**35. Implementing the sliding window protocol in java/C.**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h> // For sleep function

#include <time.h> // For srand and time

#define WINDOW\_SIZE 4 // Sliding window size

#define TOTAL\_FRAMES 10 // Total frames to send

// Function to simulate frame transmission

void sendFrames(int frames[], int start, int window) {

printf("\nSender: Sending frames %d to %d...\n", start, start + window - 1);

}

// Function to simulate receiver acknowledgment

int receiveFrames(int frames[], int start, int window) {

int i;

for (i = start; i < start + window && i < TOTAL\_FRAMES; i++) {

if (rand() % 5 == 0) { // Simulate random error (20% probability)

printf("Receiver: Frame %d is lost! ?\n", i);

return i; // Return the lost frame number

} else {

printf("Receiver: Frame %d received ?\n", i);

}

}

return -1; // No frame loss

}

int main() {

int frames[TOTAL\_FRAMES];

int base = 0; // Base frame index

int nextFrame;

int i; // Declare loop variable outside the loop

srand(time(0));

// Initialize frame numbers

for (i = 0; i < TOTAL\_FRAMES; i++)

frames[i] = i;

while (base < TOTAL\_FRAMES) {

// Send frames in the window

sendFrames(frames, base, WINDOW\_SIZE);

// Simulate receiver acknowledgment

int lostFrame = receiveFrames(frames, base, WINDOW\_SIZE);

if (lostFrame == -1) {

printf("Sender: All frames acknowledged ?\n");

base += WINDOW\_SIZE; // Move window forward

} else {

printf("Sender: Timeout! Resending from Frame %d... ??\n", lostFrame);

base = lostFrame; // Resend from lost frame

}

sleep(1); // Simulate transmission delay

}

printf("\nAll frames sent successfully! ??\n");

return 0;

}

A screenshot of a computer

AI-generated content may be incorrect.