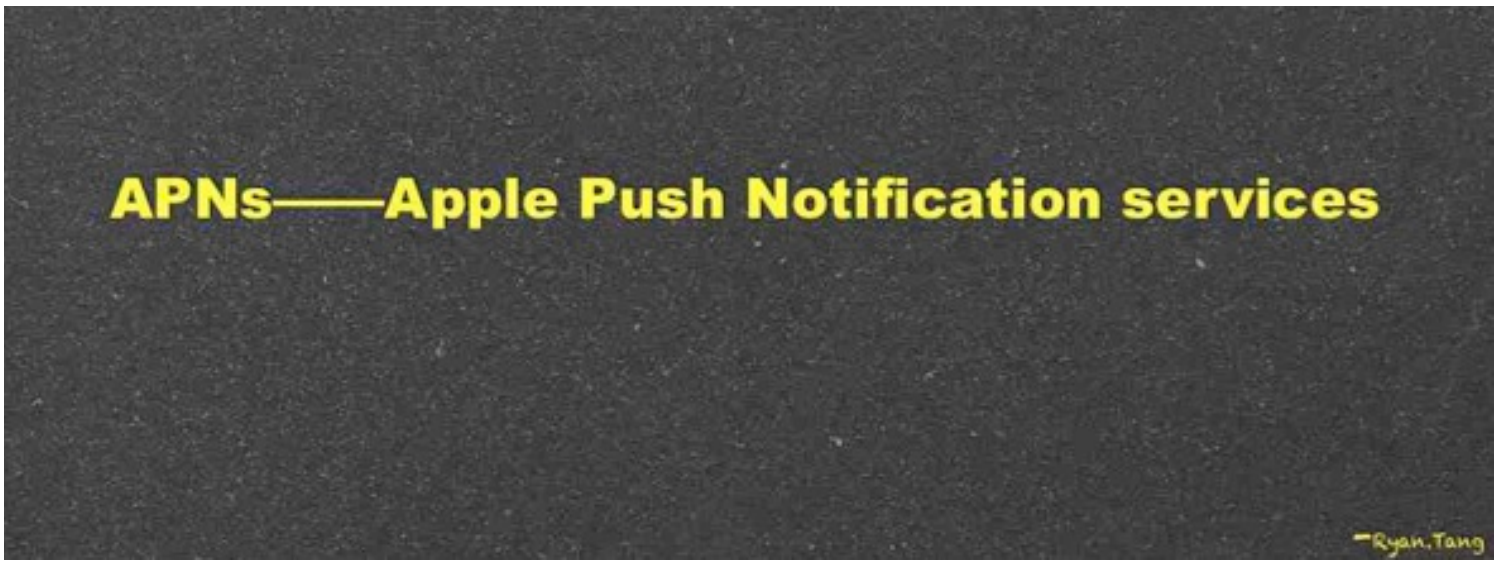
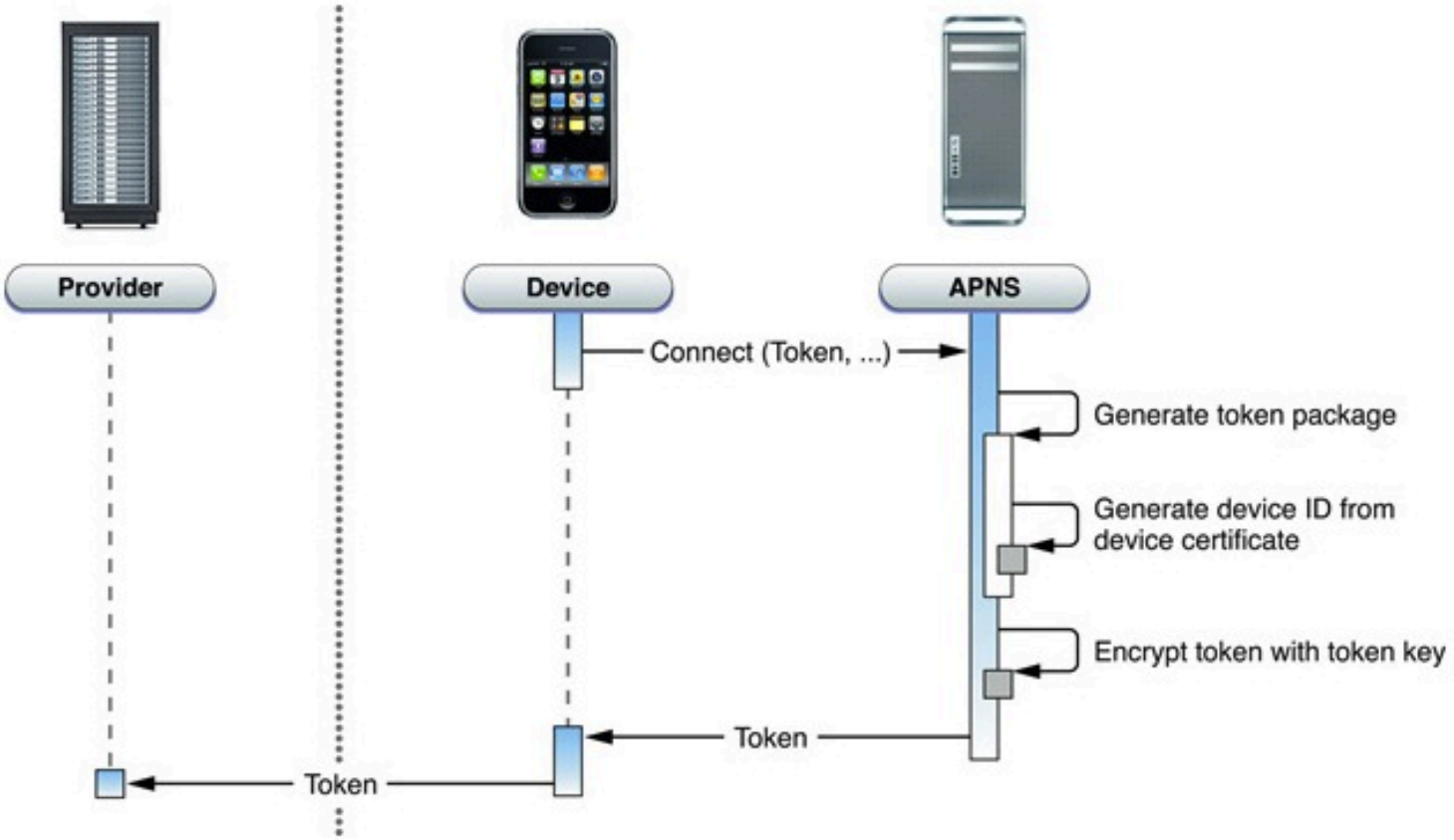
关于苹果 APNs 推送机制(1)

推送是解决轮询所造成的流量消耗和电量消耗的一个比较好的解决方案，在 Android 上，虽然 Google 提供了 GCM（之前为 C2DM），但在国内基本等于没用，各大 Android 应用基本都自己架设推送 Server 或是使用第三方推送平台，例如新浪微博使用第三方推送平台“个推”（非广告）。今天要学习的是苹果提供的推送服务 APNs（Apple Push Notification services）基本原理和工作流程。



苹果的推送服务 APNs 基本原理简单来说就是苹果利用自己专门的推送服务器（APNs）接收来自我们自己应用服务器的需要被推送的信息，然后推送到指定的 iOS 设备上，然后由设备通知到我们的应用程序，设备以通知或者声音的形式通知用户有新的消息。推送的前提是装有我们应用的设备需要向 APNs 服务器注册，注册成功后 APNs 服务器会返给我们一个 device_token，拿到这个 token 后我们将这个 token 发给我们自己的应用服务器，当有需要被推送的消息时，我们的应用服务器会将消息按指定的格式打包，然后结合设备的 device_token 一并发给 APNs 服务器，由于我们的应用和 APNs 维持一个基于 TCP 的长连接，APNs 将新消息推送到我们设备上，然后在屏幕上显示出新消息来。整个过程基本就这样，下面我们看一下设备注册 APNs 的流程图：



上图完成了如下步骤：

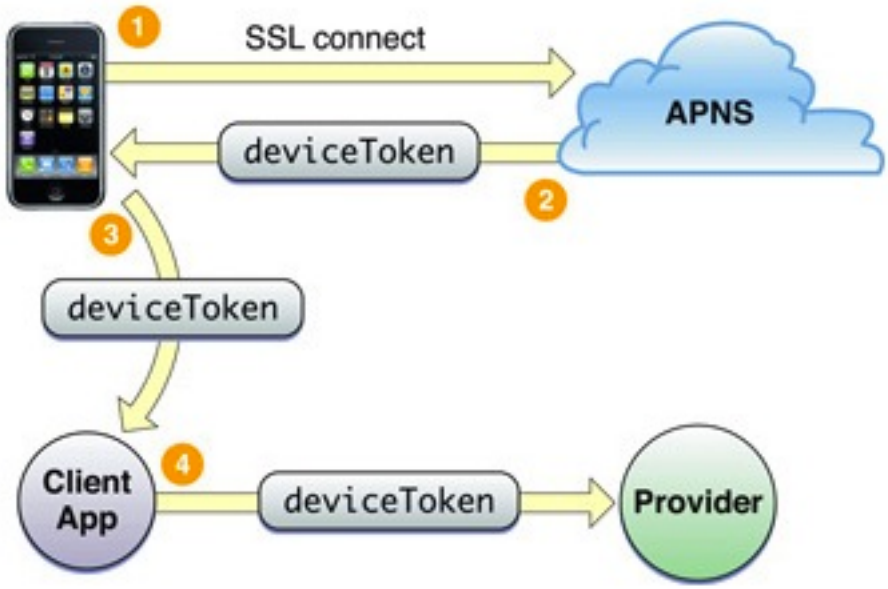
1.Device 连接 APNs 服务器并携带设备序列号

2.连接成功，APNs 经过打包和处理产生 device_token 并返回给注册的 Device

3.Device 携带获取的 device_token 向我们自己的应用服务器注册

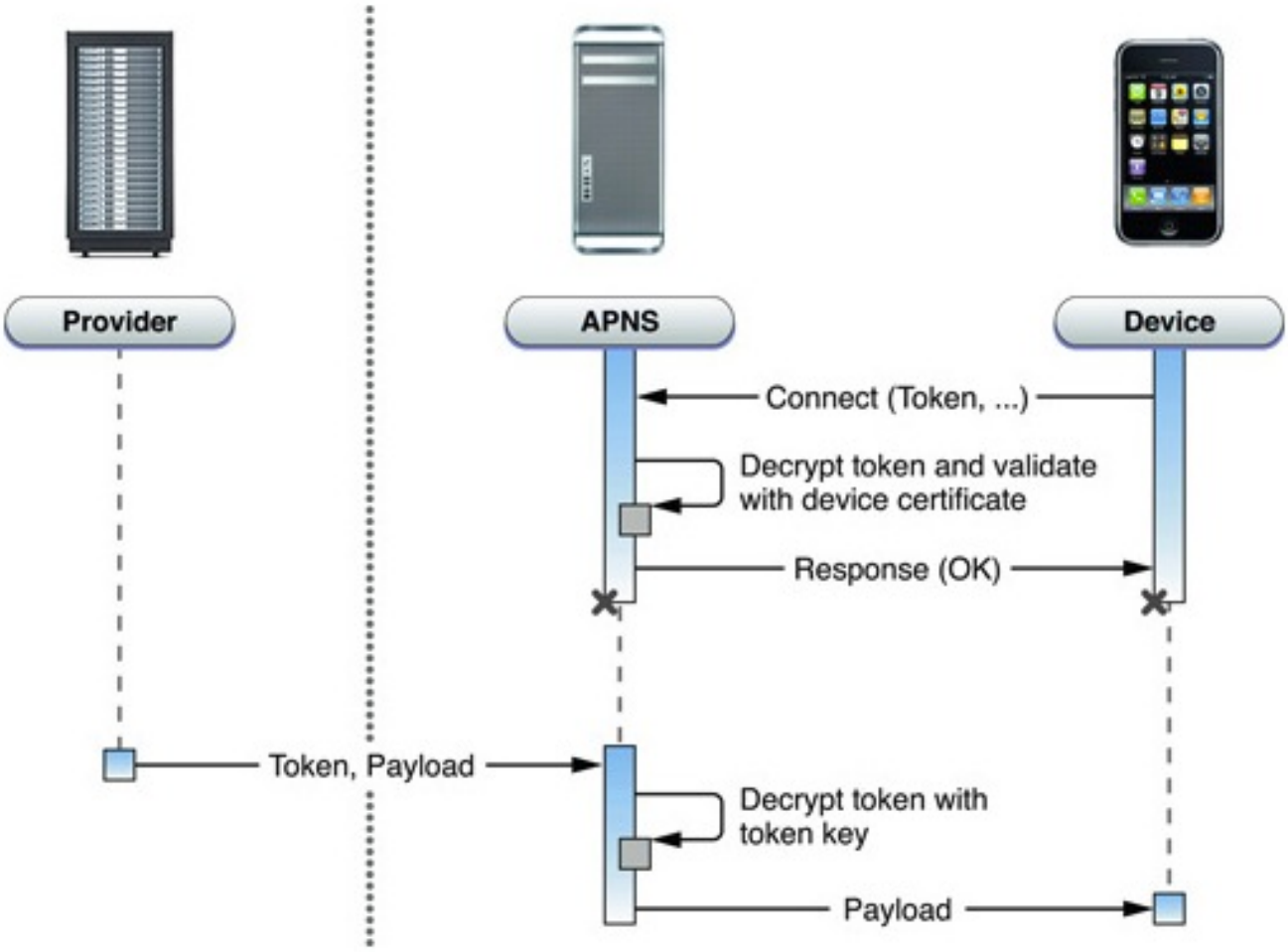
4.完成需要培推送的 Device 在 APNs 服务器和我们自己的应用服务器注册

执行顺序如下所示：



这里要提到的一点是，我们的设备和 APNS 服务器之间的通讯是基于 SSL 协议的 TCP 流通讯，二者之间维持一个长连接，当从 APNS 服务器注册成功后，一定要将 device_token 发送给我们的应用服务器，因为在推送过程中，首相是由我们的应用服务器（上图中 Provider）将需要推送的消息结合 device_token 按指定格式（后面会提到）打包然后发送给 APNS 服务器，然后由 APNS 服务器推送给我们的设备。

好了，注册设备的过程完成了，接下来就是如何推送了：



推送的过程经过如下步骤：

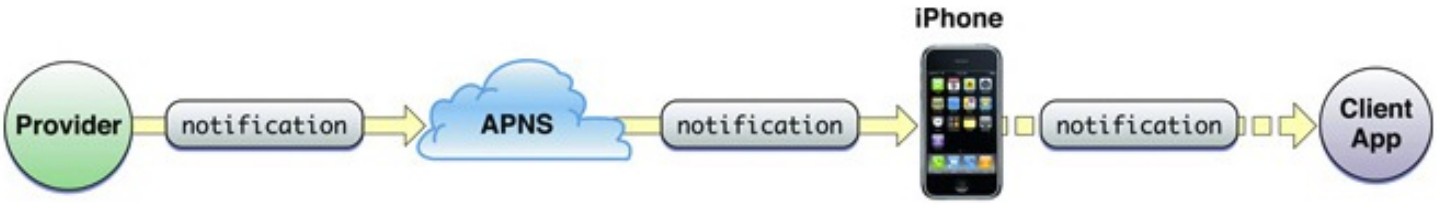
1.首先，安装了具有推送功能的应用，我们的设备在有网络的情况下会连接苹果推送服务器，连接过程中，APNS 会验证 device_token，连接成功后维持一个长连接；

2.Provider(我们自己的服务器)收到需要被推送的消息并结合被推送设备的 device_token 一起打包发送给 APNS 服务器；

3.APNS 服务器将推送信息推送给指定 device_token 的设备；

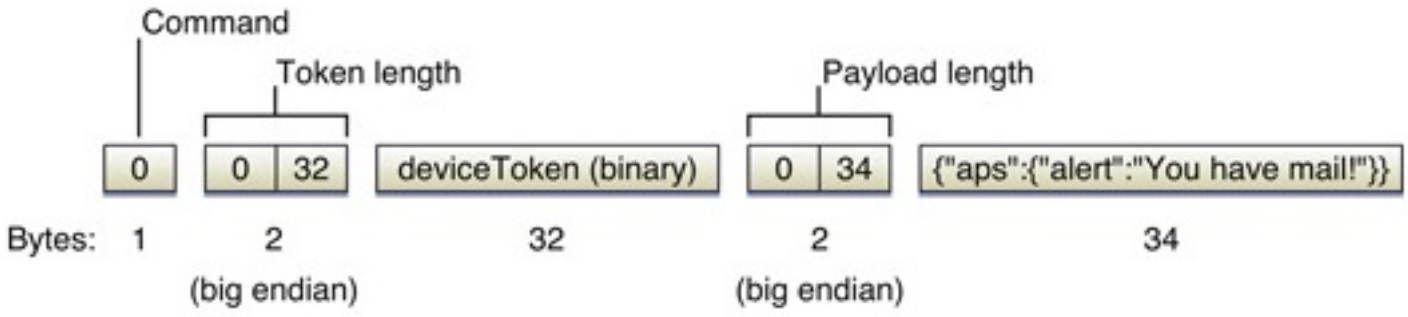
4.设备收到推送消息后通知我们的应用程序并显示和提示用户（声音、弹出框）

比较直观的流程参照下图：



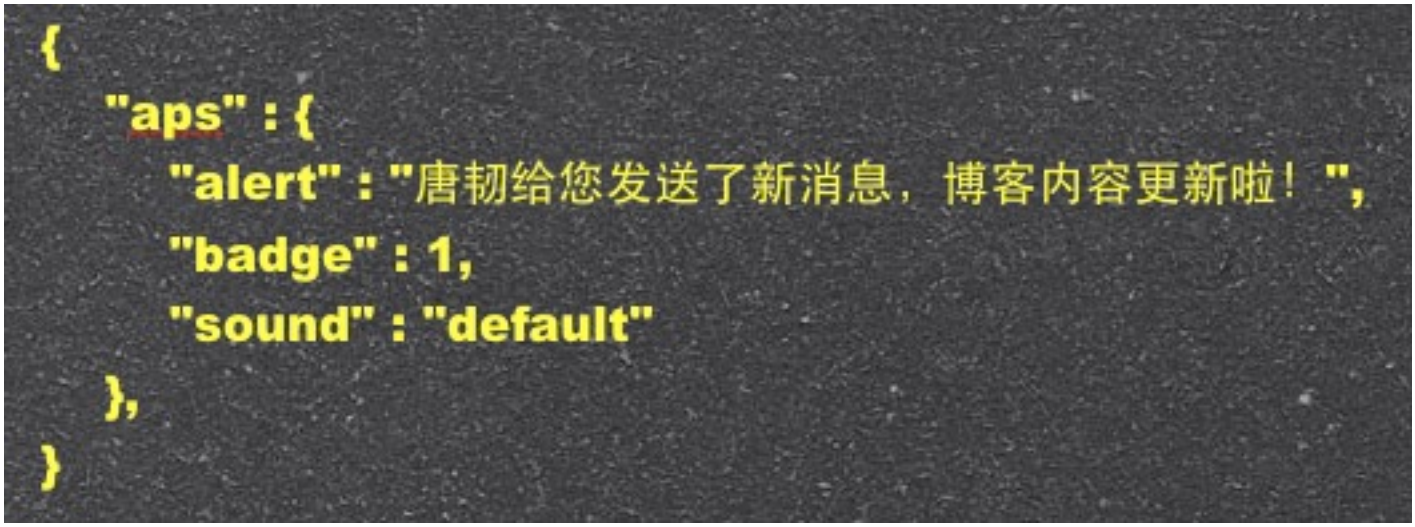
上图显示了我们的应用服务器将消息推送到我们的 App 的完整路径，其实真正完成推送的是 APNS 服务器，我们自己的应用服务器只是将需要推送的消息告诉苹果服务器，至于如何维护消息队列或如何保证消息能被推送到指定的设备上，这些都由苹果 APNS 给我们做完了。

上面提到了将 device_token 和推送消息打包的过程，那么，接下来就看看这个信息包结构是怎样的：



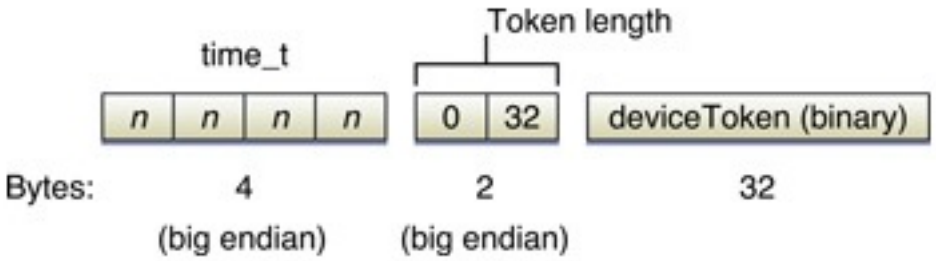
上图显示的这个消息体就是我们的服务器(Provider)发送给 APNS 服务器的消息结构，APNS 验证这个结构正确并提取其中的信息后，再将消息推送到指定的设备。这个结构

体包括五个部分，第一个部分是命令标示符，第二个部分是我们的 device_token 的长度，第三部分是我们的 device_token 字符串，第四部分是推送消息体（Payload）的长度，最后一部分也就是真正的消息内容了，里面包含了推送消息的基本信息，比如消息内容，应用 Icon 右上角显示多少数字以及推送消息到达时所播放的声音等。接下来我们拆解看一下 Payload（消息体）的结构：



这其实就是个 JSON 结构体，alert 标签的内容就是会显示在用户手机上的推送信息，badge 显示的数量（注意是整型）是会在应用 Icon 右上角显示的数量，提示有多少条未读消息等，sound 就是当推送信息送达是手机播放的声音，传 defalut 就标明使用系统默认声音，如果传比如“beep.wav”就会播放在我们应用工程目录下名称为 beep.wav 的音频文件，比如当手机锁屏时 QQ 在后台收到新消息时的滴滴声。

有这么一种情况，当我们将应用从设备卸载后，推送的消息改如何处理呢。我们知道，当我们将应用从设备卸载后，我们是收不到 Provider 给我们推送的消息的，但是，如何让 APNS 和 Provider 都知道不去向这台卸载了应用的设备推送消息呢？针对这个问题，苹果也已经帮我们解决了，那就是 Feedback service。他是 APNS 的一部分，APNS 会持续的更新 Feedback service 的列表，当我们的 Provider 将信息发给 APNS 推送到我们的设备时，如果这时设备无法将消息推送到指定的应用，就会向 APNS 服务器报告一个反馈信息，而这个信息就记录在 feedback service 中。按照这种方式，Provider 应该定时的去检测 Feedback service 的列表，然后删除在自己数据库中记录的存在于反馈列表中的 device_token，从而不再向这些设备发送推送信息。连接 Feedback service 的过程同样使用 Socket 的方式，连接上后，直接接收由 APNS 传输给我们的反馈列表，传输完成后断开连接，然后我们根据这个最新的反馈列表在更新我们自己的数据库，删除那些不再需要推送信息的设备的 device_token。从 Feedback service 读取的数据结构如下：



结构中包含三个部分，第一部分是一个时间戳，记录的是设备失效后的时间信息，第二个部分是 device_token 的长度，第三部分就是失效的 device_token，我们所要获取的就是第三部分，跟我们的数据库进行对比后，删除对应的 device_token，下次不再向这些设备发送推送信息。