

实验3：基于UDP服务设计可靠传输协议并编程实现之3-3（拥塞控制）

姓名：丁彦添 学号：1911406

实验内容

利用UDP协议在用户空间实现面向连接的可靠数据传输。

功能包括：建立连接、差错检测、确认重传等。流量控制采用拥塞控制机制，完成给定测试文件的传输。

此次实验在实验3-2的基础上，将停等机制改成基于可变长滑动窗口的流量控制机制，采用固定窗口大小，支持累积确认，完成给定测试文件的传输。

协议设计

大致流程

建立和断开连接和实验3-2设计差别不大。

连接建立

- 发送端向接收端发送长度为2的标志，包含SEQ和ACK码（此时SEQ码为预设值，ACK码为随机值）
- 接收端收到第一次握手之后，检查SEQ码是否和预设值相同，若相同，发给发送端长度为2的标志（ACK码为上一次SEQ码+1）
- 发送端收到第三次握手之后，检查ACK码是否正确，若正确，建立连接成功并发送第三次握手信息（但是第三次握手信息不会被接收端接收了）

建立连接后，发送文件名到接收端

拥塞控制机制（本次实验的重点）

- 设计两个状态，分别为慢启动状态和拥塞状态，发送窗口初始化为1。
- 首先判断状态，当状态为慢启动状态，且发送窗口没有超过窗口上限时，发送窗口乘以2；当窗口为拥塞控制状态时，发送窗口加1（特别的，当发送窗口大小乘以2之后超过了窗口上限时，判断其进入拥塞状态）
- 然后发送数据包，用index值记录发送到了第几个包。每一次发包个数为发送窗口大小（如果个数不够那么有多少发多少），之后接收ACK，记录时间。
- 然后接收ACK，记录接收到的ACK的数量。在 `recvfrom` 中记录结束时间，如果超时，那么就不再接收ACK。同时，每次接收到没有超时的ACK，接收的包数+1
- 接收方接收到包，先进行差错检测，若正确，进行解包，判断其位置，载入到程序内
- 因为接收方有很大概率收到乱序的包，所以也要乱序载入进程序。
- 发送方，如果接收的包数等于发送的包数，那么维持状态（慢启动或拥塞）不变；若不等于，将发送窗口减少到1，状态变成慢启动，同时视本次发送无效，重新发送本趟的所有包（即不改变index）。

例如，假如说发送窗口为8，本次发包发送了41，42，43，...，48，但是接收到的ACK个数小于8，那么视本次发送无效，进入慢启动状态，设发送窗口为1，发送31，若成功，设发送窗口为2，发送32，33.....

- 可以知道，如果在网络传输过程中发生报文错误或者丢包的话，也会影响ACK的个数，这种实现方式可以避免丢失。
- 如果发送方发了一个包，但就是接不到接收方发回的ACK，那么程序会卡住。

断开连接：两次挥手

- 客户端向服务器发送长度为2的标志，包含WAVE和ACK码，关闭客户端
- 服务器接收第一次挥手之后，检查WAVE码是否和预设值相同，若相同，发送WAVE和ACK码,关闭服务器。

程序运行截图

