

1 项目简介

本项目基于洪泛(Flooding)技术实现了一个资源共享系统，可为用户提供文件共享、共享资源下载等服务。另外，该系统为用户提供了预置的**配置信息修改服务**。

为了更好地为测试者提供**直观的**测试界面以及**便捷的**测试方法，本项目提供了集成的可执行文件，通过模拟所在拓扑图的不同节点，测试者能够模拟不同对等方的真实操作。

2 系统实现

2.1 开发环境和工具

操作系统： Windows 10 1903

处理器： Intel® Core™ i7-8809G CPU @ 3.10GHz

RAM: 16.0GB

开发工具： PyCharm 2019.2.3 (Professional Edition)

开发语言： Python 3.7.6

2.2 系统设计

本系统包含两个已封装的类：Config、Connection。

2.2.1 Config

Config 类：

get_attr(i)：获取对等方“peer_i”的信息，并打包以字典形式返回

set_attr(i)：对用户输入的数据检查后，修改对等方“peer_i”的信息。

modify(i, attr_dict)：将对等方“peer_i”的修改后信息写入配置文件。

get_ttl()：返回 time to live 值。

get_peer_num()：获得当前网络环境中对等方的数量。

__init__()：构造函数。

2.2.2 Connection

Connection 类：

set_num(num): 设置节点编号。

set_ip(ip_addr): 设置节点 IP 地址。

set_server_port(server_port): 设置节点的服务端端口。

set_client_port(client_port): 设置节点的客户端端口

set_peer_list(peer_list): 设置节点的邻居列表。

set_share_dir(share_dir): 设置资源共享本地目录。

query(root, filename): 递归查询 root 目录下是否存在名为 filename 的文件。

tcp_handler(conn, addr): tcp 服务端在与客户端连接后的处理方法。

tcp_server(): tcp 服务端基础建立。

__send(conn, filename): 在 conn 连接中发送名为 filename 的文件。

__save(conn): 保存从 conn 连接中收到的文件。

tcp_client_notice(ip, port, msg): 节点调用自身的客户端端口进行信息发送。

update_peer_attr(): 更新邻居属性字典。

update_ttl(msg): 更新消息中的 time to live。

2.2.3 其他函数

filemd5.py:

get_file_md5(file): 计算文件 file 的 md5 值。

compare_file_md5(file, md5): 计算文件 file 的 md5 值并与传入的 md5 进行比较。

process.py:

tcp_server(ID): 用于多进程的 tcp_server 函数。

tcp_client(role_num, filename): 用于多进程的 tcp_client 函数。

3 关键技术点

3.1 配置文件读取/修改

本系统所有对等方信息都存储在 config.ini 文件中，存储形式如下（以 Peer_0 为例）：

```
[Peer - 0]
ip_addr = 127.0.0.1
server_port = 10800
client_port = 10810
share_dir = 0/
peer_list = [1, 2]
```

设计时采用 **configparser** 包，将其用于配置信息的**格式化文件读取**。在用户进行配置信息的修改时，系统将**对新信息的正确性予以确认**，并给予用户相应的**详细错误反馈**，方便用户进行修改更正。

3.2 服务端支持并发

本系统应用**多线程**，将 TCP 服务端对连接的处理方法封装，并在产生连接时**新建一个处理线程**，此时服务端继续监听其他并发连接。考虑到运行效率，在本项目中支持最高**5 个并发**。

3.3 文件传输检验

本系统在进行文件传输前后，将对传输文件的正确性进行验证。

在本项目中，选用文件的 **MD5 值** 为标准进行校验，通过在传输前后分别计算文件的 MD5 值，系统在接收完成后计算所接收文件的 MD5 值并与收到的 MD5 值进行比对，并反馈验证信息。

3.4 多进程模拟服务端

为了便于测试者的使用，本系统采用多进程的方式在主进程中用**进程池**的方法模拟各个对等方服务端的进程，并编写了简易的 UI，方便测试者对系统反馈的各类信息进行阅读。

4 运行实例

本项目预置测试拓扑图如下：

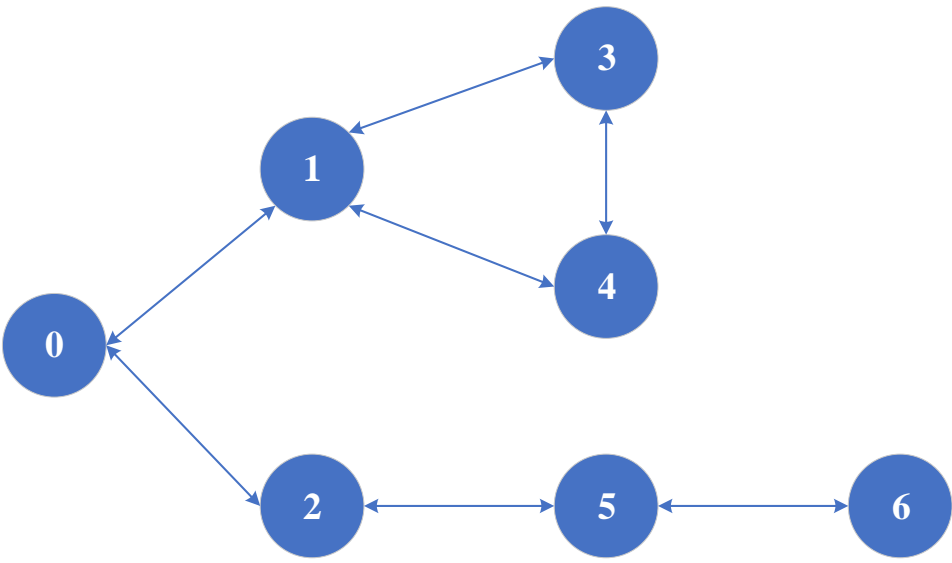


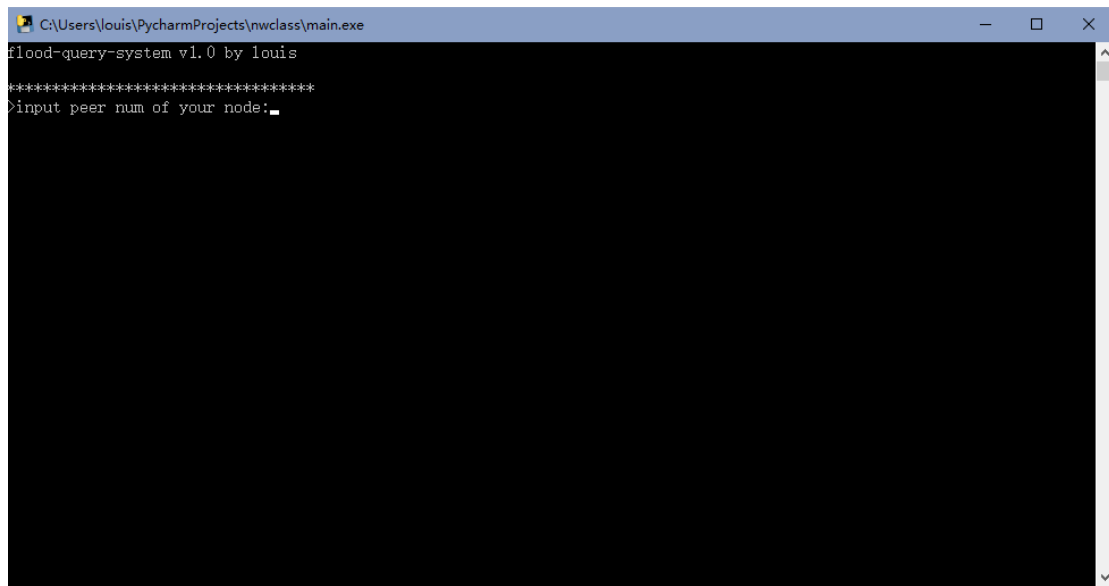
图 4-1 网络拓扑结构

本项目预置测试文件目录如下：

表 4-1 测试文件目录

对等方编号	共享目录	原有文件
0	/0/	0.txt
1	/1/	testpackage 文件夹 (含 test.docx, test.txt)
2	/2/	2 文件夹 (含 2.txt)
3	/3/	3.txt
4	/4/	4.txt
5	/5/	5.txt
6	/6/	6.txt

Step 1 运行 `dist/main.exe`

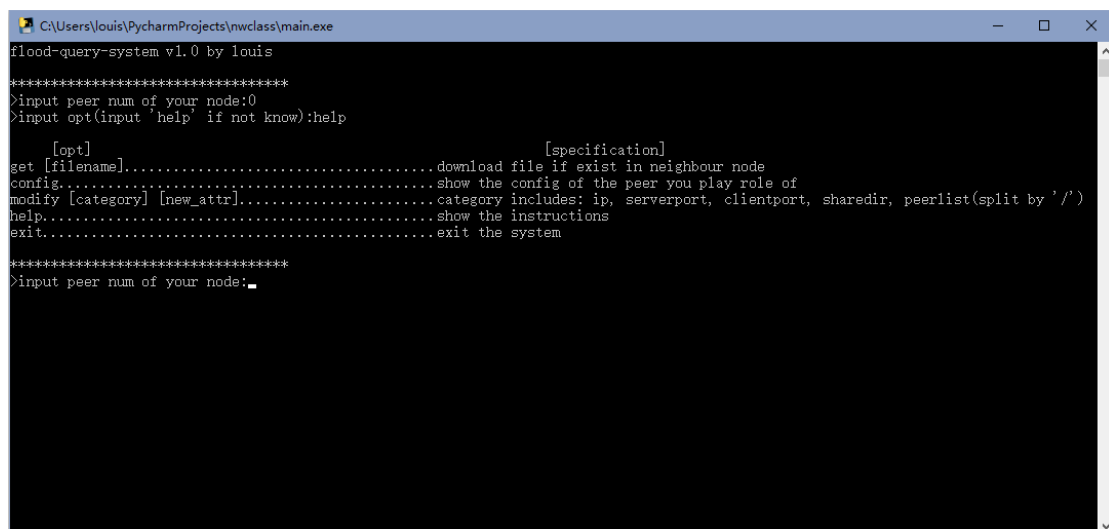


```
C:\Users\louis\PycharmProjects\nwclass\main.exe
flood-query-system v1.0 by louis
*****
>input peer num of your node: _
```

图 4-2 运行 `dist/main.exe`

Step 2 输入模拟的节点编号（0~6），这里以 0 为例

Step 3 测试命令 1: help

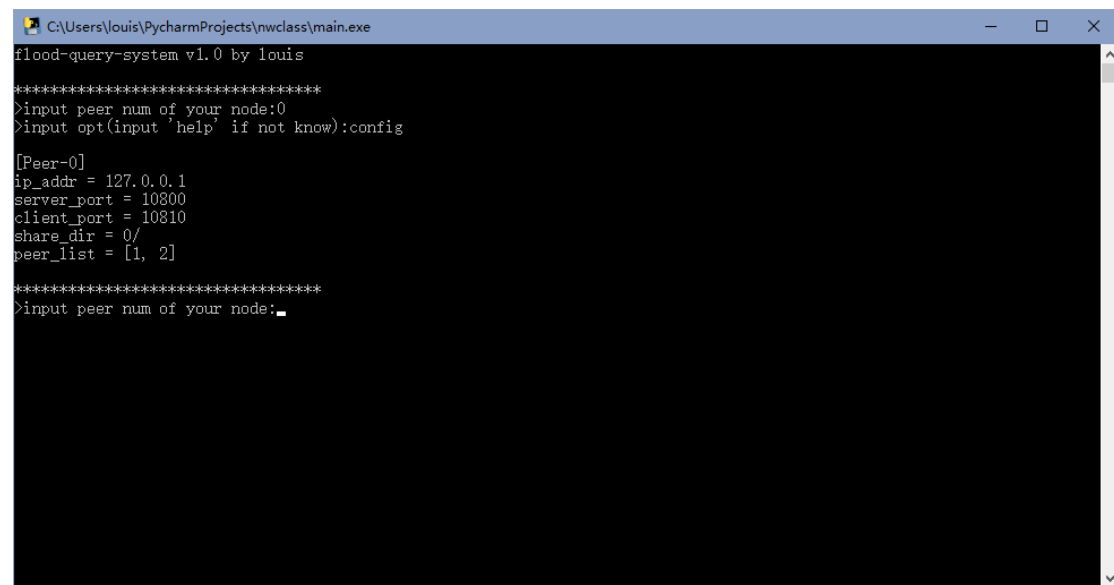


```
C:\Users\louis\PycharmProjects\nwclass\main.exe
flood-query-system v1.0 by louis
*****
>input peer num of your node:0
>input opt(input 'help' if not know):help

[opt]                                [specification]
get [filename].....download file if exist in neighbour node
config.....show the config of the peer you play role of
modify [category] [new_attr].....category includes: ip, serverport, clientport, sharedir, peerlist(split by '/')
help.....show the instructions
exit.....exit the system
*****
>input peer num of your node: _
```

图 4-3 测试命令 help

Step 4 测试命令 2: config

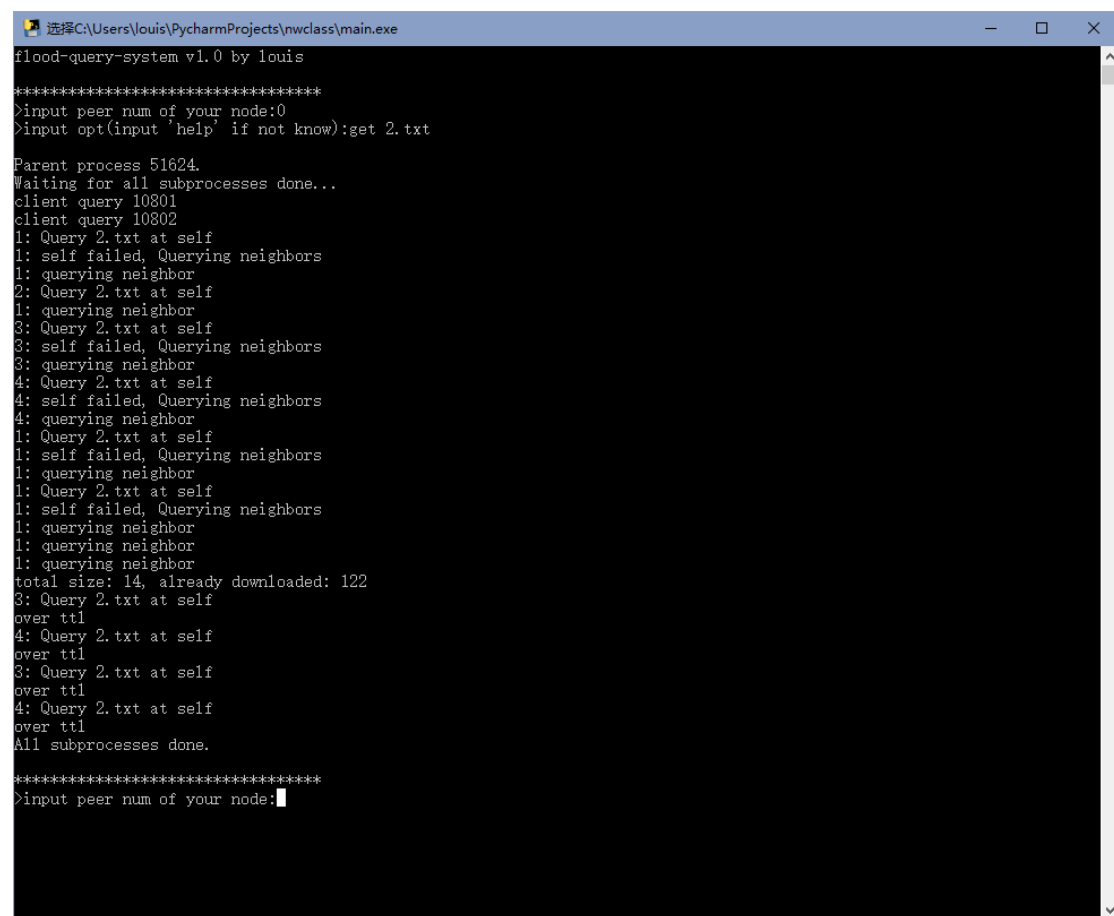


```
C:\Users\louis\PycharmProjects\nwclass\main.exe
flood-query-system v1.0 by louis
*****
>input peer num of your node:0
>input opt(input 'help' if not know):config

[Peer-0]
ip_addr = 127.0.0.1
server_port = 10800
client_port = 10810
share_dir = 0/
peer_list = [1, 2]
*****
>input peer num of your node:█
```

图 4-4 测试命令 config

Step 5 测试命令 3: get 2.txt (文件传输测试)



```
选择C:\Users\louis\PycharmProjects\nwclass\main.exe
flood-query-system v1.0 by louis
*****
>input peer num of your node:0
>input opt(input 'help' if not know):get 2.txt

Parent process 51624.
Waiting for all subprocesses done...
client query 10801
client query 10802
1: Query 2.txt at self
1: self failed, Querying neighbors
1: querying neighbor
2: Query 2.txt at self
1: querying neighbor
3: Query 2.txt at self
3: self failed, Querying neighbors
3: querying neighbor
4: Query 2.txt at self
4: self failed, Querying neighbors
4: querying neighbor
1: Query 2.txt at self
1: self failed, Querying neighbors
1: querying neighbor
1: Query 2.txt at self
1: self failed, Querying neighbors
1: querying neighbor
1: querying neighbor
1: querying neighbor
total size: 14, already downloaded: 122
3: Query 2.txt at self
over ttl
4: Query 2.txt at self
over ttl
3: Query 2.txt at self
over ttl
4: Query 2.txt at self
over ttl
All subprocesses done.
*****
>input peer num of your node:█
```

图 4-5 测试命令 get 文件

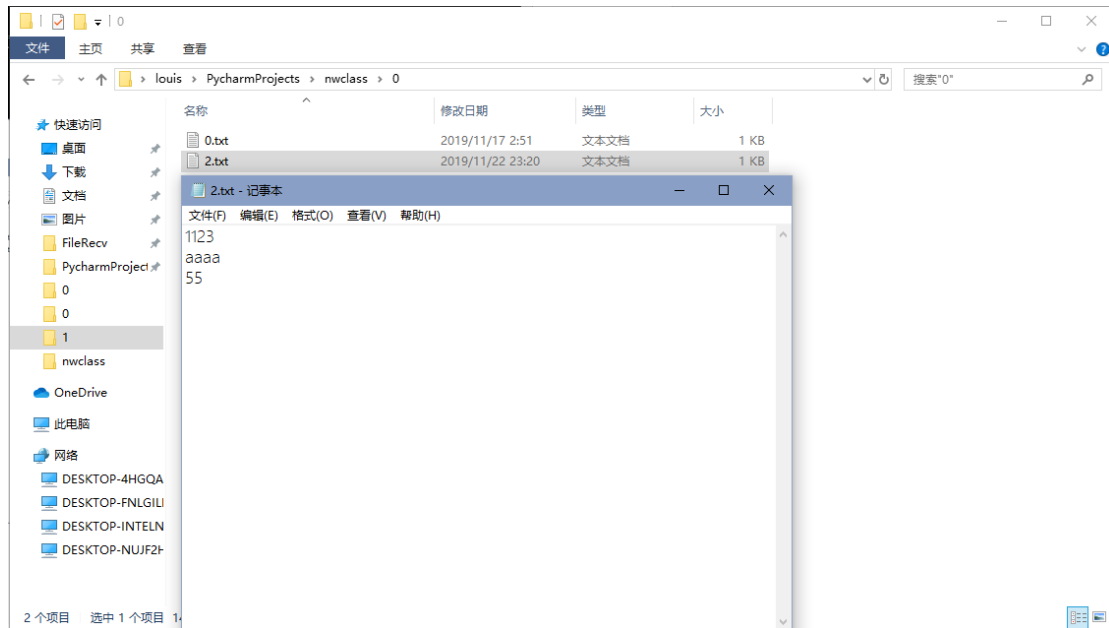


图 4-6 测试效果

Step 6 测试命令 4: testpackage (文件夹传输测试)

```
C:\Users\louis\PycharmProjects\nwclass\main.exe
>input peer num of your node:0
>input opt(input 'help' if not know):get testpackage
Parent process 47840.
Waiting for all subprocesses done...
client query 10801
client query 10802
1: Query testpackage at self
2: Query testpackage at self
2: self failed, Querying neighbors
2: querying neighbor
5: Query testpackage at self
5: self failed, Querying neighbors
5: querying neighbor
5: querying neighbor
2: Query testpackage at self
2: self failed, Querying neighbors
2: querying neighbor
6: Query testpackage at self
6: self failed, Querying neighbors
6: querying neighbor
5: Query testpackage at self
over ttl
5: Query testpackage at self
over ttl
total size: 12095, already downloaded: 12316
All subprocesses done.
*****
>input peer num of your node:
```

图 4-7 测试命令 get 文件夹

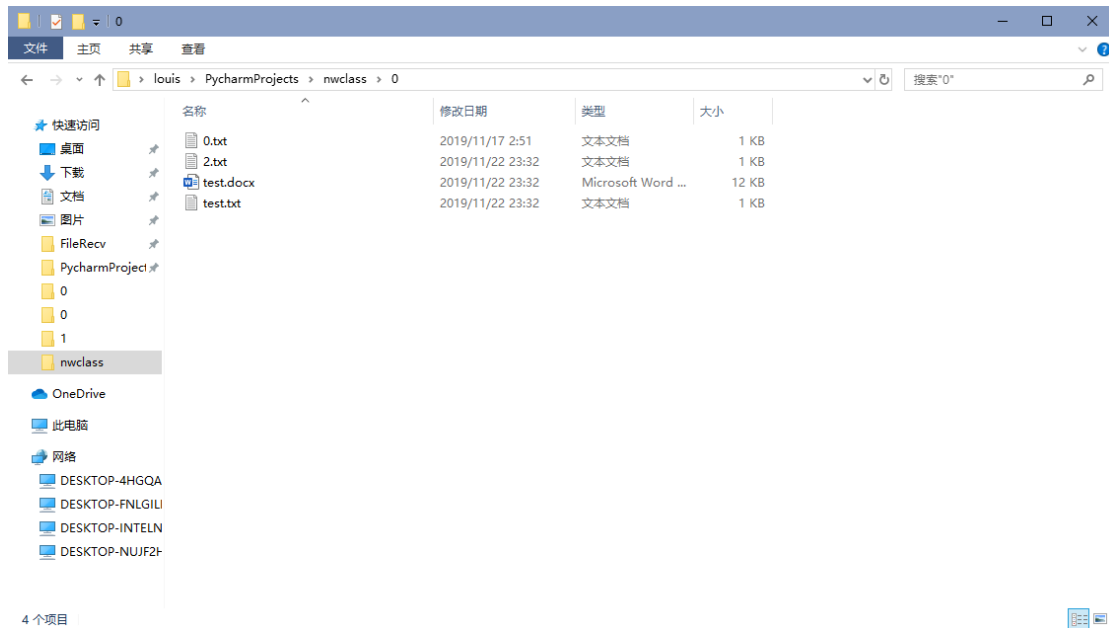


图 4-8 测试效果

注：如出现网络中断提示（connect abort），请重试。

Step 7 测试命令 5: modify（以 modify ip 127.0.0.2 为例）

```
CAUsers\louie\PycharmProjects\nwclass\main.exe
*****
>input peer num of your node:0
>input opt(input 'help' if not know):config

[Peer-0]
ip_addr = 127.0.0.1
server_port = 10800
client_port = 10810
share_dir = 0/
peer_list = [1, 2]

*****
>input peer num of your node:0
>input opt(input 'help' if not know):modify ip 127.0.0.2

{'ip_addr': '127.0.0.1', 'server_port': '10800', 'client_port': '10810', 'share_dir': '0/', 'peer_list': [1, 2]}
[('ip_addr', '127.0.0.1', '127.0.0.2')]
Modify success

*****
>input peer num of your node:0
>input opt(input 'help' if not know):config

[Peer-0]
ip_addr = 127.0.0.2
server_port = 10800
client_port = 10810
share_dir = 0/
peer_list = [1, 2]
```

图 4-9 测试命令 modify ip 127.0.0.2

Step 8 测试完成: exit