Ad-Hoc on Android using Wi-Fi Direct

Geetika Dhand, Kavita Sheoran

Abstract: ***Starting with API level 14, ICECREAM SANDWITH, Android introduced API for peer to peer communication using Wi-Fi Direct. Since then many applications have been released that use this technology for data transfer and communication between two devices. But most of these have been used for just two-person communication. In this paper we propose to create a peer to peer network using the same technology. This paper focuses on creating a local wireless network of android smartphones using Wi-Fi direct. This paper also defines an algorithm using which each device can get the IP address of each of its peer. This paper will also help those who are trying to create applications on android using Wi-Fi Direct.***

Index Terms: Wi-Fi Direct, Android, Peer to peer networks.

I. INTRODUCTION

The last decade has seen a rapid increase in Telecommunication technologies. With the fast pace of development, mobile phone was invented, and the old telephone could now fit into our hands. This mobility soon made it a very popular device. The seemingly magical device has a lot of stationary infrastructure working at its background. There are cell phone towers and exchanges that keep track of all the connections. But in case of a calamity, all the infrastructure fails. In such a scenario there is a need of a readily deployable network. Now- a- days much research is going on in the field of such Ad-Hoc networks. At the same time a new technology called Wi-Fi Direct has emerged which allows for seamless creation of a network between two devices. This paper explores to create a local Wireless Network using Wi-Fi Direct. In a typical Wi-Fi network, clients discover and associate to WLANs, which are created and announced by Access Points (APs). In this way, a device unambiguously behaves either as an AP or as a client, each of these roles involving a different set of functionalities. A major novelty of Wi-Fi Direct is that these roles are specified as dynamic, and hence a Wi-Fi Direct device must implement both the role of a client and the role of an AP (sometimes referred to as SoftAP). These roles are therefore logical roles that could even be executed simultaneously by the same device, for instance by using different frequencies (if the device has multiple physical radios) or time-sharing the channel through virtualization techniques. To establish a communication, then, P2P devices must agree on the role that each device will assume.

II. BACKGROUND

A) Wi-Fi Direct overview

According to Wi-Fi Direct Technical Overview [1], a device can dynamically assume the role of either an access point or a client.

Revised Manuscript Received on July 05, 2020.

Geetika Dhand, Assistant Professor**,** Department of CSE, Maharaja Surajmal Institute of Technology, Janak Puri New Delhi, India.

Kavita Sheoran, Assistant Professor**,** Department of CSE, Maharaja Surajmal Institute of Technology, Janak Puri New Delhi, India.

Wi-Fi Direct devices, formally known as P2P Devices, communicate by establishing P2P Groups, which are functionally equivalent to traditional Wi-Fi infrastructure networks. The device implementing AP-like functionality in the P2P Group is referred to as the P2P Group Owner (P2P GO), and devices acting as clients are known as P2P Clients. Given that these roles are not static, when two P2Pdevices discover each other they negotiate their roles (P2PClient and P2P GO) to establish a P2P Group. Once theP2P Group is established, other P2P Clients can join the group as in a traditional Wi-Fi network. Legacy clients can also communicate with the P2P GO, as long as they are not 802.11b-only devices and support the required security mechanisms (see Section II-D). In this way, legacy devices do not formally belong to the P2P Group and do not support the enhanced functionalities defined in Wi-Fi Direct, but they simply "see" the P2P GO as a traditional AP.

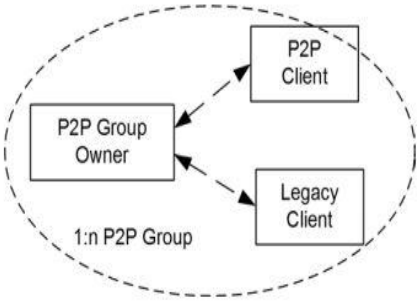

Figure. 1P2p Components and Topology

B) Contemporary Applications

With the rollout of an API for Wi-Fi Direct in Android-android.net.wifip2p [2], many applications were developed. Most of them focused on file sharing between two android enabled smartphones. One of the examples will include SuperBeam [3]. This app can be used to transfer any file, image audio, etc. between two devices using Wi-Fi Direct. But such apps use a point to point approach (see Figure 2). Here the sending application becomes the group owner using Wi-Fi Direct API and the receiver can join the network to receive these files.

Figure. 2Point to point communication in Wi-Fi Direct

This paper explains about creating a peer to peer network using Wi-Fi direct. In our network instead of following a point to point approach we create a one to many associations of Group Owner to Client as shown in Fig. 3

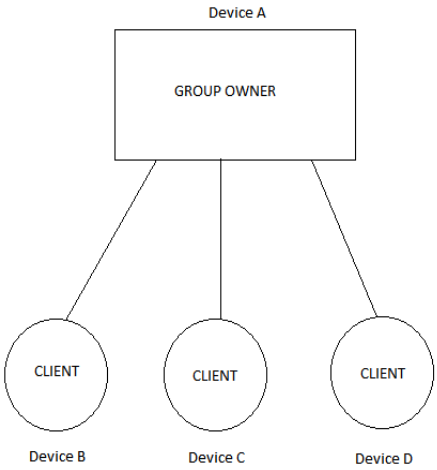

Figure. 2 One to many network with the group owner

Once all clients join the network of Group Owner (GO), they know the IP address of the Group Owner. But these clients don't know the IP address of other clients. There is a limitation in the Android API that even the GO does not know the IP of the clients. We achieve this as by creating the IP-MAC mapping table as explained in sectionIII D.

III. PROPOSED ARCHITECTURE

A) Network Formation

In our test we found that when a new peer requested to join a network, all it sees is the peer list. Now if it initiates the connect request to a client, the connection setup fails. So, we have to initiate the request to a Group Owner only. So, first step involves finding out the GO. This is achieved as follows:

Initially a peer discovery is started. When peers are available, we get a list of WifiP2pDevice type[]. We iterate through the peer list and check if a device is Group Owner. Now if a group owner is found, then a connect request is initiated to the desired Group Owner. Otherwise the connect request is initiated to the first peer in the peer list. The process can be summarized as in the flowchart in Fig. 4.

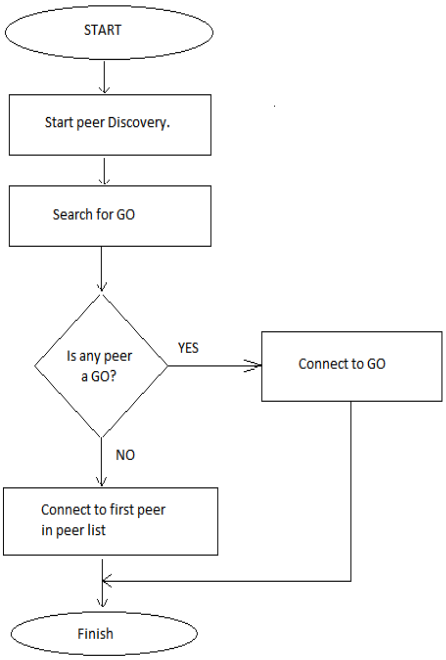

Figure. 3 Flowchart for joining network

Once all peers connect in this manner a one to many network will be created. Once the devices are connected in this manner we can begin the MAC IP table creation algorithm. Before discussing that let us first discuss some basic concepts which will be used in the algorithm.

B) Identification

MAC addresses serve as the basic device identifiers in our approach. But the MAC address used in android Wi-Fi direct implementation is different from the MAC address of the device. This MAC can be obtained on receiving the intent WIFI_P2P_THIS_DEVICE_CHANGED_ACTION. Along with the MAC, the name of the device is also received. Care must be taken that throughout the implementation only this MAC must be used as this is the MAC that peers see in the peer list.

C) Message Types

Two types of messages are used throughout the project. These are

1. Hello Packet
2. Message Packet

1) Hello Packet:

Whenever a device first joins the network, send this hello packet. This packet contains the p2p MAC and the IP address of the current device. Whenever a device receives a hello packet, it updates it MAC-IP table as illustrated in section III D.

2) Message Packet:

Any text message sent from the device is not sent directly. It is encapsulated as the Message packet. This packet also contains the MAC and P2p name of the sending device along with the message string.

When a device receives this type of message, it may store it in database and update the view as necessary.

D) MAC- IP table creation

When the client has successfully associated with the GO, it knows the IP address of the GO. But in API there is no provision to fetch the IP address of the client. One way to access will be to access the DHCP files on the GO. But that table as shown in Figure 5.

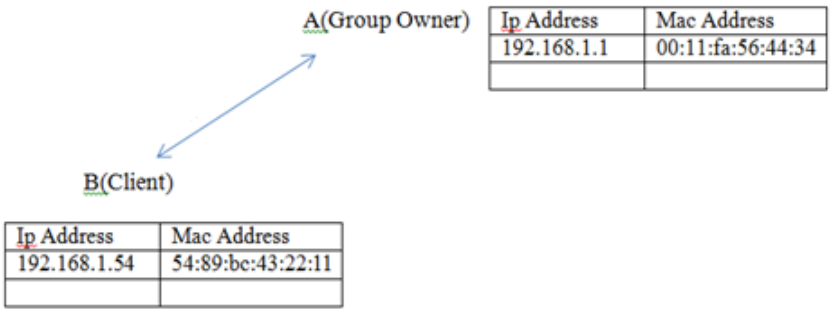

Figure 5The client as well as GO enter their own MAC and IP in the

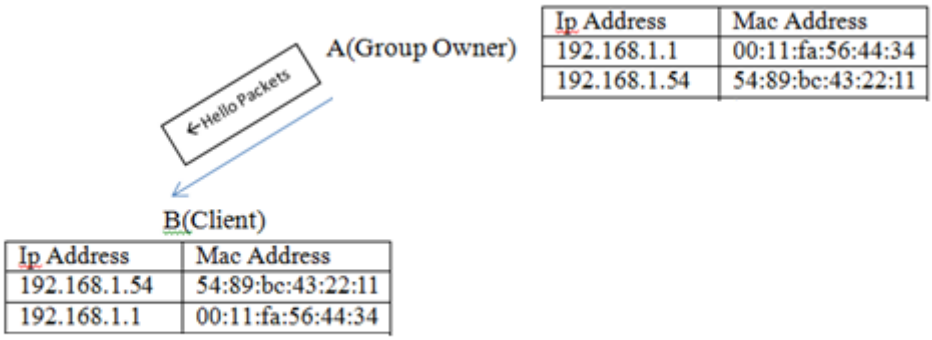

Figure. 6 The GO makes the entry in its table and broadcasts the table to client who modified its table

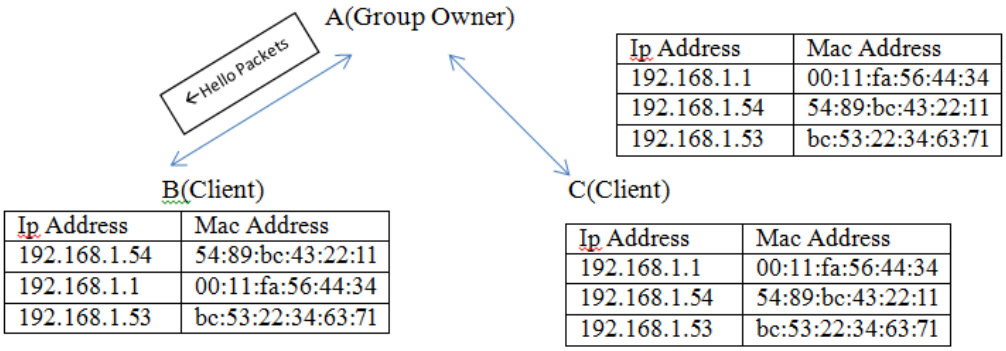

Figure. 7 When a new client sends its hello packet, GO modifies its own table and broadcasts the entry to each IP in its table.

As soon as a device receives a Hello Packet, it extracts the MAC and IP from the Packet and then makes this entry in its table. So when the GO receives the Hello packet from client, it modifies its table. Now the GO broadcasts its table to all the IPs in its table by creating a Hello Packet for each entry of the table. As soon as the client receives this, it modifies its table as shown in Figure 6. Now the client as well the GO both know the IP of each other.

Similarly when a third client will join the network, the same process will happen. The GO will modify its table and broadcast it to every client associated with it. Now every device has the IP address of every other device as shown in Figure. 7

Now each device can initiate a direct communication with other client. When it wants to communicate with a particular peer in the peer list, it can get its MAC and find the corresponding IP address from its MAC IP table.

IV. CONCLUSION

This paper can act as a base for all those who want to extend the use of Wi-Fi direct on Android. We provide a clear method of creating a one to many network and then creating a MAC IP table at each connecting device. This can extend the previously limited use cases of android applications.

REFERENCES

1. Device to device communications with WiFiDirect: overview and experimentation
2. Wi-Fi Alliance, P2P Technical Group, Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.0, December 2009.
3. http://developer.android.com/reference/android/net/wifi/p2p/package-summary.html
4. http://forum.xdadevelopers.com/showthread.php?t=2177133
5. https://play.google.com/store/apps/details?id=com.majedev.superbeam