```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>
#include <signal.h>

#define BUFFER_SIZE 512
#define PORT 8882
int state = 0;

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);
}

void sig_handler(int signo);

int main(){

    signal(SIGALRM, sig_handler);

    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");

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

    while(1){
        switch(state){
            case 0:
                printf("Enter message (0):\n");
```

```
70                  fgets(d_pkt.data, sizeof(d_pkt.data), stdin);
71                  d_pkt.sequence_number = 0;
72                  if(sendto(sock, &d_pkt, sizeof(d_pkt), 0, (struct sockaddr*)
    &other, s_len) == -1){
73                      die("sendto()");
74                  }
75                  alarm(5);
76                  state = 1;
77                  break;
78
79              case 1:
80                  FD_SET(sock, &listen_for);
81                  select(sock + 1, &listen_for, NULL, NULL, &timer);
82                  if(FD_ISSET(sock, &listen_for)){
83                      if(recvfrom(sock, &a_pkt, sizeof(a_pkt), 0, (struct
    sockaddr*) &other, &s_len) == -1){
84                          die("recvfrom()");
85                      }
86                      if(rand()%100 == 0){
87                          state = 1;
88                          break;
89                      }
90                      if(a_pkt.sequence_number != 0){
91                          break;
92                      }
93                      printf("Received ack packet with sequence number %d\n",
    a_pkt.sequence_number);
94                      state = 2;
95                      break;
96                  }
97                  else{
98                      state = 0;
99                      break;
100                 }
101             case 2:
102                 printf("Enter message (1):\n");
103                 fgets(d_pkt.data, sizeof(d_pkt.data), stdin);
104                 d_pkt.sequence_number = 1;
105                 if((sendto(sock, &d_pkt, sizeof(d_pkt), 0, (struct sockaddr*)
    &other, s_len)) == -1){
106                     die("sendto()");
107                 }
108                 alarm(5);
109                 state = 3;
110                 break;
111
112             case 3:
113                 FD_SET(sock, &listen_for);
114                 select(sock + 1, &listen_for, NULL, NULL, &timer);
115                 if(FD_ISSET(sock, &listen_for)){
116                     if((recvfrom(sock, &a_pkt, sizeof(a_pkt), 0, (struct
    sockaddr*) &other, &s_len)) == -1){
117                         die("recvfrom()");
118                     }
119                     if(rand()%100 == 0){
120                         state = 3;
121                         break;
122                     }
123                     if(a_pkt.sequence_number != 1){
124                         break;
125                     }
126                     printf("Received ack packet with sequence number %d\n",
    a_pkt.sequence_number);
127                     state = 0;
128                     break;
129                 }
130                 else{
131                     state = 2;
132                     break;
```

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

    void sig_handler(int signo){
        state -= 1;
    }
```