BMS COLLEGE OF ENGINEERING
BENGALURU Autonomous Institute, Affiliated to VTU

A Lab Report on

Computer Networks Lab

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:
Raghav KR
1BM21CS150

Under the Guidance of:
Dr. Nandini Vineeth
Assistant Professor
BMSCE

Department of Computer Science and Engineering
BMS College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560019

2022-2023

CERTIFICATE

This is to certify that the Lab work entitled "COMPUTER NETWORKS LAB" was carried out by **Raghav KR(1BM21CS150),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **COMPUTER NETWORKS - (22CS4PCCON) work prescribed for the said degree.**

Prof. Swathi Sridharan Assistant Professor Department of CSE BMSCE, Bengaluru

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

INDEX

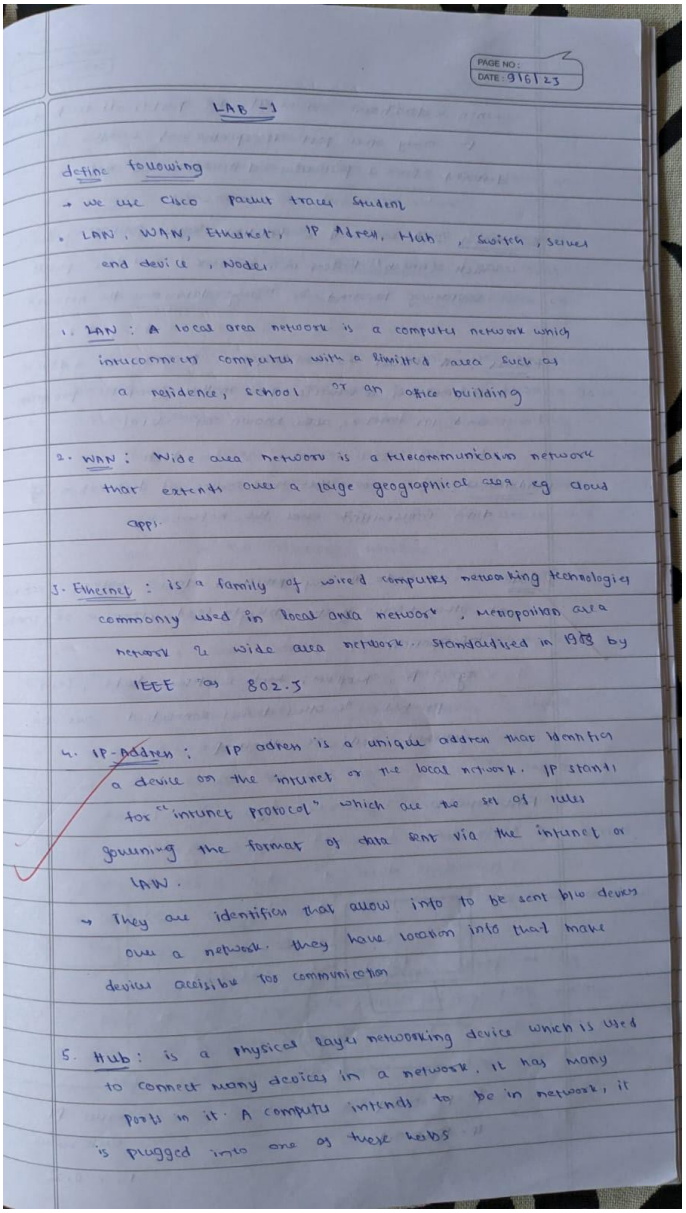
Sl. No.	Date	Experiment Title
		CYCLE - 1
1	15/06/2023	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.
2		Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.
3		Configure default route, static route to the Router.
4		Configure DHCP within a LAN and outside LAN.
5		Configure Web Server, DNS within a LAN.
6		Configure RIP routing Protocol in Routers
7.1		Configure OSPF routing protocol
7.2		To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

8.1		To construct a VLAN and make the PC's communicate among a VLAN.
8.2		To construct a WLAN and make the nodes communicate wirelessly.
9		Demonstrate the TTL/ Life of a Packet.
10	10/08/2023	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.
CYCLE - 2		
11	17/08/2023	Write a program for error detecting code using CRC CCITT (16-bits).
12		17/08/2023 Write a program for congestion control using Leaky bucket algorithm.
13		Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.
14	24/08/2023	Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

LAB 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-



PAGE NO:
DATE: 9/6/23

LAB -1

define following

→ we use Cisco packet tracer student

• LAN, WAN, Ethernet, IP Adress, Hub, Switch, Server and device, Nodes

1. LAN : A local area network is a computer network which interconnects computers with a limited area such as a residence, school or an office building

2. WAN : Wide area network is a telecommunication network that extends over a large geographical area eg cloud appl.

3. Ethernet : is a family of wired computer networking technologies commonly used in local area network, Metropolitan area network & wide area network. Standardized in 1983 by IEEE as 802.3

4. IP-Address : IP adress is a unique address that identifies a device on the internet or the local network. IP stands for "internet protocol" which are the set of rules governing the format of data sent via the internet or LAN.

→ They are identifiers that allow info to be sent b/w devices over a network. they have location info that make devices accisible for communication

5. Hub : is a physical layer networking device which is used to connect many devices in a network, it has many ports in it. A computer intends to be in network, it is plugged into one of these hubs

when a dataframe arrives at the port, it is broadcast to every other port irrespective of whether it is destined for a particular destination or not.

6. Switches : A switch is a data link layer networking device which connects devices in a network & uses packet switching to send & recieve data over the network

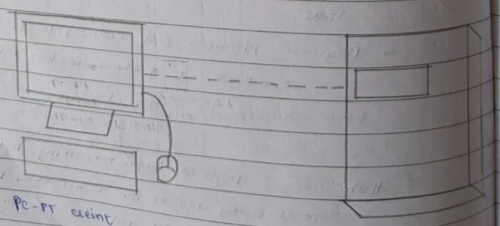
7. Servers : A server is a computer program or a device that provides a service to another computer program and its user, also known as client

8. End devices : They are either the source or destination of data transmitted over the network

9. Node : a node is any physical device within a network of other tools that able to send, receive or config information.

Eg : PC, modems, tetli switches, hubs, servers, Printers & other devices connected over the internet with or ethernet.

First network



PC-PT client
IP - 10.0.0.1

Server - PT
web server
10.0.0.2

PAGE NO:
DATE:

Observation for the week (all things done.)

1. Open Cisco packet tracer
2. go to help section in the top ribbon and click on contents. → This takes us to documentation
3. on the left most part, index exists under which there is a tab called getting started, under that "My first PT lab" menu can be seen, click on that
4. skip the "viewing help & tutorials" section as it familiarises you with the UI & head to "Creating your first network" section which is a subheading
   → There are 10 substeps under this
   → on the bottom left corner, the first icon is end devices, click on that, then all devices will appear in the adjacent window
   → drag & drop the "PC-PT" to "Server-PT" on the work space
   → in the same bottom left corner, there is a "lightning icon" which is connections
   → click on that and connect using "copper straight through" which is the third icon
   → then check the red lights or pink indicating that it is not a working connection.
   → on the right most column ribbon of the screen select delete icon & delete connection
   → replace straight copper with copper cross-over connection the lights turn green.
   → hovering mouse is letting mouse status as "up"
   → click on the PC & turn it on / off on the GUI image (find power button)
   → toggle this button for both servers & PC & notice the connection turning red & green

PAGE NO:
DATE:

→ three ways to access to server, first is to hover mouse, second is click on it to open the config window and third to use inspect tool and click on PC & check dhcp radio is empty

→ ~~ope~~ click on pc, in the config tab, set dns & IP address as 10.0.0.1 & 10.0.0.2 for client & servers respectively

→ You can also configure into the desktop tab to ensure that port status is checked.

→ open server config, set name, ip and click add ensuring dns is on.

→ Add description using ① button.

→ ~~Save~~ click on tiled background.

→ save the file

UI command prompt output

→ positive ping (exciting)

PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128
reply from 10.0.0.1 : bytes = 32 time = 5ms TTL = 128
reply from 10.0.0.1 : bytes = 32 time = 4ms TTL = 128
reply from 10.0.0.1 : bytes = 32 time = 5ms TTL = 128

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 = 5ms, Average = 3ms

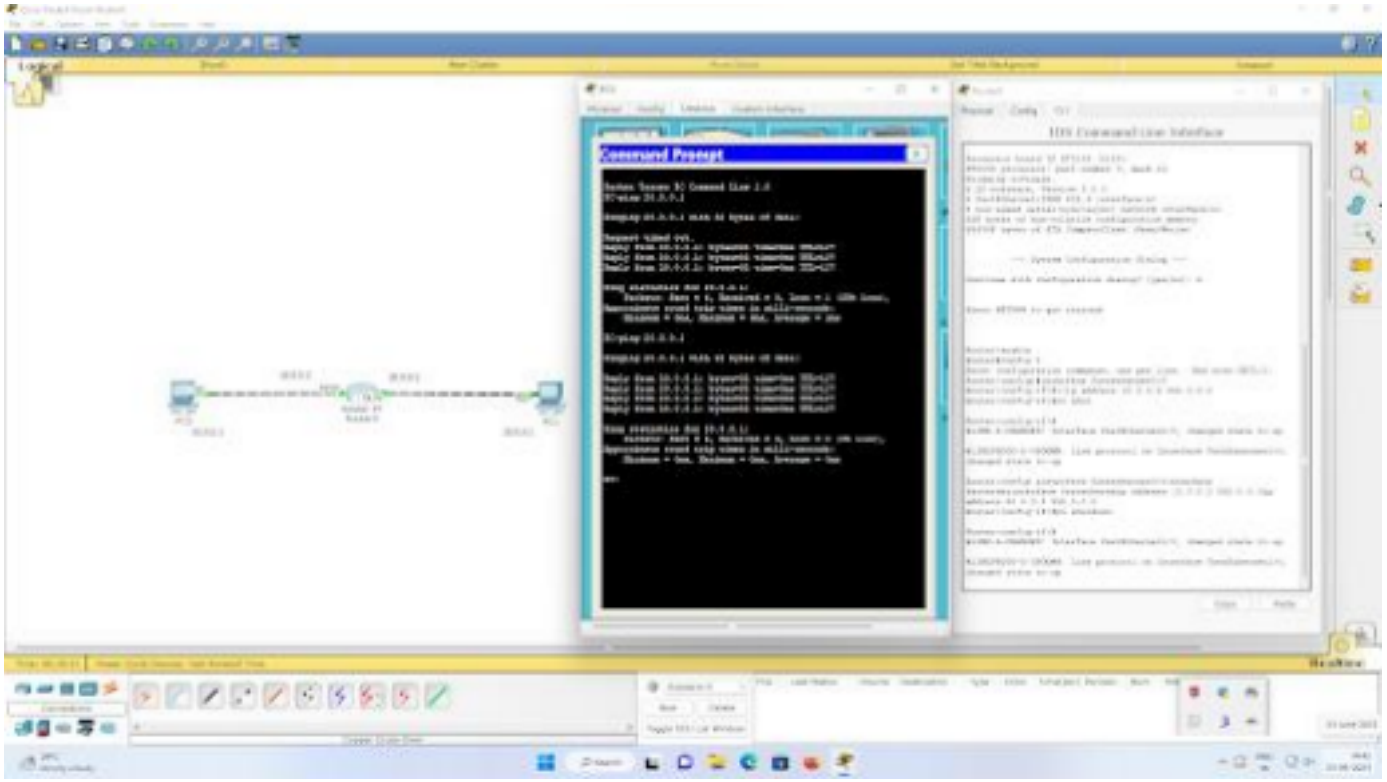
PAGE NO :
DATE :

→ -ve pinging (not existing.

```
PC > Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data :

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

9/6/23

System Output-



LAB 2

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

Observation-

PAGE NO :
DATE : 16/6/23

LAB - 2

Q. Create a topology & simulate sending a simple PDU from src → destination using simple hub & switch as connecting domain

Step 1: drag and drop a generic hub PT on the work-space

Step 2: drag and drop 6 pc's & configure the ip addresses for each of the 6 pcs and make the ip 10.0.0.1 - 10.0.0.6

Step 3: Connect the pc's to the hub & transmit data blw them

Step 4: drag & drop the switch and repeat the same

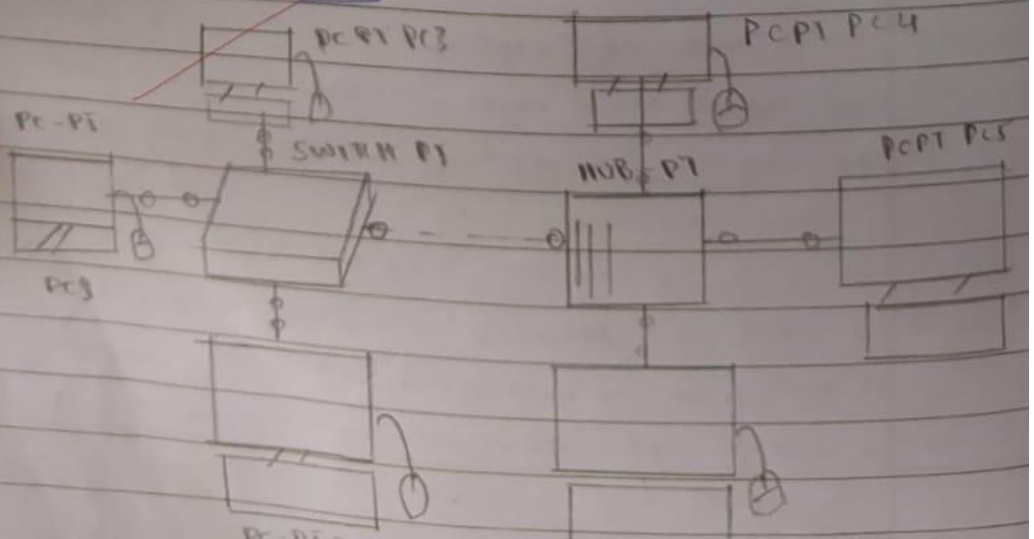
Step 5: test package transmission by sending PDU: blw the PC's the process is a success

Step 6: Connect the hub & the switch on & transmit data blw them the process succeeds

Step 7: Transmit data blw hub pc & switch pc and try again toggling the switch on & off

from ip 10.0.0.1 → 10.0.0.5.

diagram of topology



PAGE NO:
DATE:

Command Prompt Pinging a Pc

Pc > Ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5 : byte = 32 time = 0ms TTL = 120
Reply from 10.0.0.5 : bytes = 32 time = 0ms TTL = 120
Reply from 10.0.0.5 : bytes = 32 time = 0ms TTL = 120
Reply from 10.0.0.05 : bytes = 32 time = 0ms TTL = 120

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 = 0ms, Average = 0ms

when switch is turned off

Pinging 10.0.0.4 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out
Request timed out.

?

Ping statistics for 10.0.0.4:
Packets: sent = 4, received = 0, lost = 4 (100% loss)

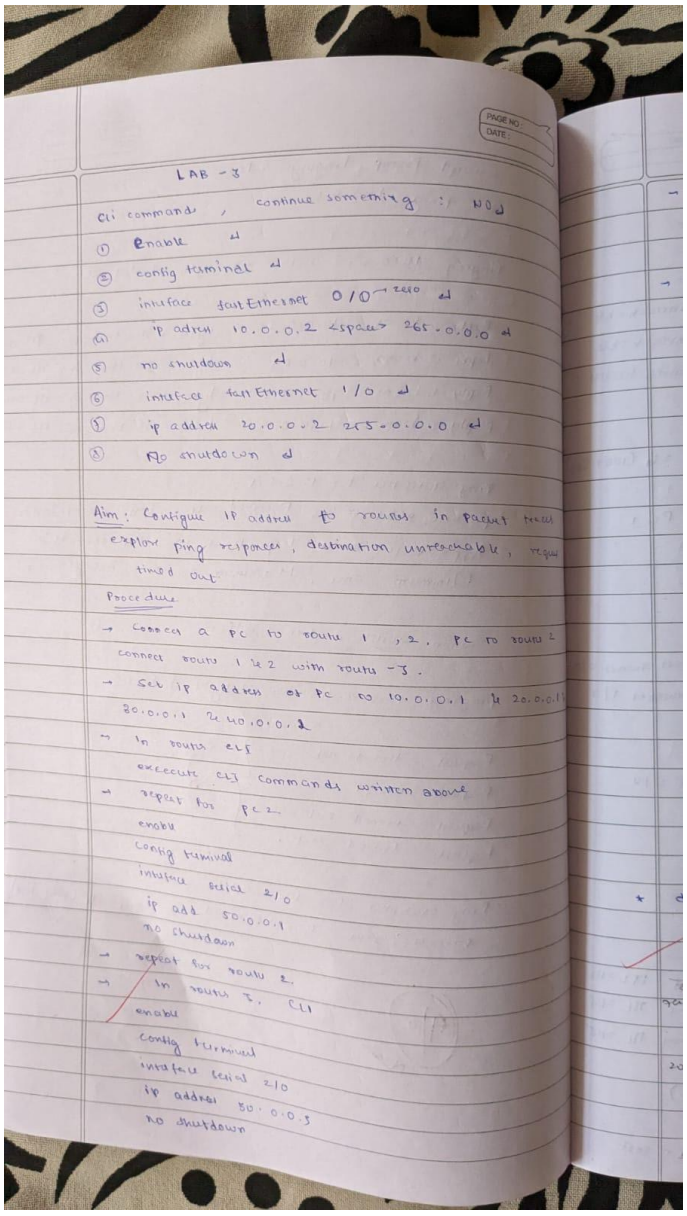
4/6

9/10

System Output-

LAB 3
Configure default route, static route to the Router.

Observation-



PAGE NO :
DATE :

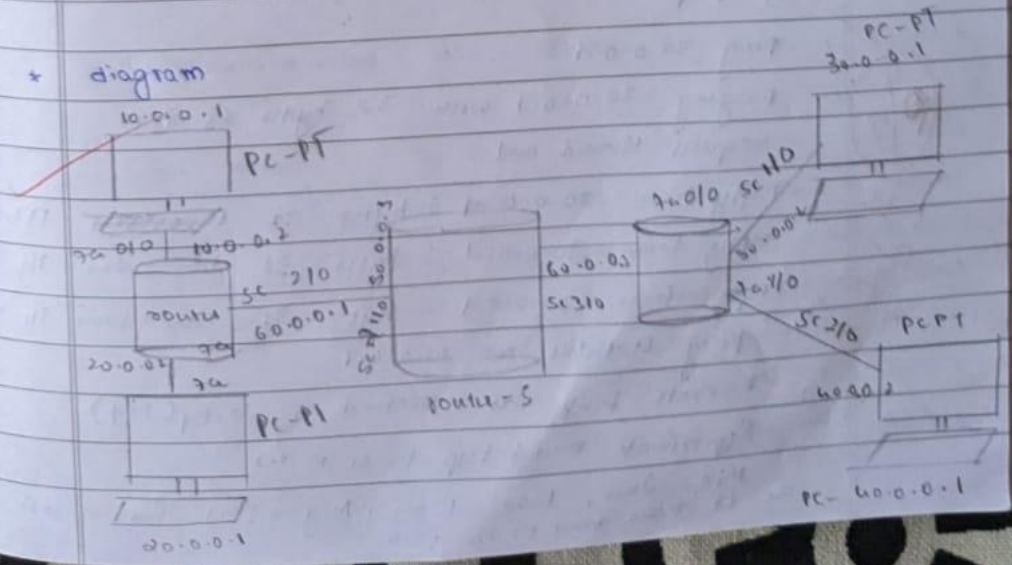
→ interface serial 2/0
ip address 60.0.0.3
no shutdown

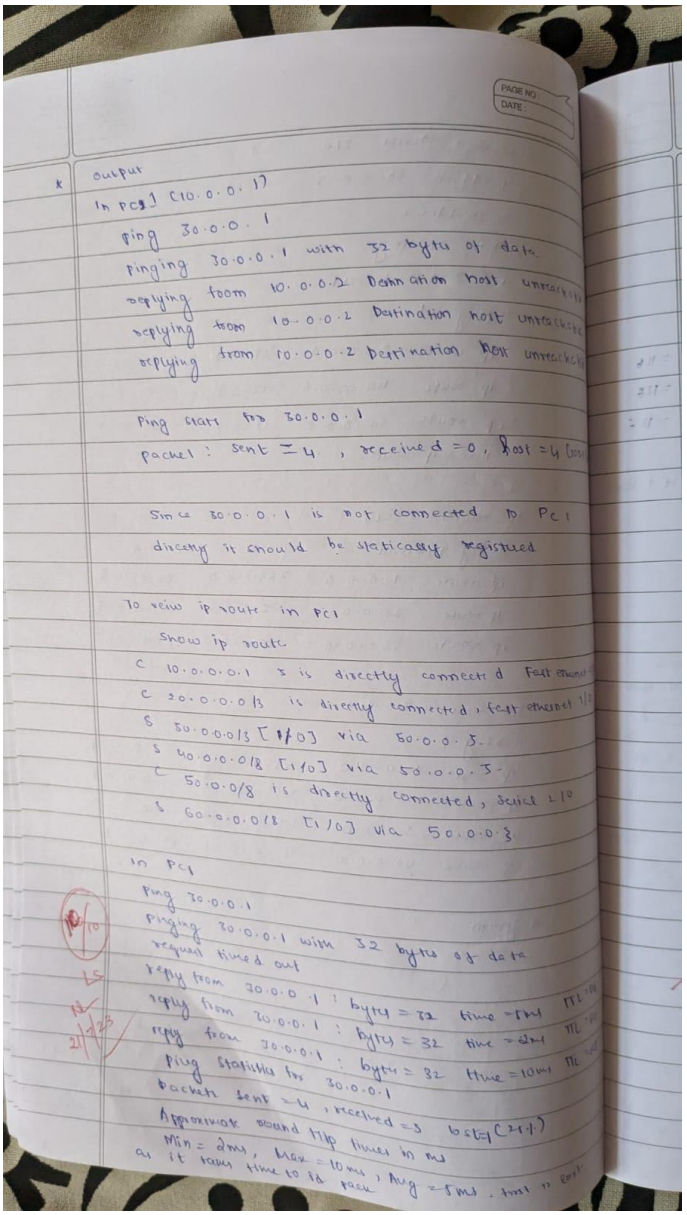
→ static routing
in router 1 CLI
config t
ip route 30.0.0.0 255.0.0.0 50.0.0.3
ip route 40.0.0.0 255.0.0.0 50.0.0.5
ip route 60.0.0.0 255.0.0.0 50.0.0.3
show ip route

in router 2 CLI
config terminal
ip route 10.0.0.0 255.0.0.0 60.0.0.3
ip route 20.0.0.0 255.0.0.0 60.0.0.5
ip route 50.0.0.0 255.0.0.0 60.0.0.7

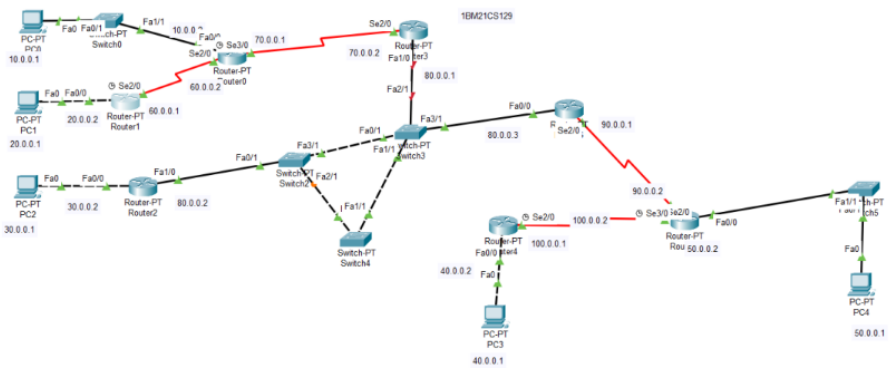
in router 3 CLI
config t
ip route 10.0.0.0 255.0.0.0 50.0.0.1
ip route 20.0.0.0 255.0.0.0 50.0.0.0.1
rp route 30.0.0.0 255.0.0.0 60.0.0.0
ip route 40.0.0.0 255.0.0.0 60.0.0.1

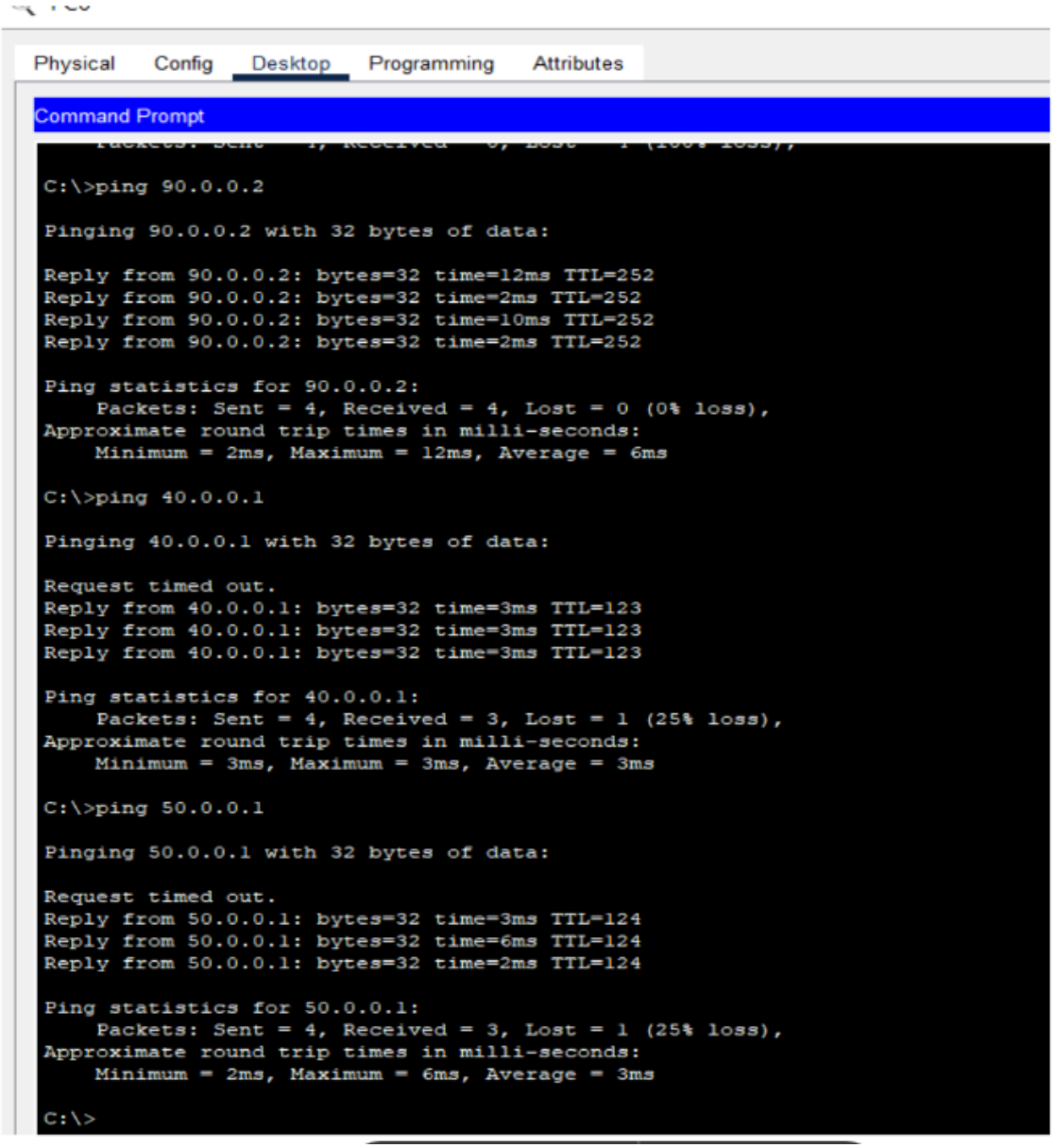
* diagram




PAGE NO.
DATE
output
In PC$ ping 10.0.0.1
ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data
replying from 10.0.0.2 Destination host unreachable
replying from 10.0.0.2 Destination host unreachable
replying from 10.0.0.2 Destination host unreachable
Ping stats for 30.0.0.1
packet : sent = 4 , received = 0 , lost = 4
Since 30.0.0.1 is not connected to PC1 directly it should be staticaly registered
To veiw ip route in PC1
show ip route
C 10.0.0.0/8 is directly connected Fast ethernet
C 20.0.0.0/8 is directly connected, fast ethernet 1/0
S 30.0.0.0/8 [1/0] via 50.0.0.5
S 40.0.0.0/8 [1/0] via 50.0.0.5
C 50.0.0.0/8 is directly connected, serial 2/0
S 60.0.0.0/8 [1/0] via 50.0.0.3
in PC1
Ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data
request timed out
reply from 30.0.0.1 : bytes = 32 time = 1ms
reply from 30.0.0.1 : bytes = 32 time = 2ms
reply from 30.0.0.1 : bytes = 32 time = 10ms
Ping statistics for 30.0.0.1
Packets sent = 4 , received = 3
Approximate round trip times in ms
Min = 2ms, Max = 10ms, Avg = 5ms
as it takes time to fix path

Topology:


1BM21CS129
PC-PT PC0
10.0.0.1
Switch0
Router0
Router1
PC-PT PC1
20.0.0.1
Router2
PC-PT PC2
30.0.0.1
Switch2
Switch3
Switch4
Router3
Router4
PC-PT PC3
40.0.0.1
PC-PT PC4
50.0.0.1
Router-PT Rou
50.0.0.2
Switch5
70.0.0.1
70.0.0.2
80.0.0.1
80.0.0.2
80.0.0.3
90.0.0.1
90.0.0.2
100.0.0.1
100.0.0.2
60.0.0.1
60.0.0.2
20.0.0.2
30.0.0.2
40.0.0.2



Experiment 4

Configure DHCP within a LAN and outside LAN

PAGE NO:
DATE:

Week - 4 30/6/22

Configure default route, static route to the router.

- connect devices as shown in topology
- set ip address to all PC's and routers for router.

enable
config t
interface four Ethernet / serial 0/0
ip address 10.0.0.1 255.0.0.0
no shut

- Default route for router 1, router 5 & router 3

router 1 : enable
config t
ip route 0.0.0.0 000.0.0 60.0.0.5

router 3 : enable
config t
ip route 0.0.0.0 0.0.0.0 50.0.0.5

router 5 : enable
config t
IP route 0.0.0.0 0.0.0.0 10.000.1

- Static routing for router 0, route 6, router 4 & router 2.

→ for router 0 ip route 20.0.0.0 255.0.0.0 60.0.0.1
IP route 80.0.0.0 255.0.0.0 70.0.0.1
ip route 40.0.0.0 255.0.0.0 70.0.0.1
ip route 50.0.0.0 255.0.0.0 70.0.0.1
ip route 80.0.0.0 255.0.0.0 70.0.0.1
ip route 100.0.0.0 255.0.0.0 70.0.0.1
ip route 110.0.0.0 255.0.0.0 70.0.0.1

Similarly for router 2, router 4 & router 2

O/P:

ping 20.0.0.1

Pinging from 20.0.0.1 with 32 bytes of data

request timed out

reply from 20.0.0.1 : bytes = 32 time = 7ms TTL = 14

reply from 20.0.0.1 : bytes = 32 time = 7ms TTL = 14

reply from 20.0.0.1 : bytes = 32 time = 7ms TTL = 14

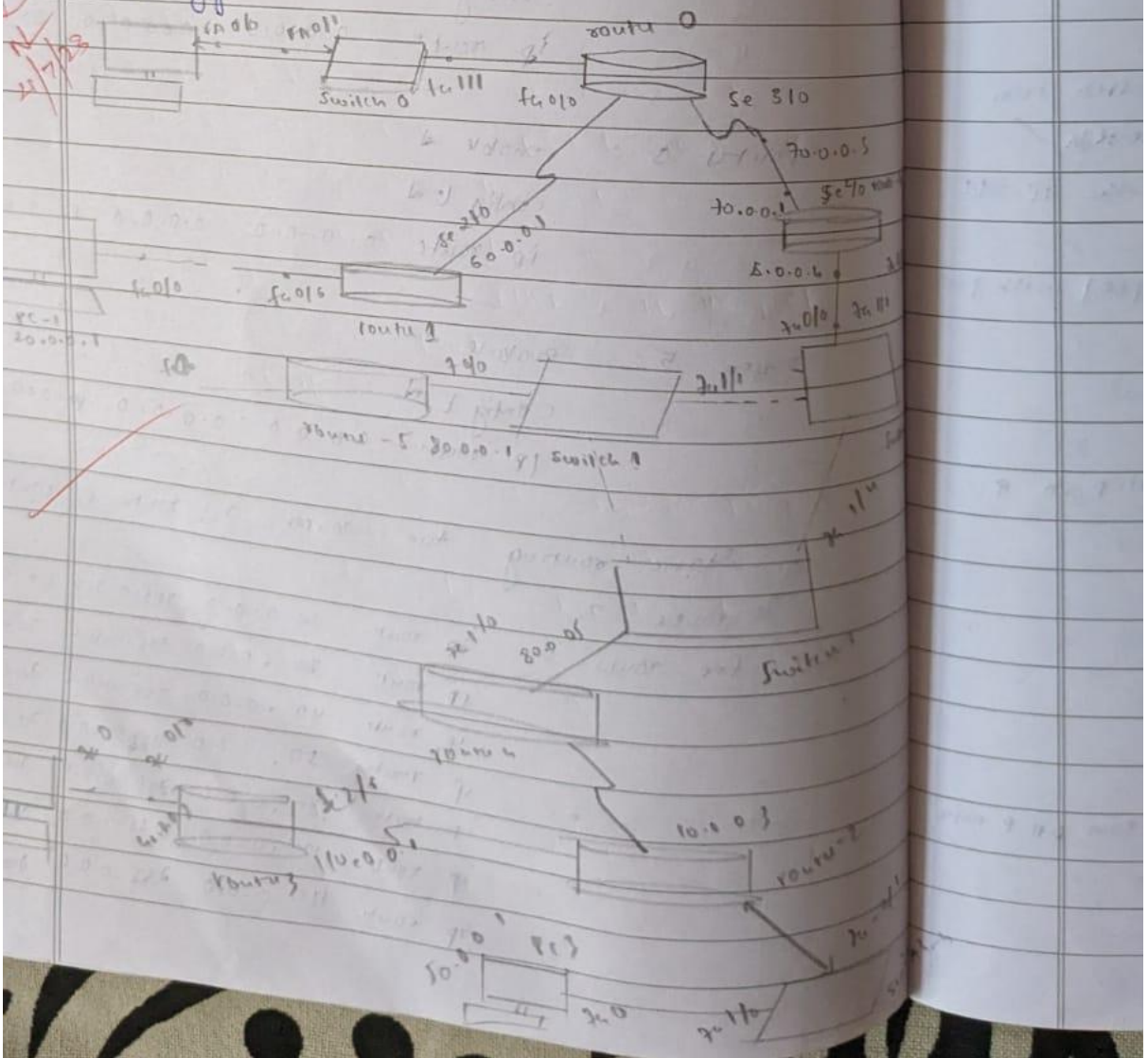
ping statistics for 20.0.0.1 :

packets : Sent = 4, received = 3, lost = 1 (25% loss)

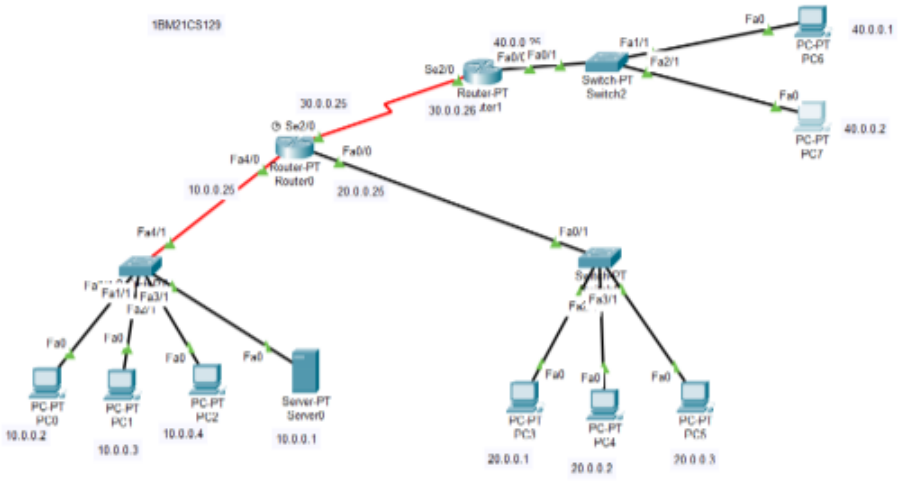
Approximate round trip times in ms

min = 6ms, max = 7ms, Avg = 6 ms

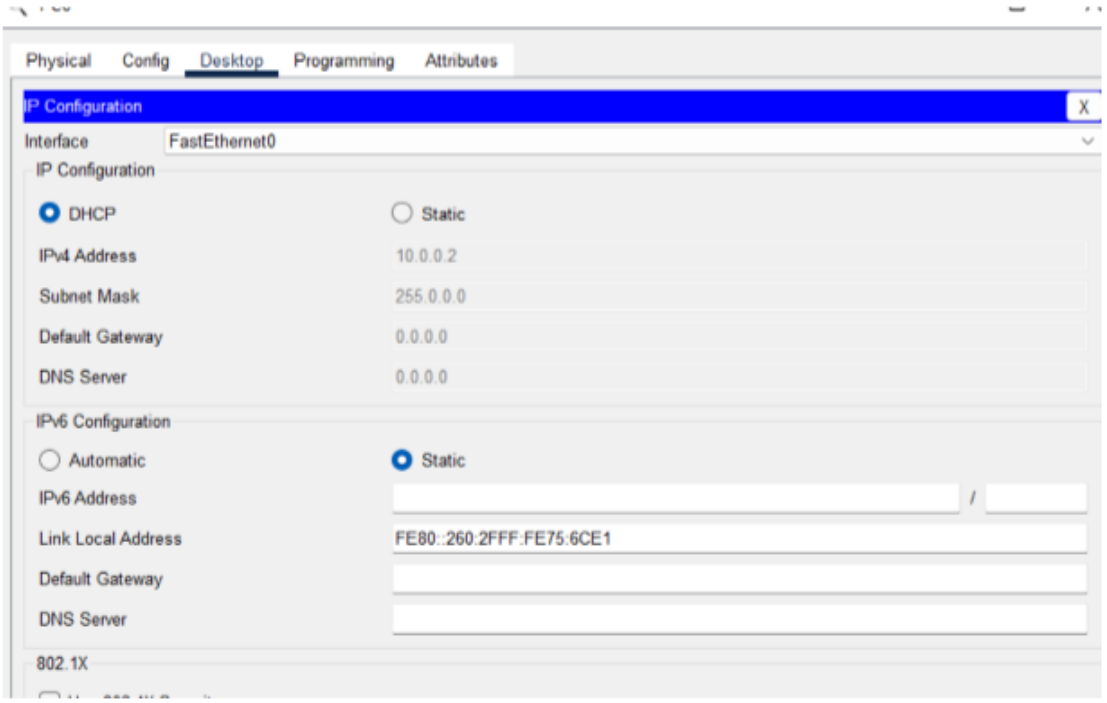
topology



Topology and output:



Output:



Experiment 5

Configure RIP routing Protocol in Routers

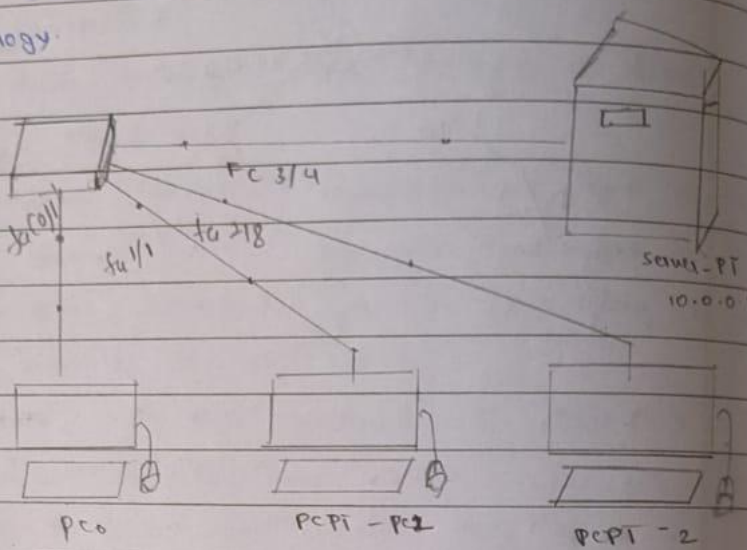
PAGE NO:
DATE:

LAB - 5

Aim: Configure DHCP within a LAN and outside LAN

for DHCP within LAN

topology:



Procedure:

1. Drag and drop 3 PC's and 1 server, 1 switch from the devices. Connect the 3 PC's to the switch
2. Connect the server to the switch. Give the IP address to the server as 10.0.0.1
3. go to the services → DHCP, add the pool with pool name as Server Pool

Start IP 10.0.0.2

Subnet mask

4. Now go to the PC1 and switch to DHCP in IP configuration. IP add will be assigned to the PC's [PC0, PC1, PC2]

for DHCP outside the lan,

Procedure

1. repeat the procedure we did for the lan DHCP with a lan

PAGE NO :
DATE :

2. Now add a router, another set of PCs and a switch [Ion-2]

3. Config the ip address of the ports from fa 0/0 to router 0 using following commands

→ router enable ↵
router config t
router (config) # ip interface fast Ethernet 4/0
ip address 10.0.0.25 255.0.0.0
noshut ↵
exit

Similarly repeat for router 0/0 port

check default gateway during server/pools.

4. Now, give the command for the helper ip address.
router # config t
interface fast ethernet 0/0
ip-helper-address 10.0.0.1

5. Go to the PC's [PC3, PC4, PC5] and switch to DCP in ip config mation.

6. Connect router 1 to the router 0, router 1 should be connected to the switch and 2 PC's

7. Config the ip of router 1 for Send 2/0 & fa 0/0 with 30.0.0.26 and 40.0.0.25 respectively

8. config ip of router 0 of serial 0/0 with ip 30.0.0.25

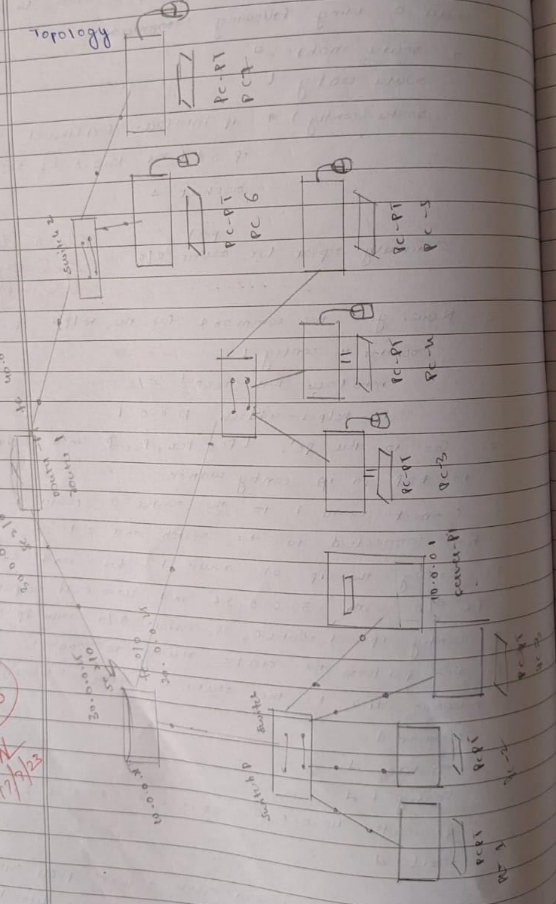
9. Now perform the state routing to connect network 40 to the router 0 with following command
enable ↵
Config t ↵
ip route 40.0.0.0 255.0.0.0 30.0.0.26 ↵
exit ↵

10 Go to the server and create 2 more pools with Starting address 20.0.0.2 & 40.0.0.2. give default gateway a 10.0.0.25

PAGE NO.:
DATE: 21/09/23

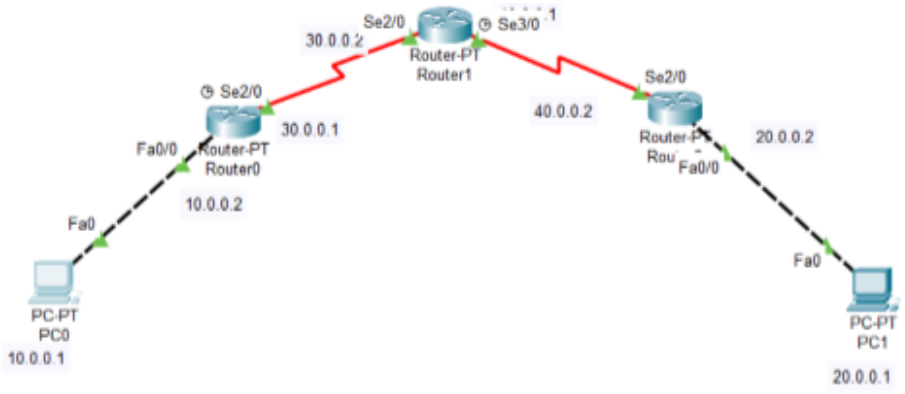
12. PC's in the network 40 with be provided with ip-address 40.0.0.2, 40.0.0.3 using DHCP

Topology

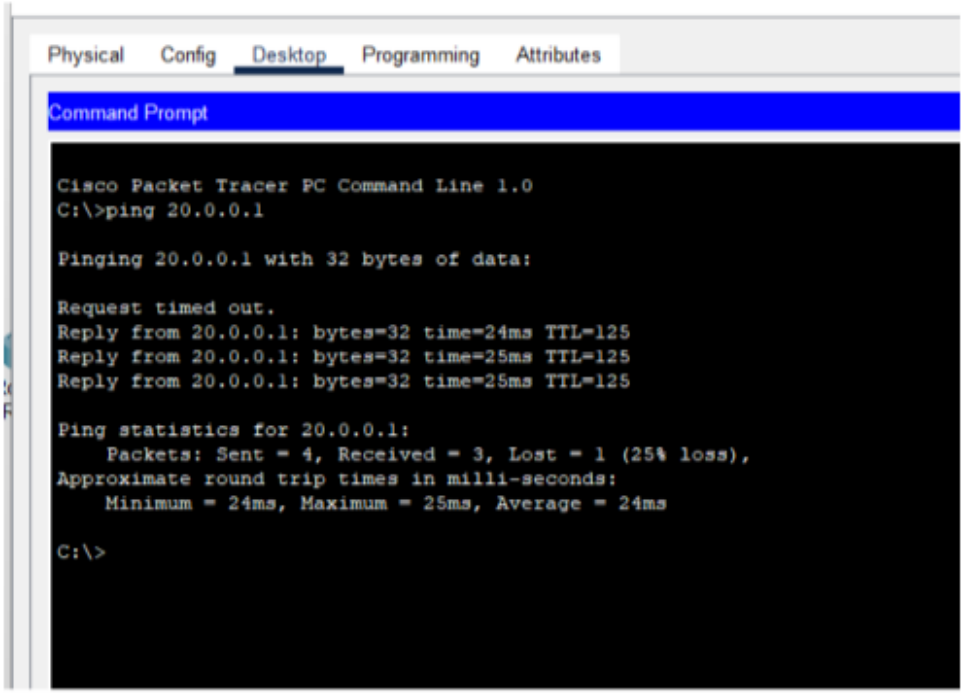


10/10
17/9/23

9/10



Output:



Experiment 6

Domain name server configuration

21/03/23

PAGE NO :
DATE :

Domain name server (DNS). Q-?

Procedure.

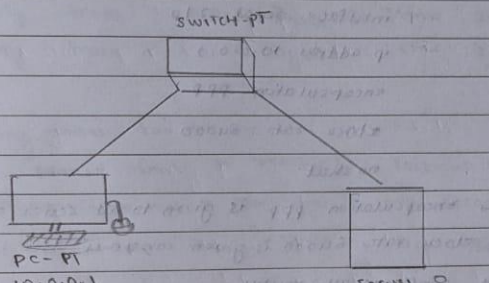
→ Create a topology by placing a Pc, a server & a switch on the workspace.

→ config pc & server IP address + gateway.

→ open web browser of pc to set ip to set ip of server.

→ open config DNS of server with name (with name) & URL.

→ Edit index.html to display USN and name topology.



9/10

Output:

28/3/23

Web Boowser.

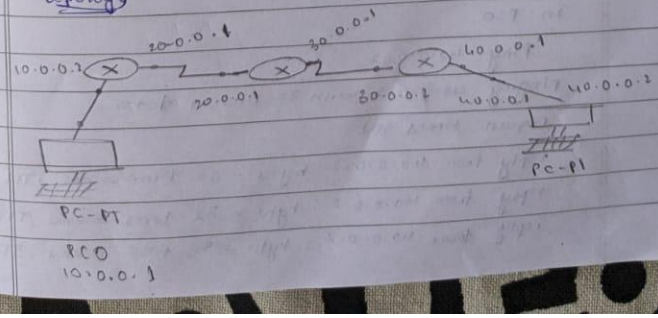
URL : https://helloraghav.com

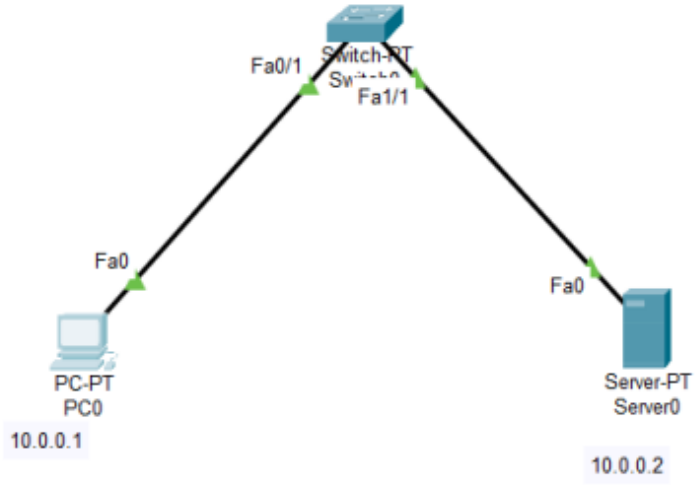
USN : 1BM21CS150

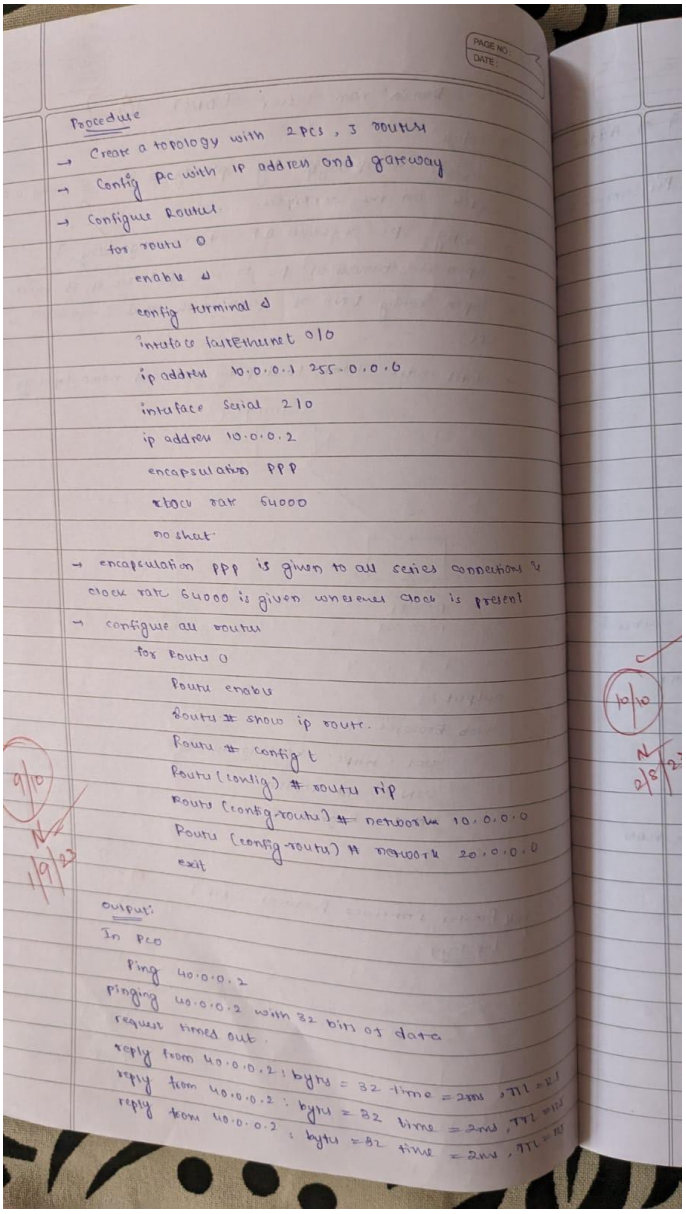
NAME: Raghav.KR

Configure Routing Interface Protocol. (RIP) Q-?

topology







Procedure

→ Create a topology with 2 PCs, 3 routers
→ Config Pc with ip address and gateway
→ Configure Routers
for router 0
enable
config terminal
interface fastEthernet 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 & clock rate 64000 is given wherever 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

output:
In PC0
Ping 40.0.0.2
Pinging 40.0.0.2 with 32 bits of data
request timed out
reply from 40.0.0.2 : byte = 32 time = 2ms, TTL =
reply from 40.0.0.2 : byte = 32 time = 2ms, TTL =
reply from 40.0.0.2 : byte = 32 time = 2ms, TTL =

PAGE NO :
DATE :

Ping statistics for 40.0.0.2

packets : Sent = 4, received = 3, lost = 1 (25% lost)

Approximate round trip times in ms

Minimum = 2ms, Maximum = 13ms, Avg = 8ms

Ping 40.0.0.2

pinging 40.0.0.2 with 32 bit of data

Reply from 40.0.0.2 bytes = 32 time : 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32 time : 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32 time : 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32 time : 2ms TTL = 125

Ping stats for 40.0.0.2:

packets sent 4 received 4 lost = 0 (0% loss)

Approximate round trip times in milli-seconds

min = 2ms, max = 13ms, Avg = 8ms

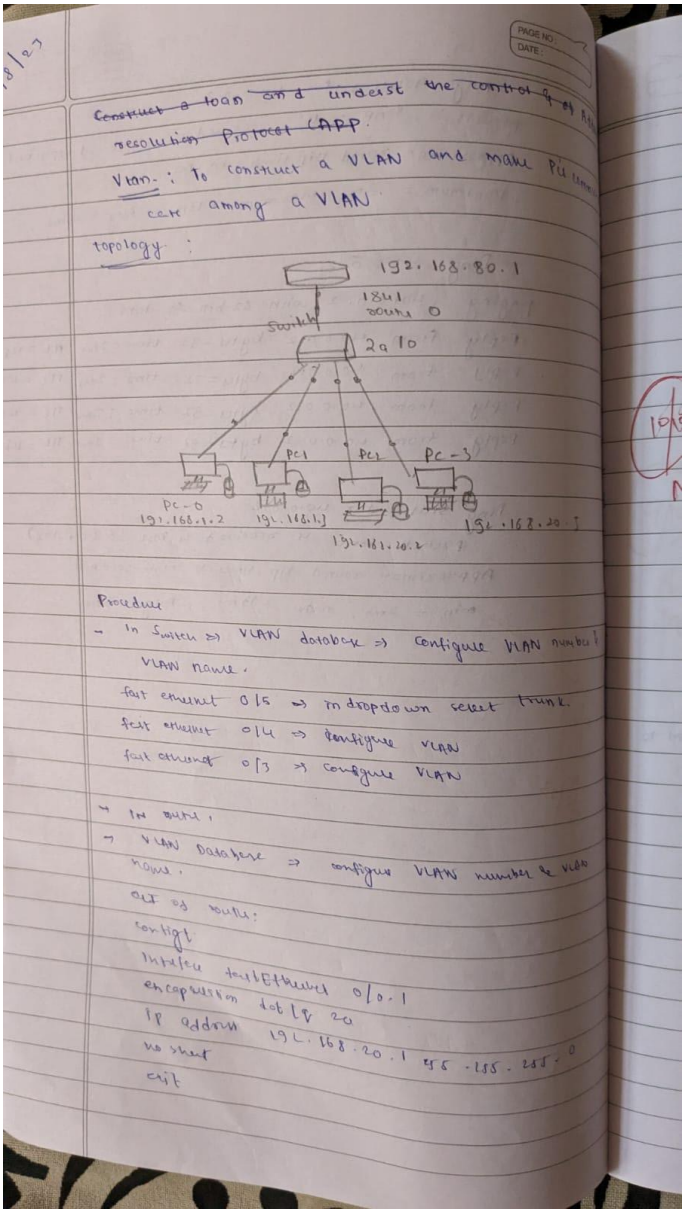
10/10

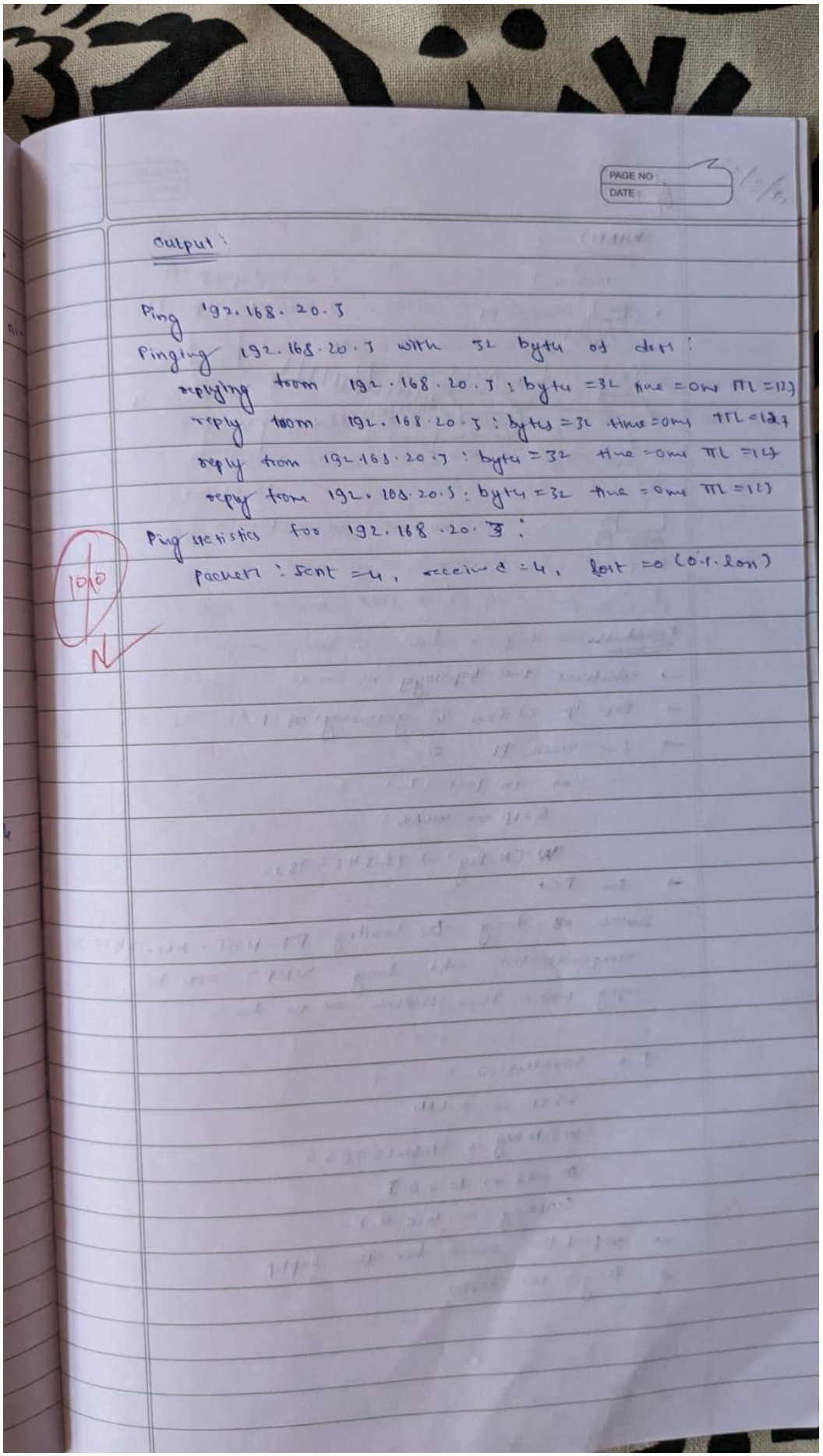
N

2/8/23

Experiment 7

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





Output :

Ping 192.168.20.3

Pinging 192.168.20.3 with 32 byte of data :

replying from 192.168.20.3 : byte = 32 time = 0ms TTL = 127

reply from 192.168.20.3 : byte = 32 time = 0ms TTL = 127

reply from 192.168.20.3 : byte = 32 time = 0ms TTL = 127

reply from 192.168.20.3 : byte = 32 time = 0ms TTL = 127

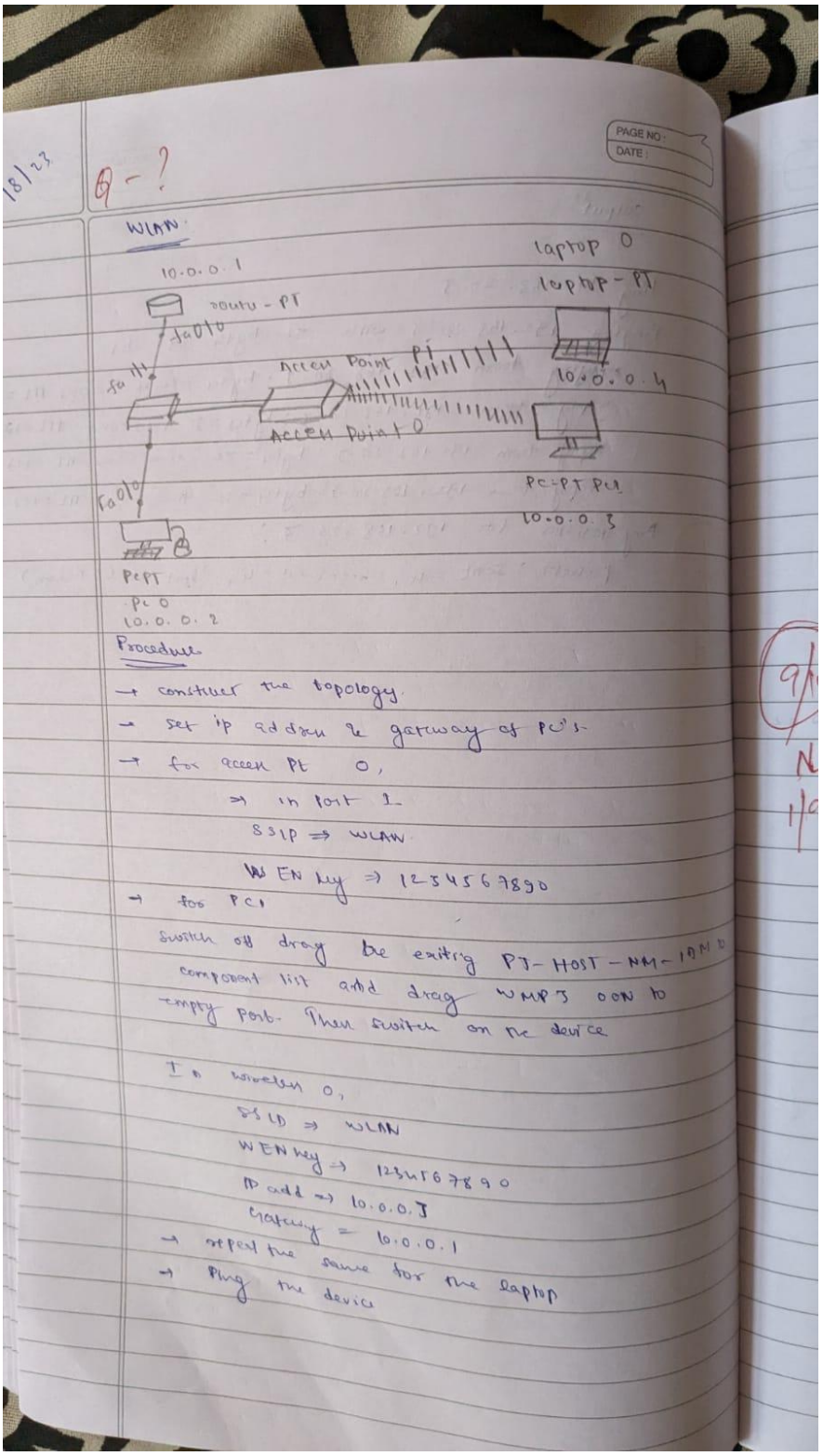
Ping statistics for 192.168.20.3 :

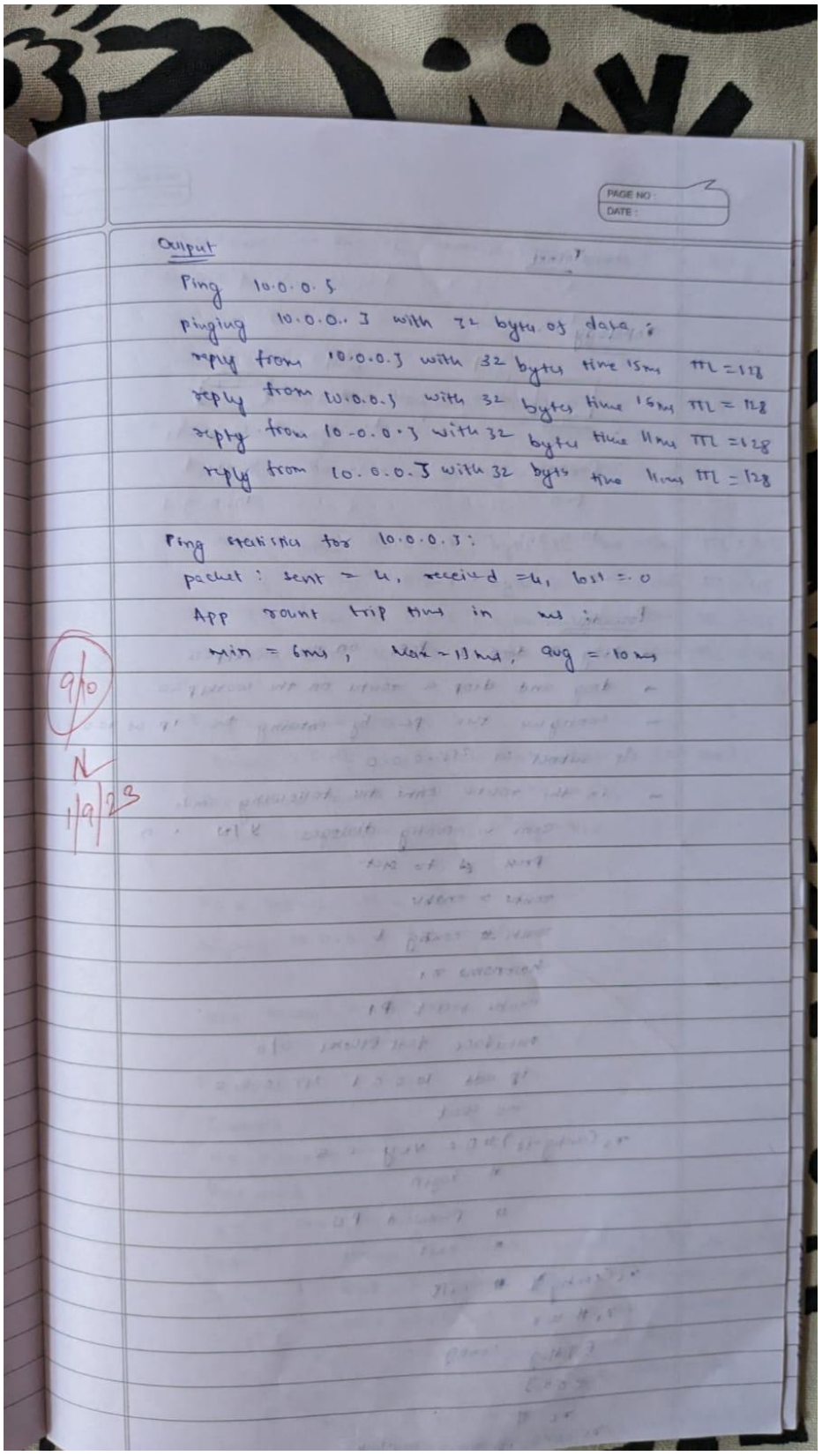
packet : sent = 4, received = 4, lost = 0 (0% loss)

10/10

Experiment 8

To construct a WLAN and make the nodes communicate wirelessly





Experiment 9

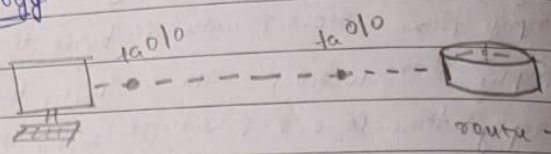
To understand the operation of TELNET by accessing the router in server room from a PC

in IT office.

PAGE NO:
DATE:

Telnet

Topology



Pc-PT
Pc0
10.0.0.2

router-PT
router1
10.0.0.1

Procedure:

→ drag & drop a PcPt on the workspace
→ drag and drop a router on the workspace
→ configure the pc by entering the ip as 10.0.0.2 & subnet as 255.0.0.0,
→ in the router enter the following cmds

```
cont w config dialogue y/n : n
Press ↵ to start
router > enable
router # config t
hostname r1
enable secret p1
interface fa1 Ethernet 0/0
ip add 10.0.0.1 255.0.0.0
no shut
r1(config-if)# line vty 0 5
# login
# password p0
# exit
r1(config)# exit
r1# wr
Building config ...
[OK]
r1#
r1 conn is now available
```

PAGE NO :
DATE :

→ headover to the Pc terminal (cmd prompt) & ping 10.0.0.1

output

Packet Tracer Pc Command Line 1.0
Pc > Ping 10.0.0.1

Pinging 10.0.0.1 w 32 bytes of data:

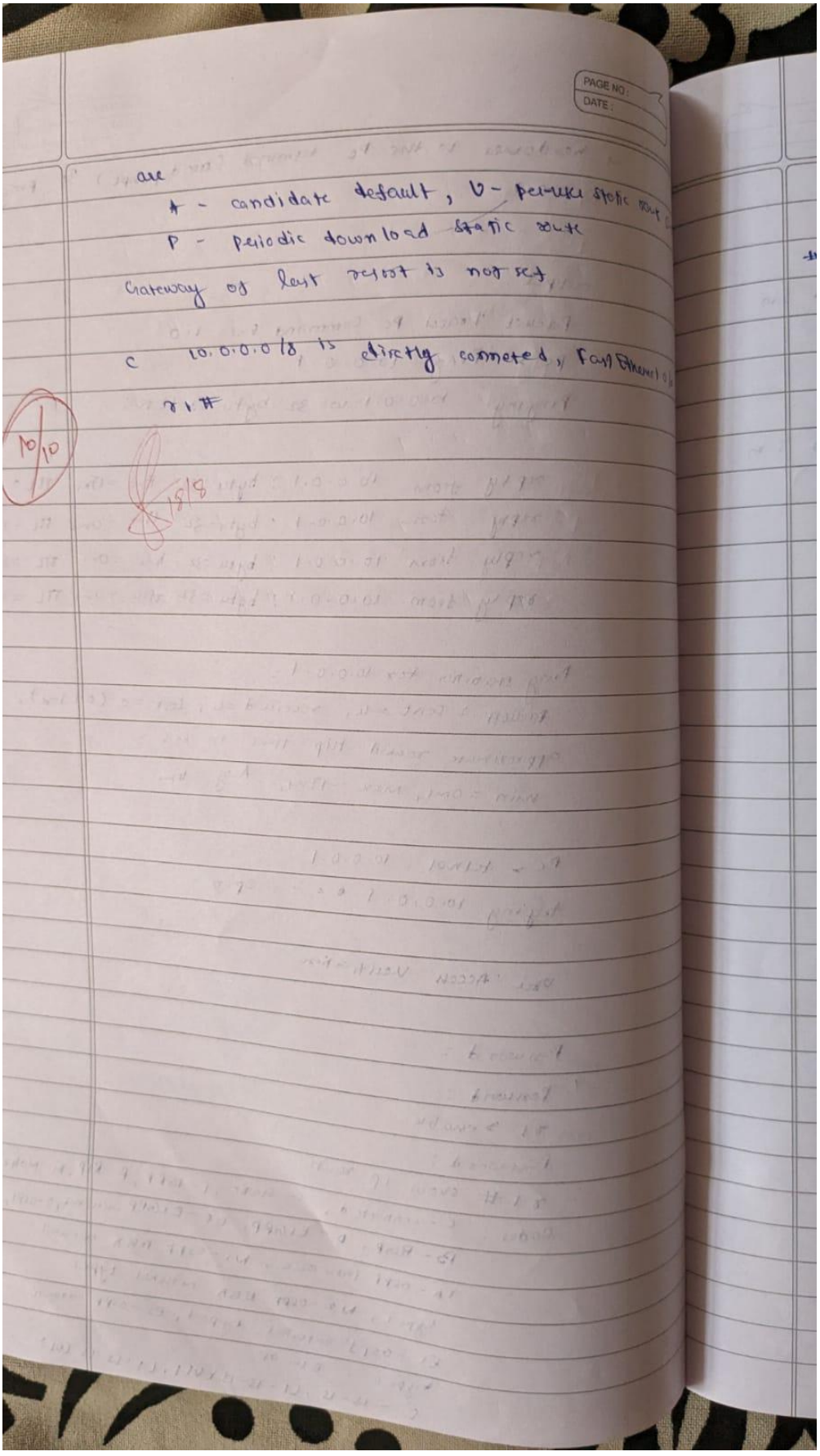
reply from 10.0.0.1 : bytes=32 time=17ms TTL=255
reply from 10.0.0.1 : bytes=32 time=0ms TTL=255
reply from 10.0.0.1 : bytes=32 time=0ms TTL=255
reply from 10.0.0.1 : bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.1 :
Packets : Sent = 4, received = 4, lost = 0 (0% loss),
Approximate round trip times in ms :
Min = 0ms, Max = 17ms, Avg = 4ms

Pc > telnet 10.0.0.1
trying 10.0.0.1 ... Open

User Access Verification

Password :
Password :
R1 > enable
Password :
R1 # show ip route
Codes : C - connected, S - static, I - IGRP, R - RIP, M - mobile
B - BGP, D - EIGRP, EX - EIGRP external, O - OSPF,
IA - OSPF inter area, N1 - OSPF NSSA external
type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external
type 2 . E1 - OS
i - IS-IS, L1 - IS-IS level 1, L2 - IS-IS level 2



Write a program for congestion control using Leaky bucket algorithm.

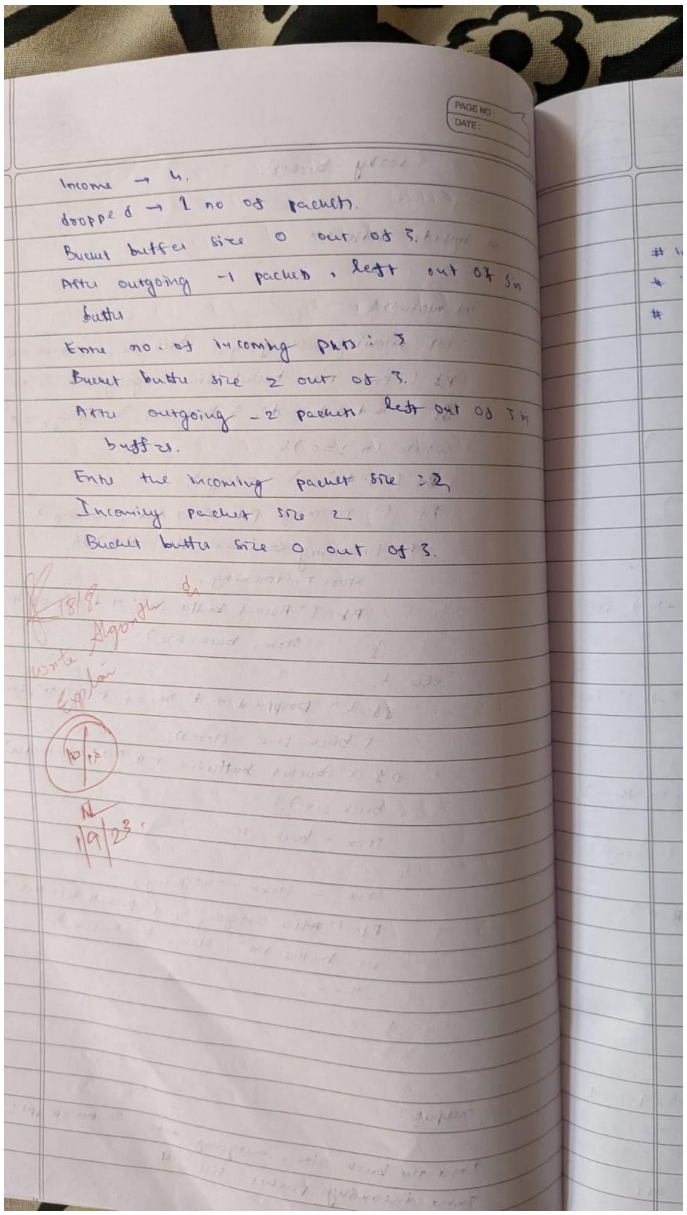
PAGE NO :
DATE :

leaky - bucket

```
#include <stdio.h>

int main(){
    int incoming, outgoing, buck_size, n, store=0;
    pf("Enter the bucket size, outgoing rate & no of ips:");
    sf("%d %d %d", &buck_size, &outgoing, &n);
    while(n!=0){
        sf("%d", &incoming);
        pf("incoming pkt size %d\n", incoming);
        if(incoming <= (bucket_size - store)){
            store += incoming;
            pf("Bucket buffer size: %d out of %d\n",
        }           store, buck_size);
    else{
        pf("Dropped %d no. of packets\n", incoming -
            (buck_size - store));
        pf("bucket buffer size %d out of %d\n", store,
            buck_size);
        store = buck_size;
    }
    store = store - outgoing;
    pf("After outgoing %d packets left out of %d
    in buffer\n", store, buck_size);
        n--;
    }
}
```

output :

Enter the buck size, outgoing rate & no. of ips : 5 4 4
Enter incoming packet size : 4

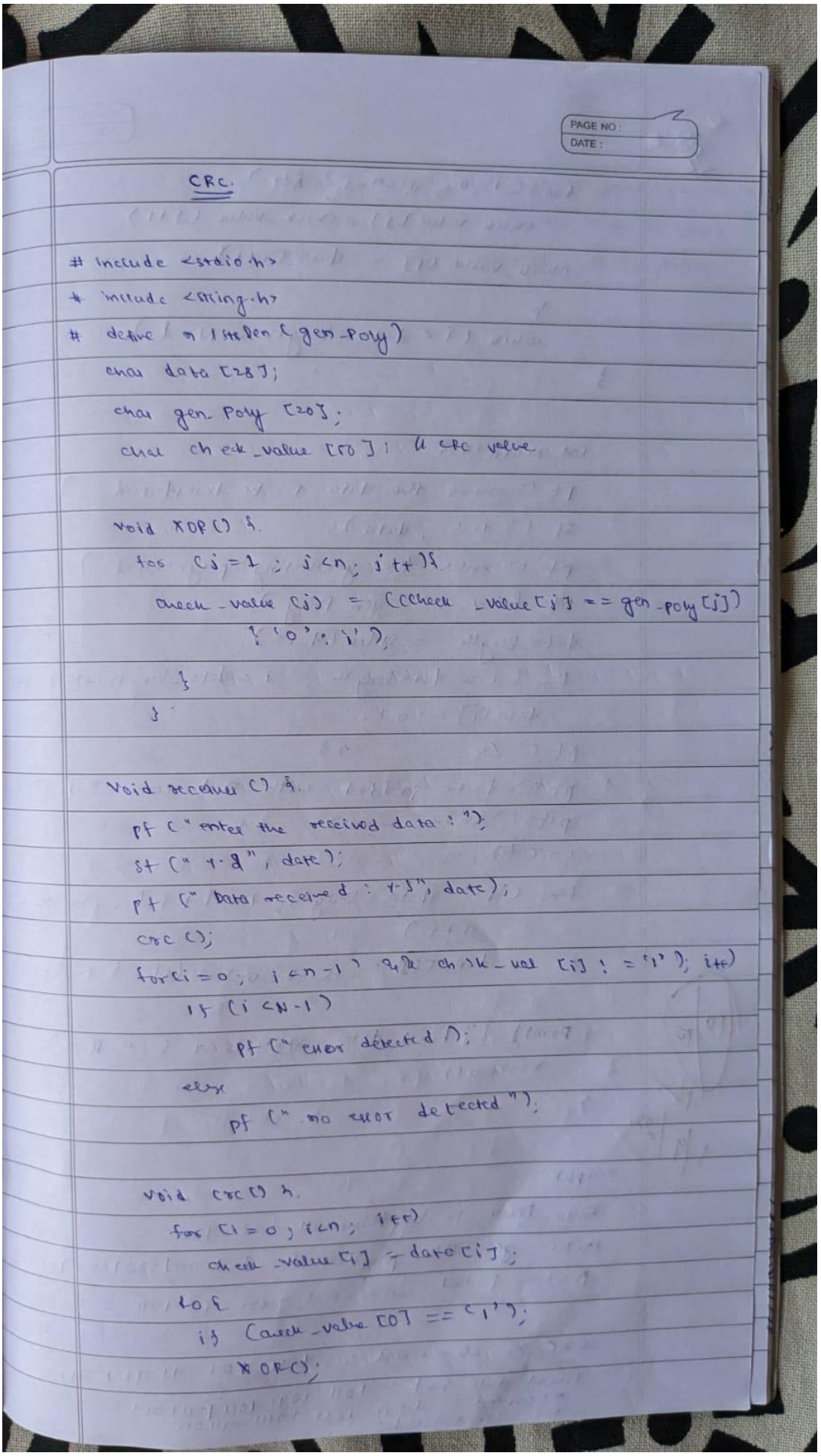


Output:

```
Enter bucket size, outgoing rate and no of inputs: 10 10 2
Enter the incoming packet size : 30
Incoming packet size 30
Dropped 20 no of packets
Bucket buffer size 0 out of 10
After outgoing 0 packets left out of 10 in buffer
Enter the incoming packet size : 10
Incoming packet size 10
Bucket buffer size 10 out of 10
After outgoing 0 packets left out of 10 in buffer
```

Experiment 13

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



PAGE NO :
DATE :

CRC.

```
# include <stdio.h>
# include <string.h>
# define n strlen(gen_poly)
  char data[28];
  char gen_poly[20];
  char check_value[20]; // crc value

  void XOR() {
    for(j=1; j<n; j++){
      check_value[j] = ((check_value[j] == gen_poly[j])
          ? '0' : '1');
    }
  }

  Void receiver() {
    pf("enter the received data : ");
    sf("%s", data);
    pf("Data received : %s", data);
    crc();
    for(i=0; i<n-1) && (check_val[i] != '1'); i++)
      if (i<N-1)
        pf("error detected");
      else
        pf("no error detected");

  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() {
pf(" enter the data to be transfused : ");
sf("%s", data);
pf(" enter the generating polynomial ");
sf("%s", gen_Poly);
data_length = strlen(data);
for (i = data length ; i < data_length + N-1; i++)
    data[i] = '0';
pf("\n ---- ");
pf(" data padded with n-1 zeros %s\n", data);
pf("\n ---- ");
crc();
for (i = data_length; i < data_length + N-1;
        i++)
    data[i] = check_value(i - data_length);

Printf(" Final data sent is %s ", Data);
receiver();
}
```

output :

Enter Data to be transmitted : 1011

Enter generator polynomial : 10001000100010001

Data padded with n-1 zeros is : 1011 000000000 000 00000

crc or check val : 1011 101 11011 1011

final data sent : 1011 1011 1011 10111011

received : 1011 1011 1011 10111011

No error detected

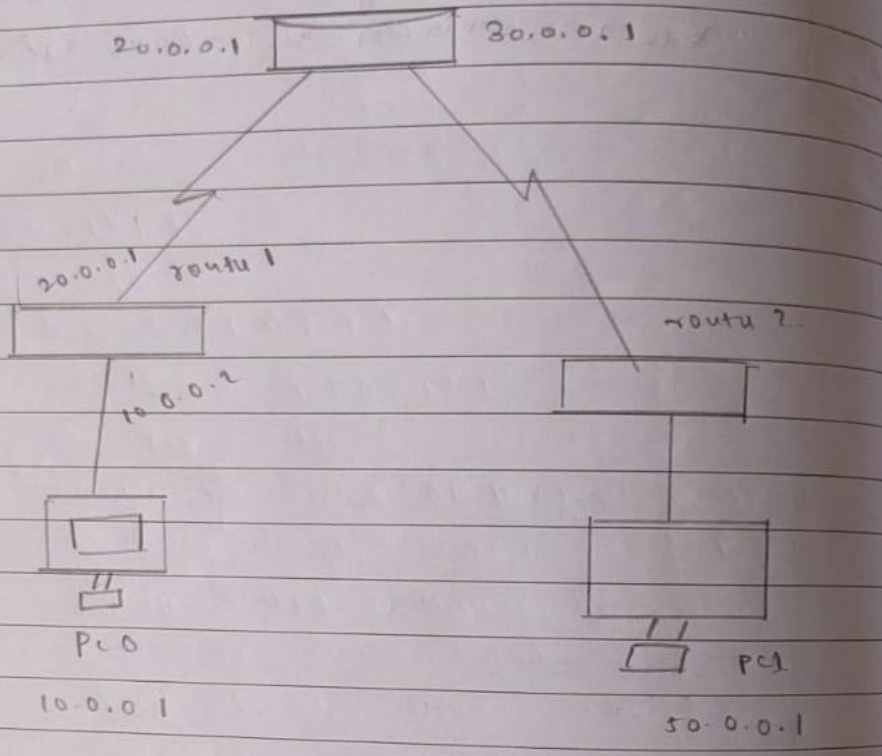
```
Enter the frame bits:1011
Message after appending 16 zeros:1011000000000000000
generator:10001000000100001


quotient:1011
transmitted frame:1011011000101101011
Enter transmitted freme:1011011000101101011
CRC checking


last remainder:0000000000000000

Received freme is correct
```

Experiment 7

Demonstrate the TTL/ Life of a Packet

PAGE NO :
DATE :

Q. Demonstration of TTL.



Procedure :

→ Select 2 PC's and connect them to different routers connect the two routers to another router
→ Enter the IP address to each PC and also set the respective gateway.
→ Also set the dynamic and all the static routing for all PCs
→ Now select a single PDU from one PC & send it while the PDU is moving click on the capture first and check on the packet and a window pops up
→ Btw every router there will be a time difference of 1 in TTL

PAGE NO:
DATE:

output :

1)

0	4	8	16	19	31
4	IHL	DSCP			
FD : 0x6			0x		0x0
TTL : 225		PRO : 0x1		CHKSUM.	
SRC IP : 10.0.0.1					
DST IP : 50.0.0.1					
OPT : 0x0					0x0
Data (variable length)					

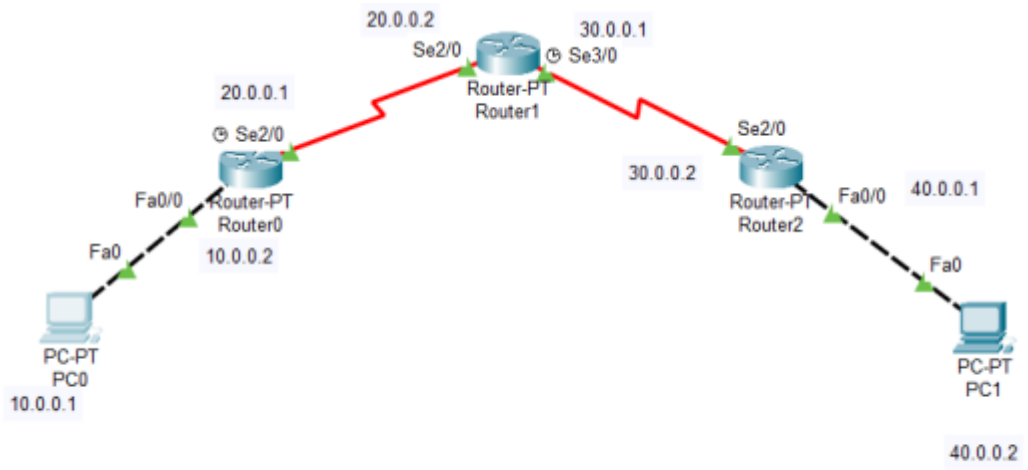
0	4	8	16	19
4	IHL	DSCP	TL : 28	
ID : 0x6		0x	0x0	
TTL : 254		PRO : 0x1	CHKSUM	
SRC IP : 10.0.0.1				
DST IP : 50.0.0.1				
OPT : 0x0				0x0
Data (variable length).				

8/10

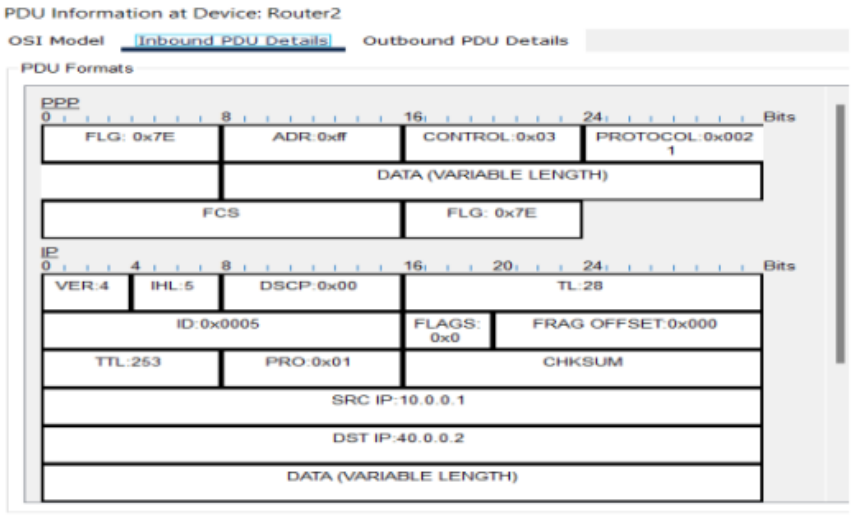
N
1/9/23

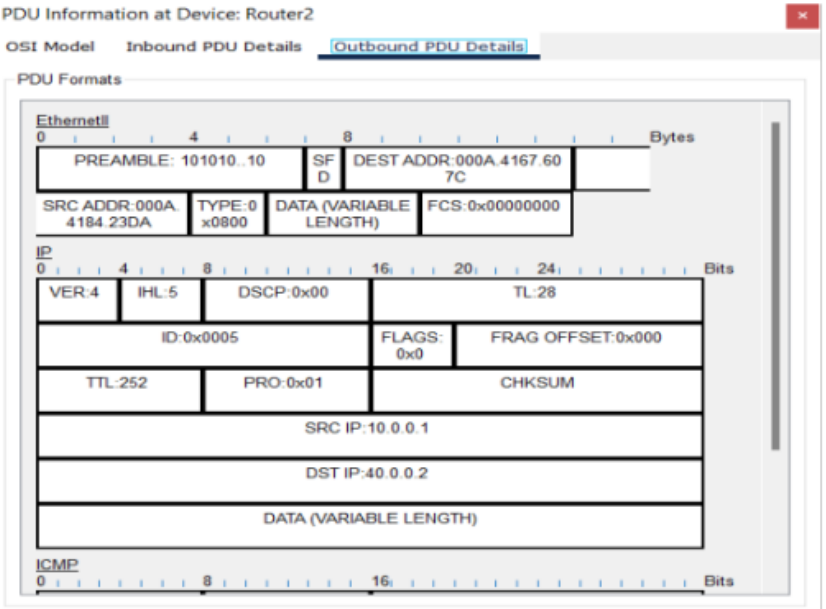
observation :

Output and topology



Output:





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

PAGE NO :
DATE :

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

code:

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())
fileContent = clientSocket.recv(1024).decode()
print("\n From Server: \n")
print(fileContent)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName = "127.0.0.0"
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")
```

10/10

1/9/23

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

- bind() → method bind it to specific IP & port so that it can listen to incoming req on the IP & port.
- listen() → A server has listen() method which puts it in listen mode for incoming connections.
- Accept() & close() → accept() initiate a connection and the close method closes the connection.

Client output

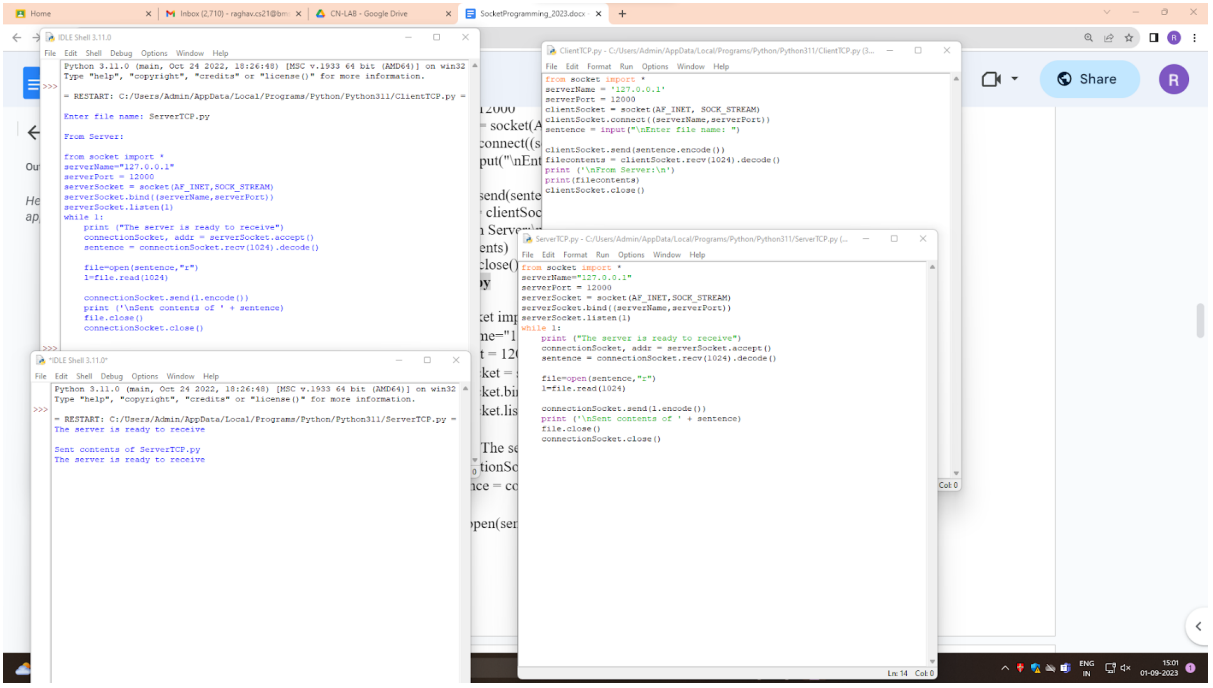
```
--> The server is ready to receive
```

Server output

```
Enter the file name : ServerTCP.py
from Server :

/*
    contents of the file will be
    displayed here.
*/
```

10/10

1/9/23



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

PAGE NO :
DATE :

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

Code :

ClientUDP.py

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter file : ")
clientSocket.sendto(bytes(sentence, "utf-8"),
    (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print("\n 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_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("server ready to receive").
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
```

10/10

7/9/23

PAGE NO :
DATE :

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

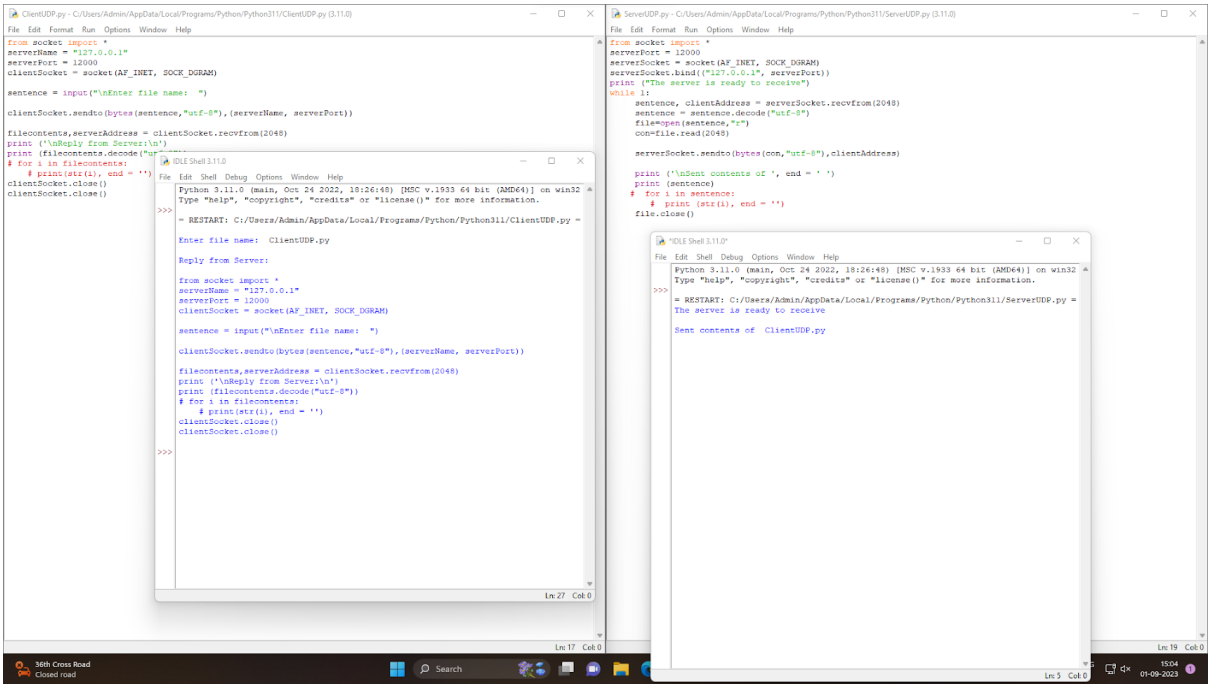
```
The server is ready to receive.

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

Server UDP output

```
Enter the file name: serverUDP.py
reply from server:

/*
    contents of the file are
    displayed here
*/
```

10/10

19/23



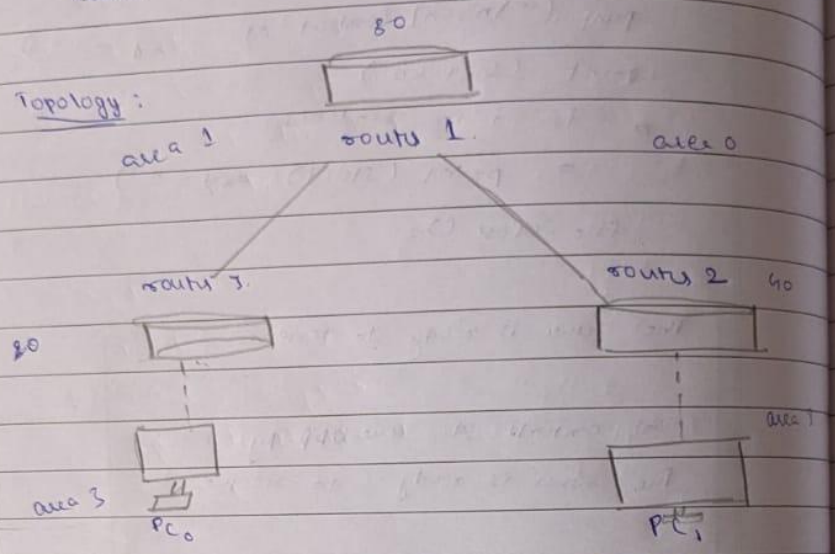
Experiment 6

Configure OSPF routing protocol

PAGE NO :
DATE :

Aim: To configure OSPF routing protocol and connect areas

Topology:



→ Procedure:

1. Create the above topology with 3 routers & 2 PCs and set the address of 2 PC's
2. Now provide the ip addresses of the router interfaces as well as the "encapsulation ppp" command on the router-router interface.
3. for router-0 & router-1 in network 20 & 30 provide clock rate 64000.
4. for each router, enable ip routing by configuring OSPF in all routers & providing the router id.
5. Create dedicated loopback interface for all the routers
6. Create a virtual link to connect area 0 to area 3.

→ CLI commands:

1) configuring ip address to all router-interfaces.

router>1

enable

PAGE NO :
DATE :

config t

interface se 2/0

ip address 20.0.0.2 255.0.0.0

encapsulation PPP

no shut

ip address 30.0.0.1 255.0.0.0

encapsulation PPP :

clock rate 64000

no shut

2) enabling routing by configuring ospf in all routers.

router 0:

router ospf 1

router-id 1.1.1.1

network 10.0.0.0 0.255.255.255. ~~0.0.0~~ area 3

~~encapsulation PPP~~

network 20.0.0.0 0.0.0 255.255.255 area 1

exit

router 1:

router ospf 1

router id 2.2.2.2

network 10.0.0.0 0.0.0 255.255.255 area 1

network 30.0.0.0 0.255.255.255 area 0

exit

router 2:

router ospf 1

router-id 3.3.3.3

network 50.0.0.0.0 0.255.255.255 area 0

network 40.0.0.0.0 0.255.255.255 area 2

exit

PAGE NO:
DATE:

3. creating dedicated loopback interface for all routers

router 0.

interface se 2/0
interface loopback 0.
ip add 172.16.1.252 255.255.0.0
no shut

router 1.

interface se 2/0
interface loopback 0
ip add 172.16.1.253. 255.255.0.0
no shut.

router 2.

interface se 2/0
interface loopback 0.
ip add 172.16.1.254 255.255.0.0
no shut

4) create virtual link b/w router-0 router-2

router-0

router ospf 1
area 1 virtual-link 2.2.2.2
exit

router-1

router ospf 1
area 1 virtual-link 1.1.1.1
exit

Ping command

PAGE NO :
DATE :

PC> Ping 40.0.0.10

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

PC> Pinging 40.0.0.10 with 32 bytes of data

Reply from 40.0.0.10 bytes = 32 time = 9ms TTL=11

Reply from 40.0.0.10 bytes = 32 time = 9ms TTL=15

Reply from 40.0.0.10 bytes = 32 time = 9ms TTL=11

reply from 40.0.0.10 bytes = 32 time = 9ms TTL=11

Ping Statistics for 40.0.0.10

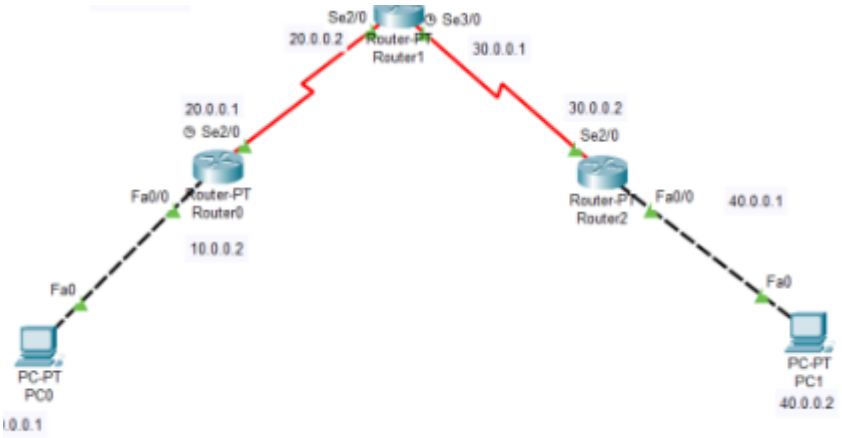
packets : sent = 4, received = 4, lost = 0

Outcome :

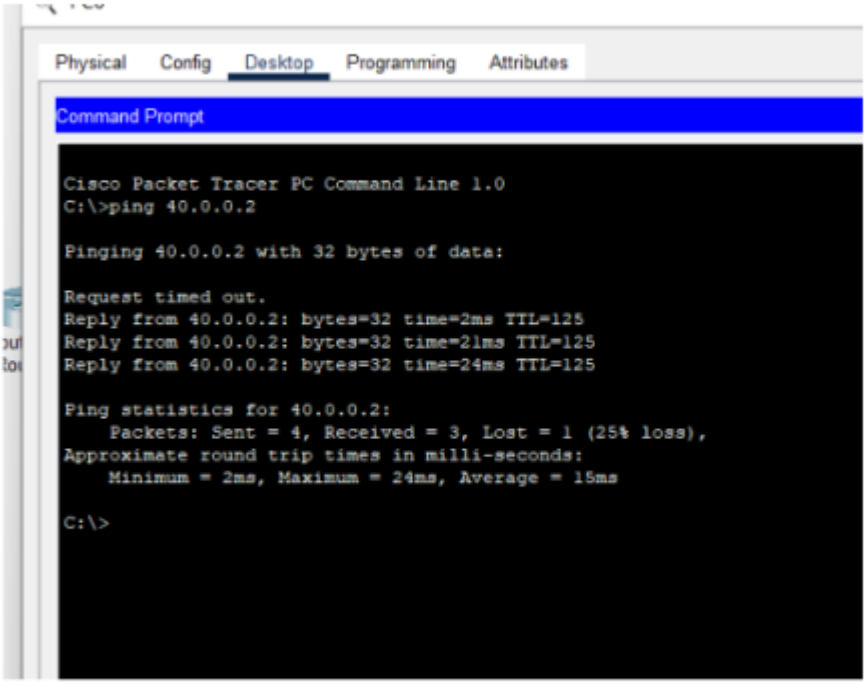
OSPF stands for open shortest path first is a link layer / app layer protocol that is used to find the best route for packet transfer & involves the creation of a virtual link

8/10

N 2/9/23



Output:



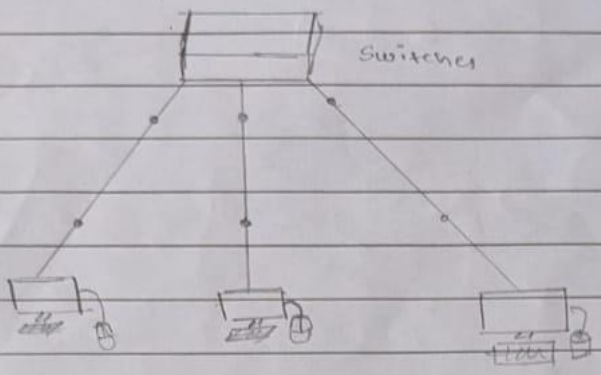
Experiment 9

To construct simple LAN and understand the concept and operation of Address Resolution

Protocol (ARP)

ARP (address resolution Protocol)

Topology:



Procedure:

- arp -a → to chk arp tables (from cmd prompt)
- arp -d → to clear arp.
- click capture (foward

output:

In PC0,

arp -a

No arp entries found

Ping 10.0.0.2.

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2 : bytes =32 time = 12ms TTL=1

Reply from 10.0.0.2 : bytes =32 time = 12ms TTL=

Reply from 10.0.0.2 : bytes =32 time =12ms TTL=

Reply from 10.0.0.2 : bytes =32 time =12ms TTL=

Ping statistics for 10.0.0.2:

Packets sent =4, received =4, lost =0 (0% lo

Approximate round trip time in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Avg = 0ms

PAGE NO :
DATE :

...ing dedicated loopback interface for ea router

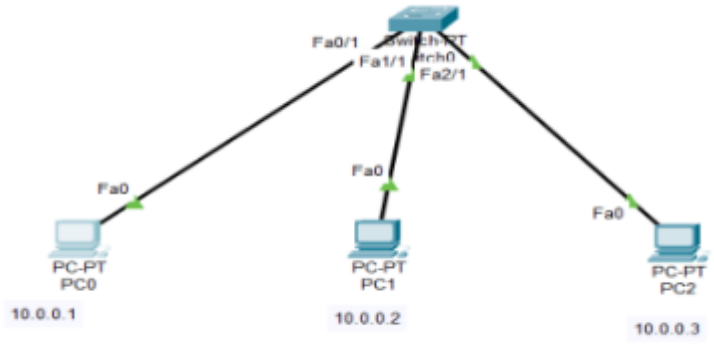
classmate
Date
Page

arp -a

internet Address	Physical address	type
10.0.0.2	0001.4230.e7e9	dynamic

10/10

11/9/23



Output:

PC0

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
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<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
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 = 0ms, Average = 0ms

C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms 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

C:\>arp -a
  Internet Address      Physical Address      Type
  10.0.0.2              0050.0f21.c5d2        dynamic
  10.0.0.3              00d0.d326.7e75        dynamic

C:\>
```