WIRELESS NETWORK PROJECT
ProximityCall App

Alessio Lazzaron
alessio.lazzaron@studenti.unipd.it

February 8, 2021

Abstract

This article summed up my work for the Wireless Network course project, where I developed an Android application that allow the communication between people in a confined area without using mobile network.

1 Introduction

The aim of the project is to develop an Android application that uses wireless technologies, present in our phones, for a real purpose.
This application is made for normal people that want to communicate between them but there is no mobile network or anything else similar. So I imagine an emergency context where, for example, there are people that live in a small village in the mountains and for some reason the electricity fail and maybe there isn't mobile network. So people want to communicate to each other to know if something serious happened. In this context you can used a portable radio and communicate with AM or FM signal but you must have the radio. An cheaper way is to install, in the phone, this application.
To allow the communication between people, that are approximately close to each other, I used the Wi-Fi direct standard that is present also in the older smartphone. In the next paragraph I analyze the architecture, strengths and weaknesses of this standard and why I choose it.

1.1 Wifi Direct

Wi-Fi Direct [1], initially called Wi-Fi P2P (Wi-Fi Peer-to-Peer), is a standard for Wi-Fi Alliance and it is a Wi-Fi technology that allows two or more devices (also of different nature: Tablet, PC, TV, etc ...) to connect directly to each other without relying on a Wireless Router.
For communication, Wi-Fi Direct devices have to establish a group. In each group there is only one *Group Owner*, that plays a role of access point, and one or many *Group Client.* Basically when the group is created, other devices can join into the group as a clients, so the owner can connect with multiple devices at the same time. The group owner plays a central role in this architecture, indeed he can determine the lifetime and the communications of the group, also he remains in office until the group closes.
Basically the Wi-Fi Direct protocol was designed to work in two different context that are show in Figura 1. In *One-to-one* configuration the connection is available between only two devices while in *One-to-many* configuration connection occurs between multiple devices. In this configuration the laptop will be the Group owner and all the others the Client, so in this way you can interact with the owner to have access to other devices.

To find new device Wifi-direct adopt a specific method which is divided into two phases:

- *Scan*: During Scan phase, each device collects information about nearby devices by scanning all supported wireless channels (the available chan-

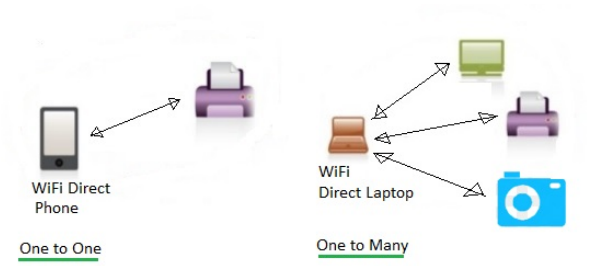

Figure 1: Wi-fi Direct context of work

nels are 1, 6, 11). Also this phase has a specific pre-defined duration.

- *Find*: In this phase the devices alternate between two states: a *search state* in which the device performs active scanning by sending Probe Requests; and a *listen state* where the device listens for Probe Requests and then responds with Probe Responses. A *Probe request* is a special frame sent by a client requesting information from either a specific access point, or all access points in the area.

To create a group there are three different way:

- *standard*: first of all, the devices needs to discover each other and then they negotiate on which device will be the group owner. So to determine who will be the owner, the two devices send, between each other, a numeric value via the three-way handshake mechanism (also used by TCP/IP) and whoever declares the highest value will become the group owner. When the roles are established is created a secure network with Wi-Fi Protected Setup and finally the setting of an IP address through the exchange of DHCP.

- *autonomous*: in this way a device elects itself as group owner and announces its presence through beacon messages. So if a device want to connect in this group he already know that he is a client.

- *persistent*: when a group has been formed, with the methods above, the owner is able to declare that group is persistent. So a device can recognize, during the scanning phase, that group had already been formed previously by looking at a specific flag present in the Beacon frame of the Probe Response. For example, a laptop could create a Persistent Group that comprised of the laptop and a printer. But this group is not always on, so create, every time, a new group is a long procedure which takes time. It is easier if the setting of the group are saved and used when you need to turn it back on so when the laptop send a print request.

1.2 Why used Wi-Fi direct?

In the last paragraphs I analyze the WI-Fi direct architecture and it can be summed up as a protocol that simplifies the connection from one device to another, avoiding using a third device. But inside a smartphone there are other technologies that allow the same thing, for example Bluetooth. So in this paragraph I list the difference between these protocols and why I choose the Wi-Fi Direct protocol.

- *Bandwidth*: Wi-fi Direct promises 250 Mbps while Bluetooth 4.0 achieves a maximum of 25 Mbps, so ten times less fast. However Wi-fi Direct is not suitable for streaming.

- *Range*: Wi-fi Direct supports a theoretical maximum distance of about 250m while Bluetooth 4.0 has a much shorter range of action, in fact it is about 7m.

- *Connectivity*: for how it was created Bluetooth can manage only 7 users, while the number of clients that can participate in a Wi-Fi Direct group is not specified so a number bigger than 7.

- *Power Management*: although both protocols are classified as ”low energy” and are thought for a specific class of device that used rechargeable batteries, however Bluetooth 4.0 is better than Wi-Fi Direct, in fact it promises to be able to work continuously, with a common rechargeable battery, for almost more than a year.

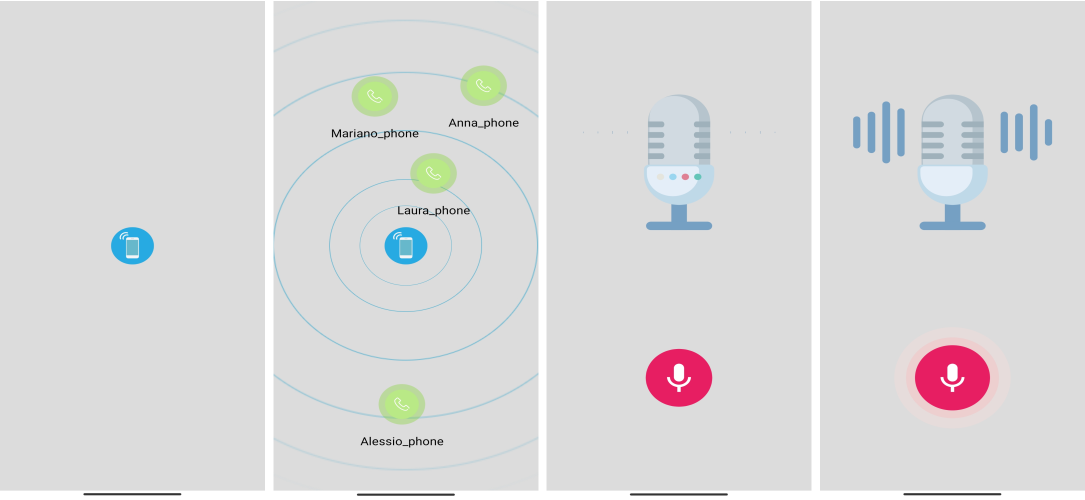

Figure 2: ProximityCall App user interface: a) Homepage; b)Homepage with discovered devices; c) Call-page; d) Call-page where user is talking

The main reason why I used Wi-Fi Direct is not for the connectivity property but for the range, Bluethoot has too narrow a range.

1.3 Limitations of Wi-Fi Direct Technology

For my point of view Wi-Fi Direct has two main limitations:

1. Communication between client devices are not supported inside a group. So the client can only communicate with the Group owner and to each other.

2. Not support for multi hop communication. This is due to the protocol that client can not become a owner in another group so client can not establish connections to multiple devices.

2 Application's Design

I develop a simple and minimal user interface so that it was intuitive for the user therefor not require a guide or list of instruction to be used.

The user interface is splitted in two different part with distinct function: Homepage and Call-page.
The *Homepage* is the first view that see the user and, as show in Figura 2a, there aren't many option indeed there is only one button on the center of the view. When the user click on this button the discovery phase of available peer devices starting and this is showing to the user with an animation. Every device that was found is a user that he installed, on his own smartphone, the application and he is ready to communicate. If some device is found these appear to the view as show in igura 2b.
At this point the user can decide which device to connect to initiate the call and he can do this simply by touching a founded device in the Homepage. When he does that the connection between the two devices is established and they are brought into the *Call-page*. This view is shown is igura 2c. Now the user can do two things: talk with the other user or stay and listen to him. For the first option the user must click to the record button and talk, when he finished or there is some noise inside the room that he don't want the other user listen, he clicks on the same button to close the mic. The second option is easier than the first one because the user doesn't have

to do anything and listen to what the other user has to tell him. Also it is possible, to both users, to talk at the same time.
When a user want to close the call, he basically turn back to the Homepage, with the back button that is present in Android phone, and the connection between them will be closed. Also for the other user will be automatically returned to the home because the connection was closed.

3 Implementation

In this section will describe all the technologies used for the development phase and how is the life cycle of the application.

3.1 Developer Ambient

For the development of the application it was used the IDE Intellij Idea on Windows 10 Home 64bit. I used the last version of Android API (version 30) where I found everything I needed to use Wi-Fi Direct. Also I used the library of Lottie animation [2] and Ripple-Background [3] for the aesthetic part. Moreover the whole application was developed in Java.
The mobile devices used for application development and testing are as follows:

- Xiaomi Mi 9T with Android 10
- Xiaomi Redme 7 with Android 9.1 Pie

3.2 Application lifecycle

In Android to use the Wi-Fi Direct you must use the *WifiP2pManager*[4] class. This object allow you to create different async threads to manage the communication between multiple devices. In my application I used one thread to find all near available devices and other two different threads to established the connection and to get information about the other device respectively. I created only Wi-Fi Direct group with only two devices: one owner and one client, I was forced to do that for the protocol limitation.
After the connection was established I initialized a TCP Socket communication between the two devices for the voice packet, where the owner take the server part of the protocol TCP and the group client take the client part. When one user decide to close the call he close also the socket and after that the application close the connection between the two devices.

4 Conclusion

During the development of the application I encountered some problems especially related to the Android development environment, although it does not bind the developer but it's a fairly complex environment for novice developers like me.
Also Wi-Fi Direct technology it was not what I expected although it has a range of more than 200m this is a theoretical value and with some obstacle the real range is less. Moreover it doesn't offer the possibility to the group to communicate between them so the only way to implement a multy-hop communication is a custom solution.
Overall I value the experience positively and I am happy with the application that I developed.

References

[1] Maicol Landi. Communication device to device through wi-fi direct technology: An experimental evaluation. 2015.

[2] Lottiefiels, "https://lottiefiles.com/".

[3] Ripple-background, "https://github.com/skyfishjy/android-ripple-background".

[4] Wifip2pmanager, "developer.android.com/ reference/android/net/wifi/p2p/wifip2pmanager".