

1. 长连接心跳机制

互联网推送消息的方式很常见，特别是移动互联网上，手机每天都能收到好多推送消息，经过研究发现，这些推送服务的原理都是维护一个长连接（要不不可能达到实时效果），但普通的 `socket` 连接对服务器的消耗太大了，所以才出现像 `MQTT` 这种轻量级低消耗的协议来维护长连接。

android 系统的推送和 IOS 的推送区别：

所有的推送功能必须有一个客户端和服务器的长连接，因为推送是由服务器主动向客户端发送消息，如果客户端和服务端之间不存在一个长连接那么服务器是无法来主动连接客户端的。因而推送功能都是基于长连接的基础上的。

IOS 长连接是由系统来维护的，也就是说苹果的 IOS 系统在系统级别维护了一个客户端和苹果服务器的长链接，IOS 上的所有应用上的推送都是先将消息推送到苹果的服务器然后将苹果服务器通过这个系统级别的长链接推送到手机终端上，这样的几个好处为：1.在手机终端始终只要维护一个长连接即可，而且由于这个长链接是系统级别的不会出现被杀死而无法推送的情况。2.省电，不会出现每个应用都各自维护一个自己的长连接。3.安全，只有在苹果注册的开发者才能够进行推送，等等。

android 的长连接是由每个应用各自维护的，但是 google 也推出了和苹果技术架构相似的推送框架，`C2DM`，云端推送功能，但是由于 google 的服务器不在中国境内推送无法使用，android 的开发者不得不自己去维护一个长链接，于是每个应用如果都 24 小时在线，那么都得各自维护一个长连接，这种电量和流量的消耗是可想而知的。虽然国内也出现了各种推送平台，但是都无法达到只维护一个长连接这种消耗的级别。

推送的实现方式：

- 一：客户端不断的查询服务器，检索新内容，也就是所谓的 `pull` 或者轮询方式
- 二：客户端和服务端之间维持一个 `TCP/IP` 长连接，服务器向客户端 `push`
- 三：服务器有新内容时，发送一条类似短信的信令给客户端，客户端收到后从服务器中下载新内容，也就是 `SMS` 的推送方式

苹果的推送系统和 `googleC2DM` 推送系统其实都是在系统级别维护一个 `TCP/IP` 长连接，都是基于第二种的方式进行推送的。第三种方式由于运营商没有免费开放

这种信令导致了这种推送在成本上是无法接受的，虽然这种推送的方式非常的稳定，高效和及时。

2. 数据压缩

在实际编程中，网络带宽的有效数据负载率是一个值得考虑的问题。特别地，对于移动客户端来说，网络资源往往并不是很丰富，为了尽可能地节省网络资源，往往需要尽可能地增加数据包的有效数据率。

一种非常简单的压缩方法是进程长度编码，这种方法使用数据及数据长度这样简单的编码代替同样的连续数据，这是无损数据压缩的一个实例。这种方法经常用于办公计算机以更好地利用磁盘空间、或者更好地利用计算机网络中的带宽。对于电子表格、文本、可执行文件等这样的符号数据来说，无损是一个非常关键的要求，因为除了一些有限的情况，大多数情况下即使是一个数据位的变化都是无法接受的。

对于视频和音频数据，只要不损失数据的重要部分一定程度的质量下降是可以接受的。通过利用人类感知系统的局限，能够大幅度的节约存储空间并且得到的结果质量与原始数据质量相比并没有明显的差别。这些有损数据压缩方法通常需要在压缩速度、压缩数据大小以及质量损失这三者之间进行折衷。

有损图像压缩用于数码相机中，大幅度地提高了存储能力，同时图像质量几乎没有降低。用于 DVD 的有损 MPEG-2 编解码视频压缩也实现了类似的功能。