

Experiment - 3

AIM- To implement sliding window protocol (Go back N).

Description -

The Data link layer is used for error and flow control in computer networking. Sliding window protocol is a data link layer protocol that helps to implement both error control and flow control.

The key to Go-back-N is that we can send several packets before receiving acknowledgments, but the receiver can only buffer one packet. We keep a copy of the sent packets until the acknowledgments arrive.

Go-back-N 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 acknowledgement 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.

The receive window makes sure that the correct data packets are received and that the correct acknowledgments are sent. In Go-Back-N, the size of the receive window is always 1. The receiver is always looking for the arrival of a specific packet. Any packet arriving out of order is discarded and needs to be resent.

Algorithms:

1. Input number of frames and window size.
2. Initialize current frame = 0
3. do while current frame < number of frames:
 - a) Start with sender side.
 - b) Send frames from current frame to $\min(N, \text{current frame} + \text{window size}) - 1$
 - c) Move to receiver side.
 - d) Input one by one which sent frames were successfully received. Do not accept frames that are out of order.
 - e) Send Acknowledgements for frames that were received successfully.
 - f) Move to sender side.

g) Check which acknowledgements were received successfully and update the variable current frame accordingly.

Code:

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int n;
    cout<<"Enter the number of frames: ";
    cin>>n;
    int wSize;
    cout<<"Enter the window size: ";
    cin>>wSize;
    cout<<endl;

    int currFrame = 0;
    while(currFrame != n){
        cout<<"Sender Side: "<<endl;
        for(int i=currFrame; i<min(n, currFrame+wSize); i++){
            cout<<"Sending Frame "<<i<<"..."<<endl;
        }
        cout<<endl;

        cout<<"Receiver Side: "<<endl;
        vector<int> ACKs;
        for(int i=currFrame; i<min(n, currFrame+wSize); i++){
            char ch;
            cout<<"Did you receive Frame "<<i<<"? (y or n) ";
            cin>>ch;
            if(ch == 'n'){
                break;
            }else{
                cout<<"Frame Received"<<endl;
                cout<<"Sending ACK for frame "
                    <<i<<"..."<<endl;
                ACKs.push_back(i);
            }
        }
        cout<<endl;

        cout<<"Sender Side:"<<endl;
```

```

int maxACK = currFrame - 1;
for (int i=0; i<ACKs.size(); i++){
    char ch;
    cout<<"Did you receive the ACK for frame "
        <<ACKs[i]<<"? ";
    cin>>ch;
    if(ch == 'y'){
        maxACK = max(maxACK, ACKs[i]);
        cout<<"Successful Transmission for all frames till "
            << ACKs[i] << endl;
    }
}
currFrame = maxACK + 1;
cout << "-----" << endl;
}
return 0;
}

```

Output:

```

File Edit View Search Terminal Help
prince@pp-asus:~/lab/CN/3.GoBackN$ g++ goBackN.cpp
prince@pp-asus:~/lab/CN/3.GoBackN$ ./a.out
Enter the number of frames: 6
Enter the window size: 3

Sender Side:
Sending Frame 0...
Sending Frame 1...
Sending Frame 2...

Receiver Side:
Did you receive Frame 0? (y or n) y
Frame Received
Sending ACK for frame 0...
Did you receive Frame 1? (y or n) y
Frame Received
Sending ACK for frame 1...
Did you receive Frame 2? (y or n) n

Sender Side:
Did you receive the ACK for frame 0? n
Did you receive the ACK for frame 1? y
Successful Transmission for all frames till 1
-----

```

File Edit View Search Terminal Help

Sender Side:

Sending Frame 2...

Sending Frame 3...

Sending Frame 4...

Receiver Side:

Did you receive Frame 2? (y or n) y

Frame Received

Sending ACK for frame 2...

Did you receive Frame 3? (y or n) y

Frame Received

Sending ACK for frame 3...

Did you receive Frame 4? (y or n) y

Frame Received

Sending ACK for frame 4...

Sender Side:

Did you receive the ACK for frame 2? n

Did you receive the ACK for frame 3? n

Did you receive the ACK for frame 4? y

Successful Transmission for all frames till 4

Sender Side:

Sending Frame 5...

Receiver Side:

Did you receive Frame 5? (y or n) y

Frame Received

Sending ACK for frame 5...

Sender Side:

Did you receive the ACK for frame 5? y

Successful Transmission for all frames till 5

prince@pp-asus:~/lab/CN/3.GoBackN\$

Learnings -

1. Retransmits all the frames that sent after the frame which suspects to be damaged or lost.
2. If error rate is high, it wastes a lot of bandwidth.
3. No searching of frame is required neither on sender side nor on receiver. Maximum window size allowed is $N - 1$.
4. The Go-Back-N protocol simplifies the process at the receiver. The receiver keeps track of only one variable, and there is no need to buffer out-of-order packets; they are simply discarded.