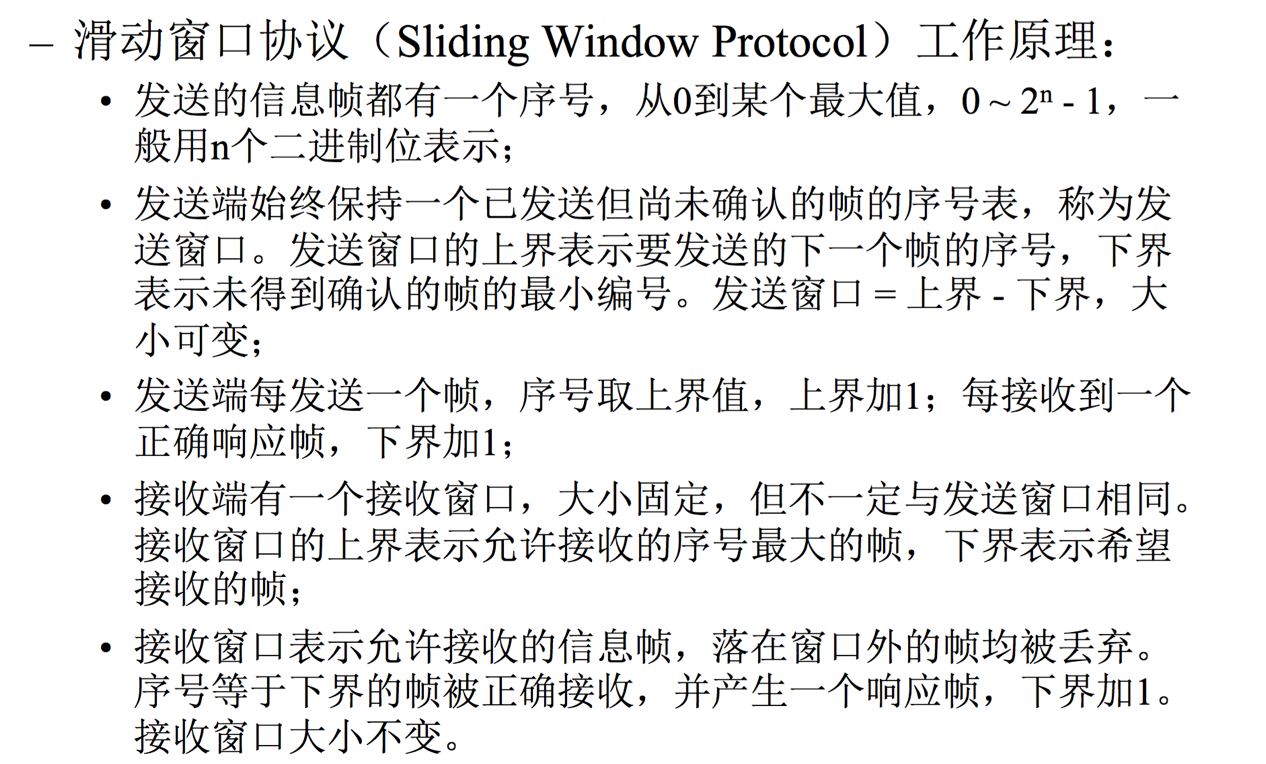
**滑动窗口协议实验报告**

2013011371 计34 沈哲言

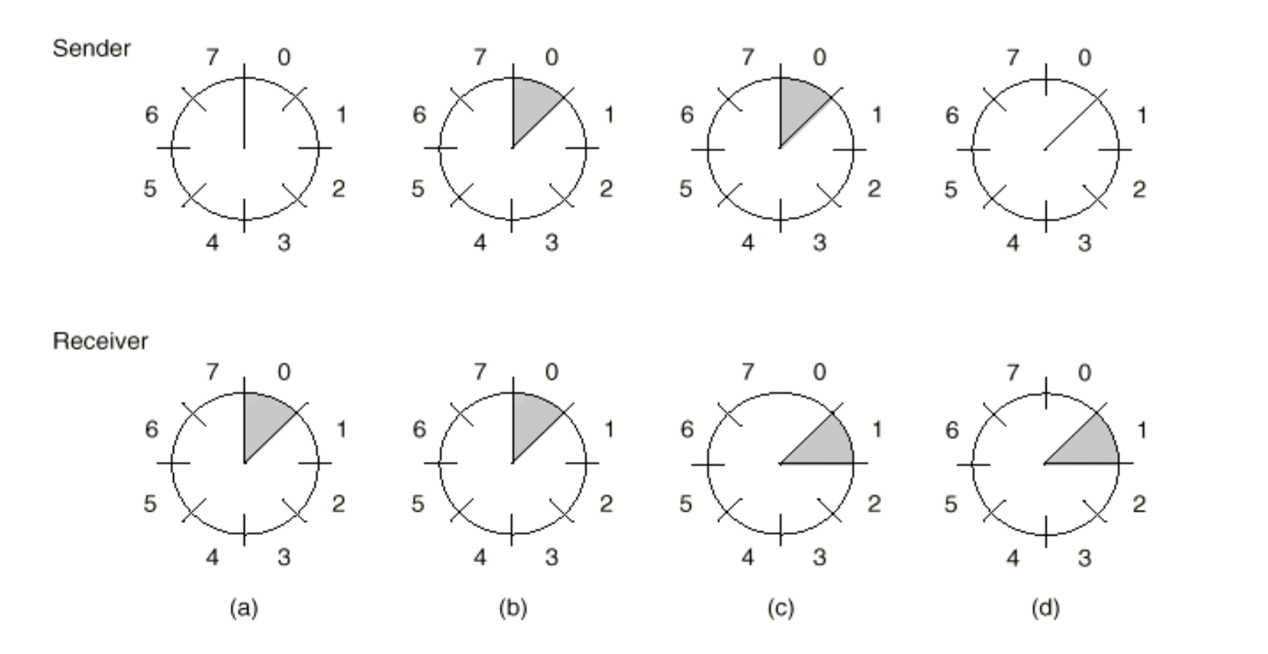
1. 实验内容
2. 熟悉NetRiver实验系统，学会在实验系统上进行编辑、编译、运行和调试
3. 掌握滑动窗口协议，在理解协议的基础上利用C/C++语言实现1bit滑动窗口协议和退后n帧协议的发送方部分，使得能够响应系统的发送请求、接收帧消息和超时消息
4. 实验原理



这是基础的滑动窗口协议的原理，本实验要求实现的1bit重传和退后n帧协议都是在基本的协议上的扩展，下面分别叙述

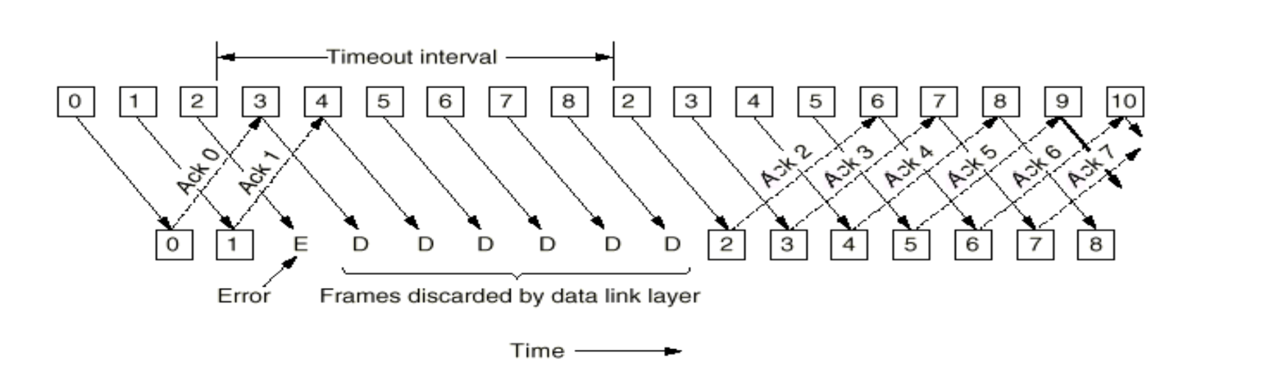
（1）1bit滑动窗口协议

这个协议其实是单工停等协议的双工版本，允许双向发送，但发送方还是只能一次发一帧，等待确认之后再发下一帧，发送方窗口大小为1，即没有接到当前发送帧的响应时不能发下一帧，接收方窗口大小也为1，即只能接收期待的帧，其余一律丢弃，下图为原理图



（2）退后n帧协议

接收方的窗口大小依然是1，区别在于发送方窗口大小调整为大于1的常数n（本实验中为4），也即发送端可以在未收到响应帧数目小于n的时候连续发送，并未这些帧单独设置计时器和缓冲区，一旦有一个帧发送失败，从这个帧起的之后所有帧都要重发，原理图如下



1. 实现思路

首先设置全局缓冲区和发送端上下界（即待发送帧编号和最早的未确认帧编号），然后分别响应不同的消息类型

1. 发送请求

先将帧存入缓冲区，让后判断当前帧是否在发送窗口内，如果是就发送否则返回

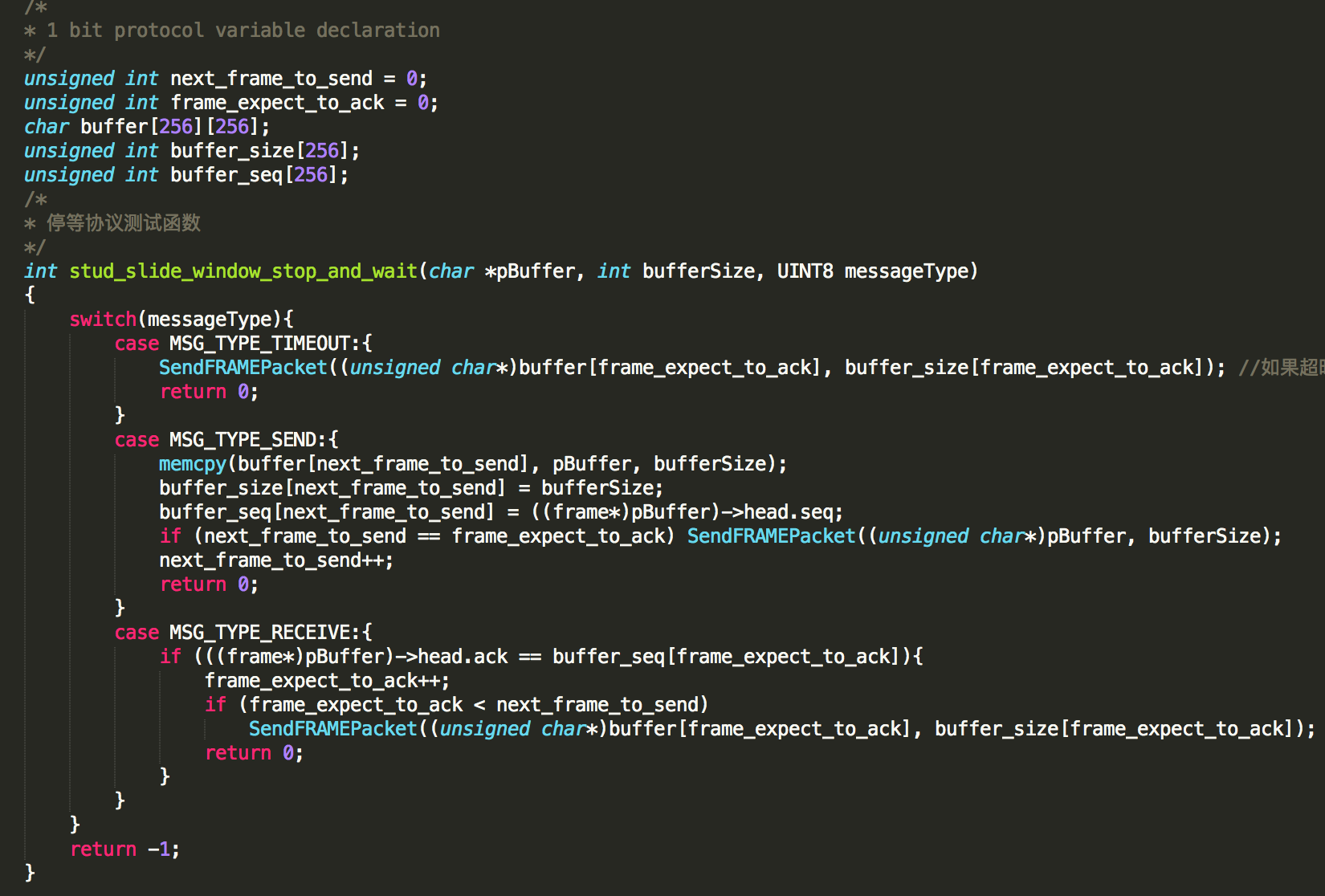
1. 接收帧消息

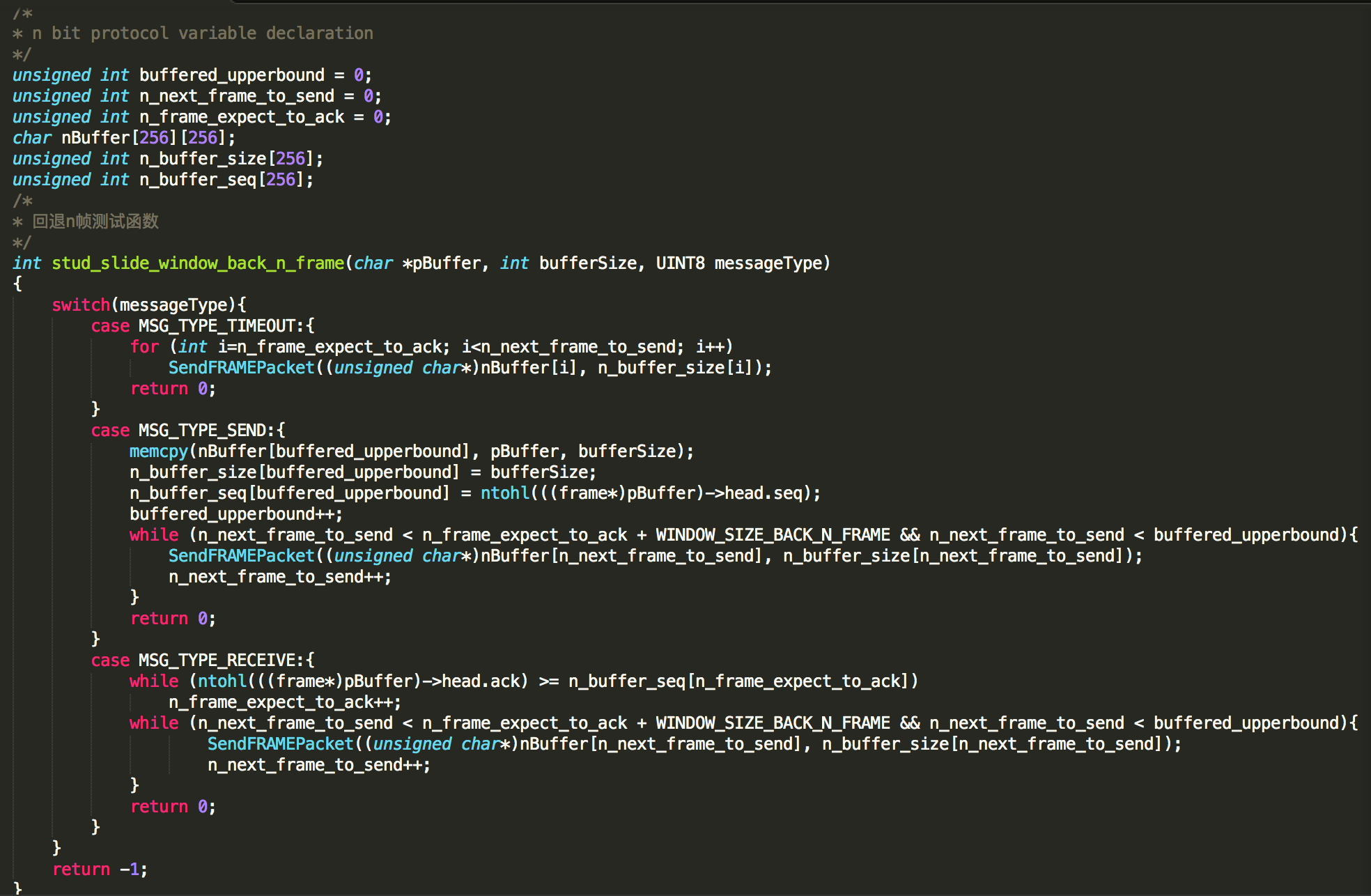
判断确认帧号是否和待确认帧好一致，如果一致更新待确认帧号，并且将缓存的帧在窗口大小允许的条件下发出

1. 接收超时消息

从待确认帧起重发帧（1bit重发1帧，退后n帧重发超时帧后所有帧）

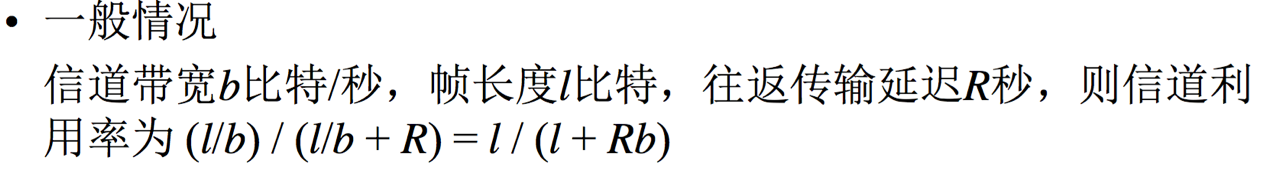
1. 源代码





1. 思考题

（1）1bit重传协议本质上基于停等的方式，双方同时开始的话有一半的重复帧，传输时间长，退后n帧协议大大提高的信道的利用率，一次可以发送n帧，可以充分利用高带宽的优势同时部分弥补延迟大的劣势，这个结论可以从信道利用率公式上看出



这是1bit的情况，nbit的话上下的l同时乘以n可以提高利用率

（2）退后n帧重传要设置比较大的发送端缓冲区，以便在重传的时候能够顺利进行，这要消耗一定的内存空间；其次，退后n帧重传可能会重发已经正确发送的帧，假设每一帧的出错概率为p，n帧中需要重发某些帧的概率为1-（1-p）^n，在n很大时这个概率是很高的，故如果信道误码率高的情况下，很大的n将导致不断重发，这显然是对带宽的浪费