# WiFi Direct in Android

Mobile Computing
MEIC/MERC 2014/15

Nuno Santos

WiFi Direct: What Is It?

- Wi-Fi Direct allows devices to connect directly to each other using methods similar to traditional Wi-Fi, except without a pre-established access point
  - Instead, they negotiate to establish one peer device to act as a software access point for all the other devices in the group
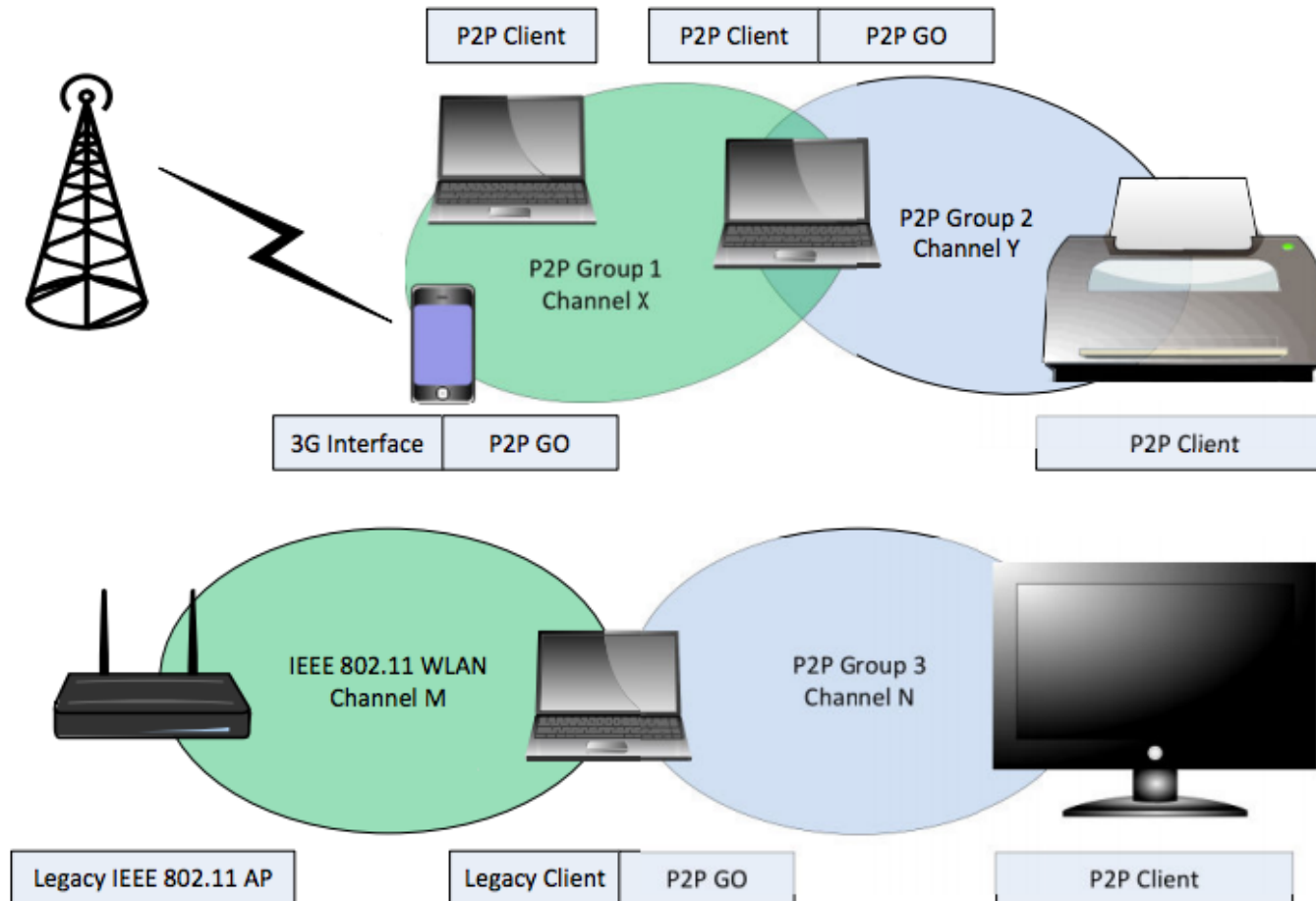
Client 1

Acts as Software Access Point
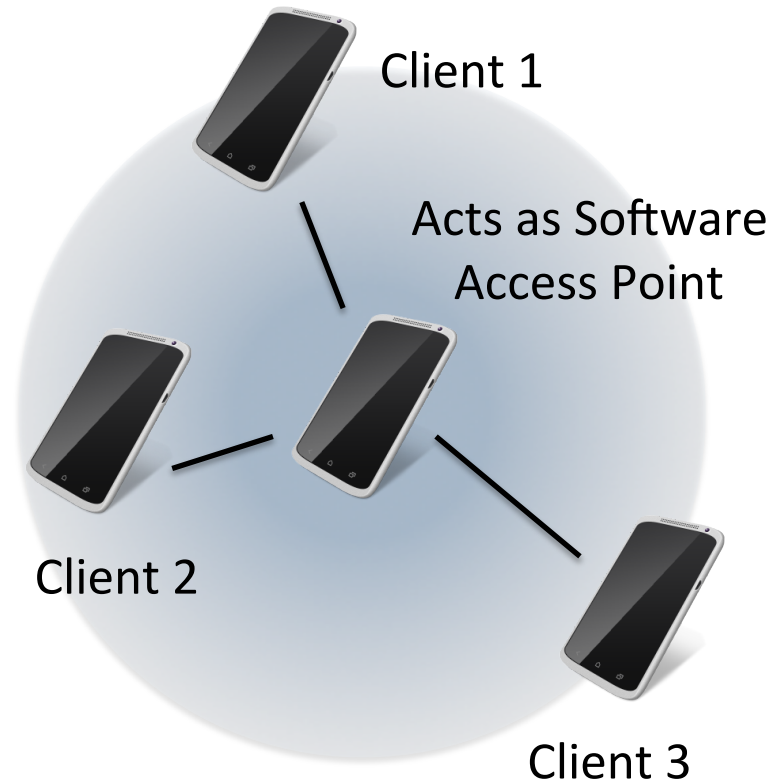
Client 2

Client 3

Devices Form P2P Groups

- **WiFi Direct device roles**
  - WiFi AP and client

- **Devices form P2P Groups**
  - Group Owner (GO) is the AP
  - Clients connect to the GO
  - The GO assigns IPs to clients

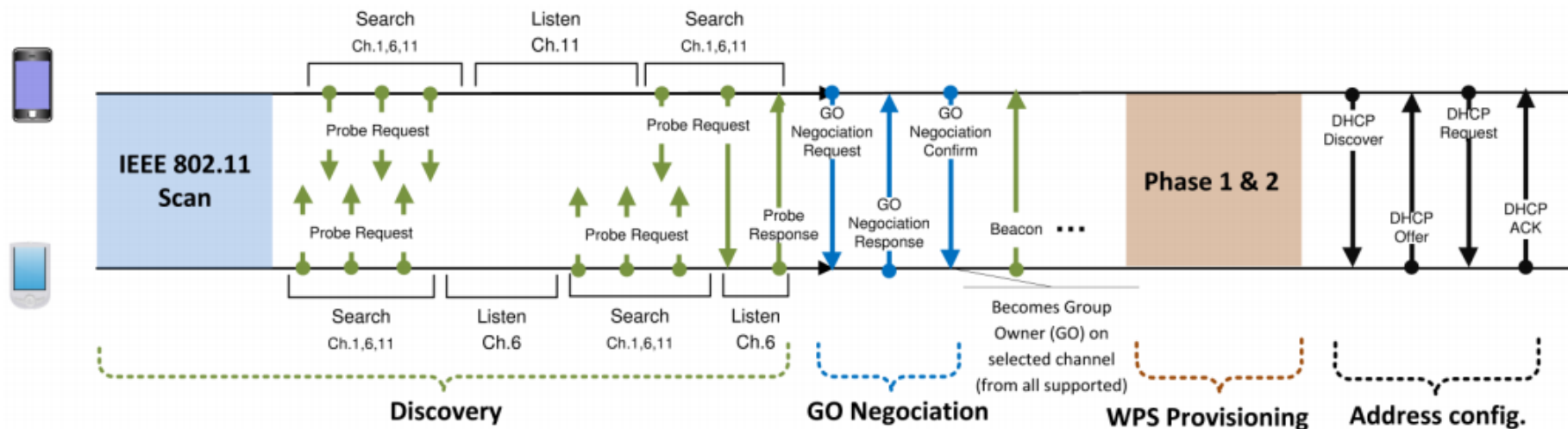**P2P Group**

GO of Group

Clients of Group
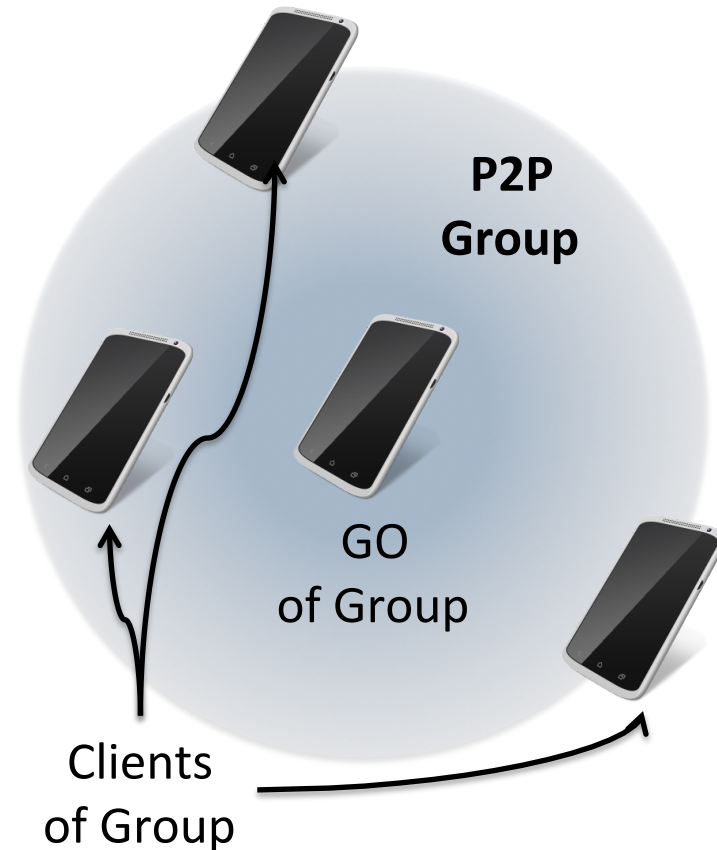
WiFi Direct Supports Many Topologies

P2P Groups Must Be Managed by App Developers

1. Group formation
   – Discovery, GO negotiation, Address config
2. Data transfer
3. Group tear down

Group Formation

THE TERMITE TESTBED

Termite: WiFi Direct Simulator

- Wdirect testable only w/ real devices
  - Emulator does not emulate WiFi Direct

- Use Termite to emulate WDirect nets

- Termite consists of:
  - Sandalone console
  - Library packaged with apps

- Tertmite workings
  - Console takes net topology script
  - Fires net events to emulator instances
  - Library emulates WDirect events
  - App captures and handles WDSim events

**Emulator Instances
w/ Termite Lib**

Termite Console
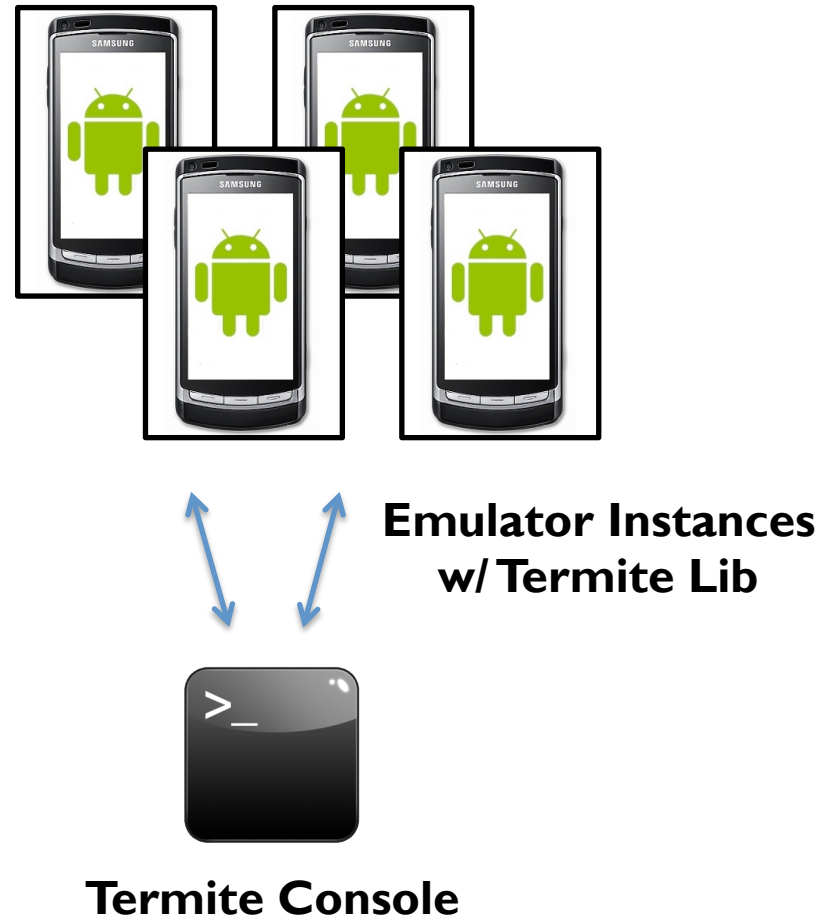
Termite Software Architecture

Software Stack
WiFi Direct

Software Stack
WDSim

App
WiFi Direct API
Android on Hardware

App
WDSim API
Android on Emulator

WDSim Console
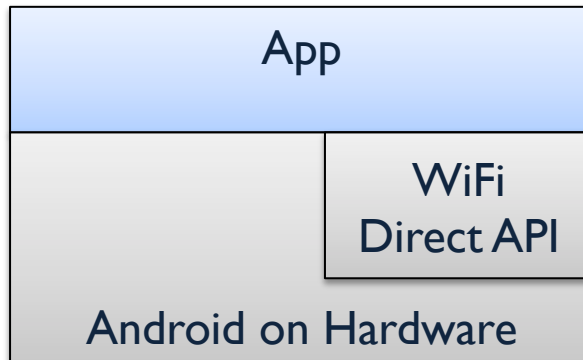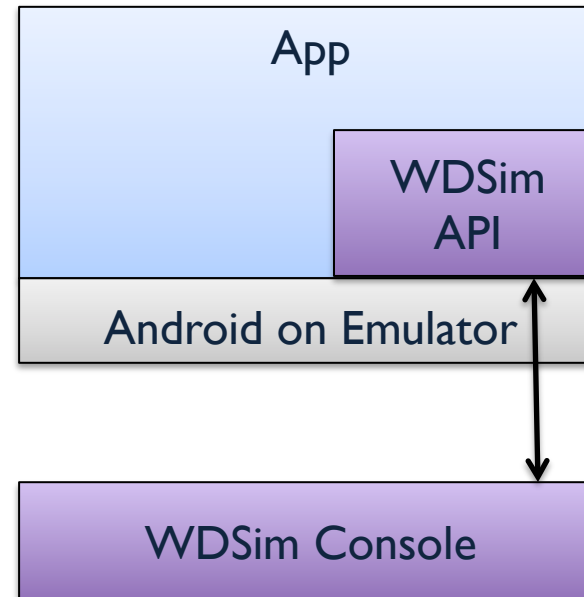
Attention: Termite API not fully compatible with WDirect

WIFI DIRECT ON ANDROID

Finding Nearby Peers

- There are two methods of finding nearby peers:

  - Peer discovery finds all nearby peers with WiFi Direct
    - http://developer.android.com/training/connect-devices-wirelessly/wifi-direct.html

  - Service discovery finds all nearby peers with WiFi Direct that are running the same app
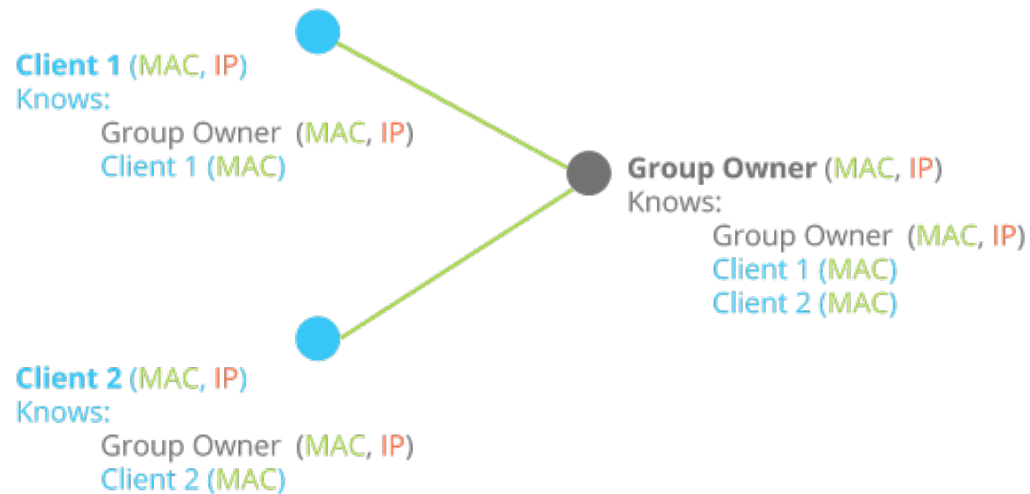   - http://developer.android.com/training/connect-devices-wirelessly/nsd-wifi-direct.html

Quirks If You Use Service Discovery

- In order to find each other, both devices must call WiFiP2pManager.discoverServices, not just one. This method puts the devices in the FIND phase, in which they both advertise and listen for nearby services. If you do not call WiFiP2pManager.discoverServices, your device will not advertise itself, and other devices will not be able to find it. Issue report on the Android issue tracker.
  - https://code.google.com/p/android/issues/detail?id=37425

- The Android specification claims that service discovery continues until the phone finds a device, but it was not very reliable in our tests. Sometimes devices failed to find each other, but later found each other if we called WiFiP2pManager.discoverServices a second time.

Connect to Peers

- Follow the developer guide:
  - http://developer.android.com/guide/topics/connectivity/wifip2p.html
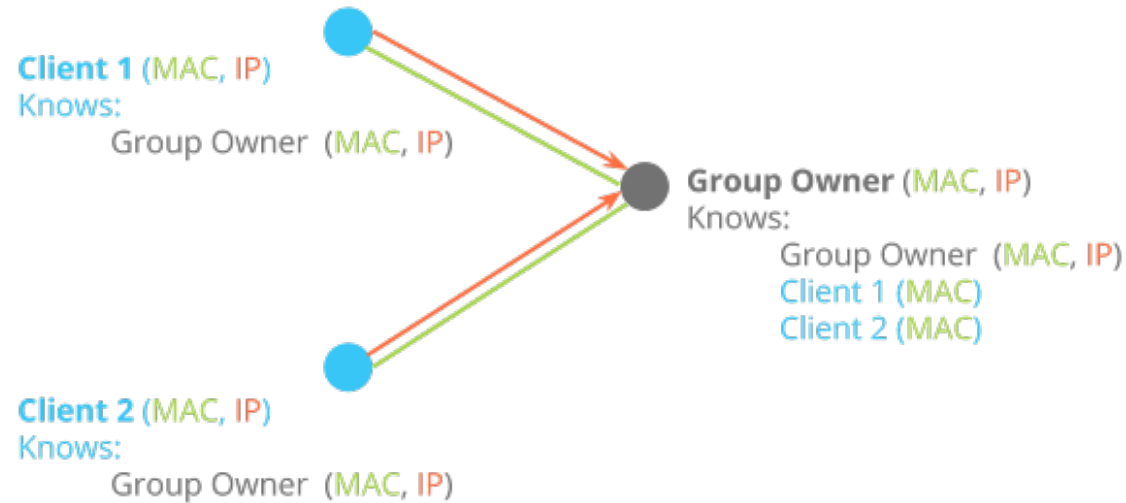
- Quirks:
  - If you already have a pending connection with a peer, you cannot attempt to connect to a second peer. If you wish to automatically connect to all the peers you discover, you must add the devices to a queue, and connect to them one at a time. We recommend sorting peers into three queues:
    - available
    - connecting (max size: 1)
    - connected

Make Peers Talk [1/4]

Client 1 (MAC, IP)
Knows:
    Group Owner (MAC, IP)
    Client 1 (MAC)

Group Owner (MAC, IP)
Knows:
    Group Owner (MAC, IP)
    Client 1 (MAC)
    Client 2 (MAC)

Client 2 (MAC, IP)
Knows:
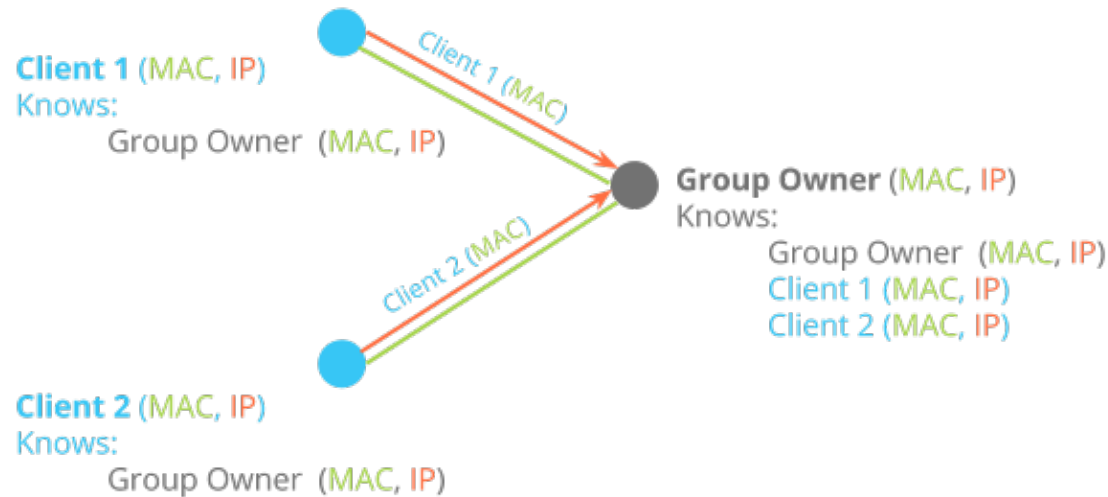   Group Owner (MAC, IP)
    Client 2 (MAC)

- Every phone has a MAC address and an IP address. By the nature of the WiFi Direct connection, the group owner is aware of all the clients MAC addresses, but the clients are not aware of each other
- In order to get clients to talk to each other, they must be aware of each other. To solve this problem, we needed to make the group owner distribute a list of the clients MAC and IP addresses to all clients

Make Peers Talk [2/4]

Client 1 (MAC, IP)
Knows:
    Group Owner (MAC, IP)

Group Owner (MAC, IP)
Knows:
    Group Owner (MAC, IP)
    Client 1 (MAC)
    Client 2 (MAC)

Client 2 (MAC, IP)
Knows:
    Group Owner (MAC, IP)

- Unfortunately, the Wi-Fi Direct API does not give the group owner direct access to the IP addresses of the clients. The group owner only has access to its clients' MAC addresses. To get access the clients' IP addresses, initiate a socket connection from the client to the group owner.

Make Peers Talk [3/4]

Client 1 (MAC, IP)
Knows:
  Group Owner (MAC, IP)

Client 1 (MAC)

Group Owner (MAC, IP)
Knows:
  Group Owner (MAC, IP)
  Client 1 (MAC, IP)
  Client 2 (MAC, IP)

Client 2 (MAC)

Client 2 (MAC, IP)
Knows:
  Group Owner (MAC, IP)

- Once that socket connection is established, the client can send a string containing its MAC address to the group owner, so the group owner would be able to add the IP of the socket connection to the known MAC of the client, thereby creating a MAC, IP pair

Make Peers Talk ^{4/4}
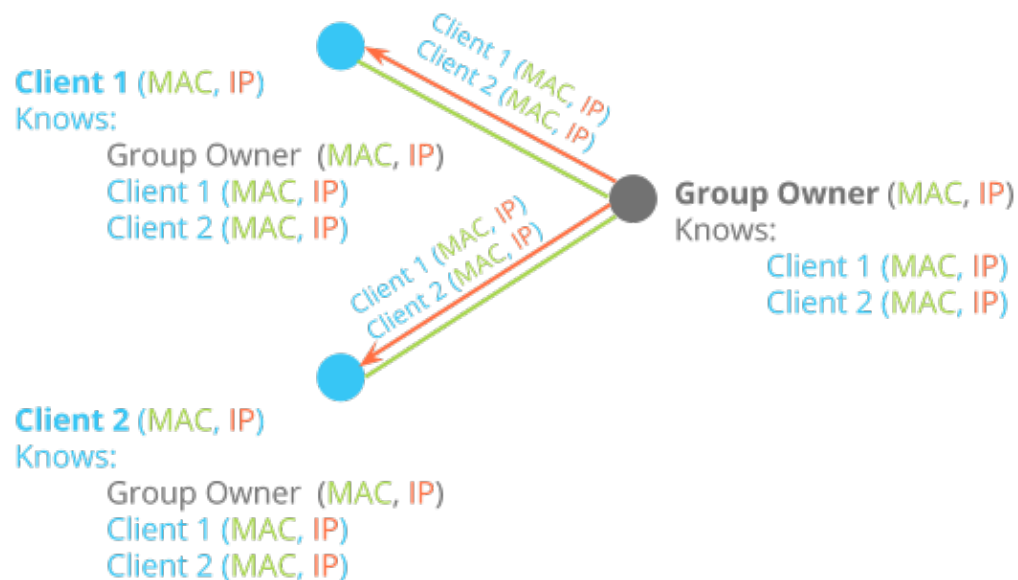
- The group owner can store all IP, MAC address pairs in a hashmap, and distribute this hash map via socket connection back to all the clients

Relevant Pointers

- android.net.wifi.p2p
  http://developer.android.com/reference/android/net/wifi/p2p/package-summary.html

- Connecting with Wi-Fi Direct
  http://developer.android.com/training/connect-devices-wirelessly/wifi-direct.html