

Experiment No: - 9

Experiment Name: - Sliding Window Protocol (Selective Repeat)

Name: - Shrikrushna C Gundre

Roll No: - DCSE 12

PRN No: - 2122010191

❖ **Selective Repeat Protocol**

Why Selective Repeat Protocol?

The go-back-n protocol works well if errors are less, but if the line is poor it wastes a lot of bandwidth on retransmitted frames. An alternative strategy, the selective repeat protocol, is to allow the receiver to accept and buffer the frames following a damaged or lost one.

Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors):

- Receiver must be able to accept packets out of order.
- Since receiver must release packets to higher layer in order, the receiver must be able to buffer some packets.

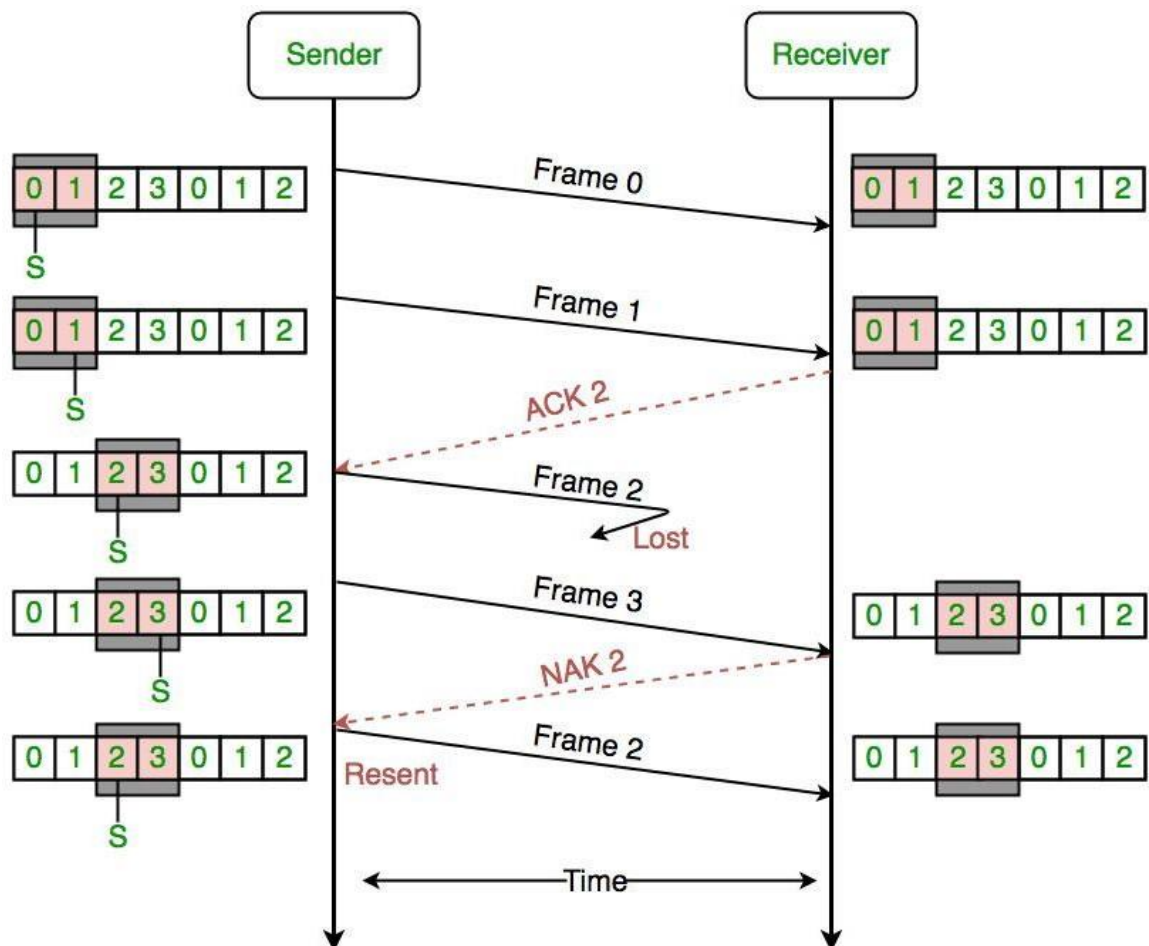
Retransmission requests:

- **Implicit** – The receiver acknowledges every good packet, packets that are not ACKed before a time-out are assumed lost or in error. Notice that this approach must be used to be sure that every packet is eventually received.
- **Explicit** – An explicit NAK (selective reject) can request retransmission of just one packet. This approach can expedite the retransmission but is not strictly needed.
- One or both approaches are used in practice.

Selective Repeat Protocol (SRP):

This protocol (SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintains a window of size. SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. SRP also requires full-duplex link. backward acknowledgments are also in progress.

- Sender's Windows (Ws) = Receiver's Windows (Wr).
- Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the size of the window is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.
- Sender can transmit new packets as long as their number is within W of all unACKed packets.
- Sender retransmits un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
- Receiver ACKs all correct packets.
- Receiver stores correct packets until they can be delivered in order to the higher layer.
- In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m .



➤ **Code**

```
#include<stdio.h>

int frames[12] = {1,2,3,4,5,6,7,8,9,10,11,12};
int nack[12] = {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};

void sendFrames(int start, int end);

void retransmit(int start);

void getFrameNo(int no, int windowSize);

int main(){

    int windowSize = 0, ack, start = 0, win = 0;

    do{

        printf("\n Enter window size :");

        scanf("%d",&windowSize);

        if(windowSize < 2)

            printf("\n Enter window size greater than 1");

    }while(windowSize < 2);

    win += windowSize;

    sendFrames(start,win);

    while(win < 12){

        printf("\n Enter no of frames which you have received negative ack :");

        scanf("%d",&ack);

        if(ack == 0){

            start += (win-start);

            win += windowSize ;

            sendFrames(start,win);

        }else{

            printf("\n Retransmitting the frame..");
```

```

        getFrameNo(ack>windowSize);
        for(int i=windowSize;i>=0;i--){
            if(nack[i] != -1){
                retransmit(nack[i]-1);
            }
        }
    }
}

printf("\n All frames transmitted successfully...!");
return 0;
}

void retransmit(int index){
    printf("\n Frame %d is retransmitted..",frames[index]);
}

void sendFrames(int start, int end){
    for(int i=start;i<end;i++){
        printf("\n Frame %d is transmitted..",frames[i]);
    }
}

void getFrameNo(int no, int windowSize){
    int i=0;
    while(no > 0 && i < windowSize){
        int res = no%10;
        nack[i] = res;
        i++;
        no /= 10;
    }
}

```

}

➤ Output

```
Enter window size :4

Frame 1 is transmitted..
Frame 2 is transmitted..
Frame 3 is transmitted..
Frame 4 is transmitted..
Enter no of frames which you have received negative ack :23

Retransmitting the frame..
Frame 2 is retransmitted..
Frame 3 is retransmitted..
Enter no of frames which you have received negative ack :0

Frame 5 is transmitted..
Frame 6 is transmitted..
Frame 7 is transmitted..
Frame 8 is transmitted..
Enter no of frames which you have received negative ack :78

Retransmitting the frame..
Frame 7 is retransmitted..
Frame 8 is retransmitted..
Enter no of frames which you have received negative ack :0

Frame 9 is transmitted..
Frame 10 is transmitted..
Frame 11 is transmitted..
Frame 12 is transmitted..
All frames transmitted successfully...!
```