**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4 基于连续 ARQ 协议的可靠通信**

**1. 实验目的**

掌握基于连续 ARQ 协议的可靠通信的相关原理。

**2. 实验内容**

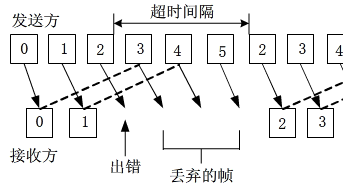
采用 UDP Socket 编程接口作为模拟物理层接口实现帧的发送和接收，协议采用双工方式进行数据通信。假设 Host1 和 Host2 分别向对方发送大文件，Host1 先发送一帧到 Host2，通过数据链路层的帧每次完成数据块的可靠传输，采用 GBN 协议，差错编码采用 CRC-CCITT标准。以教材协议 5 为基础，在帧末尾增加 CRC 校验字段。

**3. 实验原理**

实验采用了退后N帧协议。在发送完一个帧后，不用停下来等待确认，而是可以连续发送多个数据帧。收到确认帧时，任可发送数据，这样就减少了等待时间，整个通信的通吞吐量提高。

如果前一个帧在超时时间内未得到确认，就认为丢失或被破坏，需要重发出错帧及其后面的所有数据帧。这样有可能有把正确的数据帧重传一遍，降低了传送效率。

线路很差时，使用退后N帧的协议会浪费大量的带宽重传帧。



**4. 实验环境**

Python pycharm

Java

c

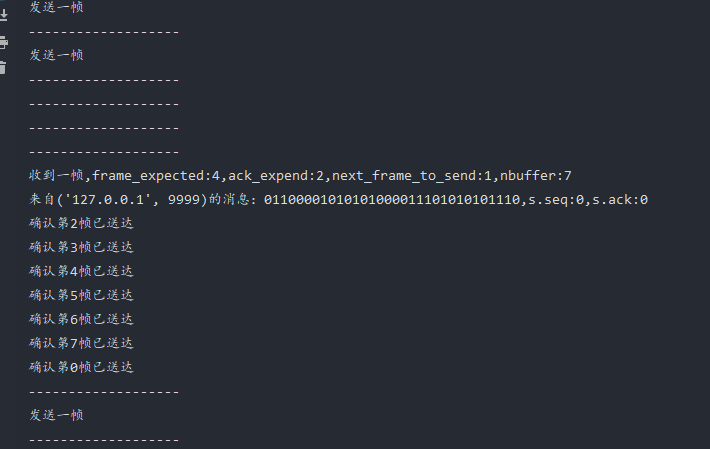
**5. 实验步骤**

Python：

首先读入相关的配置参数，并初始化socket。由于连续ARQ协议是双工协议，所以不区分client和server。按照教材协议5的基础框架，构建一个大的while循环，维护一个状态变量flag\_control，利用此来判断接收到下一个包的时候需要执行什么样的操作。值得指出的是用于维护超时判断的表，在发送一个包还未接收到确认的时候立刻将其对应的计时器（时间）计入表中，然后再每次事件（flag\_control）的判断（wait\_for\_event）中，只需要查看当前表的第一项是否超时即可，如果超时了就执行相应的重传操作，稍待确认的话就将该表对应数量的计时器弹出即可。

实验结果如下：

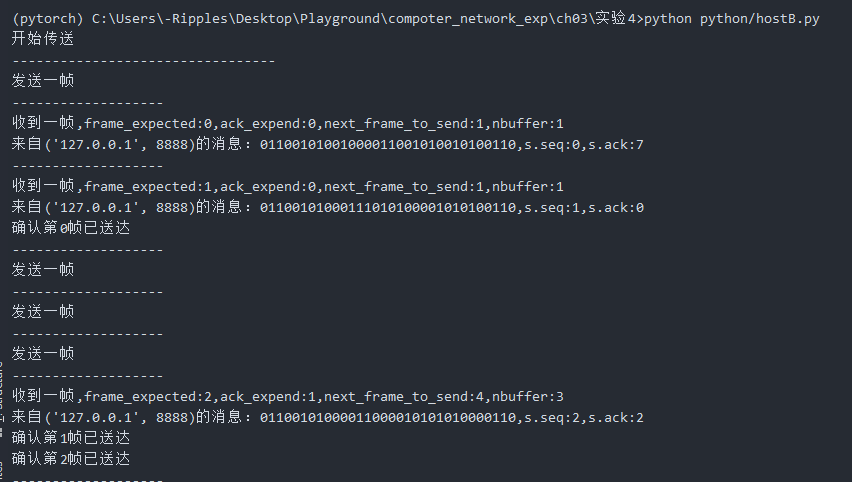
Host A：

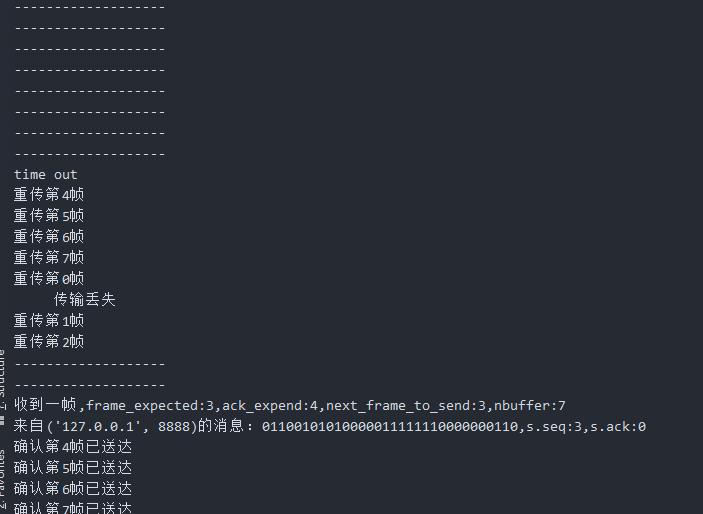


结果说明：

如果出现一行分隔符但没有对应的说明，即为在等待对应的确认，窗口已经扩大到了最大值。

host B：





结果说明：

上图展示了超时重传的机制，上面的分隔符表示每次wait\_for\_event都在等待确认包（自己的窗口已经扩到了最大），最终出发了timeout进行重传。

**6. 实验总结**

通过这次实验，我们基本掌握了基于连续 ARQ 协议的可靠通信的核心思想。