VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS LAB

Submitted by

RAKSHITHA D N (1BM22CS415)

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 "**COMPUTER NETWORKS LAB**" carried out by **RAKSHITHA DN (1BM22CS415),** 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

.

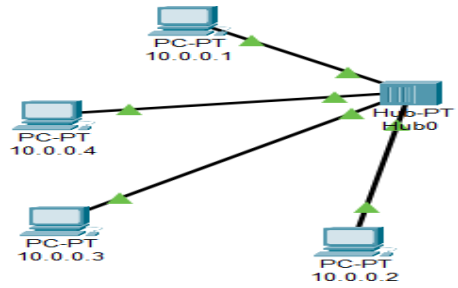
Sowmya T
Assistant Professor
Department of CSE
BMSCE, Bengaluru

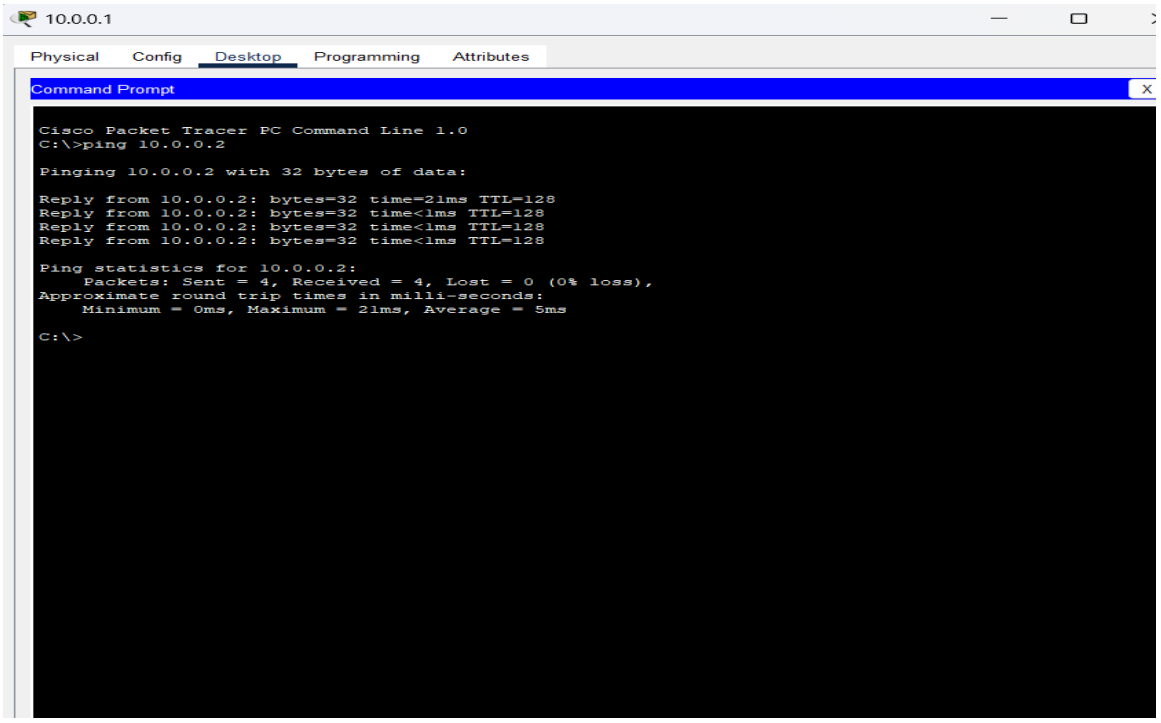
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

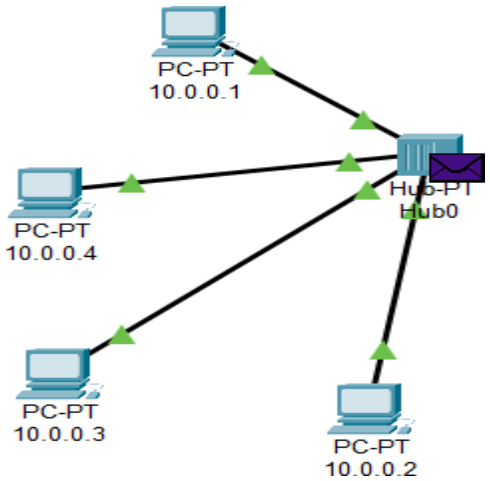
Index

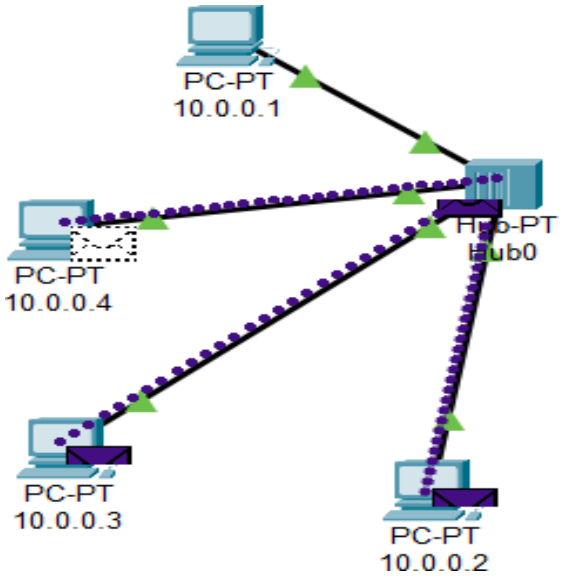
LAB PROGRAM-01

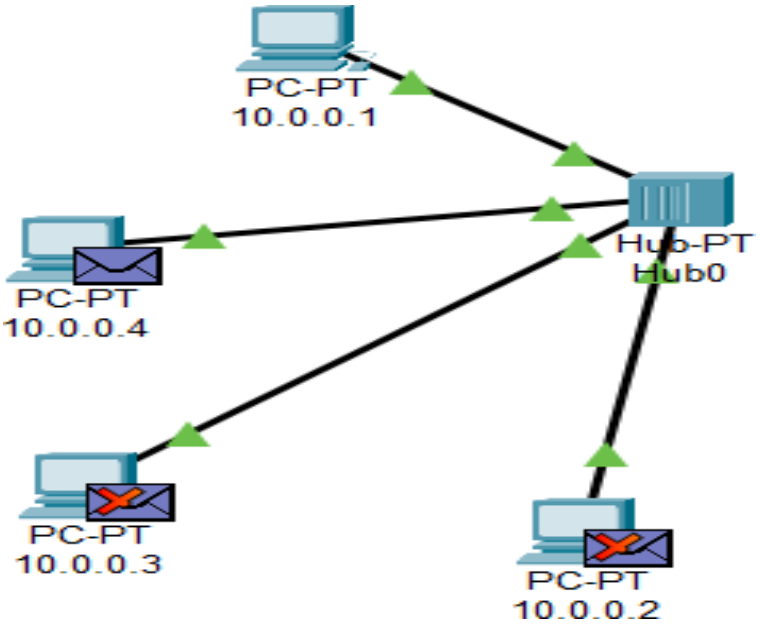
Create a topology consisting of three or more device connected with help of a hub and a switch

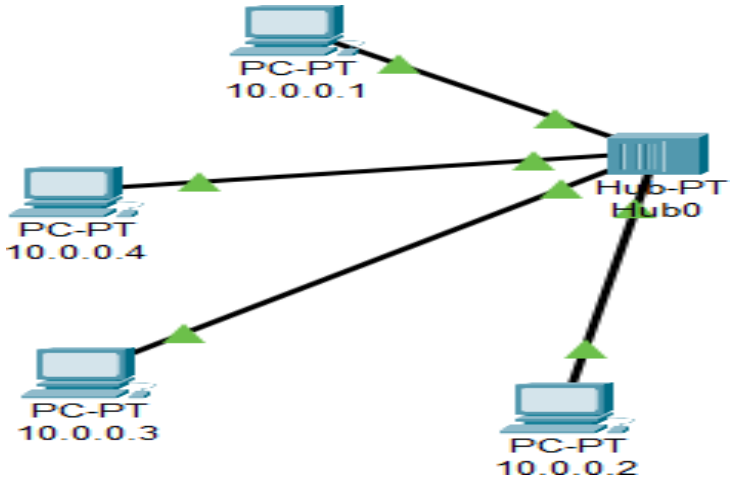


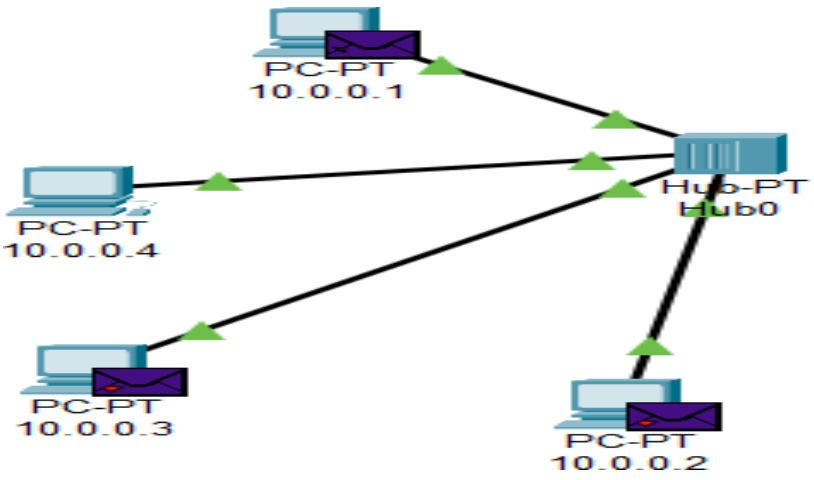

10.0.0.1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=21ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 21ms, Average = 5ms
C:\>

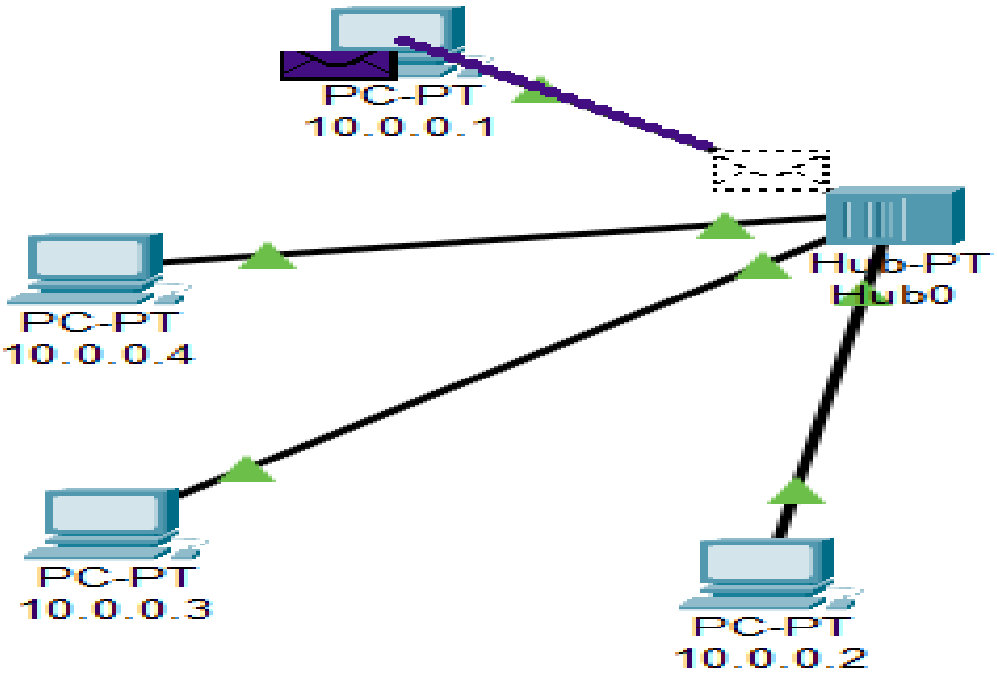

PC-PT
10.0.0.1
Hub-PT
Hub0
PC-PT
10.0.0.4
PC-PT
10.0.0.3
PC-PT
10.0.0.2

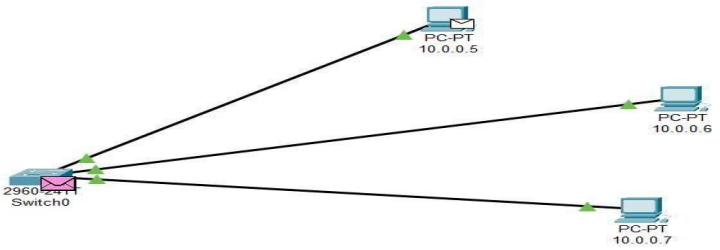

PC-PT
10.0.0.1
Hub-PT
Hub0
PC-PT
10.0.0.4
PC-PT
10.0.0.3
PC-PT
10.0.0.2

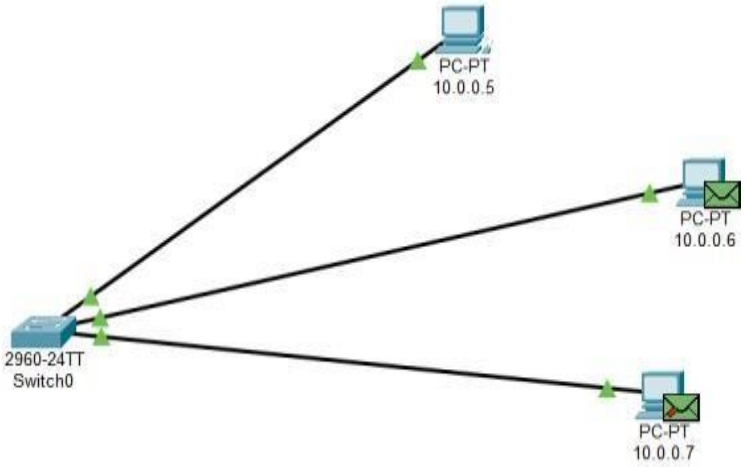

PC-PT
10.0.0.1
Hub-PT
Hub0
PC-PT
10.0.0.4
PC-PT
10.0.0.3
PC-PT
10.0.0.2

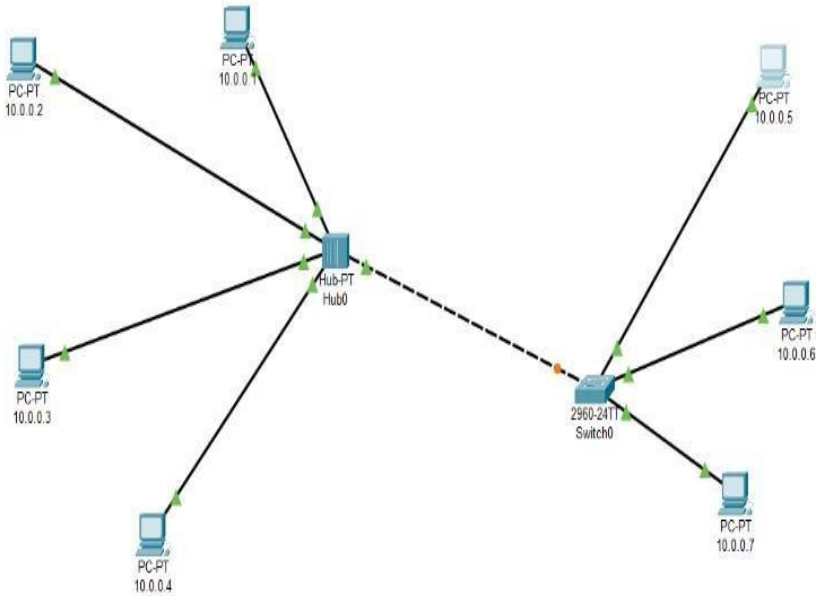

PC-PT
10.0.0.1
Hub-PT
Hub0
PC-PT
10.0.0.4
PC-PT
10.0.0.3
PC-PT
10.0.0.2

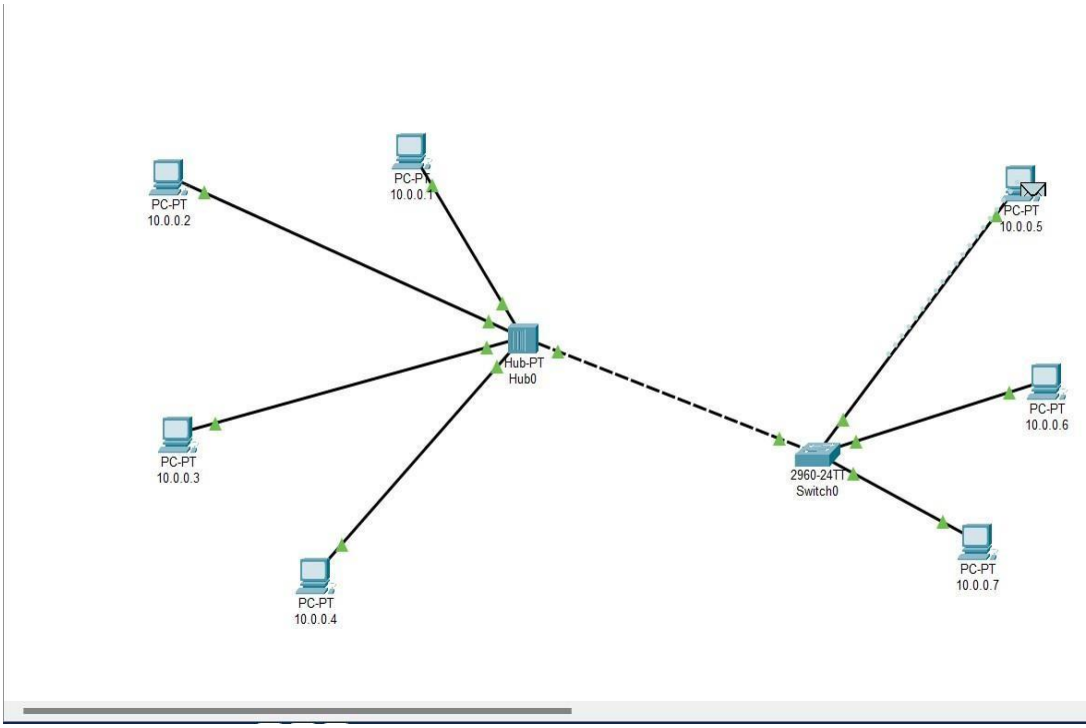

PC-PT
10.0.0.1
Hub-PT
Hub0
PC-PT
10.0.0.4
PC-PT
10.0.0.3
PC-PT
10.0.0.2

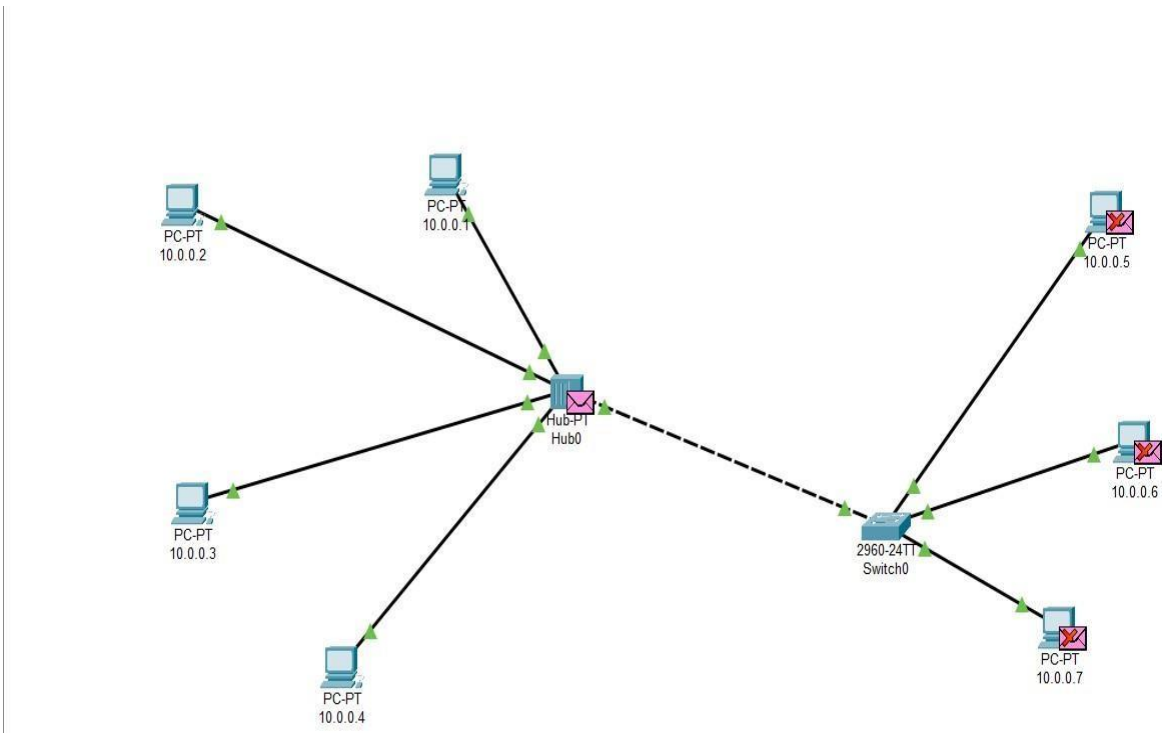

PC-PT
10.0.0.1
Hub-PT
Hub0
PC-PT
10.0.0.4
PC-PT
10.0.0.3
PC-PT
10.0.0.2

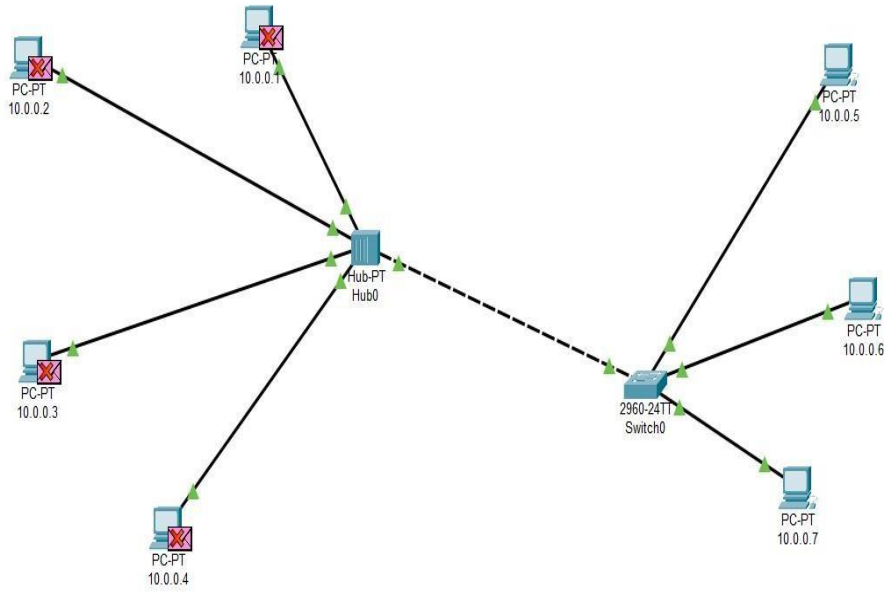

PC-PT
10.0.0.5
PC-PT
10.0.0.6
Switch0
PC-PT
10.0.0.7

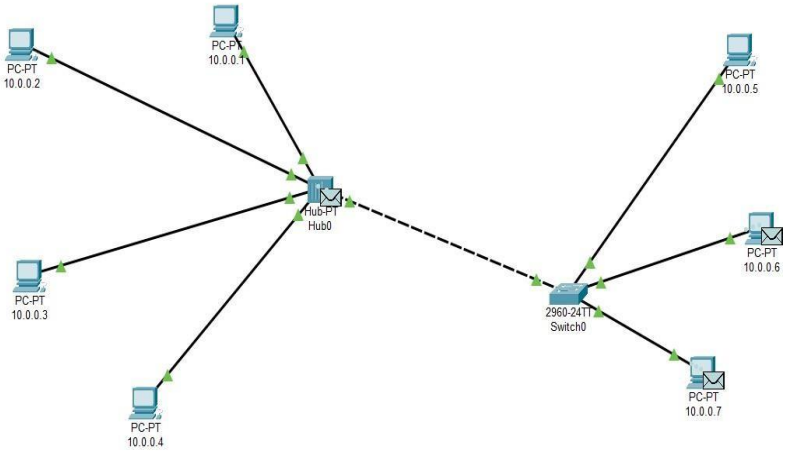

PC-PT
10.0.0.5
PC-PT
10.0.0.6
2960-24TT
Switch0
PC-PT
10.0.0.7


PC-PT
10.0.0.2
PC-PT
10.0.0.1
PC-PT
10.0.0.5
Hub-PT
Hub0
PC-PT
10.0.0.6
PC-PT
10.0.0.3
2960-24TT
Switch0
PC-PT
10.0.0.7
PC-PT
10.0.0.4


PC-PT
10.0.0.2
PC-PT
10.0.0.1
PC-PT
10.0.0.5
Hub-PT
Hub0
PC-PT
10.0.0.6
PC-PT
10.0.0.3
2960-24TT
Switch0
PC-PT
10.0.0.7
PC-PT
10.0.0.4


PC-PT
10.0.0.2
PC-PT
10.0.0.1
PC-PT
10.0.0.5
Hub-PT
Hub0
PC-PT
10.0.0.6
PC-PT
10.0.0.3
2960-24TT
Switch0
PC-PT
10.0.0.7
PC-PT
10.0.0.4


PC-PT
10.0.0.2
PC-PT
10.0.0.1
PC-PT
10.0.0.5
Hub-PT
Hub0
PC-PT
10.0.0.6
PC-PT
10.0.0.3
2960-24TT
Switch0
PC-PT
10.0.0.7
PC-PT
10.0.0.4


PC-PT
10.0.0.2
PC-PT
10.0.0.1
PC-PT
10.0.0.5
Hub-PT
Hub0
PC-PT
10.0.0.6
PC-PT
10.0.0.3
2960-24TT
Switch0
PC-PT
10.0.0.7
PC-PT
10.0.0.4

Create a topology and simulate sending a simple PDU from source to destination using simple hub and switch as connecting domains.

Steps Involved:

Step1: Drag and drop 3 generic PC's and a generic switch. Connect 3 PC's as peripherals to the switch after setting the IP addresses as 10.0.0.1, 10.0.0.2 and 10.0.0.3 for PC1, PC2 and PC3 respectively and connect them.

Step2: Drag and Drop 3 more generic PC's and a generic hub. Set the IP addresses of PC4, PC5 and PC6 as 10.0.0.4, 10.0.0.5 and 10.0.0.6 respectively connect all the three PC's to the hub.

Step3: Turn on the switch and send a PDU from PC1 to PC2 via switch

Step4: Send a PDU from PC4 with ip address 10.0.0.6 to PC6 via hub. Hub will send PDU to every PC connected to it PC6 will acknowledge and receive it

Step5: connect switch and hub. Send a PDU from PC1 to PC6 via switch and hub.

Command prompt [observation]

PC> PPing 10.0.0.6

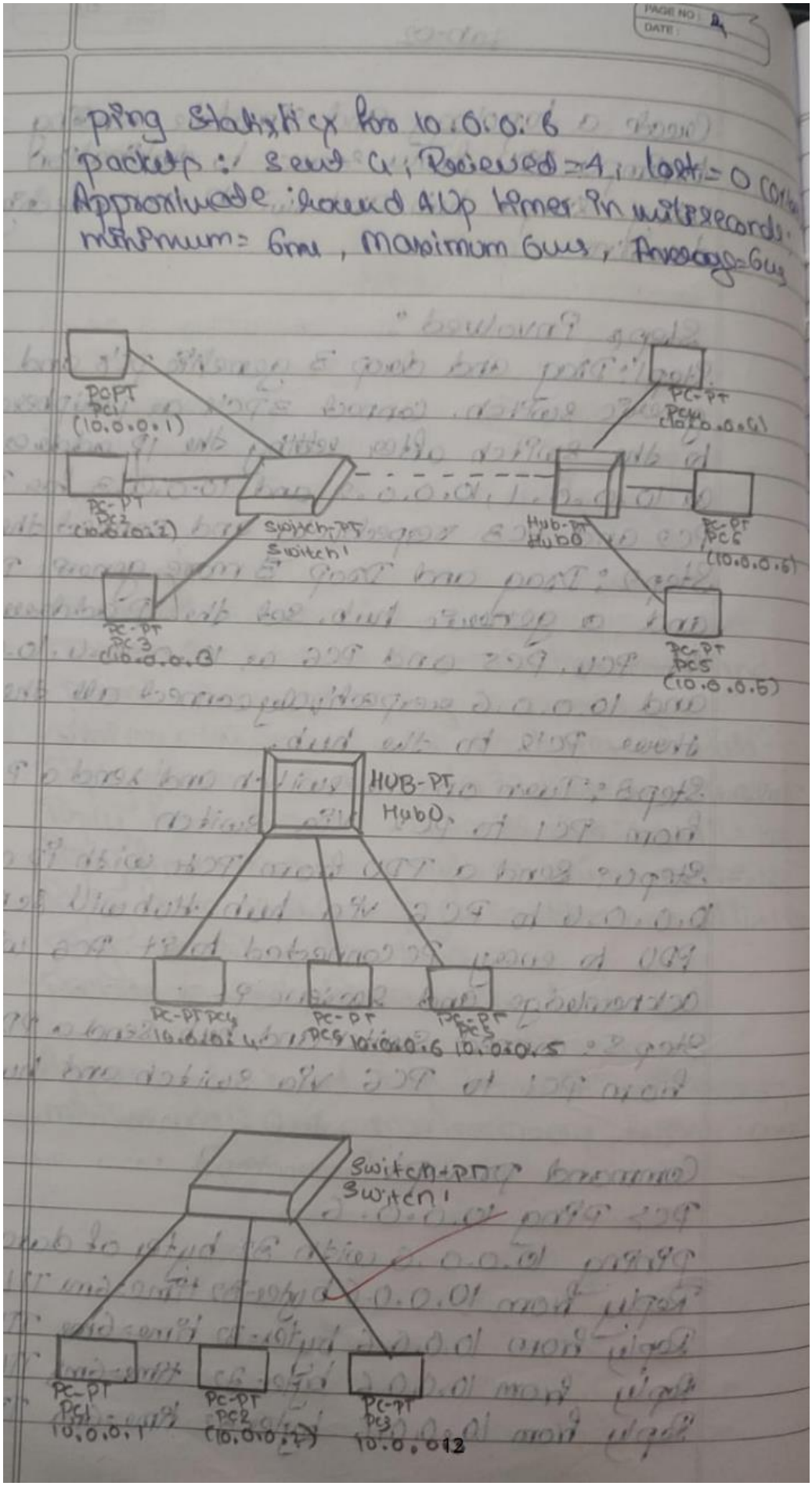
PPinging 10.0.0.6 with 32 bytes of data

Reply from 10.0.0.6 bytes=32 time=6ms TTL=128

Reply from 10.0.0.6 bytes=32 time=6ms TTL=128

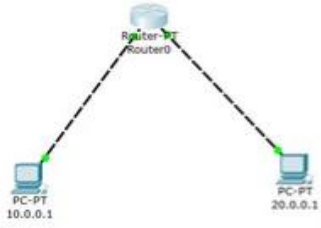
Reply from 10.0.0.6 bytes=32 time=6ms TTL=128

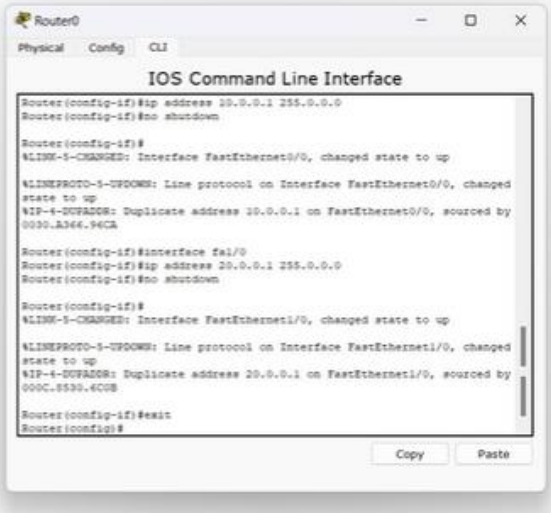
Reply from 10.0.0.6 bytes=32 time=6ms TTL=128

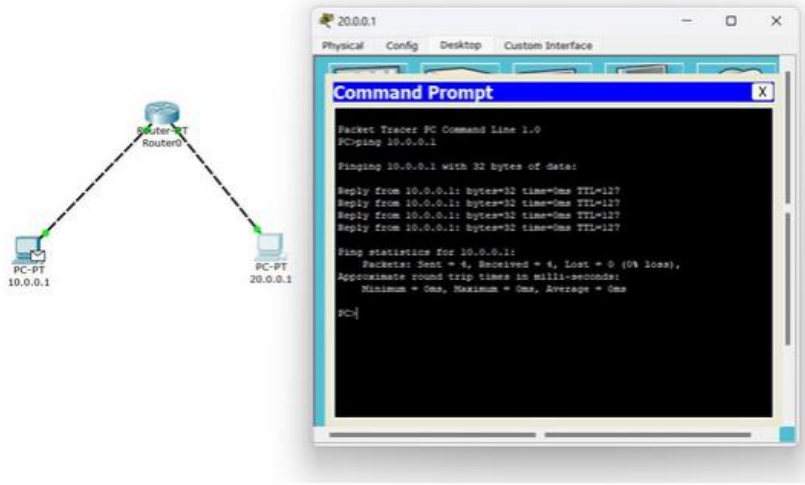

ping Statistics for 10.0.0.6
packets: Sent = 4, Received = 4, Lost = 0
Approximate round trip times in milliseconds:
Minimum = 6ms, Maximum = 6ms, Average = 6ms
PC-PT
PC1
(10.0.0.1)
PC-PT
PC2
(10.0.0.2)
PC-PT
PC3
(10.0.0.3)
Switch-PT
Switch1
Hub-PT
Hub0
PC-PT
PC4
(10.0.0.4)
PC-PT
PC6
(10.0.0.6)
PC-PT
PC5
(10.0.0.5)
HUB-PT
Hub0
PC-PT PC4
PC-PT
PC6
10.0.0.6
PC-PT
PC5
10.0.0.5
Switch-PT
Switch1
PC-PT
PC1
10.0.0.1
PC-PT
PC2
(10.0.0.2)
PC-PT
PC3
10.0.0.3

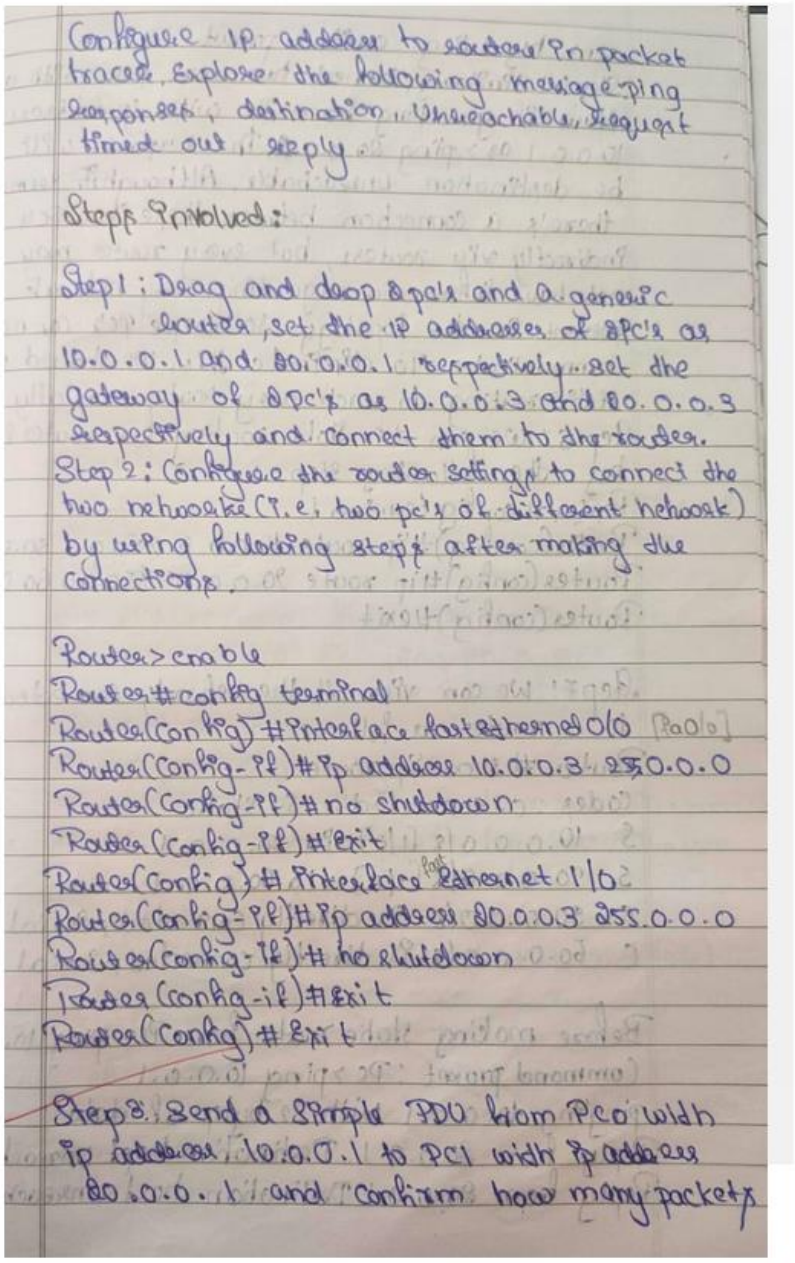
LAB PROGRAM-02

Create a topology consisting of two devices connected with a help of router.









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

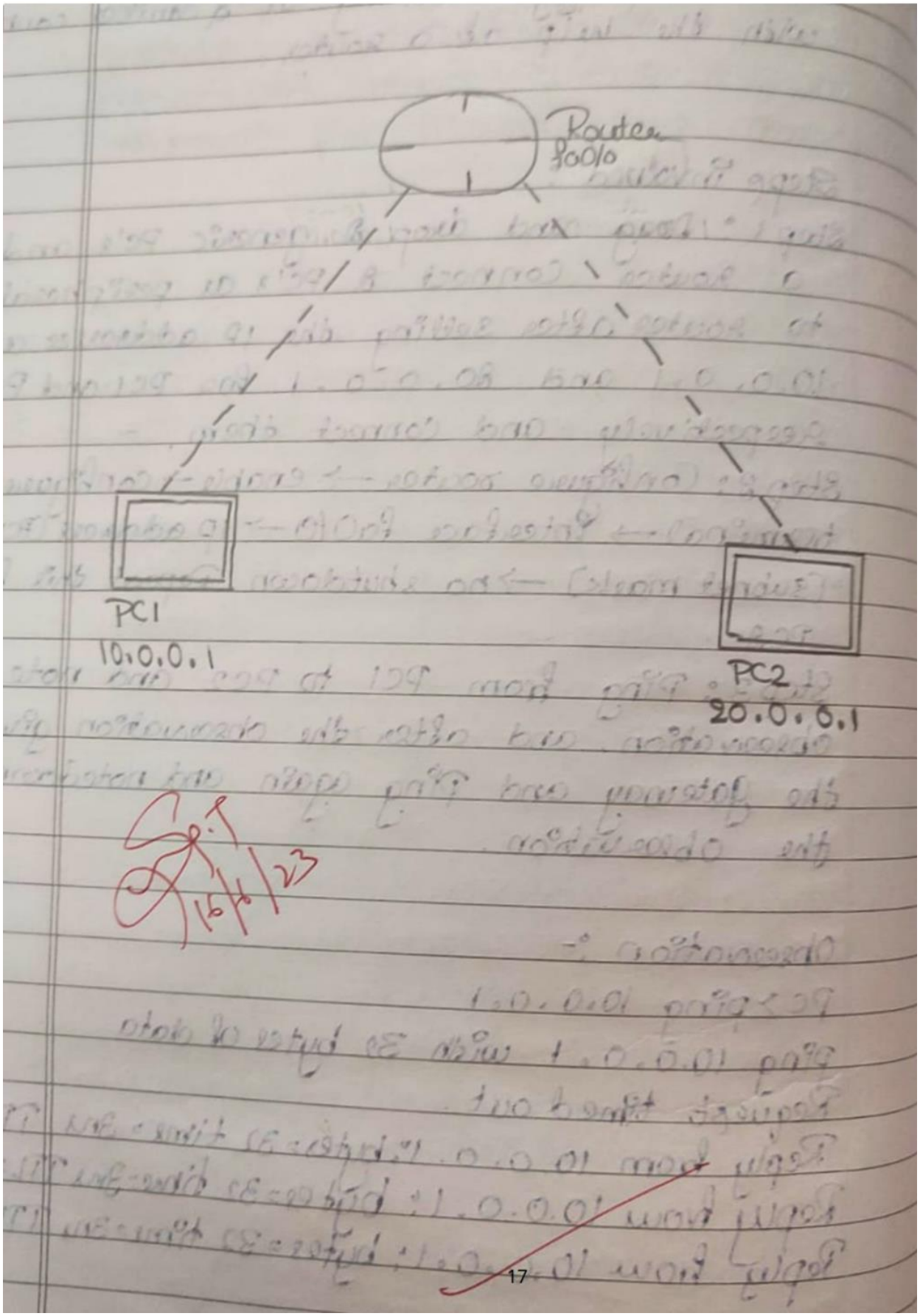
Steps Involved:

Step 1: Drag and drop 2 pc's and a generic router, set the IP addresses of 2 PC's as 10.0.0.1 and 20.0.0.1 respectively. Set the gateway of 2 pc's as 10.0.0.3 and 20.0.0.3 respectively and connect them to the router.

Step 2: Configure the router setting to connect the two networks (i.e. two pc's of different network) by using following steps after making the connections.

Router>enable
Router#config terminal
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 10.0.0.3 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastethernet 1/0
Router(config-if)#ip address 20.0.0.3 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit

Step 3: Send a simple PDU from PC0 with ip address 10.0.0.1 to PC1 with ip address 20.0.0.1 and confirm how many packets



17

sent by using command prout.
Step 6: Similarly, connect two more pcs with ip
Now if you ping from Pc with ip address
10.0.0.1 as > ping 20.0.0.2, the response will
be destination unreachable. Although it seemed
there's a connection between these two Pc's
indirectly via routers, but every router may
not have information regarding every network
present in the topology so these pc's cannot
communicate. To eliminate this, we should use
static routing to reach every router manually
Step 6: We can do static routing for routers
by the following steps:

```
Router # config termina l
Router(config)#ip route 10.0.0.0 255.0.0.0 50.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 60.0.0.1
Router(config)#exit
```

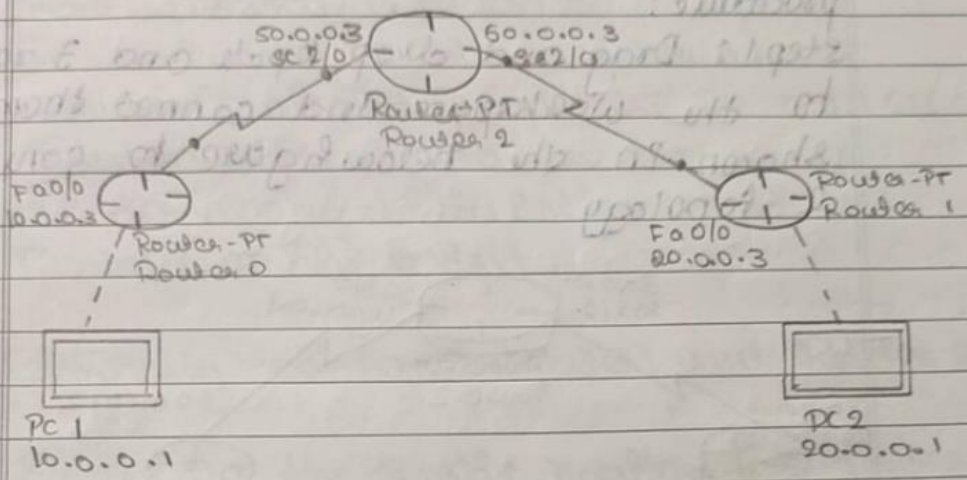
Step 7: We can view all the networks connected
to a router as follows:

```
Router # show ip route
Codes : C - connected, S - static
S  10.0.0.0/8 [1/0] via 50.0.0.1
S  20.0.0.0/8 [1/0] via 60.0.0.1
C  50.0.0.0/8 is directly connected, Serial e/0
C  60.0.0.0/8 is directly connected, Serial e/0
```

Before making static route from Pc2 ping 10.0.0.1

```
Command prompt : Pc > ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: Destination host unreachable
Reply from 20.0.0.1 Destination host unreachable
```

Reply from 20.0.0.1: Destination host unreachable.
Request timed out
ping statistics for 10.0.0.1:
Packets: sent=4, Recieved=0, lost=4 (100% loss)



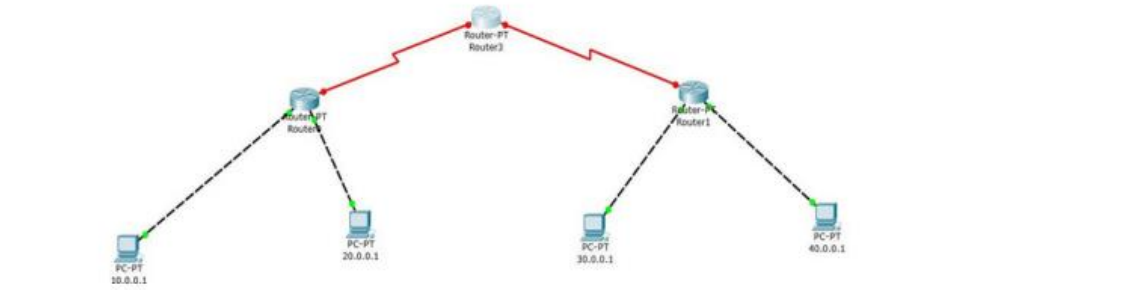
After static route, from PC1 to PC2
Command prompt: PC>ping 20.0.0.1
Ping 20.0.0.1 with 32 bytes of data
Request timed out
Reply from 20.0.0.1: bytes=32 time=3ms TTL=125
Reply from 20.0.0.1: bytes=32 time=3ms TTL=125
Reply from 20.0.0.1: bytes=32 time=ms TTL=125

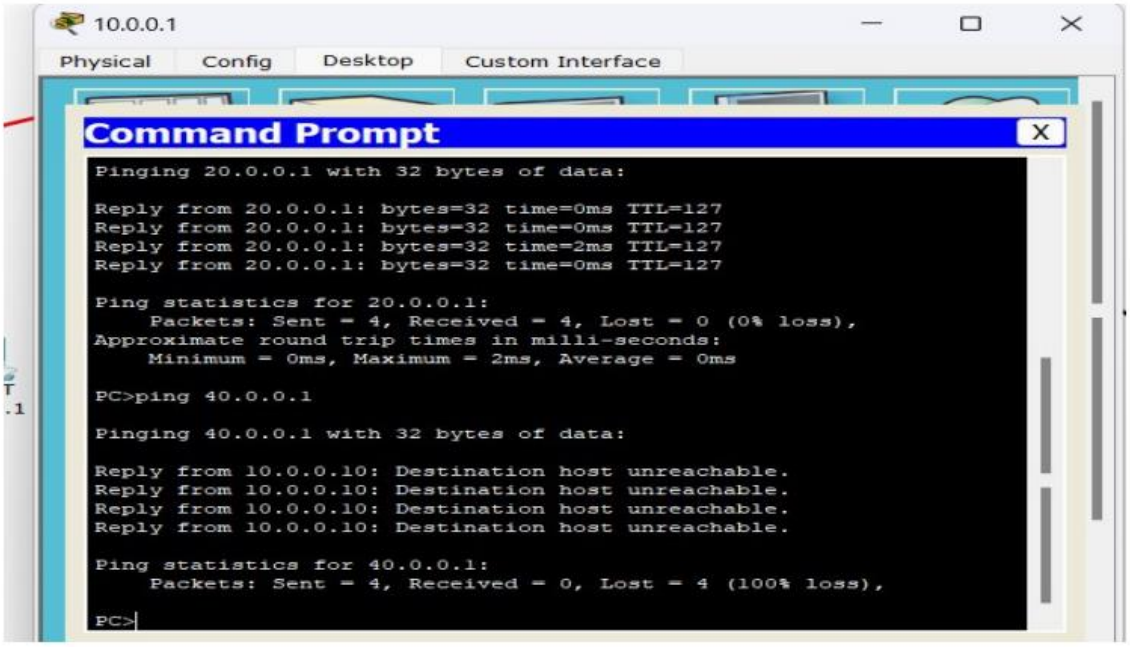
ping statistics for 20.0.0.1:
Packets: sent=4, Recieved=3, lost=1 (25% loss)

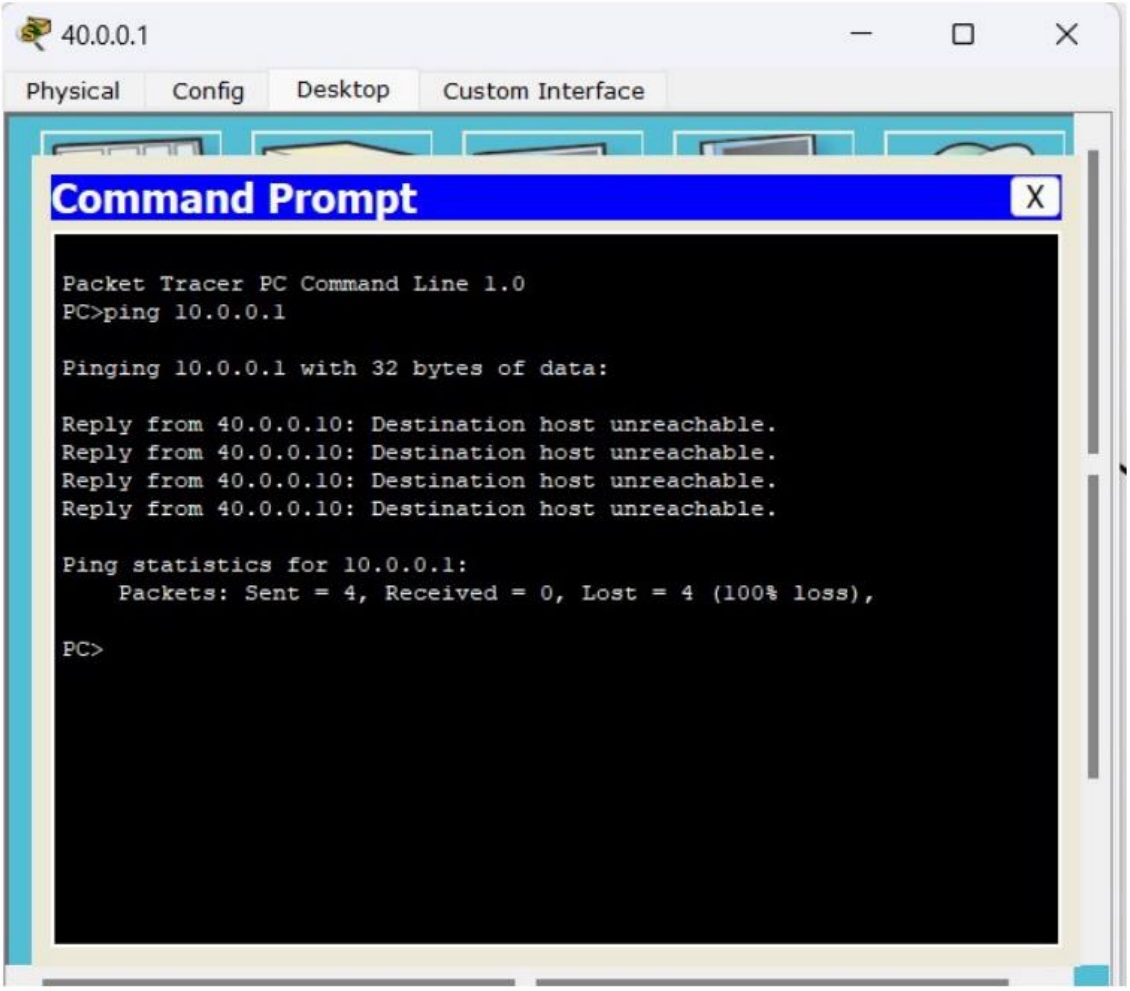
23/6/23

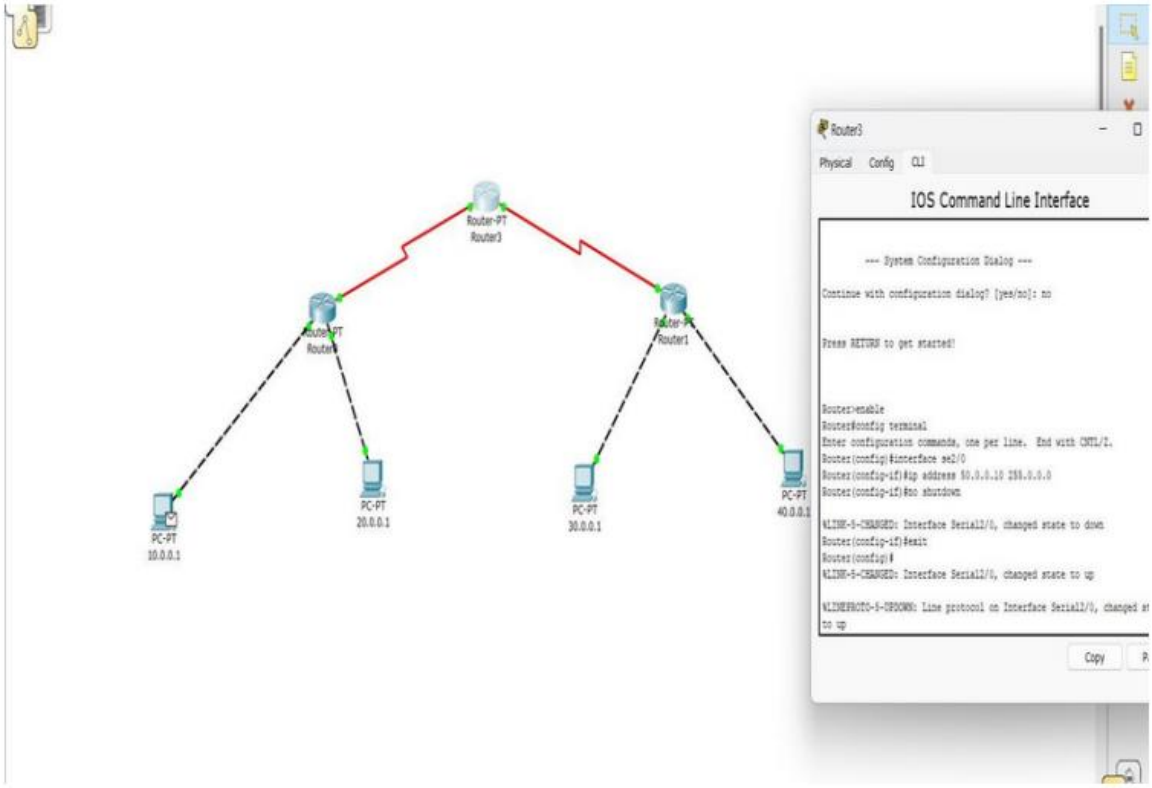
LAB PROGRAM-03

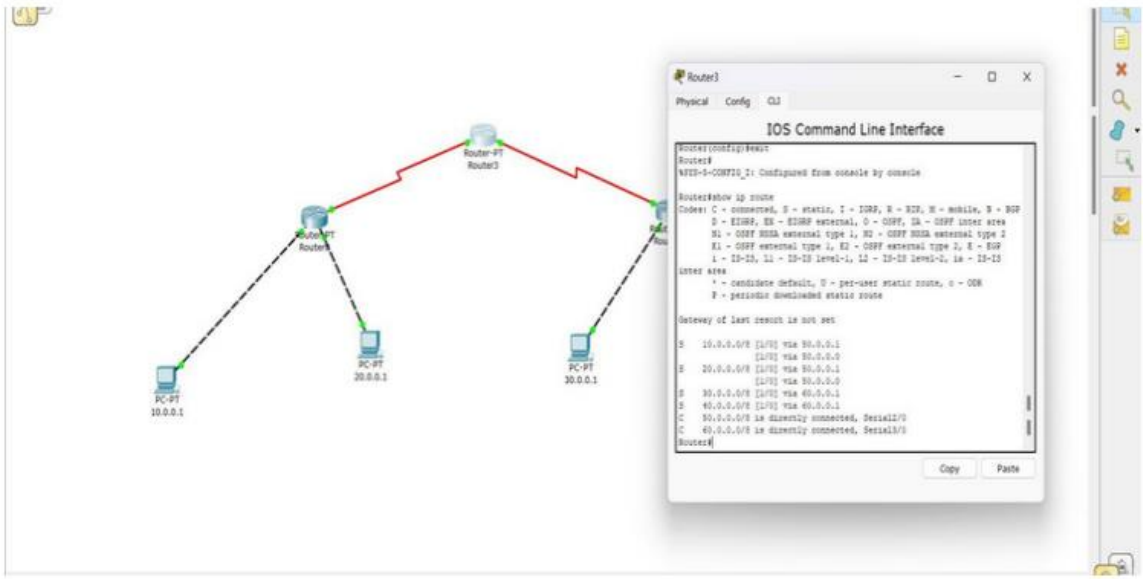
Configure default route, static route to router.

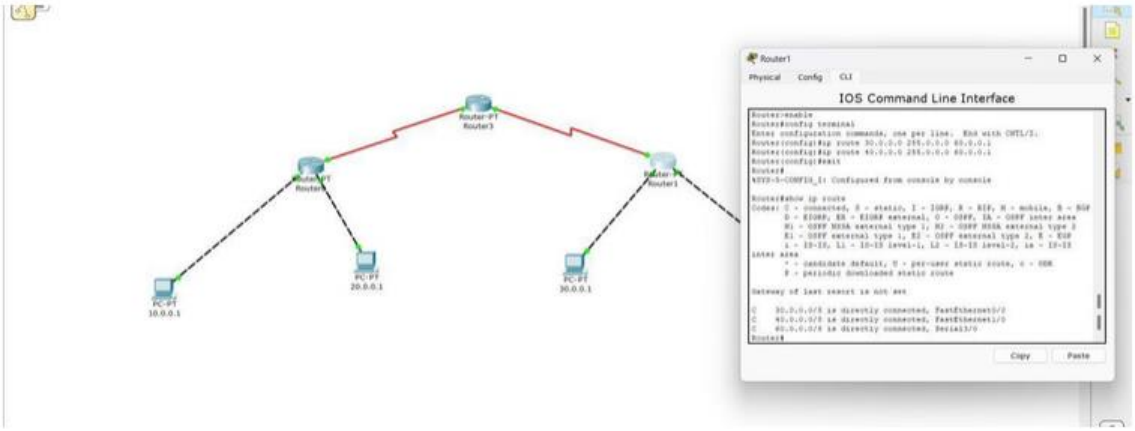


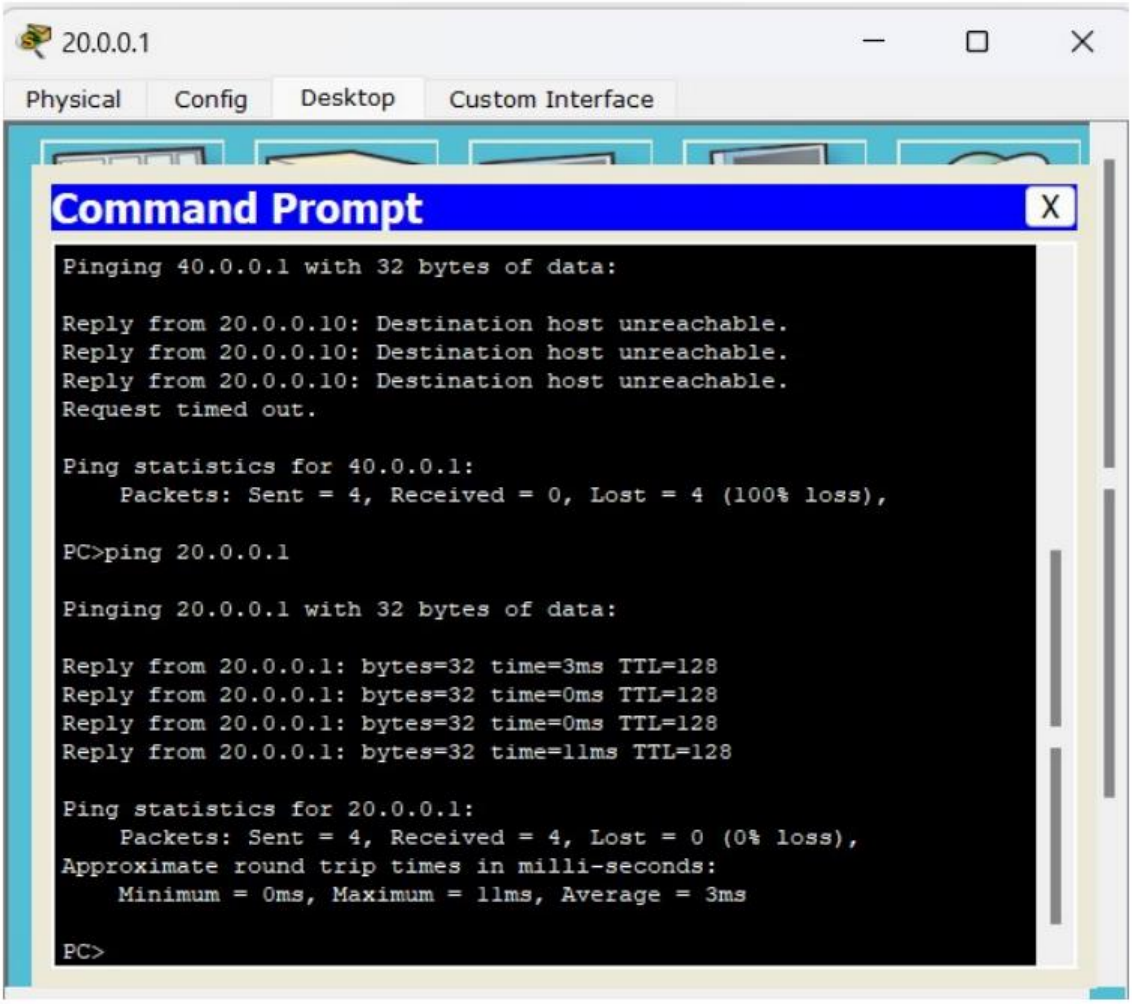



40.0.0.1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 40.0.0.10: Destination host unreachable.
Reply from 40.0.0.10: Destination host unreachable.
Reply from 40.0.0.10: Destination host unreachable.
Reply from 40.0.0.10: Destination host unreachable.
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>


Router3
Physical
Config
CLI
IOS Command Line Interface
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]: no
Press RETURN to get started!
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface se2/0
Router(config-if)#ip address 50.0.0.10 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Copy
Router-PT
Router3
Router-PT
Router0
Router-PT
Router1
PC-PT
10.0.0.1
PC-PT
20.0.0.1
PC-PT
30.0.0.1
PC-PT
40.0.0.1


Router3
Physical
Config
CLI
IOS Command Line Interface
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#show ip route
Gateway of last resort is not set
Copy
Paste
Router-PT
Router3
Router-PT
Router0
PC-PT
10.0.0.1
PC-PT
20.0.0.1
PC-PT
30.0.0.1


Router1
Physical
Config
CLI
IOS Command Line Interface
Copy
Paste

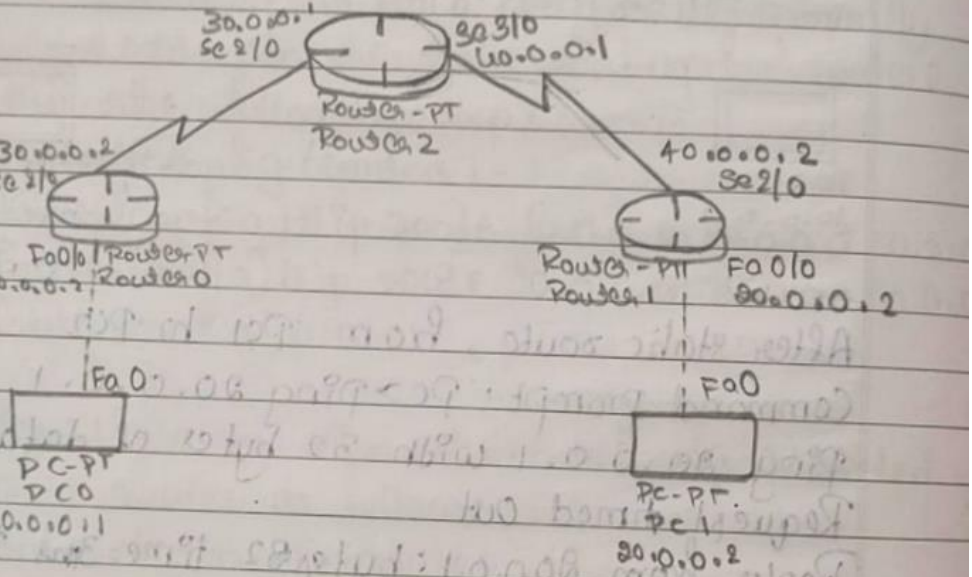

20.0.0.1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 20.0.0.10: Destination host unreachable.
Reply from 20.0.0.10: Destination host unreachable.
Reply from 20.0.0.10: Destination host unreachable.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=3ms TTL=128
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=11ms TTL=128
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 11ms, Average = 3ms
PC>

Default routing and static routing

Aim: To configure default route, static ro to the routers

procedure:

Step 1: Drag and drop 8pc's and 3 routers to the workspace and connect them as shown in the below figure to construct a topology



Step 8: Set the ip address of 1st PC0 as 10.0.0.1 and connect them as shown in the above figure and 2nd PC1 as 20.0.0 also, set the gateway of 2 pcs as 10.0.0.2 and 20.0.0.2 respectively

Step 3: place different network ip addresses as 30.0.0.1 and 40.0.0.1 to the left and right of Router 2 and start Configuring the router interfaces for router0,

Router > enable
Router # config terminal
Router(config) interface fastethernet 0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)# no shut down
Router(config-if)# exit
Router(config) # interface serial
Router(config)# ip address 30.0.0.1 255.00.0
Router(config-if)#no shutdown
Router(config-if)# exit
Router(config) # exit

Similarly, configure Router1 and Router 2

Steps: Do the static routing for router 2 and default routing for router 0 and router 1 as.

for Router 0,

Router > enable
Router #config terminal
Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

for Router 1,

Router # config terminal
Router(config)# ip route 0.0.0.0 00.0.0 40.0.0.1

for Router 2,

Router # config terminal
Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.2
Router(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2
Router(config)# exit
Router #

Now, you can check the routing information

as follows.

Router0,
Router # show ip route
c-connect s-static * - candidate default
Gateway of last resort is 30.0.0.1 to
network 0.0.0.0

C 10.0.0.0/8 is directly connected, Fast
C 30.0.0.0/8 is directly connected, Serial
S* 0.0.0.0/0 [1/0] via 30.0.0.1

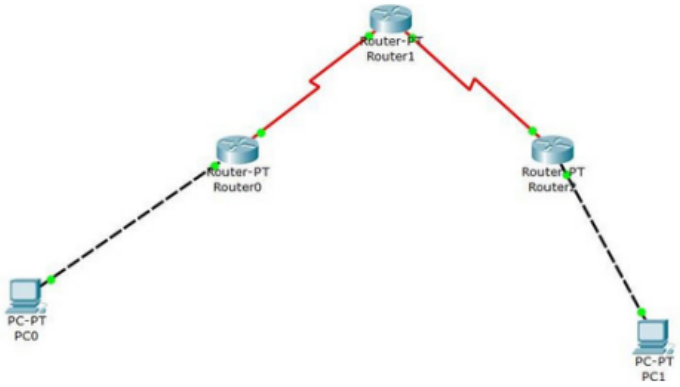
Router2,
Router # show ip route
C-connected S-Static
S 10.0.0.0/8, [1/0] via 30.0.0.2
S 20.0.0.0/8 [1/0] via 40.0.0.2
C 30.0.0.0/8 is directly connected, Serial
C 40.0.0.0/8 is directly connected, Seria

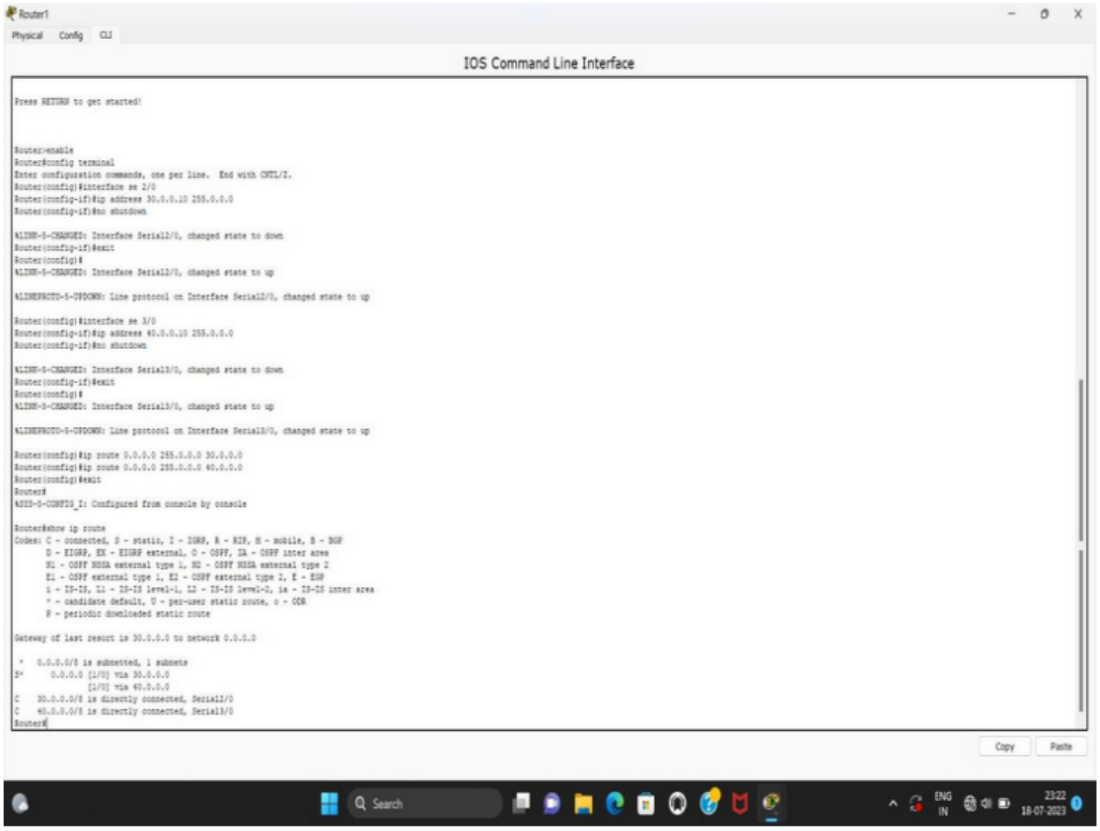
Ping operations: (Result)
from PC0 ping PC1

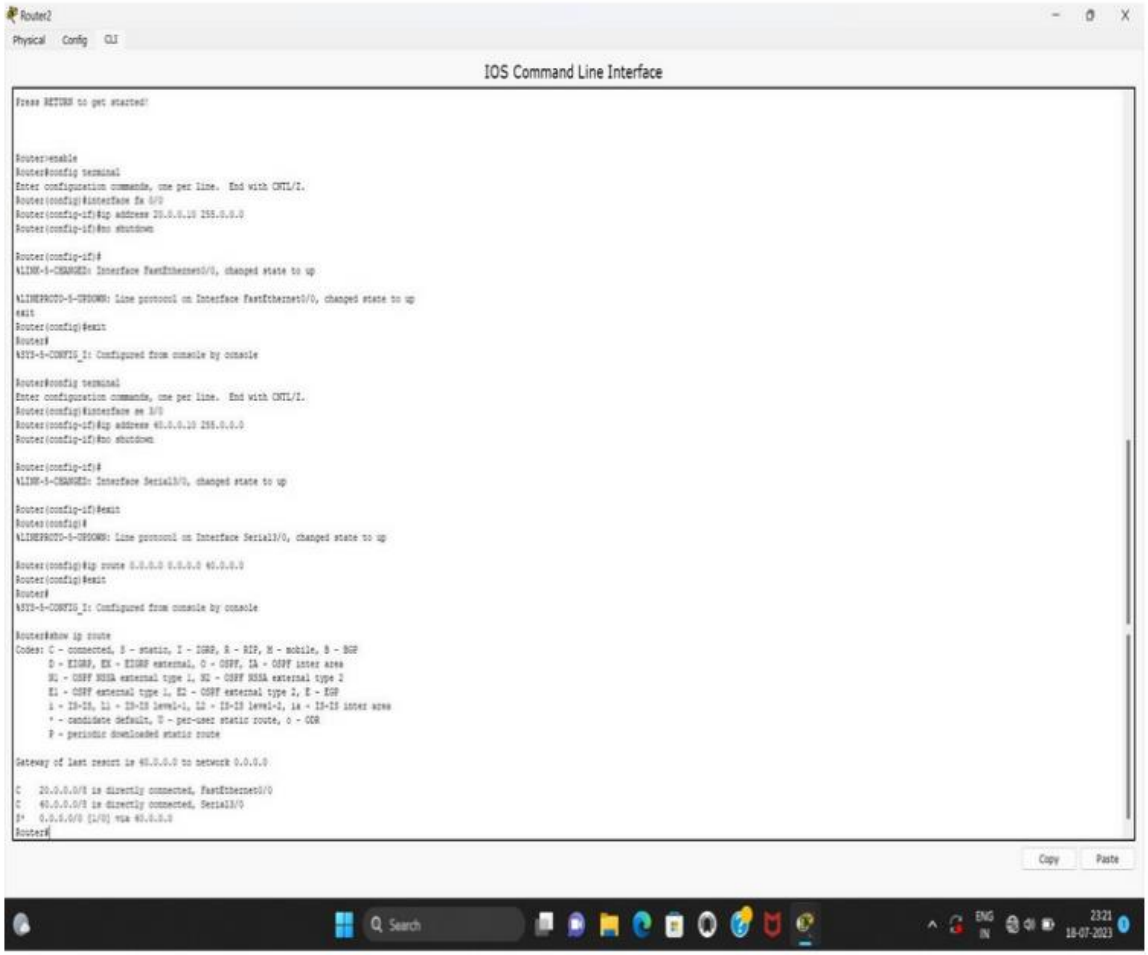
PC> Ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of da
Reply from 20.0.0.1: bytes=32 time=2ms TT
Reply from 20.0.0.1: bytes=32 time=1ms TT
Reply from 20.0.0.1: bytes=32 time=1ms TT
Reply from 20.0.0.1: bytes=32 time=25ms T
Ping statistics for 20.0.0.1:
Packets: Sent=4, Received=4, Lost=0
(0% loss), Approximate=2ms, Maximu

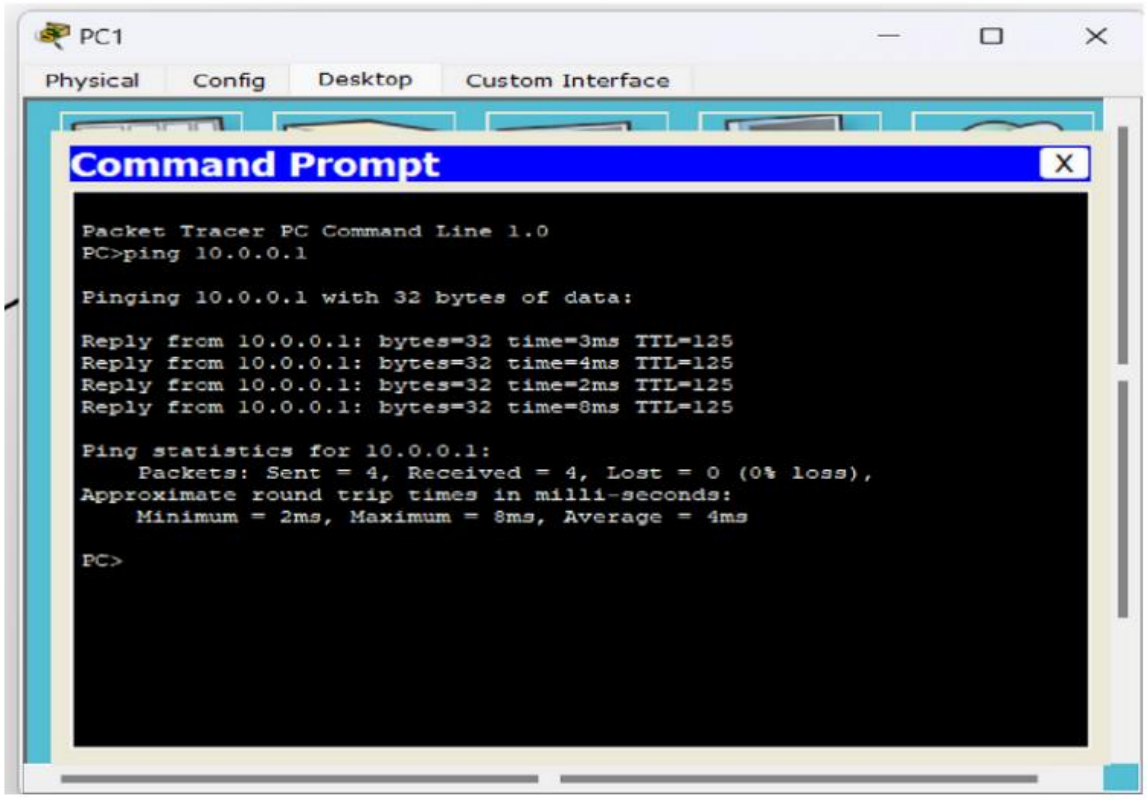
LAB PROGRAM-03

Default routing:-



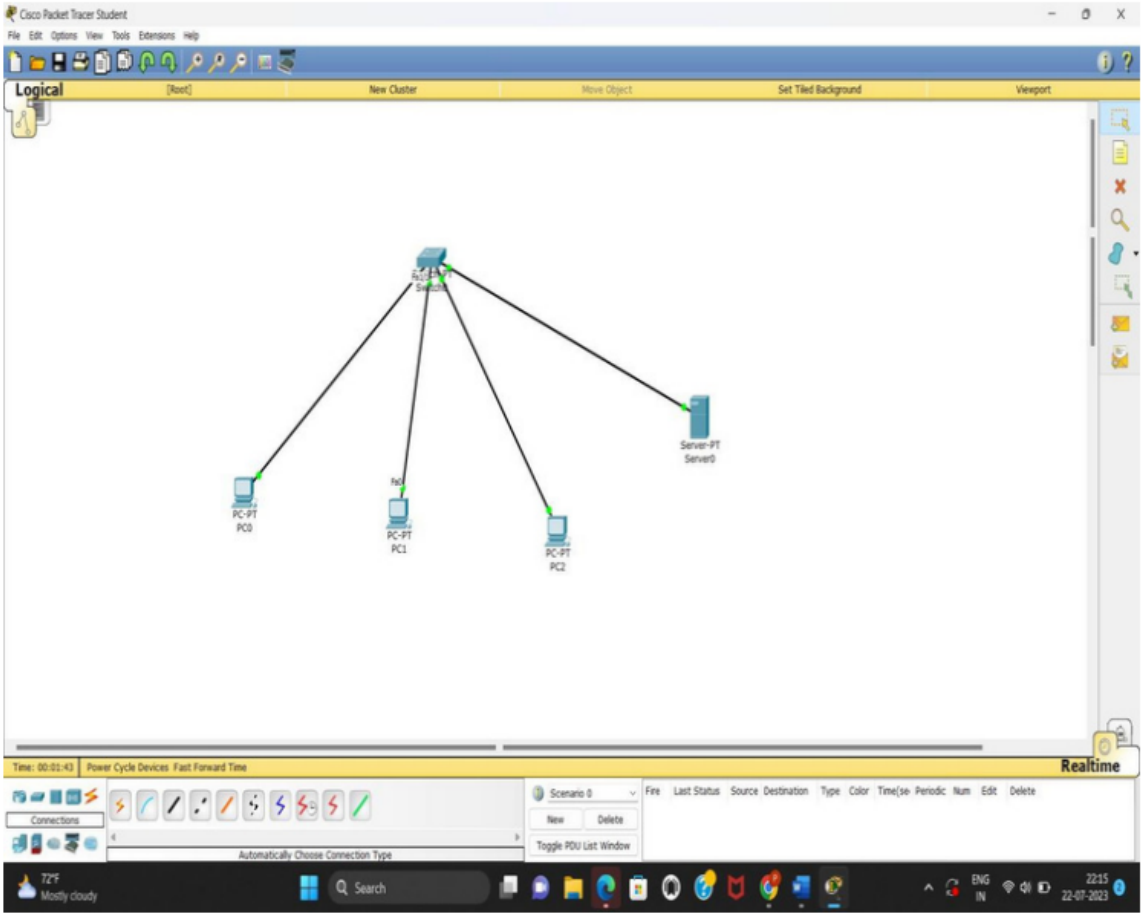


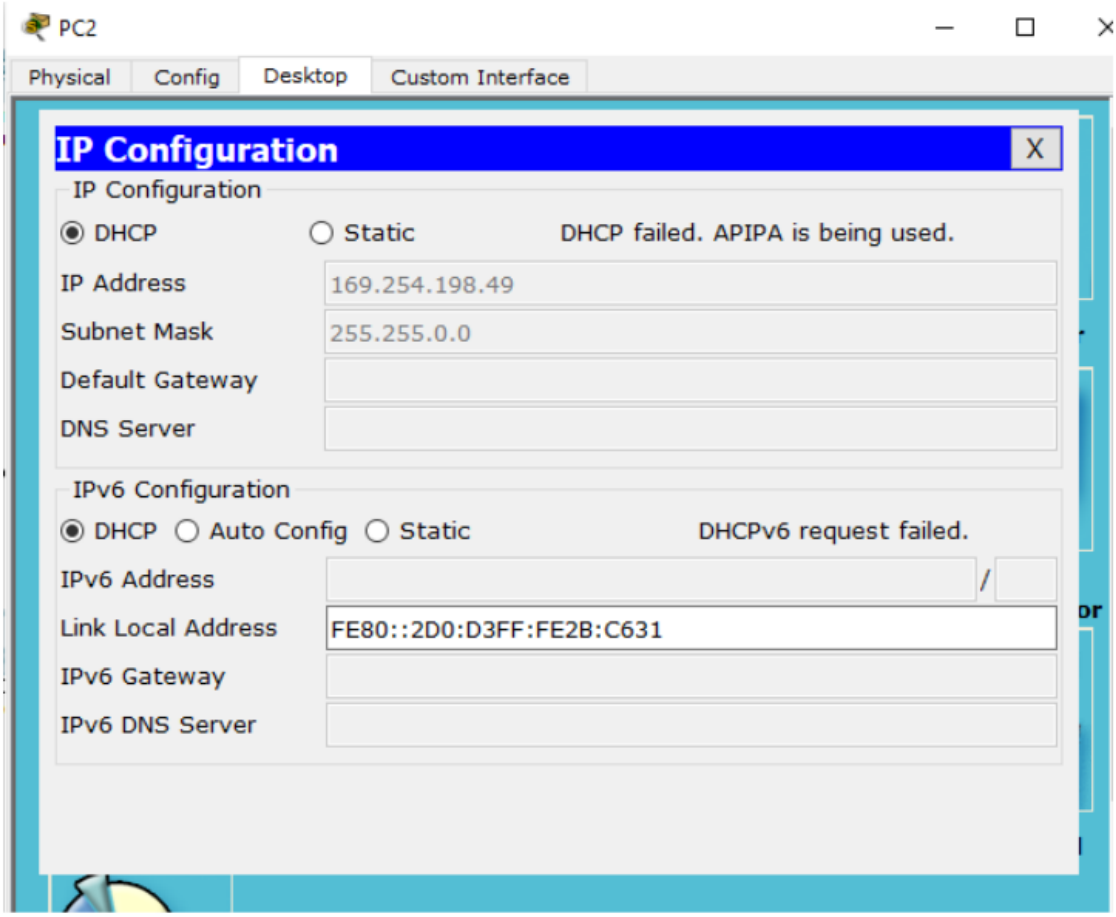

Router2
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started!
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fa 0/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface se 3/0
Router(config-if)#ip address 40.0.0.10 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
Router(config)#ip route 0.0.0.0 0.0.0.0 40.0.0.0
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#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 40.0.0.0 to network 0.0.0.0
C 20.0.0.0/8 is directly connected, FastEthernet0/0
C 40.0.0.0/8 is directly connected, Serial3/0
S* 0.0.0.0/0 [1/0] via 40.0.0.0
Router#
Copy
Paste
Search
ENG
IN
23:21
18-07-2023

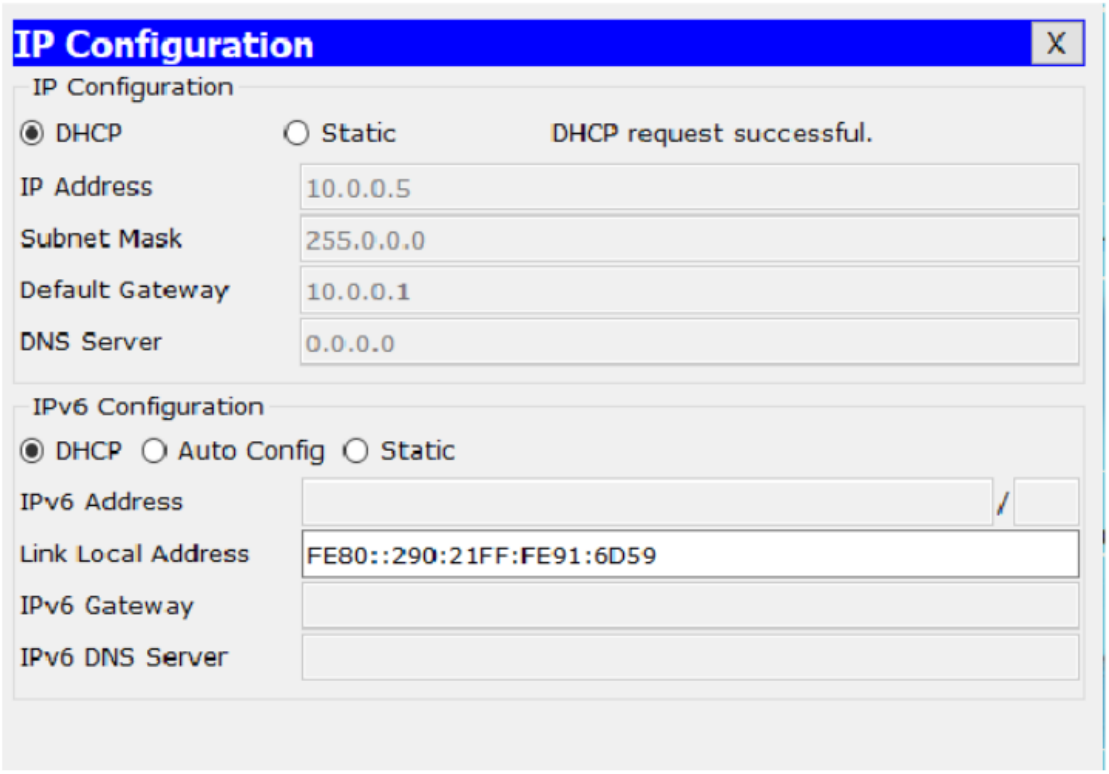

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=3ms TTL=125
Reply from 10.0.0.1: bytes=32 time=4ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 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),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 4ms
PC>

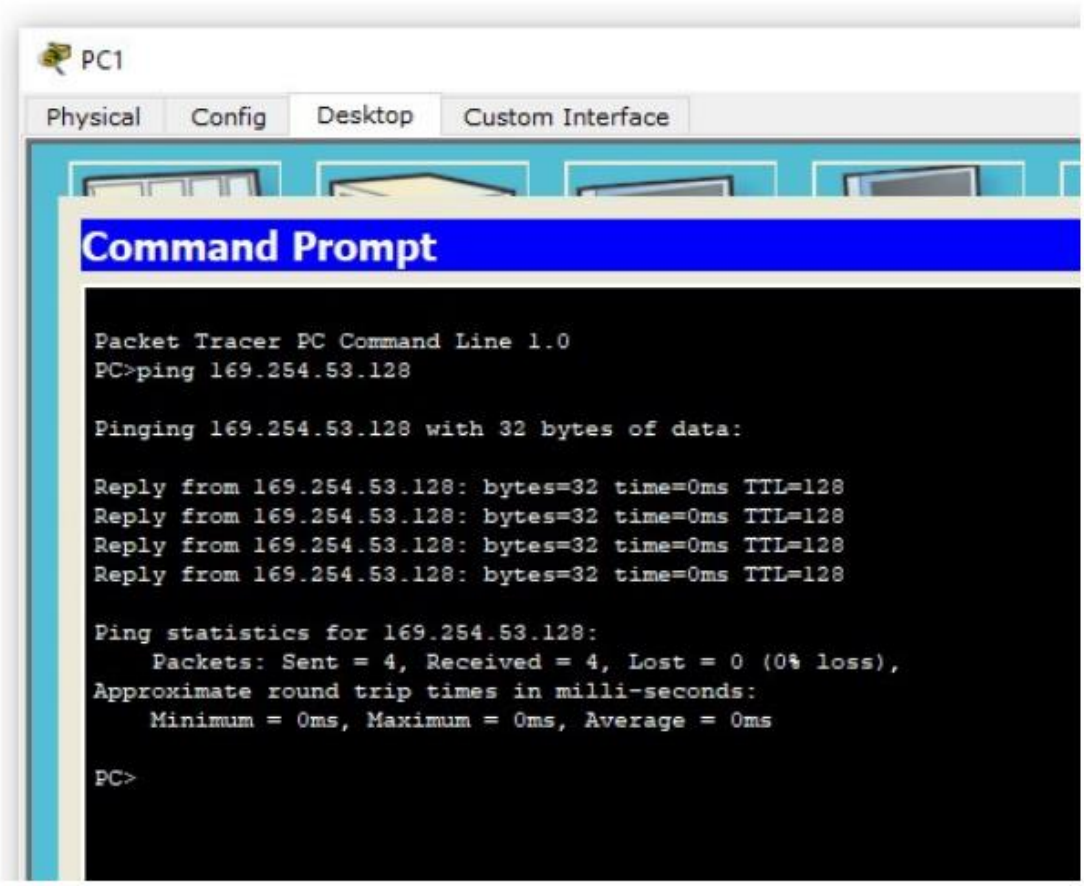
LAB PROGRAM-04

To configure DHCP routing for internal and external LAN.




PC2
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
DHCP failed. APIPA is being used.
IP Address
169.254.198.49
Subnet Mask
255.255.0.0
Default Gateway
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
DHCPv6 request failed.
IPv6 Address
Link Local Address
FE80::2D0:D3FF:FE2B:C631
IPv6 Gateway
IPv6 DNS Server

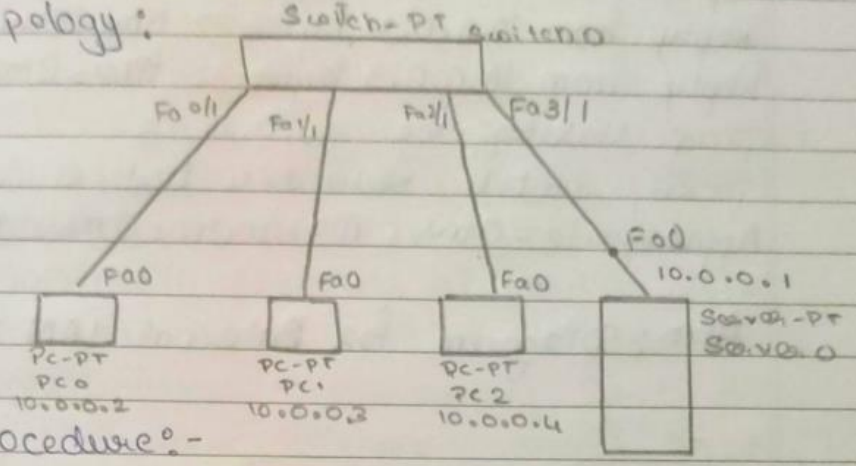

IP Configuration
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
10.0.0.5
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::290:21FF:FE91:6D59
IPv6 Gateway
IPv6 DNS Server


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 169.254.53.128
Pinging 169.254.53.128 with 32 bytes of data:
Reply from 169.254.53.128: bytes=32 time=0ms TTL=128
Reply from 169.254.53.128: bytes=32 time=0ms TTL=128
Reply from 169.254.53.128: bytes=32 time=0ms TTL=128
Reply from 169.254.53.128: bytes=32 time=0ms TTL=128
Ping statistics for 169.254.53.128:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

DHCP within LAN and outside LAN

Aim: To configure DHCP within LAN

Topology:



procedure:-

1. Drag and drop3end device and switch and 1 Server to the workspace and connect them as shown in the above topology
2. Set the ip address of Server as 10.0.0.1 and in service tab, turn on the DHCP and create a serverpool with start address as 10.0.0.2 and save.
3. In all the PC's go to desktop → ip configuration and turn on DHCP, the ip address will be assigned as 10.0.0.2, 10.0.0.3 etc..., These ip addresses are provided by the server through DHCP protocol.

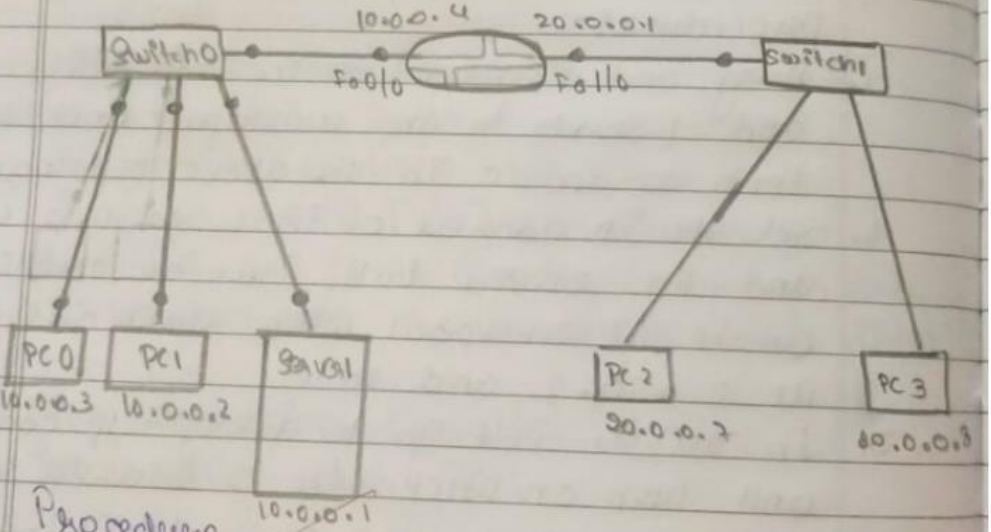
Observation

As soon as the server is assigned a server pool with the appropriate starting IP address all devices within that server pool will have an IP address starting from that starting address

Result:- ping 10.0.0.3

pingPng 10.0.0.3 with 32 bytes of data
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=0ms TTL=127
Reply from 10.0.0.3 bytes=32 time=0ms TTL=127
ping statistics for 10.0.0.3
packets Sent=4 Received=4 Lost=0 (0% loss)
Approximate=0ms, Maximum=0ms, Average=0ms

Aim:- Diagram for Internal LAN DHCP routing



Procedure

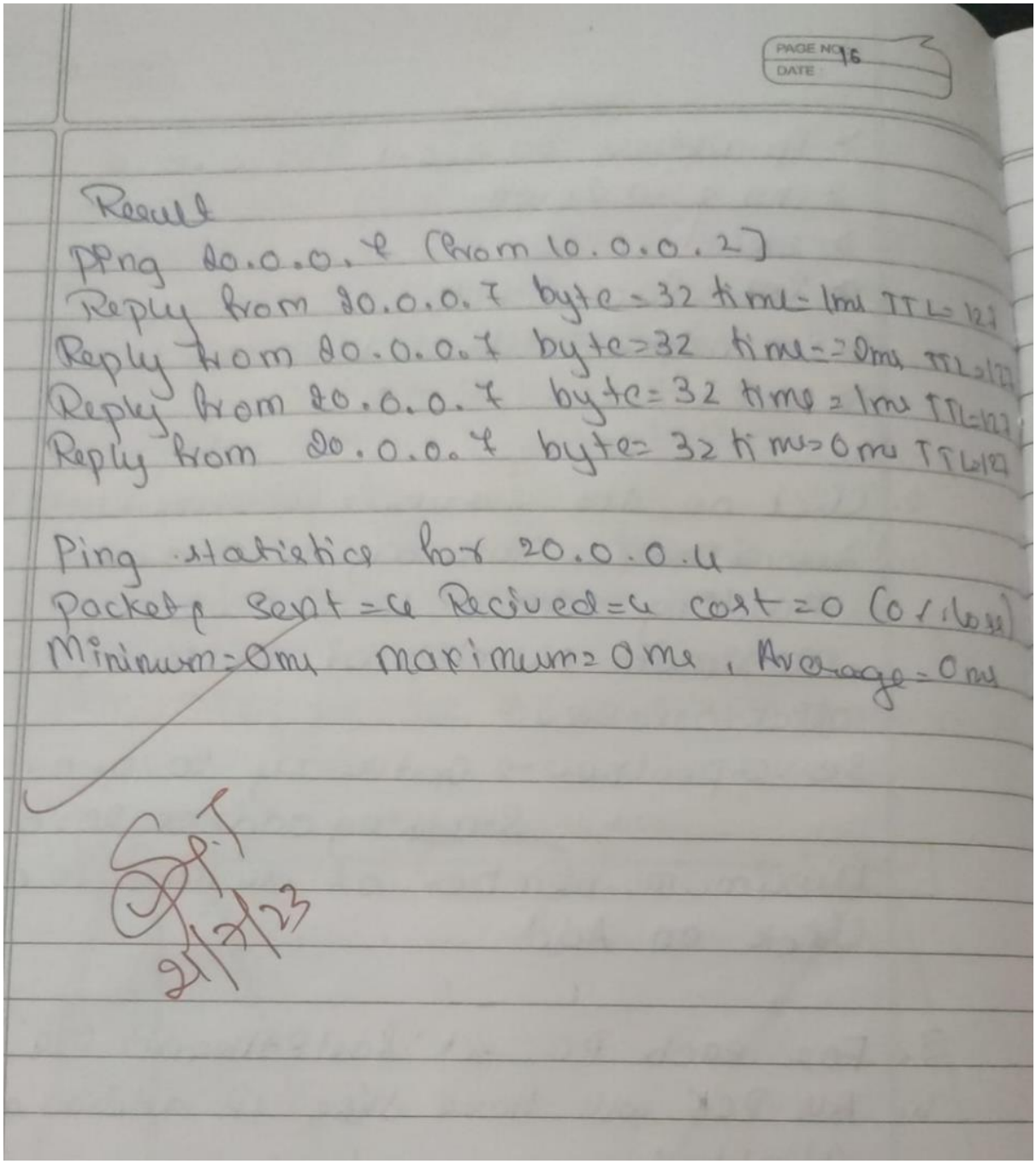
1. Drag and drop the end devices, Connect them as shown, click on the router → CLI
> enable
> configure terminal
> interface Fa0/0
> ip address 10.0.0.4 255.0.0.0
> no shutdown
> exit
> interface Fa1/0

> ip address 20.0.0.1 255.0.0.0
> no shut down
> exit
> interface Fa1/0
> ip helper-address 10.0.0.1
> exit

2. Click on the server → services → DHCP
serverpool → Gateway 10.0.0.4
starting address 10.0.0.2
maximum Number of devices = 10
click on save
serverpool(new) → Gateway 20.0.0.1
starting address 20.0.0.2
Maximum number of devices = 100
Click on Add

3. For each PC, set FastEthernet 0/0 to DHCP
4. All PCs will have their IP address automatically allotted
5. Ping PC's to test connectivety

Observation
For a server, IP address must be allotted statically
Unless a gateway in assigned to the server for serverpool(new), the PC's and the server LAN will have a random address; 169.254.0.1
Once the gateway is assigned to the server the PC's will automatically be assigned the gateways.


PAGE NO. 16
DATE :

Result

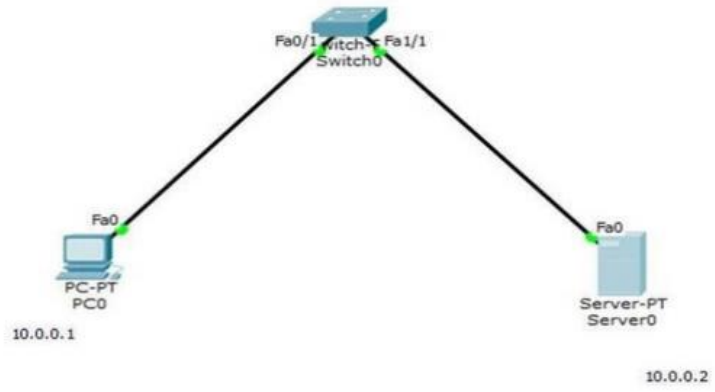
Ping 20.0.0.4 [from 10.0.0.2]
Reply from 20.0.0.4 byte=32 time=1ms TTL=127
Reply from 20.0.0.4 byte=32 time=0ms TTL=127
Reply from 20.0.0.4 byte=32 time=1ms TTL=127
Reply from 20.0.0.4 byte=32 time=0ms TTL=127

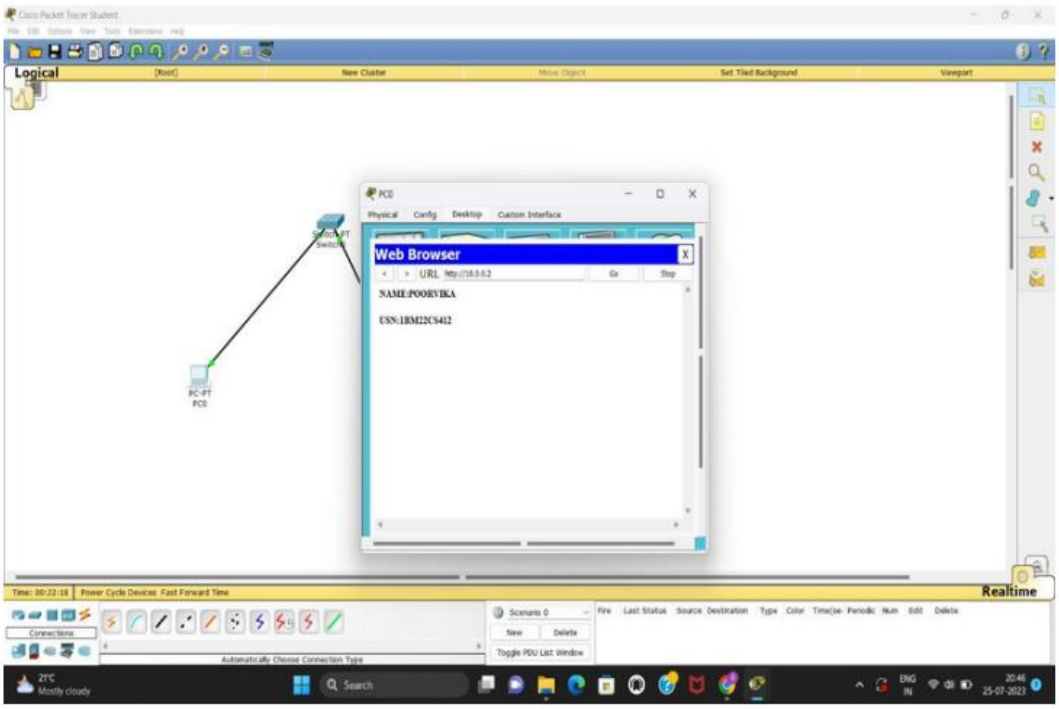
Ping statistics for 20.0.0.4
Packets Sent=4 Recived=4 Lost=0 (0% loss)
Minimum=0ms maximum=0ms, Average=0ms

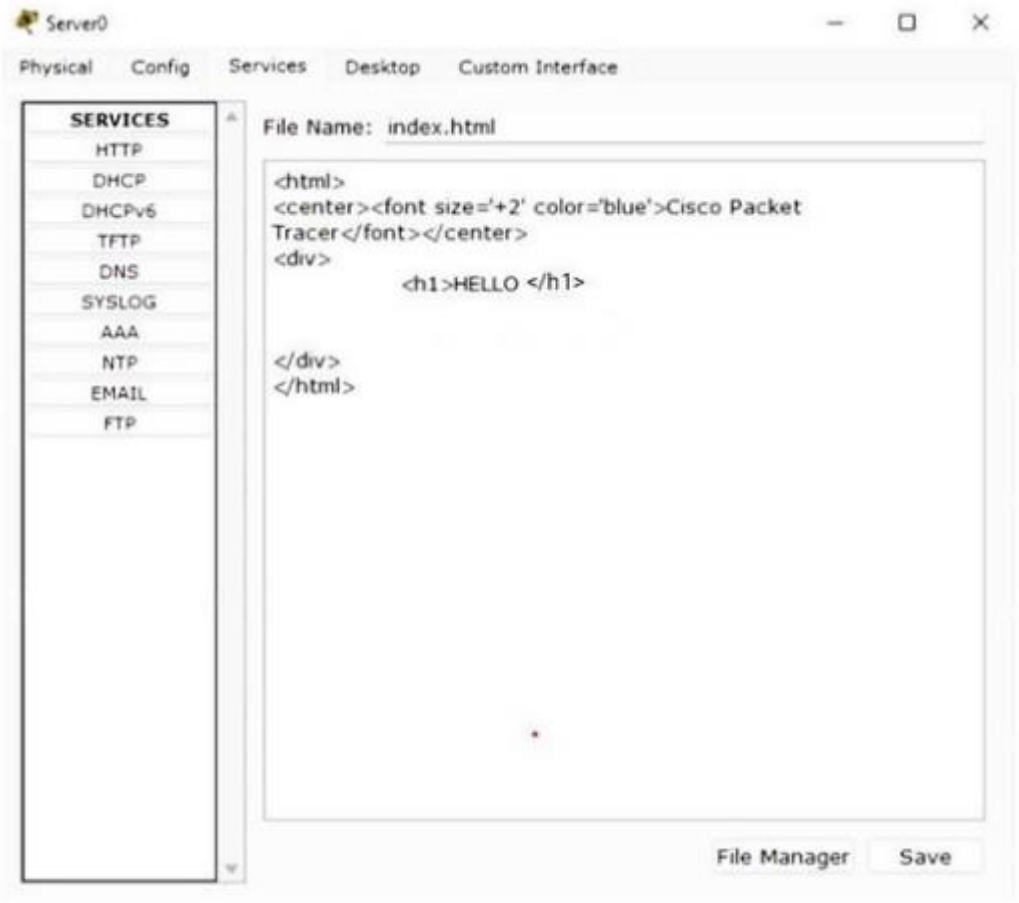
Sp.T
21/7/23

LAB PROGRAM-05

DNS(Using one pc and server):-

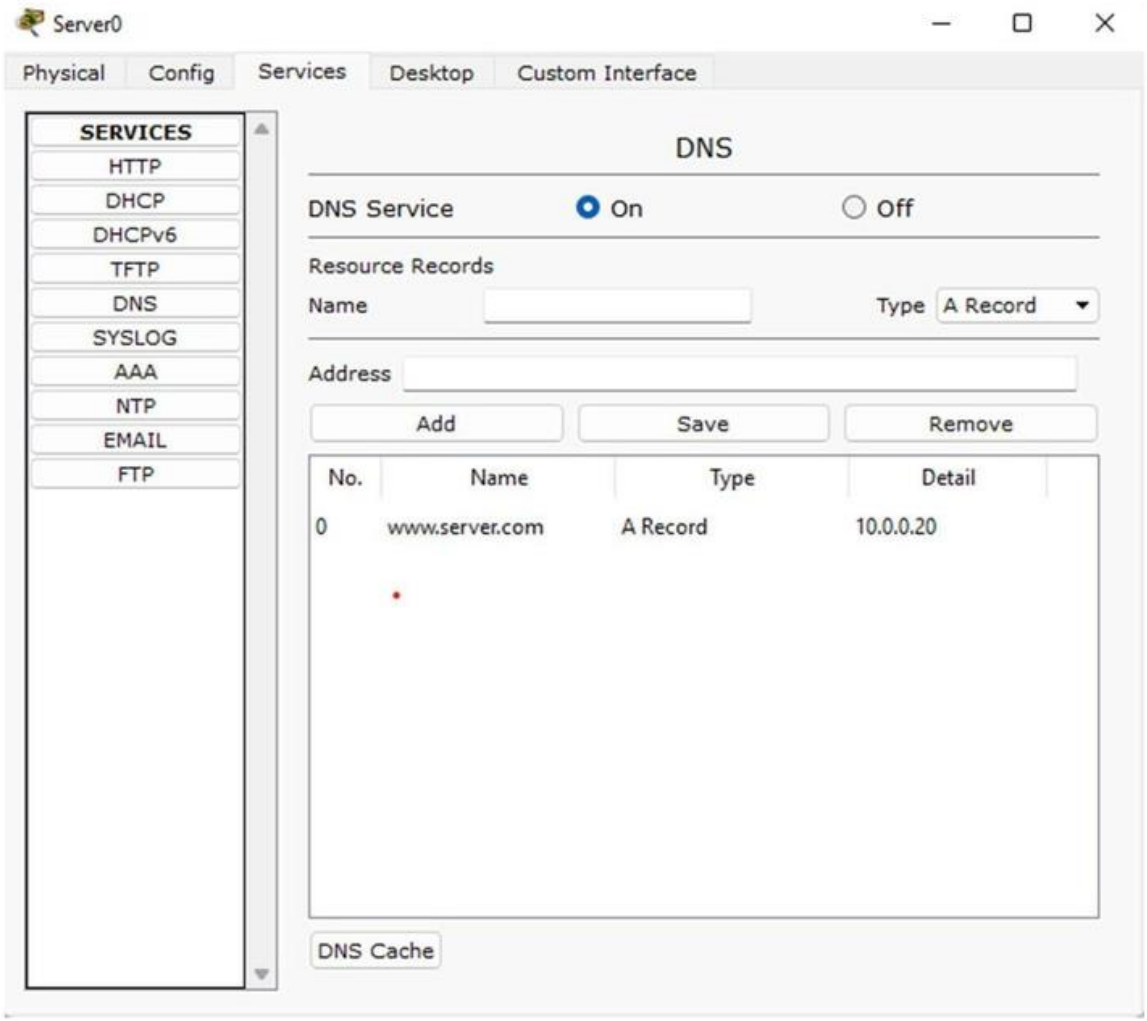




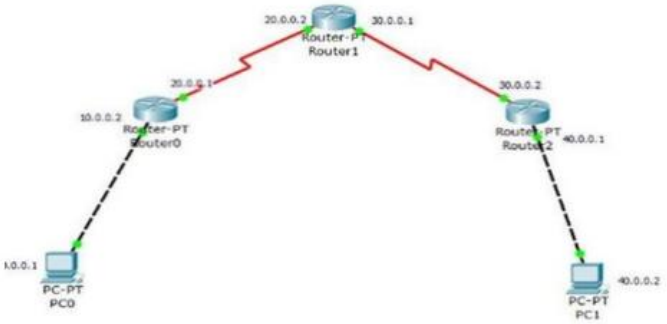

Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
File Name: index.html
<html>
Cisco Packet Tracer

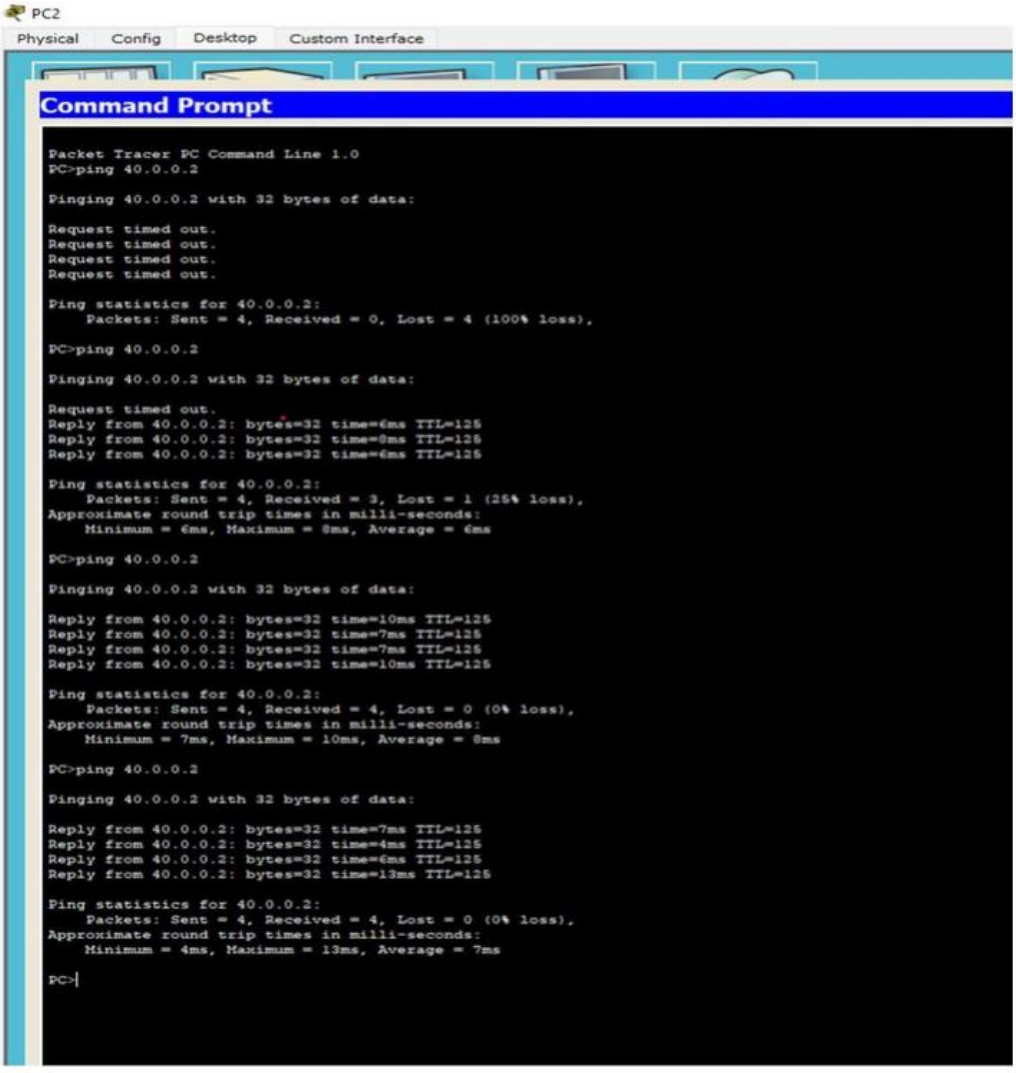
<h1>HELLO </h1>

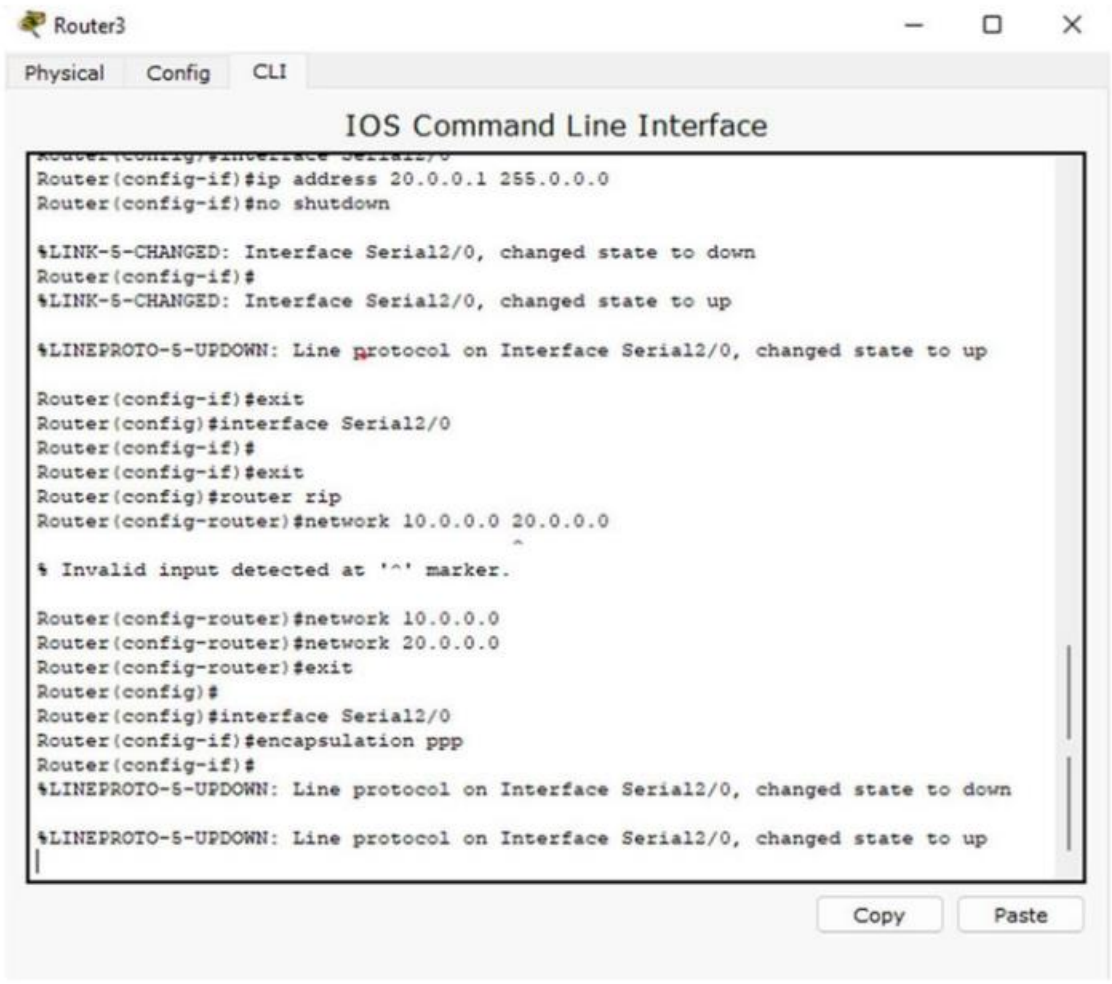
</html>
File Manager
Save

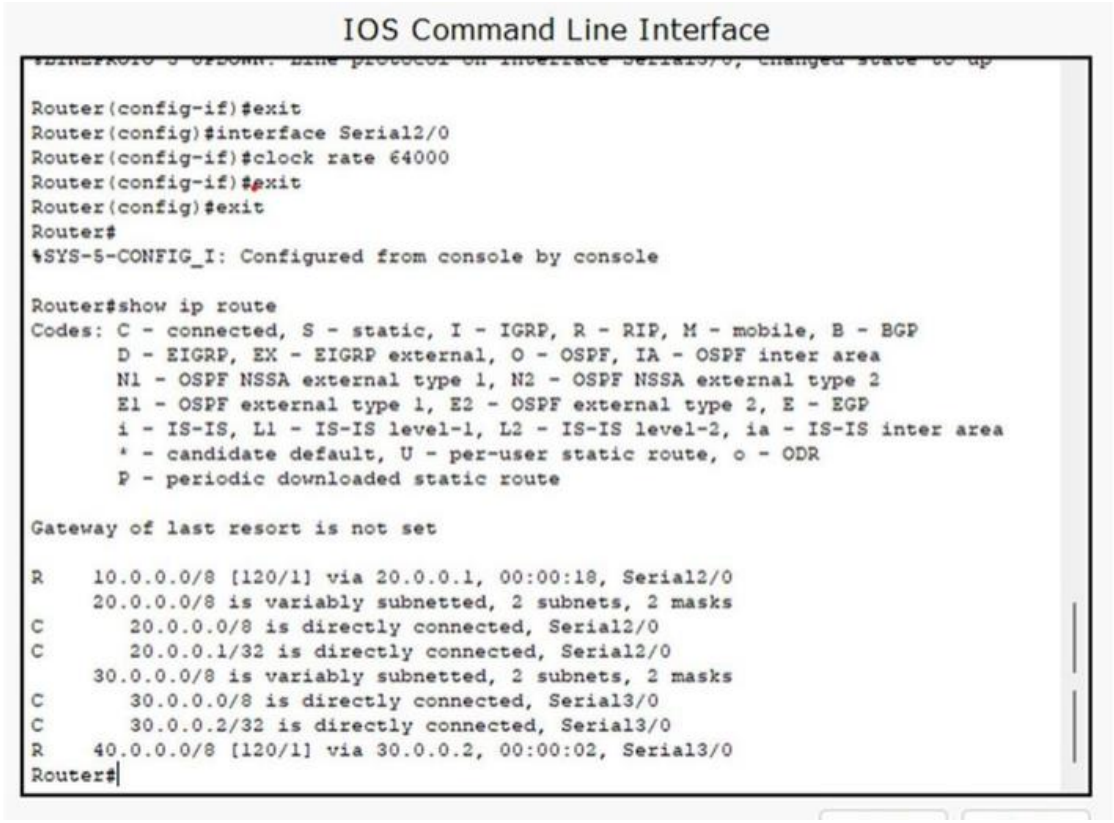

Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
Type
A Record
Address
Add
Save
Remove
No.
Name
Type
Detail
0
www.server.com
A Record
10.0.0.20
DNS Cache

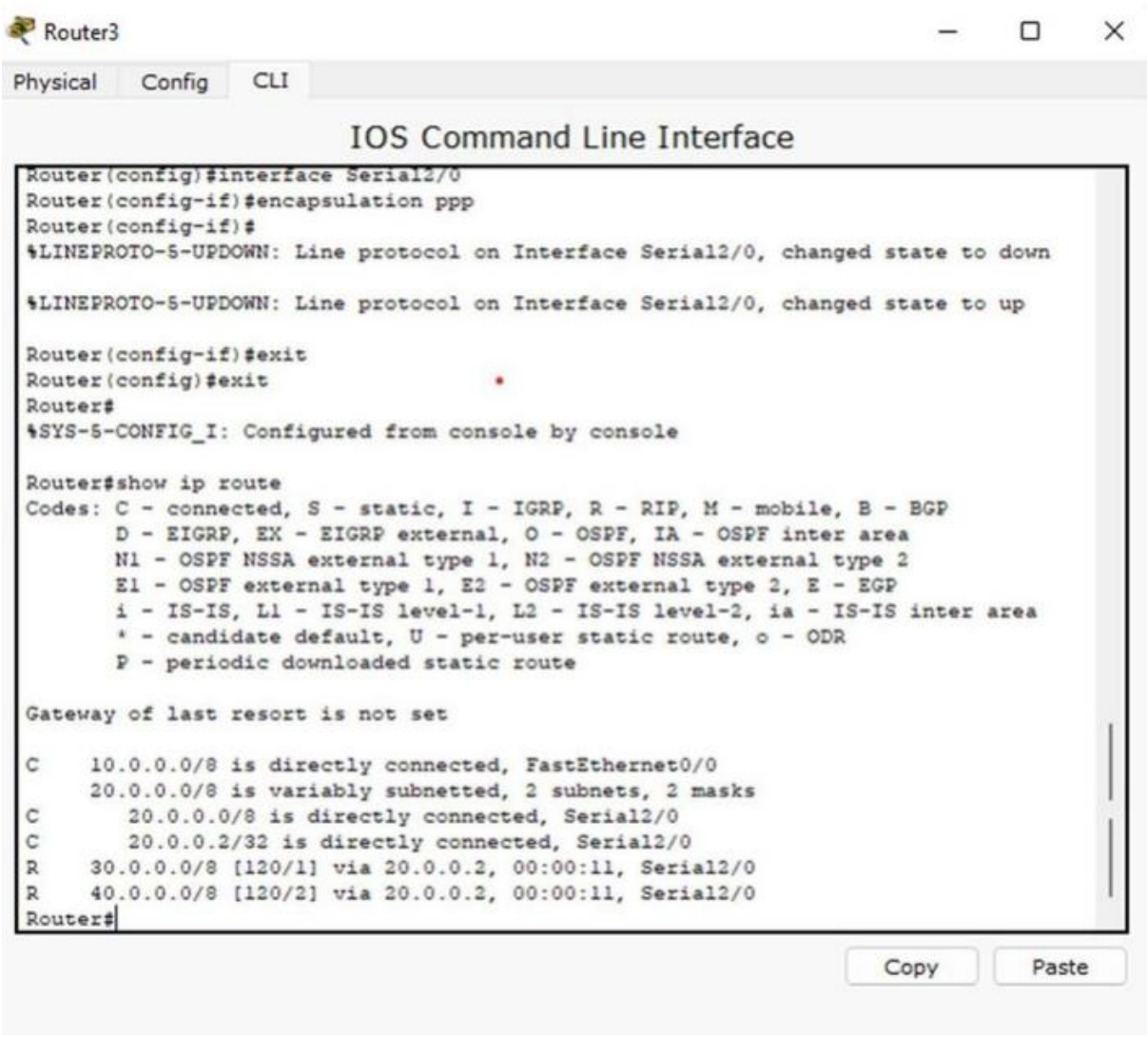
RIP(Using 3 routers and 2 pcs):-



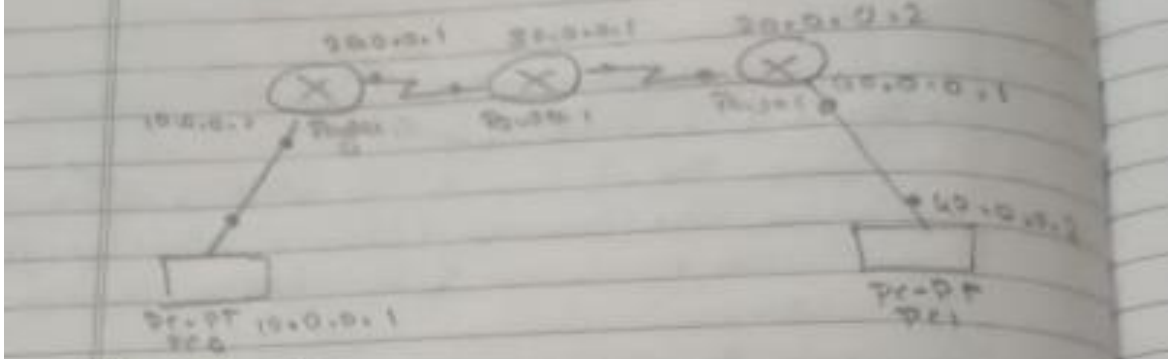



Router3
Physical
Config
CLI
IOS Command Line Interface
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0 20.0.0.0
% Invalid input detected at '^' marker.
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#encapsulation ppp
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Copy
Paste


IOS Command Line Interface
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#clock rate 64000
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#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
R 10.0.0.0/8 [120/1] via 20.0.0.1, 00:00:18, Serial2/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.1/32 is directly connected, Serial2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial3/0
C 30.0.0.2/32 is directly connected, Serial3/0
R 40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:02, Serial3/0
Router#


Router3
Physical
Config
CLI
IOS Command Line Interface
Router(config)#interface Serial2/0
Router(config-if)#encapsulation ppp
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#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
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.2/32 is directly connected, Serial2/0
R 30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:11, Serial2/0
R 40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:11, Serial2/0
Router#
Copy
Paste

Router Interface Protocol (RIP)



Procedure :-
Create a topology with 2 PC's 3 Routers
Configure PC with ip address and gateway
Configure Routers
for Router 0
- enable
- config terminal
- Interface fast ethernet 0/0
- ip address 10.0.0.1 255.0.0.0
- Interface Serial 2/0
- ip address 10.0.0.2
- Encapsulation PPP
- clock rate 64000
- No shut

→ Encapsulation PPP is given to all serial connections and clock rate 64000 is given whenever clock is present

→ Configure all routers
for Router 0
Router enable

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

o/p:
In PC0

ping 40.0.0.2
Ping 40.0.0.2 with 32 bytes of data
Request timed out
Reply from 40.0.0.2 byte=32 time=2ms TTL=125
Reply from 40.0.0.2 bytes=32 time=2ms TTL=125
Reply from 40.0.0.2 byte=32 time=2ms TTL=125
Reply from 40.0.0.2 bytes=32 time=0ms TTL=125

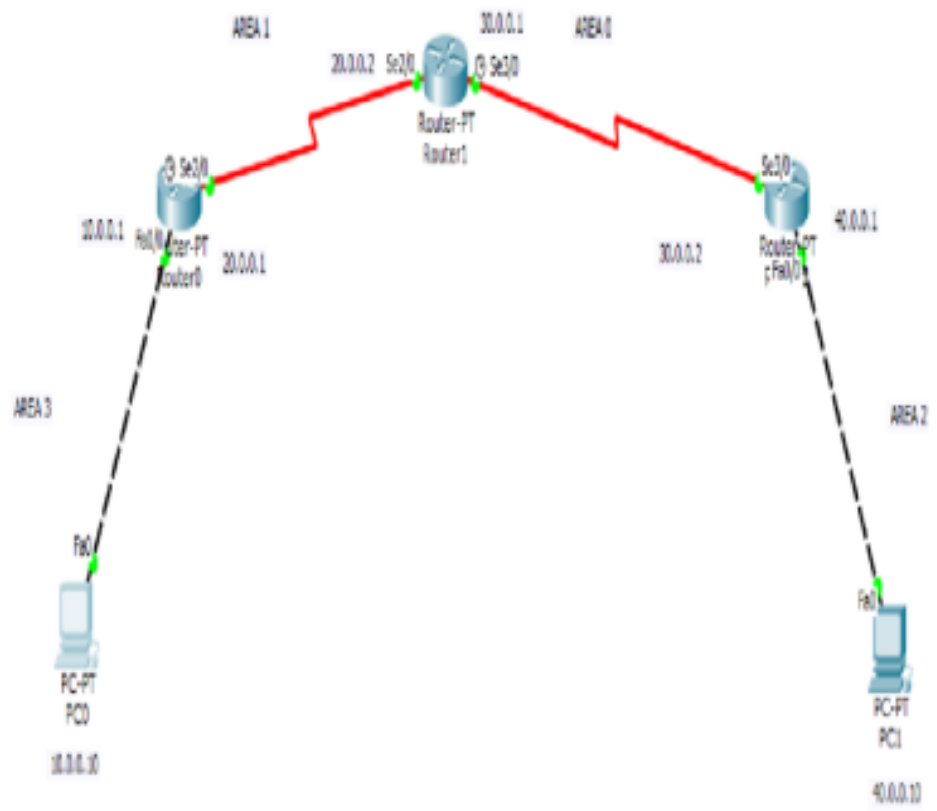
Ping statistics for 40.0.0.2
Packets: Sent=6, Received=3 Lost=1 (25% loss)
Approximate round trip time in milliseconds
Minimum=0ms Maximum=13ms Average=4ms

14/8/23

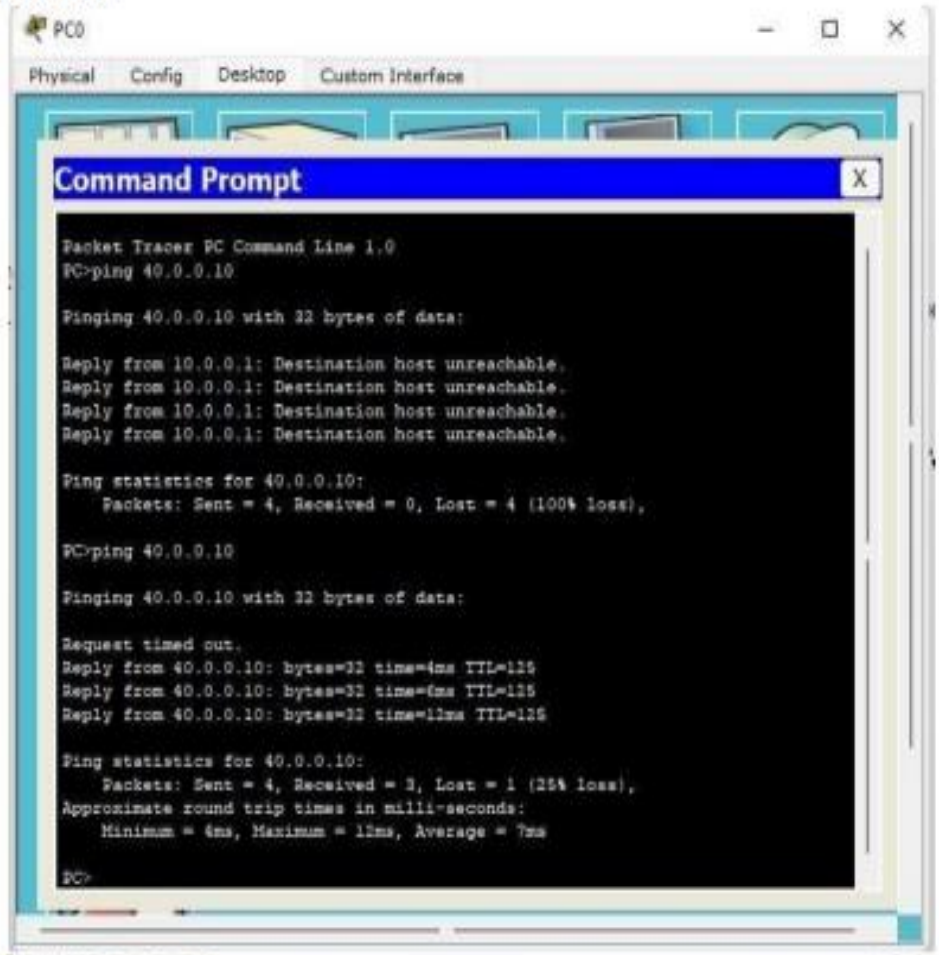
LAB PROGRAM -07

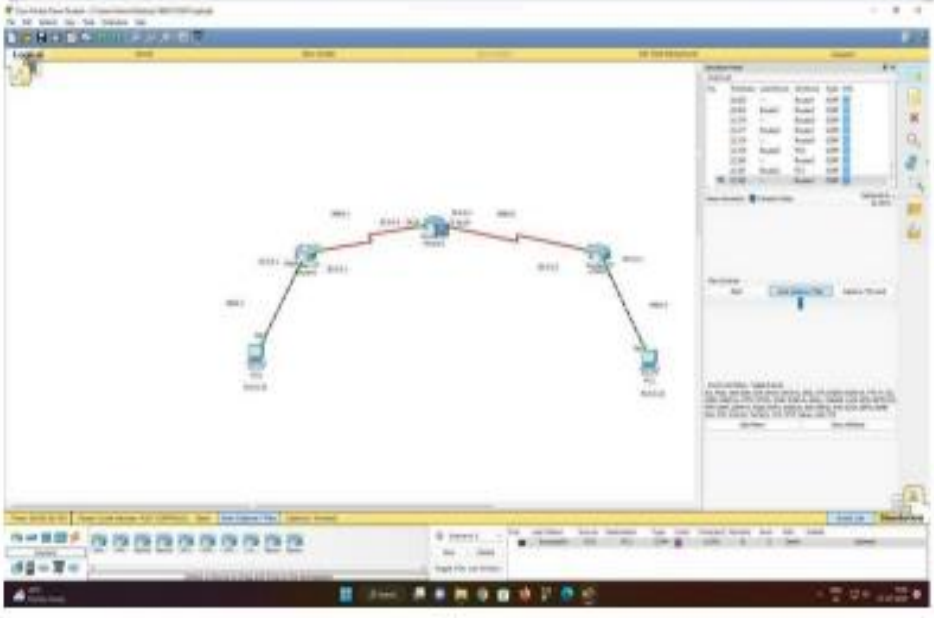
Configure OSPF routing protocol.

TOPOLOGY:



OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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>



OSPF (Open Shortest Path First)

Procedure :-

- Drag and drop 2 pc's and 3 routers
- Set the ip address of 2pc's as 192.168.1.0 and 192.168.2.0
- Set the gateway of 2pc's
- Connect them to the routers. connect the pc's and routers as shown above.

Step 1:-

Router 0:

Router(config-if)exit

Router(config)#interface serial 3/0

Router(config-if)#exit

Router(config)#router ospf 1

Router(config-router)#network 192.168.1.0. 0.0.0.255 area 0

Router(config-router)#network 10.0.0.0 0.255.255.255 area 0

Router(config-router)#network 19.0.0.0 0.255.255.255 area 0

Router R1.

R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit

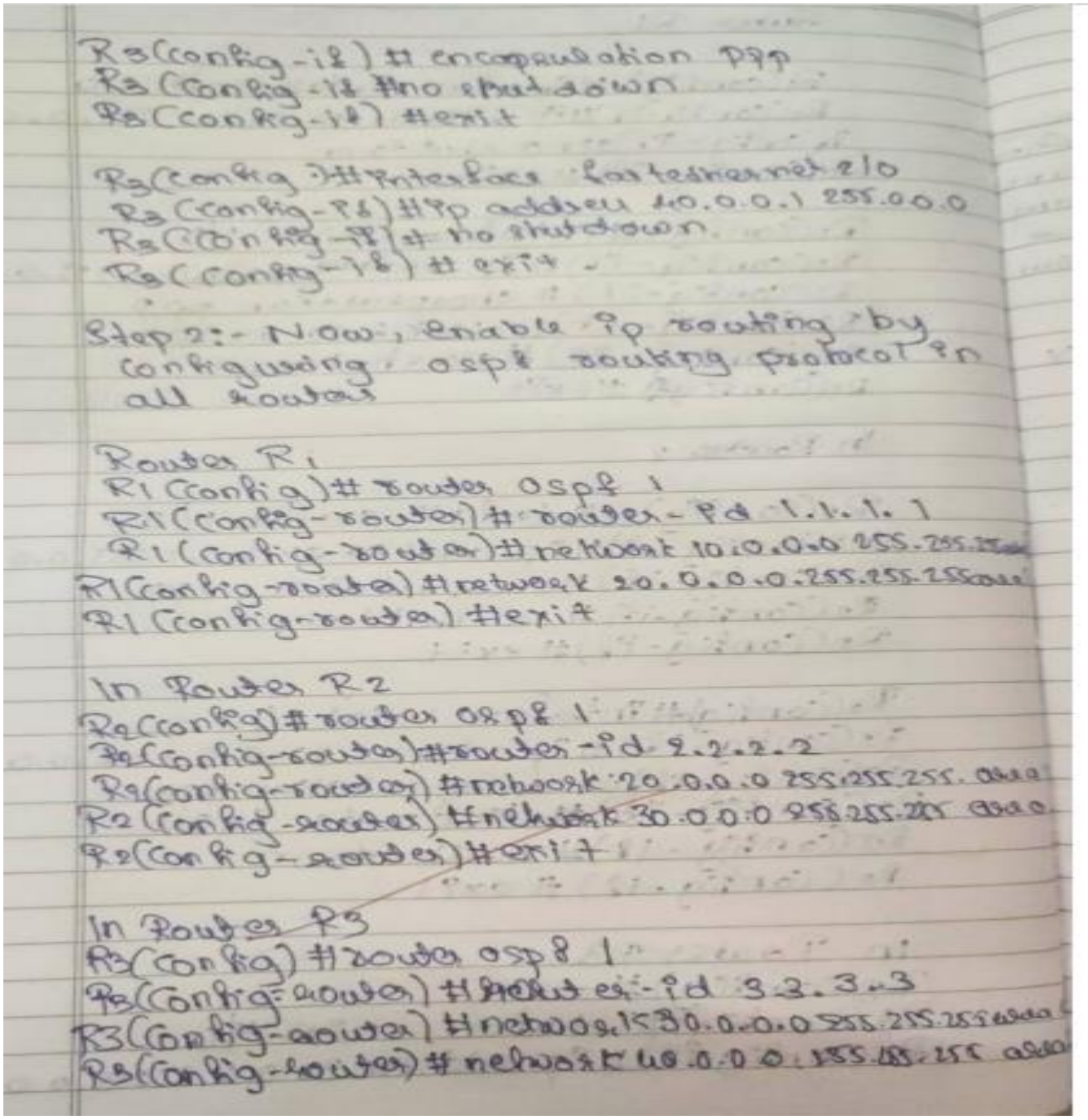
R1(config)#interface serial 1/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#encapsulation ppp
R1(config-if)#clock rate 64000
R1(config-if)#no shutdown
R1(config)#exit

In Router 2

R2(config)#interface serial 1/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#encapsulation ppp
R2(config-if)#no shutdown
R2(config-if)#exit

R2(config)#interface serial 1/1
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#encapsulation ppp
R2(config-if)#clock rate 64000
R2(config-if)#no shutdown
R2(config-if)#exit

In Router R3
R3(config)#
R3(config)#interface serial 1/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0



R3(config-if)# encapsulation ppp
R3(config-if)# no shut down
R3(config-if)# exit

R3(config)# interface fastethernet 1/0
R3(config-if)# ip address 40.0.0.1 255.0.0.0
R3(config-if)# no shutdown
R3(config-if)# exit

Step 2:- Now, enable ip routing by configuring ospf routing protocol in all routers

Router R1
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.0.0.0 255.255.255
R1(config-router)# network 20.0.0.0.255.255.255
R1(config-router)# exit

In Router R2
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 20.0.0.0 255.255.255.0
R2(config-router)# network 30.0.0.0 255.255.255.0
R2(config-router)# exit

In Router R3
R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 30.0.0.0 255.255.255
R3(config-router)# network 40.0.0.0 255.255.255

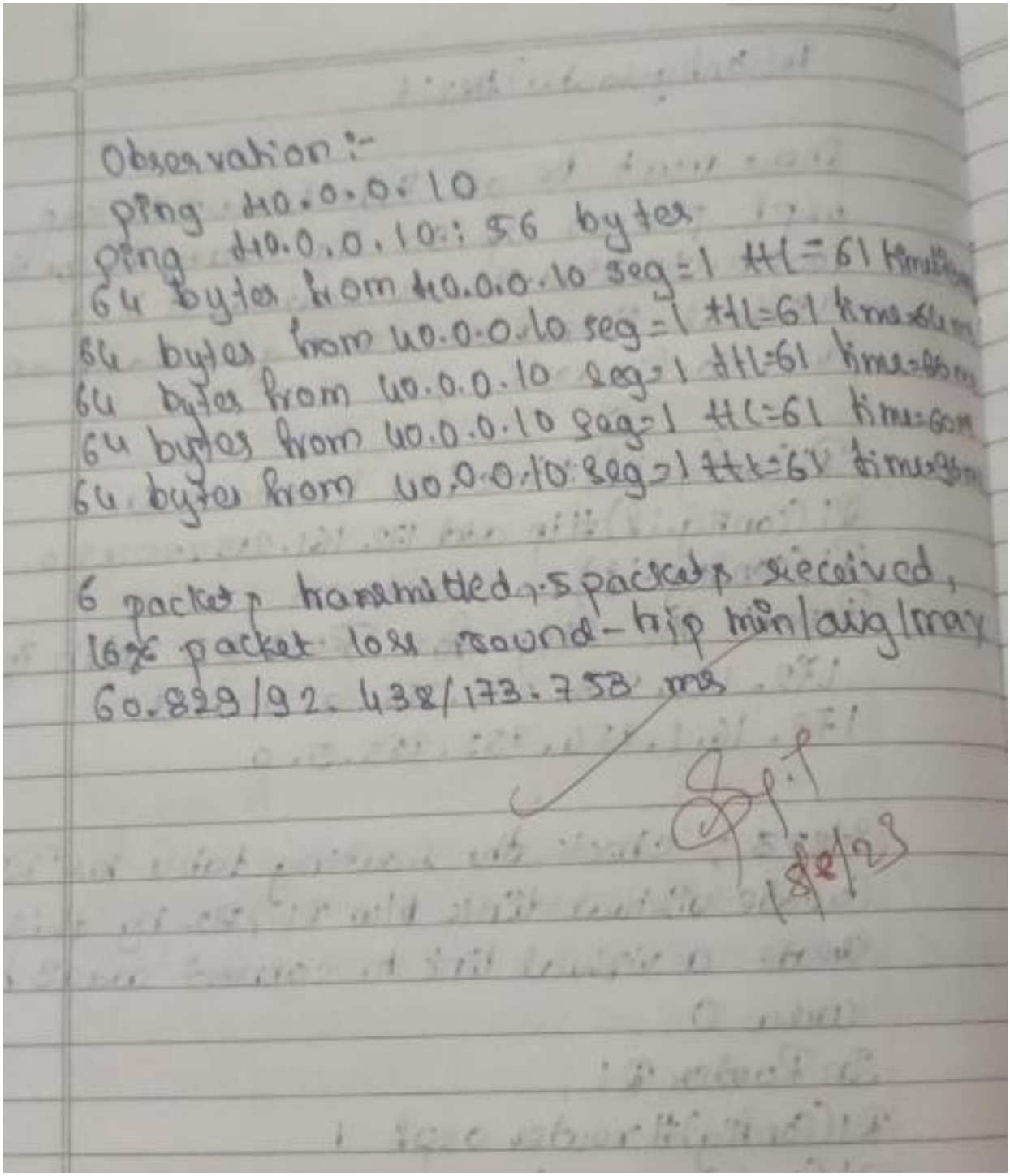
R2(config-router)#exit

There must be one interface up to keep ospf process up. So it's better to configure loopback address to routers. It is a virtual interface never goes down once we configured.

R1(config-if)#interface loopback 0
R1(config-if)#ip add 172.161.255.255.255.0.0
R1(config-if)#no shutdown
Repeat the step for R2 & R3 with ipadd 172.16.1.253.255.255.0.0 and 172.16.1.254.255.255.0.0

Step 3 :- check the routing table for R3
Create virtual link b/w R1, R2 by this we create a virtual link to connect area 3 to area 0.
In Router R1,
R1(config)#router ospf 1
R1(config-router)#area 1 virtual link 2.2.2.2
*Feb 10.10:29:29:767 %.OSPF-5-ADJCHG Process 1, Nbr 2.2.2.2 on OSPF_VL0 from LOADING to FULL, loading Done.
Repeat this step for Router 2, and Router

Now check the routing table for R3, R1 and R3 get updates about Area 3



Observation :-

ping 10.0.0.10

ping 10.0.0.10 : 56 bytes

64 bytes from 10.0.0.10 seq=1 ttl=61 time=

64 bytes from 10.0.0.10 seq=1 ttl=61 time=

64 bytes from 10.0.0.10 seq=1 ttl=61 time=

64 bytes from 10.0.0.10 seq=1 ttl=61 time=

64 bytes from 10.0.0.10 seq=1 ttl=61 time=

6 packets transmitted, 5 packets received,

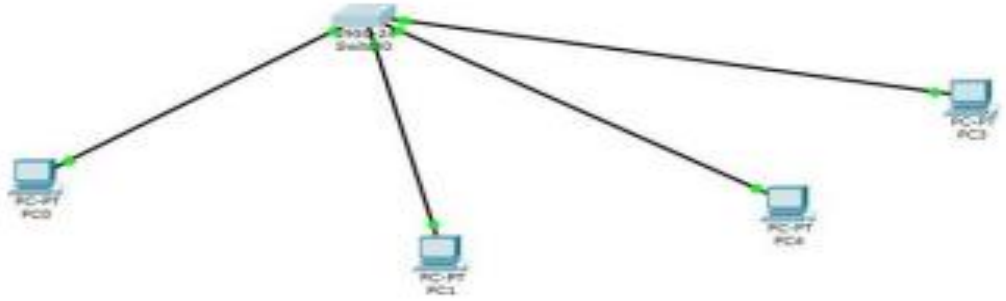
16% packet loss round-trip min/avg/max

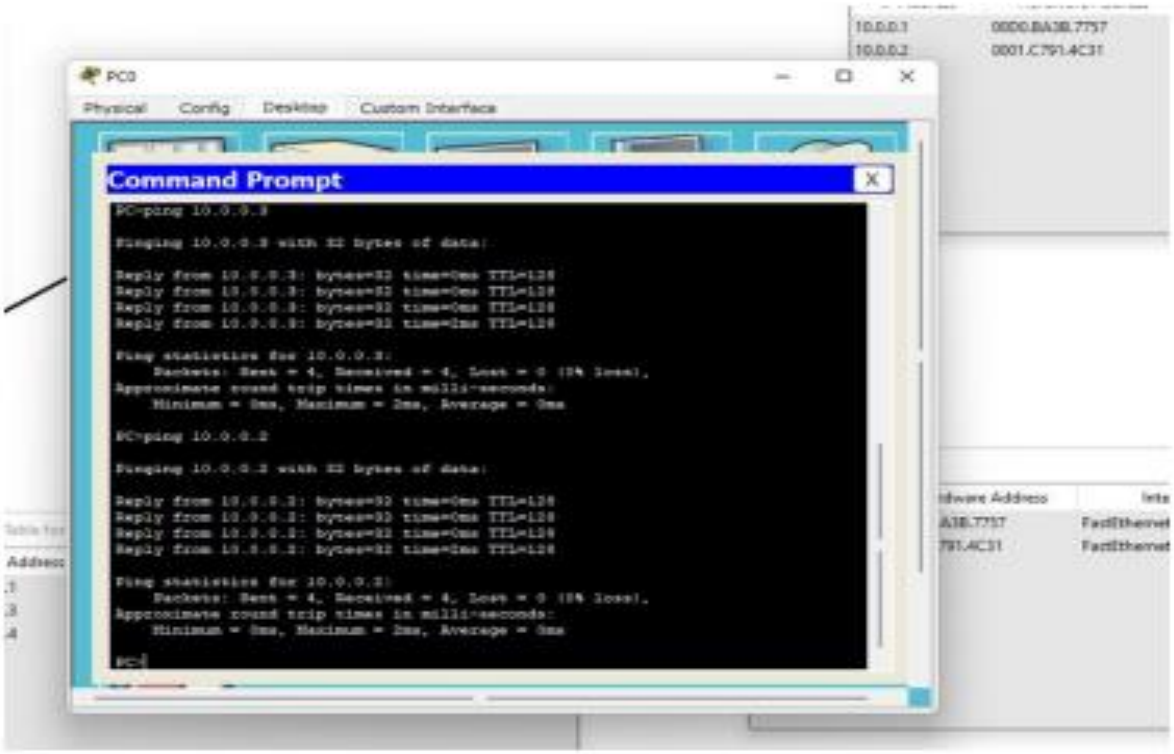
60.829/92.438/173.758 ms

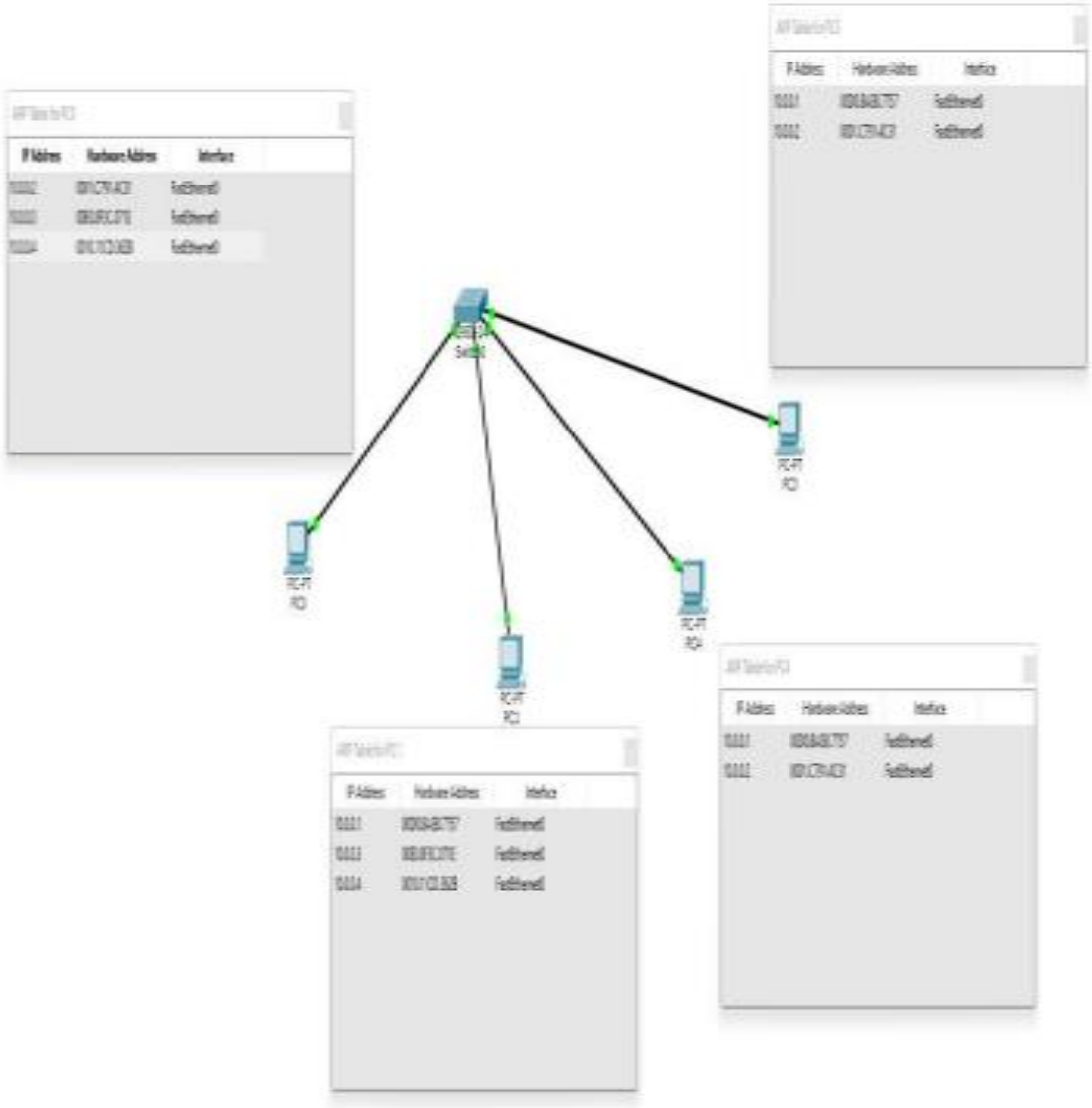
18/2/23

LAB PROGRAM-08

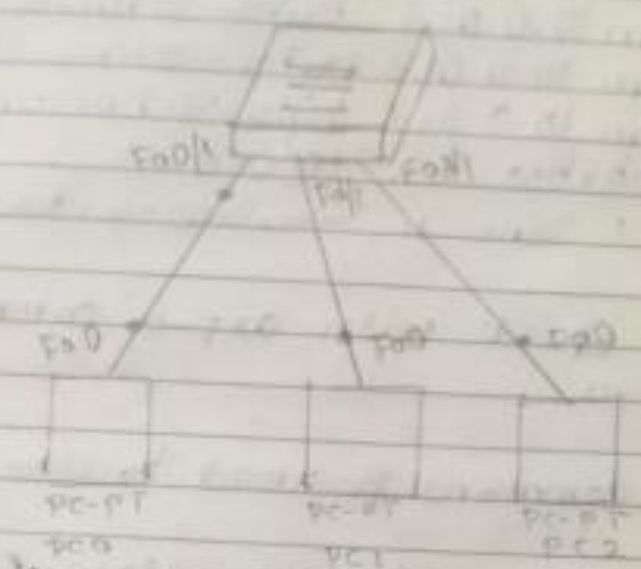
To construct simple LAN and understand the concept and operation of address resolution protocol (ARP)





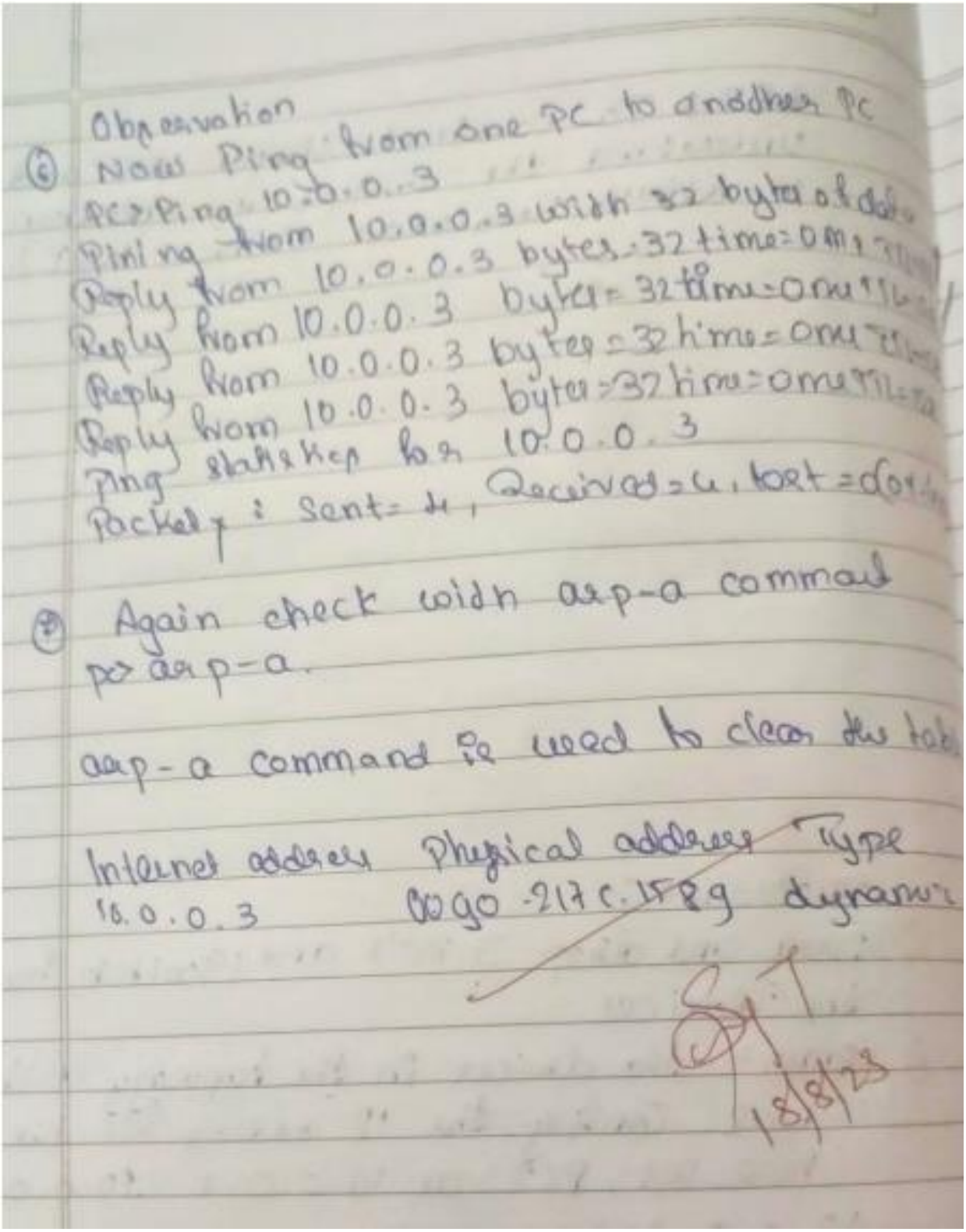


ARP :- To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)



procedure :-

1. Drag and drop 3 PC's and 1 Switch from the devices
2. Connect the devices in the topology as shown above Config the IP address for the PC's PC0, PC1, PC2 as 10.0.0.1, 10.0.0.2, 10.0.0.3 respectively.
3. Config Now, In CLI use the command "arp -a" to see ARP table, Initially the ARP table will be empty
4. Also in CLI of Switch, the command show mac address table can be given for every transaction to see how the switch learns from transactions and build the address table



Observation

⑥ Now Ping from one PC to another PC

PC> Ping 10.0.0.3

Pinging from 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 bytes=32 time=0ms

Reply from 10.0.0.3 bytes=32 time=0ms

Reply from 10.0.0.3 bytes=32 time=0ms

Reply from 10.0.0.3 bytes=32 time=0ms

Ping statistics for 10.0.0.3

Packets: Sent=4, Received=4, lost=0

⑦ Again check with arp-a command

PC> arp-a.

arp-a command is used to clear the table

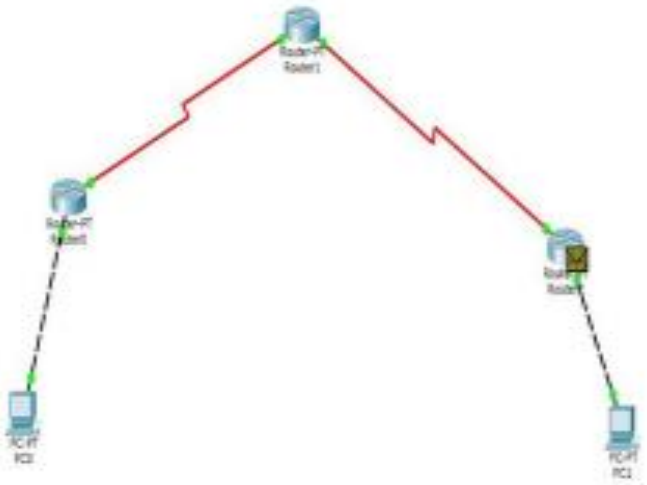
Internet address Physical address Type

10.0.0.3 0090.217C.1F8g dynamic

18/8/23

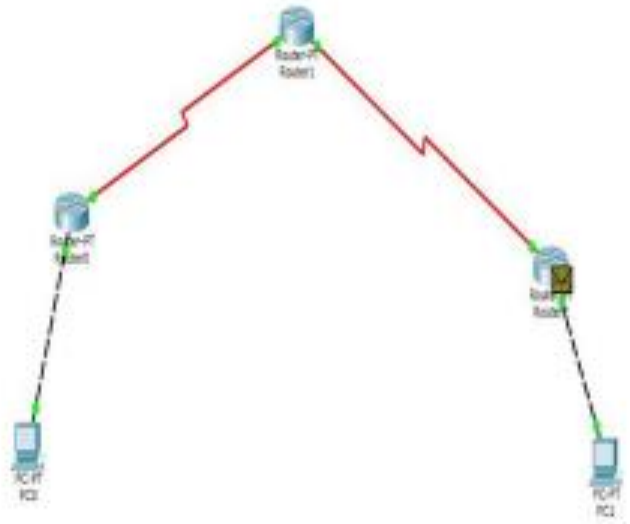
LAB PROGRAM-09

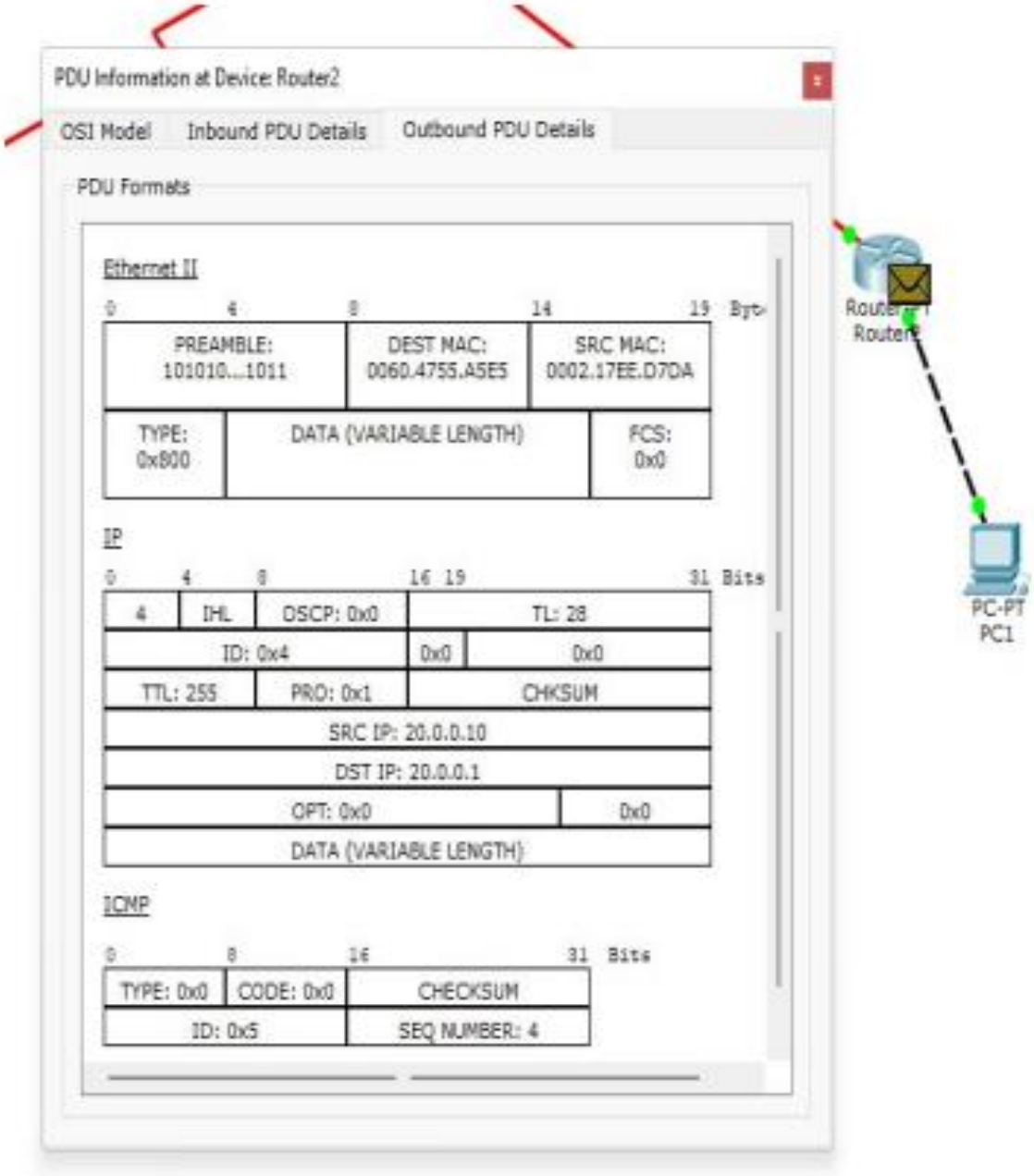
Demonstrate the TTL/Life of a Packet

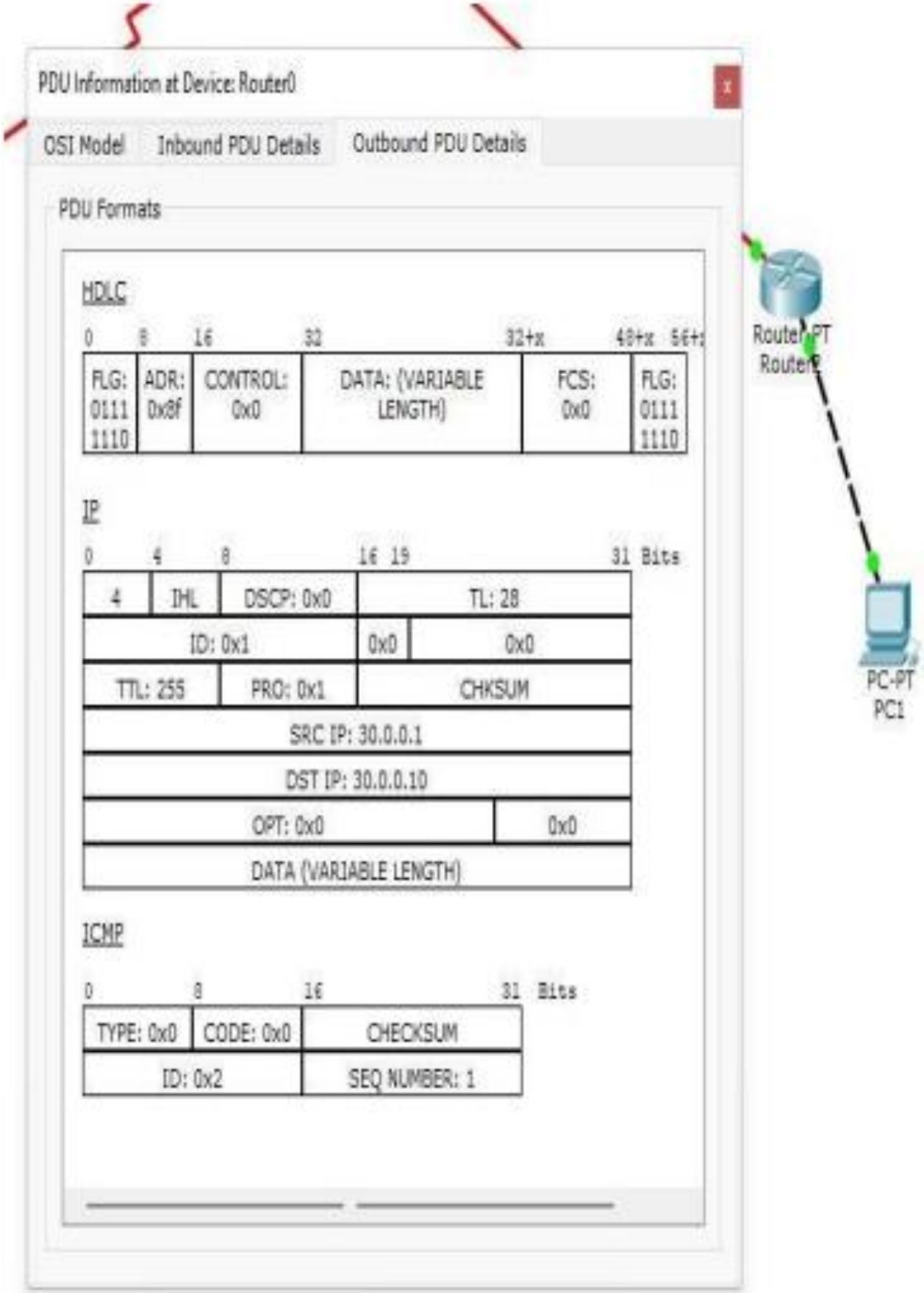


LAB PROGRAM-09

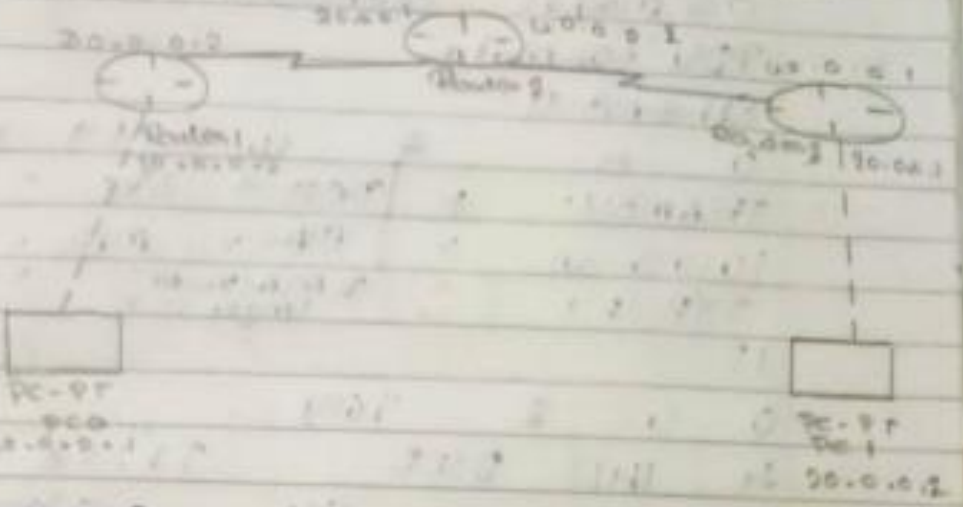
Demonstrate the TTL/Life of a Packet




PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byte
PREAMBLE: 101010...1011
DEST MAC: 0060.4755.A5E5
SRC MAC: 0002.17EE.D7DA
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
OSCP: 0x0
TL: 28
ID: 0x4
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 20.0.0.10
DST IP: 20.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 4
PC-PT
PC1

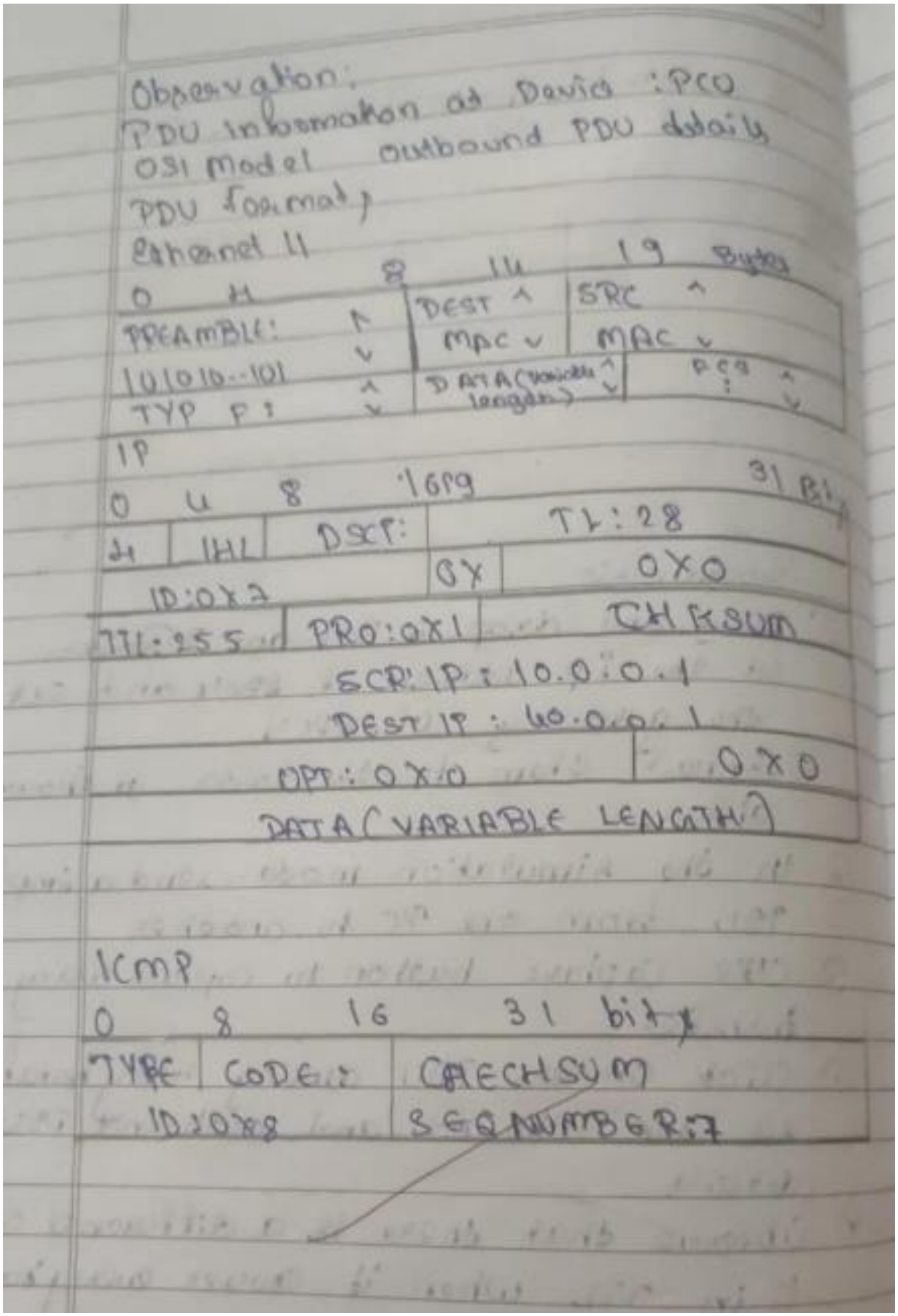

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0
8
16
32
32+x
40+x
56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0
4
8
16
19
31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x1
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 30.0.0.1
DST IP: 30.0.0.10
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0
8
16
31 Bits
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x2
SEQ NUMBER: 1
Router-PT
Router0
PC-PT
PC1

Demonstrate the TTL/life of a Packet



Procedure:-

1. Drag and drop 2 PC's and 3 Routers. Set the ip address of PC's and set the gateway of PC's. Connect them to the routers as shown above.
2. In the simulation mode - send a simple PDU from one PC to another.
3. Use Capture button to capture every transfer.
4. Click on the PDU during every hop to see the Inbound and outbound PDU details.
5. Observe that there is a difference of 1 in TTL when it crosses every router.


Observation:
PDU information at Device :PC0
OSI model outbound PDU details
PDU format
Ethernet II

0	8	16	19 Bytes
PREAMBLE: 10101010--101 TYPE:		DEST MAC	SRC MAC
		DATA (variable length)	PCS:

IP

0	4	8	16	31 Bits
4	IHL	DSF:	TL: 28	
ID: 0x7		0X	0X0	
TTL: 255		PRO: 0X1	CHKSUM	
SCR IP: 10.0.0.1				
DEST IP: 40.0.0.1				
OPT: 0X0			0X0	
DATA (VARIABLE LENGTH)				

ICMP

0	8	16	31 bits
TYPE	CODE:	CHECHSUM	
ID: 0X8		SEQ NUMBER: 7	

PDU Information at Device: Router 1

OSI Model | Inbound PDU details | Outbound PDU

PDU Formats

Ethernet II

0	4	8	14	19 Bytes
PREAMBLE: 10101010 ... 101		↑ DEST MAC ↓		SRC ↑ MAC: ↓
TYPE:		DATA (VARIABLE LENGTH)		FCS: ↑ ↓

IP

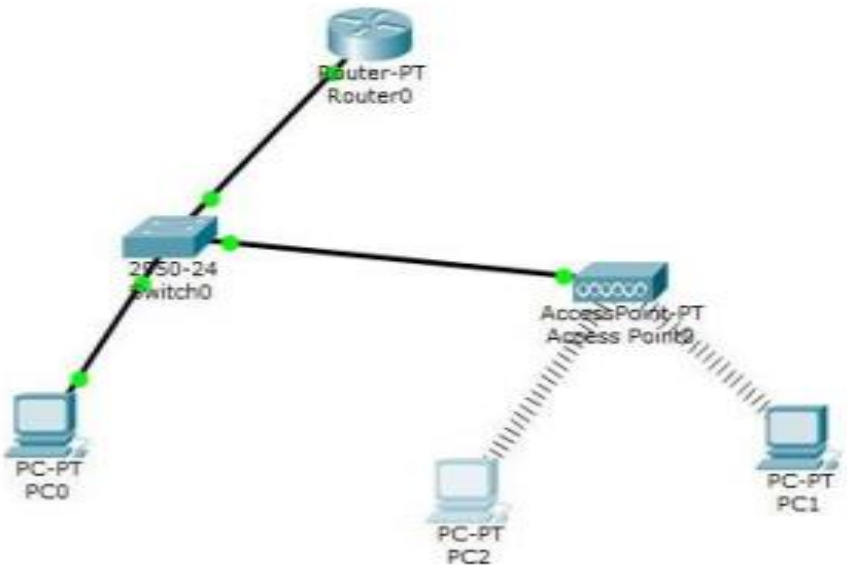
0	4	8	16	19	31 Bits
4	IHL	DSCP:	TL: 28		
ID: 0X6			0X	0X0	
TTL: 255		PRO: 0X1	CHKSUM		
SRC IP: 10.0.0.1					
DEST IP 40.0.0.1					
OPT: 0X0				0X0	
DATA (VARIABLE LENGTH)					

ICMP

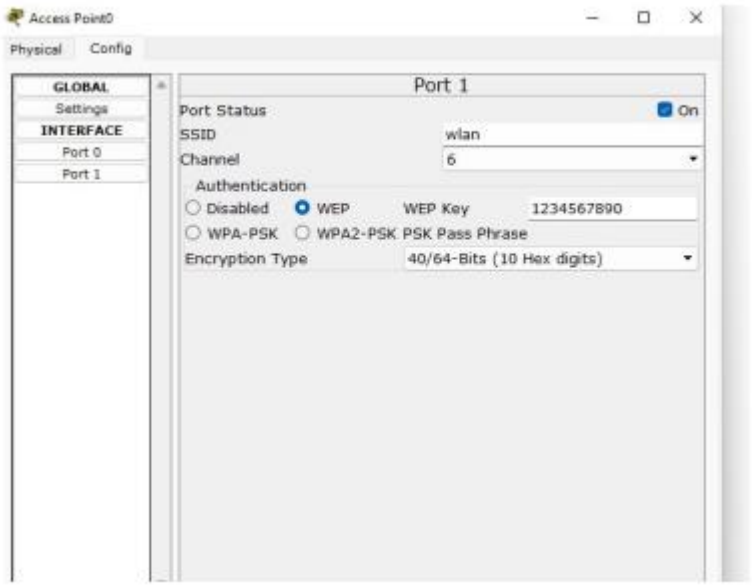
0	8	16	31 bits
TYPE:	CODE	CHECKSUM	
ID: 0X7		SEQ NUMBER: 6	

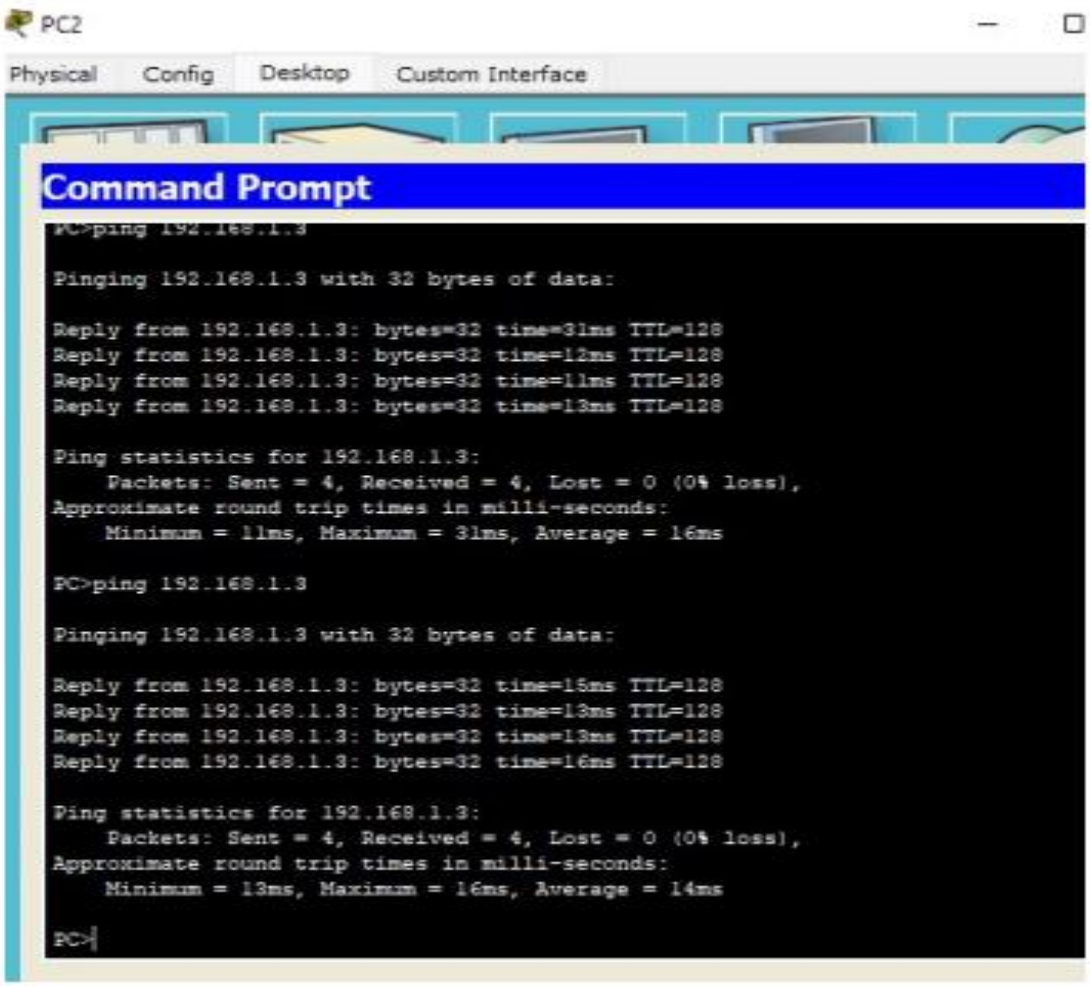
LAB PROGRAM:10

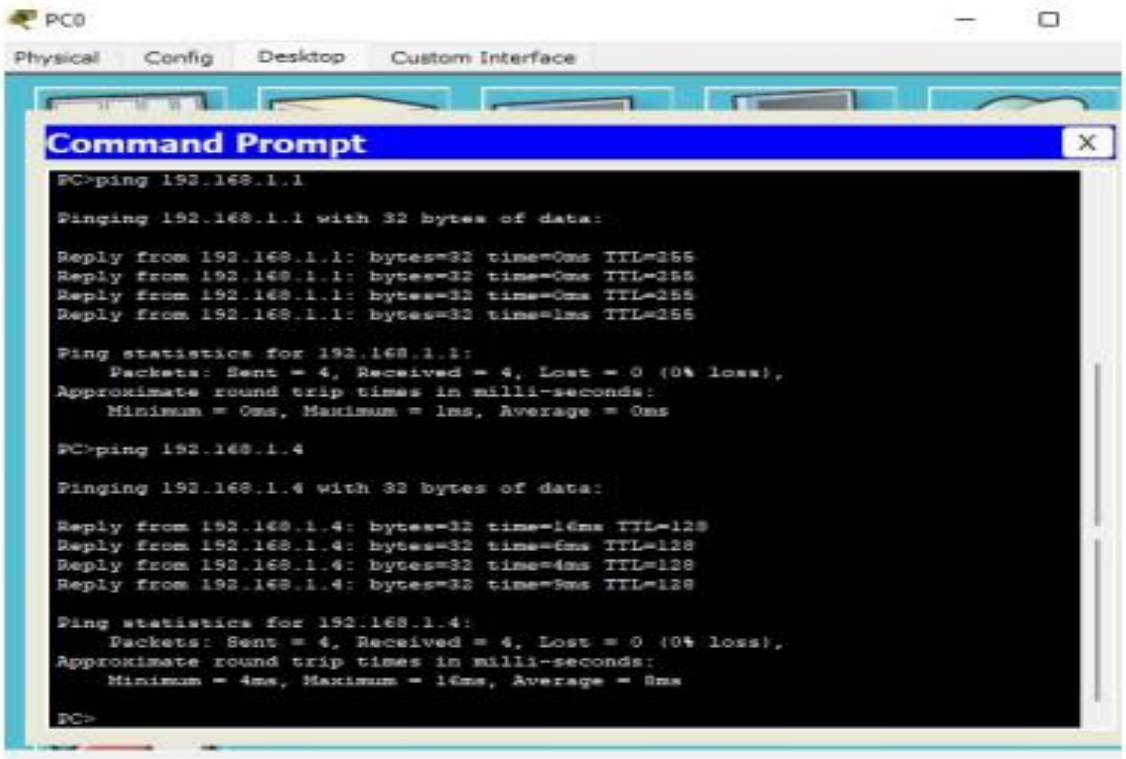
WLAN:



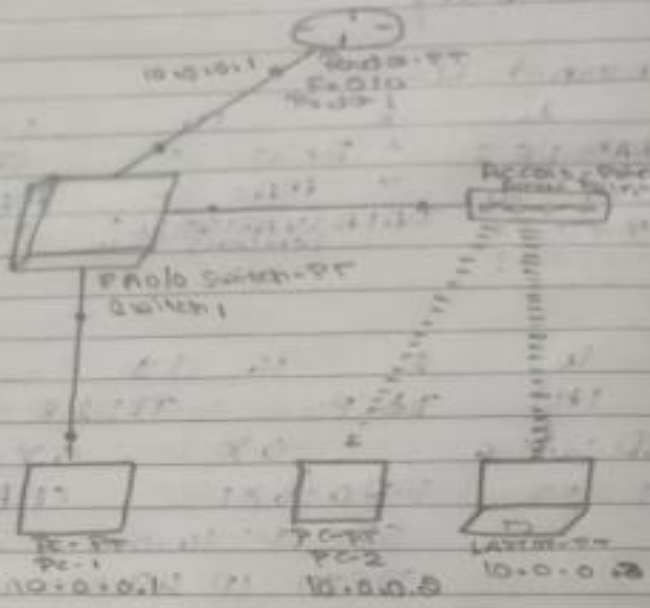

PC2
Physical
Config
Desktop
Custom Interface
Link Information
Connect
Profiles
Below is a list of available wireless networks. To search for more wireless networks, click the Refresh button. To view more information about a network, select the wireless network name. To connect to that network, click the Connect button below.
Wireless Network Name
CH
Signal
wlan
1
72%
Site Information
Wireless Mode Infrastructure
Network Type Mixed B/G
Radio Band Auto
Security WEP
MAC Address 0004.9A8N.D9W
Refresh
Connect
2.4GHz
Adapter is Active
Wireless-N Notebook Adapter
Wireless Network Monitor v1.0
WPC300N
PC2
Physical
Config
Desktop
Custom Interface
Link Information
Connect
Profiles
More Information
Infrastructure Mode
You have successfully connected to the access point
2.4GHz
Signal Strength
Link Quality
Adapter is Active
Wireless-N Notebook Adapter
Wireless Network Monitor v1.0
WPC300N


Access Point0
Physical
Config
GLOBAL
Settings
INTERFACE
Port 0
Port 1
Port 1
Port Status
On
SSID
wlan
Channel
6
Authentication
Disabled
WEP
WEP Key
1234567890
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Encryption Type
40/64-Bits (10 Hex digits)


PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=31ms TTL=128
Reply from 192.168.1.3: bytes=32 time=12ms TTL=128
Reply from 192.168.1.3: bytes=32 time=11ms TTL=128
Reply from 192.168.1.3: bytes=32 time=13ms TTL=128
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 11ms, Maximum = 31ms, Average = 16ms
PC>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=15ms TTL=128
Reply from 192.168.1.3: bytes=32 time=13ms TTL=128
Reply from 192.168.1.3: bytes=32 time=13ms TTL=128
Reply from 192.168.1.3: bytes=32 time=16ms TTL=128
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 13ms, Maximum = 16ms, Average = 14ms
PC>


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>ping 192.168.1.4
Pinging 192.168.1.4 with 32 bytes of data:
Reply from 192.168.1.4: bytes=32 time=16ms TTL=128
Reply from 192.168.1.4: bytes=32 time=6ms TTL=128
Reply from 192.168.1.4: bytes=32 time=4ms TTL=128
Reply from 192.168.1.4: bytes=32 time=9ms TTL=128
Ping statistics for 192.168.1.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 16ms, Average = 8ms
PC>

To construct a WLAN and make the nodes communicate wirelessly



Procedure:

* Drag and Drop 2 PC's, 1 Laptop, one Switch and 1 Power, and Connect them as shown above
* Configure PC1 and Power (as PC normally done
* Configure Access Point-Port1 -> SSID Name- any name (WLAN here)
* Select WEP and give any 10 digit hex key (as 1234567890)
* Configure PC2 and Laptop with Wireless standards
* Switch off the device, Drag the existing PT-HOST-NM-1AM to the Component listed

in the Link Drag component wireless interface to the empty part. Switch on the device

* In the config tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and Gateway to the device

Observation

PC> Ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2 bytes=32 time=0ms TTL=255

Reply from 10.0.0.2 bytes=32 time=0ms TTL=255

Reply from 10.0.0.2 bytes=32 time=0ms TTL=255

Reply from 10.0.0.2 bytes=32 time=0ms TTL=255

Ping Statistics for 192.168.1.1:

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

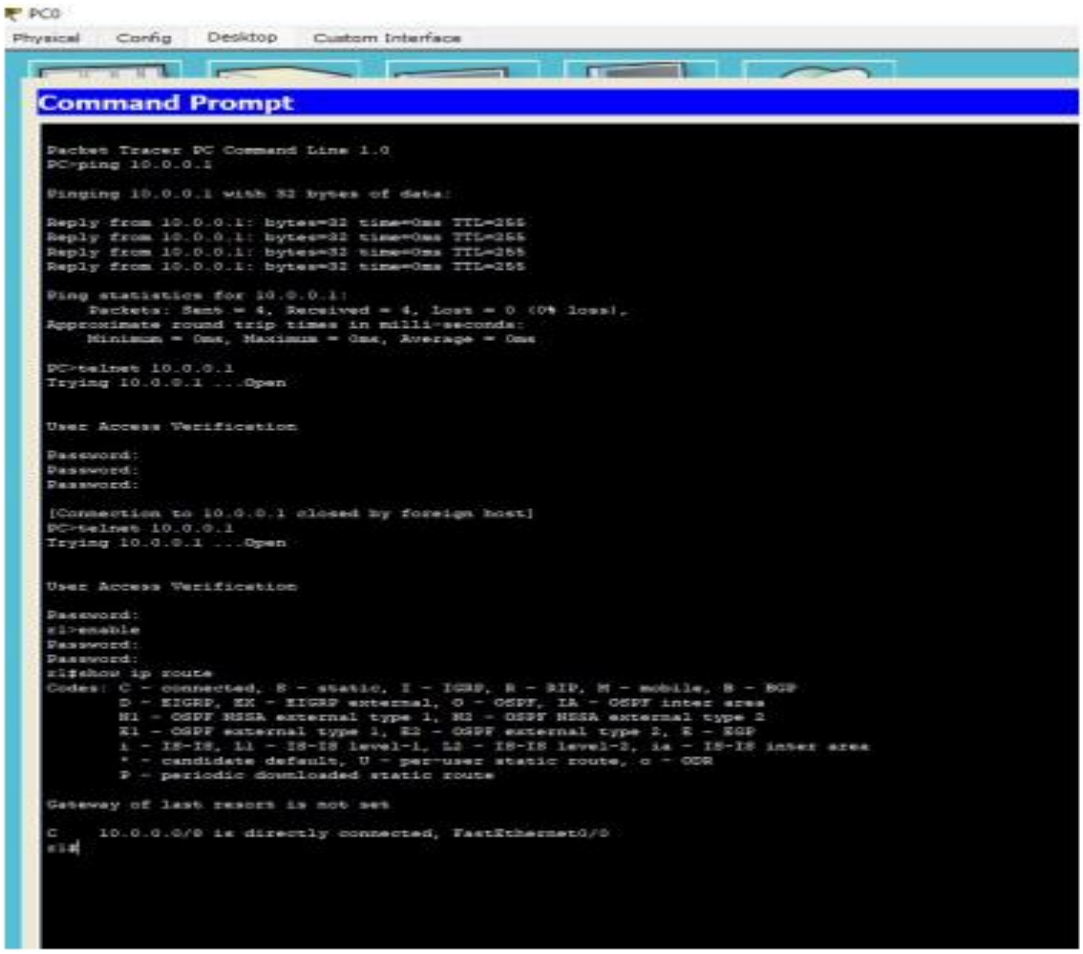
Approximate round trip in milliseconds

Minimum = 0ms, Maximum = 1ms, Average=0ms

LAB PROGRAM-11

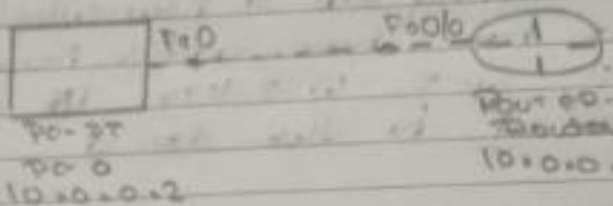
TELNET:

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface termial FA0/0
                              ^
% Invalid input detected at '^' marker.

Router(config)#interface termial fa0/0
                              ^
% Invalid input detected at '^' marker.

Router(config)#interface fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#hostname r1
r1(config)#enable secret p1
r1(config)#interface fastethernet 0/0
r1(config-if)#ip address 10.0.0.1 255.0.0.0
r1(config-if)#no shutdown
r1(config-if)#
r1(config-if)#line vty 0 5
r1(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
% Login disabled on line 136, until 'password' is set
% Login disabled on line 137, until 'password' is set
r1(config-line)#password p0
r1(config-line)#
r1(config-line)#exit
r1(config)#exit
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#wr
Building configuration...
[OK]
r1#
```


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:
Password:
Password:
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
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#

Telnet:

To understand the operation of TELNET by accessing the router



Procedure:-

* Drag and drop 1 PC and 1 Router. Set the IP address and connect them as shown.
* Click on router → CLI
  - > enable
  - > config t
  - > hostname R1
  - > enable secret R1
  - > interface fastethernet 0/0
  - > ip address 10.0.0.1 255.0.0.0
  - > no shutdown
  - > line vty 0 5 [To allow virtual terminal access for 6 users]
  - > login
  - > password p0
  - > exit
  - > exit
  - > wr - [To save changes in router]

Observation.

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)

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
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
* - candidate default, U - per-user static route
P - periodic downloaded static route

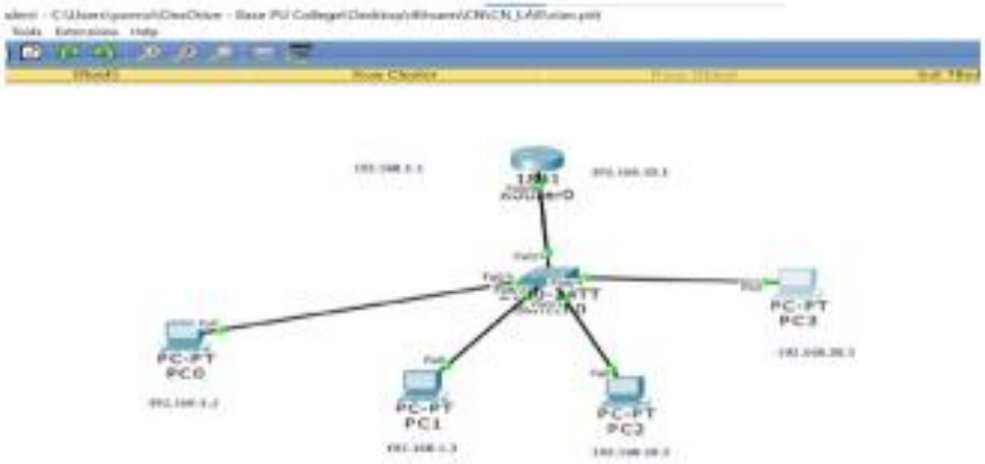
Gateway of last resort is not set

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

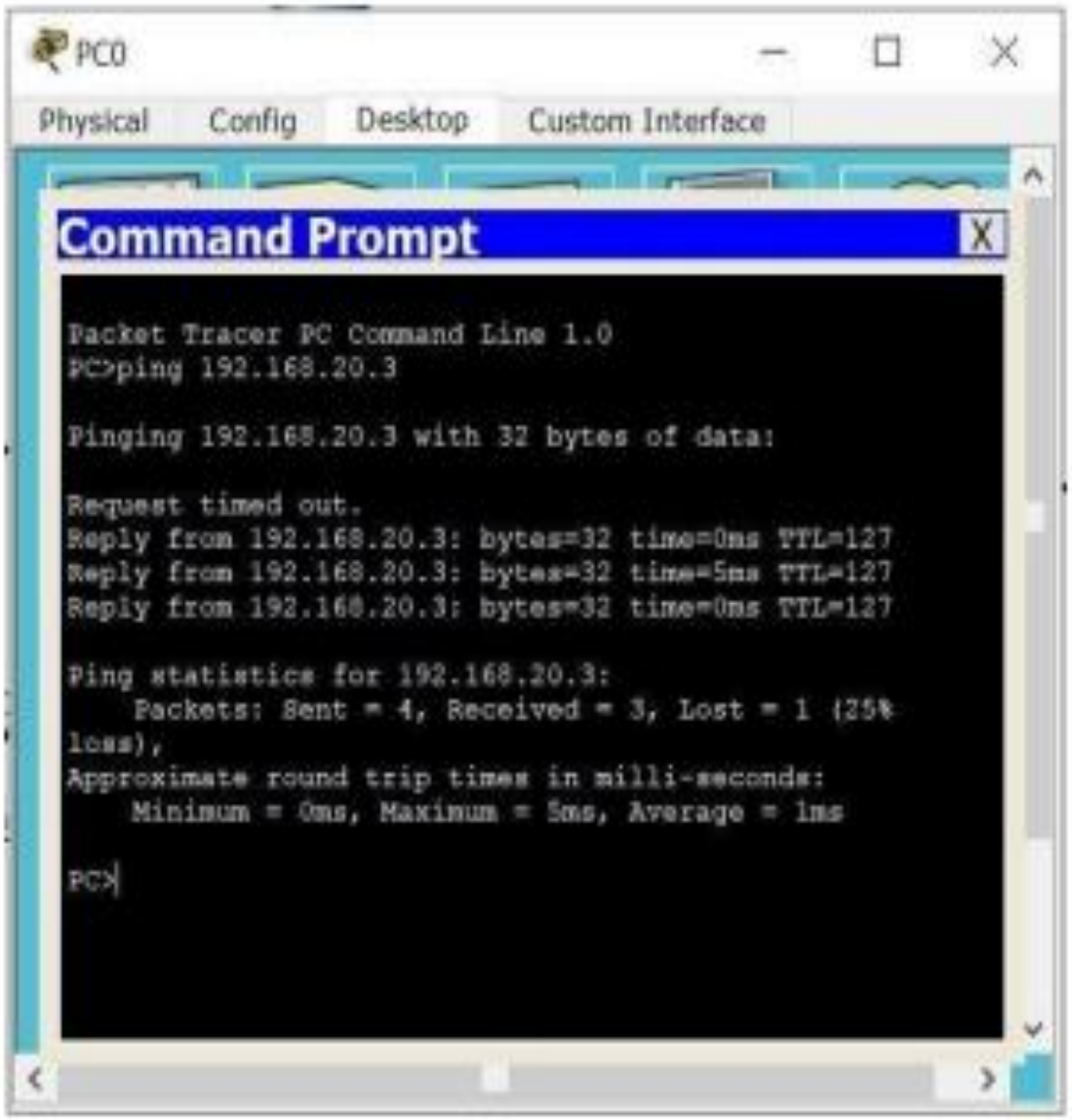
LAB PROGRAM-12

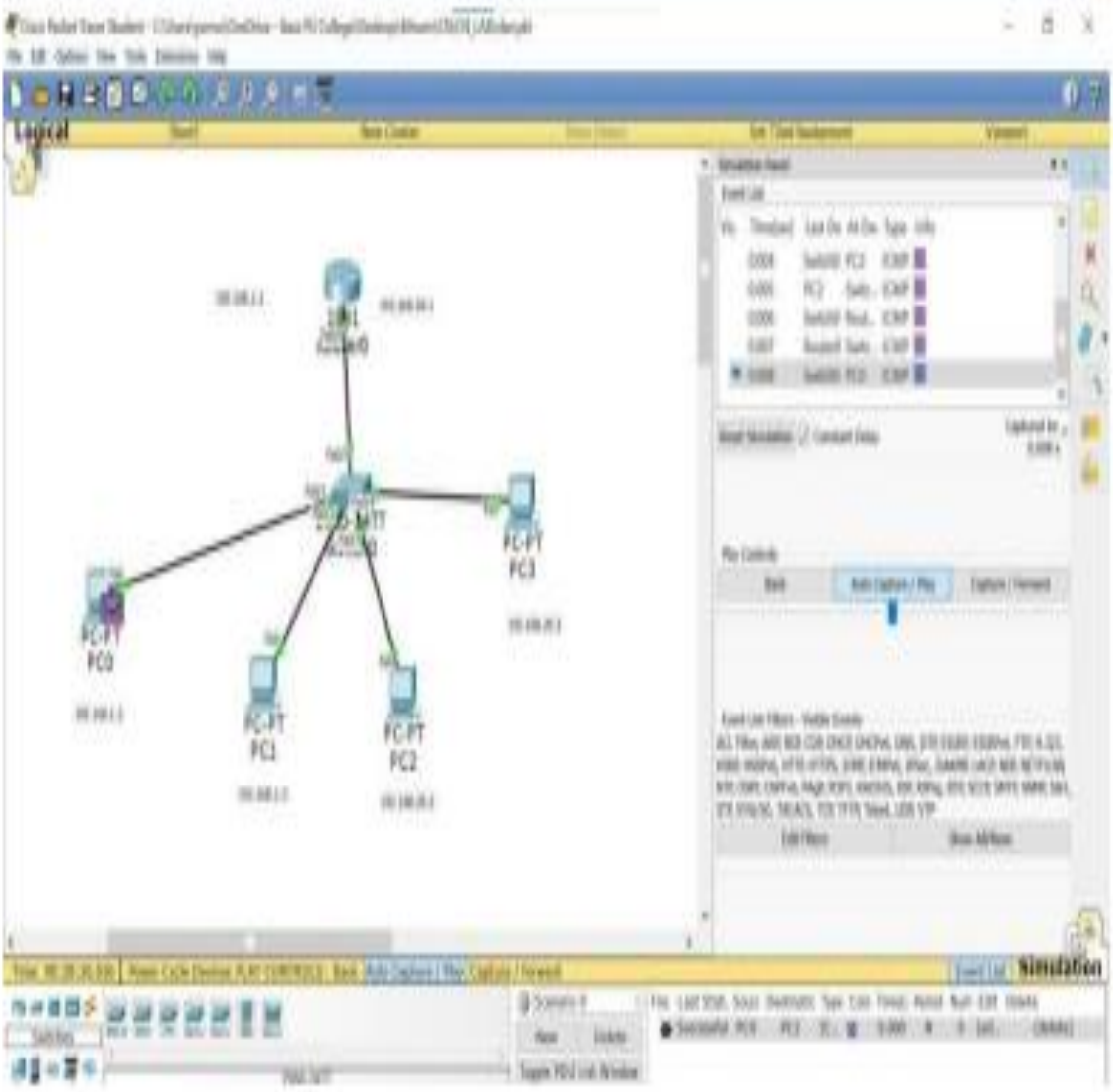
To construct a VLAN and make a pc communicate among VLAN.

TOPOLOGY:

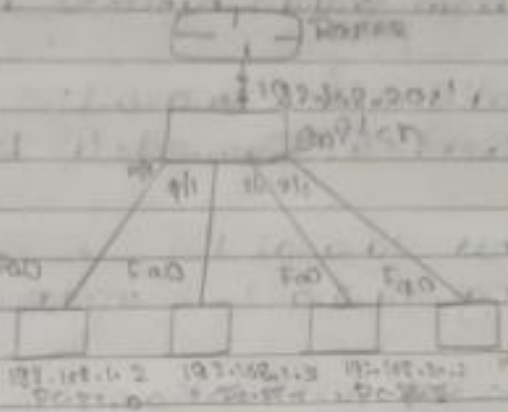


OUTPUT:


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

Virtual LAN :- To construct a VLAN & make the PC's communicate among VLAN.



Procedure :-

* To construct a new VLAN we use cisco type address
* Create a topology as shown above. Choose the 1841 router.
* First PC0 & PC1 will be in physical LAN and PC2 and PC3 will be in VLAN
* Configure router i.e set ip address for the Fa 0/0 Interface as 192.168.1.1
* And set ip address of PC0 & PC1 as 192.168.1.2 and 192.168.1.3 and gateway as 192.168.1.1
* Now we can check that PC0 and PC1 can communicate with each other
* For PC2 and PC3 set IP address as 192.168.20.2 and 192.168.20.3 and gateway as 192.168.20.1

Switch configuration:

* In switch go to config and select VLAN database and VLAN no and name:-
  Ex: VLAN Number 20
  VLAN Name NewVLAN
* Click on add ⇒ select the interface i.e fa 0/1 (mean the switch from router) & make it trunk.
* VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk
* 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
* And make (select) the interface that are connecting VLAN PC's to the switch
* Here it is fa 0/1 & fa 0/1 and select and make VLAN as 20: new VLAN

Router Configuration:-

* Open config select VLAN database enter the no and name of VLAN created go to CLI

Router(VLAN)#exit

Apply ~~completed~~

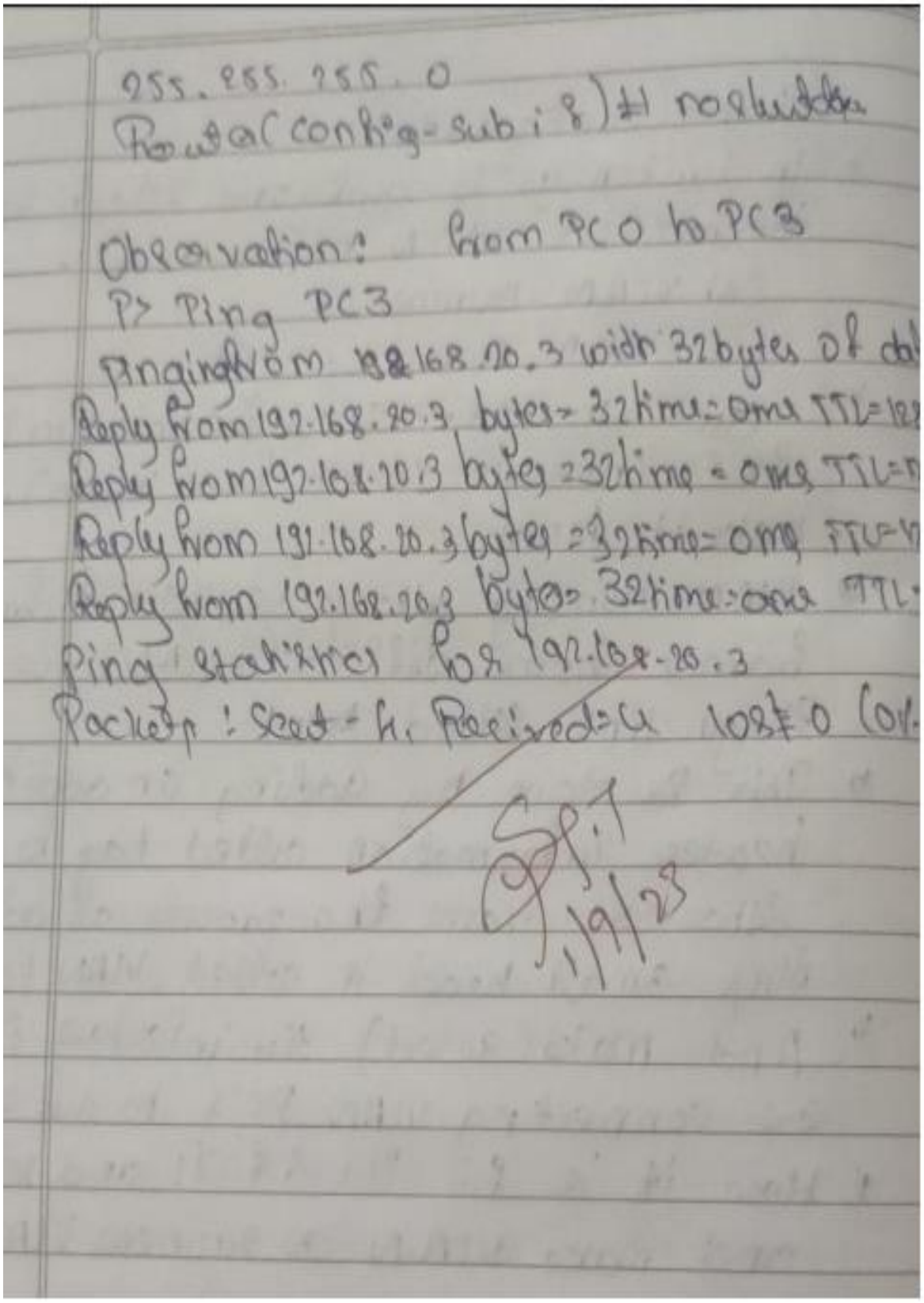
existing

Router #config t

> interface fa 0/0

> encapsulation dot1q 2

> ip address 192.168.20.1


255.255.255.0
Router(config-sub i 8) #1 no shut
Observation: from PC0 to PC3
P> Ping PC3
Pinging from 192.168.20.3 with 32 bytes of da
Reply from 192.168.20.3 bytes=32 time=0ms TTL=12
Reply from 192.168.20.3 bytes=32 time=0ms TTL=1
Reply from 192.168.20.3 bytes=32 time=0ms TTL=1
Reply from 192.168.20.3 bytes=32 time=0ms TTL=
Ping statistics for 192.168.20.3
Packets: Sent=4, Received=4 lost=0 (0%
Sp.1
9/9/23

LAB PROGRAM-13

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

CODE:

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

#define N strlen(gen_poly) char data[28]; char check_value[28]; char
gen_poly[10]; int data_length,i,j; void XOR(){ for(j = 1;j < N; j++)
check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');} void receiver(){
printf("Enter the received data: "); scanf("%s", data); printf("Data received:
%s", data); crc(); for(i=0;(i<N-1) && (check_value[i]!='1');i++); if(i<N-1)
printf("\nError detected\n\n");
else printf("\nNo error
detected\n\n");
} void crc(){
for(i=0;i<N;i++)
check_value[i]=data[i];
do{ if(check_value[0]=='1')
XOR(); for(j=0;j<N-1;j++)
check_value[j]=check_value[j+1];
check_value[j]=data[i++];
}while(i<=data_length+N-1);
} int

main()
```

```
{ printf("\nEnter data to be transmitted: ");
scanf("%s",data); printf("\n Enter the
Generating polynomial: ");
scanf("%s",gen_poly); data_length=strlen(data);
for(i=data_length;i<data_length+N-1;i++)
data[i]='0';
printf("\n Data padded with n-1 zeros : %s",data); crc();
printf("\nCRC or Check value is :
%s",check_value);
for(i=data_length;i<data_length+N-1;i++)
 data[i]=check_value[i-data_length]; printf("\n Final
data to be sent : %s",data); receiver();
return 0;
}
```

OUTPUT:

```
Enter data to be transmitted: 1000100000100001

 Enter the Generating polynomial: 1011

 Data padded with n-1 zeros : 10001000001000001000
CRC or Check value is : 100
 Final data to be sent : 10001000001000001100
Enter the received data: 10001000001000001100
Data received: 10001000001000001100
No error detected
```

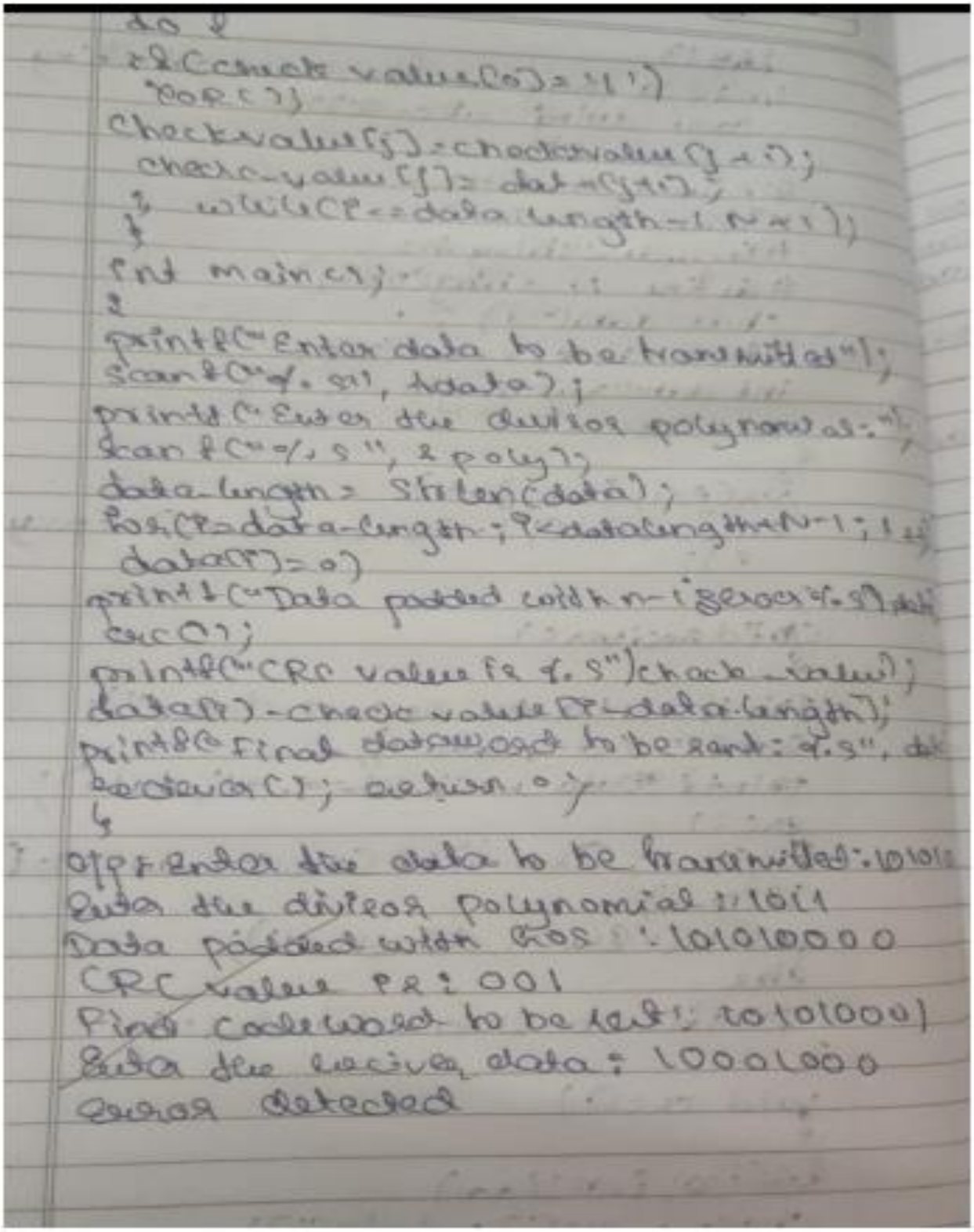
```
 Enter data to be transmitted: 1000100000100001

  Enter the Generating polynomial: 1011

  Data padded with n-1 zeros : 10001000001000001000
 CRC or Check value is : 100
  Final data to be sent : 10001000001000001100
 Enter the received data: 10010000001000001100
 Data received: 10010000001000001100
 Error detected
```

LAB 13

Write a program for error detecting code using CRC-CCITT

program

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define N strlen(poly)
char data[30];
char check_value[30];
int data[30],length, i, j;
void xor()
{
for(j=1; j<N; j++)
check_value[j] = ((check_value[j]==poly[j])?'0':'1');
}
void receiver()
{
printf("Enter the received data:");
scanf("%s", data);
printf("Data received = %s", data);
crc();
for(i=0; (i<N-1)&&(check_value[i]=='0'); i++);
if(N-1<i)
printf("\nError data detected\n");
else
printf("\nNo error data detected\n");
}
void crc()
{
for(i=0; i<N; i++)
check_value[i] = data[i];
```



```
do {
if(check value[0]=='1')
for(j=1;j<N;j++)
check value[j]=check value[j+1];
check value[j]=data[i++];
} while(i<=data length+N-1);
}
int main()
{
printf("Enter data to be transmitted");
scanf("%s",data);
printf("Enter the divisor polynomial:");
scanf("%s",&poly);
data length = Strlen(data);
for(i=data length;i<data length+N-1;i++)
data[i]='0';
printf("Data padded with n-1 zeros:%s\n",
data);
printf("CRC value is %s",check value);
data[i]=check value[i-data length];
printf("Final data word to be sent:%s",data);
receiver();return 0;
}
O/P: Enter the data to be transmitted:10101
Enter the divisor polynomial:1011
Data padded with 3 0's:10101000
CRC value is 001
Final codeword to be sent:10101001
Enter the received data:10001000
Error detected
```

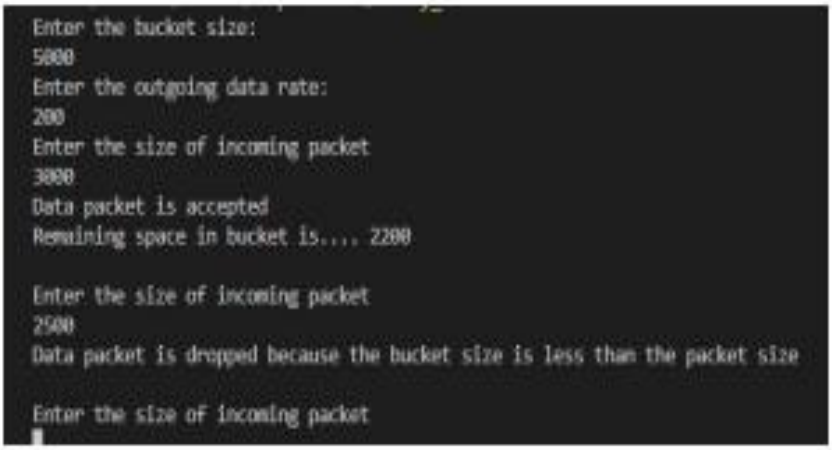
LAB PROGRAM-14

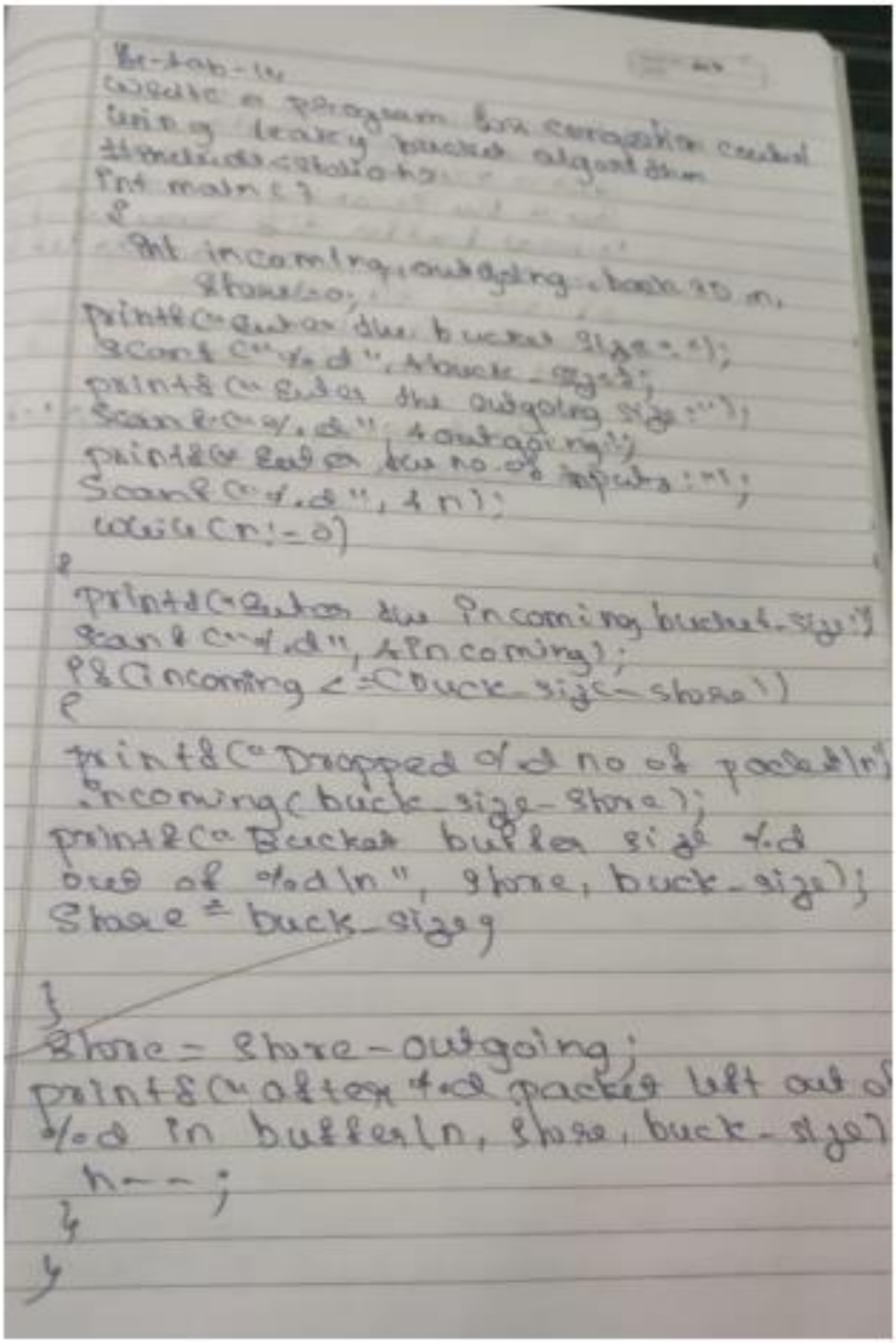
Write a program for congestion control using Leaky bucket algorithm.

CODE:

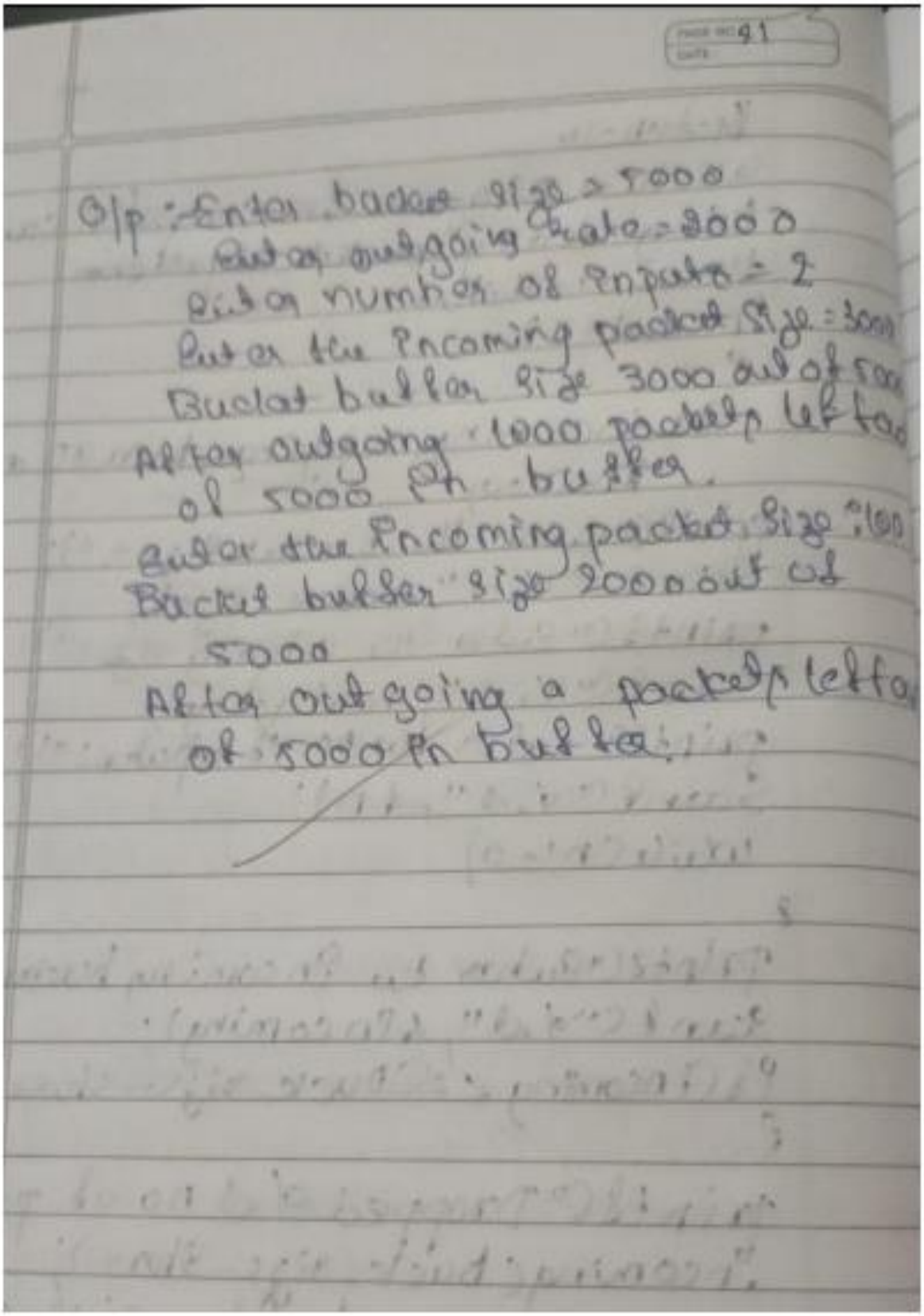
```
#include<stdio.h> void
main()
{ int b_size,d_rate,in_d_rate,rem_b_size;
printf("Enter the bucket size:\n");
scanf("%d",&b_size); rem_b_size=b_size;
printf("Enter the outgoing data rate:\n");
scanf("%d",&d_rate); while(1) {
printf("Enter the size of incoming packet\n");
scanf("%d",&in_d_rate); if(in_d_rate<=b_size)
{ if(in_d_rate<=rem_b_size) { rem_b_size=rem_b_size-
in_d_rate; rem_b_size=rem_b_size+d_rate;
printf("Data packet is accepted\n"); printf("Remaining
space in bucket is....
%d\n",rem_b_size); printf("\n");
} else{ printf("Data packet is dropped because the bucket size is less than
the packet
size\n"); printf("\n");
}
}
}
}
```

OUTPUT:


Enter the bucket size:
5000
Enter the outgoing data rate:
200
Enter the size of incoming packet
3000
Data packet is accepted
Remaining space in bucket is.... 2200
Enter the size of incoming packet
2500
Data packet is dropped because the bucket size is less than the packet size
Enter the size of incoming packet



Ex-tab-14
Write a program for congestion control using leaky bucket algorithm
#include<stdio.h>
int main()
{
int incoming, outgoing, buck_size, n,
store=0;
printf("Enter the bucket size:");
scanf("%d", &buck_size);
printf("Enter the outgoing size:");
scanf("%d", &outgoing);
printf("Enter the no. of inputs:");
scanf("%d", &n);
while(n!=0)
{
printf("Enter the incoming bucket size");
scanf("%d", &incoming);
if(incoming <= (buck_size-store))
{
printf("Dropped %d no of packets\n",
incoming(buck_size-store));
printf("Bucket buffer size %d
occupied out of %d\n", store, buck_size);
store = buck_size;
}
store = store-outgoing;
printf("After %d packets left out of
%d in buffer\n", store, buck_size);
n--;
}
}


91
O/p : Enter bucket size : 5000
Enter outgoing rate : 2000
Enter number of inputs : 2
Enter the incoming packet size : 300
Bucket buffer size 3000 out of 5000
After outgoing 1000 packets left from
of 5000 in buffer.
Enter the incoming packet size : 100
Bucket buffer size 2000 out of
5000
After out going a packets left fro
of 5000 in buffer.

LAB PROGRAM-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.

SOLUTION:

ClientTCP.py

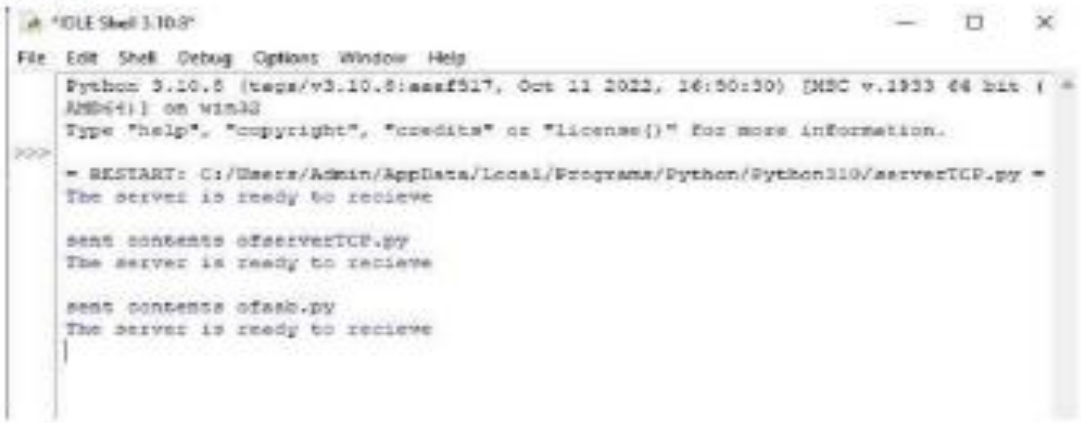
```
from socket import * serverName =
'127.0.0.1' serverPort = 12000 clientSocket =
socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode()) filecontents
= clientSocket.recv(1024).decode() print
("\nFrom Server:\n") print(filecontents)
clientSocket.close()
```

ServerTCP.py

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

connectionSocket.send(l.encode())
print ("\nSent contents of " + sentence)
```

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter file name:aab.py

From Server:

Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
class Node:
    def __init__(self,data):
        self.data=data
        self.left=None
        self.right=None
        self.height=1

class AVL Tree:
    def getHeight(self,root):
        if not root:
            return 0
        return root.height

    def getBalance(self,root):
        if not root:
            return 0
        return self.getHeight(root.left)-self.getHeight(root.right)

    def rightRotate(self,z):
        y=z.left
        T3=y.right

        y.right=z
        z.left=T3

        z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
        y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

        return y

    def insert(self,root,data):
        if not root:
            return Node(data)
        if data < root.data
            root.left=self.insert(root.left,data)
        else:
            root.right=se
>>>
```

Server:



LAB - 15

Using TCP/IP sockets, write a client server program to make client sending the file name of the server to read back the contents of the requested file if present

Client TCP.py

```
from socket import *
servername = '127.0.0.1'
serverport = 12000
clientsocket = socket(AF_INET, SOCK_STREAM)
clientsocket.connect((servername,))
sentence = input("\n Enter file name:")
clientsocket.send(sentence.encode())
filecontents = clientsocket.recv(1024).decode()
print("\n From Server:\n")
print(filecontents)
clientsocket.close()
```

Server TCP.py

```
from socket import *
servername = '127.0.0.1'
serverport = 12000
serversocket = socket(AF_INET, SOCK_STREAM)
serversocket = bind((servername, serverport))
serversocket.listen(1)
while 1:
print("The server is ready to receive")
f = open(sentence, "r")
l = f.read(1024)
connectionsocket.send(l.encode())
print("\n Sent contents of " + sentence)
```

file.close()
ConnectionSocket.close()

O/P:
Server side:
The server is ready to receive

Client side:
Enter file name: serverTCP.py
The contents of file serverTCP displayed here

Server side:
Sent contents of serverTCP.py

LAB PROGRAM-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.

SOLUTION:

```
ClientUDP.py from socket import * serverName = "127.0.0.1"
serverPort = 12000 clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))

filecontents,serverAddress = clientSocket.recvfrom(2048) print
("\nReply from Server:\n") print
(filecontents.decode("utf-8")) # for i in filecontents:
# print(str(i), end = '')
clientSocket.close() clientSocket.close()

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

OUTPUT:

Client:

```
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/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()

>>>
```

Server:

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =
The server is ready to receive

Sent contents of  serverUDP.py
```

LAB - 1c

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

Client UDP.py

```
from socket import *
servername = "127.0.0.1"
serverport = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter the file name:")
file contents, serverAddress = clientSocket.recvfrom(2048)
print("\n Reply from server:\n")
print(file contents.decode("utf-8"))
# for i in fileContents:
# print (str(i), end="")
clientSocket.close()
clientSocket.close()
```

serverUDP.py

```
from socket import *
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(20)
Server socket.sendto(bytes(con,'UTF
client address)
print("sent contents of and:")
print(sentence)
# for i in sentence
# print(str(i), end="")
file.close()

o/p: Server side
The server is ready to receive

Client side
Enter file name: ServerUDP.py
The contents of the file server UD
are displayed here

Server side
Sent content of servaUDP.py

LAB-1c

Tool explanation - Wireshark

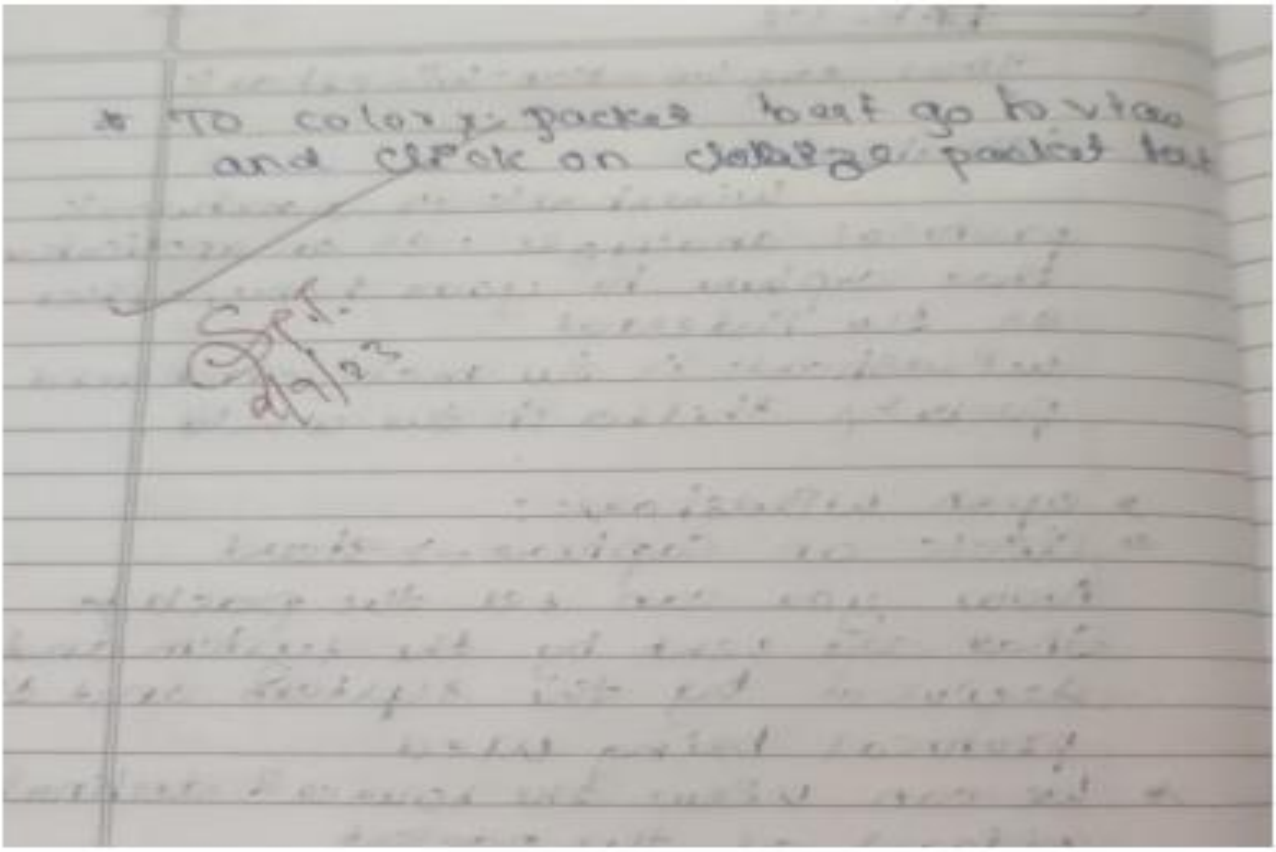
Wireshark:

Wireshark is a network protocol analyzer, or an application that captures to your home, office or the Internet.

Wireshark is the most often used packet sniffer in the world.

* Open Wireshark:
* Click on capture -> start
  Now you can see the packets that are sent by the system and received by the system and the protocol being used
* We can view the source & destination address of the packet
* If we click on a particular packet now you can see the ASCII code in the bottom of the page
* Wireshark uses different colours for the different protocols to represent
* To see only your system participate in network in display filter type ip address == ip address of the PC

  Ex: ip address = 10.124.2.1
* We can even filter packets by the type of the protocol
  Go to analyze & display filters to see the filter & click on stop



* To colour packet text go to view and click on colorize packet text