

十分对不起大家，出题仓促有些疏漏，对没说清楚/错误的地方补充和修正如下：

关于几个文件：

1. network用于模拟丢包/延时/拥塞等情况，在network.py中设置它的地址。
2. 将USocket.py中的network设为这个地址，发送的segment会经由network转发（不需要自己实现），sendto和recvfrom的参数/返回值如python socket，直接调用就可以。
3. 测试时先运行network.py
4. network最终将会统计通过的数据量来计算performance
5. rdt.py里accept函数的第一个返回值类型是 'RDTsocket'

关于要求：

- close：理解为“不能再进行任何收发”，但应等待已发送的数据正确发完。正确关闭指一个connection结束之后（主动调用close()或对方断开），程序仍能正常继续执行后续的部分。
- 合理的效果：假设network的处理速度为R，理想速率为R/连接数量/2，实际达到的不低于30%理想值即认为合理。例如alice.txt大约有150KB，将rate设置为10240 即10KB/s，算上线程等等overhead，一个连接可以在50秒内单向完成发送。
- 文档末尾的几个条件：
 - 保证丢包率小于10%用于判断丢包原因是拥塞还是link不可靠，即不发生拥塞的情况下丢包应该比较罕见。
 - 错误率用于设置合适的包长度。更长的payload更容易出错，且重传代价更大。最终错误率可能会变但只会比0.00001小。
 - 延时不会很混乱，可以比较稳定地估计RTT并设置合适的timeout以及控制发送速率。
- 测试情景其实可能比现实简单很多，不一定要按照书上的介绍来写，原理上能work就行。比如拥塞控制可以直接delay发送或者减小segment size而不需要设置window。
- Echo server 见<https://docs.python.org/3/library/socket.html?highlight=example>底部的example，简便起见不需要listen()。

对题意有其他任何疑问和异议可以随时在群里提问指出，附件是添加了一些解释的network.py