

**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，  可靠数据传输协议-GBN协议的设计与实现 | | | | | |
| 姓名 | 田雪洋 | | 院系 | 计算机科学与技术学院 | | |
| 班级 | 1937102 | | 学号 | 1190202110 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物207 | | 实验时间 | 2021.10.30 | | |
| 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： |
| 实验2：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。  实验3：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。  实验2：  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，可以不进行差错检测，可以利用UDP协议差错检测  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) 计时器实现方法：对于阻塞的socket 可用int setsockopt(int socket,int level, int option\_name, const void\* option\_value, size\_t option\_len)函数设置套接字发送与接收超时时间；对于非阻塞socket 可以使用累加sleep时间的方法判断socket 接受数据是否超时(当时间累加量超过一定数值时则认为套接字接受数据超时)。  实验3：  1) 基于 UDP 实现的 GBN 协议，可以不进行差错检测，可以利用 《计算机网络》实验指导书 20 UDP 协议差错检测；  2) 自行设计数据帧的格式，应至少包含序列号 Seq 和数据两部分；  3) 自行定义发送端序列号 Seq 比特数 L 以及发送窗口大小 W，应满 足条件 W+1<=2L。  4) 一种简单的服务器端计时器的实现办法：设置套接字为非阻塞方 式，则服务器端在 recvfrom 方法上不会阻塞，若正确接收到 ACK 消息， 则计时器清零，若从客户端接收数据长度为-1（表示没有接收到任何数 据），则计时器+1，对计时器进行判断，若其超过阈值，则判断为超时， 进行超时重传。  5) 为了模拟 ACK 丢失，一种简单的实现办法：客户端对接收的数 据帧进行计数，然后对总数进行模 N 运算，若规定求模运算结果为零则 返回 ACK，则每接收 N 个数据帧才返回 1 个 ACK。当 N 取值大于服务 器端的超时阀值时，则会出现服务器端超时现象。  6) 当设置服务器端发送窗口的大小为 1 时，GBN 协议就是停-等协 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。  由于实验2的停等协议是实验3停等协议的特例，所以下面仅叙述实验3的实验过程   1. GBN协议数据分组格式，确认分组格式，各个域的作用  |  |  | | --- | --- | | 序号 | 数据 |   本次实验采取的GBN协议的数据分组格式和确认分组格式相同，都采用了序号+数据的分组格式，其中  序号表示该数据分组是第几个分组，字段长度为8个字节。  数据表示该次传输的数据，字段长度为0-1024字节，即最大能够传输1024字节。而在确认分组中，数据域的字段长度为0.    SR和GBN采用相同的数据分组和确认分组，各个域的作用相同。     1. 协议两端程序流程图   GBN协议接收端：    GBN协议发送端：    SR协议接收端：    SR协议发送端：     1. 数据分组丢失验证模拟方法   本次实验中采取的模拟数据包丢失的方法，是在接收端中，对于接收到的数据帧，以一定的概率发送ACK报文(在实验中使用的概率为20%)，剩余的情况接收端不发送ACK即表现为ACK报文丢失。  通过这种方法即实验了数据分组的丢失模拟。   1. 协议典型交互过程   **GBN协议：**  在发送方端，发送端把需要发送的分组加入发送窗口，然后向接收端发送窗口内的分组。对于接收端来说，若接收到了期望收到的分组，则向发送方返回一个该分组的ACK，否则发送上一个正确接收的分组的ACK。对于发送端来说，发送端收到ACK分组后会将waitSeq移动到收到的ACK序列号+1处，然后进入下一轮循环。这样不断循环便能完成所有的文件发送。  **SR协议：**  SR协议与GBN协议的不同之处在于拥有一个接收窗口，因此需要为每个分组都设置一个计时器，当经过一次循环后，将重发发送窗口中超时的那些分组。对于接收方来说，如果收到乱序但序列号在接收窗口内的分组，则接收方会把它缓存，然后向发送方返回该分组的ACK。当收到按序到达的分组时，接收方会检查接收窗口内是否有能合并的已接收的分组，当这些分组能够从接收窗口的最左端合并时，先将这些分组合并，然后一起提交给上层协议，并滑动接收窗口到未提交给上层协议的位置处。这样不断循环便能完成所有的文件的发送。   1. 程序实现的主要类（或函数）及其主要作用     由于SR是在GBN基础上改进而来的，因此，他们主体结构类似，下面对于SR不同于GBN的设计，将以红字标出    1.    该函数用于判断发送端是否需要进行发送数据包，当序列号在发送窗口中，并且该序列号未被确认，这时，可以进行数据发送。    SR除了未确认时可以进行数据发送，若ack数组被标记为重传，也需要进行数据发送  2.    该函数是用于对收到的ack进行解析，若收到的ack为想要收到的ack，则滑动窗口右移，并把相应位的ack标记为已确认。    SR协议，若收到的ack为想要收到的ack，则滑动窗口右移，并把相应位的ack标记为已确认。若收到的ack不是想要的，同样进行确认，但滑动窗口不移动  3.    该函数的作用是从系统获取时间  4.    该部分的作用是从文件中读入要发送的文件  5.    这部分的内容是超时处理，若发生超时，从base开始，之后的所有ack都变为可用，并将确认总数减少。    SR协议把原来的单个变量timeout记时改为使用clocktimer数组进行计时，当该ack未被确认时，遍历clocktimer数组，对所有ack未被确认的时间+1，若发生超时错误，把接受到的totalSeq减少相应数量，并把nextseqnum重新移到base处。  5.    该函数的作用是根据输入的丢失率，随机生成一个数字，判断是否丢失，丢失则返回TRUE  **6.全双工处理**  **对于全双工，只需要使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**。即发送端和接收端的身份互换即可。   1. 源代码   由于本次实验源代码过多，在这里就不粘到报告中，详细源代码见源代码文件夹，下面给出各个cpp文件的含义：   |  |  | | --- | --- | | 文件名称 | 含义 | | GBN\_client.cpp | GBN接收端 | | GBN\_server.cpp | GBN发送端 | | SR\_client.cpp | SR接收端 | | SR\_server.cpp | SR发送端 | | stop\_wait\_client.cpp | 停等协议接收端 | | stop\_wait\_client.cpp | 停等协议发送端 | |
| 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.-time功能演示：      2.GBN功能演示：    可以看到，当接收端seq为6的数据包丢失后，接收端一直发送他期望收到的ack5，而发送端当收到后，会立刻重传从seq为6的数据包和之后的数据包  接收文件和发送文件对比：  接收文件：    发送文件：    可以看到文件实现了准确无误的传输。  3.GBN客户端向服务器端发送文件演示：      接收文件和发送文件对比：  服务器端接收文件：    客户端发送文件：    可以看到文件实现了准确无误的传输。  2.SR功能演示：    可以看到，当接收端seq为16，1,2,3的数据包丢失后，接收端只会发送未确认的数据包  接收文件和发送文件对比：  接收文件：    发送文件：    可以看到文件实现了准确无误的传输。  3.SR客户端向服务器端发送文件演示：      接收文件和发送文件对比：  服务器端接收文件：    客户端发送文件：    可以看到文件实现了准确无误的传输。 |
| 问题讨论： |
| 对实验过程中的思考问题进行讨论或回答。  1.开始不知道怎么进行模拟丢包，后来通过查阅相关资料和阅读实验指导书，知道了可以通过引入一个概率因子，然后不处理当前收到的数据包进行模拟。  2.对于双向数据传输的实现，其实就是在客户端和服务器端同时实现GBN和SR的发送方和接收方。  3.SR协议的重传机制相对GBN更加复杂，但却效率更高，因为不需要重传发送窗口的所有文件，SR使用了空间换时间。 |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  同过本次实验，更加深刻的理解到了GBN和SR协议的全双工通信，对可靠数据通信协议有了更深刻的认识，同时，也意识到，仅仅是理解了原理，是不够的，还会在工程上面临许多困难，比如计时器如何设计，如何循环利用序列号，当程序出现问题后，这种需要双方通信的程序如何debug |