



哈尔滨工业大学  
Harbin Institute of Technology

# 计算机网络 课程实验报告

实验名称	可靠数据传输协议（停等协议与 GBN 协议）					
姓名			院系	计算机科学与技术		
班级			学号			
任课教师			指导教师			
实验地点			实验时间			
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						

**实验目的：**

本次实验的主要目的。

1. 理解可靠数据传输的基本原理；掌握停等协议的工作原理；掌握基于 UDP 设计并实现一个停等协议的过程与技术。
2. 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 设计一个简单的停等协议，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) 改进所设计的停等协议，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) 基于所设计的停等协议，实现一个 C/S 结构的文件传输应用。（选作内容，加分项目，可以当堂完成或课下完成）
- 5) 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）。
- 6) 模拟引入数据包的丢失，验证所设计协议的有效性。
- 7) 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）
- 8) 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。

**1.停等协议的设计与实现****1.1简单实现**

停等协议是发送双方传输数据的一种协议方式。停等协议的接收、发送双方仅需设置一个帧的缓冲存储空间和帧序号只取0或1的两个状态标志位，便可有效地实现数据重发并确保接收方接受的数据不会重复。简单来说，停等协议就是发送方发送数据，等到接收方回复ack的时候才继续发送下一段数据。

**1.2验证：**

在实验过程中在数据发送过程中引入一个随机丢包以及随机丢失ack的情况，对于简单的停等协议来说，发生丢包或者丢失ack的情况下，也就是发送方并没有接收到接收方回复的ack的情况下，接收方当等待一定时间之后将会重传当前分组，直到接收到当前分组对应的ack的时候才会继续发送下一个分组。

**1.3双向数据传输：**

当实现双向数据传输的时候，主要的方式就是在客户端和服务端都设置了接收和发送窗口，对于停等协议来说两种窗口的大小都设置为1，当客户端向服务器端发送数据的时候启用客户端的发送窗口和服务端的接收窗口；当客户端从服务器端下载数据的时候启用客户端的接收窗口和服务端的发送窗口。

**1.4 C/S结构的文件传输应用：**

对于C/S结构的文件传输来说，由于上一步实现了双向数据传输，因此在文件传输应用的时候实现的是可以提供客户端上传与下载数据的应用。在实现过程中根据客户端输入的命

令决定是上传还是下载，同时将这一命令发送给服务器端，在服务器端决定是否执行这个服务，如果服务端拒绝提供服务，那么本次连接中断；如果服务器端同意提供服务，以客户端上传文件为例讲述下述过程，客户端下载文件过程类似。当服务器端允许开始服务后，客户端进入传输数据状态，而服务器端进入接受数据状态。当数据传输结束之后客户端向服务器端发送特定报文内容标志传送任务结束，则本次客户端与服务器端的服务完成。

## 2.GBN协议的设计与实现

### 2.1GBN协议简单实现：

与简单停等协议相比，GBN协议事实上就是把发送窗口的长度设置为 $n$ ，而接收窗口的长度仍保持为1，也就是GBN协议不接受乱序到达的数据，如果数据乱序到达，那么在接收处将会决定放弃这一部分数据；而GBN协议采用的是累积确认的方式，也就是说如果当前等待接收的数据编号为1,2,3，而收到了编号为3的ack，那么就表示这三个数据已经都正确接收了。GBN协议流程图如下所示：

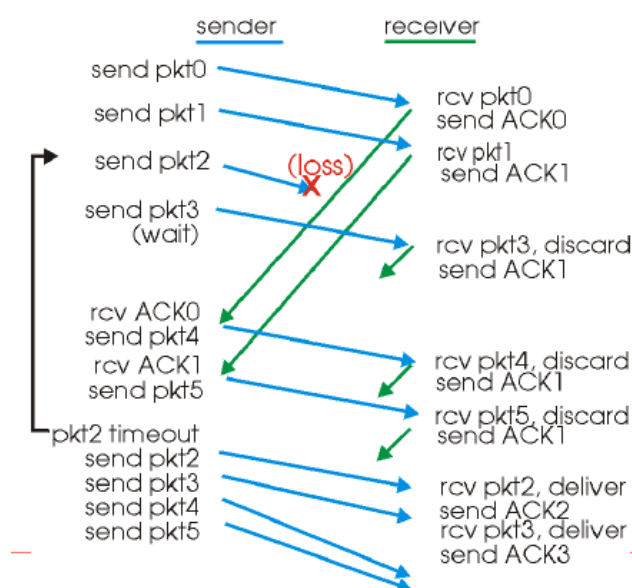


图2.1 GBN流程图

### 2.2验证：

GBN验证方法和上述停等协议中基本一致，可见上述验证过程叙述。

### 2.3双向数据传输：

GBN双向数据传输方法和上述停等协议中基本一致，可见上述验证过程叙述。

### 2.4改进为SR协议：

为了解决GBN只能顺序接收数据的情况，在SR协议中，对于发送方来说：从上层收到数据后，SR发送方检查下一个可用于该帧的序号。如果序号位于发送窗口内，则发送该数据帧。否则要么将数据缓存，要么返回给上层之后再传输；同时当发送方接收到某个ACK的时候，将这个被确认的帧标记为已接收。如果该帧序号是窗口的下界，则窗口向前移动到最小序号的未被确认帧处。如果窗口移动了且有序号在窗口内的未发送帧，则发送这些帧。而对于超时事件来说，SR协议只重传超时的帧，并不重传所有超时帧之前的帧。而对于SR接收方，只要是接收窗口内的帧都可以接收，SR接收方将确认一个正确接收的帧而不管其是否按序，失序的帧将被缓存，并返回给发送方一个该帧的确认帧，直到所有帧（序号更小的帧）都被收到为止，这时将一批帧顺序交付上层，然后向前移动滑动窗口。详细内容可见MOOC讲解截图如下：

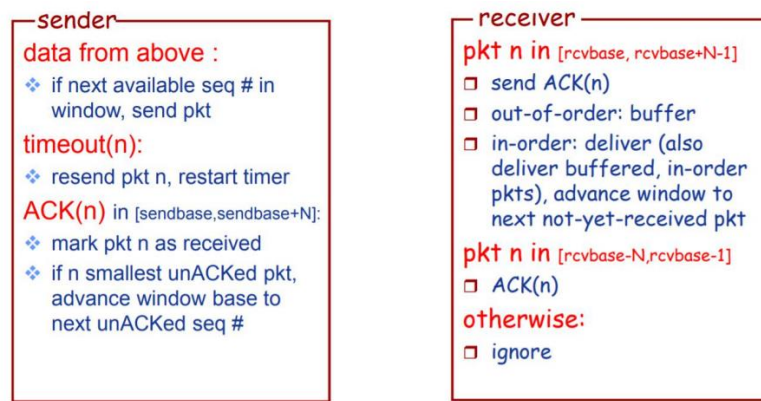


图 2.2 SR 协议图

最终设计的基于 SR 协议的文件传输应用流程如下：

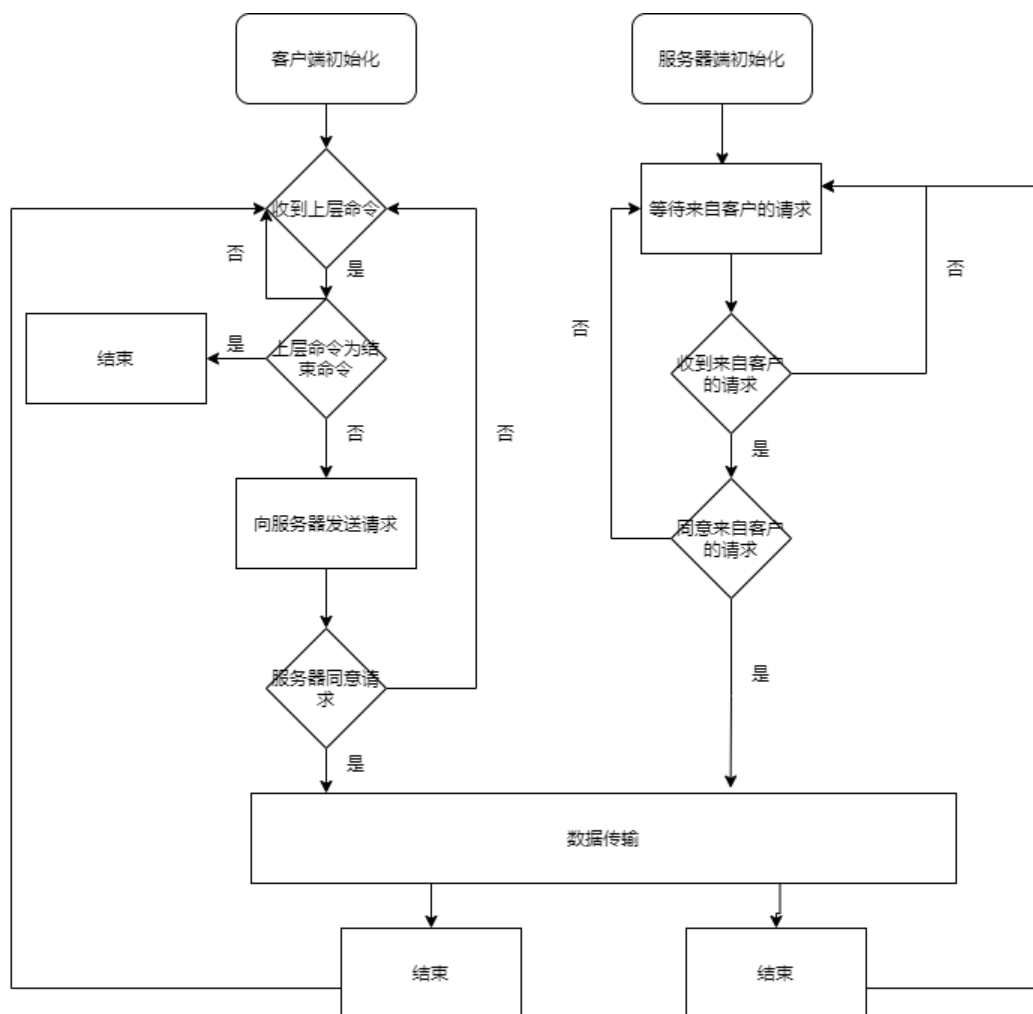


图 2.3 文件传输应用流程图

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。

### 1.GBN:

由于停等协议就是GBN协议中发送窗口长度设置为1，因此停等协议与GBN协议验证过程类似，统一演示如下：

获取服务器端时间：

```
The Winsock 2.2 dll was found okay
*****
| -time to get current time |
| -quit to exit client |
| -testgbn [X] [Y] to test the gbn |
*****
-time
2021/10/7 16:50:15
*****
```

图1.1 获取服务器端时间

模拟丢包：

```
begin to test gbn protocol, please don't abort the process
The loss ratio of packet is 0.20, the loss ratio of ack is 0.20
Ready for file transmission
recv a packet with a seq of 1
send a ack of 1
recv a packet with a seq of 2
send a ack of 2
The packet with a seq of 3 loss
recv a packet with a seq of 4
The ack of 2 loss
recv a packet with a seq of 5
send a ack of 2
recv a packet with a seq of 6
send a ack of 2
recv a packet with a seq of 7
send a ack of 2
recv a packet with a seq of 8
The ack of 2 loss
The packet with a seq of 9 loss
recv a packet with a seq of 10
The ack of 2 loss
recv a packet with a seq of 11
send a ack of 2
recv a packet with a seq of 12
send a ack of 2
```

图1.2 模拟丢包

在实现模拟丢包的过程中，设置的模拟丢包率和ack丢包率都是0.2，由上面的运行截图中可以看出，由于客户端发送的标号为2的ack报文丢失，因此在服务器端标号为2的数据发生超时之后要重传标号为2后面的所有数据，这样在一定的程度上就造成了资源的浪费，因此引入了SR协议。

### 2.SR文件传输：

在SR实现的文件传输服务中提供了客户端上传与下载文件的功能，演示结果如下：

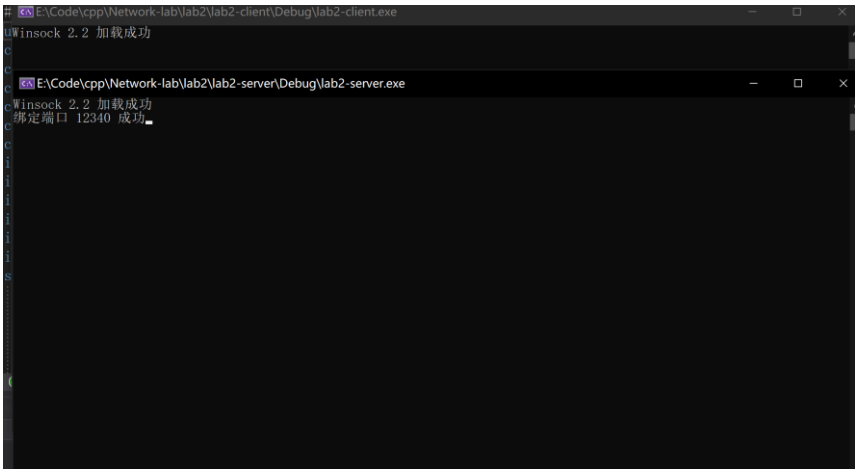


图2.1 初始状态

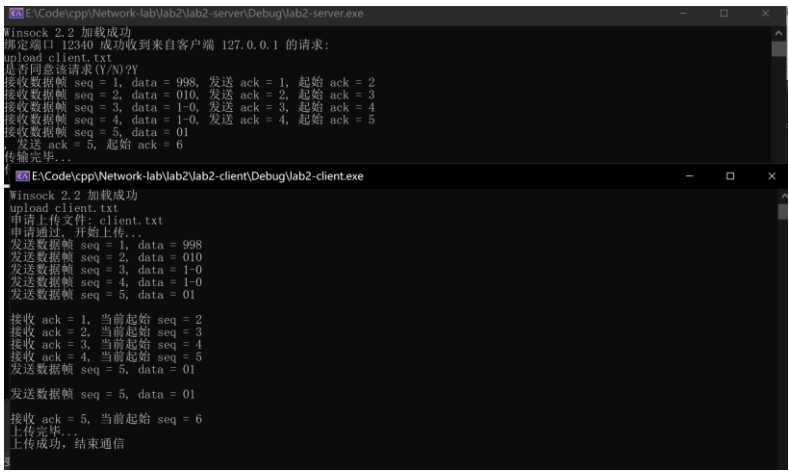


图2.2 客户端申请上传文件

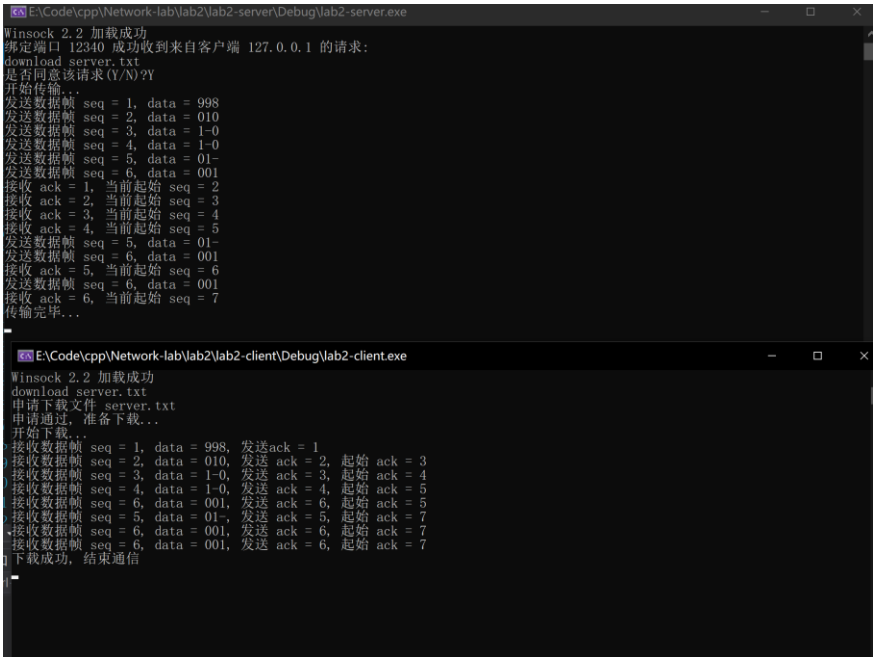


图2.3 客户端申请下载文件

问题讨论:

对实验过程中的思考问题进行讨论或回答。

1. 由于UDP协议是无连接的传输协议，无法保证可靠的数据传输，因此如果在上层使用的时候需要上层来实现可靠的数据传输的部分。而且和TCP协议不同，UDP是一种无连接的协议，因此在传输数据之前并不需要三次挥手建立连接，尽管在本次实验中一部分代码确实使用了类似于连接的方式实现，但是需要注意的是那只是逻辑上的连接，并不存在真实的连接过程。
2. SR协议的出现主要就是为了解决在使用GBN的过程中的一定量的资源浪费的情况，事实上SR协议、GBN协议、停等协议这三者在本质上的区别就是接收窗口和发送窗口大小的区别，至于其他的一些区别都是一些小的区别。
3. 需要注意的是因为SR协议为每一个发送窗口的数据都设置了一个计时器，每次都重传超时的部分。在实际实现的过程中每次只需要比对发送窗口最低位是否超时，如果超时则重传，如果没有超时则表示发送窗口中没有超时的数据。

心得体会：

结合实验过程和结果给出实验的体会和收获。

通过本次实验对于基于UDP协议的可靠数据传输有了更加深刻的认识，掌握了停等协议，GBN协议以及SR协议，同时对于socket编程也有了更深入的了解。