

# 计算机网络及应用 课程设计实验

## 基于中央定位服务器 P2P 网络聊天系统设计

姓名：钟清扬

学号：2016011481

班级：自 65

日期：2019-1-6

计算机网络及应用 课程设计实验 .....	1
1. 需求分析 .....	3
2. 总体设计 .....	3
2.1 通信功能设计 .....	3
2.2 界面架构设计 .....	4
3. 详细设计 .....	6
3.1 数据报结构设计 .....	7
3.2 登录与“基本信息栏”功能实现 .....	7
3.3“好友标签栏”设计与实现 .....	7
3.4“聊天对话框”设计与实现 .....	7
3.5“特色工具栏”设计与实现 .....	8
3.6 群聊功能设计与实现 .....	8
4.项目总结 .....	10
5.参考文献 .....	10

# 1. 需求分析

设计基于中央服务器的 P2P 网络聊天系统，并设计通信数据报格式以实现大文件发送与群聊功能。在实际生活中应用计算机网络课程内容，实现 QQ/微信的基本功能。

设计需求与功能实现情况如下：

MyChat 功能一览表

项目要求	功能名称	实现情况
必做要求	账号登录上线/下线	☑
	查询好友是否在线	☑
	P2P 通信	☑
	文件传输（10MB 以上）	☑
	友好的用户界面	☑
选做内容	群聊功能	△
	聊天记录查询	☑
	动态表情	☑
	语音发送	
创新内容	聊天双方头像显示	☑
	聊天页面自动切换	☑
	接收消息提示	☑
	绿色运行	☑

△：群聊功能目前处于 Beta 版，暂不能稳定运行

# 2. 总体设计

本项目设计可分为通信功能设计与界面架构设计两部分，以下分别进行简述。

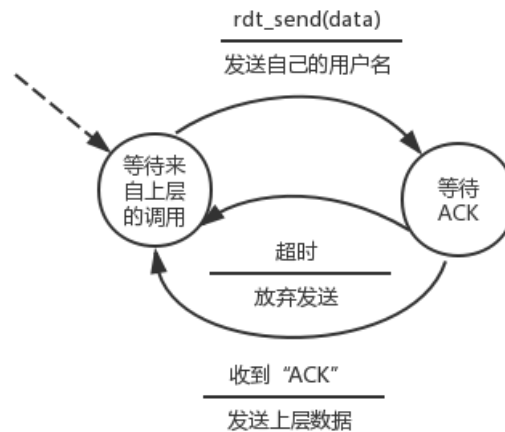
## 2.1 通信功能设计

P2P 是一种用于不同 PC 用户之间，不经过中继设备直接交换数据或服务的技术，允许每个人可以直接连接到其他用户的计算机，并进行文件的交换，而不需要连接到服务器上再进行浏览与下载。程序采用 CS 架构实现用户的上线和下线功能、查询好友的在线状态功能，采用 P2P 架构实现与好友一对一聊天功能、文件传输功能。每次发起 P2P 连接、采用 P2P 发送消息或文件前，程序均会向服务器发起好友 IP 查询，以保证好友在线且 IP 地址有效。

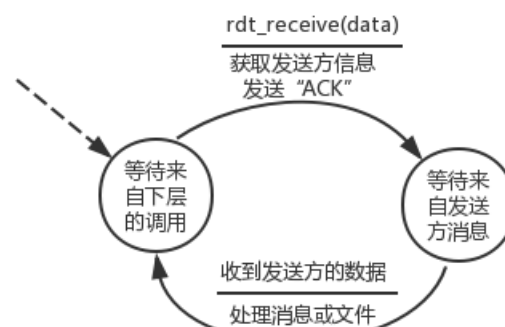
本项目采用 TcpClient 和 TcpListener 完成通信功能 TcpListener 和 TcpClient 类对 Socket 进行了封装。TcpListener 在服务器负责监听和接受客户端的连接请求，TcpClient 类提供可用于 TCP 网络通信的 API。每次发送消息时创建 TcpClient 对象对指定 IP 端口进行连接，调用 GetStream 的方法将发挥一个 NetworkStream 实例，对数据进行读取和写入。

本程序采取为每个用户分配固定的端口的的方法，以“10000”+学号后 4 位作为用户端口号，侦听相应端口号即可获取其他用户向自己发送的信息。调用 BeginAcceptTcpClient 方法异步接受连接，当有新的连接传入时，触发 AsyncCallback 委托绑定的方法，异步接受传入的连接尝试并在收到聊天后单独开线程处理接收信息；同时进入循环监听，重新开始异步操作来接受传入的连接尝试。

每次进行 P2P 通信时，为保证信息可靠传输，发送消息的用户获取好友最新 IP 后，将首先向该 IP 发送自己的用户名声明自己的身份，接收信息的用户收到信息后将回复“ACK”至发起对话的用户，表示信息接收成功；收到 ACK 后，用户将正式开始信息的发送，接收方将根据收到的数据报格式判断收到的是文件、表情还是消息，从而进行相应的处理与显示。



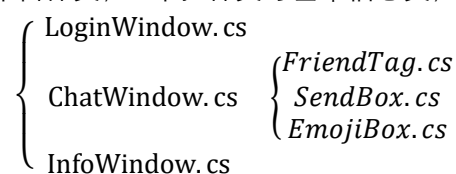
(发送方 FSM)



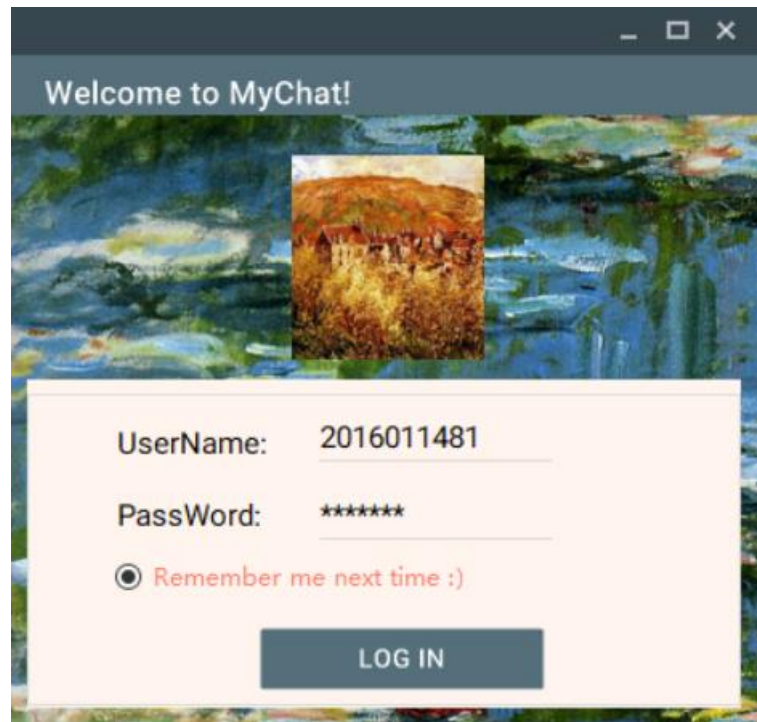
(接收方 FSM)

## 2.2 界面架构设计

本项目设计共设计 3 个窗体类，3 个控件类与基本信息类，总体架构如下：



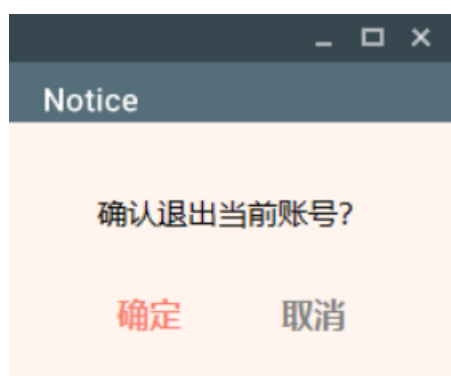
其中 LoginWindow, ChatWindow 与 InfoWindow, 分别用于实现登录/上线功能, P2P 传输等核心功能, 与错误信息提示功能。FreindTag, SendBox, EmojiBox 分别为好友标签, 聊天对话, 与表情栏控件。窗体具体运行说明详见 README。



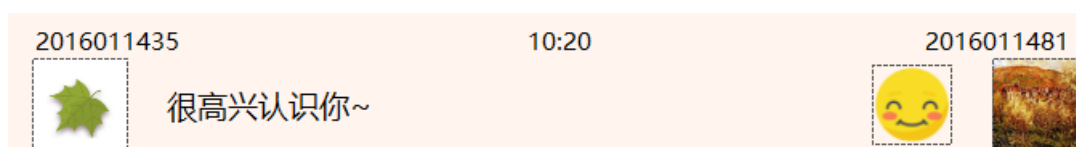
(登录界面 LoginWindow)



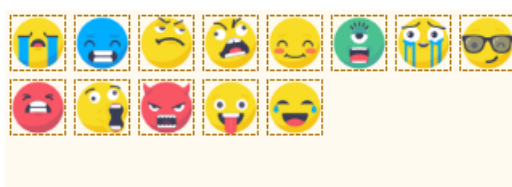
(聊天界面 ChatWindow)



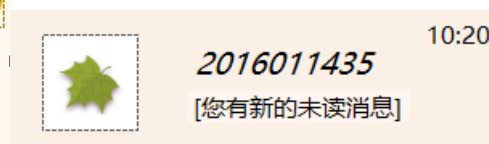
(消息提示框 InfoWindow)



(聊天对话控件 SendBox)



(表情栏控件 EmojiBox)



(好友标签控件 FreindTag)

其余与功能相关的类均在 BasicInfo.cs 中，具体类如下：

```

BasicInfo.cs {
    Class BasicInfo
    Class QueryResult
    Class Friend
    Class GroupTalk
    Enum DialogueType
}

```

其中，BasicInfo 类以静态存储重要的全局变量，如当前用户名、中央服务器 IP 与端口、是否处于群聊状态、当前对话好友等；QuerResult 用于记录查询好友状态时由中央服务器返回的是否在线与 IP 地址等信息；DialogueType 用与区分当前是用户发送的消息还是自己发送的消息以便聊天对话框对对话进行显示；Friend 类时用于处理与 P2P 相关功能的核心类，负责与界面进行信息传输，包括主动/被动添加好友时生成新的好友类、初始化聊天对话框，在聊天对话框内显示当前聊天内容等，当前用户将使用 MyFriendList 列表存储所有好友信息，方便信息的发送、接收与显示；Grouptalk 类主要处理和群聊相关的信息加工，包括报头的添加与拆除。

### 3. 详细设计

界面的具体运行操作已在 Readme 中进行了详细介绍，此处不再赘述，仅对实现原理进行分析与说明。

## 3.1 数据报结构设计

除通信实现时采用了发送“用户名”-发送“ACK”-发送信息的可靠传输模式，及动态请求最新 IP 代替定期更新 IP 列表外，本项目自行设计了 P2P 数据报格式以分别实现消息，文件，表情的传输。

### 消息发送格式

直接发送文字消息即为 P2P 消息传输。

### 文件发送格式

[FileName]+ 文件名+[Length]+ 文件长度+文件内容。

传输文件时每次写入长度为 1024 字节 Packet，文件发送或接收完毕时聊天对话框内显示[文件]+[文件名]信息。

### 表情发送格式

[Emoji]+当前表情编号。

### 群聊发送格式

[Group][Number] +群聊人数+ [Info]+（用户名 1+IP）+ //+（用户名 2+IP）+ // +……+ [Sender] +发送用户名+ [Message] +传输信息。

其中“[]”中内容均为指定关键字，“//”为群聊用户信息分隔符。接收方将根据关键字与分隔符分割报头获取相关信息。

## 3.2 登录与“基本信息栏”功能实现

登录即为基本的请求上线功能，程序将记录登录用户名并在基本信息栏进行显示。程序实现了记录所有新添好友或新登录用户的头像信息的功能。该信息保存在运行程序当前目录下的“./ PortraitList.txt”中。每次出现新用户时，文件内容与全局静态变量中的 PortraitList 列表将进行刷新。程序升级完善后可以实现上传头像与更换头像功能。

## 3.3“好友标签栏”设计与实现

好友标签栏由 FriendLayoutPanel 实现，每个好友类 Friend 生成时均会生成相应的 FriendTag 控件添加到 FriendLayoutPanel。其中点击 Tag 实现聊天对话框切换功能交由委托由 Friend 类实现。

## 3.4“聊天对话框”设计与实现

聊天对话框由 ChatPanel 实现。无论主动或被动添加好友时，均会生成新的好友类，每个好友类 Friend 生成时均会初始化相应的 ChatPanel 控件并添加到主窗体 ChatWindow 中。和好友进行对话时，首先将判断该信息是当前用户发送的信息还是用户接收到的信息，从而

生成相应的 SendBox 空间添加到当前好友的 ChatPanel 中进行显示。其中收到新消息时 ChatPanel 自动切换功能由 Friend 类中的委托实现。

与每个用户单独打开新的对话窗体相比,聊天对话框的主要难点在于需要判断当前对话的好友以保证信息的正确传输,并在接收信息时及时更新当前对话好友,对聊天对话框进行切换。与直接采用富文本框显示聊天对话相比,采用添加控件的方式可以使每一轮对话显示发送者的用户名、头像与发送时间,界面更加友好,图片显示与动态表情显也很容易实现。

## 3.5“特色工具栏”设计与实现

### 发送表情

实际上发送照片或 gif 与发送文件的原理和实现方式完全相同,本项目实现发送动态表情功能时定义新的数据报格式,采用与发送消息相同的形式。

事先将动态表情进行存储,用户点击需要发送的表情时,程序会将信息编码为“[Emoji]+表情编号”形式进行发送,同时聊天对话框对发送的表情进行显示;接收到表情信号时,以同样方式进行解码,从而实现发送动态表情功能。

### 发送文件

除按照数据报格式发送与接收文件外,接收到的文件将保存在当前目录下“./(用户名)”文件夹下,同时聊天对话框将显示[文件]+[文件名]表示信息的传输。程序完善升级后可以添加委托实现点击文件名打开相应存储路径的功能。

### 保存/查询聊天记录

每个好友类 Friend 中均含有 WordList 列表记录当前所有对话内容,点击查询标志时当前对话框内的所有对话将存储在指定位置(详见 Readme),每次保存后最后一次保存聊天内容的位置将会被记录;多次保存同一对话框内的聊天对话,原先的聊天内容不会被多次写入。

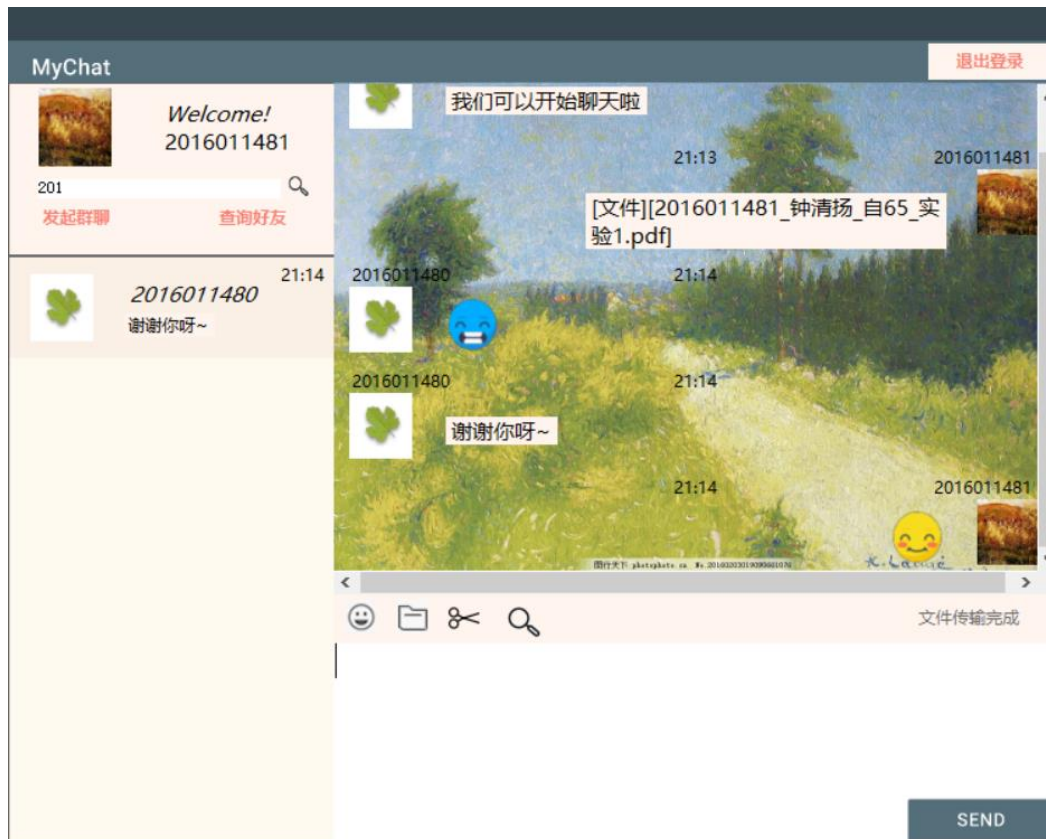
特色工具栏对话效果以 2016011481 与 2016011480 对话为例进行展示,可以看出文件传输、表情发送、消息发送功能良好。(见下页图)

## 3.6 群聊功能设计与实现

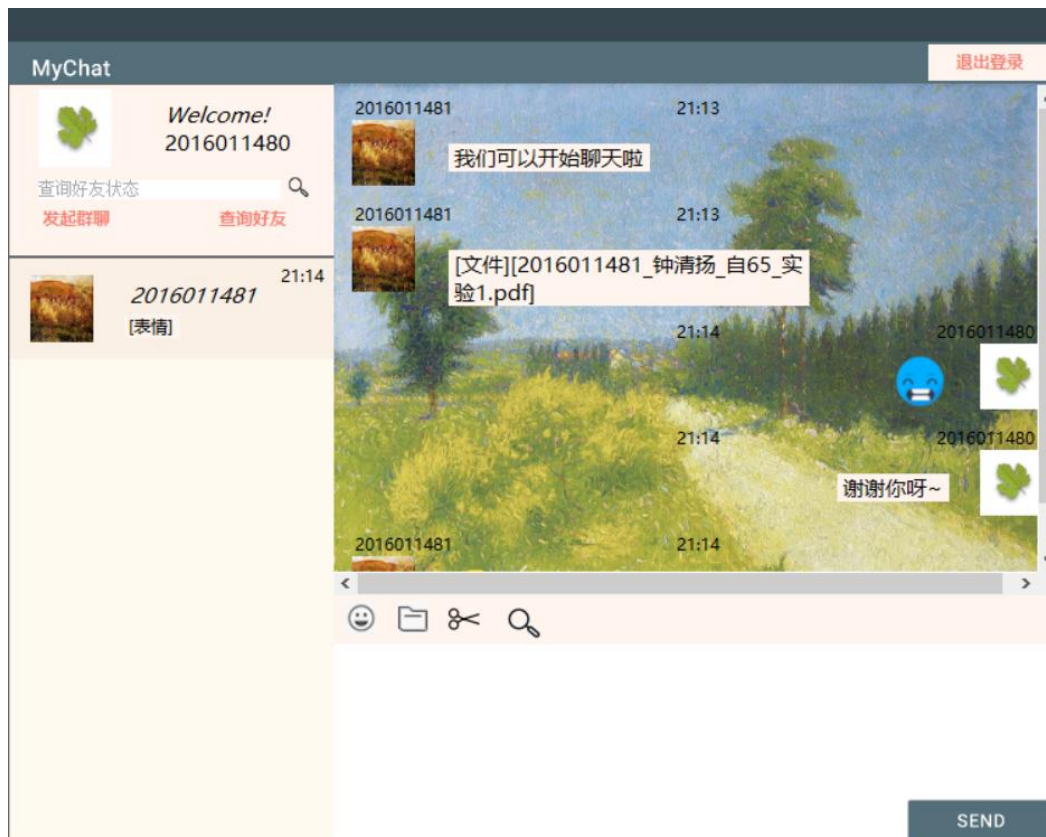
为简化界面设计难度,规定当前用户只能对当前所有已添加对话的好友发起群聊,而不能选择添加群聊的对象。所有被添加进群聊的在线好友都会产生与被动发起对话时相同的响应:用户标签栏中添加新的好友标签(包括群聊的专用头像与群聊名称等)并生成新的群聊对话框。

实际上,群聊的实现原理是将需要发送的消息向所有成员进行人为广播,P2P 对话时,需要记录用户的名称,广播时则需要记录所有群聊好友的名称-IP 地址键值对与当前群聊人数。为尽可能减少对界面设计架构的改动,预想群聊功能实现后,群聊对话框中的发送方头像将始终为群聊预置头像,每条接收信息均显示为“发送人”+发送信息或发送表情的形式。





(2016011481 向 2016011480 对话)



(2016011480 向 2016011481 对话)

## 4.项目总结

最初得知计算机网络课程的大作业题目时,我感到非常沮丧。微信和 QQ 作为生活中不可或缺的社交工具,其功能乃至界面都已经给人留下了太过深刻的印象。利用课程所学内容设计 P2P 网络聊天系统,无非是模仿 QQ/微信以实现基本功能,难以超越经典的沮丧感一度深深地扼住了我的咽喉。

为了能让自己的作业耳目一新,起初我为它设计了很多有趣的创新功能。除选做任务外,我还曾设想增添截图,抖窗,更换聊天背景,更换用户头像等功能。然而真正实现起来才发现信息的传递与控件的配合非常不易,每调试一个功能都意味着大半天精力的付出。到最后项目功能越做越少,在考试周日渐逼近的压力下我不得已遵循了“世上无难事,只要肯放弃”的原则。虽然心有不甘,但总体而言还算满意。虽然写作业时觉得计算机网络课程的大作业性价比太低,但看到好看的界面流畅运行时还是禁不住内心小小的欢喜。

值得一提的是,设计项目时我选择了为每个用户保留固定端口的做法。虽然此方法将占用很多端口资源,但考虑到这门课程的所有选课人数不过百人,因而资源消耗还算可以接受。这种看似愚蠢的做法给程序调试带来了极大的便利,在身边同学不得不借用多台电脑进行调试时,我完全可以同时运行多个进程自己测试,很大程度上提高了调试的效率。

完成作业时遇到了很多小问题,比如写作业时打开 Backlog 文档以记录调试中出现的问題,结果测试传文件的时候一直提示进程被占用,调试了一小时后才发现传送的正是已打开的 Backlog 文档;写表情栏控件时辛苦设计好的界面忽然从控件类变成了普通类,后来才发现窗体首类必须是界面类才能用设计器进行设计;委托应用不熟练导致简单的切换页面操作重复返工等。总而言之,课程大作业教会我相信科学拒绝魔法,使我在应用课程知识之余很好地锻炼了编程基本功。

另外,由于参考文献过多,很多没有及时记录,这里只能列出参考最多的几条;MyChat 设计整体采用了莫奈的油画风格,但很多图片名字都是随手起的,如有冒犯世界艺术瑰宝还请多多谅解,不要嘲笑设计者的孤陋寡闻。

## 5.参考文献

[1]Csharp 入门经典(第四版).pdf

[2][https://blog.csdn.net/kevin\\_jot/article/details/54947037](https://blog.csdn.net/kevin_jot/article/details/54947037)

[3]<https://blog.csdn.net/tandashuai/article/details/41966917>