1. Go Back to N

/\*

    Name: Sudhansu Sekhar Swain

    Roll: 118CS0689

\*/

#include<bits/stdc++.h>

using namespace std;

bool isAccepting(){    //this function returns random acceptance for a frame

    return rand()%2;

}

void go\_back\_n(){

    queue<int>packets;              //window used as sliding window

    vector<int>senderWindowBuffer;  //buffer to store temporary window frame

    int windowSize,sequenceNumber,currentPacket;

    cout<<"Enter size of the sliding window"<<endl;

    cin>>windowSize;

    sequenceNumber = windowSize + 1;    //this is used to avoid duplicate frame value sent from the sender

    currentPacket =  0;

    bool accepted[sequenceNumber] = {0};    //stores the acceptance value

    bool sendPackets = true;

    do{

        for(int i = 0;i < windowSize;i++){  //sending the window of frames atonce

            int packetSequenceNumber = (currentPacket%sequenceNumber);

            senderWindowBuffer.push\_back(packetSequenceNumber);

            currentPacket++;

        }

        bool ackReceived = false;

        while(!ackReceived){

            while(!packets.empty())

                packets.pop();

            for(int i:senderWindowBuffer)

                packets.push(i);

            while(!packets.empty()){

                int frontPacket = packets.front();

                if(( !accepted[frontPacket] && !isAccepting())){    //if the last frame is not accepted

                    cout<<frontPacket<<" is not accepted"<<endl;

                    break;

                }

                if(!accepted[frontPacket])                          //if this is accepted for the first time

                    cout<<frontPacket<<" is accepted"<<endl;

                accepted[frontPacket] = true;                       //value changed as a result of acceptance

                packets.pop();

            }

            ackReceived = packets.empty();

            if(ackReceived)

                cout<<"Cumulative Ack is received at sender side!"<<endl;

        }

        memset(accepted,0,sizeof(accepted));                        //accepted array is cleared for status of next window of frames

        senderWindowBuffer.clear();                                 //buffer cleared for next set of frames

        cout<<"Want to send more packets? Press 0 for No"<<endl;    //dynamism is achieved as more frames can be sent in the runtime

        cin>>sendPackets;

    }while(sendPackets);

}

int main(){

    go\_back\_n();

    return 0;

}

1. Selective Repeat

/\*

    Name: Sudhansu Sekhar Swain

    Roll: 118CS0689

\*/

#include<bits/stdc++.h>

using namespace std;

bool isAccepting(){

    return rand()%2;

}

int main(){

    int windowSize,sequenceNumber,currentPacket;

    cout<<"Enter the windowSize"<<endl;

    cin>>windowSize;

    sequenceNumber = windowSize + 1;                    // To avoid duplication of packets

    currentPacket =  0;

    int senderWindow[windowSize];

    bool nAck[sequenceNumber] = {0};                         // Status of the neg. Acks of each frames

    bool sendPackets = true;                            // Whether sender wants to send more packets

    bool accepted[sequenceNumber] = {0};                    // Accepted record of packets

    vector<int>receiverWindow;

    do{

        for(int i = 0;i < windowSize;i++){

            senderWindow[i] = (currentPacket%sequenceNumber);

            cout<<"Sender has sent "<<senderWindow[i]<<" packet"<<endl;

            currentPacket++;

        }

        cout<<"------All packets sent------"<<endl;

        bool nAckFound = true;

        while(nAckFound){

            for(int i=0;i<windowSize;i++){

                int packetSequenceNumber = senderWindow[i];

                if((!accepted[packetSequenceNumber] and isAccepting()) || (nAck[packetSequenceNumber] and isAccepting())){

                    receiverWindow.push\_back(packetSequenceNumber);

                    cout<<packetSequenceNumber<<" has been accepted"<<endl;

                    nAck[packetSequenceNumber] = false;

                    accepted[packetSequenceNumber] = true;

                }

                else if(!accepted[packetSequenceNumber]){

                    nAck[packetSequenceNumber] = true;

                    cout<<"Negative Acknowledgement Sent for "<<packetSequenceNumber<<endl;

                }

            }

            bool anyNAckExists = false;

            for(int i=0;i<windowSize;i++){

                int packetNumber = senderWindow[i];

                if(nAck[packetNumber])

                    anyNAckExists = true;

            }

            nAckFound = anyNAckExists;

        }

        memset(nAck,0,sizeof(nAck));

        memset(accepted,0,sizeof(accepted));

        receiverWindow.clear();

        cout<<"Do you want to send more packets ? Press 0 for No or Any number for Yes: "<<endl;

        cin>>sendPackets;

    }while(sendPackets);

}