

Lab 5 Writeup

My name: 李易

My Student ID: 191830079

This lab took me about 25 hours to do. I did attend the lab session.

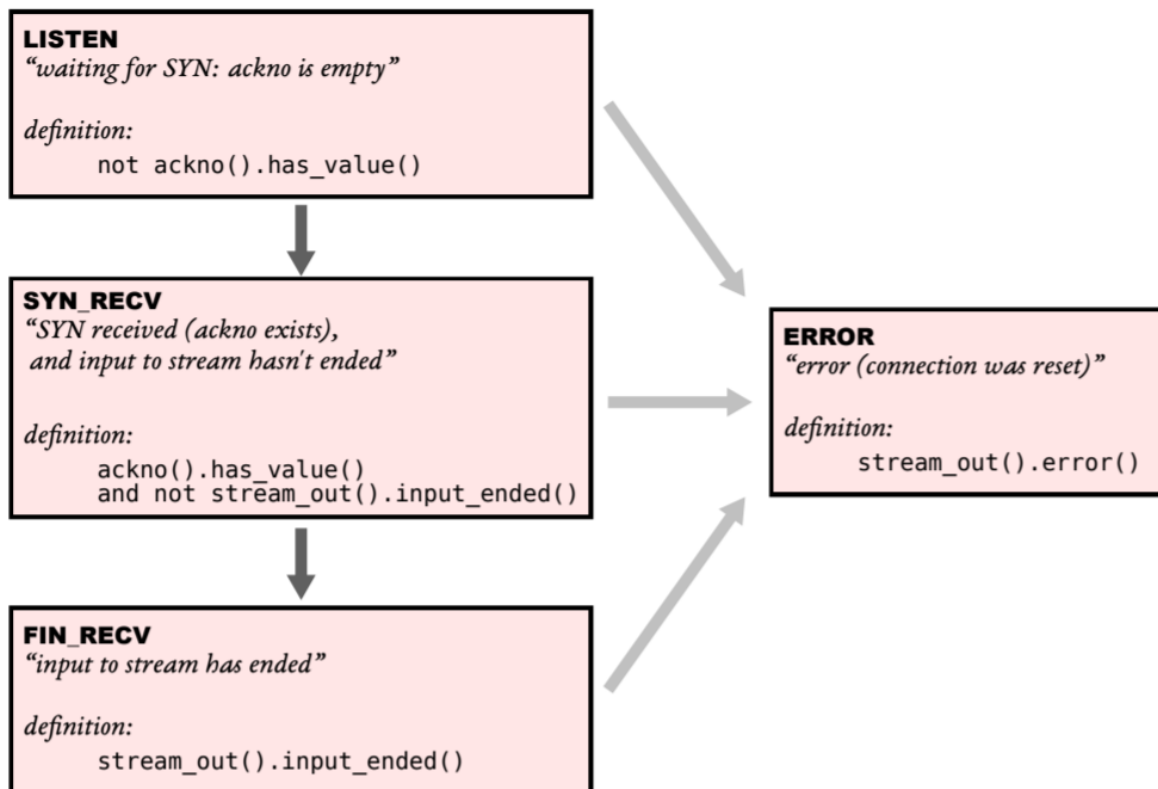
Program Structure and Design of the TCPConnection:

这是至今最为复杂的一个lab，代码量看上去并不大，但涉及的逻辑细节很多，很容易踩坑。

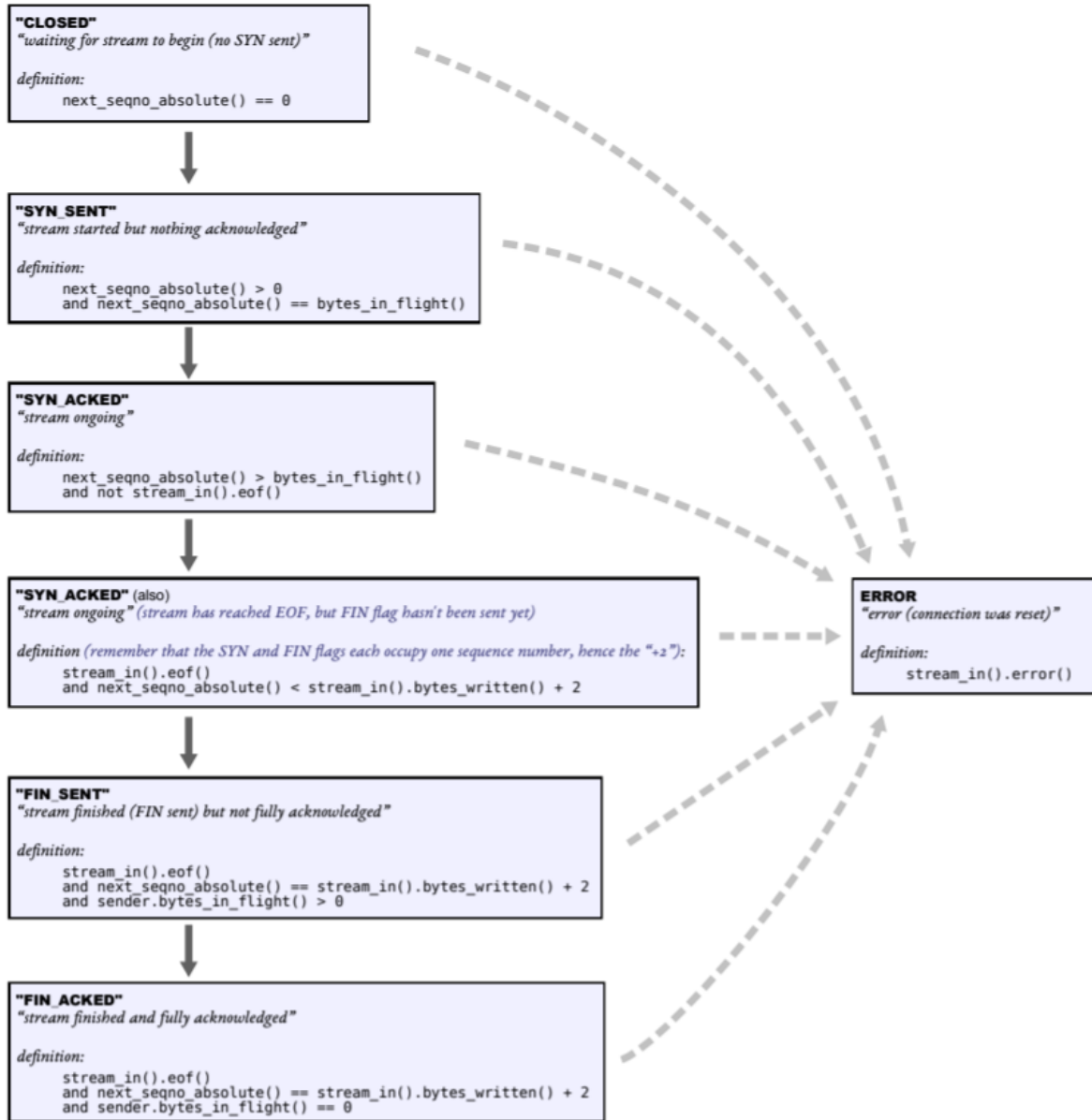
实现部分主要包括接收segment时的函数逻辑、两种shutdown的函数逻辑。

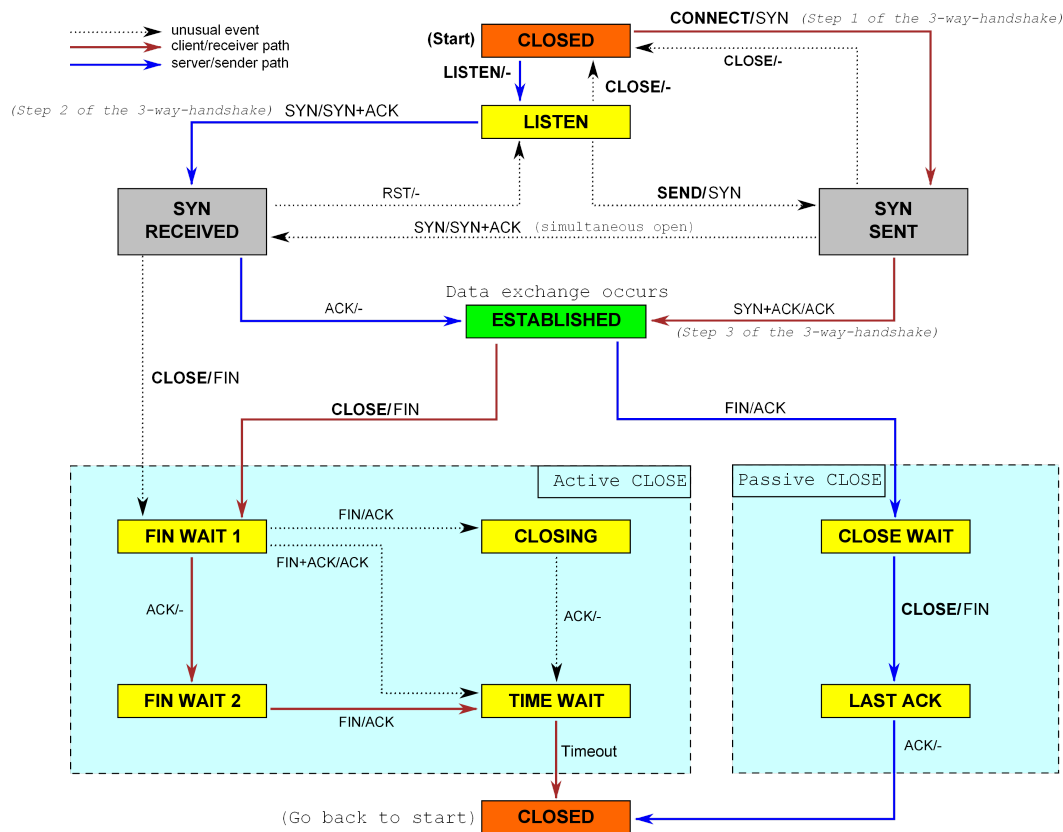
前一部分主要涉及前几个lab中给出的状态机示意图，需要把receiver和sender在建立连接时三次握手的过程彻底理解，尤其要注意在出现rst时的特殊状况，需要作为边界情况特别处理。

Evolution of the TCP receiver (as tested by the test suite)



Evolution of the TCP sender (as tested by the test suite)





第二部分的逻辑反而简单一些，因为讲义上说得非常清楚，细节也并不多。unclean_shutdown函数需要在设置发送方、接收方error之后将含rst的segment发出去（push到_segments_out里），由于调用unclean_shutdown的情况不同（比如超过最大重传次数的时候就对当前segment置rst即可），所以看情况调用fill_window()和send_empty_segment()方法生成新segment。

Implementation Challenges:

难点在于接收segment时建立连接的逻辑细节和clean_shutdown的理解。

segment_received中实现复杂之处在于需要同时从client和server两个角度去思考，同时要利用sender和receiver两者的接口。可以按照状态机提示来一步一步完成这个函数。

初始，发送方处于close状态，接收方处于Listen状态，发送方利用connect发送syn包（ack=0），从而使得client进入SYN_SENT状态（这里client因为是主动连接）。此时发送的工作由fill_window()完成。

```
if(!_syn_sent)
{
    TCPSegment seg;
    _syn_sent = true;
    seg.header().syn = true;
    _segment_handler(seg);
    return;
}
```

接收方收到syn包后，发送syn=1，ack=1的空包，进入SYN_RECV状态。（同样利用connect()），这是被动连接的情况。

此时主动发起连接的对等方只接受syn=1，ack=1的空包，否则若ack=0，则是两者同时主动发起连接，这时转变身份，发送syn=1，ack=1的空包，进入SYN_RECV状态即可。

```
if(SYN_SENT && !_receiver.ackno().has_value())
```

```

{
    //only accept empty segment with ack
    if(seg.payload().size() > 0 || !seg.header().ack)
    {
        //simultaneously open
        if(!seg.header().ack && seg.header().syn)
        {
            _receiver.segment_received(seg);
            //send empty seg with ack,syn to get further ack(established)
            _sender.send_empty_segment();
        }
        return;
    }
}
}

```

后续逻辑就比较简单，按讲义上来就好。被动方再收到一个含ack的包就代表第三次握手也完成，连接进入established状态。

clean_shutdown部分要理解讲义中四个条件的相关论述。针对本地TCPConnection无法确认的第四个条件，我们有两种处理方法：

1. linger after stream finish($10 * \text{_cfg.rt_timeout}$)
2. 如果在本地端sender的bytestream结束之前（应用层调用end_input()并且stream为空，读完了），receiver就已经结束（input_ended()代表此时收到了FIN），说明对等方(remote peer)想主动结束，这时本地端再发送的segment将有最大的seqno(比之前都大)，这时由于满足第三条条件（The outbound stream has been fully acknowledged by the remote peer），remote peer必然发送了针对这个segment的ack，故也就确认了之前所有的segment。

需要注意不管上述哪个都必须建立在条件一二三同时满足的基础上。

```

(_sender.stream_in().eof() && _receiver.stream_out().input_ended() &&
 !_sender.bytes_in_flight())

```

Remaining Bugs:

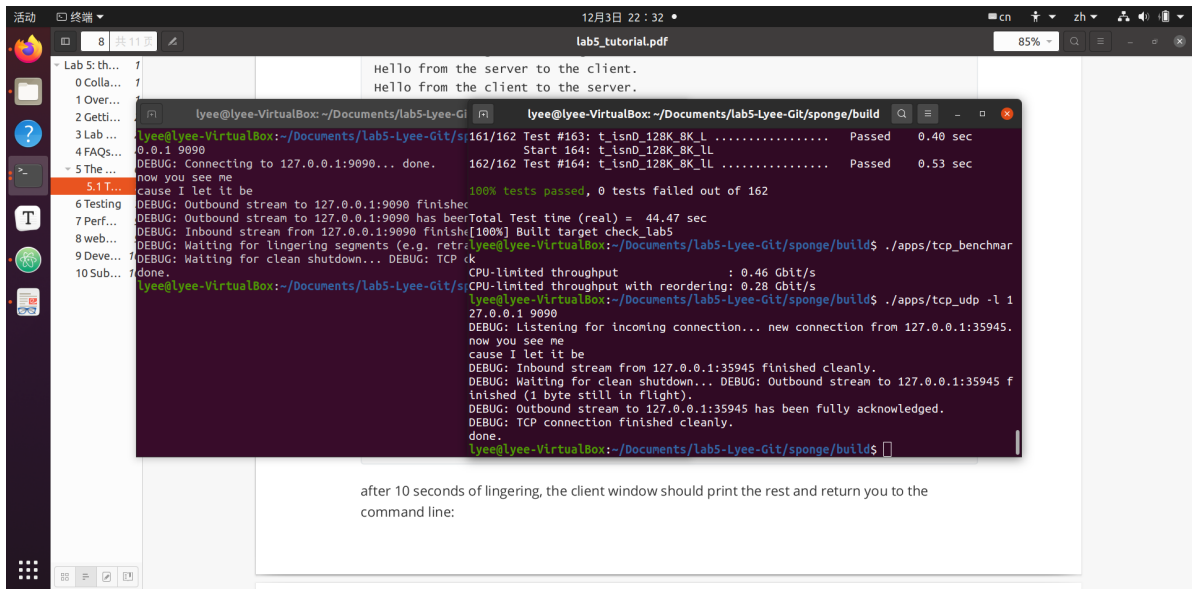
```

lyee@lyee-VirtualBox: ~/Documents/lab5-Lyee-Git/sponge/build
Start 158: t_isnS_128K_8K_LL
156/162 Test #158: t_isnS_128K_8K_LL ..... Passed 0.24 sec
Start 159: t_isnR_128K_8K_L
157/162 Test #159: t_isnR_128K_8K_L ..... Passed 0.60 sec
Start 160: t_isnR_128K_8K_L
158/162 Test #160: t_isnR_128K_8K_L ..... Passed 0.28 sec
Start 161: t_isnR_128K_8K_LL
159/162 Test #161: t_isnR_128K_8K_LL ..... Passed 1.12 sec
Start 162: t_isnD_128K_8K_L
160/162 Test #162: t_isnD_128K_8K_L ..... Passed 0.70 sec
Start 163: t_isnD_128K_8K_L
161/162 Test #163: t_isnD_128K_8K_L ..... Passed 0.40 sec
Start 164: t_isnD_128K_8K_LL
162/162 Test #164: t_isnD_128K_8K_LL ..... Passed 0.53 sec

100% tests passed, 0 tests failed out of 162

Total Test time (real) = 44.47 sec
[100%] Built target check_lab5
lyee@lyee-VirtualBox:~/Documents/lab5-Lyee-Git/sponge/build$ ./apps/tcp_benchmark
CPU-limited throughput : 0.46 Gbit/s
CPU-limited throughput with reordering: 0.28 Gbit/s
lyee@lyee-VirtualBox:~/Documents/lab5-Lyee-Git/sponge/build$

```



```
lyee@lyee-VirtualBox: ~/Documents/lab5-Lyee-Git/sponge/build
[100%] Testing webget...
Test project /home/lyee/Documents/lab5-Lyee-Git/sponge/build
Start 31: t_webget
1/1 Test #31: t_webget ..... Passed 0.63 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) = 0.63 sec
[100%] Built target check_webget
lyee@lyee-VirtualBox: ~/Documents/lab5-Lyee-Git/sponge/build$ ./apps/webget www.n
etlab4njucs.top /ALPHA
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Fri, 03 Dec 2021 15:42:16 GMT
Content-Type: application/octet-stream
Content-Length: 45
Last-Modified: Fri, 27 Aug 2021 01:46:38 GMT
Connection: close
ETag: "612843fe-2d"
Accept-Ranges: bytes

ALPHA,NJUer!
Welcome to computer networking!
lyee@lyee-VirtualBox: ~/Documents/lab5-Lyee-Git/sponge/build$
```

还可以通过更改之前的lab内容进一步改进性能。