

Develop a java program to implement a sliding window protocol in the data link layer.

Sliding Window Protocol (Go-Back-N)

The sliding window protocol is a method used in the data link layer for reliable and efficient data transmission. It allows a sender to send multiple frames before needing an acknowledgment for the first one, which helps in utilizing the available bandwidth more effectively.

There are two main types of sliding window protocols:

1. **Go-Back-N (GBN):** In this protocol, the sender can send multiple frames specified by a window size. If an error is detected in a frame, the receiver discards it and all subsequent frames, and the sender retransmits all frames starting from the erroneous one.
2. **Selective Repeat (SR):** This protocol allows the receiver to accept out-of-order frames and only retransmits the specific frames that were corrupted or lost.

Explanation:

1. **Initialization:**
 - The SlidingWindowProtocol class is initialized with the window size and the number of frames to be sent.
 - A queue window is used to represent the frames currently in the window that are awaiting acknowledgment.
2. **Sending Frames:**
 - The sendFrames method sends frames within the window size.
 - For each frame sent, it prints a message indicating the frame's transmission.
 - It then simulates the reception of an acknowledgment (ACK) or its loss using a random boolean.
3. **Handling Acknowledgments:**
 - If an ACK is lost, the entire window is retransmitted (as per Go-Back-N).
 - If an ACK is received, the corresponding frame is removed from the window, allowing the next frame to be sent.
4. **Retransmission:**
 - The retransmitWindow method retransmits all frames in the current window when an acknowledgment is lost.

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Random;
import java.util.Scanner;
```

```
class SlidingWindowProtocol {

    private int windowSize;
    private int numFrames;
```

```

private Queue<Integer> window;

// Constructor to initialize the protocol
public SlidingWindowProtocol(int windowSize, int numFrames) {
    this.windowSize = windowSize;
    this.numFrames = numFrames;
    this.window = new LinkedList<>();
}

// Method to simulate the sending of frames using Go-Back-N
public void sendFrames() {
    int currentFrame = 1; // Start with frame 1
    Random random = new Random();
    while (currentFrame <= numFrames) {
        // Send frames within the window size
        while (window.size() < windowSize && currentFrame <= numFrames) {
            System.out.println("Sending Frame " + currentFrame);
            window.add(currentFrame);
            currentFrame++;
        }

        // Simulate receiving an acknowledgment
        if (!window.isEmpty()) {
            int ackFrame = window.peek(); // Assume ACK for the first frame in the window
            boolean isLost = random.nextBoolean(); // Simulate ACK loss randomly

            if (isLost) {
                System.out.println("ACK for Frame " + ackFrame + " lost. Retransmitting window...");
                retransmitWindow();
            } else {
                System.out.println("ACK received for Frame " + ackFrame);
                window.poll(); // Remove the acknowledged frame from the window
            }
        }
    }

    System.out.println("All frames sent successfully.");
}

// Method to retransmit all frames in the current window
private void retransmitWindow() {
    for (int frame : window) {
        System.out.println("Retransmitting Frame " + frame);
    }
}

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

```

```

        System.out.print("Enter the number of frames to be sent: ");
        int numFrames = scanner.nextInt();

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

        SlidingWindowProtocol protocol = new SlidingWindowProtocol(windowSize, numFrames);
        protocol.sendFrames();
    }
}

```

Expected Output

Enter the number of frames to be sent: **5**

Enter the window size: **3**

Sending Frame 1

Sending Frame 2

Sending Frame 3

ACK received for Frame 1

Sending Frame 4

ACK for Frame 2 lost. Retransmitting window...

Retransmitting Frame 2

Retransmitting Frame 3

Retransmitting Frame 4

ACK received for Frame 2

ACK received for Frame 3

Sending Frame 5

ACK received for Frame 4

ACK received for Frame 5

All frames sent successfully.

Key Points:

- The program simulates Go-Back-N by sending multiple frames and handling possible ACK loss.
- The window size determines how many frames can be sent before needing an acknowledgment.
- The program provides a basic simulation of how data packets might be managed in a real-world scenario.