

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 可靠数据传输协议设计与实现 | | | | | |
| 姓名 | 马旭 | | 院系 | 计算科学与技术学院 | | |
| 班级 | 1603106 | | 学号 | 1160300601 | | |
| 任课教师 | 聂兰顺 | | 指导教师 | 聂兰顺 | | |
| 实验地点 | 格物楼207 | | 实验时间 | 2018.11.7 | | |
| 实验课表现 | 出勤、表现得分(10) |  | 实验报告  得分(40) |  | 实验总分 |  |
| 操作结果得分(50) |  |
| 教师评语 | | | | | | |
|  | | | | | | |

****

目录

[1. 实验目的 3](#_Toc529985172)

[2.实验内容 3](#_Toc529985173)

[3.实验过程 3](#_Toc529985174)

[3.1.GBN协议数据格式 3](#_Toc529985175)

[3.2.确认分组格式 3](#_Toc529985176)

[3.3基本流程图 3](#_Toc529985177)

[3.4协议典型交互过程 5](#_Toc529985178)

[3.5数据分组丢失验证模拟方法 6](#_Toc529985179)

[3.6程序实现的主要类及主要函数 7](#_Toc529985180)

[4.实验结果 12](#_Toc529985181)

[4.1.基本功能实现： 12](#_Toc529985182)

[4.2.实现GBN协议： 12](#_Toc529985183)

[4.3.实现SR协议： 14](#_Toc529985184)

[5.问题讨论 15](#_Toc529985185)

[6.心得体会 16](#_Toc529985186)

# 实验目的

理解滑动窗口协议的基本原理；掌握 GBN 的工作原理；掌握基于

UDP 设计并实现一个 GBN 协议的过程与技术。

# 2.实验内容

1) 基于UDP设计一个简单的GBN协议，实现单向可靠数据传输（服

务器到客户的数据传输）。

2) 模拟引入数据包的丢失，验证所设计协议的有效性。

3) 改进所设计的 GBN 协议，支持双向数据传输；（选作内容，加分

项目，可以当堂完成或课下完成）

4）将所设计的 GBN 协议改进为 SR 协议。（选作内容，加分项目，

可以当堂完成或课下完成）

# 3.实验过程

## 3.1.GBN协议数据格式

对一个字节数组进行封装，封装成一个类，如下：

// 第一位字节为seq,其余位为数据data

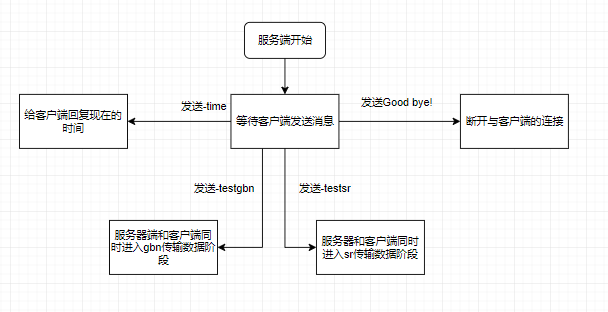
private byte[] allData = new byte[1471];

## 3.2.确认分组格式

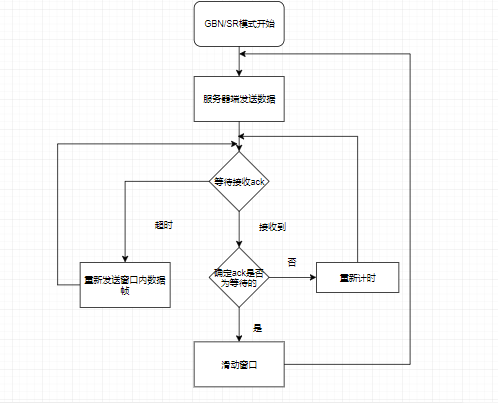
直接使用一个字节当作ack分组格式。

## 3.3基本流程图

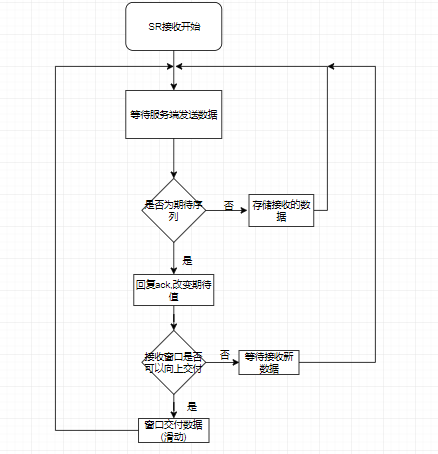
**1. 服务器端大致流程**



1. **GBN/SR传输具体流程**

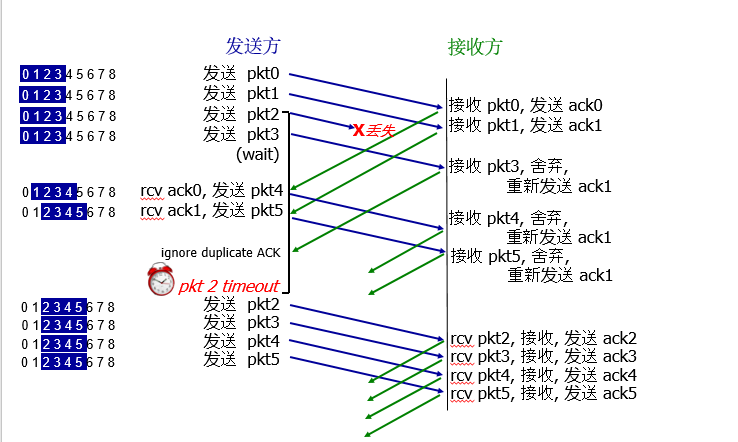


1. **SR具体接收流程**

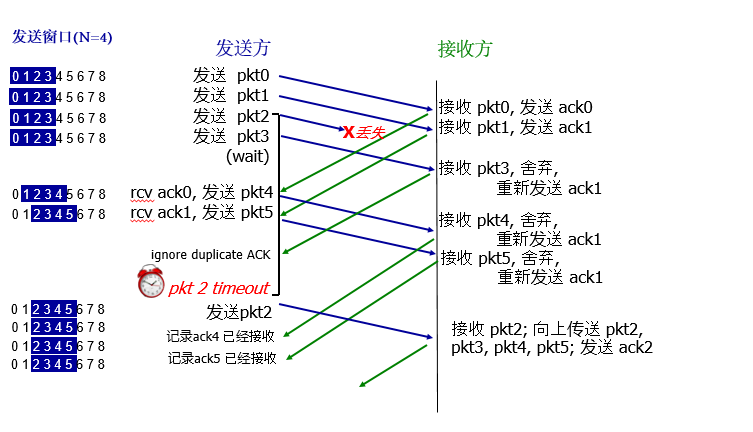


## 3.4协议典型交互过程

**1. GBN协议**



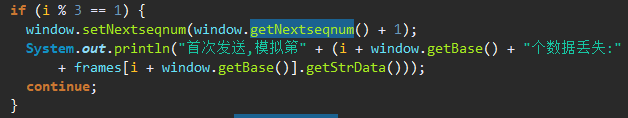
1. **SR协议**



## 3.5数据分组丢失验证模拟方法

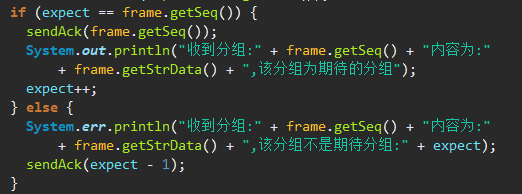
**1. 丢失模拟：**

通过使用整除，当发送数据的序号可以被某个数整除就实际不发送，但是标记上标记发送。如下：



**2．验证丢失：**

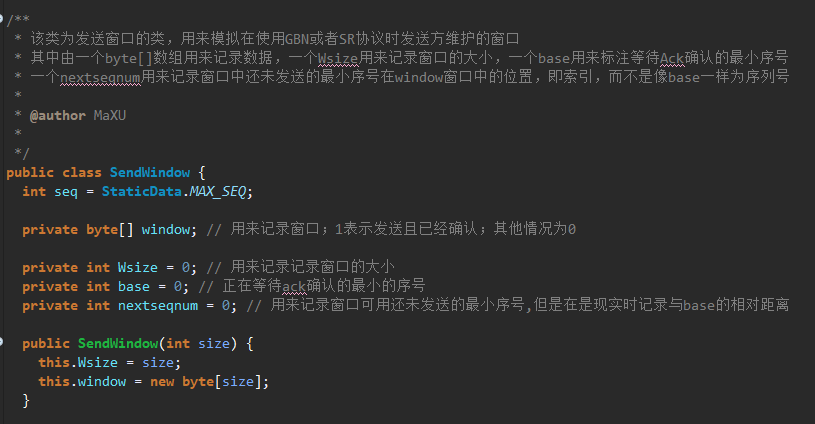
在接收端使用红色字体打印丢弃的分组，使用白字打印接收的分组。如下：



## 3.6程序实现的主要类及主要函数

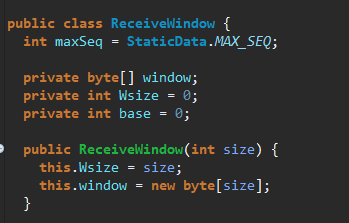
**1. 发送窗口类：**

模拟发送窗口，具有滑动方法等滑动窗口的动作。如下：



1. **接收窗口类**

一个模拟SR协议的接收窗口而创建的类，具有接收窗口的动作，如下：



1. **服务器类EchoServer**

一个服务器类主要用来当作发送文件的服务器端。其主要具备下面几个函数：

(1).根据接收数据决定进行特定的处理，receive函数



(2).进入gbn模式，用于处理对应的gbn请求，ingbn函数。



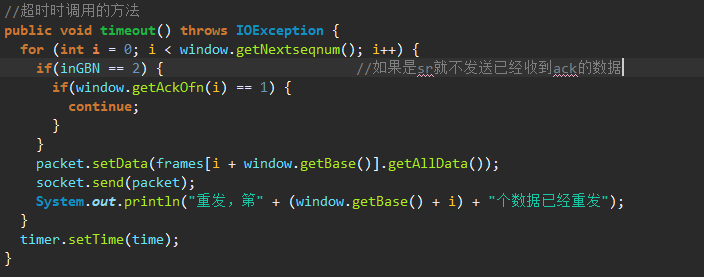
(3).进入sr模式，用于处理对应的sr请求，insr函数。



(4). 发送数据函数，用于将数据发送给客户端，sendData函数。



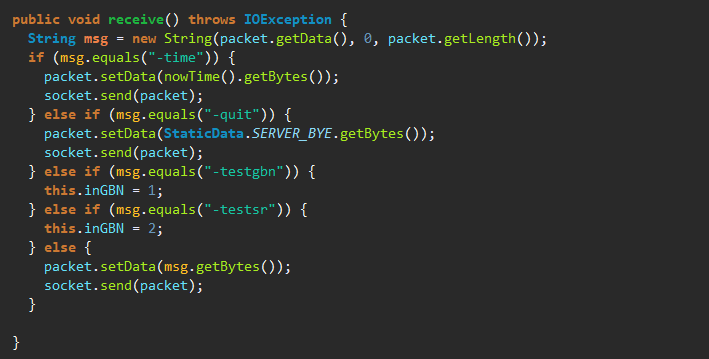
(5).超时处理函数，用于超时时对数据进行发送。timeout函数:



1. **客户端类EchoClient**

一个客户端类用于模拟客户端用于接收数据。其主要方法如下：

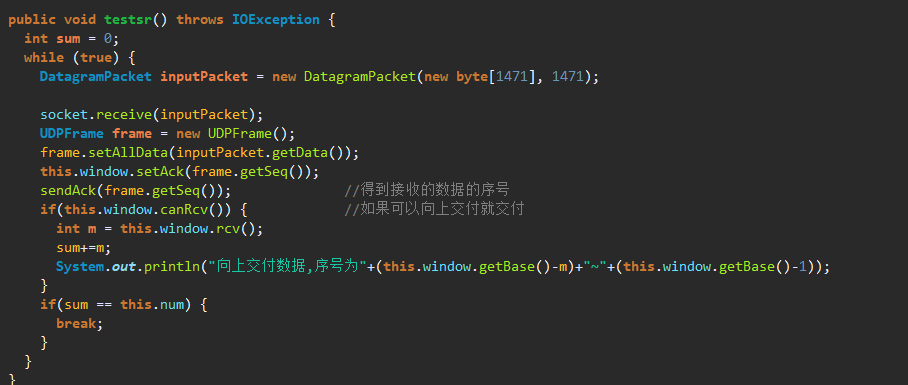
(1). 根据接收数据决定进行特定的处理，receive函数



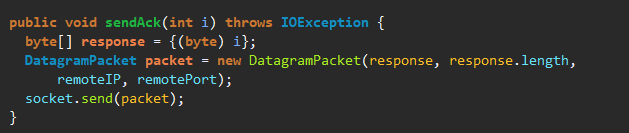
(2).进入gbn模式，用于处理对应的gbn请求，testgbn函数。



(3).进入sr模式，用于处理对应的sr请求，insr函数。



(4).回复ack函数，向发送方回复ack



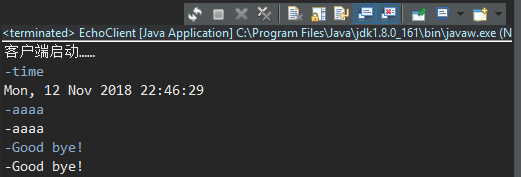
1. **计时器类Timer**

用于发送时的计时。



# 4.实验结果

### 4.1.基本功能实现：

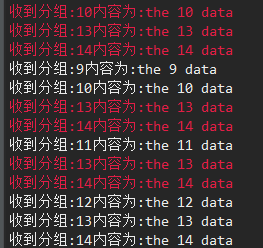


### 4.2.实现GBN协议：

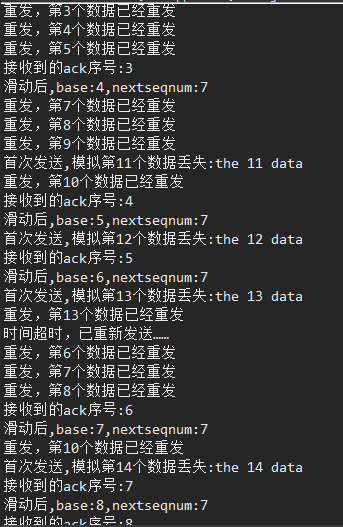
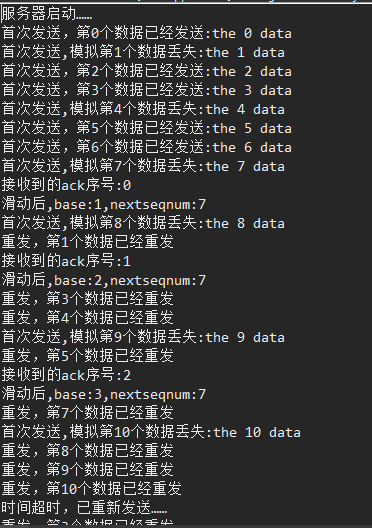
窗口大小：10

帧个数：15

运行结果如下：



其中红字表示被舍弃的数据，白字表示被正确接收的数据。



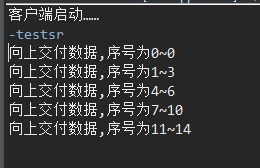
这是服务器端打印的发送数据。

### 4.3.实现SR协议：

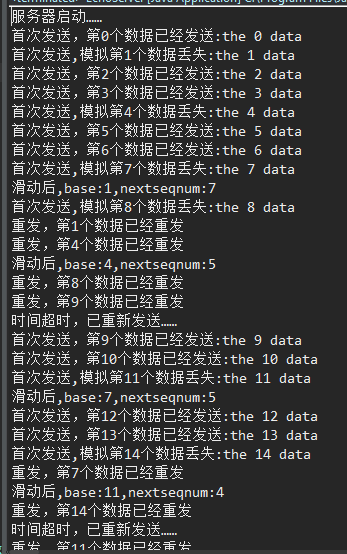
窗口大小：10

帧个数：15

客户端接收数据情况如下：



服务器端打印如下：



# 5.问题讨论

1. 在SR协议中，发送窗口大小和接收窗口大小的和要小于最大的序列号，否则在接收端会出现不知道具体接收的序号是不是需要接收，就会出现序号错乱的现象。

2. 在实验中，窗口中设置ack时，因为会出现延时到达的 ack序号不在窗口中的情况，这样就会导致出现数组越界的错误。这样的话在设置ack时就需要综合考虑各种情况，尤其是base + size > max\_seq ；在进行判断时需要分段判断。

3. 计时器的使用，如果每个帧都使用一个计时器就会非常浪费线程，并且难以维护，所以就在整个文件传输中只使用了一个计时器。

# 6.心得体会

1. 通过实验对GBN协议和SR协议有了更加深入的了解。

2. 通过Java中的UDP编程有了一些了解。

3. 在写实验时首先要把需要的类写好，然后再将整体的框架写好，否则在实现主要功能时会因为个别需要的类没有写好而束手束脚。