

## Selective Repeat Sliding Window Protocol

Selective repeat protocol, also called Selective Repeat ARQ (Automatic Repeat reQuest), is a data link layer protocol that uses sliding window method for reliable delivery of data frames. Here, only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

It uses two windows of equal size: a sending window that stores the frames to be sent and a receiving window that stores the frames received by the receiver. The size is half the maximum sequence number of the frame. For example, if the sequence number is from 0 – 15, the window size will be 8.

### Working Principle

Selective Repeat protocol provides for sending multiple frames depending upon the availability of frames in the sending window, even if it does not receive acknowledgement for any frame in the interim. The maximum number of frames that can be sent depends upon the size of the sending window.

The receiver records the sequence number of the earliest incorrect or un-received frame. It then fills the receiving window with the subsequent frames that it has received. It sends the sequence number of the missing frame along with every acknowledgement frame.

The sender continues to send frames that are in its sending window. Once, it has sent all the frames in the window, it retransmits the frame whose sequence number is given by the acknowledgements. It then continues sending the other frames.

### Retransmission requests:

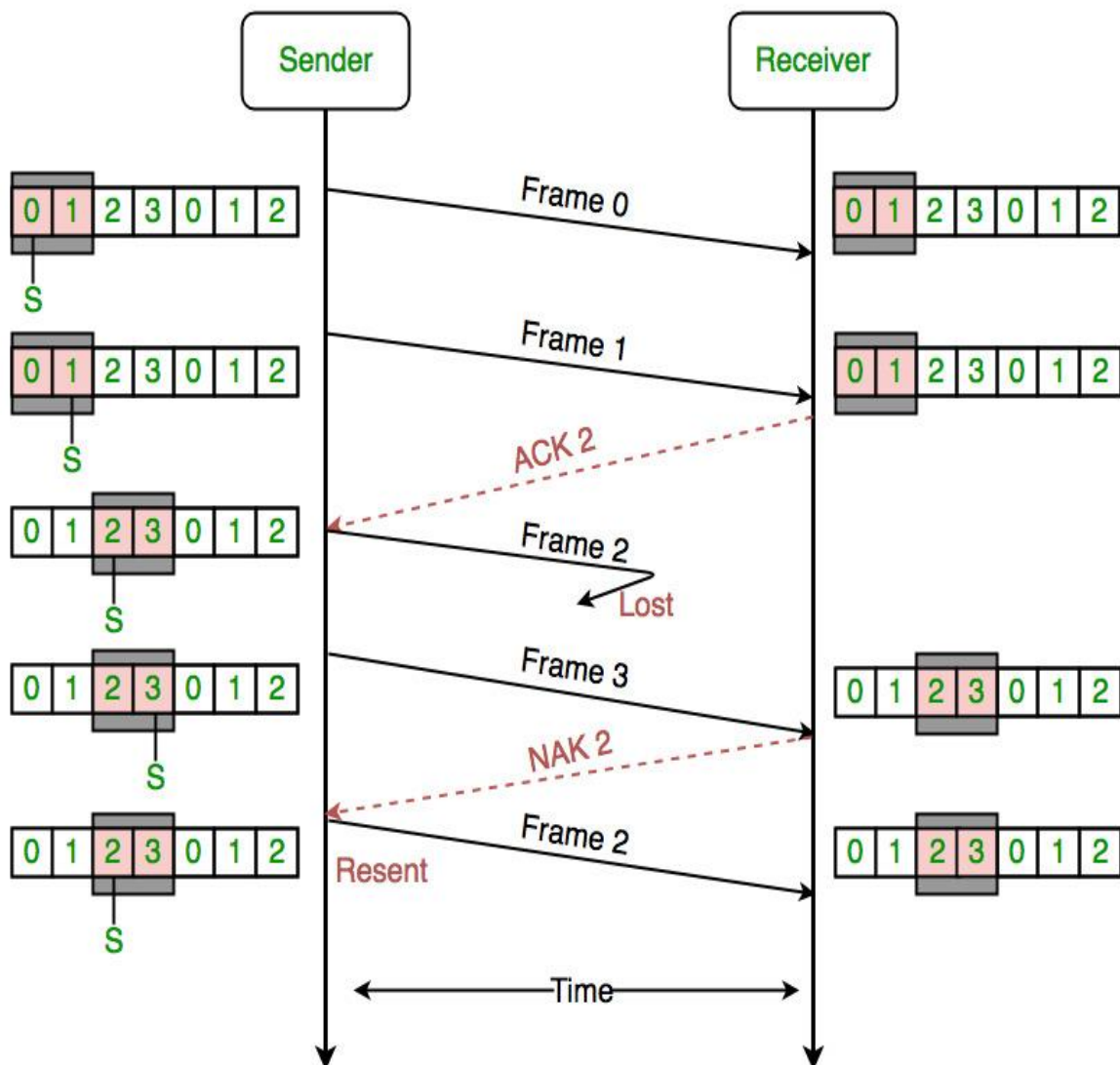
- **Implicit** – The receiver acknowledges every good packet, packets that are not ACKed before a time-out are assumed lost or in error. Notice that this approach must be used to be sure that every packet is eventually received.
- **Explicit** – An explicit NAK (selective reject) can request retransmission 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 maintain 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 acknowledgements are also in progress.

- Sender's Windows (  $W_s$  ) = Receiver's Windows (  $W_r$  ).

- 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 window size 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  $2^m$ .



**Figure** – the sender only retransmits frames, for which a NAK is received

Efficiency of Selective Repeat Protocol (SRP) is same as GO-Back-N's efficiency:

Efficiency =  $N/(1+2a)$

Where  $a$  = Propagation delay / Transmission delay

Buffers =  $N + N$

Sequence number =  $N$  (sender side) +  $N$  (Receiver Side)

## Sender Site Algorithm of Selective Repeat Protocol

```
begin
    frame s; //s denotes frame to be sent
    frame t; //t is temporary frame
    S_window = power(2,m-1); //Assign maximum window size
    SeqFirst = 0; // Sequence number of first frame in window
    SeqN = 0; // Sequence number of Nth frame window
    while (true) //check repeatedly
        do
            Wait_For_Event(); //wait for availability of packet
            if ( Event(Request_For_Transfer)) then
                //check if window is full
                if (SeqN-SeqFirst >= S_window) then
                    doNothing();
                end if;
                Get_Data_From_Network_Layer();
                s = Make_Frame();
                s.seq = SeqN;
                Store_Copy_Frame(s);
                Send_Frame(s);
                Start_Timer(s);
            end if;
        end while;
end begin
```

```

        SeqN = SeqN + 1;
    end if;

    if ( Event(Frame_Arrival) ) then
        r = Receive_Acknowledgement();

        //Resend frame whose sequence number is with ACK
        if ( r.type = NAK) then

            if ( NAK_No > SeqFirst && NAK_No < SeqN ) then

                Retransmit( s.seq(NAK_No));

                Start_Timer(s);

            end if

            //Remove frames from sending window with
positive ACK

            else if ( r.type = ACK ) then

                Remove_Frame(s.seq(SeqFirst));

                Stop_Timer(s);

                SeqFirst = SeqFirst + 1;

            end if

        end if

        // Resend frame if acknowledgement haven't been received
        if ( Event(Time_Out)) then

            Start_Timer(s);

            Retransmit_Frame(s);

        end if

    end
end

```

## Receiver Site Algorithm of Selective Repeat Protocol

```

Begin

    frame f;

    RSeqNo = 0; // Initialise sequence number of expected frame

    NAKsent = false;

```

```

ACK = false;

For each slot in receive_window
Mark(slot)=false;
while (true) //check repeatedly
do
    Wait_For_Event(); //wait for arrival of frame
    if ( Event(Frame_Arrival) then
        Receive_Frame_From_Physical_Layer();
        if ( Corrupted ( f.SeqNo ) AND NAKsent = false) then
            SendNAK(f.SeqNo);
            NAKsent = true;
        end if
        if ( f.SeqNo != RSeqNo AND NAKsent = false ) then
            SendNAK(f.SeqNo);
            NAKsent = true;
            if ( f.SeqNo is in receive_window ) then
                if ( Mark(RSeqNo) = false ) then
                    Store_frame(f.SeqNo);
                    Mark(RSeqNo) = true;
                end if
            end if
        else
            while ( Mark(RSeqNo))
                Extract_Data(RSeqNo);
                Deliver_Data_To_Network_Layer();
                RSeqNo = RSeqNo + 1;
                Send_ACK(RSeqNo);
            end while
        end if
    end if
end if

```

```
        end if  
    end while  
end
```