

6. Write a Program to implement Sliding window protocol for Goback N.

WHAT IS GO BACK-N

The Go Back-N ARQ or Go Back Automatic Repeat Request is a way to implement sliding window protocol. This protocol is used for flow control in networking and is a part of the data-link layer. The sender's window is of size N and the receiver's window size is always one.

The window sizes for Go Back-N are:

1. SENDER WINDOW SIZE (WS)

The WS is equal to N. If we say the protocol is GB-4, then $WS = 4$. In order to implement pipelining, the window size should always be greater than one. If window size is one, the protocol reduces to stop-and-wait protocol.

2. RECEIVER WINDOW SIZE (WR)

WR is always one in Go Back-N.

Now we will understand how Go Back-N actually works with the help of an example. In the diagram below the sender's window has a size of four, or Go Back-4. Assume that we've plenty of sequence numbers, for the sake of explanation. Now the sender has dispatched the packets zero, one, two and three. After acknowledging the packets zero and one, the receiver is now awaiting packet two and the sender window has additionally slid to similarly transmit the packets four and five.

Now suppose packet two is misplaced within the network. The receiver will discard all the packets which the sender has transmitted after packet two as it's awaiting the packet with sequence number two. On the sender side, for every packet dispatched there is a time-out timer to expire for packet range two. Now from the remaining transmitted packet five the sender will go back to the packet with sequence number two within the current window and transmit all the packets until packet number five. That's why it is referred to as Go Back-N. In the Go Back-N approach the sender has to move back N places from the closing transmitted packet within the unacknowledged window and not from the factor where the packet is misplaced.

It is based on the Sliding Window Technique.

Term Description Sliding A fixed-size buffer that allows sender to send multiple frames without waiting for ACK for each one. **Window Size (N)** Maximum number of frames that can be sent without acknowledgment. **Sequence Numbers** Frames are numbered (e.g., 0 to 7 if 3-bit numbering is used) to keep track. **Timeout** If ACK is not received in time, sender retransmits all frames from the last unacknowledged one.

ALGORITHM:

Step 1: Start the process

Step 2: Declare all necessary variables.

Step 3: Define the window Size by getting input from the user.

Step 4: Sender transmits all frames present in the window.

Step 5: When acknowledgment is not received for a frame, then the sender retransmits the frames from the last transmitted frame.

Step 6: After all frames are sent successfully stop the process

Program:

```
#include<stdio.h>

int main()
{
    int windowSize,sent=0,ack,i;
    printf("enter window size\n");
    scanf("%d",&windowSize);
    while(1)
    {
        for( i = 0; i < windowSize; i++)
        {
            printf("Frame %d has been transmitted.\n",sent);
            sent++;
            if(sent == windowSize)
                break;
        }
        printf("\nPlease enter the last Acknowledgement received .\n");
        scanf("%d",&ack);

        if(ack >= windowSize)
            break;
        else
            sent = ack+1;
    }
}
```

```
}  
return 0;  
}
```

Output: