

# Lab 3 Writeup

---

My name: [徐研]

My SUNet ID: [211240047]

I collaborated with: []

I would like to thank/reward these classmates for their help: [czy]

This lab took me about [9] hours to do. I [did] attend the lab session.

## Program Structure and Design of the TCPSender:

- **对于fill\_window函数：**即先判断是否需要发送syn信号，然后在发送完信号后，在还没有将window填满之前，将stream里面的数据包装成segment发送。并且要根据stream的上层输入状态判断这个时候要不要发送fin信号，发送完fin信号后不再发送其余数据了
- **ack\_received函数：**即如果收到了比之前ack过的序号还大的ack序号，那么这个时候将发送但未ack中的TCPSegment逐一验证看是否已经收到ack，并将窗口往后移动
- **tick函数：**主要是自己添加实现的TCPTimer要能够记录距离上一次开始计时过去了多久，然后在每次调用时查看是否expired了，如果是的话需要重新发送sent但未ack中seqno最小的segment
- 其余函数照着tutorial的要求写即可

## Implementation Challenges:

- 首先是对于发送fin和syn的时机把握不对，导致测试时处于的state不正确，后面查看了tutorial上的状态转移图更正。并且在不断试验中发现，syn需要单独发送，不能和任何一个包一起（必须先告诉receiver表示要开始发送内容了）。但是fin可以和其他包一起，但是需要判断加上fin长度为1，有没有超过当前的window\_size，需要注意
- 一开始忘记在发送syn的时候设定计时器了，导致test13 failed了，即每次发包都要设定计时器
- 对window\_size的组成理解有误，应该是发送了但是还没有ack的包+准备但是还没发送的包=window\_size
- fin不能每次都单独发送，否则在测验时会遇到“Piggyback FIN in segment when space is available”这个错误，如果能和之前一个包一起发那就一起发，否则放到下一轮单独发

- 在装填window\_size时，如果读到的包的大小为0，要及时从while循环里退出来，否则可能会死循环。考虑到的可能情况是：这个时候stream的size为0，但是还没有input\_ended，所以也还没有eof，故没有发送信号，就会一直满足小于window\_size情况，陷到死循环里
- 刚开始判断结束发包要么是填满了window\_size，要么是stream里面没有东西了，但是其实要根据是否发送了fin信号判断。因为有可能fin信号在下一轮才能发送，但此时stream里面没东西了，导致fin信号无法继续发送，错误
- 在ack\_received部分，对于序列号的计算都要转化成uint64\_t计算，否则可能会溢出
- 一开始只判断如果参数ack小于之前已经判断过的ack值，那么直接返回。但是还需要判断参数ack小于\_next\_seqno值。不然的话如果参数ack发过来一个很大的数，那么所有outstanding\_segment全都以为已经被acked了，但是这个参数其实是无效的，如果不在特定范围内，需要直接返回
- 遇到了“When filling window, treat a '0' window size as equal to '1' but don't back off RTO”的错误提醒，后面意识到如果window\_sz为0，那么不能double RTO。但是因为每次这种情况下会把window\_sz边为1，所以均为double。之后在每次收到window\_size时，有单独的判断记录这次收到的window\_size是0还是不是，下一次是否需要double RTO

## Remaining Bugs:

暂无

- Optional: I had unexpected difficulty with: [describe]
- Optional: I think you could make this lab better by: [可以每次直接git pull上一次代码下来，不需要手动复制]
- Optional: I was surprised by: [describe]
- Optional: I'm not sure about: [其中我用于判断是否要back off RTO的布尔变量should\_double，如果一开始赋值为false，会导致text13错误，但是改成true能过test。但是每一次ack\_received都会重新赋值，不太理解初始化对其的影响]

## 通过样例截图

```
33/33 Test #55: t_socket_dt ..... Passed    0.01 sec

100% tests passed, 0 tests failed out of 33

Total Test time (real) =  1.89 sec
[100%] Built target check_lab3
```