# Name: Srujan Patwardhan

# Class: TY CS D

# Roll No: 2

# PRN: 12210847

**Code:**

import java.util.Random;

import java.util.Scanner;

public class SlidingWindowProtocols {

    public static void goBackN(int[] packets, int windowSize) {

        int base = 0;

        int nextSeqNum = 0;

        int n = packets.length;

        boolean[] ackReceived = new boolean[n];

        Random random = new Random();

        while (base < n) {

            while (nextSeqNum < base + windowSize && nextSeqNum < n) {

                System.out.println("Sending packet: " + packets[nextSeqNum]);

                nextSeqNum++;

            }

            for (int i = base; i < nextSeqNum; i++) {

                if (!ackReceived[i]) {

                    if (random.nextInt(10) < 7) { // 70% chance of successful ACK

                        System.out.println("Received ACK for packet: " + packets[i]);

                        ackReceived[i] = true;

                        base++;

                    } else {

                        System.out.println("ACK for packet " + packets[i] + " lost or delayed.");

                        break;

                    }

                }

            }

            if (!ackReceived[base]) {

                System.out.println("Timeout occurred. Resending packets starting from: " + packets[base]);

                nextSeqNum = base;

            }

        }

        System.out.println("All packets successfully sent using Go-Back-N.");

    }

    public static void selectiveRepeat(int[] packets, int windowSize) {

        int n = packets.length;

        boolean[] ackReceived = new boolean[n];

        Random random = new Random();

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

            int end = Math.min(i + windowSize, n);

            for (int j = i; j < end; j++) {

                System.out.println("Sending packet: " + packets[j]);

            }

            for (int j = i; j < end; j++) {

                if (!ackReceived[j]) {

                    if (random.nextInt(10) < 7) { // 70% chance of successful ACK

                        System.out.println("Received ACK for packet: " + packets[j]);

                        ackReceived[j] = true;

                    } else {

                        System.out.println("ACK for packet " + packets[j] + " lost or delayed.");

                    }

                }

            }

            for (int j = i; j < end; j++) {

                if (!ackReceived[j]) {

                    System.out.println("Retransmitting packet: " + packets[j]);

                    if (random.nextInt(10) < 7) { // 70% chance of successful ACK after retransmission

                        System.out.println("Received ACK for retransmitted packet: " + packets[j]);

                        ackReceived[j] = true;

                    }

                }

            }

        }

        System.out.println("All packets successfully sent using Selective Repeat.");

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of packets: ");

        int numPackets = scanner.nextInt();

        int[] packets = new int[numPackets];

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

            packets[i] = i + 1;

        }

        System.out.print("Enter the window size: ");

        int windowSize = scanner.nextInt();

        System.out.println("\nChoose protocol to simulate:");

        System.out.println("1. Go-Back-N");

        System.out.println("2. Selective Repeat");

        int choice = scanner.nextInt();

        switch (choice) {

            case 1:

                System.out.println("\nSimulating Go-Back-N protocol...");

                goBackN(packets, windowSize);

                break;

            case 2:

                System.out.println("\nSimulating Selective Repeat protocol...");

                selectiveRepeat(packets, windowSize);

                break;

            default:

                System.out.println("Invalid choice.");

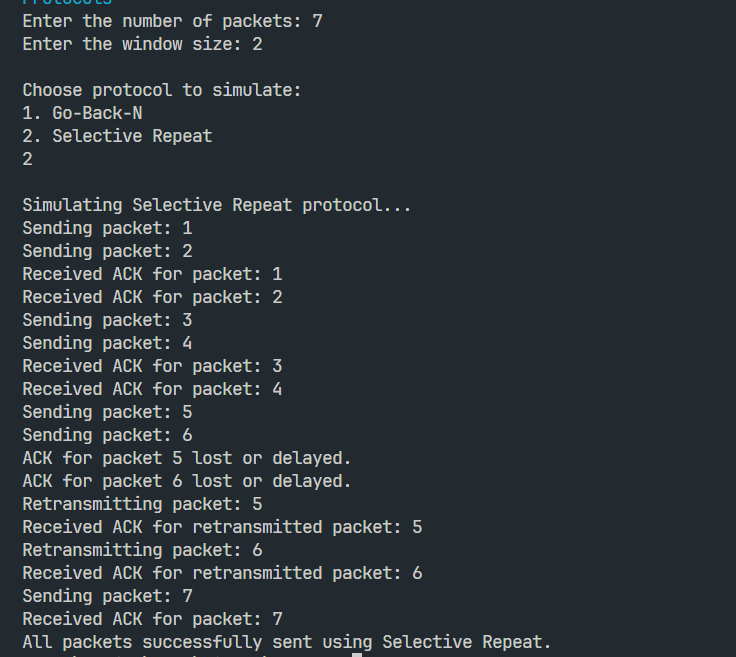
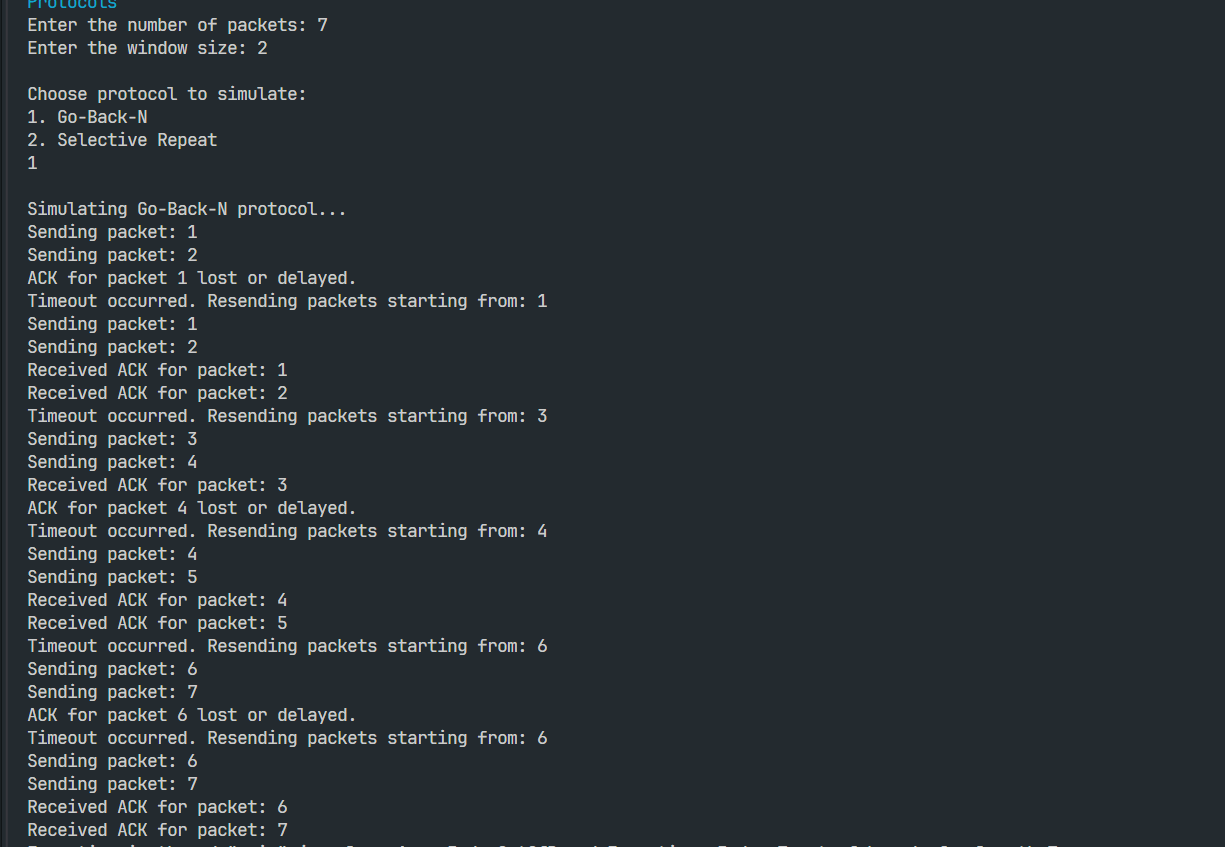
        }

        scanner.close();

    }

}

**Output:**

****