

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 可靠数据传输协议 | | | | | |
| 姓名 | 王丁子睿 | | 院系 | 计算机科学与技术 | | |
| 班级 | 1803104 | | 学号 | 1183710211 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物楼213 | | 实验时间 | 2020/11/6 | | |
| 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 | | | | | | |
|  | | | | | | |

**计算学部**

|  |
| --- |
| 实验目的： |
| 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。  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： |
| 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； 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。 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； 8. 将所设计的GBN协议改进为SR协议。 |
| 实验过程： |
| 1. 首先启动服务器server.py，此时由于客户端未启动，所以所有接收来自客户端的请求均会产生异常，同时间歇性有超时的提示      1. 启动客户端client.py，观察接收到的包和ack的信息。 2. 换用双向传输模式，观察接收到的包和ack的信息。 |
| 实验结果： |
| 1. 根据不同的传输协议，客户的终端会每隔一定时间接收到一个数据和ack，同时由于丢包的存在，接收的时间间隔并不固定    1. 对于停等协议，可以看到接收数据的ack按照0、1交错出现      * 1. 对于滑动窗口协议，设置窗口大小为10，ack的范围为16，可以看到接收数据的ack按照从0到15的顺序循环出现      * 1. 对于选择重发协议，设置窗口大小为10，ack的范围为16，可以看到由于丢包现象的存在，接收数据的ack并没有特定的顺寻，但均在0到15之间      1. 以滑动窗口协议为例，进行双向传输，可以看到在收到结果的同时，会有本地产生的超时信息 |
| 问题讨论： |
| 1. 分组格式   对于三种协议，数据分组的格式均为：”ack message”的形式，其中最后一个分组为”-1 <EOF>”，用来指示当前文件已经传输完毕。   1. 程序执行流程图    1. 停止等待协议      * 1. 滑动窗口协议      * 1. 选择重传协议      1. 数据分组丢失验证模拟方法   在客户端设定一个阈值，每次接收到一个包之后，生成一个随机数，若该随机数小于设定的阈值，则认为没有接收到这个包，不对该包做处理，从而模拟丢包的情况。 |
| 心得体会： |
| 在本实验中，需要特别注意的地方就是如何判断一个文件已经传输完毕，一个比较直观的想法是：在传输文件的最后添加一个<EOF>来指示当前是否为一个文件的结尾。  但是这样的问题在于，某些文件的最后不适合添加一个这样的信息，例如图片文件。此外，SR协议中，即使发送完最后一个包，也不代表传输已经结束。  因此上述代码采用的方式为：send跳出while True循环后，再发送一个信息给客户端，说明文件已经发送完毕。由于此时文件已经被客户端完全接收，因此这一方法不会有提前终止的情况。  但对于多文件传输，这一方法无法区分某个包属于哪个文件，因此，在头部中标明文件的总长度，才是最合适的选择。 |