**ST.ANN'S COLLEGE OF ENGINEERING AND TECHNOLOGY**

**An Autonomous Institution**

Approved by AICTE& Permanently Affiliated to JNTUK, Kakinada

Accredited by NAAC with “A” Grade and NBA

Nayunipalli village, ChallaReddy Palem Post,Vetapalem Mandal, Chirala,

www.sacet.ac.in



**Computer Networks-Lab Manual**

**B.TECH(R20)**

**(III YEAR-I SEM)**

**(2022-2023)**

**DEPARTMENT OF COMPUTER SCIENCE AND**

**ENGINEERING**

Course Objectives:

Learn basic concepts of computer networking and acquire practical notions of protocols with the emphasis on TCP/IP. A lab provides a practical approach to Ethernet/Internet networking: networks are assembled, and experiments are made to understand the layered architecture and how do some important protocols work

Course outcomes:

By the end of the course student will be able to

->Know how reliable data communication is achieved through data link layer.

-> Suggest appropriate routing algorithm for the network.

->Provide internet connection to the system and its installation.

->Work on various network management tools

List of Experiments:

1. Study of Network devices in detail and connect the computers in Local Area Network.

2. Write a Program to implement the data link layer farming methods such as

i) Character stuffing

ii) bit stuffing.

3. Write a Program to implement data link layer farming method checksum.

4. Write a program for Hamming Code generation for error detection and correction.

5. Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.

6. Write a Program to implement Sliding window protocol for Goback N.

7. Write a Program to implement Sliding window protocol for Selective repeat.

8. Write a Program to implement Stop and Wait Protocol.

9. Write a program for congestion control using leaky bucket algorithm.

10. Write a Program to implement Dijkstra‘s algorithm to compute the Shortest path through a graph.

11. Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

12. Write a Program to implement Broadcast tree by taking subnet of hosts.

13. Wireshark

i. Packet Capture Using Wire shark

ii. Starting Wire shark

iii. Viewing Captured Traffic

iv. Analysis and Statistics & Filters.

14. How to run Nmap scan.

15. Operating System Detection using Nmap.

16. Do the following using NS2 Simulator

i. NS2 Simulator-Introduction

ii. Simulate to Find the Number of Packets Dropped

iii. Simulate to Find the Number of Packets Dropped by TCP/UDP

iv. Simulate to Find the Number of Packets Dropped due to Congestion

v. Simulate to Compare Data Rate& Throughput.

1. Study of Network devices in detail and connect the computers in Local Area Network.

Aim:-

To write a information on study of Network devices in detail and connect the computers in Local Area Network.

epeaters:

➔Repeaters are network devices operating at the physical layer of the OSI model that

amplify or regenerate an incoming signal before retransmitting it.

➔They are incorporated in networks to expand its coverage area. They are also known

as signal boosters.

➔Signals that carry information within a network can travel a fixed distance before

attenuation endangers the integrity of the data.

➔A repeater receives a signal and, before it becomes too weak or corrupted, regenerates

the original bit pattern.

➔The repeater then sends the refreshed signal.

➔A repeater can extend the physical length of a LAN.

➔The location of a repeater on a link is vital. A repeater must be placed so that a signal

reaches it before any noise changes the meaning of any of its bits.

➔If the corrupted bit travels much farther, however, accumulated noise can change its

meaning completely.

➔At that point, the original voltage is not recoverable, and the error needs to be

corrected.

➔A repeater placed on the line before the legibility of the signal becomes lost can still

read the signal well enough to determine the intended voltages and replicate them in

their original form.

epeaters:

➔Repeaters are network devices operating at the physical layer of the OSI model that

amplify or regenerate an incoming signal before retransmitting it.

➔They are incorporated in networks to expand its coverage area. They are also known

as signal boosters.

➔Signals that carry information within a network can travel a fixed distance before

attenuation endangers the integrity of the data.

➔A repeater receives a signal and, before it becomes too weak or corrupted, regenerates

the original bit pattern.

➔The repeater then sends the refreshed signal.

➔A repeater can extend the physical length of a LAN.

➔The location of a repeater on a link is vital. A repeater must be placed so that a signal

reaches it before any noise changes the meaning of any of its bits.

➔If the corrupted bit travels much farther, however, accumulated noise can change its

meaning completely.

➔At that point, the original voltage is not recoverable, and the error needs to be

corrected.

➔A repeater placed on the line before the legibility of the signal becomes lost can still

read the signal well enough to determine the intended voltages and replicate them in

their original form.

epeaters:

➔Repeaters are network devices operating at the physical layer of the OSI model that

amplify or regenerate an incoming signal before retransmitting it.

➔They are incorporated in networks to expand its coverage area. They are also known

as signal boosters.

➔Signals that carry information within a network can travel a fixed distance before

attenuation endangers the integrity of the data.

➔A repeater receives a signal and, before it becomes too weak or corrupted, regenerates

the original bit pattern.

➔The repeater then sends the refreshed signal.

➔A repeater can extend the physical length of a LAN.

➔The location of a repeater on a link is vital. A repeater must be placed so that a signal

reaches it before any noise changes the meaning of any of its bits.

➔If the corrupted bit travels much farther, however, accumulated noise can change its

meaning completely.

➔At that point, the original voltage is not recoverable, and the error needs to be

corrected.

➔A repeater placed on the line before the legibility of the signal becomes lost can still

read the signal well enough to determine the intended voltages and replicate them in

their original form.

**Network devices:-**

*Network devices or networking hardware are physical devices that are required for communication and interaction between hardware on a computer network.*

**Types of Netwok Devices:-**

**🡪 Repeaters**

**🡪 Hub**

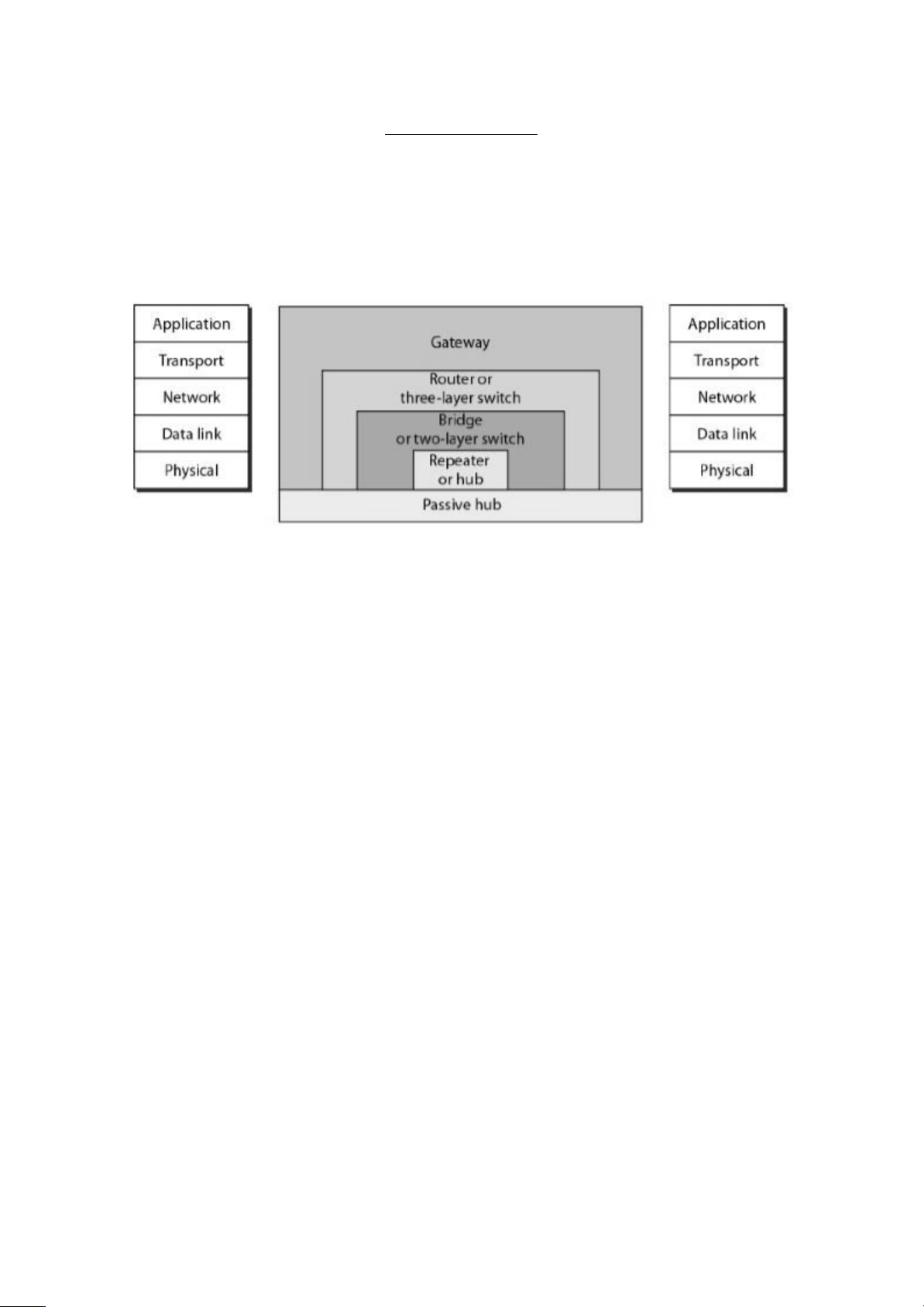
**🡪 Bridges**

**🡪 Routers**

**🡪 Gateway**

**🡪 Switch**

**Structure:-**

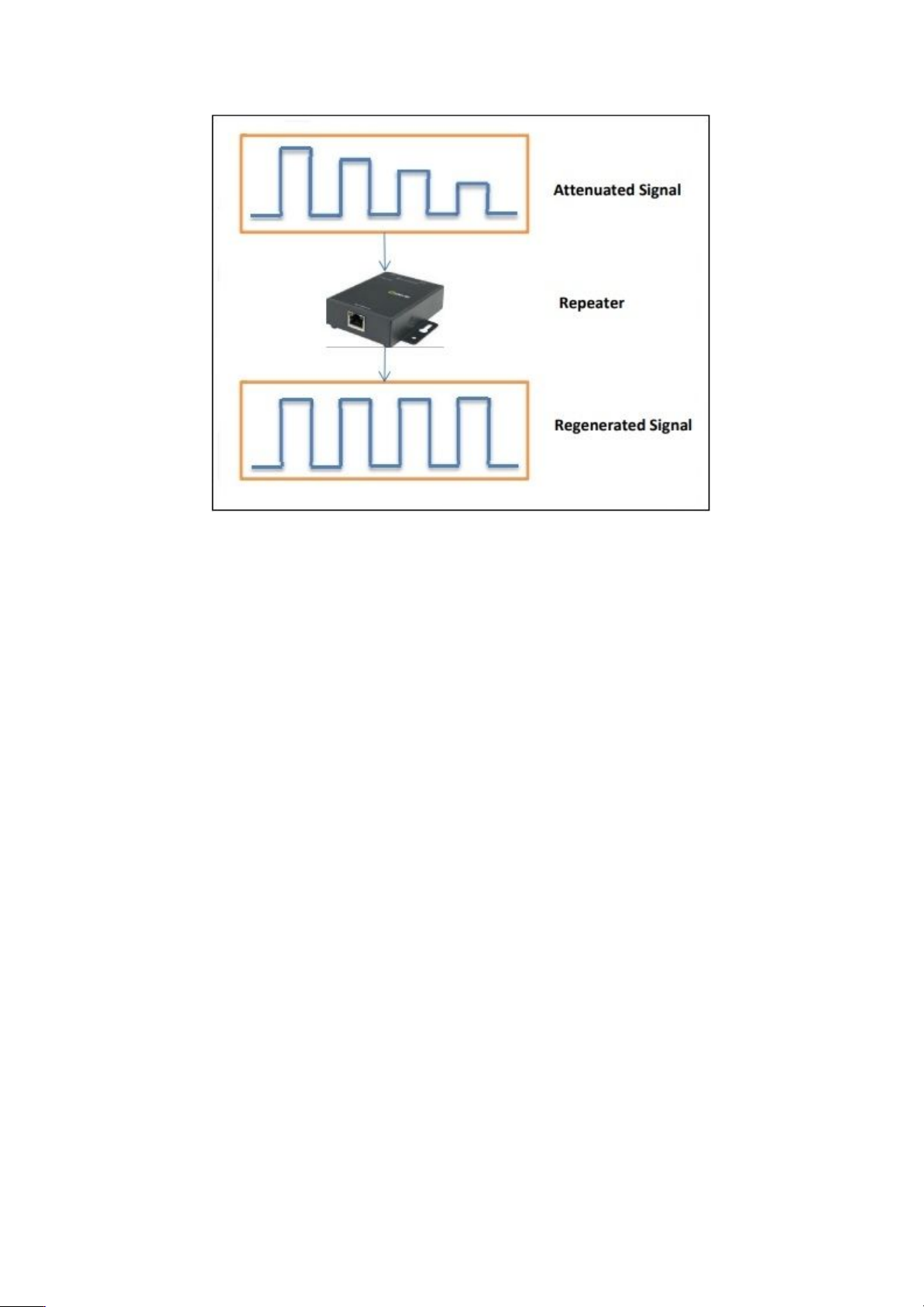


**Repeaters:-**

**🡪Repeaters are network device operating at the physical layer of the OSI model that amplify or regenerate an incoming signal before retransmitting it.**

**🡪They are incorporated in networks to expand its coverage area.They are also known as signal boosters.**

**🡪Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data.**



**Types of Repeaters:**

**🡪According to the types of signals that they regenerate,repeaters can be classified into two categories**

**.Analog Repeaters**

**.Digital repeaters**

**🡪According to the types of networks that they connect,repeaters can be categorized into two types**

**.Wired repeaters**

**.Wireless repeaters**

**🡪According to the domain of LANS they connect,repearets can be divided into two categories**

**.Local repeaters**

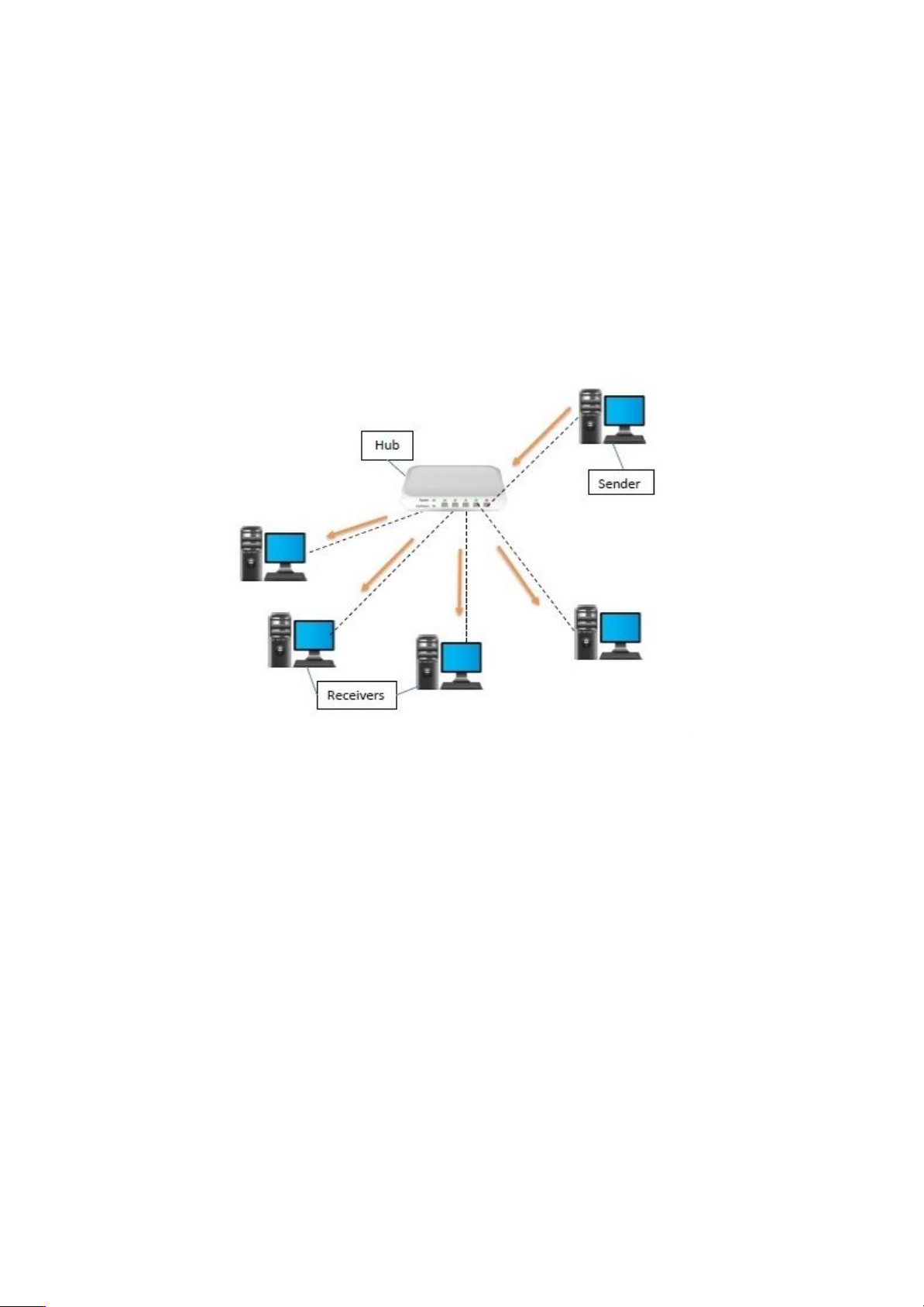
**.Remote repeaters**

**Hub:-**

**🡪A hub is a physical layer networking device which is used to connect multiple device in a network.They are generally used to connect computer in a LAN.**

**🡪A hub has many ports in it.A computer which intends to be connected to the network is plugged into one of these ports.**

**🡪When a data frame arrives at a port,it is broadcast to every other port,without considering whether it is destined for a particular destination or not.**



**Types of Hubs:**

**.Passive Hubs**

**🡪A passive hub is just a connecter .It connects the wires coming from different branches.In a star-topology ethernet LAN.A passive hub is just a point where the signals coming from different stations collide.**

**.Active Hubs**

**🡪An active hub is actually a multipart repeater.It is normally used to create connection between stations in a physical star topology.**

**Bridge:-**

🡪A bridge operates at the data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of the 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.

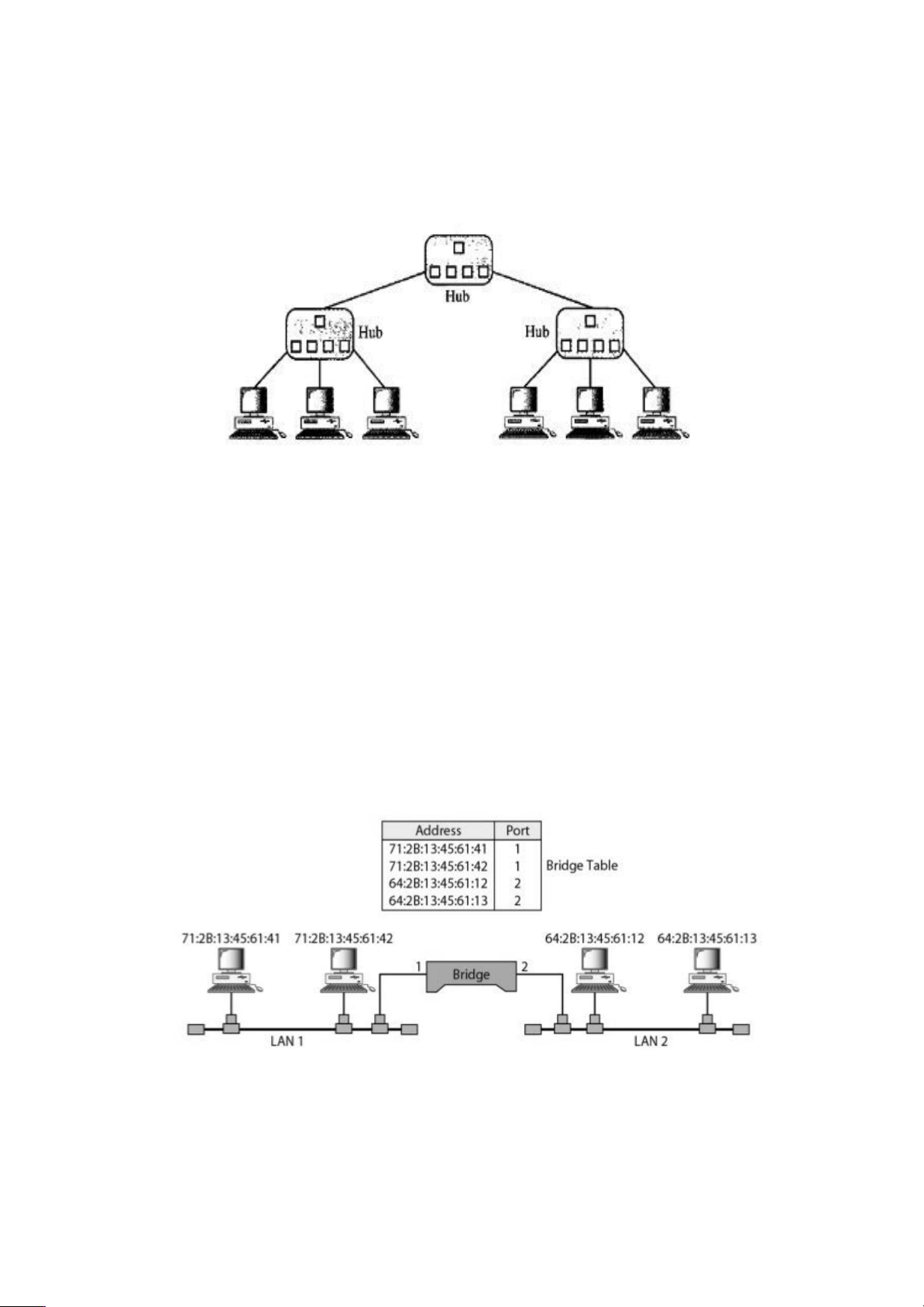
**Types of Bridges**

* **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.

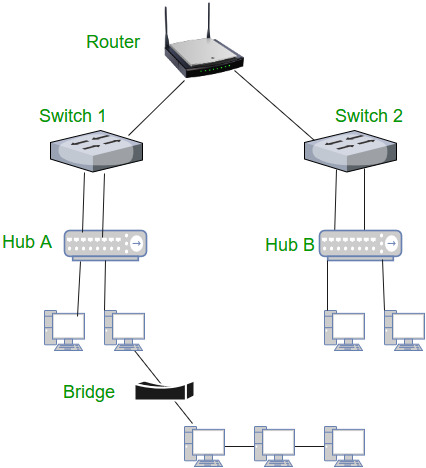
* **Source Routing Bridges**

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

****

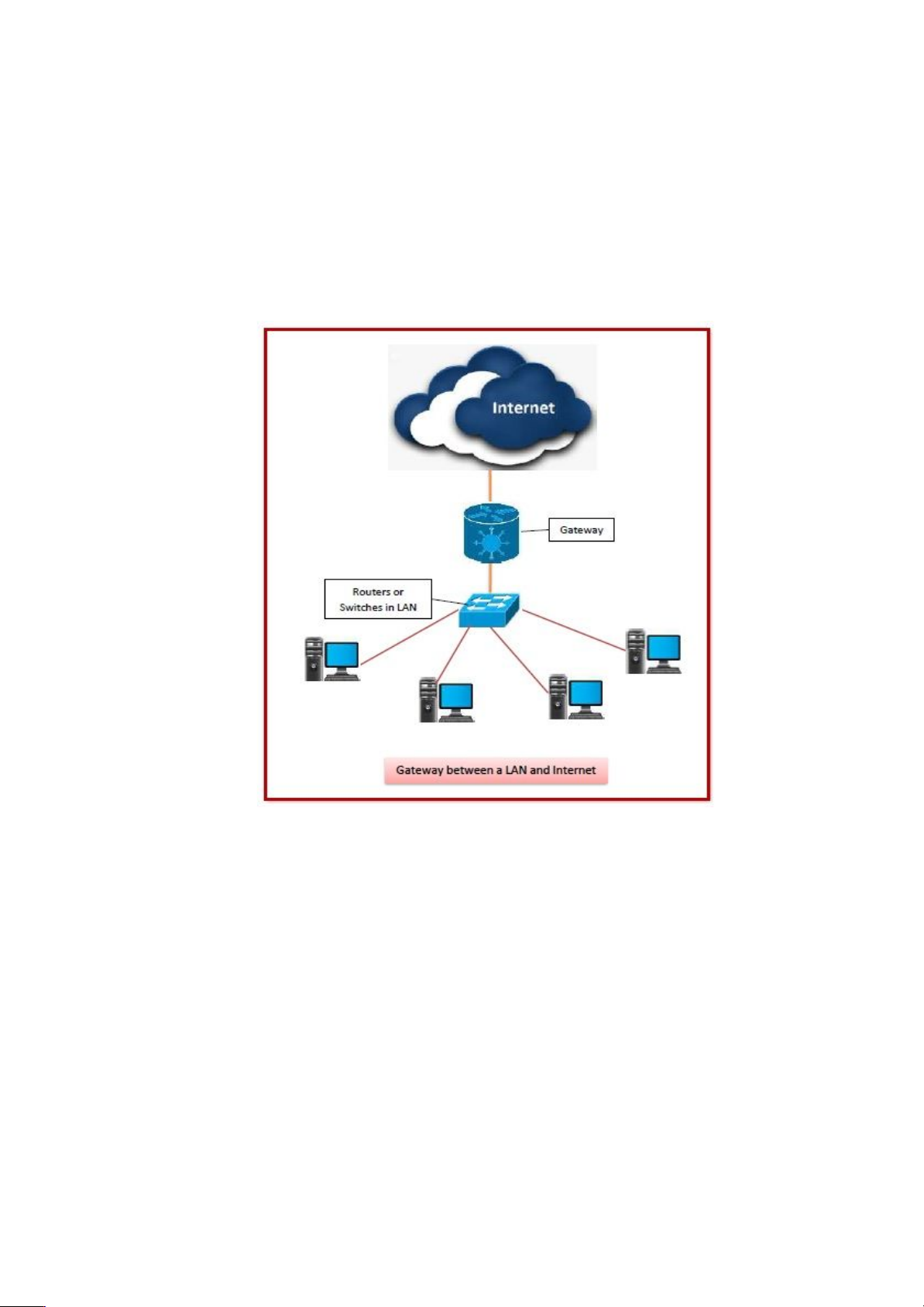
**Routers**:-

🡪 A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs and have a dynamically updating routing table based on which they make decisions on routing the data packets. The router divides the broadcast domains of hosts connected through it.



**Gateway**:-

🡪A gateway, as the name suggests, is a passage to connect two networks that may work upon different networking models. They work as 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 switches or routers. A gateway is also called a protocol converter.



**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, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to the correct port only.  In other words, the switch divides the collision domain of hosts, but the [broadcast domain](https://en.wikipedia.org/wiki/Broadcast_domain) remains the same.

Types of Switchs:

.Unmanaged Switch

🡪These are inexpensive switches commonly used in frame networks and small baseness.They can be set up by simple plugging into the network.

.Managed Switch

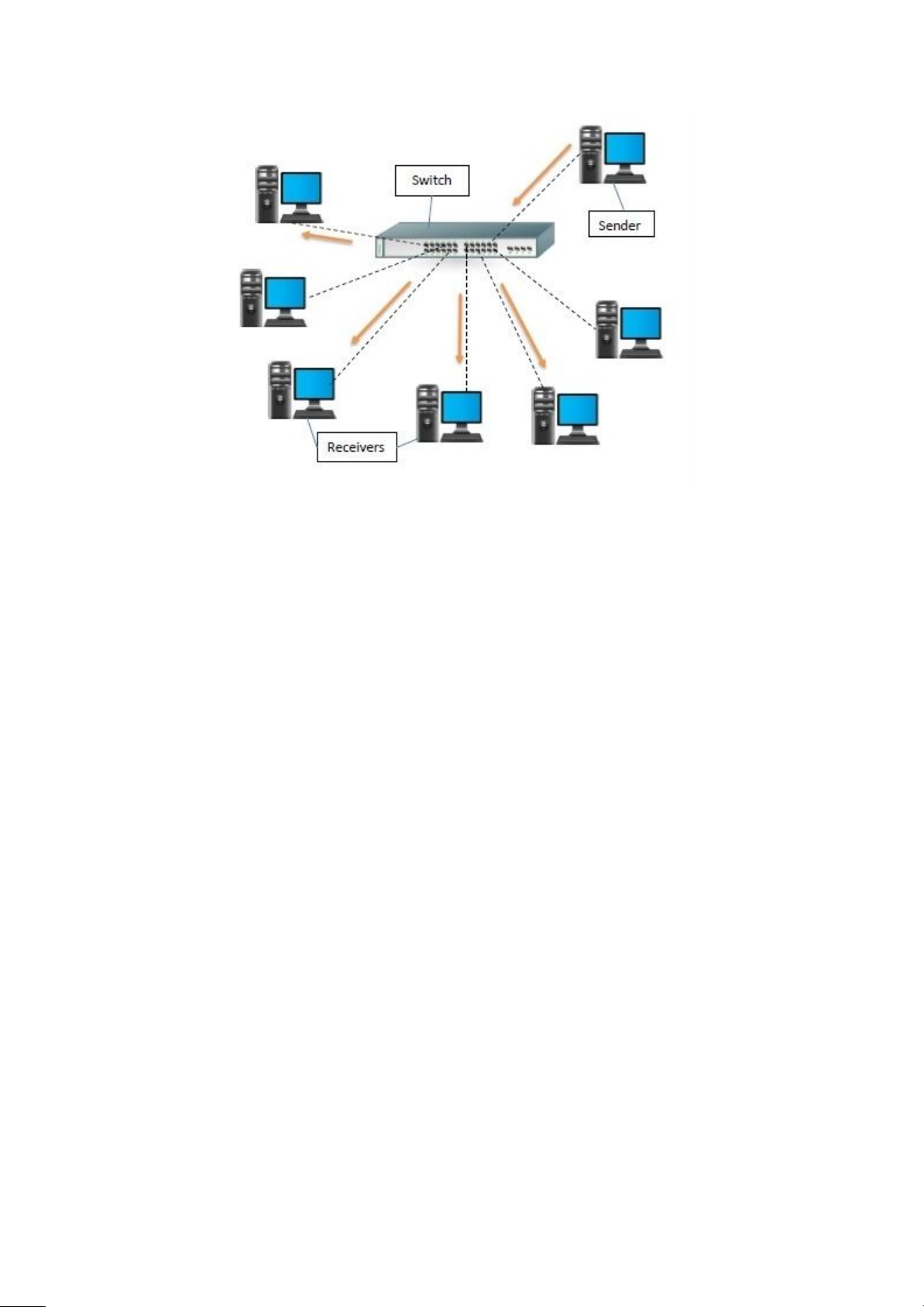
🡪These are costly switches that are used in organization with large and complex network,since they can be customized to augment the functionalities of switch.

LAN Switch:

🡪Local area network switches connect device in the internal LAN of an organization.They are also refer to as ethernet switch or data switches.

.POE Switch:

🡪Power over ethernet switches are used in poe gigabit ethernets.



2. Write a Program to implement the data link layer farming methods such as

i) Character stuffing

ii) bit stuffing.

i) Character stuffing

**Aim:-**

**To write a program to implement the data link layer framing methods by using character stuffing.**

**Source code:-**

#include<stdio.h>

main()

{

int frame[10],temp[20];

int i,n,j=0,count=0;

printf("Enter the length of the frame:");

scanf("%d",&n);

printf("Enter the bit frame: ");

for(i=0;i<n;i++)

{

scanf("%d",&frame[i]);

}

for(i=0;i<n;i++)

{

if(frame[i]==1)

{

count++;

}

else

{

count=0;

}

temp[j++]=frame[i];

if(count==3)

{

temp[j++]=0;

count=0;

}

}

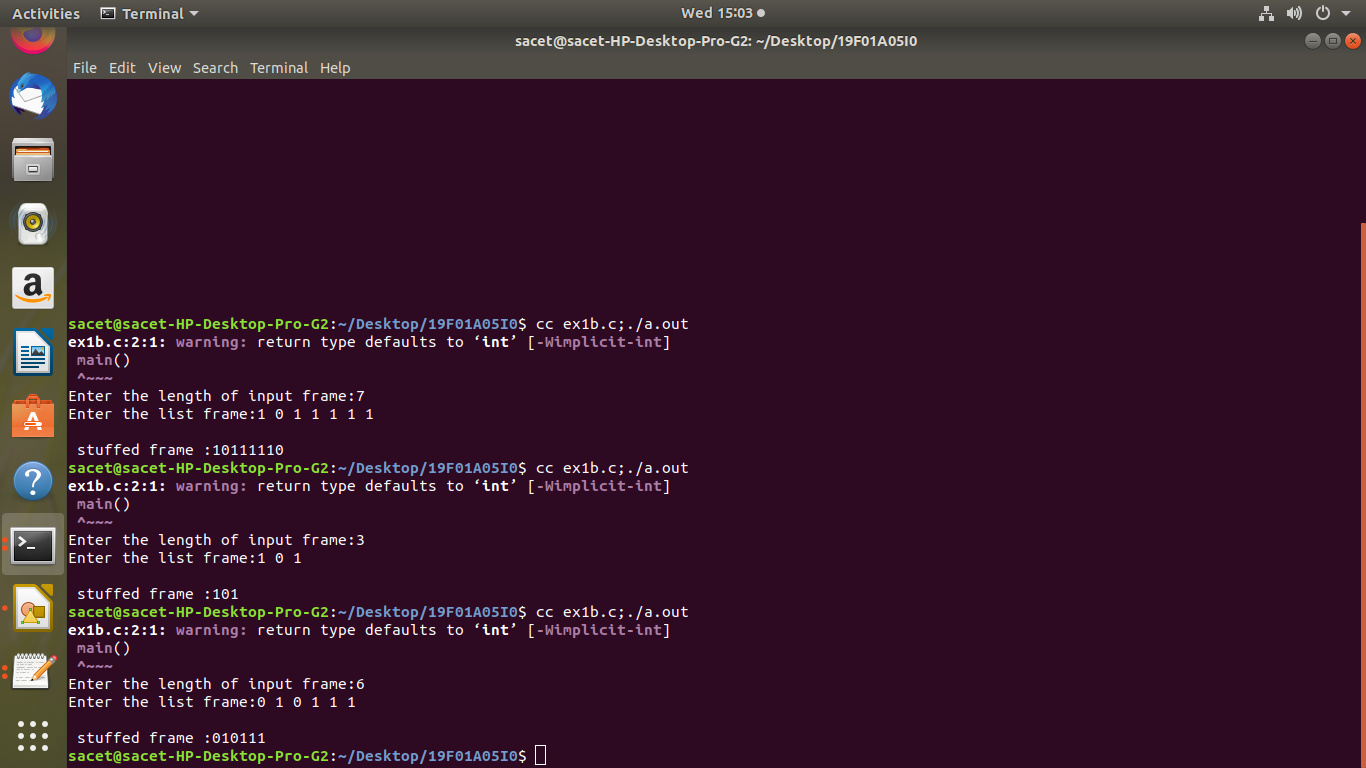
printf("\nThe mingled frame is: ");

for(i=0;i<j;i++)

printf("%d",temp[i]);

}

output:-



ii) bit stuffing.

**Aim:-**

**To write a program to implement the data link layer framing methods by using bit stuffing.**

**Source code:**

#include<stdio.h>

main()

{

int frame[10],temp[20];

int i,n,j=0,count=0;

printf("Enter the length of the frame:");

scanf("%d",&n);

printf("Enter the bit frame: ");

for(i=0;i<n;i++)

{

scanf("%d",&frame[i]);

}

for(i=0;i<n;i++)

{

if(frame[i]==1)

{

count++;

}

else

{

count=0;

}

temp[j++]=frame[i];

if(count==3)

{

temp[j++]=0;

count=0;

}

}

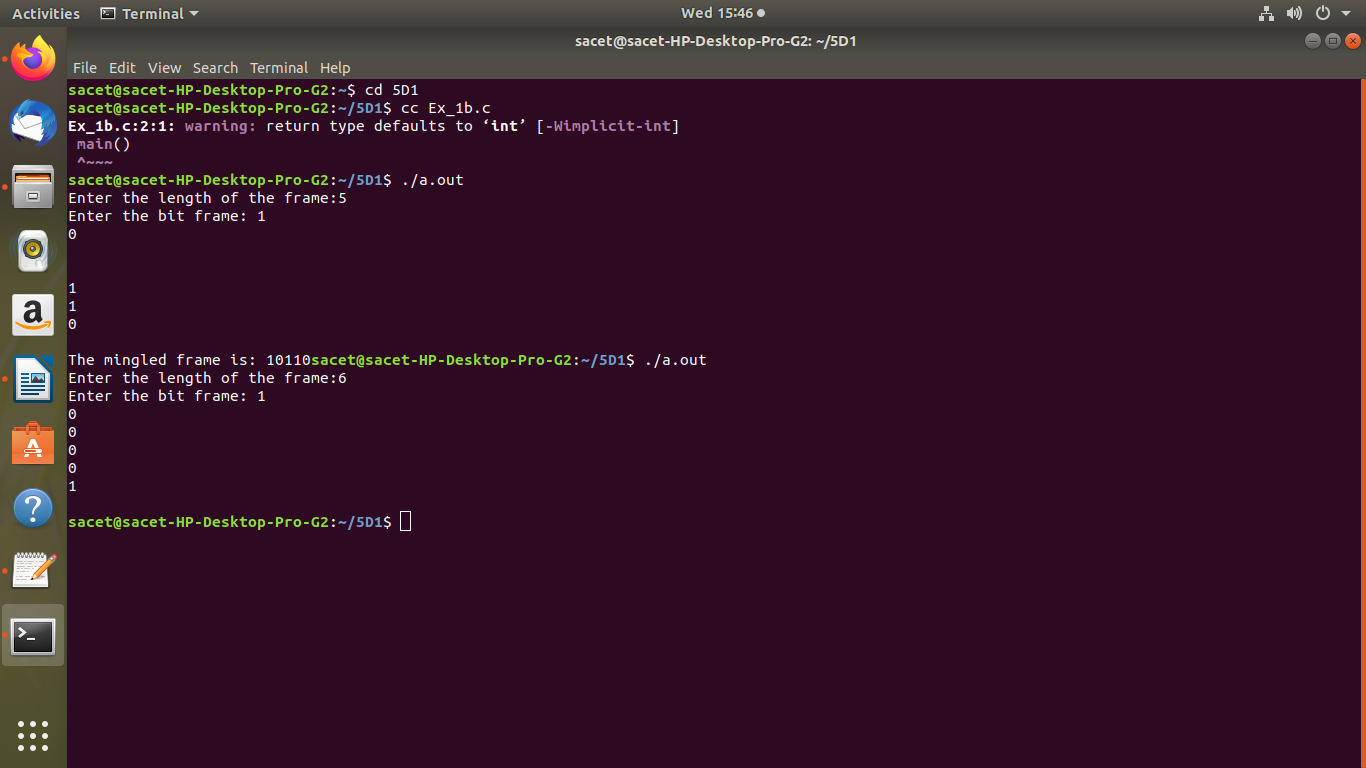
printf("\nThe mingled frame is: ");

for(i=0;i<j;i++)

printf("%d",temp[i]);

}

output:-



3. Write a Program to implement data link layer farming method checksum.

Aim:-

To write a program to implement data link layer framing method checksum

**1. gethostbyname**

The first method uses the traditional gethostbyname function to retrieve information about a hostname/domain name.

**Source code:**

#include<stdio.h> //printf

#include<string.h> //memset

#include<stdlib.h> //for exit(0);

#include<sys/socket.h>

#include<errno.h> //For errno - the error number

#include<netdb.h> //hostent

#include<arpa/inet.h>

int hostname\_to\_ip(char \* , char \*);

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

{

if(argc <2)

{

printf("Please provide a hostname to resolve");

exit(1);

}

char \*hostname = argv[1];

char ip[100];

hostname\_to\_ip(hostname , ip);

printf("%s resolved to %s" , hostname , ip);

printf("\n");

}

/\*

Get ip from domain name

\*/

int hostname\_to\_ip(char \* hostname , char\* ip)

{

struct hostent \*he;

struct in\_addr \*\*addr\_list;

int i;

if ( (he = gethostbyname( hostname ) ) == NULL)

{

// get the host info

herror("gethostbyname");

return 1;

}

addr\_list = (struct in\_addr \*\*) he->h\_addr\_list;

for(i = 0; addr\_list[i] != NULL; i++)

{

//Return the first one;

strcpy(ip , inet\_ntoa(\*addr\_list[i]) );

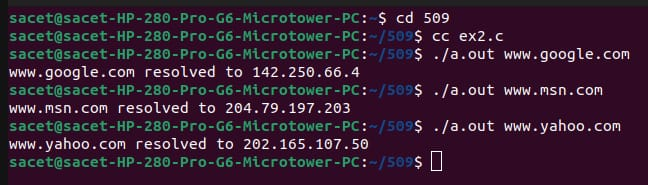
return 0;

}

return 1;

}

Output:-



**2. getaddrinfo**

The second method uses the getaddrinfo function to retrieve information about a hostname/domain name. The getaddrinfo supports ipv6 better.

**Source code:-**

#include<stdio.h> //printf

#include<string.h> //memset

#include<stdlib.h> //for exit(0);

#include<sys/socket.h>

#include<errno.h> //For errno - the error number

#include<netdb.h> //hostent

#include<arpa/inet.h>

int hostname\_to\_ip(char \* , char \*);

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

{

if(argc <2)

{

printf("Please provide a hostname to resolve");

exit(1);

}

char \*hostname = argv[1];

char ip[100];

hostname\_to\_ip(hostname , ip);

printf("%s resolved to %s" , hostname , ip);

printf("\n");

}

/\*

Get ip from domain name

\*/

int hostname\_to\_ip(char \*hostname , char \*ip)

{

int sockfd;

struct addrinfo hints, \*servinfo, \*p;

struct sockaddr\_in \*h;

int rv;

memset(&hints, 0, sizeof hints);

hints.ai\_family = AF\_UNSPEC; // use AF\_INET6 to force IPv6

hints.ai\_socktype = SOCK\_STREAM;

if ( (rv = getaddrinfo( hostname , "http" , &hints , &servinfo)) != 0)

{

fprintf(stderr, "getaddrinfo: %s\n", gai\_strerror(rv));

return 1;

}

// loop through all the results and connect to the first we can

for(p = servinfo; p != NULL; p = p->ai\_next)

{

h = (struct sockaddr\_in \*) p->ai\_addr;

strcpy(ip , inet\_ntoa( h->sin\_addr ) );

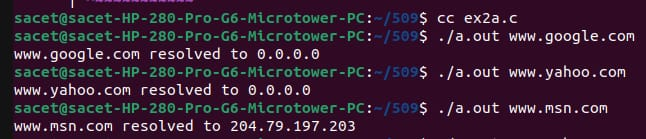
}

freeaddrinfo(servinfo); // all done with this structure

return 0;

}

Output:-



4. Write a program for Hamming Code generation for error detection and correction.

Aim:-

To write a program for Hamming Code generation for error detection and correction.

Source code:-

#include<string>

#include<math.h>

#include<stdlib.h>

#include<stdio.h>

int maxlength;

int parity;

char\*hammingstring=null;

void enterparameters(int\*length,int\*parity)

{

Printf(“enter the maximum length:”);

Scanf(%d”,length);

Printf(“enter the parity(0=even,1=odd):”);

Scanf(“%d”,parity);

}

Void checkhamming(char\*hammingstring,int parity)

{

Int i,j,k,start,length,paritynumber;

Printf(“enter the hamming code:”);

Scanf(“%s”,hammingstring);

Int errorbit=0;

Length=strlen(hammingstring);

Length--;

If(length>maxlength)

{

Printf(‘\n\*\*invalid entry-exceeds maximum code length of %d\n\n”,maxlength);

Return;

}

Paritynumber=ceil(log(length)/log(2));

For(i=0;i<paritynumber;i++)

{

Start =pow(2,i);

int paritycheck=parity;

for(j=start;j<length;j=j+(2\*start))

{

For(k=j;(k<((2\*j)-1))&&(k<length);k++)

{

Paritycheck^=(hammingstring[length-k]-‘0’);

}

}

Errorbit=errorbit+(paritycheck\*start);

}

If(errorbit==0)

{

Printf(“no error\n”);

}

Else

{

Hammingstring[length-errorbit]=’0’;

}

Printf(“the corrected hamming code is:%s\n”,hammingstring);

}

}

int main()

{

int parity;

int choice=0;

printf(“error detection/correction:\n”);

printf(“----------------\n”);

printf(“1)enter parameters\n”);

printf(“2)check hamming code\n”);

printf(“3)exit\n”);

printf(“\n enter selection:”);

scanf(%d”,&choice);

while(choice!=3)

{

If(choice==1)

Enterparameters(&maxlength,&parity);

Hammingstring=(char\*)malloc(maxlength\*sizeof(char));

Main();

}

Else if(choice==2)

{

Checkhamming(hammingstring,parity);

Main();

}

Else

{

Printf(“valid options are 1,2,0r3.quitting program.\n”);

Exit(0);

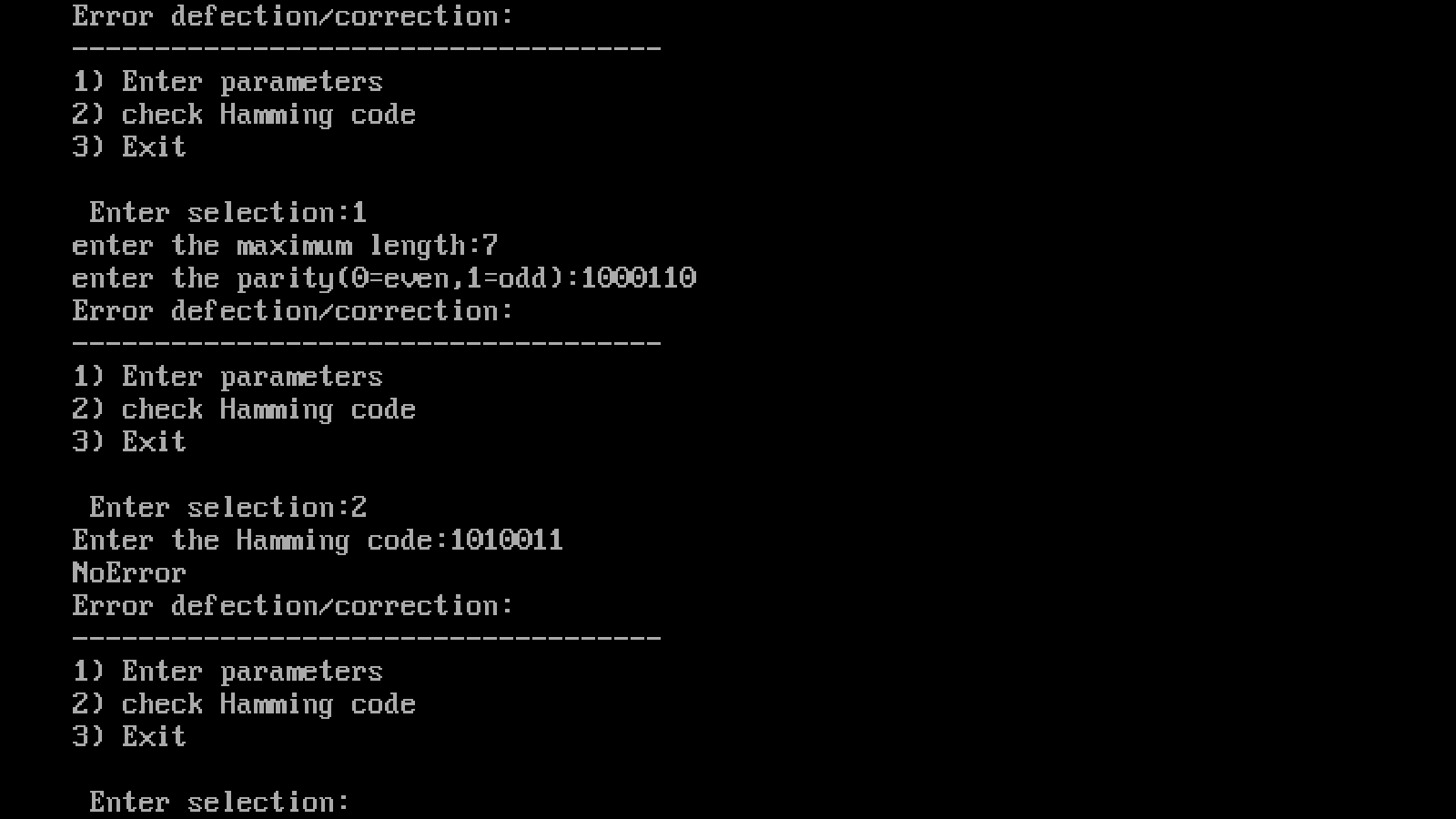
}

}

Exit(0);

}

Output:-



5. Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.

Aim:-

To Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.

Source code:-

#include<stdio.h>

#include<string.h>

int main()

{

char d[30],p[100],s[100],r[100];

int len\_divisor,i,len\_divident,flag=0;

printf("Enter message to be send: ");

scanf("%s",s);

printf("Enter the generator:");

scanf("%s",d);

strcpy(p,s);

len\_divisor=strlen(d)-1;

len\_divident=strlen(p);

for(i=len\_divident;i<len\_divident+len\_divisor;i++)

p[i]='0';

p[i]='\0';

printf("string after appending %d zeros:",len\_divisor-1);

puts(p);

while(strlen(p)>len\_divisor)

{

if(p[0]=='1')

{

for(i=0;p[i]!='\0';i++)

{

p[i]=((p[i]-'0') ^ (d[i]-'0')+'0');

}

}

else

{

for(i=0;p[i]!='\0';i++)

{

p[i]=p[i+1];

}

}

}

strcat(s,p);

printf("string after appending CRC code is :");

puts(s);

printf("\nAt receiver side:");

printf("\nEnter received message:");

for(i=0;i<strlen(s);i++)

{

scanf("%s",r[i]);

}

while(strlen(r)>len\_divisor)

{

if(r[0]=='1')

{

for(i=0;r[i]!='\0';i++)

{

r[i]=((r[i]-'0') ^ (d[i]-'0')+'0');

}

}

else

{

for(i=0;r[i]!='\0';i++)

{

r[i]=r[i+1];

}

}

}

for(i=0;i<strlen(r);i++)

{

if(r[i]=='0')

{

}

else

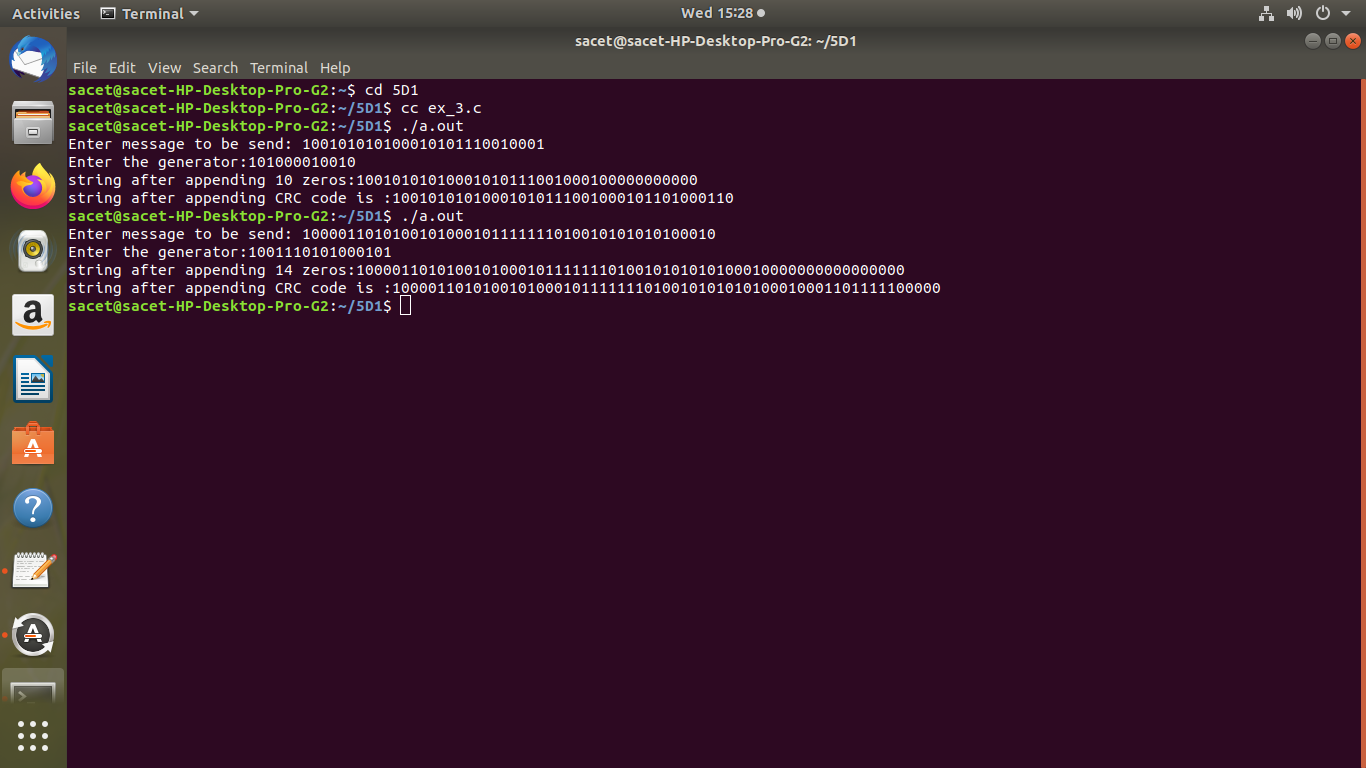
{

printf("Received wrong message");

}

}

output:-



6. Write a Program to implement Sliding window protocol for Goback N.

Aim:-

To Write a Program to implement Sliding window protocol for Goback N.

Source code:-

#include<stdio.h>

int main()

{

int w,i,f,frame[50];

printf("Enter window size : ");

scanf("%d",&w);

printf("\nEnter no.of frames:",f);

scanf("%d",&f);

printf("\nEnter %d frames:",f);

for(i=1;i<=f;i++)

{

scanf("%d",&frame[i]);

}

printf("\nWith sliding window protocol the frames will be send in following manners consuming no corruption frame\n\n");

printf("After sending %d frames at each stage sender waits for acknowledgement set by the receiver\n\n",f);

for(i=0;i<=f;i++)

{

if(i%w==0)

{

printf("%d\n",frame[i]);

printf("Acknowledgement of above frames sent is received by sender\n\n");

}

else

printf("%d\n",frame[i]);

}

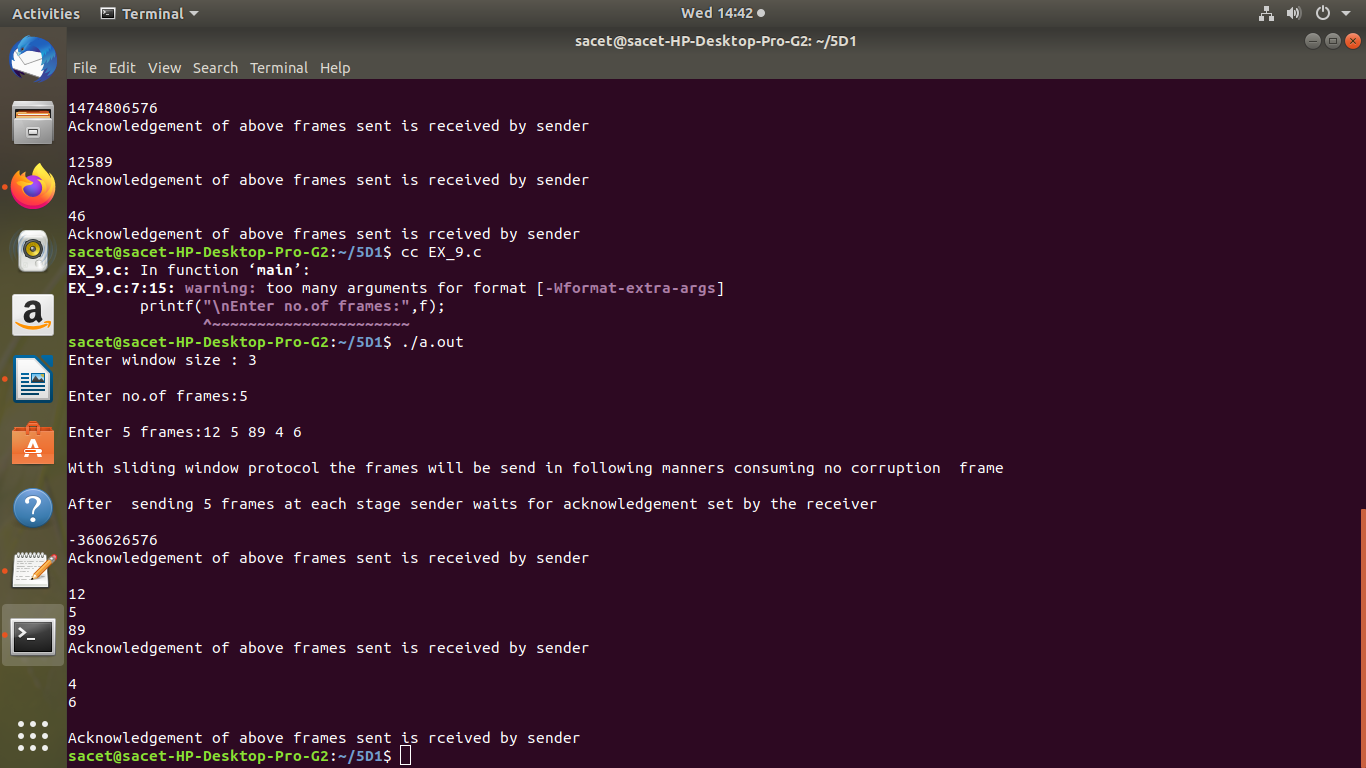
if(f%w!=0)

printf("\nAcknowledgement of above frames sent is rceived by sender\n");

return 0;

}

output:



7. Write a Program to implement Sliding window protocol for Selective repeat.

Aim:-

To Write a Program to implement Sliding window protocol for Selective repeat.

Source code:-

#include<stdio.h>

int main()

{

int w,t,f,frames[50];

printf("enter window size:");

scanf("%d",&w);

printf("\n enter number of frames to transmit:");

scanf("%d",&f);

printf("\n enter %d frames:",f);

for(i=1;i<=f;i++)

scanf("%d",&frames[1]);

printf("\n with sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames) \n\n");

printf("after sending %d frames at each stage sender waits for acknowledgement sent by the rectever \n\n",w);

for(i=1;i<=f;i++)

{

if(i%w==0)

{

printf("%d\n", frames[1]);

printf("acknowledgement of above frames sent is recieved by sender\n\n");

}

else

printf("%d\n", frames[1]);

}

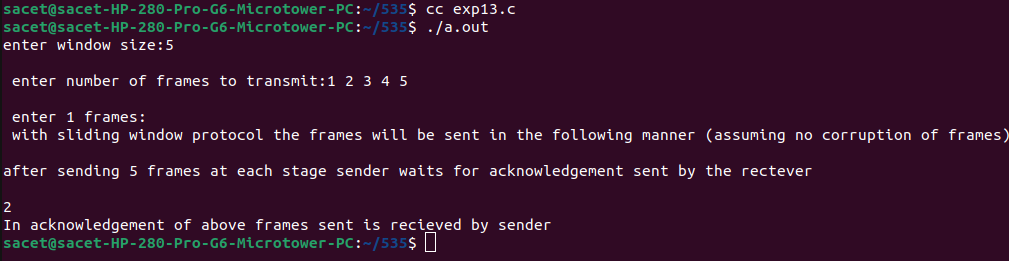
if(f%w!=0)

printf("In acknowledgement of above frames sent is recieved by sender\n");

return 0;

}

Output:-



8. Write a Program to implement Stop and Wait Protocol.

Aim:-

To Write a Program to implement Stop and Wait Protocol.

Source code:-

#include<stdio.h>

int main()

{

int framesize,sent=0,ack,i;

printf("Enter number of frames\n");

scanf("%d",&framesize);

while(1)

{

for(i=0;i<framesize;i++)

{

printf("frame %d has been transmitted.\n",sent);

sent++;

if(sent==framesize)

break;

}

printf("\n please enter the last acknowledgement receieeved.\n");

scanf("%d",&ack);

if(ack>=framesize)

break;

else

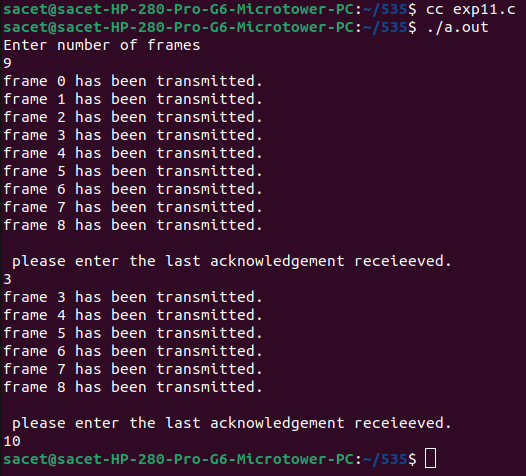
sent=ack;

}

return 0;

}

Output:-



9. Write a program for congestion control using leaky bucket algorithm.

Aim:-

To write a program for congestion control using leaky bucket algorithm.

Source code:-

#include<stdio.h>

#include<stdlib.h>

#define min(x,y) (x<y)?x:y

int main()

{

int outrate, drop=0,bsize,rem=0,nsec;

int input[10]={0},i=0,ch,x;

printf("\n Enter Bucket size and output rate:");

scanf("%d%d",&bsize,&outrate);

do

{

printf("\n Enter no of packets conning at sec %d ",i+1);

scanf("%d",&input[i]);

i++;

printf("\n Enter 1 to continue or 0 to quit...\n");

scanf("%d",&ch);

}while(ch);

nsec=i;

printf("\n Time Recived\t Sent \t Dropped \t Remaning\n");

for(i=0;i<nsec||rem;i++)

{

printf("%d",i+1);

printf("%5d",input[i]);

printf("%10d",min(input[i]+rem, outrate));

if((x= input[i]+rem-outrate)>0)

{

if(x>bsize)

{

rem=bsize; drop=x-bsize;

}

else

{

rem=x;

drop=0;

}

}

else

{

drop=0;

rem=0;

}

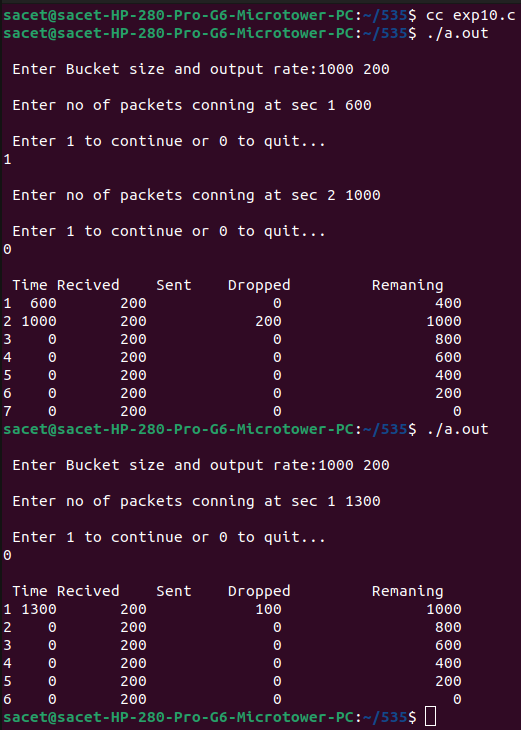
printf("%15d%20d\n",drop,rem);

}

return 0;

}

Output:-



10. Write a Program to implement Dijkstra‘s algorithm to compute the Shortest path through a graph.

Aim:-

To Write a Program to implement Dijkstra‘s algorithm to compute the Shortest path through a graph.

Source code:-

#include<stdio.h>

#define MAX 10

# define INFINITY 999

void dijkstra(int G[MAX][MAX],int n,int startnode);

int main()

{

int G[MAX][MAX],n,i,j,s;

printf("Enter no. of vertices: ");

scanf("%d",&n);

printf("Enter adjacency matrix:\n");

for(i=0;i<n;i++)

for(j=0;j<n;j++)

scanf("%d",&G[i][j]);

printf("Enter the starting node:");

scanf("%d",&s);

dijkstra(G,n,s);

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[startnode]=0;

visited[startnode]=1;

count=1;

while(count<n-1)

{

mindistance=INFINITY;

for(i=0;i<n;i++)

{

if(distance[i]<INFINITY && !visited[i])

{

mindistance=distance[i];

nextnode=i;

}

}

for(i=0;i<n;i++)

{

if(!visited[i])

if(mindistance+cost[nextnode][i]<distance[i])

{

distance[i]=mindistance+cost[nextnode][i];

pred[i]=nextnode;

}

count++;

}

}

for(i=0;i<n;i++)

if(i!=startnode)

{

printf("\nDistance of node%d=%d",i,distance[i]);

printf("\npath=%d",i);

j=i;

do

{

j=pred[j];

printf("<-%d",j);

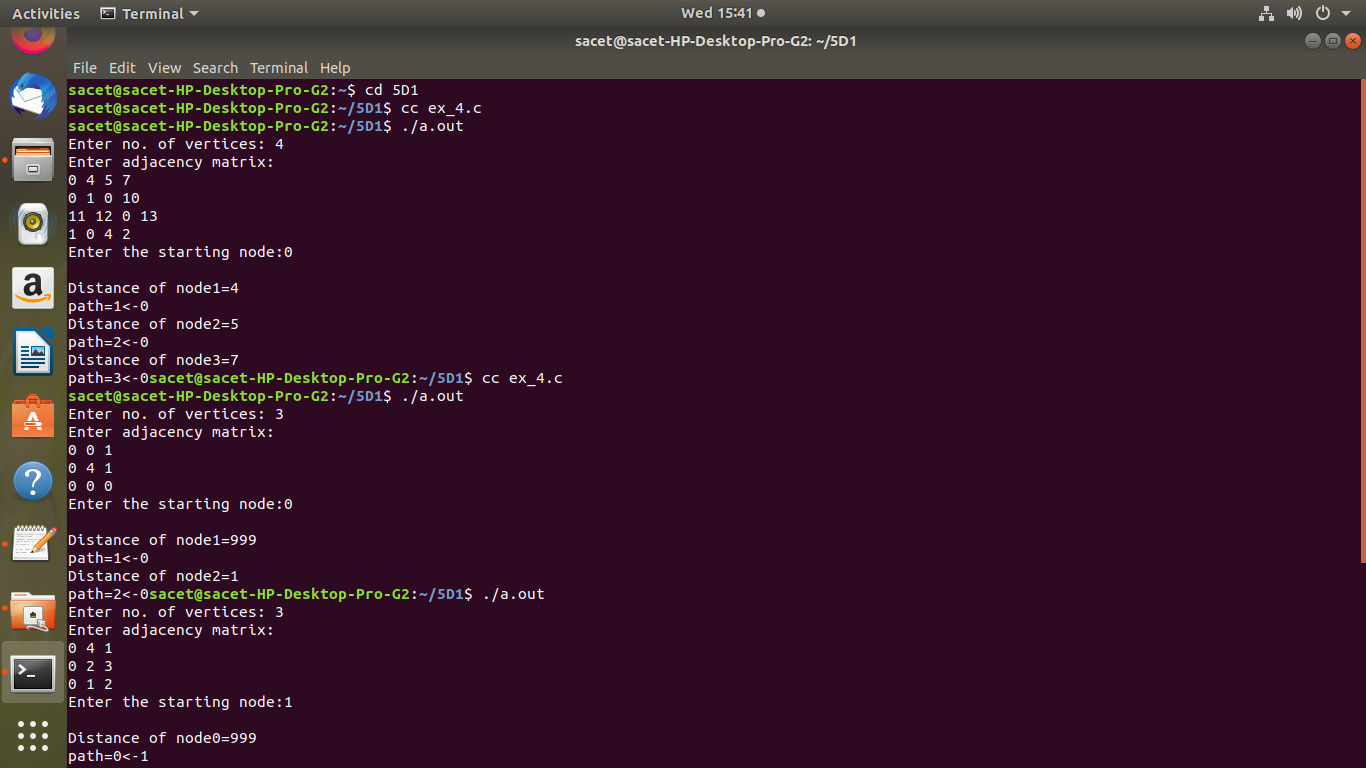
}

while(j!=startnode);

}

}

output:-



11. Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

Aim:-

To Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

Source code:-

include<stdio.h>

int dist[50][50],temp[50][50],n,i,j,k,x;

void dvr();

int main()

{

printf("\nEnter the number of nodes : ");

scanf("%d",&n);

printf("\nEnter the distance matrix :\n");

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

scanf("%d",&dist[i][j]);

dist[i][i]=0;

temp[i][j]=j;

}

printf("\n");

}

dvr();

printf("enter value of i &j:");

scanf("%d",&i);

scanf("%d",&j);

printf("enter the new cost");

scanf("%d",&x);

dist[i][j]=x;

printf("After update\n\n");

dvr();

return 0;

}

void dvr()

{

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

for (k = 1; k <= n; k++)

if (dist[i][k] + dist[k][j] < dist[i][j])

{

dist[i][j] = dist[i][k] + dist[k][j];

temp[i][j] = k;

}

for(i=1;i<=n;i++)

{

printf("\n\n\n\n router %d table ",i);

printf("\n\n From router %d \n",i);

for(j=1;j<=n;j++)

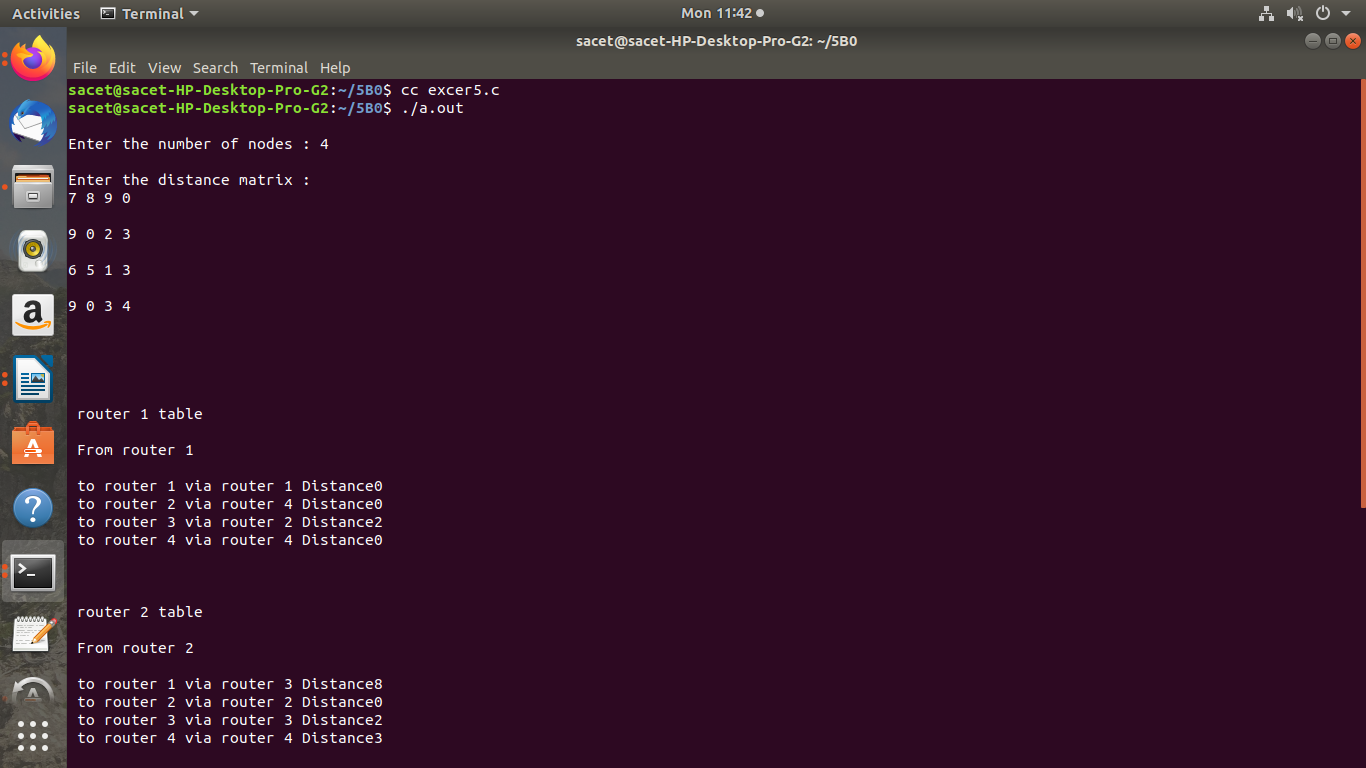
printf("\t\n to router %d via router %d Distance%d",j,temp[i][j],dist[i][j]);

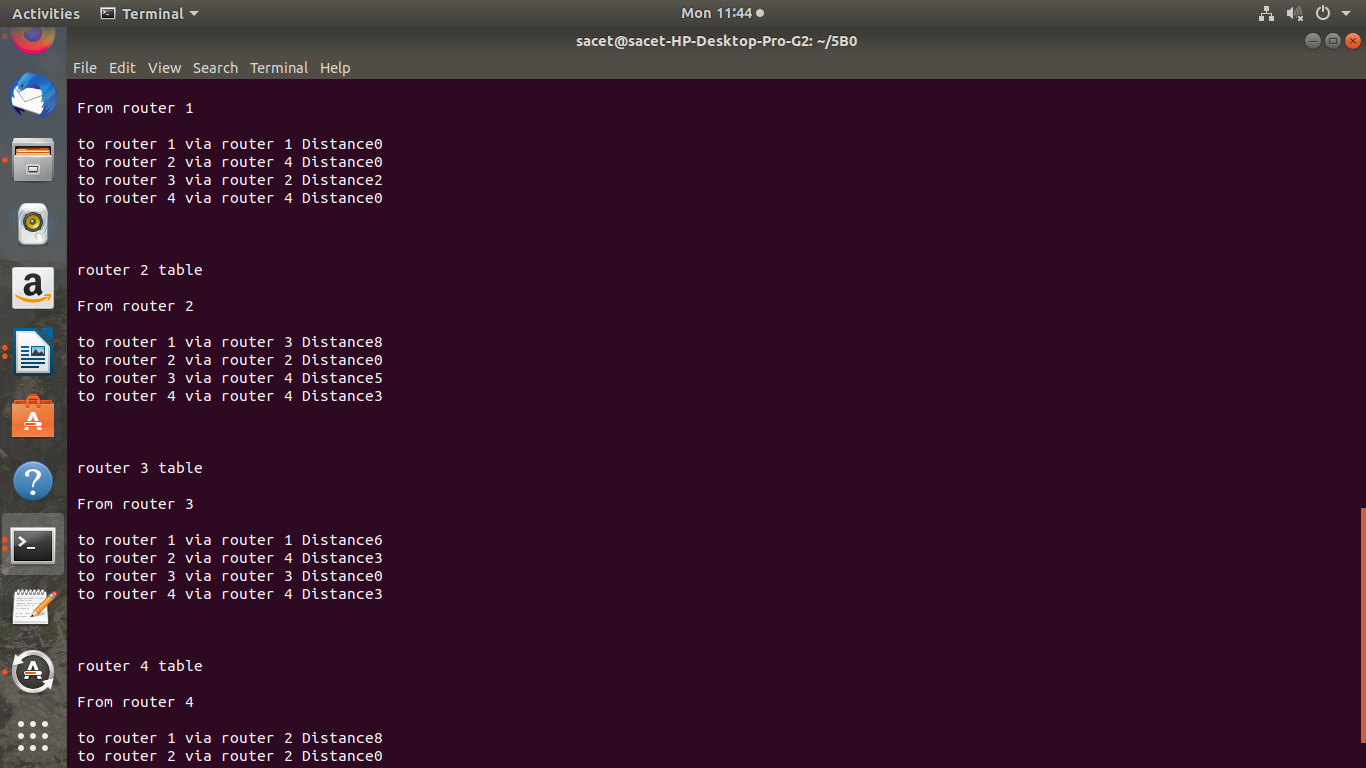
}

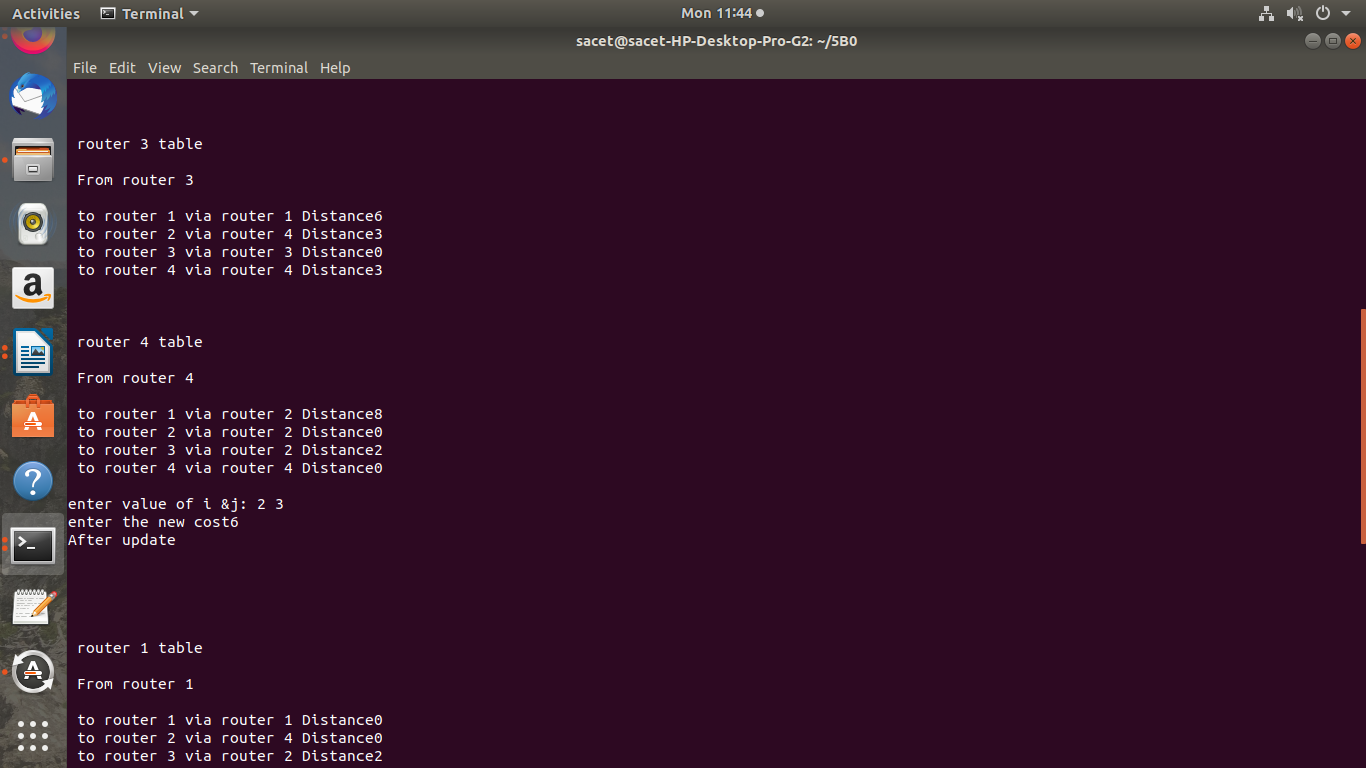
printf("\n\n");

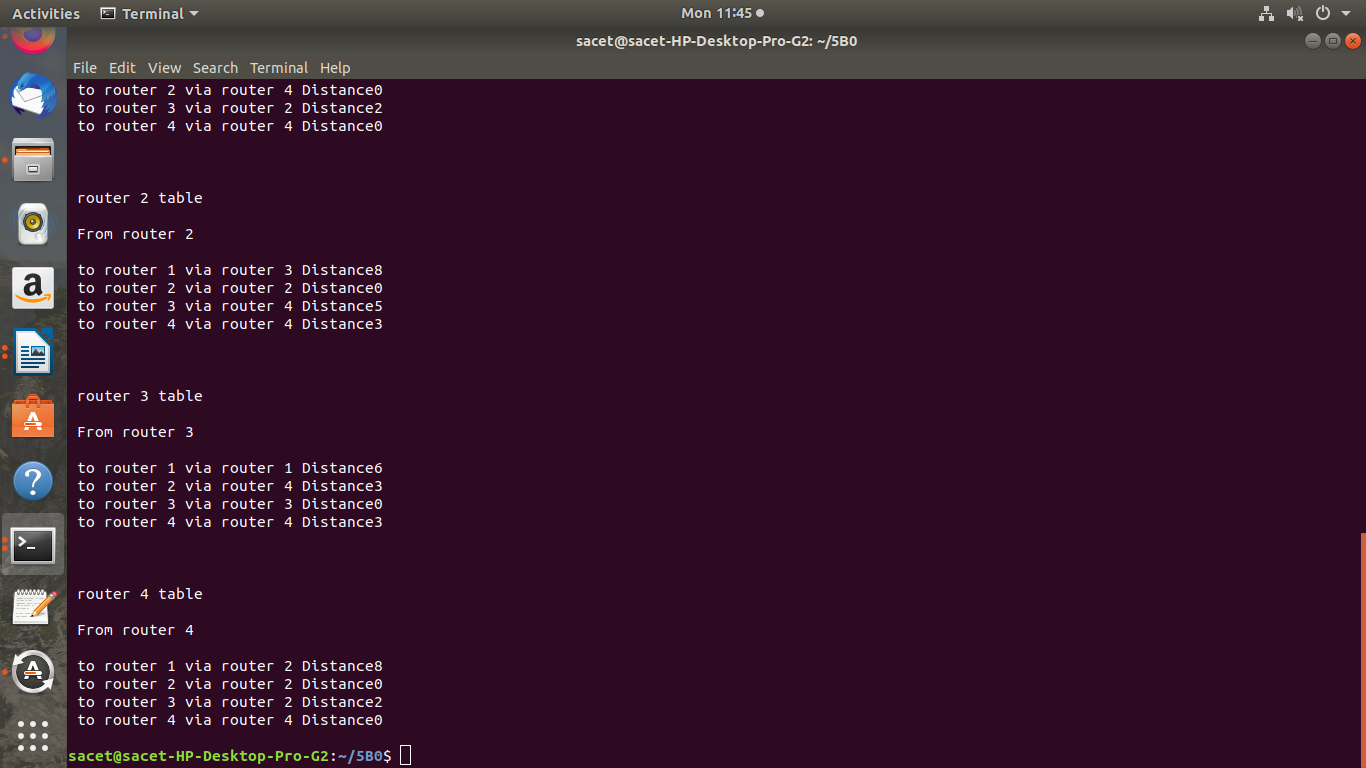
}

Output:-









12. Write a Program to implement Broadcast tree by taking subnet of hosts.

Aim:-

To Write a Program to implement Broadcast tree by taking subnet of hosts.

Source code:-

#include <stdio.h>

//#include <conio.h>

int i,j,k,x,y,u,v,n,ne=1;

int min,mincost=0,cost[10][10],parent[10]= {-1};

int find(int);

int sunion(int,int);

void main()

{

printf("\nEnter the no. of vertices:");

scanf("%d",&n);

printf("\nEnter the cost adjacency matrix:\n");

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

scanf("%d",&cost[i][j]);

if(cost[i][j]==0)

cost[i][j]=999;

}

}

printf("The edges of broadcast Tree are\n");

while(ne < n)

{

for(i=1,min=999;i<=n;i++)

{

for(j=1;j<=n;j++)

{

if(cost[i][j] < min)

{

min=cost[i][j];

x=u=i;

y=v=j;

}

}

}

j=find(u);

k=find(v);

if(sunion(j,k))

{

printf(" edge %d : (%d,%d) cost =%d\n",ne++,u,v,min);

mincost +=min;

}

cost[x][y]=cost[y][x]=999;

}

printf("\n\tMinimum cost = %d\n",mincost);

//getch();

}

int find(int i)

{

while(parent[i]>0)

i=parent[i];

return i;

}

int sunion(int i,int j)

{

if(i!=j)

{

parent[j]=i;

return 1;

}

return 0;

}

Output:-

