报文队列：

收到完整报文后，求出处理队列号并把报文放入队尾。一个队列可能会被放入多个channelID的报文，较多的队列会使分配较均匀。

如果该队列长度为1，说明没有正在被处理的报文，直接处理该报文。

如果该队列长度大于1，说明有报文正在被处理，延缓处理刚刚收到的这条报文。

该队列的某个packet的callback或errback被回调时，检查队列是否为空，如果不为空就处理下条packet，否则返回eventloop。

假设处理某个packet的典型过程如下，

1)计算 2)访问Redis缓存 3)计算 4)访问Mysql数据库 5)计算 6)访问缓存7)计算8)访问数据库9)计算10)返回packet

据作者了解， Redis缓存及Mysql数据对于单一连接收到响应与请求发送的顺序对应。

Redis及Mysql访问会通过网络，这里使用twisted网络的异步访问。 这个过程中请求了多次数据库，应该使用同一连接，所以第4步的时候应该从ConnectionPool中取出一个数据库连接，在第9步的时候选择commit或rollback。

int hashVal =HashFunc(channelId);

int queueIdx =hashVal%4096;

if Len(packetQueue(queueIdx))==1:

HandlePacket();

NormalPacketQueue

Packet-Recved