接下来我来讲解一下我的代码，首先是zyl\_reno结构体的定义，采用reno的拥塞控制算法，此结构体里的变量包括拥塞窗口的大小，阈值大小，重复ack的个数，以及到拥塞避免每窗口大小增加1的记录数值和三个状态的定义，结构体里的函数首先是接收到新的数据报的函数的处理，如果是慢启动状态，则窗口大小加一，然后根据窗口是否大于阈值判断是否进入拥塞避免阶段，如果是拥塞避免阶段，则每隔窗口大小增加一个大小，如果是快速恢复阶段则让拥塞窗口为阈值并进入拥塞避免阶段，接下来是超时的处理，不论现在在哪一个状态，都必须进入慢启动阶段并且让阈值为当前窗口的一半并让窗口为1，然后是接收到重复数据报的处理，当重复的数据报大于3的时候，如果现在是慢启动或者拥塞避免阶段则让阈值为当前窗口的一半并让窗口为阈值加3，进入快速恢复阶段，如果当前是快速恢复阶段，则窗口大小增加1。

然后是接收线程的定义，首先每次判断是否完成所有的发送确认，如果完成则结束处理并返回，如果还没有完成所有的发送确认，则判断是否超时，如果超时则利用go-back-n重传数据报，如果并不超时且校验和正确摒弃是ACK报文，则判断是新数据报还是重复数据报，如果是新数据报则直接让结构体处理即可，如果是重复数据报则发送丢失数据报，然后更改缓冲区，判断是否还有未确认的报文来看是否重新计时。

接下来是发送单个数据报，如果小于窗口大小则发送并加入缓冲区，然后发送文件中先开启接收线程在发送数据报，剩余部分与之前类似，我的讲解完毕，谢谢学姐。