

**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 | 陈东鑫 | | 院系 | 计算机科学与技术学院信息安全 | | |
| 班级 | 1903202 | | 学号 | 1190201308 | | |
| 任课教师 | 聂兰顺 | | 指导教师 | 聂兰顺 | | |
| 实验地点 | 格物213 | | 实验时间 | 2021-11-06 8:00-9:45 | | |
| 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.可靠数据传输协议-停等协议的设计与实现：  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.可靠数据传输协议-GBN 协议的设计与实现：  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.可靠数据传输协议-停等协议的设计与实现：  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）  2.可靠数据传输协议-GBN 协议的设计与实现：  1) 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）。  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；（选作内容，加分项目，可以当堂完成或课下完成）  4）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。  将GBN协议的滑动窗口设为1则变为停-等协议，故我们只需要先写GBN协议代码，就能很容易地得到停-等协议，然后在GBN的基础上添加缓存和更多计时器将其修改为SR协议。  一、设计GBN协议：  使用UDP协议传输数据，收发数据使用非阻塞方式，为了保证数据可靠传输，我们需要设计数据传输步骤，在实验指导书上面的参考代码基础上进行改进。  1.服务器端：  1）命令解析  为了测试客户端与服务器端的通信交互，方便操作，设置了此过程。  首先，服务器接收客户端发来的请求数据，  “-time”表示客户端请求获取当前时间，服务器回复当前时间；  “-quit”表示客户端退出，服务器回复“Good bye!”；  “-testgbn”表示客户端请求开始测试 GBN 协议，服务器开始进入GBN 传输状态；  其他数据，则服务器直接回复原数据。  2）数据传输数据帧格式定义  在以太网中，数据帧的 MTU 为 1500 字节，所以 UDP 数据报的数据部分应小于 1472 字节（除去 IP 头部 20 字节与 UDP 头的 8 字节），为此，定义 UDP 数据报的数据部分格式为：  IMG_256  Seq 为 1 个字节，取值为 0~255，（故序列号最多为 256 个）；  Data≤1024 个字节，为传输的数据；  最后一个字节放入 EOF0，表示结尾  3）传输开始与结束  模拟TCP三次握手与四次挥手，这里只进行一次握手与挥手。  传输开始时，向客户端发送序列号为200的分组，表示服务器准备好传输数据，等待客户端  回复205分组，表示客户端准备好传输数据，开始传输数据。  totalSeq > totalPacket时，表示已经输出分组数量大于总要传输的分组数量，但这时可能有的ACK还没有收到，我们需要检查分组ACK是否收到，若有没收到的ACK继续向客户端发送空分组，否则当客户端的ACK丢失，客户端不会向服务器端重传ACK；若都收到，则向客户端发送序列号为204的分组，表示传输完成，结束分组传输。  qt_temp  2.客户端：  1) ACK 数据帧定义  IMG_256  由于是从服务器端到客户端的单向数据传输，因此 ACK 数据帧不包含任何数据，只需要将 ACK 发送给服务器端即可。  ACK 字段为一个字节，表示序列号数值；末尾放入 0，表示数据结束。  2）命令设置  客户端的命令和服务器端的解析命令向对应，获取当前用户输入并 发送给服务器并等待服务器返回数据，如输入  “-time”得到服务器的当前 时间。  此处重点介绍“-testgbn [X] [Y]”命令，[X],[Y]均为[0,1]的小数，其中：  [X]表示客户端的丢包率，模拟网络中报文丢失；  [Y]表示客户端的 ACK 的丢失率。（使用随机函数完成）。  如果用户不输入，则默认丢失率均为 0.2。  3）丢包  使用LossInLossRatio()函数，输入丢失率（分组或客户端），在函数内首先获得1到100的随机数，根据丢失率获得随机数限制，当随机数在限制范围内表示超时，输出true；否则表示非超时，输出false。  4）传输开始与结束  收到服务器端的序列号为200的分组，回复205分组，表示准备好传输，等待服务器端的数据。  收到服务器端的序列号为204的分组后，结束分组接收。  qt_temp  3.扩展：  1）双向数据传输：  在上一步编写的程序基础上，将服务端与客户端通过多线程整合到一起，实现一个端口能够同时做到收发。  具体方法：在主线程中使用客户端程序，在进入while前创建子线程，令其handle函数服务器端函数，将这个代码复制两份，然后将其中一份的发送和接收的IP地址（SERVER\_IP,CLIENT\_IP）与端口号（SERVER\_PORT,CLIENT\_PORT）调换（若接收地址为0.0.0.0，变为发送地址时需要使用本机具体地址或环回地址进行验证），使通信的两个端口能够发送和接收IP和端口对应。  2）停-等协议：  将GBN协议中的窗口大小（SEND\_WIND\_SIZE）改为1，GBN协议变为停-等协议，且能实现双向数据传输。  3）C/S：  将输入输出的数据分别通过文件读写，在传输与接收之前分别通过read和write读和写一个数据报中数据大小的缓存，实现文件有序可靠的发送和接收。  二、设计SR协议：  在GBN协议的基础上通过为每一个发送窗口增加缓存和计时器，为接收窗口增加缓存，实现SR协议。以下描述都是在上面的GBN协议的基础上进行修改的内容：  1.服务器端：  1）缓存和重传  创建全局变量，为每个发送窗口创建缓存（cache[SEND\_WIND\_SIZE + 1][DATA\_SIZE + 1]）与计时器(counter[SEND\_WIND\_SIZE])。  每个计时器在关闭时为-1，在开启时表示超时时间。每次循环，使用click函数将所有开启且没有收到ACK的计时器加一。因为在发送与接收两端有足够的sleep函数，故可以忽略传输时延对sleep的总时间带来的影响，可以使用循环次数来表示超时计数。  发送每个分组，在cache对应序号进行缓存，并开启对应序号计时器（置0）。当接收到ACK时，关闭对应的计时器。  若超时，则将计时器重置，并重新发送cache中缓存的分组。  2）传输结束  由于每个窗口都有自己的计时器，可以不用在发送完毕后继续发送空报文。  2.客户端：  创建全局变量，为每个接收窗口创建缓存（cache[SEND\_WIND\_SIZE + 1][DATA\_SIZE + 1]）。  当分组失序到达时，首先在ack函数中进行标记，然后在cache中缓存分组。  当后到达的分组非失序时，进行累计确认，从最开始失序的分组开始（如果有），将所有缓存的分组进行接收（写入文件），保证数据有序。  其他部分与GBN协议基本相同 |
| 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。  检测用传输数据：每行1023个数字的txt文件（server\_in.txt , client\_in.txt），加上每行的换行符一行共1024个字符，便于比较。  IMG_256  GBN：  如图，在传输双方文件目录内分别有client\_in.txt和server\_in.txt作为检测用传输数据，内部数据和上面图示一样。  IMG_256  启动程序，如下图，可以实现双向传输  IMG_256  当输入-time可以获得对方的时间，输入非标准格式输入则对方返回相同的输入。  IMG_256  进行双向传输，中间可能会出现分组丢失  IMG_256  当双方分别输出“xxx send finish!”“Receive finished!”，表示传输完成  IMG_256  查看双方文件目录，会出现从对方传输得到的文件server\_out.txt和client\_out.txt  IMG_256  查看文件内容，可以发现数据分别与client\_in.txt与server\_in.txt相同，表示可以实现可靠数据传输  IMG_256  IMG_256 |
| 问题讨论： |
| 对实验过程中的思考问题进行讨论或回答。  在检测数据是否传输完成时，对于GBN协议会出现数据乱序或重复  答：GBN协议当服务器端发送完所有分组后等待ACK时不能停止传输数据，即使继续传输空数据也不要停止传输。因为在客户端后续发送的ACK都丢失时，因为后续服务器端停止发送分组，客户端不会回复ACK，则服务器端不会知道客户端已接收的分组序号，以为分组全部丢失，就会在计时器超时全部重传，则ACK丢失的部分（客户端已经接收过）就会重复，若代码逻辑有问题或鲁棒性不足还有可能出现乱序或卡死现象。 |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  设计了基于UDP的停-等协议、GBN协议、SR协议的服务端与客户端，基本实现了简单的通信双方的有序可靠的数据传输，理解了UDP协议的传输方法和特点，并了解了双向传输的原理，深入理解了停-等协议、GBN协议、SR协议的实现过程与技术，理解和掌握了可靠数据传输和滑动窗口协议的基本原理。 |