UDP & FTP Report

Project 1 in lecture Computer Network

School of software Jiacheng Zhang 2017013564

Section 1: General

UDP 实现了 server 端和 client 端之间的直接通信并发送数据。而 FTP 需要两个 TCP 连接，实现了要求的所有的 16 个命令，并实现了断点续传功能， GUI 界面和多用户，autograde.py 评测满分。

Section 2: Implement of FTP

FTP 概述：

FTP 是基于 TCP 连接的文件传输协议。它由两个 TCP 实现了连接的建立，第一个 TCP 连接是和用于和 client 端实现三次握手和 client 命令的发送还有 server 端返回处理的结果，生命周期一直存在到 server 或者 client 一方断开连接为止。第二个 TCP 连接是用来传输数据发送文件的，生命周期存在于 PORT/PASV 命令到文件发送结束为止。用户登录后可以向 server 发送请求，然后远程操作文件目录，实现文件的移动，删除，增加，重命名，查看路径等 FTP 基本操作。

FTP 运行方式：

Server 端：在 server 目录下输入 ./server -port [端口号] -root [根目录]

Port 和 root 是可选的，不填就默认端口 3000 根目录/tmp。

Client 端：在 client.py 下用命令行(python3 client.py)打开 gui 界面（python 需要事先安装 pyqt5,安装方法：Windows 下： pip install pyqt5， Linux 下 pip install python3-pyqt5）。然后首先填写本机 ip（127.0.0.1）和端口号，此端口号和运行 server 的端口号相同。点击 connect， 然后执行 USER/PASS

命令登录，就可以执行操作了。

FTP 命令：

USER [username]：发送用户名，服务器确认后会引导填写密码。

PASS [password]：发送密码，服务器确认登录。

CWD [absolutely path]：修改路径，参数必须为绝对路径。

PWD ：打印当前绝对路径。

SYST ：不需要参数，返回服务器的操作系统类型。

TYPE [transfer type] ：指定传输文件的方式（是按照 ascll 码还是二进制等）。

MKD [file name] ：在当前路径下创建文件夹。

RMD [file name] ：删除位于当前目录下的一个**空文件夹**。

RNFR [file name] ：指定要修改的文件名称。

RNTO [file name] ：指定文件被修改的名称。

PORT [h1,h2,h3,h4,p1,p2] ：客户端打开端口等待服务器的数据连接，主动传输。参数是客户端打开套接字所使用的地址和端口号。

PASV ：服务器端打开端口等待客户端的数据连接，被动传输，不需要参数。

RETR [file name]：获取当前目录下的文件。

STOR [file name]：将本地文件存储到远程服务器的目录下。

LIST [file name]：显示出当前目录下的所有文件。

QUIT ：退出连接，再次登录需要重启连接。

REST ：断点续传，不带参数。如果上一次网络断开没传完就会自动携带参数访问 server 端。

Section 3：Conclusion

我的程序有几个特色，就是 GUI 界面能够看见本地存放的文件，操作记录，传输文件的进度条和实时网速。进度条我是提取文件大小然后在传输文件的 while 循环里去更新进度的。网速我是在传输文件的时候用下载的差量除以时间的差量来解决的。另外一个就是我实现了断点续传，在传输文件的时候，我手动断开我的 server 端模拟网络中断，客户端记录下文件指针。然后服务器重启，client 重连登录，先发送 REST 指令（不用手动带参数），然后 RETR 上次没传成功的文件，就可以实现断点续传，节省网络流量和等待时间。

此次实验难度对我来说很大，主要是代码的逻辑方面，需要防止各种错误操作和错误输入，防止客户端被卡崩掉。整个实验大约进行了 2 周多的时间，每天都在搞测试和 debug，踩了无数多的坑，进行了上千次的测试。不过此次实验对我的提高还是蛮大的，通过写 FTP，我基本理解了文件传输的原理和 TCP 协议里面的一些东西。这对我理解网络底层十分有帮助。