

EC8563 Communication Networks

Laboratory

Ex.No: 1 CRC

```
Start here x distance vector routing.c x link state routing.c x Hamming Code.c x crc.c x
1 //CRC
2 #include<stdio.h>
3 #include<conio.h>
4 void main()
5 {
6     int n,t,s=0,l,a,i,f[20],m[50],g[50],j,temp,c[15],z[15];
7     printf("enter the number of bits of message:");
8     scanf("%d",&n);
9     printf("enter the message");
10    for(i=0;i<n;i++)
11    { scanf("%d",&m[i]);
12    }
13    for(i=0;i<n;i++)
14    { z[i]=m[i];
15    }
16    printf("enter the number of bits of generator:");
17    scanf("%d",&l);
18    printf("Enter the generator number");
19
20    for(i=0;i<l;i++)
21    { scanf("%d",&g[i]);
22    }
23    for(i=0;i<l-1;i++)
24    { m[n+i]=0;
25    }
26
27    for(i=0;i<n;i++)
28    { temp=i;
29    if(m[i]==1)
30    {
31        for(j=0;j<l;j++)
32        {
33            if(m[temp]==g[j])
34            {
35                m[temp]=0;
36                f[j]=0;
37            }
38        }
39    }
40    }
```

```
C:\Users\GOWTHAM\J\Desktop\output\crc.exe
enter the number of bits of message:5
enter the message 0 0 1 1
enter the number of bits of generator:3
Enter the generator number 1 0 1

The remainder is:001
The CRC is:1001101The recieved signal has errors :(
Process returned 13 (0xD)   execution time : 40.178 s
Press any key to continue.
```

Ex.No: 2 HDLC

```
Start here x distance vector routing.c x link state routing.c x Hamming Code.c x crc.c x HDLC.c x
1 //HDLC
2 #include<stdio.h>
3 #include<conio.h>
4 #include<string.h>
5 void main()
6 {
7     int a[20],b[30],s,m[20],fm[20],n,i,j,k,f,fs,count;
8     int f1[]={0,1,1,1,1,1,1,0};
9     int f2[]={0,1,1,1,1,1,1,0};
10
11    printf("\t\tTransmitter side\n\n");
12    printf("Enter the size of the message signal:\n ");
13    scanf("%d",&n);
14    printf("Enter the message signal:\n ");
15
16    for (i=0;i<n;i++)
17    scanf("%d",&a[i]);
18    i=0;
19    count=1;
20    j=0;
21
22    while(i<n)
23    {
24    {
25
26    if(a[i]==1)
27    {
28    {
29    b[j]=a[i];
30
31    for(k=i+1;a[k]==1 && k<n && count<5 ;k++)
32
33    {
34    j++;
35    b[j]=a[k];
36    count++;
37    }
38    }
39    }
```

```
C:\Users\GOWTHAM\J\Desktop\output\HDLC.exe
Transmitter side
Enter the size of the message signal:
8
Enter the message signal:
1 0 0 1 1 1 0 0
Receiver side
Received Signal
01111110100111001121292480111110
Original message signal
10011100
Process returned 13 (0xD)   execution time : 16.415 s
Press any key to continue.
```

Ex.No: 3 Hamming Code

The screenshot shows a C program for Hamming Code in a code editor and its execution output in a terminal window.

Code Editor (Hamming Code.c):

```
1 //Hamming Code
2 #include<stdio.h>
3
4 void main() {
5     int data[10];
6     int dataatrec[10],c1,c2,c3,i;
7
8     printf("Enter 4 bits of data one by one\n");
9     scanf("%d",&data[0]);
10    scanf("%d",&data[1]);
11    scanf("%d",&data[2]);
12    scanf("%d",&data[4]);
13
14    //Calculation of even parity
15    data[6]=data[0]^data[2]^data[4];
16    data[5]=data[0]^data[1]^data[4];
17    data[3]=data[0]^data[1]^data[2];
18
19    printf("\nEncoded data is\n");
20    for(i=0;i<7;i++)
21        printf("%d",data[i]);
22
23    printf("\nEnter received data bits one by one\n");
24    for(i=0;i<7;i++)
25        scanf("%d",&dataatrec[i]);
26
27    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
28    c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
29    c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
30    c=c3*4+c2*2+c1;
31
32    if(c==0) {
33        printf("\nNo error while transmission of data\n");
34    }
35    else {
36        printf("\nError on position %d",c);
37    }
38 }
```

Terminal Output (Hamming Code.exe):

```
Enter 4 bits of data one by one
1 0 0 1

Encoded data is
1001100

Enter received data bits one by one
1 0 0 1 1 0 0

No error while transmission of data

Process returned 0 (0x0)   execution time : 72.594 s
Press any key to continue.
```

Ex.No: 4 Distance Vector Routing

The screenshot shows a C program for Distance Vector Routing in a code editor and its execution output in a terminal window.

Code Editor (distance vector routing.c):

```
1 //Distance Vector Routing:
2 #include<stdio.h>
3 #include<string.h>
4 int main()
5 {
6     int i,j,k,n,a[10][10],b[10][10],source,s,d;
7     char ch;
8     printf("Enter the number of nodes");
9     scanf("%d",&n);
10    for(i=1;i<=n;i++)
11    {
12        for(j=1;j<=n;j++)
13        {
14            if(i==j)
15                a[i][j]=0;
16            else
17            {
18                printf("\n Distance between the host %d->%d:",i,j);
19                scanf("%d",&a[i][j]);
20            }
21        }
22    }
23    for(i=1;i<=n;i++)
24    {
25        for(j=1;j<=n;j++)
26        {
27            printf("%d\t",a[i][j]);
28        }
29        printf("\n");
30    }
31    do
32    {
33        printf("Enter the node to display the Routing table:\n");
34        scanf("%d",&source);
35        for(j=1;j<=n;j++)
36        {
37            if(source==j)
38            {
39                printf("Adjacent path is %d to %d:",j,j);
40                for(k=1;k<=n;k++)
41                {
42                    if(k!=j)
43                        printf("%d\t",a[j][k]);
44                }
45                printf("\n");
46            }
47        }
48        printf("Do you want to continue(y/n)");
49        scanf("%c",&ch);
50        if(ch=='y')
51            continue;
52        else
53            break;
54    } while(1);
55 }
```

Terminal Output (distance vector routing.exe):

```
Enter the number of nodes3

Distance between the host 1->2:2
Distance between the host 1->3:3
Distance between the host 2->1:2
Distance between the host 2->3:1
Distance between the host 3->1:5
Distance between the host 3->2:4

0   2   3
2   0   1
5   4   0

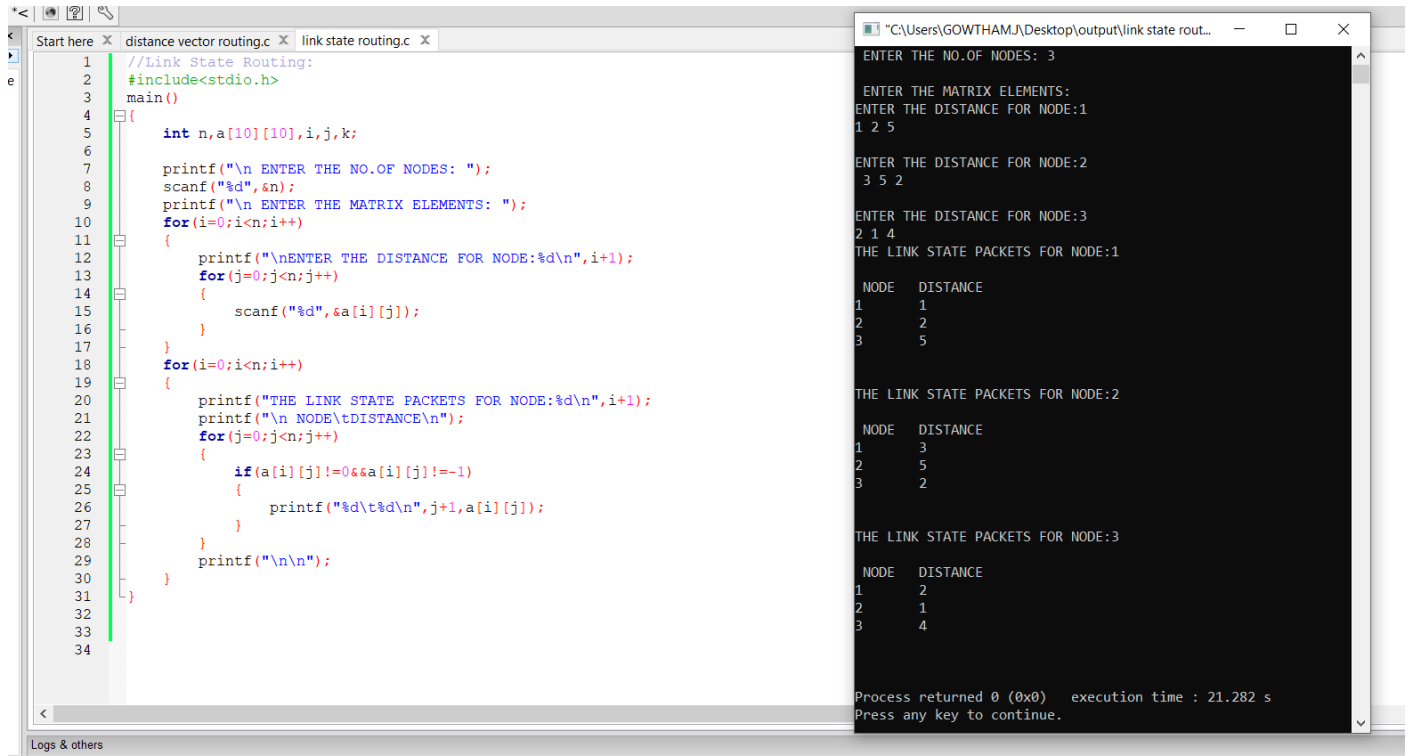
Enter the node to display the Routing table:
1
Adjacent path is 1 to 2:2
Adjacent path is 1 to 3:3

Do you want to continue(y/n)y
Enter the node to display the Routing table:
2
Adjacent path is 2 to 1:2
Adjacent path is 2 to 3:1

Do you want to continue(y/n)y
Enter the node to display the Routing table:
3
Adjacent path is 3 to 1:5
Adjacent path is 3 to 2:4

Do you want to continue(y/n)
```

Ex.No: 5 Link state Routing



```
//Link State Routing:
#include<stdio.h>
main()
{
    int n,a[10][10],i,j,k;

    printf("\n ENTER THE NO.OF NODES: ");
    scanf("%d",&n);
    printf("\n ENTER THE MATRIX ELEMENTS: ");
    for(i=0;i<n;i++)
    {
        printf("\nENTER THE DISTANCE FOR NODE:%d\n",i+1);
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<n;i++)
    {
        printf("THE LINK STATE PACKETS FOR NODE:%d\n",i+1);
        printf("\n NODE\tDISTANCE\n");
        for(j=0;j<n;j++)
        {
            if(a[i][j]!=0&&a[i][j]!=-1)
            {
                printf("%d\t%d\n",j+1,a[i][j]);
            }
        }
        printf("\n\n");
    }
}
```

ENTER THE NO.OF NODES: 3

ENTER THE MATRIX ELEMENTS:

ENTER THE DISTANCE FOR NODE:1

1 2 5

ENTER THE DISTANCE FOR NODE:2

3 5 2

ENTER THE DISTANCE FOR NODE:3

2 1 4

THE LINK STATE PACKETS FOR NODE:1

NODE	DISTANCE
1	1
2	2
3	5

THE LINK STATE PACKETS FOR NODE:2

NODE	DISTANCE
1	3
2	5
3	2

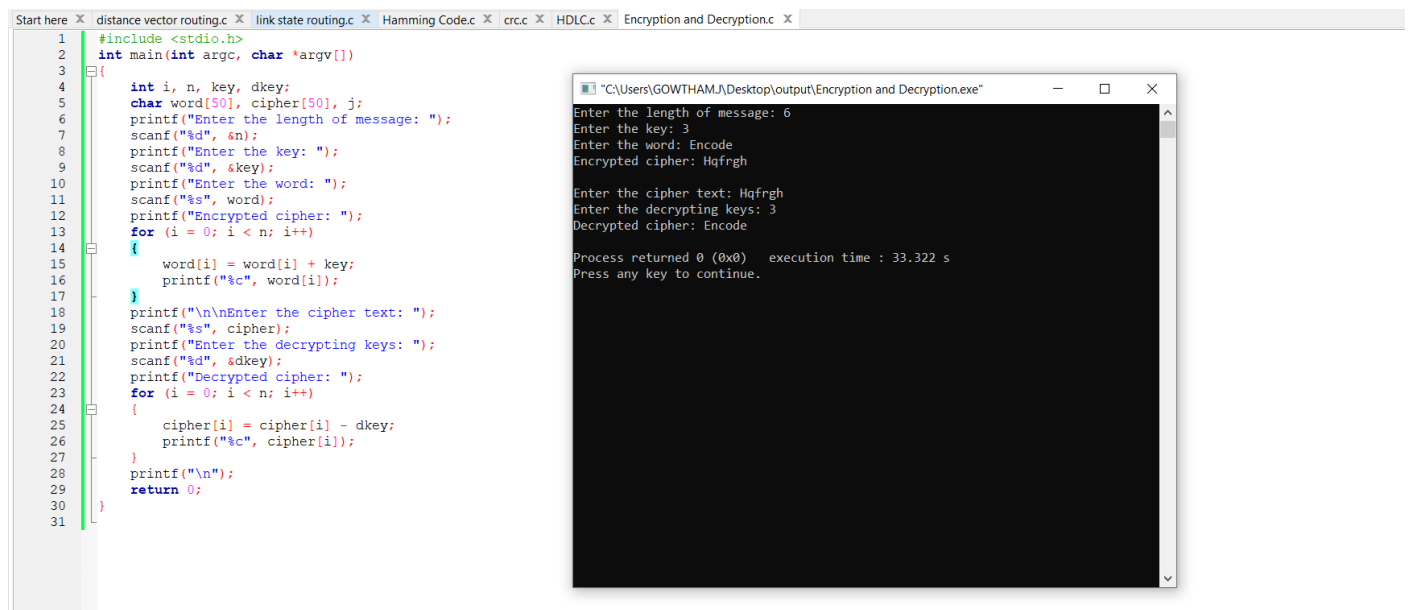
THE LINK STATE PACKETS FOR NODE:3

NODE	DISTANCE
1	2
2	1
3	4

Process returned 0 (0x0) execution time : 21.282 s

Press any key to continue.

Ex.No: 6 Encryption And Decryption



```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int i, n, key, dkey;
    char word[50], cipher[50], j;
    printf("Enter the length of message: ");
    scanf("%d", &n);
    printf("Enter the key: ");
    scanf("%d", &key);
    printf("Enter the word: ");
    scanf("%s", word);
    printf("Encrypted cipher: ");
    for (i = 0; i < n; i++)
    {
        word[i] = word[i] + key;
        printf("%c", word[i]);
    }
    printf("\n\nEnter the cipher text: ");
    scanf("%s", cipher);
    printf("Enter the decrypting keys: ");
    scanf("%d", &dkey);
    printf("Decrypted cipher: ");
    for (i = 0; i < n; i++)
    {
        cipher[i] = cipher[i] - dkey;
        printf("%c", cipher[i]);
    }
    printf("\n\n");
    return 0;
}
```

Enter the length of message: 6

Enter the key: 3

Enter the word: Encode

Encrypted cipher: Hqfrgh

Enter the cipher text: Hqfrgh

Enter the decrypting keys: 3

Decrypted cipher: Encode

Process returned 0 (0x0) execution time : 33.322 s

Press any key to continue.

Ex.No: 7 Stop And Wait ARQ

```
Start here x distance vector routing.c x link state routing.c x Hamming Code.c x crc.c x HDLC.c x Encryption and Decryption.c x wait.c x
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4
5 int main(int argc, char *argv[])
6 {
7     int i, j, noframes, x, x1 = 10, x2;
8     printf("Enter the number of frames: ");
9     scanf("%d", &noframes);
10    i = 1;
11    j = 1;
12
13    while (noframes > 0)
14    {
15        printf("Sending frame %d\n", i);
16        srand(x1++);
17        x = rand() % 10;
18        if (x % 2 == 0)
19        {
20            for (x2 = 1; x2 < 2; x2++)
21            {
22                printf("Waiting for %d seconds\n", x2);
23                sleep(x2);
24            }
25            printf("Sending frame %d\n", i);
26            srand(x1++);
27            x = rand() % 10;
28        }
29        printf("ACK for frame %d\n", j);
30        noframes -= 1;
31        i++;
32        j++;
33        printf("\n");
34    }
35    printf("End of Stop and Wait Protocol\n");
36    return 0;
37 }
```

```
C:\Users\GOWTHAM\J\Desktop\output\wait.exe
Enter the number of frames: 6
Sending frame 1
ACK for frame 1

Sending frame 2
Waiting for 1 seconds
Sending frame 2
ACK for frame 2

Sending frame 3
ACK for frame 3

Sending frame 4
Waiting for 1 seconds
Sending frame 4
ACK for frame 4

Sending frame 5
Waiting for 1 seconds
Sending frame 5
ACK for frame 5

Sending frame 6
ACK for frame 6

End of Stop and Wait Protocol

Process returned 0 (0x0)   execution time : 31.262 s
Press any key to continue.
```

Ex.No: 8 Go - Back N ARQ

```
Start here x distance vector routing.c x link state routing.c x Hamming Code.c x crc.c x HDLC.c x Encryption and Decryption.c x go.exe
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int reciever(int);
5 int simulate(int);
6
7 int main(int argc, char *argv[])
8 {
9     int temp1, temp2, temp3, temp4, i, winsize = 8, noframes, moreframes;
10    char c;
11    temp1 = temp2 = temp3 = temp4 = 0;
12    for (i = 0; i < 200; i++)
13        rand();
14    printf("Number of frames: ");
15    scanf("%d", &noframes);
16    moreframes = noframes;
17    while (moreframes > 0)
18    {
19        temp1 = simulate(winsize);
20        winsize -= temp1;
21        temp4 += temp1;
22        if (temp4 > noframes)
23            temp4 = noframes;
24        for (i = temp3 + 1; i <= temp4; i++)
25            printf("Sending frame %d\n", i);
26        temp2 = reciever(temp1);
27        temp3 += temp2;
28        if (temp3 > noframes)
29            temp3 = noframes;
30        printf("Acknowledgement for the frames upto %d\n", temp3);
31        moreframes -= temp2;
32        temp4 = temp3;
33        if (winsize <= 0)
34            winsize = 8;
35    }
36    printf("\n");
37    return 0;
38 }
```

```
C:\Users\GOWTHAM\J\Desktop\output\go.exe
Number of frames: 6
Sending frame 1
Sending frame 2
Sending frame 3
Sending frame 4
Acknowledgement for the frames upto 1
Sending frame 2
Acknowledgement for the frames upto 1
Sending frame 2
Acknowledgement for the frames upto 1
Sending frame 2
Sending frame 3
Acknowledgement for the frames upto 2
Sending frame 3
Sending frame 4
Sending frame 5
Sending frame 6
Acknowledgement for the frames upto 2
Sending frame 3
Sending frame 4
Sending frame 5
Acknowledgement for the frames upto 2
Sending frame 3
Sending frame 4
Sending frame 5
Sending frame 6
Acknowledgement for the frames upto 4
Sending frame 5
Acknowledgement for the frames upto 4
Sending frame 5
Sending frame 6
Acknowledgement for the frames upto 5
Sending frame 6
Acknowledgement for the frames upto 5
Sending frame 6
Acknowledgement for the frames upto 6

Process returned 0 (0x0)   execution time : 5.416 s
Press any key to continue.
```

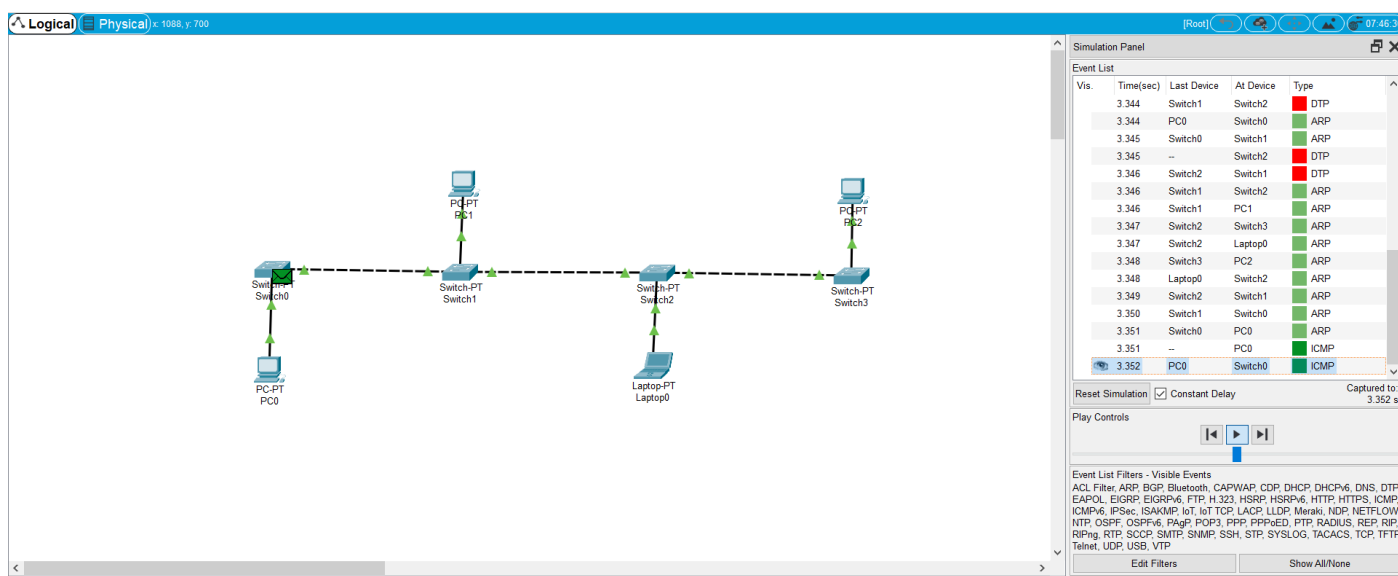
Ex.No: 9 Selective Repeat ARQ

```
Start here x distance vector routing.c x link state routing.c x Hamming Code.c x crc.c x HDLC.c x Encryption and Decryption.c x wait.c x go.c x *selective.c x
1 #include<stdio.h>
2
3 #include<conio.h>
4
5 #include<stdlib.h>
6
7 void main()
8 {
9     int temp1,temp2,temp3,temp4,temp5,i,winsize=8,noframes,moreframes;
10
11     char c;
12
13     int reciever(int);
14
15     int simulate(int);
16
17     int nack(int);
18
19     temp4=0,temp1=0,temp2=0,temp3=0,temp5 =0;
20
21     for(i=0;i<200;i++)
22     rand();
23
24     noframes=rand()/200;
25
26     printf("\n number of frames is %d",noframes);
27
28     getch();
29
30     moreframes=noframes;
31     while(moreframes>=0)
32     {
33         temp1=simulate(winsize);
34         winsize-=temp1;
35         temp4+=temp1;
36
37         temp4+=temp1;
```

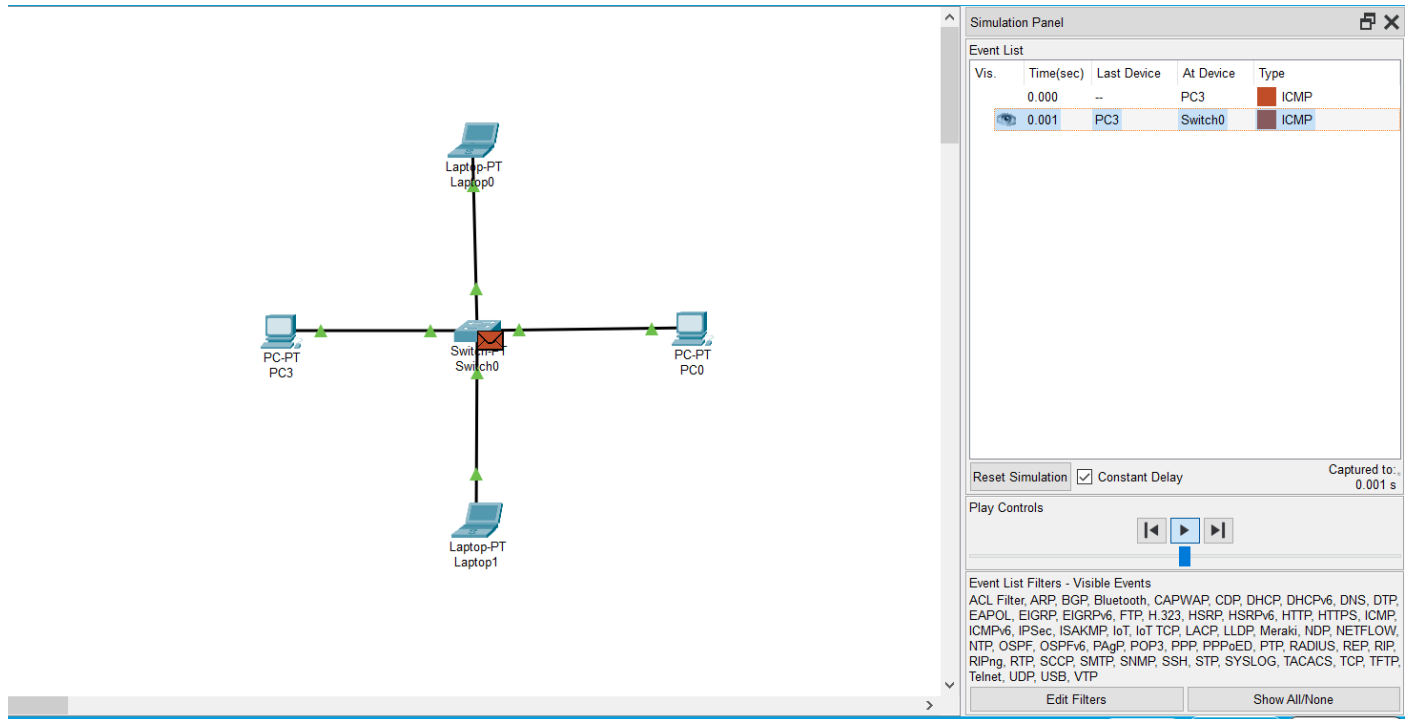
```
C:\Users\GOWTHAM\A\Desktop\output\selective.exe
number of frames is 71
sending frame 0
sending frame 1
sending frame 2
sending frame 3
sending frame 4
sending frame 5
sending frame 6
sending frame 7
No acknowledgement for the frame 1
sending frame 7
sending frame 8
No acknowledgement for the frame 1
sending frame 8
sending frame 9
sending frame 10
sending frame 11
sending frame 12
sending frame 13
sending frame 14
No acknowledgement for the frame 4
Retransmitting frame 4
Retransmitting frame 4
Retransmitting frame 4
sending frame 14
sending frame 15
sending frame 16
```

Ex.No: 10 Topologies

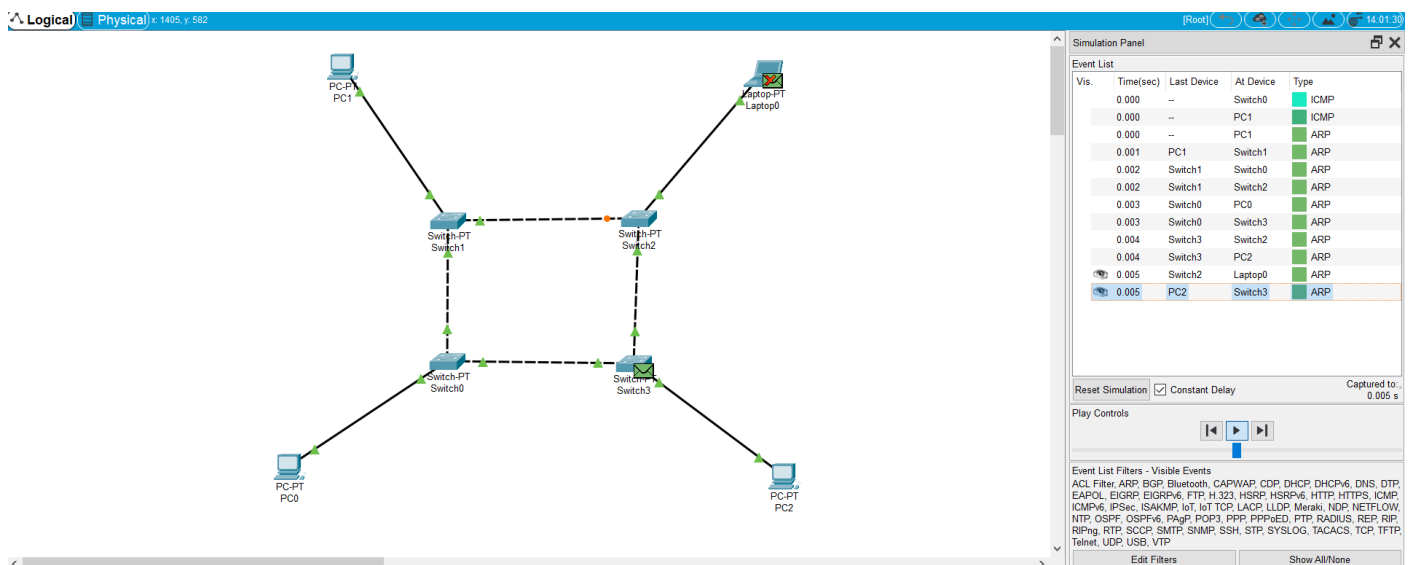
A. Bus Topology



B. Star Topology



C. Ring Topology



D. Mesh Topology

