

**计算机网络**

**课程实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 |  | | | | | |
| 姓名 |  | | 院系 |  | | |
| 班级 |  | | 学号 |  | | |
| 任课教师 |  | | 指导教师 |  | | |
| 实验地点 |  | | 实验时间 |  | | |
| 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；  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. 分组格式及各个域作用    1. 数据分组格式   采用指导书给出的数据格式    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）。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。   * 1. 确认分组格式     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。   1. 协议两端程序流程图及协议典型交互过程     左图为服务器端流程，右图为客户端流程  停/等协议两端交互    GBN协议两端交互    SR协议两端交互     1. 数据分组丢失验证模拟方法   采用指导书提供的lossInLossRatio函数，根据丢失率随机生成一个数字，判断是否丢失,丢失则返回 TRUE，否则返回 FALSE   1. 程序实现的主要类（或函数）及其主要作用   void getCurTime(char \*ptime)：获取当前系统时间，结果存入 ptime 中  void printTips()：打印提示信息  BOOL lossInLossRatio(float lossRatio)：根据丢失率随机生成一个数字，判断是否丢失  unsigned int seqIsAvailable()：当前序列号 curSeq 是否可用    BOOL seqRecvAvailable(int recvSeq)：判断这个序列号是否是所需的    void timeoutHandler(SOCKET socketServer, SOCKADDR\_IN addrClient)：超时重传函数，在停等/GBN协议中直接重传窗口内所有报文，在SR协议中只重传对应未被确认超时的报文    void ackHandler(char c)：在停等/GBN协议中是累计确认，在SR协议中是对应的报文的确认 |
| 实验结果： |
| 1. -time、-quit命令的验证   客户端    服务器端     1. GBN协议的验证   使用默认的丢包率和ack丢失率都是0.2的参数进行模拟，传输这样一个txt文件    ①客户端发起请求并成功通过握手建立起连接      ②开始发送和接收报文，数据报和ack都未丢失时，在控制台打印传输的报文，服务器端打印接收到的ack信号      ③当数据报和ack其中一个发生丢失时，触发超时重传机制，服务器端打印超时重传的信息      ④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求       1. 双向GBN的验证   在第2部分已经完成了从服务器端向客户端传输文件的过程。现在模拟从客户端向服务器端传输文件。    ①客户端发起请求并成功通过握手建立起连接    ②开始发送和接收报文，数据报和ack都未丢失时，在控制台打印传输的报文，服务器端打印接收到的ack信号      ③当数据报和ack其中一个发生丢失时，触发超时重传机制，服务器端打印超时重传的信息      ④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求       1. SR协议的验证   使用默认的丢包率和ack丢失率都是0.2的参数进行模拟，传输这样一个txt文件    ①客户端发起请求并成功通过握手建立起连接    ②开始发送和接收报文，数据报和ack都未丢失时，在控制台打印传输的报文，服务器端打印接收到的ack信号      ③当数据报和ack其中一个发生丢失时，触发超时重传机制，服务器端打印超时重传的信息          这里运气不好，6号需要反复重传  ④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求     1. 双向SR的验证   在第4部分已经完成了从服务器端向客户端传输文件的过程。现在模拟从客户端向服务器端传输文件。  ①文件信息    ②客户端发起请求，经历握手后完成连接。      注意这回是客户端打印相关文件信息，服务器端准备接收。  ③开始发送和接收报文，数据报和ack都未丢失时，服务器端控制台打印传输的报文，客户端打印接收到的ack信号      注意到SR协议允许乱序接收的特点。  ④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求 |
| 问题讨论： |
| 1.UDP 编程的主要特点   * 无连接：UDP 不需要在发送数据前建立连接，而是直接发送数据，因此其速度快于需要建立连接的协议。 * 无状态：UDP 协议没有保存之前发送或接收的数据报的状态，每个数据报都是独立处理的。 * 不保证可靠性：UDP 协议不保证数据报的到达，尽力而为。 * 数据报结构简单。 * 支持一对一、一对多、多对多、多对一的交互通信。   2.终端打印中文乱码  使用命令chcp 65001进行设置  3.带有详细注释的源代码  见附件 |
| 心得体会： |
| * 加深了对UDP传输、停等协议、GBN协议、SR协议的理解 * 熟悉了socket编程方法 * 巩固了c++代码调试方法 |