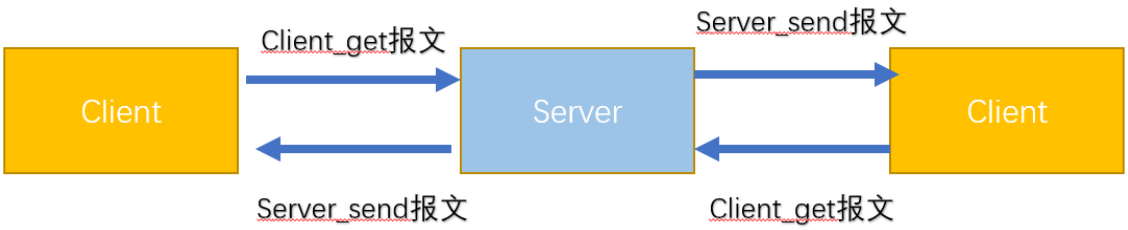
SP网络协议文档

一、SP概述。

SP （Simple Protocol）网络协议是一种能够支持简单文字对战游戏平台的 protocol(通讯协议)。是一种 client-server 协议，也就是说，请求通常是由像浏览器这样的接受方发起的。SP协议中通常包括了用户的状态、名字、行为等信息，同时预留了许多空间留做后续的扩展。

二、SP协议的组件系统。



上面就是SP协议使用的基本架构系统，报文直接传递在Client与Server之间，没有使用Proxy。每当client发送一个get报文，服务器就会返回一个send报文。系统运行的形式就是，服务器维护数据，客户登录、查询信息、对战都需经过服务器的处理和管理，它们之间的通信通过Client_get报文和Server_send报文交换信息。

客户端 User Agent:

user-agent 就是用户本身。只有Client才能发出Client_get报文，不能代理。

服务端 Web:

服务器永远不会主动向客户端发报文，只有在收到相应客户端的报文之后，服务器端才会做出相应的反应。即：Client-get报文与Server_send报文是一一对应的关系。

三、SP协议的报文格式定义。

下面我们分析一下两种报文：

1、Client-get报文：

报文的结构如下：

```
union Client_Buffet
{
  struct
  {
    char operation_number;  // explare how to resolve
    char user_name[10];     // client name (use for login )
    char user_password[10]; // client password (use for login)
    char rival_name[10];    // look for rival's name
    char whether_accept;    // whether accept another challenge
    char choice;            // cloth ,sicssors, stone
    char un_use[95];        //
```

```
  } content;
  char characters[128];
};
```

报文的解析：

```
概述：报文的总长是128字节。
组成部分：
-----operation_number 操作码【客户端选择不同的操作码发给服务器端，服务器端根据操作码做出相应的行为【详情在下面】】
-----user_name  发送者的名字，对于client来说谁发送就写谁，
-----user_password 用户的密码（可选）
-----rival_name  对方的名字，主要用于返回与你对战的对方的名字
-----whether_accept 用于是否接受对方邀请参与文字游戏
-----choice  在文字游戏中，用户做出的选择
-----un_use  预留部分可以供以后拓展功能使用
```

附：

operation_number 操作码:

操作码	数值	意义
LOGIN_TEST	0	登录验证，检查重名
ASK_MAIN_INFORMATION	1	查询用户主要信息
LOOK_FOR_RIVALS	2	查询其他用户状态
CLIENT_EXIT	3	用户退出（广播）
CLIENT_LOGIN	4	用户登入（广播）
SEARCH_BATTLE	5	询问对方是否可以接受挑战,与rival_name关联，表示询问的名字
RESPONSE_BATTLE	6	接受挑战/拒绝挑战，与whether_accept关联
FIGHTING_STATE	7	对战报文，choice 表示选择

whether_accept:

操作码	数值	意义
ACCEPT_FIGHT	1	对方同意加入游戏
REFUGE_FIGHT	0	对方拒绝加入游戏

choice:

操作吗	数值	意义
STONE	0	石头
SCISSORS	1	剪刀
CLOTH	2	布

2、Server-send报文。

```
union Server_Buffet
{
  struct
  {
    char operation_number;      // explare how to resolve
    char login_state;           // whether login successful
    char user_name[10];         // client name (use for login )
    char user_password[10];     // client password (use for login)
    char blood;                 // client blood
    char state;                 // client state
    char player_number;         // player's number (max 4)
    char members[11 * 4];       // player's (name ,state)
    char find_rival_error_code; // name not exit or  find myself to battle 0
means normal
    char peer_name[10];         // require another peer
    char user_name_choice;
    char peer_name_choice;
    char win_state;  // 0 means equal ,1 means win once ,2 means final win
    char counter_part_dump;
    char no_use[44]; // leave to use
  } content;
  char characters[128];
};
```

报文的解析:

```
-----operation_number;  操作码【客户端选择不同的操作码发给服务器端，服务器端根据操作码做出相
应的行为【详情在下面】】
-----login_state;   告知登录状态
-----user_name[10];  要发给的用户的用户名称
-----user_password[10];   密码（可选）
```

```
-----blood;          血量，在对战过程中实时显示对方血量，也可用于返回个人信息时返回体力信息

-----state;          用户的状态，用在查询个人信息时返回自己的状态
-----player_number;  目前平台在线人数，用于查询对手信息
-----members[11 * 4];  成员数组，每11为构成一个client，前10位时名字，后一位是状态 （目前能够容纳4人）
-----find_rival_error_code;    用名字查询对手时返回的错误处理码【详情见下】
-----peer_name[10];    用在对战确认阶段，向一个用户展示谁向他发起了挑战，存储发起挑战者的名字

-----user_name_choice; 下面两个用于向用户显示，每一轮双方的选择结果，及时反馈对战信息。
-----peer_name_choice; //数值同client-get的choice
-----win_state;  获胜状态【详情见下】
-----counter_part_dump 判断是否是对战对方dump，如果是回到空闲状态
-----no_use[44]; 预留空间
```

login_state:

操作码	数值	意义
LOGIN_STATE_FAIL	0	登陆失败
LOGIN_STATE_SUCCESS	1	登陆成功
LOGIN_PLAYER_FULL	2	人员已满

state:

操作码	数值	意义
LOGIN_IN	0	正在登陆
BASE_UI	1	空闲
READY_BATTLE	2	准备对战
COMBATING	3	正在对战

find_rival_error_code:

操作码	数值	意义
USER_AGREE_BATTLE	0	同意对战
USER_NOT_EXIT	1	用户不存在
USER_NOT_FREE	2	用户不空闲
PEER_REFUSED	3	对方拒绝

win_state:

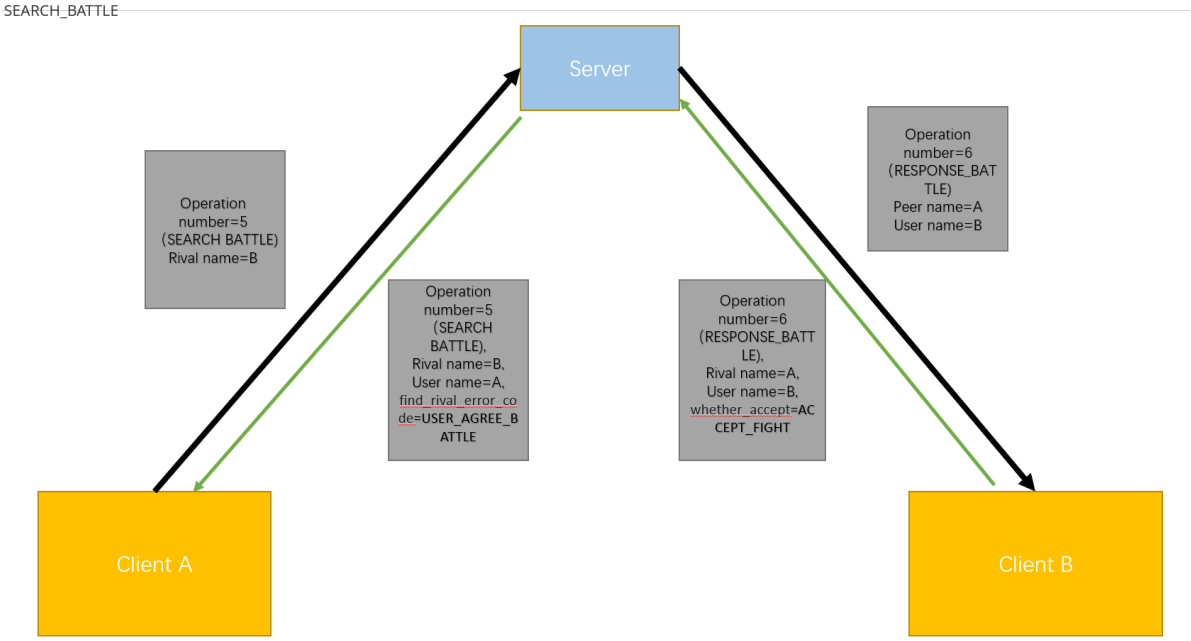
操作码	数值	意义
EQUAL	0	平局
WINONCE	1	本轮赢了
LOSEONCE	2	本轮输了
FINALWIN	3	最终赢了
FINALLOSE	4	最终输了

counter_part_dump：

操作码	数值	意义
IS_DUMP	1	对战对端挂了
NOT_DUMP	0	有人退出了，但是对战对方没挂

四、SP协议的运行实例。

本处邀请对方挑战，且对方同意为例展示运行细节：



三、SP协议的设计思路。

本协议秉承简单、易懂和可扩展的理念。没有设计复杂的依赖关系。考虑到客户端每一次请求都是独立的，即：连续两个client_get报文不会相互影响，连续两个Server_get报文也不会相互影响。所以设计出操作码，用户输入什么操作码，服务器就执行操作码对应的程序，将所需信息放入报文，传送给客户端。