

天津大学



网络聊天室实验报告

学生姓名_____赵万旭_____

学院名称_____智能与计算学部_____

班 级_____软件工程 2 班_____

学 号_____3019244207_____

一、需求分析

1. 整体需求

设计网络通信协议，运用 socket 编程技术，开发一个网络聊天室，实现多客户端的接入与通信功能，然后对其进行系统的评价与测试。

2. 功能需求

基本功能
1). 服务端可以建立网络聊天室并等待客户端连接。
2). 客户端可以连入网络聊天室，并成功将自己的消息发送至服务端。
3). 多个客户端能及时收到并显示彼此的聊天消息。
4). 新用户加入和旧用户退出不会影响聊天室的正常运作。
扩展功能：
1). 系统能正确处理粘包、掉线等异常情况。
2). 系统进行了高并发的优化，并对其并发性能开展了评价实验。
3). 系统支持文件传输功能
4). 系统支持语音聊天功能。

二、系统设计

2.1 架构设计

本实验利用 python 语言进行编程，引用了一些 python 库来进行辅助设计。

服务器端在正常接收消息和文件时开启一个进程，使用同一个 IP 地址和端口号进行连接，在进行语音通话时为避免冲突，开启另外一个端口启用线程进行语音传输，如图 1 所示，IP 地址及端口号如图 2

所示

```
if __name__ == '__main__':  
    server1 = ThreadServer1()  
    server2 = ThreadServer2()  
    server1.start()  
    server2.start()
```

图 1. 实现不同功能的两个线程

```
IP = '127.0.0.1' #服务器IP  
PORT = 6789 #服务器端口  
PORT1 = 5678 #语音通话时的端口
```

图 2. IP 地址及端口号

将语音通话另外开一个新的进程有以下好处：

- 可以在语音通话的同时进行文字聊天，两者并发进行，互不影响，提高了聊天室的性能
- 语音模块单独实现，能够对其进行很好地调试，及时发现问题并处理问题

2.2 界面设计

2.2.1 图形界面设计



用户登陆窗口

IP地址 127.0.0.1

端口号 6789

用户名

登录

图 3. 用户登录界面

默认 IP 地址和端口号，只需输入用户名进入聊天室，点击登录即可进入，并会显示登陆成功，一旦输入用户名为空，系统会自动提示，并需要用户输入姓名



图 4. 用户登录提示界面

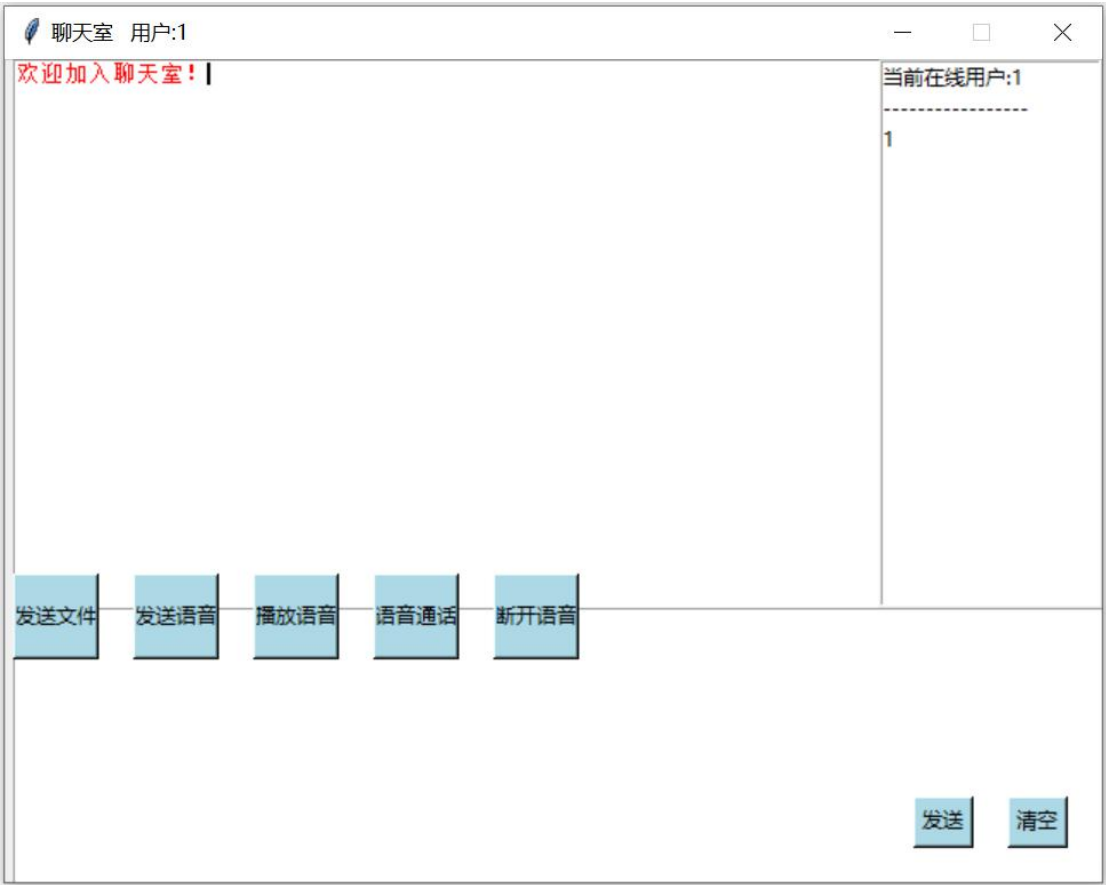


图 5. 聊天室界面

2.2.2 文件传输设计

点击发送文件按钮，即可选择文件进行传输，界面如图所示，点击所需要传输的文件即可

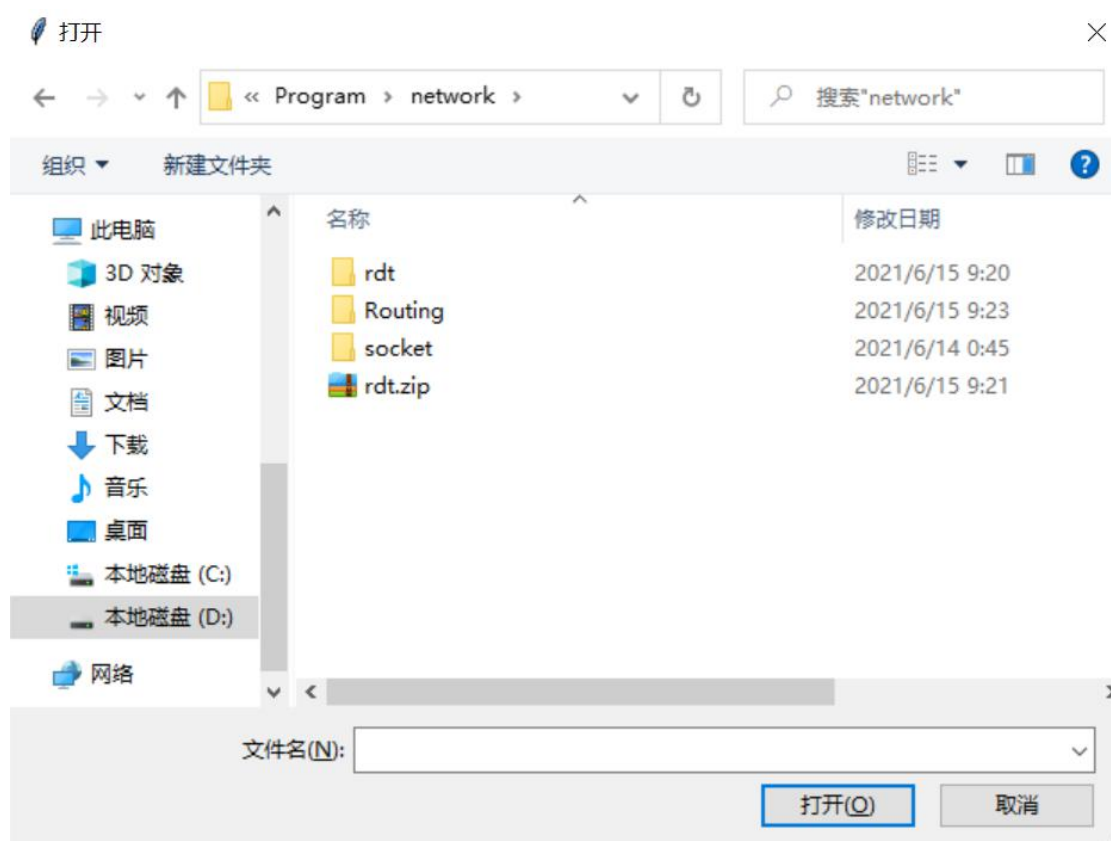


图 6. 文件传输

2.2.3 语音通话设计

语音通话我实现了两种方式，一种是语音消息的设置，一种是语音通话的设置。

1). 语音消息

点击发送语音即可对语音进行录入，并将其打包成文件发给聊天室的其他用户，点击播放语音即可听到语音消息。

2). 语音通话

点击语音通话即可与聊天室其他用户进行通话，点击断开语音即可断开

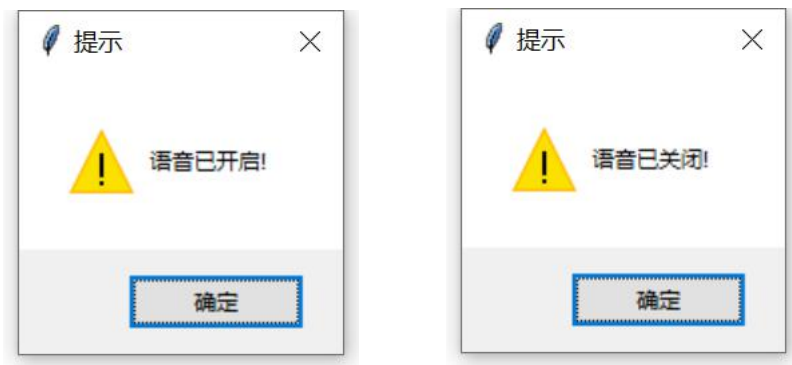


图 7. 语音提示

2.2.4 文本输入设置

文本输入设置了“发送”按钮和“清空”按钮，“发送”按钮将文本消息进行发送，“清空”按钮可以将输入框中的文本消息清零

三、协议设计

在本次实验中，所有协议均采用 TCP 协议，实现数据的可靠传输和稳定传输

四、实现方式

4.1 基本功能

4.1.1 文字聊天功能



图 8. 文字聊天

我在文字聊天中实现了如下功能

- 1) 首先是在聊天界面中将自己所发送的消息与其他人进行了区分，自己所发的消息为绿色，其他人所发的消息为黑色
- 2) 同时实现了在聊天界面中显示发送者的用户名和发送时间，用

蓝色进行标识

3)在聊天界面右侧显示了当前在线用户人数以及在线用户的用户名列表

```
timeNow = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
message ={'depcode':1,'msg': entryIuput.get()+':'+timeNow, 'user': user }
```

图 9. 文字聊天实现

图 9 变量 timeNow 获取当前的时间，将当前时间与发送消息和用户的名字放在了 message 里面，并通过 depcode 将其标识为文本，发送给服务器端，服务器端对其进行解析，并显示给其他用户

4. 1. 2 统计用户数量

服务器端查看接入的用户个数及其状态，将其进行统计，进而显示当前在线用户数量

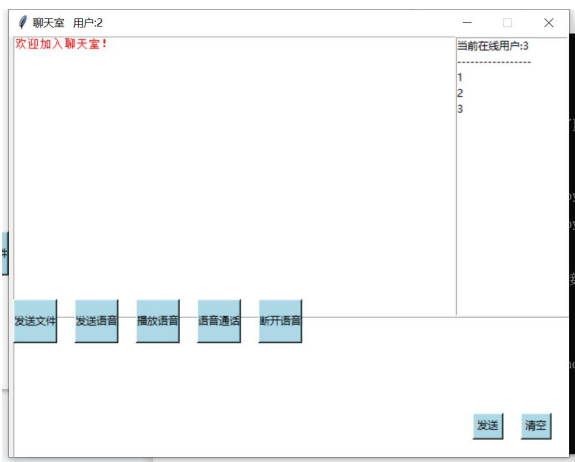


图 10. (a)

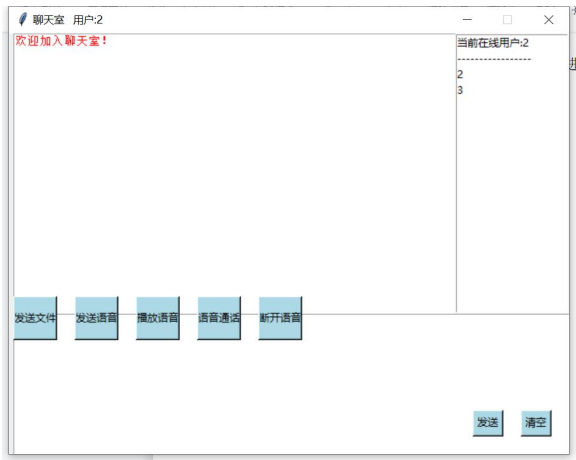


图 10. (b)

图 10（a）为三个用户接入情况，图 10（b）为用户 1 退出后统计情况

用户的登录和退出对聊天室的正常运作不产生影响

在用户名的处理上采用加（）形式，例如，已有用户“a”登录，那么下一个以用户名“a”登录的用户名字将改为“a(1)”

4.2 拓展功能

4.2.1 沾包、掉线等异常情况

TCP 粘包就是指发送方发送的若干包数据到达接收方时粘成了一包，从接收缓冲区来看，后一包数据的头紧接着前一包数据的尾，出现粘包的原因是多方面的，可能是来自发送方，也可能是来自接收方。

在对沾包问题的处理上采用多次收发，每个包的大小定义为 1024 个 bit，每次接收准确大小数据，避免沾包

对于掉线的情况，通过查看客户端的心跳即可确认客户端是否在线，即每隔一段时间用户发送一个心跳包给服务器，如果服务器长久未收到心跳包则判定该用户掉线

4.2.2 并发调优与压力测试

```
Thread_pool = ThreadPoolExecutor(max_workers=1000)
```

a. 创建线程池

```
while True:
    conn,addr = self.ServerSocket.accept()
    # threadRecv = threading.Thread(target=self.receive,args=(conn,addr))
    # threadRecv.start()
    threadRecv = Thread_pool.submit(self.receive,conn,addr)
```

b. 启动线程池

图 11. 线程池

通过线程池对性能进行模拟，测试在接入 1000 个用户的并发连接时刻的各个性能指标，将优化前与优化后进行对比，得出如图 12 所示的结果


```

Please enter the number of connected clients:1000
-----
Total time:                126.23 s
Average delay:             1.467s
Minimum delay:             0.365s
Maximum delay:             87.341s
Client request statistics: 189027
Client access statistics:  1000

```

```

Please enter the number of connected clients:1000
-----
Total time:                89.15 s
Average delay:             0.234s
Minimum delay:             0.034s
Maximum delay:             23.356s
Client request statistics: 70789
Client access statistics:  1000

```

图 12. 优化前后各性能对比

4.2.3 文件传输

```
from tkinter import filedialog
```

```
file = filedialog.askopenfilename(initialdir=os.path.dirname(__file__))
```

图 13. 文件处理

如图 13 所示，引入 filedialog 模块，并利用文件选择框 askopenfilename 打开文件，将文件打包传给服务器端，同时，各用户接收到服务器端发来的该用户发送的文件，可以选择是否接收

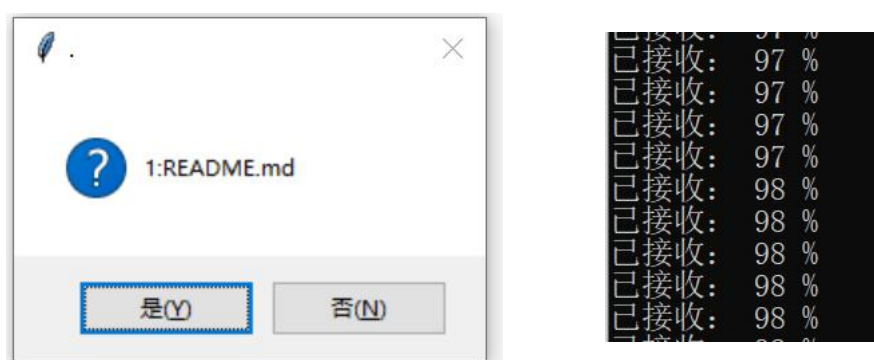


图 14. 文件接收

如图 14 所示，用户可以选择是否接收用户“1”发来的文件

README.md 文件，在终端上会显示文件的传输进程

4.2.4 语音通话

1) 语音消息

在发送语音消息时将语音打包成一个 wav 文件，其传输过程与文件相同

2) 语音通话

语音通话利用 pyaudio 模块

```
chunk_size = 1024 # 512
audio_format = pyaudio.paInt16
channels = 1
rate = 20000
```

图 15. 语音通话参数

如图 15 为语音通话的一些参数

服务器端单独创建一个进程为语音通话做准备

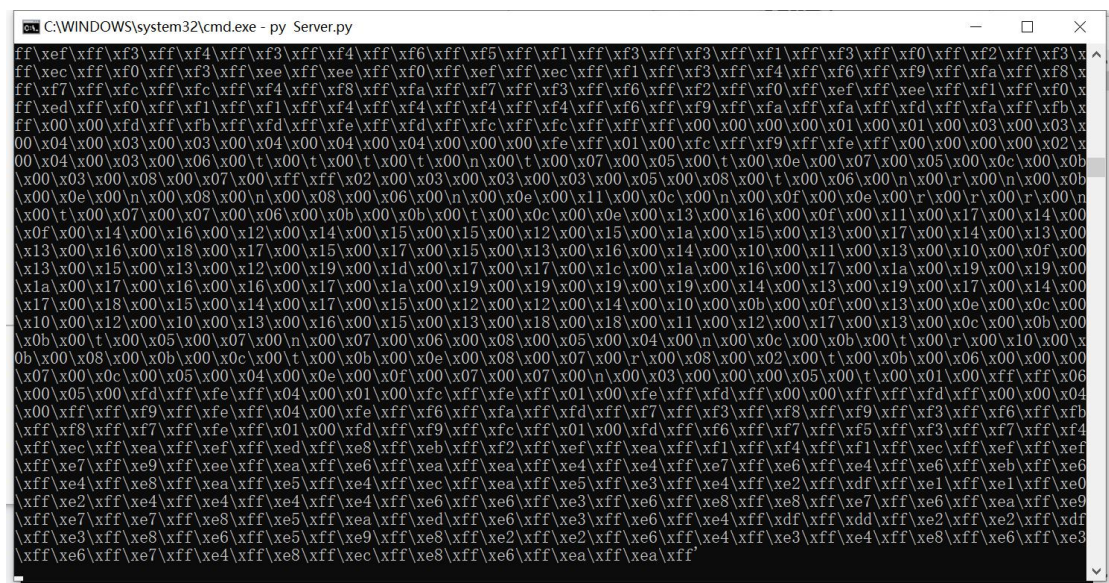


图 16. 语音通话数据

图 16 所示为语音通话中产生的数据在终端的显示，表明语音数据传输成功

五、性能评价

本聊天室性能较好，能够满足多人聊天的需求，并加入一些简单的功能

以下功能基本实现

基本功能
1). 服务端可以建立网络聊天室并等待客户端连接。
2). 客户端可以连入网络聊天室，并成功将自己的消息发送至服务端。
3). 多个客户端能及时收到并显示彼此的聊天消息。
4). 新用户加入和旧用户退出不会影响聊天室的正常运作。
扩展功能：
1). 系统能正确处理粘包、掉线等异常情况。
2). 系统进行了高并发的优化，并对其并发性能开展了评价实验。
3). 系统支持文件传输功能
4). 系统支持语音聊天功能。

六、总结

本次实验是进入大学以来难度最大的一个实验，历时时间之久，耗费工夫之大，首屈一指，虽然过程中几多苦难几多艰辛，行百里者半九十，坚持下来就是胜利！开始做实验也是非常痛苦，无从下手，不知所措，但随着一个个功能慢慢的晚上，心中的喜悦也是一天天增加，没想到自己居然拥有这么大的潜力！在实验的过程中，对 socket 和计算机网络整体都有了很深刻的认识，受益匪浅。希望在以后的日子里，我都能够携带着这种百折不挠的精神，一路且歌且行！