

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: Anirudha P Kakrannaya

Register No:



1. Algorithm/Flowchart

Algorithm for Stop and Wait

- Step 1: Start
- Step 2: Take total number of frames from user
- Step 3: Send frames one by one
- Step 4: Ask for user whether respective frame is received or not
- Step 5: If not send negative ACK to sender and if it is then send ACK to sender
- Step 6: Stop

Algorithm for Go back N

- Step 1: Start
- Step 2: Take total number of frames from user
- Step 3: Take window size from user
- Step 4: Send frames in window, if frame is received then slide the window and send all frames in window
- Step 5: If frame is not received then send frames of window which comes after the frame is lost.
- Step 6: Stop

Algorithm for Selective Repeat

- Step 1: Start
- Step 2: Take total number of frames from user
- Step 3: Send all frames
- Step 4: Take number of frames received
- Step 5: Also take the which frames are received
- Step 6: Send only frames which are not received
- Step 7: Stop

2. Program



```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv) {
    int val;
    while(1){
        printf("\nEnter 1.Stop And Wait protocol\n"
               "2.Go back N\n3.Selective repeat\n");
        scanf("%d",&val);
        switch(val){
            case 1:StopAndWait();
            break;
            case 2:GoBackN();
            break;
            case 3:Selectiev();
            break;
            default:return(0);
        }
    }
    return (EXIT_SUCCESS);
}

```

Main function of program all ARQ functions are called from here.

```

void StopAndWait(){
    int n,i=0,frame;
    printf("Enter the total number of frames ");
    scanf("%d",&n);
    while(i!=n){
        printf("\nEnter received frame ");
        scanf("%d",&frame);
        if(frame==i+1){
            printf("Transmitting..... ACK to frame %d\n",frame);
            i++;
        }else{
            printf("Negative ACK.... to frame %d\n",i+1);
        }
    }
}

```

Function to implement stop and wait ARQ algorithm.

```

void GoBackN() {
    int n,i=0,frame,size,t;
    printf("Enter the total number of frames ");
    scanf("%d",&n);
    printf("Enter window size of frames ");
    scanf("%d",&size);
    printf("Sending frames ");
    for(int j=0;j<size;j++){
        printf("%d,",j+1);
    }
}

```



```

for(int j=0;j<n;j++){
    printf("\nIs Frame %d is received (1 or 0)",j+1);
    scanf("%d",&t);
    if(t==1){
        printf("\nSending ACK to frame %d\nSliding window",j+1);
        printf("\nIn window ");
        for(int k=j+1;k<j+1+size;k++){
            if(k<n)
                printf("%d ",k+1);
        }
    }else{
        printf("\nRetransmitting frames\t");
        for(int m=j;m<j+size;m++){
            if(m<n)
                printf(" %d",m+1);
        }
        j--;
    }
}
}

```

Function to implement Go back N algorithm.

```

void Selectiev(){
    int n,i=0,frame,m,size;
    printf("Enter the total number of frames ");
    scanf("%d",&n);
    printf("Sending frames ");
    for(int j=0;j<n;j++){
        printf("%d,",j+1);
    }
    printf("\nEnter the number of frames received\n");
    scanf("%d",&m);
    int a[n];
    printf("\nEnter received frame\n");
    for(int i=0;i<m;i++)
        scanf("%d",&a[i]);
    int trigger=0;
}

```



```

for(int i=0;i<n;i++){
    trigger=0;
    for(int j=0;j<m;j++){
        if((i+1==a[j])){
            trigger=1;
        }
    }
    if(trigger==0){
        // printf("Negative ACK to frame %d\n",i+1);
        printf("Retransmitting frame %d\n",i+1);
        a[m++]=i+1;
    }
}

```

```

for(int i=0;i<n;i++){
    for(int j=0;j<n;j++){
        if(a[i]<a[j]){
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
}
printf("\nSorting ");
for(int i=0;i<n;i++){
    printf("%d ",a[i]);
}

```

```

}

```

Function to implement Selective Repeat Algorithm.



3. Results

```
Enter 1.Stop And Wait protocol
2.Go back N
3.Selective repeat
1
Enter the total number of frames 5

Enter received frame 1
Transmitting..... ACK to frame 1

Enter received frame 3
Negative ACK.... to frame 2

Enter received frame 2
Transmitting..... ACK to frame 2

Enter received frame 3
Transmitting..... ACK to frame 3

Enter received frame 5
Negative ACK.... to frame 4

Enter received frame 4
Transmitting..... ACK to frame 4

Enter received frame 5
Transmitting..... ACK to frame 5
```

Output for simple Stop and Wait protocol



```
Enter 1.Stop And Wait protocol
2.Go back N
3.Selective repeat
2
Enter the total number of frames 5
Enter window size of frames 3
Sending frames 1,2,3,
Is Frame 1 is received (1 or 0)1

Sending ACK to frame 1
Sliding window
In window 2 3 4
Is Frame 2 is received (1 or 0)0

Retransmitting frames    2 3 4
Is Frame 2 is received (1 or 0)1

Sending ACK to frame 2
Sliding window
In window 3 4 5
Is Frame 3 is received (1 or 0)0

Retransmitting frames    3 4 5
Is Frame 3 is received (1 or 0)1

Sending ACK to frame 3
Sliding window
In window 4 5
Is Frame 4 is received (1 or 0)1

Sending ACK to frame 4
Sliding window
In window 5
Is Frame 5 is received (1 or 0)0

Retransmitting frames    5
Is Frame 5 is received (1 or 0)1

Sending ACK to frame 5
```

Output for Go back N protocol




```

Enter 1.Stop And Wait protocol
2.Go back N
3.Selective repeat
3
Enter the total number of frames 10
Sending frames 1,2,3,4,5,6,7,8,9,10,
Enter the number of frames received
4

Enter received frame
2
8
1
3
Retransmitting frame 4
Retransmitting frame 5
Retransmitting frame 6
Retransmitting frame 7
Retransmitting frame 9
Retransmitting frame 10
Buffer 2 8 1 3 4 5 6 7 9 10
Sorting 1 2 3 4 5 6 7 8 9 10

```

Output for Selective repeat protocol

4. Analysis and Discussions

Stop-and-wait protocol and Go Back N are the methods evolved for handling the flow control of the network data transfers. These methods mainly differentiated by the techniques they follow such as stop-and-wait uses the concept of the acknowledging each data unit before sending another data unit. Conversely, Go Back N allows the transition of the several data units before sending an acknowledgement.

Go Back N Protocol and Selective Repeat Protocol are the sliding window protocols. The sliding window protocol is primarily an error control protocol, i.e. it is a method of error detection and error correction. The basic difference between go-back-n protocol and selective repeat protocol is that the Go back N protocol" retransmits all the frames that lie after the frame which is damaged or lost. The "selective repeat protocol" retransmits only that frame which is damaged or lost.

Disadvantages of stop-and-wait:

Fairly slow: the sender can send at most one new packet per RTT

Not robust: if the ack can get lost, when the receiver gets a packet, the receiver cannot tell if it is a retransmission or a new packet to deal with the last problem, must distinguish new packets from retransmissions.



Go Back N has to retransmit all the packets from the start of its sliding number whenever a packet timeout/is lost/has error. So, when the window size and bandwidth product is large, and the channel is lossy, Go Back N will often retransmit a large number of packets. The channel gets stuffed with these retransmissions where many of the packets are redundant, thus leading to lower channel utilization compared to Stop and Wait.

That's why we have Selective Repeat, where we only retransmit the packets that haven't been acknowledged.

The selective repeat is a more efficient protocol as it does not waste bandwidth for the frames which are properly received but, its complexity and expense favors the use of the go-back-n protocol.

5. Conclusions

C program for ARQ mechanism is developed and verified with result.

6. Comments

a. Limitations of the experiment

Only simple logic is used may not work in complex cases.

b. Limitations of the results obtained

Results are based on user input for particular ARQ function.

c. Learning

Learnt to create simple program to implement ARQ

d. Recommendations

This program further modified like adding frame sequence number and Ack type.

