# 极光推送技术原理：移动无线网络长连接

## 移动互联网应用现状

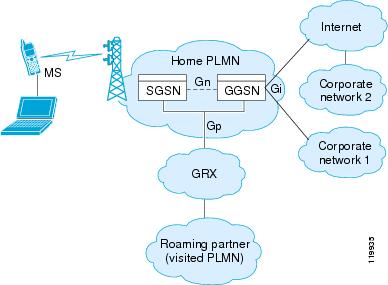
因为手机平台本身、电量、网络流量的限制，移动互联网应用在设计上跟传统 PC 上的应用很大不一样，需要根据手机本身的特点，尽量的节省电量和流量，同时又要尽可能的保证数据能及时到达客户端。

为了解决数据同步的问题，在手机平台上，常用的方法有2种。一种是定时去服务器上查询数据，也叫Polling，还有一种手机跟服务器之间维护一个 TCP 长连接，当服务器有数据时，实时推送到客户端，也就是我们说的 Push。

从耗费的电量、流量和数据送达的及时性来说，Push 都会有明显的优势，但 Push 的实现和维护成本相对较高。在移动无线网络下维护长连接，相对也有一些技术上的难度。本文试图给大家介绍一下我们[极光推送](http://www.jpush.cn/)在 Android 平台上是如何维护长连接。

## 移动无线网络的特点

因为 IP v4 的 IP 量有限，运营商分配给手机终端的 IP 是运营商内网的 IP，手机要连接 Internet，就需要通过运营商的网关做一个网络地址转换（Network Address Translation，NAT）。简单的说运营商的网关需要维护一个外网 IP、端口到内网 IP、端口的对应关系，以确保内网的手机可以跟 Internet 的服务器通讯。



http://www.cisco.com/en/US/i/100001-200000/110001-120000/119001-120000/119935.jpg

图片源自 cisco.com.

NAT 功能由图中的 GGSN 模块实现。

大部分移动无线网络运营商都在链路一段时间没有数据通讯时，会淘汰 NAT 表中的对应项，造成链路中断。

## Android 平台上长连接的实现

为了不让 NAT 表失效，我们需要定时的发心跳，以刷新 NAT 表项，避免被淘汰。

Android 上定时运行任务常用的方法有2种，一种方法用 Timer，另一种是AlarmManager。

### Timer

Android 的 Timer 类可以用来计划需要循环执行的任务，Timer 的问题是它需要用 WakeLock 让 CPU 保持唤醒状态，这样会大量消耗手机电量，大大减短手机待机时间。这种方式不能满足我们的需求。

### AlarmManager

AlarmManager 是 Android 系统封装的用于管理 RTC 的模块，RTC (Real Time Clock) 是一个独立的硬件时钟，可以在 CPU 休眠时正常运行，在预设的时间到达时，通过中断唤醒 CPU。

这意味着，如果我们用 AlarmManager 来定时执行任务，CPU 可以正常的休眠，只有在需要运行任务时醒来一段很短的时间。极光推送的 Android SDK 就是基于这种技术实现的。

## 服务器设计

当有大量的手机终端需要与服务器维持长连接时，对服务器的设计会是一个很大的挑战。

假设一台服务器维护10万个长连接，当有1000万用户量时，需要有多达100台的服务器来维护这些用户的长连接，这里还不算用于做备份的服务器，这将会是一个巨大的成本问题。那就需要我们尽可能提高单台服务器接入用户的量，也就是业界已经讨论很久了的 C10K 问题。

### C2000K

针对这个问题，我们专门成立了一个项目，命名为C2000K，顾名思义，我们的目标是单机维持200万个长连接。最终我们采用了多消息循环、异步非阻塞的模型，在一台双核、24G内存的服务器上，实现峰值维持超过300万个长连接。

## 后记

稳定维护长连接是推送平台的一个基础，[极光推送团队](http://blog.jpush.cn/)将会在这方面长期投入，以保证用户能有效的节省电量、流量，同时数据能实时送达。

# 即时通信技术-Websocket

以前不管使用HTTP轮询或使用TCP长连接等方式制作在线聊天系统，都有天然缺陷，随着Html5的兴起，其中有一个新的协议WebSocket protocol，可实现浏览器与服务器全双工通信(full-duplex)，它可以做到：浏览器和服务器只需要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。这个新的协议的特点正好适合这种在线即时通信。