

Experiment-5**Write a Program to implement Sliding window protocol for Go-back N**

Aim: To implement Sliding window protocol for “Go-back N”

Description:**Sliding window protocol:**

- Sliding window protocols are data link layer protocols for reliable and sequential delivery of data frames.
- The sliding window is also used in Transmission Control Protocol. In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.

Types Of Sliding Window Protocol:

1. One Bit Sliding Window Protocol
2. Go-Back N-ARQ (Automatic Repeat Request)
3. Selective Repeat Protocol

1. One-Bit Sliding Window Protocol

- In one – bit sliding window protocol, the size of the window is 1. So, the sender transmits a frame, waits for its acknowledgment, then transmits the next frame. Thus, it uses the concept of stop and waits for the protocol.

2.Go-Back N-ARQ (Automatic Repeat Request)

- In Go-Back-N ARQ, N is the sender's window size. Suppose we say that Go-Back-3, which means that the three frames can be sent at a time before expecting the acknowledgment from the receiver.
- It uses the “principle of protocol pipelining” in which the multiple frames can be sent before receiving the acknowledgment of the first frame.
- The number of frames that can be sent at a time totally depends on the size of the sender's window. So, we can say that 'N' is the number of frames that can be sent at a time before receiving the acknowledgment from the receiver.
- If the acknowledgment of a frame is not received within an agreed-upon time period, then all the frames available in the current window will be retransmitted.



- In Go-Back-N, N determines the sender's window size, and the size of the receiver's window is always 1.

Program:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <math.h>
```

```
int n, r;
```

```
struct frame
```

```
{
```

```
    char ack;
```

```
    int data;
```

```
}
```

```
frm[10];
```

```
int sender(void);
```

```
void recvack(void);
```

```
void resend_gb(void);
```

```
int main()
```

```
{
```

```
    int c;
```

```
    sender();
```

```
    recvack();
```

```
    resend_gb();
```

```
    printf("\n All frames are sent succesfuuly\n");
```

```
}
```

```
int sender()
```



Exp No:

Date:

Page No:

```
{  
  
int i;  
  
printf("\n Enter No.Of Frames to be send:");  
  
scanf("%d", &n);  
  
for (i = 1; i <= n; i++)  
{  
  
printf("\n Enter Data Into The Frames[%d]", i);  
  
scanf("%d", &frm[i].data);  
  
frm[i].ack = 'y';  
  
}  
  
return 0;  
  
}  
  
void recvack()  
  
{  
  
int i;  
  
r = rand() % n;  
  
frm[r].ack = 'n';  
  
for (i = 1; i <= n; i++)  
{  
  
if (frm[i].ack == 'n')  
  
printf("\n The Frame Number %d Is Not Received\n", r);  
  
}  
  
}  
  
void resend_gb()  
  
{
```



Exp No:

Date:

Page No:

```
int i;

printf("\n Resending The Frame %d", r);

for (i = 1; i <= n; i++)

{

    sleep(2);frm[i].ack = 'y';

    printf("\n The Frame Received is %d", frm[i].data);

}

}
```

Output:

```
Enter No.Of Frames to be send:4

Enter Data Into The Frames[1]1

Enter Data Into The Frames[2]2

Enter Data Into The Frames[3]3

Enter Data Into The Frames[4]4

The Frame Number 1 Is Not Received

Resending The Frame 1
The Frame Received is 1
The Frame Received is 2
The Frame Received is 3
The Frame Received is 4
All frames are sent succesfuuly
```