> 心跳机制是定时发送一个自定义的结构体(心跳包)，让对方知道自己还活着，以确保连接的有效性的机制。

android系统的推送和iOS的推送有什么区别：

首先我们必须知道，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，如果客户端和服务器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是上的。

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，其他的原因你懂的。所以导致这个推送无法使用，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：

1. 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
2. 客户端和服务器之间维持一个TCP/IP长连接，服务器向客户端push
3. 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。
如果想了解android中各种推送方式请参考这个链接：Android实现推送方式解决方案 这篇博客已经介绍的非常好了。

所谓的心跳包就是客户端定时放送简单的信息给服务器端，告诉它我还在而已。代码就是每隔几分钟发送一个固定信息给服务器端，服务器端回复一个固定信息。如果服务器端几分钟后没有收到客户端信息则视客户端断开。比如有些通信软件长时间不适用，要想知道它的状态是在线还是离线，就需要心跳包，定时发包收包。

心跳包之所以叫心跳包是因为：它像心跳一样每隔固定时间发一次，以此来告诉服务器，这个客户端还活在。事实上这是为了保持长连接，至于这个包的内容，是没有什么特别规定的，不过一般都是很小的包，或者只包含包头的一个空包。

在TCP机制里面，本身是存在有心跳包机制的，也就是TCP选项:SO_KEEPALIVE. 系统默认是设置的2小时的心跳频率。

心跳包的机制，其实就是传统的长连接。或许有的人知道消息推送的机制，消息推送也是一种长连接，是将数据有服务器端推送到客户端这边从而改变传统的“拉”的请求方式。下面我来介绍一下安卓和客户端两个数据请求的方式

1. push 这个也就是有服务器推送到客户端这边 现在有第三方技术 比如极光推送。
2. pull 这种方式就是客户端向服务器发送请求数据（http请求）

实现轮询

原理

其原理在于在android端的程序中，让一个SERVICE一直跑在后台，在规定时间之内调用服务器接口进行数据获取。这里的原理很简单，当然实现起来也不难；然后，这个类之中肯定要做网络了数据请求，所以我们在Service中建立一个线程（因为在android系统中网络请求属于长时间操作，不能放主线程，不然会导致异常），在线程中和服务器进行通信。

最后，这个逻辑写完后，我们需要考虑一个问题，如何进行在规定时间内调用该服务器，当然可以用Thread+Handler(这个不是那么稳定),也可以使用AlamManager+Thread（比较稳定），因为我们需要其在后台一直运行，所以可以依靠系统的Alammanager这个类来实现，Alammanager是属于系统的一个闹钟提醒类，通过它我们能实现在规定间隔时间调用，并且也比较稳定，这个service被杀后会自己自动启动服务。