

基于 Android 系统终端实现 3G WIFI 无线漫游

摘要:近年来 3G 网络得到迅速发展,其用户量也迅速增长。3G 网络覆盖范围广,而 WIFI 网络带宽高。因此,实现 3G 与 WIFI 网络的互通将具有十分重要的意义。分析了基于 Android 系统手机终端的 3G 网络和 WIFI 网络的连接机制,根据此连接机制针对当前兴起的 Android 系统应用程序的开发进行了分析,实现了由 3G 到 WIFI 的无线漫游,该漫游主要是通过 Android 中的 ListView 数据显示 adapter 以及 SharedPreferences 实现了连接节点 SSID 的显示及存储。此 WIFI 节点漫游节省了用户使用的 3G 流量,可以使用户充分利用当前兴起的无线网络资源。

随着人们日常生活和工作节奏的不断加快,人们对信息获取的便捷程度、对移动办公、移动娱乐的重视程度正与日俱增。无线接入是满足这一高速增长需求的唯一途径,无线接入的两个主角是 3G(3rdGeneratiON)技术和 W1FI(Wireless Fidelity)技术。

本论文针对目前比较流行的 Android 系统手机终端的 3G、WIFI 连接机制实现了 3G 和 WIFI 的无线漫游。

1 Android 平台介绍

Android 是 Google 公司最新推出的一个手机平台,是一个真正意义上的开放性移动设备综合平台。它包括操作系统、用户界面和应用程序及移动电话工作所需的全部软件,而且不存在任何以往阻碍移动产业创新的专有权障碍。Android 是由 Unix+Java 构成的。它的架构和工作原理决定了 Android 即目前最具可移植性的手机操作系统。Android 是开源软件,能够有效降低软件的成本,从而降低手机的整体价格,最终让每个用户能够自由的获取信息。Android 提供了一个从操作系统到应用程序的完整的软件栈,同时又允许人们定制差异化,以形成自己的竞争优势,这对于手机制造商来说,具有强大的吸引力。开源软件带来的另一个好处是允许厂商自主解决问题。在此我们借助 android 系统的开源性及可定制的特性在此基础上增加了 3G、WIFI 的无线漫游机制。

2 WIFI 漫游中 ListView 的作用

Activity 是构成应用程序中最重要的构造块,在 Android 应用程序中,几乎都是通过 Activity 来与用户进行交互,所以 Activity 主要负责的就是创建与用户进行交互的显示窗口,你可以在这些窗口里通过调用 setContentView(View)方法来显示你自己的用户界面。在实际开发应用程序的过程中通常需要用到列表显示功能,那么在 Android 应用中,我们是通过 Android 中的一个特殊组件 ListView 来进行列表展示,它以列表的形式展示具体内容,并且能够根据数据的长度自适应显示。列表的显示需要 3 个元素。

1. ListView。用来展示列表的 View。
2. 适配器。用来把数据映射到 ListView 上的中介。
3. 数据。具体的将被映射的字符串、图片、或者基本组件。

ListView 绘制的过程如下：首先，系统在绘制 ListView 之前，将会先调用 getCount 方法来获取 Item 的个数。之后每绘制一个 Item 就会调用一次 getView 方法，在此方法内就可以引用事先定义好的 xml 来确定显示的效果并返回一个 View 对象作为一个 Item 显示出来。也正是在这个过程中完成了适配器的主要转换功能，把数据和资源以开发者想要的效果显示出来。也正是 getView 的重复调用，使得 ListView 的使用更为简单和灵活。这两个方法是自定 ListView 显示效果中最为重要的，同时只要重写好了前两个方法，ListView 就能完全按开发者的要求显示。而 getItem 和 getItemId 方法将会在调用 ListView 的响应方法的时候被调用到。所以要保证 ListView 的各个方法有效的話，这两个方法也得重写。ListView 作为一个数据显示的机制，在用户添加完 SSID 节点可以通过这种机制显示出来，同时还添加了删除，修改按钮及 SSID 序号。这些 SSID 节点作为 WiFi 漫游的搜索节点，按照用户的经常使用的级别定义优先顺序。

2.1 SharedPreferences 实现数据存储

Android 平台给我们提供了一个 SharedPreferences 类，它是一个轻量级的存储类，特别适用于保存软件配置参数。SharedPreferences 是以键值对来存储应用程序的配置信息的一种方式，它只能存储基本数据类型。一个程序的配置文件仅可以在本应用程序中使用，或者说只能在同一个包内使用，不能在不同的包之间使用。实际上 SharedPreferences 是采用了 XML 格式将数据存储到设备中，在 DDMS 中的 File Explorer 中的 / data / data / ~ package name > / shares prefs 目录下。

```
SharedPreferences ssidvalues = getSharedPreferences  
(SSID_INFOS, 0);
```

```
Editor editor = sharedPreferences.editor();获取编辑器  
WiFiConfigure.ssidvalues.editor ( ) .putString  
( WiFiConfigure.SSIDNAME, AccessPointDialog.ssid )  
.putLong(SSIDNUM, ++num); editor.commit();提交修改
```

生成的 SSID_Infos.xml 文件内容如下：

```
<? xml version = '1.0' encoding = 'utf-8' standalone =  
'yes' ? >  
<map><string name = " ADDRESS" >www.csdn.net  
</string>  
<long name = " num" value = " 0" /><string name = "  
SSID" >ssid1</string></map>
```

因为 sharedPreferences 是使用 xml 文件保存数据，getSharedPreferences(nAME, mode) 方法的第一个参数用于指定该文件的名称，名称不用带后缀，后缀会由 Android 自动加上。

如果希望 SharedPreferences 使用的 xml 文件能被其他应用读和写，可以指定 Context.MODE_WORLD_READABLE 和 Context.MODE_WORLD_WRITEABLE 权限。另外 Activity 还提供了另一个 getPreferences(mode)方法操作 SharedPreferences，这个方法默认使用当前类不带包名的类名作为文件的名称。访问 SharedPreferences 中的数据代码如下：

```
SharedPreferences sharedPreferences =  
getSharedPreferences (SSID_INFOS, Context.MODE_PRIVATE);  
String name = ssidvalues.getString(SSIDNAME, "");
```

Long num = ssidvalues.getLong(SSIDNUM, 0); 如果访问其他应用中的 Preference，前提条件是该 preference 创建时指定了 Context.MODE_WORLD_READABLE 或者 Context.MODE_WORLD_WRITEABLE 权限。在此实现中是将数据存储为用户自己定义的 SharedPreference 文件中，用户将 SSID 节点添加到用户定义的 SSIDINFO 文件中。

2.2 3G WIFI 无线漫游功能设计

在基于 android 系统手机中的 WIFI 和 3G 都有相应的设置选项，在 WIFI 设置关闭的情况下默认通过 3G 网络连接上网。用户会经常去一些有 WIFI 热点的地方，这些热点是给用户免费开放的，可以通过 WIFI 网络连接到互联网。这些经常用的 AP 节点就是用户设置的 SSID 节点。

WIFI AP 节点认证连接过程：

①搜索过程。终端进入 WIFI 后，首先搜索候选网络的 AP 信息，AP 搜索过程就是终端扫描信道获取 AP 信息的过程，也叫 AP 扫描过程。在此过程中，终端在某个频道上发出请求消息，然后等待 AP 回应，然后转到下一个频道。IEEE802.11 标准规定，AP 在收到终端发出的请求信息后，必须回复携带自身信息的包给终端，终端成功收到 AP 回复后，就能够确知 AP 的存在并获取妙的相关信息。

② 认证过程。终端在发起切换连接上 WIFI 一个新的 AP 时，需要通过 AP 的认证。在进入认证阶段后，AP 先发送一个待验证的内容片段给终端，终端解析此内容后将解析结果返回给 AP，再判断终端解析后的内容是否正确，如果正确则通过认证，否则失败。3 个子过程缺少一个过程都会导致此次认证失败，因此保证了 AP 认证机制的可靠性。尽管这 3 个过程比较复杂，在终端与 AP 之间产生了多次的交互，然而这些过程并不占用太多的时间。

③ 重关联过程。在经过认证后，需要与连接上的 AP 进行重关联，然后通过 AP 连接上网。在重关联过程中，终端先发送重关联请求帧给 AP，该帧包含了终端一些信息，如终端所支持的数据比特率及终端将要进行连接的 AP 在网络中的 SSID。在接收到请求后，AP 处理请求并为该终端建立一条关联 ID 号同时分配一部分存储空间来存储相关信息。然后 AP 向终

端发送重关联响应来通终端关联成功还是失败，同时包含 AP 建立重关联的相关信息。如果关联成功，则终端可以使用此 AP 联网并与其他已经联网的终端进行通信了。

④ 地址配置。终端在连接上 AP 后，AP 要给终端分配一个 IP 地址，使其通过 IP 地址连接上互联网。

以上步骤是 WiFi 在热点区域连接 AP 节点认证的过程。在 android 系统中 WiFi 如果设定为关闭即使是在有 WiFi 热点区域也不会进行搜索连接。为此我们添加了 WiFi 漫游过程。首先每隔一个固定间隔时间打开 WiFi 设置，查找是否有 WiFi 热点，如果有就进行以上的 WiFi 连接 AP 节点的过程，然后查找用户设定的 SSID 节点，查找到和用户设定的一致的 SSID 节点则进行网络连接。网络连接终端可以发起切换，在与 3G 网络断开后使用存储的信息与 WiFi 的 AP 建立连接，通过 WiFi 上网，从 3G 网络切换到 WiFi 网络连接到 Internet。

2.3 WiFi 漫游功能模块框图

为了能够在间隔固定时间查询是否有 WiFi 热点区域，特启动了一个漫游线程，该线程主要完成了如果已经在 WiFi 热点区域则打开 WiFi 设置，将 WiFi 的 AP 节点与用户设定的 SSID 进行比较如果为设定的 SSID 节点则进行连接，否则即使在热点区域仍然用 3G 网络连接。

在 3G 连接的情况下实现探测 WiFi 节点流程图，如果探测到的 SSID 在设定的 list 列表中，将 3G 关闭并自动连接到设定的 SSID，实现流程图如图 1 所示。

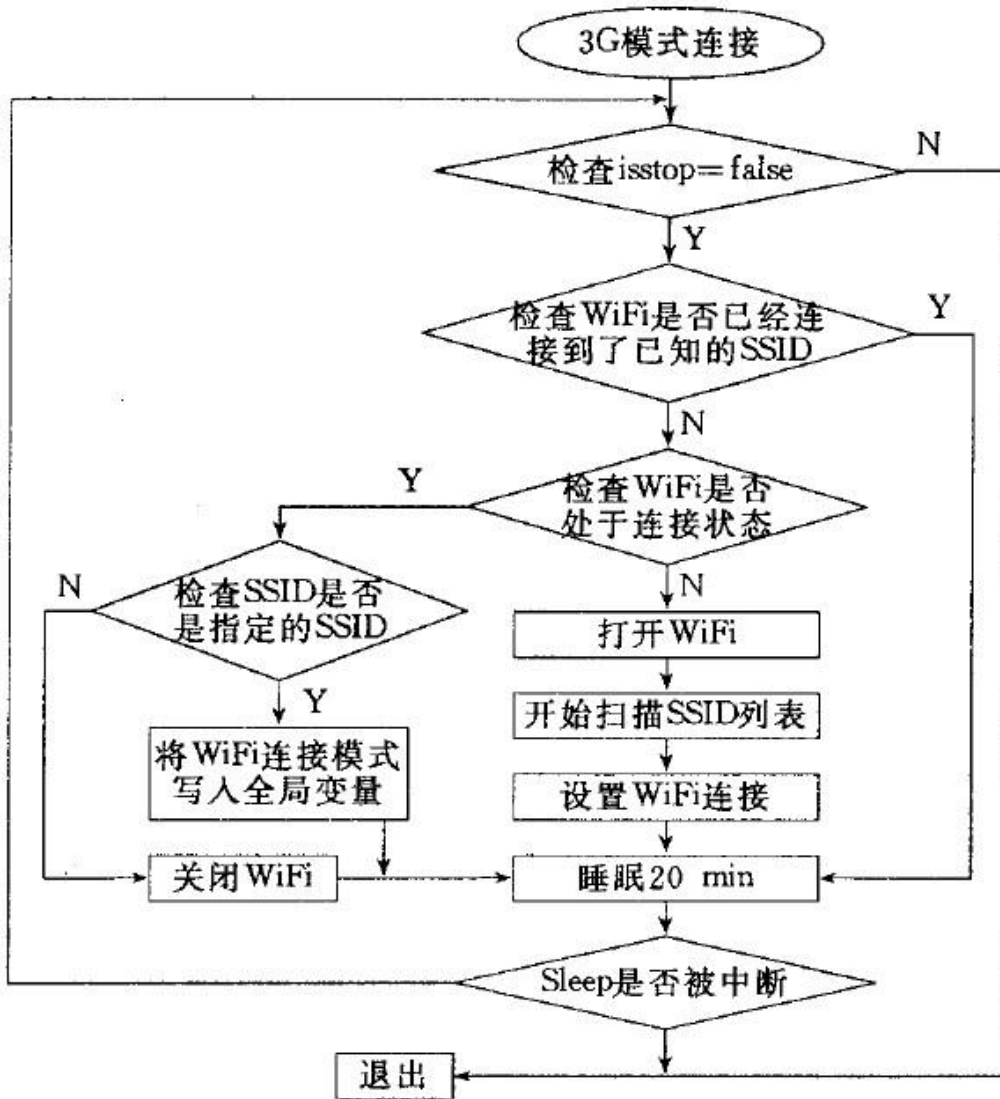


图 1 实现流程图

3 结束语

本文提出了一种基于 Android 系统终端控制上实现 3G 到 WIFI 以及 WIFI 到 3G 的无线漫游机制，分析了 3G 到 WIFI 的连接、认证、得到 AP 信息的过程。

该机制的实现还主要用到了 android 中的 ListView 用来进行数据显示，利用了 adapter 实现了 IListView 的数据传输，以及利用 SharedPreferences 实现简单数据存储。该漫游机制的实现方便了用户使用，减少了用户使用 3G 网络流量的费用。