基于时间戳的消息机制

|  |  |  |
| --- | --- | --- |
| 时间 | 作者 | 说明 |
|  |  | 创建文档 |

# 概述

本文主要描述了因dashfire需求基于时间戳的消息同步，参考RakNet的时间戳的实现提出我们基本的思路和实现案。

# RakNet参考

根据1018的介绍，RakNet内部支持了Ping和带时间戳的消息，以下做了一些简单的分析。

## Ping

RackNet提供两种Ping功能，一种是内部自动进行的Ping功能，它用来支持带timestamp的消息等其它功能，一种是提供对外的Ping的API。内部自动进行的Ping每隔一定的时间它会发送一次消息计算Ping值。每次得到的Ping值会记录下来以便后边根据一定的算法得到较准确的Ping值。

Ping值计算受消息的接收和发送过程和网络传输的影响，为了尽量降低消息接收和发送的时间的影响，Ping消息一般需要立即发送和接收到后立即处理。

Ping功能的准确性影响到消息中时间戳的准确性，据说RakNet的消息时间戳精度是比较高的，没有实测。

## timestamp

RackNet的消息可以带上时间戳。它的实现主要是建立在Ping的基础上。RakNet内部保存了网络的Ping值。

A: 1000

B: 3000

MsgX:1000|xxxx

MsgX:1000|xxxx

Ping: 5

MsgX:3000|xxxx

A在1000的时刻时B的时刻为3000。A发送带时间戳的消息时会加入本地的时间戳。B接收到该消息后会根据消息的时间和当前Ping值算出A发送消息的本地的相对时间，并将该消息的时间戳替换为本地的相对时间。

# DashFire

## 目的

使用带时间戳的消息的目的主要是: 为了使多个客户端在不同的延迟的情况下表现一致，具体的是为了预测的实现，动作的加速，并保证客户端逻辑的正确性。

## 流程

根据综合情况，Dashfire主要是在服务器加时间戳，客户端根据服务器时间戳和Ping值情况推算服务器时间。主要流程如下图：

Server

ClientA

ClientB

1) MsgX

2) MsgX|timestamp:xxx

3) MsgX|timestamp:yyy

3)MsgX|timestamp:zzz

## 工作

当前的网络库已经提供了Ping的功能，还需要做的工作有：

1. 需要实现Ping值的收集和时间戳推算算法。该算法需要在IOManager层实现。
2. 实现客户端的各功能的预测或加速

## 风险

使用当前这种方式的风险：1. Ping值计算的精确性的风险。2. 增大客户端预测或加速等的逻辑复杂度。