Practical 4 – Computer Networks Lab

Name: Neeraj Belsare

Roll No.: 79

Batch: A4

PRN: 202101040133

Title:

Sliding Window Mechanism

Aim:

Write a program to implement sliding window mechanisms using

1. Stop and Wait ARQ
2. Go Back N ARQ
3. Selective Repeat ARQ

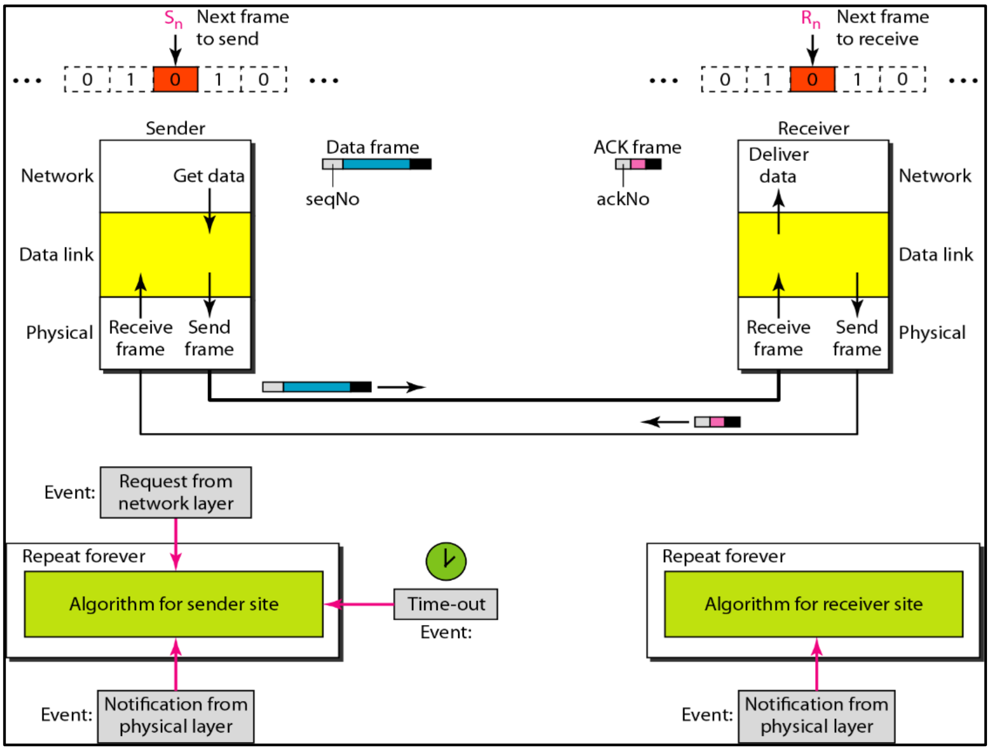
Theory:

1. **Stop-and-Wait ARQ**

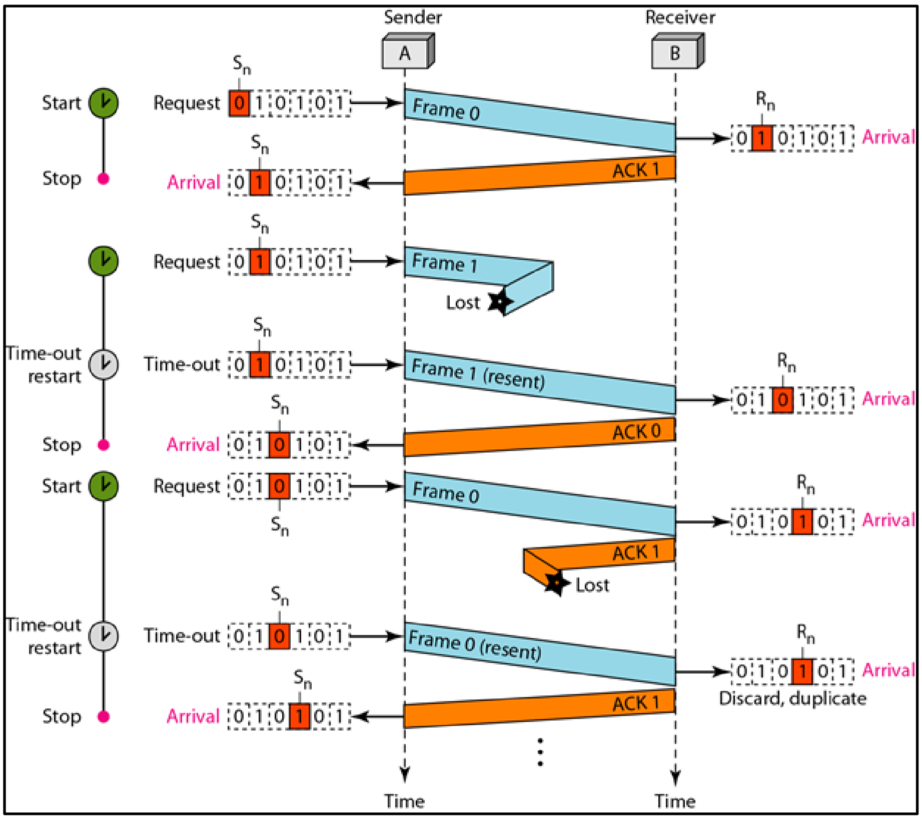
   Stop-and-Wait ARQ (Automatic Repeat reQuest) is a simple error control protocol used for reliable data transmission over a communication channel. In Stop-and-Wait ARQ, the sender sends a single data frame to the receiver and waits for an acknowledgement (ACK) from the receiver. If the sender receives the ACK, it knows that the frame has been successfully received, and it can then send the next frame. If the sender does not receive the ACK within a specified timeout indicating an error, it retransmits the same frame.

   Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting the frame when the timer expires.

Design of the Stop-and-Wait ARQ Protocol



Flow Diagram of the Stop-and-Wait ARQ Protocol



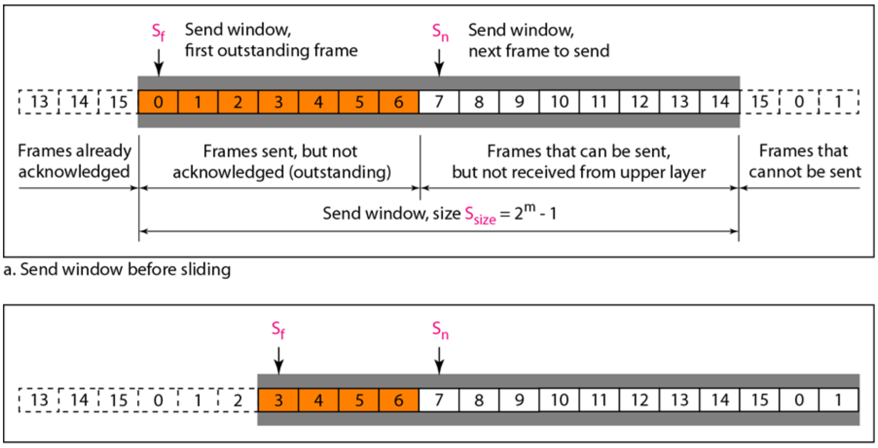
2. Go Back N ARQ

Go-Back-N is an Automatic Repeat reQuest (ARQ) protocol for reliable data communication. It ensures data packets are transmitted and received correctly over an unreliable network. If an acknowledgement (ACK) is not received for a sent packet within a certain time window, all packets starting from the unacknowledged packet are retransmitted. The send window is an abstract concept defining an imaginary box of size 2^{m-1} with three variables: S_f, S_n, and S_{size}. The send window slides one or more slots when a valid acknowledgement arrives.

S_f is Send window, first outstanding frame (frame sent but not yet acknowledged). S_n is Send window, next frame to send. S_{size} is Send window size = 2^m - 1.

The receive window is an abstract concept defining an imaginary box of size 1 with a single variable R_n. The window slides when a correct frame has arrived; it slides one slot at a time. R_n is next frame to receive.

Send window for Go-Back-N

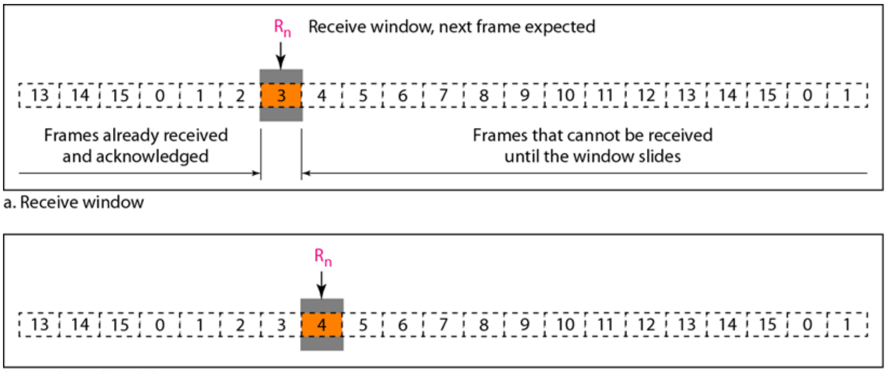


S_f Send window, first outstanding frame
S_n Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$
S_f
S_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Send window before sliding

b. Send window after sliding

Receive window for Go-Back-N ARQ

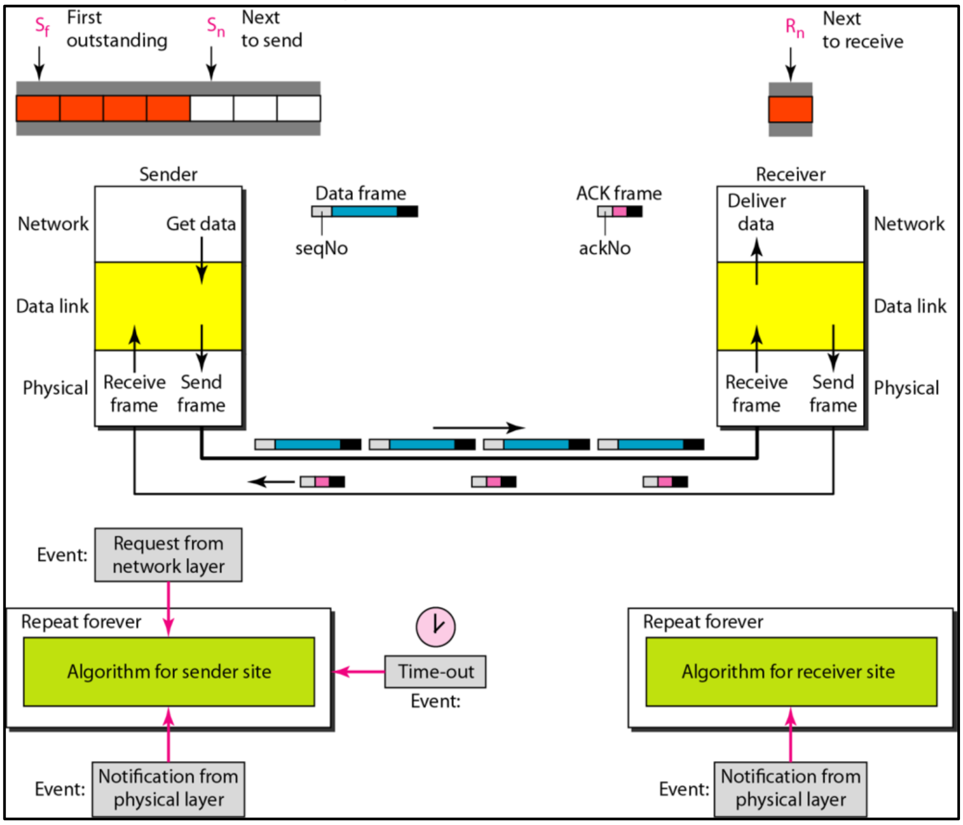


R_n Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until the window slides
R_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

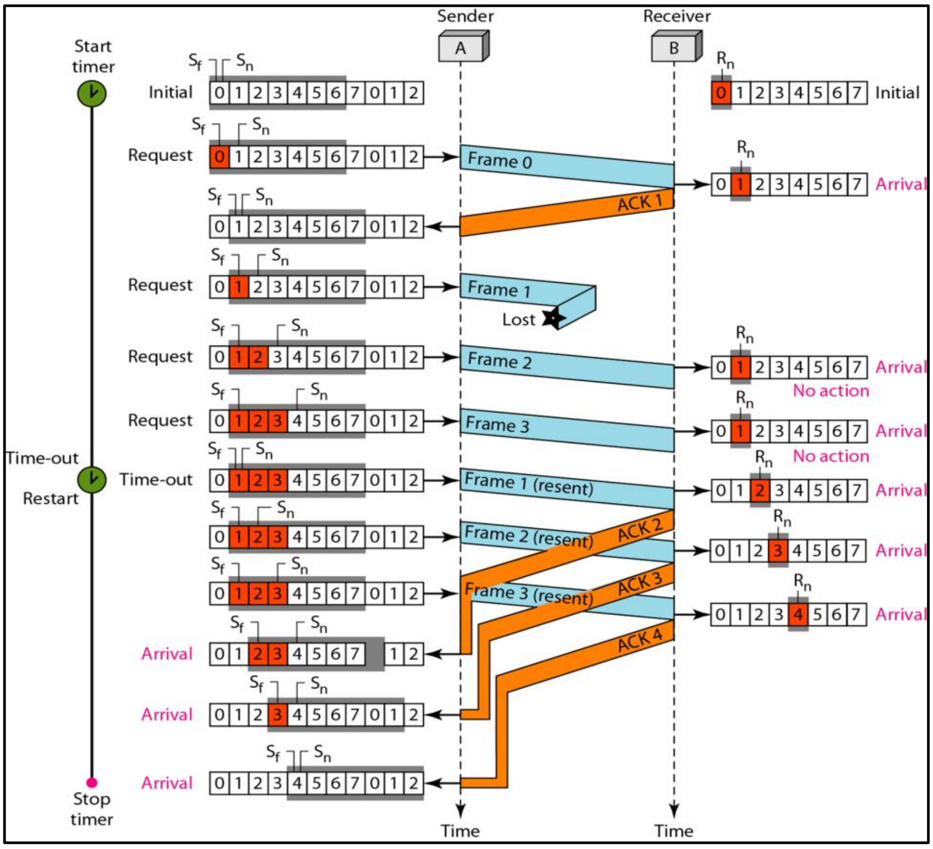
a. Receive window

b. Window after sliding

Design of Go-Back-N ARQ



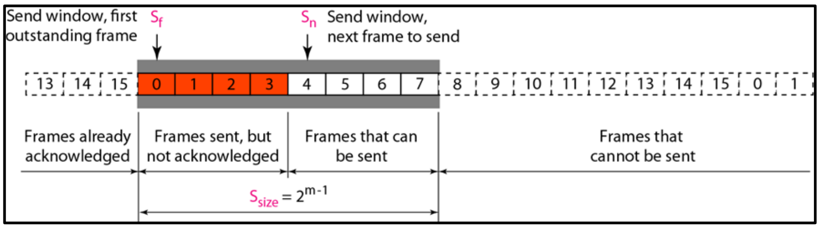
Flow Diagram of the Go-Back-N ARQ Protocol



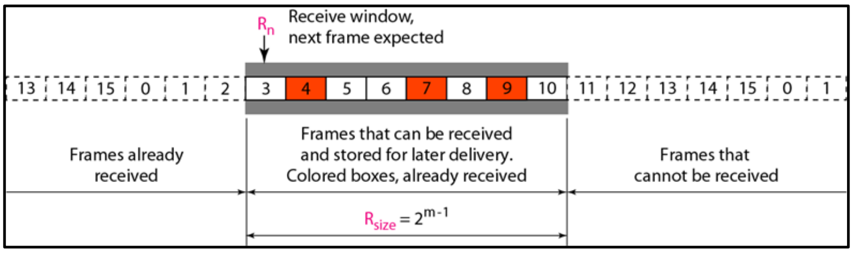
3. Selective Repeat ARQ

Selective Repeat is another Automatic Repeat reQuest (ARQ) protocol used for reliable data communication. Unlike Go-Back-N, Selective Repeat allows the receiver to individually acknowledge correctly received frames, enabling the sender to retransmit only the frames that were not received correctly.

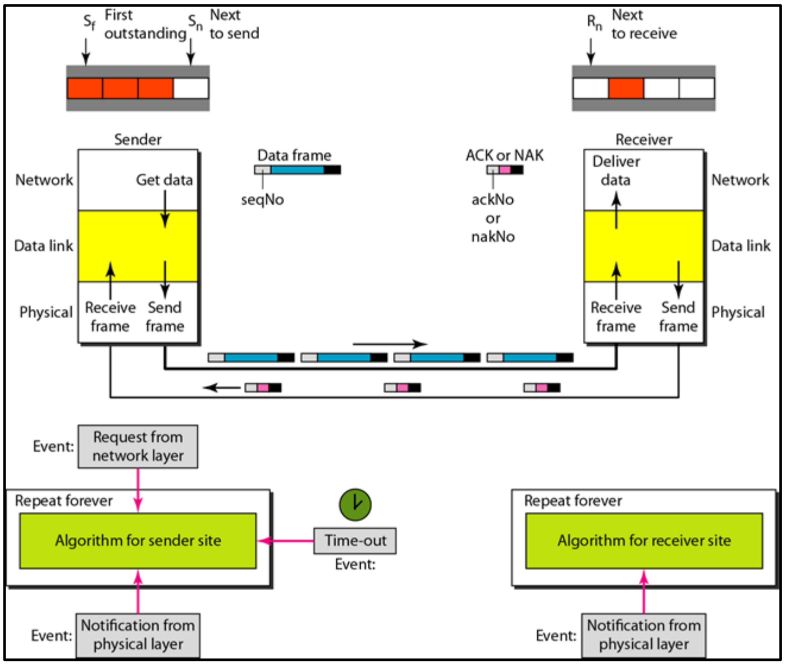
Sender window for Selective ARQ


Send window, first outstanding frame
S_f
S_n
Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged
Frames that can be sent
Frames that cannot be sent
$S_{size} = 2^{m-1}$

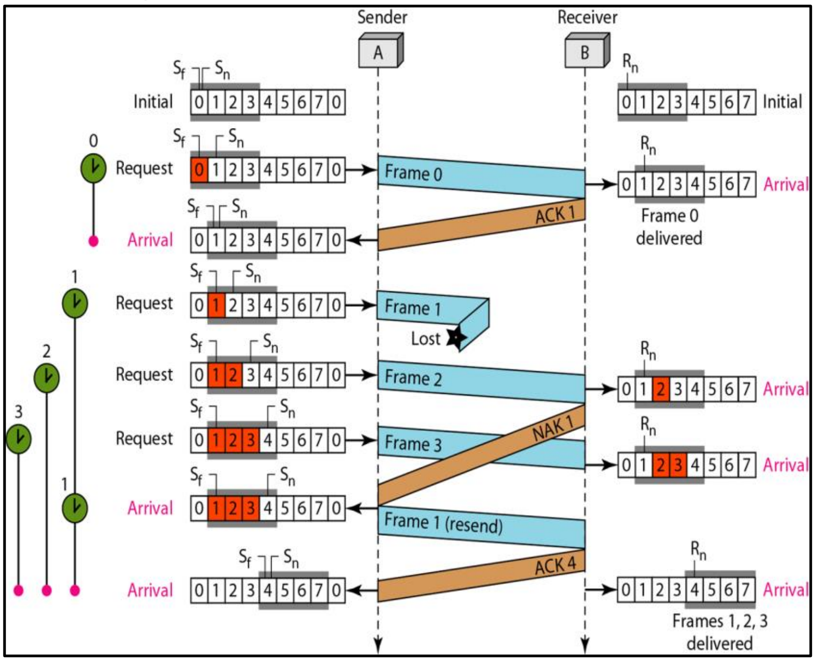
Receiver window for Selective ARQ


R_n
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already received
Frames that cannot be received
$R_{size} = 2^{m-1}$

Design for Selective ARQ



Flow Diagram of the Selective Repeat ARQ Protocol



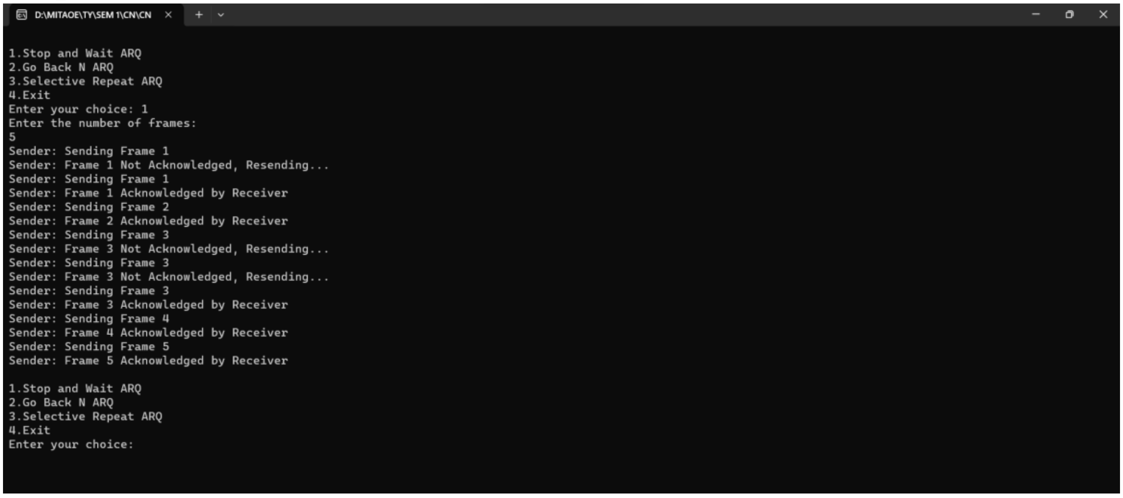
Procedure/code:

```
#include<iostream>
#include<cstdlib>
#include<ctime>
#include<unistd.h>
#include<math.h>
#define TOT_FRAMES 500
#define FRAMES_SEND 10

using namespace std;

class stop_wait_arq {
        public:
                void execute() {
                        srand(time(0));
```

```
            int frameNumber = 1, n;
            int receiverAck;

            cout << "Enter the number of frames: " << endl;
            cin >> n;
            while (frameNumber <= n) {
                sleep(1);
                cout << "Sender: Sending Frame " << frameNumber
<< endl;
                sleep(1);

                receiverAck = rand() % 2;
                if (receiverAck == 1) {
                    cout << "Sender: Frame " << frameNumber <<
" Acknowledged by Receiver" << endl;
                    frameNumber++;
                } else {
                    cout << "Sender: Frame " << frameNumber <<
" Not Acknowledged, Resending..." << endl;
                }

                sleep(1);
            }
        }
};

class go_back_n_arq {
    public:
        void execute() {
            int n,N;
            int no = 0;
            srand(time(NULL));

            cout<<"Enter the number of frames: ";
            cin>>n;

            cout<<"Enter the window size: ";
            cin>>N;

            int i=1;
            while(i<=n) {
                int x=0;
                for(int j = i; j < i+N && j <= n; j++) {
                    cout << "Sent frame " << j << endl;
                    sleep(1);
                    no++;
                }
                for(int j = i; j < i+N && j <= n; j++) {
                    int flag = rand() % 2;
                    if(!flag) {
                        cout << "Acknowledgment for frame "
<< j << " received." << endl;
                        sleep(1);
                        x++;
                    } else {
                        cout << "Frame " << j << " not received."
<< endl;
                        sleep(1);
                        cout << "Retransmitting window" <<
endl;
                        break;
                    }
                }
                cout<<endl;
                i+=x;
            }
            cout << "Total number of transmissions: " << no << endl;
        }
};

class selective_repeat_arq {
    private:
        int f,N;

    public:
        void execute() {
            printf("Enter the number of bits for the sequence number: ");
            scanf("%d", &N);

            f=pow(2, N-1);

            for(int j = 0; j < f; j++) {
                printf("\nSender: Frame %d is sent.", j);
                sleep(1);
            }

            for(int i=0; i<f; i++) {
                printf("\n");
                int flag = rand() % 2;
                if(!flag) {
                    printf("\nReceiver: Frame %d received correctly.\nAcknowledgement for Frame %d received.", i, i);
                    sleep(1);
                } else {
                    printf("\nReceiver: Frame %d received correctly.\nAcknowledgement for Frame %d lost.",i,i);
                    printf("\nSender timeout.\nResending the frame...");
                    sleep(1);
                }
            }
        }
};

int main() {
    selective_repeat_arq sr;
    stop_wait_arq st;
    go_back_n_arq gb;

    int ch=0;
    do {
        cout<<"\n1.Stop and Wait ARQ\n2.Go Back N ARQ\n3.Selective Repeat ARQ\n4.Exit\nEnter your choice: ";
        cin>>ch;
        switch(ch) {
            case 1:
                st.execute();
                break;
            case 2:
                gb.execute();
                break;
            case 3:
                sr.execute();
                break;
            default:
                cout<<"\nInvalid Choice. Please try again!";
        }
    } while(ch!=4);
}
```

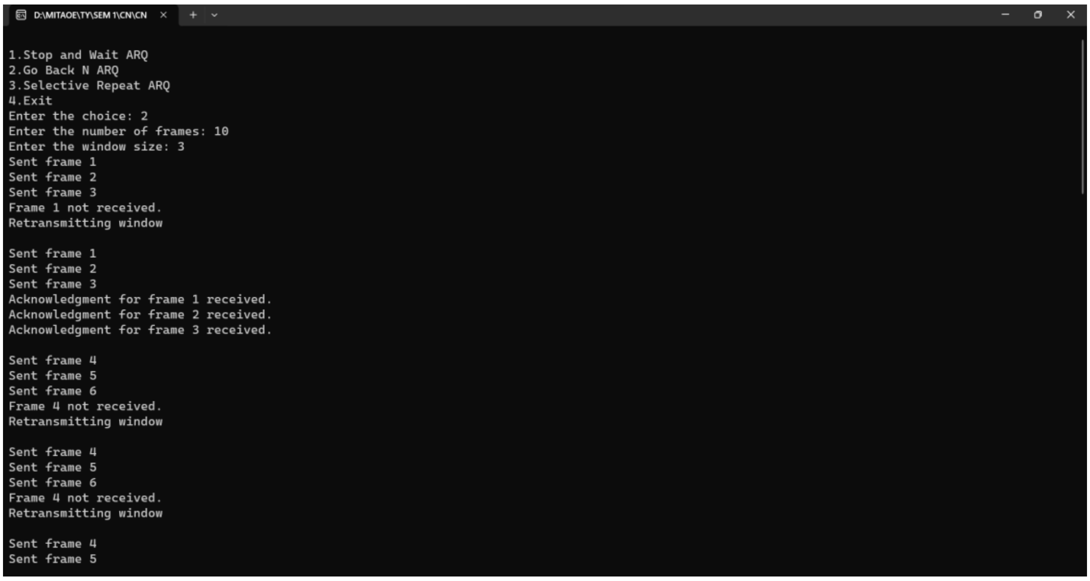
Output:

1. Stop-and-Wait ARQ:

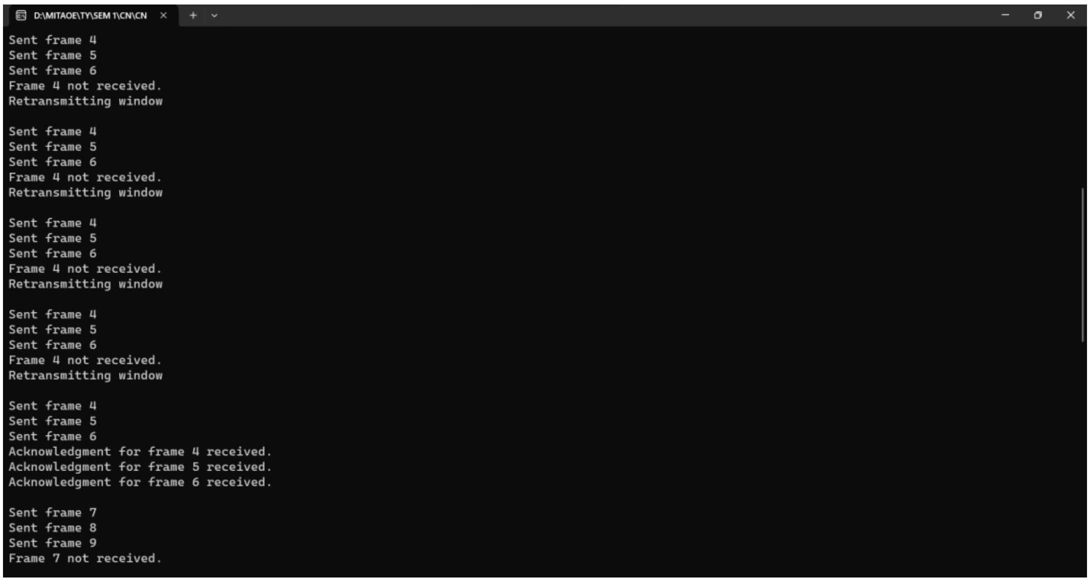


```
1.Stop and Wait ARQ
2.Go Back N ARQ
3.Selective Repeat ARQ
4.Exit
Enter your choice: 1
Enter the number of frames:
5
Sender: Sending Frame 1
Sender: Frame 1 Not Acknowledged, Resending...
Sender: Sending Frame 1
Sender: Frame 1 Acknowledged by Receiver
Sender: Sending Frame 2
Sender: Frame 2 Acknowledged by Receiver
Sender: Sending Frame 3
Sender: Frame 3 Not Acknowledged, Resending...
Sender: Sending Frame 3
Sender: Frame 3 Not Acknowledged, Resending...
Sender: Sending Frame 3
Sender: Frame 3 Acknowledged by Receiver
Sender: Sending Frame 4
Sender: Frame 4 Acknowledged by Receiver
Sender: Sending Frame 5
Sender: Frame 5 Acknowledged by Receiver

1.Stop and Wait ARQ
2.Go Back N ARQ
3.Selective Repeat ARQ
4.Exit
Enter your choice:
```

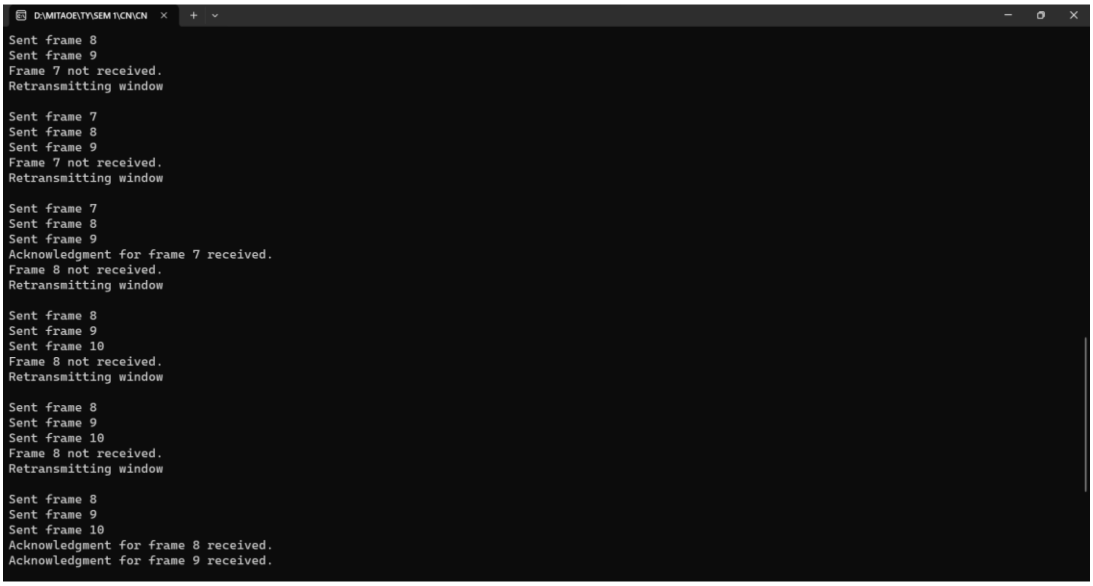
2. Go-Back-N ARQ:



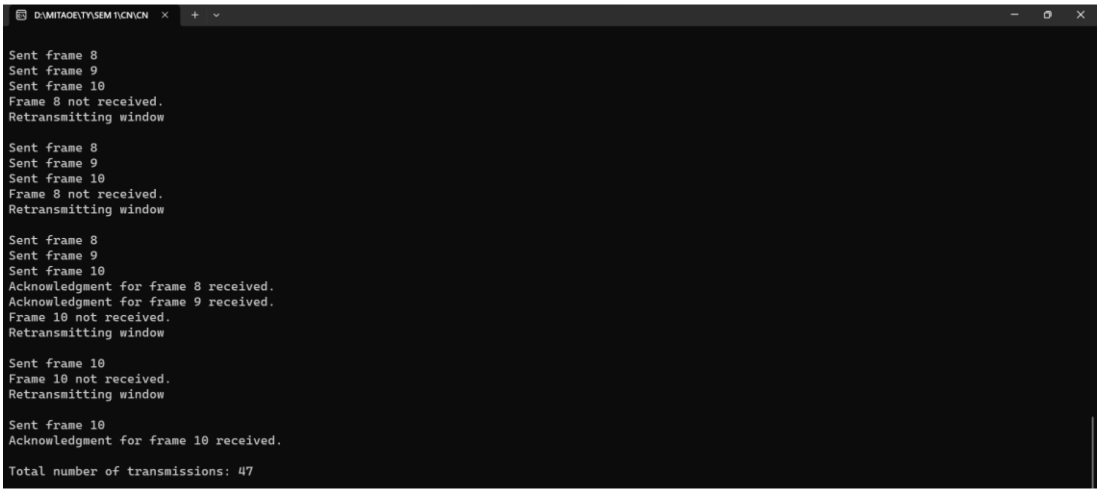
```
1.Stop and Wait ARQ
2.Go Back N ARQ
3.Selective Repeat ARQ
4.Exit
Enter the choice: 2
Enter the number of frames: 10
Enter the window size: 3
Sent frame 1
Sent frame 2
Sent frame 3
Frame 1 not received.
Retransmitting window

Sent frame 1
Sent frame 2
Sent frame 3
Acknowledgment for frame 1 received.
Acknowledgment for frame 2 received.
Acknowledgment for frame 3 received.

Sent frame 4
Sent frame 5
Sent frame 6
Frame 4 not received.
Retransmitting window

Sent frame 4
Sent frame 5
Sent frame 6
Frame 4 not received.
Retransmitting window

Sent frame 4
Sent frame 5
```



```
Sent frame 4
Sent frame 5
Sent frame 6
Frame 4 not received.
Retransmitting window

Sent frame 4
Sent frame 5
Sent frame 6
Frame 4 not received.
Retransmitting window

Sent frame 4
Sent frame 5
Sent frame 6
Frame 4 not received.
Retransmitting window

Sent frame 4
Sent frame 5
Sent frame 6
Frame 4 not received.
Retransmitting window

Sent frame 4
Sent frame 5
Sent frame 6
Acknowledgment for frame 4 received.
Acknowledgment for frame 5 received.
Acknowledgment for frame 6 received.

Sent frame 7
Sent frame 8
Sent frame 9
Frame 7 not received.
```

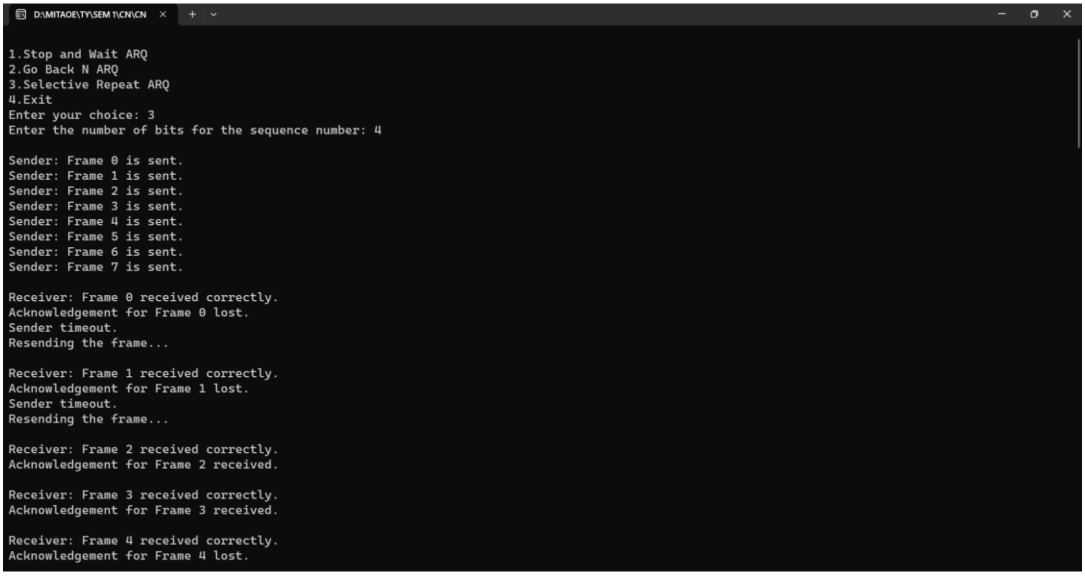


```
Sent frame 8
Sent frame 9
Frame 7 not received.
Retransmitting window

Sent frame 7
Sent frame 8
Sent frame 9
Frame 7 not received.
Retransmitting window

Sent frame 7
Sent frame 8
Sent frame 9
Acknowledgment for frame 7 received.
Frame 8 not received.
Retransmitting window

Sent frame 8
Sent frame 9
Sent frame 10
Frame 8 not received.
Retransmitting window

Sent frame 8
Sent frame 9
Sent frame 10
Frame 8 not received.
Retransmitting window

Sent frame 8
Sent frame 9
Sent frame 10
Acknowledgment for frame 8 received.
Acknowledgment for frame 9 received.
```

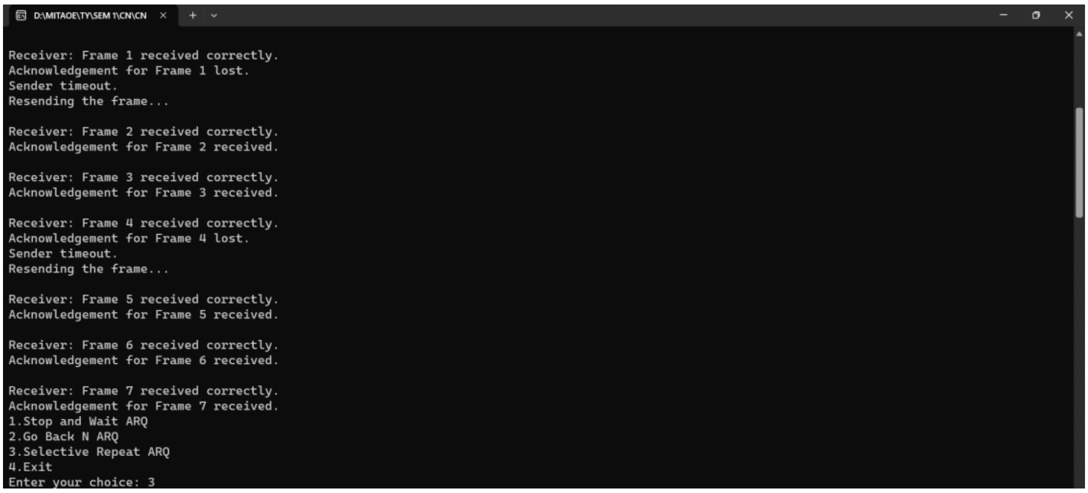


```
Sent frame 8
Sent frame 9
Sent frame 10
Frame 8 not received.
Retransmitting window

Sent frame 8
Sent frame 9
Sent frame 10
Frame 8 not received.
Retransmitting window

Sent frame 8
Sent frame 9
Sent frame 10
Acknowledgment for frame 8 received.
Acknowledgment for frame 9 received.
Frame 10 not received.
Retransmitting window

Sent frame 10
Frame 10 not received.
Retransmitting window

Sent frame 10
Acknowledgment for frame 10 received.

Total number of transmissions: 47
```

3. **Selective Repeat ARQ:**



```
1.Stop and Wait ARQ
2.Go Back N ARQ
3.Selective Repeat ARQ
4.Exit
Enter your choice: 3
Enter the number of bits for the sequence number: 4

Sender: Frame 0 is sent.
Sender: Frame 1 is sent.
Sender: Frame 2 is sent.
Sender: Frame 3 is sent.
Sender: Frame 4 is sent.
Sender: Frame 5 is sent.
Sender: Frame 6 is sent.
Sender: Frame 7 is sent.

Receiver: Frame 0 received correctly.
Acknowledgement for Frame 0 lost.
Sender timeout.
Resending the frame...

Receiver: Frame 1 received correctly.
Acknowledgement for Frame 1 lost.
Sender timeout.
Resending the frame...

Receiver: Frame 2 received correctly.
Acknowledgement for Frame 2 received.

Receiver: Frame 3 received correctly.
Acknowledgement for Frame 3 received.

Receiver: Frame 4 received correctly.
Acknowledgement for Frame 4 lost.
```



```
Receiver: Frame 1 received correctly.
Acknowledgement for Frame 1 lost.
Sender timeout.
Resending the frame...

Receiver: Frame 2 received correctly.
Acknowledgement for Frame 2 received.

Receiver: Frame 3 received correctly.
Acknowledgement for Frame 3 received.

Receiver: Frame 4 received correctly.
Acknowledgement for Frame 4 lost.
Sender timeout.
Resending the frame...

Receiver: Frame 5 received correctly.
Acknowledgement for Frame 5 received.

Receiver: Frame 6 received correctly.
Acknowledgement for Frame 6 received.

Receiver: Frame 7 received correctly.
Acknowledgement for Frame 7 received.
1.Stop and Wait ARQ
2.Go Back N ARQ
3.Selective Repeat ARQ
4.Exit
Enter your choice: 3
```

Conclusion:

In this assignment, three key Automatic Repeat reQuest (ARQ) protocols have been studied and implemented: Stop-and-Wait, Go-Back-N, and Selective Repeat.

Stop-and-Wait ARQ:

Simple, but inefficient due to its one-frame-at-a-time approach.

Go-Back-N ARQ:

Allows multiple frames in flight, but inefficiency arises from retransmitting all unacknowledged frames.

Selective Repeat ARQ:

Efficient, allowing selective acknowledgement and buffering of out-of-order frames.

Understanding these protocols is vital for designing reliable communication systems, each catering to specific network conditions and trade-offs between efficiency and complexity.