



# 《计算机网络》 测试文档 (LFTP)

学 生 姓 名 :

黄梓轩

学 号 :

16340090

时 间 :

2018 年 12 月 4 日

## 一. 引言

本文档为LFTP项目的测试文档，主要介绍项目的使用方法，测试的环境、数据和流程等等的基本情况，以及项目中各个要求的完成情况。

## 二. 测试环境

服务端为阿里云服务器，操作系统为Ubuntu 14.04.5 LTS，带宽仅为1m，ip地址为120.77.172.46，装有python 2.7.6。

客户端为自己的电脑，操作系统为Win10，带宽为校园网带宽，装有python 2.7.15，13001-13099端口开启了udp服务。

（本程序只用到了服务端13001-13003三个端口，由于带宽限制，本服务器仅支持2个客户端，要想运行更多可以改变server.py的supportPortNum。13001为监听客户端的端口，若要修改，请修改《server.py》的recv\_port变量）

## 三. 使用方法

根据实验项目设计要求，使用方法如下。

首先将4个文件《server.py》，《sender.py》，《receiver.py》，《dataSolver.py》这四个文件上传到服务端某个文件夹中，如下图

```
root@iZwz95zbktshj9i0c7il6sZ:~/hzx# ls
dataSolver.py receiver.py sender.py server.py
```

将4个文件《client.py》，《sender.py》，《receiver.py》，

《dataSolver.py》放在自己的电脑的某个文件夹中。

 <b>dataSolver.py</b> 类型: PY 文件	修改日期: 2018/11/23 11:00 大小: 2.37 KB
 <b>receiver.py</b> 类型: PY 文件	修改日期: 2018/11/24 11:05 大小: 2.40 KB
 <b>sender.py</b> 类型: PY 文件	修改日期: 2018/11/24 17:04 大小: 3.70 KB
 <b>client.py</b> 类型: PY 文件	修改日期: 2018/11/23 23:58 大小: 2.41 KB

服务端输入下述命令即可启动服务器。

```
python server.py
```

客户端按照老师要求，可以输入以下两条命令发送文件或接受文件。

```
python client.py lsend ip mylargefile
```

```
python client.py lget ip mylargefile
```

ip为服务器的ip地址，我这里应为120.77.172.46，mylargefile为发送文件的路径，以下为示例。

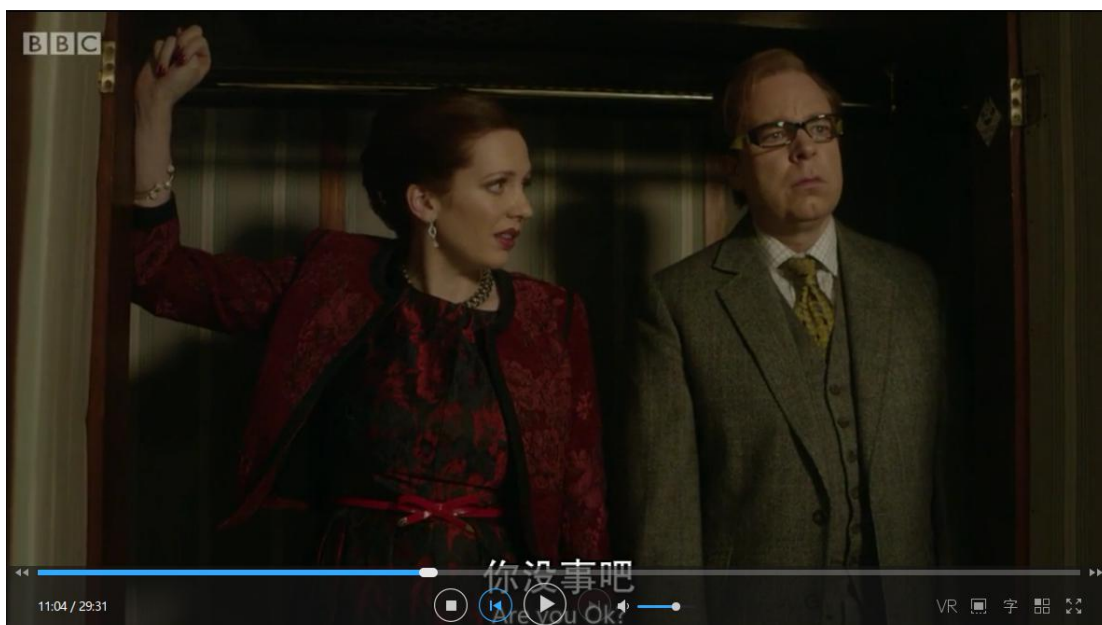
```
python client.py lsend 120.77.172.46 1.bmp
```

## 四. 测试数据

发送和接受一张bmp图片，文件大小为11.5 MB



发送和接受一个mkv视频，文件大小为273 MB（由于网速太慢，因此文件大小尽量小一点）



## 五. 测试流程

先在云服务器上运行服务器。

```
root@iZwz95zbktshj9i0c7il6sZ:~/hzx# python server.py  
waiting on port: 13001
```

部分debug提示信息：

端口繁忙：

当服务端可用端口都被占用了，服务端向客户端发送报文提示没有可用的端口并终止程序。

```
C:\Users\de11\Desktop\LFTP>python client.py lsend 120.77.172.46 1.bmp  
2018-12-04 13:15:13  
no port available , stop  
2018-12-04 13:15:13
```

发送文件不存在：

当所要发送的文件不存在，提示文件不存在并终止客户端程序，不发送报文到客户端。

```
C:\Users\de11\Desktop\LFTP>python client.py lsend 120.77.172.46 3.bmp  
2018-12-04 13:16:09  
the send file is not exit , stop sending  
2018-12-04 13:16:09
```

接收文件不存在：

当所要接收的文件不存在，服务端发送报文给客户端提示文件不存在并终止客户端程序，客户端收到报文并得知文件大小不存在后，结束客户端程序。

```

root@iZwz95zbktshj9i0c7il6sZ:~/hzx# python server.py
waiting on port: 13001
addr ('120.236.174.169', 56837)
file not exit , stop to send file
command is get , file is 2.bmp

```

```

C:\Users\de11\Desktop\LFTP>python client.py lget 120.77.172.46 2.bmp
2018-12-04 13:17:46
the file you want to get is not exit , stop requesting
2018-12-04 13:17:46

```

## 验证一对一发送:

先测试一下只有一个客户端的运行效果, 发送1.bmp (11m)

```

C:\Users\de11\Desktop\LFTP>python client.py lsend 120.77.172.46 1.bmp
2018-12-03 20:09:37
new addr ('120.77.172.46', 13002)
RTT is 0.5 congWin is 1024 sendPackNum 1 flowControlWindowSize 153600 recentAckNum 0
RTT is 0.5 congWin is 2048 sendPackNum 2 flowControlWindowSize 152592 recentAckNum 1008
RTT is 0.5 congWin is 4096 sendPackNum 4 flowControlWindowSize 152592 recentAckNum 3024
RTT is 0.5 congWin is 8192 sendPackNum 8 flowControlWindowSize 152592 recentAckNum 7056
RTT is 0.5 congWin is 16384 sendPackNum 16 flowControlWindowSize 152592 recentAckNum 15120
RTT is 0.5 congWin is 32768 sendPackNum 32 flowControlWindowSize 152592 recentAckNum 31248
RTT is 0.5 congWin is 65536 sendPackNum 64 flowControlWindowSize 152592 recentAckNum 63504
RTT is 0.5 congWin is 130048 sendPackNum 127 flowControlWindowSize 152592 recentAckNum 127008
RTT is 0.5 congWin is 208896 sendPackNum 137 flowControlWindowSize 141248 recentAckNum 204624
RTT is 0.5 congWin is 300032 sendPackNum 122 flowControlWindowSize 124969 recentAckNum 294336

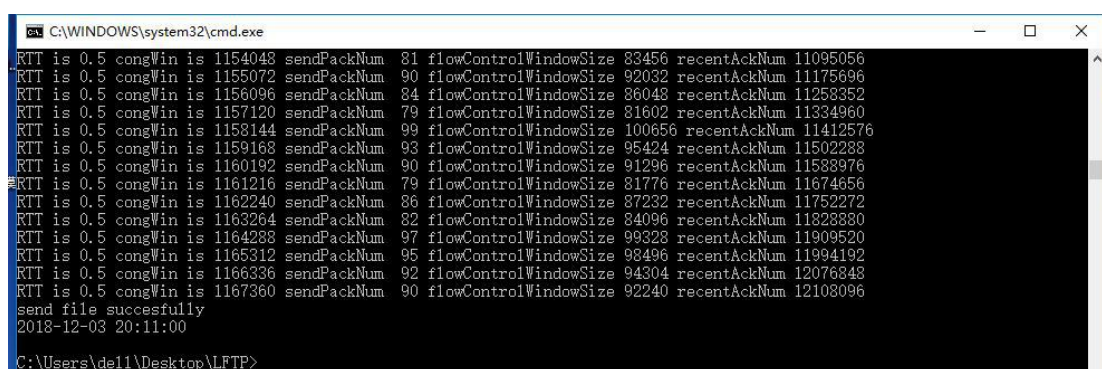
```

上图可以看到运行时间为20:09:37, 刚运行的时候, 第一条信息是服务端处理该客户端的ip和端口, 初始客户端监听13001端口, 而客户端发送请求到该端口后, 服务端创建一个新进程监听剩下的可用的端口(13002), 并告知客户端, 由此图可知客户端已经知道新的端口是13002了。

后面的是每隔一个RTT输出一次的debug信息, RTT是固定的0.5(由于程序读取文件, 发送等等过程需要时间, 实际RTT要比0.5大), 当前的congWin(阻塞控制窗口), sendPackNum(该RTT发送的包的数量), flowControlWindowSize(流控制窗口), recentAckNum(最近收到的ack值), 开始的时候发包限制主要是阻塞控制窗口, 初始的发包指数增长, 后来增速变慢, 这主要是服务端的带宽限制导致。



最后发送完毕后，可以看到结束时间为20:11:00，起始时间是20:09:37，传输时间共83s，由于传输文件1.bmp共11827kb大小，计算得传输的粗略速度为142.5KB/s，云服务器带宽只有1mb/s（约125KB/s左右），因此也大致符合。



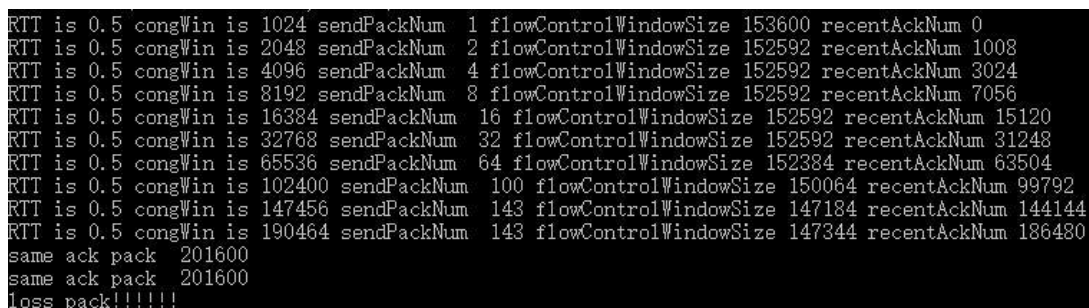
```

C:\WINDOWS\system32\cmd.exe
RTT is 0.5 congWin is 1154048 sendPackNum 81 flowControlWindowSize 83456 recentAckNum 11095056
RTT is 0.5 congWin is 1155072 sendPackNum 90 flowControlWindowSize 92032 recentAckNum 11175696
RTT is 0.5 congWin is 1156096 sendPackNum 84 flowControlWindowSize 86048 recentAckNum 11258352
RTT is 0.5 congWin is 1157120 sendPackNum 79 flowControlWindowSize 81602 recentAckNum 11334960
RTT is 0.5 congWin is 1158144 sendPackNum 99 flowControlWindowSize 100656 recentAckNum 11412576
RTT is 0.5 congWin is 1159168 sendPackNum 93 flowControlWindowSize 95424 recentAckNum 11502288
RTT is 0.5 congWin is 1160192 sendPackNum 90 flowControlWindowSize 91296 recentAckNum 11588976
RTT is 0.5 congWin is 1161216 sendPackNum 79 flowControlWindowSize 81776 recentAckNum 11674656
RTT is 0.5 congWin is 1162240 sendPackNum 86 flowControlWindowSize 87232 recentAckNum 11752272
RTT is 0.5 congWin is 1163264 sendPackNum 82 flowControlWindowSize 84096 recentAckNum 11838880
RTT is 0.5 congWin is 1164288 sendPackNum 97 flowControlWindowSize 99328 recentAckNum 11909520
RTT is 0.5 congWin is 1165312 sendPackNum 95 flowControlWindowSize 98496 recentAckNum 11994192
RTT is 0.5 congWin is 1166336 sendPackNum 92 flowControlWindowSize 94304 recentAckNum 12076848
RTT is 0.5 congWin is 1167360 sendPackNum 90 flowControlWindowSize 92240 recentAckNum 12108096
send file sucesfully
2018-12-03 20:11:00
C:\Users\del1\Desktop\LFTP>

```

## 验证一对多发送：

然后在本地测试运行两个客户端程序。



```

RTT is 0.5 congWin is 1024 sendPackNum 1 flowControlWindowSize 153600 recentAckNum 0
RTT is 0.5 congWin is 2048 sendPackNum 2 flowControlWindowSize 152592 recentAckNum 1008
RTT is 0.5 congWin is 4096 sendPackNum 4 flowControlWindowSize 152592 recentAckNum 3024
RTT is 0.5 congWin is 8192 sendPackNum 8 flowControlWindowSize 152592 recentAckNum 7056
RTT is 0.5 congWin is 16384 sendPackNum 16 flowControlWindowSize 152592 recentAckNum 15120
RTT is 0.5 congWin is 32768 sendPackNum 32 flowControlWindowSize 152592 recentAckNum 31248
RTT is 0.5 congWin is 65536 sendPackNum 64 flowControlWindowSize 152384 recentAckNum 63504
RTT is 0.5 congWin is 102400 sendPackNum 100 flowControlWindowSize 150064 recentAckNum 99792
RTT is 0.5 congWin is 147456 sendPackNum 143 flowControlWindowSize 147184 recentAckNum 144144
RTT is 0.5 congWin is 190464 sendPackNum 143 flowControlWindowSize 147344 recentAckNum 186480
same ack pack 201600
same ack pack 201600
loss pack!!!!!!

```

最后几行可以看到，由于带宽限制，每个RTT发送143个包的话已经发生丢包了，三次重复的ack，直接提示丢包。

再看下图，当传输久了后，接收方缓冲区已经开始变小了，尽管阻塞控制窗口很大，但发送的包的数量还是根据流控制窗口而设定。同时也能看到下图中每个RTT才发送40-60个包，这主要是因为此时开启了两个客户端程序，因此原本就低的带宽变得更低了。

```

RTT is 0.5 congWin is 90880 sendPackNum 70 flowControlWindowSize 72528 recentAckNum 1548288
RTT is 0.5 congWin is 91904 sendPackNum 68 flowControlWindowSize 70544 recentAckNum 1592640
RTT is 0.5 congWin is 92928 sendPackNum 69 flowControlWindowSize 71312 recentAckNum 1633968
RTT is 0.5 congWin is 93952 sendPackNum 64 flowControlWindowSize 66464 recentAckNum 1674288
RTT is 0.5 congWin is 94976 sendPackNum 63 flowControlWindowSize 65424 recentAckNum 1713600
RTT is 0.5 congWin is 96000 sendPackNum 63 flowControlWindowSize 64640 recentAckNum 1752912
RTT is 0.5 congWin is 97024 sendPackNum 62 flowControlWindowSize 64224 recentAckNum 1791216
RTT is 0.5 congWin is 98048 sendPackNum 64 flowControlWindowSize 66000 recentAckNum 1830528
RTT is 0.5 congWin is 99072 sendPackNum 61 flowControlWindowSize 63296 recentAckNum 1869840
RTT is 0.5 congWin is 100096 sendPackNum 58 flowControlWindowSize 59743 recentAckNum 1913184
RTT is 0.5 congWin is 101120 sendPackNum 56 flowControlWindowSize 57669 recentAckNum 1951488
RTT is 0.5 congWin is 102144 sendPackNum 48 flowControlWindowSize 49872 recentAckNum 1991808
RTT is 0.5 congWin is 103168 sendPackNum 47 flowControlWindowSize 48925 recentAckNum 2032128
RTT is 0.5 congWin is 104192 sendPackNum 45 flowControlWindowSize 46512 recentAckNum 2070432

```

### 验证一对多接受:

文件虽然已经传输上服务器了，但传输的文件是视频和图片，因此无法查看正确与否，因此再启动客户端程序从服务端接受文件。

```

C:\Users\de11\Desktop\LFTP>python client.py lget 120.77.172.46 1.bmp
2018-12-03 21:50:29
begin to recv file 1.bmp size is 12110454
seqNum 486864 != ackNum 485856
seqNum 488880 != ackNum 485856
seqNum 490896 != ackNum 485856

```

可以看到接受文件的时候输出的debug信息，包括开始时间，接收文件的名字和大小，以及乱序或丢包时导致的seq Num != ack Num的情况。

最后传输完成后，程序关闭。程序在21:50:29开始，在21:52:22结束，耗时113s，传输文件1.bmp共11827kb大小，传输速率大概为104.7KB/s，与之前的上传速度相比稍慢，但还是与服务器1m带宽大致相符。

```

file has already recieved succesfully
file has already written in the disk
2018-12-03 21:52:22

```

打开一下传输成功的两个文件1.mkv和1.bmp，发现没有错误。能



够完整显示，证明程序没有问题。

