项目四 强力回音壁软件设计方案

版本号v1.0

## 1.任务需求描述

1. 客户机能尽自己最大努力向服务器发送数据
2. 服务器能将所有的客户数据全部无损回弹
3. 客户机能统计流量和延时
4. 能对比多线程和基于select框架
5. 服务器具有多线程和各线程基于select的结构，获得更大的服务量和更高的性能。
6. 多个客户机（或多线程设计）向服务器发起大流量“攻击”

## 2.需求分析

1. 客户机能尽自己最大努力向服务器发送数据，使用多线程，每个线程中使用一个套接字全力向服务器发送大量简单重复的字符串。
2. 服务器将所有客户数据全部无损回弹，使用多种不同的处理方式进行相应。
   1. 采用select方式，轮询所有套接字。
   2. 采用多线程，每个线程中单套接字，并发处理各个套接字。
   3. 采用多线程与select结合的方式。
3. 客户机能统计流量和延时。
   1. 流量，通过每个线程完成后，返回发送或接收的字符总数，来计算流量。
   2. 延时，则通过在每一帧信息中加入时间戳，在客户端接收到响应之后，通过帧中时间戳计算出延时。

## 3.关键技术分析

相关的决策及依据。

1. 线程：使用CreatThread function创建新的线程，通过向特定的函数输入特定的参数，来完成线程的创建，执行指定的任务。

CreateThread function：HANDLE WINAPI CreateThread( \_In\_opt\_  LPSECURITY\_ATTRIBUTES  lpThreadAttributes, \_In\_    SIZE\_T         dwStackSize, \_In\_    LPTHREAD\_START\_ROUTINE lpStartAddress, \_In\_opt\_  LPVOID          lpParameter, \_In\_    DWORD          dwCreationFlags, \_Out\_opt\_  LPDWORD         lpThreadId);

1. 客户端接收数据速度，会影响服务器端转发的速度，优先接收服务器的响应数据，再尽全力发送数据。

## 4.数据结构设计

客户端：

 struct InPara{

SOCKET sock;

TIME first\_send; //用于记录发送第一帧的时间

TIME first\_recv; //用于记录接收第一帧的时间

long send\_sum; //需要发送的数据大小

long recv\_sum; //最终接收到响应的数据大小

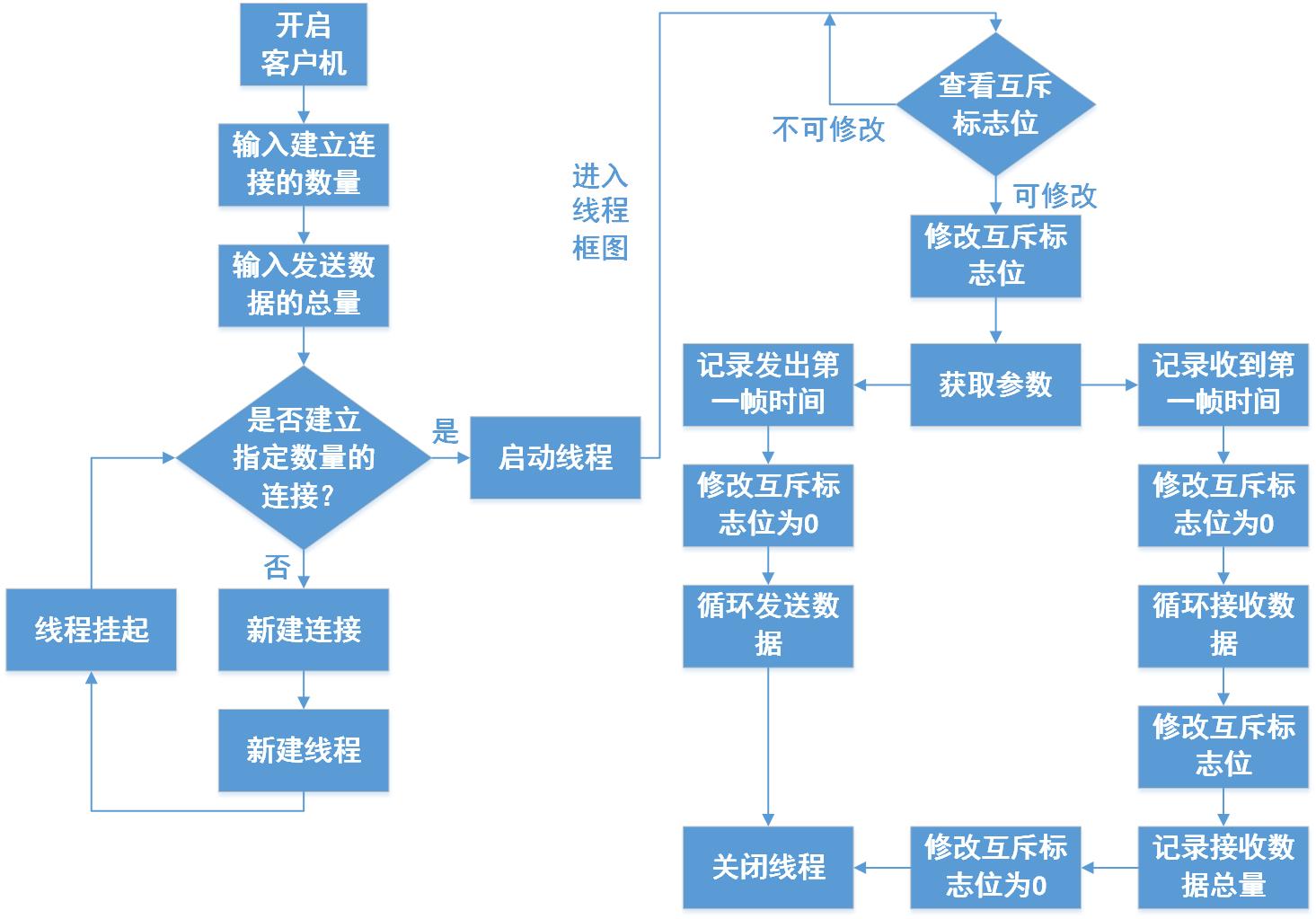
char flag; //为0时代表可使用，为1时代表主函数在使用，

//为2时代表发送线程在使用，为3时代表接收线程在使用

}

## 5.程序框架及流程设计

客户端 ：



## 6.软件设计方案修订

根据调试和测试情况，可能对本版设计方案有所修订，本处记录修订的原因和修订后的方案版本号

6.1、修订原因

6.2、修订后的版本号