项目三 有状态服务设计

版本号v1.0

## 1.任务需求描述

1. 具有用户登录认证功能（模拟CHAP协议的认证方法）
2. 在客户与服务器之间传输文件
3. 差错控制协议设计
4. 支持多用户同时在线

## 2.需求分析

1. 具有用户登录认证功能）

需模拟CHAP协议的认证方法，即：

连接上服务器后，服务器将向客户机发送质询数据，该质询数据由N个整数构成。N的数量是随机的；每个整数的值是随机的；甚至每个整数的长度可能是两个字节，也可能是4个字节，这也是随机的。 客户机将这N个整数相加（溢出进位去掉），把总和与密钥相异或，得到质询结果，传回给服务器。 服务器收到质询结果后，验证这个结果是否正确，并给与响应。

分析：模拟CHAP的认证方法，需考虑正确传输整数的方法与字节序问题。

1. 在客户与服务器之间传输文件

具体功能实现：

客户连接服务器之后，向服务器发送查看文件目录、上传或下载文件的命令，并在命令后加上需要上传或下载的文件名。服务器收到命令后，如果是客户机上传，客户机将文件名上传后，继续将文件内容上传到服务器上。 如果客户机发布的是一个下载命令，则服务器收到命令后，就会按要求将指定文件内容传输到客户机，客户机收到后存盘。如果客户机需要文件不存在，则连接会关闭。

1. 差错控制协议设计，使文件传输功能能够在模拟的差错环境下正确传输。

## 3.关键技术分析

1） 质询报文的构成格式分析：应包含整数的长度，整数数值等要素。由于整数长度可变，且存在网络字节序的问题，实现上可考虑应用位域的方法或按位运算的方法组织整数。2）认证算法的实现

分析：项目需求中制定的认证算法模拟CHAP协议认证过程。CHAP协议认证过程的加密算法在需求分析中已经说明。简而言之，在成功获得服务器传来的整数后，进行加法，异或的简单运算，并准确传回给服务器比较，即可完成认证。

## 4.数据结构设计

服务器，客户端PDU通用。PDU包括质询报文PDU，文件列表传输报文PDU，文件传输报文PDU，控制命令报文PDU。以下列出各PDU并注释。

**1. 质询报文PDU**

#define INQUIRING\_PDU\_TYPE 1 //质询报文类型代号

#define CTL\_BUF\_SIZE 255 //一个报文载荷所能存储的最大字节数

typedef struct InquiryPDUS {

char type; //INQUIRING\_PDU\_TYPE = 1

char num; //存储质询数个数（范围1-16）

char each\_len[4]; //表达每个质询数的长度。每个表达占2位，值1、2、3、0分别表示质询数长度为1、2、3、4。4个字节32bit，最多能表达16个数。

char inquiry\_buf[INQUIRY\_BUF\_SIZE]; //每个质询数，高位存储在前，低位存储在后，符合网络字节序

}InquiryPDU;

**2. 控制命令报文PDU**

#define CTL\_PDU\_TYPE 2

#define CTL\_BUF\_SIZE 255 //一个报文载荷所能存储的最大字节数

//以下为命令取值，改为2开头的两位数，与CTL\_PDU\_TYPE看齐

#define REQ\_ACCOUNT 20 //Server ask Client

#define REQ\_FILELIST 21 //Client to Server

#define REQ\_DOWNFILE 22 //Client to Server

#define REQ\_UPFILE 23 //Client to Server

#define RE\_ACCESS 25 //客户机登陆成功

#define RE\_TIMEOUT 26 //差错控制协议，等待报文超时

#define RE\_ERR\_ACCOUNT 27 //用户名错误

#define RE\_ERR\_INQUIRY 28 //质询结果不通过

#define RE\_ERR\_FILENAME 29 //文件名错误

typedef struct CtlPDUS{

unsigned char type; //类型，可选参数CTL\_PDU\_TYPE = 2

unsigned char cmd; //控制命令，可选参数见于上文

unsigned char len; //报文总长度

char filename[CTL\_BUF\_SIZE]; //载荷，具体包括用户名、请求上下载文件名等

}CtlPDU;

**3. 文件/文件列表传输报文PDU**

#define FILELIST\_PDU\_TYPE 3 //文件列表传输类型

#define FILE\_PDU\_TYPE 4 //文件传输类型

#define FILE\_BUF\_SIZE 255 //一个报文载荷所能存储的最大字节数

typedef struct FilePDUS{

unsigned char type; //可选参数FILE\_PDU\_TYPE = 3 或FILE\_PDU\_TYPE = 4

unsigned char No; //ACK代号，可选参数0 或1

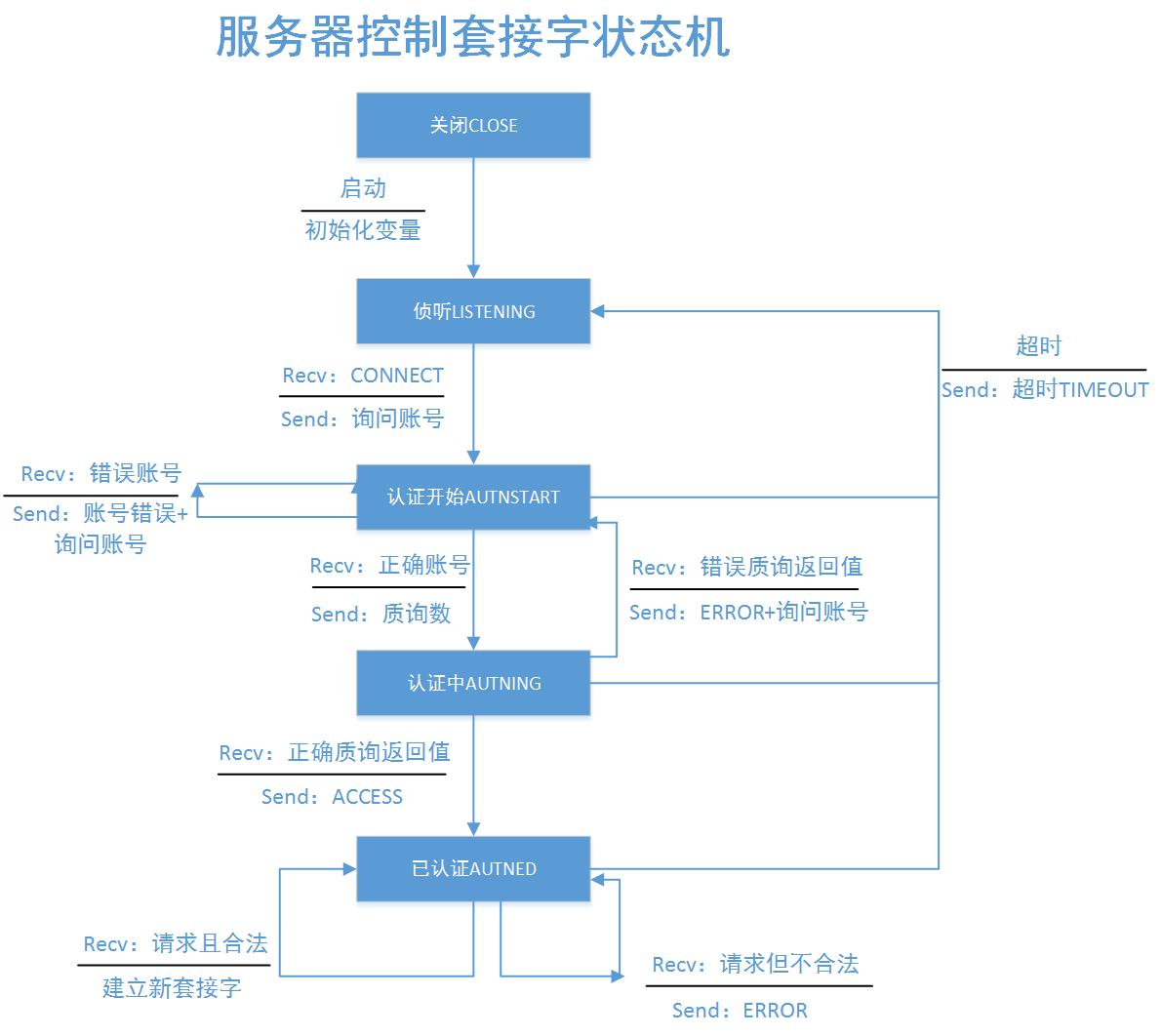
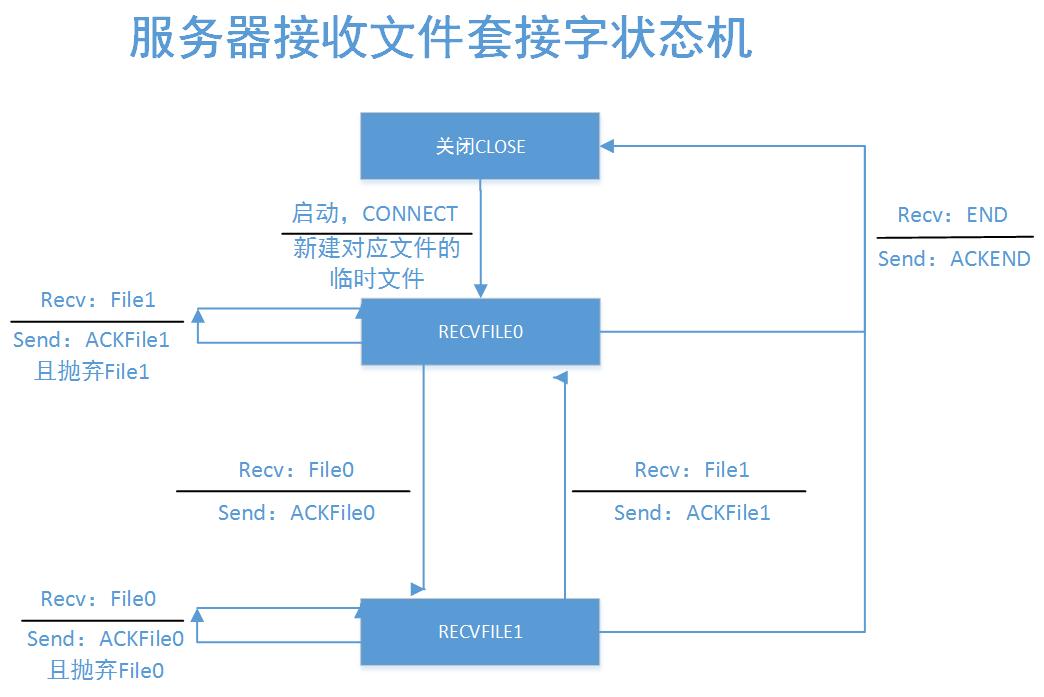
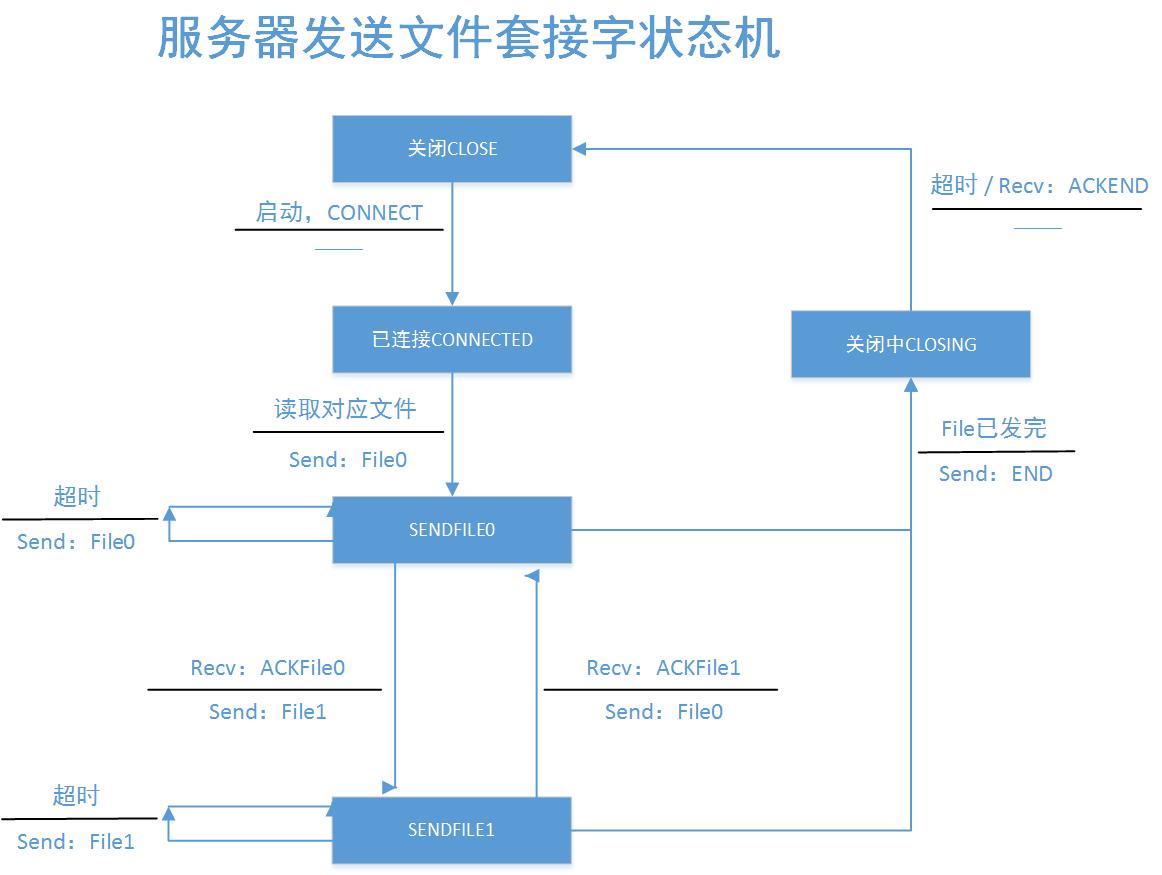
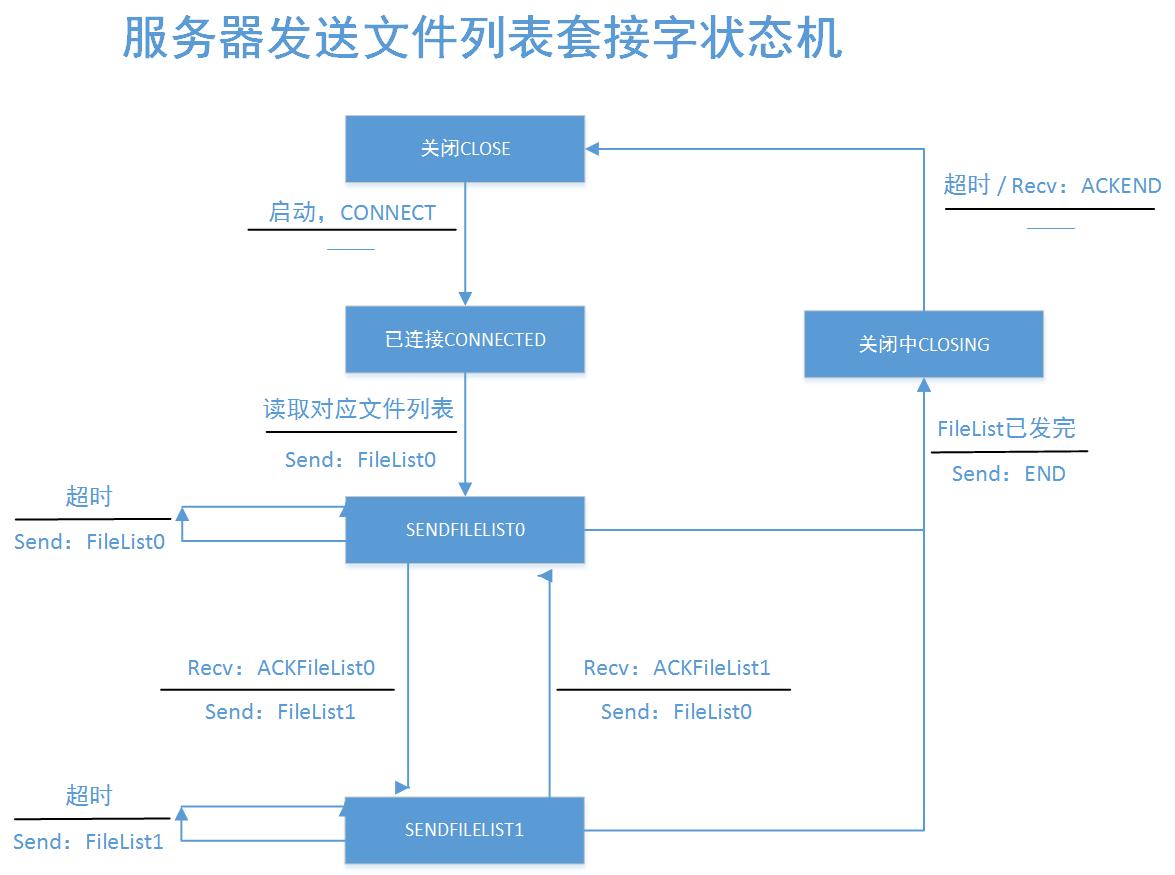
unsigned char len; //默认为报文长度总长度。当值为0时，表示传输结束

char buf[FILE\_BUF\_SIZE]; //载荷，具体包括文件列表各文件名或文件数据流

}FilePDU;

## 5.程序框架及流程设计

以下给出服务器在4种关键状态下的有限状态机。客户端的状态机可以对应推导出。



## 6.软件设计方案修订

6.1、修订原因

NULL

6.2、修订后的版本号

NULL