VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Mahantesh V N (1BM23CS175)

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
Academic Year 2024-25 (odd)

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 Network (23CS5PCCON)" carried out by **Mahantesh V N(1BM23CS175),**who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Praveen N Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

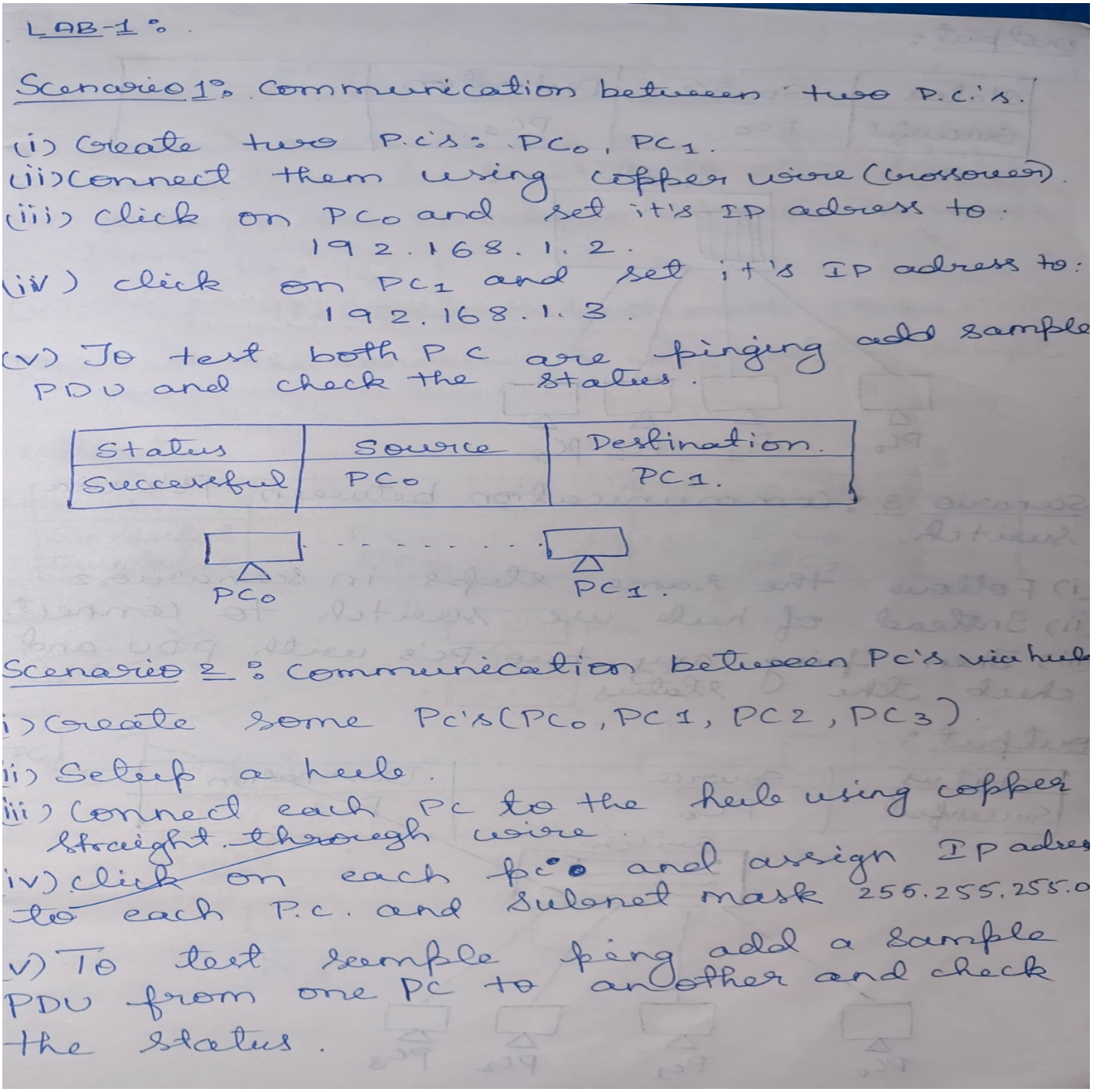
Index

Github: https://github.com/Mahanteshvn/CN_LAB

Program 1

Aim : 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:



LAB-1 :

Scenario 1 : Communication between two P.C.'s.

(i) Create two P.C's : PC_0, PC_1.
(ii) Connect them using copper wire (crossover).
(iii) Click on PC_0 and set it's IP adress to :
192.168.1.2.
(iv) click on PC_1 and set it's IP adress to :
192.168.1.3.
(v) To test both PC are pinging add sample PDU and check the states.

Status	Source	Destination.
Successful	PC_0	PC_1.

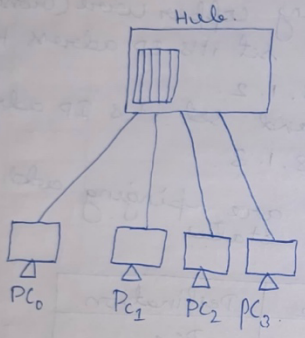
PC_0 PC_1.

Scenario 2 : Communication between Pc's via hub

i) Create some Pc's (PC_0, PC_1, PC_2, PC_3)
ii) Setup a hub.
iii) Connect each Pc to the hub using copper straight through wire.
iv) click on each pc and assign IP adres to each P.C. and Subnet mask 255.255.255.0
v) To test sample ping add a sample PDU from one PC to another and check the status.

output:

Status	Source	Destination	
Successful	PC_0	PC_1.	

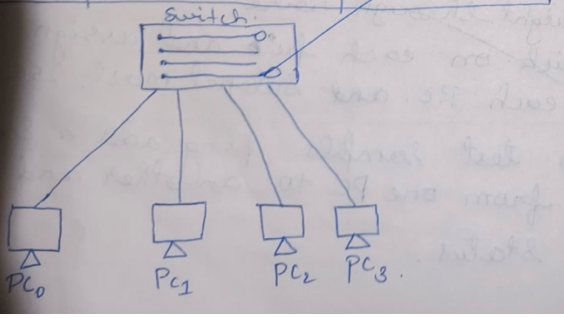


Scenario 3: Communication between Pc's via switch.

(i) Follow the same steps in scenario 2.
(ii) Instead of hub use switch to connect.
(iii) Then ping any two Pc's with PDU and check the status.

output:

Status	Source	Destination
Successful	PC_0	PC_1.

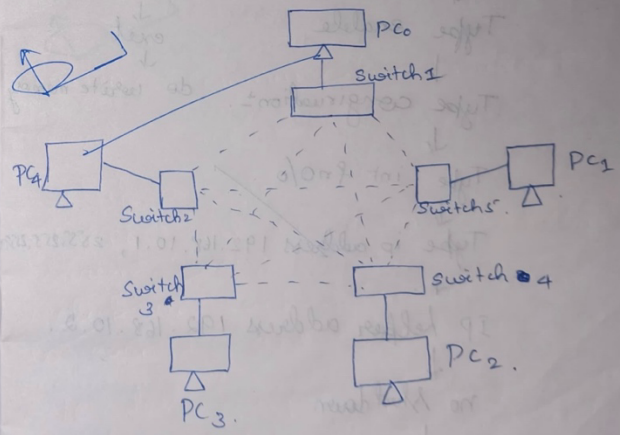


Scenario 4: Mesh network using switch.

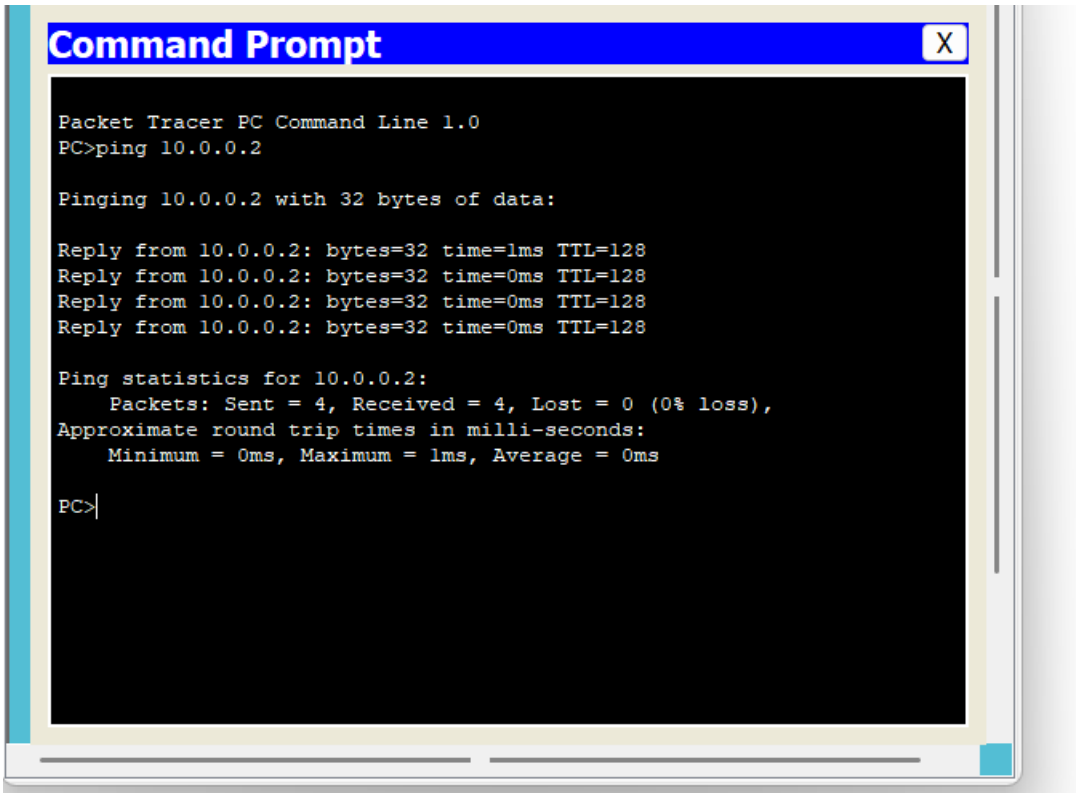
(i) setup Pc's in a mesh fatteen, for each Pc setup an unique switch.
(ii) Connect each Pc to an unique switch.
(iii) Connect each switch to other switch.
(iv) setup the IP address of each Pc where from: 192.168.1.2
(v) Setup the Subnet mask to 255.255.255.0
(vi) Test the connectivity by adding simple PDU.

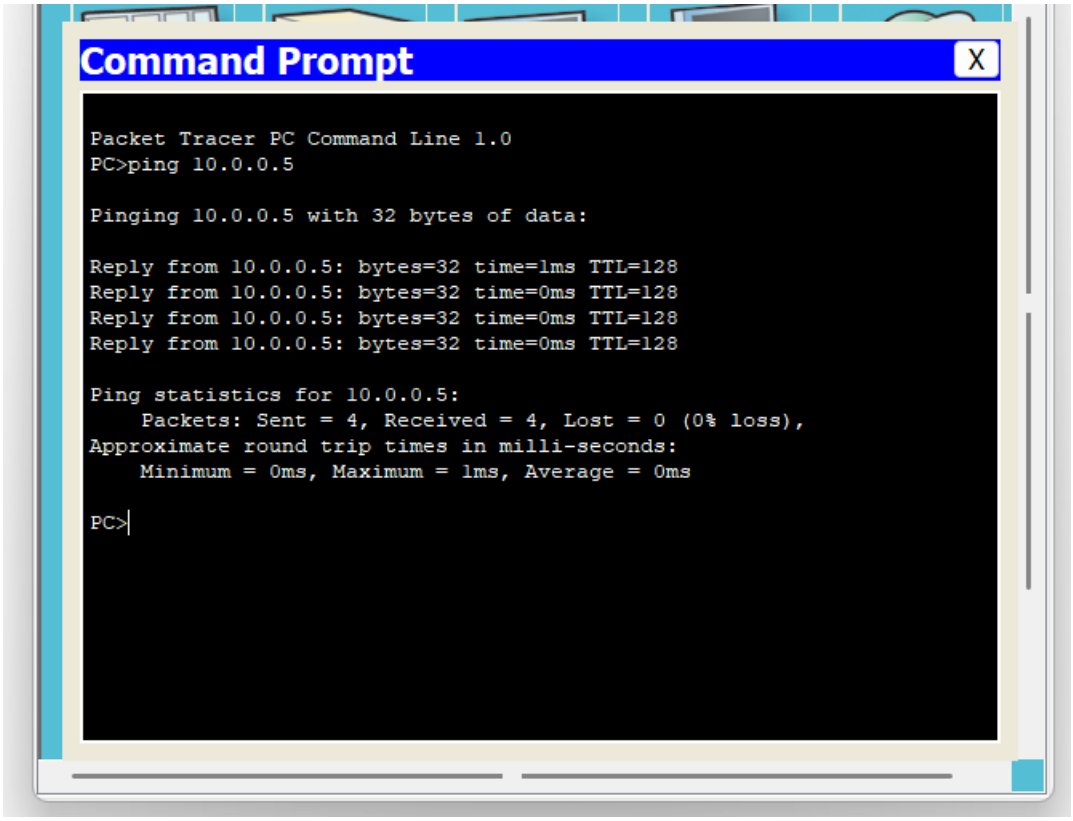
output:

Status	Source	Destination
Successful	PC_0	PC_4.
Successful	PC_1	PC_2.



Output:

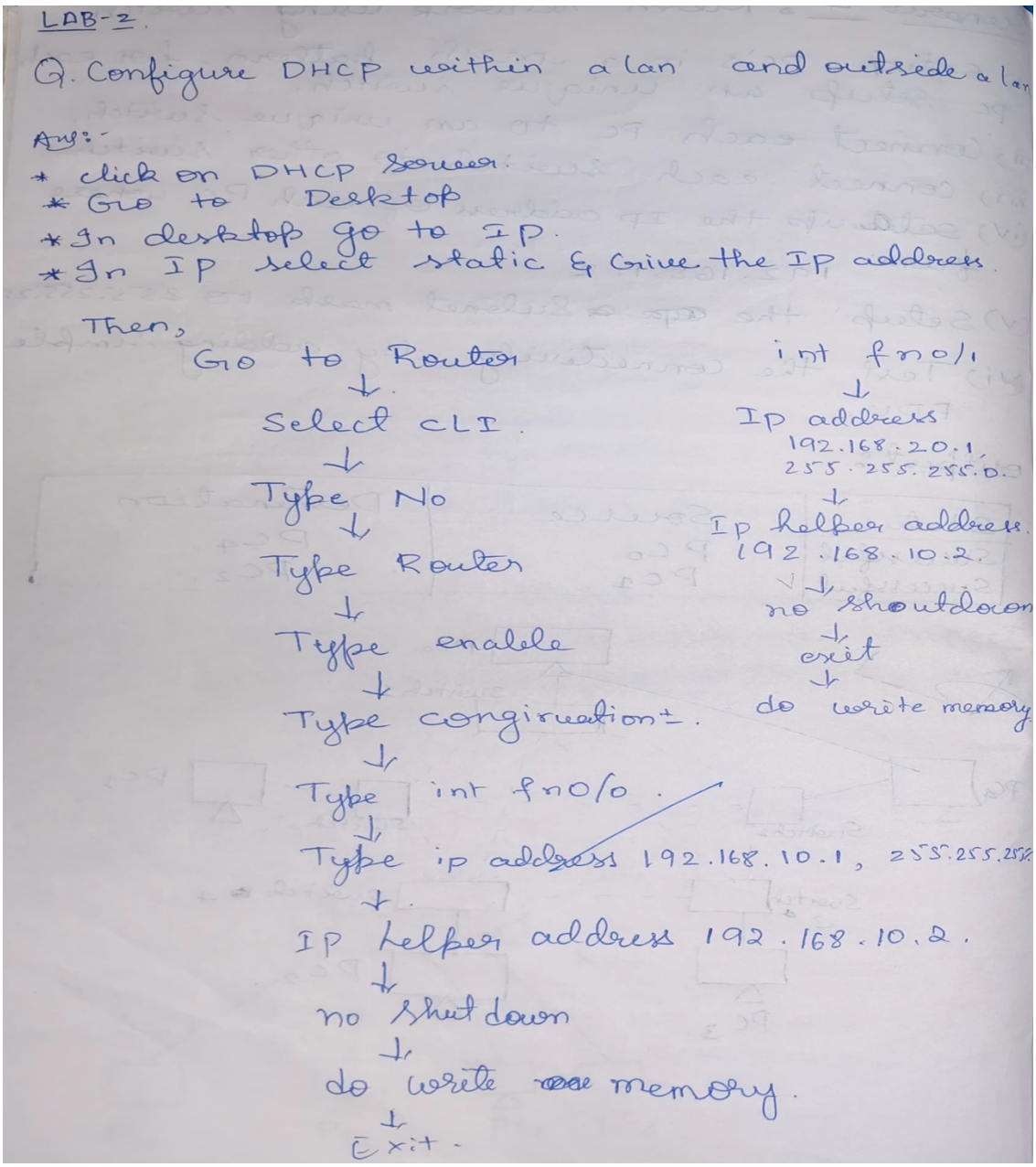

Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=1ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

Program 2

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

Observation:



DHCP.

↳ Desktop IP configuration.

↳ Static.

↳ 192.168.10.2.

↳ 192.168.10.1.

Gateway.

*Server :.

•Services :.

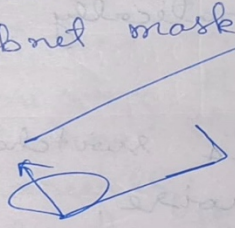
DHCP

Pool name

Gateway

Start IP

Subnet mask



ON

Switch one

192.168.10.1

192.168.10.3

255.255.255.0

ON.

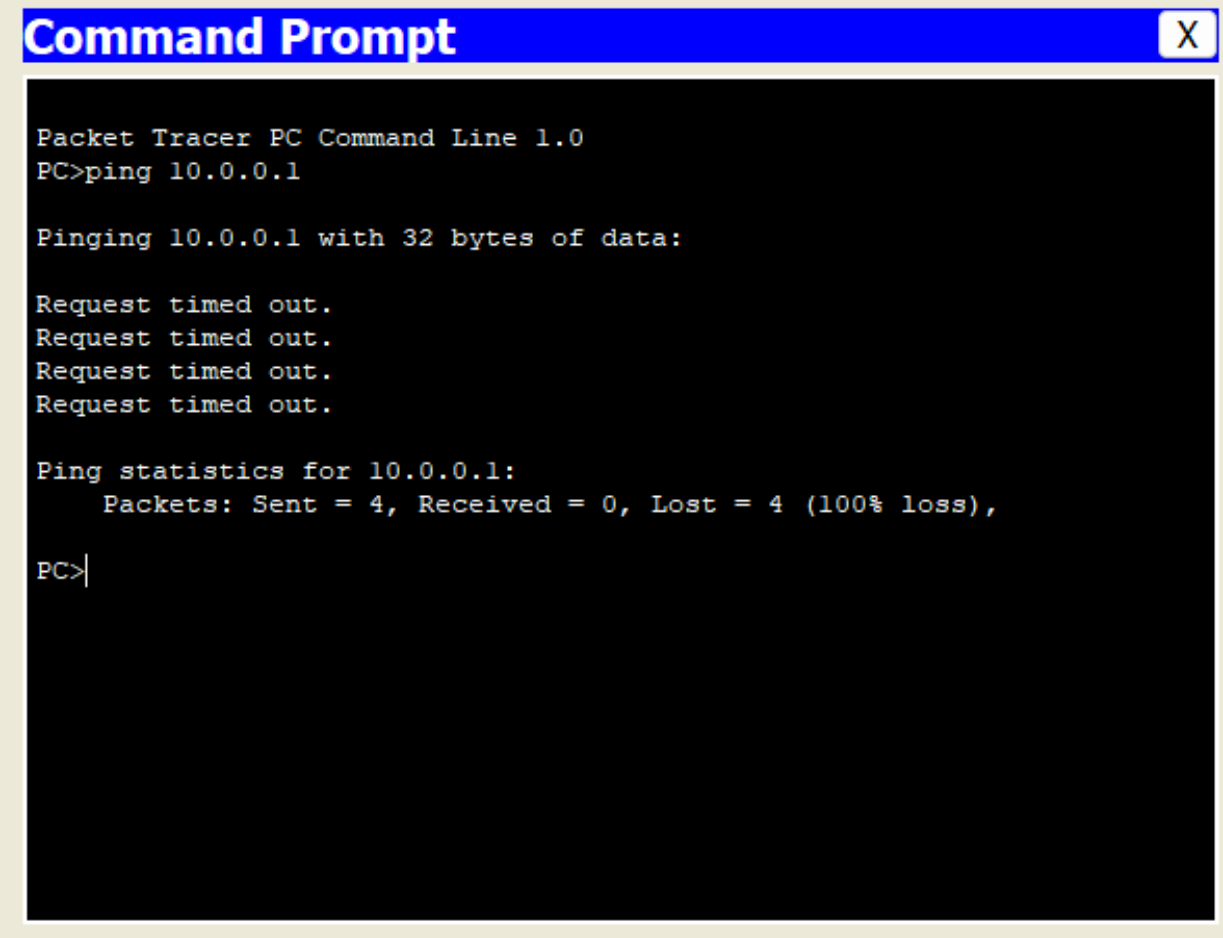
Switch Two.

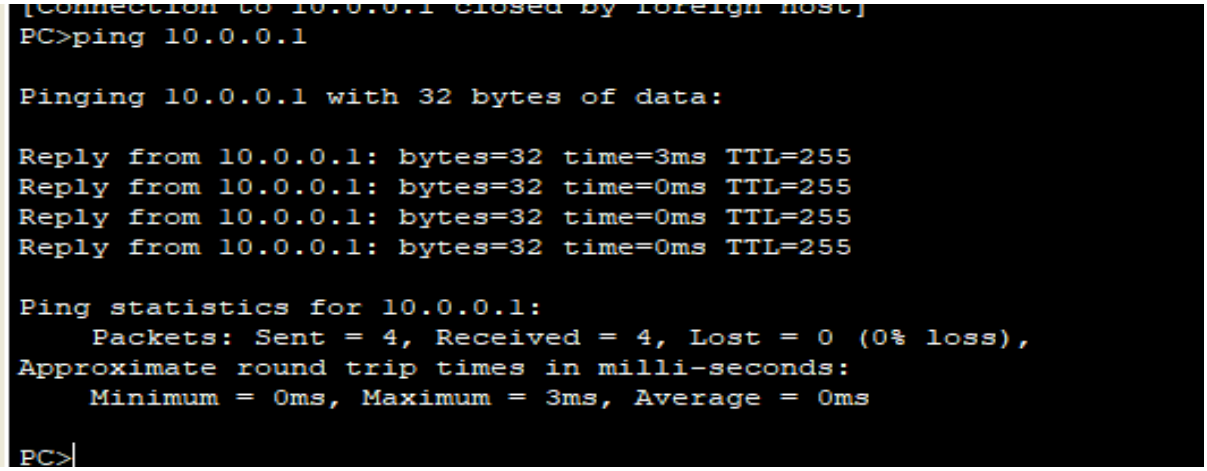
192.168.20.1

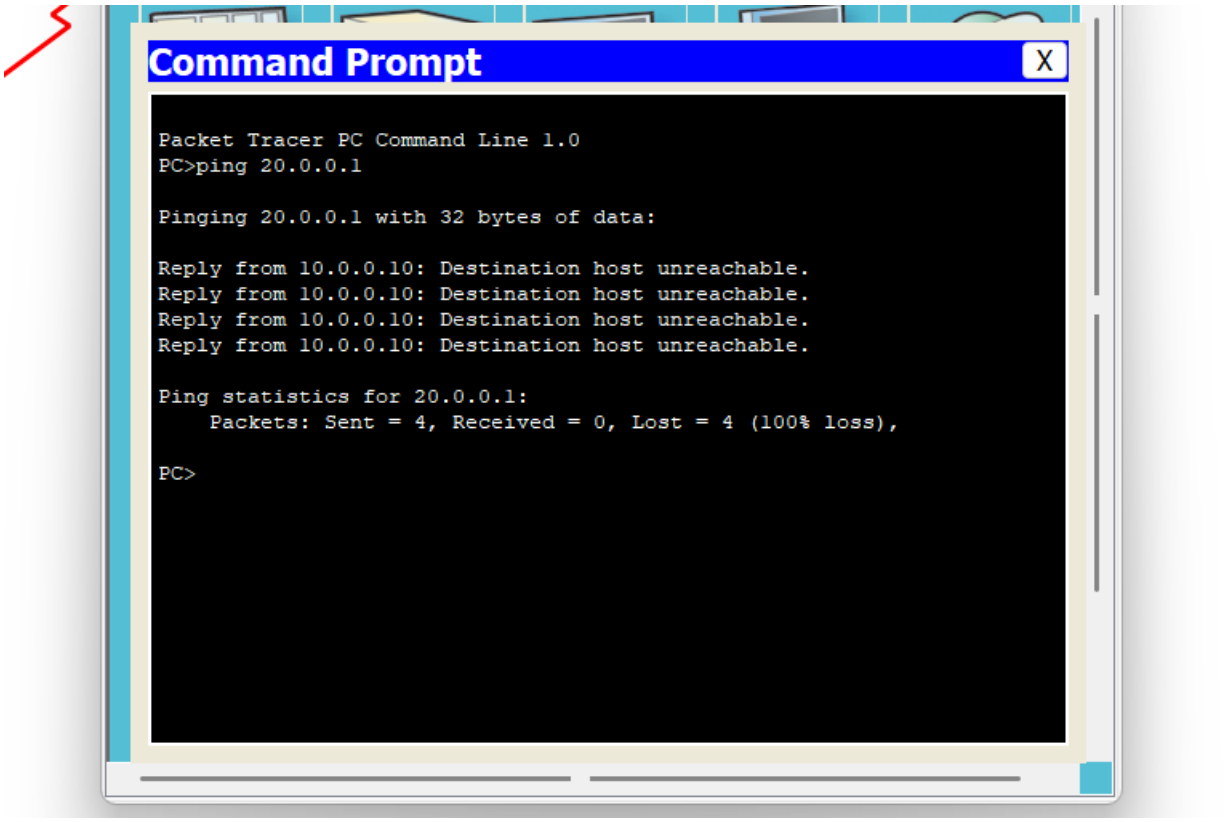
192.168.20.2

255.255.255.0

Output:


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>

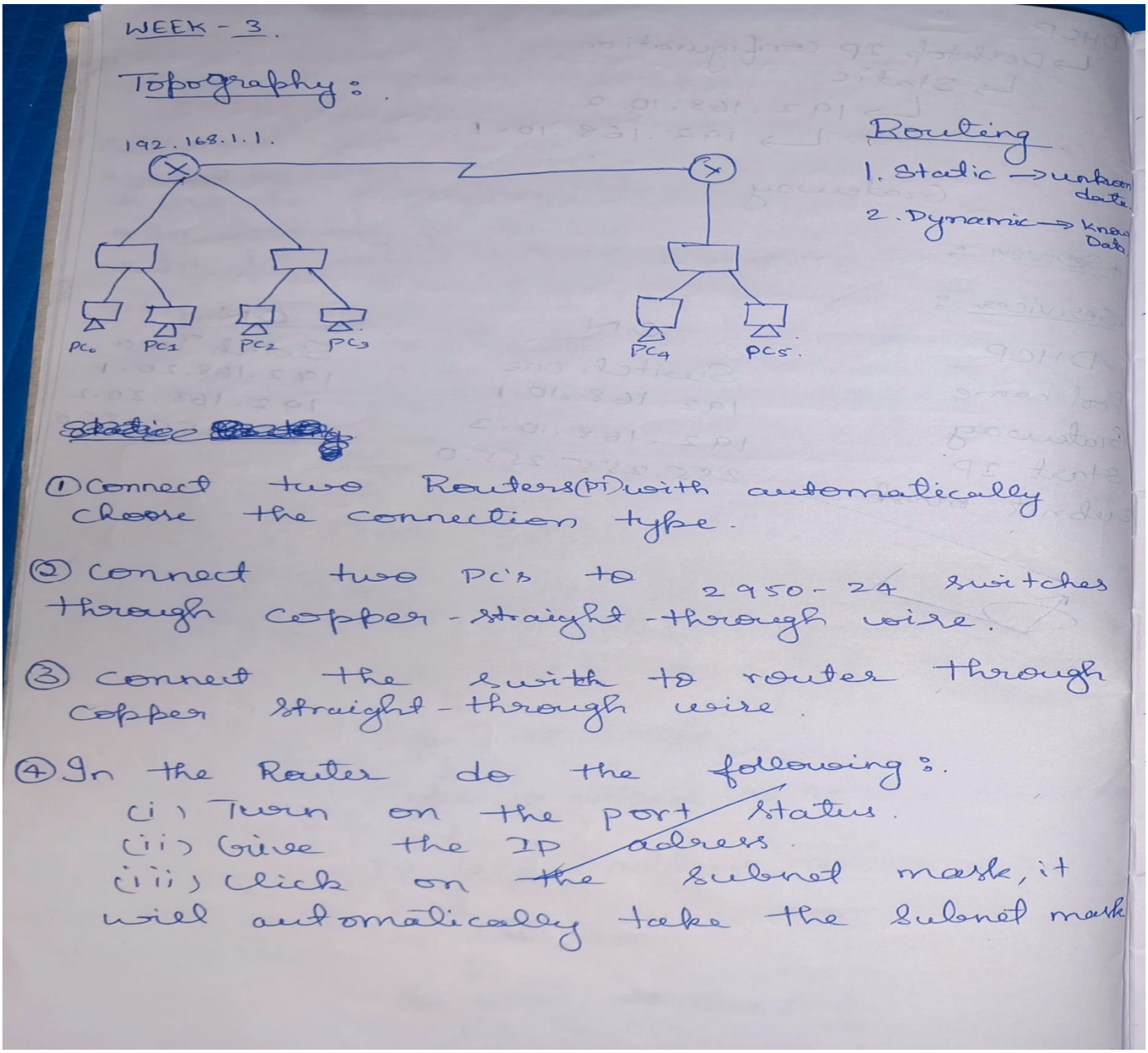

PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=3ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 0ms
PC>


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>

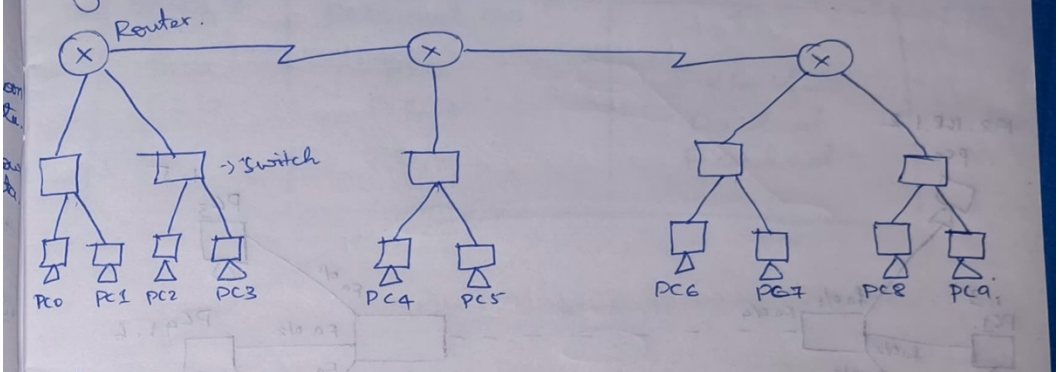
Program 3

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

Observation:



When we try to ping PC_0 to PC_4 it is not pinging due to Static Routing we can solve it by:



~~The router~~

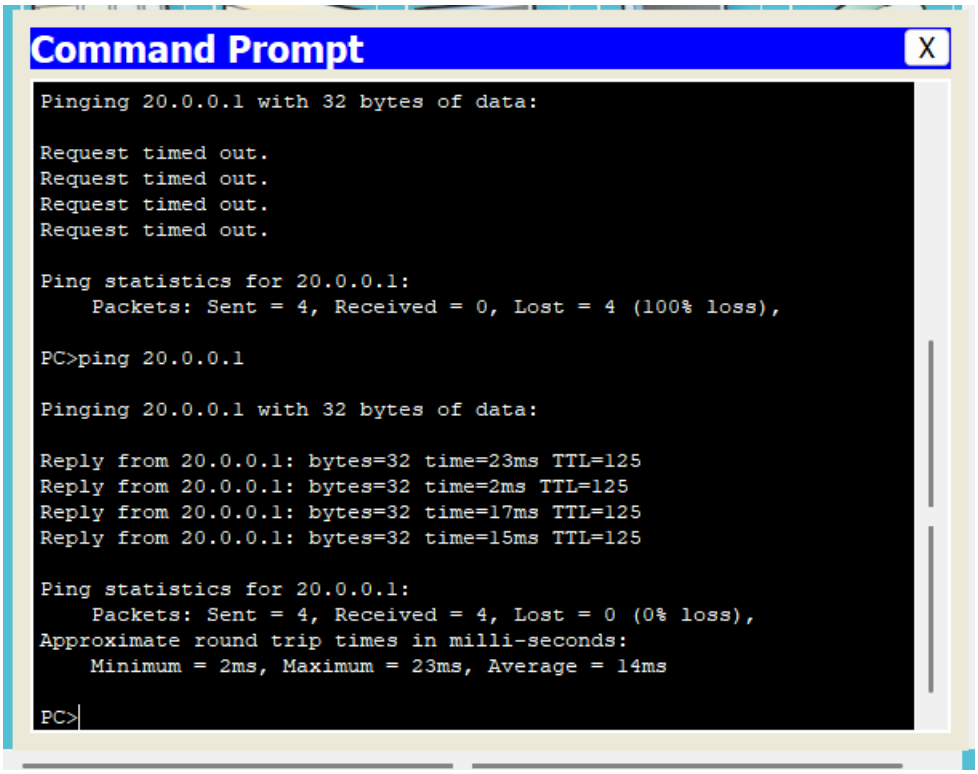
Now if we try to ping any PC from any pc it's working due to Static Routing. It is done by going to Router, Selecting the Static, then, Giving the network adress, Giving the mask adress as 255.255.255.0 manually and for the new connection repeat the above steps.

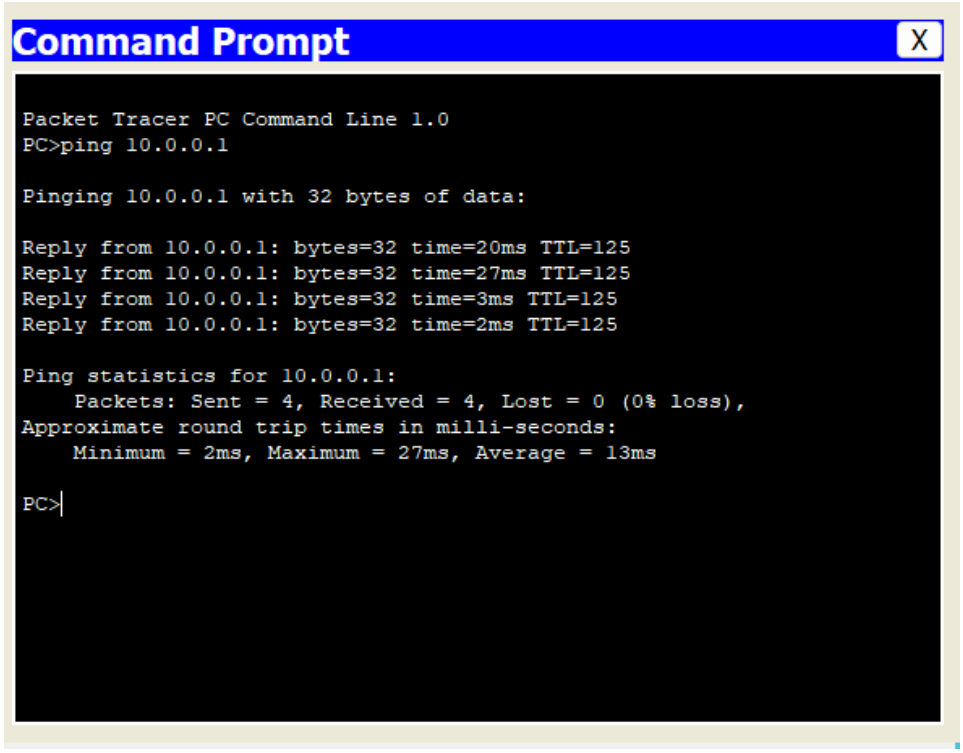
OUTPUT:

Source	Destination	Status
PC0	PC5	Successfull.

~~Dynamic Routing~~

Output:


Command Prompt
X
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 20.0.0.1:
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=23ms TTL=125
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125
Reply from 20.0.0.1: bytes=32 time=17ms TTL=125
Reply from 20.0.0.1: bytes=32 time=15ms TTL=125
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 23ms, Average = 14ms
PC>

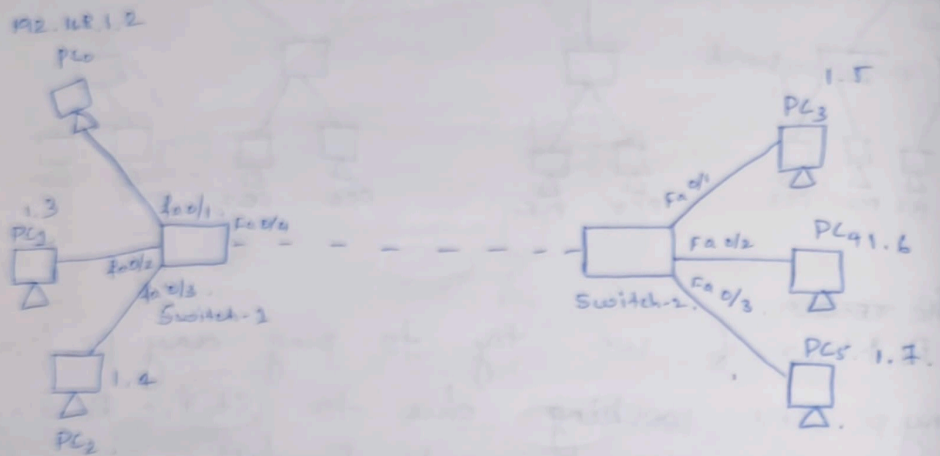

Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=20ms TTL=125
Reply from 10.0.0.1: bytes=32 time=27ms TTL=125
Reply from 10.0.0.1: bytes=32 time=3ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 27ms, Average = 13ms
PC>

Program 4

Aim : Configure DHCP within a LAN and outside LAN.

Observation:

WEEK-4



Switchport mode trounk.

fa0/1	PC0	fa 0/2 PC1	fa 0/3 P2.
fa0/a1	PC3	fa 0/2 PC4	fa 0/3 PC5.

(i) Go to switch 1 then select CLI.

then,

(ii) Switch > enable.

(iii) switch # config t

(iv) Switch (config)# int fa 0/1.
Switchport access vlan 10.

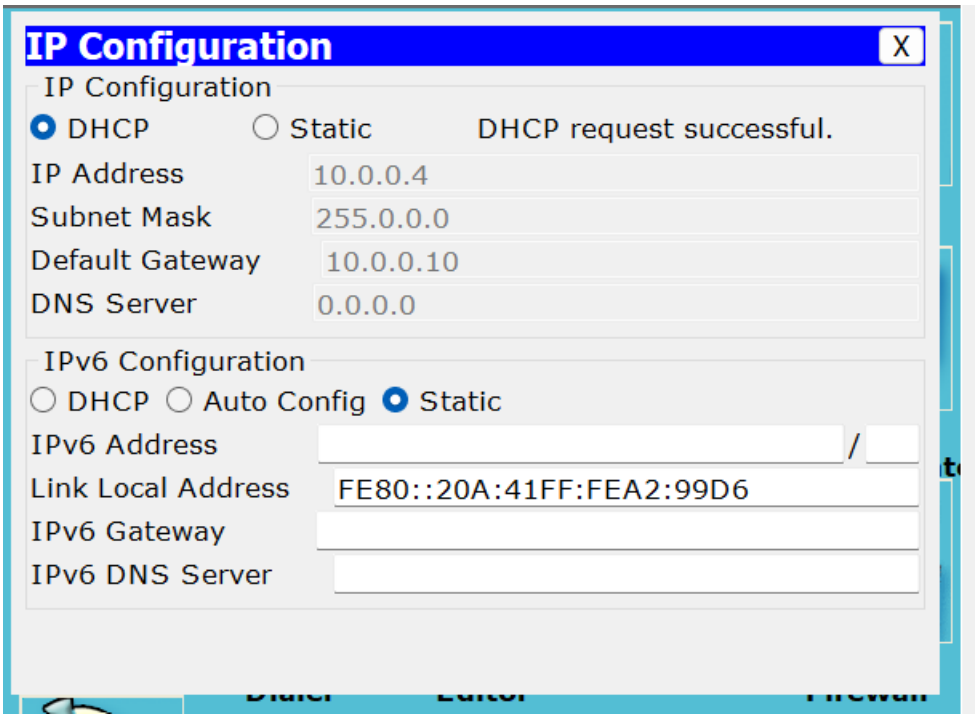
int fa 0/3
Switchport acess vlan 30

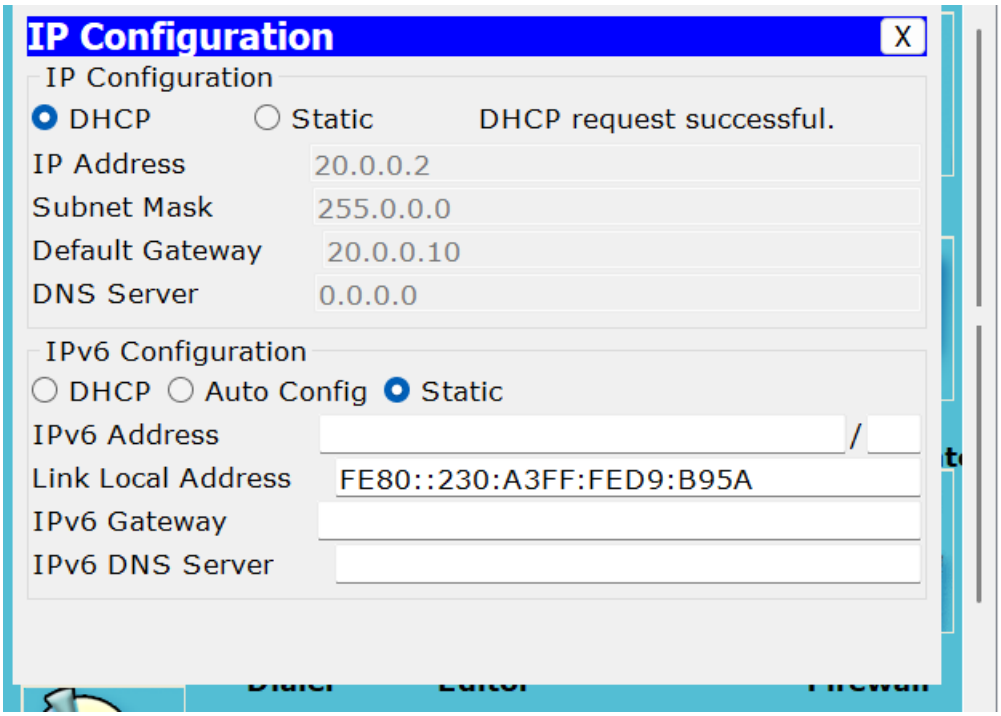
) Repeat the same for switch 2 then, the output will be.

UTPUT:

Source	Destination	Status
PC0	PC1	Successful
PC0	PC4	Failed
PC0	PC5	Failed

Output:


IP Configuration
X
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
10.0.0.4
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.10
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
/
Link Local Address
FE80::20A:41FF:FEA2:99D6
IPv6 Gateway
IPv6 DNS Server

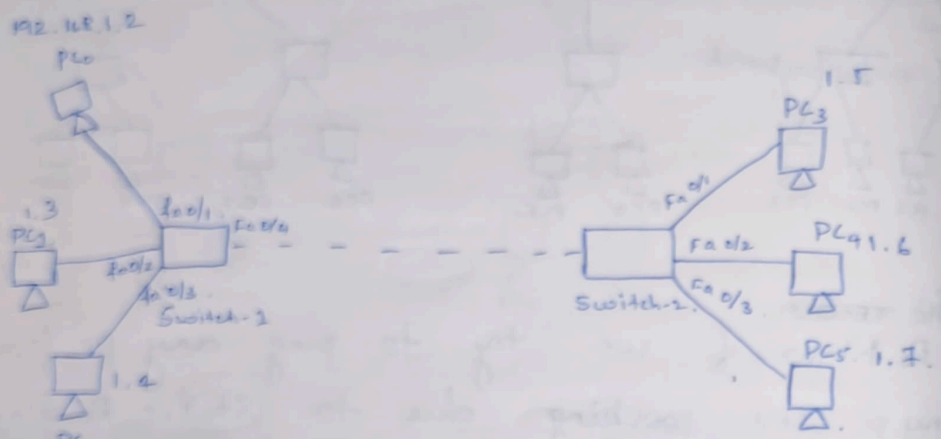

IP Configuration
X
IP Configuration
DHCP
Static
DHCP request successful.
IP Address
20.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.10
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
/
Link Local Address
FE80::230:A3FF:FED9:B95A
IPv6 Gateway
IPv6 DNS Server

Program 5

Aim : Configure RIP routing Protocol in Routers

Observations :

WEEK-4



Switchport mode trounk.

fa0/1	PC0	fa 0/2 PC1	fa 0/3 P2.
fa0/a1	PC3	fa 0/2 PC4	fa 0/3 PC5.

(i) Go to switch 1 then select CLI.

then,

(ii) Switch > enable.

(iii) Switch # config t

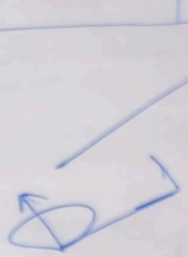
(iv) Switch (config)# int fa 0/1.
Switchport access vlan 10.

int fa 0/3
Switchport acess vlan 30

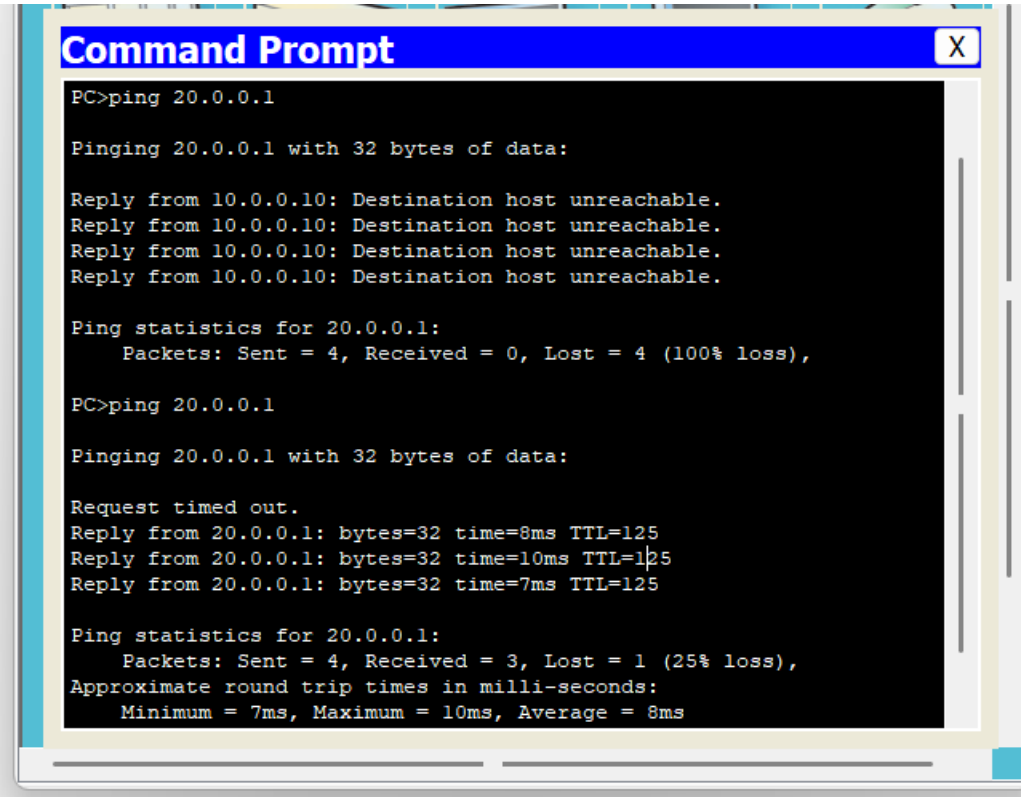
) Repeat the same for switch 2 then, the output will be

UTPUT:

Source	Destination	Status
PC0	PC1	Successful
PC0	PC4	Failed
PC0	PC5	Failed



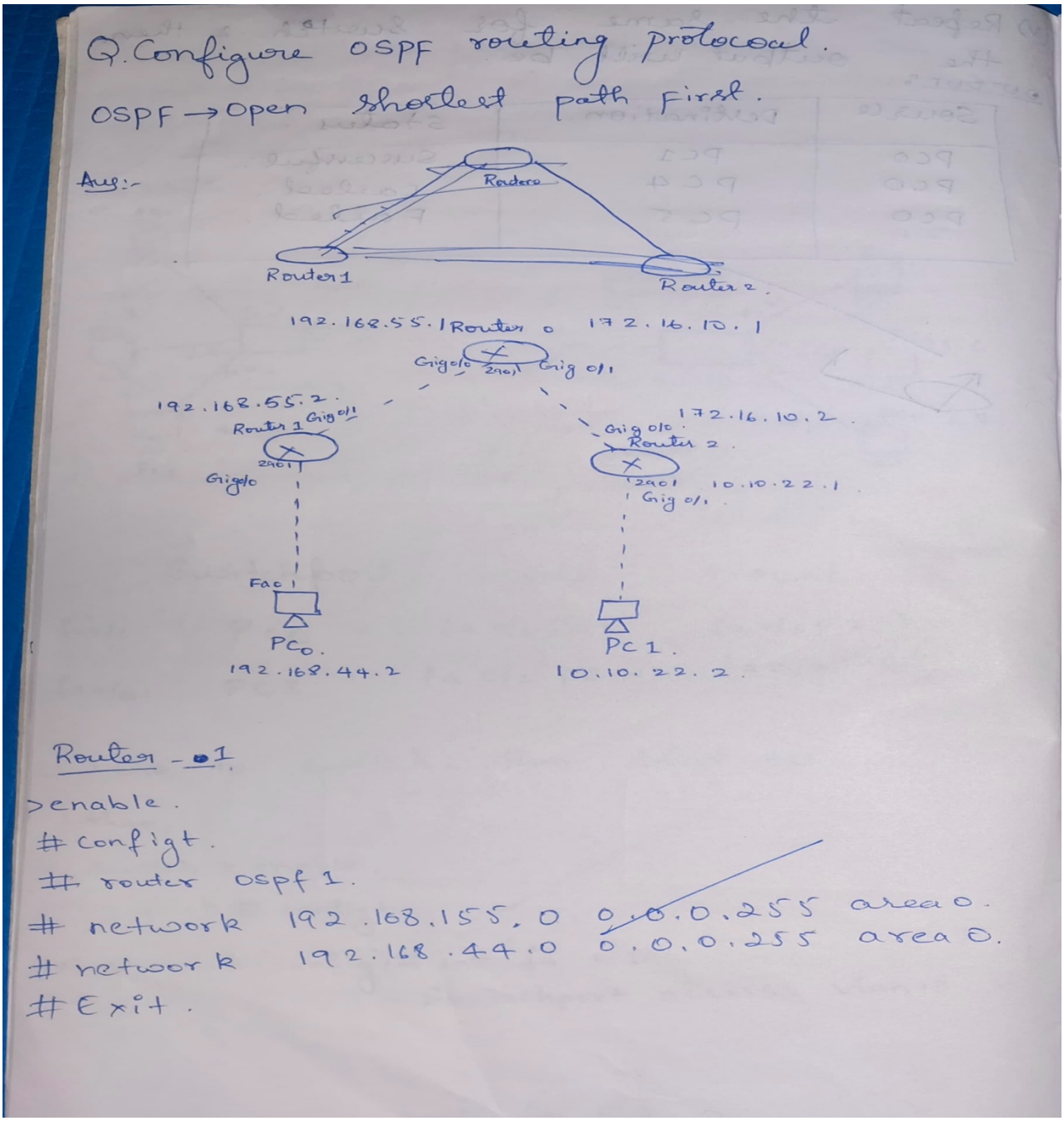
Output:


Command Prompt
X
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Ping statistics for 20.0.0.1:
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:
Request timed out.
Reply from 20.0.0.1: bytes=32 time=8ms TTL=125
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=7ms TTL=125
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 10ms, Average = 8ms

Program 6

Aim : Configure OSPF routing Protocol

Observations:



Router - 0 :-

router ospf 1.

network 192.168.55.0 0.0.0.255.

network 172.16.0.0 0.0.255.255.

exit.

Router - 2 :-

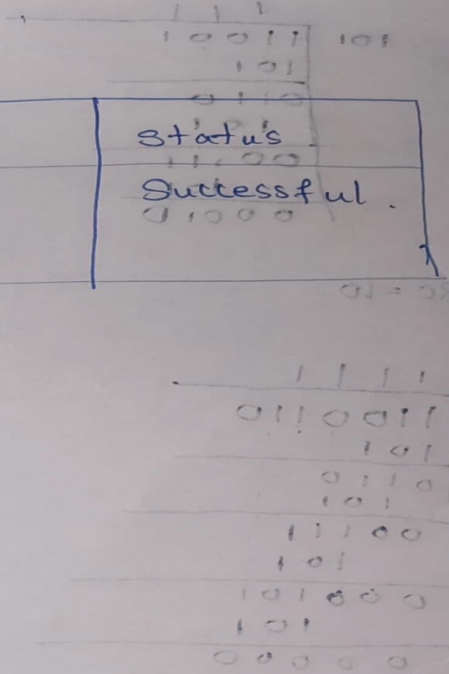
router ospf 1.

network 172.16.0.0 0.0.255.255 area 0.

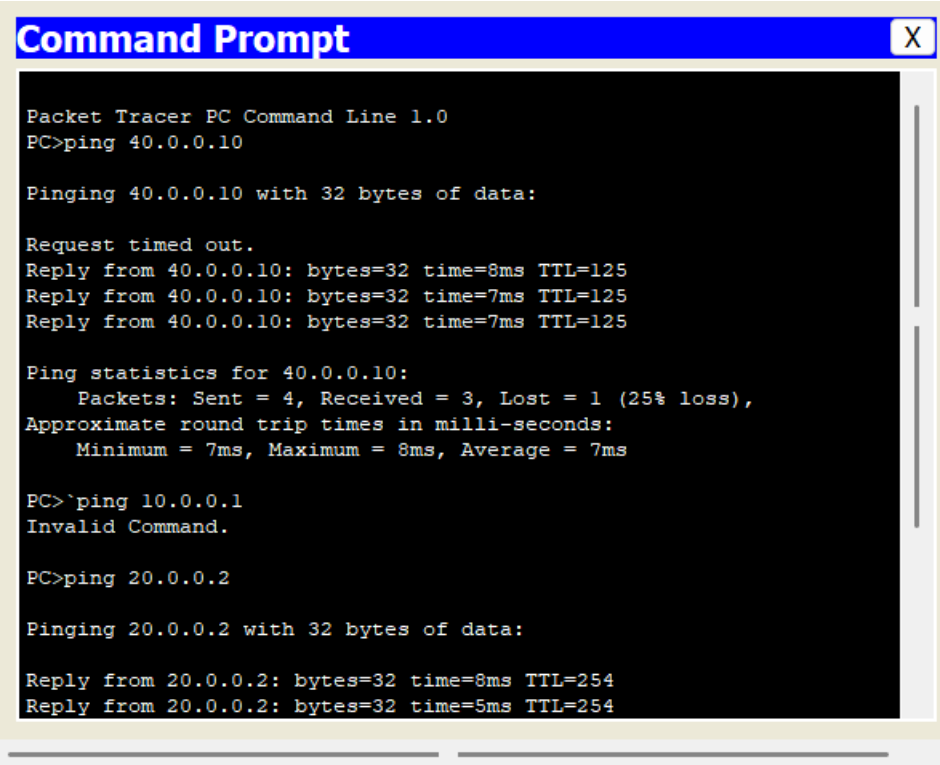
network 10.0.0.0 0.255.255.255 area 0.

OUTPUT :-

Source	Destination	Status
PC0	PC1	Successful.



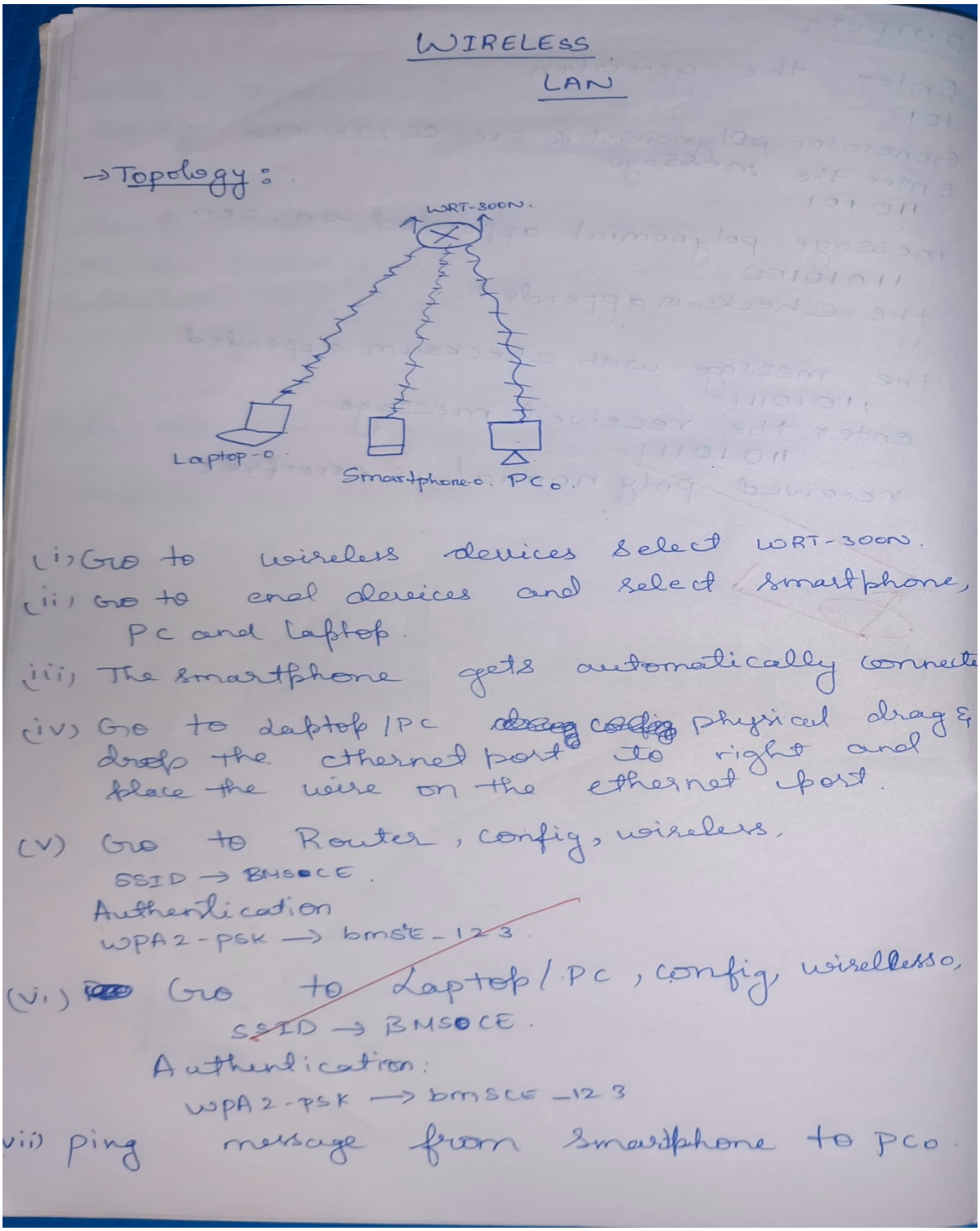
Output:


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:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms 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 = 7ms, Maximum = 8ms, Average = 7ms
PC>`ping 10.0.0.1
Invalid Command.
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time=8ms TTL=254
Reply from 20.0.0.2: bytes=32 time=5ms TTL=254

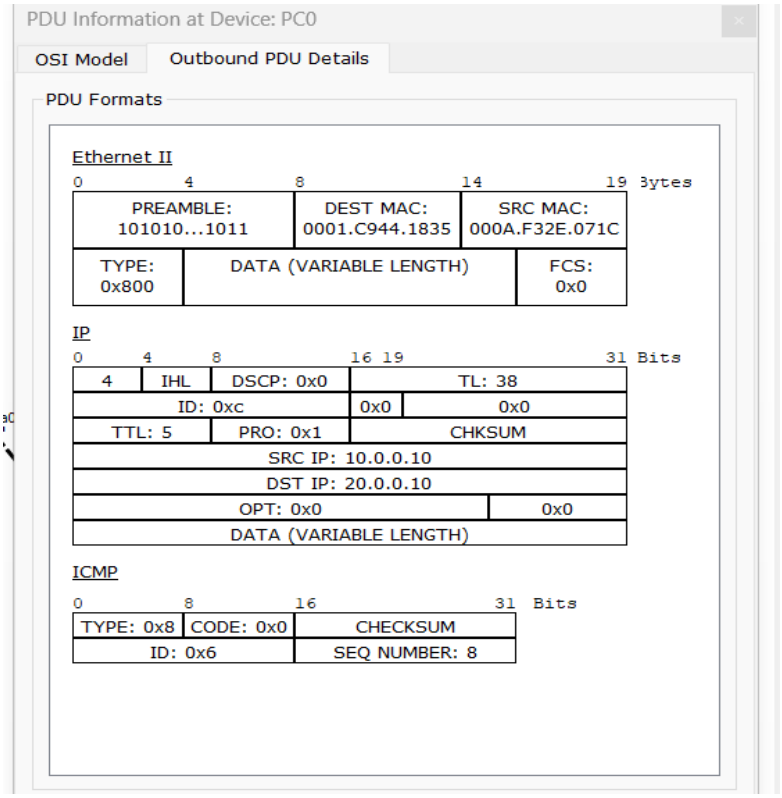
Program 7

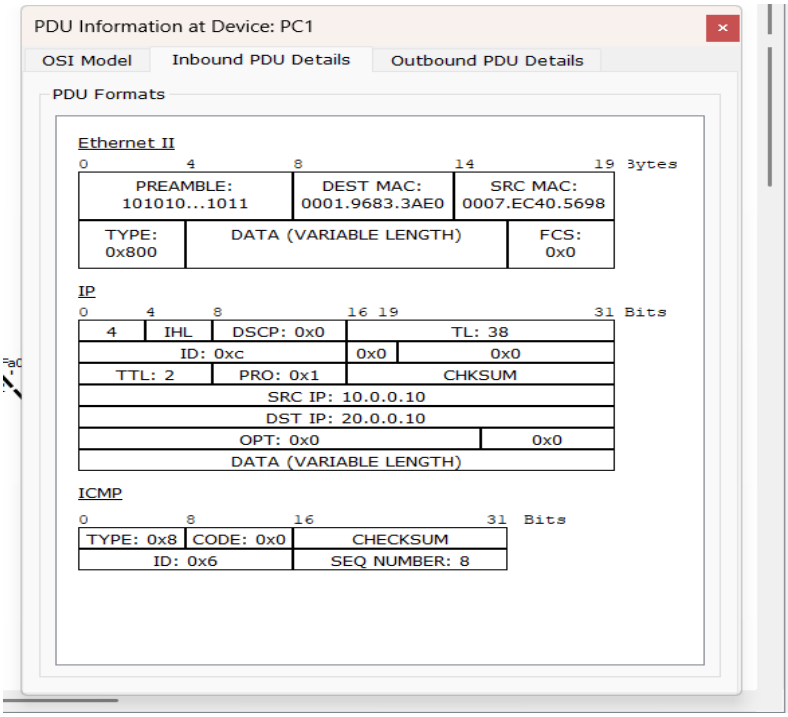
Aim : Demonstrate the TTL/ Life of a Packet

Observations:



Output:

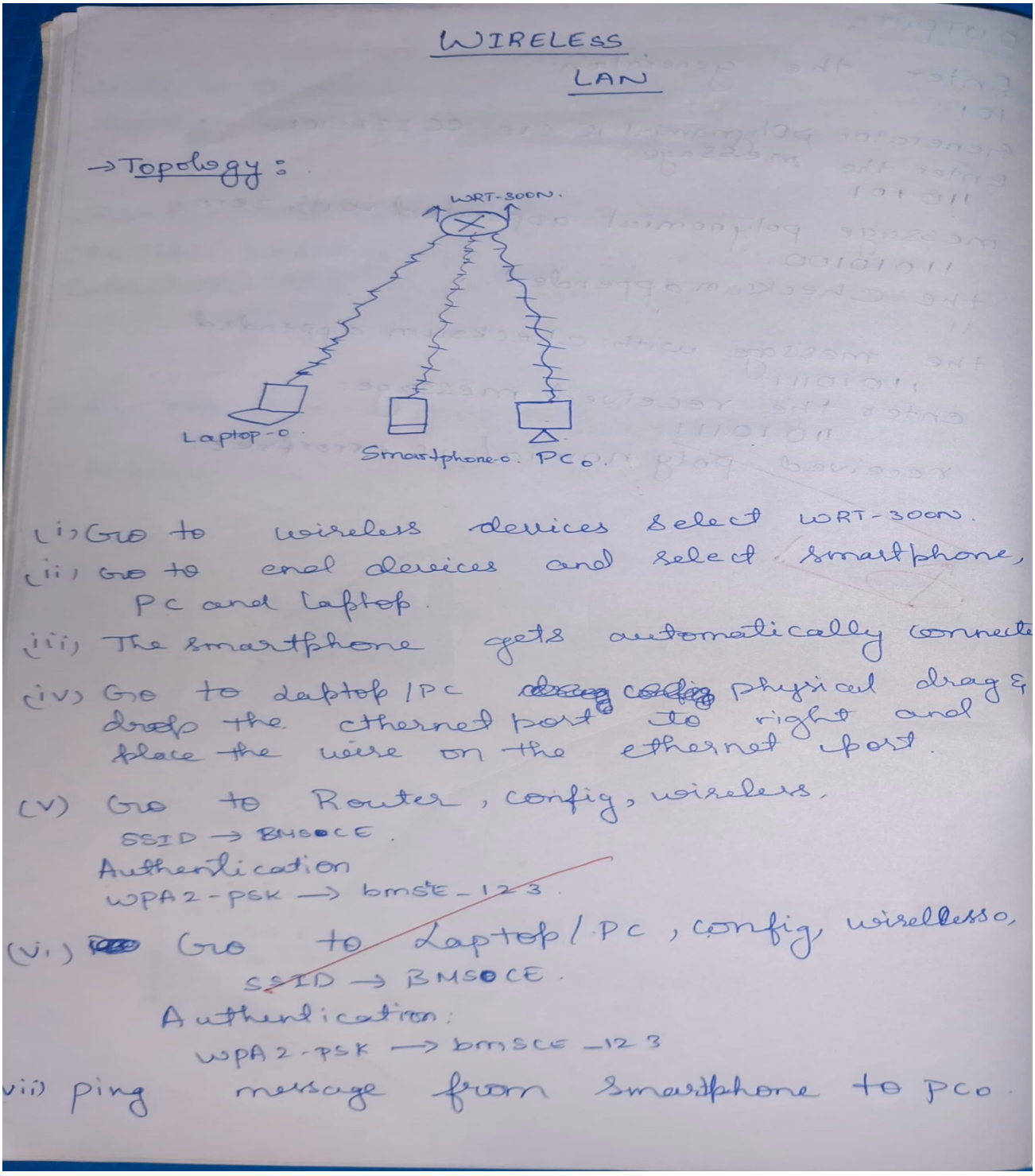

PDU Information at Device: PC0
OSI Model
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Bytes
PREAMBLE: 101010...1011
DEST MAC: 0001.C944.1835
SRC MAC: 000A.F32E.071C
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 38
ID: 0xc
0x0
0x0
TTL: 5
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.10
DST IP: 20.0.0.10
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x6
SEQ NUMBER: 8


PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Bytes
PREAMBLE: 101010...1011
DEST MAC: 0001.9683.3AE0
SRC MAC: 0007.EC40.5698
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 38
ID: 0xc
0x0
0x0
TTL: 2
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.10
DST IP: 20.0.0.10
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x6
SEQ NUMBER: 8

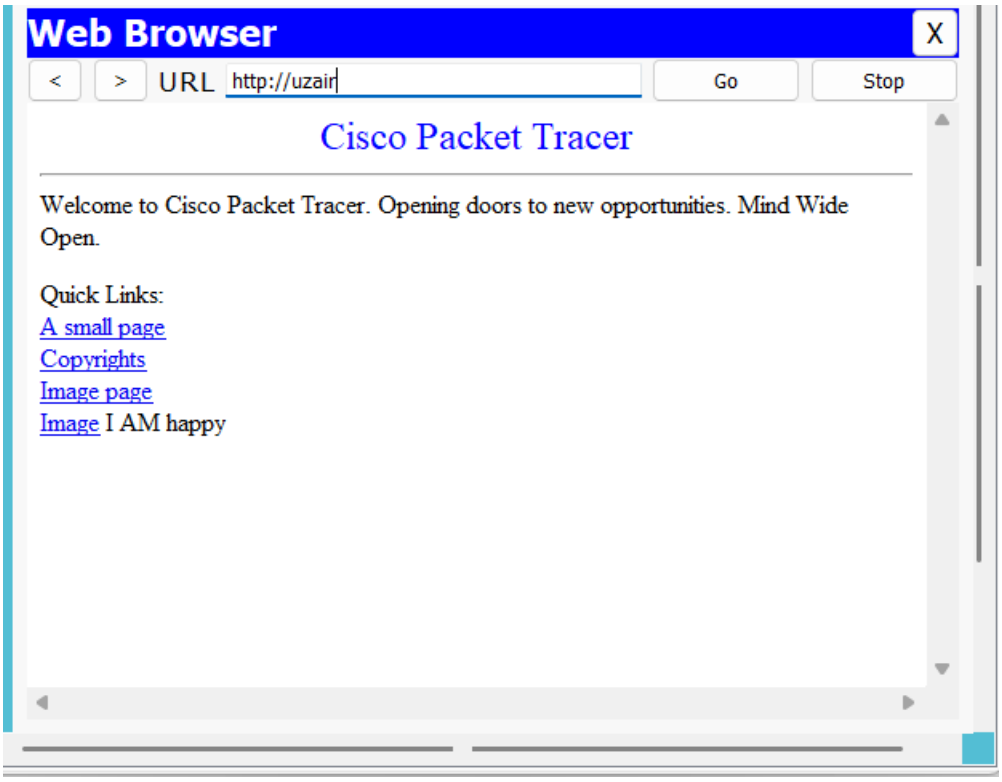
Program 8

Aim : Configure Web Server, DNS within a LAN.

Observation:



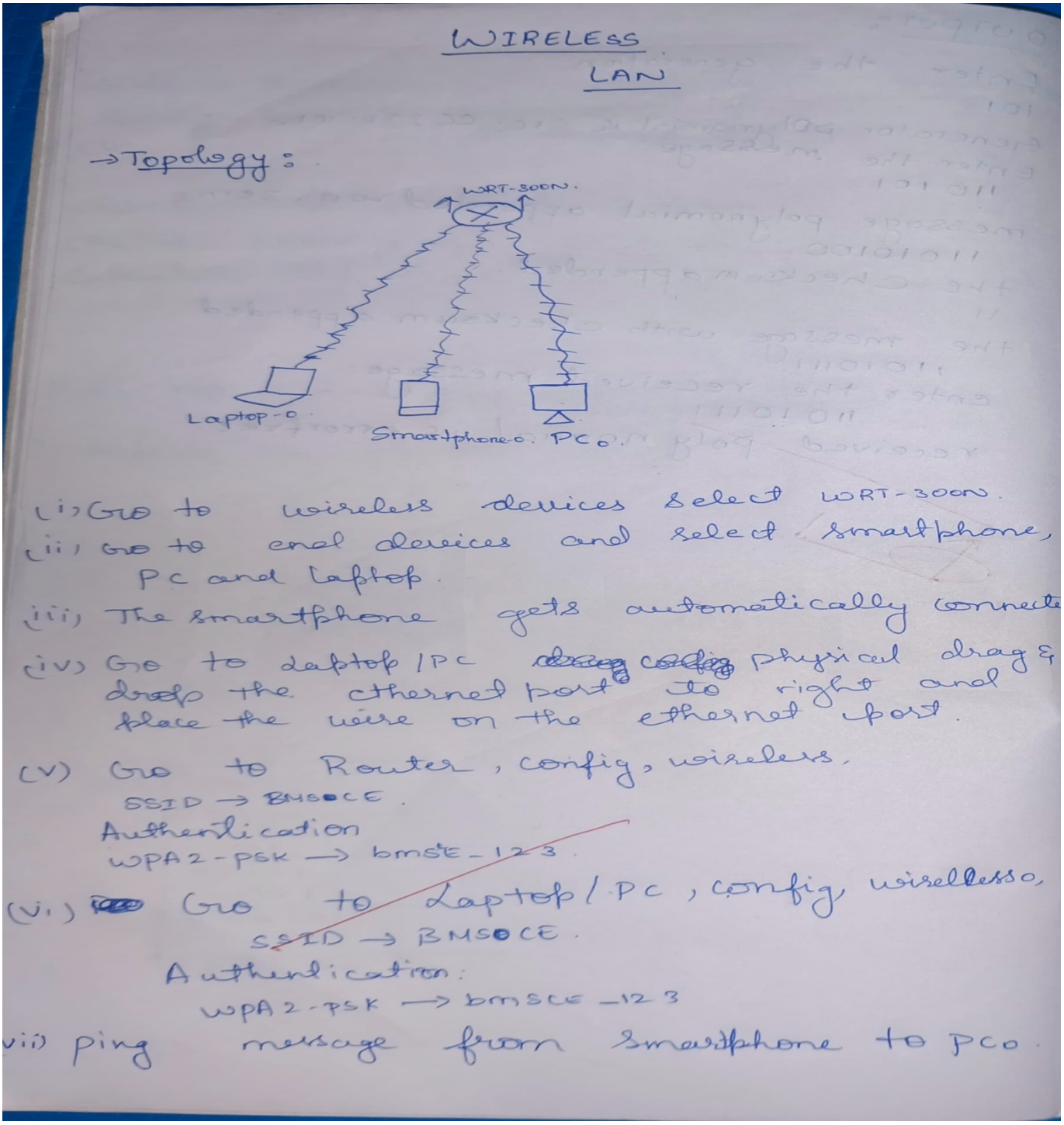
Output:


Web Browser
X
<
>
URL
http://uzair
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image I AM happy

Program 9

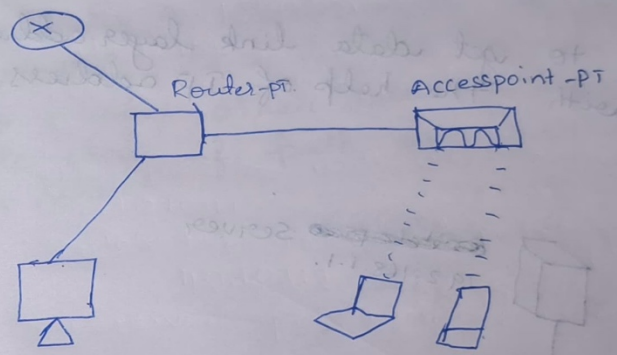
Aim : To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Observation:



OUTPUT:

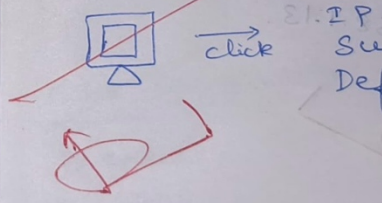
Source	Destination	Last Status
Smartphone0	PC0	Successful
Laptop0	PC0	Successful



Router:

IP Addresses: 192.168.0.1

Subnet mask: 255.255.255.0

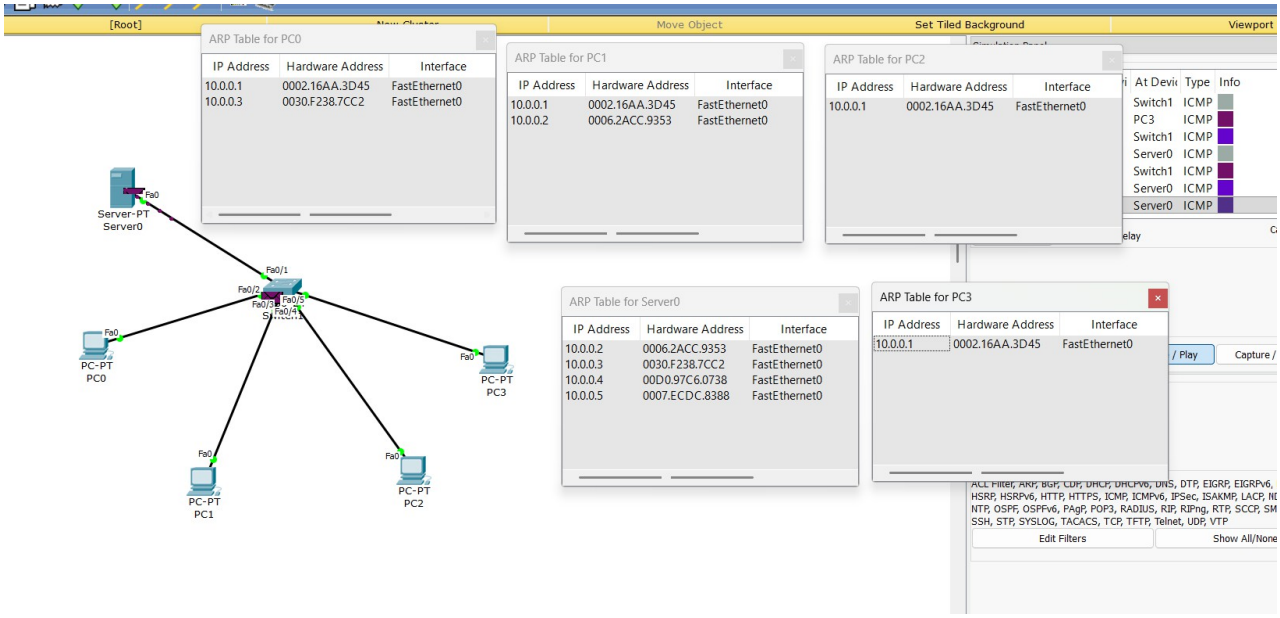


IP address: 192.168.0.2

Subnet mask: 255.255.255.0

Default Server: 192.168.0.1

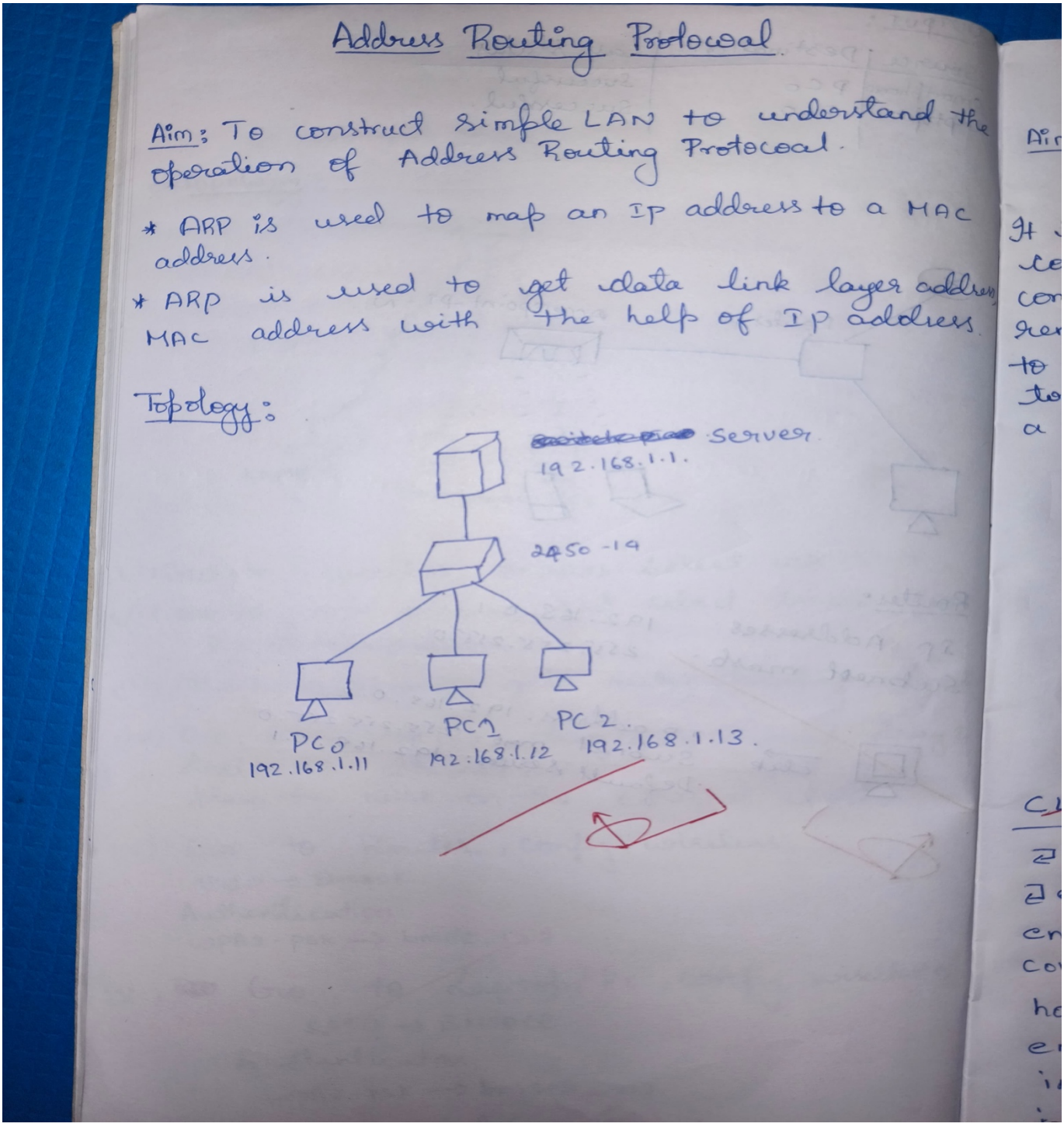
Output:


[Root]
Move Object
Set Tiled Background
Viewport
ARP Table for PC0
IP Address Hardware Address Interface
10.0.0.1 0002.16AA.3D45 FastEthernet0
10.0.0.3 0030.F238.7CC2 FastEthernet0
ARP Table for PC1
IP Address Hardware Address Interface
10.0.0.1 0002.16AA.3D45 FastEthernet0
10.0.0.2 0006.2ACC.9353 FastEthernet0
ARP Table for PC2
IP Address Hardware Address Interface
10.0.0.1 0002.16AA.3D45 FastEthernet0
ARP Table for Server0
IP Address Hardware Address Interface
10.0.0.2 0006.2ACC.9353 FastEthernet0
10.0.0.3 0030.F238.7CC2 FastEthernet0
10.0.0.4 00D0.97C6.0738 FastEthernet0
10.0.0.5 0007.ECDC.8388 FastEthernet0
ARP Table for PC3
IP Address Hardware Address Interface
10.0.0.1 0002.16AA.3D45 FastEthernet0
At Devi Type Info
Switch1 ICMP
PC3 ICMP
Switch1 ICMP
Server0 ICMP
Switch1 ICMP
Server0 ICMP
Server0 ICMP
Server-PT Server0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
Edit Filters
Show All/None

Program 10

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

Observation:



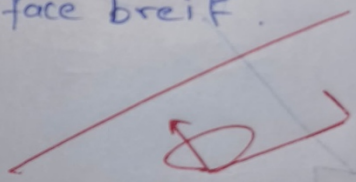
To cross check commanel:

Show ip interface brief.

Command of PC0

* Ping 192.168.1.1.
  telnet 192.168.1.1
* password: tp
  password: tp
  Ri >en.
  password: tp (secret)
  password: tp (secret)

R1# : Show ip interface breif.



Output:

```
PC>telnet 10.0.0.3
Trying 10.0.0.3 ...
% Connection timed out; remote host not responding
```

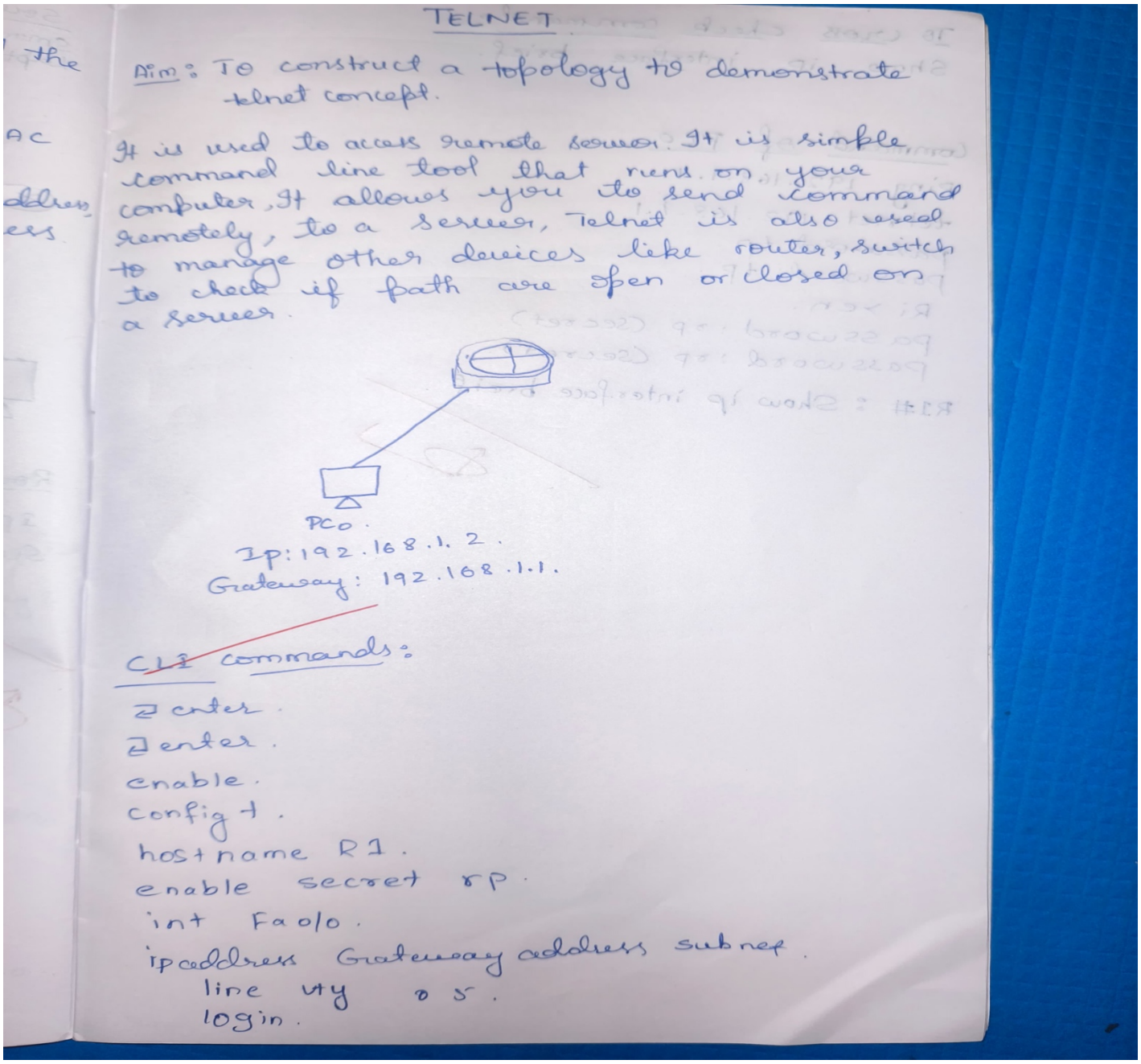
```
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open

[Connection to 10.0.0.1 closed by foreign host]
```

```
[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>ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1
ms

r1>
```

Program 11

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

Observation:



TELNET

Aim: To construct a topology to demonstrate telnet concept.

It is used to acess remote server. It is simple, command line tool that runs on your computer, It allows you to send command remotely, to a server, Telnet is also used to manage other devices like router, switch to check if path are open or closed on a server.

PC0
Ip: 192.168.1.2
Gateway: 192.168.1.1

CLI commands:

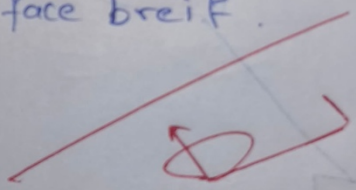
enter.
enter.
enable.
config t.
hostname R1.
enable secret rp.
int Fa0/0.
ip address Gateway address subnet.
line vty 0 5.
login.

To cross check commanel:

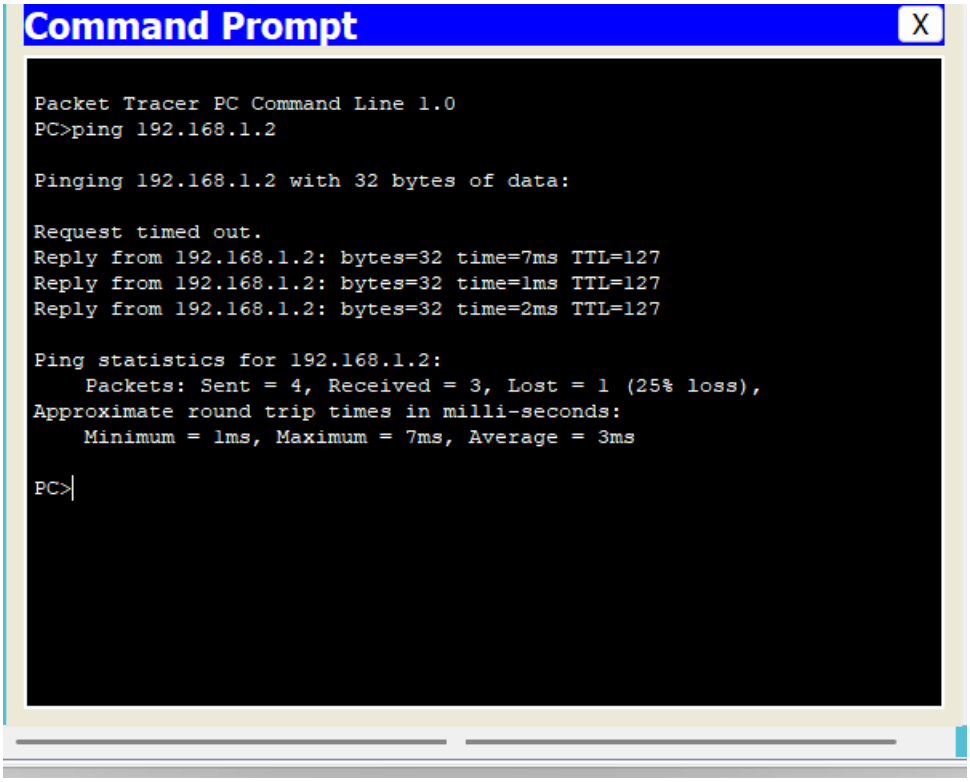
Show ip interface brief.

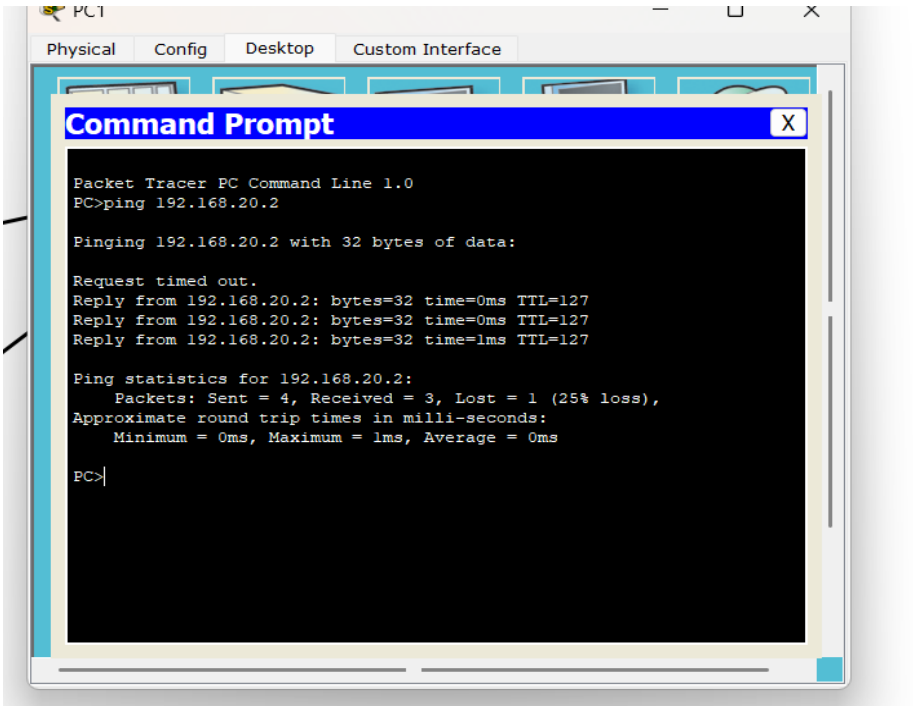
Command of PC0

* Ping 192.168.1.1
  telnet 192.168.1.1
* password: tp
  password: tp
  Ri>en.
  password: cp (secret)
  password: cp (secret)
  R1# : Show ip interface breif



Output:

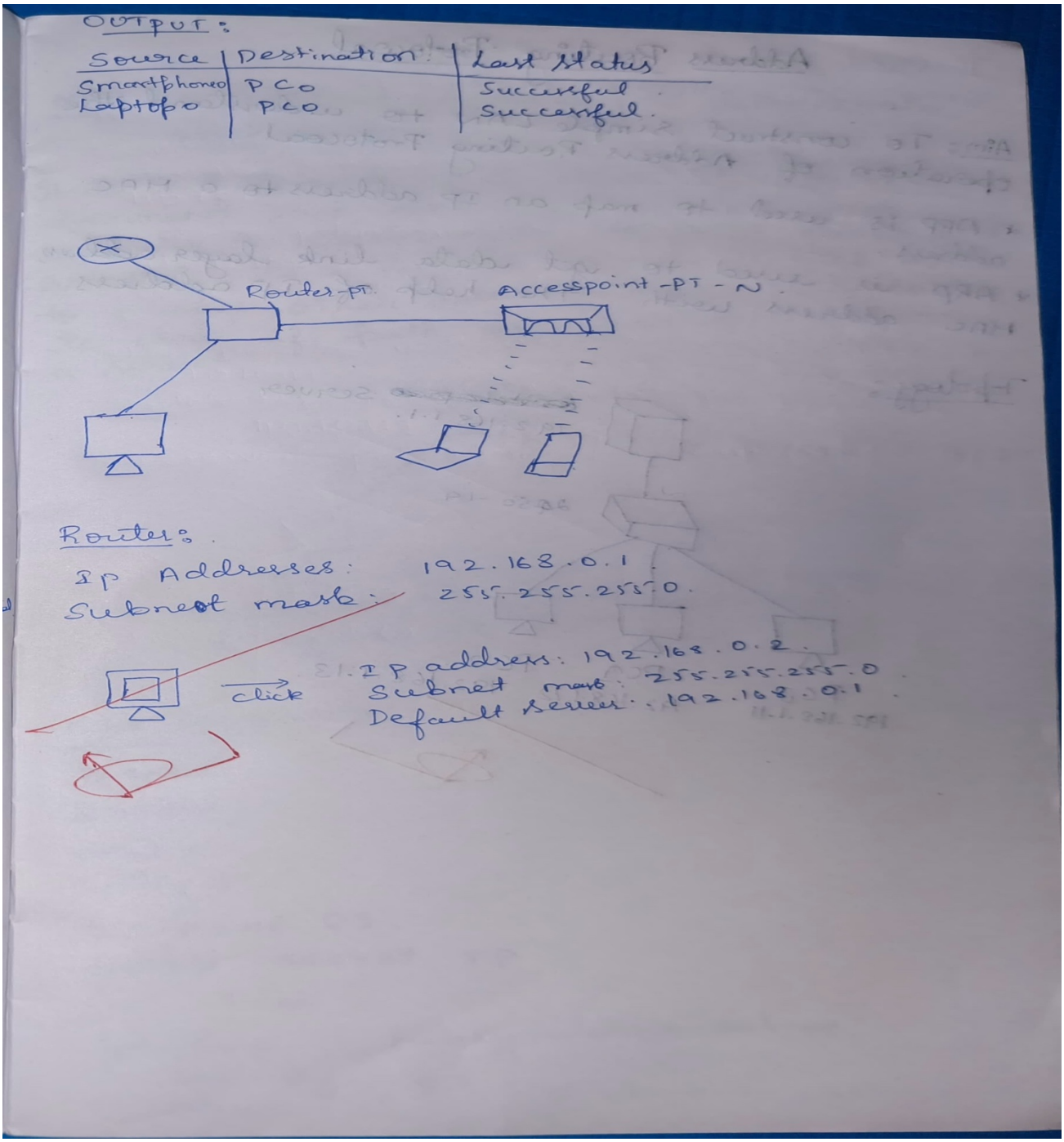

Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.2: bytes=32 time=7ms TTL=127
Reply from 192.168.1.2: bytes=32 time=1ms TTL=127
Reply from 192.168.1.2: bytes=32 time=2ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 7ms, Average = 3ms
PC>

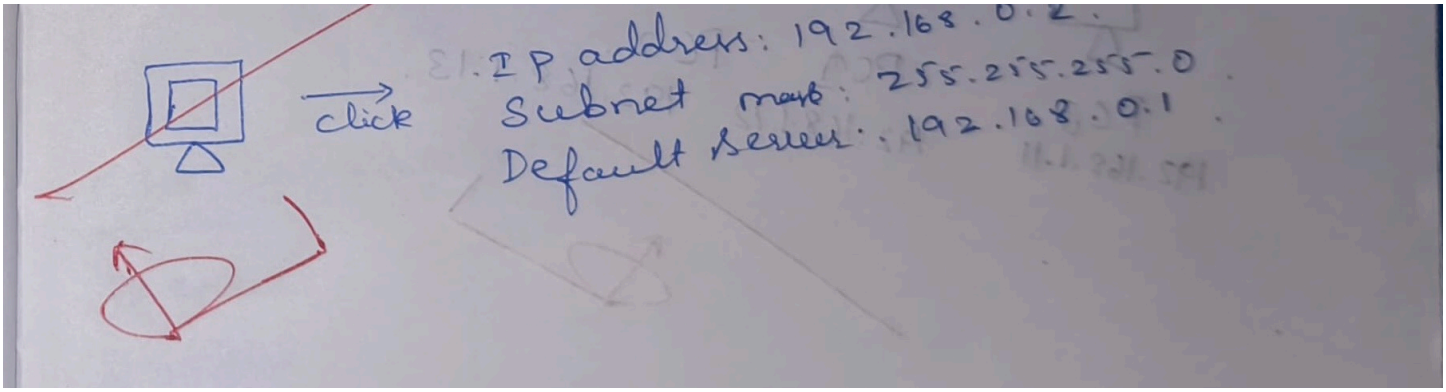

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.20.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

Program 12

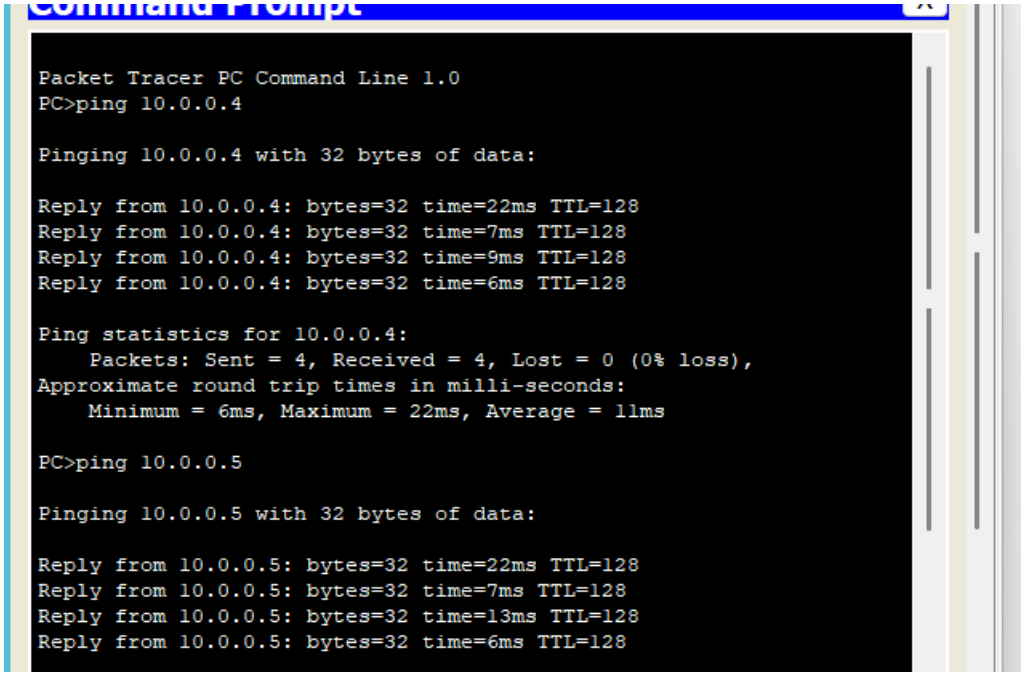
Aim : To construct a WLAN and make the nodes communicate wirelessly

Observation:




click
IP address: 192.168.0.2
Subnet mask: 255.255.255.0
Default Server: 192.168.0.1

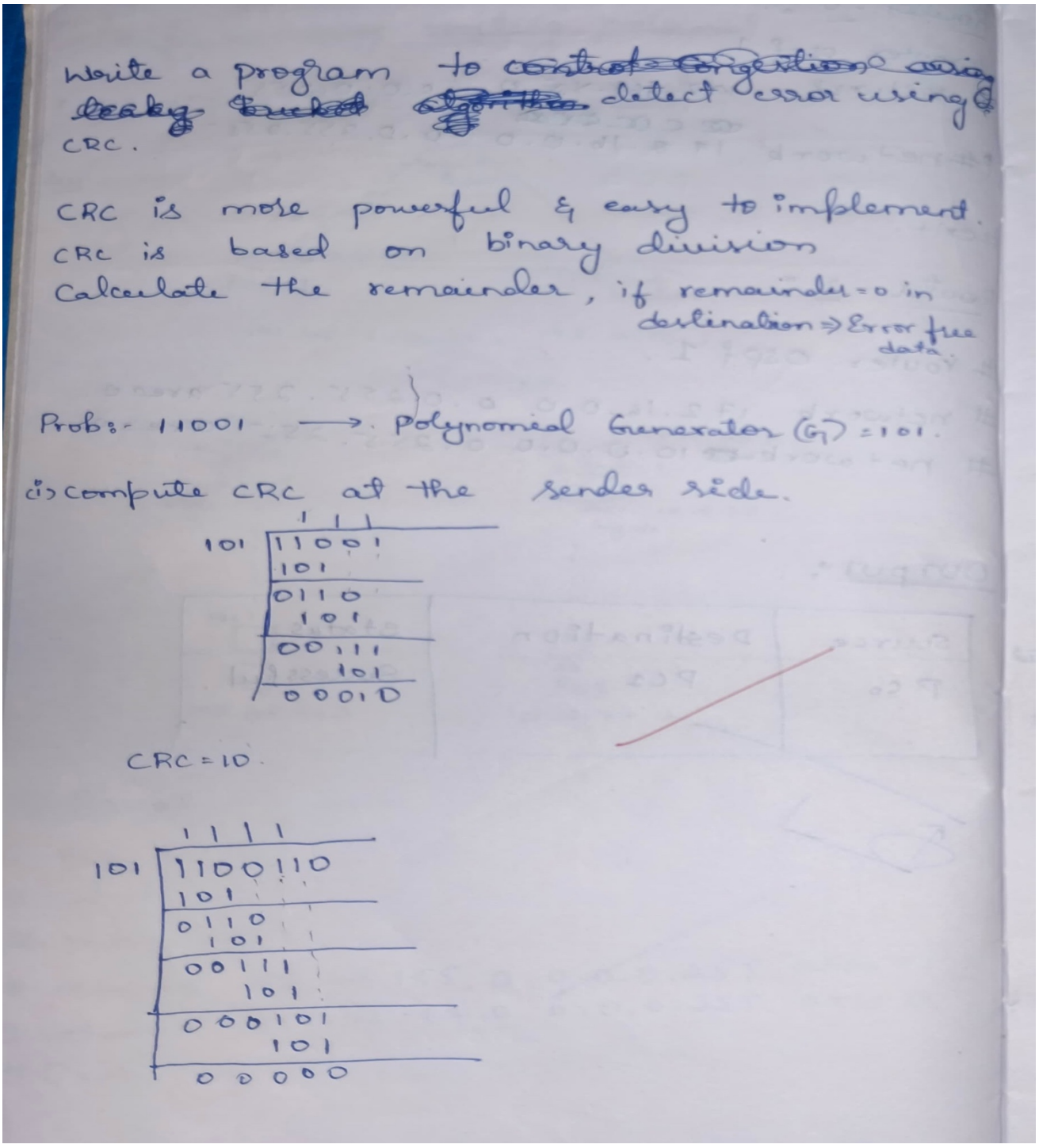
Output:


Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=22ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms TTL=128
Reply from 10.0.0.4: bytes=32 time=9ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 22ms, Average = 11ms
PC>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=22ms TTL=128
Reply from 10.0.0.5: bytes=32 time=7ms TTL=128
Reply from 10.0.0.5: bytes=32 time=13ms TTL=128
Reply from 10.0.0.5: bytes=32 time=6ms TTL=128

Program 13

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

Observation:



Code:

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

void computeCRC(char data[], char poly[], char remainder[])
{
   char temp[200];
   int dataLen = strlen(data);
   int polyLen = strlen(poly);

   strcpy(temp, data);

   // Append (polyLen - 1) zeros to data
   for (int i = 1; i < polyLen; i++)
      strcat(temp, "0");

   printf("\nData padded (message + zeros): %s\n", temp);

   // Perform binary division
   for (int i = 0; i <= strlen(temp) - polyLen; i++)
   {
      if (temp[i] == '1')
      {
         for (int j = 0; j < polyLen; j++)
            temp[i + j] = (temp[i + j] == poly[j]) ? '0' : '1';
      }
   }

   // CRC = last (polyLen-1) bits
   strcpy(remainder, temp + (strlen(temp) - (polyLen - 1)));
}

int checkReceived(char received[], char poly[])
{
   char temp[200];
```

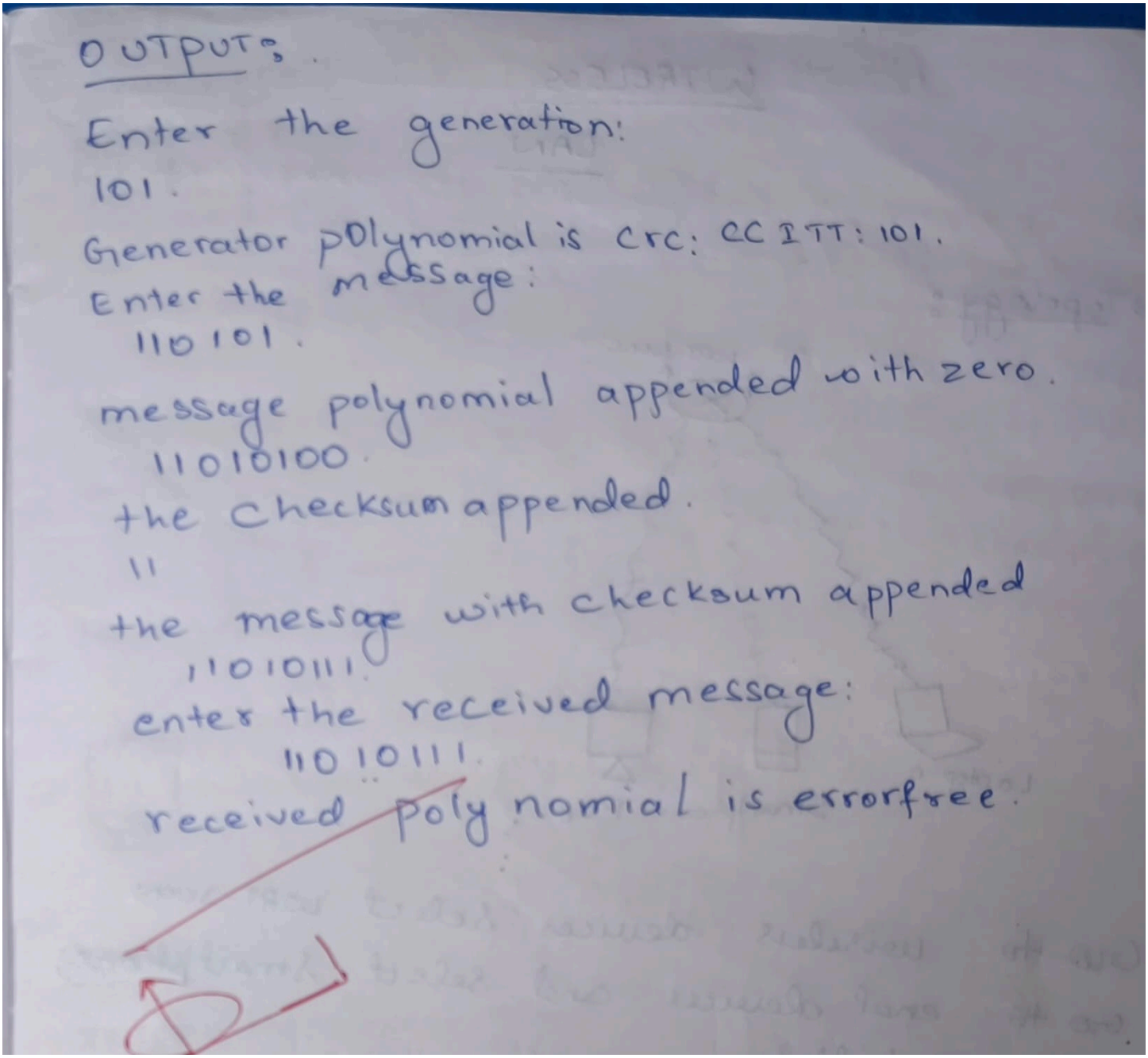
```
    int polyLen = strlen(poly);

    strcpy(temp, received);

    // Perform division on received message
    for (int i = 0; i <= strlen(temp) - polyLen; i++)
    {
        if (temp[i] == '1')
        {
            for (int j = 0; j < polyLen; j++)
                temp[i + j] = (temp[i + j] == poly[j]) ? '0' : '1';
        }
    }

    // Check any '1' in remainder portion
    for (int i = strlen(temp) - (polyLen - 1); i < strlen(temp); i++)
        if (temp[i] == '1')
            return 0; // Error detected

    return 1; // No error
}

int main()
{
    char message[100], poly[30], crc[30], transmitted[150], received[150];

    printf("Enter the message bits       : ");
    scanf("%s", message);

    printf("Enter the polynomial (g(x))  : ");
    scanf("%s", poly);

    computeCRC(message, poly, crc);

    printf("CRC value (remainder)        : %s\n", crc);
```

```
    // Form transmitted message
    strcpy(transmitted, message);
    strcat(transmitted, crc);

    printf("Message to be transmitted     : %s\n", transmitted);

    // Receiver side
    printf("\nEnter received message bits (can modify 1 bit to test error): ");
    scanf("%s", received);

    if (checkReceived(received, poly))
        printf("\n■     No Error: Message received correctly.\n");
    else
        printf("\n+ ERROR detected in received message!\n");

    return 0;
}
```

Output:

```
Enter the message bits        101101
Enter the polynomial (g(x)) : 1101

Data padded (message + zeros): 101101000
CRC value (remainder)         101
Message to be transmitted     101101101

Enter received message bits (can modify 1 bit to test error): 101101101

■  No Error: Message received correctly.
```

Output:


OUTPUT:
Enter the generation:
101.
Generator polynomial is CRC: CCITT: 101.
Enter the message:
110101.
message polynomial appended with zero.
11010100.
the checksum appended.
11
the message with checksum appended
11010111.
enter the received message:
11010111.
received polynomial is errorfree.