Network Programming Lab

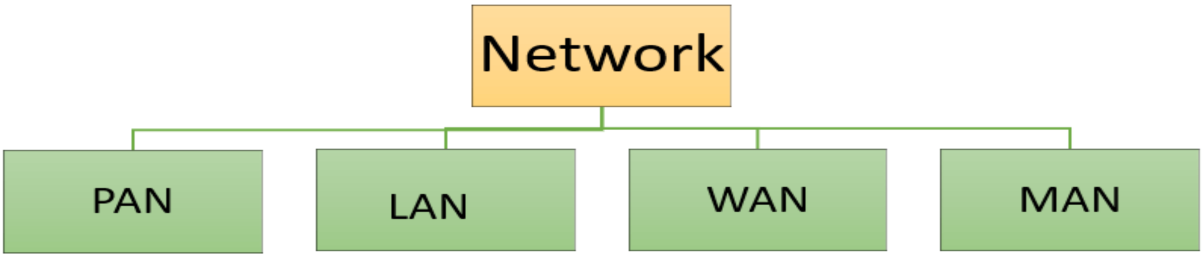
Touching new heights of excellence!

II Year-IV Semester:
Computer Science and Engineering

1. Study of Different Type of LAN & Network Equipments

Ans.

There are various types of computer networks available. We can categorize them according to their size as well as their purpose.



What is PAN (Personal Area Network):- PAN is a computer network formed around a person. It generally consists of a computer, mobile, or personal digital assistant. PAN can be used for establishing communication among these personal devices for connecting to a digital network and the internet.

Characteristics of PAN:-

- It is mostly personal devices network equipped within a limited area.
- Allows you to handle the interconnection of IT devices at the surrounding of a single user.
- PAN includes mobile devices, tablet, and laptop.
- It can be wirelessly connected to the internet called WPAN.
- Appliances use for PAN: cordless mice, keyboards, and Bluetooth systems.

Advantages of PAN:-

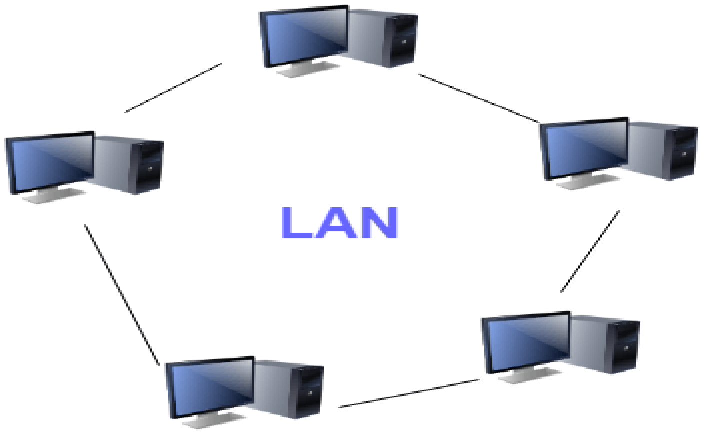
- PAN networks are relatively secure and safe
- It offers only short-range solution up to ten meters
- Strictly restricted to a small area

Disadvantages of PAN

Here are important cons/ drawback of using PAN network:

- It may establish a bad connection to other networks at the same radio bands.
- Distance limits.
-

What is LAN?



A **L**ocal **A**rea **N**etwork (LAN) is a group of computer and peripheral devices which are connected in a limited area such as school, laboratory, home, and office building. It is a widely useful network for sharing resources like files, printers, games, and other application. In general, LAN will be used as one type of transmission medium.

It is a network which consists of less than 5000 interconnected devices across several buildings.

Characteristics of LAN:-

- It is a private network, so an outside regulatory body never controls it.
- LAN operates at a relatively higher speed compared to other WAN systems.
- There are various kinds of media access control methods like token ring and ethernet.

Advantages of LAN

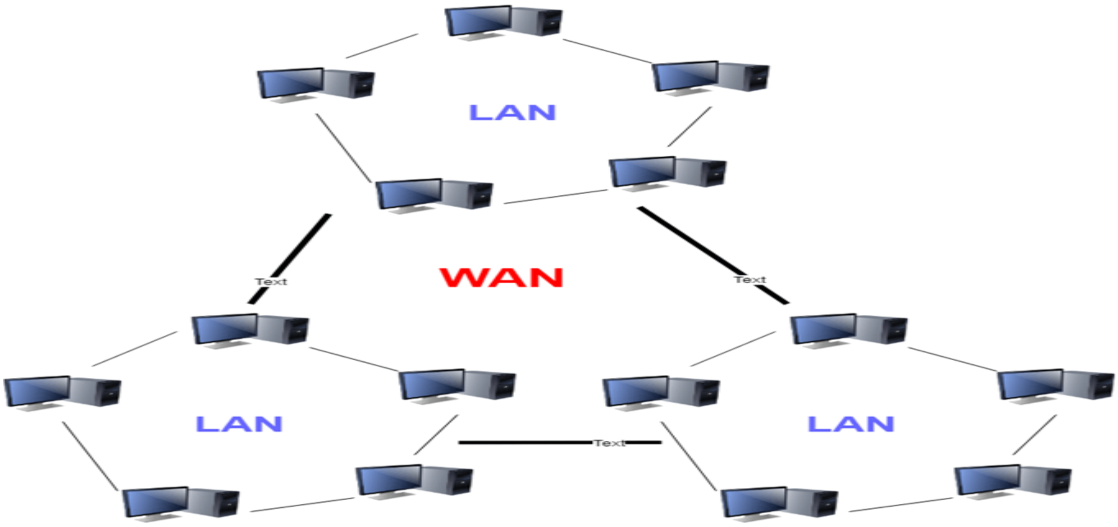
- Computer resources like hard-disks, DVD-ROM, and printers can share local area networks.
- Data of all network users can be stored on a single hard disk of the server computer.
- You can easily transfer data and messages over networked computers.
- It will be easy to manage data at only one place, which makes data more secure.

Disadvantages of LAN

- LAN will indeed save cost because of shared computer resources, but the initial cost of installing Local Area Networks is quite high.
- The LAN admin can check personal data files of every LAN user, so it does not offer good privacy.
- Unauthorized users can access critical data of an organization in case LAN admin is not able to secure centralized data repository.
- Local Area Network requires a constant LAN administration as there are issues related to software setup and hardware failures

What is WAN?

WAN (Wide Area Network) is another important computer network that which is spread across a large geographical area. WAN network system could be a connection of a LAN which connects with other LAN's using telephone lines and radio waves. It is mostly limited to an enterprise or an organization.



Advantages of WAN

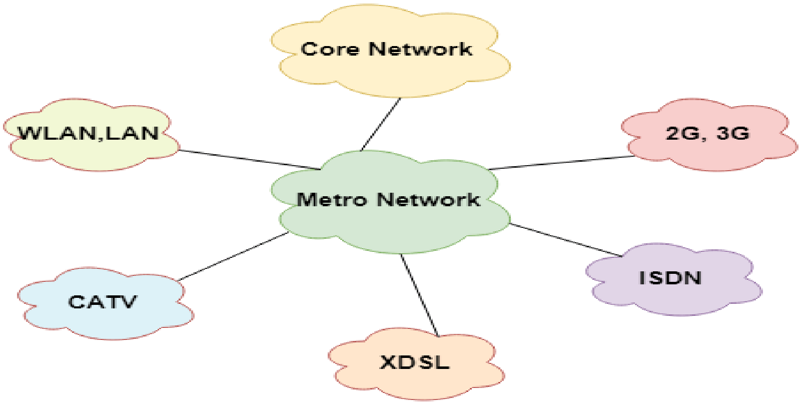
- WAN helps you to cover a larger geographical area. Therefore business offices situated at longer distances can easily communicate.
- Contains devices like mobile phones,laptop,tablet,computers, gaming consoles, etc.
- WLAN connections work using radio transmitters and receivers built into client devices.

Disadvantage of WAN

- The initial setup cost of investment is very high.
- It is difficult to maintain the WAN network. You need skilled technicians and network administrators.
- There are more errors and issues because of the wide coverage and the use of different technologies.
- Offers lower security compared to other types of networks.

What is MAN?

A Metropolitan Area Network or MAN is consisting of a computer network across an entire city, college campus, or a small region. This type of network is large than a LAN, which is mostly limited to a single building or site. this type of network allows you to cover an area from several miles to tens of miles.



Characteristics of MAN

It mostly covers towns and cities in a maximum 50 km range

- Mostly used medium is optical fibers, cables
- Data rates adequate for distributed computing applications.

Advantages of MAN

- It offers fast communication using high-speed carriers, like fiber optic cables.
- It provides excellent support for an extensive size network and greater access to WANs.
- The dual bus in MAN network provides support to transmit data in both directions concurrently.
- A MAN network mostly includes some areas of a city or an entire city.

Disadvantages of MAN

- You need more cable to establish MAN connection from one place to another.
- In MAN network it is tough to make the system secure from hackers

Other Types of Networks

Apart from above mentioned here, are some other important types of networks:

- WLAN (Wireless Local Area Network)
- Storage Area Network
- System Area Network
- Home Area Network
- POLAN- Passive Optical LAN
- Enterprise private network
- Campus Area Network
- Virtual Area Network

Let's see all of them in detail:

1) WLAN

WLAN (Wireless Local Area Network) helps you to link single or multiple devices using wireless communication within a limited area like home, school, or office building. It gives users an ability to move around within a local coverage area which may be connected to the network. Today most modern day's WLAN systems are based on IEEE 802.11 standards.

2) Storage-Area Network (SAN)

A Storage Area Network is a type of network which allows consolidated, block-level data storage. It is mainly used to make storage devices, like disk arrays, optical jukeboxes, and tape libraries.

3) System-Area Network

System Area Network is used for a local network. It offers high-speed connection in server-to-server and processor-to-processor applications. The computers connected on a SAN network operate as a single system at quite high speed.

4) Passive Optical Local Area Network

POLAN is a networking technology which helps you to integrate into structured cabling. It allows you to resolve the issues of supporting Ethernet protocols and network apps.

POLAN allows you to use optical splitter which helps you to separate an optical signal from a single-mode optical fiber. It converts this single signal into multiple signals.

5) Home Area Network (HAN):

A Home Area Network is always built using two or more interconnected computers to form a local area network (LAN) within the home. For example, in the United States, about 15 million homes have more than one computer.

This type of network helps computer owners to interconnect with multiple computers. This network allows sharing files, programs, printers, and other peripherals.

6) Enterprise Private Network :

Enterprise private network (EPN) networks are build and owned by businesses that want to securely connect numerous locations in order to share various computer resources.

7) Campus Area Network (CAN):

A Campus Area Network is made up of an interconnection of LANs within a specific geographical area. For example, a university campus can be linked with a variety of campus buildings to connect all the academic departments.

8) Virtual Private Network:
A VPN is a private network which uses a public network to connect remote sites or users together. The VPN network uses "virtual" connections routed through the internet from the enterprise's private network or a third-party VPN service to the remote site.
It is a free or paid service that keeps your web browsing secure and private over public WiFi hotspots.

Network Devices :-
1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. Also, they do not have intelligence to find out best path for data packets which leads to inefficiencies and wastage.

Types of Hub

- **Active Hub:-** These are the hubs which have their own power supply and can clean, boost and relay the signal along with the network. It serves both as a repeater as well as wiring centre. These are used to extend the maximum distance between nodes.

- **Passive Hub :-** These are the hubs which collect wiring from nodes and power supply from active hub. These hubs relay signals onto the network without cleaning and boosting them and can’t be used to extend the distance between nodes.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

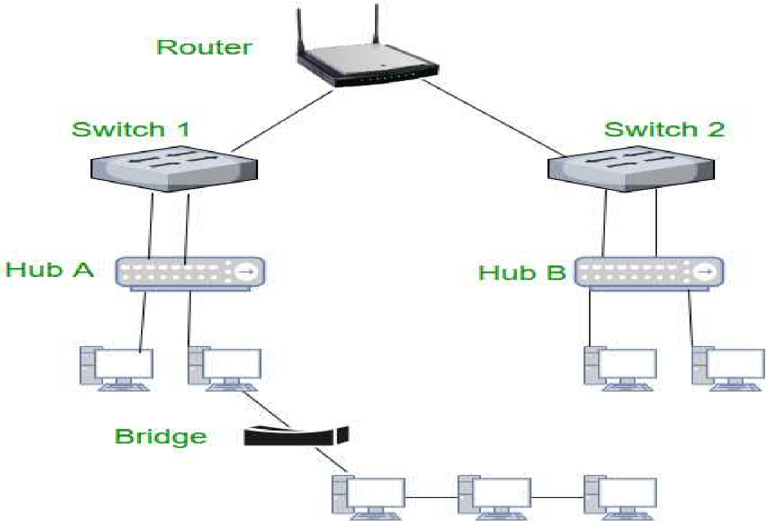
Types of Bridges

- **Transparent Bridges:-** These are the bridge in which the stations are completely unaware of the bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

- **Source Routing Bridges:-** In these bridges, routing operation is performed by source station and the frame specifies which route to follow. The hot can discover frame by sending a special frame called discovery frame, which spreads through the entire network using all possible paths to destination.

4. Switch – A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only.

5. <u>Routers</u> – A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.



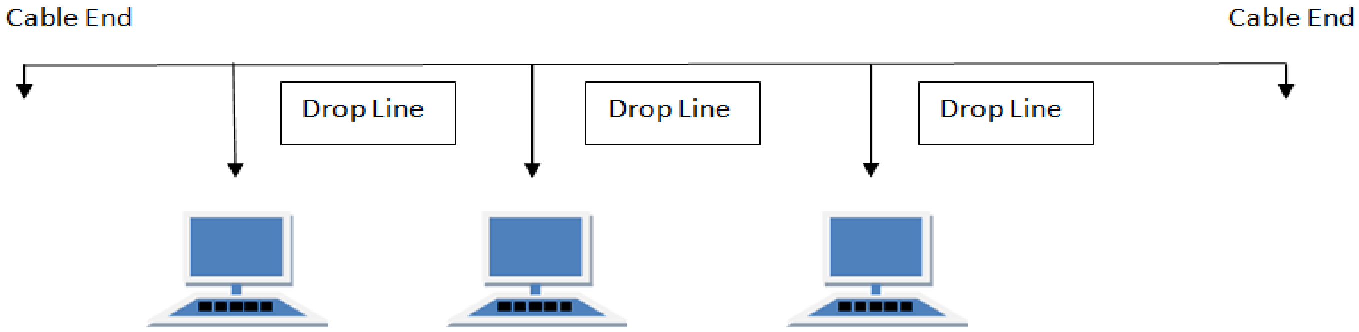
6. Gateway – A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

Exp.-2 Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc

Ans. - Types of Network Topology

Network Topology is the schematic description of a network arrangement, connecting various nodes(sender and receiver) through lines of connection.

1. **BUS Topology**



Bus topology is a network type in which every computer and network device is connected to single cable. When it has exactly two endpoints, then it is called **Linear Bus topology**.

Features of Bus Topology

1. It transmits data only in one direction.
2. Every device is connected to a single cable

Advantages of Bus Topology

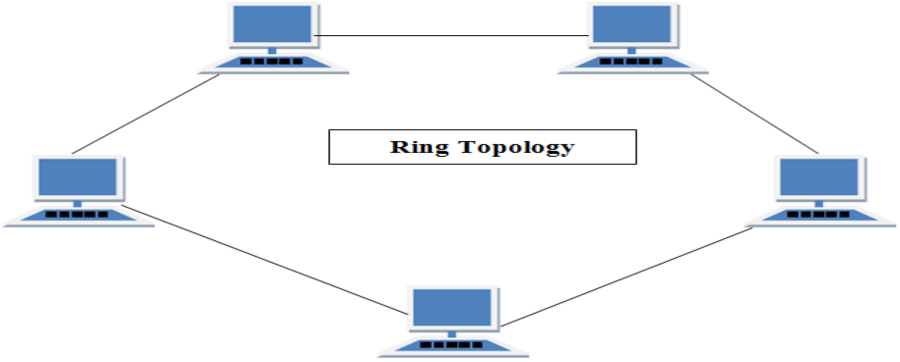
1. It is cost effective.
2. Cable required is least compared to other network topology.
3. Used in small networks.
4. It is easy to understand.
5. Easy to expand joining two cables together.

Disadvantages of Bus Topology

1. Cables fails then whole network fails.
2. If network traffic is heavy or nodes are more the performance of the network decreases.
3. Cable has a limited length.

4. It is slower than the ring topology.

2. **RING Topology:-** It is called ring topology because it forms a ring as each computer is connected to another computer, with the last one connected to the first.



Features of Ring Topology

1. The transmission is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called **Dual Ring Topology**.
2. In Dual Ring Topology, two ring networks are formed, and data flow is in opposite direction in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.
3. Data is transferred in a sequential manner that is bit by bit. Data transmitted, has to pass through each node of the network, till the destination node.

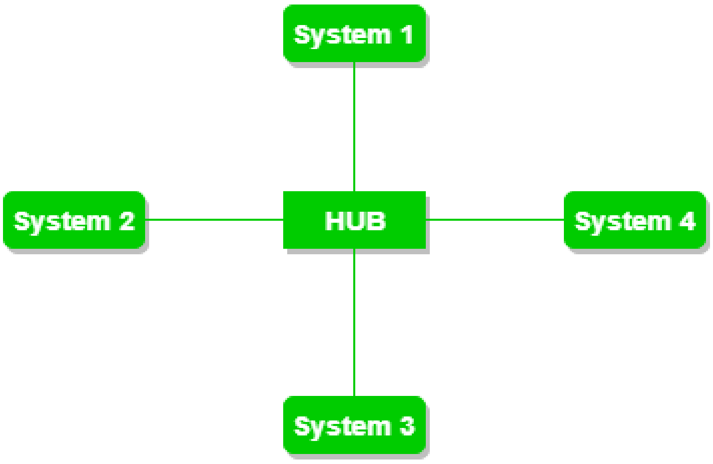
Advantages of Ring Topology

1. Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data.
2. Cheap to install and expand

Disadvantages of Ring Topology

1. Troubleshooting is difficult in ring topology.
2. Adding or deleting the computers disturbs the network activity.
3. Failure of one computer disturbs the whole network.

3. **Star Topology:-** In star topology, all the devices are connected to a single hub through a cable. This hub is the central node and all others nodes are connected to the central node.



Features of Star Topology

1. Every node has its own dedicated connection to the hub.
2. Hub acts as a repeater for data flow.
3. Can be used with twisted pair, Optical Fibre or coaxial cable.

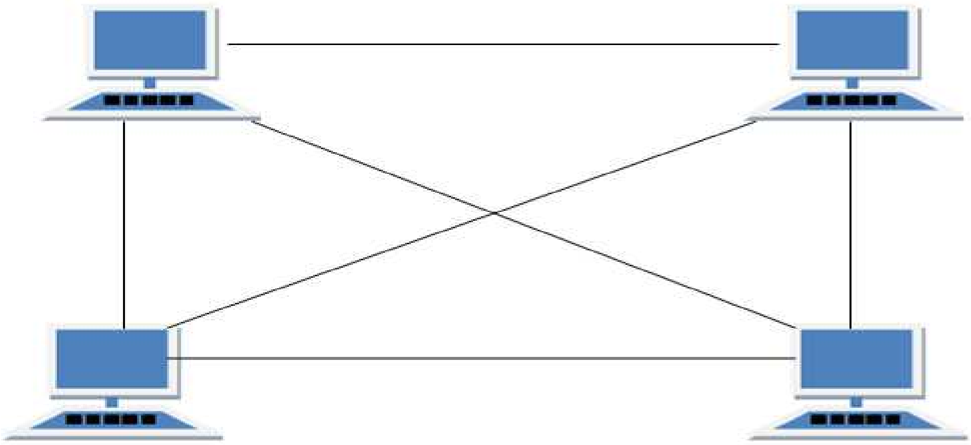
Advantages of Star Topology

1. Fast performance with few nodes and low network traffic.
2. Hub can be upgraded easily.
3. Easy to troubleshoot.
4. Easy to setup and modify.
5. Only that node is affected which has failed, rest of the nodes can work smoothly.

Disadvantages of Star Topology

1. Cost of installation is high.
2. Expensive to use.
3. If the hub fails then the whole network is stopped because all the nodes depend on the hub.
4. Performance is based on the hub that is it depends on its capacity

4. **MESH Topology:-**It is a point-to-point connection to other nodes or devices. All the network nodes are connected to each other. Mesh has n(n-1)/2 physical channels to link n devices.



Features of Mesh Topology

1. Fully connected.
2. Robust.
3. Not flexible.

Advantages of Mesh Topology

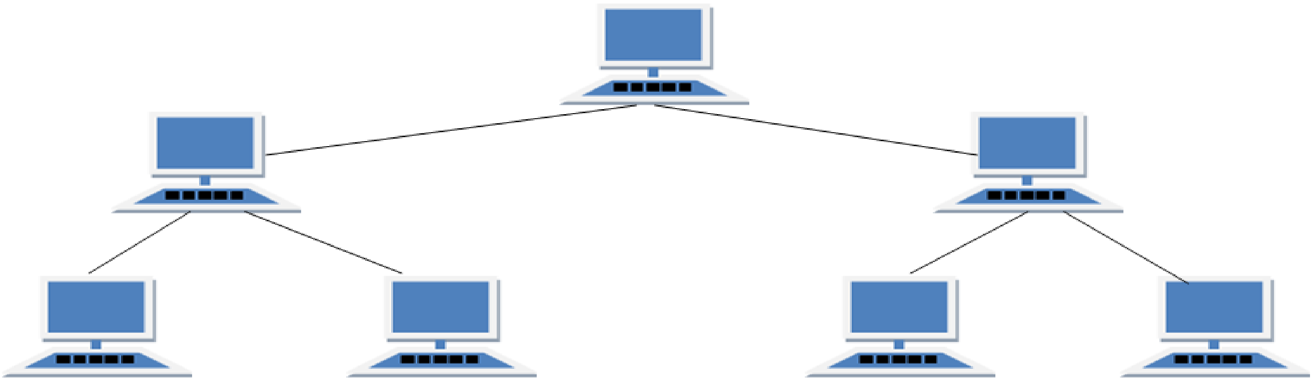
1. Each connection can carry its own data load.
2. It is robust.
3. Fault is diagnosed easily.
4. Provides security and privacy.

Disadvantages of Mesh Topology

1. Installation and configuration is difficult.
2. Cabling cost is more.
3. Bulk wiring is required.

TREE Topology

It has a root node and all other nodes are connected to it forming a hierarchy. It is also called hierarchical topology. It should at least have three levels to the



hierarchy.

Features of Tree Topology

1. Ideal if workstations are located in groups.
2. Used in Wide Area Network.

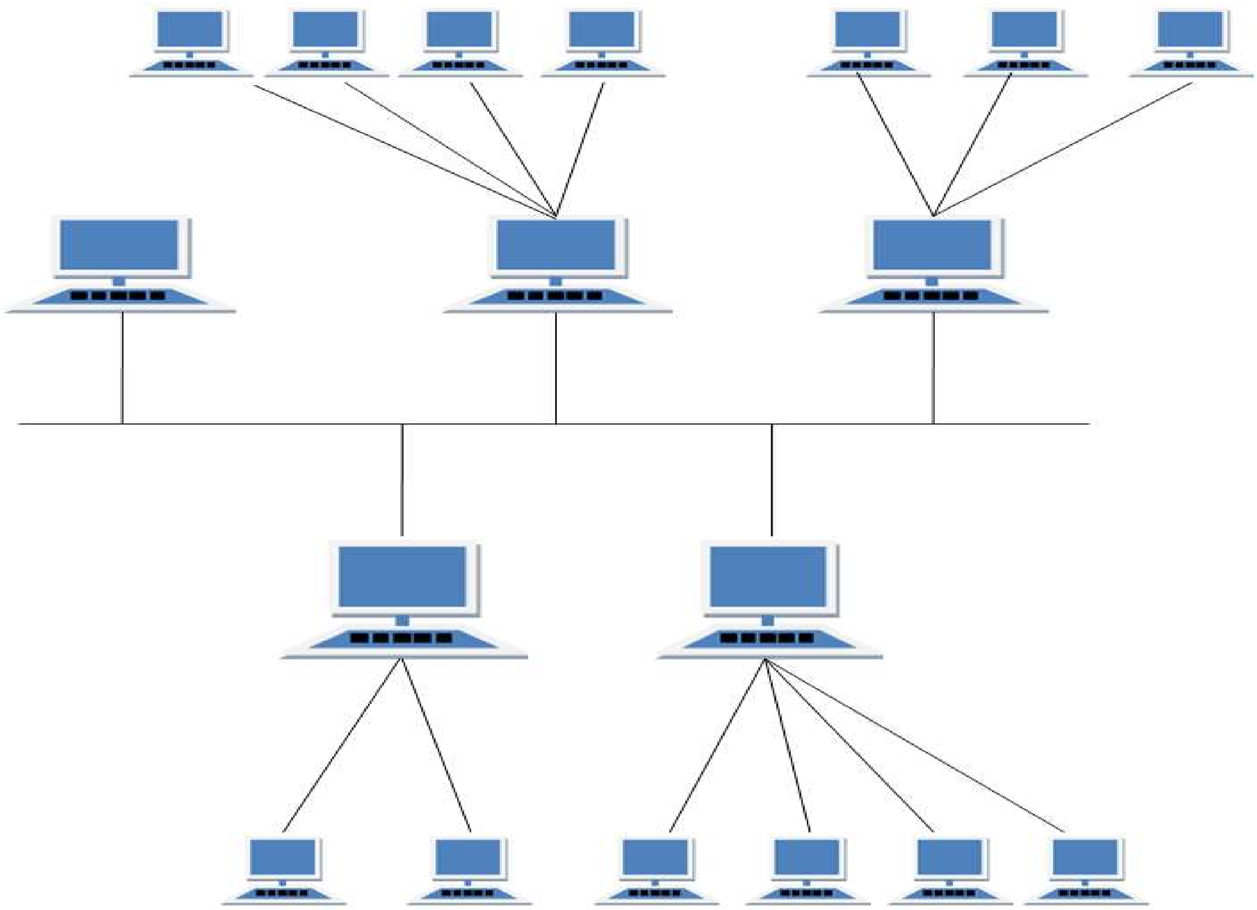
Advantages of Tree Topology

1. Extension of bus and star topologies.
2. Expansion of nodes is possible and easy.
3. Easily managed and maintained.
4. Error detection is easily done.

Disadvantages of Tree Topology

1. Heavily cabled.
2. Costly.
3. If more nodes are added maintenance is difficult.
4. Central hub fails, network fails.

HYBRID Topology:- It is two different types of topologies which is a mixture of two or more topologies. For example if in an office in one department ring topology is used and in another star topology is used, connecting these topologies will result in Hybrid Topology (ring topology and star topology).



Features of Hybrid Topology

1. It is a combination of two or topologies
2. Inherits the advantages and disadvantages of the topologies included

Advantages of Hybrid Topology

1. Reliable as Error detecting and trouble shooting is easy.
2. Effective.
3. Scalable as size can be increased easily.
4. Flexible.

Disadvantages of Hybrid Topology

1. Complex in design.
2. Costly.

4. Write a program to implement various types of error correcting techniques

```
#include<stdio.h>
#include<conio.h>
void main() {
    int data[7],rec[7],i,c1,c2,c3,c;
    printf("this works for message of 4bits in size \nenter message bit one by one:  ");
    scanf("%d%d%d%d",&data[0],&data[1],&data[2],&data[4]);
    data[6]=data[0]^data[2]^data[4];
    data[5]=data[0]^data[1]^data[4];
    data[3]=data[0]^data[1]^data[2];
    printf("\nthe encoded bits are given below: \n");
    for (i=0;i<7;i++) {
        printf("%d ",data[i]);
    }
    printf("\nenter the received data bits one by one: ");
    for (i=0;i<7;i++) {
        scanf("%d",&rec[i]);
    }
    c1=rec[6]^rec[4]^rec[2]^rec[0];
    c2=rec[5]^rec[4]^rec[1]^rec[0];
    c3=rec[3]^rec[2]^rec[1]^rec[0];
    c=c3*4+c2*2+c1 ;
    if(c==0) {
        printf("\ncongratulations there is no error: ");
    } else {
        printf("\nerron on the postion: %d\nthe correct message is \n",c);
        if(rec[7-c]==0)
                    rec[7-c]=1; else
                    rec[7-c]=0;
        for (i=0;i<7;i++) {
            printf("%d ",rec[i]);
        }
    }
    getch();
}
```

5. Write a program to implement various types of framing methods

Ans. Type-1

Aim-Implement the data link layer framing methods such as Bit Stuffing.

<u>Theory</u>

Security and Error detection are the most prominent features that are to be provided by any application which transfers data from one end to the other end. One of such a mechanism in tracking errors which may add up to the original data during transfer is known as Stuffing. It is of two types namely Bit Stuffing and the other Character Stuffing. Coming to the Bit Stuffing, 01111110 is appended within the original data while transfer of it. The following program describes how it is stuffed at the sender end and de-stuffed at the 16eceiver end.

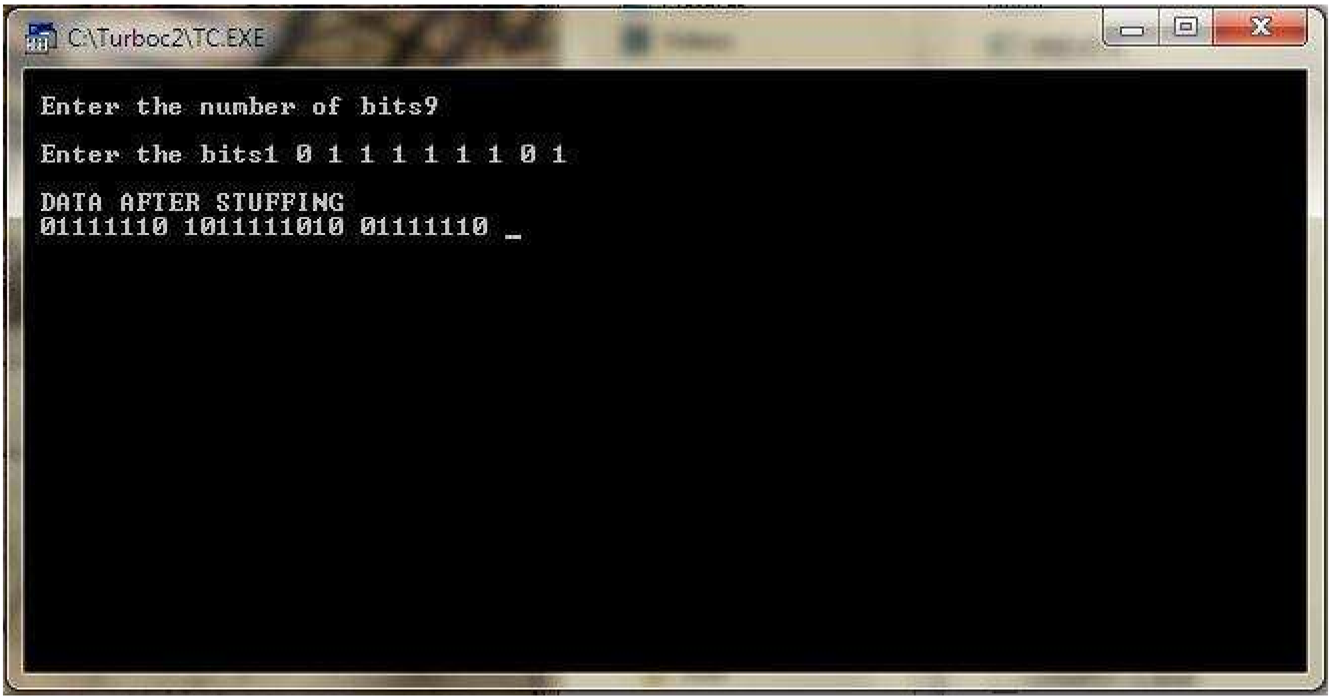
Program:

```
#include
main()
{
int a[15];
int i,j,k,n,c=0,pos=0;
clrscr();
printf("\n Enter the number of bits");
scanf("%d",&n);
printf("\n Enter the bits");
for(i=0;i<n;i++)
 scanf("%d",&a[i]);
for(i=0;i<n;i++)
 {
if(a[i]==1)
{
c++;
if(c==5)
{
pos=i+1;
c=0;
for(j=n;j>=pos;j--)
{
k=j+1;
a[k]=a[j];
}
a[pos]=0;
n=n+1;
}
}
else
c=0;
}

printf("\n DATA AFTER STUFFING \n");
printf(" 01111110 ");

for(i=0;i<n;i++)
 {
printf("%d",a[i]);
}
printf(" 01111110 ");
getch();
}
```

Output:



```
Enter the number of bits9

Enter the bits1 0 1 1 1 1 1 1 0 1

DATA AFTER STUFFING
01111110 1011111010 01111110 _
```

</n;i++)
</n;i++)
</n;i++)

Type-2

implement the data link layer framing methods such as Character Stuffing and also De-stuff it

Theory

Coming to the Character Stuffing, DLESTX and DLEETX are used to denote start and end of character data with some constraints imposed on repetition of charaters as shown in the program below clearly.

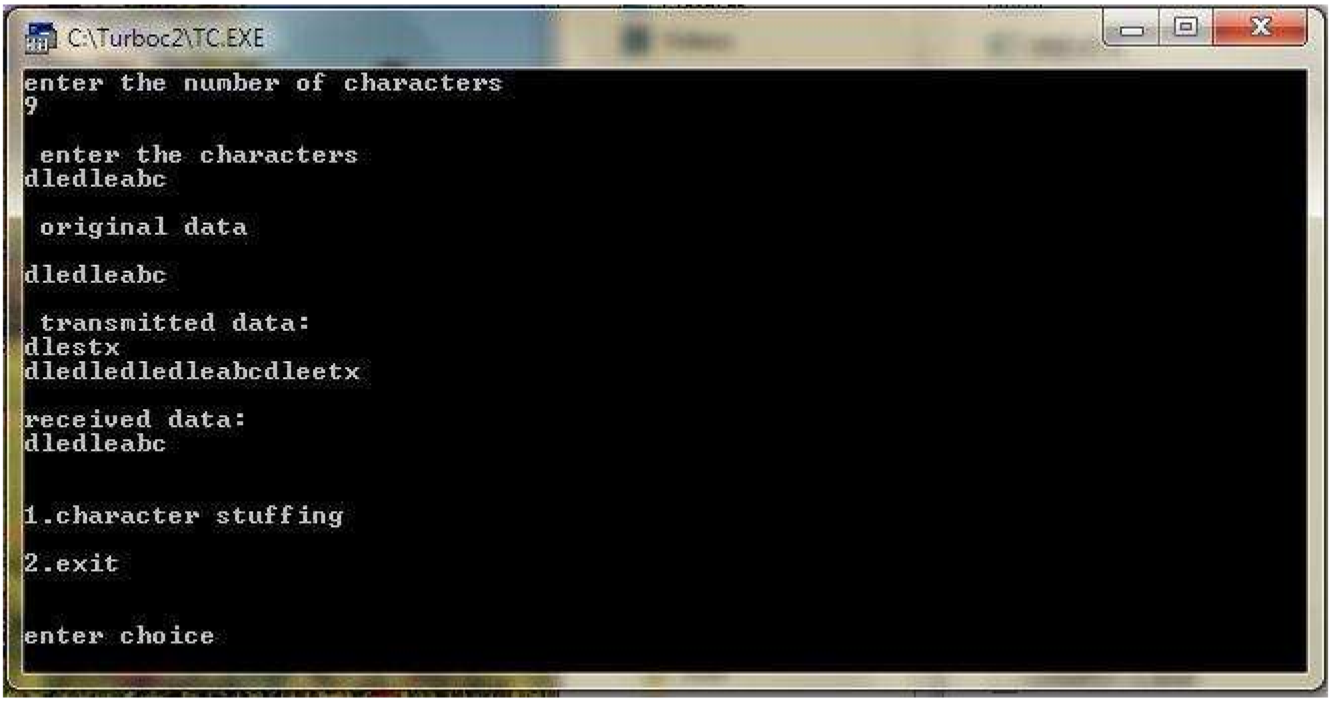
Program:

```
#include
#include
#include
void charc(void);
void main()
{
int choice;
while(1)
{
printf("\n\n\n1.character stuffing");
printf("\n\n2.exit");
printf("\n\n\nenter choice");
scanf("%d",&choice);
printf("%d",choice);
if(choice>2)
printf("\n\n invalid option....please renter");
switch(choice)
{
case 1:
charc();
break;
case 2:
exit(0);
}
}
}
void charc(void)
{
char c[50],d[50],t[50];
int i,m,j;
```

```
clrscr();
printf("enter the number of characters\n");
scanf("%d",&m);
printf("\n enter the characters\n");
for(i=0;i<m+1;i++)
 {
scanf("%c",&c[i]);
}
printf("\n original data\n");
for(i=0;i<m+1;i++)
 printf("%c",c[i]);
d[0]='d';
d[1]='l';
d[2]='e';
d[3]='s';
d[4]='t';
d[5]='x';
for(i=0,j=6;i<m+1;i++,j++)
 {
if((c[i]=='d'&&c[i+1]=='l'&& c[i+2]=='e'))
{
d[j]='d';
j++;
d[j]='l';
j++;
d[j]='e';
j++;
m=m+3;
}
d[j]=c[i];
}
m=m+6;
m++;
d[m]='d';
m++;
d[m]='l';
m++;
```

```
d[m]='e';
m++;
d[m]='e';
m++;
d[m]='t';
m++;
d[m]='x';
m++;
printf("\n\n transmitted data: \n");
for(i=0;i<m;i++)
 {
printf("%c",d[i]);
}
for(i=6,j=0;i<m-6;i++,j++)
 {
if(d[i]=='d'&&d[i+1]=='l'&&d[i+2]=='e'&&d[i+3]=='d'&&d[i+4]=='l'&&d[i+5]=='e')
i=i+3;
t[j]=d[i];
}
printf("\n\nreceived data:");
for(i=0;i<j;i++)
 {printf("%c",t[i]);
}
}
```

Output:



```
enter the number of characters
9

 enter the characters
dledleabc

 original data

dledleabc

 transmitted data:
dlestx
dledledledleabcdleetx

received data:
dledleabc


1.character stuffing

2.exit


enter choice
```

```
</j;i++)
</m-6;i++,j++)
</m;i++)
</m+1;i++,j++)
</m+1;i++)
</m+1;i++)
```

6. **Write two programs in C: hello_client and hello_server**

a. The server listens for, and accepts, a single TCP connection; it reads all the data it can from that connection, and prints it to the screen; then it closes the connection

```
// TCP Server

#include <unistd.h>

#include <stdio.h>

#include <sys/socket.h>

#include <stdlib.h>

#include <netinet/in.h>

#include <string.h>
```

```
#define PORT 8080

int main(int argc, char const *argv[])

{

    int server_fd, new_socket, valread;

    struct sockaddr_in address;

    int opt = 1;

    int addrlen = sizeof(address);

    char buffer[1024] = {0};

    char *hello = "Hello from server";

    // Creating socket file descriptor

    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0)

    {  perror("socket failed");

       exit(EXIT_FAILURE);

    }


    // Forcefully attaching socket to the port 8080

    if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR |
SO_REUSEPORT,&opt, sizeof(opt)))

    {   perror("setsockopt");

        exit(EXIT_FAILURE);

    }

    address.sin_family = AF_INET;

    address.sin_addr.s_addr = INADDR_ANY;

    address.sin_port = htons( PORT );
    // Forcefully attaching socket to the port 8080
    if (bind(server_fd, (struct sockaddr *)&address, sizeof(address))<0)
    {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }
    if (listen(server_fd, 3) < 0)
    {
        perror("listen");
        exit(EXIT_FAILURE);
    }
    if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
(socklen_t*)&addrlen))<0)
    { perror("accept");
        exit(EXIT_FAILURE);
    }
    valread = read( new_socket , buffer, 1024);
    printf("%s\n",buffer );
    send(new_socket , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");
    return 0;
}
```

// Tcp Client

```
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
#define PORT 8080

int main(int argc, char const *argv[])
{
    struct sockaddr_in address;
    int sock = 0, valread;
    struct sockaddr_in serv_addr;
    char *hello = "Hello from client";
    char buffer[1024] = {0};
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        printf("\n Socket creation error \n");
        return -1;
    }

    memset(&serv_addr, '0', sizeof(serv_addr));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);


    // Convert IPv4 and IPv6 addresses from text to binary form
    if(inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr)<=0)
    {
        printf("\nInvalid address/ Address not supported \n");
        return -1;
    }


    if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
    {
        printf("\nConnection Failed \n");
        return -1;
    }
    send(sock , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");
    valread = read( sock , buffer, 1024);
    printf("%s\n",buffer );
    return 0;
}
```

Steps to compile and Run

<u>Open one Terminal in Linux and compile and run TCP server program</u>

1. Compile the server program

```
gcc tcpserver.c -o tcpser)
```

1. Run server using

```
./tcpser
```

Output

```
[root@localhost TCP]# gcc tcpserver.c -o tcpser

[root@localhost TCP]# ./tcpser

Client : Hello from client

Hello message sent.
```

Open one Terminal in Linux and compile and run TCP client program

1. compile tcp client program

```
gcc tcpclient.c -o tcpcli)
```

2. Run tcp client

```
./tcpcli)
```

Output

```
[root@localhost TCP# gcc tcpclient.c -o tcpcli

[root@localhost TCP]# ./tcpcli

Hello message sent.
```

7. Write an Echo_Client and Echo_server using TCP to estimate the round trip time from client to the server. The server should be such that it can accept multiple connections at any given time.

Round trip time(RTT) is the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgement of that signal to be received. This time therefore consists of the propagation times between the two point of signal.

On the Internet, an end user can determine the RTT to and from an IP(Internet Protocol) address by pinging that address. The result depends on various factors :-

- The data rate transfer of the source’s internet connection.
- The nature of transmission medium.
- The physical distance between source and destination.
- The number of nodes between source and destination.
- The amount of traffic on the LAN(Local Area Network) to which end user is connected.
- The number of other requests being handled by intermediate nodes and the remote server.
- The speed with which intermediate node and the remote server function.

At sender Side:

```
#include <string.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>

#define MYPORT 3490
#define SIZE_TO_SEND 1000
#define MY_IP "127.0.0.1"

int main(int argc, char *argv[]) {
 int sockfd,sockfd2;
 char tosend = 's'; //a char (1byte) to send to receivers
 char ack;
 struct sockaddr_in my_addr,rcvr_addr;
 struct timeval start,end;
 int sin_size = sizeof(my_addr),i,k,num_packet_sent,optval;
 double t1,t2;

 //open TCP socket,bind and accept RECEIVERS connections
 if( (sockfd = socket(PF_INET, SOCK_STREAM, 0)) ==-1){
  perror("socket error");
  exit(1);
 }
 my_addr.sin_family = AF_INET;
 my_addr.sin_port = htons(MYPORT);
 my_addr.sin_addr.s_addr = inet_addr(MY_IP);
 memset(my_addr.sin_zero, '\0', sizeof(my_addr.sin_zero));
 //allow reuse of port
 optval = 1;
 if (setsockopt(sockfd,SOL_SOCKET,SO_REUSEADDR,&optval,sizeof(int)) == -1) {
  perror("setsockopt");
  exit(1);
 }
 //bind(socketfd, struct about my address,sizeofmy address);

if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof my_addr) == -1) {
  perror("bind");
  exit(1);
 }
 listen(sockfd,10);

 sockfd2 = accept(sockfd, (struct sockaddr *)&rcvr_addr, &sin_size);

 //connections OK
 //send 100 packet of size 1 byte and for each send wait for ack

 t1=0.0; t2=0.0;
 printf("Sending 100 messages 1 byte each and wait for ack.\n");
 for(num_packet_sent=0;num_packet_sent<100;num_packet_sent++){
  if(gettimeofday(&start,NULL)) {
   printf("time failed\n");
   exit(1);
  }
  send(sockfd2,&tosend,sizeof(char),0);
  optval=recv(sockfd2,&ack,sizeof(char),0);
  if(optval==-1) {
   perror("Receive error");
   exit(1);
  }
  else{
   if(gettimeofday(&end,NULL)) {
    printf("time failed\n");
    exit(1);
   }
            t1+=start.tv_sec+(start.tv_usec/1000000.0);
            t2+=end.tv_sec+(end.tv_usec/1000000.0);
  }
 }
 //calculate and print mean rtt

 printf("RTT = %g ms\n",(t2-t1)/100);
 printf("close sockets and exit\n");
 shutdown(sockfd2,2);
 shutdown(sockfd,2);
 exit(0);
}
```

At Receiver Side

```
#include <string.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
```

```
#define SENDER_PORT 3490
#define SENDER_IP "127.0.0.1"

int main(int argc, char *argv[]) {

 int sockfd;
 int rcv_num,loop_count,i;
 char buf;
 struct sockaddr_in sender_addr;

 //open socket and connect
 if( (sockfd = socket(PF_INET, SOCK_STREAM, 0)) ==-1){
  perror("socket error"); // do some error checking!
  exit(1);
 }

 sender_addr.sin_family = AF_INET;
 sender_addr.sin_port = htons(SENDER_PORT);
 sender_addr.sin_addr.s_addr = inet_addr(SENDER_IP);
 memset(sender_addr.sin_zero, '\0', sizeof(sender_addr.sin_zero));

 if ((connect(sockfd,(struct sockaddr *)&sender_addr,sizeof(sender_addr))) ==-1){
  perror("connect error"); // do some error checking!
  exit(1);
 }

 //connection established
 printf("Connection to sender established\n");
 //reads 100 packets of 1 byte and sends them back as ack packets
 printf("Receive 100 packets of 1 byte and send then back\n");
 for(i=0;i<100;i++){
  rcv_num = recv(sockfd,&buf,sizeof(char),0);
  if(rcv_num!=0) {
   //send  ack
   send(sockfd,&buf,sizeof(char),0);
  }
  else{
   perror("Receive error");
   exit(1);
  }
 }
 printf("\tDone\nClose socket and exit\n");
 close(sockfd);
 exit(0);
}
```

Compilation & Run –

Server side:

```
gcc serverrtt.c -o serverrtt
./server
```

Client side:
gcc clientrtt.c -o clientrtt
./client

Output

At Receiver Terminal

```
Connection to sender established
Receive 100 packets of 1 byte and send then back
    Done
Close socket and exit
```

At Server

```
Sending 100 messages 1 byte each and wait for ack.
RTT = 2.65503e-05 ms
close sockets and exit
```

Exp.-9. Simulate Bellman-Ford Routing algorithm in NS2.

```
/*
Distance Vector Routing in this program is implemented using
Bellman Ford Algorithm:-
*/
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);//Enter the nodes
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];//initialise the
distance equal to cost matrix
            rt[i].from[j]=j;
        }
    }
        do
        {
            count=0;
            for(i=0;i<nodes;i++)//We choose arbitary vertex k
and we calculate the direct distance from the node i to k
using the cost matrix
            //and add the distance from k to node j
            for(j=0;j<nodes;j++)
            for(k=0;k<nodes;k++)
                if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
                {//We calculate the minimum distance
                    rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                    rt[i].from[j]=k;
                    count++;
                }
        }while(count!=0);
        for(i=0;i<nodes;i++)
        {
            printf("\n\n For router %d\n",i+1);
            for(j=0;j<nodes;j++)
            {
                printf("\t\nnode %d via %d Distance %d
",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
        }
    printf("\n\n");
    getch();
}


/*
A sample run of the program works as:-
Enter the number of nodes :
3
Enter the cost matrix :
0 2 7
2 0 1
7 1 0
For router 1
node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 3 Distance 3
For router 2
node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1
For router 3
node 1 via 1 Distance 3
node 2 via 2 Distance 1
node 3 via 3 Distance 0
*/
```