**计算机网络编程实验要求说明**

一、实验报告格式

实验报告格式举例如下：

第3章 实验1 XXXXXXXX（XXX代表实验名称）

1. 实验目的

XXXXXXXXXXXXX

2. 实验内容

XXXXXXXXXXXXXX

3. 实验原理

XXXXXXXXXXXXXXX

4. 实验环境

XXXXXXXXXXXXXXXXX

5. 实验步骤

XXXXXXXXXXXXXx

6. 实验总结

XXXXXXXXXXXXXXXX

二、实验所用编程语言、环境和要求

实验编程语言采用C/C++、Java、Python3三种，每个编程实验均需这三种编程语言的实现版本。开发环境选择Windows平台，C/C++、Java、Python3的集成开发环境分别采用Visual Studio 2015、Eclipse 4.8、PyCharm教育版。程序最基本要求是界面实现控制台版本，运行中需要用到的参数以配置文件的形式保存。

三、实验分组

编程实验采用分组完成的模式，每个小组6-7人组成，每个班分成4-5组。设定组长一名，每班学委将分组情况上报授课教师。

四、实验验收方式

每个实验最终提交的材料包括：实验报告、集成环境下的源码工程文件、可运行的执行程序、软件安装部署运行和使用说明。

验收采用小组汇报验收的方式，在提交上述材料的同时，小组汇报自己的成果，包括但不限于：设计思想、运行演示等。

第3章 数据链路层

实验1编程：循环冗余校验CRC生成和校验程序

配置文件关键要点：

待发送的数据信息二进制比特串（32位）

InfoString1=0110XXXXXXXXXXXXXXXXXXXX110

收发双方预定的生成多项式采用CRC-CCITT=X16+X12+X5+1，对应的二进制比特串（17位）

GenXString=10001000000100001

接收的数据信息二进制比特串（32位）

InfoString2=0110XXXXXXXXXXXXXXXXXXXX110

程序运行屏幕输出要点：

首先显示待发送的数据信息二进制比特串

然后显示收发双方预定的生成多项式采用CRC-CCITT，对应的二进制比特串

计算循环冗余校验码CRC-Code

显示生成的CRC-Code，以及带校验和的发送帧

显示接收的数据信息二进制比特串，以及计算生成的CRC-Code

计算余数

显示余数，为零表示无错，不为零表示出错

实验2 编程：实现透明传输程序

分别实现零比特填充和字节填充。

比特填充配置文件关键要点：

待发送的数据信息二进制比特串（32位）

InfoString1=0110XXXXXX11111111111XXXXXXXX110

帧起始和结束标志二进制比特串

FlagString=01111110

比特填充程序运行屏幕输出关键要点：

屏幕显示帧起始标志、帧数据信息和帧结束表示

比特填充，显示比特填充后的发送帧

显示比特删除后的接收帧

字节填充配置文件关键要点：

待发送的数据信息十六进制串（64位）

InfoString1=347D7E807E40AA7D

帧起始和结束标志十六进制串

FlagString=7E

字节填充程序运行屏幕输出关键要点：

屏幕显示帧起始标志、帧数据信息和帧结束表示

字节填充，显示字节填充后的发送帧

显示字节删除后的接收帧

实验3 编程：基于停止等待协议的可靠通信

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。

发送程序配置文件关键要点：

数据传输目的UDP端口

UDPPort=8888

增添发送过滤程序，模拟传输出错或丢数据帧，下面两项指明每发送多少帧出现一次出错或丢帧，此例表示每10帧中一帧出错，每10帧中一帧丢失

FilterError=10

FilterLost=10

发送程序运行屏幕输出关键要点：

显示next\_frame\_to\_send变量的值，以及正在发送帧的编号

显示经过过滤器后是正确发送、模拟传输出错还是模拟帧丢失（实际没有发送）

显示接收到确认帧，确认帧的确认序号

或者显示超时

回到开始重复一直到文件发送完成

接收程序运行屏幕输出关键要点：

显示frame\_expected变量的值，

接收帧是否出错（CRC余数是否为零），正确则显示接收帧的发送帧序号

显示发送回确认帧，以及确认帧的确认序号

回到开始重复一直到文件接收完成

实验4 编程：基于连续ARQ协议的可靠通信

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。

配置文件关键要点：

数据传输目的UDP端口

UDPPort=8888

增添发送过滤程序，模拟传输出错或丢数据帧，下面两项指明每发送多少帧出现一次出错或丢帧，此例表示每10帧中一帧出错，每10帧中一帧丢失

FilterError=10

FilterLost=10

Host1程序运行屏幕输出关键要点：

显示ack\_expected, next\_frame\_to\_send和frame\_expected变量的值，以及正在发送帧的编号和确认序号

显示经过过滤器后是正确发送、模拟传输出错还是模拟帧丢失（实际没有发送）

显示接收到对方帧，该帧的发送序号和确认序号，以及当前frame\_expected变量的值

或者显示超时，重传帧的发送序号，以及ack\_expected, next\_frame\_to\_send变量的值

回到开始重复一直到文件发送完成

Host2程序运行屏幕输出关键要点：

显示frame\_expected变量的值

接收帧是否出错（CRC余数是否为零），正确则显示接收帧的发送帧序号

显示ack\_expected, next\_frame\_to\_send和frame\_expected变量的值，以及正在发送帧的编号和确认序号

显示经过过滤器后是正确发送、模拟传输出错还是模拟帧丢失（实际没有发送）

或者显示超时，重传帧的发送序号，以及ack\_expected, next\_frame\_to\_send变量的值

回到开始重复一直到文件接收完成