VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

PRAGNYA B S(1BM21CS132)

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

June-2023 to September-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" carried out by **PRAGNYA B S (1BM21CS132),** 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 academic semester June-2023 to September-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.

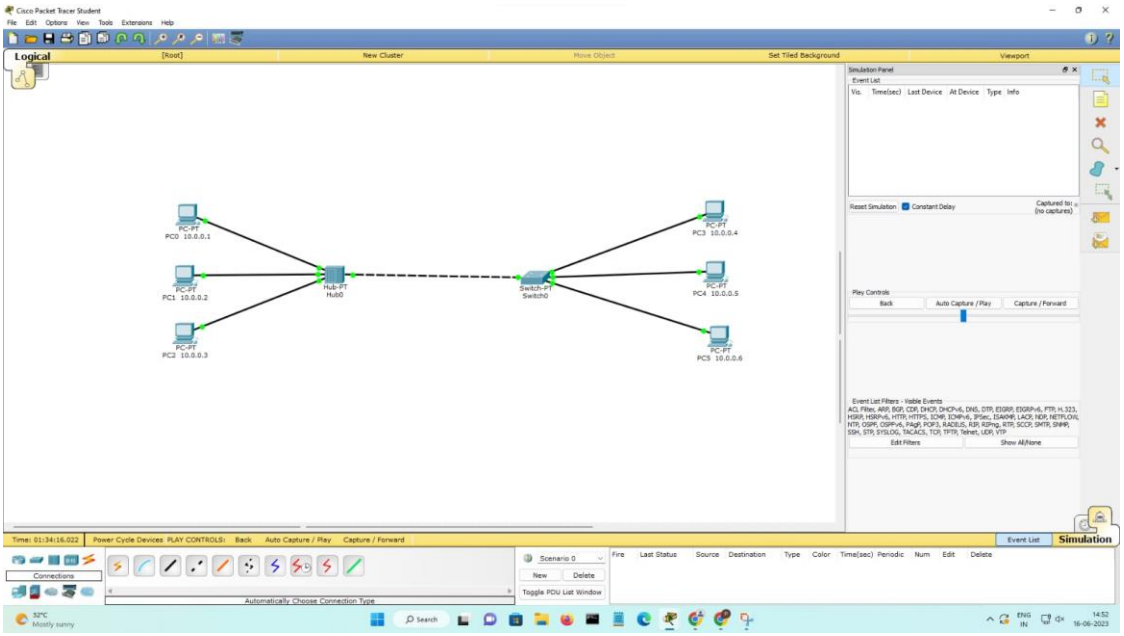
Prof.Swathi Sridharan
Assistant Professor
Department of CSE
BMSCE, Bengaluru

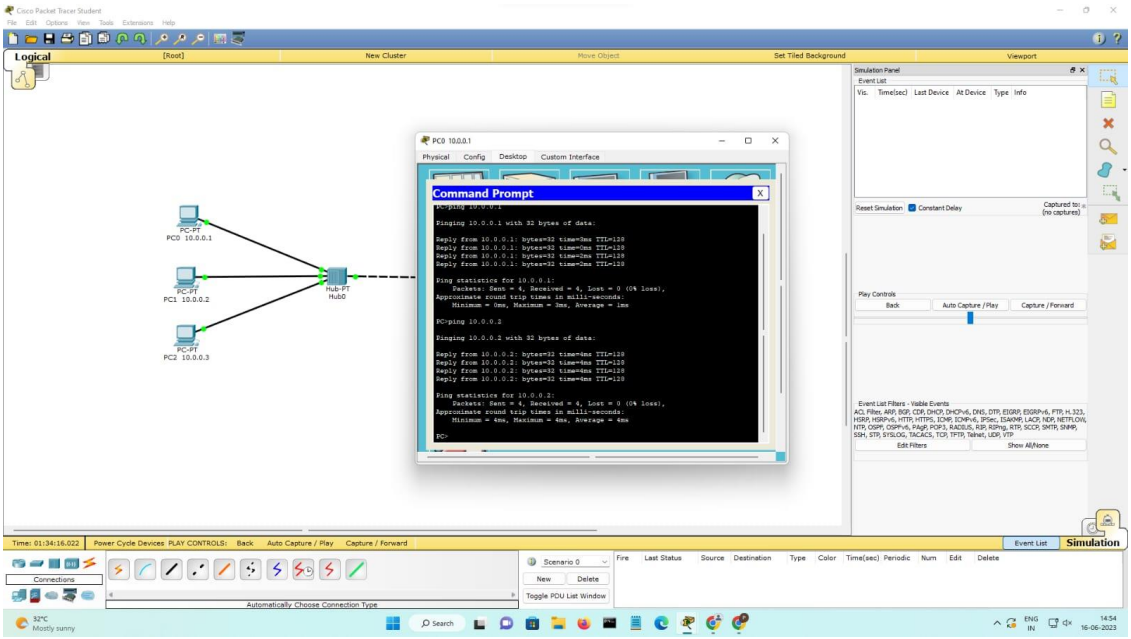
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

INDEX

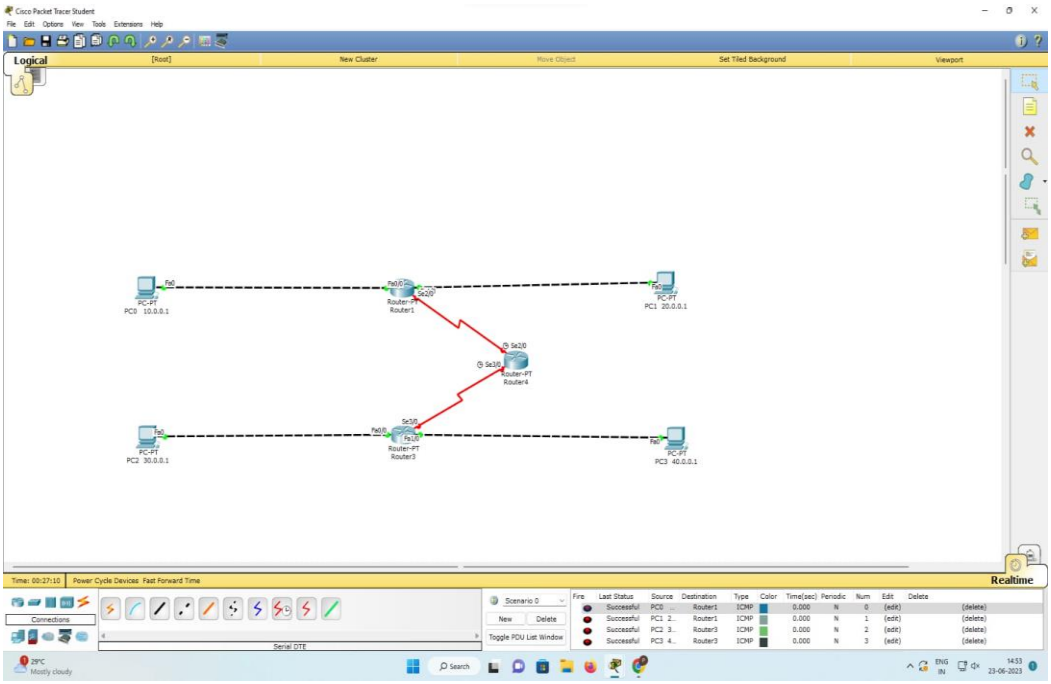
Experiment #	Unit #	Name of Experiment
		CYCLE 1
1	2	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.
2	3	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply
3	3	Configure default route, static route to the Router
4	5	Configure DHCP within a LAN and outside LAN. ,.
5	3	Configure RIP routing Protocol in Routers
6	3	Configure OSPF routing protocol
7	3	Demonstrate the TTL/ Life of a Packet
8	5	Configure Web Server, DNS within a LAN.
9	2	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
10	5	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.
11	3	To construct a VLAN and make the PC's communicate among a VLAN
12		To construct a WLAN and make the nodes communicate wirelessly
		CYCLE - 2
13	2	Write a program for error detecting code using CRC-CCITT (16-bits).
14		Write a program for congestion control using Leaky bucket algorithm.
15	4	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.
16	4	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.
17	3,4,5	Tool Exploration -Wireshark

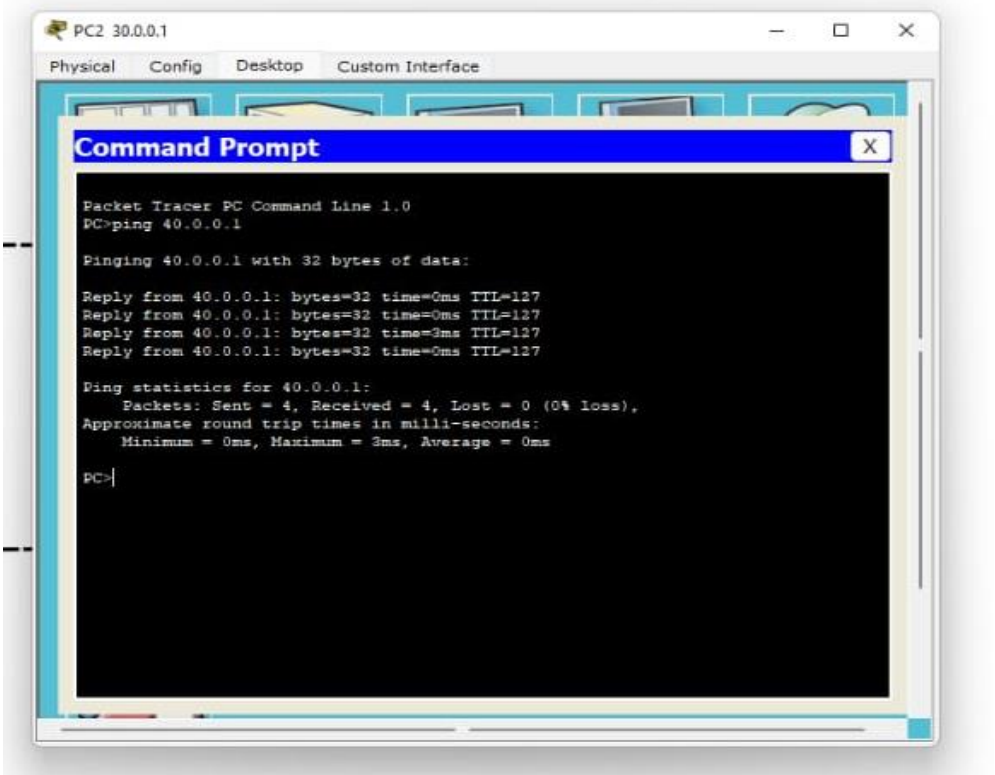
1) Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message

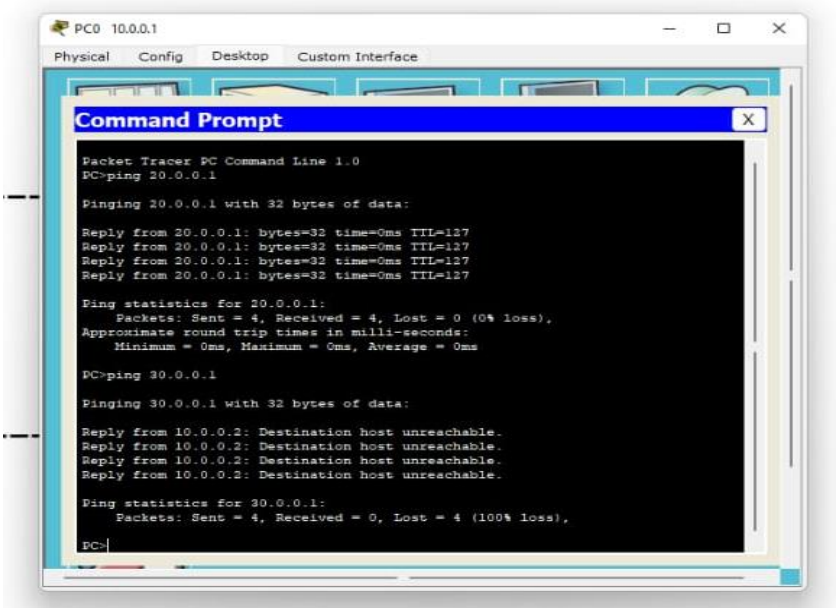




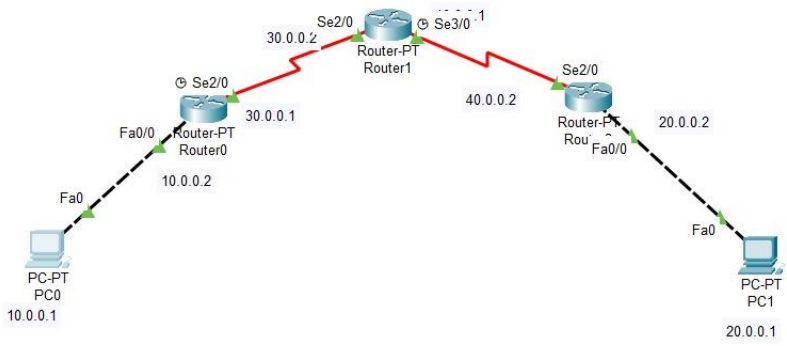
2) Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

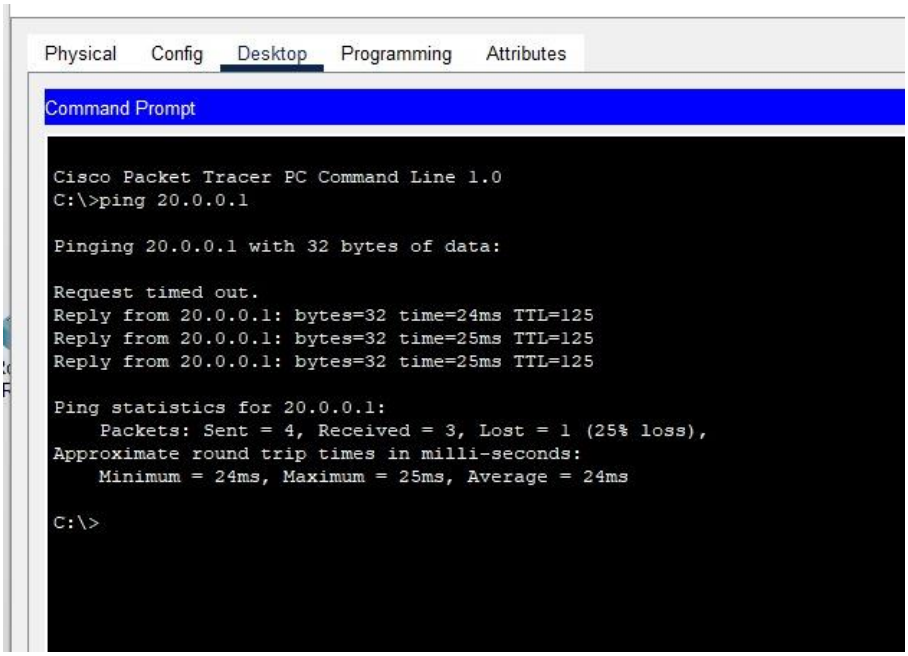




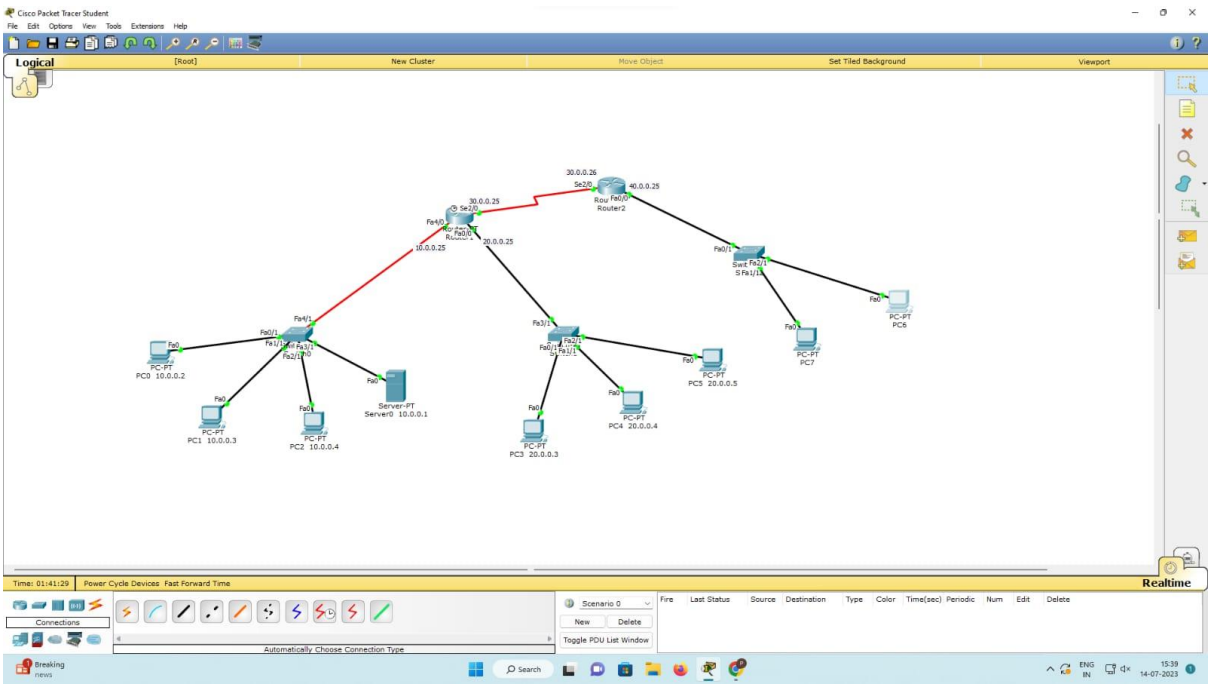


3) Configure default route, static route to the Router

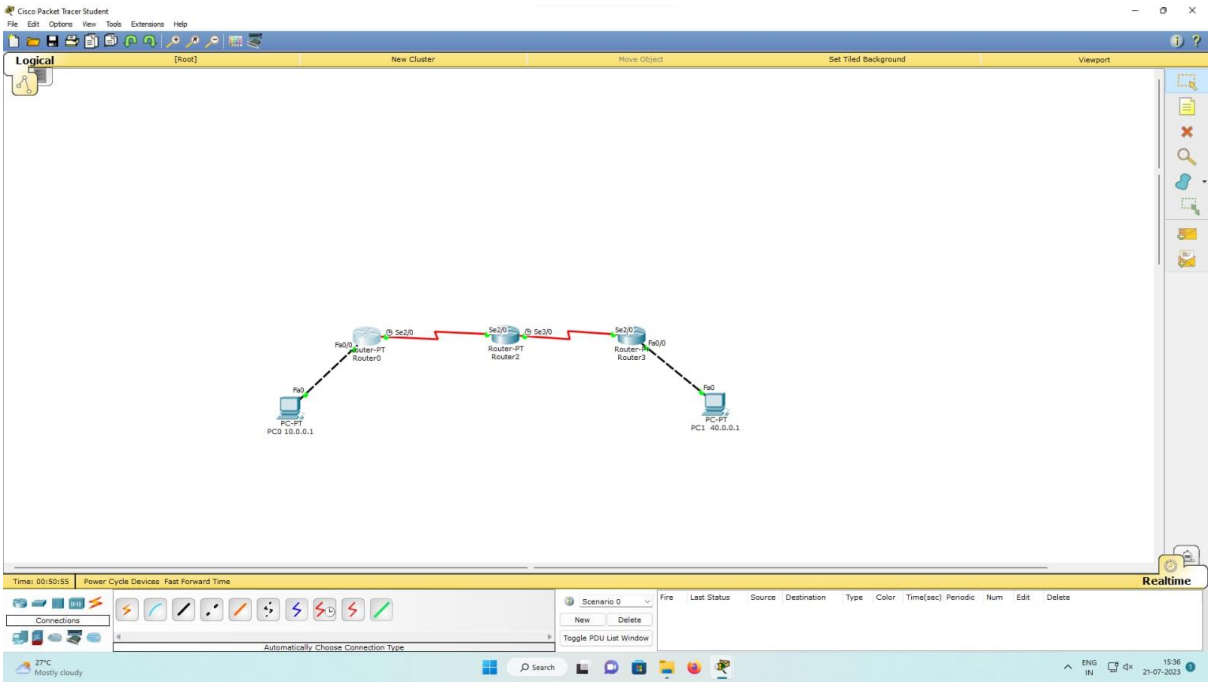


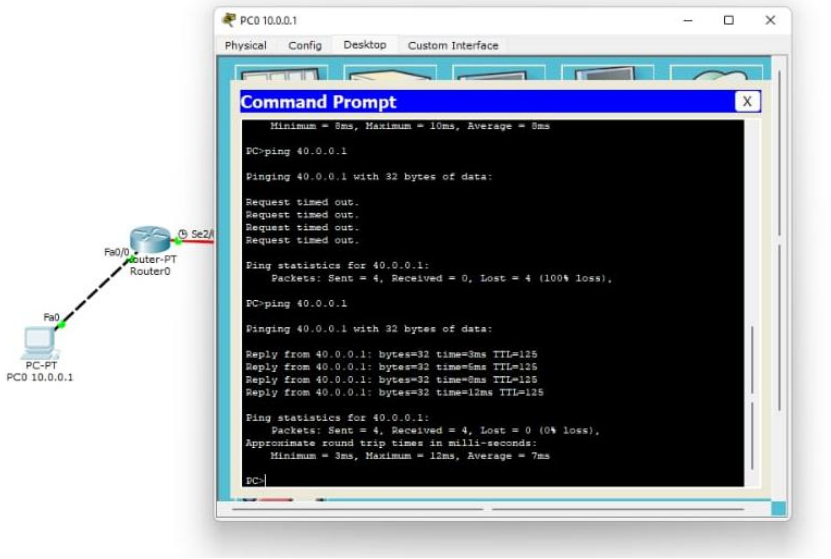


4) Configure DHCP within a LAN and outside LAN. ,.

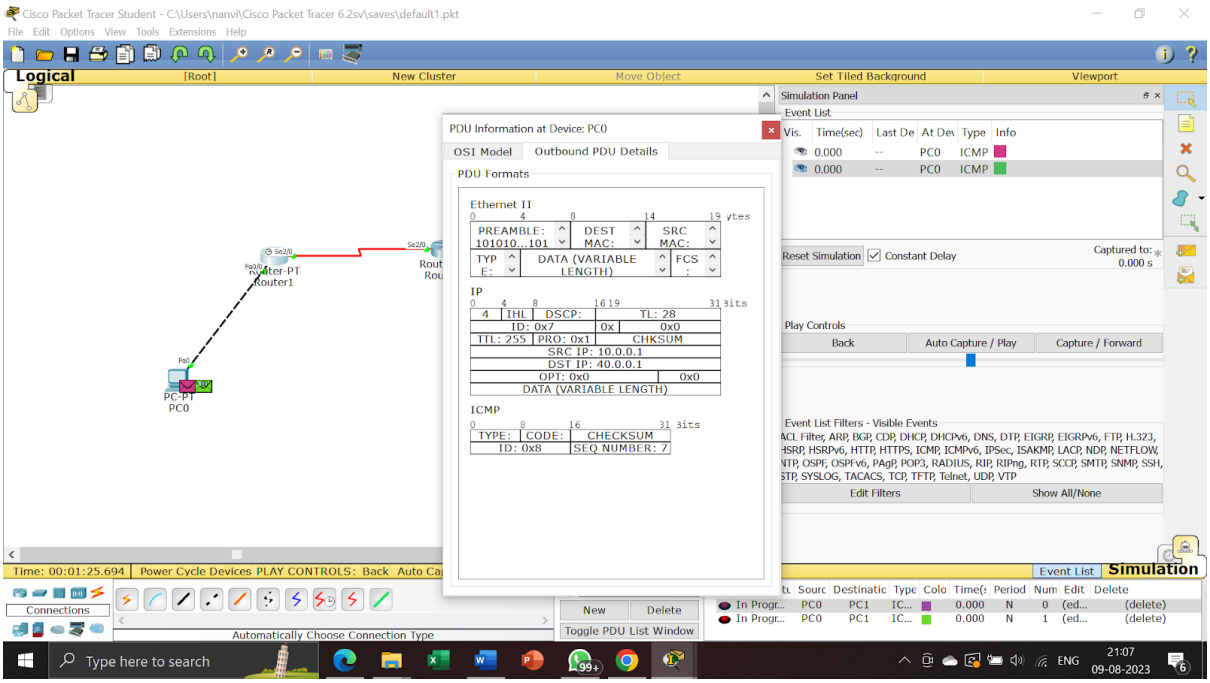


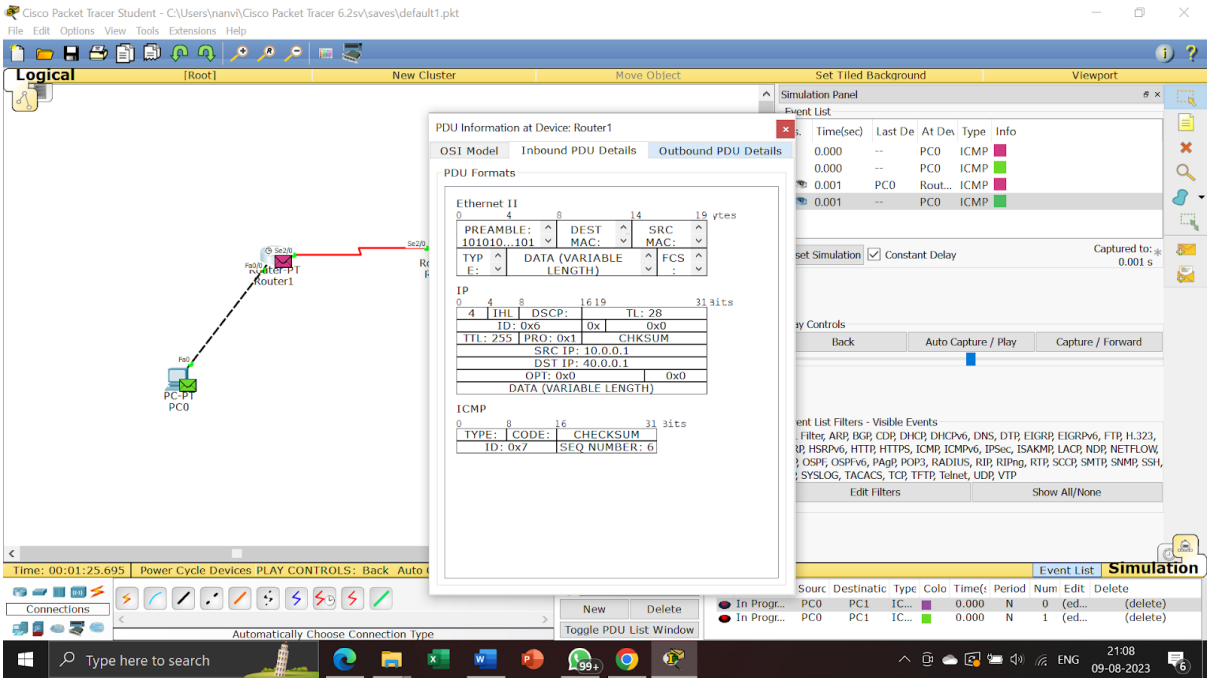
5) Configure RIP routing Protocol in Routers

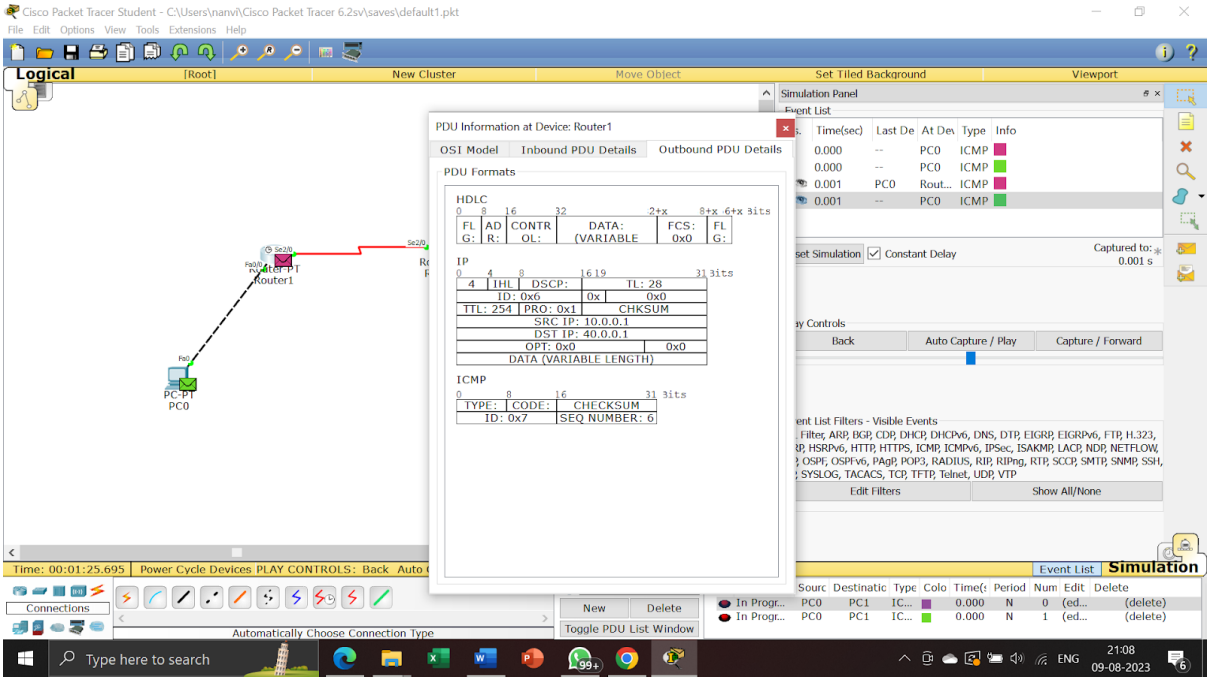


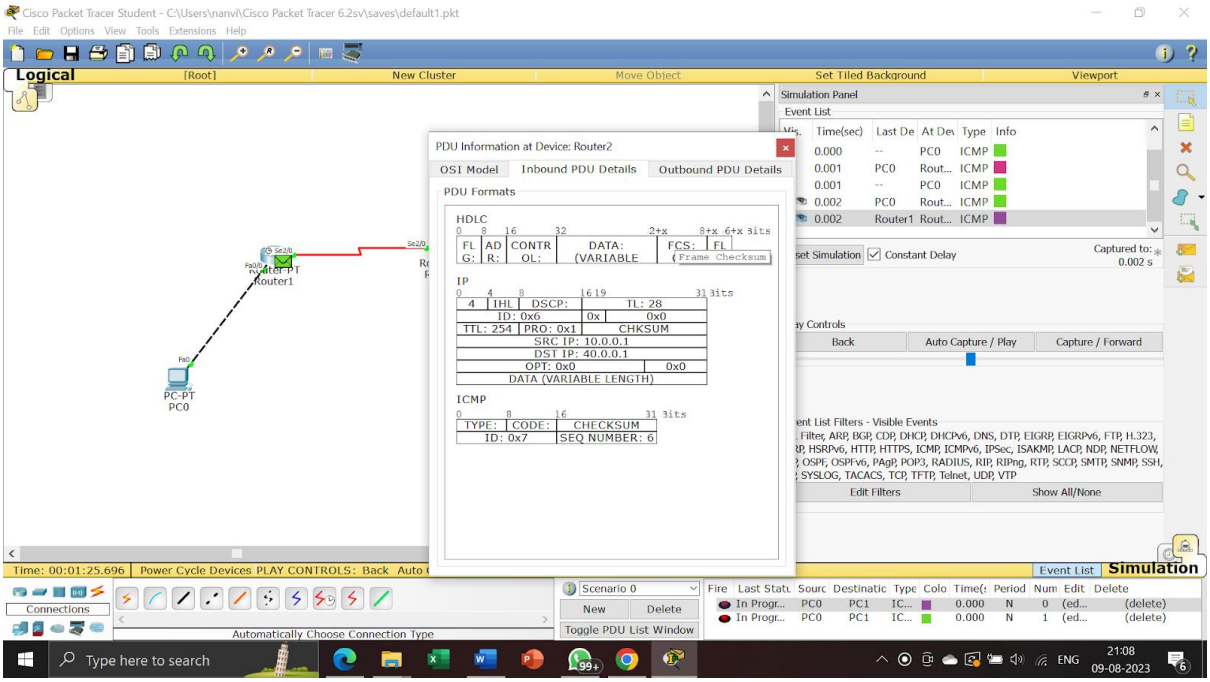


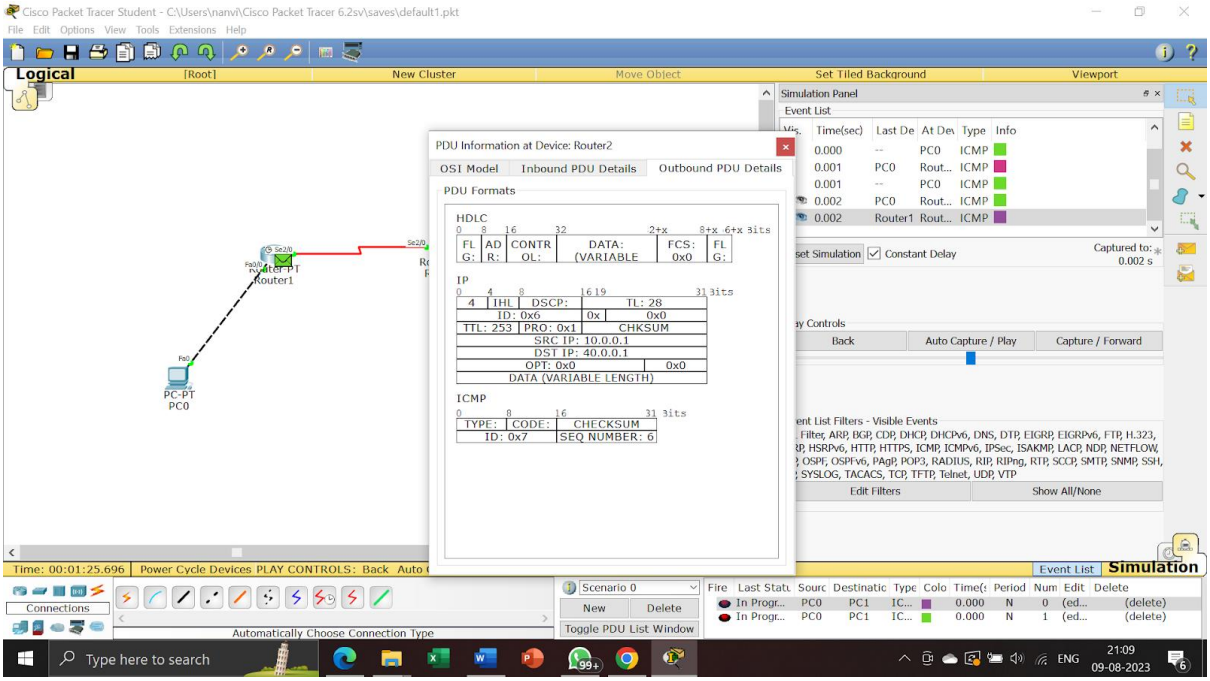
7) Demonstrate the TTL/ Life of a Packet



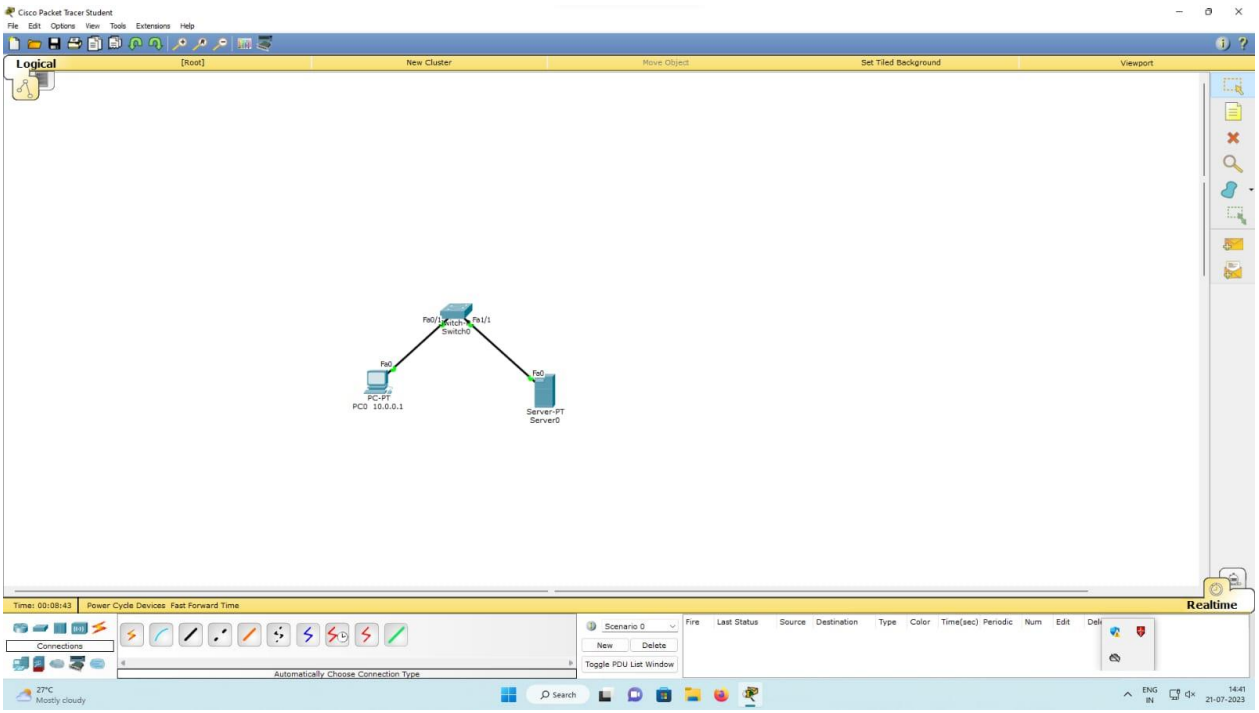

Cisco Packet Tracer Student - C:\Users\nanvi\Cisco Packet Tracer 6.2sv\saves\default1.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...101
DEST MAC:
SRC MAC:
TYPE:
DATA (VARIABLE LENGTH)
FCS:
IP
4 IHL DSCP: TL: 28
ID: 0x6 0x 0x0
TTL: 255 PRO: 0x1 CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: CODE: CHECKSUM
ID: 0x7 SEQ NUMBER: 6
Router-PT Router1
PC-PT PC0
Simulation Panel
Event List
Time(sec) Last De At De Type Info
0.000 -- PC0 ICMP
0.000 -- PC0 ICMP
0.001 PC0 Rout... ICMP
0.001 -- PC0 ICMP
Constant Delay
Captured to: 0.001 s
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:01:25.695
Power Cycle Devices PLAY CONTROLS: Back
Event List
Simulation
Connections
Automatically Choose Connection Type
New
Delete
Toggle PDU List Window
Type here to search
ENG
21:08
09-08-2023

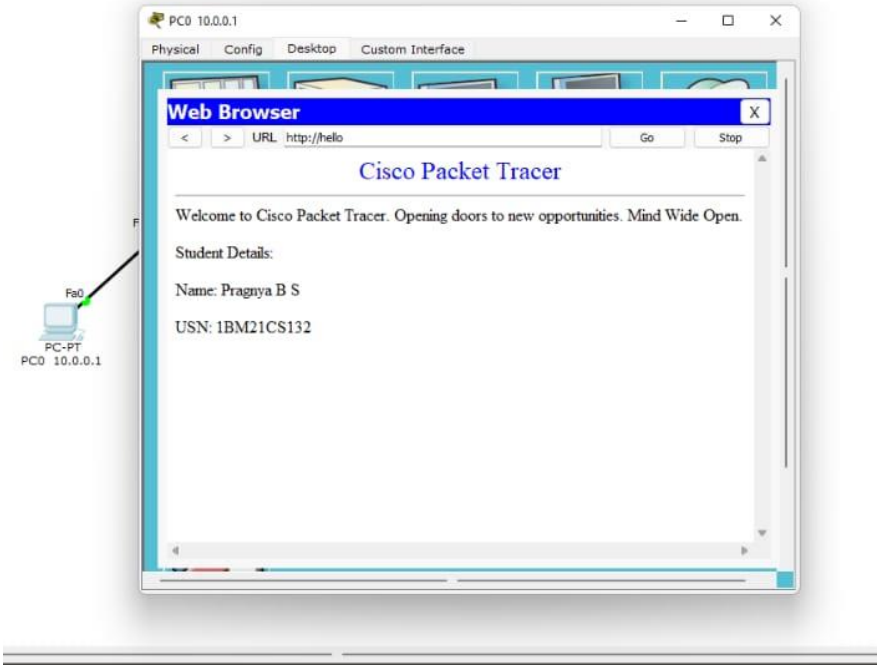

Cisco Packet Tracer Student - C:\Users\nanvi\Cisco Packet Tracer 6.2sv\saves\default1.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FLG:
ADR:
CONTROL:
DATA: (VARIABLE
FCS: 0x0
FLG:
IP
4 IHL DSCP: TL: 28
ID: 0x6 0x 0x0
TTL: 254 PRO: 0x1 CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: CODE: CHECKSUM
ID: 0x7 SEQ NUMBER: 6
Router-PT Router1
PC-PT PC0
Simulation Panel
Event List
Time(sec) Last De At De Type Info
0.000 -- PC0 ICMP
0.000 -- PC0 ICMP
0.001 PC0 Rout... ICMP
0.001 -- PC0 ICMP
Constant Delay
Captured to: 0.001 s
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:01:25.695
Power Cycle Devices PLAY CONTROLS: Back
Event List
Simulation
Connections
Automatically Choose Connection Type
New
Delete
Toggle PDU List Window
Type here to search
ENG
21:08
09-08-2023

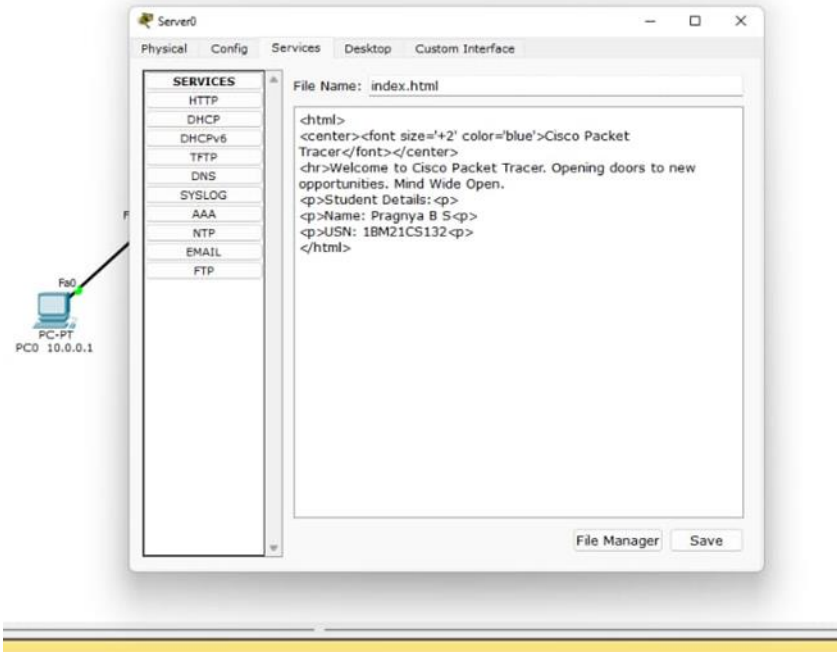

Cisco Packet Tracer Student - C:\Users\nanvi\Cisco Packet Tracer 6.2sv\saves\default1.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FL G: | AD R: | CONTR OL: | DATA: (VARIABLE | FCS: | FL
{ Frame Checksum }
IP
4 | IHL | DSCP: | TL: 28
ID: 0x6 | 0x | 0x0
TTL: 254 | PRO: 0x1 | CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0 | 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: | CODE: | CHECKSUM
ID: 0x7 | SEQ NUMBER: 6
Router1
PC-PT PC0
Simulation Panel
Event List
Time(sec) Last De At De Type Info
0.000 -- PC0 ICMP
0.001 PC0 Rout... ICMP
0.001 -- PC0 ICMP
0.002 PC0 Rout... ICMP
0.002 Router1 Rout... ICMP
Constant Delay
Captured to: 0.002 s
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:01:25.696
Power Cycle Devices PLAY CONTROLS: Back
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Type here to search
ENG
21:08
09-08-2023


Cisco Packet Tracer Student - C:\Users\nanvi\Cisco Packet Tracer 6.2sv\saves\default1.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FL G: | AD R: | CONTR OL: | DATA: (VARIABLE | FCS: 0x0 | FL G:
IP
4 | IHL | DSCP: | TL: 28
ID: 0x6 | 0x | 0x0
TTL: 253 | PRO: 0x1 | CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0 | 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: | CODE: | CHECKSUM
ID: 0x7 | SEQ NUMBER: 6
Router1
PC-PT PC0
Simulation Panel
Event List
Time(sec) Last De At De Type Info
0.000 -- PC0 ICMP
0.001 PC0 Rout... ICMP
0.001 -- PC0 ICMP
0.002 PC0 Rout... ICMP
0.002 Router1 Rout... ICMP
Constant Delay
Captured to: 0.002 s
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:01:25.696
Power Cycle Devices PLAY CONTROLS: Back
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Type here to search
ENG
21:09
09-08-2023

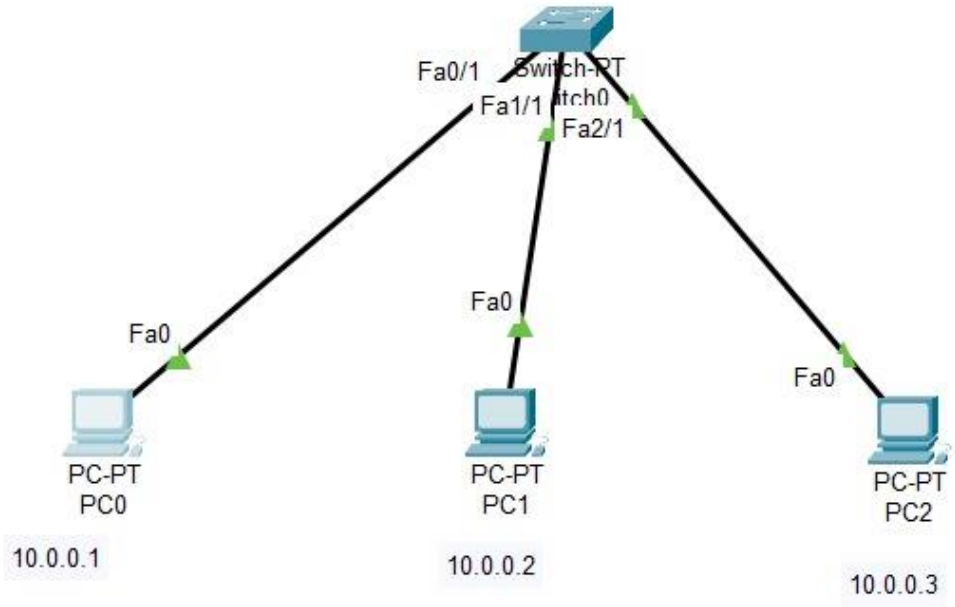
8) Configure Web Server, DNS within a LAN

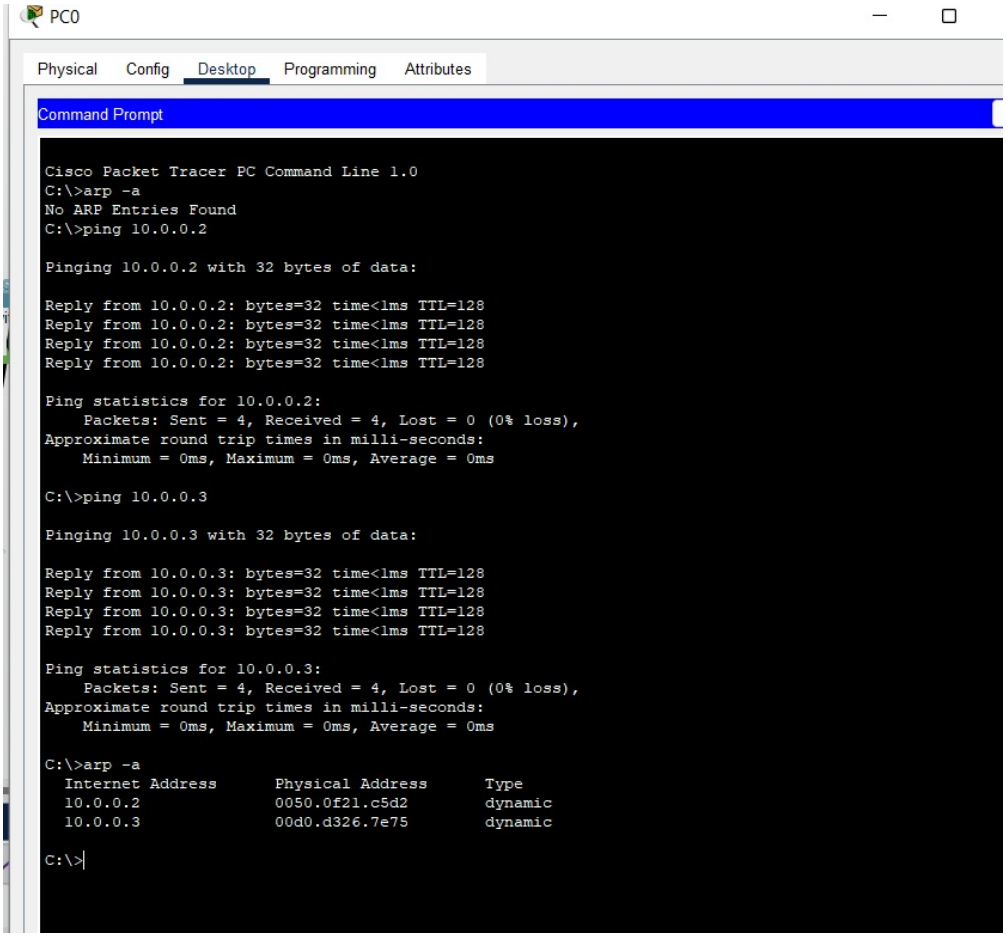




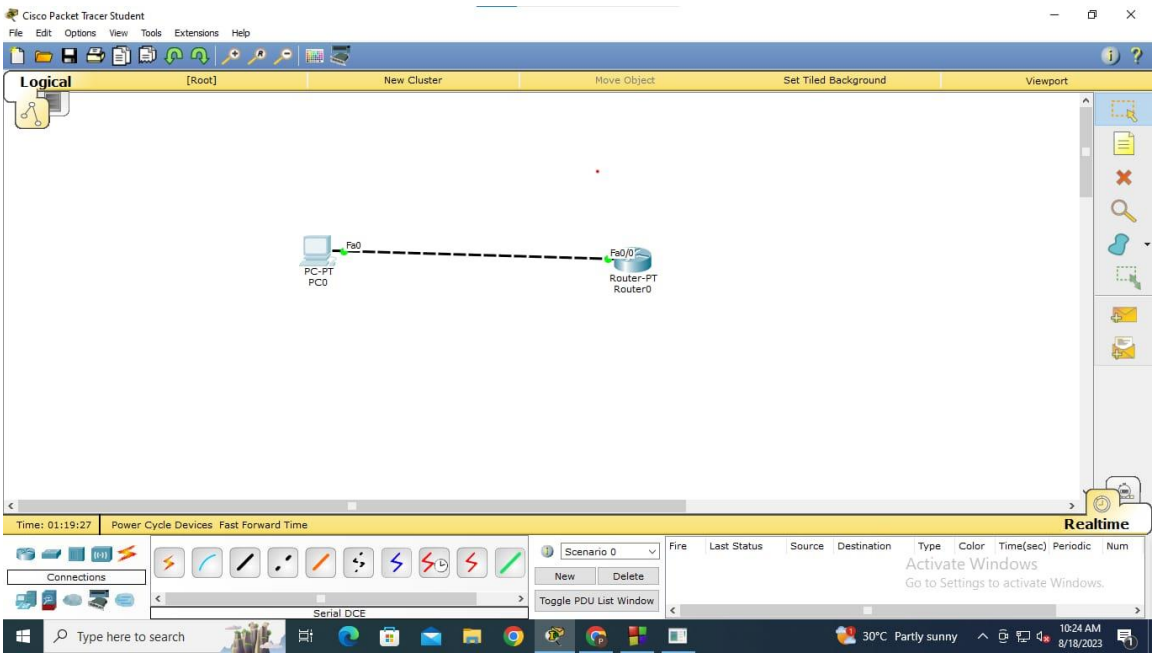


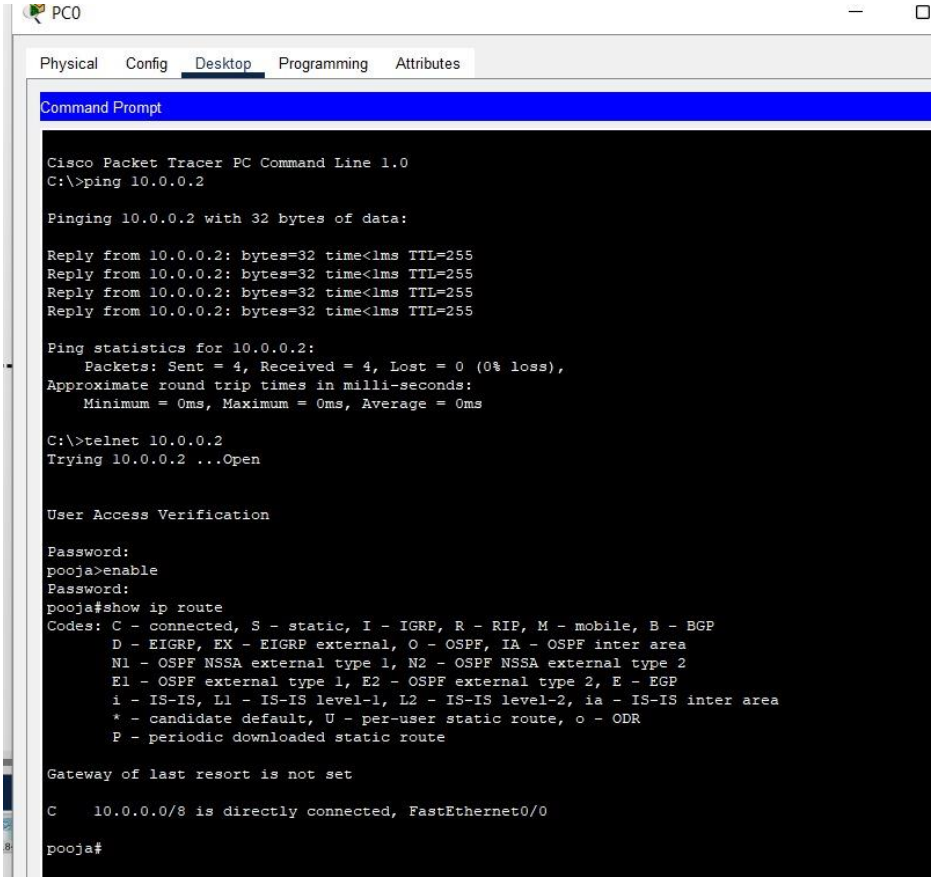
9) To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)



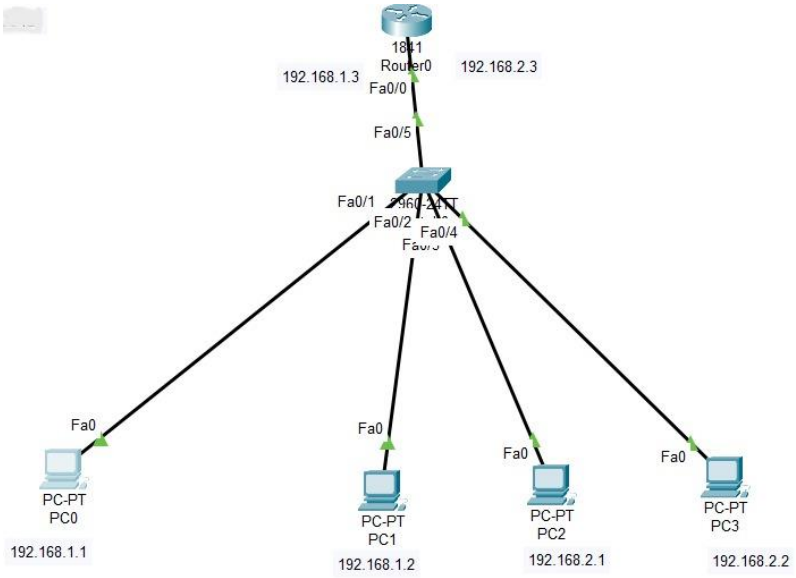


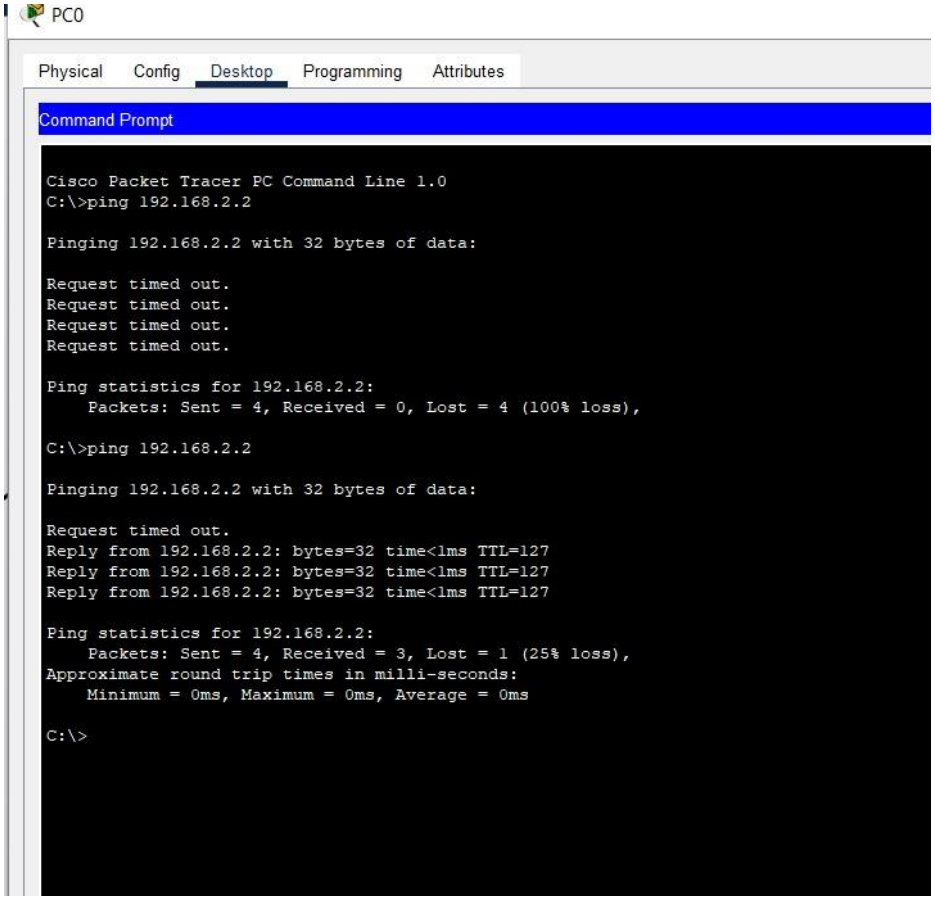
10) To understand the operation of TELNET by accessing the router in server room from a PC in IT office.



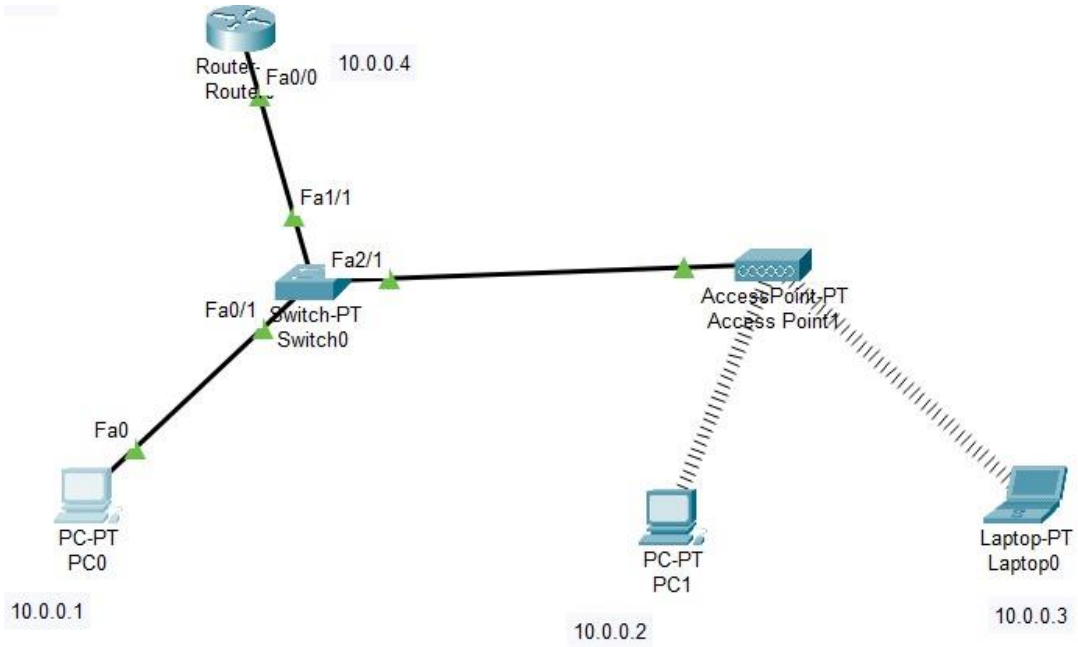


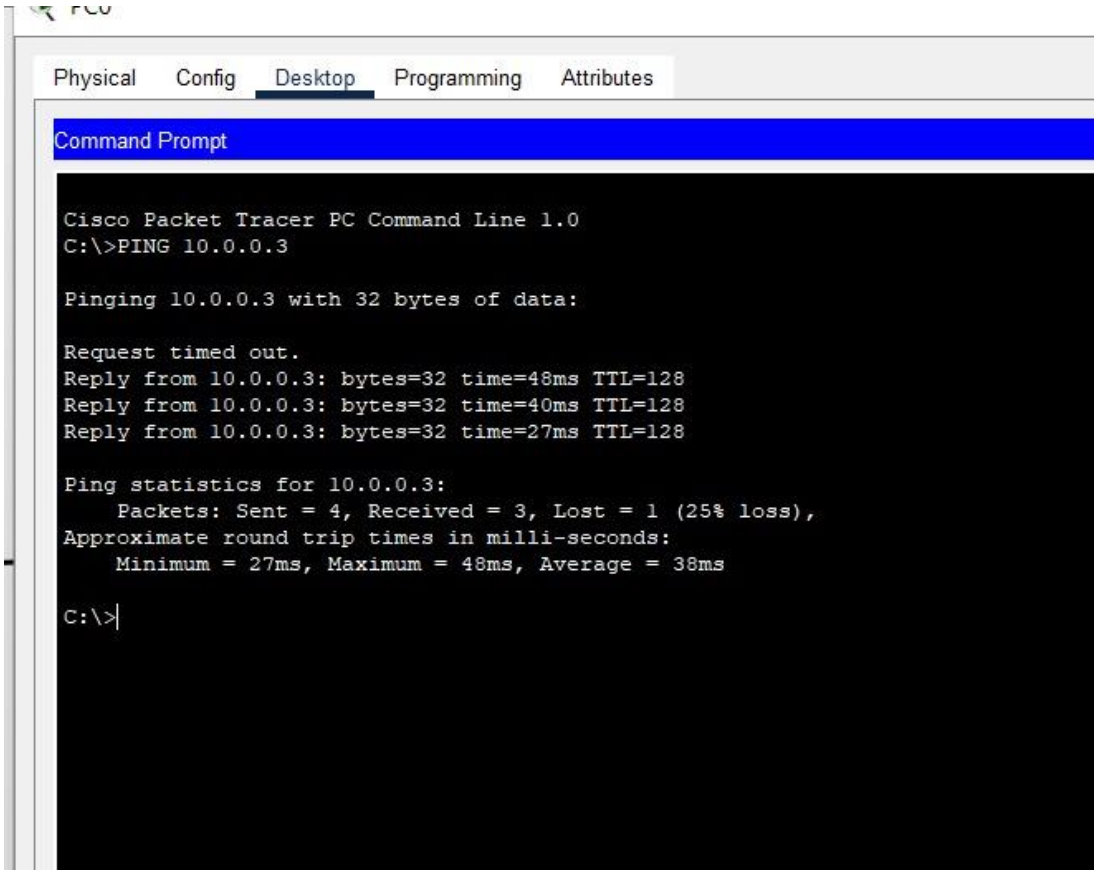
11) To construct a VLAN and make the PC's communicate among a VLAN





12) To construct a WLAN and make the nodes communicate wirelessly





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

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

// CRC-CCITT polynomial: x^16 + x^12 + x^5 + 1 (0x1021)
#define CRC_POLY 0x1021

// Function to calculate CRC-CCITT checksum
unsigned short calculateCRC(const char *data, int length) {
    unsigned short crc = 0xFFFF; // Initial value
    for (int i = 0; i < length; i++) {
        crc ^= (unsigned short)data[i] << 8;
        for (int j = 0; j < 8; j++) {
            if (crc & 0x8000)
                crc = (crc << 1) ^ CRC_POLY;
            else
```

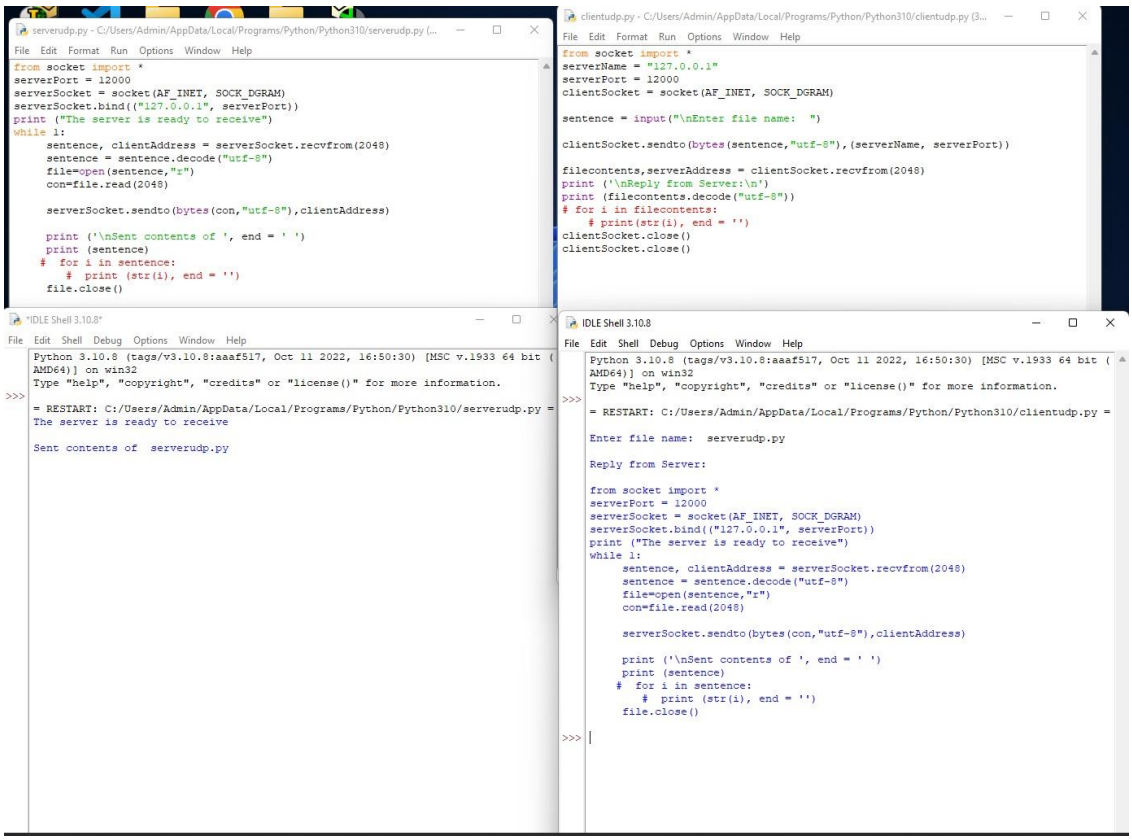
```
            crc <<= 1;
        }
    }
    return crc;
}

int main() {
    char data[100]; // Replace with your actual data
    printf("Enter data: ");
    scanf("%s", data);

    int dataLength = strlen(data);
    unsigned short checksum = calculateCRC(data, dataLength);

    printf("Calculated CRC: 0x%04X\n", checksum);

    // Simulating error by changing a bit
    // data[2] ^= 0x01; // Uncomment this line to introduce an error

    // Verify the received data
    unsigned short receivedChecksum;
    printf("Enter received CRC: ");
    scanf("%hx", &receivedChecksum);

    if (receivedChecksum == checksum) {
        printf("Data is error-free.\n");
    } else {
        printf("Data contains errors.\n");
    }

    return 0;
}
```

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


quotient:1011
transmitted frame:10111011000101101011
Enter transmitted freme:10111011000101101011
CRC checking


last remainder:0000000000000000

Received freme is correct
```

14) Write a program for congestion control using Leaky bucket algorithm.

```
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
```

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



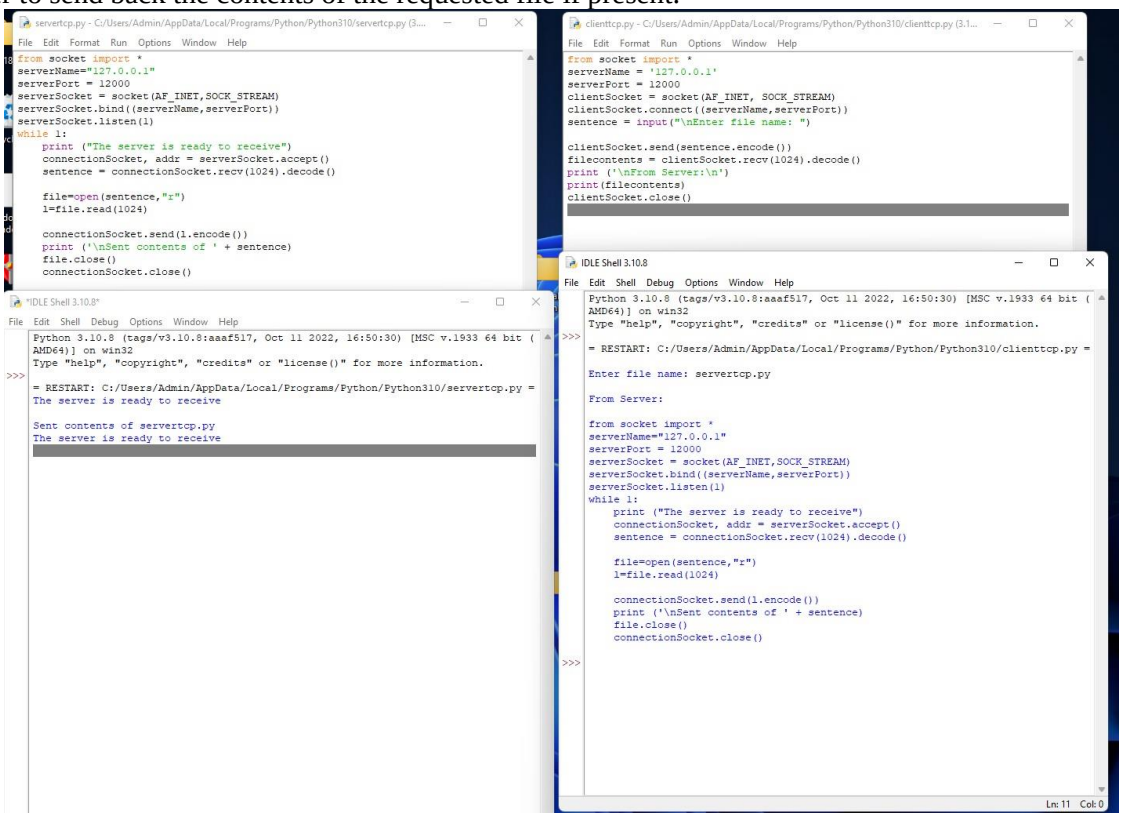
```
# serverudp.py
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")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

```
# 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()
```

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

Sent contents of  serverudp.py
```

```
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.
>>>
= 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()

>>>
```

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.



```
# 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)
    file.close()
    connectionSocket.close()
```

```
# 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()
```

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

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

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

Enter file name: servertcp.py

From Server:

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

    file=open(sentence,"r")
    l=file.read(1024)

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()

>>>
```