VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

COURSE TITLE

Submitted by

PRATEEK GUPTA (1BM20CS111)

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 **PRATEEK GUPTA(1BM20CS111),** 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 **Course Title - (20CS5PCCON)** work prescribed for the said degree.

M Lakshmi Neelima	**Dr. Jyothi S Nayak**
Assistant professor Department of CSE BMSCE, Bengaluru	Professor and Head Department of CSE BMSCE, Bengaluru

`

Index

10|11|22

EXP-1

Aim!- Implementing star topology using hubs and switches.

Procedure :- 1) End devices are connected through the hub.

2) Hubs are connected by using switch.

3) IP address of end devices are configured.

4) ~~A simple PDU file~~ Connection between all devices are checked.

5) A simple PDU file is transmitted between a source and destination.

Result!- Message Transmission between any devices is successful.

Observation :- i) PDU is first sent to the hub

ii) Hub will broadcast to all the device connected to it, if any of the receiving device is destination it will read message otherwise discard it.

iii) Initially switch will broadcast to all the ports. Later fill the detail of ip address & ports in a table. And later on this table is used to broadcast a message to particular port.

-> Hub :- when source sends a packed in network the hub source the packet and ends broadcast over the network, i.e, it sends data to all the end devices in network and node whereit matches.

Result :-

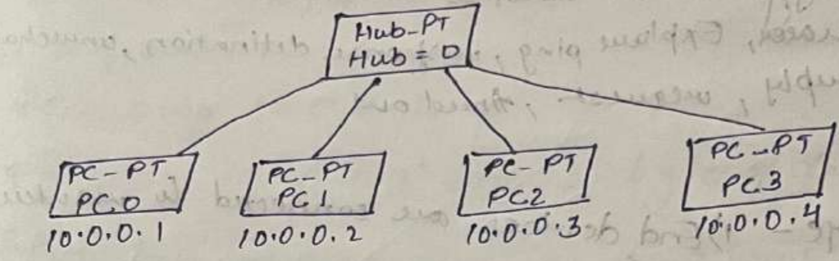
```
PC > ping 10.0.0.1
Pinging 10.0.0.1 with 32 byte of data
Reply from 10.0.0.1 : byte=32 , time=0ms
Reply from 10.0.0.1 : byte = 32   times=0ms
Reply from 10.0.0.1 : byte = 32   times=0ms
Reply from 10.0.0.1 :  byte = 32   time =0 ms
Packet sent=4 , received=4, Lost=0
```

-> Switches :- When source device send a message to the switch once a connection is established which takes some time called learning time the switch received the packed.

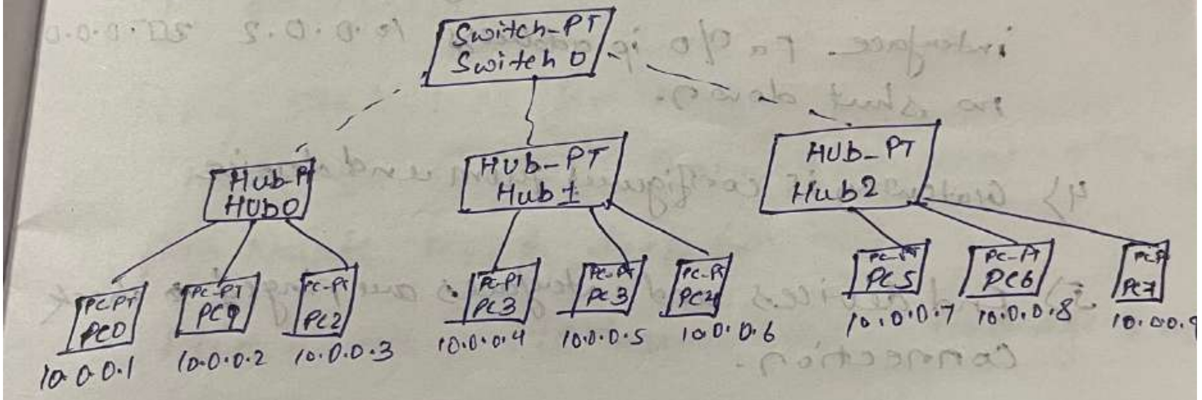
Result :-

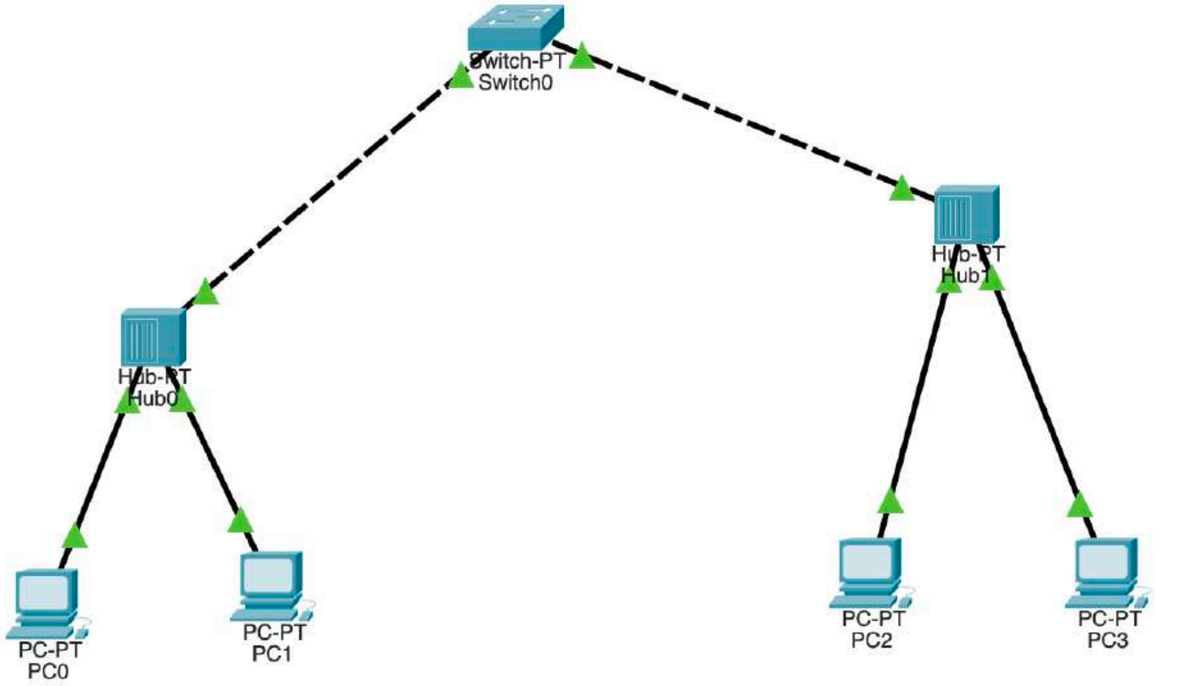
```
PC > Ping 10.0.0.1
Pinging 10.0.0.1 with 32 byte of data
Reply from 10.0.0.1 : byte=32 . time=0ms
Reply from 10.0.0.1 : byte=32   time=0ms
Reply from 10.0.0.1 : byte = 32   time=0ms
Reply from 10.0.0.1 : byte = 32   time=0 ms
Ping statistics for 10.0.0.3
Packet sent=4 , Received=4, Lost=0.
```

Using Hub:-



Using Switch:-




Switch-PT
Switch0
Hub-PT
Hub0
Hub-PT
Hub1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3

17/11/22

Exp-2

Aim — Configure IP address to router in packet tracer, Explain ping, response, detination, unreachable reply, request, timed out.

Proceduye — 1) End devices are connected to router

2) IPaddress is configured to end devices

3) Config IP Address & subnet mask using commands enable, config terminal, interface Fa 0/0 ip address 10.0.0.2 255.0.0.0 no shut down.

4) Gateway is configured from end device

5) End devices and interfaces are pinged to check connection.

Topology — Star Topology.

Result — Succesfully pinged end devices.

Observation — When we configure both end devices and router with appropriate ip address and by configuring subnetmask of interface of router as 255.0.0.0 and gateway of PC0 is set as 10.0.0.2 which is of fa0/0 interface followed by same for PC1.

Then we could succesfully Ping.

when gateway of end devices is not configured then we get request time out.

After configure):-

PC> ping 10.0.0.1

Pinging 10.0.0.1 with 32 byte of data

Reply from 10.0.0.1 byte=32 time 6ms TTL=128
Reply from 10.0.0.1 byte=32 time=2ms TTL=128
Reply from 10.0.0.1 byte=32 time=5ms TTL=128
Reply from 10.0.0.1 byte=32 time=3ms TTL=120

Ping statistics for 10.0.0.1

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

Approximate round trip (in ms)

Min=2ms Maximum=6ms Average=4ms

Before configure:-

PC > Ping 10.0.0.1

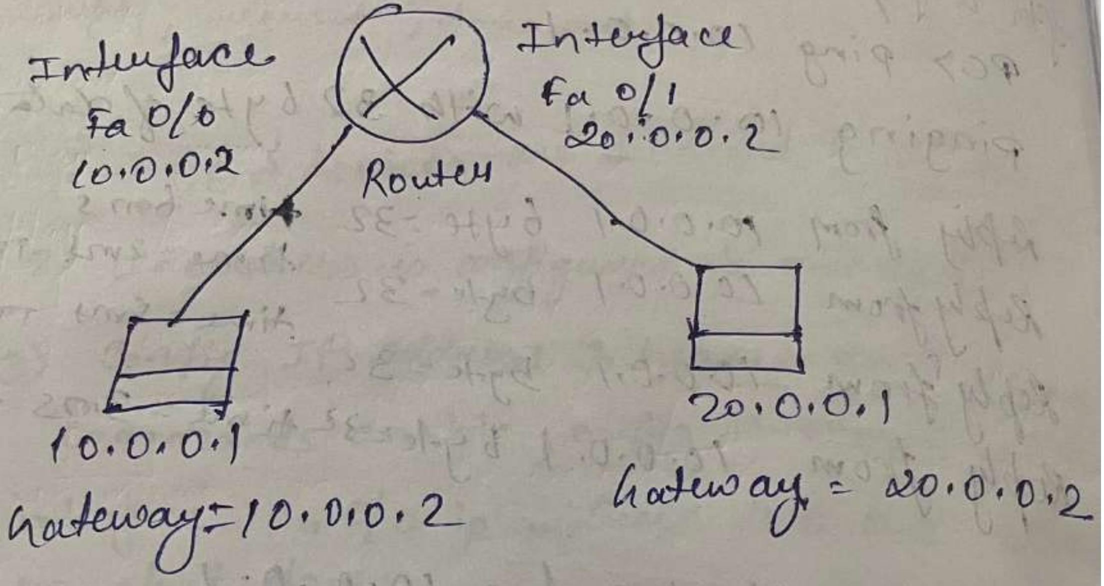
~~PC>~~ pinging 10.0.0.1 with 32 byte of data

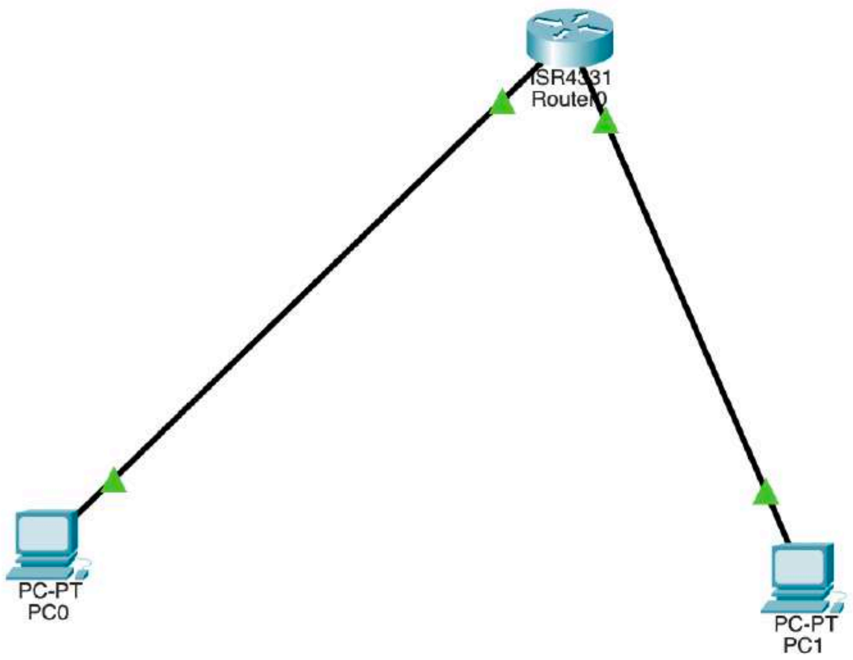
Request timed out
Request timed out
Request timed out
Request timed out.

Ping statistics for 10.0.0.1

Packet: Sent=4, Received=0, Lost=4 (100% loss

Topology :-


Interface
Fa 0/0
10.0.0.2
Router
Interface
Fa 0/1
20.0.0.2
10.0.0.1
Gateway = 10.0.0.2
20.0.0.1
Gateway = 20.0.0.2

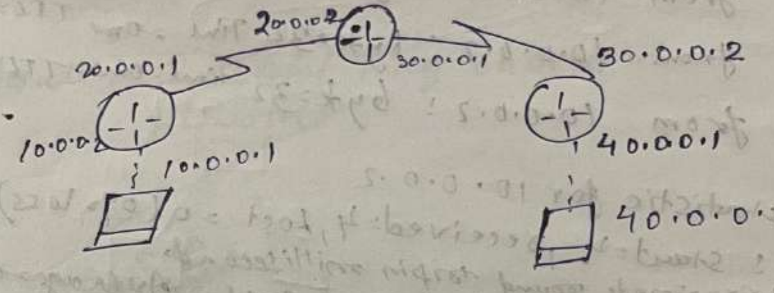

ISR4331
Router0
PC-PT
PC0
PC-PT
PC1

24/11/22

Ex-3

Aim - Configure default router to route

Topology - Star Topology.



Procedure !-

1> Select two end devices, configure there fastethernet ip address to 10.0.0.1 & 40.0.0.1

2> ~~select~~ Configure the three router in their CLI

3> started pinging from end devices to their nearest router, it shows request timed out.

4> Therefore we need to set up path for end devices from one router to other router end.

5> For each router in previlage mode run command show ip route. It shows networks that are directly connected to that router.

6> Their must be two router path for each router.

7> Then start pinging each devices, it shows succesful replay

Observation!– Before configuring router paths.

From one device PC-PT(10.0.0.1) ping next router (10.0.0.2)

Time=0ms TTL=255

Replay from 10.0.0.2 : byte=32 Time=0ms TTL=255

Replay from 10.0.0.2 : byte=32 Time=0ms TTL=255

Replay from 10.0.0.2 : byte=32 Time=0ms TTL=255

Replay from 10.0.0.2 : byte=32

Ping statistic for 10.0.0.2

Packet : Sent=4 , received=4, Lost =0 (0% loss)

Approximate round tripin millisecond.

minimum=0ms Maximum=0ms Average=0ms

* Then from the same end device PC-PT (10.0.0.1) ping 20.0.0.1 , 20.0.0.2

It shows request timeout.

Then set gateway for end devices.

Then pinged 20.0.0.1 it showe successful Replay.

Then pinged 30.0.0.1 , it shows destination host unreachable.

After configuring router paths.

* Ping end device. PC-PT(10.0.0.1) to PC-PT2(40.0.0.1)

Pinging ~~fro~~. 40.0.0.1 with 32 byte of data

Replay from 40.0.0.1 : byte=32 time=16ms TTL=125

Replay from 40.0.0.1 : byte=32 time=21ms TTL=125

Replay from 40.0.0.1 : byte=32 time=2ms TTL=125

Replay from 40.0.0.1 : byte=32 time=14ms TTL=125

Ping statistic for 40.0.0.1

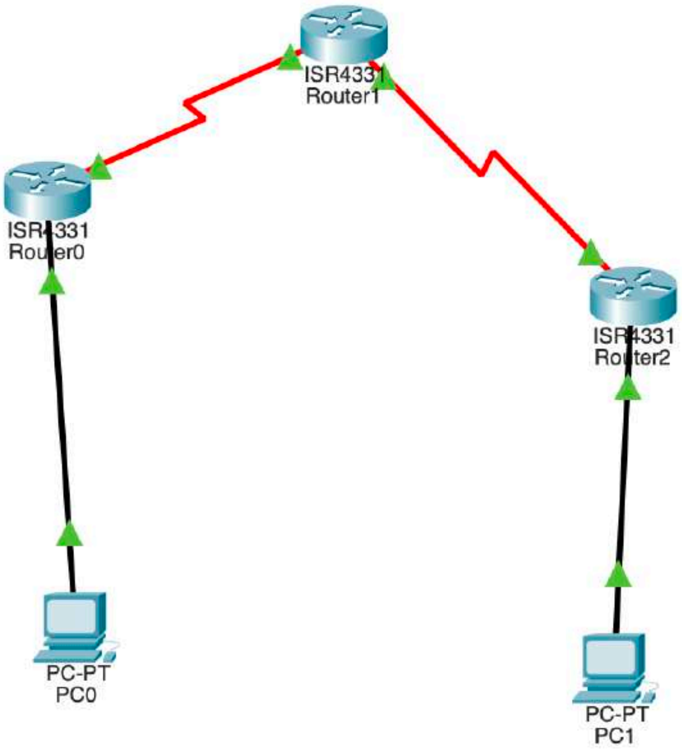
Packets : sent = 4, Received=4, Lost =0 (0% Loss)

Approximate round trip in millisecond.

Minimum=2ms , Maximum=21ms , Average=13ms.

Result :- Successful ping only on manually connecting networks.

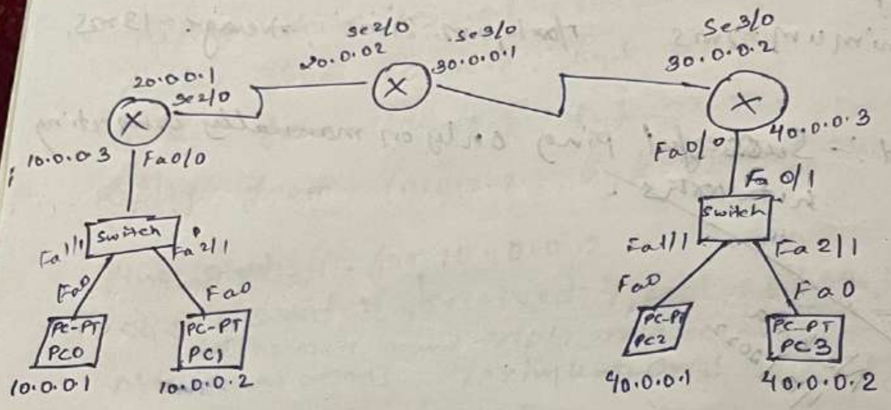
Neelima
1/12/2022


ISR4331
Router1
ISR4331
Router0
ISR4331
Router2
PC-PT
PC0
PC-PT
PC1

1/12/22

Ex-4

Aim - Configuring default route to the router.



Procedure :- A topology was created using 3 router, 2switch and 2PC-PT connected to each switch using copper straight and serial DCE connection.

* Default gateway and unique IP address were configured for each PC.

* IP address were configured for each interface using CLI.

Observation!- *Pinging PC 2 from PC0 gave destination host unrechable.

* IP routes for each router was viewed using: show ip route.

* Static ip route was configured for router 1 by ip route > destination network > subnet mask > nexthop address.

* Default ip route configure was for route 0 and route 2 by 0.0.0.0 0.0.0.0 next_hop address.

* pinging PC2 from PC0 gave required reply.

Result:-

PC> ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 : byte =32 time=0ms TTL=255
Reply from 10.0.03 : byte=32 time = 1ms TTL=255
Reply from 10.0.0.3: byte =32 time =0ms TTL=255
Reply from 10.0.0.3: byte=32 time =0ms TTL=255

PC> ping 20.0.0.1

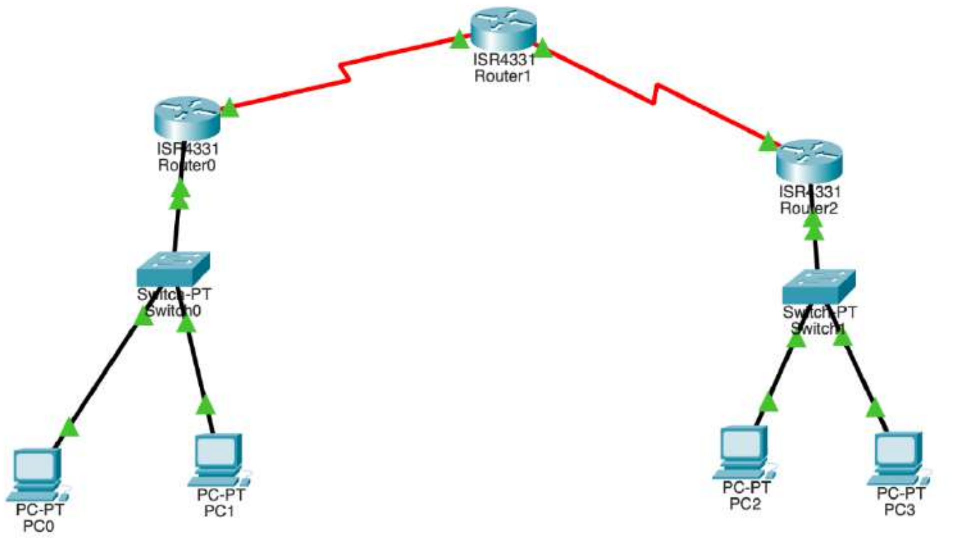
Pinging 20.0.0.1 with 32 byte of data

Reply from 20.0.0.1 : byte = 32 time=0ms TTL=255
Reply from 20.0.01 : byte =32 time=0ms TTL=255
Reply from 20.0.0.1: byte = 32 time=5ms TTL = 255
Reply from 20.0.0.1; byte =32 time =0ms TTL=255

Ping statistics for 20.0.0.1
Package sent=4, Received= 4, Loss = 0 (0% Loss)
Approximate round trip in ms.
Maximum =5ms Minimum=0ms Average=1ms

```
PC> ping 30.0.0.2
Pinging 30.0.0. 2 with 32 byte of data :
Request timed out.
Reply from 30.0.0.2 : byte=32 time=12ms TTL=253
Request timed out
Reply from 30.0.0.2 : byte=32 time=2ms TTL=253

Ping statistics for 30.0.0.2
Packet sent = 4, Received = 2, Lost = 2(50% Loss)
Approx round trip in ms :
Minimum = 2ms   Maximum = 12ms   Average = 7ms.

PC> ping 50.0.0.1
~~Reply~~: Pinging 50.0.0.1 with 32 byte of data :
Reply from 50.0.0.1 : byte=32 time= 2ms TTL=125
Reply from 50.0.0.1 : byte=32 Time=3ms TTL=125
Reply from 50.0.0.1 : byte=32 Time=7ms TTL=125
Reply from 50.0.0.1 : byte=22 Time=10ms TTL=125

Ping statistic for 50.0.0.1
Packet sent=4, Recieved=4, Loss=0(0% loss)
Approx round trip in ms.
Minimum= 2ms  Maximum=10ms  Average=5ms.
```

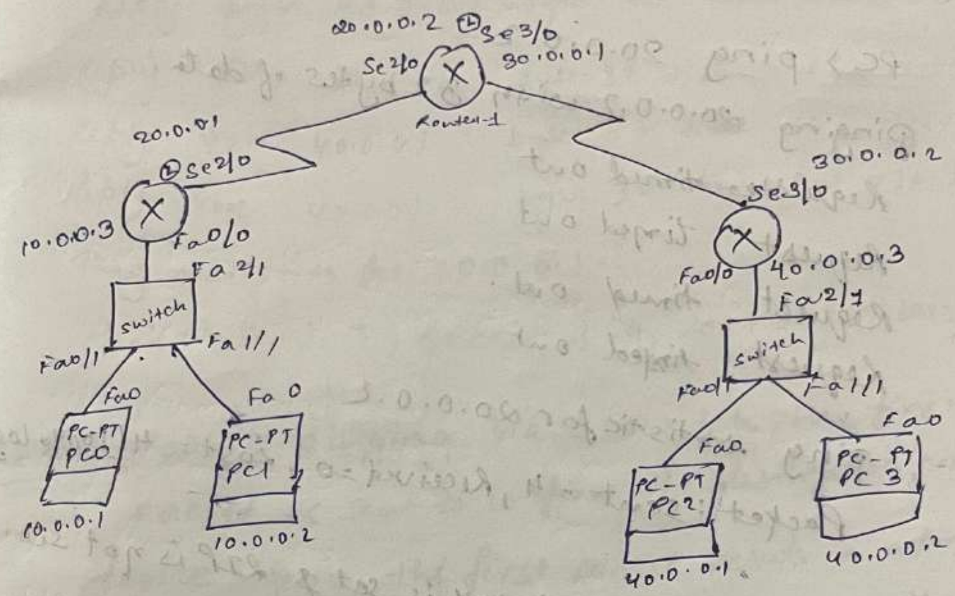
Neelima
1/12/2022


ISR4331
Router1
ISR4331
Router0
ISR4331
Router2
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3

8/12/22

Ex-5

Aim - Configure RIP Routing Pryotocol in Routers.



Procedure :- Created ~~a~~ a Topology using 3 routers, 2 switches, 2PC-PT connected to each switch using copper straight and DCE cable.

* Default gateway and unique IP address was configured for each PC.

* IP address were configured for each interface using CLI.

~~Observations~~

* Configure router ip address using ip address <address> <subnet-mask>

* Encapsulation ppp and clock rate 64000 used to specify PPP protocol and clock rate ~~o~~ in router 0 and 1 for the serial port.

* ~~[illegible]~~ configure RIP routing route rip then network <address> command, where network refers.

Output !—

(i) When no gateway or RIP is set

```
PC > ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Requeste timed out
Request timed out
Request timed out.
Request timed out

ping statistic for 20.0.0.2
Packet : sent = 4, Received =0 , Lost = 4(100% loss)
```

(ii) When only gateway is set & RIP is not set.

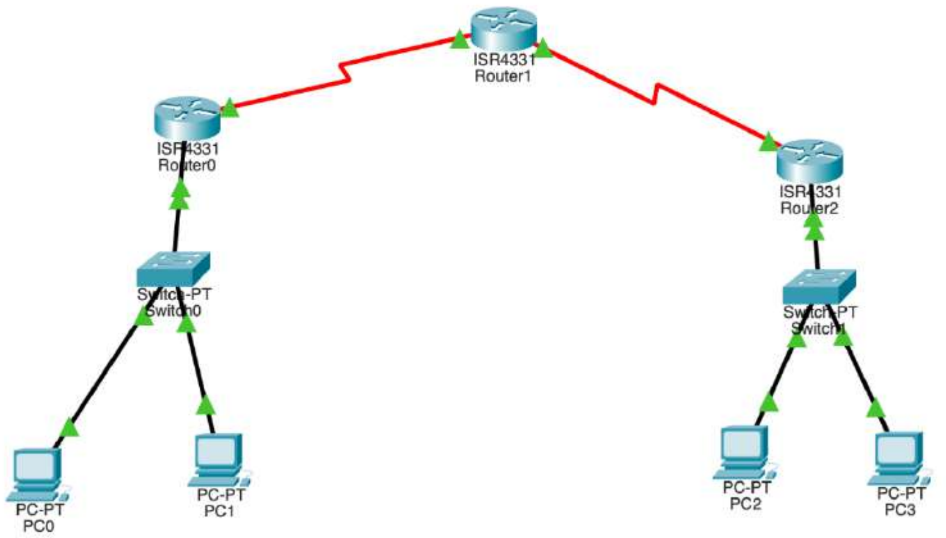
```
PC > Ping 30.0.0.1
Pinging 30.0.0.1 with 32byte of data:
Reply from 10.0.0.3 : Destination host unrechabl
Reply from 10.0.0.3 : Destination host unrechabl
Reply from 10.0.0.3 : Destination host unrechabl
Reply from 10.0.0.3 : Destination host unrechablo
Ping Statistics for 30.0.0.1
Packet : Sent=4 , Received=0, Lost=4 (100% loss).
```

(iii) when both gateway & RIP is set.

```
PC> ping 40.0.0.1
Reply from 40.0.0.1 : byte = 32   time = 9ms  TTL=125
Reply  from  40.0.0.1 : byte=32  time = 2 ms  TTL=125
Reply  from  40.0.0.1 : byte=32  time = 16 ms  TTL=125
Reply  from  40.0.0.1 : byte=32  time = 9ms  TTL=125
Ping statistics for 40.0.0.1
Packet sent 4, Received = 4, Lost = 0 (0% loss)
```

Note:- while pinging via a switch, the first packet is sent to confirm the destination end device. Hence, the first packet resuts in a timed out situation. 4 packets are sent & 3 packets are received.

Neelima
8/12/2022

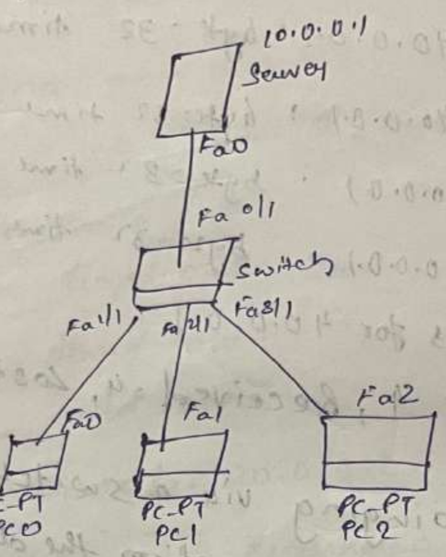

ISR4331
Router1
ISR4331
Router0
ISR4331
Router2
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3

15/12/22

Ex-6

Aim:- Configure DHCP Server.

Topology :-



Procedure :- 1) Configure ip address for the server 10.0.0.1

2) Select server and configure DHCP in server and turn on service.

3) Assign any poolname and give start ip address of the server & the subnet mask is also added.

4) We can ~~either~~ add DNS server same that of server's ip address.

5) Oncewe click save the pool is created

6) When we click on the end devices in order to config the ip address and select DHCP in the ip configuration the ip address is automatically configured.

observation :-

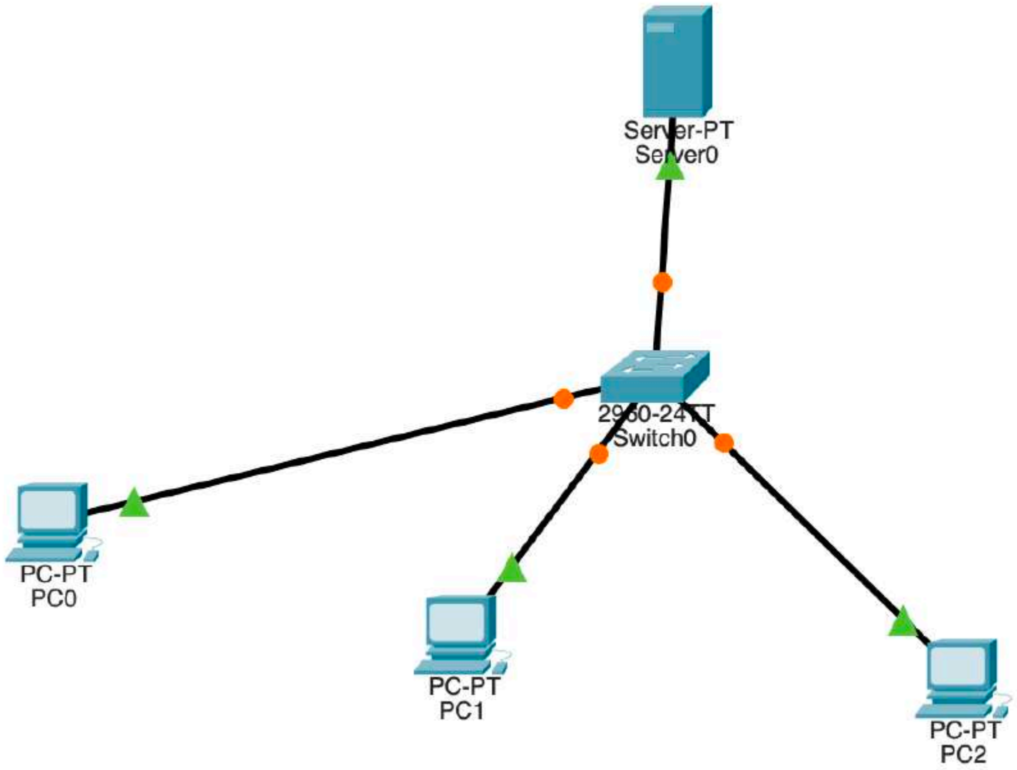
DHCP(Dynamic host configuration Protocol)
we don't have to give the ip for enddevices manually
it dynamically configure.

Result :-

```
PC > ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data.
Reply from 10.0.0.3 : byte =32 time =81ms TTL = 128
Reply from 10.0.0.3 : byte =32 time = 0ms TTL = 128
Reply from 10.0.0.3 : byte = 32 time =0ms TTL = 128
Reply from 10.0.0.3 : byte = 32 time =0ms TTL = 128

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

```
PC > ping 10.0.0.1
Pinging 10.0.0.1 with 32byte of data
Reply from 10.0.0.1 : byte= 32 time=0ms TTL = 128
Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128
Reply from 10.0.0.1 : byte = 32 time =0ms TTL = 128
Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128

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

Neelima S
15/12/2022

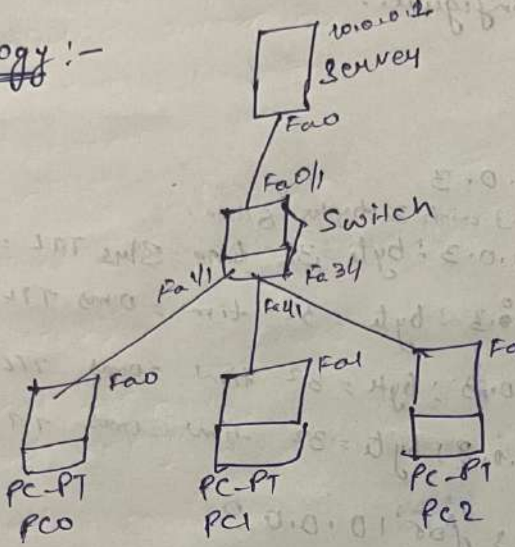

Server-PT
Server0
2960-24TT
Switch0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2

10/12/22

Ex-7

Aim :- Configure WebServer and DNS Server.

Topology :-



Procedure :- ① Configure ip address of server 10.0.0.1

② Select server and config DHCP server and turn on service

③ Assign any pool name and give start ip address of the server

④ We can add DNS server same that of server's IP address

⑤ We click on end devices to config IP address select DHCP in the ip and the ip address is config automatically.

⑥ click on server and services and HTTP make it on.

⑦ Then on DNS make it on and put name : www.qwerty.com type : A Record Address: 10.0.0.1 → same as server Ip.

⑧ ~~Then~~ click on add and the Goto ~~end~~ web browser in end devices and type www.qwerty.com.

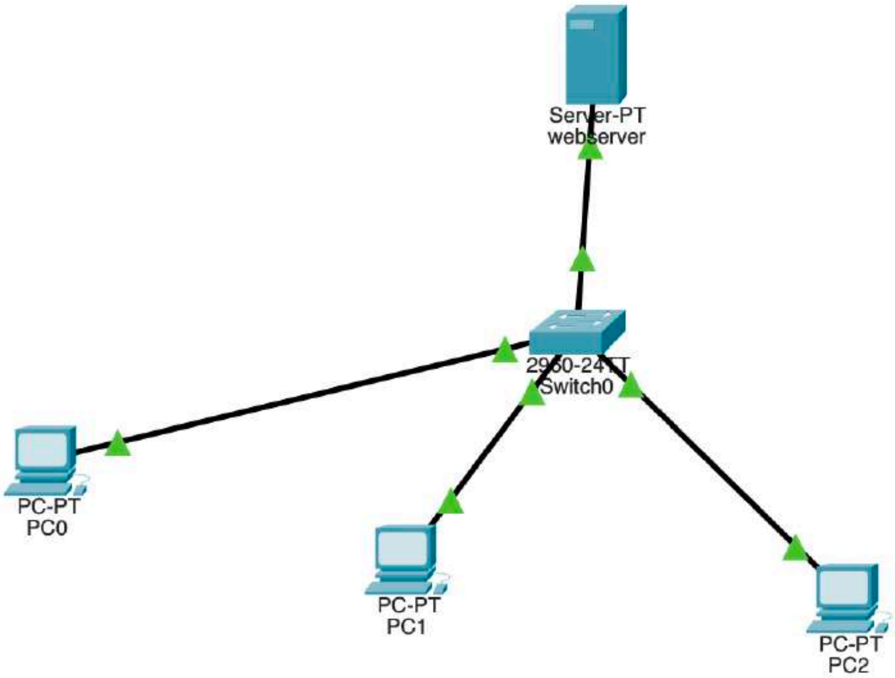
Observation:-

DNS server maps domain name with IP address of the server.

Result:- ① If configured properly the page of Cisco packet Tracker is opened.

② If not configured properly i.e DNS server and default gateway, 'Host Unresolved' is shown.

Neelima
15/12/2022


Server-PT
webserver
2960-24TT
Switch0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2

LAB-8

29/12/22 Write a Program for error detection using CRC - 16 bit.

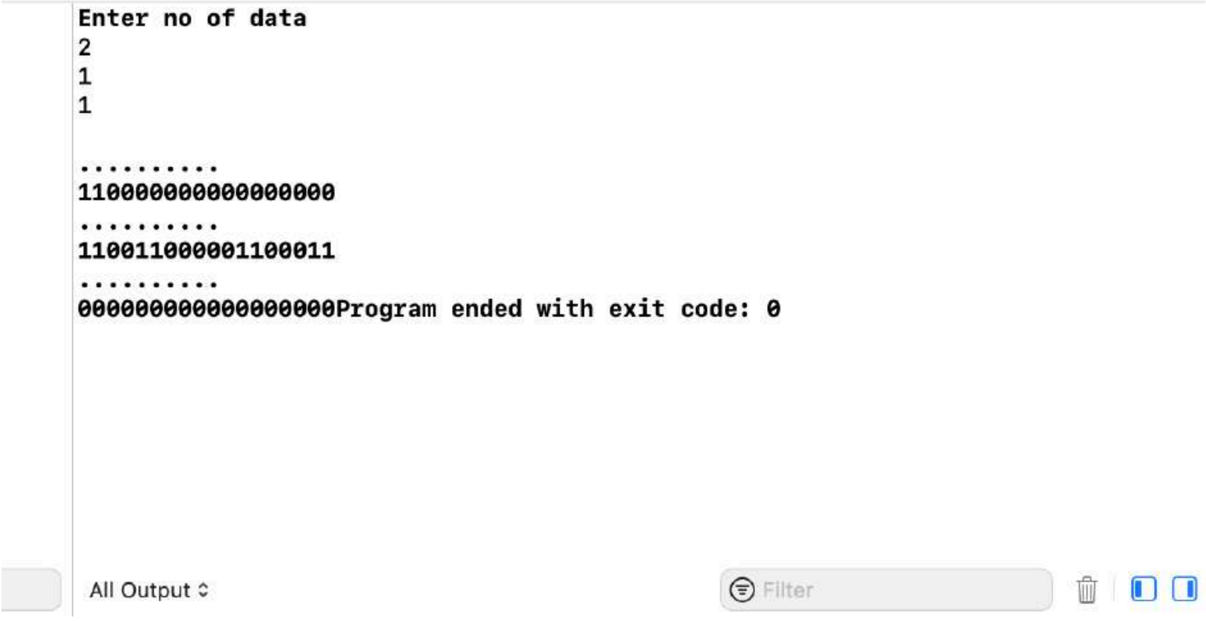
```
-> #include <iostream.h>
   using namespace std;
   int main()
   { int i=0, j=0, n;
     cout << "Enter no. of data entry: ";
     cin >> n;
    int a[n+16]
    int g[] = {1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1
    int rem[n];
    int m= n+16;
    for(i=0; i <m; i++)
     { if (i>=n){
          a[i] =0; }
      else{
          cin >> a[i];
         if (i<n){
           rem[i]= a[i]; }}

  for(i=0; i <n; i++)
   { if (a[i] != 1)
       continue;
   for(j=0; j <17; j++)
   {a[i+j] = a[i+j]^g[j]; }}
```

```
for (i=0; i <n; i++)
    {
    a[i] = rem[i];
    }
    for (~~int~~ i=0; i<m; i++)
    {
     cout << a[i]; }
```

Output :-

```
3
1
1
1
111 0 11 00 000 11 0 0 0 11
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0
```

5/1/2023



```
Enter no of data
2
1
1

..........
110000000000000000
..........
110011000001100011
..........
000000000000000000Program ended with exit code: 0
```

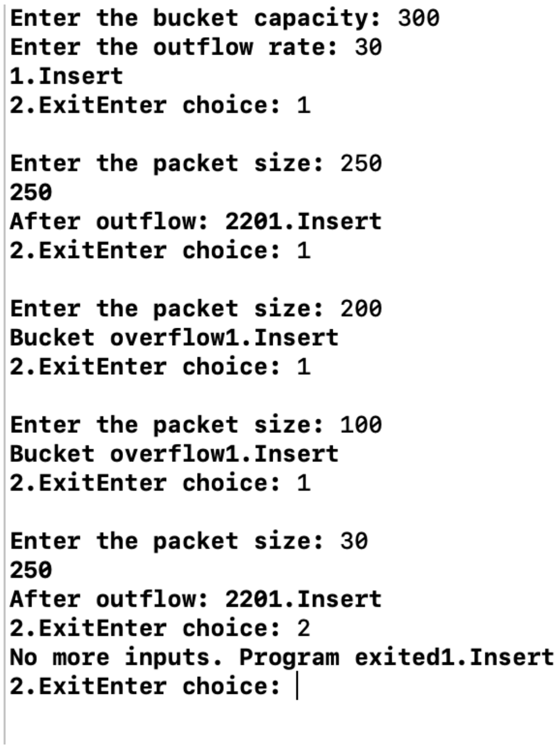
5/1/22

Leaky Bucket

```
→ #include <iostream>
using namespace std;
int main ()
{
cout << "Enter bucket size" << endl;
int bucketsize;
int filled=0;
int outputrate, inputrate, choice;
cin >> bucketsize;
cout << "Enter ~~packetsize~~ outputrate" << endl;
 cin >> outputrate;
do {
 cout << "Enter packet size" << endl;
 cin >> input~~packet~~ rate;
~~do{ cout << "enter~~
   if (inputrate <= bucketsize) {
   if ( filled+inputrate > bucketsize)
     {
     cout << "packet too big" << endl;
   }
  else {
  filled = filled + inputrate;
       }
     }
else {
cout << "Packet is too big" << endl;
 }
if(filled <= outputrate)
   { filled=0; }

else
{ filled = filled-outputrate; }

cout << "Amount of bucket filled" << filled;
cout << "Do you want to continue (1 to yes  2 to no)" << endl;
cin >> choice;
} while(choice==1); }
```

```
Output :- Enter Bucket size . 500
          Enter ouput rate  50
          Enter packet size  700
          Packet too big.
          Do you want to continue (1 to yes 2 to no) 1
          Enter packet size 200
          Bucket filled  150
          Do you want to continue (1 to yes 2 to no) 1
          Enter packet size 250
          Bucked filled 350.
          Do you want to continue (1 to yes 2 to no) 2.
```

12/1/2023.



```
Enter the bucket capacity: 300
Enter the outflow rate: 30
1.Insert
2.ExitEnter choice: 1

Enter the packet size: 250
250
After outflow: 2201.Insert
2.ExitEnter choice: 1

Enter the packet size: 200
Bucket overflow1.Insert
2.ExitEnter choice: 1

Enter the packet size: 100
Bucket overflow1.Insert
2.ExitEnter choice: 1

Enter the packet size: 30
250
After outflow: 2201.Insert
2.ExitEnter choice: 2
No more inputs. Program exited1.Insert
2.ExitEnter choice: 
```

Bellman Foud.

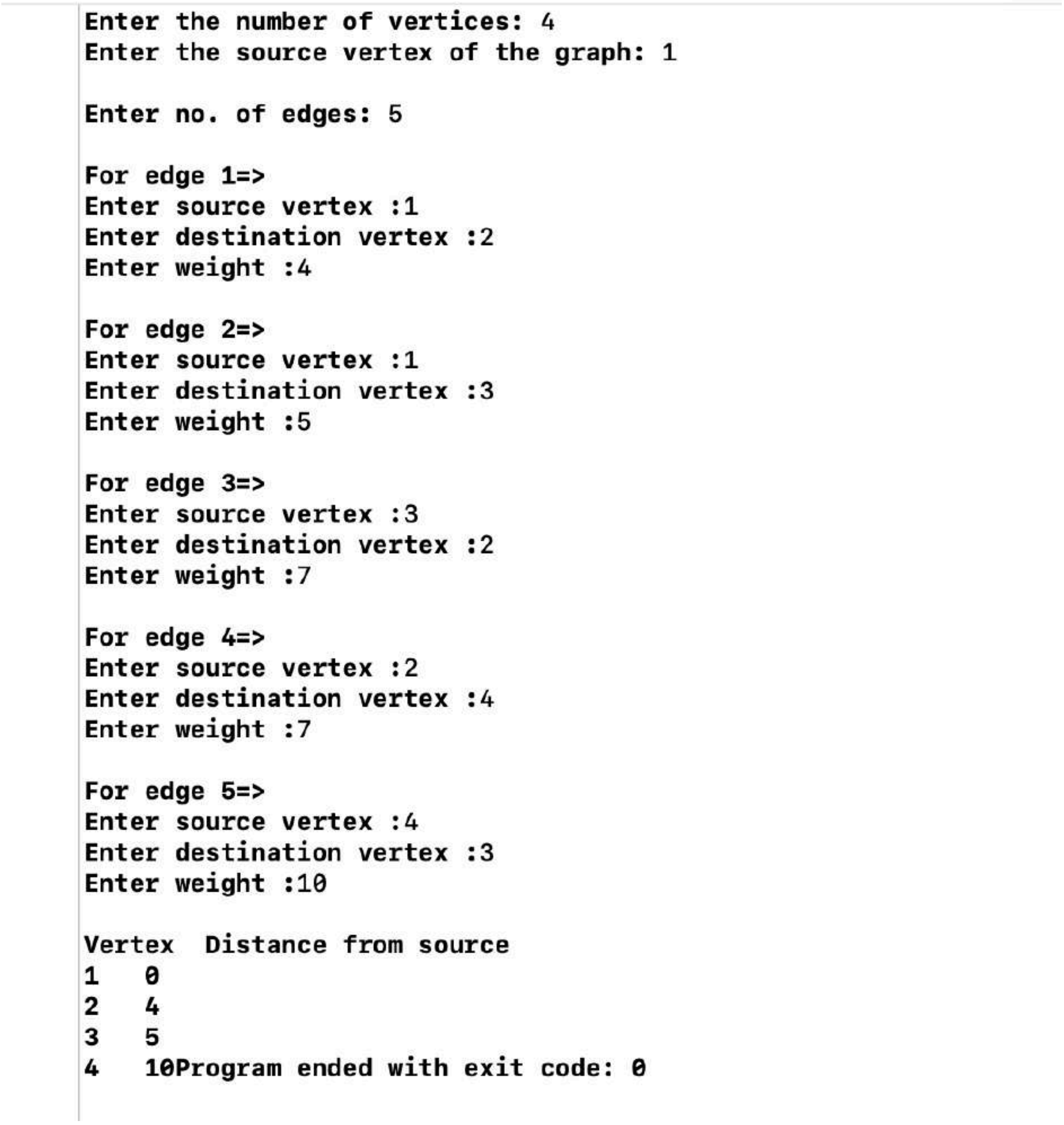
12/1/23

~~→ #include <stdio.h>~~
~~#include <stdlib.h>~~
~~int Bellman-Foud (int G[20][20], int V, int E,~~
~~int edge [20][20])~~

```
→ #include <iastream>
  #define MAX 70
  usingnamespace std;
  typedef struct edge
   {int src;
    int dest;
    int wt;
   }edge;
void bellman-ford (int nv, edge e[], int src-graph,
                   int ne)
{ int u,v,weight, i, j=0;
  int dis[MAX];
  for (i=0; i< nv; i++)
  {dis[i]=999; }
  dis [src-graph] =0;
  for (i=0; i< nv-1; i++)
  {for j=0; j<ne; j++)
   { U=e[j].src;
     v=e [j].dest;
     weight = e[j].wt;
    if (dis[U]! =999 && dis[U]+weight <dis[v])
      dis[v]= dis[U]+weight;
   }
  }
for (j=0 ;j<ne ;j++)
 { U=e[j] .src;
   V=e[j].dest;
   weight =e[j].wt;
if (dis[v] +weight  < dis[v])
{cout << "Negative cycle Present";   }
 }
cout << "vertex" << "Distance from source";
for (i=1 ; i<=nv; i++)
   cout << "\n" << i << "\t" << dis[i];
}
  int main ()
{ int nv, ne, src-graph;
 edge e[MAX];
 Cout << "Enter the no. of vertices!";
  cin>> nv
  cout<< "Enter the source vertex of graph"
  ~~cood~~ .
   cin >> src-graph
   cout << "Enter no. of edges : "
   cin >> ne ;
   for (i=0 ; i<ne; i++)
  { cout << "for edge" << i+1 << "=>" ;
   cout << "\n Enter source vertex !";
    cin >> e[i].src;
   Cout << "Enter destination vertex";
    cin >> e[i]. dest;
     cout << "Enter weight";
      cin >> e[i].wt
     }
       bellman-ford (nv, e, src-graph, ne);
 return 0;
 }
```

2/2/202

```
Output :-
Enter no. of vertices : 4
Enter graph in matrix form
   0   5   4   999
   5   0   6    3
 999   3   1    6
   2   0   1    4
Enter source vertex : 1

vertex 1 -> cost =0  parent =0
vertex 2 -> cost =5  parent =1
vertex 3 -> cost =4  parent =1
vertex 4 -> cost =8  parent =2

No negative weight cycle.
```



```
Enter the number of vertices: 4
Enter the source vertex of the graph: 1

Enter no. of edges: 5

For edge 1=>
Enter source vertex :1
Enter destination vertex :2
Enter weight :4

For edge 2=>
Enter source vertex :1
Enter destination vertex :3
Enter weight :5

For edge 3=>
Enter source vertex :3
Enter destination vertex :2
Enter weight :7

For edge 4=>
Enter source vertex :2
Enter destination vertex :4
Enter weight :7

For edge 5=>
Enter source vertex :4
Enter destination vertex :3
Enter weight :10

Vertex  Distance from source
1   0
2   4
3   5
4   10Program ended with exit code: 0
```

12/1/23

Dijkstra

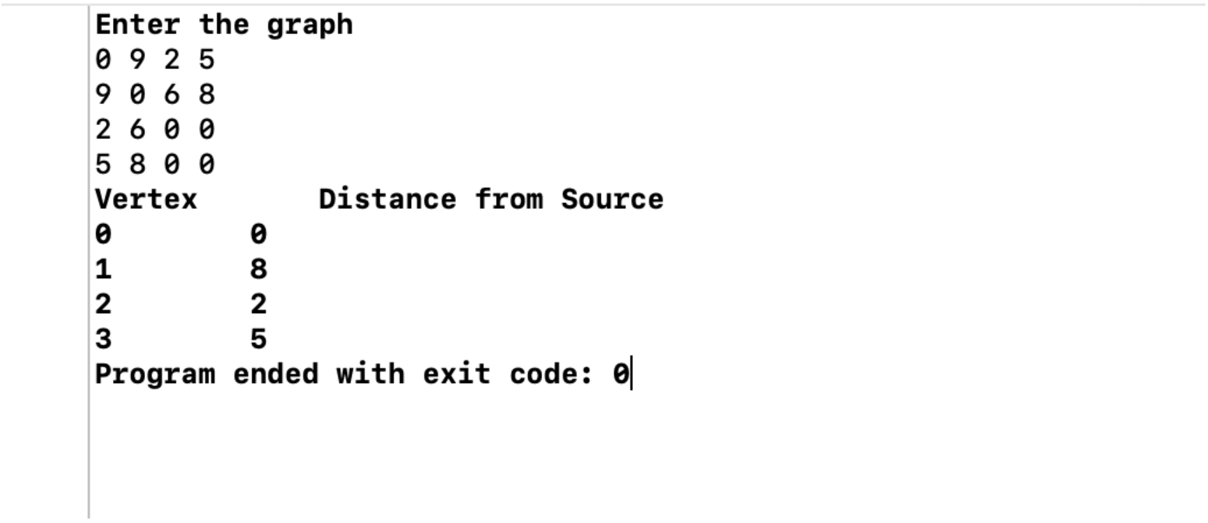
```
#include <stdio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra(int G[MAX][MAX], int n, startnode);
int main() {
int G[MAX][MAX], i, j, n, u;
printf("Enter no. of vertices : ");
scanf("%d", &n);
printf("Enter adjacency matrix ");
for (i=0 ; i<n ; i++)
 for (j=0 ; j<n ; j++)
scanf("%d", &G[i][j]);
printf("Enter starting node : ");
scanf("%d", &u);
dijkstra(G, n, u);
return 0;
}
void dijkstra(int G[MAX][MAX], int n, int startnode)
{
 int cost[MAX][MAX], distance[MAX], pred[MAX];
 int visited[MAX], count, mindistance, nextnode, i, j;
for (i=0; i<n; i++)
for (j=0 ; j<n ; j++)
 if(G[i][j]==0)
 cost[i][j] =infinity;
else
 cost[i][j] = G[i][j];
for (i=0 ; i<n ; i++) {
distance[i]=cost[startnode][i];
 pred[i] =startnode; visited[i]=0;
}
distance[Start node] =0;
visited[Start node]=1
}
```

```
for (i=0 ; i<n ; i++)
  if (i != startnode)
    printf("Distance of node %d = %d", i , distance[i]);
    printf(" Path = %d", i);
        j = i ;
     do {
        j = pred[j]
        printf(" <- %d", j); }
     while (j != startnode);
  }}
```

Output:-

```
Enter no.of vertices : 4
  Enter the adjacency matrix :
    0   5   4   999
    5   0   6   3
  999   3   1   6
    2   0   1   4
Enter the starting node : 1
Distance of node 0 = 5
Path = 0 <- 1
Distance of node 2 = 4
Path = 2 <- 3 <- 1
Distance of node 3 = 3
Path = 3 <- 1
```

2/2/2022



```
Enter the graph
0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0
Vertex       Distance from Source
0        0
1        8
2        2
3        5
Program ended with exit code: 0
```

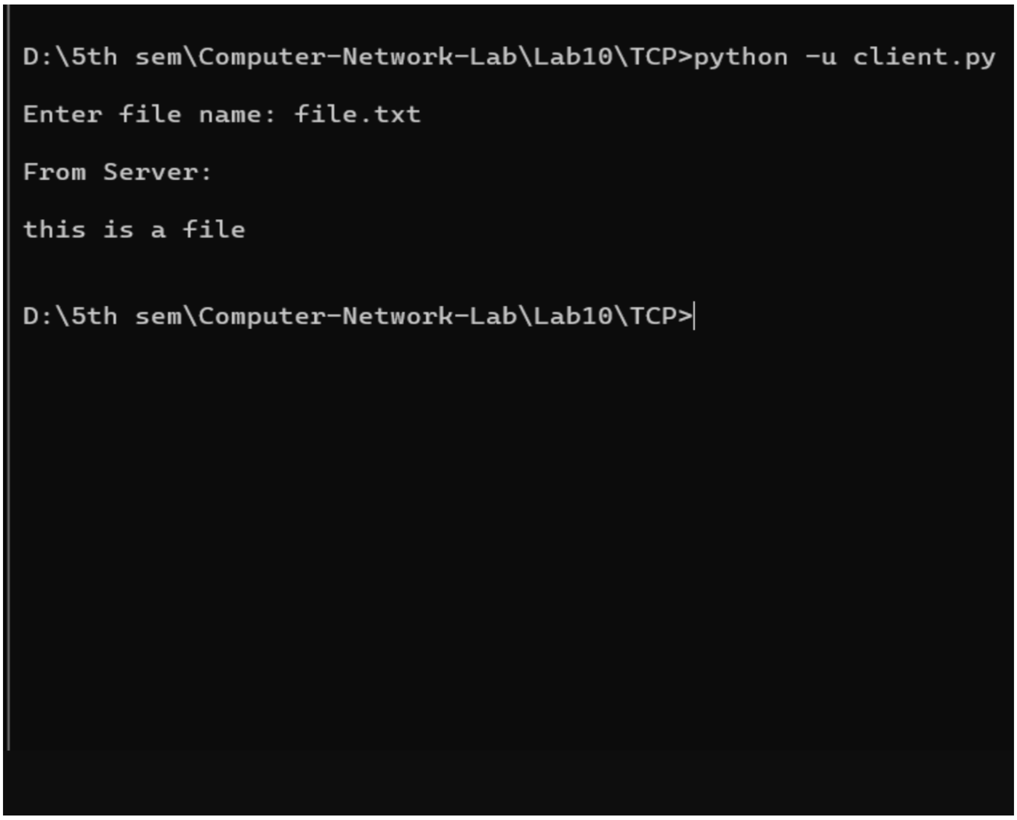
TCP/IP Socket   Write client-Server program

=> client TCP

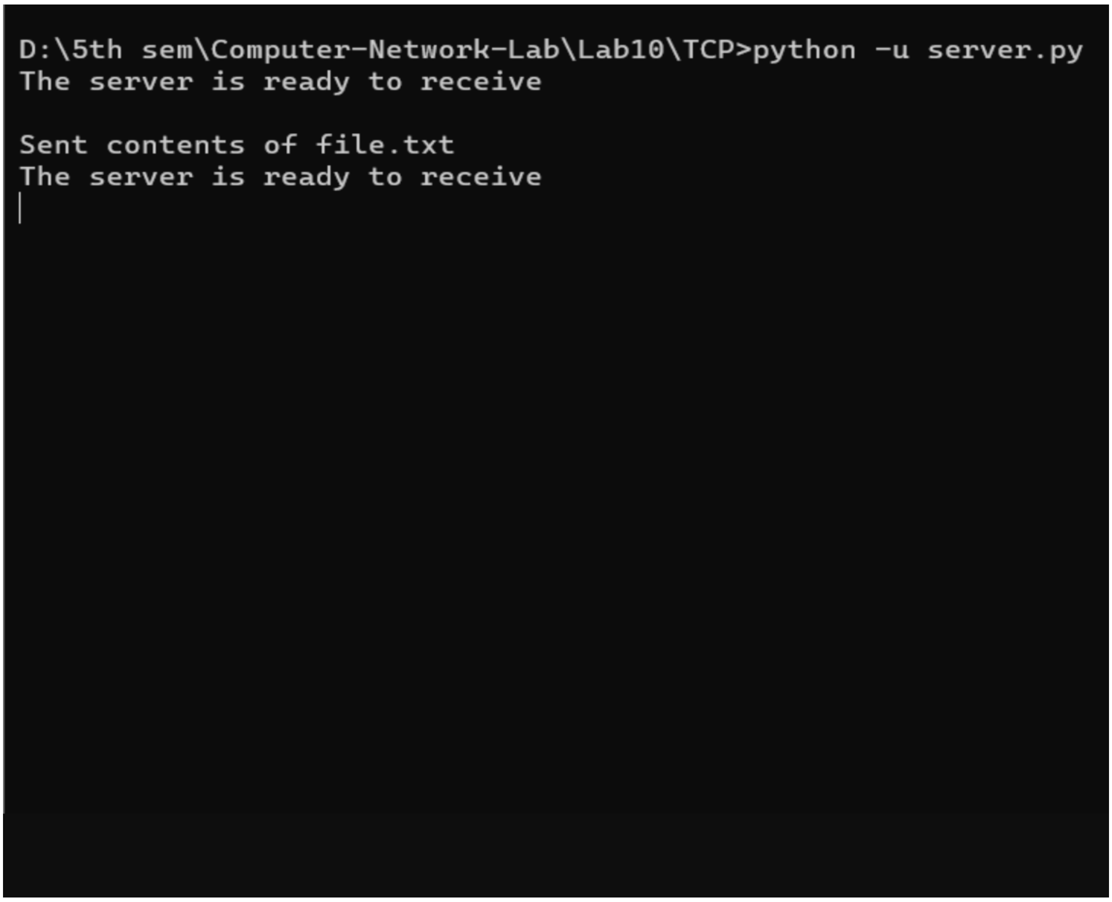
```
from socket import*
serverName = '127.0.0.1'
ServerPort = 12000
Client Socket = Socket (AF_INET, SOCK-STREAM)
ClientSocket . connect((serveName, serve Port))
sentence =input(" Enterfilename").
clientSocket.send (sentence .encode())
filecontents =clientSocket.recv(1024).decode()
print("from Server")
print (file contents)
Client Socket.close()
```

=> Server TCP.

```
from socket import *
Server name='127.0.0.1'
Server Port = 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 = server Socket.accept()
  sentence=connection Socket.rec (1024).decode
  file=open(sentence, "r")
  l=file.read (1024)
connection Socket.send(l.encode())
print ("sent contents of " +sentence)
 file.close()
Connection socket.close().
```



```
D:\5th sem\Computer-Network-Lab\Lab10\TCP>python -u client.py

Enter file name: file.txt

From Server:

this is a file


D:\5th sem\Computer-Network-Lab\Lab10\TCP>
```



```
D:\5th sem\Computer-Network-Lab\Lab10\TCP>python -u server.py
The server is ready to receive

Sent contents of file.txt
The server is ready to receive
```

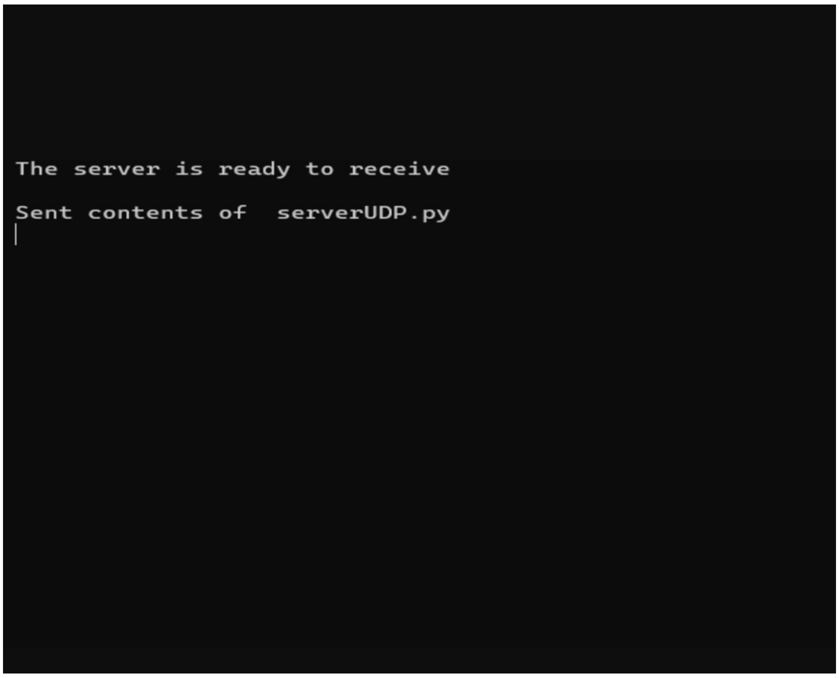
UDP Socket   write client - Server program

⇒ Cliend UDP

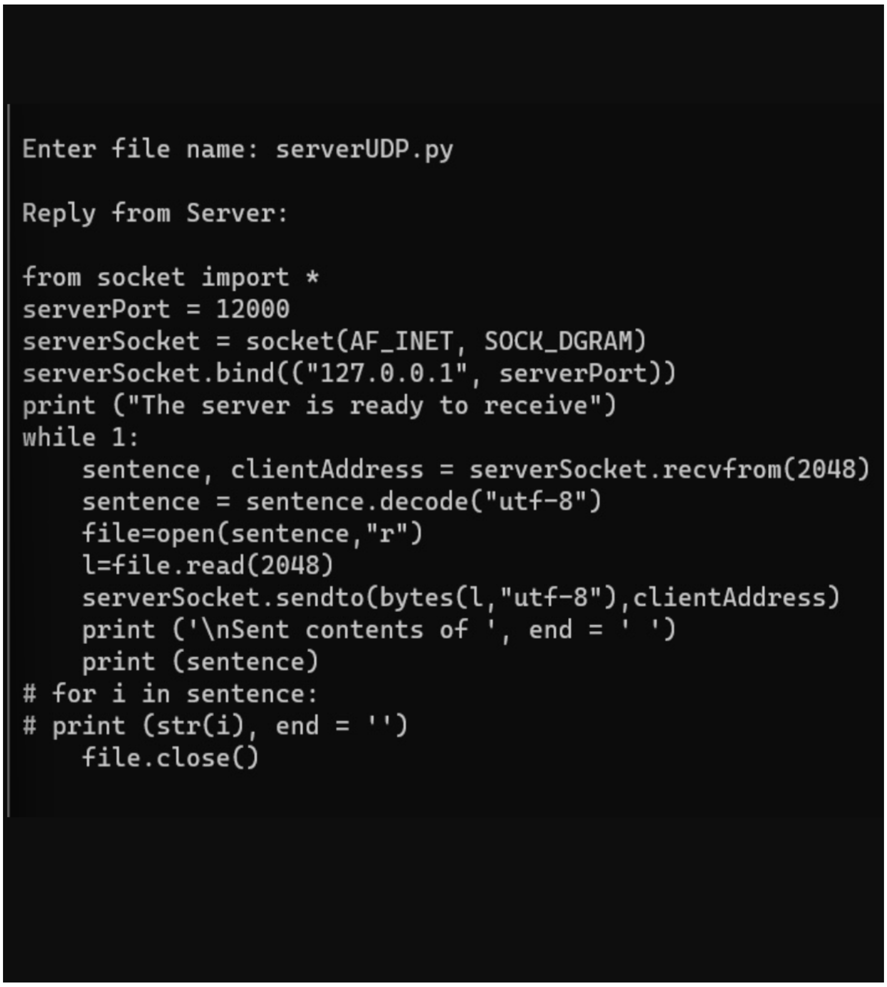
```
from Socket import *
 serverName = '127.0.0.1'
 server Port  = 12000
clientSocket= Socket(AF_INET, SOCK_DGRAM)
Sentence= input("Enter filename")
clientSocket. sentto (bytes(Sentence, "utf - 8"), (serverName,
                                                  serverPort)]
file contents, serverAddress = cliantSocket.recvfrom(2048)
print("Reply from server")
print(file contents.decode ("utf-8"))
client Socket .close()
```

⇒ Server UDP

```
 from socket import *
 server Port 1200
 Server Socket=Socket (AF_INET, SOCK_DGRAM)
 Server Socket.bind("127.0.0.1", server port)
 printf ("The server the ready")
while(1)
sentence, client Address= serverSocket. recvfrom(2048)
sentence = sentence.decode("utf-8")
 file = open(sentence, "r")
 l = file.read(2048)
serverSocket sendto(bytes(l, "utf-8"), client Address)
printf("Sent content of !, end='')
print(sentence)
file. close().
```



```
The server is ready to receive

Sent contents of  serverUDP.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()
```