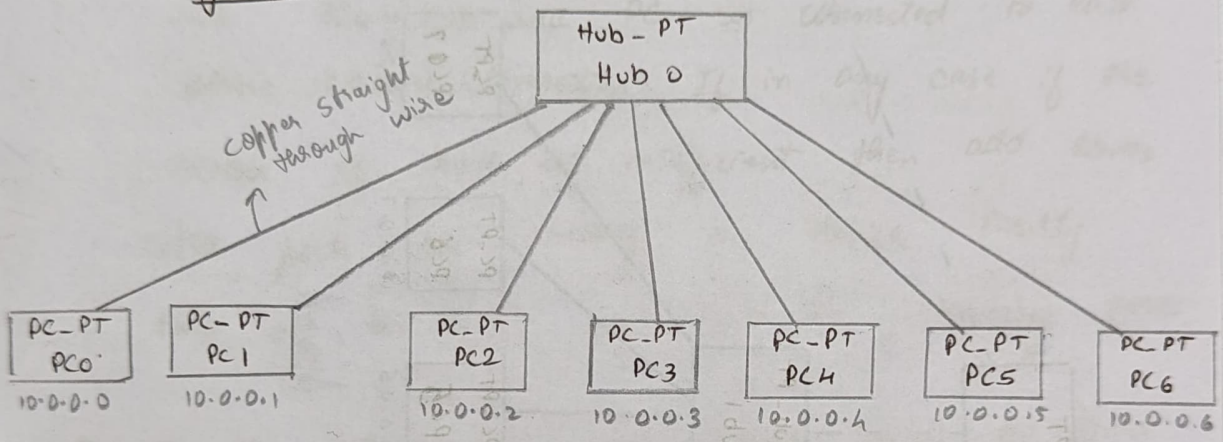
# Lab - 1.

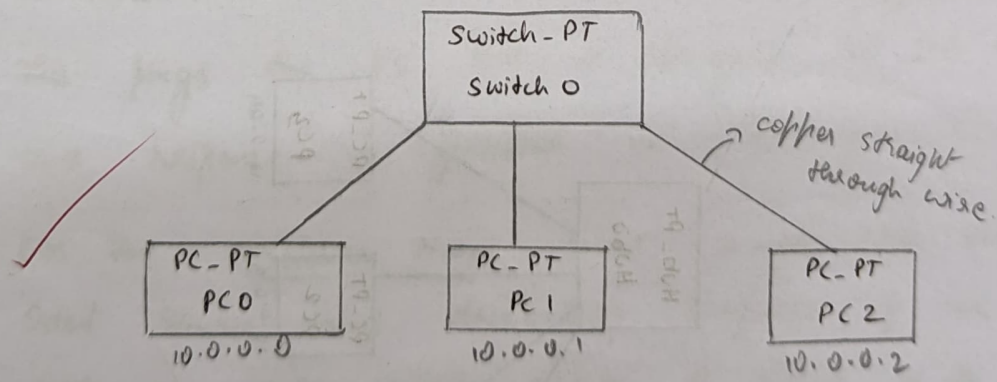
**Aim :** Creating a topology and simulate sending a simple PDU from source to destination using simple hub and switch as connecting domain.

## Topology :

### Using hub :

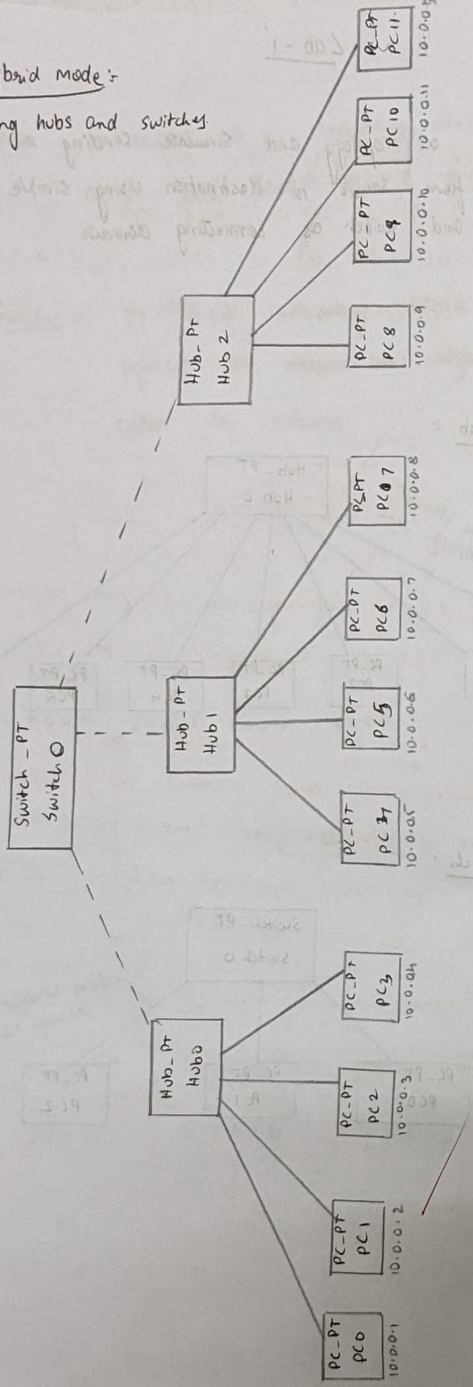



### Using switch :



Hybrid mode :

using hubs and switches




Procedure :

while ~~creating~~ using a hub :

⊙ Insert a generic hub and add seven PC's to a workspace. Set the IP address for each PC in configuration tab. Make sure that all the IP address are different from each other. Connect all the PC's to the hub using a copper straight wire. The hub and PC is connected to each other's ethernet connection. If in any case if the number of ports is insufficient then add some extra ports by clicking on device. Firstly turn off the device and add the necessary ports.

⊙ In the "Real time" tab select source PC and in desktop tab, select command prompt option in the command prompt type. Now use a ping command in the command prompt : ping 10.0.0.3 This pings the PC that is assigned to IP 10.0.0.3 and response is generated in PC0.

⊙ For the simulation time select simple PDU and select source and destination computer. if we click on auto capture option it allows us to see how the ~~ports~~ are transferred to and from the ~~dev~~ device.

⊙ Using a switch:-

→ Add a generic switch and then PC's to the workspace. Configure the IP address of each PC's in the configuration tab. Make sure that the IP is different for each device. Connect all PC's to switch using a copper straight wire. If no of ports are insufficient then add extra ports by clicking on device. Firstly turn off device and then add the necessary ports.

→ In the "Real time" tab select the source PC and in the desktop tab select command prompt option. Then in the command prompt ping the destination PC by inserting its IP.

→ In the "Simulation time" select simple PDU and select source and destination computer. If we click on auto capture option, it allows us to see how packets are transferred

Hybrid mode :

⊙ Add a switch (3 hubs + 12 PC's) to workspace. Then connect all the 3 hubs to switch and 4 PC's to each of the hub using a copper cross over and copper straight wires respectively. Set the IP for each of the PC in configure tab and note the IP of each below the PC.

⊙ In the "Real time mode" click on the PC you want to send packet. Open its command prompt specify destination PC by specifying its IP address. Response is sent by destination PC to source PC.

⊙ In the "Simulation mode" add a simple PDU by selecting the pair of PC's and click on auto capture from the right panel.

Observation :

→ Hub :

Learning Outcomes :- When source PC sends a packet in network the hub source sends it to all the end devices in the network and where it matches the specified address accepts the packet and acknowledge it.

The remaining nodes ignore the message.

Communication between hub and the end devices is established through copper straight wires as they belong to different layers. Number of ports can be added if needed by clicking on the device and adding the necessary ports.

Result :- PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1 : byte = 32 time = 0 ms

Reply from 10.0.0.1 : byte = 32 time = 0 ms

Reply from 10.0.0.1 : byte = 32 time = 0 ms

Reply from 10.0.0.1 : byte = 32 time = 0 ms

Ping statistics for 10.0.0.1

Packets sent = 4, received = 4, lost = 0.

Switches :-

Learning Outcomes :- When source device sends a message to the switch and once a connection is made, which takes some time called the "learning time" the switch receives the packet. It broadcasts the packet to all the connected devices to locate the destination.

Once the destination is located, the message is sent only to that device.

Connection between the switch and end device is established using copper straight wire through as they belong to different network layers. And the number of ports can be added if needed, by clicking on device and turning it off and adding the necessary ports and turn the device on again.

Result :-

PC > Ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data.

Reply from 10.0.0.1 : bytes = 32 time = 0 ms

Reply from 10.0.0.1 : bytes = 32 time = 0 ms

Reply from 10.0.0.1 : bytes = 32 time = 0 ms

Reply from 10.0.0.1 : bytes = 32 time = 0 ms

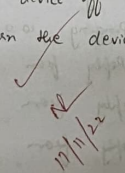
Ping statistics for 10.0.0.1

Packets sent = 4, received = 4, lost = 0

## Hybrid mode

**Learning Outcomes :-** Switch and hub are connected through copper cross over wire as they belong to the same network layer but PC and hubs are connected through copper straight wire as they belong to different network layers.

A Message from the source PC to destination PC is sent through the hub which then sends to all its connected PC's and the switch. The switch then sends the message to all its connected PC. The destination PC acknowledges that it has received the message by sending a acknowledged back to the source PC. The number of ports can be added if needed by clicking on device, turn the device off and add the necessary ports then turn the device on.



## Exp - 2

**Aim :-** Configuring IP address to routers in packet tracer. Explore ping, responses destination unreachable, reply, request, timed out.

**Procedure :-**

1. End devices are connected to router
2. IP address is configured to end devices
3. Config IP address and 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 for end devices
5. End devices and interface are pinged to check connection.

**Topology :-** Star topology

**Result :-** Successfully pinged end devices.

PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data.

Reply from 10.0.0.1 bytes = 32 time = 6ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 2ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 5ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 3ms 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 = 2ms maximum = 6ms Avg = 4ms.

PC > ping 20.0.0.1

pinging 20.0.0.1 with 32 byte of data:

Request timed out

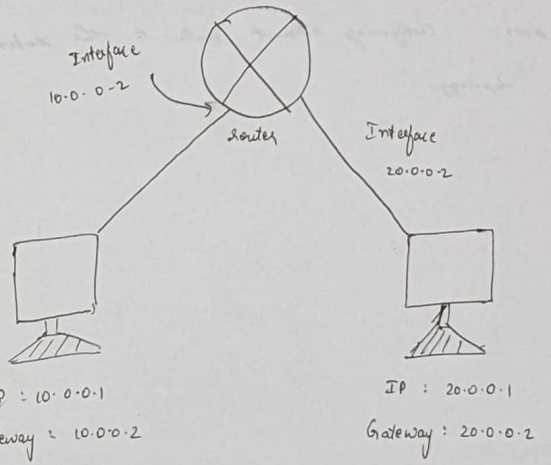
Request timed out

Request timed out

Request timed out

Ping statistics for 20.0.0.1:

Packets : sent = 4, Received = 0, lost = 4 (100% Loss)




Observation :

① When we configure both end devices and router with appropriate ip address and by configuring subnet mask of interface of router as 255.0.0.0 and gateway of PC0 is set as 10.0.0.2 which is of Fa 0/0 interface followed by same for PC1.

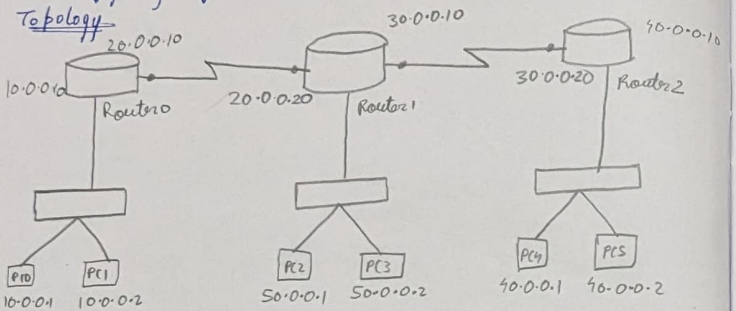
② Then we could [redacted] successfully ping

③ When gateway of end devices is not configured then we get request time out.

# LAB - 3

AIM: Configuring default route to the router

Topology




PROCEDURES

1. Use 3 gen routers & 6 gen PCs in the workspace along with 3 switches
2. Place a rot for each device & specify the IP Address.
3. Use Copper straight wires to Connect router & switch.
4. Use Copper straight wires to connect Switch PC has 3 attributes
5. Click on a PC to set Attributes for a PC, each PC IP Address, & gateway. This needs to be done for all 6 PCs.
6. For Router 1 the Config is done in the CLI. The IP Address & subnet mask is set. Router 2 is a default router for router 1 & this is done by command IP route 0.0.0.0 0.0.0.0 40.0.0.2
7. Router 2 IP Address & subnet masks are set for All 3 Interfaces. It has static Routing done by commands
8. Router 3 is Config in both Interface with IP Address & subnet mask. The default Router for router 3 is router2.
9. Ping command is Executed from 10.0.0.1 to 20.0.0.1 & from 10.0.0.1 to 30.0.0.2

OBSERVATION

- 1 router have 2 default routers
- Default router for Router 3 is Also Middle Router
- Middle Router is Made default then there are chances of packets Going to switch are sent to the Router

RESULT

Ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

Request timed out

Reply from 30.0.0.2 bytes = 32, time = 4ms TTL=125

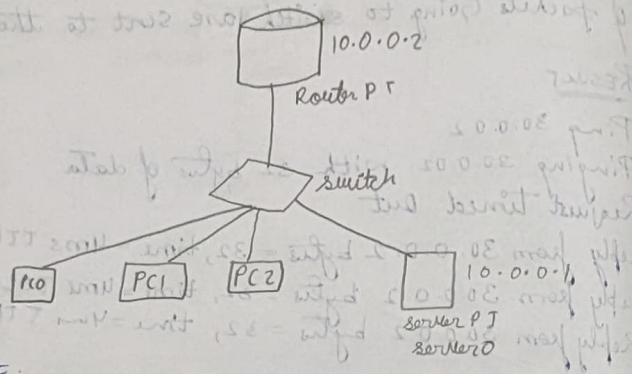
Reply from 30.0.0.2 bytes = 32, time = 4ms TTL=125

Reply from 30.0.0.2 bytes = 32, time = 4ms TTL=125

6/12/22

## LAB4

Aim: Configuring DHCP within a LAN in a packet tracer.




PROCEDURE:

1. Use the generic router one generic switch one generic server & 3 generic PCs in the workspace.
2. Use copper straight wire to connect PCs & server.
3. Use fibre to connect switch & router.
4. Configure the servers by Adding IP address, subnet mask & gateway.
5. Config the router by setting IP address, subnet mask & executing this command in cli, ip route 10.0.0.2 255.0.0.0
6. Click on server & Go to services, & select DHCP, switch it on & Add the gateway server. ip address & subnet mask
7. Click on the 1st PC & to IP config. Select DHCP Repeat this for all.
8. Ping Command is Executed from 10.0.0.3 to 10.0.0.6

OBSERVATION

Pool of ip addresses exit from which IP addresses can be dynamically allocated.
This is called dynamic Host Configuration Protocol (DHCP)

RESULT

Ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

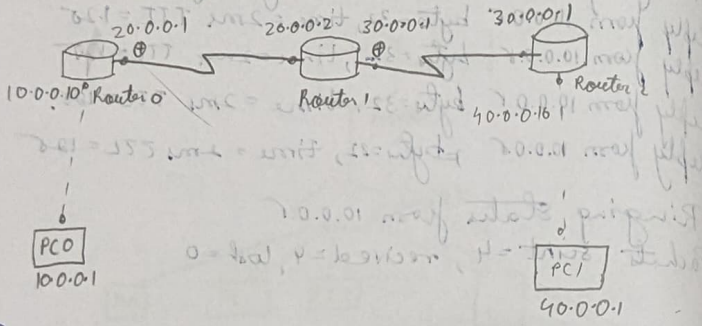
Reply from 10.0.0.5 bytes=32, time=2ms TTL=128
Reply from 10.0.0.5 bytes=32, time=2ms TTL=128
Reply from 10.0.0.5 bytes=32, time=2ms TTL=128
Reply from 10.0.0.5 bytes=32, time=2ms TTL=128

Pinging status from 10.0.0.5

Packets sent=4, recived=4, lost=0

# LAB 6

**AIM:** Configuring RIP Routing Protocol In Routers




**Procedure**

- Use 3 generic routers, 2 generic PC & place routers to indicate respective IP addresses.
- Use serial DCE cable to connect Routers and use Copper cross cable to connect PC with router1 & router3.
- Set IP addresses, gateway & subnetmask as 10.0.0.1 10.0.0.10, 255.0.0.0 for PC0 set 40.0.0.1, 50.0.0.10 255.0.0.0 for PC1
- Interface PC0 & router1
  - → Interface fastEthernet 0/0
  - → ip address 20.0.0.10 255.0.0.0
  - → No shut.
- for Interfacing serial 1/0 of router 1
  - → Interface serial 2/0
  - → Encapsulation PPP
  - → Clock rate 64000
  - → no shut.
- Use above commands for Interfacing Router which has clock symbol in cable has to it & for other Interfaces.
- Once all the lights are turned green follow the commands below each router
  - → router rip
  - → network 10.0.0.0
  - → network 20.0.0.0
  - → Exit
- Repeat the same command for router 2 & 3

**OBSERVATION**

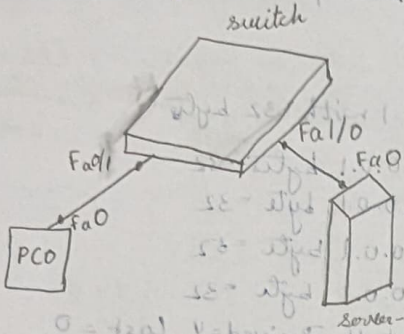
- Use of RIP routing becomes easy when large no of router are present.

**RESULT.**

Pinging 10.0.0.1 with 32 bytes
Reply from 10.0.0.1 bytes = 32
Reply from 10.0.0.1 byte = 32
Reply from 10.0.0.1 byte = 32
Reply from 10.0.0.1 byte = 32
packets : sent=4, Recived=4, Lost = 0

LAB6

AIM: Demonstration of Web server & DNS




PROCEDURE

- Place An Pc, Server & Switch & the set the IP Addresses of PC & server As 10.0.0.1 & 10.0.0.2
- Open Web browser from dashtop tab of the PC & type "http://10.0.0.2"
- It will display a default page.
- Open Server & In services Enable HTTP & change & Add the Contents in the HTTP.
- After Save refresh the browser of PC0 to the updated changes.

Activate DNS:

- Enable DNS on the server to Activate it
- Enter the name & the IP Address needed to be mapped.
- Click on add to add the new mapping.
- Now Give the name In the Web browser you saved to check if its working.

Custom page

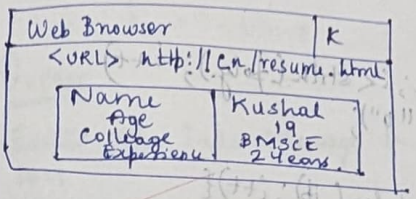
- Create a new page Test.html & save it in http servie.
- Change hyperlink In Index.html to link to the created file.
- Check the output In the webbrowser of the PC0 by cliking on hyperlink.

OBSERVATIONS

- We can view the webpage when we type 'csebmsc' in browser because 10.0.0.2 Address is mapped to the name 'csebmsc'.

Mapping Is required bcoz its difficult for users to remember IPs. Hence Its mapped.

RESULT



LAB7.

AIM: Write a program for Error detection Using CRC

```
#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[7]="100010000001";
                    //x^16+x^12+x^5+1
    cout<<"Enter the input message in binary"<<endl;
    cin>>ip;
    crc(ip, op, poly, 1);
    cout<<"The transmitted message in binary"<<endl;
    op+strlen(ip)<<endl;
    cout<<"Enter the recived message in binary"<<endl;
    cin>>recv;
    if(crc(recv, op, poly, 0))
        cout<<"No Error in data"<<endl;
    else
        cout<<"Error in data tranimission has occured"<<endl;
    return 0;
}
```

OUTPUT

```
Enter the Input Message in binary:
1011
The Transmitted Message is: 1011101100010110101011
1011101100010110101011
No Error in data
```

LAB 8

Write a program for Leaky bucket Algorithm.

```
#Include <unistd.h>
using namespace std;
#define bucketSize 500
void bucketInput (inta, intb)
{
  if (a > bucketSize)
  cout << "\n\t\t Bucket Overflow ";
  else {
    Sleep(5)
    while (a>b){
      cout << "\n\t\t" << b << " bytes outputted";
      a-=b;
      Sleep(5);
    }
    if(a>b)
    cout << "\n\t\t lost" << a << " bytes sent";
  cout <<"\n\t Bucket output Succesfull";
  }
}
int main()
{
int op, pktSize;
  Cout <<" Enter output rate:";
    Cin >>op;
  for (int i=1; i<=5; i++)
    {
    Sleep (rand() % 10);
  pkt size = rand()%700;
Cout <<"\n Packet no" << i <<"\t Packet Size =" << pktSize
  }
cout << endl;
return 0;
}
```

OUTPUT

Enter output rate: 100

Packet no 1 Packet Size = 186
100 bytes outputted
Lost 86 bytes
Bucket output Succesfull.

Packet no 2 Packet Size = 215
100 bytes outputted
100 bytes outputted
Lost 15 bytes sent
Bucket output Succesfull.

Packet no 3 Packet Size = 535
Packet Overflow.

Packet no 4 Packet Size = 492
100 bytes Outputted
100 " "
100 " "
100 " "

Packet 5 Packet Size = 521
Bucket Overflow.

# Lab 9

Implement Dijkstra's Algorithm to compute the shortest path for a given topology.

```
#include <iostream>
using namespace std;
int a[30][30], source, d[30], p[30];
void alg(int a[][30], int n){
    int s[n];
    for(int i=0; i<n; i++){
        d[i] = a[source][i];
        p[i] = source;
        s[i] = 0;
    }
    s[source]=1;
    for(int c=0; c<n; c++){
        int min=9999, u;
        for(int j=0; j<n; j++){
            if(d[j] < min && s[j]!=1){
                min = d[j];
                u=j;
            }
        }
        s[u] = 1;
        for(int i=0; i<n; i++){
            a[i]=min+a[u][i];
            p[i]=u;
        }
    }
}

int main(){
    int n;
    cout<<"Enter the no of vertices:"<<endl;
    cin>>n;
    cout<<"Enter the Adjacency matrix (Enter 9999 for infinity):"<<endl;
    for(int i=0; i<n; i++){
        int k=i;
        while(k!=source){
            cout<<k<<"<-";
            k=p[k];
        }
        cout<<source<<"=";
        cout<<d[i]<<endl;
    }
    return 0;
}
```

OUTPUT:

```
Enter the no of vertices
5
Enter the Adjacency matrix (Enter 9999 for Infinity):
0 10 9999 9999 5
9999 0 1 9999 2
9999 9999 0 4 9999
7 9999 6 0 9999
9999 3 9 2 0
Enter the source vertex:
0
The shortest paths from vertex 0 are:
0=0
1<-4<-0=8
2<-1<-4<-0=9
3<-4<-0=7
4<-0=5
```

LAB 10

USING TCP/IP sockets, write a client-server program to make client sending the file name & the server to UDP/IP

```
#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];
int n;
class router {
char adj_new [MAX], adj_old [
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 carual () {
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;
cout for (int i=0; i<n; i++)
if (i!=j) cout << (char) ('A'+i) << " ";
cout << "\n Enter 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 << "\n Destination Router:";
for (int i=0; i<n; i++)
cout << (char) ('A'+i) << " ";
cout << "\n Outgoing Line:";
for (int i=0; i<n; i++) cout << adj_new [i] << " ";
cout << "\n HOP 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];
}
};
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
  }
}
```