计算机网络实验四：TCP协议的实现

郑懿 未央-水木01 2020012859

1 实现的功能

1.1 TCP的三次握手

```
54 9.262604037    10.0.2.15         182.61.200.7      TCP        54 61992 → 80 [SYN] Seq=0 Win=8192 Len=0
55 9.295236547    182.61.200.7      10.0.2.15         TCP        60 80 → 61992 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
56 9.296651312    10.0.2.15         182.61.200.7      TCP        54 61992 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
57 9.313690880    10.0.2.15         182.61.200.7      HTTP      131 GET / HTTP/1.1
58 9.314089202    182.61.200.7      10.0.2.15         TCP        60 80 → 61992 [ACK] Seq=1 Ack=78 Win=65535 Len=0
```

1.2 双向发送较小规模的数据

按照助教要求，不在此贴截图，已经自测`curl www.baidu.com`成功。

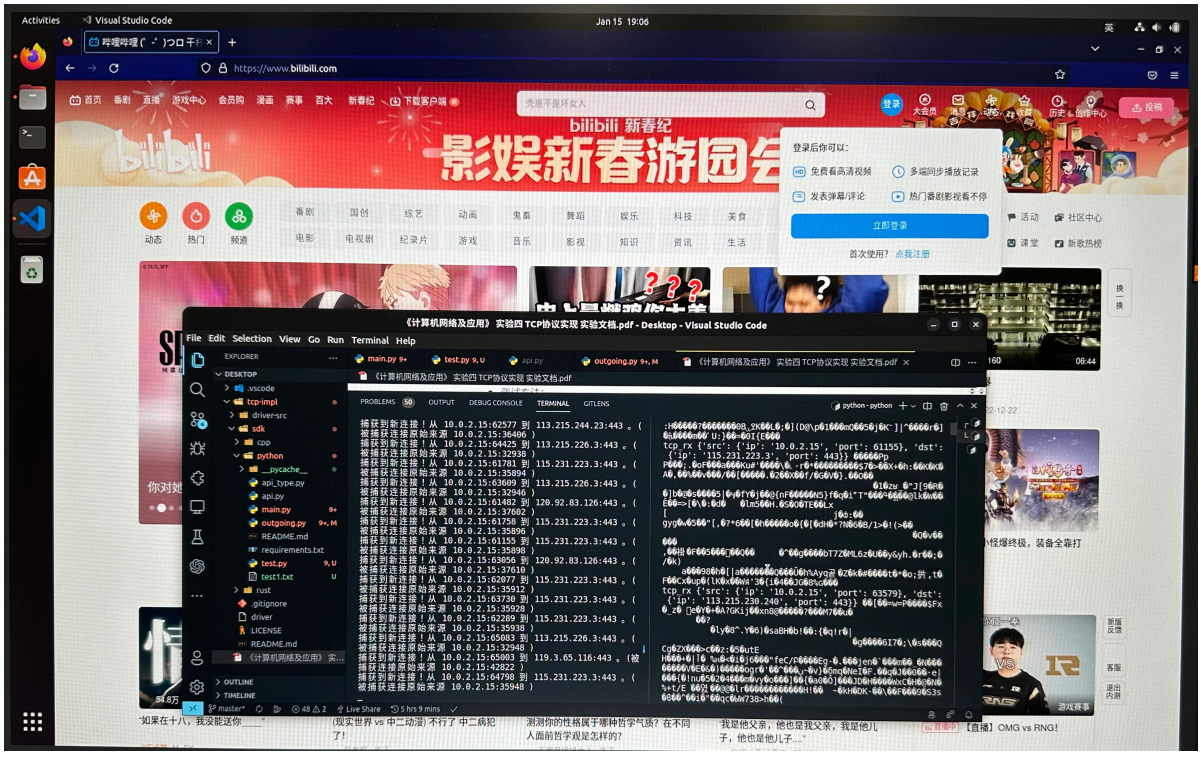
1.3 TCP的四次挥手

```
61 9.350794051    10.0.2.15         182.61.200.7      TCP        54 61992 → 80 [FIN, ACK] Seq=78 Ack=2782 Win=8192 Len=0
62 9.351157405    182.61.200.7      10.0.2.15         TCP        60 80 → 61992 [ACK] Seq=2782 Ack=79 Win=65535 Len=0
63 9.380267543    182.61.200.7      10.0.2.15         TCP        60 80 → 61992 [FIN, ACK] Seq=2782 Ack=79 Win=65535 Len=0
64 9.381181707    10.0.2.15         182.61.200.7      TCP        54 61992 → 80 [ACK] Seq=79 Ack=2783 Win=8192 Len=0
```

1.4 中等规模数据的上传和下载

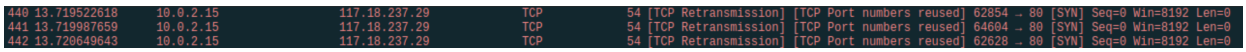
按照助教要求，不在此贴截图，都已经自测通过。

1.5 通过浏览器访问网页



VSCode中左侧运行的是`sudo ./driver`，右侧运行`python main.py`，上方打开了哔哩哔哩网站。

1.6 超时重传



2 实现思路简述

实现的关键是要搞清楚状态之间的转换。首先根据文档中的提示，我将ConnectionIdentifier类型转换为了一个四元组来作为每个连接的标识符。然后，我将要实现的函数打包在了一个类中，并在主函数部分提供了main文件访问的接口，这样做可以使得同时发起多个连接是可能的。另外，本实验中使用scapy库自动生成了报文。到此，预备部分结束。

先从简单的开始，简单的四个主要是app_connect，app_send，app_fin和app_rst。app_connect打包好一个SYN报文段，将其放入发送缓存区，视情况开启定时器并将报文段发出，将状态改为SYN_SENT。app_send直接将数据打包好，将其加入发送缓存区，视情况开启定时器并将其发出。app_fin函数打包好一个FIN报文，然后将其放入发送缓存区，视情况开启定时器并将其发出，将状态改为FIN_WAIT_1。app_rst函数打包好一个RST报文，然后将其发出（不需要放缓存区，因为直接关闭了）。以上四个函数都是基础的操作，打包报文并发出。

然后是最为复杂的tcp_rx，需要分状态讨论。优先级最高的是对面发来RST的情况，这时候直接关闭并通知应用层即可。然后处理SYN_SENT状态下收到正确的SYNACK的情况，此时需要回复ACK并且将状态改为ESTABLISHED。剩下的情况我们首先判断发来的ACK是否会导致发送缓存区的移动，因为TCP是累计确认的，如果收到一个ACK，应该将发送缓存区中序号在ACK以前的都清除。然后我们检查发来的报文段是否带有报头以外的数据，如果有，且未失序，则递交给应用层，且发回一个ACK。最后我们来分析状态的转移，如果是一个FIN报文且序号对，则必然需要回复一个ACK，并且按照原有状态的不同分别进行相应的状态改变；如果序号不对，直接丢弃。最后剩下的就是其他情况下普通的ACK，如果序号对，分情况改变状态；如果序号不对，丢弃。

为了避免使用多线程编程，在类内维护了一个tick函数，不断检查是否超时，如果超时则重传发送缓存区的第一个报文段。类外会有一个面向main文件的接口tick，每次调用当前存在的所有连接对象内部的tick。

最后，在类外维护main文件可能调用的其他函数接口，并将所有的连接存储在一个列表中，就实现了TCP连接的所有功能。

（注：只阅读这里的文字版思路可能会有点不太好理解，具体思路都已经以注释的形式标注在代码旁，结合代码理解效果更好。）

3 我认为的亮点

- 用类进行了包装，在类外维护了main文件可以调用的接口，实现多连接可以并存
- 用一种自己可以理解清楚的方式实现了状态转移的代码

4 我学到的东西

- 代码能力和调试能力提升了很多
- 锻炼了耐心
- 更加深入地理解了TCP的所有状态间的转移
- 了解了wireshark抓包工具的使用
- 更加深入的了解了TCP的报头以及伪首部等部分
- 对程序设计中的字节、位、字符串以及各种进制的数之间的关系更加清晰
- 对大端序、小端序有了更深的了解

5 我的建议

- 难度真的有点大，不仅考验课程知识，更是非常考验代码能力和信息检索能力
- 学习曲线比较陡峭，从零开始到大概读懂文档可能都需要很久，从大概写出第一版到最终bug-free会更久
- 可能需要更多的时间，一周也许不太够
- 希望可以放在学期中间而不是期末之后，考完期末可能比较疲倦
- 实验套件很不错，但也许需要多几个助教一起维护和答疑，否则一个助教实在比较累