

Experiment No: - 8

Experiment Name: - Sliding Window Protocol (Go Back N)

Name: - Shrikrushna C Gundre

Roll No: - DCSE 12

PRN No: - 2122010191

❖ Go Back N Protocol

The three main characteristic features of GBN are:

1. Sender Window Size (WS)

It is N itself. If we say the protocol is GB10, then $W_s = 10$. N should be always greater than 1 in order to implement pipelining. For $N = 1$, it reduces to Stop and Wait protocol.

$$\text{Efficiency Of GBN} = N/(1+2a)$$

$$\text{where } a = T_p/T_t$$

If B is the bandwidth of the channel, then

Effective Bandwidth or Throughput

$$= \text{Efficiency} * \text{Bandwidth}$$

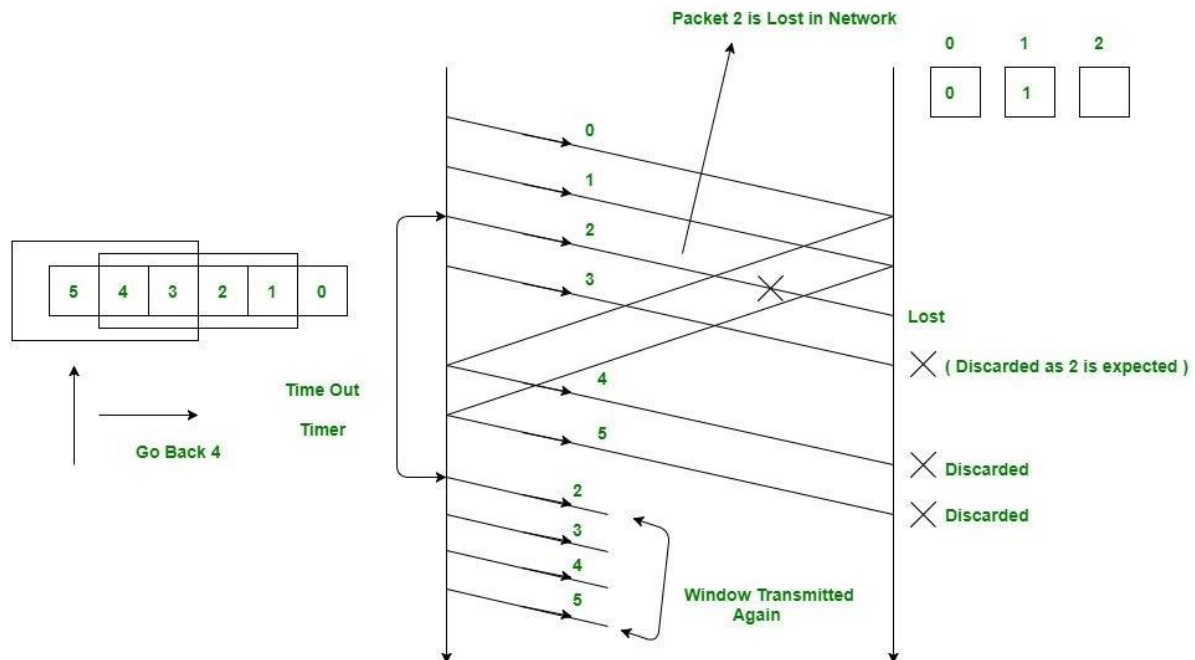
$$= (N/(1+2a)) * B$$

2. Receiver Window Size (WR)

WR is always 1 in GBN.

Now what exactly happens in GBN, we will explain with a help of example. Consider the diagram given below. We have sender window size of 4. Assume that we have lots of sequence numbers just for the sake of explanation. Now the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, receiver is now expecting packet 2 and sender window has also slid to further transmit the packets 4 and 5. Now suppose the packet 2 is lost in the network, Receiver will discard all the packets which sender has transmitted after packet 2 as it is expecting sequence number of 2. On the sender side for every packet send there is a time out timer which will expire for packet number 2. Now from the last transmitted packet 5 sender will go back to the packet number 2 in the current window and transmit all the packets till packet number 5. That's why it is called Go Back N. Go back means sender has to go back

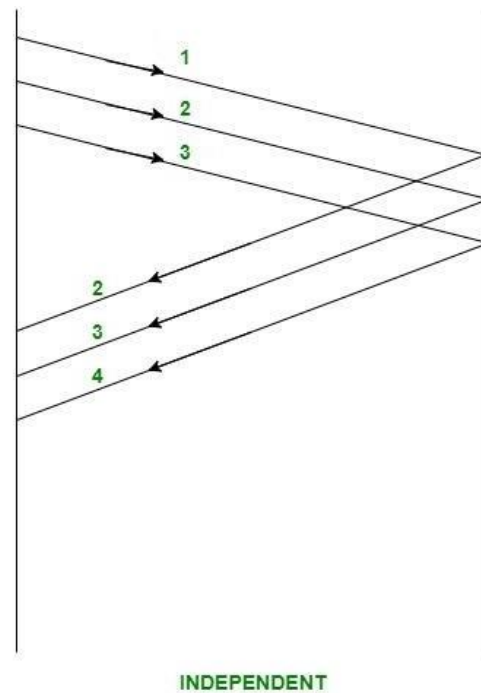
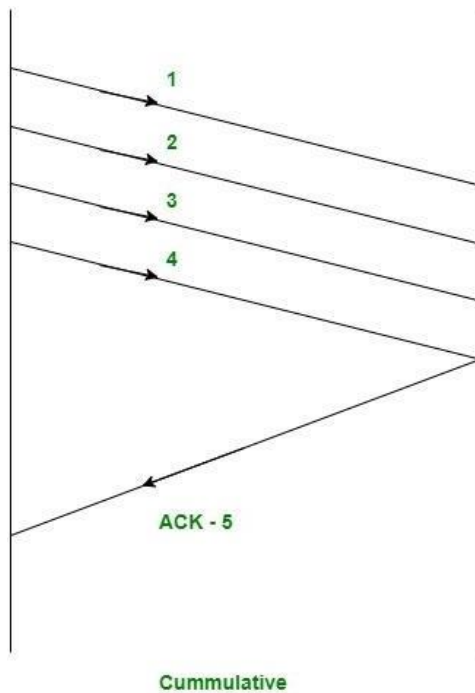
N places from the last transmitted packet in the unacknowledged window and not from the point where the packet is lost.



3. Acknowledgements

There are 2 kinds of acknowledgements namely:

- Cumulative Ack: One acknowledgement is used for many packets. The main advantage is traffic is less. A disadvantage is less reliability as if one ack is the loss that would mean that all the packets sent are lost.
- Independent Ack: If every packet is going to get acknowledgement independently. Reliability is high here but a disadvantage is that traffic is also high since for every packet we are receiving independent ack.



GBN uses Cumulative Acknowledgement.

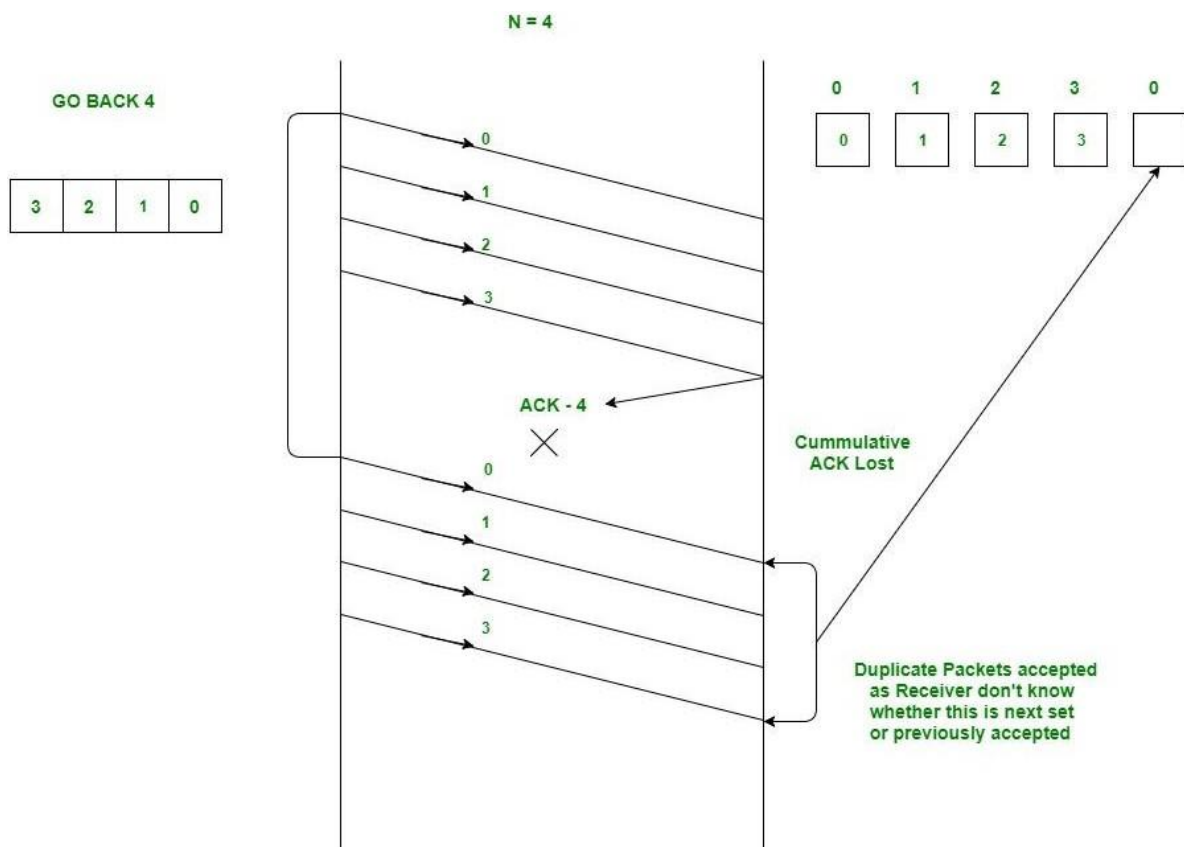
At the receiver side, it starts a acknowledgement timer whenever receiver receives any packet which is fixed and when it expires, it is going to send a cumulative Ack for the number of packets received in that interval of timer. If receiver has received N packets, then the Acknowledgement number will be $N+1$. Important point is Acknowledgement timer will not start after the expiry of first timer but after receiver has received a packet.

Time out timer at the sender side should be greater than Acknowledgement timer.

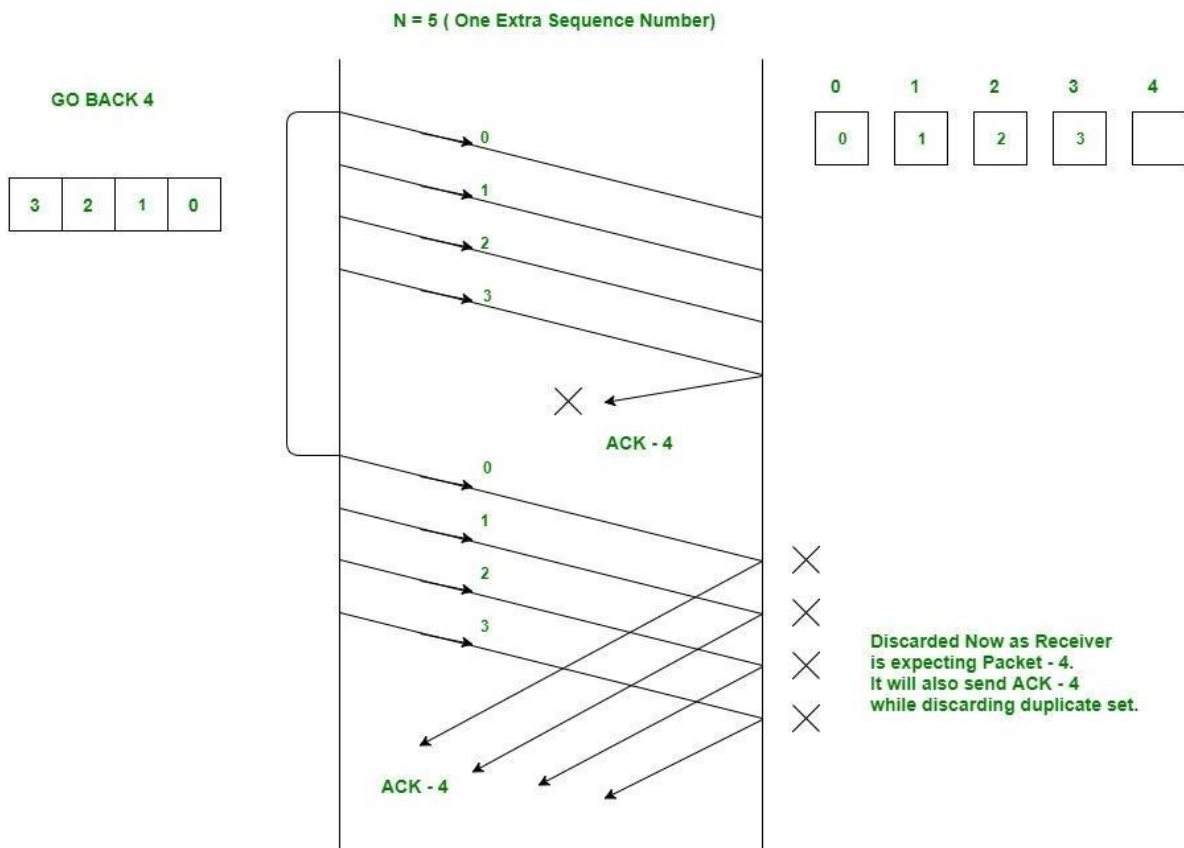
Example: Consider an example of GB4.

- Sender window size is 4 therefore we require a minimum of 4 sequence numbers to label each packet in the window.
- Now suppose receiver has received all the packets (0, 1, 2 and 3 sent by sender) and hence is now waiting for packet number 0 again (We can not use 4 here as we have only 4 sequence numbers available since $N = 4$).
- Now suppose the cumulative ack for the above 4 packets is lost in the network.
- On sender side, there will be timeout for packet 0 and hence all the 4 packets will be retransmitted again.
- Problem now is receiver is waiting for new set of packets which should have started from 0 but now it will receive the duplicate copies of the previously accepted packets.
- In order to avoid this, we need one extra sequence number.
- Now the receiver could easily reject all the duplicate packets which were starting from 0 because now it will be waiting for packet number 4 (We have added an extra sequence number now).

Trying with Sequence numbers 4.



Now Trying with one extra Sequence Number.



➤ Code

```
#include<stdio.h>

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

int main(){

    int windowSize=0, ack, start=0;

    do{

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

        scanf("%d",&windowSize);

        if(windowSize < 2)

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

    }while(windowSize < 2);

    send(0,windowSize);
```

```

while(windowSize < 10){

    printf("\n Enter 1 if you got the 1st frame otherwise enter -1 :");

    scanf("%d",&ack);

    if(ack == -1){

        printf("\n Retransmitting the frames..");

        send(start,windowSize);

    }else{

        start++;

        windowSize++;

        send(start,windowSize);

    }

}

while(start < 10){

    printf("\n Enter 1 if you got the 1st frame otherwise enter -1 :");

    scanf("%d",&ack);

    if(ack == -1){

        printf("\n Retransmitting the frames..");

        sendSigle(start);

    }else{

        start++;

        sendSigle(start);

    }

}

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

return 0;

}

void sendSigle(int start){

```

```

for(int i=start;i<10;i++){

    printf("\n Frame %d is transmitted..",frames[i]);

}

}

void send(int start, int end){

    for(int i=start;i<end;i++){

        printf("\n Frame %d is transmitted..",frames[i]);

    }

}

```

➤ Output

```

Enter window size :4

Frame 1 is transmitted..
Frame 2 is transmitted..
Frame 3 is transmitted..
Frame 4 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 2 is transmitted..
Frame 3 is transmitted..
Frame 4 is transmitted..
Frame 5 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 3 is transmitted..
Frame 4 is transmitted..
Frame 5 is transmitted..
Frame 6 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :-1

Retransmitting the frames..
Frame 3 is transmitted..
Frame 4 is transmitted..
Frame 5 is transmitted..
Frame 6 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 4 is transmitted..
Frame 5 is transmitted..
Frame 6 is transmitted..
Frame 7 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 5 is transmitted..
Frame 6 is transmitted..
Frame 7 is transmitted..
Frame 8 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :-1

Retransmitting the frames..
Frame 5 is transmitted..
Frame 6 is transmitted..
Frame 7 is transmitted..
Frame 8 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 6 is transmitted..
Frame 7 is transmitted..
Frame 8 is transmitted..
Frame 9 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

```

```

Frame 7 is transmitted..
Frame 8 is transmitted..
Frame 9 is transmitted..
Frame 10 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 8 is transmitted..
Frame 9 is transmitted..
Frame 10 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 9 is transmitted..
Frame 10 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :-1

Retransmitting the frames..
Frame 9 is transmitted..
Frame 10 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

Frame 10 is transmitted..
Enter 1 if you got the 1st frame otherwise enter -1 :1

All frames transmitted successfully...!

```