Short answers to the discussion questions in CNNS lab assign 4

22920202202763

Diss1a: Your transport protocol implementation picks an initial sequence number when establishing a new connection. This might be 1, or it could be a random value. Which is better, and why?

```
answer:
    使用 1 个随机值是比较好的，因为固定使用 1 会使得伪造 TCP 连接的第一道门槛 —— SYN 可以被以 seq = 1 轻松伪造，降低安全性。
```

Diss1b: Our connection setup protocol is vulnerable to the following attack. The attacker sends a large number of connection request (SYN) packets to a particular node, but never sends any data. (This is called a SYN flood.) What happens to your implementation if it were attacked in this way? How might you have designed the initial handshake protocol (or the protocol implementation) differently to be more robust to this attack?

```
answer:
    what happens: server 端的带宽会因为处理大量的 SYN 请求和回送 ACK 包而满载，导致收到正常请求时无法处理 ACK 回复，进而无法进行正常数据传输。
    the initial handshake protocol: 在协议初始化的时候，建立一个 HashMap 统计一定时间内同一个 (srcAddr, destAddr) 收到的各种包的数量，如果在这段时间内收到某一个节点的包出现异常，则将这一TCPSock 关闭。
```

***Diss1c*: What happens in your implementation when a sender transfers data but never closes a connection? (This is called a FIN attack.) How might we design the protocol differently to better handle this case?**

```
answer:
    what happens: server 端不会关闭这一个 sender 的连接，这一个 TCPSock 会被持续占用。
    design: 在每一个节点连接时记录下最后一个数据包的接收时间和类型，对异常连接进行传开并传送 FIN 数据包。
```