VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

JAI SHANKAR K S(1BM20CS062)

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
October-2022 to Feb-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **COMPUTER NETWORKS**" carried out by **JAI SHANKAR K S(1BM20CS062),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (20CS5PCCON)** work prescribed for the said degree.

Dr. Nandhini Vineeth
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

`

Index

Cycle 1

Experiment No 1

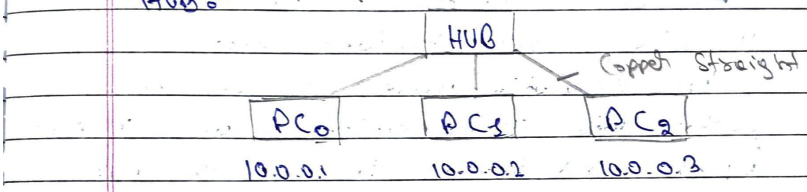
Date : 10/11/22
Page No :

Experiment:-1

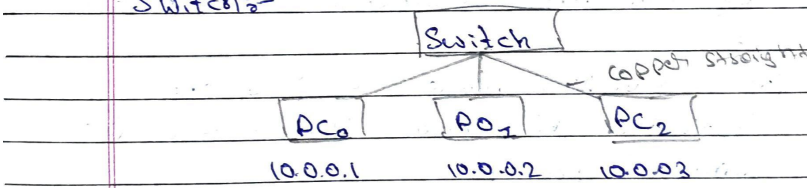
Aim:- Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.

Topology:-

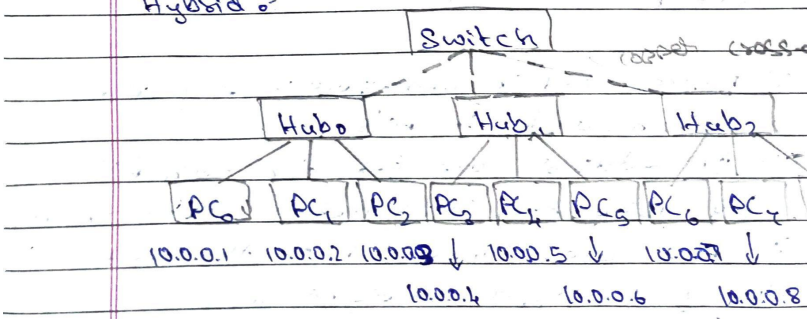
Hub:-



Switch:-



Hybrid:-



Date :
Page No :

Procedure:-

⇒ Hub

(i) Devices used in workspace :- A generic hub and seven PC's

(ii) IP address are set for each PC's by using the configuration tab in the PC.

(iii) PCs are connected to the Hub using a copper straight wire

(iv) In simulation mode, we can how packed data is transmitted and received by the end devices through the hub. message is transmited to all devices by only the destination device receives it.

(v) in real time mode, we ping the destination PC from the source PC's command prompt.

⇒ Switch

(i) Devices used in workspace :- A generic switch and 4 PC's

(ii) IP address are set for each PCs by using the configuration tab in the PC.

(iii) PCs are connected to the switch using a copper straight wire.

(iv) In simulation mode, PDU is established between two end devices, packet transfer can be seen.

(v) In real-time mode, we ping the destination PC from the source PC's command prompt.

Date :
Page No :

⇒ Hybrid

(i) Devices used in workspace:- One generic Switch, 3 generic Hubs and 12 PC's

(ii) IP address is set for all the PC's

(iii) PC's are connected to the Hub's using a copper straight wire and hubs are connected to the switch using a cross-over wire.

(iv) In Simulation mode, PDV is established between two PCs, packet transfer can be seen

(v) In real time mode, we ping the destination PC (can be any PC in the network) from the source PC's command prompt.

Observation:-

⇒ Hub:-

Learning outcome:-

The hub sends message to all the end devices, but the message is read only by the destination device (Note:- message is not sent to the source device)

Result:-

> Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Reply from 10.0.0.4: bytes=32 time=1ms TTL=128

Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0.MST TL=128

Ping statistic for 10.0.0.4:
Packets: Sent=4, Received=4, Lost=0(0% loss),
Approximate round trip times in milli-second
Minimum=0ms Maximum=1ms, Average=0ms

=> Switch
Learning Outcome:
Switch take time to establish connection with a device called learning time.
Message cannot be done until the green light is established. message is only sent to the destination device.

Result:-
AC> ping 10.0.0.3
Pinging 10.0.0.3 with 32bytes of data:

Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3
Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds
Minimum=0ms, Maximum=3ms, Average=0ms

=> Hybrid:

Learning Outcome:

The switch first sends the message to ~~off the~~ the hub to which the destination end device is connected then the hub sends to all the devices it is connected to then it is recieved by destination devices.

Result:-

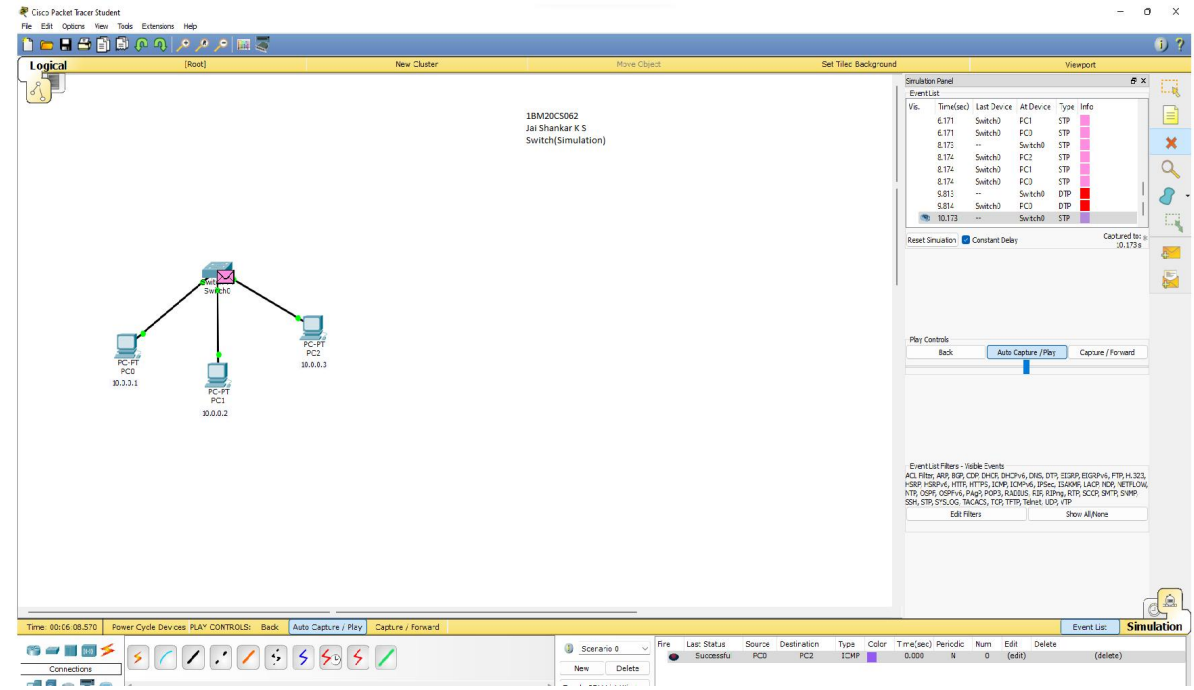
Pinging 10.0.0.12

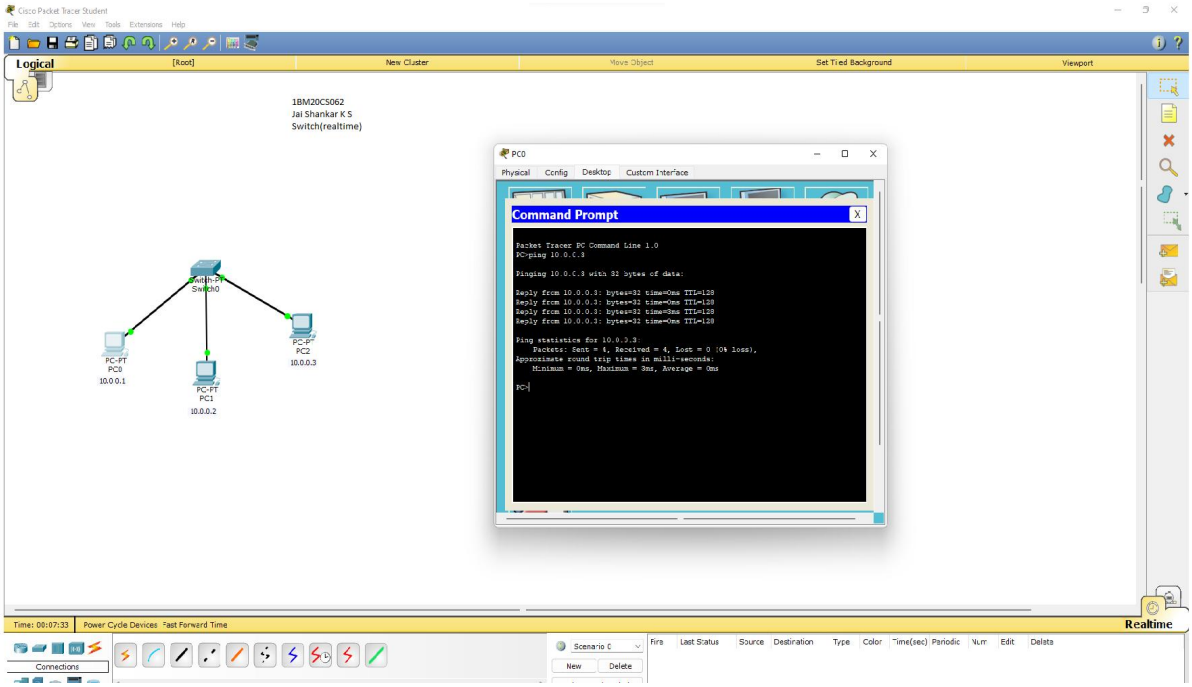
Pinging 10.0.0.12 with 32 bytes of data:

Reply from 10.0.0.12: bytes=32 time=0ms TTL=128
Reply from 10.0.0.12: bytes=32 time=0ms TTL=128
Reply from 10.0.0.12: bytes=32 time=0ms TTL=128
Reply from 10.0.0.12: bytes=32 time=0ms TTL=128

Ping Statistics for 10.0.0.12:
Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milliseconds:
Minimum=0ms, Maximum=0ms, Average=0ms

Snapshot of Output





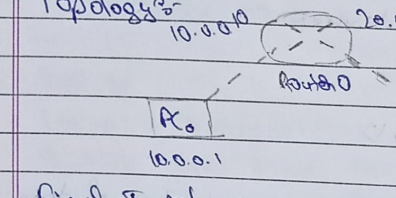
Experiment No 2

Date 25/8/22
Page No :

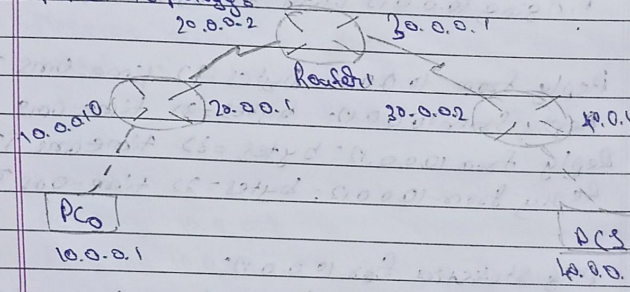
Experiment - 2

Aim: Configuring IP address to Router in packet tracer ; Explore the following messages ping Responces, Destination unreachable ; Request timed out, Reply.

Topology:



Final Topology:



Procedure:

=> For 1 Router and 2 PC's

(i) Using a copper cross-over wire connect the two PC's to the router

(ii) Configure each PC with a specific IP address. Also enter gateway for both PC's

(ii) Open Router CLI and do the following:-

```
Router > enable
Router# config t
Router(config) # interface fastethernet 0/0
Router(config-if) # ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# interface fastethernet 1/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# exit
Router# exit
```

(iv) After entering these commands the lights between PC and Router will become green

=> For 3-Router & 2-PC's

(i) Add 3 routers and 2 PC's to workspace and connect PC's with router by using copper crossover and connect Routers by using Serial PC.

(ii) Each PC is configured by a specific IP address and IP address is given by clicking specific & after that enter gateway for both PCs

(iii) Follow the following command in CLI of 3-routers after setting IP address & gateway

(ii) Open Router CLI and do the following:-

```
Router > enable
Router# config t
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# interface fastethernet 1/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# exit
Router# exit
```

(iv) After entering these commands the lights between PC and Router will become green

=> For 3-Router & 2-PC's

(i) Add 3 routers and 2 PC's to workspace and connect PC's with router by using copper crossover and connect Router's by using Serial PC.

(ii) Each PC is configured by a specific IP address and IP address is given by clicking specific PC after that enter gateway for both PCs

(iii) Follow the following command in CLI of 3-routers after setting IP address & gateway

(viii) Now Ping between any two PCs

Observation:-

=> For Single Router:-
Learning Outcome:-
We used router to set a connection between two end devices at first it shows "Request timed out". Later Pinging again shows the following result

Result:-

Ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data

Reply from 20.0.0.2 bytes=32 time=0ms TTL=127
Reply from 20.0.0.2 bytes=32 time=0ms TTL=127
Reply from 20.0.0.2 bytes=32 time=0ms TTL=127
Reply from 20.0.0.2 bytes=32 time=0ms TTL=127

Ping Statistics for 20.0.0.2:
Packets Sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip time in milli-seconds.
Minimum=0ms, Maximum=0ms, Average=0ms.

=> For three routers:-
Outcome:-
(i) · Initially we get Request timed out then later we get the result.

Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

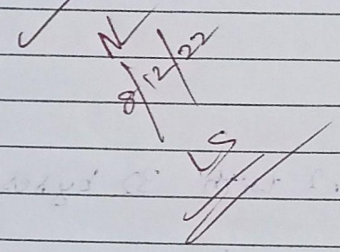
Reply from 20.0.0.1 : bytes=32 Time=1ms TTL=12
Reply from 40.0.0.1 : bytes=32 Time=8ms TTL=128
Reply from 20.0.0.1 : bytes=32 Time=6ms TTL=128
Reply from 20.0.0.1 : bytes=32 Time=11ms TTL=128

Ping statistics for 40.0.0.1

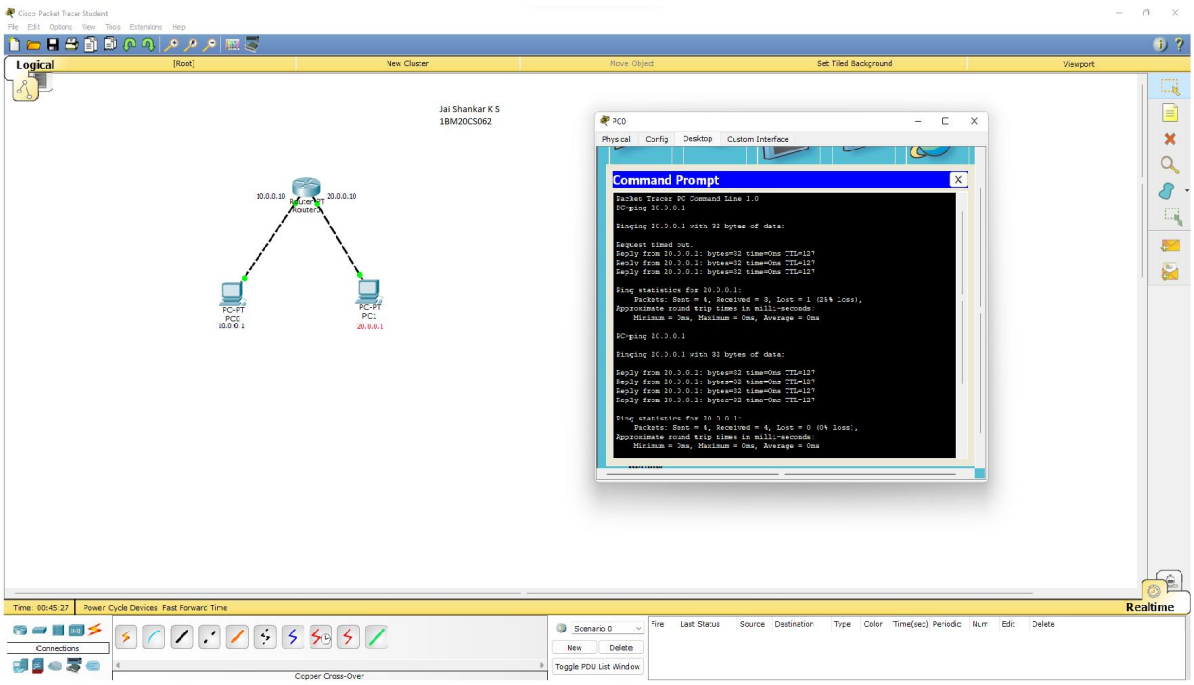
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)

Approximate round trip time in milli seconds

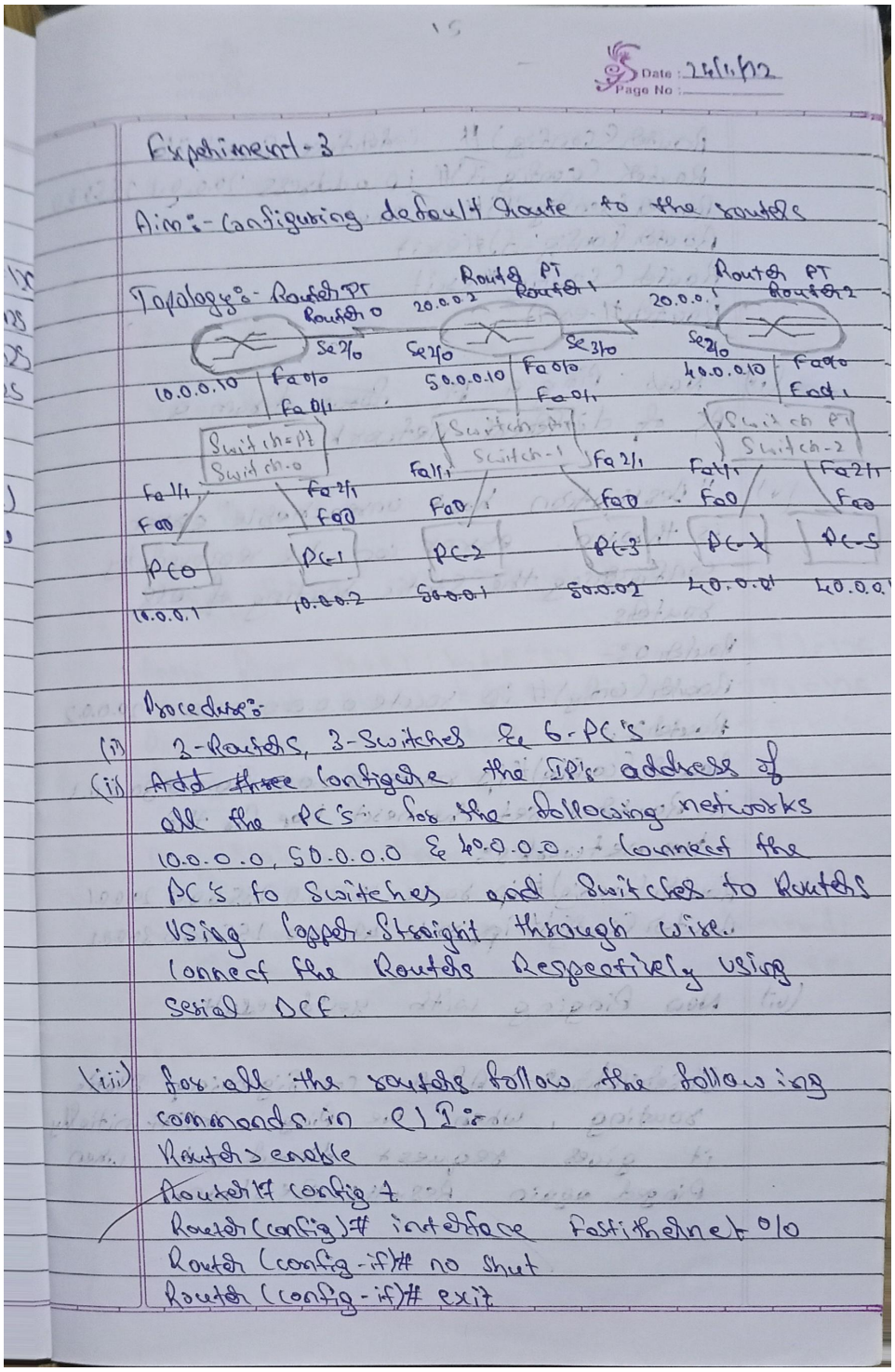
Minimum = 6ms, Maximum = 11ms Average = 7ms



Snapshot of Output



Experiment No 3



Experiment - 3

Aim:- Configuring default route to the routers

Topology:- Router PT Router 0 | Router PT Router 1 | Router PT Router 2

10.0.0.10 | Se2/0 | Fa0/0 | Fa0/1 | Se2/0 | 20.0.0.2 | Se3/0 | 50.0.0.10 | Fa0/0 | Fa0/1 | Se2/0 | 20.0.0.1 | 40.0.0.10 | Fa0/0 | Fa0/1

Switch-PT Switch-0 | Switch-PT Switch-1 | Switch-PT Switch-2

Fa1/1 | Fa2/1 | Fa0 | Fa0 | Fa1/1 | Fa2/1 | Fa0 | Fa1/1 | Fa0 | Fa0

PC0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5

10.0.0.1 | 10.0.0.2 | 50.0.0.1 | 50.0.0.2 | 40.0.0.1 | 40.0.0.2

Procedure:-

(i) 3-Routers, 3-Switches & 6-PC's

(ii) ~~Add three~~ Configure the IP's address of all the PC's for the following networks 10.0.0.0, 50.0.0.0 & 40.0.0.0. Connect the PC's to Switches and Switches to Routers using copper Straight through wire. Connect the Routers Respectively using serial DCE.

(iii) For all the routers follow the following commands in CLI:-

Router> enable

Router# config t

Router(config)# interface FastEthernet 0/0

Router(config-if)# no shut

Router(config-if)# exit

Router(config)# interface Serial 2/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# exit
Router# exit.

(iv) Now Ping a PC from a PC of different network

(v) "Destination host unreachable" error is thrown. error can be removed by configuring the static routing of all routers.

Router 0:-
Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router 1:-
Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1
in Router 1 set the next hop for the two networks:-
Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.1

(vi) Now Pinging with yeild result.

Observation:- After configuring static routing, when we Ping initially it gives request timed out. when Pinged again Results are shown.

Results:-

Ping 50.0.0.1

Pinging 50.0.0.1 with 32 bytes of data.

Request timed Out
Reply from 50.0.0.1 : bytes = 32, Time = 1ms, TTL = 126
Reply from 50.0.0.1 bytes = 32, Time = 2ms, TTL = 126
Reply from 50.0.0.1 bytes = 32, Time = 3ms, TTL = 126

Ping 50.0.0.1

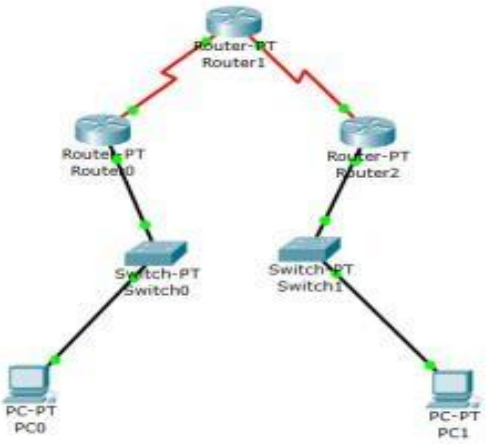
Pinging 50.0.0.1 with 32 bytes of data:

Reply from 50.0.0.1 : bytes = 32 time 2ms TTL = 126
Reply from 50.0.0.1 bytes = 32 time 3ms TTL = 126
Reply from 50.0.0.1 bytes = 32 time 2ms TTL = 126
Reply from 50.0.0.1 bytes = 32 time = 1ms TTL = 126

Ping Statistics for 50.0.0.1
Packets Sent = 4, Recieved = 4, Lost = 0 (0% Loss)
Approximate round trip time in milliseconds
Minimum = 2ms, Maximum = 10ms, Average 5ms

8/12/22

Snapshot of Output




```
PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

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

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: Destination host unreachable.
Request timed out.
Reply from 40.0.0.1: Destination host unreachable.
Reply from 40.0.0.1: Destination host unreachable.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```

```
PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=14ms TTL=125
Reply from 40.0.0.2: bytes=32 time=11ms TTL=125
Reply from 40.0.0.2: bytes=32 time=11ms TTL=125
Reply from 40.0.0.2: bytes=32 time=11ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 14ms, Average = 11ms

PC>
```

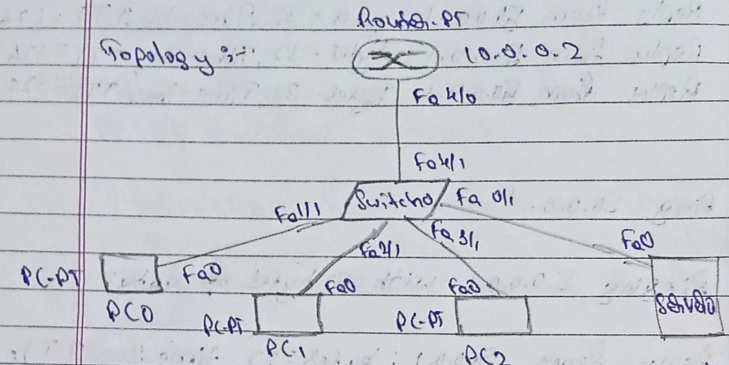
Experiment No 4

Date :
Page No :

Experiment - 4

Aim :- Configuring DHCP within a LAN in a packet Tracer.

Topology :-



Procedure :-

(i) Add :- 3 - PC's, 1 - Server, 1 - Switch, & 1 - Router

(ii) Configure the server with IP address and gateway. (IP address - 10.0.0.1) (Gateway 10.0.0.2)

(iii) In Router, follow the following commands in CLI :-

interface fastethernet 4/0
ip address 10.0.0.1 255.0.0.0

(iv) In Server, go to services and enable service

Default Gateway = 10.0.0.2
DSN Server = 10.0.0.1
Start IP : 10.0.0.3
TFTP Server : 10.0.0.1

max. users : 8
Save it.

(v) for all PC's → Desktop → IP configuration
Shift from static to DHCP

(vi) Now all configuration is done Ping any PC from a PC to see the result.

Observation:

Learning Outcome:
Server automatically provides IP address for the PC's

Result:-
Ping 10.0.0.5

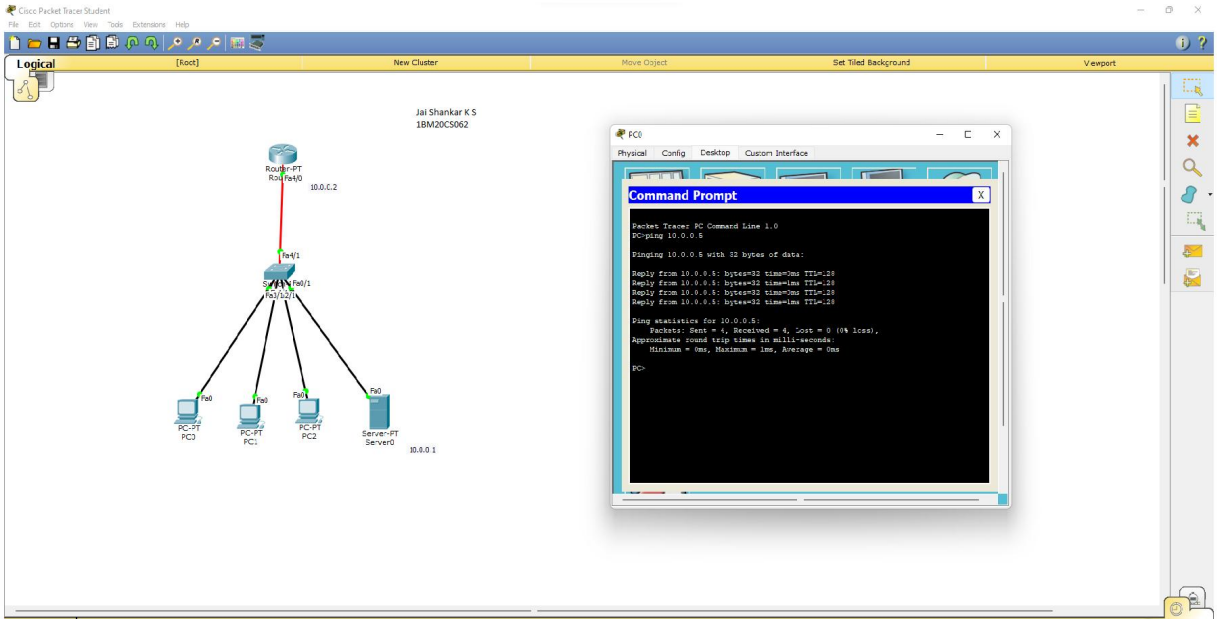
Pinging 10.0.0.5 with 32 bytes of data:

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

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

8/12/22

Snapshot of Output



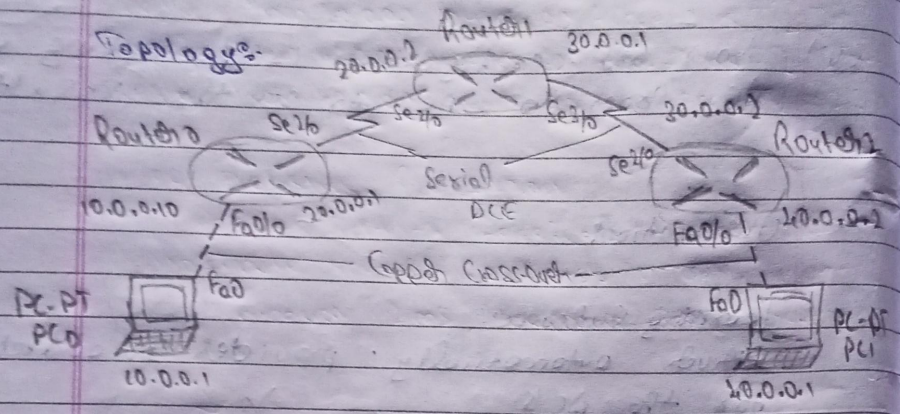
Experiment No 5

Date : 8/12/22
Page No :

Experiment - 5

Aim:- Configuring RIP Routing Protocol in Routers

Topology:-



Procedure:-

(i) Add 3-Routers and 2-PC's to the workspace

(ii) Configue the PC with IP address respectively PC_0 - 10.0.0.1 & PC_1 - 40.0.0.1

(iii) Connect the PC's to the Routers using copper crossover cable and Router to Router using Serial DCE
PC0 - Router0 PC1 → Router2
Router0 → Router1 Router1 → Router2

(iv) Add gateway for both PC's
PC0 - 10.0.0.10 PC1 - 40.0.0.2

(iv) Interface PC - Router => for all
interface fastethernet 0/0
ip address 10.0.0.10 255.0.0.0
no shut

Interface Router - Router => for all with clock
interface serial 2/0 symbol
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut

Remove clock rate for others

(v) Now follow the following commands => for all routers
router rip
network 10.0.0.0
network 20.0.0.0
exit
Use respective network address for all routers.

Observations:

Learning Outcomes:
Using RIP protocol routing is easy when many routers are involved.

Result :-

Ping 10.0.0.1

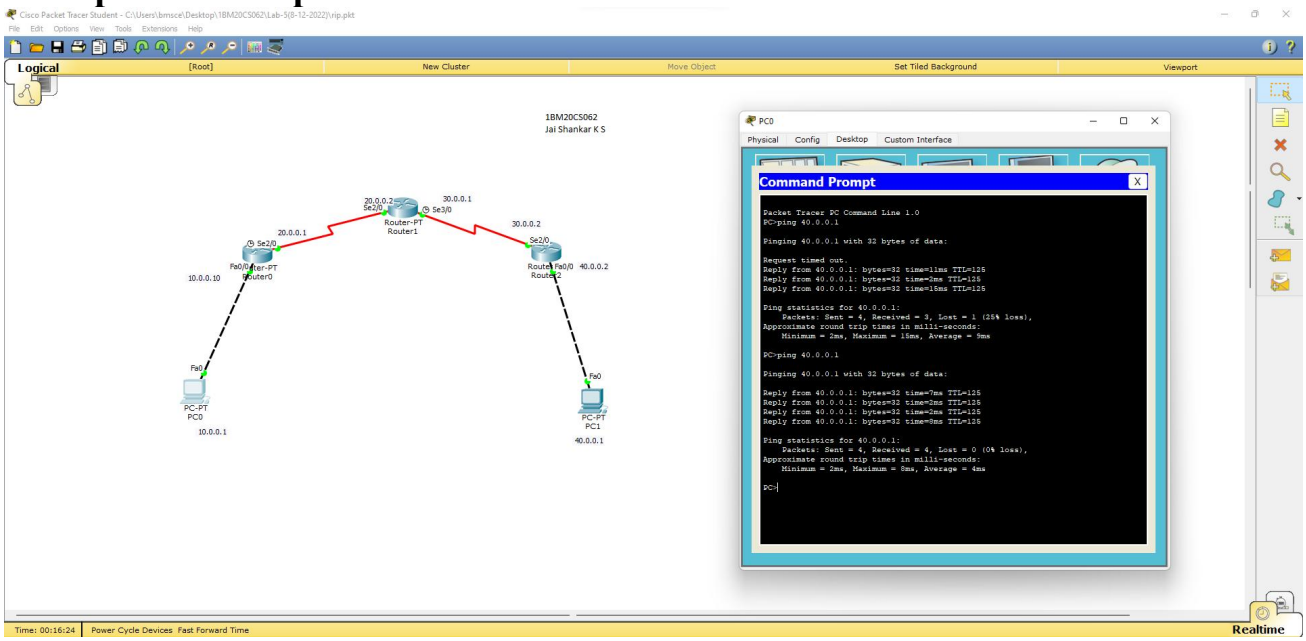
Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1 : bytes=32 time=1ms TTL=128
Reply from 10.0.0.1 : bytes=32 time=1ms TTL=128
Reply from 10.0.0.1 : bytes=32 time=1ms TTL=128
Reply from 10.0.0.1 : bytes=32 time=8ms 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 = 1ms, Maximum = 8ms, Average = 4ms

29/12/22

Snapshot of Output



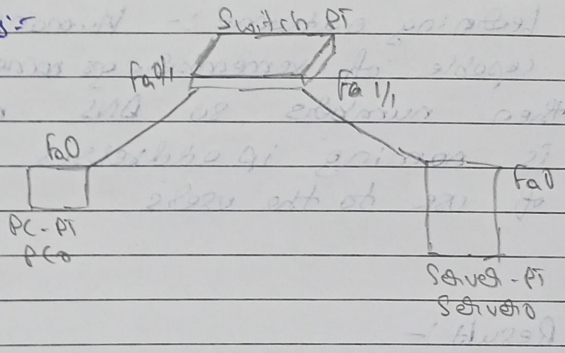
Experiment No 6

Date: 15/12/22
Page No:

Experiment-6

Aim:- Demonstration of WEB server and DNS using Packet Tracer.

Topology:-



Procedure:-

(i) Add a PC, switch and a server to the workspace
(ii) Connect Switch-PC & Switch-Server using copper straight-through cable
(iii) Add IP address for PC & Server as 10.0.0.1 & 10.0.0.10.
(iv) In the server -> Services -> DNS, Turn it on
add a server name bmsce.in
add IP address 10.0.0.10
save it.
(v) In PC -> desktop -> web-browser search for the server0 ip address the server web pages are shown Some Results are shown when searched with the name saved in DNS.

(vi) Now add new webpage mycv.html to the http and link it to the index.html

Observation :-

Learning outcome :- Humans are capable of remembering names better than numbers, so DNS helps in naming ip addresses for ease of use to the users.

Result :-

PC0

Web browser

< > URL http://bmsce.in Go Stop

BMSCE

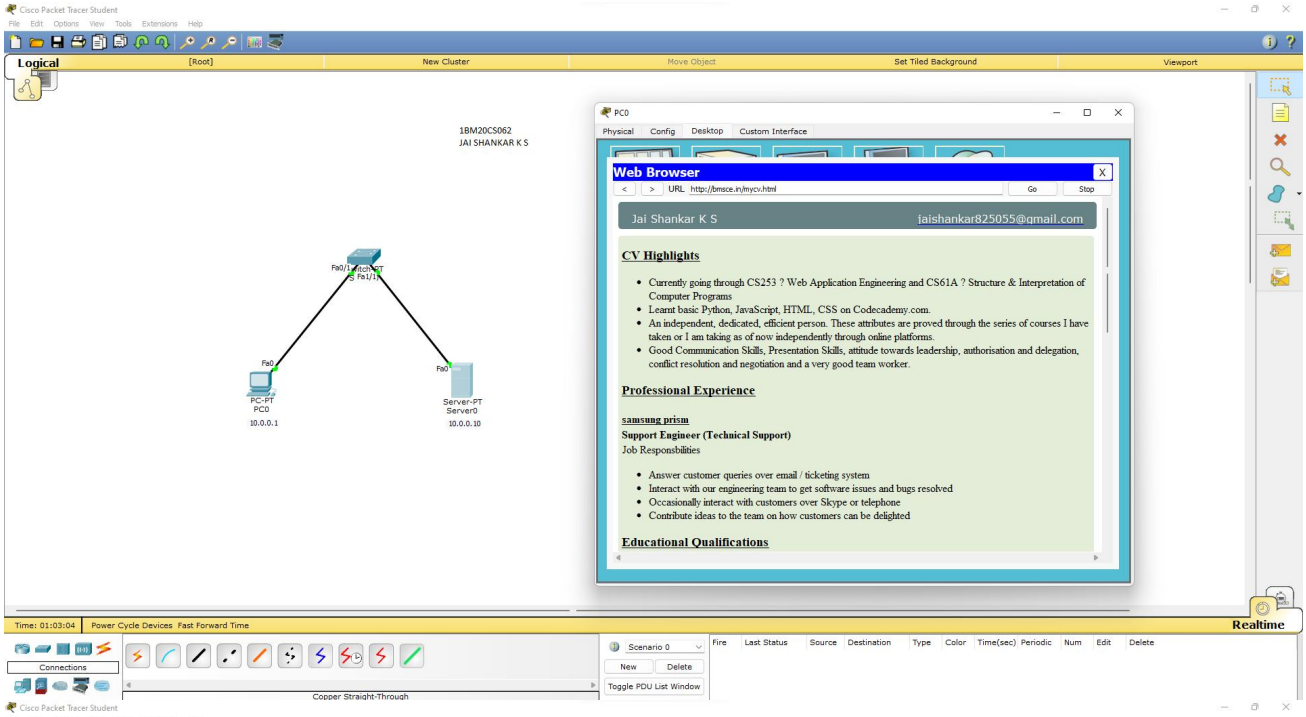
Welcome to BMSCE

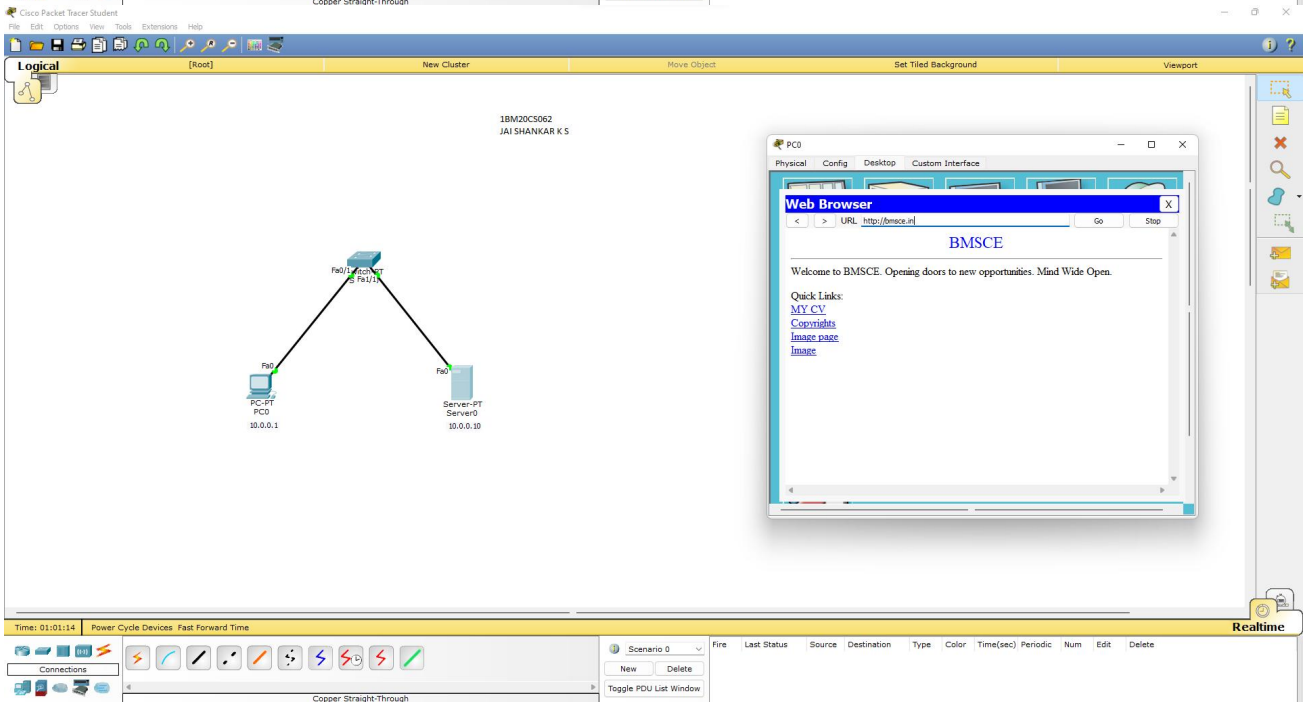
Quick links:

My CV

Copyrights

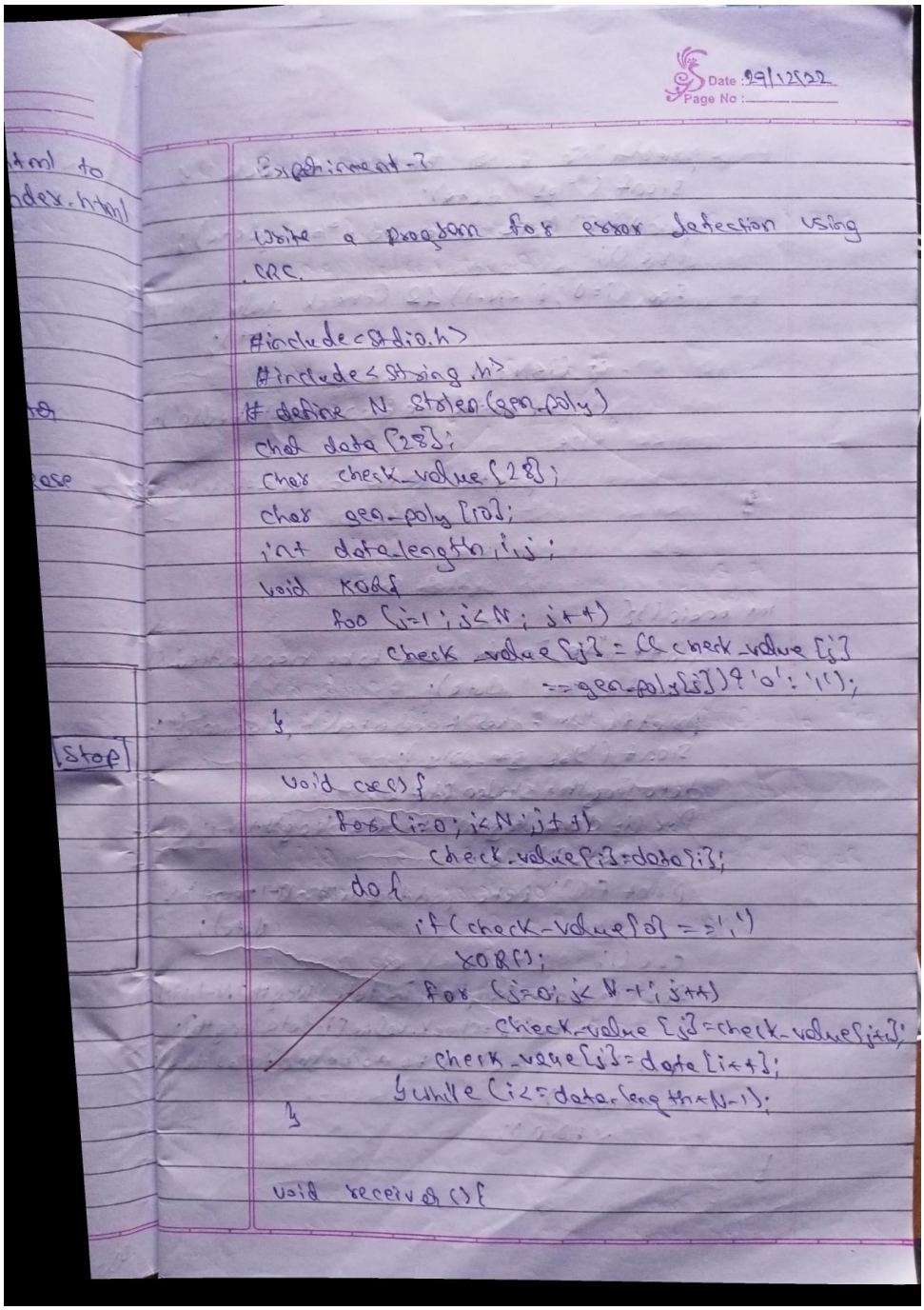
Snapshot of Output





Cycle 2

Experiment No 7



Date 29/12/22
Page No :

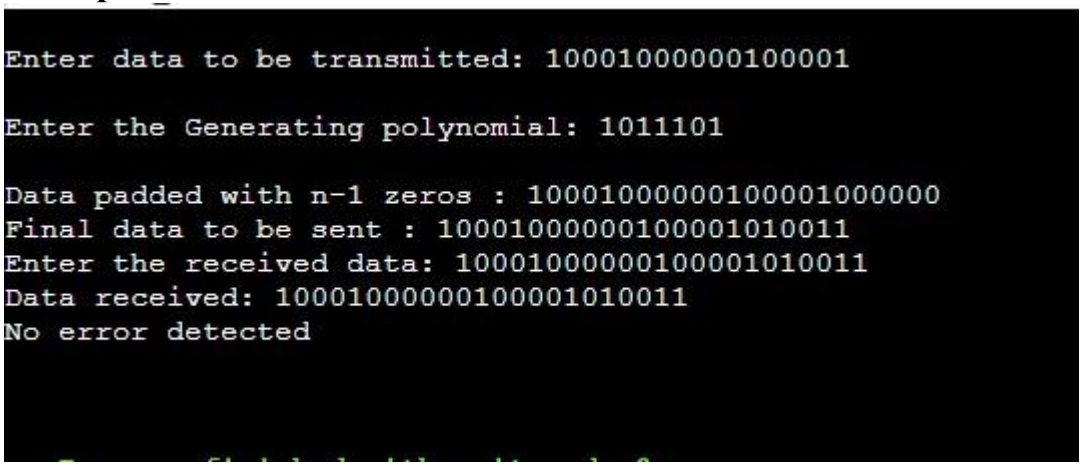
Experiment-7

Write a program for error detection using CRC.

```
#include<stdio.h>
#include<string.h>
#define N strlen(gen_poly)
char data[28];
char check_value[28];
char gen_poly[10];
int data_length,i,j;
void XOR(){
    for(j=1;j<N;j++)
        check_value[j]=(check_value[j]
            ==gen_poly[j])?'0':'1';
}

void 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);
}

void receiver(){
```

```
printf("\n Enter the received data :");
scanf("%s", data);
printf("Data received : %s.", data);
rec();
for(i=0; i<N-1) && (check_value[i]=='1');
    i++);
    if (i<N-1)
        printf("\n Error detected\n");
    else
        printf("\n No Error detected\n");
}

int main(){
    printf("\n Enter the data to be transmitted:");
    scanf("%s", data);
    printf("\n Enter the generator polynomial:");
    scanf("%s", gen_poly);
    data_length = strlen(data);
    for(i=data_length; i<data_length+N-1; i++)
        data[i]='0';
    printf("\n Data padded with N-1 zeros: %s",
                                    data);
    crc();
    for(i=data_length; i<data_length+N-1; i++)
        data[i]=check_value[i-data_length];
    printf("\n final data to be sent : %s", data);
    receiver();
    return 0;
}
```

Output – Screen shot



Experiment No 8

```
#include <bits/stdc++.h>
#include <limits.h>
#include <stdio.h>
using namespace std;
#define V 4
int minDistance(int dist[], bool sptSet[])
{
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++)
        if (sptSet[v] == false && dist[v] <= min)
            min = dist[v], min_index = v;
    return min_index;
}
void printSolution(int dist[])
{
    printf("Vertex \t\t Distance from Source\n");
    for (int i = 0; i < V; i++)
        printf("%d \t\t %d\n", i, dist[i]);
}
void dijkstra(int graph[V][V], int src)
{
    int dist[V];
    bool sptSet[V];
    for (int i = 0; i < V; i++)
        dist[i] = INT_MAX, sptSet[i] = false;
    dist[src] = 0;
    for (int count = 0; count < V - 1; count++)
    {
        int u = minDistance(dist, sptSet);
        sptSet[u] = true;
        for (int v = 0; v < V; v++)
            if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX && dist[u] + graph[u][v] < dist[v])
                dist[v] = dist[u] + graph[u][v];
    }
    printSolution(dist);
}
int main()
{
    int graph[V][V];
    cout << "Enter the graph " << endl;
    for (int i = 0; i < V; i++)
    {
        for (int j = 0; j < V; j++)
            cin >> graph[i][j];
    }
    dijkstra(graph, 0);
    return 0;
}
```

Output – Screen shot

```
Enter no. of vertices: 4
Enter graph in matrix form:
0 3 5 7

9 0 3 4
2 5 0 8
6 4 3 0

Enter source : 1

Distance from source 1 to vertex1 is 0
Distance from source 1 to vertex2 is 3
Distance from source 1 to vertex3 is 5
Distance from source 1 to vertex4 is 7
No negative weight cycle exists
```

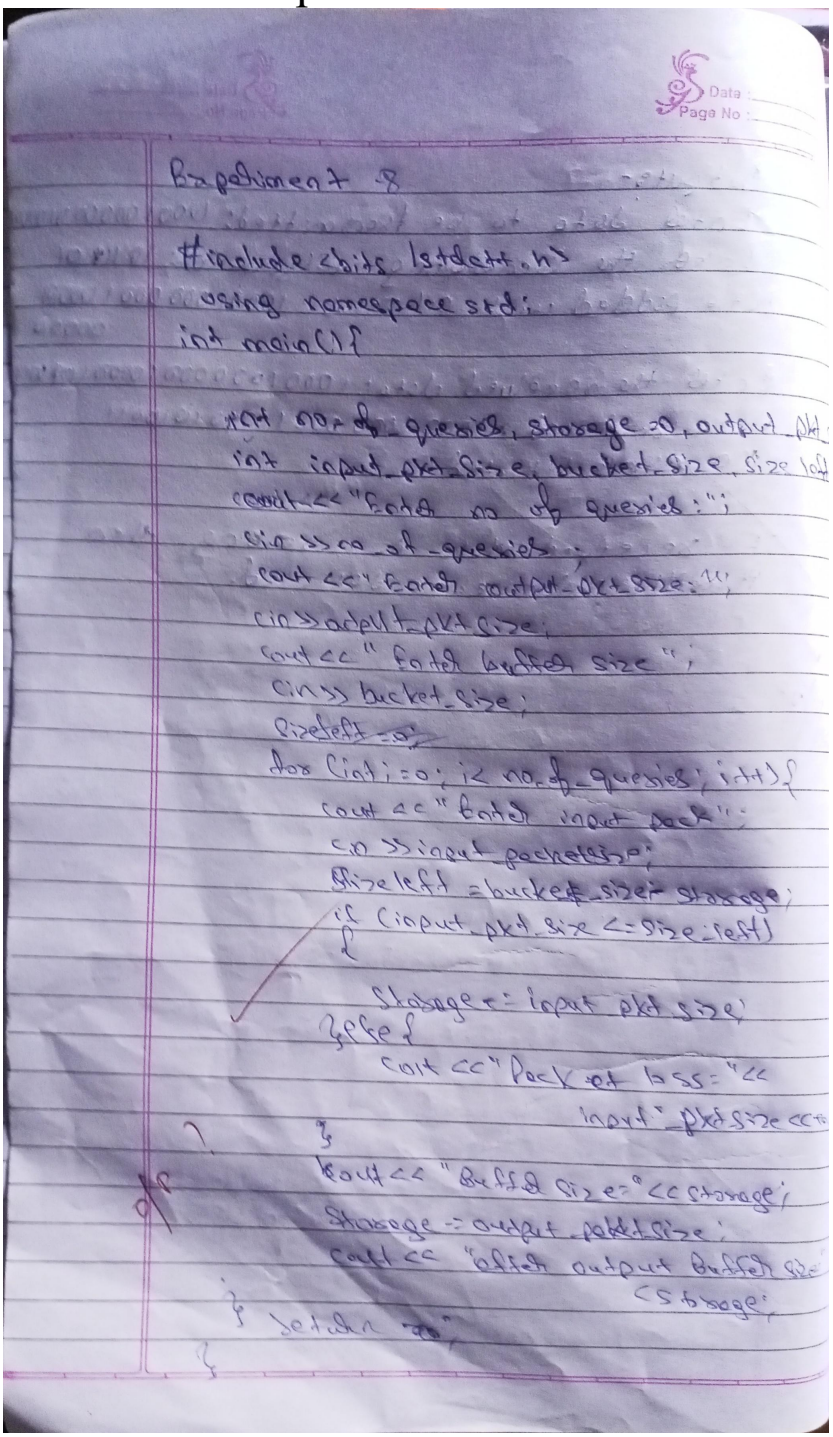
Experiment No 9

```
#include <bits/stdc++.h>
using namespace std;
int Bellman_Ford(int G[100][100] , int V, int E, int edge[100][2]) {
int i,u,v,k,distance[100],S,flag=1;
for(i=0;i<V;i++)
distance[i] = 1000 ;
cout<<"\nEnter source : ";
cin>>S;
distance[S-1]=0;
for(i=0;i<V-1;i++){
for(k=0;k<E;k++){
u = edge[k][0];
v = edge[k][1];
if(distance[u]+G[u][v] < distance[v])
distance[v] = distance[u] + G[u][v];
}
}
for(k=0;k<E;k++){
u = edge[k][0];
v = edge[k][1] ;
if(distance[u]+G[u][v] < distance[v])
flag = 0 ;
}
if(flag)
for(i=0;i<V;i++)
cout<<"\nDistance from source "<<S<<" to vertex"<<i+1<<" is "<<distance[i];
return flag;
}
int main()
{
int V,edge[100][2],G[100][100],i,j,k=0;
cout<<"Enter no. of vertices: ";
cin>>V;
cout<<"Enter graph in matrix form:\n";
for(i=0;i<V;i++)
for(j=0;j<V;j++)
{
cin>>G[i][j];
if(G[i][j]!=0)
edge[k][0]=i,edge[k++][1]=j;
}
if(Bellman_Ford(G,V,k,edge))
cout<<"\nNo negative weight cycle exists\n";
return 0;
}
```

Output – Screen shot

```
Enter no. of vertices: 4
Enter graph in matrix form:
0 3 5 7
9 0 3 4
2 4 0 8
6 4 3 0

Enter source : 1

Distance from source 1 to vertex1 is 0
Distance from source 1 to vertex2 is 3
Distance from source 1 to vertex3 is 5
Distance from source 1 to vertex4 is 7
No negative weight cycle exists
```

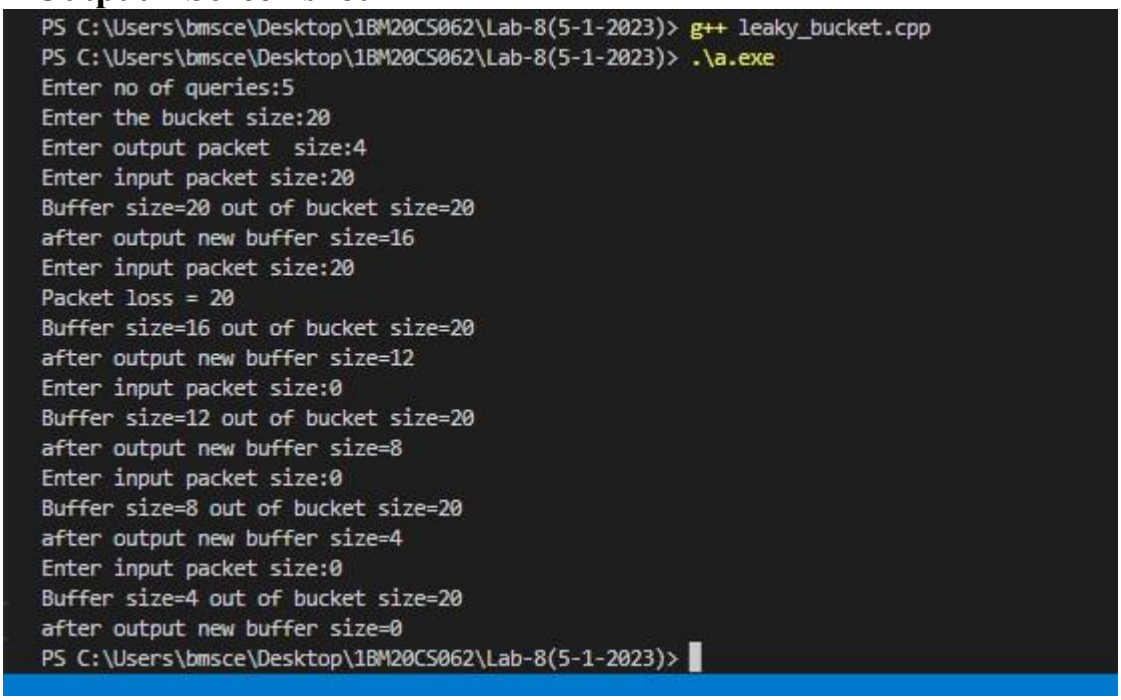
Experiment No 10



Expt no 8

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int no_of_queries, storage=0, output_pkt_
    int input_pkt_size, bucket_size, size_left
    cout<<"Enter no of queries:";
    cin>>no_of_queries;
    cout<<"Enter output_pkt_size:";
    cin>>output_pkt_size;
    cout<<"Enter bucket size";
    cin>>bucket_size;
    sizeleft=0;
    for(int i=0; i<no_of_queries; i++){
        cout<<"Enter input pack";
        cin>>input_packetsize;
        sizeleft=bucket_size-storage;
        if(input_pkt_size<=size_left)
        {
            storage+=input_pkt_size;
        }else{
            cout<<"Packet loss="<<
                input_pkt_size<<
        }
        cout<<"Buffer size="<<storage;
        storage-=output_pkt_size;
        cout<<"after output Buffer size
            <<storage;
    }getch();
}
```

Output – Screen shot



Experiment No 11

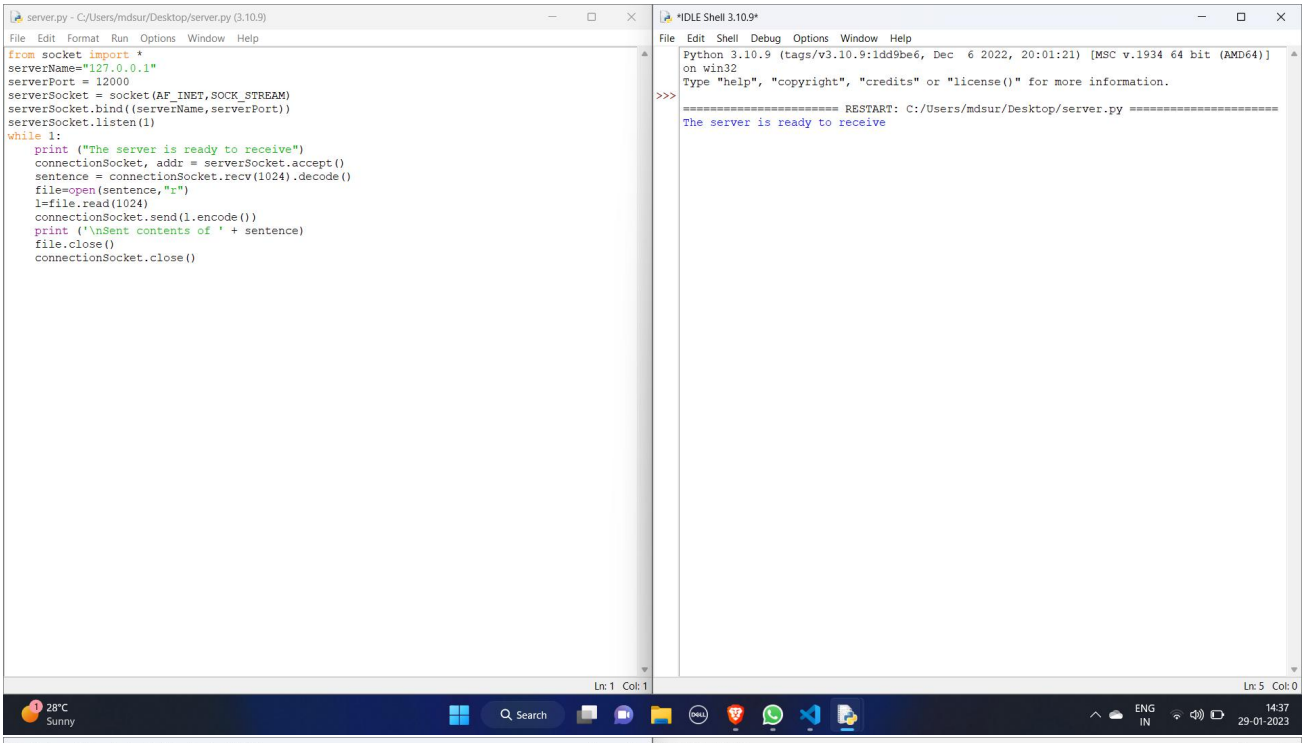
Clint.py

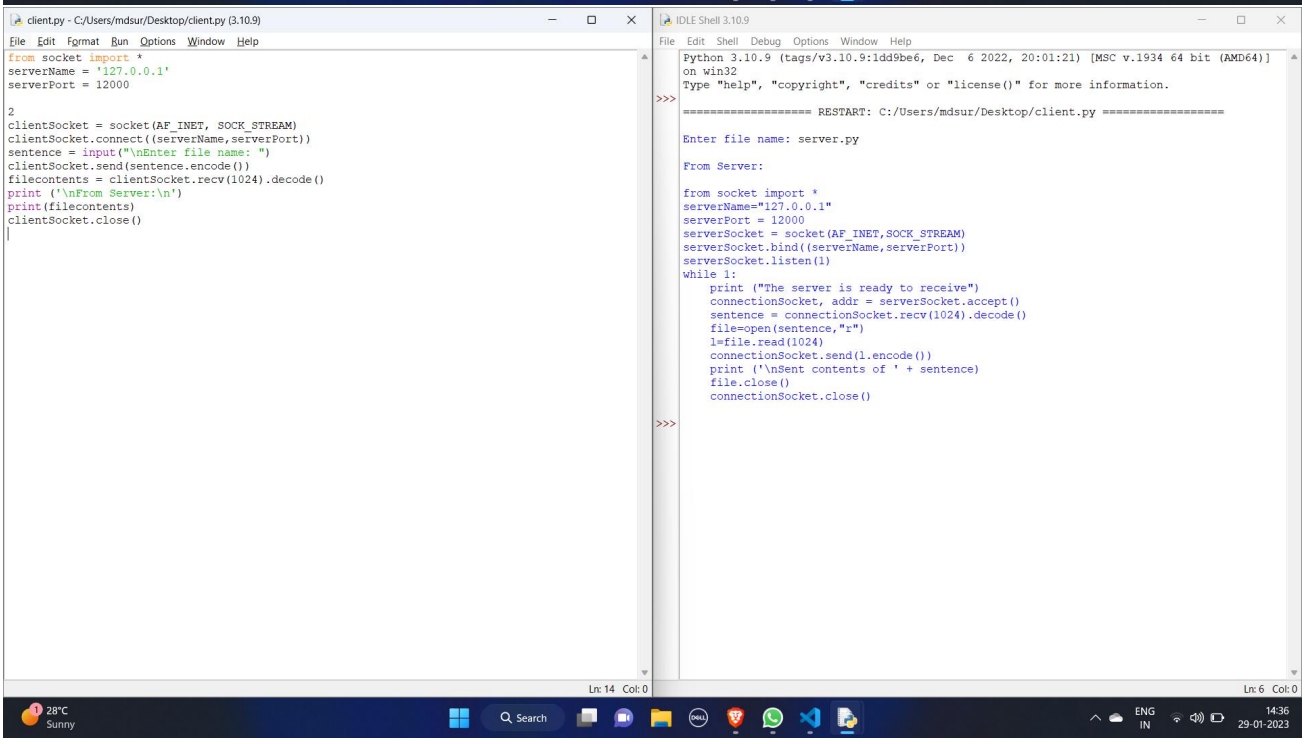
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000

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()
```

Server.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()
```

Output – Screen shot





Experiment No 12

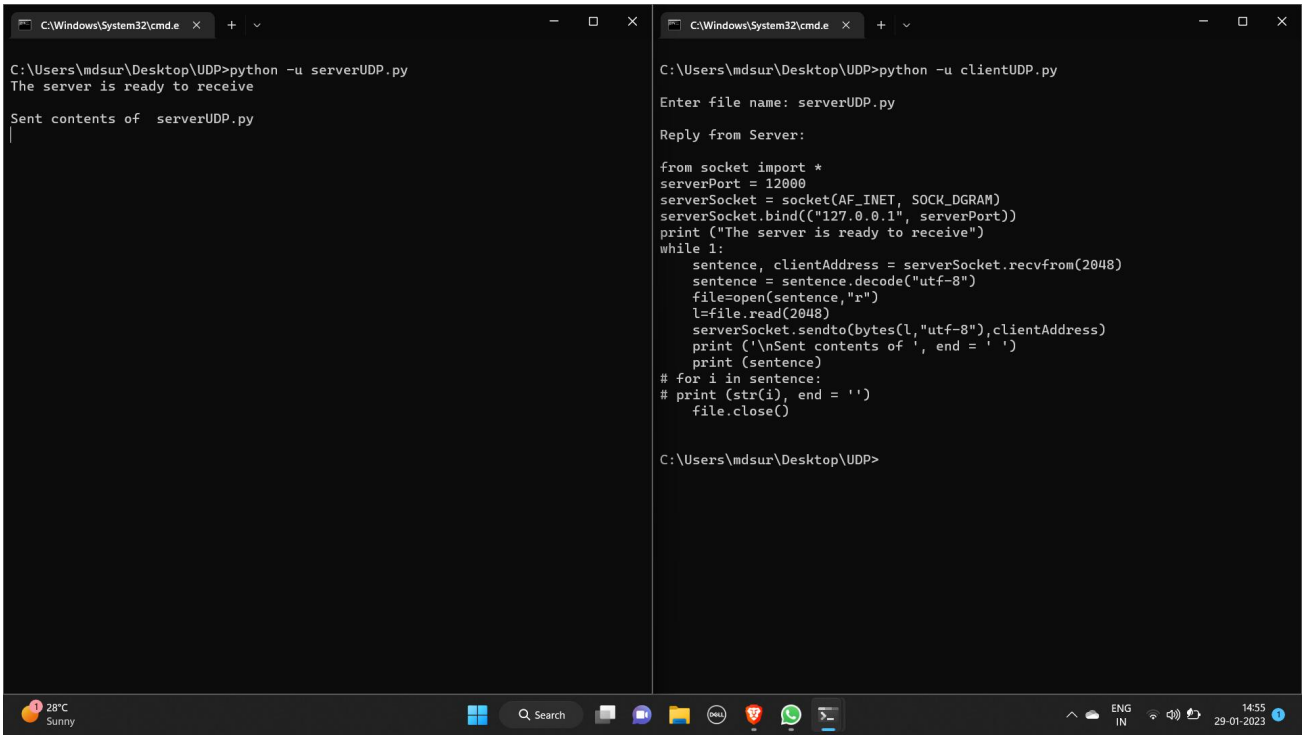
Clint.py

```
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()
```

Server.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()
```

Output – Screen shot


C:\Windows\System32\cmd.e
C:\Users\mdsur\Desktop\UDP>python -u serverUDP.py
The server is ready to receive
Sent contents of serverUDP.py
C:\Windows\System32\cmd.e
C:\Users\mdsur\Desktop\UDP>python -u 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")
l=file.read(2048)
serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
C:\Users\mdsur\Desktop\UDP>
28°C
Sunny
Search
ENG
IN
14:55
29-01-2023