

计算机网络课程设计实验

基于中央定位服务器的 P2P 网络聊天系统设计

自 66 林文镔 2016011503

目录

1	需求分析	2
2	总体设计	2
3	详细设计	2
3.1	中央服务器连接模块	2
3.2	客户端通信模块	3
3.2.1	本地服务器建立	3
3.2.2	协议设计	3
3.2.3	文本消息发送	4
3.2.4	文件消息发送	4
3.2.5	图片发送发送	4
3.2.6	语音消息发送	4
3.2.7	建群消息发送	4
3.2.8	接收方处理	4
3.3	后端数据库模块	5
3.4	前端用户界面模块	5
3.4.1	LoginFrame	5
3.4.2	MianFrame	6
3.4.3	MessagePanel	7
3.4.4	ContactLabel	7
3.4.5	AddGroupDialog	7
3.4.6	AudioDialog	8
3.4.7	EmojiDialog	8
4	结果分析	8
4.1	必做内容	8
4.2	选做内容	9
5	总结	9

1. 需求分析

本项目需要实现一个基于中央定位服务器的 P2P 网络聊天系统，该系统应实现账号登录上线/下线、通讯录、P2P 通信、文件传输、群聊、聊天记录查询、语音发送、表情发送等功能，并提供友好的用户界面，保证系统的美观易用。

2. 总体设计

根据需求，本项目需要实现中央服务器连接模块、客户端通信模块、后端数据库模块和前端用户界面模块这四个模块。具体的结构如下：

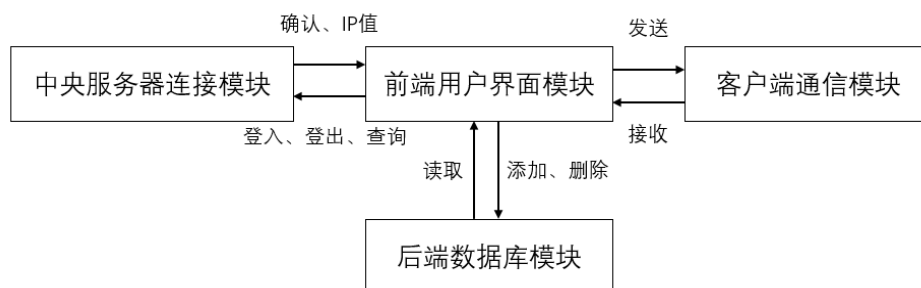


图 1: 总体架构

状态机如下：

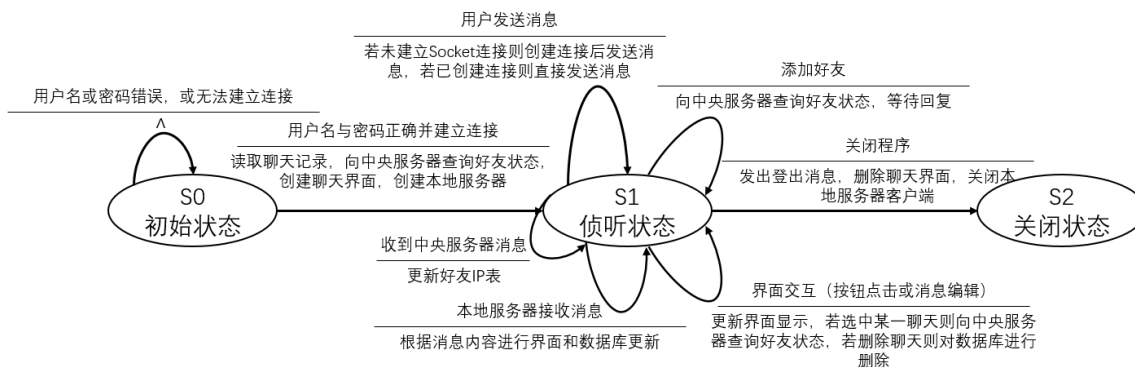


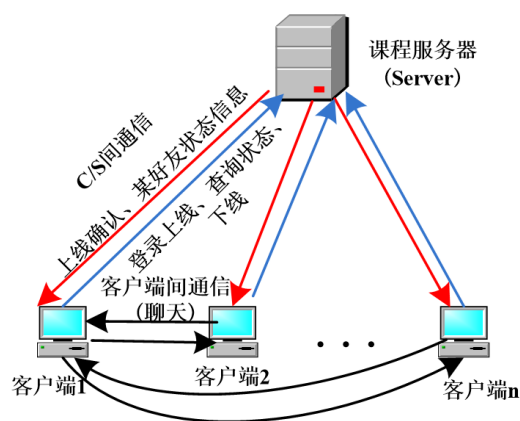
图 2: 状态机

3. 详细设计

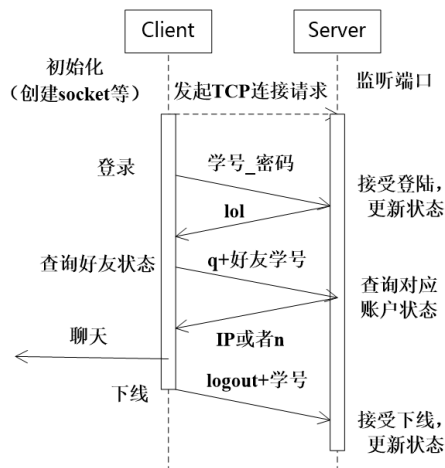
本项目使用 Java 作为编程语言进行开发，使用 jdk1.8.0 进行开发并在 jre1.8.0 下运行测试。

3.1 中央服务器连接模块

服务器连接模块负责进行用户登入、登出以及查询好友 IP，其消息格式已由助教给出，如下：



(a) 总体架构



(b) C/S 通信流程

图 3: 中央服务器通信

在本项目的完成过程中，发现中央服务器对不正确学号的登录和查询也有固定的回复消息，故在基本的登入、登出以及查询 IP 的基础上，本项目还根据服务器回复的其他消息，对用户输入的错误学号给出了必要的提示。

具体到程序上，本项目进行中央服务器连接使用的是 Java 中的 Socket 类，在连接服务器后开启消息接收线程，数据的读写使用的是最基本的 read 和 print 接口。

需要说明的是，本项目中时常需要同时查询多个好友的 IP，但由于消息的接受顺序可能会出现错乱，且服务器发送的 IP 以及不在线消息是不指明学号的，因此程序中仅在接收到一个来自服务器的回复后才查询下一个学号。

3.2 客户端通信模块

本项目中，客户端之间的 P2P 通信需要发送的消息类型有：文字消息、文件消息、图片消息、语音消息和建群消息这 5 种。由于本项目中的 P2P 通信具有群聊功能，在协议设计需要保证一对一和一对多的兼容性。

3.2.1 本地服务器建立

在 P2P 通信中，需要每一个终端在完成登录后立即开启本地服务器，进行消息监听，本地服务器基于 ServerSocket 类实现。尽管服务器本身也是可以发送消息的，但为保证程序的有序性，在本项目中，服务器不发送消息，仅负责监听，用户需要向其他终端发送消息时，需要主动发起向对方服务器的连接，而不是通过本地服务器发送消息。

3.2.2 协议设计

在 P2P 通信中，发送的消息需要包含发送方 ID 以及接收方 ID，其中接收方 ID 可能是群聊的 ID。本项目中所谓用户 ID 即用户的学号，而群聊则是将群成员的 ID 用 “,” 分隔后得到的一个长字符串，便于通过群聊 ID 找到群成员。

由于本项目中发送的消息类型共有 5 种，在消息发送时还需要将消息类型一同发送，便于接收方识别解析。而不同的消息后跟随的消息格式可能不同，需要接收方针对不同的消息类型进行处理。

具体而言，在 Java 中的 `DataOutputStream` 类提供了 `writeUTF` 这一接口，使用该接口写入字符串时，将自动在字符串前添加该字符串长度的标识，为读取带来的便利。

不同类型的消息的前缀如下：

消息类型	前缀
文本消息	txt@
文件消息	fil@
图片消息	img@
语音消息	aud@
建群消息	grp@

表 1: 消息前缀表

消息的发送使用的是 Java 的 `Socket` 类，连接到接收方的 `ServerSocket` 后即可进行传输。

3.2.3 文本消息发送

发送文字消息仅需要在消息前缀的基础之后，使用 `writeUTF` 接口发送一段字符串即可。

3.2.4 文件消息发送

文件消息的发送需要首先发送文件长度、文件名以及文件内容，分别使用 `writeInt`、`writeUTF` 和 `write` 接口，文件内容以字节形式进行发送。

3.2.5 图片发送发送

图片的发送与文件发送本质上是相同的，区别只在于消息的前缀。

3.2.6 语音消息发送

语音发送与文件发送本质上也是相同的，仅以消息前缀进行区分。

3.2.7 建群消息发送

建群消息的发送方与接收方实际上在消息解析中并不需要使用，但为了保证协议的统一性，仍然需要填写消息的发送方和接收方。本项目中，建群消息的发送者即为建群者，接收方填充为“@”字符串，进行占位。消息内容为群聊 ID，使用 `writeUTF` 接口发送。

3.2.8 接收方处理

接收方对信息的处理方式主要分为两种，对于非建群消息，需要将收到的信息封装为自定义的消息类——`Message` 类，该类包含消息的发送者、接受者、消息类型、时间戳、以及消息内容。其中对于文字消息，内容为对应的文本，对于另外三类文件形式的消息，消息内容为文件名，而文件本身将被存储至特定的聊天文件夹中。

对应于消息发送时使用的 `writeUTF`、`writeInt` 和 `write`，接收方使用 `readUTF`、`readInt` 和 `read` 函数进行读取。需要说明的是，考虑到文件传输时，发送方调用的函数接口虽然是将整个文件直接传送，但实际传输时是需要进行分片的，接收方的缓冲区也不是立即被填满的，故接收方需要对数

据进行分片读取。在读取到的字节数小于文件大小时，循环读取缓冲区，将读取到的字节写入文件中。

若消息的发送者并未出现在本机的好友列表中，且接收方 ID 为本机而非群聊，则在通讯录中添加发送者 ID。

而对于建群消息，接收方将在通讯录中添加该群聊 ID。

同时，在通讯录和聊天记录中添加内容时，接收方都将在数据库中写入相关信息，以便后续登录时进行读取。

3.3 后端数据库模块

为实现对通讯录以及聊天记录的存储，本项目使用了数据库进行必要的数据存储，同时为保证程序的绿色运行，需要使用可内嵌至 JVM 的数据库驱动程序，故本项目使用了 Apache Derby¹作为数据库驱动，其工作原理如下：

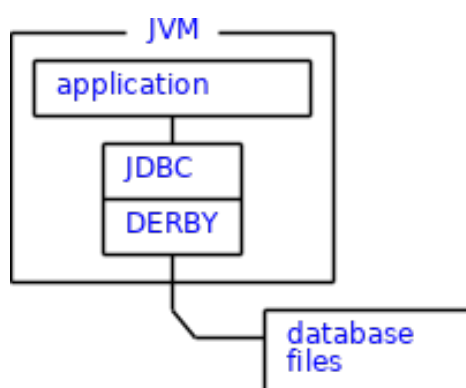


图 4: 嵌入式数据库工作原理

基于此数据库驱动，本项目实现了 DatabaseManager 类，提供后端的数据库维护功能。

本项目中的数据库含有两个表格，一个记录通讯录，即好友和群组的 ID。另一个表格一个记录聊天记录，聊天记录的格式与自定义的消息类 Message 是对应的。

在本项目的 DatabaseManager 类中主要实现了读取通讯录、聊天记录；向通讯录和聊天记录添加内容以及删除内容的接口。

3.4 前端用户界面模块

用户界面方面，本项目共定义了 LoginFrame、MianFrame、MessagePanel、ContactLabel、AddGroupDialog、AudioDialog、EmojiDialog 这 7 个类，以下一一介绍。

3.4.1 LoginFrame

LoginFrame 类继承自 JFrame，是用户登录界面：

¹<https://db.apache.org/derby/>



图 5: 用户登录界面

该界面提供用户名与密码的输入框，要求用户名为合法的学号，密码为“net2018”，当用户选择登录后，向中央服务器查询用户是否存在，存在则进入聊天界面。

3.4.2 MianFrame

MainFrame 类继承自 JFrame，是聊天主界面：

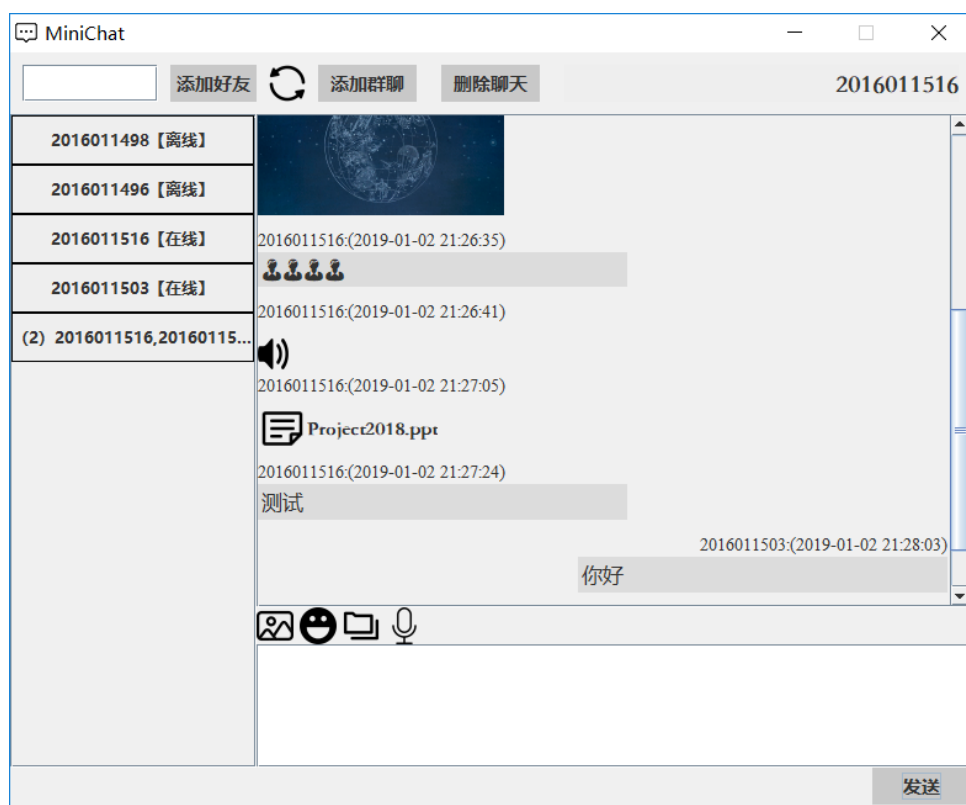


图 6: 聊天主界面

该界面左侧为通讯录，存放好友列表，对于非群组显示好友是否在线，当有未读消息时，对应的 ID 左侧将显示未读消息数目，即图中小括号中的数字表示未读消息数。该部分以 JList 为容器放置 ContactLabel 对象。

界面上方一栏中，从左向右依次为好友学号输入框、添加好友按钮、刷新按钮、添加群聊按钮、删除聊天按钮和当前聊天标签。最左侧的输入框，用于输出需要查找的好友学号，输入学号后点击添加好友按钮即可添加好友。由于程序中不进行好友在线状态的自动刷新，用户可通过点击刷新按钮更新好友在线状态。点击添加群聊按钮后将弹出添加群聊对话框 AddGroupDialog，这部分在后续介绍。点击删除聊天可在通讯录中删除当前聊天，同时清空对应的聊天记录。当前聊天标签显示当前的会话对象。

界面的右侧上部为聊天记录显示部分，可显示不同种类的消息，文本可直接显示，图片在一定的尺寸约束下直接显示，文件将显示文件名，语音将提供播放按钮。聊天记录同样以 JList 为容器，放置的是 MessagePanel 对象。

聊天记录下方的是四个按钮分别为图片、表情、文件、语音的发送按钮。点击文件发送按钮和图片发送按钮将弹出文件选择框，对于图片发送按钮将限制可选择的文件类型。点击语音发送和表情发送将分别弹出 AudioDialog 和 EmojiDialog，此部分在后续介绍。

界面右侧下部为文本编辑区以及文本发送按钮。

3.4.3 MessagePanel

MessagePanel 类继承自 JPanel，为聊天记录中的一项，可根据聊天消息的类型生成不同的显示效果，具体可见聊天主界面图。

3.4.4 ContactLabel

ContactLabel 类继承自 JLabel，为通讯录中的一项，显示好友 ID、在线状态以及未读消息数。点击其中的一项可切换至对应的会话中，界面右上方的当前聊天标签以及聊天记录也将做出相应的更新。

3.4.5 AddGroupDialog

AddGroupDialog 类继承自 JDialog，为添加群聊对话框。据当前通讯录列表，生成一系列可选成员项，用户可通过勾选成员添加群聊，同时程序会判断群聊是否存在，若该群聊为创建则向各个选中的群成员发送建群消息。

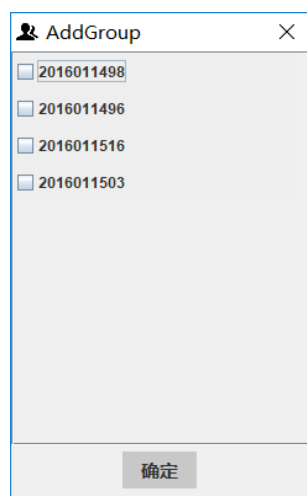


图 7: 添加群聊对话框

3.4.6 AudioDialog

AudioDialog 类继承自 JDialog，为语音发送对话框。点击左上方按钮可开始录音，再次点击则录音结束，同时将录音以 wav 格式存放在特定的聊天记录文件夹中。点击右侧的播放按钮可播放录音，点击取消可删除录音，点击发送则将发送语音消息。



图 8: 语音发送对话框

3.4.7 EmojiDialog

AudioDialog 类继承自 JDialog，为表情发送对话框，每个表情都有对应的 unicode，当选中某个表情时，将在文本编辑区添加对应的表情，每个表情对应的 unicode 存放在 emojis.json 文件中，使用对应的库²，可获取每个表情的 unicode 等信息。最终表情在聊天记录中的显示与编辑时略有不同，这是由于字体的不同，编辑是使用的字体为“Segoe UI Emoji”，而聊天记录中使用的字体为“微软雅黑”，前者对表情显示效果支持更好，而在显示中文时，可能出现一定的乱码，故在聊天记录中使用的是对中文支持更好的“微软雅黑”。

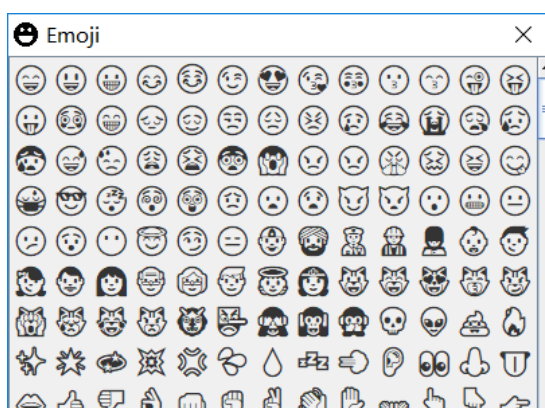


图 9: 表情发送对话框

4. 结果分析

4.1 必做内容

本项目实现了必做功能中的全部内容，测试中可完成大小在 2G 以内的文件传输任务。对于一些突发的好友断线或好友关闭程序重新连接都能够处理，并对异常情况给出提示。

²<https://github.com/vdurmont/emoji-java>

4.2 选做内容

选做内容方面，本项目实现了群聊、聊天记录查询、语音发送、表情发送功能。在要求的这些功能外，实现了图片在聊天记录中的显示以及删除好友的功能。

绿色运行方面，由于本项目使用 Java 进行编程设计，需要通过 eclipse 导出 jar 文件，而 jar 只能在配置了合适版本 Jre 的电脑上才能运行。故在导出 jar 后使用软件 launch4j³将文件导出为 exe 格式的可执行文件，同时绑定了编程中使用的 Jre 环境，最终仅需要点击 exe 文件即可运行。

5. 总结

1. 本次大作业的中使用 Java 的 Swing 组件进行 UI 设计，且直接在代码中进行布局，并未使用辅助工具。总体上，使用体验并不好，或许是我个人没有掌握要领，导致实现一些特定的显示效果上花费了较多的时间。具体来说，而在聊天记录显示中，换行效果较为人性化的 JTextArea 对表情显示支持不好，而对表情支持较好的 JTextPane 对长字符不支持换行，需要修改底层的 StyledEditorKit。此外，还有先前提到的使用“Segoe UI Emoji”字体时，偶然出现中文乱码的问题。
2. 总体上，本次大作业的设计架构自认为是较为清晰的，这也使得实现具体功能时效率较高，而主要的开发时间花费在了对一些细节问题的处理上。例如，好友断线重连或 IP 变化时，对 Socket 的重新构建；添加群聊时，检查群聊是否已存在；取消录音或重新录音时将上一录音文件删除；维护处于群聊中但非好友的通信等等细节问题。
3. 以前接触的 Socket 编程时，对计算机网络内部的运行原理并没有了解，仅仅将 Socket 作为一个封装完好的黑盒。虽然将其视作黑盒在很大程度上也并不影响编程，但有了计算机网络的知识之后，确实能对黑盒内部的工作方式有一定的理解。尤其在本项目实现的文件传输中，分片读取的做法确实需要对 TCP 的传输原理有一定的认识才能理解到其中的精髓。

参考文献

- [1] [Online] Available: <https://docs.oracle.com/javase/tutorial/>
- [2] [Online] Available: https://db.apache.org/derby/papers/DerbyTut/embedded_intro.html
- [3] [Online] Available: <https://coderanch.com/t/556838/java/Transferring-file-file-data-socket>
- [4] [Online] Available: http://www.java2s.com/Tutorials/Java/Swing_How_to/JList/index.htm
- [5] [Online] Available: http://www.java2s.com/Tutorials/Java/Swing_How_to/JTextPane/Wrap_long_words_in_JTextPane.htm

³<http://launch4j.sourceforge.net/>