Experiment 5: ARQ Mechanisms in DLL

Aim: To implement receiver algorithms for the different ARQ mechanisms at the Data Link Layer

Objective: After carrying out this experiment, students will be able to:

- implement receiver algorithms for the different ARQ mechanisms at the Data Link Layer
- Analyze the differences between the ARQ mechanisms

Problem statement: You are required to write a program that can receive frames at the data link layer. Assume that the user is entering the frames as the transmitter. You are required to implement stop and wait, go back N and selective repeat ARQ mechanisms. Consider that you have to transmit and receive a total of 20 frames using $W_T=W_R=1$, $W_T=5$ and $W_R=1$ and $W_T=W_R=5$ for stop and wait, go back N and selective repeat respectively

Analysis: While analyzing your program, you are required to address the following points:

- Difference between stop and wait, go back N and selective repeat.
- Comparison of the disadvantages of the different ARQ mechanisms.

MARKS DISTRIBUTION

Component	Maximum Marks	Marks Obtained
Preparation of Document	7	
Results	7	
Viva	6	
Total	**20**	

Submitted by: K Srikanth

Register No: 17ETCS002124

1. Algorithm/Flowchart

In the following algorithm, sender's data is transmitted to receiver which has to acknowledge of its reception to the sender. Since we cannot simulate the situation in a single computer. A random number is generated on every iteration (while sending 'I'th frame) and divided by 2, checking the remainder can help us evaluate two cases.

Case1: Remainder is 0: Interpreted as successful acknowledgement

Case2: Remainder is not zero: Interpreted as time-out and no acknowledgement, hence attempt for retransmission.

1. Start
2. Input total number of frames: N
3. For I=0:N do
    1. Input 'I'th Frame to be sent
    2. End for I
4. For I=0:N do
    1. Generate a random number : rand()
    2. If (rand()%2) do
        a. Copy from sender's data buffer to receiver's data buffer (rdata[I]=sdata[I])
    3. Else
        a. I=I-1
        b. Retransmit
    4. Endif
5. For I=0:N do
    1. Print rdata[I]
    2. End for I
6. End

2. Program

Stop and Wait

```c
#include <stdio.h>
#include <stdlib.h>
#include<time.h>
// K Srikanth 17ETCS002124
int main(int argc, char** argv) {
    int nframes,i;
    printf("Enter the Number of Frames");
    scanf("%d",&nframes);
    int sdata[nframes];
    int rdata[nframes];
    printf("Input the data to be transmitted \n");
    for(i=0;i<nframes;i++)
    {
        scanf("%d",&sdata[i]);
    }
    for(i=0;i<nframes;i++)
    {
        printf("Frame %d has been sent.\n",i+1);
        if(rand()%2==0)
        {
            rdata[i]=sdata[i];
            printf("Frame %d is successfully received.\n",i+1);
         }
        else
        {
            printf("Timed out! Initiating re-transmission.\n");
            i=i-1;
        }
    }
    printf("The Frames received are :");
    for(i=0;i<nframes;i++)
    {
        printf("%d ",rdata[i]);
    }
    return (EXIT_SUCCESS);
}
```

Figure 1 C Program for Stop and Wait Protocol

Go Back N

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
// K Srikanth 17ETCS002124
int main(int argc, char **argv)
{
    srand(time(0));
    int n, sender_window_size = 0;
    printf("Enter the Input Window Size : ");
    scanf("%d", &sender_window_size);
    printf("Enter the Number of Frames : ");
    scanf("%d", &n);
    int frames[n], i, j;
    printf("Sender Message :  ");
    for (i = 0; i < n; i++)
        scanf("%d", &frames[i]);
    int last_ack, bottom, sent = 0;
    do
    {
        printf("Frames Transmitted: ");
        for (i = sent; i < sent + sender_window_size && i < n; i++)
        {
            printf("%d  ", i + 1);
        }
        printf("\n");
        printf("Frames Acknowledged: ");
        bottom = sent;
```

Figure 2 C Program for Go Back N Protocol

```
        for (j = 0; j < sender_window_size && sent < n; j++)
        {
            last_ack = sent + random() % 2;
            if (last_ack == sent + 1)
            {
                printf("%d   ", bottom + j + 1);
                sent++;
            }
            else
            {
                printf("\nNAK for frame-%d received\n", bottom + j + 1);
                printf("Initiating Retransmisson");
                break;
            }
        }
        printf("\n");
    } while (sent < n);
    printf("%d Frame's successfully sent\n", sent);
    printf("Message recieved  at reciever :");
    for (i = 0; i < sent; i++)
    {
        printf("%d  ", frames[i]);
    }
    return (EXIT_SUCCESS);
}
```

Figure 3 C Program for Go Back N Protocol Continued

Selective Repeat

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
// K Srikanth 17ETCS002124
int main(int argc, char **argv)
{
    srand(time(0));
    int n, sender_window_size = 0;
    printf("Enter the Input Window Size : ");
    scanf("%d", &sender_window_size);
    printf("Enter the Number of Frames :");
    scanf("%d", &n);
    int frames[n], i, j;
    printf("Sender Message :  ");
    for (i = 0; i < n; i++)
        scanf("%d", &frames[i]);
    int last_ack, bottom, sent = 0, nak = 0, frame_no;
    do
    {
        if (nak)
        {
            printf("NAK %d retransmitted \n", nak);
            printf("Cumulative ACK %d received\n", sent);
            nak = 0;
        }
        printf("Frames Transmitted: ");

        for (i = sent; i < sent + sender_window_size && i < n; i++)
        {
            printf("%d  ", i + 1);
        }
        printf("\n");
        bottom = sent;
        for (j = 0; j < sender_window_size && sent < n; j++)
        {
            last_ack = sent + random() % 2;
            frame_no = bottom + j + 1;
```

Figure 4 C Program for Selective Repeat Protocol

```
            if (last_ack == sent + 1)
            {
                if (nak)
                    printf("Frame %d received, ACK-%d received   \n", frame_no, nak - 1);

                else
                    printf("Frame %d received, ACK-%d received   \n", frame_no, frame_no);

                sent++;
            }
            else
            {
                printf("Frame %d transmission failed,", frame_no);
                printf("NAK-%d received   \n", frame_no);
                printf("Initiating Retransmisson\n");
                nak = bottom + j + 1;
                sent++;
            }
        }
    } while (sent < n || nak);
    printf("%d frames successfully sent\n", sent);
    printf("Message recieved  at reciever :");
    for (i = 0; i < sent; i++)
    {
        printf("%d  ", frames[i]);
    }
    return (EXIT_SUCCESS);
}
```

Figure 5 C Program for Selective Repeat Protocol Continued

3. Results

Stop and Wait

```
❯ ./a.out
Enter the Number of Frames 2
Input the data to be transmitted
101000010
101111000
Frame 1 has been sent.
Timed out! Initiating re-transmission.
Frame 1 has been sent.
Timed out! Initiating re-transmission.
Frame 1 has been sent.
Timed out! Initiating re-transmission.
Frame 1 has been sent.
Frame 1 is successfully received.
Frame 2 has been sent.
Frame 2 is successfully received.
The Frames received are :101000010 101111000 %
```

Figure 6 C Program Output for Stop and Wait Protocol

Go Back N

```
❯ ./a.out
Enter the Input Window Size : 2
Enter the Number of Frames : 2
Sender Message :  10101010
10101010
Frames Transmitted: 1  2
Frames Acknowledged: 1
NAK for frame-2 received
Initiating Retransmisson
Frames Transmitted: 2
Frames Acknowledged: 2
2 Frame's successfully sent
Message recieved  at reciever :10101010  10101010
```

Figure 7 C Program Output for Go Back N Protocol

Selective Repeat

```
❯ ./a.out
Enter the Input Window Size : 2
Enter the Number of Frames :2
Sender Message :  10101010
10101101
Frames Transmitted: 1  2
Frame 1 received, ACK-1 received
Frame 2 transmission failed,NAK-2 received
Initiating Retransmisson
NAK 2 retransmitted
Cumulative ACK 2 received
Frames Transmitted:
2 frames successfully sent
Message recieved  at reciever :10101010  10101101
```

Figure 8 C Program Output for Selective Repeat Protocol

4. Analysis and Discussions

1. Difference between stop and wait, go back N and selective repeat.

The sliding window protocol is used to transmit frames and receive acknowledgement.

- In stop and wait protocol frames are sent one by one, and if negative or no acknowledgement is received for a frame then the frame is retransmitted.
- Go back N sends a set of frames at once and the receiver sends acknowledgement one by one, suppose any frame in sliding window receives negative or no acknowledgment, frames are re-transmitted from that particular frame. This protocol just sees the recent frame that received acknowledgement and slides the window above it.
- Selective repeat sends the frames in a collection and receiver sends acknowledgements for the frames, the frames for which negative acknowledgement are (NAK) received are stored in a buffer and transmitted again later. When the receiver receives that frame again, it replies with a collective acknowledgement called cumulative acknowledgement, acknowledging the same and all that followed.

2. Comparison of the disadvantages of the different ARQ mechanisms.

The main difference between three lies between sender's window size and receiver's window size.

	Sender's size	Receiver's size
Stop & wait	1	1
Go Back N	N	1
Selective repeat	N	M

Comparison of the disadvantages of the different ARQ mechanisms. ARQ mechanism of Stop & wait protocol introduces reasonable delay in transmission, as it waits for acknowledgement of each frame. So it is not a good idea for it to be implemented in case where there are large number of frames to be transmitted.

5. Conclusions

The above program successfully computes the Stop & wait protocol. In this method of flow control, the sender sends a single frame to receiver & waits for an acknowledgment. The main advantage of stop & wait protocols is its accuracy. Next frame is transmitted only when the first frame is acknowledged. So, there is no chance of frame being lost. However, its disadvantages outweigh its advantages.

6. Comments

a. Limitations of the experiment

As considered, program doesn't simulate the exact scenario, where sender and receiver are setup on two different systems. It is just a single case here, while in reality data is lost in different means. It can be no acknowledgement or negative acknowledgement cannot be illustrated properly.

b. Limitations of the results obtained

Program ran for 20 frames, output considerably lengthy. So just presented illustration for 7 frames.

c. Learning

Sliding window protocols, stop & wait protocol, go back N and selective repeat. its advantages and disadvantages.