

实验3-3：设计并实现拥塞机制

1811439 吴继强

一、实验要求

在任务3-2的基础上，选择实现一种拥塞控制算法，也可以是改进的算法，完成给定测试文件的传输

二、实验环境

- Visual Studio 2017

三、实验设计

- 建立连接（在实验3-1中有所叙述）
- 发送文件名，加载传输文件的二进制形式于程序内
- 设计拥塞机制
 - 采用单线程编程，重构 `sendmessage` 函数

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）

然后发包（实验3-1有所叙述），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（几乎每次实验都会）收到的是乱序的包，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**，重新发送这次包（即不改变index）。

例如，假如说发送窗口为8，本次发包发送了31, 32, 33, ..., 38，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。

还需要实现的：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。

- 接收方载入数据到文件
- 断开连接