VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

COMPUTER NETWORKS

Submitted by

Nandini Khastagir (1BM20CS093)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING BENGALURU-560019

October-2022 to Feb-2023

(Autonomous Institution under VTU)

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 **Nandini Khastagir (1BM20CS093),** 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 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.

Lohith J J
Assistant Professor
Department of CSE
BMSCE, Bengaluru
`

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

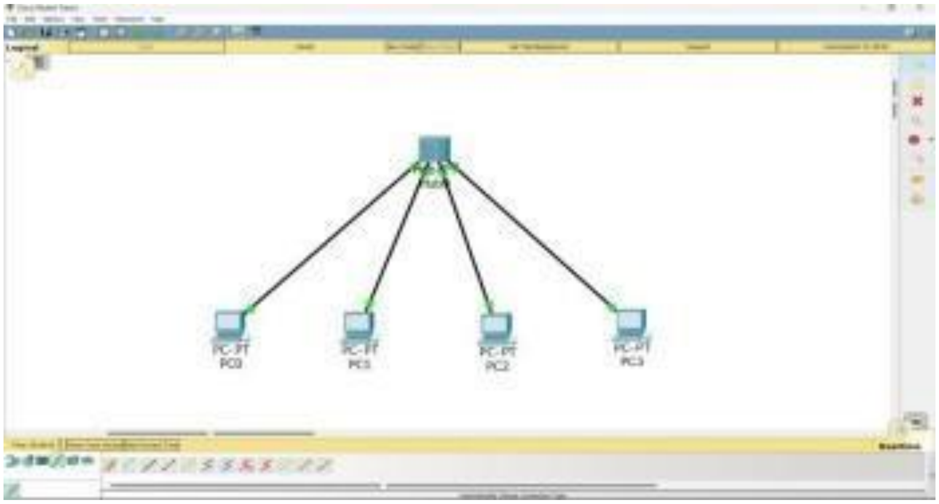
Index

Cycle-1 Experiment No 1

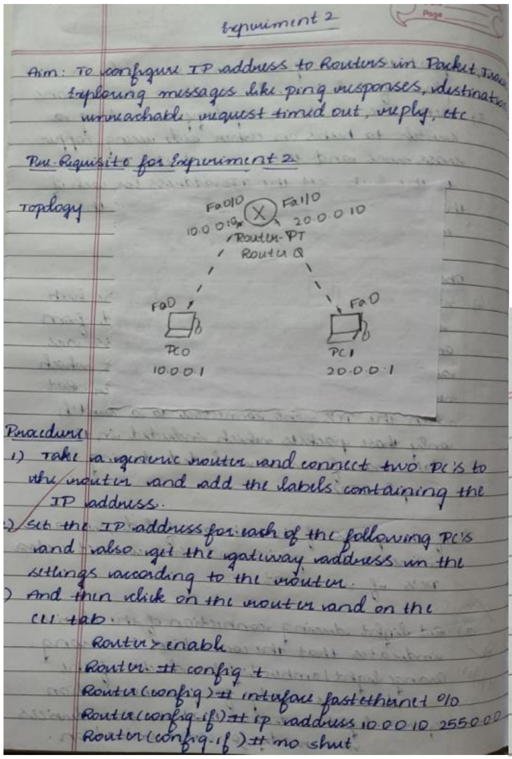
Aim of the program

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

Hub Topology



Procedure



Experiment 2

Aim: To configure IP address to Routers in Packet Tracer exploring messages like ping responses, destination unreachable, request timed out, reply, etc

Pre-Requisite for Experiment 2

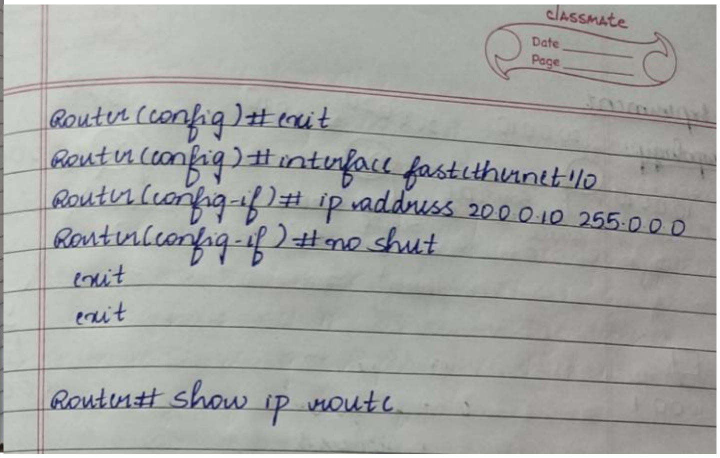
Topology

Fa0/0
Fa1/0
10.0.0.10
20.0.0.10
Router-PT
Router0
Fa0
Fa0
PC0
10.0.0.1
PC1
20.0.0.1

Procedure

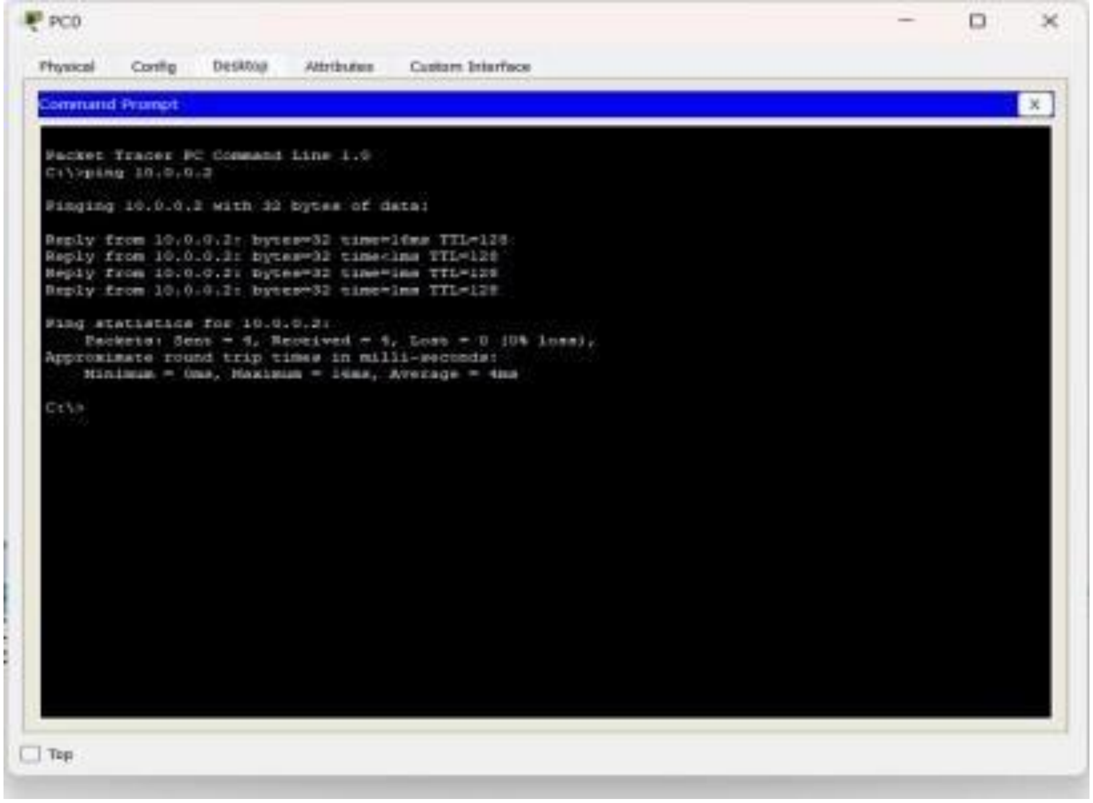
1) Take a generic router and connect two PC's to the router and add the labels containing the IP address.
2) Set the IP address for each of the following PC's and also get the gateway address in the settings according to the router.
3) And then click on the router and on the CLI tab.

```
Router>enable
Router# config t
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 10.0.0.10 255.0.0.0
Router(config-if)# no shut
```



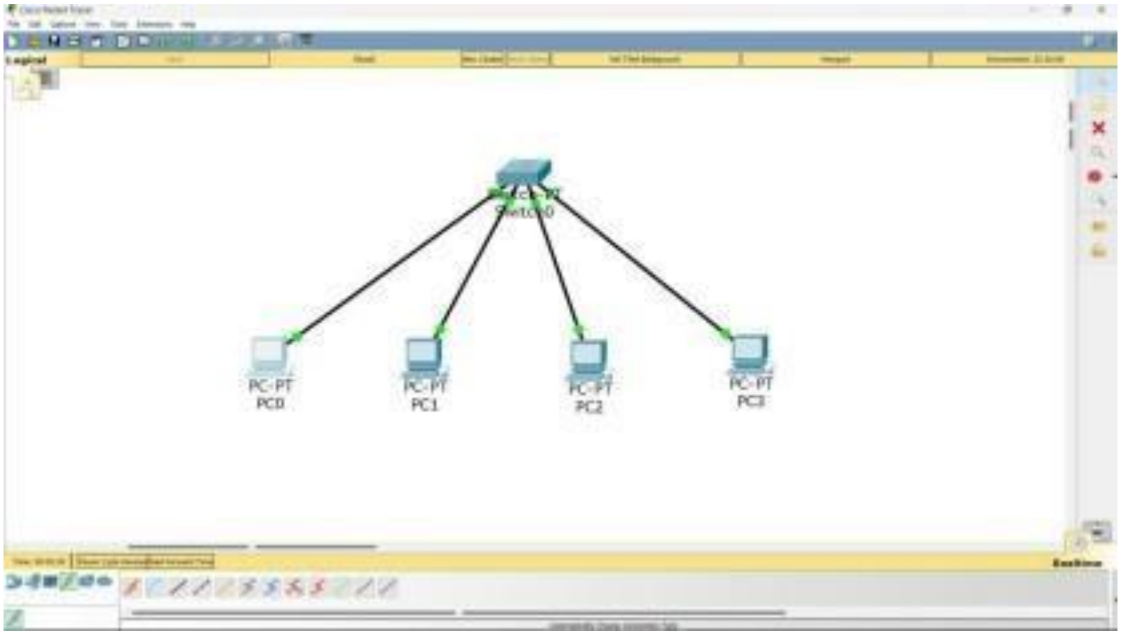
```
Router(config)# exit
Router(config)# interface fastethernet 1/0
Router(config-if)# ip address 20.0.0.10 255.0.0.0
Router(config-if)# no shut
 exit
 exit

Router# show ip route
```

Output

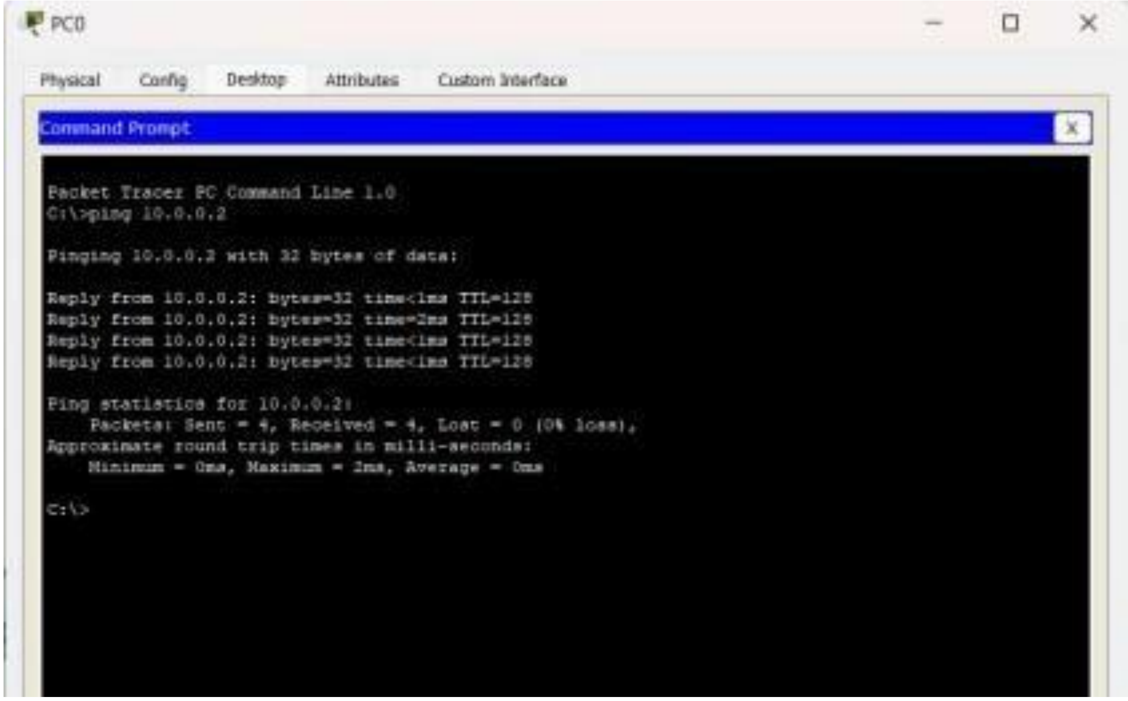


```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=14ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 14ms, Average = 4ms

C:\>
```

Switch Topology



Output



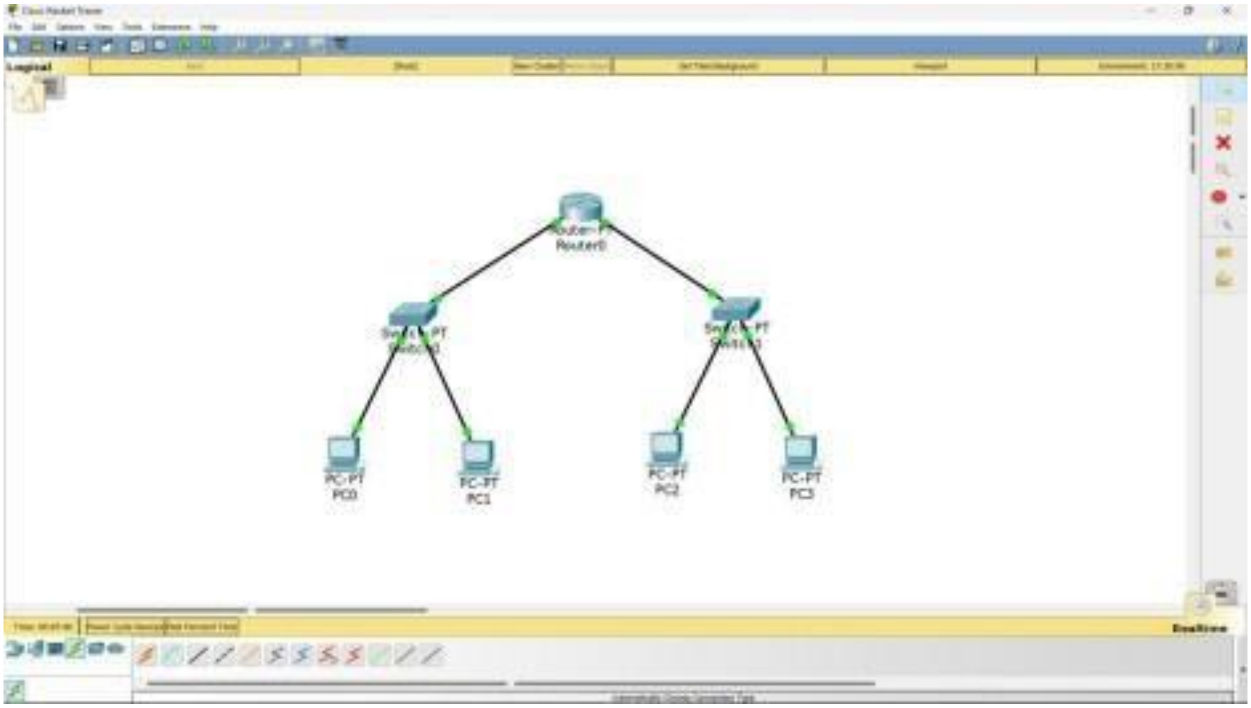
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=2ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>
```

Experiment No 2

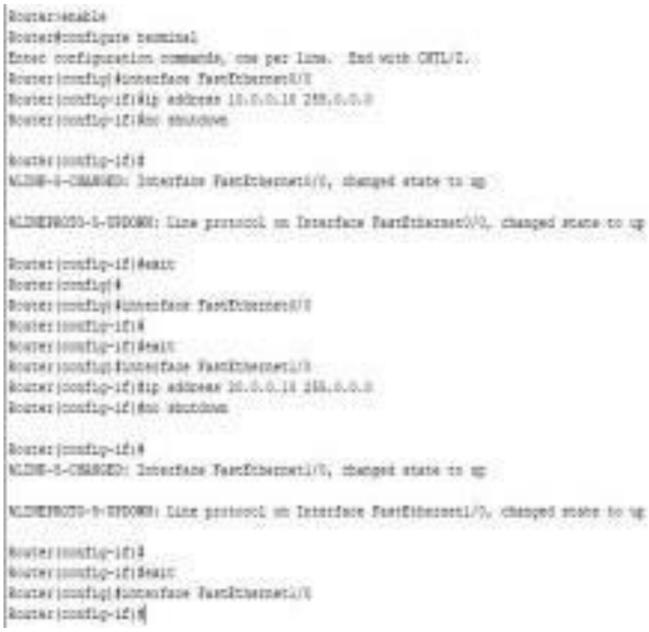
Aim of the program

Configuring IP address to Routers in Packet Tracer. Exploring the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.

Topology

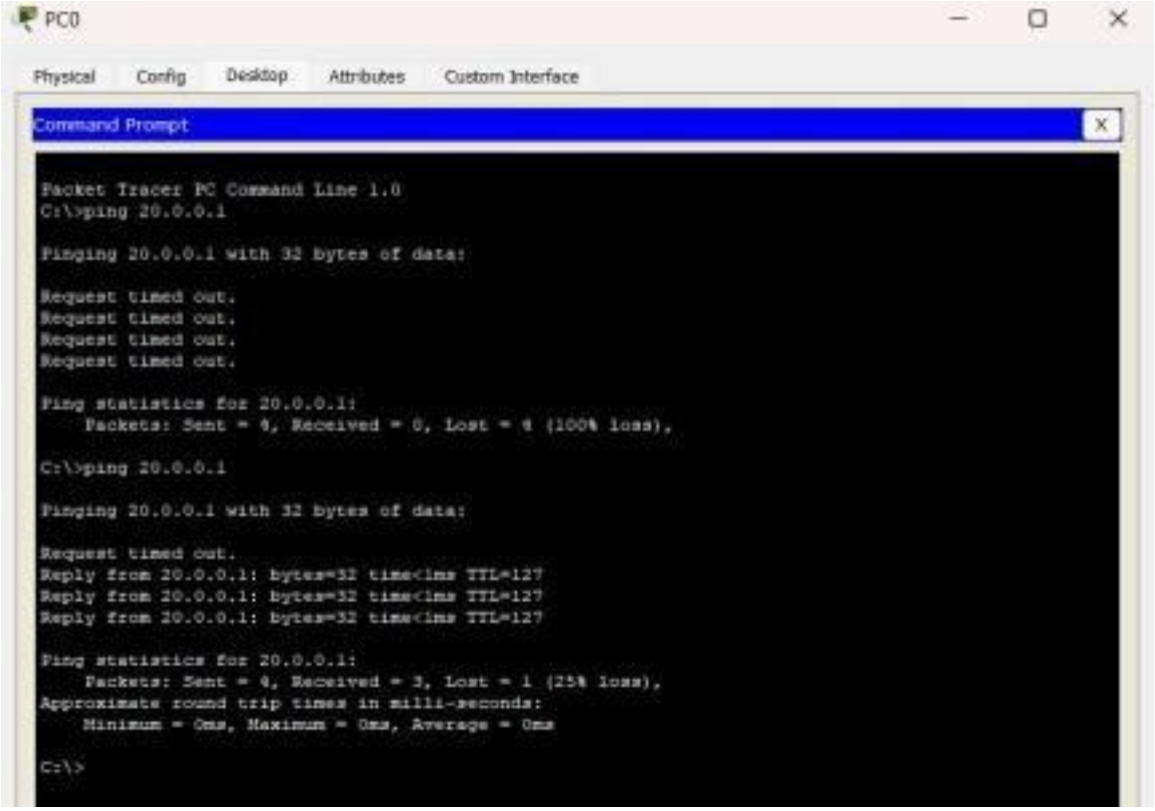


Procedure



```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.0.0.10 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up

Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
```

Output



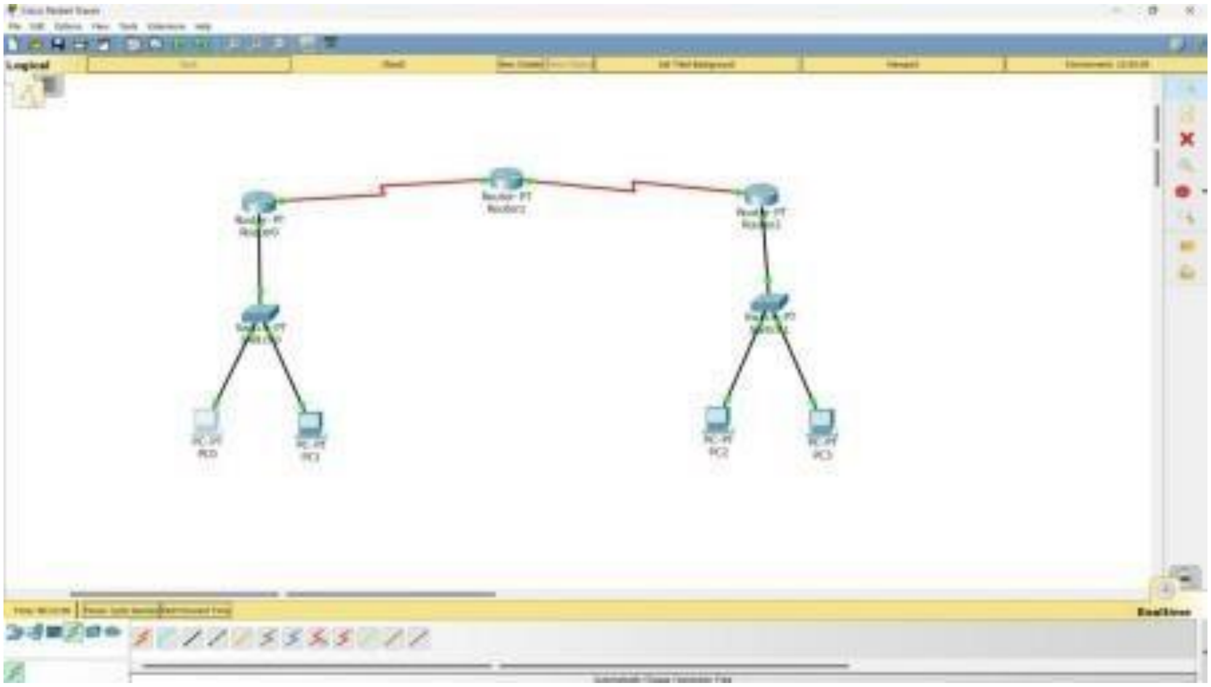
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.1

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),

C:\>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<1ms TTL=127
Reply from 20.0.0.1: bytes=32 time<1ms TTL=127
Reply from 20.0.0.1: bytes=32 time<1ms TTL=127

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Experiment No 3

Aim of the program

Configuring static and default route to the Router

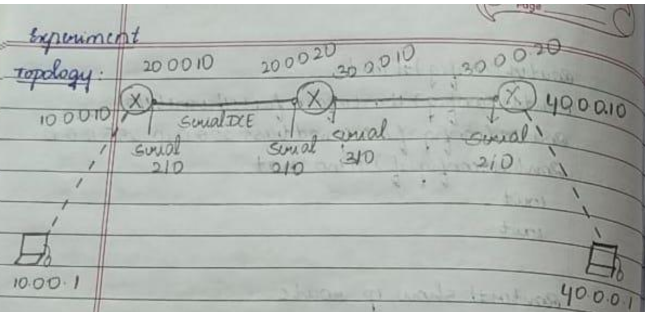
Topology for static routing



Procedure

Experiment

Topology:



Procedure

1) connect three routers serially and two PC's towards the first router and second PC towards second router.
2) set the PC's IP address and gateway settings.
3) Connect the routers using serial DCE and the routers and PC's using copper-cross-over wire.
4) click on the router and go to CLI tab enter the following commands

```
Router>enable
        config terminal
        interface fastEthernet 0/0
        ip address 10.0.0.10 255.0.0.0
        no ~~exit~~ shut
        exit
```

5) continue the process for all the above routers and check the simple PDO.

Observation

1) we set the IP address for each of the following routers till all the wires turn green.

2) when we click on cmd in desktop and enter the command ping 40.0.0.1
It says that the destination is unreached bcz the routers only know about the directly connected networks.
we should manually configure the different routers → this is known as static routing

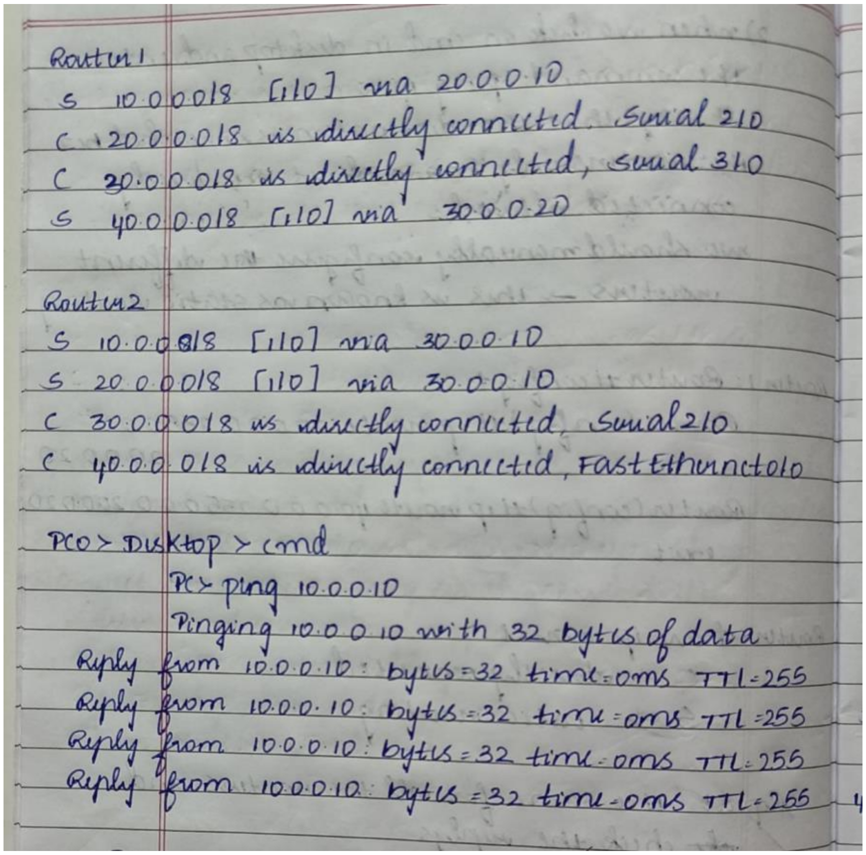
```
Router 1: Router#config t
          Router(config)#ip route 30.0.0.0 255.0.0.0
                                            20.0.0.20
          Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.20
            exit
```

```
Router#show ip route
        C → directly connected
        S → static routing
```

3) Repeat the same for all the routers and ~~ok~~ check the replys.

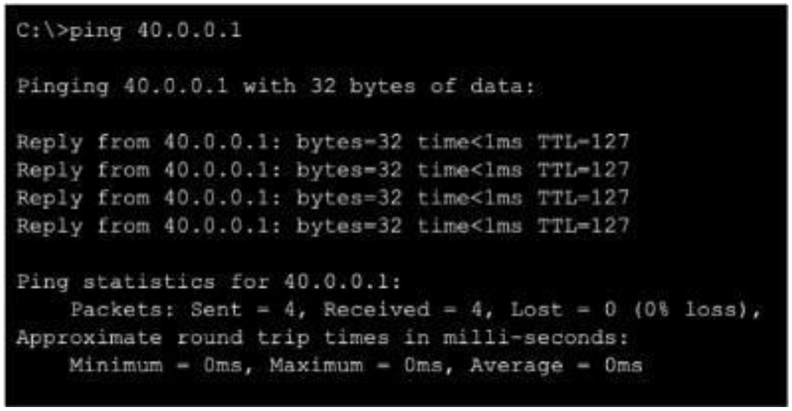
```
cmd: ping 10.0.0.10
```

```
show ip route
Router0
C  10.0.0.0/8 is directly connected, FastEthernet0/0
C  20.0.0.0/8 is directly connected, serial2/0
S  30.0.0.0/8 [1/0] via 20.0.0.20
S  40.0.0.0/8 [1/0] via 20.0.0.20
```



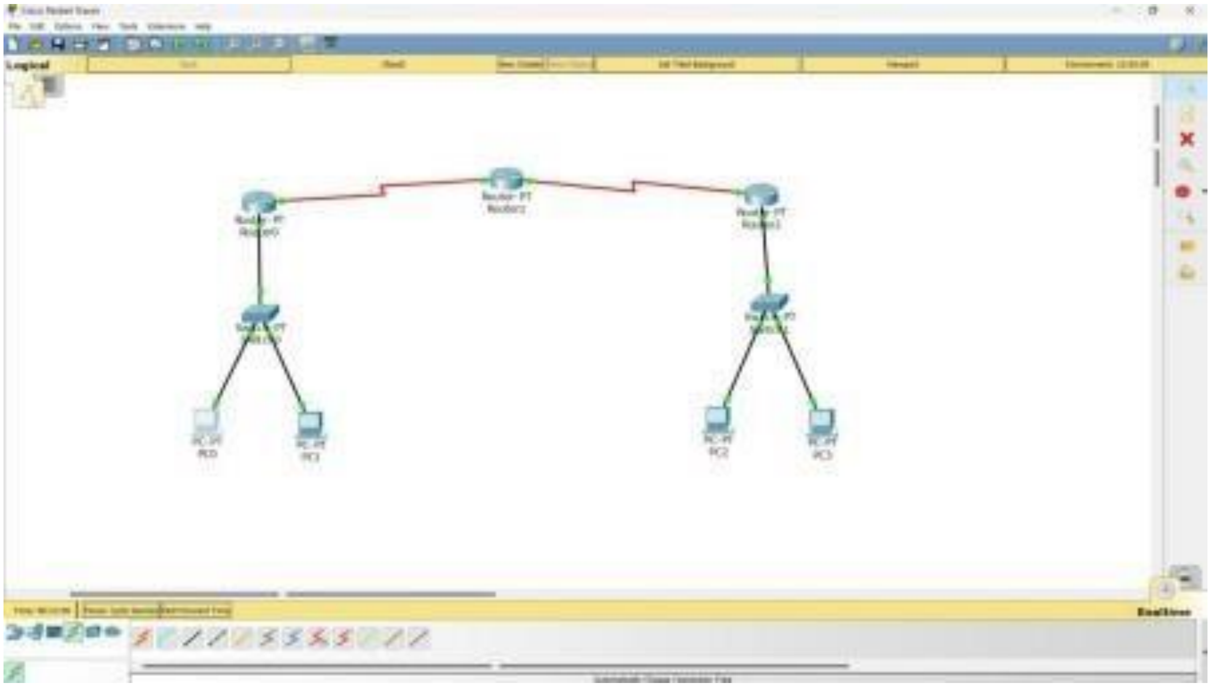
```
Router1
S  10.0.0.0/8 [1/0] via 20.0.0.10
C  20.0.0.0/8 is directly connected, Serial 2/0
C  30.0.0.0/8 is directly connected, Serial 3/0
S  40.0.0.0/8 [1/0] via 30.0.0.20

Router2
S  10.0.0.0/8 [1/0] via 30.0.0.10
S  20.0.0.0/8 [1/0] via 30.0.0.10
C  30.0.0.0/8 is directly connected, Serial2/0
C  40.0.0.0/8 is directly connected, FastEthernet0/0

PC0 > Desktop > cmd
          PC> ping 10.0.0.10
          Pinging 10.0.0.10 with 32 bytes of data
   Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
   Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
   Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
   Reply from 10.0.0.10: bytes=32 time=0ms TTL=255
```

Output

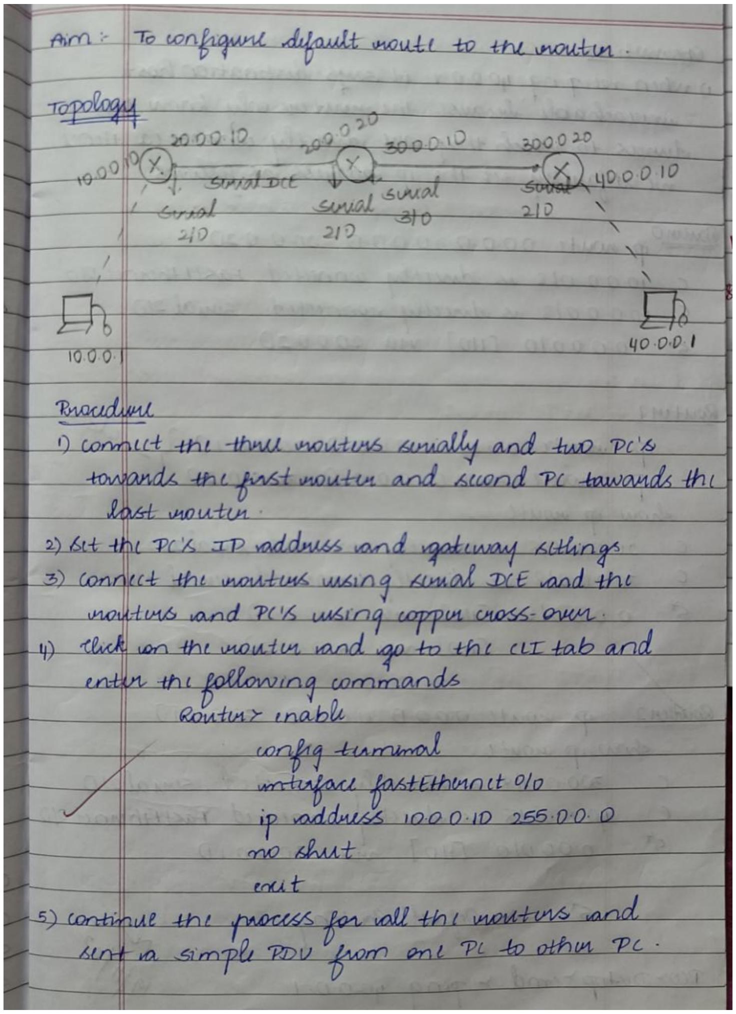


```
C:\>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time<1ms TTL=127
Reply from 40.0.0.1: bytes=32 time<1ms TTL=127
Reply from 40.0.0.1: bytes=32 time<1ms TTL=127
Reply from 40.0.0.1: bytes=32 time<1ms TTL=127

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Topology for default routing



Procedure



Aim :- To configure default route to the router.

Topology

10.0.0.10
20.0.0.10
20.0.0.20
30.0.0.10
30.0.0.20
40.0.0.10
serial DCE
serial 2/0
serial 2/0
serial 3/0
~~Serial~~ 2/0
10.0.0.1
40.0.0.1

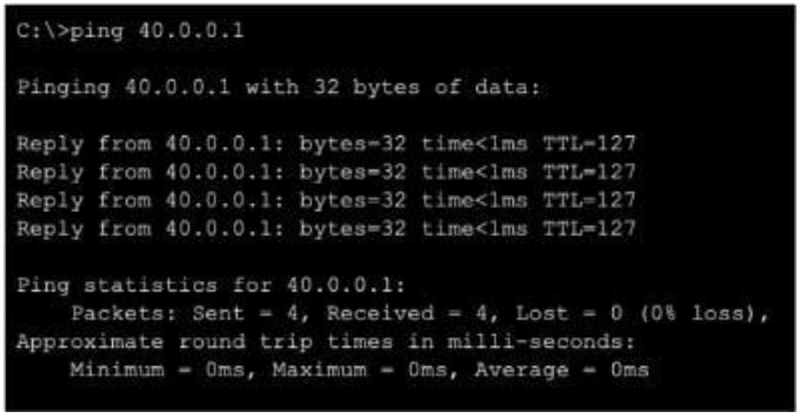
Procedure

1) connect the three routers serially and two PC's towards the first router and second PC towards the last router.
2) set the PC's IP address and gateway settings
3) connect the routers using serial DCE and the routers and PC's using copper cross-over.
4) click on the router and go to the CLI tab and enter the following commands

```
Router> enable
        config terminal
        interface fastEthernet 0/0
        ip address 10.0.0.10 255.0.0.0
        no shut
        exit
```

5) continue the process for all the routers and sent a simple PDU from one PC to other PC.

Output



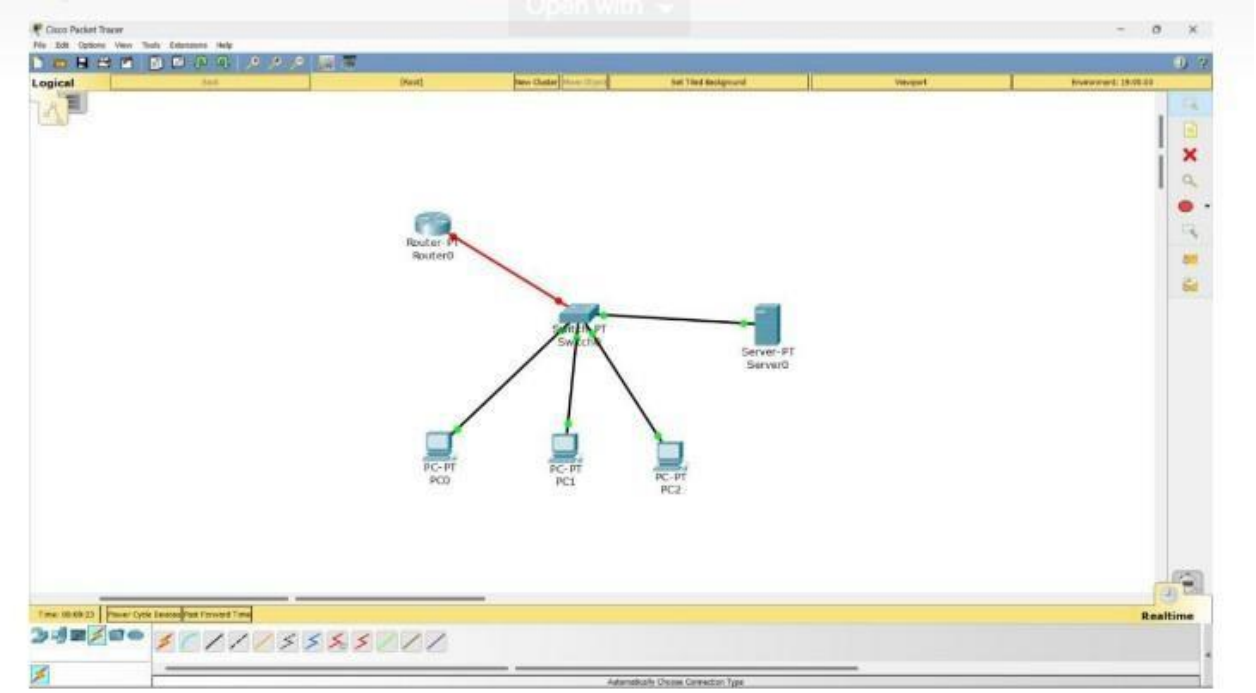
```
C:\>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time<1ms TTL=127
Reply from 40.0.0.1: bytes=32 time<1ms TTL=127
Reply from 40.0.0.1: bytes=32 time<1ms TTL=127
Reply from 40.0.0.1: bytes=32 time<1ms TTL=127

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Experiment No 4

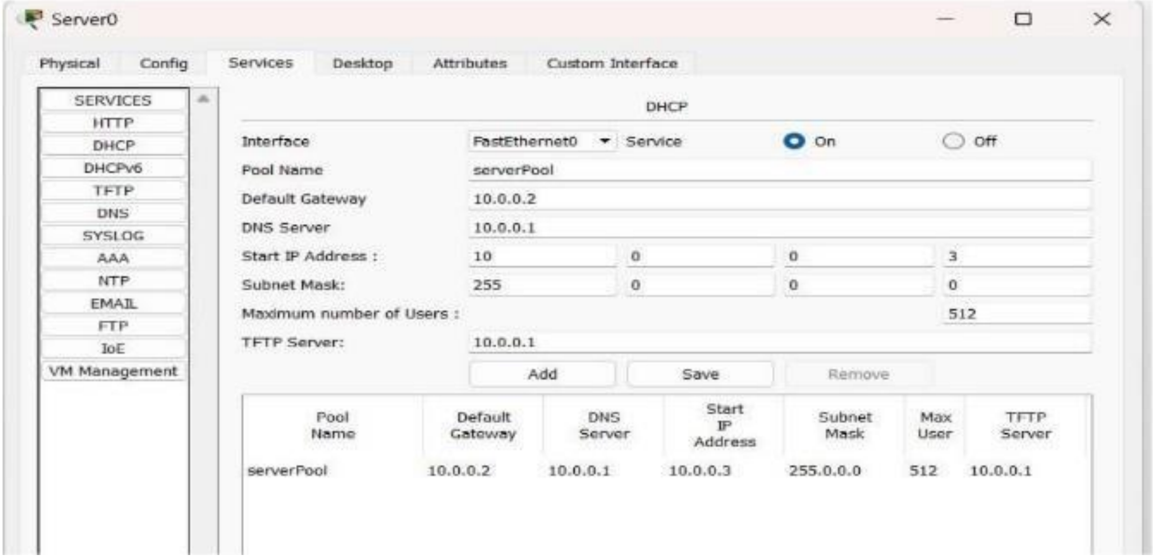
Aim of the program

Configuring DHCP within a LAN in a packet Tracer

Topology



Procedure



DHCP

Interface FastEthernet0 Service ● On ○ Off

Pool Name serverPool

Default Gateway 10.0.0.2

DNS Server 10.0.0.1

Start IP Address : 10 0 0 3

Subnet Mask: 255 0 0 0

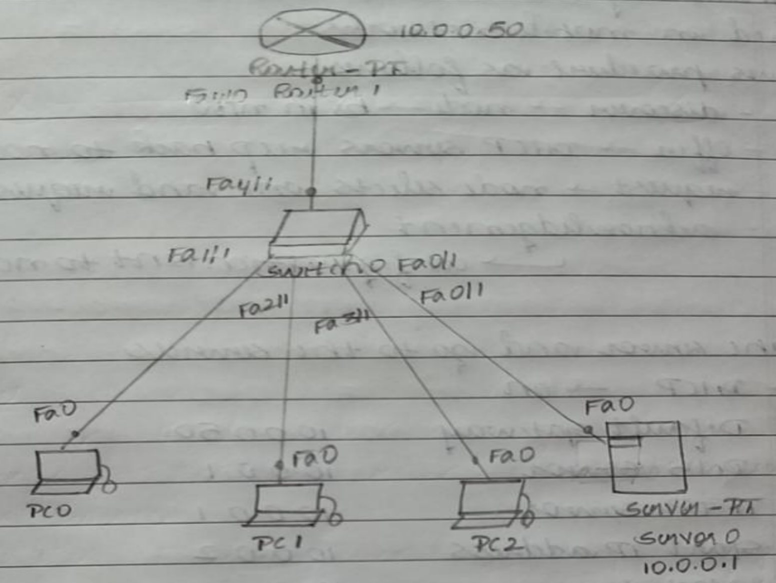
Maximum number of Users : 512

TFTP Server: 10.0.0.1

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server
serverPool	10.0.0.2	10.0.0.1	10.0.0.3	255.0.0.0	512	10.0.0.1

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

Topology



Procedure

1) connect the devices according to above topology.
2) set the IP address of the server as 10.0.0.1 and gateway settings to 10.0.0.50.
3) configure the router using the following commands

```
Router > CLI tab
        enable
        config terminal
        Interface fastethernet4/0
        ip address 10.0.0.50 255.0.0.0
        no shut
exit
```

Dynamic Host configuration Protocol (DHCP)

- It dynamically allocates IP address
- applied in mobile networks
- follows procedure as follows

D - discover → node → bx in n/w
O - Offer → DHCP servers reply back to node
R - request → node selects one and requests
A - acknowledgement
→ configuration sent to node.

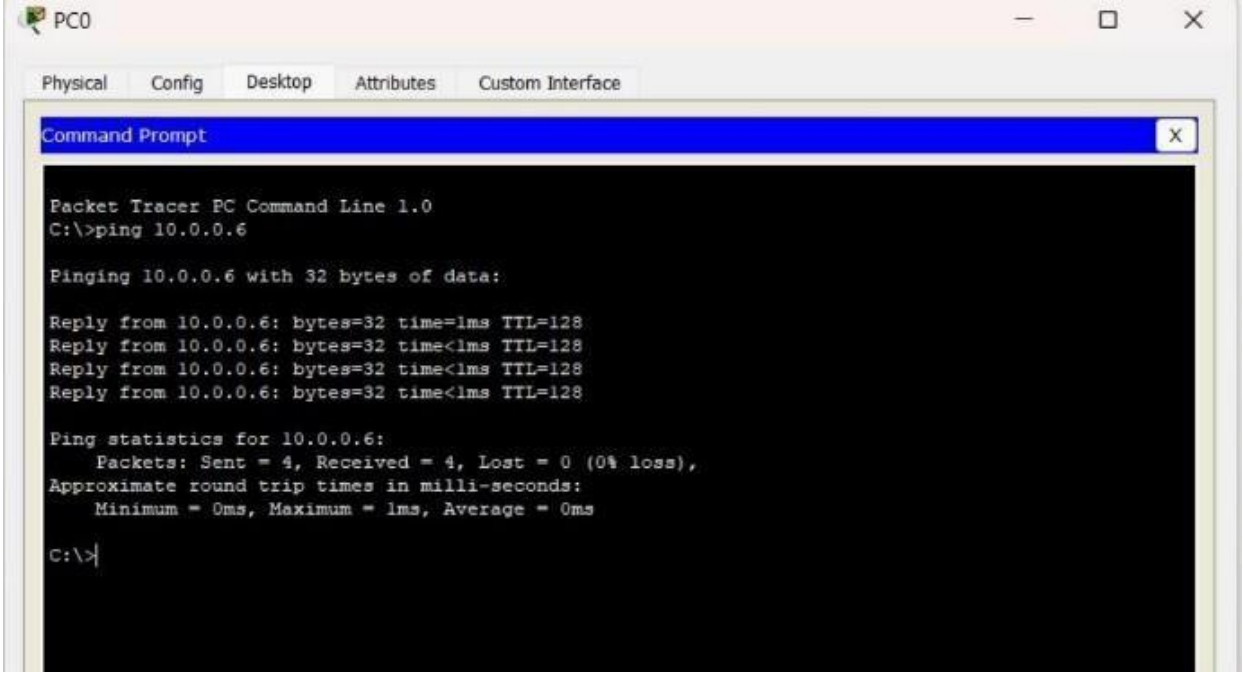
Click on the server and go to the services
→ DHCP → on

Default Gateway	10.0.0.50
DNS Server	10.0.0.1
TFTP Server	10.0.0.1
Start IP address	10.0.0.2
Max. User	500

→ click on $\boxed{\text{save}}$

PC0 → Desktop > IP config > DHCP > DHCP request successful
PC1 → Desktop > IP config > DHCP > DHCP request successful
PC2 → Desktop > IP config > DHCP > DHCP request successful

Output



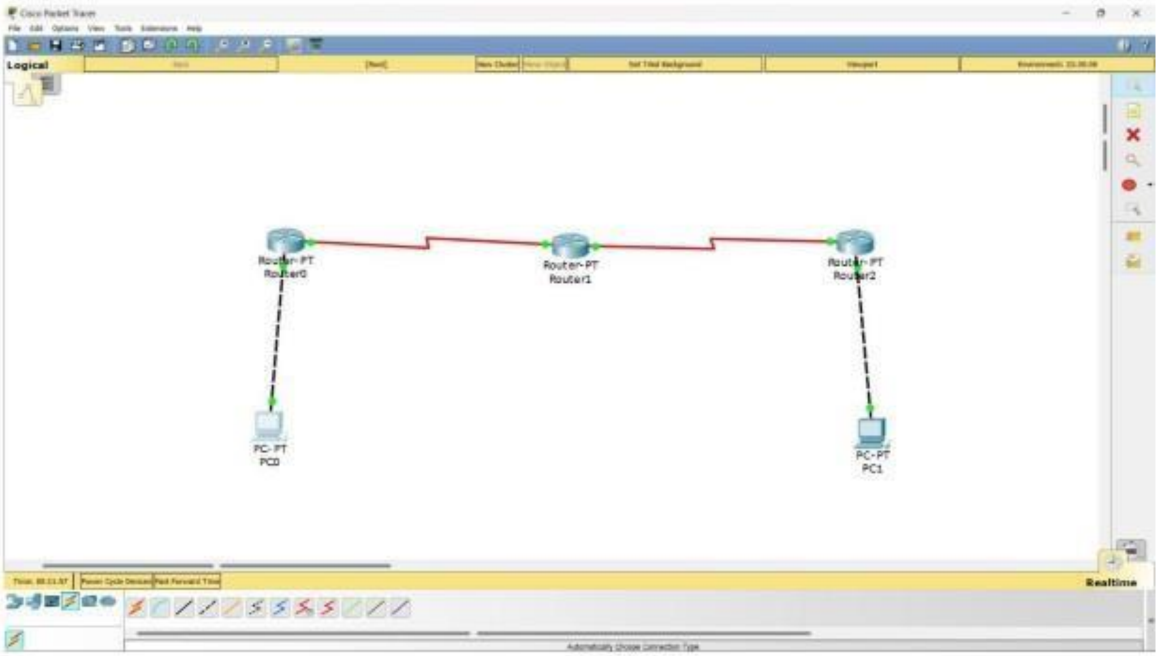
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data:

Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128
Reply from 10.0.0.6: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

Experiment No 5

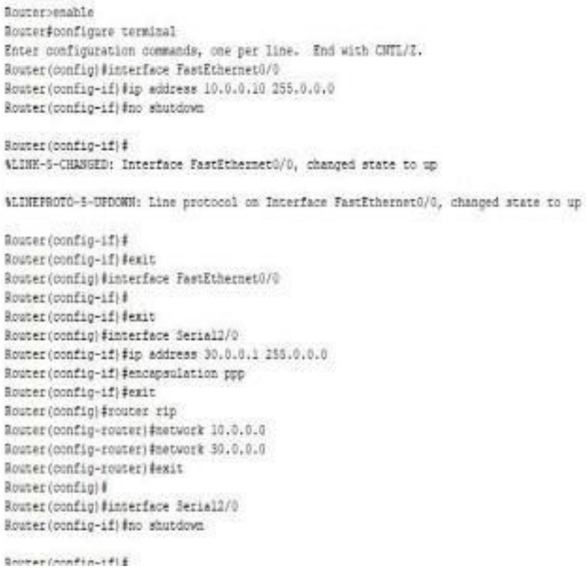
Aim of the program

Configuring RIP Routing Protocol in Routers

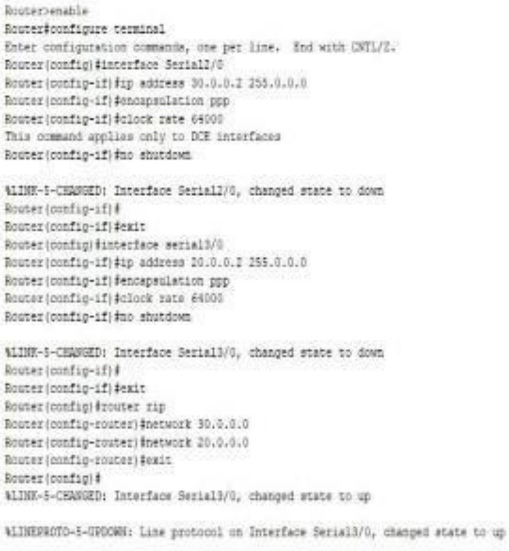
Topology



Procedure



```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.10 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#no shutdown

Router(config-if)#
```

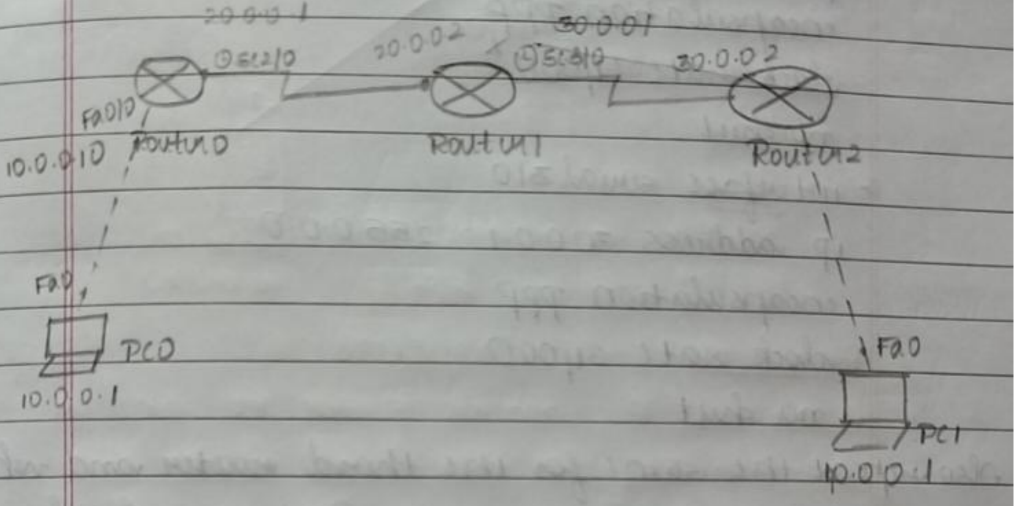


```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
This command applies only to DCE interfaces
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
Router(config-if)#exit
Router(config)#interface serial3/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
```

Routing Information Protocol (RIP)

Aim: To configure RIP Routing Protocol in Routers.

Topology:



Procedure

1. connect the routers and PC's according to the above topology. connect the two routers with a clocked serial DCE connection.
2. Set the IP address and the gateway for both the PC's.
3. Router0 > enable

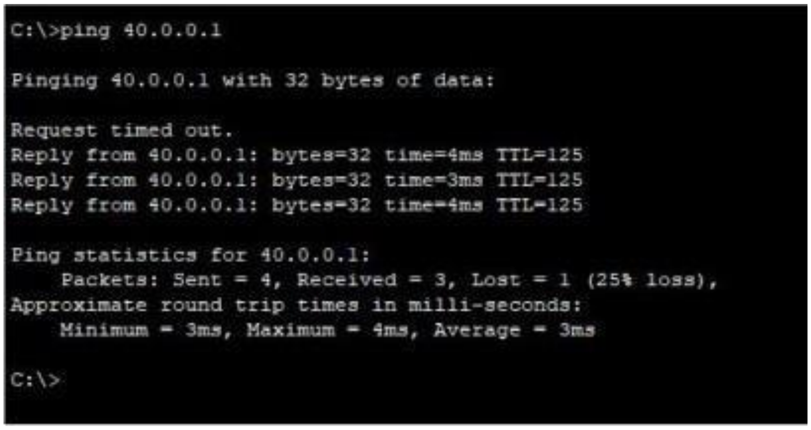
```
        config t
        interface fastethernet 0/0 ~~10.0~~
        ip address 10.0.0.10 255.0.0.0
        no shut
      > interface serial 2/0
        ip address 20.0.0.1 255.0.0.0
        encapsulation PPP (point-to-point protocol)
        clock rate 64000
        no shut
```

```
Router1 > enable
          config t
          interface serial 2/0~~.0.02~~
          ip address 20.0.0.2  255.0.0.0
          encapsulation PPP
          ~~clock rate 64000~~ x
          no shut
        > interface serial 3/0
          ip address 30.0.0.1  255.0.0.0
          encapsulation PPP
          clock rate 64000
          no shut
```

4. Also repeat the same for the third router and when we ping we get Destination host unreachable.

```
Router0 > router rip
        # network 10.0.0.0
        # network 20.0.0.0
          exit

Router1 > router rip
          network 20.0.0.0
          network 30.0.0.0
          exit

Router2 > router rip
          network 30.0.0.0
          network 40.0.0.0
          exit
```

Output:



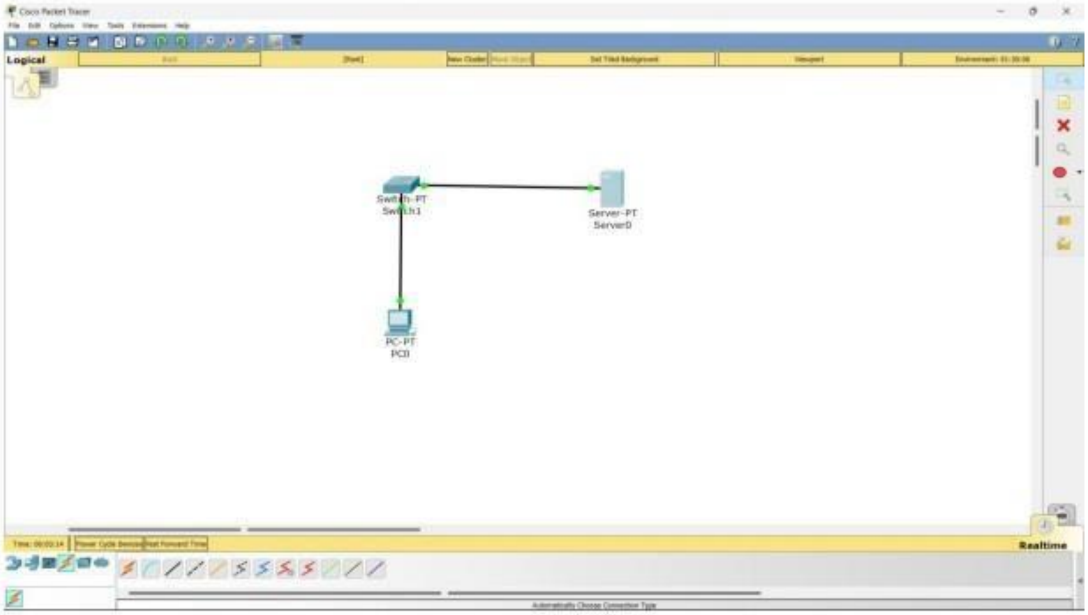
```
C:\>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125
Reply from 40.0.0.1: bytes=32 time=3ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 4ms, Average = 3ms

C:\>
```

Experiment No 6

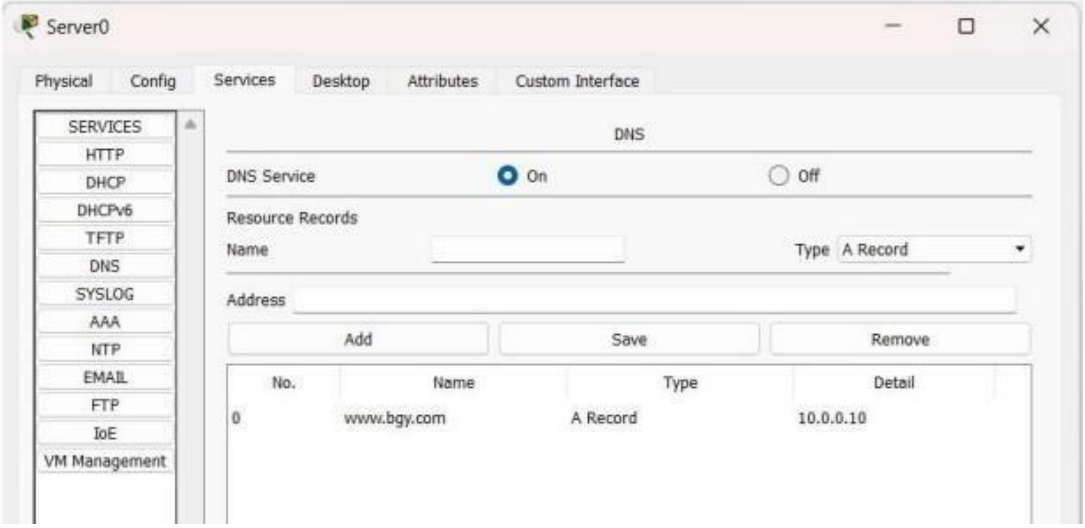
Aim of the program

Demonstration of WEB server and DNS using Packet Tracer

Topology



Procedure



DNS

DNS Service ● On ○ Off

Resource Records

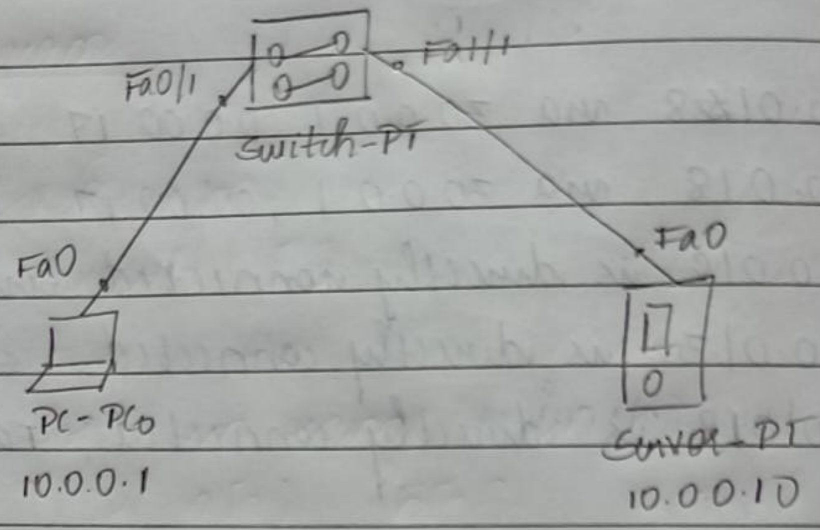
Name

Type A Record

Address

No.	Name	Type	Detail
0	www.bgy.com	A Record	10.0.0.10

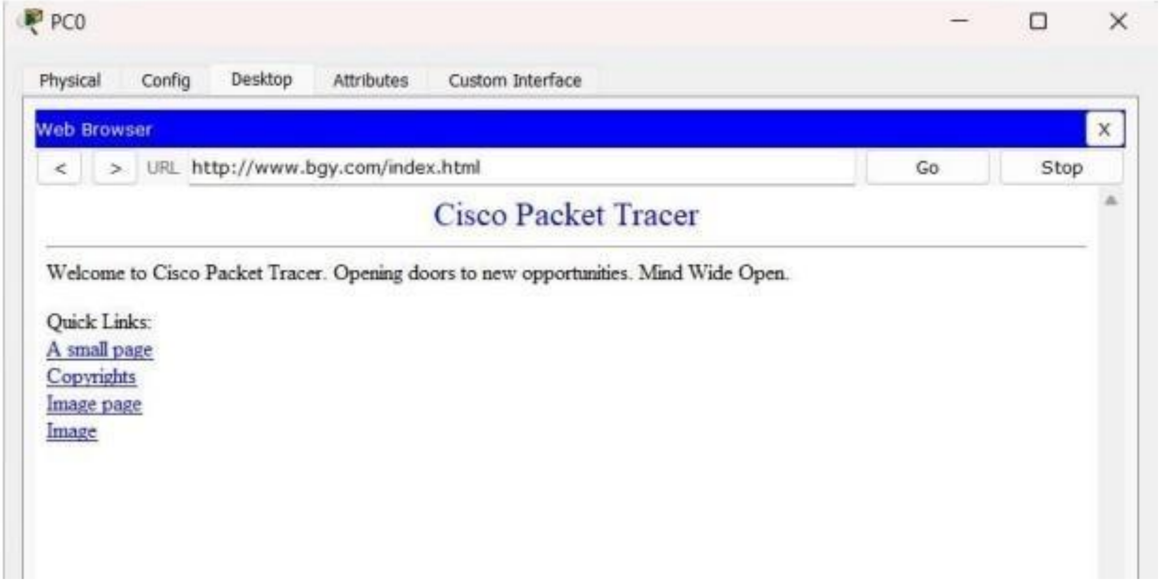
Aim: Demonstration of WEB Server and DNS using packet Tracer.



Procedure

1. A generic PC, generic server-PT and a switch PT are joined as above.
2. Set the IP address for the PC as well as server.
3. In the services tab of the server, HTTP is switched on and DNS is on.
5. In DNS section of server, assign a name and IP address 10.0.0.10 and click on ADD
6. PC > Desktop > Web browser
   a) Enter IP address of server (10.0.0.10) in the URL and click enter.

Output



http://www.bgy.com/index.html

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

Cycle-2

Experiment No 1

Aim of the Experiment

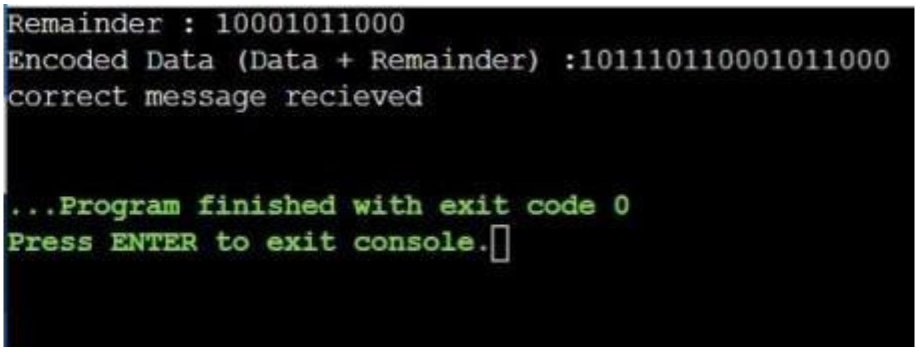
Write a program for error detecting code using CRC-CCITT (16-bits).

Code

```
#include<iostream>
#include<string.h>
using namespace std;
int crc(char *ip,char *op, char *poly, int mode){
   strcpy(op,ip);
   if(mode){
      for(int i=1;i<strlen(poly);i++){
         strcat(op,"0");
      }
   }
   for(int i=0;i<strlen(ip);i++){
      if(op[i] == '1'){
         for(int j=0;j<strlen(poly);j++){
            if(op[i+j] == poly[j]){
               op[i+j] = '0';
            }
            else
               op[i+j] = '1';
         }
      }
   }
   for(int i=0;i<strlen(op);i++)
      if(op[i] == '1')
         return 0;
   return 1;
}
int main()
{
   char ip[50],op[50],recv[50];
   char poly[] = "10001000000100001";
   cout<< "Enter input in binary: "<<endl;
   cin>>ip;
   crc(ip,op,poly,1);
   cout<<"Transmitted message is"<<ip<<op+strlen(ip)<<endl;
```

```
    cout<<"Enter recieved message in binary"<<endl;
    cin>>recv;
    if(crc(recv,op,poly,0))
        cout<<"No errors"<<endl;
    else
        cout<<"Error in data"<<endl;
    return 0;
}
```

Output



```
Remainder : 10001011000
Encoded Data (Data + Remainder) :1011101100010110000
correct message recieved


...Program finished with exit code 0
Press ENTER to exit console.
```

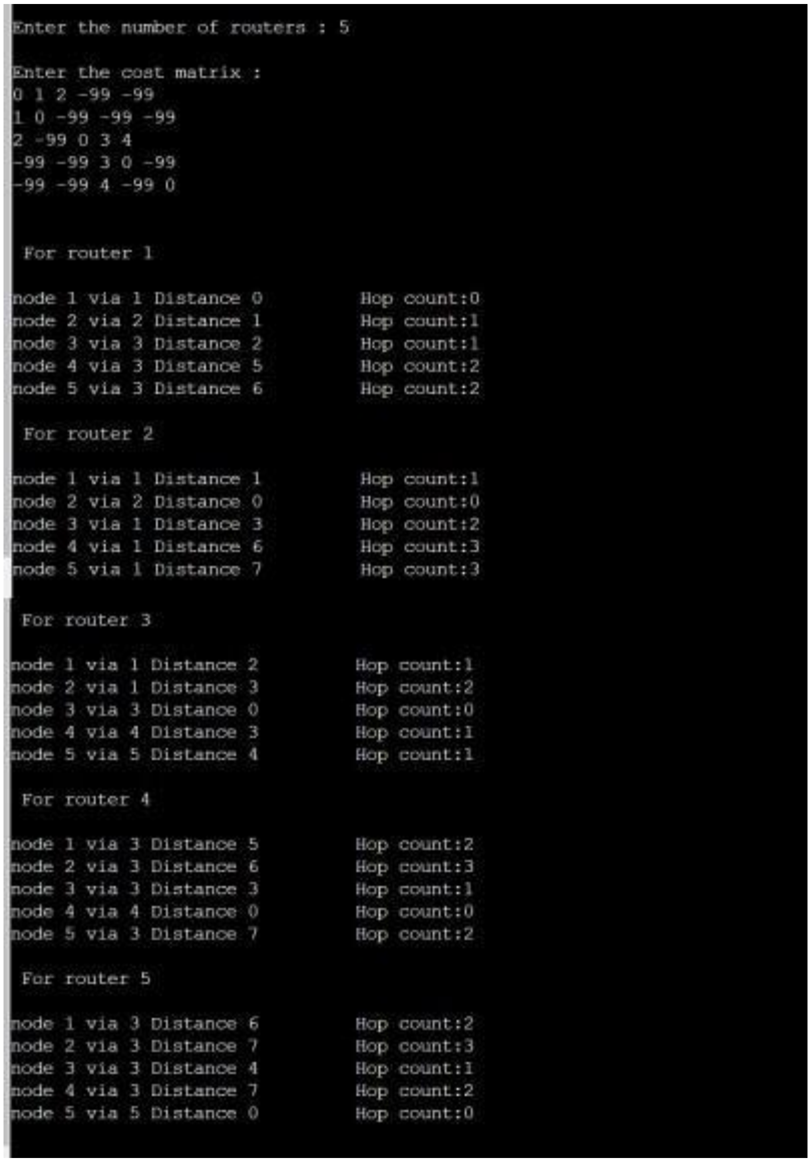
Experiment No 2

Aim of the Experiment

Write a program for distance vector algorithm to find suitable path for transmission.

Code

```
#include <bits/stdc++.h>
using namespace std;
#define MAX 10
int n;
class router {
char adj_new[MAX], adj_old[MAX];
int table_new[MAX], table_old[MAX];
 public:
 router( ){
for(int i=0;i<MAX;i++) table_old[i]=table_new[i]=99;
 }
void copy( ){
for(int i=0;i<n;i++) {
 adj_old[i] =adj_new[i];
 table_old[i]=table_new[i];
 }
 }
int equal( ) {
for(int i=0;i<n;i++)
if(table_old[i]!=table_new[i]||adj_new[i]!=adj_old[i])return 0;
return 1;
 }
void input(int j) {
 cout<<"Enter 1 if the corresponding router is adjacent to router"
<<(char)('A'+j)<<" else enter 99: "<<endl<<" ";
for(int i=0;i<n;i++)
if(i!=j) cout<<(char)('A'+i)<<" ";
 cout<<"\nEnter matrix:";
for(int i=0;i<n;i++) {
if(i==j)
 table_new[i]=0;
else
 cin>>table_new[i];
 adj_new[i]= (char)('A'+i);
 }
 cout<<endl;
 }
void display(){
 cout<<"\nDestination Router: ";
for(int i=0;i<n;i++) cout<<(char)('A'+i)<<" ";
 cout<<"\nOutgoing Line: ";
for(int i=0;i<n;i++) cout<<adj_new[i]<<" ";
 cout<<"\nHop Count: ";
for(int i=0;i<n;i++) cout<<table_new[i]<<" ";
 }
void build(int j) {
for(int i=0;i<n;i++)
for(int k=0;(i!=j)&&(k<n);k++)
if(table_old[i]!=99)
if((table_new[i]+table_new[k])<table_new[k]) {
 table_new[k]=table_new[i]+table_new[k];
 adj_new[k]=(char)('A'+i);
 }
 }
} r[MAX];
void build_table( ) {
int i=0, j=0;
while(i!=n) {
for(i=j;i<n;i++) {
 r[i].copy();
 r[i].build(i);
 }
for(i=0;i<n;i++)
if(!r[i].equal()) {
 j=i;
break;
 }
 }
}
int main() {
 cout<<"Enter the number the routers(<"<<MAX<<"): "; cin>>n;
for(int i=0;i<n;i++) r[i].input(i);
 build_table();
for(int i=0;i<n;i++) {
 cout<<"Router Table entries for router "<<(char)('A'+i)<<":-";
 r[i].display();
 cout<<endl<<endl;
 }
}
```



```
Enter the number of routers : 5

Enter the cost matrix :
0 1 2 -99 -99
1 0 -99 -99 -99
2 -99 0 3 4
-99 -99 3 0 -99
-99 -99 4 -99 0


 For router 1

node 1 via 1 Distance 0         Hop count:0
node 2 via 2 Distance 1         Hop count:1
node 3 via 3 Distance 2         Hop count:1
node 4 via 3 Distance 5         Hop count:2
node 5 via 3 Distance 6         Hop count:2

 For router 2

node 1 via 1 Distance 1         Hop count:1
node 2 via 2 Distance 0         Hop count:0
node 3 via 1 Distance 3         Hop count:2
node 4 via 1 Distance 6         Hop count:3
node 5 via 1 Distance 7         Hop count:3

 For router 3

node 1 via 1 Distance 2         Hop count:1
node 2 via 1 Distance 3         Hop count:2
node 3 via 3 Distance 0         Hop count:0
node 4 via 4 Distance 3         Hop count:1
node 5 via 5 Distance 4         Hop count:1

 For router 4

node 1 via 3 Distance 5         Hop count:2
node 2 via 3 Distance 6         Hop count:3
node 3 via 3 Distance 3         Hop count:1
node 4 via 4 Distance 0         Hop count:0
node 5 via 3 Distance 7         Hop count:2

 For router 5

node 1 via 3 Distance 6         Hop count:2
node 2 via 3 Distance 7         Hop count:3
node 3 via 3 Distance 4         Hop count:1
node 4 via 3 Distance 7         Hop count:2
node 5 via 5 Distance 0         Hop count:0
```

Experiment No 3

Aim of the Experiment

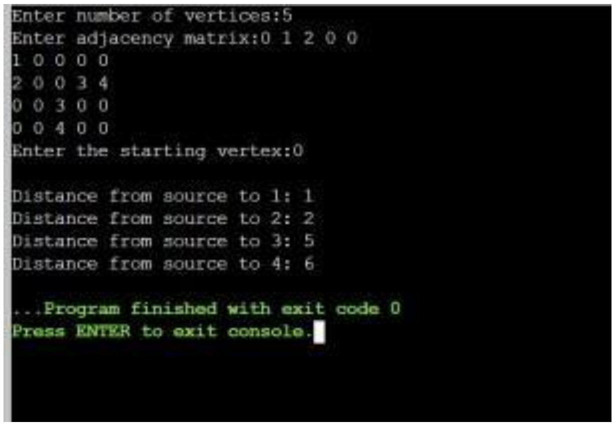
Implement Dijkstra’s algorithm to compute the shortest path for a given topology.

```
#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;
}
```

Code



```
Enter number of vertices:5
Enter adjacency matrix:0 1 2 0 0
1 0 0 0 0
2 0 0 3 4
0 0 3 0 0
0 0 4 0 0
Enter the starting vertex:0

Distance from source to 1: 1
Distance from source to 2: 2
Distance from source to 3: 5
Distance from source to 4: 6

...Program finished with exit code 0
Press ENTER to exit console.
```

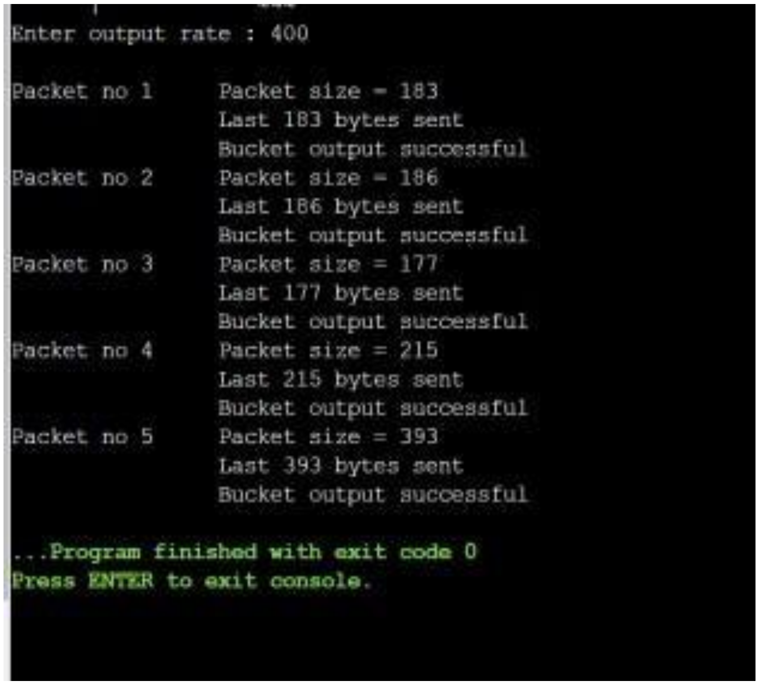
Experiment No 4

Aim of the Experiment

Write a program for congestion control using leaky bucket algorithm.

```
   #include<stdio.h>

   int main(){
   int incoming, outgoing, buck_size, n, store = 0;
   printf("Enter bucket size, outgoing rate and no of inputs: ");
   scanf("%d %d %d", &buck_size, &outgoing, &n);

    while (n != 0) {
    printf("Enter the incoming packet size : ");
    scanf("%d", &incoming);
    printf("Incoming packet size %d\n", incoming);
   if (incoming <= (buck_size - store)){
   store += incoming;
   printf("Bucket buffer size %d out of %d\n", store, buck_size);
  } else {
  printf("Dropped %d no of packets\n", incoming - (buck_size - store));
  printf("Bucket buffer size %d out of %d\n", store, buck_size);
  store = buck_size;
  }
  store = store - outgoing;
 printf("After outgoing %d packets left out of %d in buffer\n", store, buck_size);
 n--;
 }
}
```

OUTPUT:



```
Enter output rate : 400

Packet no 1     Packet size = 183
                Last 183 bytes sent
                Bucket output successful
Packet no 2     Packet size = 186
                Last 186 bytes sent
                Bucket output successful
Packet no 3     Packet size = 177
                Last 177 bytes sent
                Bucket output successful
Packet no 4     Packet size = 215
                Last 215 bytes sent
                Bucket output successful
Packet no 5     Packet size = 393
                Last 393 bytes sent
                Bucket output successful

...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No 5

Aim of the Experiment

Code

Server:

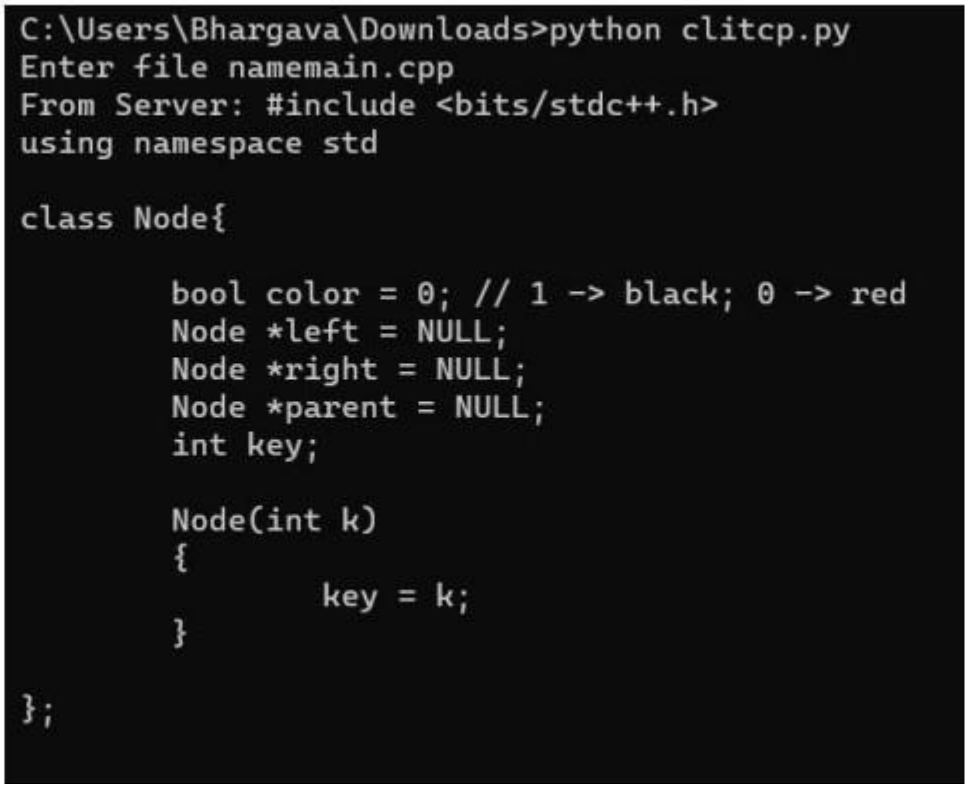
```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Client:

```
 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()
```

Output



```
C:\Users\Bhargava\Downloads>python clitcp.py
Enter file namemain.cpp
From Server: #include <bits/stdc++.h>
using namespace std

class Node{

        bool color = 0; // 1 -> black; 0 -> red
        Node *left = NULL;
        Node *right = NULL;
        Node *parent = NULL;
        int key;

        Node(int k)
        {
                key = k;
        }

};
```

Experiment No 6

Aim of the Experiment

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.

Code

Server:

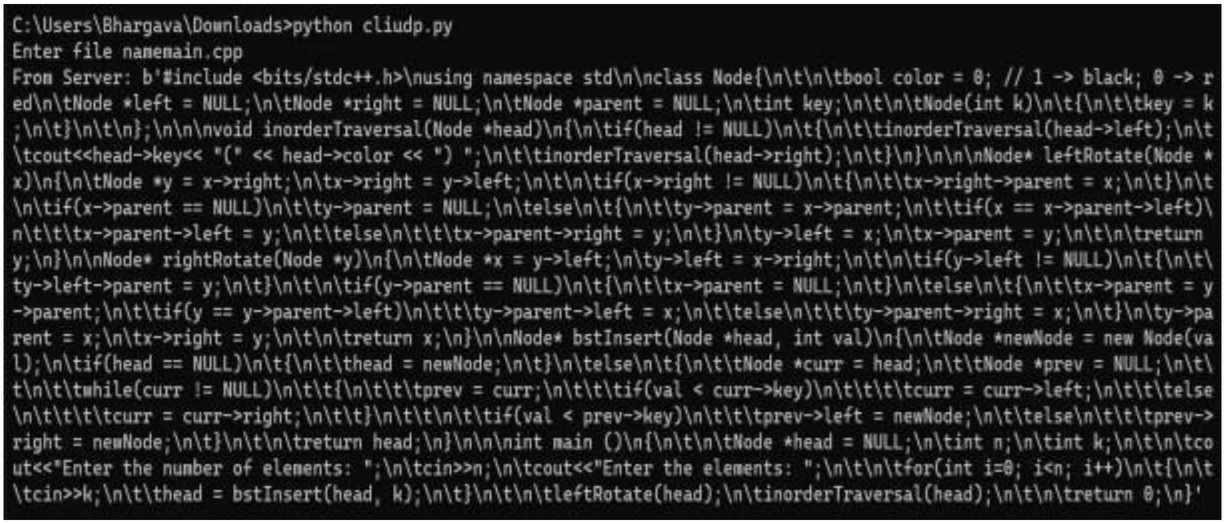
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
l=file.read(2048)
serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
```

Client:

```
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()
```

Output



```
C:\Users\Bhargava\Downloads>python cliudp.py
Enter file namemain.cpp
From Server: b'#include <bits/stdc++.h>\nusing namespace std\n\nclass Node{\n\t\n\tbool color = 0; // 1 -> black; 0 -> r
ed\n\tNode *left = NULL;\n\tNode *right = NULL;\n\tNode *parent = NULL;\n\tint key;\n\t\n\tNode(int k)\n\t{\n\t\tkey = k
;\n\t}\n\t\n};\n\n\nvoid inorderTraversal(Node *head)\n{\n\tif(head != NULL)\n\t{\n\t\tinorderTraversal(head->left);\n\t
\tcout<<head->key<< "(" << head->color << ") ";\n\t\tinorderTraversal(head->right);\n\t}\n}\n\n\nNode* leftRotate(Node *
x)\n{\n\tNode *y = x->right;\n\tx->right = y->left;\n\t\n\tif(x->right != NULL)\n\t{\n\t\tx->right->parent = x;\n\t}\n\t
\n\tif(x->parent == NULL)\n\t\ty->parent = NULL;\n\telse\n\t{\n\t\ty->parent = x->parent;\n\t\tif(x == x->parent->left)\
n\t\t\tx->parent->left = y;\n\t\telse\n\t\t\tx->parent->right = y;\n\t}\n\ty->left = x;\n\tx->parent = y;\n\t\n\treturn
y;\n}\n\nNode* rightRotate(Node *y)\n{\n\tNode *x = y->left;\n\ty->left = x->right;\n\t\n\tif(y->left != NULL)\n\t{\n\t\
ty->left->parent = y;\n\t}\n\t\n\tif(y->parent == NULL)\n\t{\n\t\tx->parent = NULL;\n\t}\n\telse\n\t{\n\t\tx->parent = y
->parent;\n\t\tif(y == y->parent->left)\n\t\t\ty->parent->left = x;\n\t\telse\n\t\t\ty->parent->right = x;\n\t}\n\ty->pa
rent = x;\n\tx->right = y;\n\t\n\treturn x;\n}\n\nNode* bstInsert(Node *head, int val)\n{\n\tNode *newNode = new Node(va
l);\n\tif(head == NULL)\n\t{\n\t\thead = newNode;\n\t}\n\telse\n\t{\n\t\tNode *curr = head;\n\t\tNode *prev = NULL;\n\t\
t\n\t\twhile(curr != NULL)\n\t\t{\n\t\t\tprev = curr;\n\t\t\tif(val < curr->key)\n\t\t\t\tcurr = curr->left;\n\t\t\telse
\n\t\t\t\tcurr = curr->right;\n\t\t}\n\t\t\n\t\tif(val < prev->key)\n\t\t\tprev->left = newNode;\n\t\telse\n\t\t\tprev->
right = newNode;\n\t}\n\t\n\treturn head;\n}\n\n\nint main ()\n{\n\t\n\tNode *head = NULL;\n\tint n;\n\tint k;\n\t\n\tco
ut<<"Enter the number of elements: ";\n\tcin>>n;\n\tcout<<"Enter the elements: ";\n\t\n\tfor(int i=0; i<n; i++)\n\t{\n\t
\tcin>>k;\n\t\thead = bstInsert(head, k);\n\t}\n\t\n\tleftRotate(head);\n\tinorderTraversal(head);\n\t\n\treturn 0;\n}'
```