哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

<table>
<tr><td>实验名称</td><td colspan="5">可靠数据传输协议-GBN 协议的设计与实现</td></tr>
<tr><td>姓名</td><td colspan="2">卢兑玳</td><td>院系</td><td colspan="2">计算机科学与技术学院</td></tr>
<tr><td>班级</td><td colspan="2">1803501</td><td>学号</td><td colspan="2">L170300901</td></tr>
<tr><td>任课教师</td><td colspan="2"></td><td>指导教师</td><td colspan="2"></td></tr>
<tr><td>实验地点</td><td colspan="2"></td><td>实验时间</td><td colspan="2"></td></tr>
<tr><td rowspan="2">实验课表现</td><td>出勤、表现得分(10)</td><td></td><td rowspan="2">实验报告
得分(40)</td><td rowspan="2"></td><td rowspan="2">实验总分</td><td rowspan="2"></td></tr>
<tr><td>操作结果得分(50)</td><td></td></tr>
<tr><td colspan="6">教师评语</td></tr>
<tr><td colspan="6"></td></tr>
</table>

计算学部

实验目的：

（注：实验报告模板中的各项内容仅供参考，可依照实际实验情况进行修改。）

本次实验的主要目的。

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。

实验内容：

概述本次实验的主要内容，包含的实验项等。

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）

实验过程：

以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

GBN 协议：

首先，实现 GBN 的单向传输，创建一个套接字，并绑定在指定的端口上。客户端（接收端）请求数据，读取控制台的请求信息，并解析该命令。根据不同的命令，请求不同的数据。

当执行单向传输的命令时，客户端首先发送请求信息，然后服务器端解析请求，进行一个握手阶段，首先服务器向客户端发送一个 205 大小的状态码（我自己定义的）表示服务器准备好了，可以发送数据；客户端收到 205 之后回复一个 200 大小的状态码，表示客户端准备好了，可以接收数据了；服务器收到 200 状态码之后，就开始使用 GBN 发送数据了，服务器端读取本地文件，放到缓存中，发送给客户端。

在发送端设置分组丢失率和ACK丢失率（默认二者都为0.2），ACK采用累积确认（取数据帧的第一个字节，为 ACK 对应分组的序列号），当收到一个字段的序列号时，在其之前的所有分组全都确认被收到。当发生超时情况时，发送端重新发送整个窗户中的所有数据分组。

GBN 双向传输也是相同的原理。只不过发送端变成了客户端。

SR协议：

连接过程（握手阶段）与 GBN 相同，都是根据205状态码和200状态码来判断是否能够正常建立连接。不同的是，SR 协议，接收端也有一个窗口，当发送的数据在窗口中是，即使前面有还没有收到的分组，也会将该组缓存下来，并发送该分组的ACK，采用超时重传技术，即哪个分组的 ack超时了，就发送哪个分组；当收到在窗口外的分组时，则丢掉。

所设计 GBN 协议数据分组格式、确认分组格式、各个域作用：

GBN 协议数据分组格式：

Seq：为 1 个字节，取值为 0~255，（故序列号最多为 256 个）；

Data：小于等于1024 个字节，为传输的数据；

0：最后一个字节放入 EOF0，表示结尾。

确认分组格式：

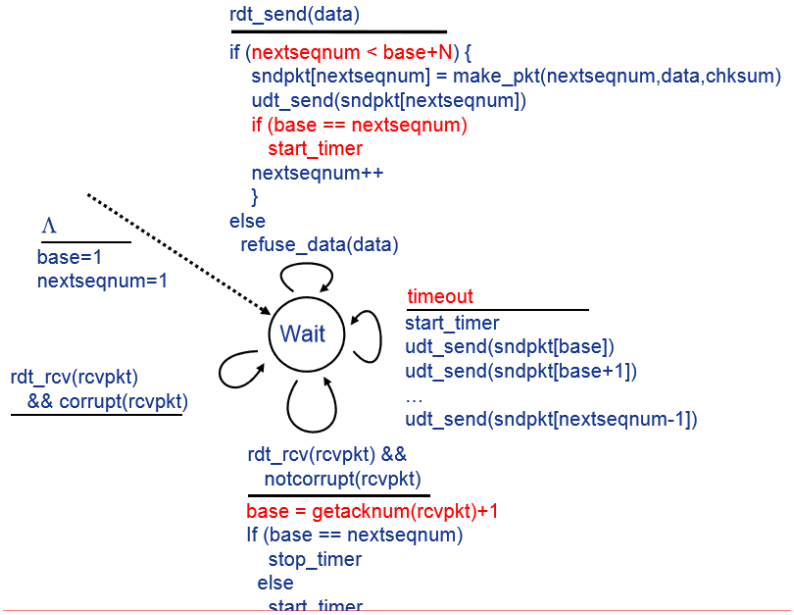
ACK	0

ACK 字段为一个字节，表示序列号数值；

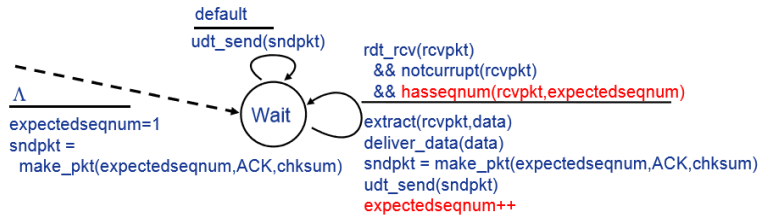
末尾放入 0，表示数据结束。

GBN 协议两端程序流程图：

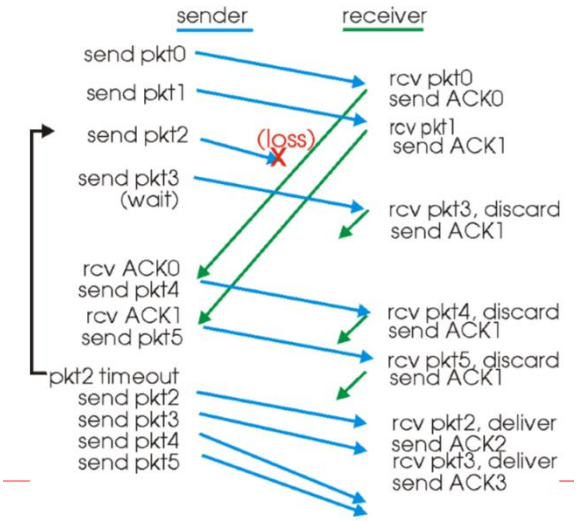
服务器端（发送端）：



客户端（接收端）：



GBN 协议典型交互过程：



所设计SR协议数据分组格式、确认分组格式、各个域作用：

SR协议数据分组格式：

Seq：为 1 个字节，取值为 0~255，（故序列号最多为 256 个）；

Data：小于等于 1024 个字节，为传输的数据；

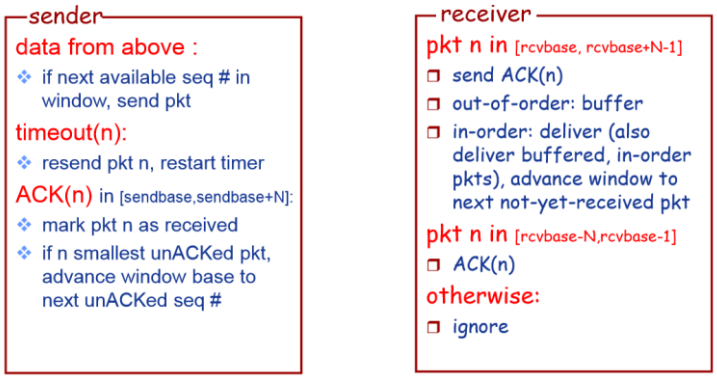
0：最后一个字节放入 EOF0，表示结尾。

确认分组格式：

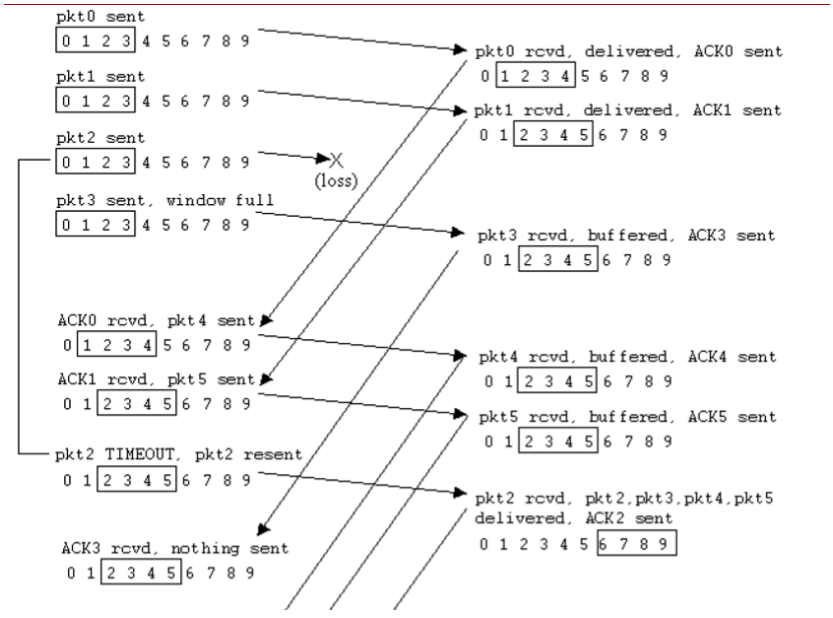
ACK 字段为一个字节，表示序列号数值；

末尾放入 0，表示数据结束。

SR协议两端程序流程图：



SR协议典型交互过程：



数据分组丢失验证模拟方法：

设置一个分组丢失率，将丢失率乘上100，随机生成一个0到100的数，检查该数是否在0到丢失率的范围内。如果在，则分组丢失；否则，没有丢失。

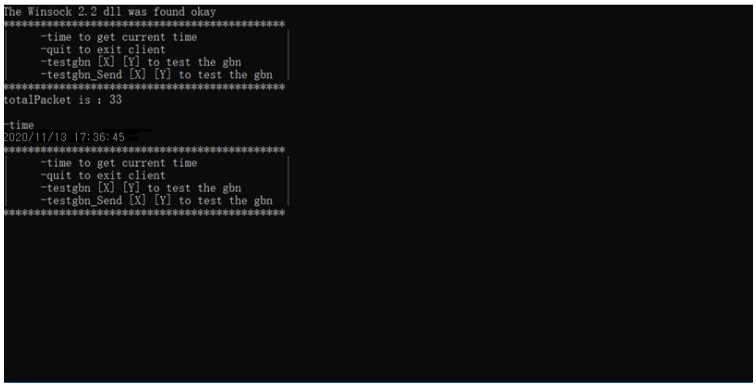
ACK 的丢失模拟也是同样的方法，二者调用相同的模拟丢失函数。

程序实现的主要类（或函数）及其主要作用：

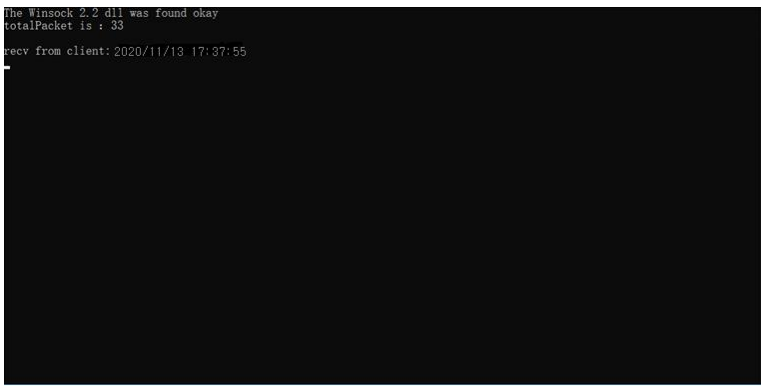
主要有4个项目：
GBN：GBN_server.cpp，GBN_client.cpp
SR：SR_client.cpp，SR_server.cpp
主要函数及其作用：
void getCurTime(char *ptime)：获取当前系统时间，结果存入 ptime 中
BOOL lossInLossRatio(float lossRatio)：根据丢失率随机生成一个数字，判断是否丢失,丢失则返回 TRUE，否则返回 FALSE
void printTips()：打印提示信息：
-time 从服务器端获取当前时间
-quit 退出客户端
-testgbn [X] 测试 GBN 协议实现可靠数据传输
[X] [0,1] 模拟数据包丢失的概率
[Y] [0,1] 模拟 ACK 丢失的概率
bool seqIsAvailable()：当前序列号 curSeq 是否可用
void timeoutHandler()：超时重传处理函数，滑动窗口内的数据帧都要重传
void ackHandler(char c)：收到 ack，累积确认，取数据帧的第一个字节

实验结果：

采用演示截图、文字说明等方式，给出本次实验的实验结果。
GBN 实验运行结果
-time获取时间测试：
客户端：

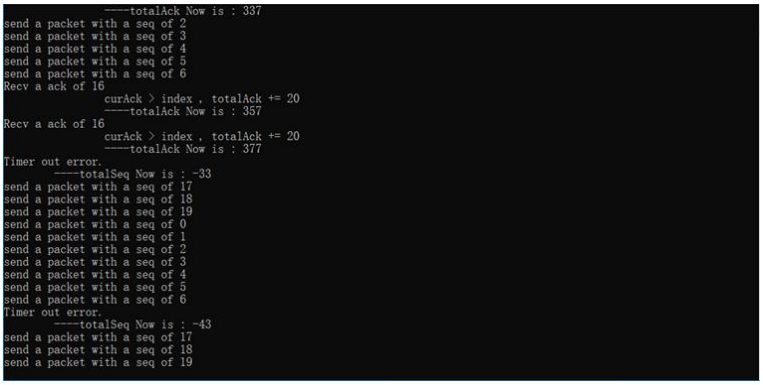


服务器端：

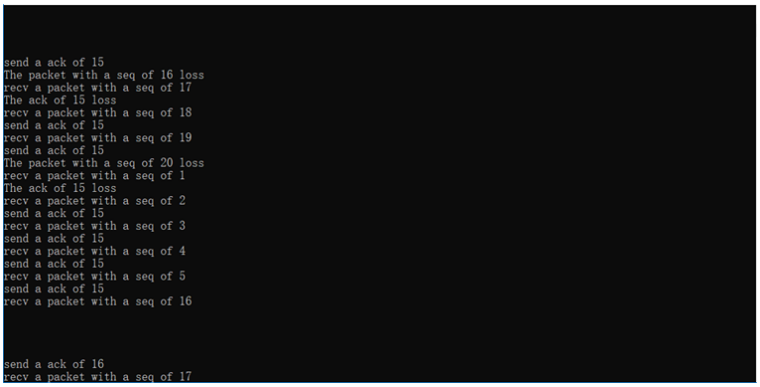


-testgbn 单向传输：

服务器端（发送端）：



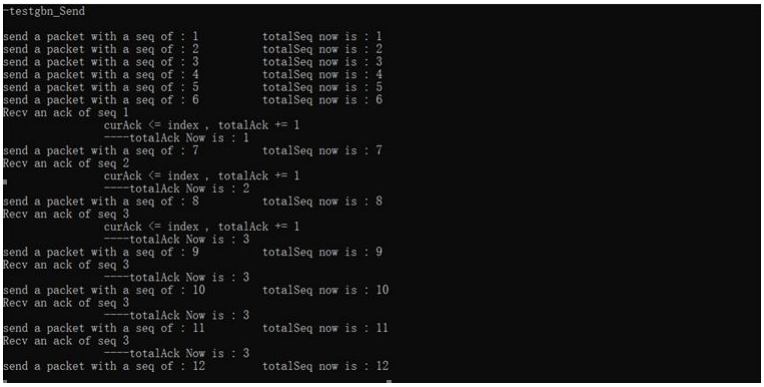
客户端（接收端）：



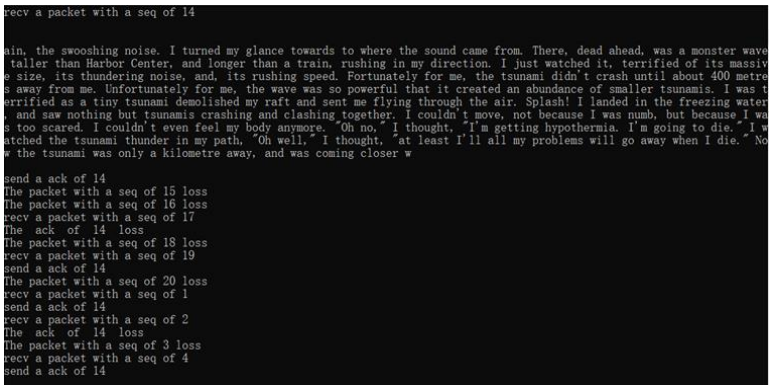
GBN 双向传输：

-testgbn_Send：

客户端（发送端）：

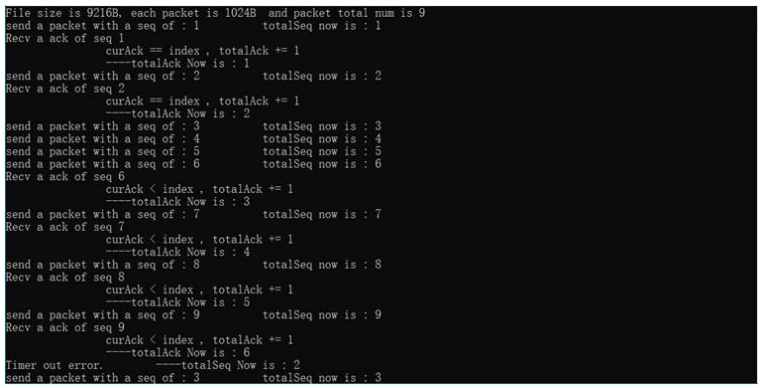


服务器端（接收端）：

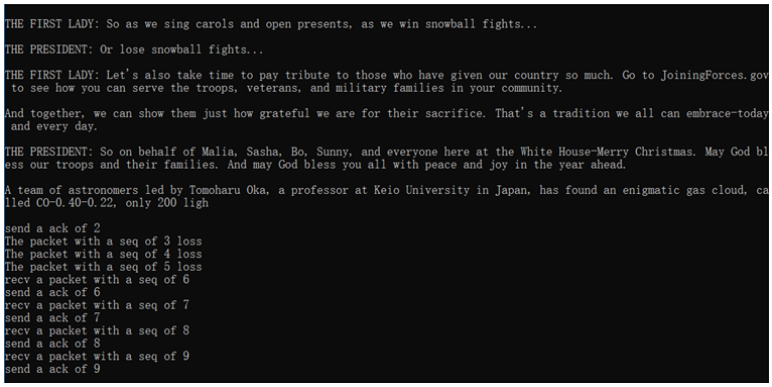


SR 运行结果：

服务器端（发送端）：



客户端（接收端）：



问题讨论：

对实验过程中的思考问题进行讨论或回答。对实验过程中的思考问题进行讨论或回答。

SR协议和GBN协议的区别主要在哪里？

GBN特点：

因为网络中流量控制的原因，它需要对这些被发送的、未被确认的分组的数目N，否则就会造成网络的拥塞。在GBN协议中，发送方可以再窗口大小N的限制内发送足够多的分组，接收方接收到分组后就发送ACK给发送方（例如：接受到分组0，发送ACK 0），当如果发送方接收到连续的ACK（例如ACK 0和ACK 1）时，该窗口就向前滑动，发送方方

便传输新的分组。在接收方，分组丢失了话，就必须从那个分组起再重新传那个丢失的分组号码之后的所有的分组（例如分组2丢失，因此分组3、4、5被认为是失序的分组被丢弃，必须再重新传）但是，这样的话无需接收方准备一定空间的缓存来储存分组。

SR特点：

SR协议相比GBN协议而言，其在接收方增加了接收窗口，对于接收窗口内乱序到达的分组进行缓存，当有一定数量的分组确认后将接收窗口向前滑动；在发送方，增加针对于每个数据包的计时器，不采取累计确认机制，对于每个数据包超时单独进行重传。

心得体会：

结合实验过程和结果给出实验的体会和收获。

本次实验虽然做的很仓促，但是确实实现很多功能。并让我对socket编程有了进一步的了解，并且，对GBN和SR协议有了更深的认识。