

CN LAB 4

Name: Rachit Nimje

Batch: 1

Class: TY-CS-D

PRN: 12210952

Roll no: 12

Title: Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

Code:

```
import java.util.*;

public class SlidingWindowProtocolSimulation {
    private static final Random random = new Random();
    private static final double LOSS_PROBABILITY = 0.3;

    public static void main(String[] args) {
        int windowSize = 3;
        int totalFrames = 5;

        System.out.println("Go-Back-N Simulation:");
        simulateProtocol(windowSize, totalFrames, true);

        System.out.println("\nSelective Repeat Simulation:");
        simulateProtocol(windowSize, totalFrames, false);
    }

    private static void simulateProtocol(int windowSize, int totalFrames,
boolean isGoBackN) {
        int base = 0;
        Set<Integer> receivedFrames = new HashSet<>();

        while (base < totalFrames) {
            System.out.println("Current window: [" + base + ", " +
Math.min(base + windowSize - 1, totalFrames - 1) + "]" );

            for (int i = base; i < Math.min(base + windowSize, totalFrames);
i++) {
                if (!receivedFrames.contains(i)) {
                    boolean frameLost = random.nextDouble() < LOSS_PROBABILITY;
                    System.out.println("Sending frame " + i + (frameLost ? "
(lost)" : " (received)"));
                    if (!frameLost) {
                        receivedFrames.add(i);
                    }
                }
            }

            if (isGoBackN) {
                if (receivedFrames.contains(base)) {
```

```

        while (receivedFrames.contains(base) && base < totalFrames)
    {
        System.out.println("Acknowledging frame " + base);
        base++;
    }
    } else {
        System.out.println("Timeout: Go-Back-N to frame " + base);
    }
    } else {
        for (int i = base; i < Math.min(base + windowSize,
totalFrames); i++) {
            if (receivedFrames.contains(i)) {
                System.out.println("Acknowledging frame " + i);
                if (i == base) {
                    while (receivedFrames.contains(base) && base <
totalFrames) {
                        base++;
                    }
                }
            } else {
                System.out.println("Timeout: Resend frame " + i);
            }
        }
    }

    System.out.println("New base: " + base);
    System.out.println();
}

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

```

Output:

```
/home/halogen/.jdk/openjdk-22.0.2/bin/java -javaagent:
Go-Back-N Simulation:
Current window: [0, 2]
Sending frame 0 (received)
Sending frame 1 (received)
Sending frame 2 (received)
Acknowledging frame 0
Acknowledging frame 1
Acknowledging frame 2
New base: 3

Current window: [3, 4]
Sending frame 3 (received)
Sending frame 4 (received)
Acknowledging frame 3
Acknowledging frame 4
New base: 5

All frames transmitted successfully
```

```
Selective Repeat Simulation:
Current window: [0, 2]
Sending frame 0 (lost)
Sending frame 1 (received)
Sending frame 2 (lost)
Timeout: Resend frame 0
Acknowledging frame 1
Timeout: Resend frame 2
New base: 0

Current window: [0, 2]
Sending frame 0 (lost)
Sending frame 2 (received)
Timeout: Resend frame 0
Acknowledging frame 1
Acknowledging frame 2
New base: 0
```

```
Current window: [0, 2]
Sending frame 0 (received)
Acknowledging frame 0
Acknowledging frame 1
Acknowledging frame 2
Timeout: Resend frame 3
Timeout: Resend frame 4
New base: 3
```

```
Current window: [3, 4]
Sending frame 3 (lost)
Sending frame 4 (received)
Timeout: Resend frame 3
Acknowledging frame 4
New base: 3
```

```
Current window: [3, 4]
Sending frame 3 (lost)
Timeout: Resend frame 3
Acknowledging frame 4
New base: 3
```

```
Current window: [3, 4]
Sending frame 3 (lost)
Timeout: Resend frame 3
Acknowledging frame 4
New base: 3
```

```
Current window: [3, 4]
Sending frame 3 (received)
Acknowledging frame 3
Acknowledging frame 4
New base: 5
```

```
All frames transmitted successfully
```

```
Process finished with exit code 0
```