

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 可靠数据传输协议-GBN协议的设计与实现 | | | | | |
| 姓名 | 朱明彦 | | 院系 | 计算机科学与技术学院 | | |
| 班级 | 1603109 | | 学号 | 1160300314 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物213 | | 实验时间 | 2018年11月3日星期六 | | |
| 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： |
| 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： |
| 通过实现GBN协议与SR协议的全双工通信，对于可靠数据通信的认识有了提高。并且实际编程中，对于一些细节的处理是在理论的学习中无法得到的，比如序列号用完了怎么办，如何实现全双工通信等等。这种模拟使得对于协议的理解更加深刻，并对真正实现TCP协议中可靠数据传输的工程师当时的工作表示由衷的敬佩。 |