LFTP

Source Code

Github

Group Member

Name(姓名)	Student Number(学号)	Major(专业)
Weiyuan Xu(徐伟元)	16340261	Software Engineering-Computer Application(软件工程-计算机应用)
Yongqi Xiong(熊永琦)	16340258	Software Engineering-Computer Application(软件工程-计算机应用)

LFTP, a network application, supports large file transfer between two computers in the Internet.

Dependency

Operating System: `linux` `/macOS` `/windows`

Environment: `>= python 3.6`

Usage

For **macOS** user, type the command in your terminal first to enable transferring the max size of the UDP packet:

```
sudo sysctl -w net.inet.udp.maxdgram=65535
```

Before you use this program, on the server-side, you should make a folder named `data` under `code/` first to store the files to exchange with clients:

```
cd code
mkdir data
```

Then, you could run command below to run the program on the server:

```
# use python 3.x
python ./server.py [hostname port]
# default hostname: localhost
# default port: 8080
```

After that, you could run the command below to connect and exchange files with the server:

```
# use python 3.x
python ./client.py lget/lsend hostname[:port] filename
# default port: 8080
```

Hope you enjoy your time with it.

The technical requirements

- Programing language: `Python` ;
- LFTP should use a **client-server** service model;
- LFTP must include a client side program and a server side program; Client side program can not only send a large file to the server but also download a file from the server.
  > Sending file should use the following format:
  > *LFTP lsend myserver mylargefile*
  > Getting file should use the following format:
  > *LFTP lget myserver mylargefile*
  > The parameter myserver can be a url address or an IP address.
- LFTP should use **UDP** as the transport layer protocol.
- LFTP must realize **100% reliability** as TCP;
- LFTP must implement **flow control** function similar as TCP;
- LFTP must implement **congestion control** function similar as TCP;
- LFTP server side must be able to **support multiple clients** at the same time;
- LFTP should provide **meaningful debug information** when programs are executed.

The Design Doc

Detail Design Doc are here

Simple introduction of design are shown below:

- **Transport Layer: RDP Protocol**
  - **Packet Structure**

UDP packet data field
Sequence Number
Acknowledgement Number
Flag Field (ACK, SYN, RST, FIN, WRW)
rwnd
Data

  - **Fundament**
    According to the Application layer requests, *send data and receive data* function are fundamental and application needn't know the implementation. Thus, we first design two functions: **`rdp_send(data)`** to send data and **`rdp_recv(size)`** to get data. Furthermore, these two function should act like TCP, which means application just invokes functions and **knows** where it get/send data. So we need to make connection between server and client before invoking these functions with *handshake behavior*. **`makeConnection(targetAddress)`** is needed.
  - **Multiple Client**
    Since server must support multiple client, the server application(host) must handle the clients at the same time. So **multiple thread** is needed. We provide each connected client a *server program* running in different *port*. So we design **`listen(num)`** function to *listen* the connection requests from clients and maximum number of client for server to serve is `num` . The listen function provides the *listening* and helps make connection between server and client. Hence, sockets are created when connection successfully made in `listen` , we must export the serving socket for server application. **`accept()`** retrieve a serving socket and server application must run the socket in a thread and handle it.
  - **Summary**
    `rdp_send(data)` , **`rdp_recv(size)`** , **`makeConnection(targetAddress)`** , **`listen(num)`** and **`accept()`** are the most important function designed in RDP. Following, we will introduce the implementation and the detail design of them.

The Testing Doc

Detail Test Doc