1. 2015-5-19

添加“Report—Android Based Wi-Fi Direct设备连接类型与流程”

1. 2015-5-22

添加“Android Wi-Fi Direct中关于WSC的一些初步分析结果”

Report—Android Based Wi-Fi Direct设备连接类型与流程

主要内容要求：Wi-Fi Direct设备直连的类型和流程（包括交互信息、认证方法），每种方法都描绘详细的流程和目的。

由于Wi-Fi安全认证的方法和流程、以及在Wi-Fi P2P工作流程中的信息交互过程比较复杂，报告中提供的主要涉及Android系统上使用Wi-Fi P2P的主要工作流程分析，如果对某个具体阶段或流程需要更详尽的分析报告，我需要再进一步深入地了解和研究。

## Wi-Fi P2P工作流程

设备发现过程是比较复杂的，下面仅以一个简单、常见的场景（两个都未未加入P2P Group的P2P Device互相搜索，协商角色[GO or Client]，建立连接组成Group）来描述Wi-Fi P2P的主要工作流程，如图1-1所示。其它的场景包括1.一个未加入Group的P2P Device搜索某个P2P Group中的GO；2. 一个未加入Group的P2P Device/Legacy Client(P2P none-supported Device)搜索某个P2P Group中的Client；3.GO搜索周围的P2P Devices和Services；4.一个P2P Device搜索另一个已经加入某Infrastructure BSS的Device（可以同时支持P2P和STA）。



图1-1 典型Wi-Fi P2P工作流程图

两台设备想要建立连接首先需要进行设备发现，这一过程分为两个阶段，扫描阶段和发现阶段。Discovery启动后，首先进入扫描阶段，在P2P设备支持的所有频段上发送Probe Request请求帧；然后进入发下阶段，在这一阶段设备将不断在Search状态和Listen状态切换，以确保设备接收到对方发送的数据帧。

当一个P2P Device A通过Service Discovery流程找到周围的一个P2P Device B时，A就可以启动Group Formation流程进行设备连接和构建Group。这一流程也包括两个阶段，GO Negotiation阶段主要是两个设备协商角色，通过进行3次握手确定各自角色，如图1-2所示。GO协商与判定规则如图1-3所示，两台设备协商时使用GO Intent属性值判定，取值范围为0~15，值越高则表示成为GO的愿望越强烈，默认值为7。

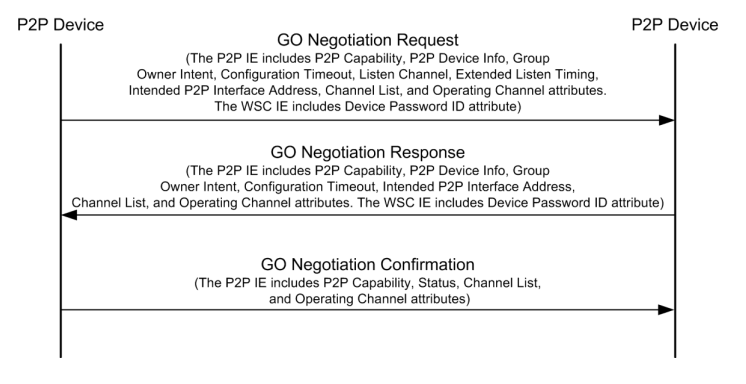


图1-2 GO Negotiation 3次帧交换流程

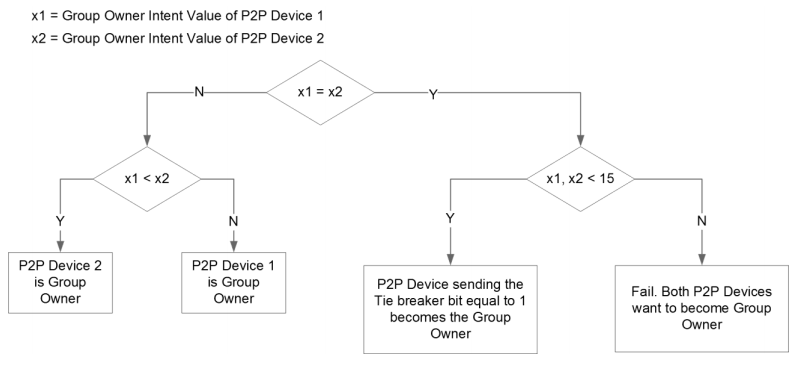


图1-3 GO决定规则

GO Negotiation 流程结束后，GO和Client的角色也随之确定，然后两个设备进入Provisioning阶段，使用WSC(Wi-Fi Simple Configuration)交换安全配置信息。P2P Client在关联到GO之前，需要先通过WSC协商安全信息。

WSC是Wi-Fi提供的基本服务，定义了13中配置方法，WSC不是Wi-Fi P2P所特有的服务功能。P2P所特有的技术3项，包括P2P Discovery、P2P Group Operation以及P2P Power Management，P2P技术依赖如图1-4所示。P2P Group Operation主要设计GO如果管理Group，包括为P2P Client Discovery提供GO Services、Group中的通信、Client从Group中断开连接、GO断开一个Client等等。



图1-4 P2P技术依赖图

其它的应用场景工作流程可能只经历用上述经典流程的部分阶段。例如，如果一个设备自主创建P2P Group，它将自动立刻成为P2P GO，其他的P2P设备通过使用标准的扫描机制发现已经建立的P2P Group，这种场景简化了设备的发现过程，只包含了P2P Device扫描阶段，设备不需要经历Find Phase（监听状态和搜索状态之间切换），也不再需要进行GO协商过程。在例如，P2P Group分为Persistent Group和Temporary Group两种，特别地，在发现阶段之后，如果一个P2P设备在过去承认自己创建的是持久型P2P Group，那么任何两台P2P设备都可以使用邀请过程（即两次握手，包括发送邀请请求和邀请响应）来快速的重新实例化P2P组，在这种场景中GO Negotiation阶段被邀请过程替换了，另外，由于可以使用存储网络凭证，所以WSC流程也简化了。对这些场景的具体描述与分析可以参考文章“Device to device communications with WiFi Direct: overview and experimentation”。

## Android Based Wi-Fi P2P工作流程

Android系统提供两种使用Wi-Fi P2P的功能，一种是上一节描述的扫描、发现、发起连接的典型P2P服务流程；另一种是P2P Pre-association Service Discovery服务，可以在没有创建网络连接的情况下直接发现周围设备提供的服务，并且由DNS based service discovery（Bonjour）或Upnp提供更高层的协议支持。

这两中连接类型的区别仅在于Group Formation procedure之前的设备发现过程，Service Discovery模式在Wi-Fi P2P协议之上使用Multicast DNS类型的服务协议以支持服务名称到设备的映射，通过服务发现对应设备。（这一段是根据Android Developer服务调用接口和说明以及Wi-Fi Peer-to-Peer (P2P) Technical Specification产生的自己的理解。）那么基于服务发现的Wi-Fi P2P工作流程将如图1-5所示。



图1-5 Service Discovery Based Wi-Fi P2P工作流程图

### 2.1 使用Wi-Fi创建P2P连接流程

在Android平台上，使用WifiP2pSettings进行P2P连接的流程如下：

1. 进入WifiP2pSettings界面；
2. 搜索周围P2P设备，并显示搜索到的设备列表；
3. 用户选择某个设备发起连接邀请。

在GO Negotiation阶段被邀请设备如果收到GON Request帧，会弹出如图2-1所示的提示框，用户需要选择“接受”，才能继续后续的流程，否则Group Formation流程终止。



图2-1 接收连接邀请设备提示框

应用程序中使用WifiP2pManager扫描设备和连接连接主要流程和接口函数如下：

1. initialize：初始化Wi-Fi信道和管理组件；
2. discoverPeers：启动P2P Device扫描流程；
3. requestPeers：获取P2P Device信息；
4. connect：向指定的P2P Device发起连接，进行协商以创建或加入一个P2P Group；
5. requestGroupInfo：获取当前的Group信息。

对应于第一节对其它场景Wi-Fi P2P工作流程，在应用程序中可以使用createGroup来自主创建一个P2P Group（自然地自己也就成为GO）。

应用程序借助WifiP2pManager与WifiP2pService交互，WifiP2pService是提供P2P功能的Java层核心模块。

### 使用Wi-Fi P2P进行服务发现流程

服务发现功能最上层基于组播域名服务（Multicast DNS），来源于Apple公司的Bonjour技术（基于组播域名服务的开放性零配置网络标准），使用该服务的设备在网络中自动组播自己的服务信息，并监听其他设备的的服务信息，从而使得局域网中的系统和服务在没有网络管理员的情况下也容易被找到。

以打印服务器为了介绍服务发现的流程：

1. 打印服务器根据零配置网络标准在局域网中找到一个可用的IP并注册一个打印服务，如名称为“Print Service”；
2. 某个Client需要打印服务时，根据服务名称“Print Service”去查找打印机，通过Bonjour（Multicast DNS，使用UDP组播DNS查询消息，根据Service名称获取对应的IP） Client找到注册“Print Service”服务的打印机并获取其IP及服务端口号；
3. Client通过服务的IP和Port直接与服务建立交互关系。

应用程序中使用WifiP2pManager 中Service Discovery有关服务的主要流程和接口函数如下：

1. addLocalService：注册并广播一个Upnp或者Bonjour服务；
2. addServiceRequest：初始化，准备进行服务发现流程；
3. discoverService：请求服务，进行服务的扫描和发现；
4. 如果发现服务则通过监听器获取服务相关信息，如IP和Port；
5. connect：向指定的P2P Device发起连接，（待确认：进行协商以创建或加入一个P2P Group）；
6. 以下的流程与扫描、建立连接之后的相同。

## WSC（Wi-Fi Simple Configuration）概述

WSC的目的就是简化无线网络配置，WSC定义两个应用场景（Usage Model），Primary UM和Secondary UM。Primary UM包括设置一个新的安全的WLAN，并为该WLAN添加无线设备；Secondary UM包括从WLAN中移除某个无线设备、通过添加新的AP或路由器来扩充WLAN的覆盖范围、密钥信息更换（Re-keying credentials）等。

Primary UM常见的两种案例包括PIN和PBC。

WSC定义的PIN码配置方法的工作流程如下：

1）打开AP和STA，用户首先从STA相关的设置选项中获取一个PIN码。

2）然后用户将STA的PIN码 通过AP的设置页面传递给AP。

3）AP和STA将基于这个PIN码 完成安全设置协商。然后STA将完成扫描、关联、四次握手等工作以加入目标AP。

Android上使用PIN码方法的操作截图见图2-2。



图2-2 Huawei G660上WPS PIN设置

PB的配置流程（Push Button Configuration，PBC）和使用更简单，但是需要设备上有支持该功能的按钮，用户只要对两个设备上单击该按钮，将触发两个设备完成安全设置协商。

Android平台上的WSC的实现贯穿App层（主要是Settings中的应用）、Framework层（主要是WifiStateMachine）和wpa\_supplicant。具体的安全协商流程和状态转换比较复杂，需要进一步学习和分析。

## 个人一点想法

如果要取消Android上的Wi-Fi Direct安全认证过程（避免用户参与连接过程），都需要修改系统。直接的方案就是取消系统提示框，如果在这个流程中不涉及其它安全设置操作的话；还可以通过修改安全配置信息，提前预置某种配置，避免安全信息的协商过程；还有，直接禁用WSC安全功能，如Wi-Fi P2P建立连接流程耦合性低并可以分离该功能的话可以尝试，当然这会导致Android系统上普通的Wi-Fi功能也没有安全保障。如果做系统修改的话，这些修改最好位于Java模块部分，如Framework层（wpa\_supplicant为C语言实现的Native Code模块）。

### 参考资料

[1] Wi-Fi Peer-to-Peer (P2P) Technical Specification Version 1.5

[2] Wi-Fi Simple Configuration Technical Specification Version 2.0.5

[3] Device to device communications with WiFi Direct: overview and experimentation

[4]邓凡平. 深入理解Android Wi-Fi、NFC和GPS卷. 北京：机械工业出版社，2014

[5]Android开发者网站关于wifip2p部分内容：<http://developer.android.com/>

## PS-报告反馈

对于WiFi P2P（WiFi Direct），WSC （或WPS）不是可选过程，系统强制要求采用PIN、PBC或NFC三种认证方式之一实现直连。

现在主要问题是需要确定WSC是不是能够禁用，如果说需要修改协议源码，在Android设备上实现是否可行？

在学习技术规范的时候，可以不用过多关注Service Discovery模式，我们使用基本模式，同时GO-negotiation过程不用作详细说明。

WSC操作处于provision阶段，需要详细考证WSC是否能够禁用。

Android Wi-Fi Direct中关于WSC的一些初步分析结果

通过对Wi-Fi P2P工作流程以及Android中WifiP2p向应用层提供的调用接口和她们实现代码的仔细分析，得到如下一些结论：

1. WSC不属于Wi-Fi P2P的直接实现和功能模块 ，无法通过修改Wi-Fi P2P部分代码禁用WSC；

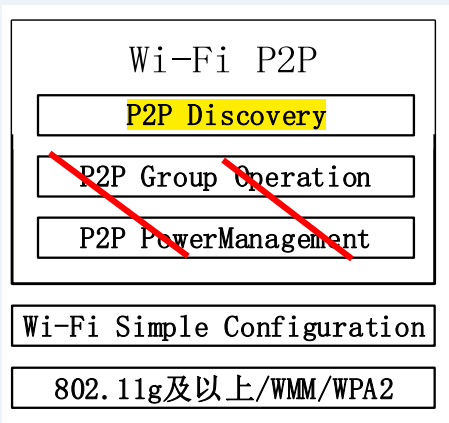


图1 P2P技术依赖图

根据图1（上一份报告中图1-4）P2P技术依赖图可知，如果去掉Wi-Fi P2P中Group管理和Power管理模块，那么Wi-Fi P2P所特有的实现模块只有P2P Discovery，也就意味着在设备发现和角色商定之后，Wi-Fi的核心功能完全由普通Wi-Fi技术支持。因此Wi-Fi Direct只是在建立连接过程中使用普通Wi-Fi 功能而间接使用了WSC功能。因此无法通过对Wi-Fi P2P部分的模块进行修改实现禁用WSC的目的。另一个佐证是，Android系统中Wi-Fi的核心进程wpa\_supplicant同时支持WifiService和WifiP2pService，在Group Negotiation阶段完成之后，WPAS也将创建一个新的wpa\_supplicant对象专门服务于P2P Group；即使有些系统中使用p2p\_supplicant支持WifiP2pService，p2p\_supplicant也仅仅是wpa\_supplicant的拷贝。

1. 如果要禁用WSC则只能从Wi-Fi标准功能模块WPAS入手，在wpa\_supplicant中研究是否可以不启用WSC，因此一旦禁用WSC也将破坏普通Wi-Fi的连接功能；

在Wi-Fi P2P Technical Specification中对使用WSC功能的P2P Invitation procedure也仅仅只有4页的描述，对WSC更是一笔带过，只是指出在此阶段进行WSC Exchange。

1. 个人的理解：Wi-Fi Direct中除了使一台设备作为GO模拟普通Infrastructure BSS中AP，其它的功能几乎与普通Wi-Fi无异；

因此下一步需要明确的问题和安排是：禁用WSC的目的是什么？如果是为了使用Wi-Fi Direct的组网和通信功能，但是又不想麻烦用户参与其连接过程（安全认证），我觉得没有必要禁用WSC，可以通过其它手段如预置安全配置信息或者通过NFC、USB简化认证操作；如果是为了使用自己的安全连接方案取代Wi-Fi提供的认证手段，那么我还需要进一步研究wpa\_supplicant，分析其能否实现WSC的禁用。