游戏同步策略

# 概述

DashFire是一个射击类形的游戏，游戏中需要用到大量的弹幕和碰撞检测，并且游戏主要模式是PVP模式，类似Dota和LOL的竞技游戏，对游戏的实时性和准确性有很高的要求。

# 同步方案选择

## C/S结构

根据同事的调研，对实时性要求较高的游戏有大多数是用的P2P结构，如DNF使用的是P2P，公司内1018使用的也是P2P的结构。和我们的游戏模式比较类似的游戏如LOL没有用P2P模式同样达到了较好的效果。考虑到P2P技术我们没有太多的积累，可能有较多的未知因素，同时我们项目的时间较紧等以上这些现状和因素，我们准备使用C/S结构。

## 几种同步方案

### 本地客户端先行

如下图。这种模式有以下步骤：

1. C1客户端移动时在本地先做移动的位移或动作，并将消息发送到服务器
2. 服务器接收到消息模拟移动，并将消息发送给其它客户端
3. 远程客户端收到消息做移动或动作

这种模式是最粗糙的同步，本地客户端，服务器和远程客户端位置不同。



### 动作确认模式

如下图。这种模式有以下步骤：

1. C1客户端移动前先将消息发送到服务器，不作移动或动作
2. 服务器接收到消息模拟移动，并将消息发送给其它客户端
3. 本地客户端和远程客户端收到消息做移动或动作

这种模式的好处是C1和C2的动作是同步的。不足是客户端和服务器端并不是完全同步的，服务器端的动作或移动先于客户端，第二是延迟较大情况下本地客户端体验不好。



### 服务器和远程客户端预测

如下图。这种模式有以下步骤：

1. 客户端和服务器同步时间
2. C1客户端移动时先将带时间戳的消息发送到服务器，同时做移动或动作
3. 服务器接收到消息做预测模拟移动，并将消息发送给其它客户端
4. 远程客户端收到消息做预测模拟移动

这种模式的好处是本地客户端体验较好，C1和C2和服务器是同步的。不足是消息发送到服务器和远程客户端之前，服务器和远程客户端的位置或动作和地本不一致。由于步骤1同步时间会有一定的不准确，这种方式到客户端C2后会放大这种不准确性。



### 服务器加时间戳先行

如下图。这种模式有以下步骤：

1. 客户端和服务器同步时间
2. C1客户端移动时先将消息发送到服务器
3. 服务器接收到消息做模拟移动，并将消息加上服务器时间戳发送给其它客户端
4. 本地客户端和远程客户端收到消息做预测模拟移动

这种模式的好处是C1和C2和服务器是同步的并且因为时间的不准确性的影响较小， 服务器不需要做模拟，本地客户端和远程客户端动作时间差很小并且取决于各自的延迟。延迟高时本地客户端体验较差。



## 同步方案

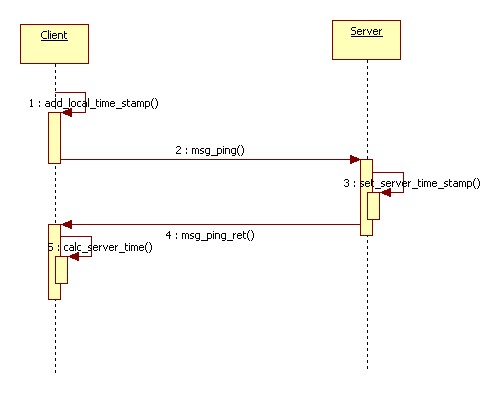
考虑到我们游戏是类DOTA游戏，本身对玩家的网络延迟较高，同时因为是竞技游戏，对游戏的准确性更高，针对以上四种基本的同步方式，第4种方式较适合当前游戏和结构。

# 移动同步

当前游戏是采用W、A、S、D控制移动，鼠标移动和点击控制方向的模式。移动时同步主要是同步按键消息，方向的控制由于是鼠标移动的方式，鼠标位置时刻改变并使用玩家的朝向时刻改变，不适合改变即同步的方案，现采用一定频率的方式进行同步。

## 时间同步

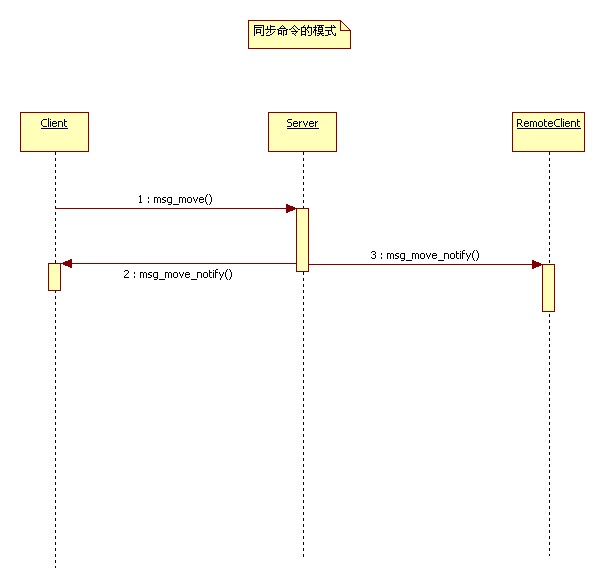
下图为时间同步的时序图：



其中步骤2和步骤4为网络传输时间，这两个时间的不确定性导致计算出的服务器时间是不准确的。可以考虑参考NTP(Network Time Protocal)协议的算法。同步的效果很多程序上依赖于此。时间校对有待进一步优化。

## 按键同步

下图为按键同步：



客户端按键发生改变时会将消息发送到服务器，服务器开始模拟移动并加上服务器端的时间戳并发送到客户端，本地客户端和远程客户端接收到消息后开始移动。因为客户端移动会落后于服务器，所以客户端要作一定的预测，这里客户端的预测关系到用户的体验，需要进一步的测试和优化。

## 方向同步

游戏中玩家的方向采用一定频率发送的方式。经测试，以10/1s的速度发送效果会好一些，低于这个频率效果较差，高于这个频率需要考虑同步的消息量。