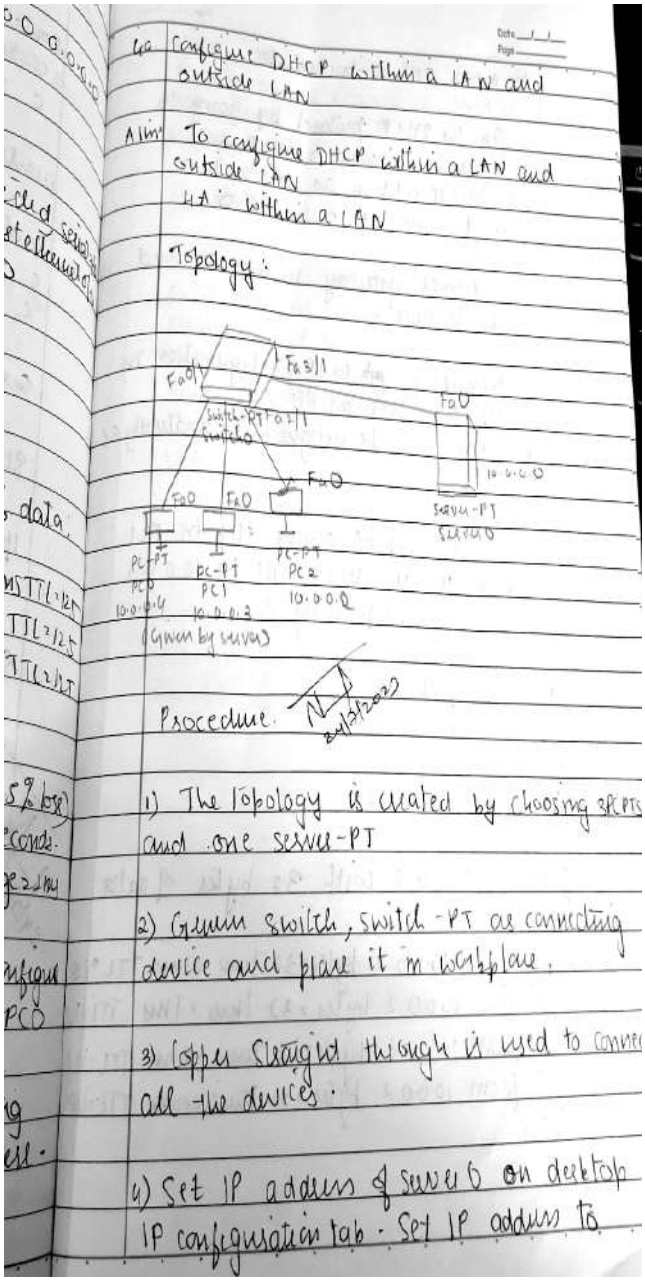
Output


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Router-PT
Router0
Router-PT
Router1
Router2
PC-PT
PC0
PC-PT
PC1


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>

EXPERIMENT 4

Configure DHCP within a LAN and outside LAN.



4a Configure DHCP within a LAN and outside LAN

Aim: To configure DHCP within a LAN and outside LAN

4A: within a LAN

Topology:

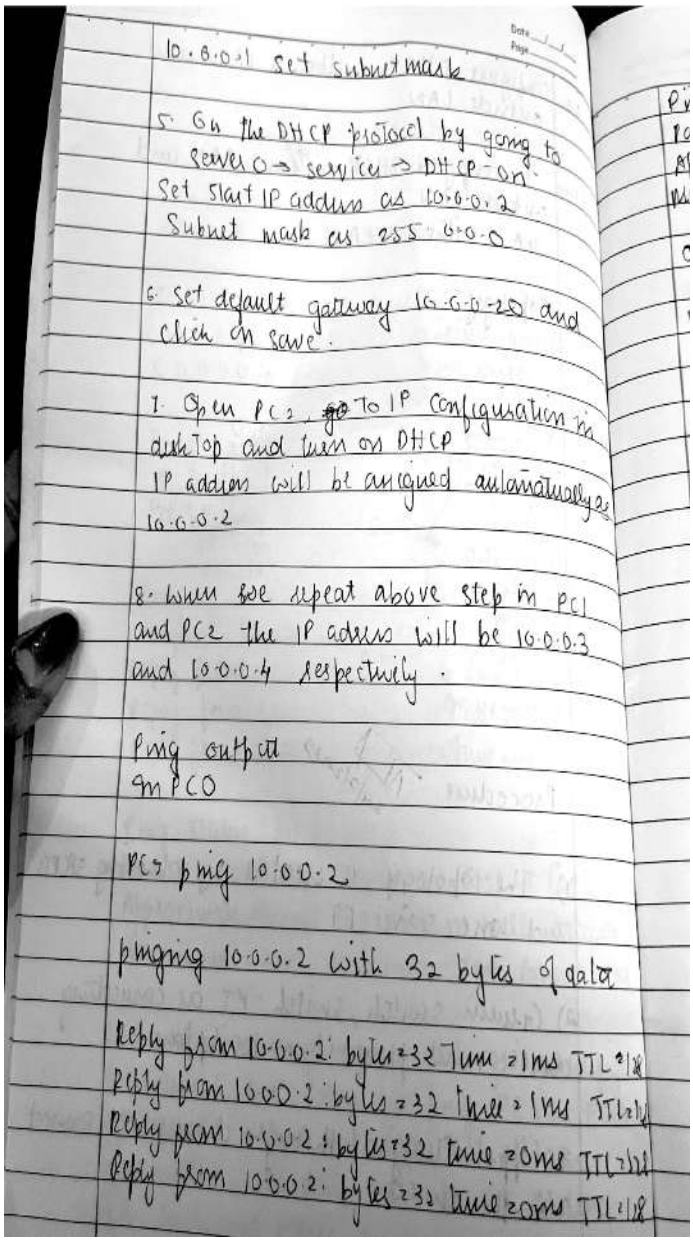
Procedure:

1) The topology is created by choosing 3PCPTs and one server-PT

2) Genuin switch, switch-PT as connecting device and place it in workplace.

3) Copper Straight through is used to connect all the devices.

4) Set IP address of server0 on desktop IP configuration tab. Set IP address to



10.0.0.1 set subnet mask

5. On the DHCP protocol by going to Server 0 → service → DHCP - on. Set start IP addres as 10.0.0.2 Subnet mask as 255.0.0.0

6. Set default gateway 10.0.0.20 and click on save.

7. Open PC2, go to IP configuration in desktop and turn on DHCP IP address will be assigned automatically as 10.0.0.2

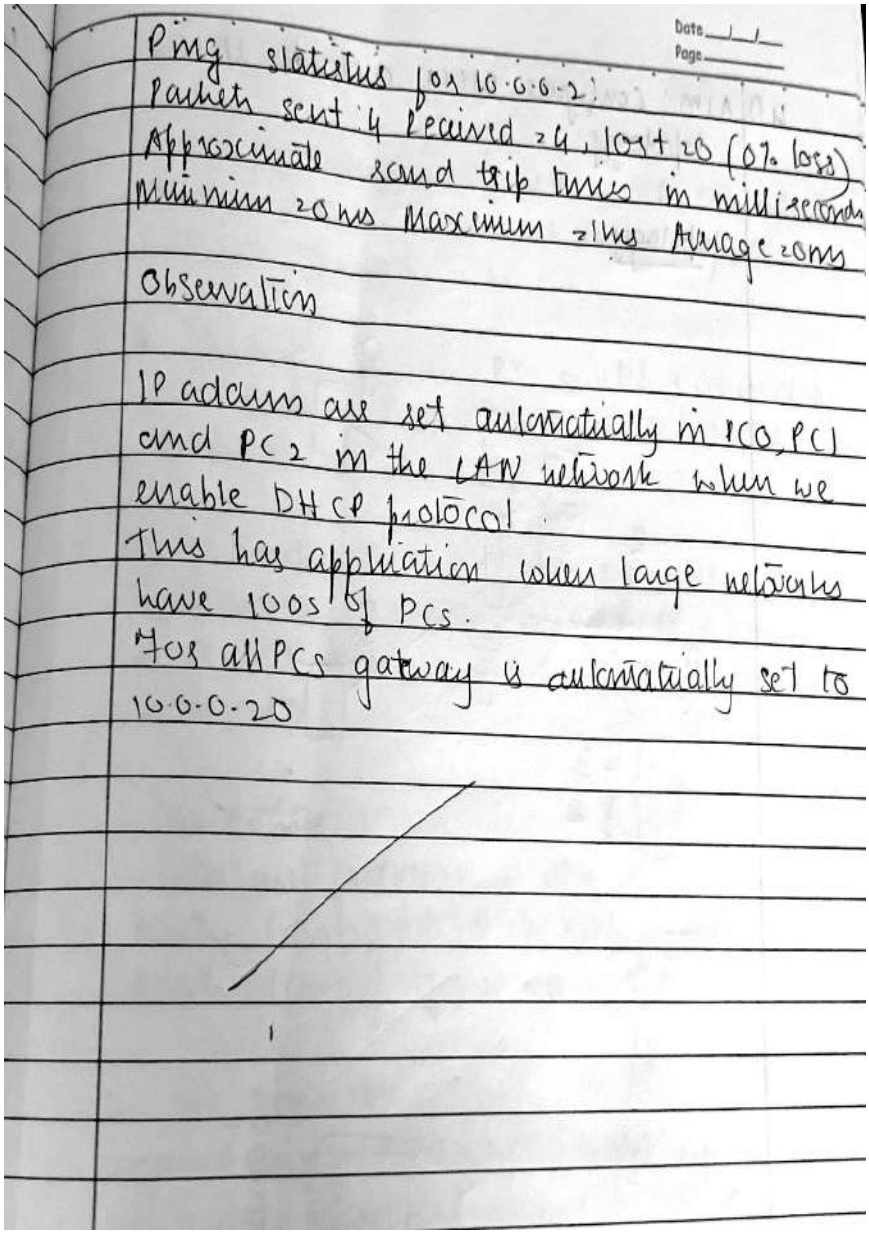
8. Now we repeat above step in PC1 and PC2 the IP adress will be 10.0.0.3 and 10.0.0.4 respectively.

Ping output on PC0

PC> ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data

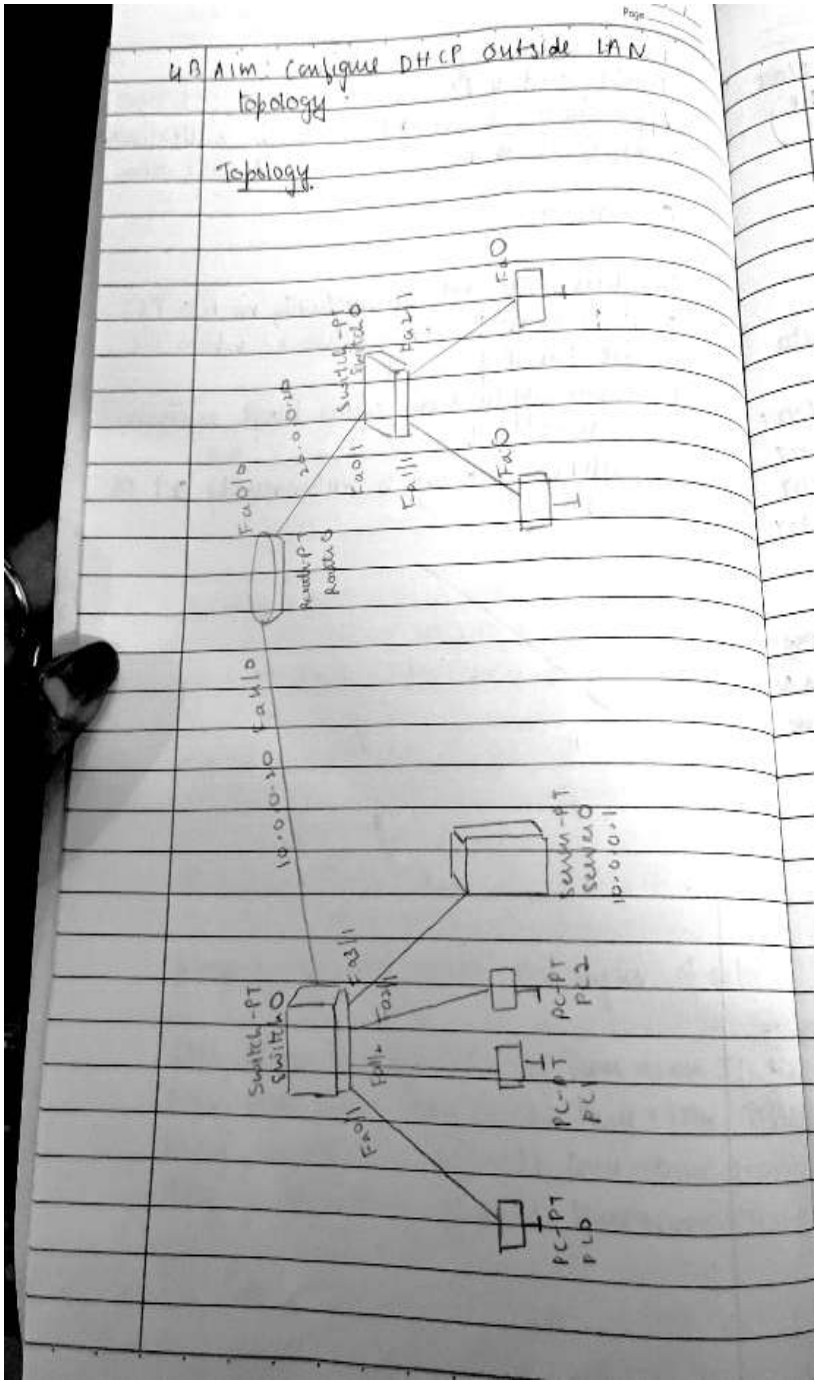
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

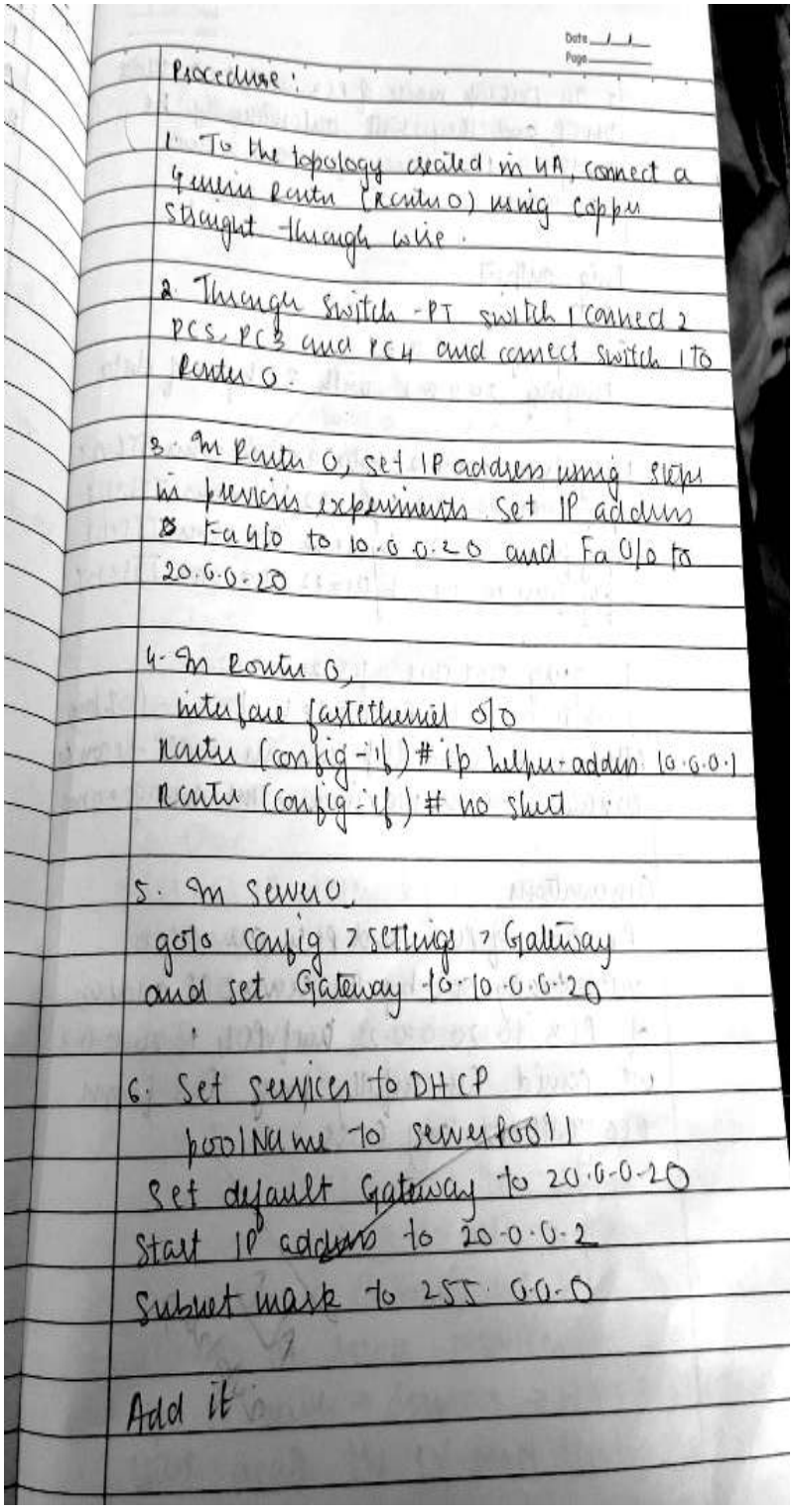

Date / /
Page

Ping statistics for 10.0.0.2:
Packets sent: 4 Received = 4, lost = 0 (0% loss)
Approximate round trip times in milliseconds
minimum = 0ms Maximum = 1ms Average = 0ms

Observation

IP adams are set automatically in PC0, PC1 and PC2 in the LAN network when we enable DHCP protocol.
This has application when large networks have 100s of PCs.
Thus all PCs gateway is automatically set to 10.0.0.20


4B Aim: Configure DHCP outside LAN
topology
Topology
Fa0/0
10.0.0.10
Fa0/1
Router-PT
Router0
Fa0/1
Fa0/0
Switch-PT
Switch0
Fa2/1
Fa0/1
Fa0/2
Fa0/3
Fa0/4
Server-PT
Server0
10.0.0.1
PC-PT
PC2
PC-PT
PC1
PC-PT
PC0



Procedure:

1. To the topology created in 4A, connect a Generic Router (Router 0) using copper Straight-through wire.

2. Through Switch-PT switch 1 connect 2 PC3, PC3 and PC4 and connect Switch 1 To Router 0.

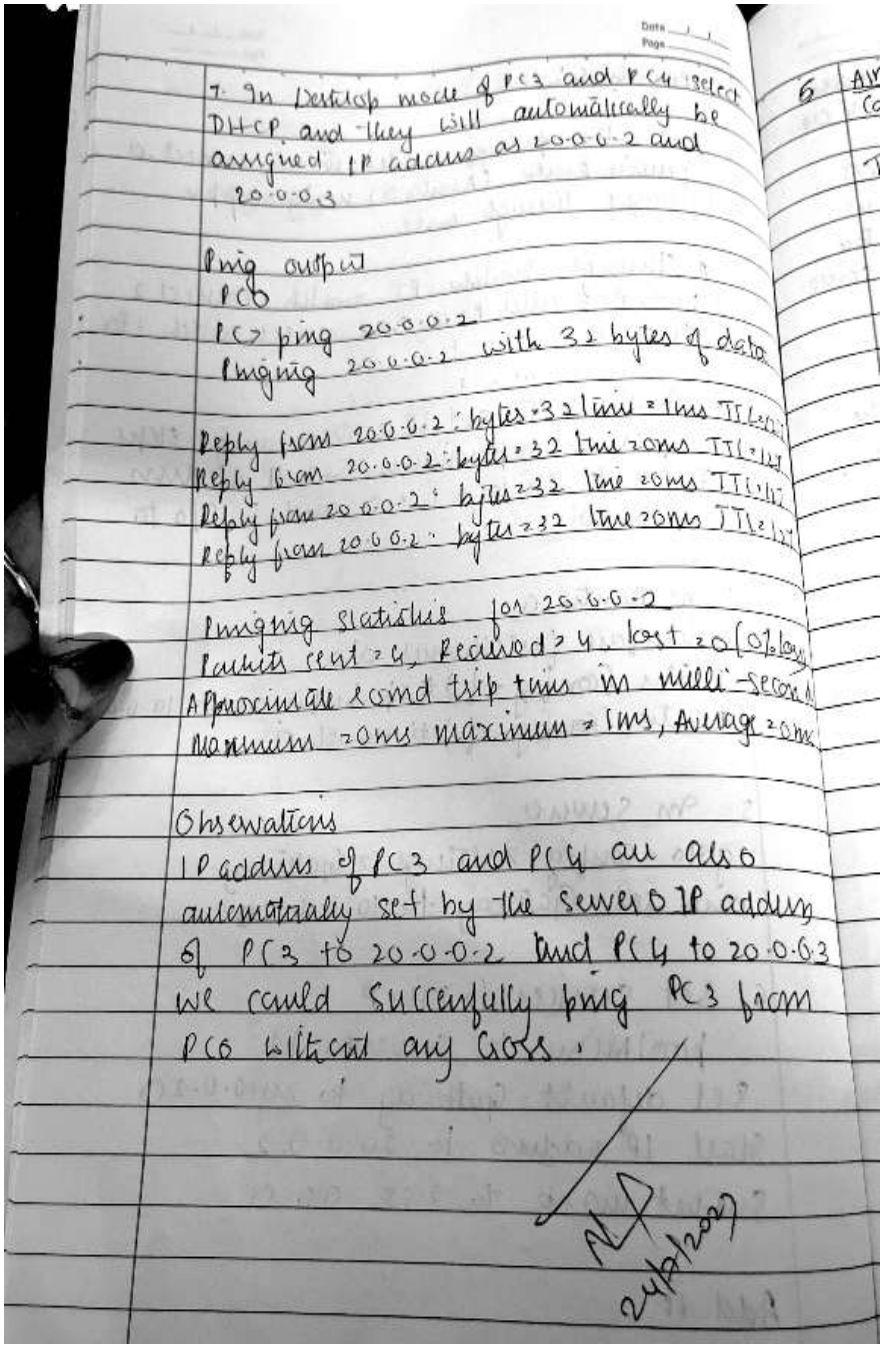
3. In Router 0, set IP address using steps in previous experiments. Set IP address Fa 4/0 to 10.0.0.20 and Fa 0/0 to 20.0.0.20

4. In Router 0,
interface fastethernet 0/0
Router (config-if)# ip helper-address 10.0.0.1
Router (config-if)# no shut

5. In Server 0,
goto config > settings > Gateway
and set Gateway to 10.10.0.0.20

6. Set services to DHCP
poolName to serverPool1
Set default Gateway to 20.0.0.20
Start IP address to 20.0.0.2
Subnet mask to 255.0.0.0

Add it



7. In Desktop mode of PC3 and PC4 select DHCP, and they will automatically be assigned IP address as 20.0.0.2 and 20.0.0.3

Ping output
PC0
PC> ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data

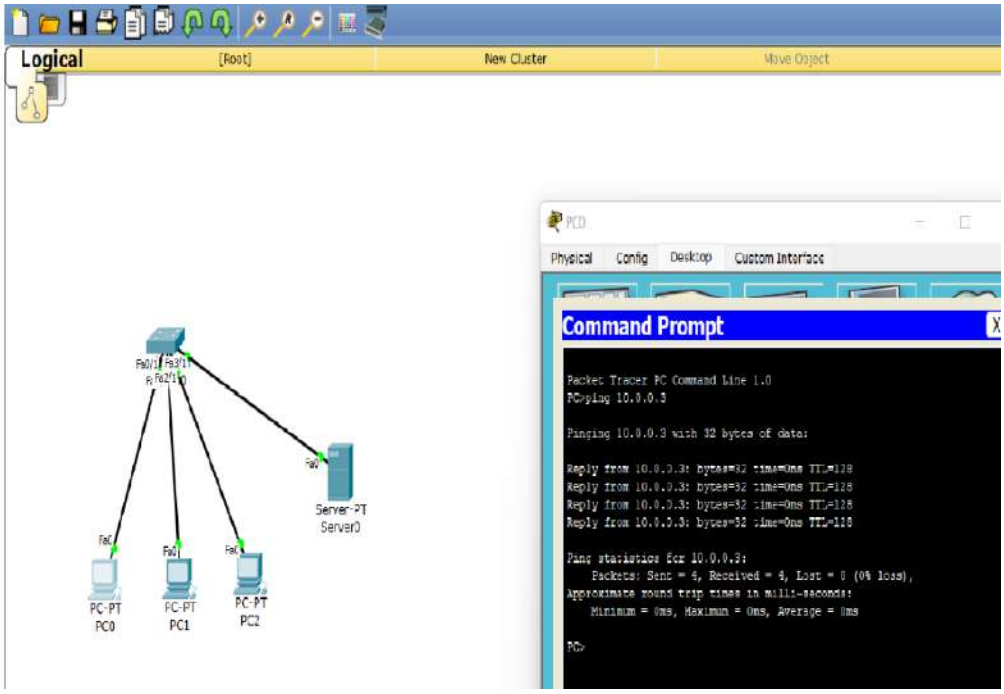
Reply from 20.0.0.2: bytes=32 time=1ms TTL=128
Reply from 20.0.0.2: bytes=32 time=0ms TTL=128
Reply from 20.0.0.2: bytes=32 time=0ms TTL=128
Reply from 20.0.0.2: bytes=32 time=0ms TTL=128

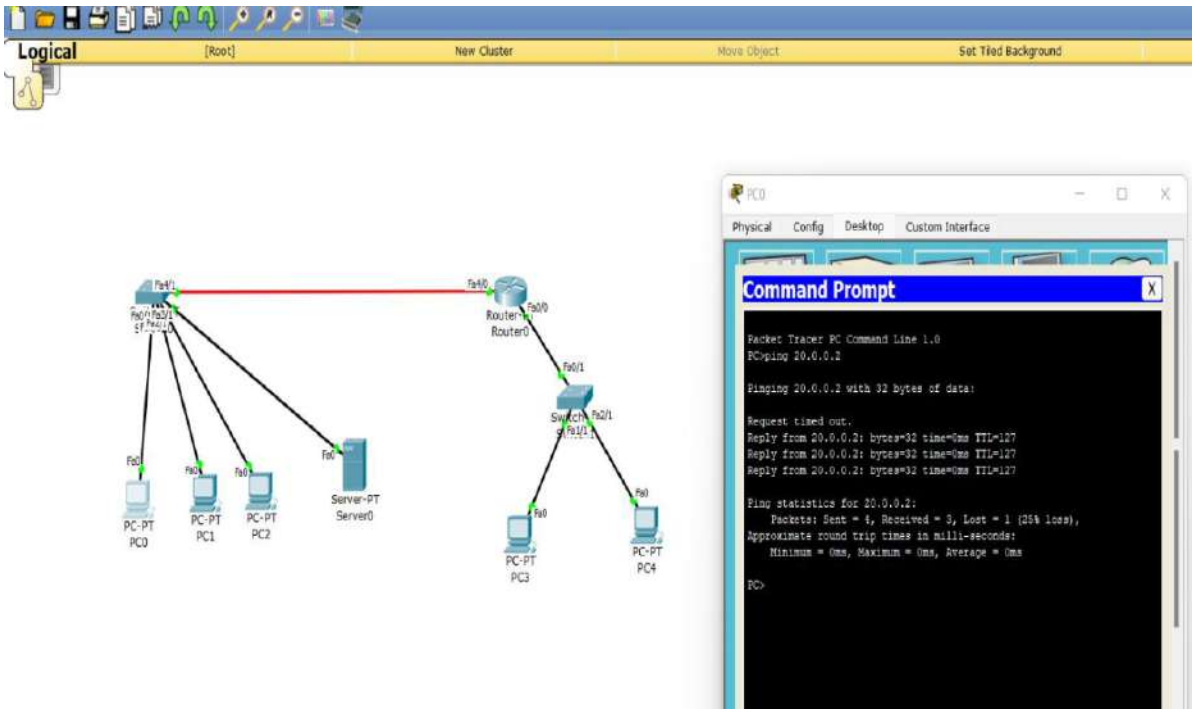
Pinging statistics for 20.0.0.2
Packets sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Observations
IP address of PC3 and PC4 are also automatically set by the server. IP address of PC3 to 20.0.0.2 and PC4 to 20.0.0.3 we could successfully ping PC3 from PC0 without any loss.

24/9/2022

Output

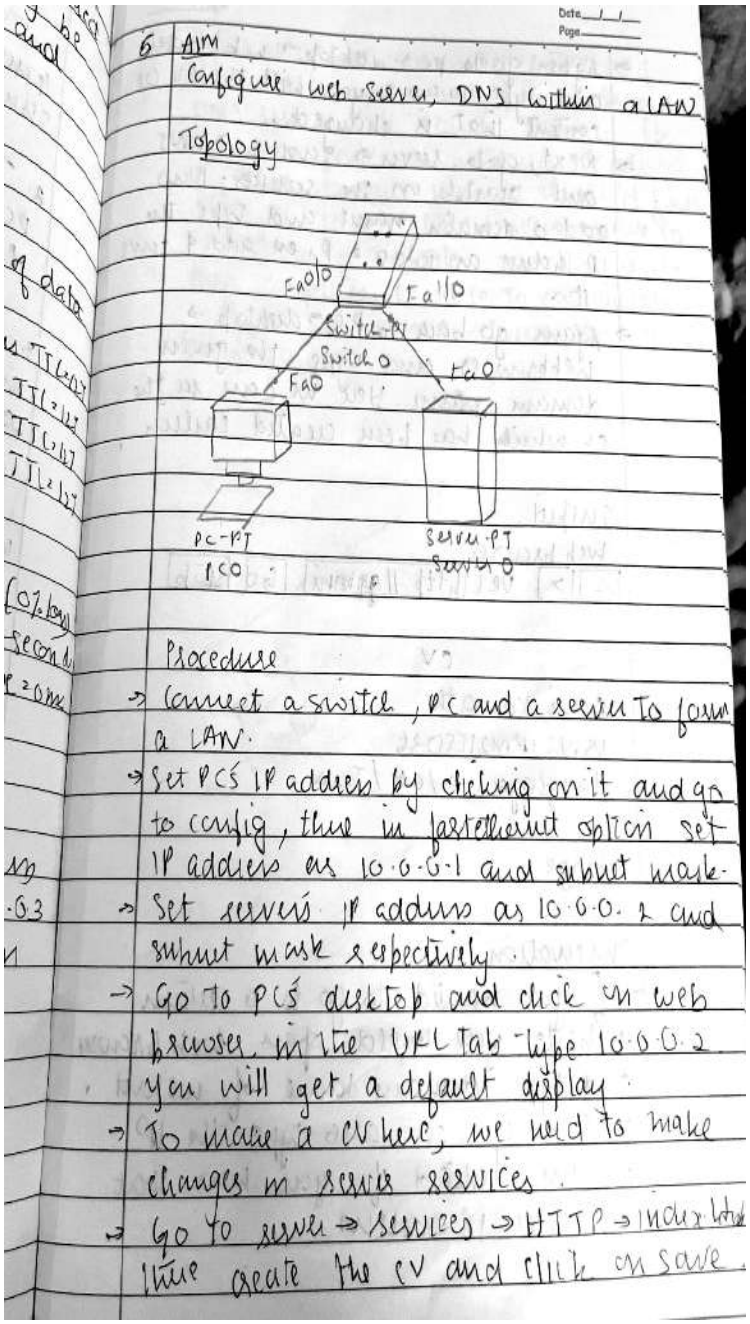

Logical
[Root]
New Cluster
Move Object
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=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 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
Server-PT
Server0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2

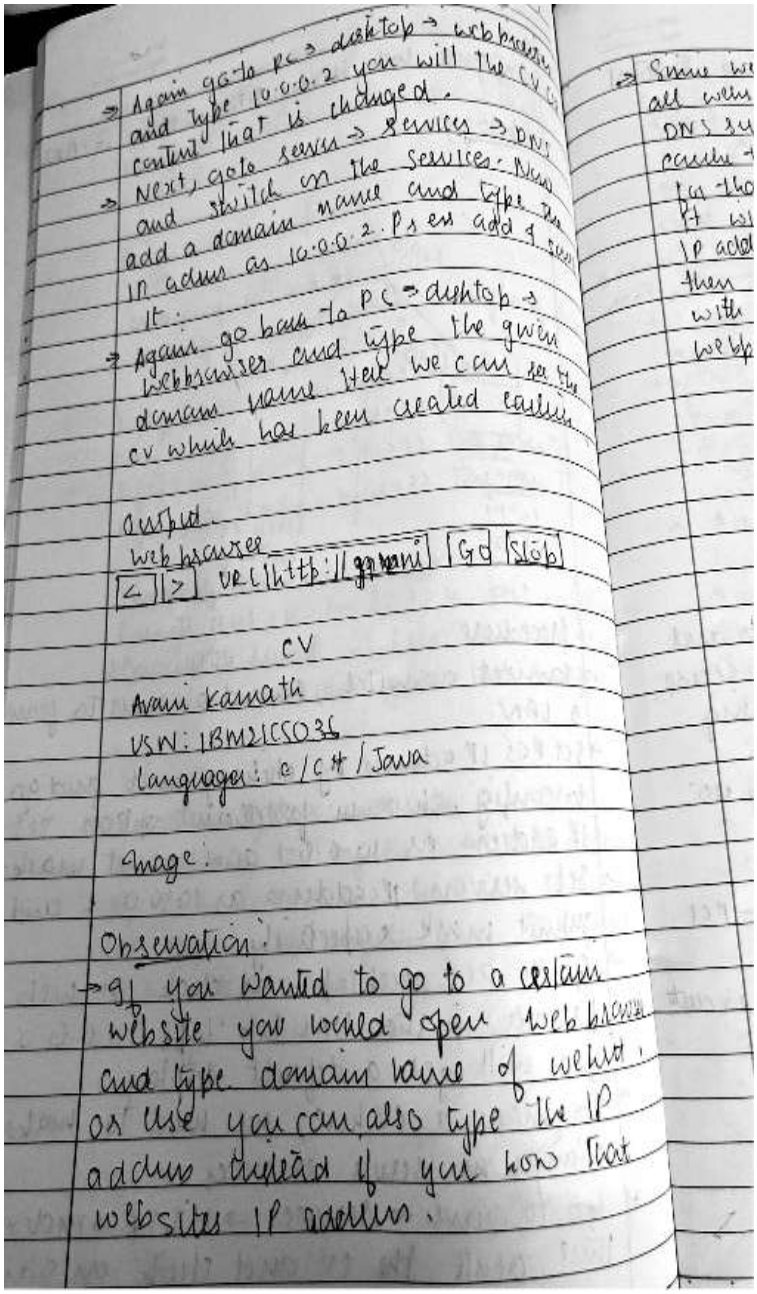

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 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>
Router-PT
Router0
Server-PT
Server0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4

EXPERIMENT 5

Configure Web Server, DNS within a LAN

observation





→ Again go to PC → desktop → web browser and type 10.0.0.2 you will the CV content that is changed.
→ Next, goto server → services → DNS and switch on the services. Now add a domain name and type the IP address as 10.0.0.2. Press add to save it.
→ Again go back to PC → desktop → web browser and type the given domain name. Here we can see the CV which has been created earlier.

Output

Web browser

URL http://avani [Go] [Stop]

CV

Avani Kamath

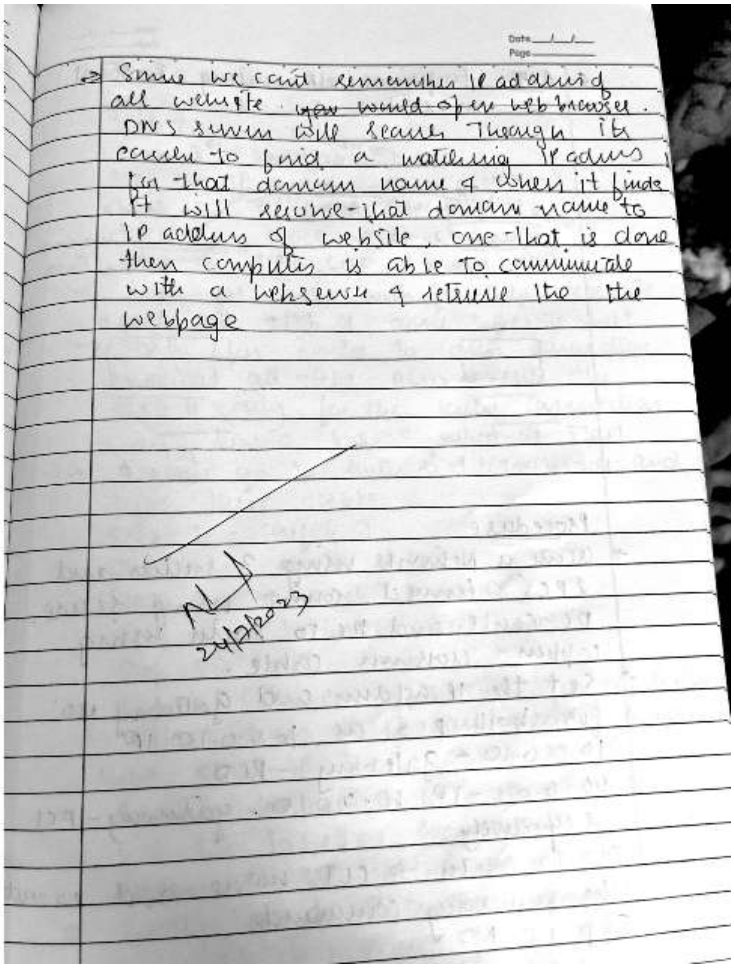
USN: 1BM21CS036

Languages: C / C# / Java

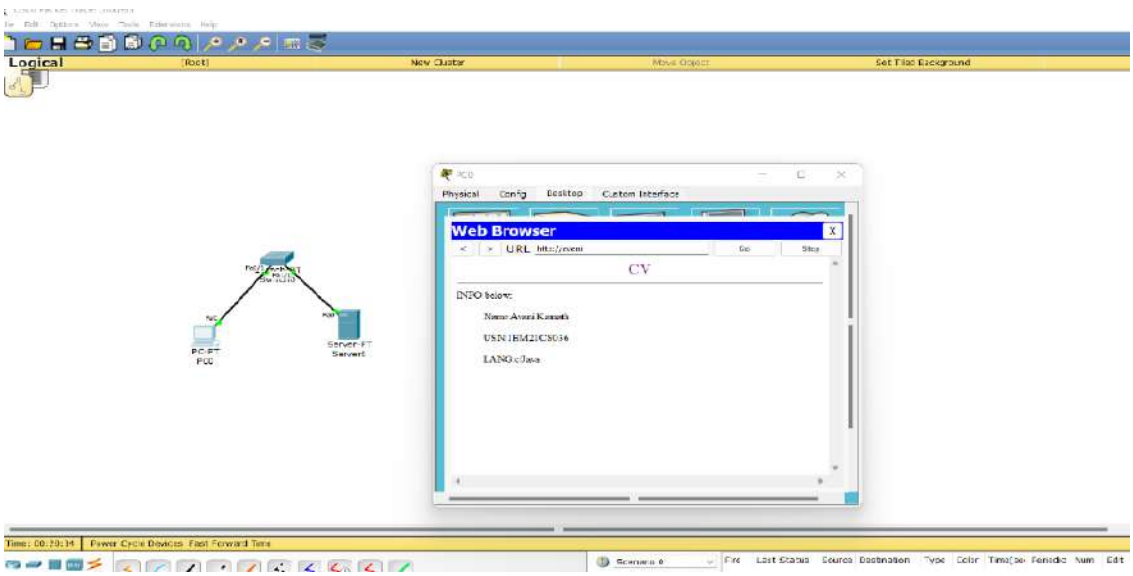
Image:

Observation:

→ If you wanted to go to a certain website you would open web browser and type domain name of website. Or else you can also type the IP address directly if you know that website IP address.


Since we can't remember IP address of all website we would open web browser.
DNS server will search through its records to find a matching IP address for that domain name & when it finds it will resolve that domain name to IP address of website. Once that is done then computer is able to communicate with a webserver & retrieve the the webpage
24/12/2023

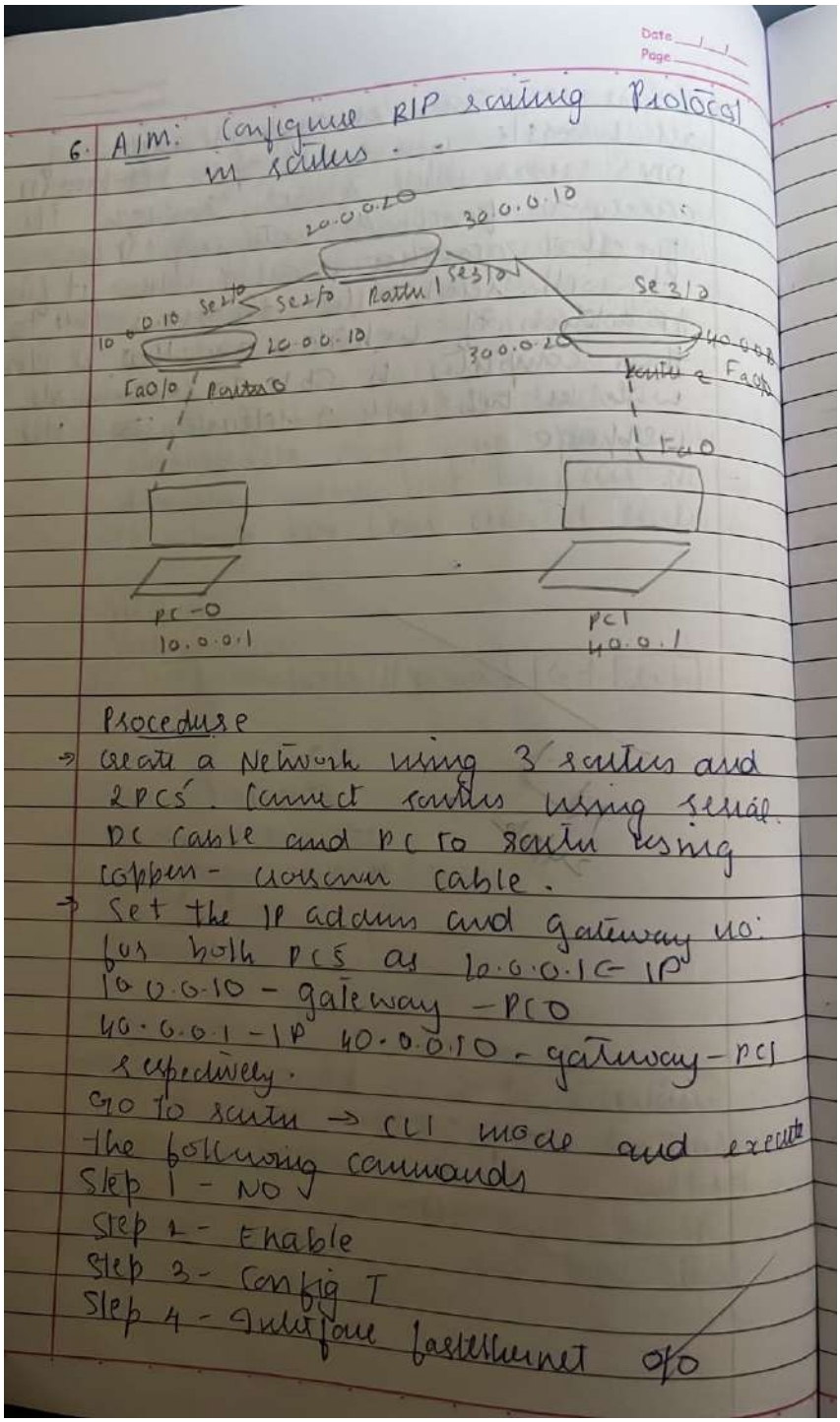
Output

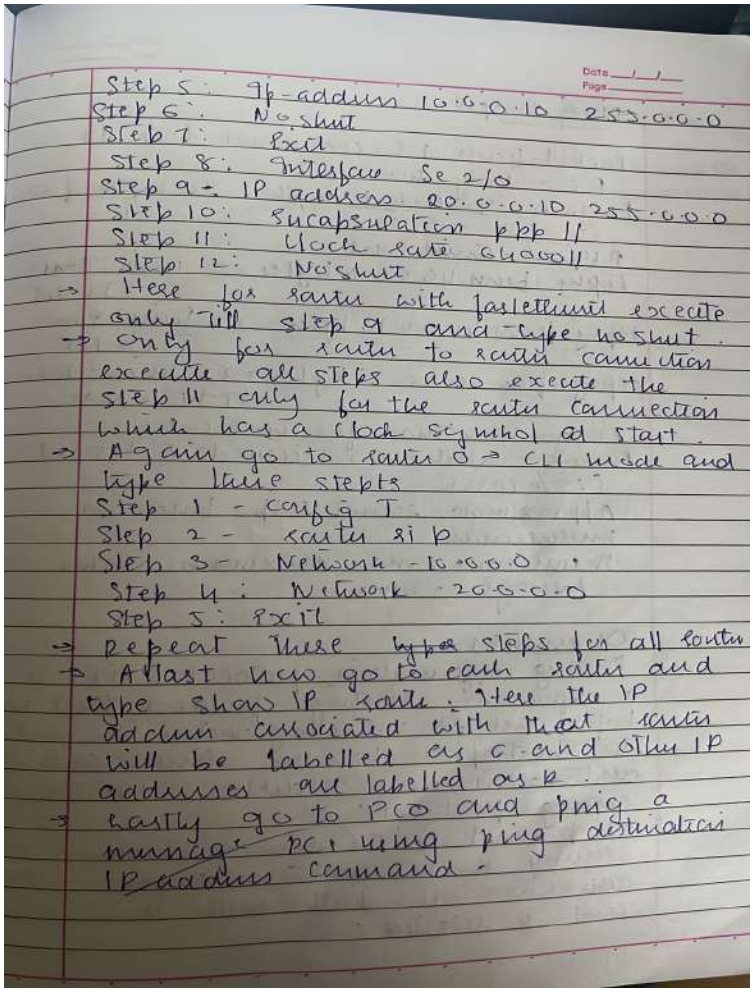

Logical
Web Browser
CV
INFO below:
Name:Avani Kamath
USN:1BM21CS036
LANG:Java

EXPERIMENT 6

Configure RIP routing Protocol in Routers

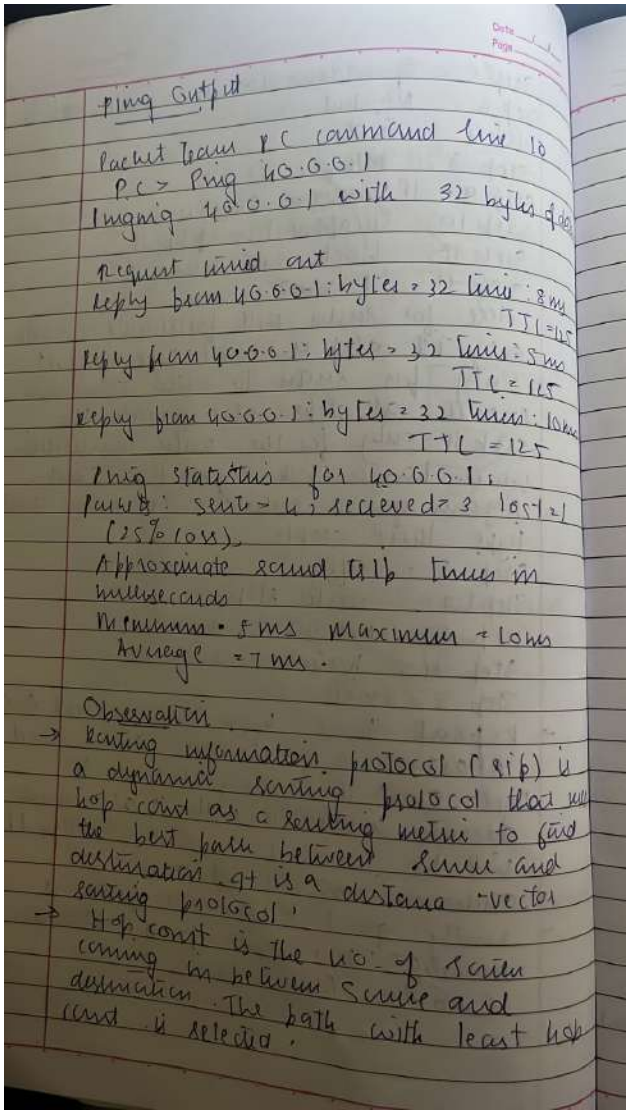
Observation





Step 5: ip-addrss 10.0.0.10 255.0.0.0
Step 6: No shut
Step 7: Exit
Step 8: Interface Se 2/0
Step 9: IP address 20.0.0.10 255.0.0.0
Step 10: Encapsulation ppp ||
Step 11: Clock rate 64000||
Step 12: No shut

→ Here for router with fastethernet execute only till step 9 and type no shut.
→ only for router to router connection execute all steps also execute the step 11 only for the router connection which has a clock symbol at start.
→ Again go to router 0 → CLI mode and type these steps
Step 1 - config T
Step 2 - router rip
Step 3 - Network - 10.0.0.0
Step 4: Network - 20.0.0.0
Step 5: exit
→ Repeat these ~~type~~ steps for all router
→ At last now go to each router and type show IP route. Here the IP address associated with that router will be labelled as C and other IP addresses are labelled as R.
→ Lastly go to PC0 and ping a message PC1 using ping destination IP address command.



Ping Output

Packet Tracer PC command line 10
PC > Ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:

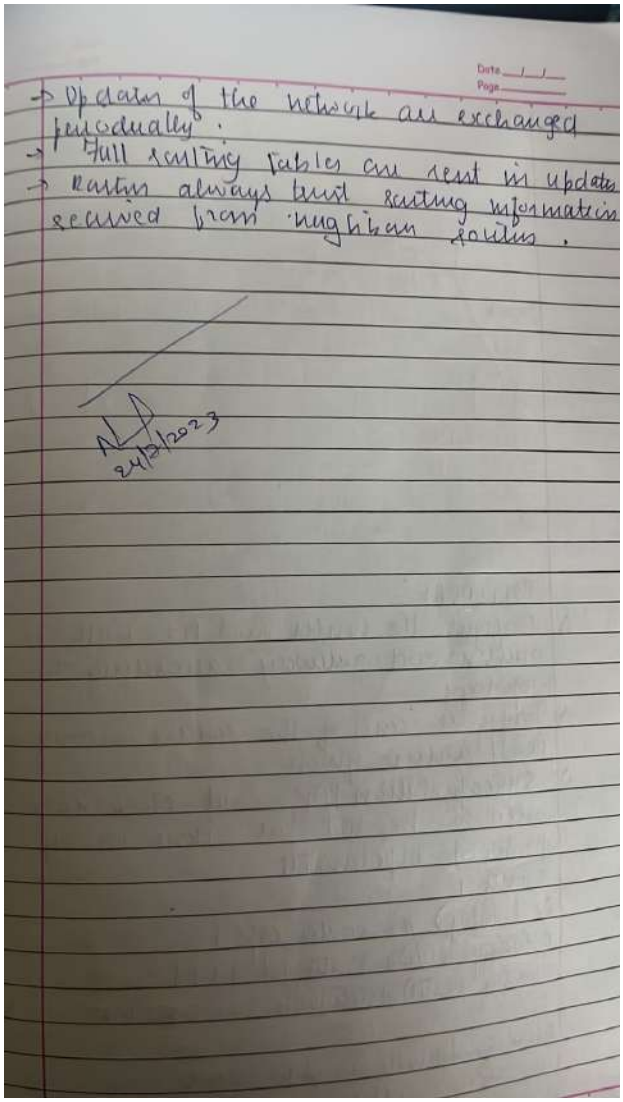
Request timed out
Reply from 40.0.0.1: bytes = 32 time: 8ms
TTL = 125
Reply from 40.0.0.1: bytes = 32 time: 5ms
TTL = 125
Reply from 40.0.0.1: bytes = 32 time: 10ms
TTL = 125

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

Observation

→ Routing information protocol (RIP) is a dynamic routing protocol that uses hop count as a routing metric to find the best path between source and destination. It is a distance vector routing protocol.

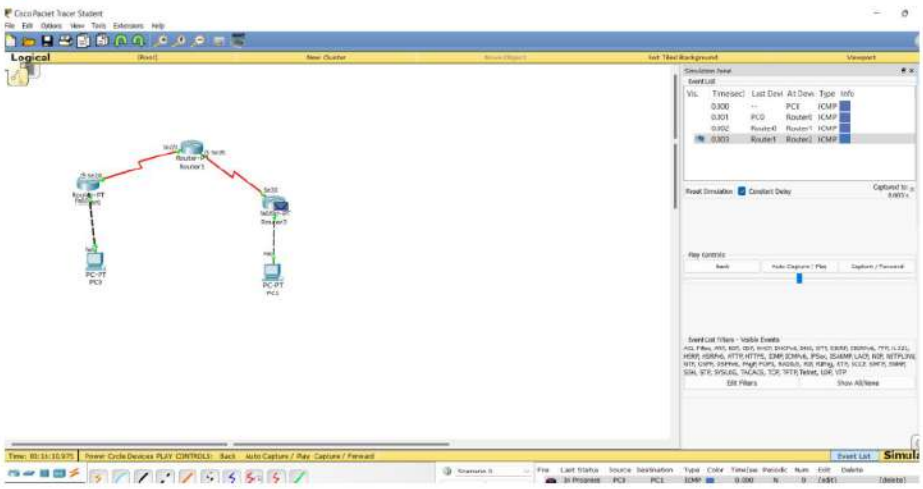
→ Hop count is the no. of routers coming in between source and destination. The path with least hop count is selected.


Date
Page
→ Updates of the network are exchanged periodically.
→ Full routing tables are sent in updates
→ Routers always trust routing information received from neighbour routers.
24/12/2023

Output

Command Prompt

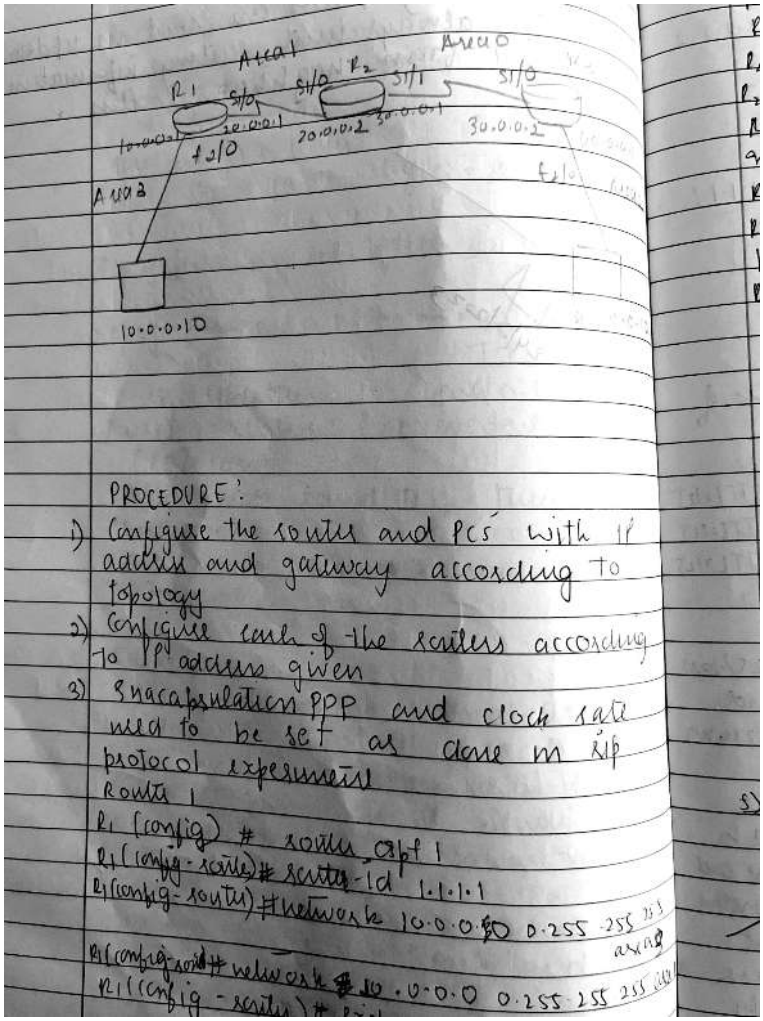
```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 14ms, Average = 10ms

PC>
```

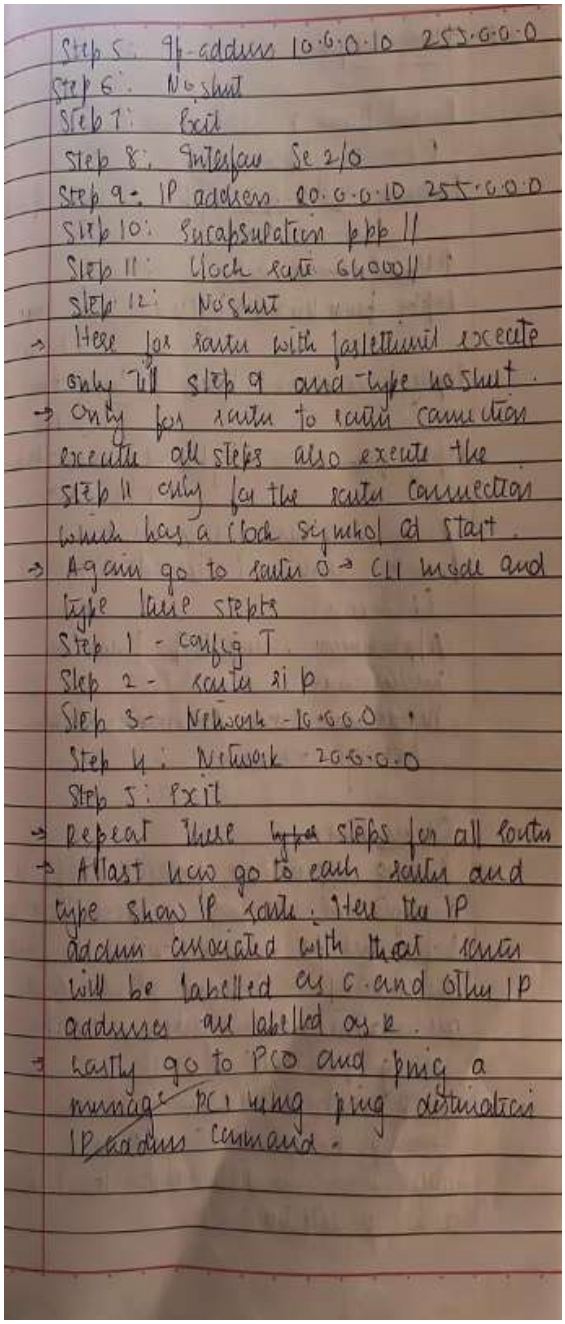


EXPERIMENT 7

Configure OSPF routing protocol

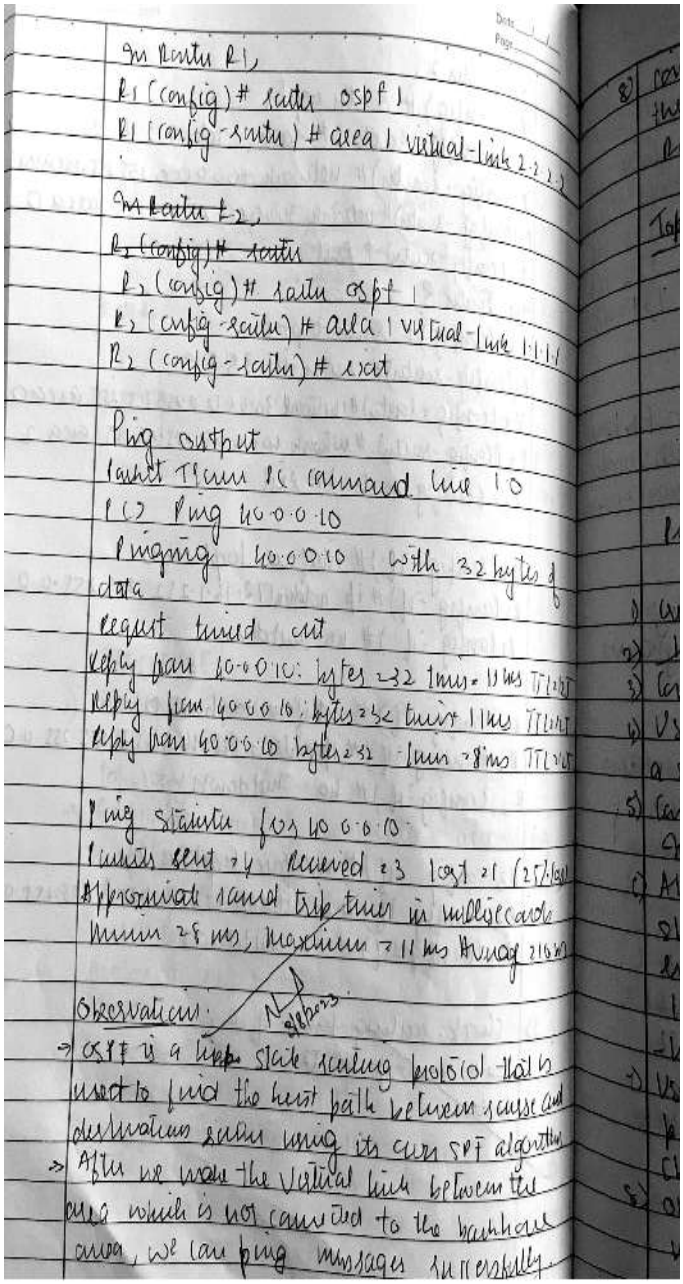
Observation



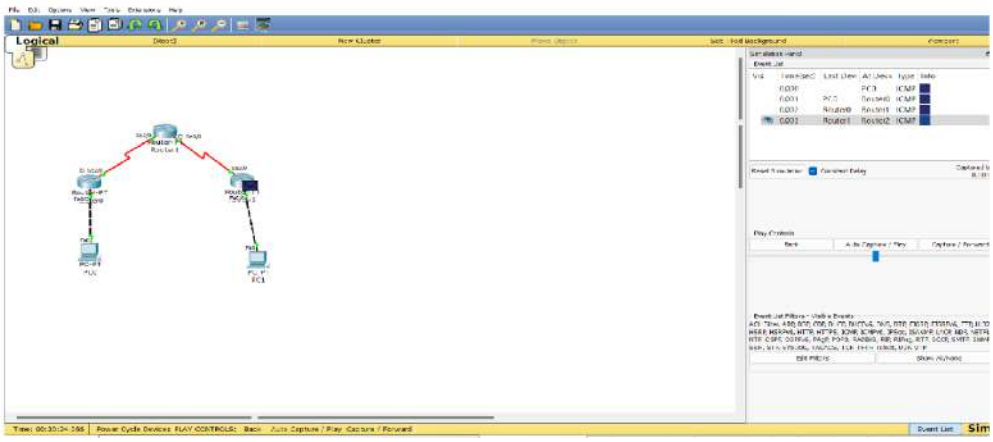


Step 5: ip address 10.0.0.10 255.0.0.0
Step 6: No shut
Step 7: Exit
Step 8: Interface Se 2/0
Step 9: IP address 20.0.0.10 255.0.0.0
Step 10: Encapsulation ppp //
Step 11: Clock rate 64000 //
Step 12: No shut

→ Here for router with fastethernet execute only till step 9 and type no shut.
→ Only for router to router connection execute all steps also execute the step 11 only for the router connection which has a clock symbol at start.
→ Again go to router 0 → CLI mode and type these steps
Step 1 - config T
Step 2 - router rip
Step 3 - Network 10.0.0.0
Step 4: Network 20.0.0.0
Step 5: Exit
→ Repeat these ~~type~~ steps for all routers
→ Atlast now go to each router and type show IP route. Here the IP address associated with that router will be labelled as C and other IP addresses are labelled as R.
→ Lastly go to PC0 and ping a running PC using ping destination IP address command.


On Router R1,
R1(config)# router ospf 1
R1(config-router)# area 1 virtual-link 2.2.2.2
On Router R2,
R2(config)# router
R2(config)# router ospf 1
R2(config-router)# area 1 virtual-link 1.1.1.1
R2(config-router)# exit
Ping output
Router Prom PC command line 10
PC> Ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data
request timed out
Reply from 40.0.0.10: bytes=32 time=11ms TTL=125
Reply from 40.0.0.10: bytes=32 time=11ms TTL=125
Reply from 40.0.0.10 bytes=32 time=8ms TTL=125
Ping statistics for 40.0.0.10
Packets sent = 4 Received = 3 Lost = 1 (25% loss)
Approximate round trip time in milliseconds
Minimum = 8ms, Maximum = 11ms Average = 10ms
Observation:
OSPF is a link state routing protocol that is used to find the best path between source and destination router using its own SPF algorithm
After we make the virtual link between the area which is not connected to the backbone area, we can ping messages successfully.

Output

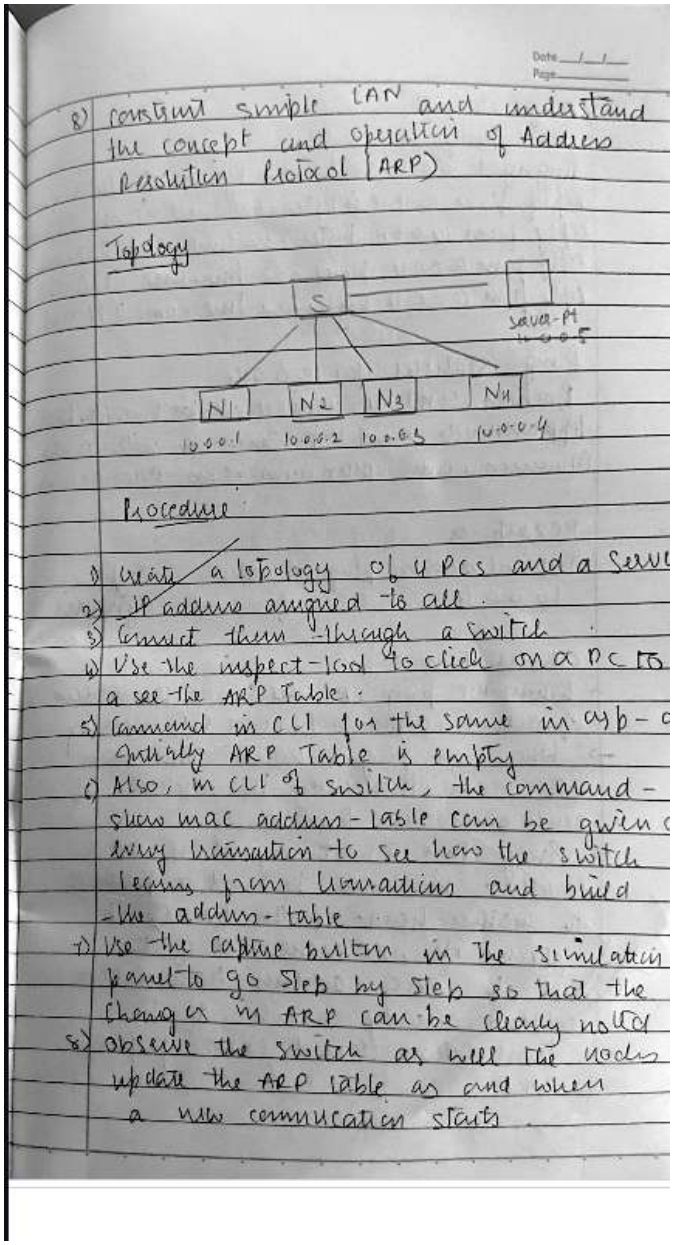


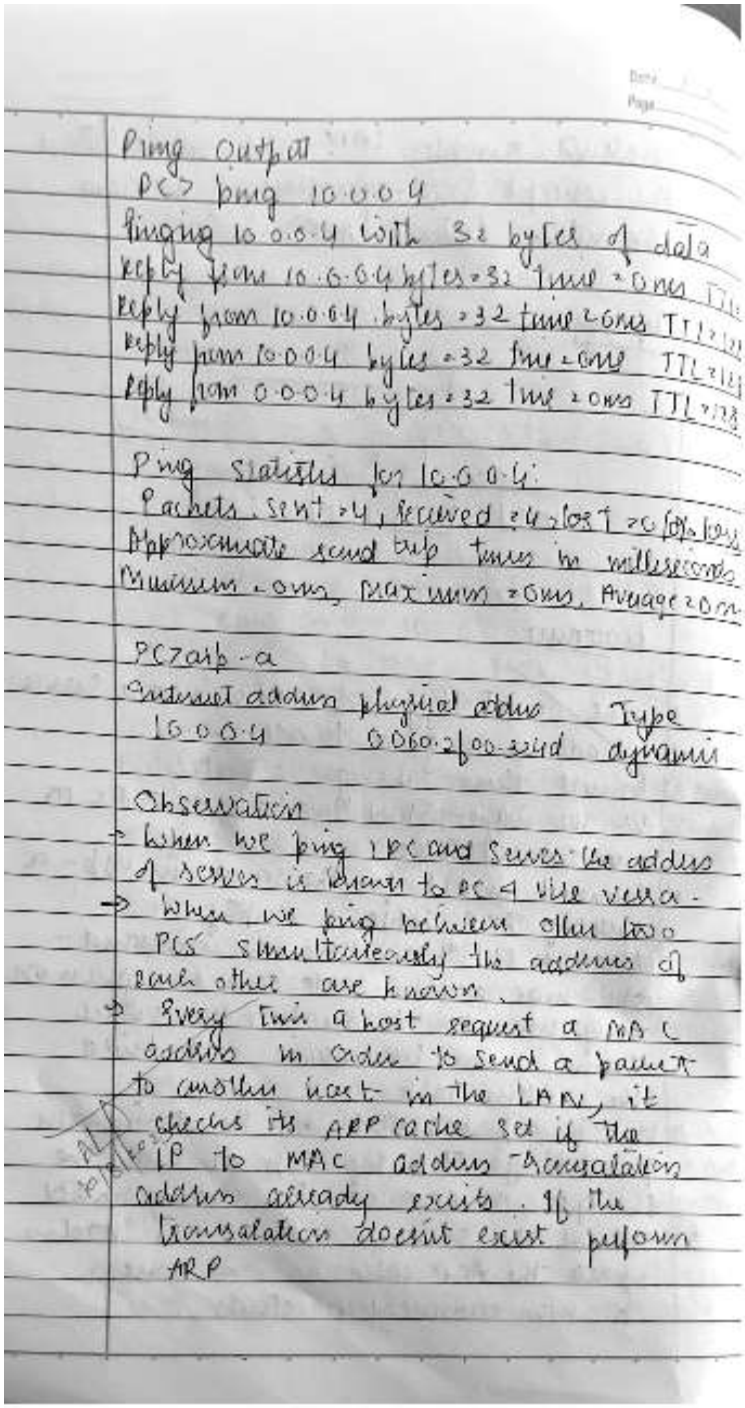
```
Command Prompt

Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=4ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=12ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 12ms, Average = 7ms

PC>
```

EXPERIMENT 8

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

Observation





Ping output

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=

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

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

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

Ping statistics for 10.0.0.4:

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

Approximate round trip times in milliseconds:

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

PC>arp -a

Internet address physical address Type

10.0.0.4 0060.2f00.3e3d dynamic

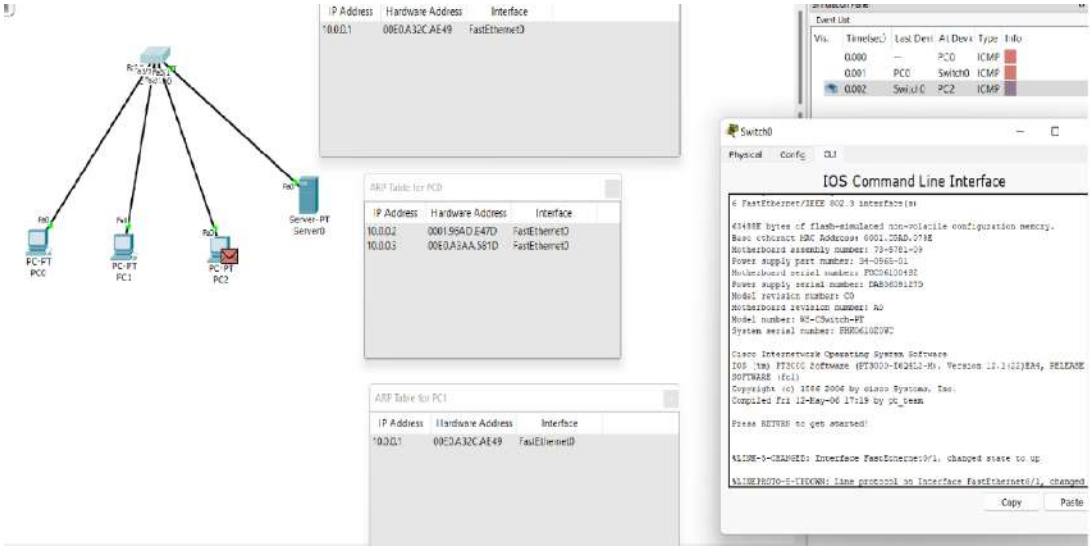
Observation

→ When we ping PC and Server the address of server is known to PC & vice versa.

→ When we ping between other two PCs simultaneously the address of each other are known.

→ Every time a host request a MAC address in order to send a packet to another host in the LAN, it checks its ARP cache set if the IP to MAC address translation address already exists. If the translation doesn't exist perform ARP

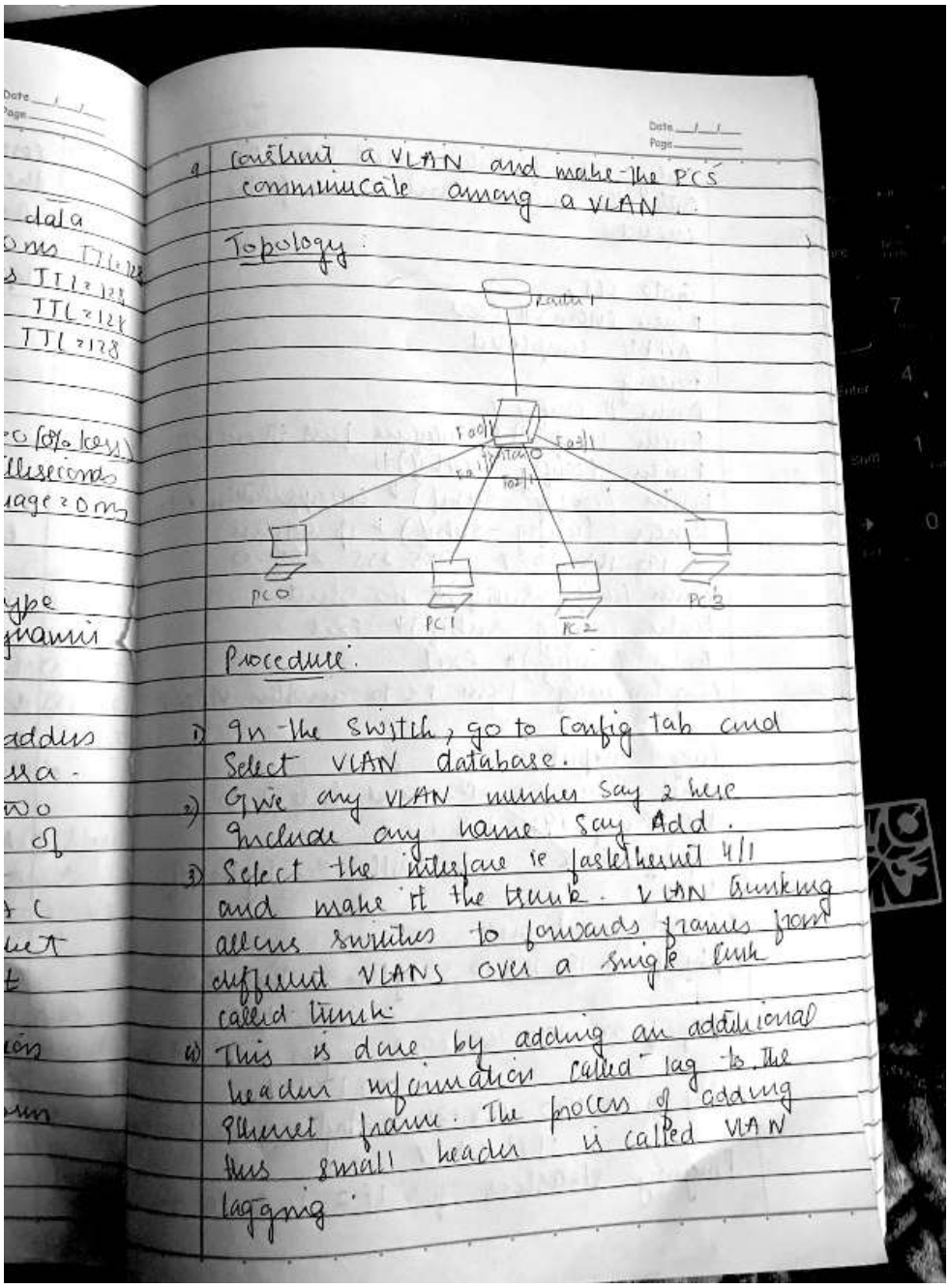
Output


IP Address
Hardware Address
Interface
10.0.0.1
00E0.A32C.AE49
FastEthernet0
Server-PT
Server0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
ARP Table for PC0
IP Address
Hardware Address
Interface
10.0.0.2
0001.96AD.E47D
FastEthernet0
10.0.0.3
00E0.A3AA.581D
FastEthernet0
ARP Table for PC1
IP Address
Hardware Address
Interface
10.0.0.1
00E0.A32C.AE49
FastEthernet0
Event List
Vis.
Time(sec)
Last Dev
At Dev
Type
Info
0.000
PC0
ICMP
0.001
PC0
Switch0
ICMP
0.002
Switch0
PC2
ICMP
Switch0
Physical
Config
CLI
IOS Command Line Interface
6 FastEthernet/IEEE 802.3 interface(s)
63488K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 0001.2BAD.D78E
Motherboard assembly number: 73-5781-09
Power supply part number: 34-0965-01
Motherboard serial number: FOC0610043E
Power supply serial number: DAB0608127D
Model revision number: C0
Motherboard revision number: A0
Model number: WS-CSwitch-PT
System serial number: FHK0610Z0WC
Cisco Internetwork Operating System Software
IOS (tm) PT3000 Software (PT3000-I4Q4S2-M), Version 12.1(22)EA4, RELEASE
SOFTWARE (fc1)
Copyright (c) 1986-2006 by cisco Systems, Inc.
Compiled Fri 12-May-06 17:19 by pt_team
Press RETURN to get started!
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
Copy
Paste

EXPERIMENT 9

To construct a VLAN and make a pc communicate among VLAN

Observation

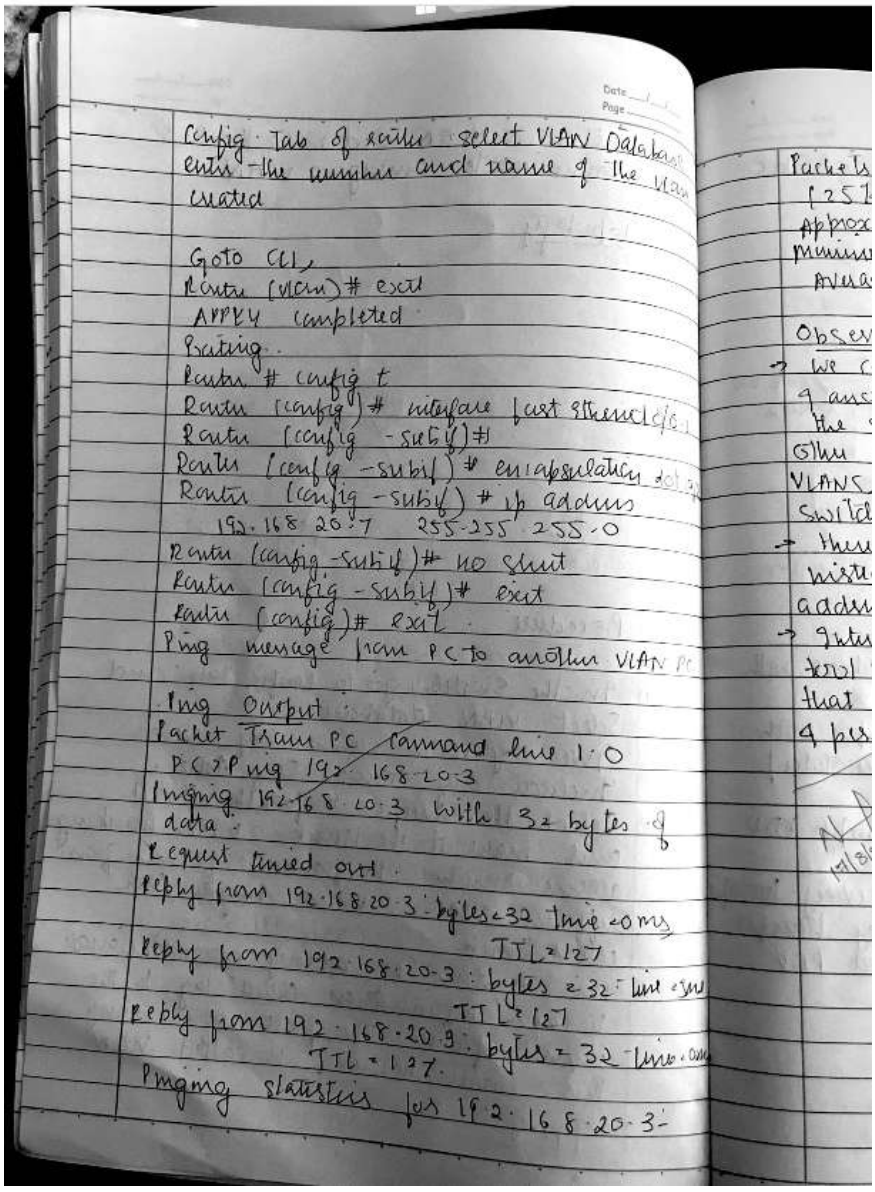


a) Construct a VLAN and make the PCs communicate among a VLAN.

Topology:

Procedure:

1) In the switch, go to Config tab and Select VLAN database.
2) Give any VLAN number say 2 here Include any name say Add.
3) Select the interface ie fastethernet 4/1 and make it the trunk. VLAN trunking allows switches to forwards frames from different VLANs over a single link called trunk.
4) This is done by adding an additional header information called tag to the Ethernet frame. The process of adding this small header is called VLAN tagging.



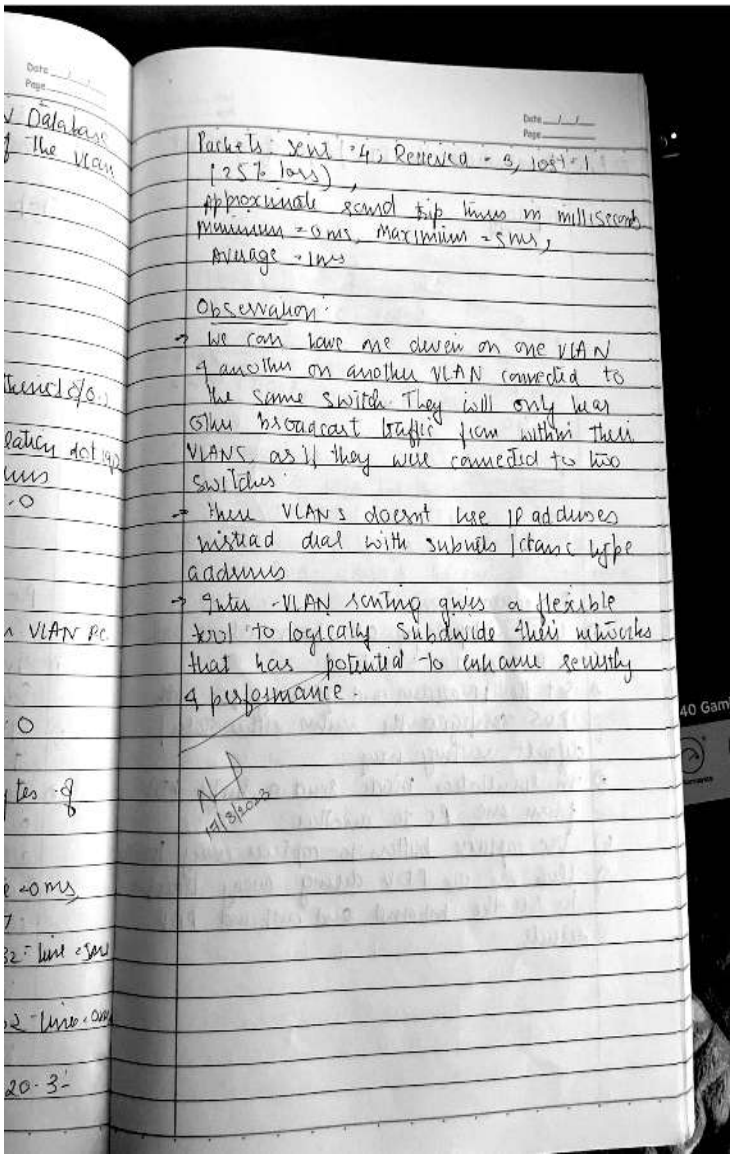
Config Tab of switch select VLAN Database
enter the number and name of the VLAN
created

Goto CLI,
Router (vlan)# exit
APPLY completed
Routing.
Router# config t
Router (config)# interface fast ethernet 0/0.1
Router (config-subif)#
Router (config-subif)# encapsulation dot1q
Router (config-subif)# ip address
192.168.20.7 255.255.255.0
Router (config-subif)# no shut
Router (config-subif)# exit
Router (config)# exit
Ping message from PC to another VLAN PC

Ping Output:
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<1ms
TTL=127
Reply from 192.168.20.3: bytes=32 time<1ms
TTL=127
Pinging statistics for 192.168.20.3:

Packets
[25
Approx
minim
Avera

Obser
→ We c
9 ans
the
Thu
VLANS
switch
→ There
mist
addr
→ Rout
tool
that
4 pc



Packets: Sent = 4, Received = 3, Lost = 1
(25% loss),
Approximate round trip times in milliseconds
Minimum = 0ms, Maximum = 5ms,
Average = 1ms

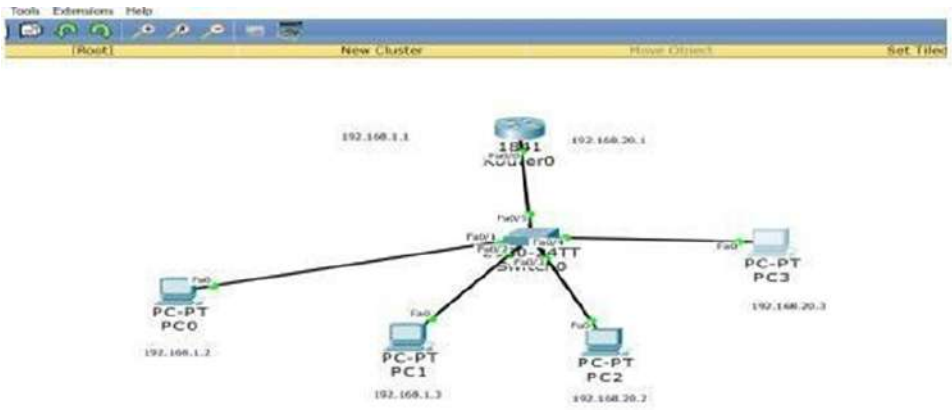
Observation:

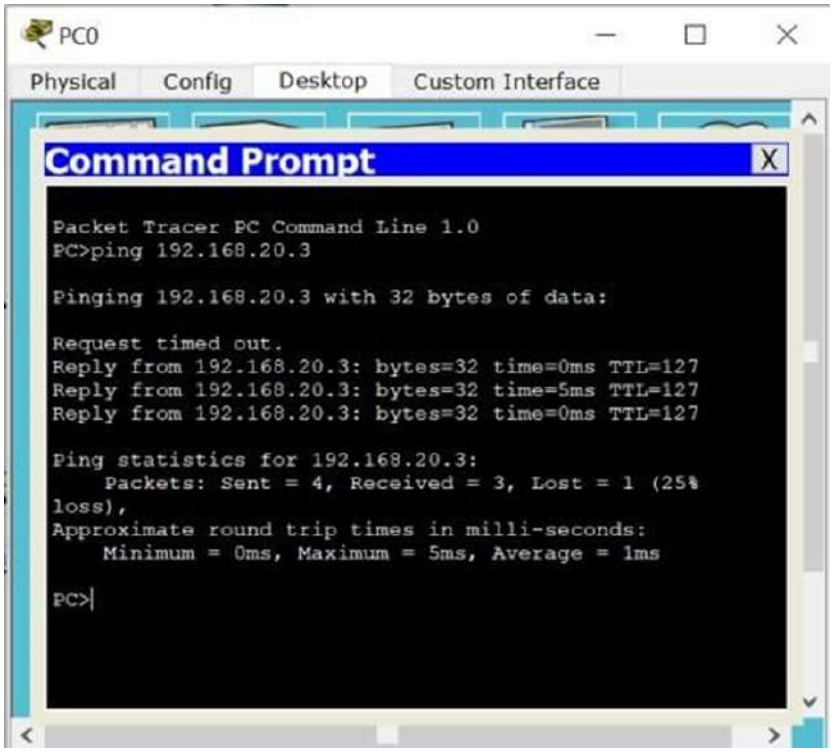
→ We can have one device on one VLAN and another on another VLAN connected to the same switch. They will only hear the broadcast traffic from within their VLANS, as if they were connected to two switches.

→ These VLANs doesn't use IP addresses instead deal with subnets (classless type addressing)

→ Inter-VLAN routing gives a flexible tool to logically subdivide their networks that has potential to enhance security & performance.

Output

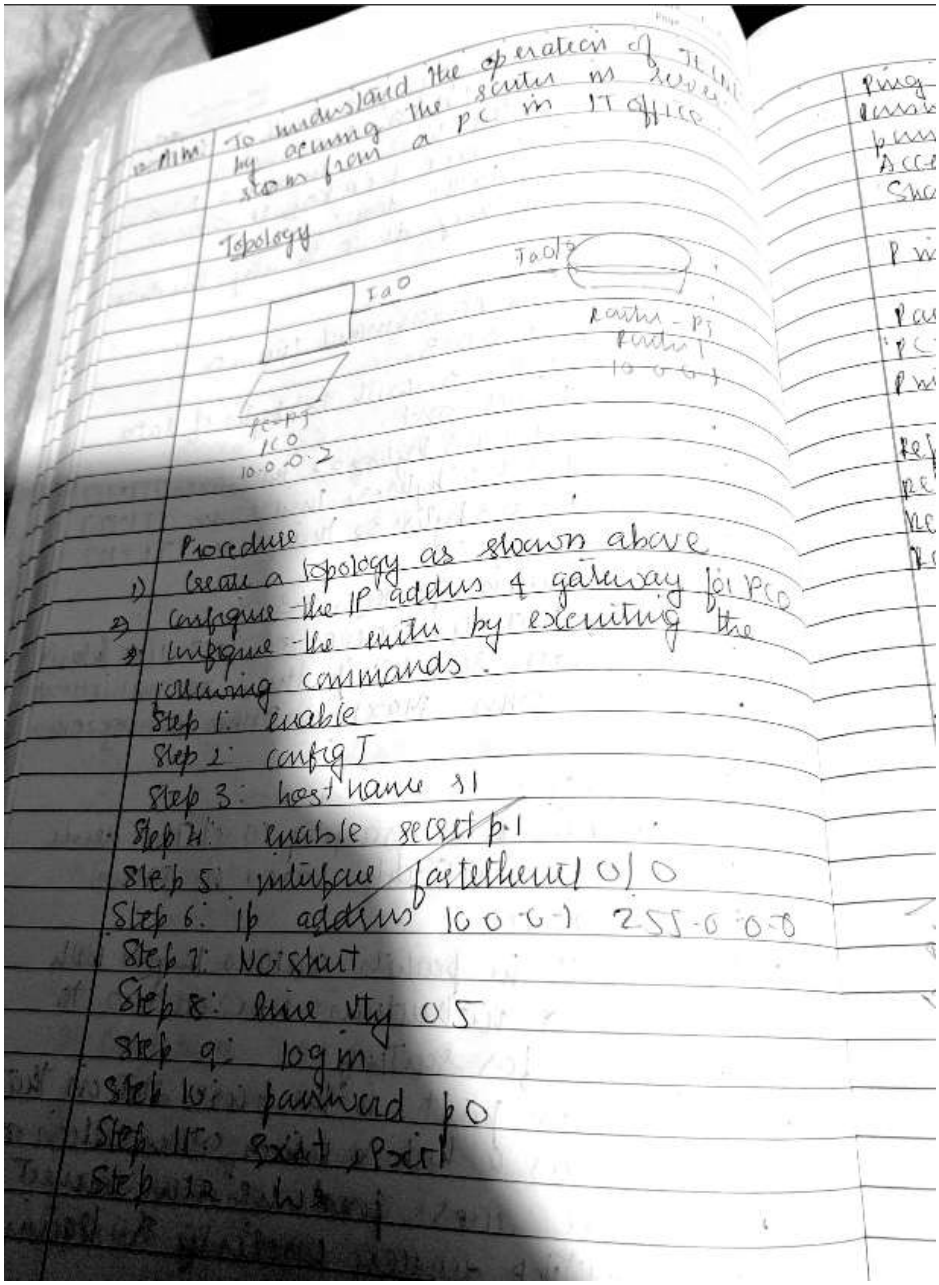


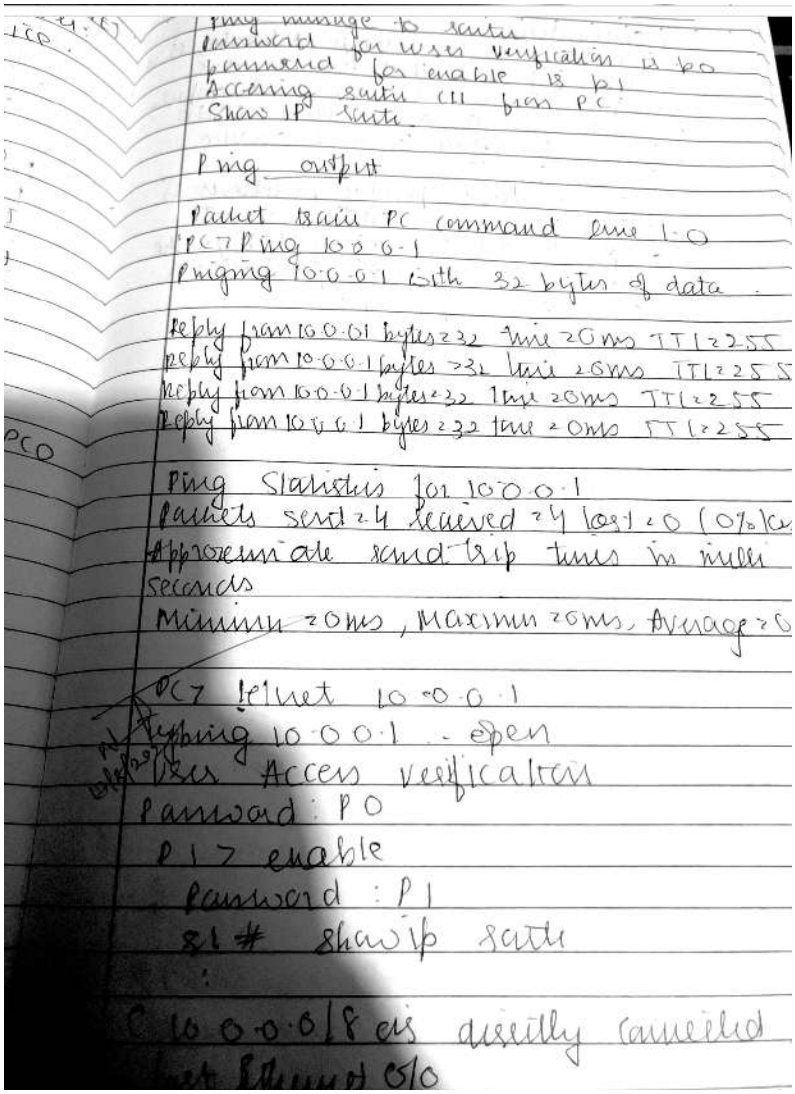

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: 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>

EXPERIMENT 10

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

Observation





ping message to switch
password for user verification is P0
password for enable is P1
Accessing switch CLI from PC
Show IP route

Ping output

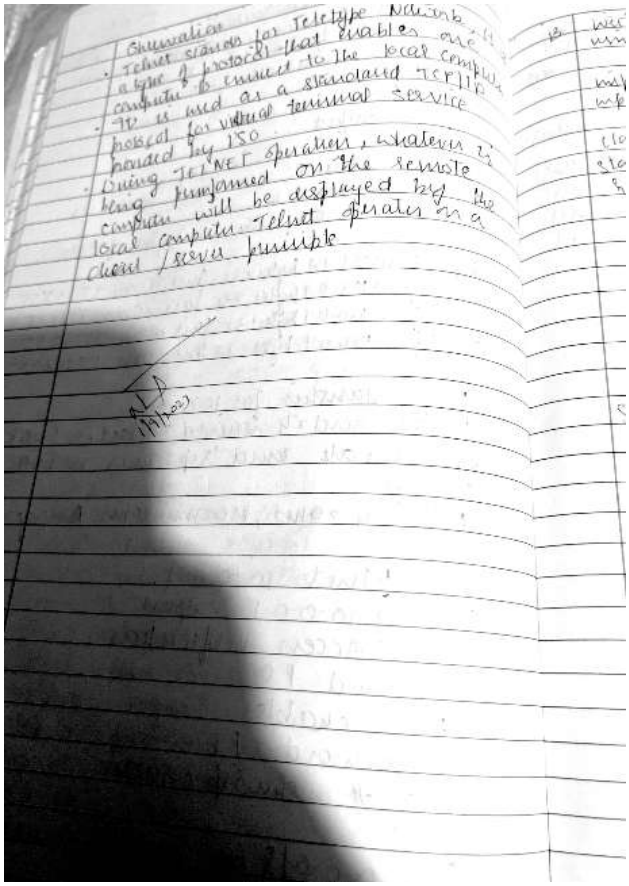
Packet tracer PC command line 1.0
PC>Ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data.

Reply from 10.0.0.1 bytes=32 time=20ms TTL=255
Reply from 10.0.0.1 bytes=32 time=20ms TTL=255
Reply from 10.0.0.1 bytes=32 time=20ms TTL=255
Reply from 10.0.0.1 bytes=32 time=20ms TTL=255

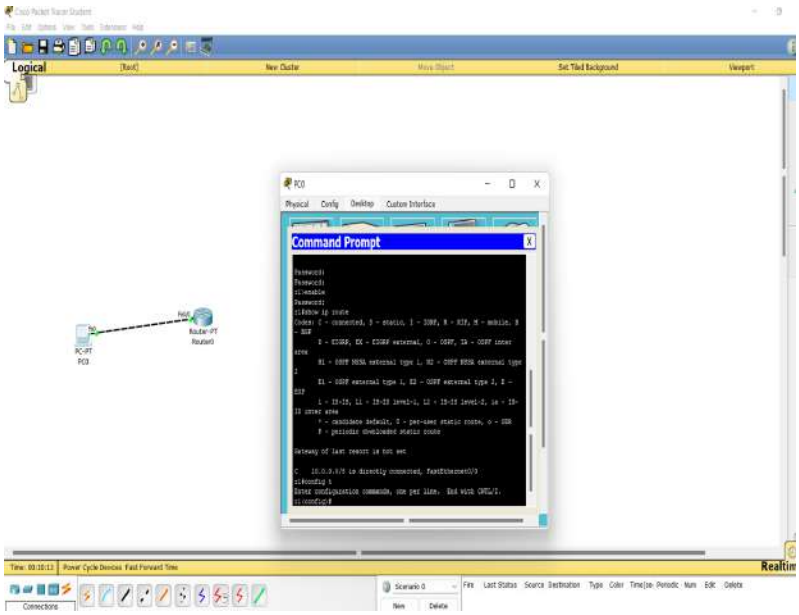
Ping Statistics for 10.0.0.1
Packets sent=4 received=4 lost=0 (0%) loss
Approximate round trip times in milli seconds
Minimum=20ms, Maximum=20ms, Average=0

PC> telnet 10.0.0.1
Trying 10.0.0.1 ... open
User Access Verification
Password: P0
R1> enable
Password: P1
R1# show ip route

[illegible] 10.0.0.0/8 is directly connected


Simulation
Telnet stands for Teletype Network
a type of protocol that enables one
computer to connect to the local computer
It is used as a standard TCP/IP
protocol for virtual terminal service
denoted by ISO
Using Telnet operation, whatever is
being performed on the remote
computer will be displayed by the
local computer. Telnet operates on a
client/server principle

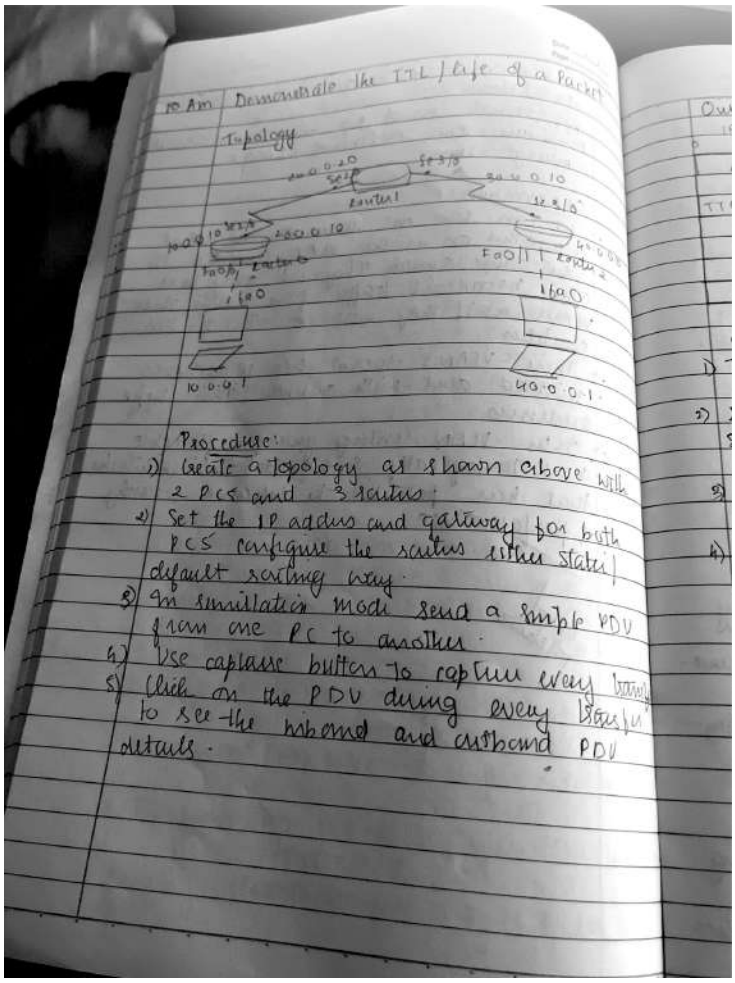
Output

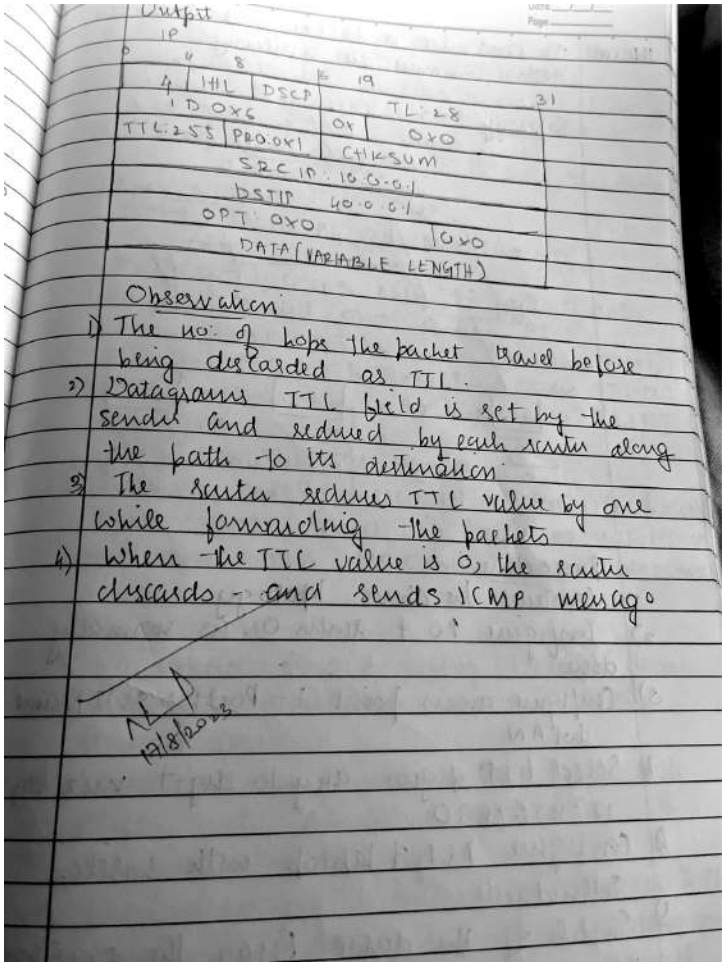

Logical
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Realtime

EXPERIMTENT 11

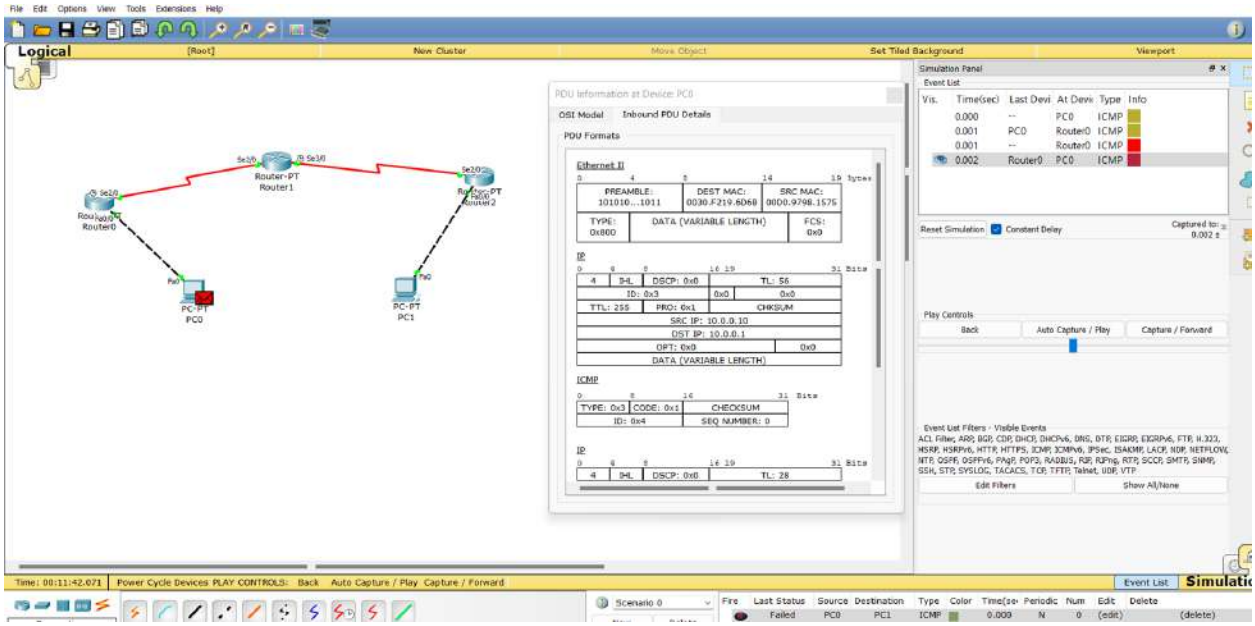
Demonstrate the TTL/ Life of a Packet

Observation




Output
IP
0 4 8 16 19 31
4 IHL DSCP TL: 28
ID 0x6 0x 0x0
TTL: 255 PRO: 0x1 CHKSUM
SRC IP: 10.0.0.1
DSTIP 40.0.0.1
OPT: 0x0 0x0
DATA (VARIABLE LENGTH)
Observation
1) The no. of hops the packet travel before being discarded as TTL.
2) Datagrams TTL field is set by the sender and reduced by each router along the path to its destination.
3) The router reduces TTL value by one while forwarding the packets
4) When the TTL value is 0, the router discards and sends ICMP messag
12/8/2023

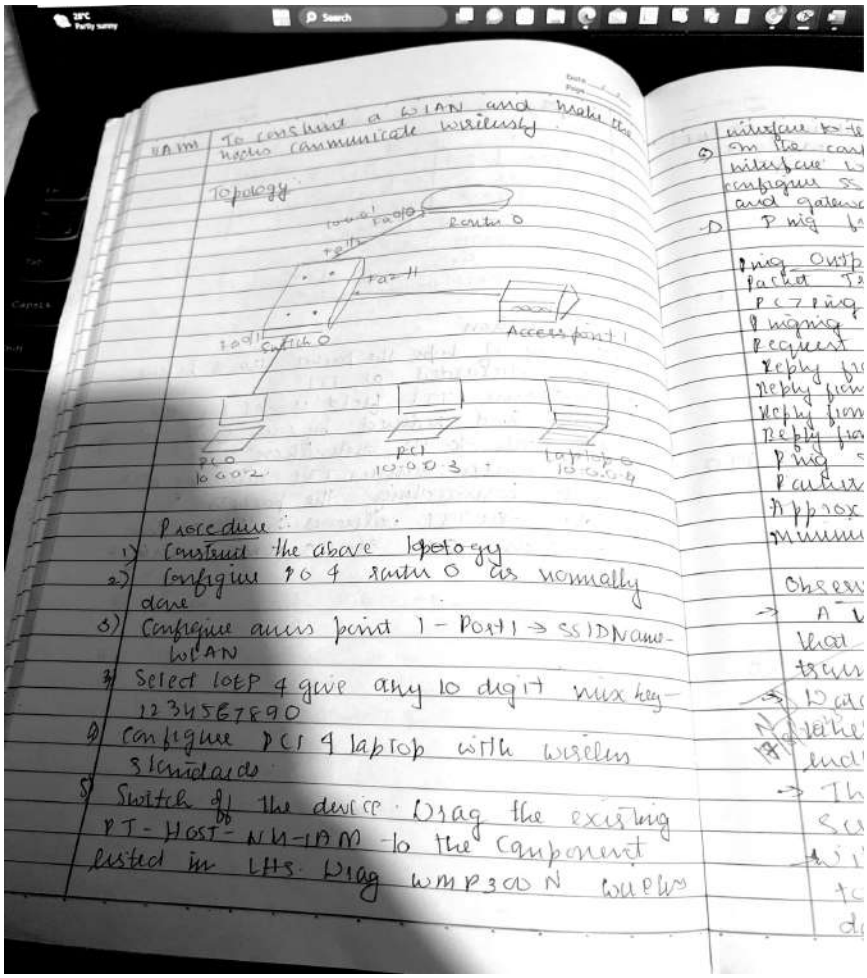
Output

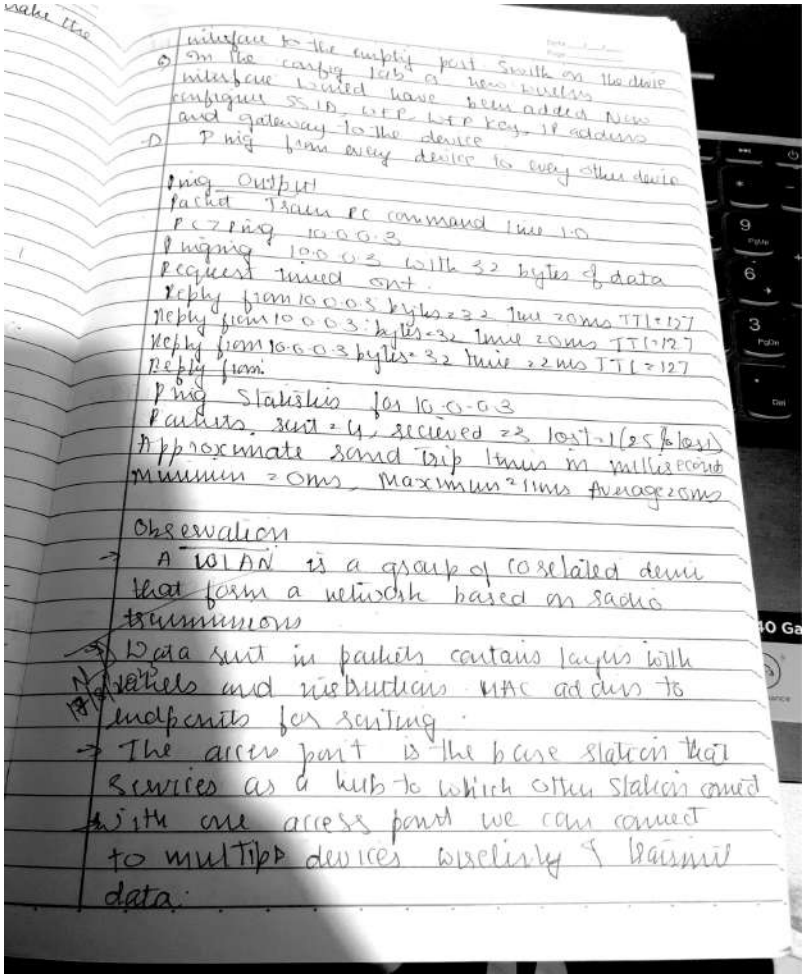

File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router0
PC-PT
PC0
PC-PT
PC1
PDU Information at Device: PC0
OSI Model
Inbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0030.F219.6D6B
SRC MAC: 00D0.9798.1575
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
ICMP
TYPE: 0x3
CODE: 0x1
CHECKSUM
ID: 0x4
SEQ NUMBER: 0
Simulation Panel
Event List
Vis. Time(sec) Last Devi At Devi Type Info
0.000 -- PC0 ICMP
0.001 PC0 Router0 ICMP
0.001 -- Router0 ICMP
0.002 Router0 PC0 ICMP
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:11:42.071
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
Failed PC0 PC1 ICMP 0.000 N 0 (edit) (delete)

EXPERIMENT 12

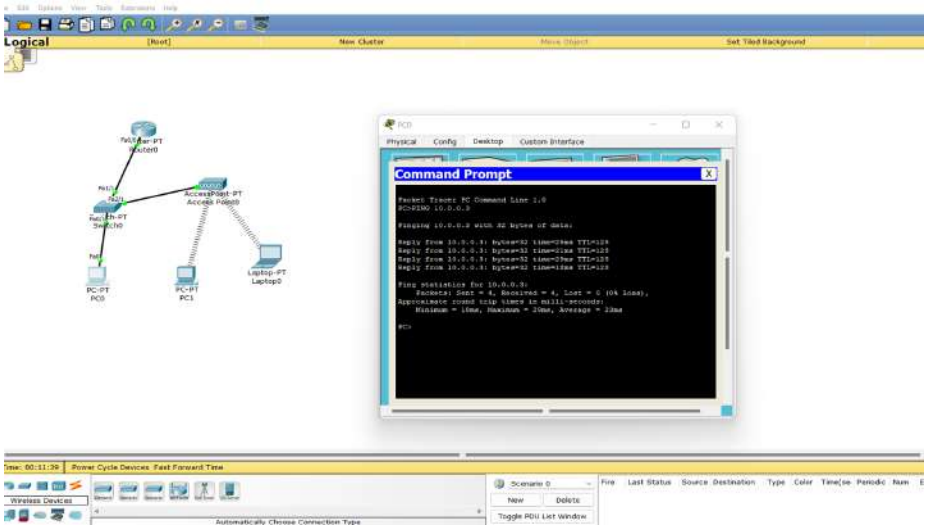
To construct a WLAN and make the nodes communicate wirelessly.

Observation




interface to the empty port Switch on the device
configure SSID, WEP, WEP key, IP address
and gateway to the device
Ping from every device to every other device
Ping Output
Packet Tracer PC command line 1.0
PC>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=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=2ms TTL=127
Reply from
Ping Statistics for 10.0.0.3
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
Observation
A WLAN is a group of correlated devi
that form a network based on radio
transmissions
Data sent in packets contains layers with
headers and destination MAC address to
endpoints for routing.
The access point is the base station that
services as a hub to which other stations connect
with one access point we can connect
to multiple devices wirelessly & transmit
data.

Output


Logical
Command Prompt
Packet Tracer PC Command Line 1.0
PC>PING 10.0.0.3
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 10ms, Maximum = 10ms, Average = 10ms
PC>
Power Cycle Devices
Fast Forward Time
Wireless Devices

EXPERIMENT 13

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

code

```
import java.util.Scanner;

import java.util.Arrays; class Program { static String Xor(String a, String b) {
String result = "";

int n = b.length();

for (int i = 1; i < n; i++)

 { result=(a.charAt(i) == b.charAt(i))?0:1;

 } return result;

} static String Div(String data, String key)

 { int pick = key.length();

String tmp = data.substring(0, pick);

int n = data.length();

 while (pick < n)

 { if (tmp.charAt(0) == '1') tmp = Xor(data, tmp) + data.charAt(pick);

 else tmp = Xor(new String(new char[pick]).replace("\0", "0"), tmp) +
data.charAt(pick); pick += 1;

}

 if (tmp.charAt(0) == '1') tmp = Xor(divisor, tmp);

 else
```

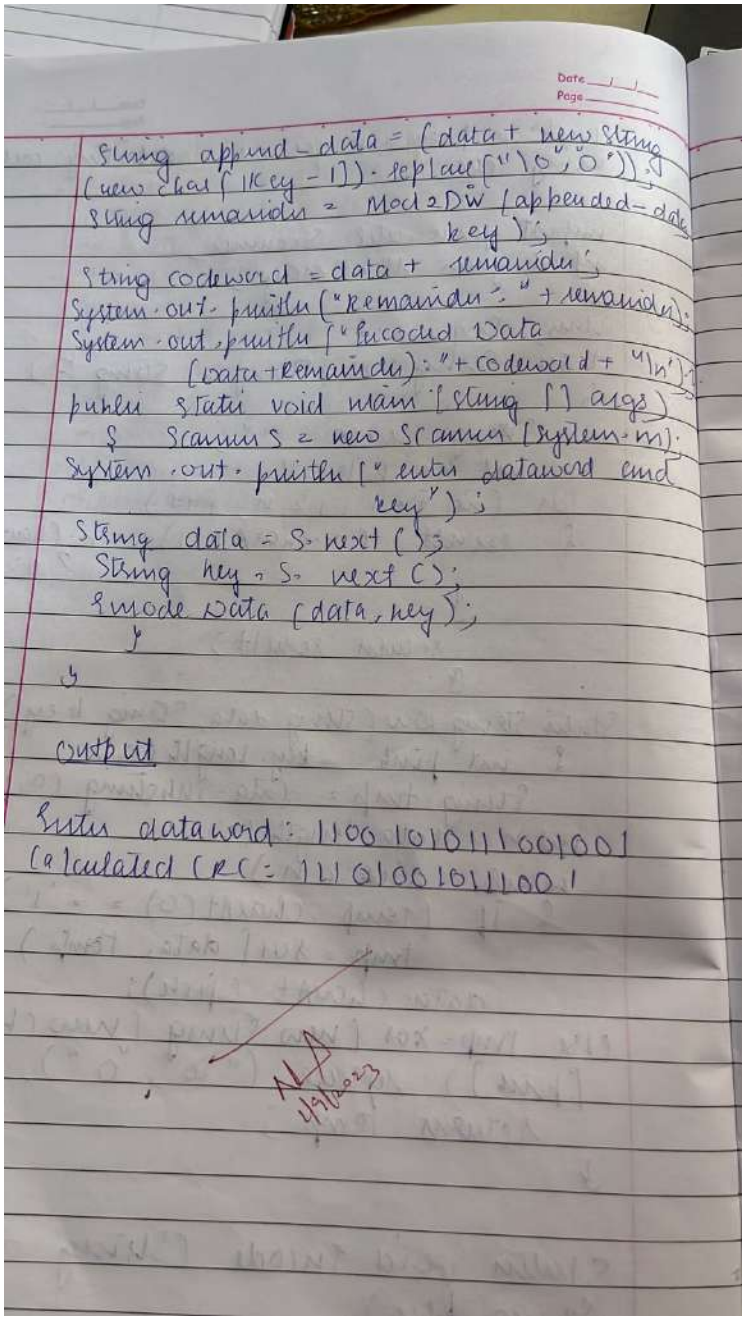
```
 tmp = Xor(new String(new char[pick]).replace("\0", "0"), tmp);

 return tmp;

}

static void Encode(String data, String key)

{ int lkey = key.length();

 String appended_data = (data + new String(new char[lkey -
1]).replace("\0", "0"));

String remainder = Mod2Div(appended_data, key); String codeword = data
+ remainder;

System.out.println("Remainder : " + remainder);
System.out.println("Encoded Data (Data + Remainder) :" + codeword +
"\n");

 }

 public static void main(String[] args)

 { Scanner s = new Scanner(System.in);

  System.out.println("enter dataword and key");

 String data = s.next();

String key = s.next();

EncodeData(data, key);

 }

}
```

Observation

Date / /
Page

13 Write a program for error detecting code using CRC

```
import java.util.Scanner;
import java.util.Arrays;

class Program {
static String xor(String a, String b)
{
    String result = "";
    int n = b.length();
    for (int i = 1; i < n; i++)
    {  result = (a.charAt(i) == b.charAt(i))
                    ? "0" : "1";
    }
    return result;
}
Static String div(String data, String key)
{ int pick = key.length();
  String tmp = data.substring(0, pick);
int n = data.length();
while (pick < n)
{ if (tmp.charAt(0) == '1')
        tmp = xor(data, tmp) +
    data.charAt(pick);
else tmp = xor(new String(new char
[pick]).replace("\0", "0"), tmp);
  return tmp;
}

Static void encode(String data,
String key)
{ int ikey = key.length();
```



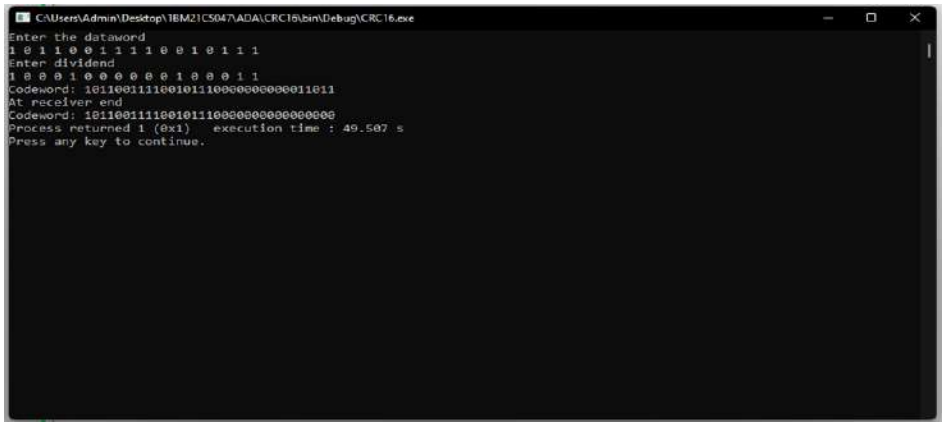
```
String appended_data = (data + new String
(new char[key.length() - 1]).replace("\0", "0"));
String remainder = Mod2Div(appended_data,
                           key);
String codeword = data + remainder;
System.out.println("Remainder: " + remainder);
System.out.println("Encoded Data
(data+Remainder): " + codeword + "\n");
public static void main(String[] args)
{ Scanner S = new Scanner(System.in);
System.out.println("enter dataword and
                    key");
String data = S.next();
String key = S.next();
encode data(data, key);
}
}
```

Output

enter dataword: 1100 1010 1110 0100 1
Calculated CRC: 1110 1001 0111 001

ALA 4/9/2023

Output


C:\Users\Admin\Desktop\18M21CS047\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.

EXPERIMENT 14

Write a program for congestion control using Leaky bucket algorithm.

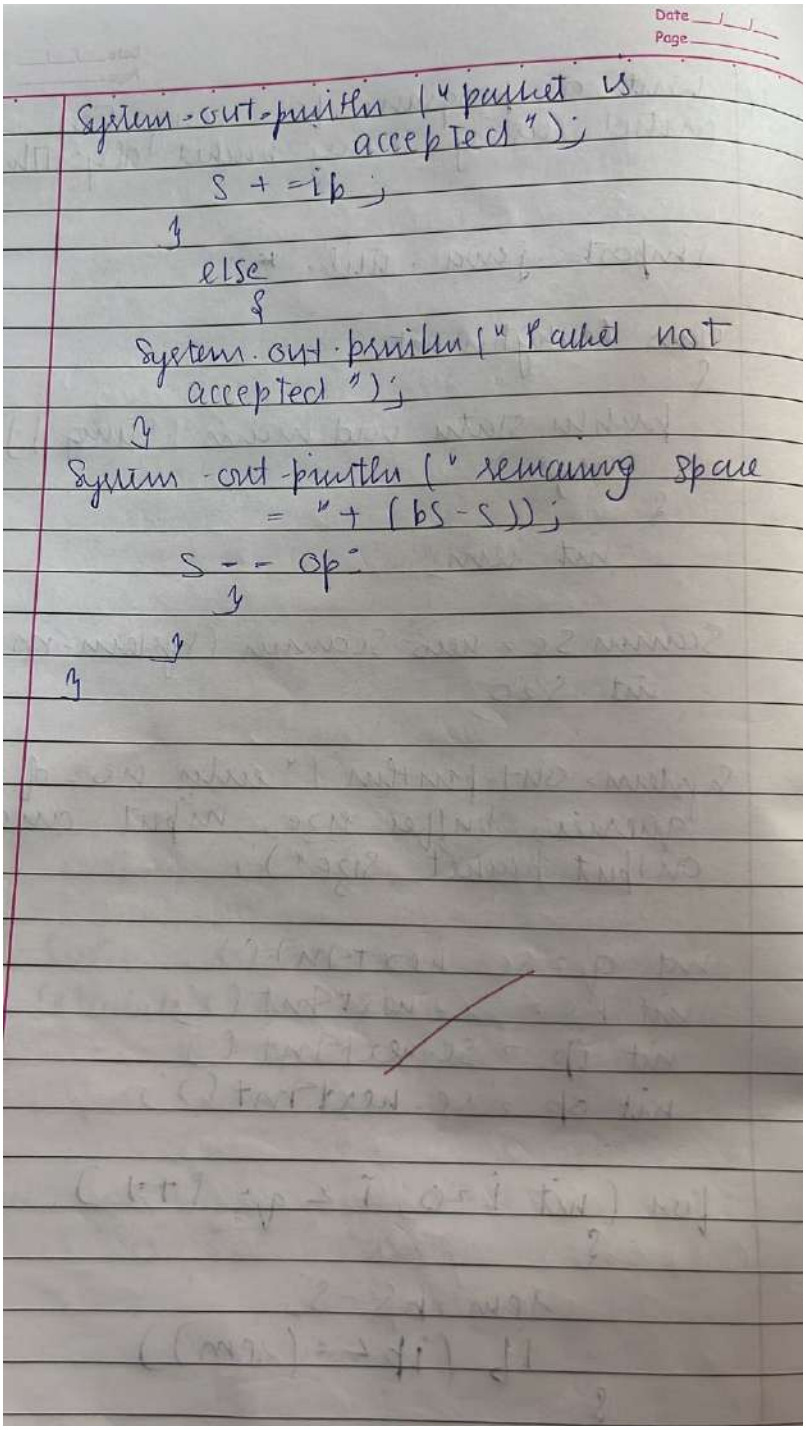
code

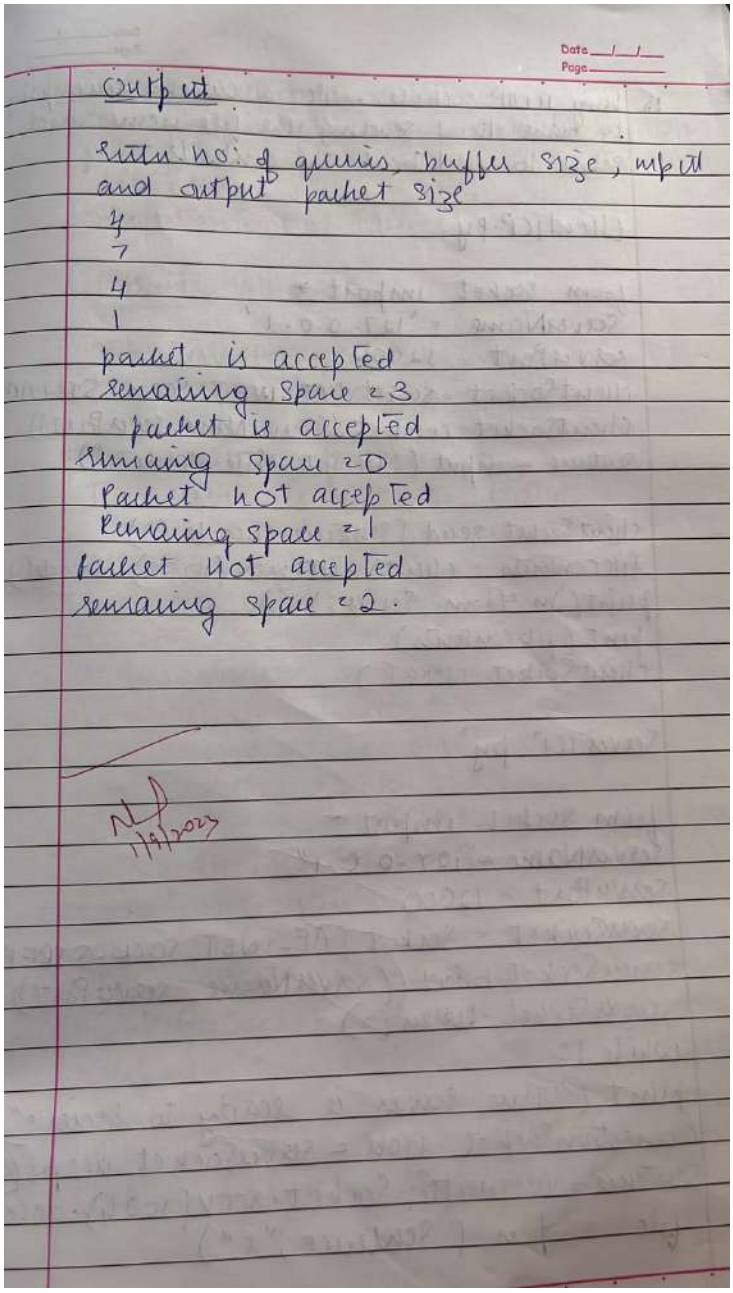
```
import java.util.*;
 class Leakybucket
{
public static void main(String[] args)
{
int rem;
 Scanner sc=new Scanner(System.in);
 int s= 0;
 System.out.println("enter no of queries,buffer size,input and output
packet size ");
int q=sc.nextInt();
 int bs=sc.nextInt();
 int ip=sc.nextInt();
int op=sc.nextInt();
for (int i = 0; i < q; i++)
 {
rem=bs-s;
 if (ip <= (rem))
 {
System.out.println("packet is accepted");
 s+=ip;
 } else
{
 System.out.println("Packet not accepted ");
 }
 System.out .println("remaining space="+(bs-s)); s -= op;
}
 }
 }
```

Observation

14 Write a program for congestion control using leaky bucket algorithm

```
import java.util.*;

class leakybucket
{
  public static void main (String [])
    args)
  {
    int rem;

Scanner sc = new Scanner (System.in);
  int S=0;

System.out.println ("enter no. of
  queries, buffer size, input and
  output packet size");

int q = sc.nextInt();
int b = sc.nextInt();
int ip = sc.nextInt()
int op = sc.nextInt();

for (int i=0; i < q; i++)
  {
    rem = bS - S;
    if (ip <= (rem))
  {
```


Date
Page
System.out.println("packet is accepted");
s += ib;
}
else
{
System.out.println("Packet not accepted");
}
System.out.println("remaining space = " + (bs - s));
s -= ob;
}
}


Date
Page
Output
Enter no: of queues, buffer size, input and output packet size
4
7
4
1
packet is accepted
remaining space = 3
packet is accepted
remaining space = 0
Packet not accepted
Remaining space = 1
packet not accepted
remaining space = 2.
11/9/2023

Output

```
enter no of queries,buffer size,input and output packet size
4 10
6
1
packet is accepted
remaining space=4
Packet not accepted
remaining space=5
packet is accepted
remaining space=0
Packet not accepted
remaining space=1
```

EXPERIMENT 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,server
Port)) sentence = input(&quot;\nEnter file
name:                                &quot;)
clientSocket.send(sentence.encode())
filecontents=
clientSocket.recv(1024).decode() print
(&#39;\nFromServer:\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,server

Port)) serverSocket.listen(1)

while 1:

print ("The server is ready to receive") connectionSocket, addr= serverSocket.accept()sentence= connectionSocket.recv(1024).decode()

file=open(sentence,&quot ;r") l=file.read(1024) connectionSocket.send(l.e ncode())

print ('\nSent contents of ' + sentence) file.close()

connectionSocket.close()

Observation

Date / /
Page

15 Using TCP/IP sockets, write a client - server program to make client sending the file name and server to send back the contents.

ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
clientSocket.connect ((serverName, serverPort))
sentence = input ("\n Enter file name :")

clientSocket.send (sentence.encode ())
fileContents = clientSocket.recv (1024).decode()
print('\n From Server :\n')
print (fileContents)
clientSocket.close ()
```

ServerTCP.py

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

Date __/__/__
Page _____

```
l = file.read(1024)
connectionSocket.send(l.encode())
print('\n Sent contents of ' + sentence)
file.close()
connectionSocket.close()
```

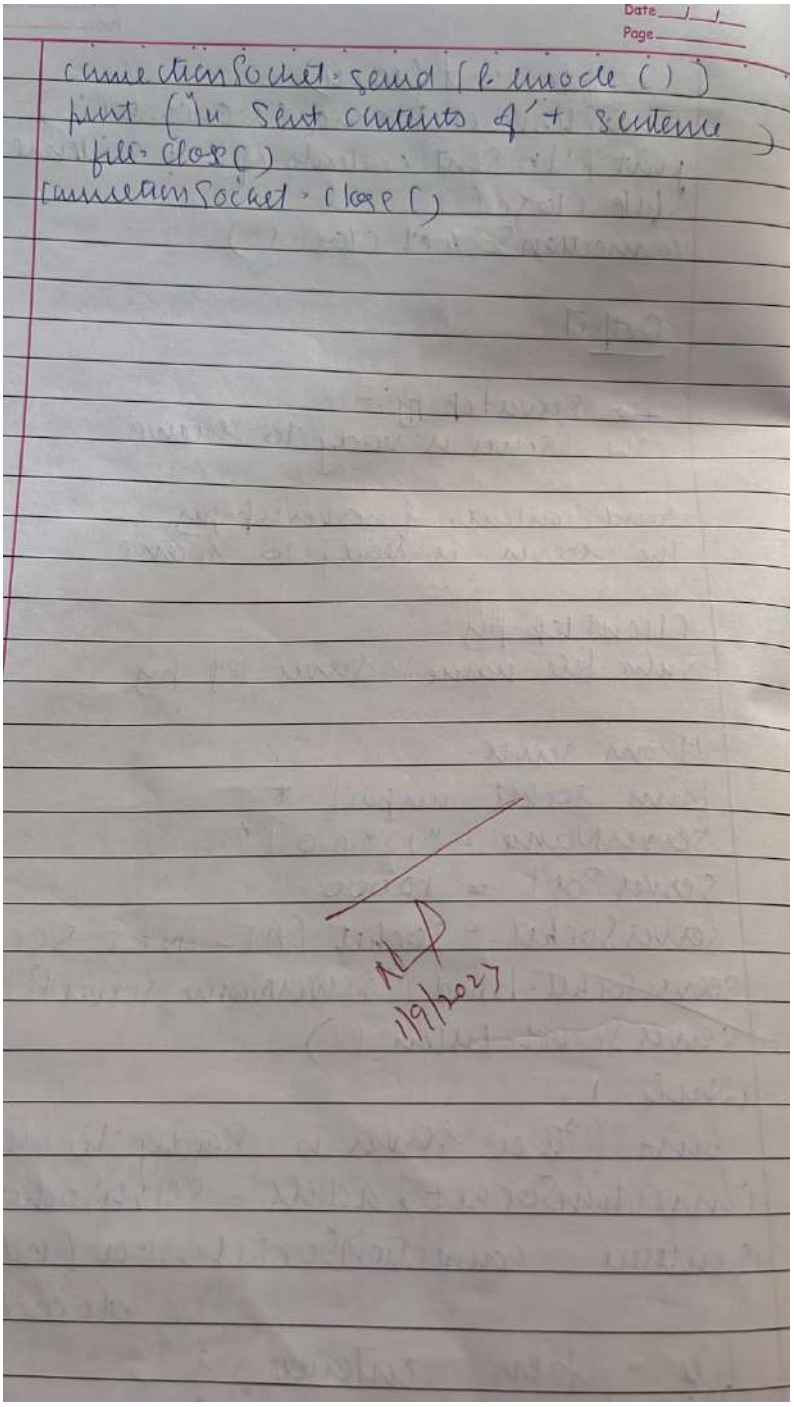
Output

~~The~~ Servertcp.py:-
The server is ready to receive

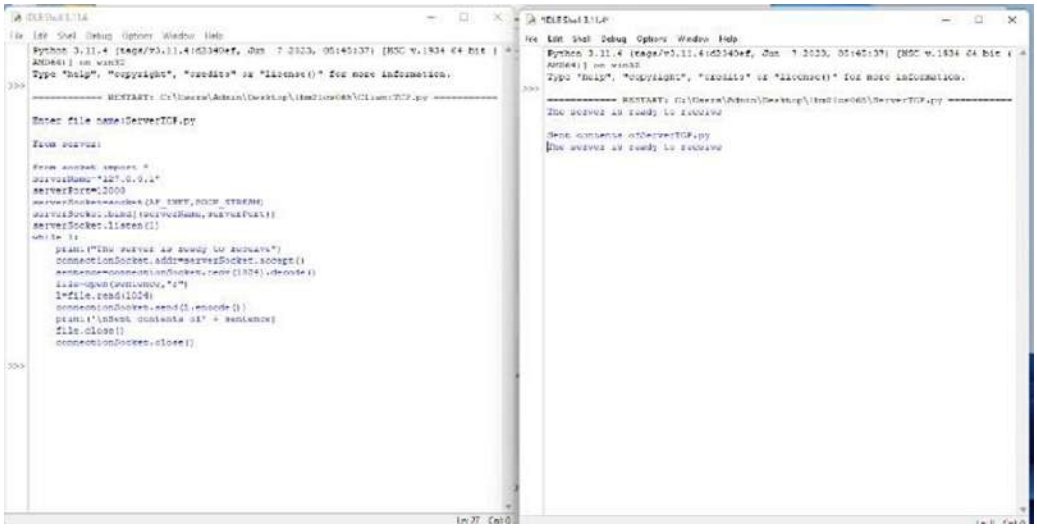
send contents of servertcp.py
The server is ready To receive

Clienttcp.py
Enter file name: Server tcp.py

From server.
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
serverSocket = socket(AF_INET, SOCK_STR
~~server~~socket.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)


Date
Page
connectionSocket.send(l.encode())
print("In Sent contents of" + sentence)
file.close()
connectionSocket.close()
NLP
1/9/2023

Output



EXPERIMENT 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.

code

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;),(serverNa

me, 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((&quot;127.0.0.1&quo
t;, serverPort)) print (&quot;The server is
ready to receive&quot;) while 1:
sentence, clientAddress =
serverSocket.recvfrom(2048) sentence =
sentence.decode(&quot;utf-8&quot;)
file=open(sentence,&quot;r&quot;)

con=file.read(2048)
serverSocket.sendto(bytes(con,&quot;utf-8&quot;
),clientAddress) print (&#39;\nSent contents of
&#39;, end = &#39; &#39;)
print (sentence)
# for i in sentence:
# print (str(i), end =
&#39;&#39;)
file.close()
```

Observation

Date __/__/__
Page _____

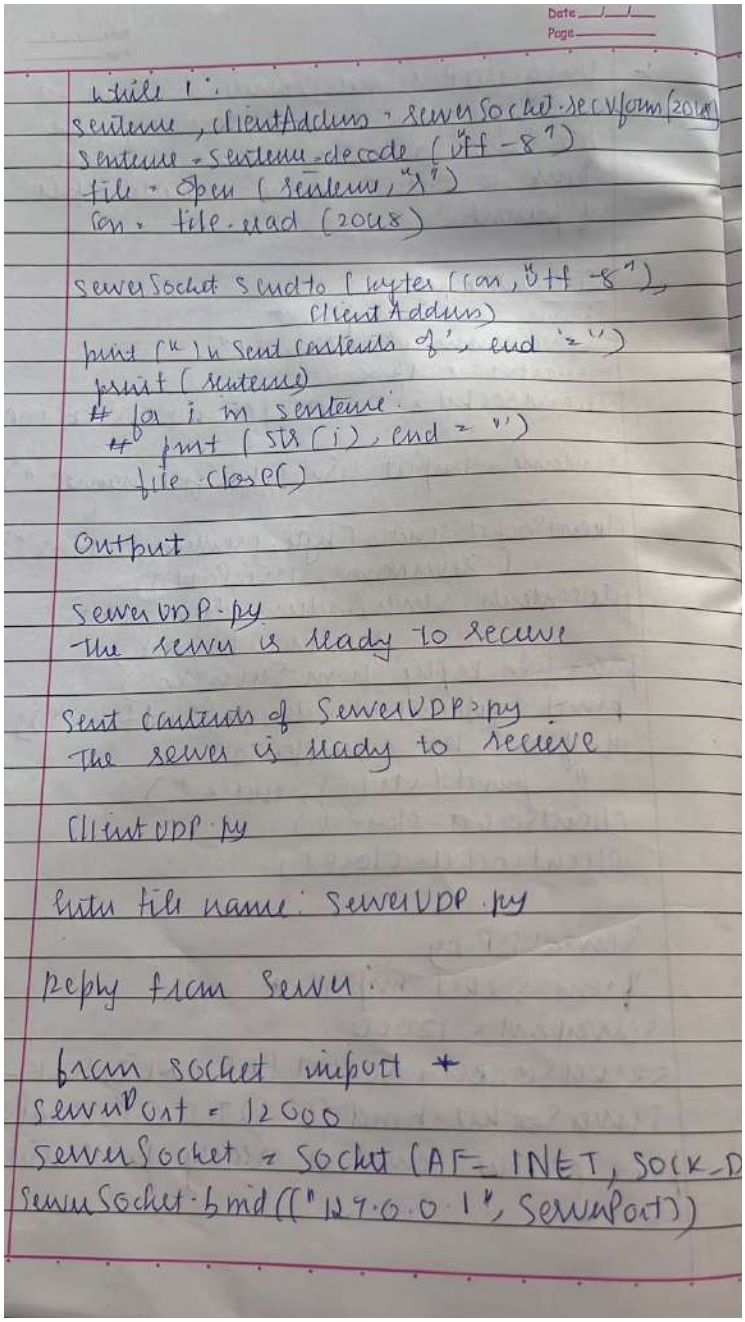
16 Using UDP sockets, write a client - server program to make client sending the file name and the server to send back to contents of the requested file if present.

ClientUDP.py

```
from socket import *
ServerName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM
sentence = input("Input file name: ")

clientSocket.sendto(bytes(sentence, "utf-8"),
        (serverName, serverPort))
fileContents, serverAddress = clientSocket.
                recvfrom(2048)
print("In Reply 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_DG
serverSocket.bind(("127.0.0.1", serverPort
print("The server is ready to receive")
```



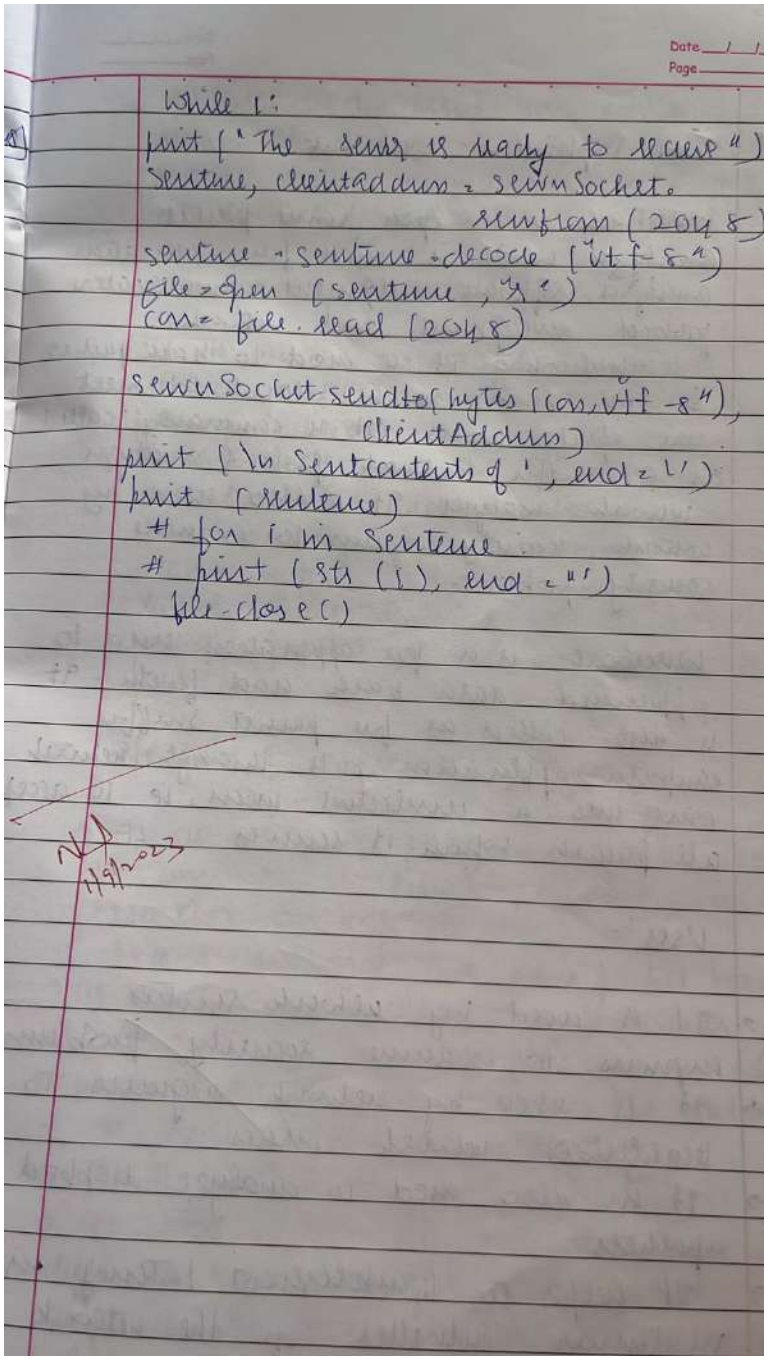
```
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file = open(sentence, "r")
con = file.read(2048)

serverSocket.sendto(bytes(con, "utf-8"),
                    clientAddress)
print("\n Sent contents of", end = ' ')
print(sentence)
# for i in sentence:
#     print(str(i), end = "")
    file.close()
```

Output

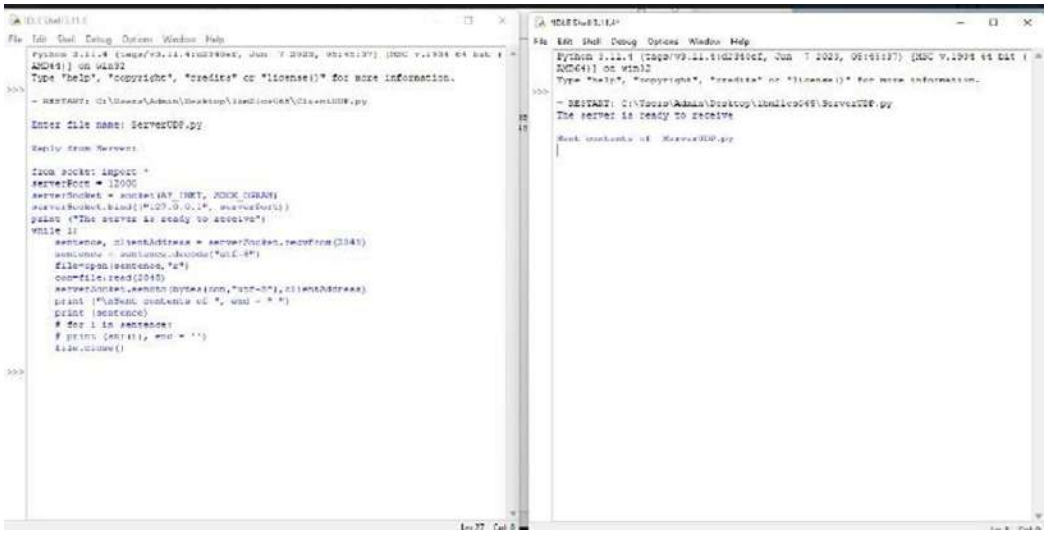
```
serverUDP.py
The server is ready to receive
```

```
Sent contents of serverUDP.py :
The server is ready to receive
```

```
clientUDP.py

Enter file name: serverUDP.py

Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_D
serverSocket.bind(("127.0.0.1", serverPort))
```


Date
Page
while 1:
print ("The server is ready to receive")
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 ("\n Sent contents of ", end = "")
print (sentence)
# for i in sentence
# print (str (i), end = "")
file.close()
11/9/2023

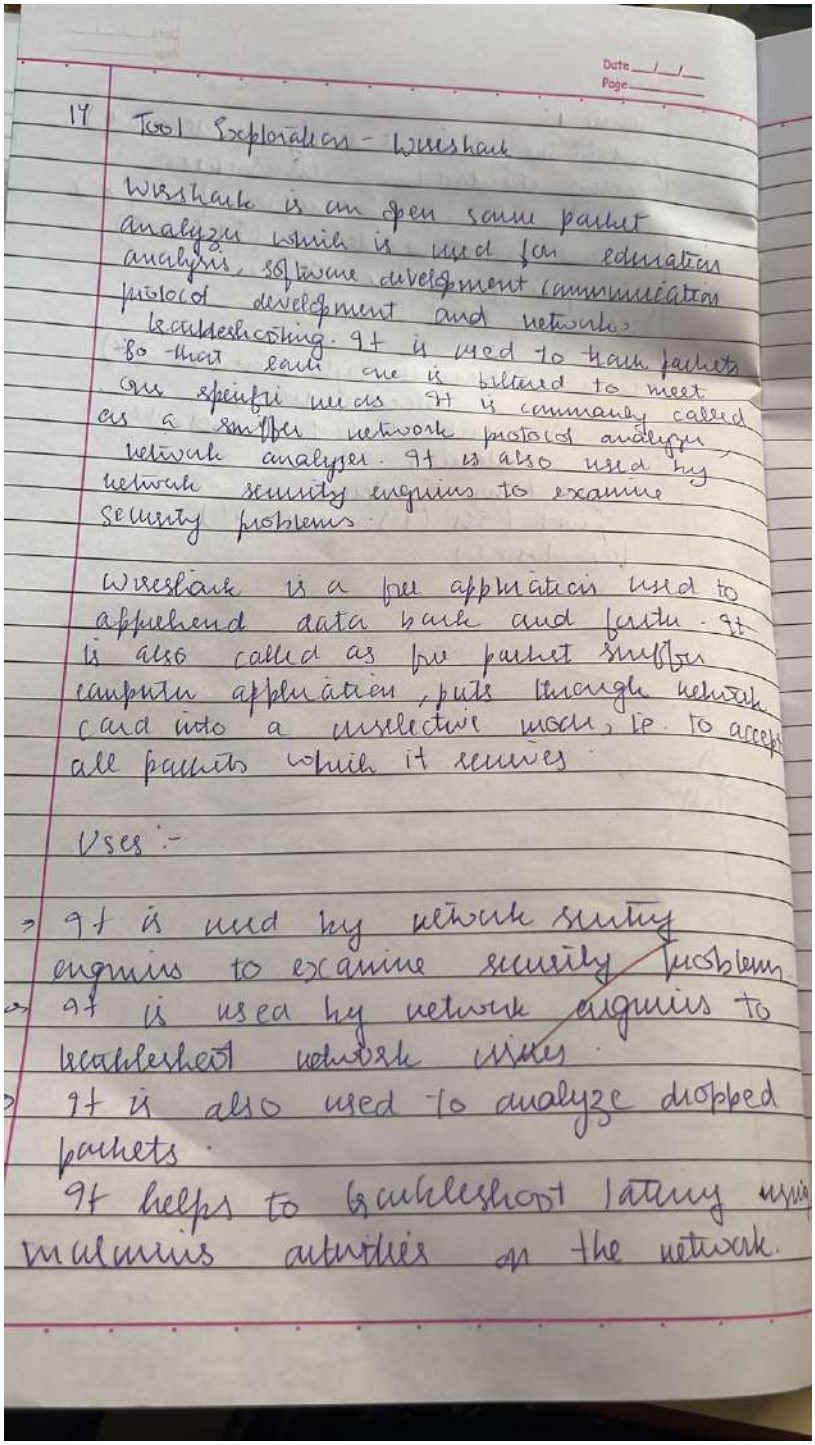
Output


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\lbm21cs048\ClientUDP.py
Enter file name: ServerUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("",12000))
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 3.11.4 (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\lbm21cs048\ServerUDP.py
The server is ready to receive

EXPERIMENT 17

Wireshark

Observation



17 Tool Exploration - Wireshark

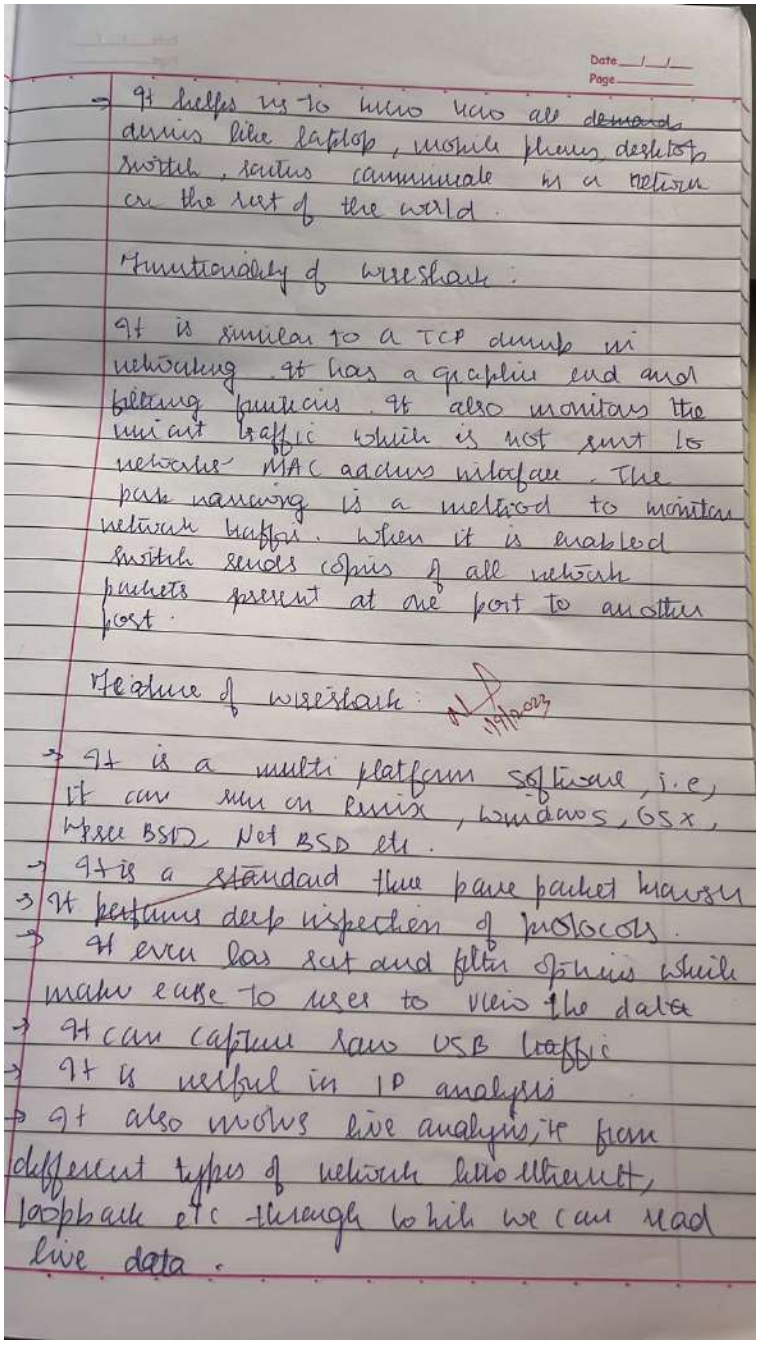
Wireshark is an open source packet analyzer which is used for education analysis, software development communication protocol development and network troubleshooting. It is used to track packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer network protocol analyzer, network analyzer. It is also used by network security engineers to examine security problems.

Wireshark is a free application used to apprehend data back and forth. It is also called as free packet sniffing computer application, puts through network card into a promiscuous mode, i.e. to accept all packets which it receives.

Uses :-

- It is used by network security engineers to examine security problems.
- It is used by network engineers to troubleshoot network issues.
- It is also used to analyze dropped packets.

It helps to troubleshoot latency issues, malicious activities on the network.



Date __/__/__
Page _____

→ It helps us to know how all devices like laptop, mobile phone, desktop switch, routers communicate in a network or the rest of the world.

Functionality of wireshark:

It is similar to a TCP dump in networking. It has a graphic end and filtering features. It also monitors the unwant traffic which is not sent to network's MAC address interface. The port mirroring is a method to monitor network traffic. When it is enabled switch sends copies of all network packets present at one port to another port.

Feature of wireshark:

→ It is a multi platform software, i.e, it can run on Linux, Windows, OSX, Hyper BSD, Net BSD etc.
→ It is a standard three pane packet browser
→ It performs deep inspection of protocols.
→ It even has sort and filter options which make ease to user to view the data
→ It can capture raw USB traffic
→ It is useful in IP analysis
→ It also allows live analysis, it from different types of network like Ethernet, loopback etc through which we can read live data.