数据通信与计算机网络
网络聊天室——设计文档

邹天韵：16300240022

2019 年 6 月 15 日

目录

1 概述

聊天室软件由四个主要文件构成，分别是client.py, Servers.py, ClientMainUI.py和ServerMainUI.py。client.py文件包含了Client类，Room类和ClientWindowDlg类三个类，控制着客户端的运行。servers.py文件包含了Server类和ServerWindowDlg类两个类，控制着服务器端的运行。ClientMainUI.py实现了客户端的图形化窗口，ServerMainUI.py实现了客户端的图形化窗口。

2 总体设计

使用了TCP和UDP连接。

启动服务器的时候开启服务器端TCP socket并加入监听列表，启动客户端的时候开启一个TCP socket，并与服务器的TCP socket相连接，当客户端下线的时候TCP连接断开。这样设计可以维护在线用户的信息，使得用户可以不加入任何聊天室而维持在线状态。

服务器端新建一个聊天室的时候开启一个UDP socket并将该socket和IP地址绑定，并加入监听列表。

用户加入某个聊天室的时候开启一个UDP socket并将该socket和自己的IP地址绑定，同时通过匹配相同的聊天室名称，将加入的聊天室的IP和UDP socket以及用户的IP和UDP socket一同存放进Room类的一个对象里。

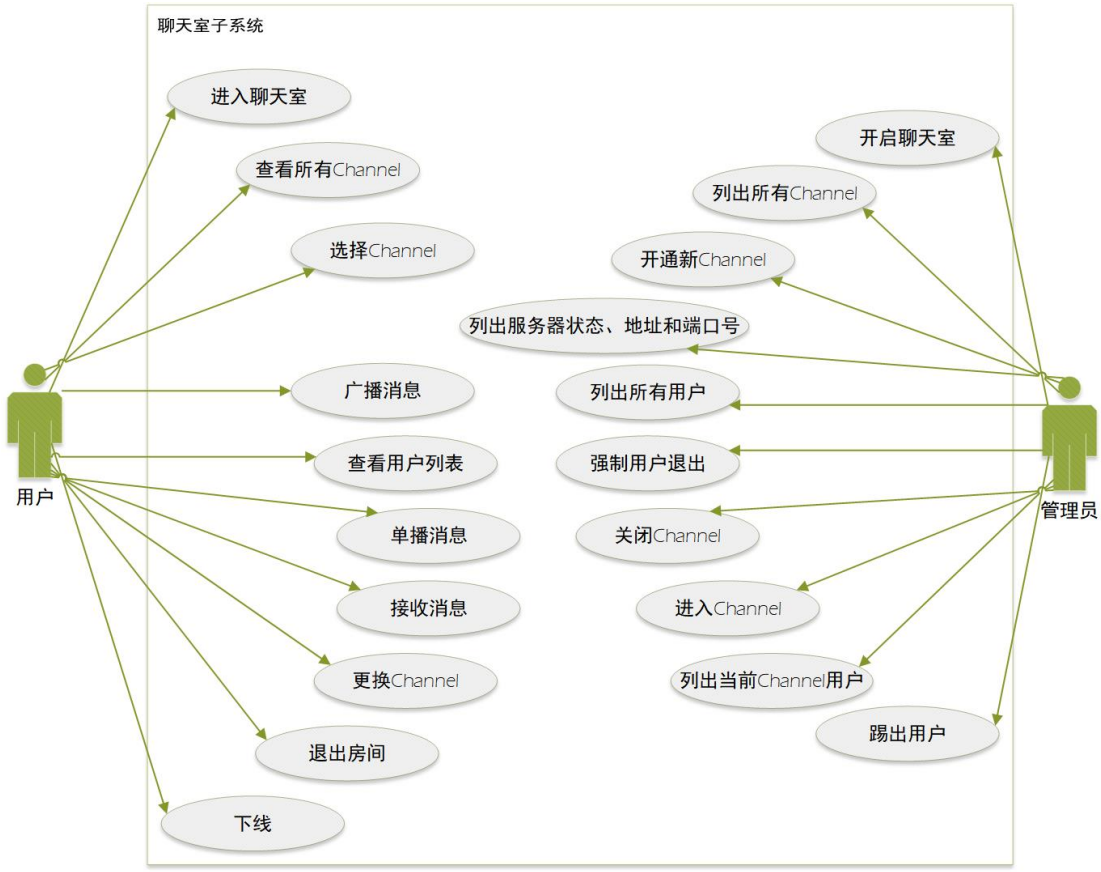

图 1: 聊天室用例图

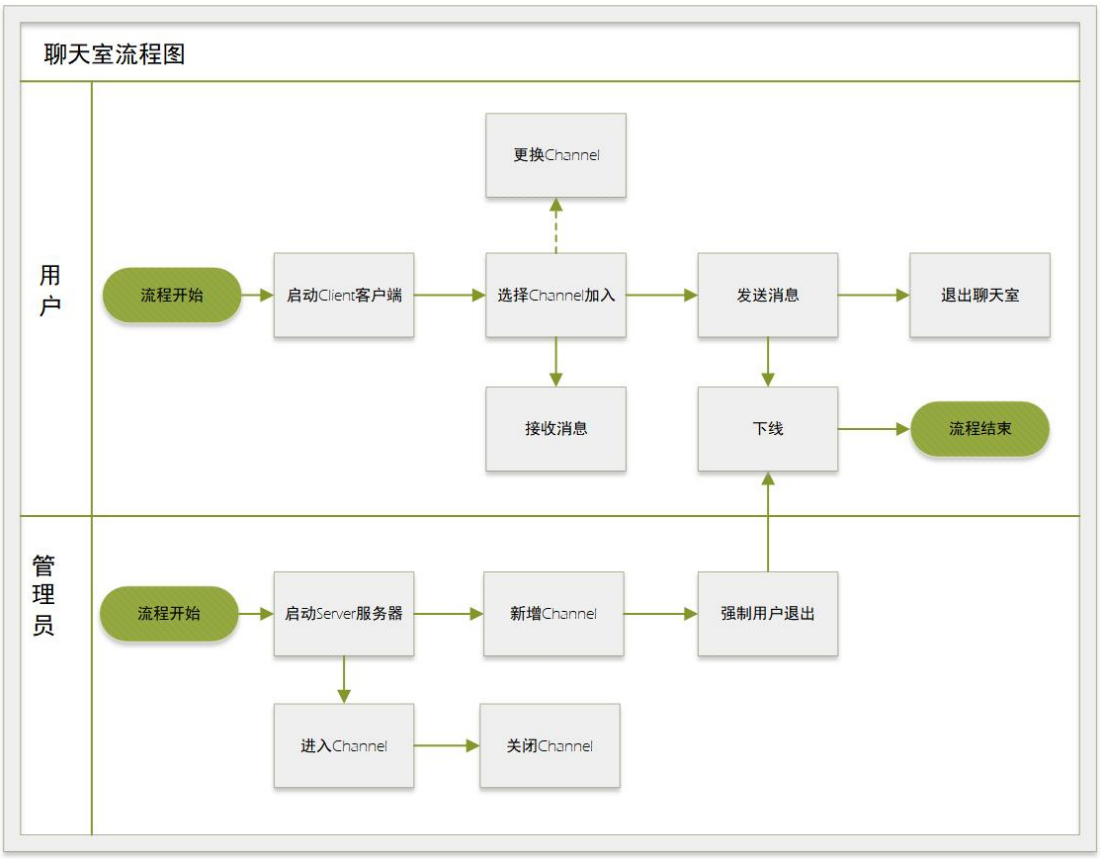

图 2: 聊天室流程图

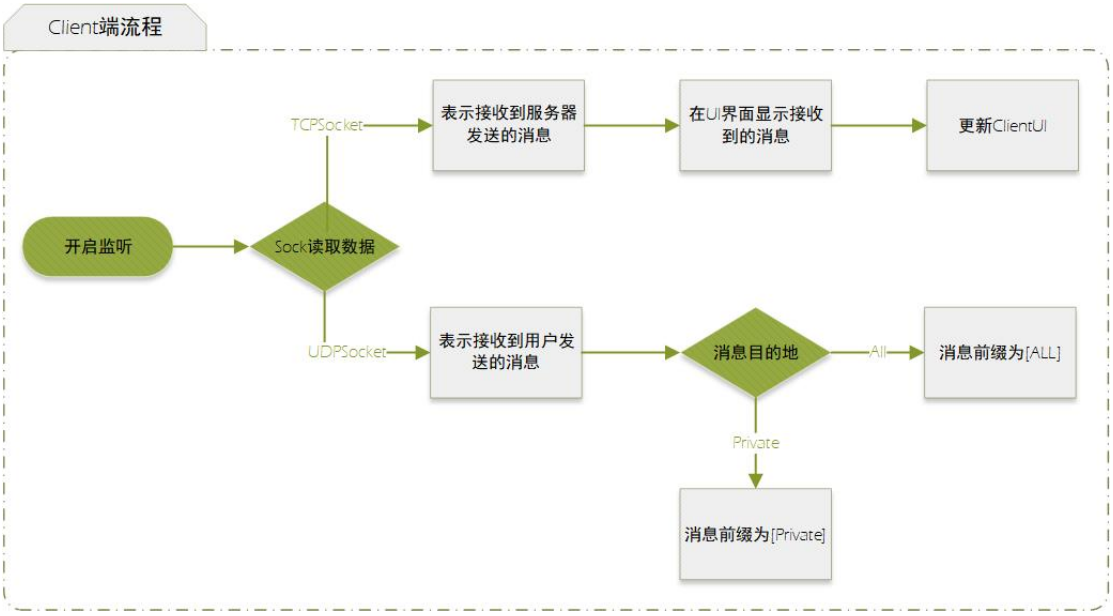

图 3: 客户端监听流程图

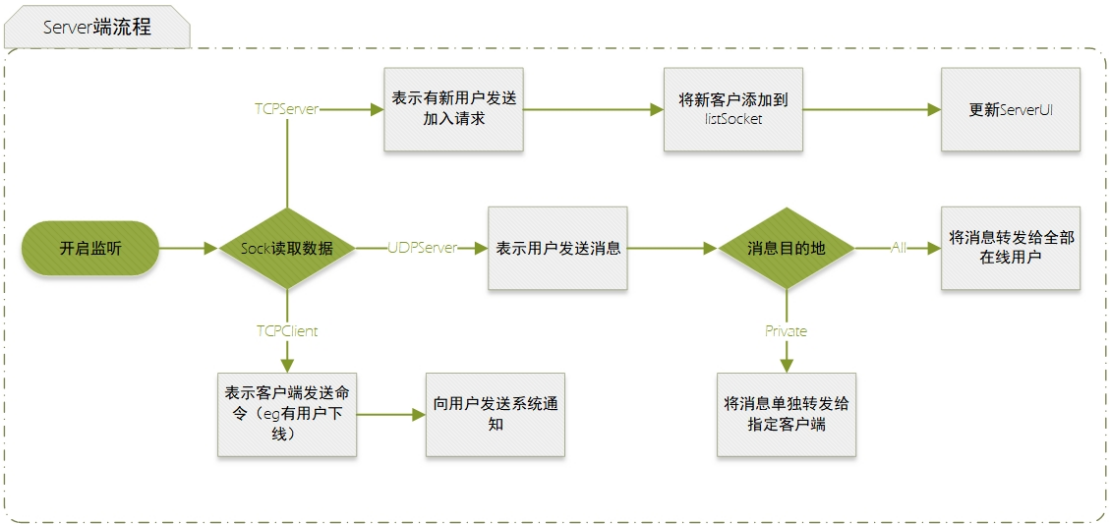

图 4: 服务器端监听流程图

3 协议设计

变长json作为协议体。json使用明文文本编码，可读性强、易于扩展、前后兼容、通用的编解码算法。json协议体为协议提供了良好的扩展性和兼容性。

使用send()进行发送的时候，Python将内容传递给系统底层的send接口,也就是说，Python并不知道这次调用是否会全部发送完成，比如MTU是1500，但是此次发送的内容是2000，那么除了包头等等其他信息占用，发送的量可能在1000左右，还有1000未发送完毕。

sendall()是对send()的包装，它会自动判断每次发送的内容量，然后从总内容中删除已发送的部分，将剩下的继续传给send()进行发送；

Head	From		To		channel	Data
CHANNELSLIST	System Message	0	all			
USERLIST	IP	port	IP	port		
EXIT SERVER						

图 5: 协议内容及填入选项

Head	sp	USERLIST		cr	lf
From	sp	127.0.0.1	5507	cr	lf
To	sp	127.0.0.1	4106	cr	lf
channel	sp	Channel1		cr	lf
Data	sp	你的PJ写的怎么样啦?		cr	lf

图 6: 报文举例

4 类的详细设计

4.1 Client端

4.1.1 Client类

表 1: Client类属性

属性名	属性说明
MainWindow	主窗口
MainFrame	UI界面
localIP	本机IP
host	服务器IP
port	服务器port
current_channel_name	当前聊天室的名字
udpSocket	当前聊天室的port
room	当前聊天室对象
socket_list	监听列表
channels	存放所有聊天室的列表
tcpSocket	当前用户的TCP socket

表 2: Client类函数

函数名	函数说明
run()	启动监听，当接收到TCP socket的消息表示接收到服务器端发来的系统消息，当接收到UDP socket的消息表示接收到了服务器端转发的其他用户发送的消息。
getList()	发送给服务器‘GET’消息，表示接受到服务器发送的消息。
updatechannelsList()	更新聊天室列表。
leaveRoom()	用户退出房间。关闭该用户在这个房间的udp socket，并从监听列表里删除。
enterRoom()	用户进入房间。先退出原来所在房间，再进入新的房间。进入房间时开启一个UDP socket并将该socket和自己的IP地址绑定，同时通过匹配相同的聊天室名称，将加入的聊天室的IP和UDP socket以及用户的IP和UDP socket一同存放进Room类的一个对象里。
exitAPP()	用户退出聊天室软件，关闭TCP socket。

4.1.2 Room类

表 3: Room类属性

属性名	属性说明
name	房间名
socket	房间socket
server	维护server （的IP, port）属性对

表 4: Room类函数

函数名	函数说明
sendMessage()	接收并解析客户端发送的消息，将用户发送的消息发送给server端

4.1.3 ClientWindowDlg类

表 5: ClientWindowDlg类属性

属性名	属性说明
signal	信号，用于连接到信号槽，接收GUI的信号。
ui_Window	UI界面。
client	当前客户端的Client对象。
room	当前房间的对象。

表 6: ClientWindowDlg类函数

函数名	函数说明
closeEvent()	调用client.exitAPP()函数控制客户端退出软件。
MessageBox()	用于提示用户已经断开与聊天室的连接的对话框。
sendMessage()	通过信号槽从GUI接收用户发送的聊天内容，并解析发送给Room对象。
leaveRoom()	调用client.leaveRoom，实现用户离开房间。
clickUserList()	当用户双击userList中的某个用户时可以实现私聊，该函数用于完成此项任务。
enterRoom()	调用client.enterRoom(channel)，实现用户进入聊天室。

4.2 Server端

4.2.1 Server类

表 7: Server类属性

属性名	属性说明
FrameUI	UI界面
localIP	服务器IP
serverPort	服务器port，默认为5000
channels	存放所有channel的信息，包括当前聊天室的所有活跃用户信息、聊天室名称、聊天室的udpServer
udpServer	存放channel的IP和udp port
userList	存放所有客户端的IP和tcp port
listSocket	监听列表

表 8: Server类函数

函数名	函数说明
run()	启动监听，当接收到tcpServer的消息表示有新用户向服务器发起连接请求，则将该用户添加到listSocket和userList中。当接收到UDP socket的消息表示接收到了用户发送的消息，则当该消息为“all”类型，服务器端将消息广播给所有用户，当该消息为“private”类型，则将该消息单独发给指定用户。当接收到tcpClient的消息表示接收到了用户端发送的命令，如某用户离开聊天室，此时服务器端会向其他用户发送系统通知。
updateUserListUI()	更新GUI用户列表。
updateChannelListUI()	更新GUI聊天室列表。
buidServer()	启动服务器。开启服务器端TCP socket并加入监听列表，启动客户端的时候开启一个TCP socket，并与服务器的TCP socket相连接，当客户端下线的时候TCP连接断开。这样设计可以维护在线用户的信息，使得用户可以不加入任何聊天室而维持在线状态。
Openchannel()	服务器端新建一个聊天室的时候开启一个UDP socket并将该socket和IP地址绑定，并加入监听列表。
updateUserINChannel()	更新聊天室中的活跃用户。
kickOut()	将用户踢出房间，并给其他在线用户发送消息。
userExit()	用户下线，并将其从listSocket和userList中移除。
roomExit()	关闭聊天室。
close()	关闭server，断开server的TCP连接

4.2.2 ServerWindowDlg类

表 9: ServerWindowDlg类属性

属性名	属性说明
ui_Window	UI界面
server	当前服务器的Server对象

表 10: ServerWindowDlg类函数

函数名	函数说明
closeEvent()	调用client.exitAPP()函数控制客户端退出软件
NewChannel()	调用client.openChannel新开启一个聊天室。
enterChannel()	接受通过信号槽传递的双击操作，实现管理员进入房间。
leaveChannel()	接收通过信号槽传递的点击按钮操作，实现管理员离开当前聊天室。
userKickOut()	通过调用client.kickOut()函数，实现将用户踢出聊天室。
userForceExit()	通过调用client.userExit()函数，实现强制用户下线。
ChannelClose()	通过调用client.roomExit()函数，实现关闭聊天室操作。