```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <time.h>
#include <sys/time.h>
#include <sys/select.h>

#define BUFFER_SIZE 512
#define PORT 8882

typedef struct data_packet{
    int sequence_number;
    char data[BUFFER_SIZE];
} Data_packet;

typedef struct ack_packet{
    int sequence_number;
} Ack_packet;

void die(char* error_message){
    perror(error_message);
    exit(1);
}

int main(){
    time_t t;

    srand((unsigned) time(&t));

    struct sockaddr_in other;
    int sock, i;

    char message[BUFFER_SIZE];
    Data_packet d_pkt;
    Ack_packet a_pkt;

    if((sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0){
        die("socket()");
    }

    struct timeval timer;
    timer.tv_sec = 5;
    timer.tv_usec = 0;
    fd_set listen_for;
    FD_ZERO(&listen_for);


    memset((char*) &other, 0, sizeof(other));
    other.sin_family = AF_INET;
    other.sin_port = htons(PORT);
    other.sin_addr.s_addr = inet_addr("127.0.0.1");

    int state = 0;
    int s_len = sizeof(other);

    while(1){
        switch(state){
            case 0:
                printf("Enter message (0):\n");
                fgets(d_pkt.data, sizeof(d_pkt.data), stdin);
                d_pkt.sequence_number = 0;
                if(sendto(sock, &d_pkt, sizeof(d_pkt), 0, (struct sockaddr*) &other, s_len) == -1){
                    die("sendto()");
                }
                state = 1;
                break;

            case 1:
                FD_SET(sock, &listen_for);
                select(sock + 1, &listen_for, NULL, NULL, &timer);
                if(FD_ISSET(sock, &listen_for)){
                    if(recvfrom(sock, &a_pkt, sizeof(a_pkt), 0, (struct sockaddr*) &other, &s_len) == -1){
                        die("recvfrom()");
                    }
                    if(rand()%100 == 0){
                        state = 1;
                        break;
                    }
                    if(a_pkt.sequence_number != 0){
                        break;
                    }
                    printf("Received ack packet with sequence number %d\n", a_pkt.sequence_number);
                    state = 2;
                    break;
                }
                else{
                    state = 0;
                    break;
                }
            case 2:
                printf("Enter message (1):\n");
                fgets(d_pkt.data, sizeof(d_pkt.data), stdin);
                d_pkt.sequence_number = 1;
                if((sendto(sock, &d_pkt, sizeof(d_pkt), 0, (struct sockaddr*) &other, s_len)) == -1){
                    die("sendto()");
                }
                state = 3;
                break;

            case 3:
                FD_SET(sock, &listen_for);
                select(sock + 1, &listen_for, NULL, NULL, &timer);
                if(FD_ISSET(sock, &listen_for)){
                    if((recvfrom(sock, &a_pkt, sizeof(a_pkt), 0, (struct sockaddr*) &other, &s_len)) == -1){
                        die("recvfrom()");
                    }
                    if(rand()%100 == 0){
                        state = 3;
                        break;
                    }
                    if(a_pkt.sequence_number != 1){
                        break;
                    }
                    printf("Received ack packet with sequence number %d\n", a_pkt.sequence_number);
                    state = 0;
                    break;
                }
                else{
                    state = 2;
                    break;
                }
        }
    }

    close(sock);
    return 0;
}
```