



中山大學
SUN YAT-SEN UNIVERSITY

《计算机网络 LFTP》

实验报告

学院名称：数据科学与计算机学院

学生姓名 1：张星 16340294

学生姓名 2：姚雪辉 15355119

专业（班级）：16 软件工程教务四班

时间：2018 年 12 月 3 日

目录

实验设计..... 2

LFTP 未完成..... 2

LFTP 新开 3

实现功能..... 3

使用说明..... 4

测试结果..... 4

代码模块..... 9

使用说明..... 10

实验分工..... 10

README..... 10

实验设计

LFTP 未完成

我们的项目有两个部分，一个是 LFTP 未完成，一个是 LFTP 新开。一开始我们打算按照要求实现所有的要求，包括 100%可靠的传输，流量控制，拥塞控制。具体的部分是新加头文件，虽然

首部
seqnum 1byte
acknum 1byte
cwnd 1byte
rwnd 1byte
checknum 2byte2
length 1byte
data

UDP 自身有校验和的部分，但是由于新加了首部的内容，还是做了校验和的部分，这样当接收方缓存满的时候，发送 rwnd 给发送方，然后发送方改成发送 1bit 的数据给接收方，直到接收

方清空缓存再恢复，以此实现流量控制。通过 cwnd 的值，实现慢启动以及拥塞避免。但是由于项目启动过晚，以及多线程的编程问题（主要是计时器调试没法完成），没有办法完成，所以最后项目做了一半。改成停等协议，没有办法实现流水线，也就没有办法实现流量控制，以及拥塞控制，此部分的项目都保存在 LFTP 未完成文件夹中。虽然没有办法完成，但是部分文件还是可以沿用的，所以虽然是停等协议，还是多做了其他功能。

LFTP 新开

在 LFTP 新开的文件夹中是已经完成的完整的项目。但是采用了更为简单的停等协议，即发送方发包，接收方收到包并确认正确无误后，发一个正确的 ACK，发送方收到 ACK 之后继续发包。我们用的是局域网测试，速度很快，但是没有丢包，所以手动设置了第二个数据包丢包，再截图里面会体现，其实如果用流水线，如果不用选择重传，而采用回退 N 步，不见得会比停等协议更加有效率，因为也是收到数据 ACK 之后，滑动窗口才会移动，所以虽然发了很多数据包，但是一旦顺序乱了，还是要从头开始重传，所以效率不是很好，这也是我们放弃流水线的一点原因。其中头文件还是沿用之前的头文件，所以虽然停等协议对 seqnum 没有要求，我们还是做了 seqnum 的处理，序号从 1-127，而不是简单的 0 和 1。由于 Socket 自带的 setTimeout 函数，可以方便实现计时器，我们设置一秒为超时时间，超过一秒发送方未收到 ACK。当然收到错误的数据包也会重传，我们沿用之前的首部数据，使用 checksum，接收方首先将文件读入 buff，然后将 buff 作为参数传入该类，根据传入参数计算出 checksum 并扩展 buff，将 checksum 放到 buff 的最后，将新的数据流作为 packet 的参数发送给接收方。接收方接到数据包之后，同样使用该类解析数据，先取出 buff 最后的 checksum，然后计算前面的 byte 数据，是否与该 checksum 相加后结果为 65535，若不等，则出错，发给发送方 NAK。Checksum 的计算依据书上的理论，将数据分为 16 比特，循环相加，回卷，最后结果取反即为 checksum。

实现功能

以上所讨论的都是协议的本身，除此之外我们实现的功能有，根据老师的要求有 9 个，除了第六条（流量控制）和第七条（拥塞控制）没有实现，其他要求都实现了。

输入命令是根据文件查找和正则表达式判断的，其中命令的前三部分是正则表达式实现的，由于链接的形式很多，用正则表达式实在没有办法判断，所以就直接判定是 IP 地址才可以，至于第四部分的文件，如果是发送文件，则会在本地查找文件，不存在就会提示命令错误，所以如果命令输入错的离谱，就会多次提示命令错误。

另外也有提供连接功能，首先是客户端发送 SYN 给服务端，服务端的端口接收到之后，会给该客户端分配一个端口，并告诉客户端，并以此实现多客户端同时上传或下载的功能。多客户端的并发使用，我们采用了服务端多线程来实现。具体思路类似于 TCP 连接，客户端首先通过命令，将连接请求发送到服务端，服务端受到请求命令后记录客户端的 IP 地址和端口，新开一个线程，将地址和端口号传入线程，然后线程新建一个 socket，使用不同的端口号，给发送端发确认信息。发送方收到后，则会往新的线程发消息，建立连接，此后就可以进行通信了，并且不会出现互相干扰的情况。

上传与下载功能，我们在客户端与服务端可以通过不同命令，来进入到相应模块来实现，此处两端的代码相似。需要注意的是，客户端选择上传功能时，我们先将文件名发送给服务端，以此作为服务端接收文件存储在磁盘的文件名，所以最好不要输入文件路径，或者本机即是服务端又是客户端，会出错。

使用说明

输入命令都在客户端完成，服务端无此功能。

若要在一台 PC 上开多个客户端，则需要更改 socket 端口号，编译后运行，以免出现冲突。

输入命令前需要确认服务端已在目标地址运行。

命令：LFTP lsend XXX.XXX.XXX.XXX 2018.mp4

若本地存在名为 2018.mp4 的文件，则向该 IP 地址上传该文件，服务端接收该文件，并保存为 2018.mp4. 若本地不存在该文件，则报错，提示该文件不存在。

命令：LFTP lget XXX.XXX.XXX.XXX 2018.mp4

从服务端下载一个名为 2018.mp4 的文件，若文件不存在，则会发送一条消息提示客户端，否则正常下载，并保存为 2018.mp4。

测试结果

首先测试客户端向服务端请求下载：服务端并无此文件，返回信息提示客户端。

客户端：

```
LFTP client start...
please input command
LFTP lget 172.19.42.52 2030.mp4
lget
172.19.42.52
2030.mp4
Send first SYN from client
Receive second SYN from server
Server Port: 50002
Connect Success
Download file name: 2030.mp4
File is not Exist
```

服务端：

```
LFTP server start...
From address:/172.19.42.52
From port:50003
Receive first SYN from client
Send second SYN from server
Connect Success
Downloading from server.
Receive the name of the file: 2030.mp4
File is not Exist
```

接着测试从客户端向服务端发文件，此文件不存在，显示提示信息，重新输入命令。

客户端：有可能会多次报错的可能，是因为经过多层判断错误。

```
LFTP client start...
please input command
LFTP lsend 172.19.42.52 2040.mp4
File does not exist
File does not exist
File does not exist
Wrong Command!!!
please input command
```

该 IP 地址不合法。

```
please input command
LFTP lsend 172.19.42.256 2030.mp4
Wrong Command!!!
```

进行下载测试：首先在一台 PC 同时启动两个客户端，需要更改端口，重新编译才可正常工作。然后分别向服务端请求下载名为 2018. mp4 和 2019. mp4 的文件。

(此过程 Github 中有视频，我们只是截取了一部分图片，完整过程可见 Github.)

```

C:\Windows\System32\cmd.exe - java
Receive count of 737 !
Receive count of 738 !
Receive count of 739 !
Receive count of 740 !
Receive count of 741 !
Receive count of 742 !
Receive count of 743 !
Receive count of 744 !
Receive count of 745 !
Receive count of 746 !
Receive count of 747 !
Receive count of 748 !
Receive count of 749 !
Receive count of 750 !
Receive count of 751 !
Receive count of 752 !
Receive count of 753 !
Receive count of 754 !
Receive count of 755 !
Receive count of 756 !
Receive count of 757 !
Receive count of 758 !
Receive count of 759 !
Receive count of 760 !
Receive count of 761 !
Receive count of 762 !
Receive count of 763 !
Receive count of 764 !
Receive count of 765 !

C:\Windows\System32\cmd.exe - java
Receive count of 428 !
Receive count of 429 !
Receive count of 430 !
Receive count of 431 !
Receive count of 432 !
Receive count of 433 !
Receive count of 434 !
Receive count of 435 !
Receive count of 436 !
Receive count of 437 !
Receive count of 438 !
Receive count of 439 !
Receive count of 440 !
Receive count of 441 !
Receive count of 442 !
Receive count of 443 !
Receive count of 444 !
Receive count of 445 !
Receive count of 446 !
Receive count of 447 !
Receive count of 448 !
Receive count of 449 !
Receive count of 450 !
Receive count of 451 !
Receive count of 452 !
Receive count of 453 !
Receive count of 454 !
Receive count of 455 !
Receive count of 456 !

```

两者连接的服务端 IP 地址相同，端口号不同，这是因为处于不同的线程中。

```

LFTP client start...
please input command
LFTP lget 172.18.34.217 2019.mp4
lget
172.18.34.217
2019.mp4
Send first SYN from client
Receive second SYN from server
Server Port: 50003
Connect Success
Download file name: 2019.mp4
Receive count of 1 !

LFTP client start...
please input command
LFTP lget 172.18.34.217 2018.mp4
lget
172.18.34.217
2018.mp4
Send first SYN from client
Receive second SYN from server
Server Port: 50002
Connect Success
Download file name: 2018.mp4
Receive count of 1 !
Receive count of 2 !

```

服务端收到请求，并开启两个不同线程进行传输，图中可看到交错顺序的序号，这是属于不同客户端的数据包。

2018.mp4	2018/12/3 21:22	MP4 文件	122,478 KB	Send count of 418!
2018.txt	2018/12/3 9:58	文本文档	183 KB	Send count of 727!
2019.mp4	2018/12/3 21:22	MP4 文件	122,478 KB	Send count of 419!
2020.flv	2018/12/2 23:03	FLV - Flash(FLV) ...	7,871 KB	Send count of 728!
2020.txt	2018/12/2 22:44	文本文档	365 KB	Send count of 420!
CommandAnalysis.class	2018/12/2 19:15	CLASS 文件	2 KB	Send count of 729!
CommandAnalysis.java	2018/12/2 23:06	Java 源文件	2 KB	Send count of 421!
CommandRegex.class	2018/12/2 19:32	CLASS 文件	2 KB	Send count of 730!
CommandRegex.java	2018/12/2 22:23	Java 源文件	2 KB	Send count of 422!
ReliablePacket.class	2018/12/2 13:03	CLASS 文件	3 KB	Send count of 731!
ReliablePacket.java	2018/12/2 12:51	Java 源文件	4 KB	Send count of 423!
test.class	2018/12/2 14:35	CLASS 文件	2 KB	Send count of 732!
UDPClient.class	2018/12/2 19:35	CLASS 文件	6 KB	Send count of 424!
UDPClient.java	2018/12/2 23:19	Java 源文件	8 KB	Send count of 733!
UDPServer.class	2018/12/3 10:16	CLASS 文件	3 KB	Send count of 425!
UDPServer.java	2018/12/3 10:15	Java 源文件	2 KB	Send count of 734!
UDPServerThread.class	2018/12/3 10:09	CLASS 文件	5 KB	Send count of 426!
UDPServerThread.java	2018/12/3 10:08	Java 源文件	8 KB	Send count of 735!
UDPUtils.class	2018/12/3 10:16	CLASS 文件	2 KB	Send count of 427!
				Send count of 736!
				Send count of 428!
				Send count of 737!
				Send count of 429!
				Send count of 738!
				Send count of 430!
				Send count of 739!

From address:/172.18.33.211 From port:50003 Receive first SYN from client Send second SYN from server Connect Success Send count of 294! Send count of 295! Send count of 296! Downloading from server. Receive the name of the file: 2019.mp4 Send count of 297! Send count of 298! Send count of 1! Send count of 299!	LFTP server start.... From address:/172.18.33.211 From port:50002 Receive first SYN from client Send second SYN from server Connect Success Downloading from server. Receive the name of the file: 2018.mp4 Send count of 1! Send count of 2! Send count of 3!
---	--

进行上传测试：同上，也是两个客户端程序同时向服务端上传文件，然后服务端进行接收。

客户端：


















Send count of 290! Send count of 291! Send count of 292! Send count of 293! Send count of 294! Send count of 295! Send count of 296! Send count of 297! Send count of 298! Send count of 299! Send count of 300! Send count of 301! Send count of 302! Send count of 303! Send count of 304! Send count of 305! Send count of 306! Send count of 307! Send count of 308! Send count of 309! Send count of 310! Send count of 311! Send count of 312! Send count of 313! Send count of 314! Send count of 315! Send count of 316! Send count of 317! Send count of 318!	Send count of 119! Send count of 120! Send count of 121! Send count of 122! Send count of 123! Send count of 124! Send count of 125! Send count of 126! Send count of 127! Send count of 128! Send count of 129! Send count of 130! Send count of 131! Send count of 132! Send count of 133! Send count of 134! Send count of 135! Send count of 136! Send count of 137! Send count of 138! Send count of 139! Send count of 140! Send count of 141! Send count of 142! Send count of 143! Send count of 144! Send count of 145! Send count of 146! Send count of 147!
--	--

处于不同的端口。


















```
LFTP client start...
please input command
LFTP lsend 172.18.34.217 2019.mp4
lsend
172.18.34.217
2019.mp4
Send first SYN from client
Receive second SYN from server
Server Port: 50003
Connect Success
Send file name: 2019.mp4
Send count of 1!
Send count of 2!

172.18.34.217
2018.mp4
Send first SYN from client
Receive second SYN from server
Server Port: 50002
Connect Success
Send file name: 2018.mp4
Send count of 1!
Send count of 2!
Send count of 3!
Send count of 4!
Send count of 5!
Send count of 6!
```

服务端进行接收，可看到交替顺序的序号，是两个客户端同时在传输。

	.classpath	2018/12/2 15:51	CLASSPATH 文件	1 KB	Receive count of 108 !
	.project	2018/12/2 15:51	PROJECT 文件	1 KB	Receive count of 278 !
	2018.mp4	2018/12/3 21:22	MP4 文件	0 KB	Receive count of 109 !
	2018.txt	2018/12/3 9:58	文本文档	183 KB	Receive count of 279 !
	2020.flv	2018/12/2 23:03	FLV - Flash(FLV) ...	7,871 KB	Receive count of 110 !
	2020.txt	2018/12/2 22:44	文本文档	365 KB	Receive count of 280 !
	CommandAnalysis.class	2018/12/2 19:15	CLASS 文件	2 KB	Receive count of 111 !
	CommandAnalysis.java	2018/12/2 23:06	Java 源文件	2 KB	Receive count of 281 !
	CommandRegex.class	2018/12/2 19:32	CLASS 文件	2 KB	Receive count of 112 !
	CommandRegex.java	2018/12/2 22:23	Java 源文件	2 KB	Receive count of 282 !
	ReliablePacket.class	2018/12/2 13:03	CLASS 文件	3 KB	Receive count of 113 !
	ReliablePacket.java	2018/12/2 12:51	Java 源文件	4 KB	Receive count of 283 !
	test.class	2018/12/2 14:35	CLASS 文件	2 KB	Receive count of 114 !
	UDPCClient.class	2018/12/2 19:35	CLASS 文件	6 KB	Receive count of 284 !
	UDPCClient.java	2018/12/2 23:19	Java 源文件	8 KB	Receive count of 115 !
	UDPServer.class	2018/12/3 10:16	CLASS 文件	3 KB	Receive count of 285 !
	UDPServer.java	2018/12/3 10:15	Java 源文件	2 KB	Receive count of 116 !

文件传输完毕，可以看到列表中出现了该文件。可以正常播放，播放过程详见 GitHub 视频。

	2018.mp4	2018/12/3 21:22	MP4 文件	122,478 KB	Receive count of 2268 !
	2018.txt	2018/12/3 9:58	文本文档	183 KB	Receive count of 2440 !
	2019.mp4	2018/12/3 21:22	MP4 文件	122,478 KB	Receive count of 2269 !
	2020.flv	2018/12/2 23:03	FLV - Flash(FLV) ...	7,871 KB	Receive count of 2441 !
	2020.txt	2018/12/2 22:44	文本文档	365 KB	Receive count of 2270 !
	CommandAnalysis.class	2018/12/2 19:15	CLASS 文件	2 KB	Receive count of 2442 !
	CommandAnalysis.java	2018/12/2 23:06	Java 源文件	2 KB	Receive count of 2271 !
	CommandRegex.class	2018/12/2 19:32	CLASS 文件	2 KB	Receive count of 2443 !
	CommandRegex.java	2018/12/2 22:23	Java 源文件	2 KB	Receive count of 2272 !
	ReliablePacket.class	2018/12/2 13:03	CLASS 文件	3 KB	Receive count of 2444 !
	ReliablePacket.java	2018/12/2 12:51	Java 源文件	4 KB	Receive count of 2273 !
	test.class	2018/12/2 14:35	CLASS 文件	2 KB	Receive count of 2445 !
	UDPCClient.class	2018/12/2 19:35	CLASS 文件	6 KB	Receive count of 2274 !
	UDPCClient.java	2018/12/2 23:19	Java 源文件	8 KB	Receive count of 2446 !
	UDPServer.class	2018/12/3 10:16	CLASS 文件	3 KB	Receive count of 2275 !
	UDPServer.java	2018/12/3 10:15	Java 源文件	2 KB	Receive count of 2447 !
	UDPServerThread.class	2018/12/3 10:09	CLASS 文件	5 KB	Receive count of 2276 !

2018.mp4 has been received

丢包测试，由于我们是在校园网内进行测试，所以很难出现丢包的情况，所以我手动设置了一个丢包触发条件，进行测试。

如图：客户端上传是遇到了丢包，然后超时重传，服务端正常接收。

```
LFTP lsend 172.18.34.217 2018.txt
lsend
172.18.34.217
2018.txt
Send first SYN from client
Receive second SYN from server
Server Port: 50002
Connect Success
Send file name: 2018.txt
Send count of 1!
Send count of 2!
Resend the lost packet.
Send count of 3!
Send count of 4!
```

D:\学习\大三上\计算机网络\Homework\TCPPa

```
D:\课程资料\大三上\LFTP\TCPBasedOnUDP\LFTP新开>java UDPServer
LFTP server start...
From address:/172.18.33.211
From port:50003
Receive first SYN from client
Send second SYN from server
Connect Success
Uploading to server.
Receive the name of the file: 2018.txt
Receive count of 1 !
Receive count of 2 !
Receive count of 3 !
Receive count of 4 !
2018.txt has been received
```

代码模块

本次实验我们采用 JAVA 语言。

ReliablePacket.java

该模块用于为 UDP 包生成校验和以及将 sequenceNumber, rwnd, cwnd 等信息插入数据流中，然后返回一个扩展的 byte 数组，依次作为参数发给另一端进行接收。另一端同样调用此类中的函数进行解析，将扩展的 byte 数组解析为原数据并进行数据包的校验。

UDPUtills.java

工具类，用于客户端与服务端都会用到的提示信息，以此作为当前连接状况的判断条件。

CommandRegex.java

用正则表达式判断命令输入正误。

```
CommandAnalysis.java
```

用文件查找判断命令正误，以及将命令拆分。

```
UDPServer.java
```

服务端主线程。

```
UDPServerThread.java
```

服务端子线程。

```
UDPClient.java
```

客户端代码

使用说明

```
客户端 javac UDPClient.java && java UDPClient
```

```
服务端 javac UDPServer.java && java UDPServer
```

实验分工

张星主要负责客户端部分，还包括首部，校验和，以及命令提示的编写。

姚雪辉主要负责服务端部分，还包括命令格式以及文件的判断编写。

README

由于上传了视频，所以使得下载变得麻烦，所以直接上传了张星姚雪辉.zip，其中不包含视频，但是包含了所有的源代码截图以及实验报告，所以想要运行程序直接下载该文件即可。