

8.stop and wait protocol

Python

```
import random
import time
def stop_and_wait():
    print("---- Stop and Wait Protocol Simulation ----\n")
    total_frames = int(input("Enter number of frames to send: "))
    frames = []
    for i in range(total_frames):
        data = input(f"Enter data for Frame {i+1}: ")
        frames.append(data)
    frame = 0
    while frame < total_frames:
        print(f"\nSender: Sending Frame {frame+1} -> \"{frames[frame]}\"")
        time.sleep(1)
        if random.random() < 0.2:
            print(f"❌ Frame {frame+1} lost during transmission!")
            print(f"Sender: Timeout! Resending Frame {frame+1}...")
            continue
        print(f"✅ Receiver: Frame {frame+1} received -> \"{frames[frame]}\"")
        time.sleep(1)
        if random.random() < 0.2:
            print(f"❌ ACK {frame+1} lost!")
            print(f"Sender: Timeout! Resending Frame {frame+1}...")
        else:
            print(f"✅ Receiver: Sending ACK {frame+1}")
            print(f"Sender: ACK {frame+1} received. Moving to next frame.")
            frame += 1
        time.sleep(1)
    print(f"\n🎉 All {total_frames} frames sent successfully using Stop-and-Wait Protocol!")

# Run the program
if __name__ == "__main__":
    stop_and_wait()
```

C

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <string.h>
#define MAX 100
int frameReceived() {
    return (rand() % 10 < 8);
}
int ackReceived() {
    return (rand() % 10 < 8);
}
int main() {
    int total_frames, frame = 1;
    char data[50][MAX];
    srand(time(NULL));
    printf("---- Stop and Wait Protocol Simulation ----\n\n");
```

```

printf("Enter total number of frames to send: ");
scanf("%d", &total_frames);
getchar();
for (int i = 1; i <= total_frames; i++) {
    printf("Enter data for Frame %d: ", i);
    fgets(data[i], MAX, stdin);
    data[i][strcspn(data[i], "\n")] = '\0';
}
while (frame <= total_frames) {
    printf("\nSender: Sending Frame %d -> Data: \"%s\"\n", frame, data[frame]);
    sleep(1);
    if (!frameReceived()) {
        printf("Error: Frame %d lost or corrupted during transmission!\n", frame);
        printf("Sender: Timeout! Resending Frame %d...\n", frame);
        sleep(1);
        continue;
    }
    printf("Receiver: Frame %d received successfully. Data: \"%s\"\n", frame, data[frame]);
    sleep(1);
    if (ackReceived()) {
        printf("Receiver: Sending ACK %d.\n", frame);
        printf("Sender: ACK %d received. Moving to next frame.\n", frame);
        frame++;
    } else {
        printf("Error: ACK %d lost during transmission!\n", frame);
        printf("Sender: Timeout! Resending Frame %d...\n", frame);
        // retransmit same frame
    }
    sleep(1);
}
printf("\nAll %d frames transmitted successfully using Stop-and-Wait Protocol.\n",
total_frames);
return 0;
}

```