**计算机网络编程**

**实验报告**

**班级：30081702**

**组长：**

**范文骁 1120170346**

**成员：**

**张翰澄 1120163682**

**邢智博 1120170939**

**李想 1120171688**

**韩世杰 1120172345**

**黄一帆 1120172825**

**袁祥博 1120173374**

**北京理工大学**

**计算机学院**

**2020年6月**

**第3章 实验3 基于停止等待协议的可靠通信**

**1. 实验目的**

掌握基于停止等待协议的可靠通信的相关原理。

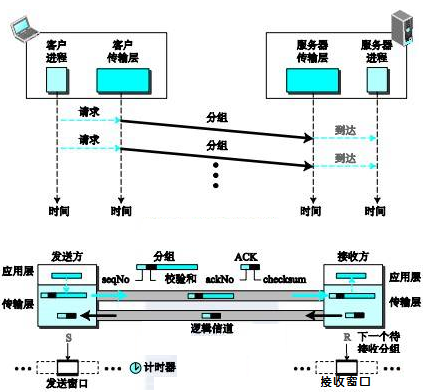
**2. 实验内容**

采用 UDP Socket 编程接口作为模拟物理层接口实现帧的发送和接收，协议采用单工方式进行数据通信。假设 Host1 要向 Host2 发送大文件，通过数据链路层的帧每次完成数据块的可靠传输，采用停止等待协议，差错编码采用 CRC-CCITT 标准。以教材协议 3 为基础，在帧末尾增加 CRC 校验字段

**3. 实验原理**

当发送窗口和接收窗口都等于1时，就是停止等待协议。发送端给接收端发送数据，等待接收端确认回复ACk，并停止发送新的数据包，开启计时器。数据包在计时器超时之前得到确认，那么计时器就会关闭，并发送下一个数据包。如果计时器超时，发送端就认为数据包丢失或被破坏，需要重新发送之前的数据包，说明数据包在得到确认之前，发送端需要存储数据包的副本。

停止等待协议是发出一个帧后得到确认才发下一个，降低了信道的利用率。



**4. 实验环境**

Python pycharm

Java

c

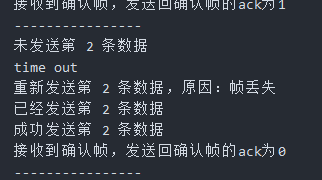
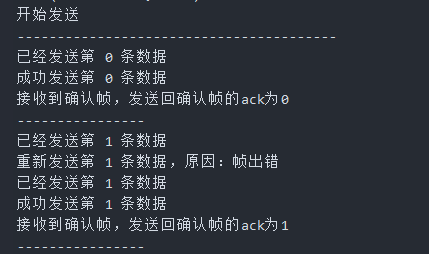
**5. 实验步骤**

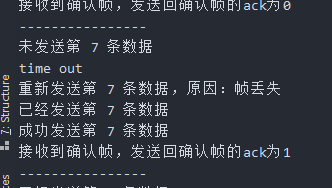
Python：

首先读入相关的配置参数，并初始化socket。Client在发送包之后立刻进入接收阶段，等到确认包的返回，而server在接受到包的时候也立刻返回一个确认包，ack和seq的相关关系按照书上的关系进行增加，checksum调用实验3-1实现的结果，出错的时候就不返回确认包，让其自动超时，或者返回一个报错的包提示client进行重传。超时的处理比较麻烦，client需要开一个单独的线程计时，如果超时就给自己发一个包来取抵消自己的recvfrom，至于所有的编号均按照ppt与课本上进行维护即可。

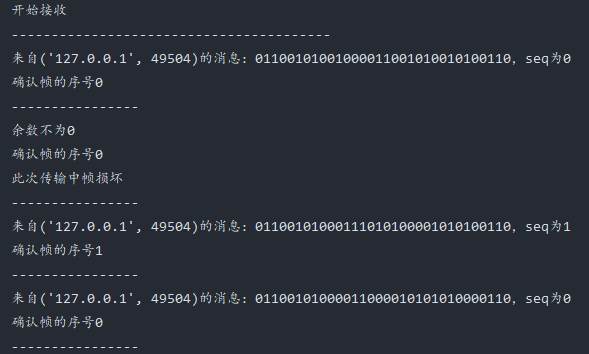
实验结果如下：

Client：





Server：



其中第二次接收对应着client的第二张图的帧损坏，超时重传只在client端体现。

**6. 实验总结**

通过这次实验，我们基本掌握了基于停止等待协议的可靠通信的核心思想。