

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <arpa/inet.h>
6  #include <sys/socket.h>
7  #include <time.h>
8  #include <sys/time.h>
9  #include <sys/select.h>
10
11 #define BUFFER_SIZE 512
12 #define PORT 8882
13
14 typedef struct data_packet{
15     int sequence_number;
16     char data[BUFFER_SIZE];
17 } Data_packet;
18
19 typedef struct ack_packet{
20     int sequence_number;
21 } Ack_packet;
22
23 void die(char* error_message){
24     perror(error_message);
25     exit(1);
26 }
27
28 int main(){
29     time_t t;
30
31     srand((unsigned) time(&t));
32
33     struct sockaddr_in other;
34     int sock, i;
35
36     char message[BUFFER_SIZE];
37     Data_packet d_pkt;
38     Ack_packet a_pkt;
39
40     if((sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0){
41         die("socket()");
42     }
43
44     struct timeval timer;
45     timer.tv_sec = 5;
46     timer.tv_usec = 0;
47     fd_set listen_for;
48     FD_ZERO(&listen_for);
49
50
51     memset((char*) &other, 0, sizeof(other));
52     other.sin_family = AF_INET;
53     other.sin_port = htons(PORT);
54     other.sin_addr.s_addr = inet_addr("127.0.0.1");
55
56     int state = 0;
57     int s_len = sizeof(other);
58
59     while(1){
60         switch(state){
61             case 0:
62                 printf("Enter message (0):\n");
63                 fgets(d_pkt.data, sizeof(d_pkt.data), stdin);
64                 d_pkt.sequence_number = 0;
65                 if(sendto(sock, &d_pkt, sizeof(d_pkt), 0, (struct sockaddr*)
&other, s_len) == -1){
66                     die("sendto()");
67                 }
68                 state = 1;

```

```

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

```