

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 可靠数据传输协议**-**停等协议的设计与实现&GBN协议的设计与实现 | | | | | |
| 姓名 | 陈啸 | | 院系 | 计算机学院自然语言处理 | | |
| 班级 | 1803104 | | 学号 | 1180300121 | | |
| 任课教师 | 聂兰顺 | | 指导教师 | 聂兰顺 | | |
| 实验地点 | 格物207 | | 实验时间 | 2020年11月7日 | | |
| 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： |
| 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GBN协议，实现单项可靠数据传输(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 4. 将所设计的GBN协议改进为SR协议 |
| 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. GBN协议数据分组格式**   |  |  |  | | --- | --- | --- | | 数据包num | 空格 | data |   例如 数据段为256，num为0，则整个数据包为“0 256”  **2. 确认分组格式**   |  |  |  | | --- | --- | --- | | ACK | 空格 | num |   例如，ACK确认报文确认的为数据包0，则ack包表现为“ACK 0”  **3. 协议两端程序流程图**  GBN协议接收端    GBN协议发送端    **4. 数据分组丢失验证模拟方法**  在实验中模拟数据包丢失的方法，采取实验报告中的建议，对于接收端，接收到的数据帧，以一定的概率发送ACK报文(在实验中使用的概率为50%)，剩余的情况接收端不发送ACK即表现为ACK报文丢失。  **5. 程序中实现的主要类(或函数)及其主要作用**  在实验中主要类有**4种**，分为GBN协议的客户端、服务器端和SR协议的客户端、服务器端。其中**由于GBN协议和SR协议都实现了全双工通信，所以客户端和服务器端都可以互相向对方发送数据，两边的实现是对称的**。  因此以GBNClient和SRClient为例进行说明。  GBNClient为GBN协议中的客户端实现，其中有\_\_send()函数作为发送方时的主要功能，即从可用窗口中发送数据，以及接收并处理来自接收方发回的ACK报文。 \_\_receive()函数主要的功能是作为接收方时，接收数据包，并以一定的概率回复ACK报文，在收到冗余ACK报文的时候丢弃回复带有next\_expected\_num的ACK报文。此外还有\_\_receive\_random\_throw()函数为模拟丢包的接收函数。  SRClient的实现与GBNClient的实现类似，主要的区别在SR协议对于每个数据包是分别确认的，因此，对于每个数据包均有单独的Timer用于计算是否超时，采取单独确认的机制，在接收端的增加了接收窗口，对于乱序到达的数据包进行缓存，而不是仅期待唯一分组的到达。具体实现如下：  GBN协议的send函数    在前面的While循环中，主要是将窗口中可用的序号全部发送出去，有一个特殊处理就是在处理所有的序列号使用完毕的情况，采取mod n除法，在序列号使用完的情况下，可以重新从头开始发送。  另外，在下半部分主要在处理接收ACK的情况，此处采用的非阻塞的方式，利用select包中的select函数，如果无法收到信息，那么就将计时器+1，并且当计时器超时的时候，将所有未确认的数据包全部重发；否则，读取ack信息，更新base信息。  GBN的receive函数    同样是利用非阻塞的方式读取，但是在回复ACK的时候，利用随机函数，以50%的概率进行回复，剩余的50%的概率模拟丢包，不发送ack确认报文；另外对于乱序到达的数据包，没有缓存，直接丢弃并回复带有当前期待收到的数据包的序号的ack报文（确定此种ack报文不会丢失）。  对于SR协议中的Send函数，主要增加的是一个ack确认数组，在已发送还未接收的窗口中的数据包置为1，已确认接收的数据包置为0；另外对于每一个数据包增加timer，在接收到对应的ack报文时，会将其置为0，否则，则将所有未确认的分组对应的计时器全部+1；receive函数，增加了接收窗口，对于在接收窗口内的数据包正确接收，在前一个接收窗口内的数据包([receive\_base – N, receive\_base-1])回复ack，其余数据全部忽略即可。 |
| 实验结果： |
| **对于GBN协议全双工通信的测试**  **源文件见gbn\_client.log和gbn\_server.log**    **对于SR协议全双工通信的测试**  源文件见sr\_client.log和sr\_server.log |
| 问题讨论： |
| **停等协议：**  数据分组格式：序号 数据，中间用一个空格隔开  确认分组格式：ACK 序号，中间用一个空格隔开  域作用：序号：保存分组的序号  ACK：用来标识这个是一个ACK包  数据：用来标识这个数据包所携带的数据。  协议两端程序流程图：  服务端：发送数据->尝试接收ACK->一段时间接收不到ACK，重新发包  客户端：接收数据->发送这个包的ACK  协议交互过程：  发送方发送数据->接收方接收数据->接收方返回ACK->发送方发送下一个包  主要类、函数和作用：实验过程已写  数据分组丢失模拟：使用一个概率来模拟数据丢失，如果随机数大于一个值就停止发这个包造成数据丢失。  UDP编程主要特点：UDP发送和接收都是整个包，自动保持包的边界；UDP包发送接收不保证一定成功，不保证按正确顺序到达。  源代码：见报告附件  **GBN协议：**  数据分组格式：序号 数据，中间用一个空格隔开  确认分组格式：ACK 序号，中间用一个空格隔开  域作用：序号：保存分组的序号  ACK：用来标识这个是一个ACK包  数据：用来标识这个数据包所携带的数据。  协议两端程序流程图：  服务端：发送窗口内数据->尝试依次接收ACK->接收ACK序号和当前分组序号比较，一样则移动窗口接收下一个包/不一样则继续等待原来的包，丢弃后面的所有数据  客户端：接收一组数据->依次发送这组数据的ACK  协议交互过程：  发送方发送一组数据->接收方接收一组数据->接收方依次返回ACK->发送方接收到ACK，移动窗口->发送方发送下一组数据  数据分组丢失模拟：使用一个概率来模拟数据丢失，如果随机数大于一个值就停止发这个包造成数据丢失。  实验验证结果：见上面实验过程 |
| 心得体会： |
| 通过这次实验，我学到了许多新知识，体会到了计算机网络的博大精深。 |