测试用例详解

根据同学们的问题和反馈，对测试用例做出如下调整和补充。希望能够减轻同学们的负担，不用纠结极端情况，关注主线任务。

（一）sender 测试

1. 评分测试用例细则：
   - Normal case without network failures (10%)
   - Loss of arbitrary amount and types of ACK messages (5% → 10%)
     说明：对于 DATA 类型的 ACK 报文丢失，需要通过超时重传，进行恢复；
     对于 START 类型的 ACK 报文丢失，sender 判断超时后，直接向 receiver 发送 END 消息，sender 在收到 END 的 ACK 消息之后，结束进程；
     对于 END 类型的 ACK 报文丢失，sender 判断超时后，连接关闭失败，sender 端主动关闭连接。
   - Reordering of ACKs for DATA messages (5%)
   - Duplication of ACKs for DATA messages (5%)
   - ~~Delay in the arrivals of ACK messages (5%)~~
     说明：ACK 报文延迟到达可能造成的后果，在模拟超时丢包、收到乱序、重复报文的情况中已经涵盖，因此不再单独测试
   - Connection failure (receive incorrect ACK for START message) (5%)
     说明： sender 判断 ACK 消息损坏后，直接向 receiver 发送 END 消息，sender 在收到 END 的 ACK 消息之后，结束进程；

2. broken_receiver 支持的错误类型

   - Error code is 1, broken_receiver will drop one DATA packet randomly during transmission (without ACK).
   - Error code is 2, broken_receiver will exchange the order of two ACK packets.
   - Error code is 3, broken_receiver will select one received packet and send its ACK twice.
   - Error code is 4, the ACK for START message will lost (broken_receiver receives START but the ACK for START message get lost).
   - Error code is 5, the ACK for END message will lost (broken_receiver receives END but the ACK for END message get lost).
   - Error code is 6, broken_receiver will send ACK with wrong checksum for START message.

(二) receiver 测试

1. 评分测试用例细则
   - Normal case without network failures (10%)
   - Loss of arbitrary levels of DATA messges (5%)
   - Reordering of DATA messages (5%)
   - Duplication of any amount for any DATA packet (5%)
   - Packet corruption (bad checksum) (5%)

- Wrong connection (receive incorrect START message) (5%)
  说明：receiver 判断 START message 的 checksum 错误以后，连接建立失败（listen 返回-1），receiver 进程结束即可。sender 仍旧按照没有接收到 START 的 ACK 消息方式处理。

2. broken_sender 支持的错误类型

- Error code is 1, broken_sender will drop one DATA packet randomly during transmission.
- Error code is 2, broken_sender will select one DATA packet and send it twice.
- Error code is 3, broken_sender will send one DATA packet with wrong checksum.
- Error code is 4, broken_sender will send two DATA packets in wrong order.
- Error code is 5, broken_sender will send the START message with wrong checksum.