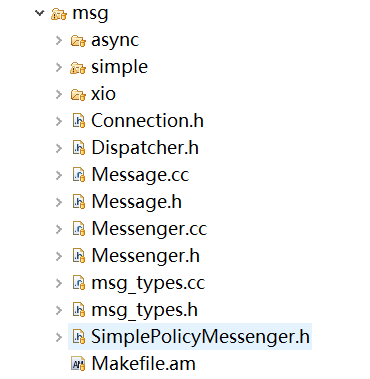
ceph作为分布式存储软件，其网络通信模块必不可少，是其关键的设计部分。Ceph有着自己的网络通信机制。本章分析ceph的网络通信实现。

先大概浏览下ceph的网络通信模块代码，集中在msg目录下面：

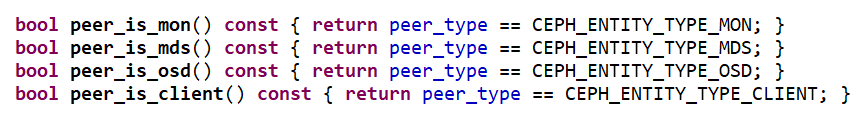


其中3个目录async,simple,xio分别是3种不同的通信模式。3种通信模式这里先不展开讨论

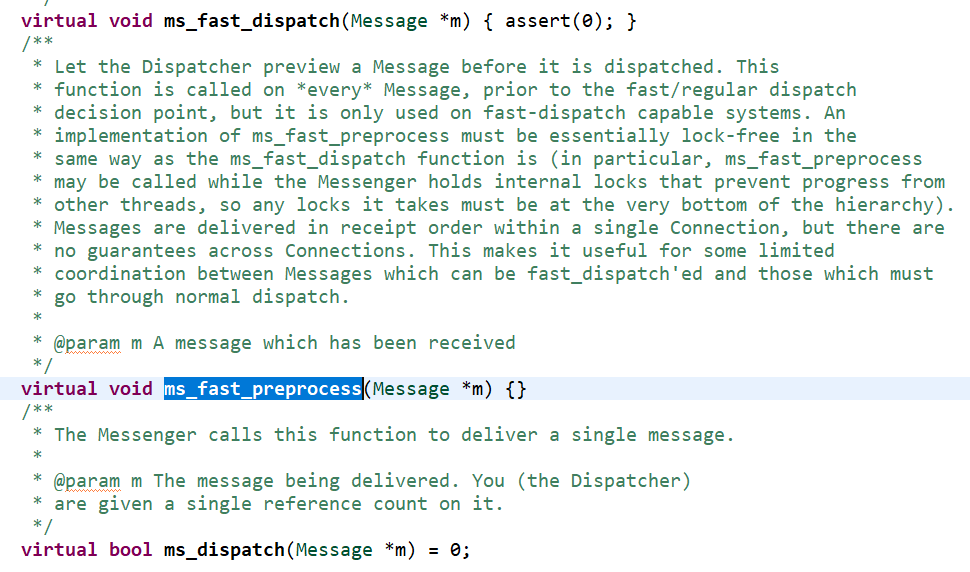
Connection.h主要定义一个网络连接以及与这次连接的相应操作。其中最主要的接口就是发送消息：

virtual int send\_message(Message \*m) = 0;

这个类是纯虚基类，需要子类继承实现。每个连接都属于不同的连接类型：

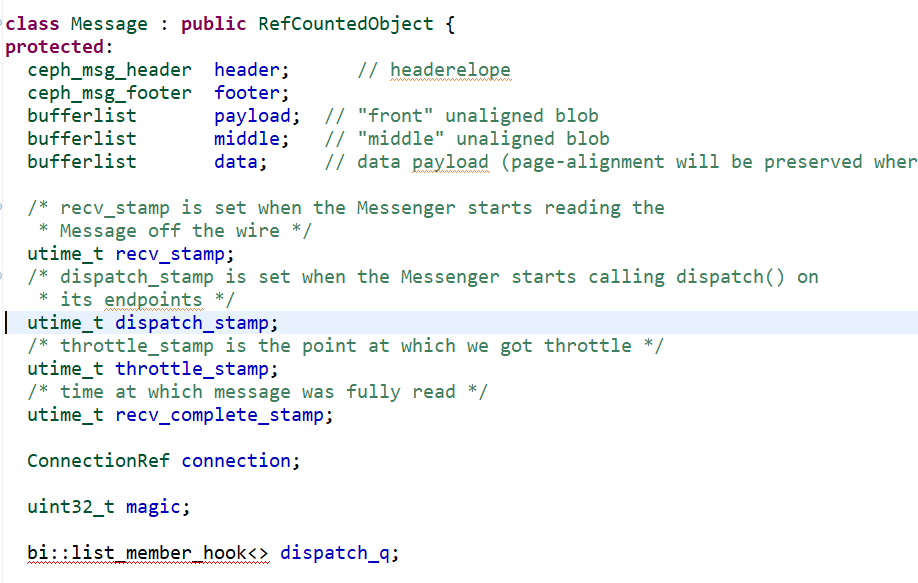
连接点是monitor，还是mds, 还是osd, 还是client, 不同的连接有不同的网络数据处理

Dispatcher.h主要定义网络消息的分发处理，主要是对接收到的网络消息分发给具体处理的应用层。类Dispatcher也是个纯虚基类，需要子类继承实现其功能。其中最主要的接口就是处理分发接收到的消息接口：



这3个接口都是消息的分发处理接口。是这个类的最主要的接口。其他的接口处理分发过程中的网络连接问题。

Message.h主要是定义网络数据结构体。类Message也是基类，任何要发送的消息，都要继承该类。



Ceph的各个网络模块通信都要设计符合message的数据格式：

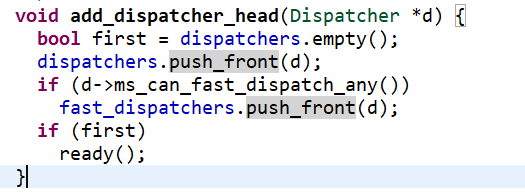
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| header | payload | middle | data | footer |

Messenger是整个网络抽象模块，定义网络模块的基本API接口。提供网络模块基本功能，能在节点之间发送和接收消息。

向一个节点发送消息的命令：

virtual int send\_message(Message \*m, const entity\_inst\_t& dest) = 0;

注册一个Dispatcher用来分发消息的命令如下：



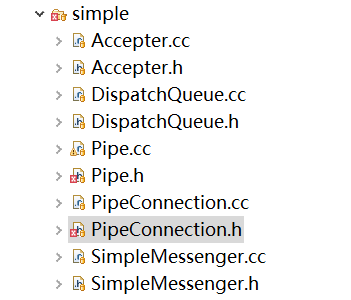
msg\_types.h定义网络连接的类型及其网络地址信息：



SimplePolicyMessenger.h主要设置一些策略，处理网络连接的策略。

上面说到了三种网络通信模式，这次专门讲解下simple模式

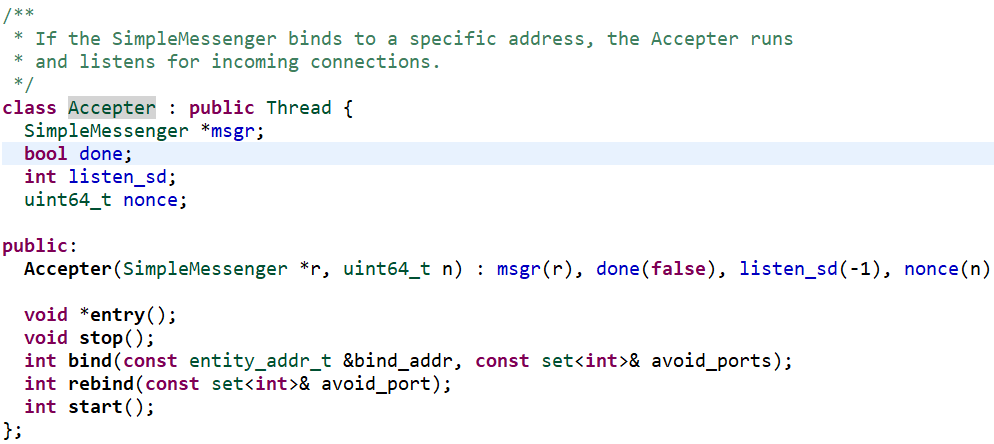
Simple代码结构如下：



同样的也大概分析了如下代码：

Accepter.h：主要是用来在Server端监听端口，接收连接，它继承了Thread类，本身是一个线程，不断的监听Server的端口：

类Accepter有如下接口：



函数entry()是函数的执行体，这个函数有个循环，不停的接收客户端的连接，并且将连接交给pipe处理。

stop()关闭连接，停止监听端口。

bind()接口建立socket监听

start()启动线程

上面粗略的描述了各个模块的基本功能。

下面整体梳理下网络架构：