

**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 | 任永鹏 | | 院系 | 计算机 | | |
| 班级 | 1737102 | | 学号 | 1173710107 | | |
| 任课教师 | 张英涛 | | 指导教师 | 张英涛 | | |
| 实验地点 | 格物213 | | 实验时间 | 周六12节 | | |
| 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。  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。  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）。  模拟引入数据包的丢失，验证所设计协议的有效性。  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）  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：  写在实验前的话：因为实验二需要做的是停等协议的相关内容，实验三是需要做滑动窗口协议的相关内容，当将滑动窗口协议内容的滑动窗口大小设置为1时，即为停等协议，因此此处的停等协议部分不在此赘述！  下面这张图是GBN协议有穷状态机的示意图：    先定义    如果窗口内有空余,就开始发送分组,直到窗口被占满,计时器开始计时,之后进入接收ACK的状态,收到ACK之后,更新滑动窗口的位置,之后如果计时器超时,就将窗口内所有的分组全部重发一次。之后开始下一次循环。  接收方:  不需要有缓存,只需要记录一个seq值,每成功接收一个数据帧,seq+1,开始循环顺序接收数据帧,对于seq不是目标值得数据帧直接丢弃,如果是符合要求的数据帧,就给发送方发送一个ACK=seq的确认数据帧,直到发送方没有数据传来为止。  GBN的实现就完成了。  Untitled Diagram (3)  SR原理：  SR协议是当接收方发现某帧出错后，其后继续送来的正确的帧虽然不能立即递交给接收方的高层，但接收方可收下来，存放在一个缓冲区中，同时要求发送方重新传送出错的那一帧。一旦收到重新传来的帧后，就可以原已存于缓冲区中的其余帧一并按正确的顺序递交高层。显然，SR减少了浪费，但要求接收方有足够大的缓冲区空间。  SR协议实现:  发送方:  在GBN发送方的基础上,增加一个计时器,对每一个未被确认的分组进行计时。在每次判断是否超时时,需要对所有的计时进行判断,与GBN重传不同的是,SR只对超时的那一个分组进行重传。  Untitled Diagram (5)  双向传输：  发送方发送数据需要占用一个固定的端口,而接收方也需要一个固定的端口来向发送方发送 ACK,所以就可以封装一个完整的协议类,类似于TCP的有连接传输一样,发送方和接收方之间在两个固定的ip和端口之间进行数据的传输,直到双方的传输结束。发送方在使用send()函数进行发送时,也可以同时使用receive()函数进行接收,两个过程并不冲突,可以同时进行。如果要同时收发,就需要同时开一个发送线程和一个接收线程,两个线程独立运行,没有冲突,这样就可以实现双向数据传输了。 |
| 实验结果：  下面上截图    从上面的图片中，可以看到，数据发送完毕，并且出现了模拟丢包的情况，可以看到当收到不是预期的数据包的时候，直接重发分组。              从上面的截图中可以看到，sr协议当遇到分组丢失的情况，会只重传丢失的分组，而不会整个窗口进行重传。同时我这里实现了双向传输，即在客户端发送文件到服务器，服务器接受完成之后会向客户端发送文件，客户端接收文件。 |
| 心得体会：  还是自己太菜了，确实学到很多东西，以后要仔细听mooc了，好多东西没记住 |