1. **Problem solving:**

The flow of this project is as follows:

Basic info:

udpserver:
ask user to enter the group name to be created
ask user to enter the maximum number of clients can be joined to the group

udpclient:
ask user to enter the group name it wants to join
ask user to choose the option it wants to request

Next:

Then:
Udpclient: send information in the form: group name + "$" + "option"--->Udpserver

Udpserver: split the info received from udpclient: group name, request(according to the special character '$')

Check if the group name received from udpclient matches the group name created before:
If matches, get the request join and the array has space to join the array:
Then add the udpclient to the array of client which is of maximum size: send "Positive" message to udpclient
Else(exception handling):
if group name does not match: send "Invalid" message to udpclient
if the array is full, i.e. reach the maximum number of clients can be joined: send "Overflow" message to udpclient

*if join successfully, then create a new thread for both udpserver and udpclient(**using pthread_create**):*
for udpserver: main thread for getting the user input and multicast the message to udpclient, new thread for receiving message from udpclient
for udpclient: main thread for getting the user input and send the request "quit" to udpserver, new thread for receiving message from udpserver

Basic idea for verifying group name and request is:

If udpserver creates the group name as "ubuntu", and udpclient enter the same group name and choose the request "join", then udpclient sends the message as "ubuntu$join"

When udpserver gets the message, it splits the message into "ubuntu", "join", then verified the pair.

The code I made for pthread is as follows(in udpserver):

```
//thread1 executes Listening function
int s = pthread_create(&thread1, NULL, listening, &args);
if(s == 0){
    printf("New thread created!\n\n");
}
else{
    printf("New thread failed to create!\n\n");
}
```

For pthread in udpclient:

```
int s = pthread_create(&thread1, NULL, ReceiveMessage, &args);
if(s == 0){
    printf("New thread created!\n\n");
}
else{
    printf("New thread failed to create!\n\n");
}
```

2. **Data Structure:**

The data for this project mainly includes:

1) client list:

Because every client is uniquely identified as the data type: struct sockaddr_in udpclient, we create an array of struct sockaddr_in to store the list of clients:

```
struct sockaddr_in clientList[maxSize];
```

2) the arguments of subroutine of new thread, because we have multiple arguments to pass to the subroutine function, we use struct to form:

```
struct argsThread{
  int udpSocket;
  char groupName[MAXBUF];
  int *currSize;
  int *maxSize;
  struct sockaddr_in* clientList;
};
```

1. **User define function:**

I defined multiple functions in udpserver.c and udpclient.c file:

1) *The declaration of these functions in udpserve.c are as follows:*

```
void showStatus(struct sockaddr_in clientList[], int currSize, int maxSize);//show all the current client list info: IP address and port number
void* listening(void* arg); //receiving message from other clients
void printAddrInfo(struct sockaddr_in udpClient);//print the corresponding IP address and port number of client
```

2) *The declaration of these functions in udpclient.c are as follows:*

void ReceiveMessage(void *args);//receive message from udpserver.c*

void SendMessage(int updSocket, struct sockaddr_in udpServer, char groupName[MAXBUF]);//send message to udpserver.c

2. Screen shots of outputs:

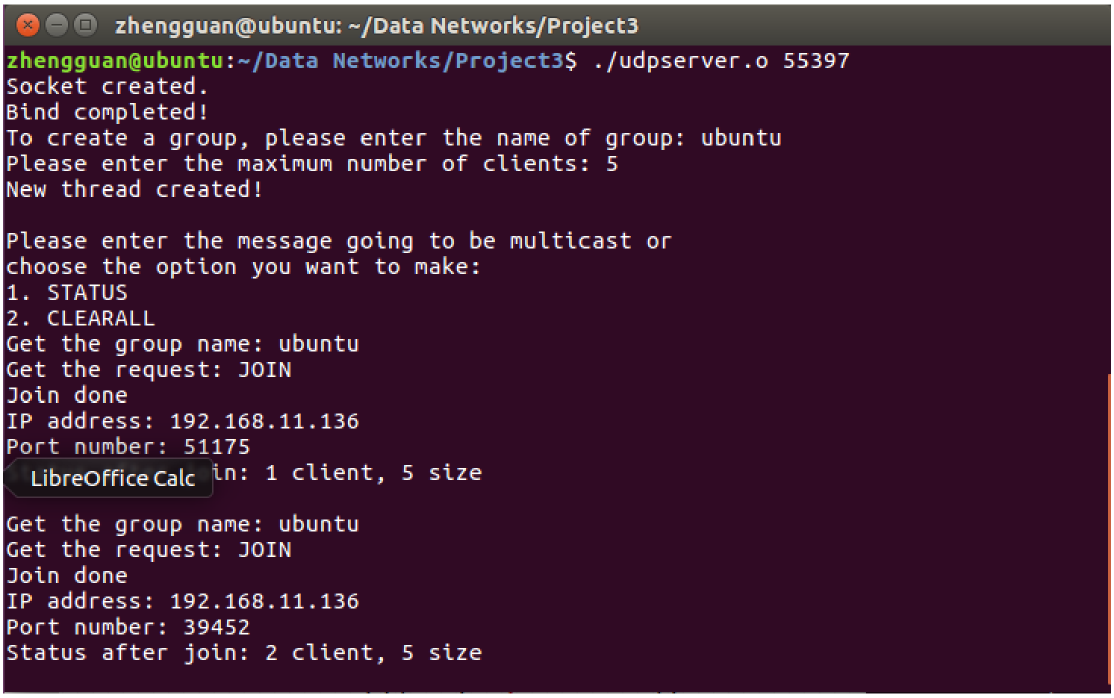
Group name: ubuntu

Maximum size of client: 5

Port number: 55397

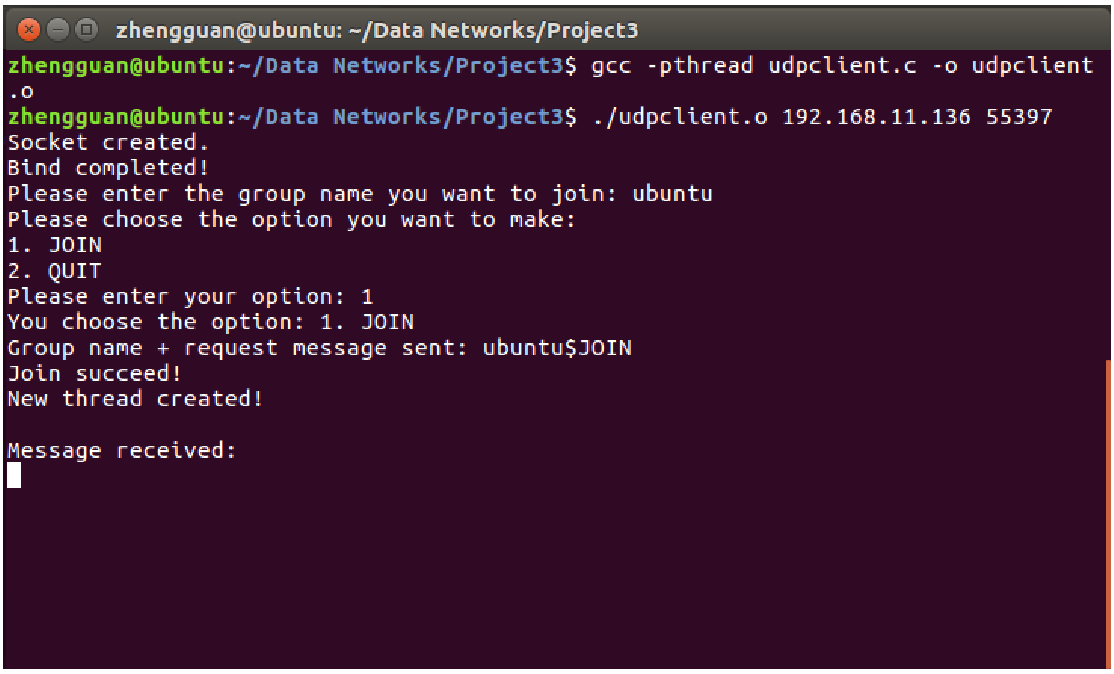
2.1 Screenshot: 2 clients to connect

udpserver:



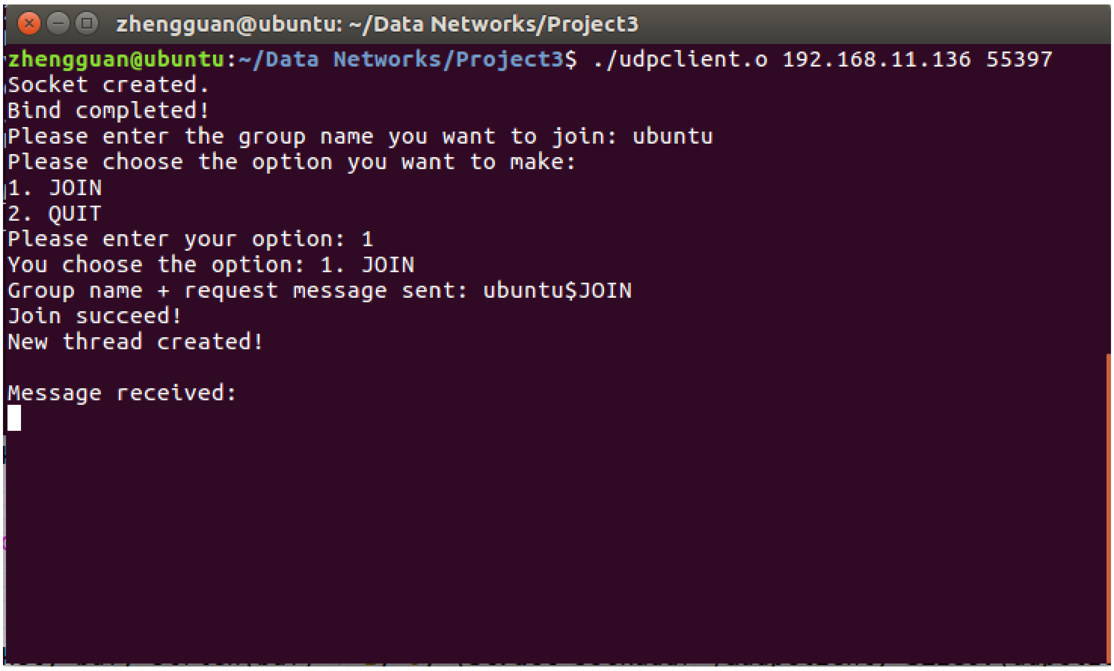
```
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpserver.o 55397
Socket created.
Bind completed!
To create a group, please enter the name of group: ubuntu
Please enter the maximum number of clients: 5
New thread created!

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 51175
                in: 1 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 39452
Status after join: 2 client, 5 size
```

udpclient:



```
zhengguan@ubuntu:~/Data Networks/Project3$ gcc -pthread udpclient.c -o udpclient
.o
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
```

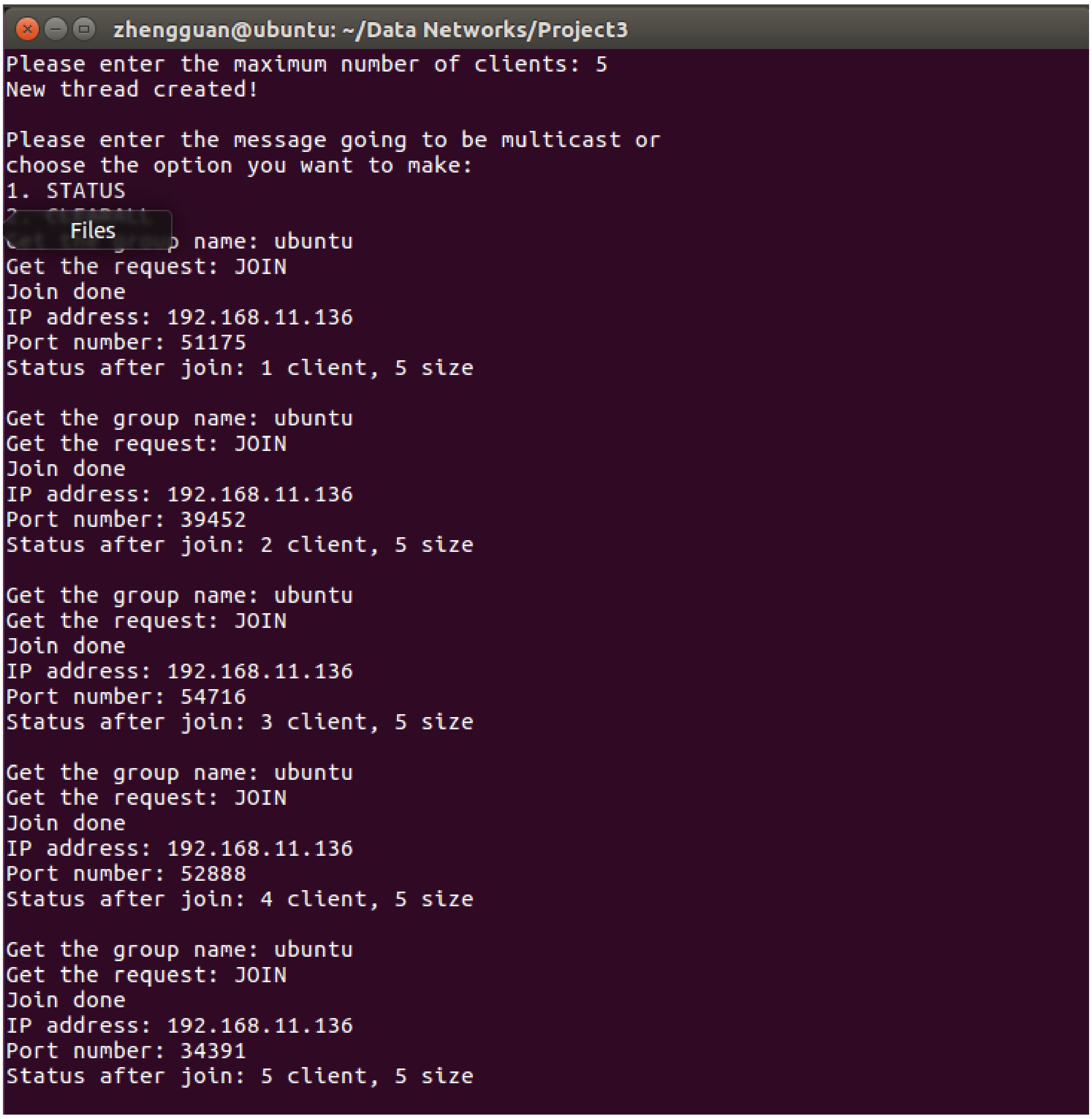
2nd udpclient:



```
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
```

2.2 Screenshot: 5 clients to connect

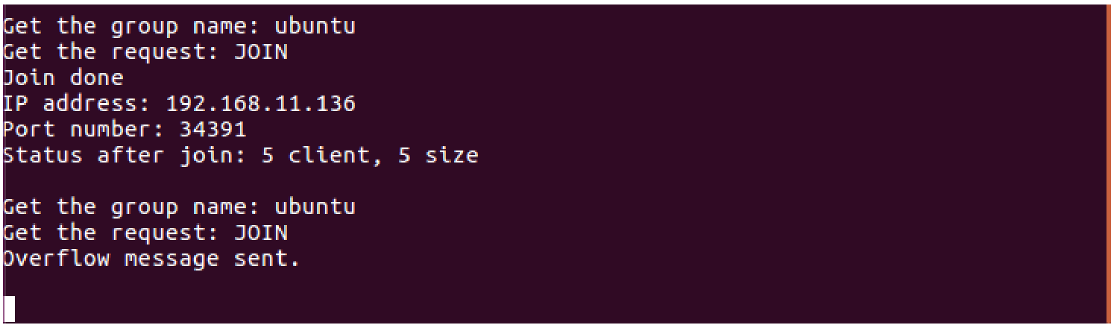
udpserver:



```
Please enter the maximum number of clients: 5
New thread created!

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. [illegible]
Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 51175
Status after join: 1 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 39452
Status after join: 2 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 54716
Status after join: 3 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 52888
Status after join: 4 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 34391
Status after join: 5 client, 5 size
```

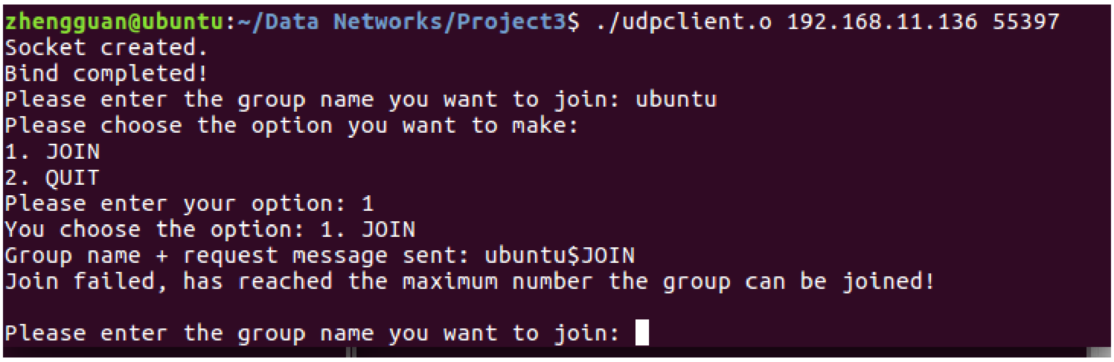
2.2 Screenshot: 6th client wants to join(i.e. overflow)

udpserver:



```
Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 34391
Status after join: 5 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Overflow message sent.
```

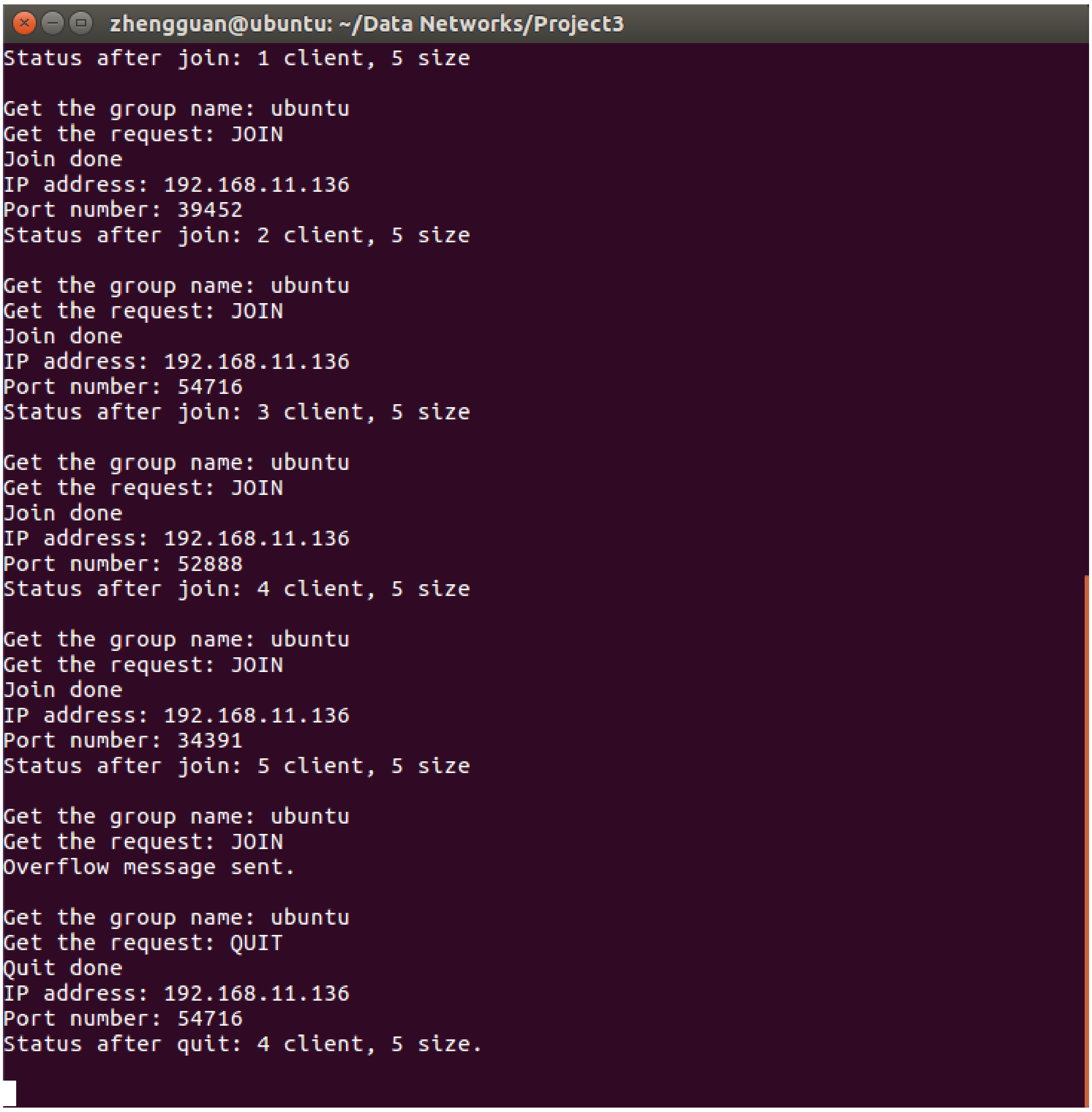
udpclients:



```
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join failed, has reached the maximum number the group can be joined!

Please enter the group name you want to join: 
```

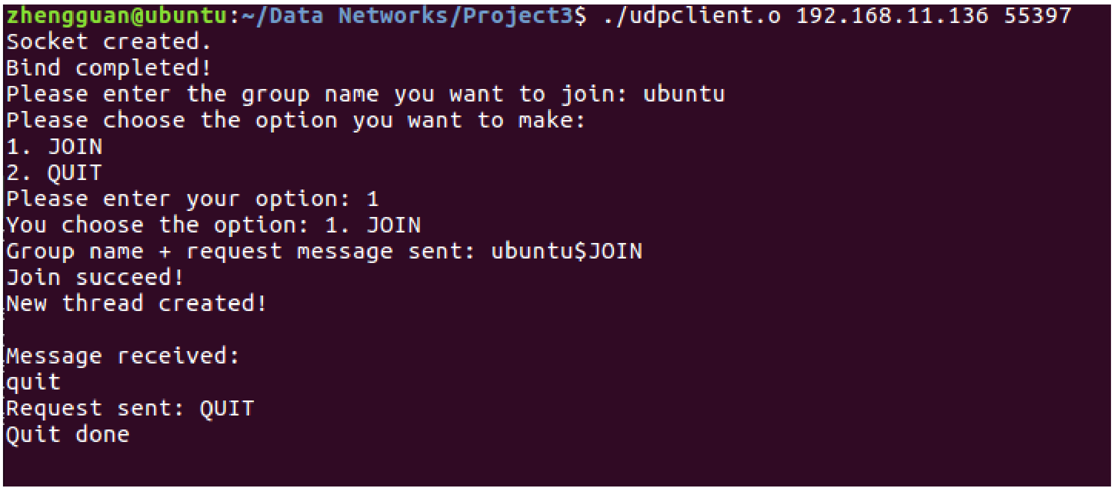
2.3 Screenshot: one udpclient quit

udpserver:



```
Status after join: 1 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 39452
Status after join: 2 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 54716
Status after join: 3 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 52888
Status after join: 4 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 34391
Status after join: 5 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Overflow message sent.

Get the group name: ubuntu
Get the request: QUIT
Quit done
IP address: 192.168.11.136
Port number: 54716
Status after quit: 4 client, 5 size.
```

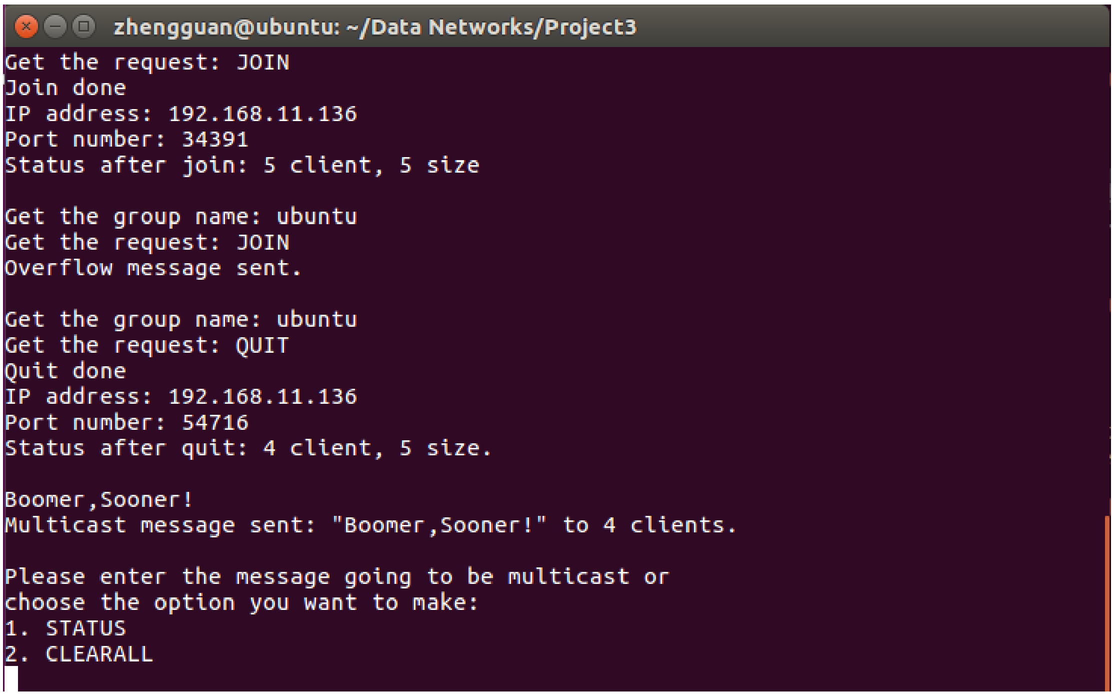
udpclient:



```
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
quit
Request sent: QUIT
Quit done
```

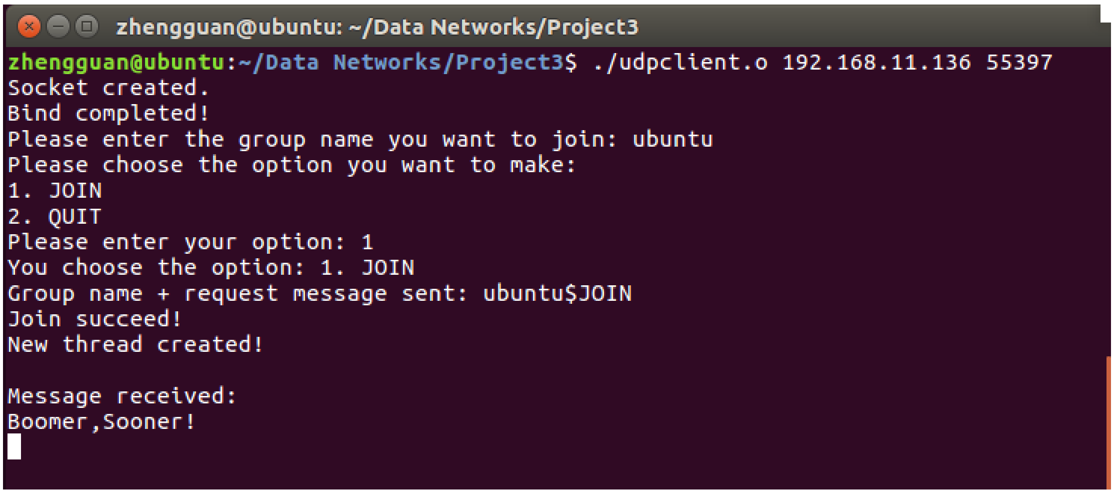
2.4 multicasting, send message "Boomer, Soomer!"

udpserver:



```
Get the request: JOIN
Join done
IP address: 192.168.11.136
Port number: 34391
Status after join: 5 client, 5 size

Get the group name: ubuntu
Get the request: JOIN
Overflow message sent.

Get the group name: ubuntu
Get the request: QUIT
Quit done
IP address: 192.168.11.136
Port number: 54716
Status after quit: 4 client, 5 size.

Boomer,Sooner!
Multicast message sent: "Boomer,Sooner!" to 4 clients.

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
```

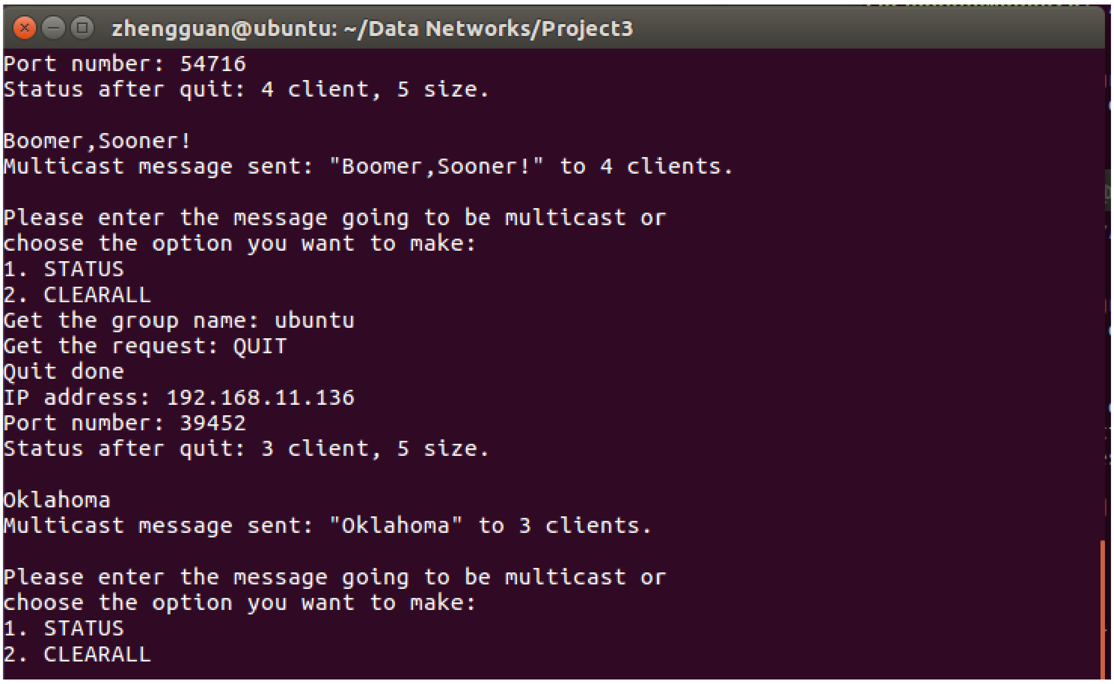
udpclient:



```
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
Boomer,Sooner!
```

2.5 one client choose to quit, and udpserver multicast "Oklahoma"

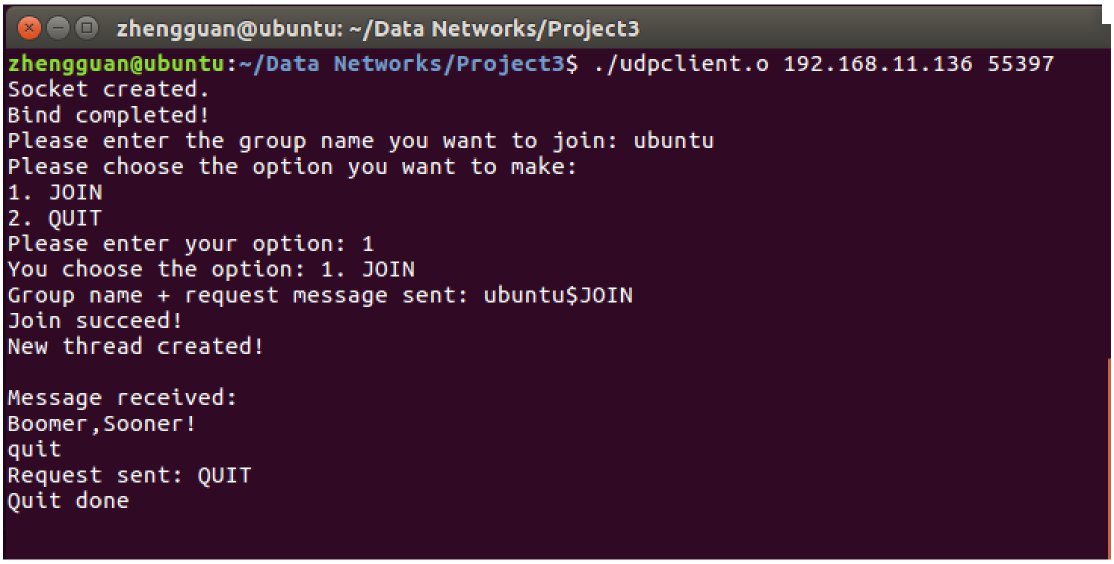
udpserver:



```
Port number: 54716
Status after quit: 4 client, 5 size.

Boomer,Sooner!
Multicast message sent: "Boomer,Sooner!" to 4 clients.

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
Get the group name: ubuntu
Get the request: QUIT
Quit done
IP address: 192.168.11.136
Port number: 39452
Status after quit: 3 client, 5 size.

Oklahoma
Multicast message sent: "Oklahoma" to 3 clients.

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
```

Status after quit becomes: 3 clients, 5 size from 4 clients, 5 size

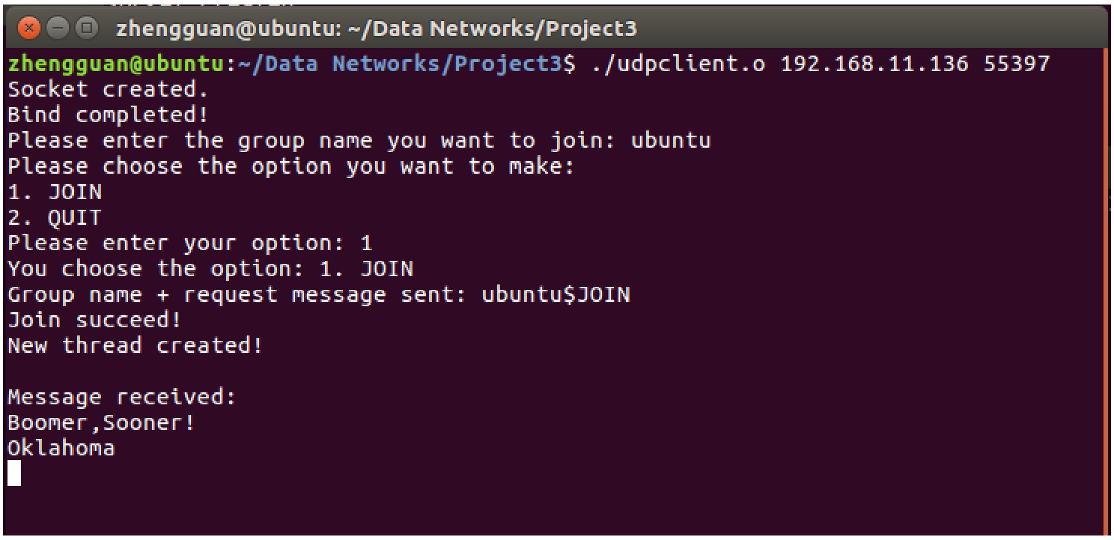
udpclient has quitted:



```
zhengguan@ubuntu: ~/Data Networks/Project3
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
Boomer,Sooner!
quit
Request sent: QUIT
Quit done
```

No new message received

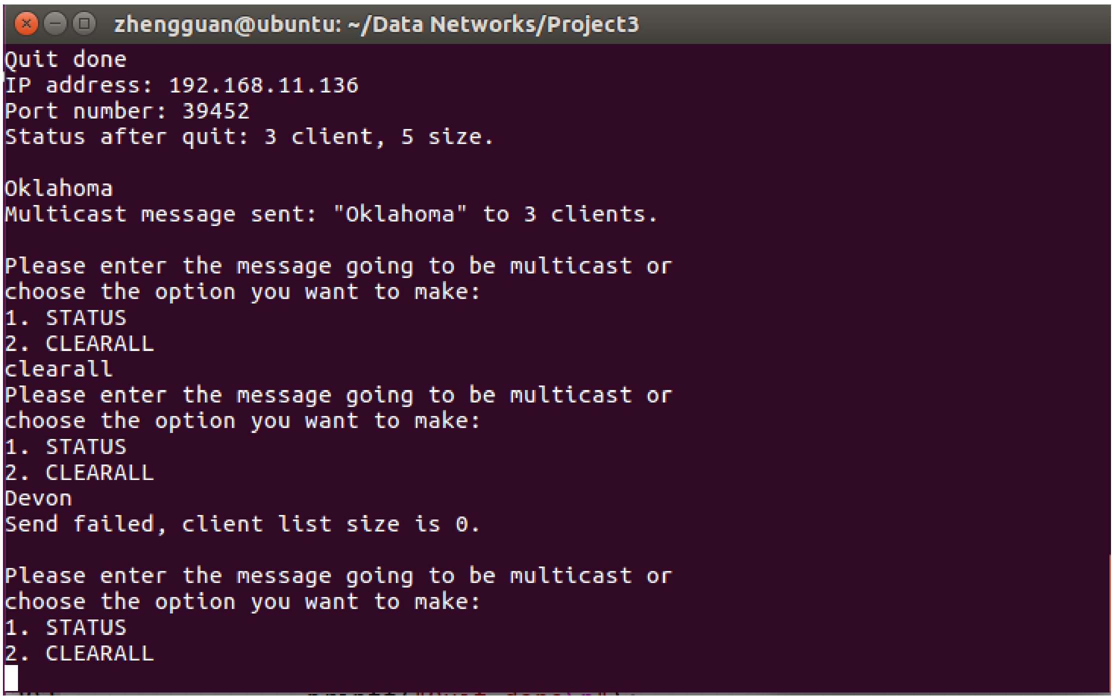
Udpclient still active:



```
zhengguan@ubuntu: ~/Data Networks/Project3
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
Boomer,Sooner!
Oklahoma
```

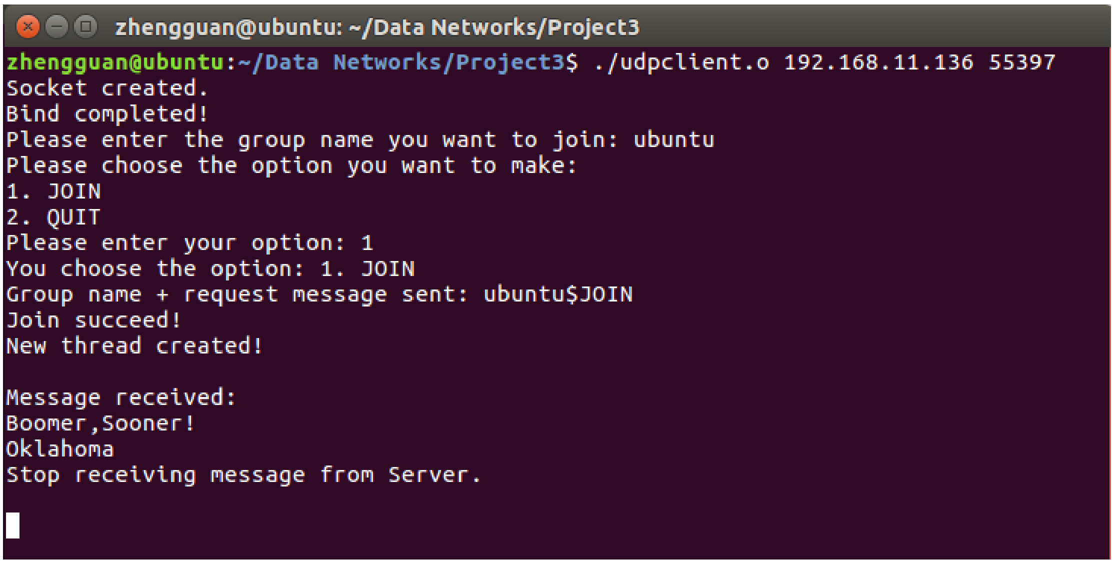
Still can receive the new message.

2.6 Server chooses to option clearall and multicast message “Devon” to clients



```
Quit done
IP address: 192.168.11.136
Port number: 39452
Status after quit: 3 client, 5 size.

Oklahoma
Multicast message sent: "Oklahoma" to 3 clients.

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
clearall
Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
Devon
Send failed, client list size is 0.

Please enter the message going to be multicast or
choose the option you want to make:
1. STATUS
2. CLEARALL
```

udpclients:



```
zhengguan@ubuntu:~/Data Networks/Project3$ ./udpclient.o 192.168.11.136 55397
Socket created.
Bind completed!
Please enter the group name you want to join: ubuntu
Please choose the option you want to make:
1. JOIN
2. QUIT
Please enter your option: 1
You choose the option: 1. JOIN
Group name + request message sent: ubuntu$JOIN
Join succeed!
New thread created!

Message received:
Boomer,Sooner!
Oklahoma
Stop receiving message from Server.
```