Batch: D2	**Roll No.:16010123325**
Experiment / assignment / tutorial No._5_	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

Experiment No.:5

TITLE: Flow control Mechanism: Go-Back-N ARQ Sliding Window Protocol using Socket programming

---

AIM: Implementation of Flow Control Mechanism: Stop and Wait ARQ / Go-Back- N / Selective Repeat Sliding Window Protocol ARQ using sockets.

Expected Outcome of Experiment:

CO2: Demonstrate Data Link Layer, MAC layer technologies & protocols and implement the functionalities like error control, flow control.

Books/ Journals/ Websites referred:

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

---

Pre-Lab/ Prior Concepts:

Java Socket Programming, Flow Control, Go-Back-Stop and Wait

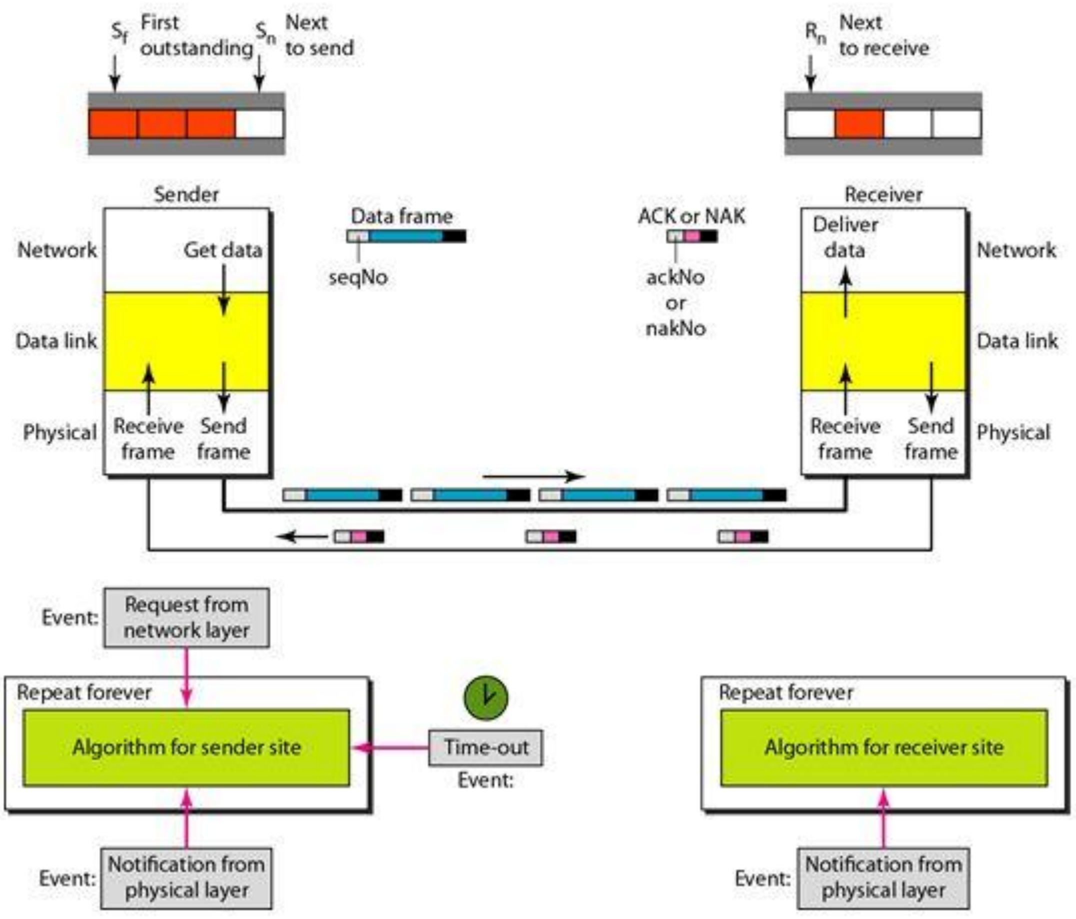
New Concepts to be learned: Window Flow Control

Go-Back-N ARQ is mainly a specific instance of Automatic Repeat Request (ARQ) protocol where the sending process continues to send a number of frames as specified by the window size even without receiving an acknowledgement(ACK) packet from the

receiver.
The sender keeps a copy of each frame until the arrival of acknowledgement.
This protocol is a practical approach to the sliding window.

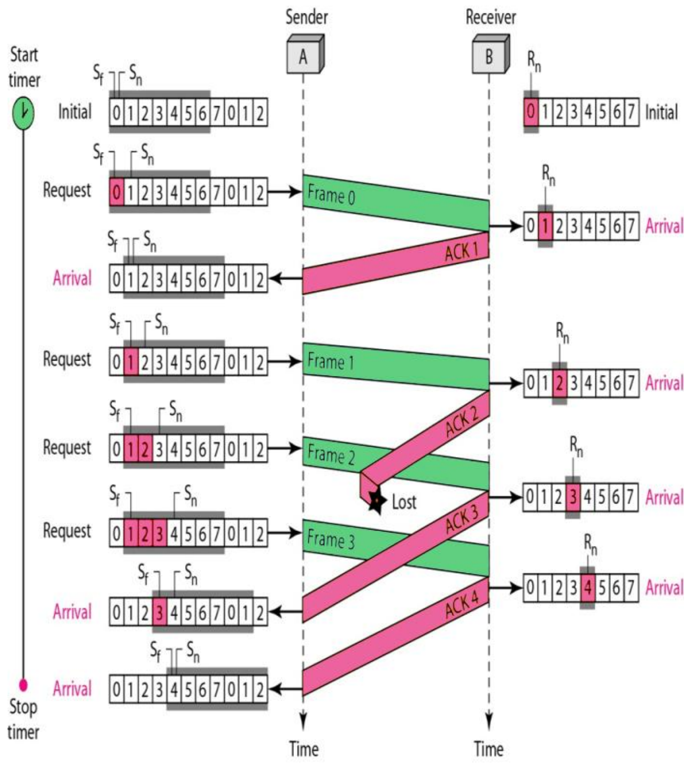
- In Go-Back-N ARQ, the size of the sender is N and the size of the receiver window is always 1.
- This protocol makes the use of **cumulative acknowledgements** means here the receiver maintains an acknowledgement timer; whenever the receiver receives a new frame from the sender then it starts a new acknowledgement timer. When the timer expires then the receiver sends the **cumulative acknowledgement** for all the frames that are unacknowledged by the receiver at that moment.
- It is important to note that the new acknowledgement timer only starts after the receiving of a new frame, it does not start after the expiry of the **old acknowledgement timer**.
- If the receiver receives a corrupted frame, then it silently discards that corrupted frame and the correct frame is retransmitted by the sender after the timeout timer expires. Thus receiver silently discards the corrupted frame. By discarding silently we mean that: “Simply rejecting the frame and not taking any action for the frame".
- In case after the expiry of the acknowledgement timer, suppose there is only one frame that is left to be acknowledged. In that case, the receiver sends the independent acknowledgement for that frame.
- In case if the receiver receives the out of order frame then it simply discards all the frames.
- In case if the sender does not receive any acknowledgement then the entire window of the frame will be retransmitted in that case.
- Using the Go-Back-N ARQ protocol leads to the retransmission of the lost frames after the expiry of the **timeout timer.**

Design of Go-Back-N ARQ



1. Take data from user about how many bit windows is case of go back n and selective repeat.
2. Generate frames randomly and show the transmission
3. Generate the random number for the frame to be lost.
4. For Go – Back – N transmit all the frames after that number till max number
5. For Selective repeat transmit the selected frame which is not received by the receiver.

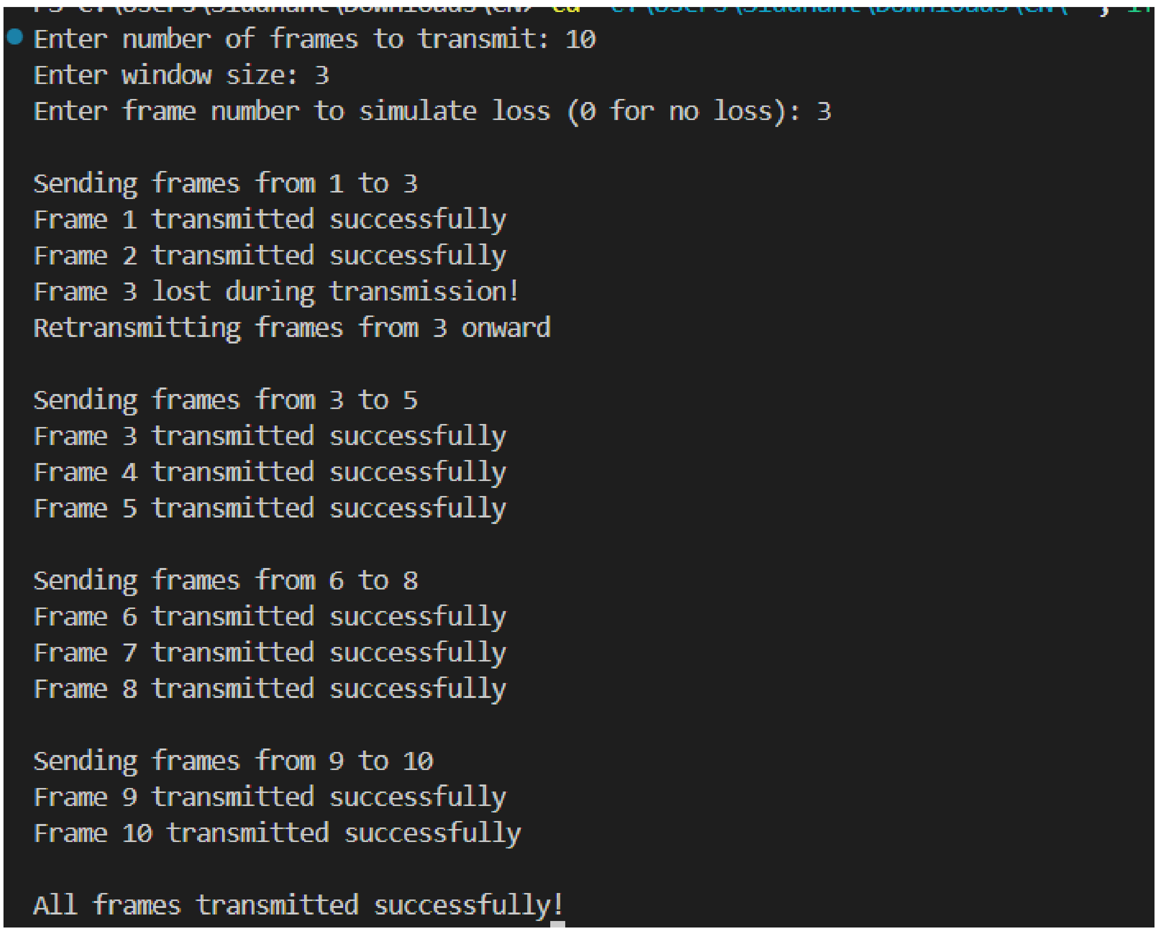
Flow Diagram:



IMPLEMENTATION: (printout of code)

```
import java.util.Scanner;

public class GoBackN {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // User input
        System.out.print("Enter number of frames to transmit: ");
        int totalFrames = sc.nextInt();

        System.out.print("Enter window size: ");
        int windowSize = sc.nextInt();

        System.out.print("Enter frame number to simulate loss (0 for no
loss): ");
        int lostFrame = sc.nextInt();

        int i = 1;
        while (i <= totalFrames) {
            int windowEnd = Math.min(i + windowSize - 1, totalFrames);
            System.out.println("\nSending frames from " + i + " to " +
windowEnd);

            boolean lossOccurred = false;

            for (int j = i; j <= windowEnd; j++) {
                if (j == lostFrame) {
                    System.out.println("Frame " + j + " lost during
transmission!");
                    lossOccurred = true;
                    break;
                } else {
                    System.out.println("Frame " + j + " transmitted
successfully");
                }
            }

            if (lossOccurred) {
                System.out.println("Retransmitting frames from " + lostFrame
+ " onward");
                i = lostFrame;
                lostFrame = 0;
            } else {
                i = windowEnd + 1;
            }
        }

        System.out.println("\nAll frames transmitted successfully!");
        sc.close();
    }
}
```

Output:



```
Enter number of frames to transmit: 10
Enter window size: 3
Enter frame number to simulate loss (0 for no loss): 3

Sending frames from 1 to 3
Frame 1 transmitted successfully
Frame 2 transmitted successfully
Frame 3 lost during transmission!
Retransmitting frames from 3 onward

Sending frames from 3 to 5
Frame 3 transmitted successfully
Frame 4 transmitted successfully
Frame 5 transmitted successfully

Sending frames from 6 to 8
Frame 6 transmitted successfully
Frame 7 transmitted successfully
Frame 8 transmitted successfully

Sending frames from 9 to 10
Frame 9 transmitted successfully
Frame 10 transmitted successfully

All frames transmitted successfully!
```

CONCLUSION:

The experiment demonstrated the Go-Back-N ARQ Sliding Window Protocol using socket programming. It showed how multiple frames can be sent without waiting for individual ACKs, improving efficiency. Cumulative acknowledgements and retransmissions ensured reliable data transfer even with lost or corrupted frames. The effect of window size and frame loss on transmission was observed. Overall, the experiment highlighted the importance of flow control and error recovery in reliable communication.

Post Lab Questions

1. Compare Go-Back-N and Stop and Wait.

Feature	Stop-and-Wait ARQ	Go-Back-N ARQ
Transmission	Sends one frame at a time	Sends multiple frames up to window size N
Efficiency	Low, idle time while waiting for ACK	High, utilizes channel efficiently
Acknowledgement	Individual ACK per frame	Cumulative ACK for all correctly received frames
Retransmission	Only lost or corrupted frame is retransmitted	All frames from the lost/corrupted frame onward are retransmitted
Window Size	1	N (sender window)
Use Case	Low-speed channels or simple protocols	High-speed networks where channel utilization is important

2. What is Flow Control and why it is necessary?

Ans: Flow Control is a technique used to control the rate of data transmission between a sender and receiver to prevent overwhelming the receiver.It ensures efficient and reliable communication by matching the sender’s transmission rate with the receiver’s processing capability.

Without flow control, faster senders could overflow receiver buffers, causing data loss and retransmissions, reducing network efficiency.

3. The maximum window size for data transmission using the selective reject protocol with n-bit frame sequence numbers is

   a) 2n b) 2n-1 c) 2n-1 d)2n-2

   Ans: 2n-1

Date: __________ Signature of Faculty In-charge