

计算机网络课程作业

何昕、冯元浩、宓楠浣、陈子阳

作业 1. 实现简单可靠的传输层协议 backTCP

- 概述

backTCP 的目标是实现面向无连接的可靠传输功能，能够解决数据包在传输过程中出现的乱序以及丢包问题。由于考虑的是无连接的网络，因此该作业不需要考虑传统 TCP 中的三次握手连接建立过程。此外，假设传输中数据不会出现错误，因此只需要考虑如何解决数据包的乱序和丢包问题。

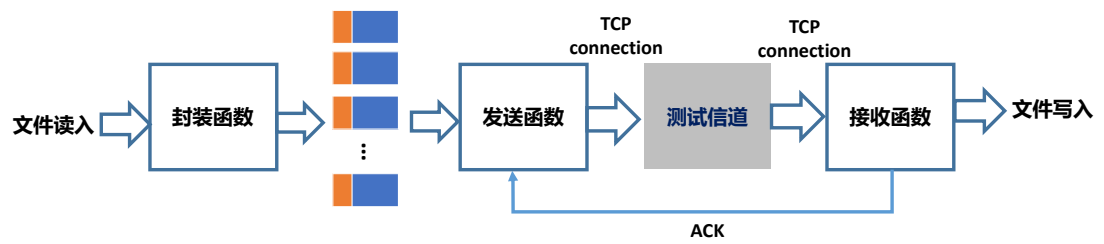
- backTCP 数据包结构

```
typedef uint8_t tcp_seq;

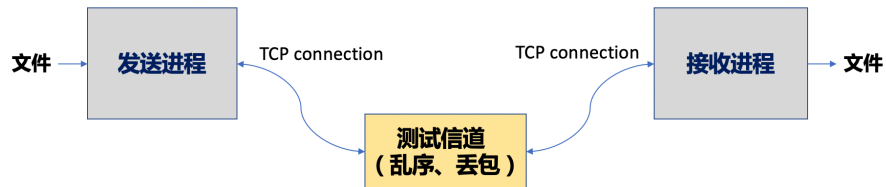
struct tcphdr {
    uint8_t    btcp_sport;           /* source port */
    uint8_t    btcp_dport;           /* destination port */
    tcp_seq    btcp_seq;             /* sequence number */
    tcp_seq    btcp_ack;             /* acknowledgment number */
    uint8_t    data_off;             /* data offset */
    uint8_t    win_size;             /* window size */
    uint8_t    flag;                 /* retransmission or not*/
} BTcpHeader;
```



- 数据包中 payload 最长不能超过 64 字节。发送缓存为 512 字节，使用滑动窗口进行发送，发送时启动 timer。当超过 10ms，在未收到 ACK 时，重传未收到确认号后面的所有帧（Go-Back-N），若同时实现选择重传协议（SR），加分。
- 说明：假设一个待传送的文件，按照“字节-空格-字节”的方式存储，其中每个字节由一个十六进制数表示，例如，0A 49 4A 49 4A 49 4A 49 4A 49 4A 41 4A 41 4C, 首先，将这个文件读入，然后通过编写的数据包封装函数，将文件按照 backTCP 数据报结构进行封装；封装后，使用滑动窗口协议进行按序发送，每个包调用一次套接字 TCP 发送函数，发送到我们提供的测试信道中；测试信道处理后，会使用 TCP 发送到接收端程序中，接收端程序确定收到的数据报序号，并确定哪些数据包在传输过程中被丢弃；对第一个丢弃的包序号前面正确收到的分组进行确认（确认帧通过 TCP 直接发送到发送端），之后，发送端重传序号后面所有的帧，例如，发送了 1, 2, 3, 4, 5 这几个数据包，其中 4 丢包了，对 3 进行确认，发送端重传 4 和 5 两个数据包。为了简化处理，我们在数据包头部加了一个 flag 字段，指示是否为重传的数据包，1 表示重传。重传的数据包经过测试信道不会丢包。



- 推荐使用 Linux 环境下 POSIX Socket，Windows 环境下的 Winsock 亦可，不做限定。
- 考查方法：发送端发送一测试文件，然后接收端检测是否能恢复该文件。我们会在测试中，使用一个虚拟测试信道，将发送出来的数据包随机调整顺序和丢弃其中一些数据包，最后检测接收端输出的文件。



- **Due date: 11.10 23:59, 提交.c 的压缩文档(.zip 格式)。**