Android Programming: Wi-Fi Detection
Mobile Internet Lab2

Liu Hanwen (517030910294)
Shanghai Jiao Tong University IEEE Honer Class
hwLiu2017@sjtu.edu.cn

April 26, 2020

Abstract

Understand the principle of Wi-Fi detection and java code. Use the easiest method to location Android phone by calculating distance from Wi-Fi sources through strength and frequency by scanning. *Key Words:* Wi-Fi, Java, Strength, Frequency, Location

1 Environment

I use Android Studio 3.6.2 with Oracle Java 1.8.0.

2 Project: Wi-Fi Detector

2.1 Brief Introfuction

In this java program, there are two classes which are important. One is MainActivity.java, we have recognized in the former project. The other is SuperWiFi.java, which abstract the implement as a independent class to make the structure of the code more clearly.

Mainly we use WiFiManager to scan Wi-Fi in the environment.

2.2 Answer For Questions

1. Why is necessary to record all the measured value rather than only the average value?

   From the code, we can find out that we store all the measured value and we use the average value as the output. I think store all the value can help us to extract proper value when there may be noisy value.

   Also storing all the values, we can construct distribution of Wi-Fi signal strength to gain more information of the specific Wi-Fi source.

2. Besides the Wi-Fi signal strength, what other information of the Routers can be got in the test?

   During the test, through WiFiManage Class, we obtain scanResults. Besides the strength which we mainly focused, we can also obtain the following informations(just for example):

   (a) getMaxSignalLevel() Get the system default maximum signal level.

   (b) getNetworkSuggestions() Get all network suggestions provided by the calling app.

   (c) is5GHzBandSupported() Check if the chipset supports 5GHz band.

   (d) isAutoWakeupEnabled() Get the persisted Wi-Fi auto wakeup feature state.

   We can find more information from the Android website:

   https://developer.android.com/reference/android/net/wifi/WifiManager

3. Why does the scanning need to be operated in thread “scanThread” ?

   Because in java code, we start scanning by changactivity, which will be called by clicking the bottom. So there is such a possibility that during once scanning, the operator clicks twice or more times on this buttom. In this case, by using thread, we can deal with this kind of condition, and give the proper results. In a nutshell, we can realize multi-thread processing.

3 Additional Questions

3.1 Introduction

From the code, we can find out that actually we can obtain all the Wi-Fi information by WiFiManager.getScanResults(). In this case, we can pick the top three powerful points as the target points to construct the positioning.

3.2 Principle

The main function to calculate the distance is:

$$Distance = 10^{\frac{27.55-(20*log_{10}(Freq))+signalLevel}{20}} \tag{1}$$

```java
public double calDistance(double signalLevelInDb, double freqInMHz) {
    double freqExp = 20 * Math.log10(freqInMHz);
    doubel signalExp = Math.abs(signalLevelInDb);
    double exp = (27.55 - freqExp + signalExp) / 20.0;
    return Math.pow(10.0, exp);
}
```

3.3 Design

In the experiment, I design three specific Wi-Fi sources:

Family: The settled routine in my home.

Hanwen: The mobile hotspot inspired by my laptop.

Hanwen's iPhone: The mobile hotspot inspired by my iPhone.

In the following demos, I choose four information obtained from scanning:

Name: Name of the Wi-Fi source.

Strength: Strength of Wi-Fi(db).

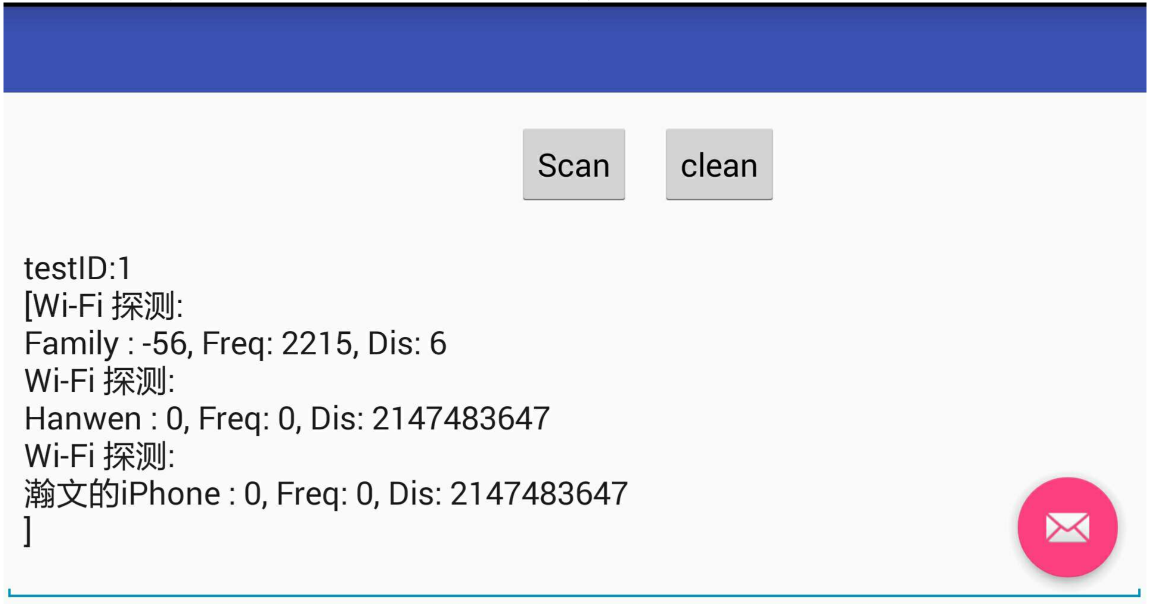
Freq: Frequency of Wi-Fi(MHZ).

Distance: Distance(m) calculated by Strength and Frequency.

3.4 Demo Experiment

3.4.1 Case 1

In case1, I only activate one Wi-Fi source: "Family"

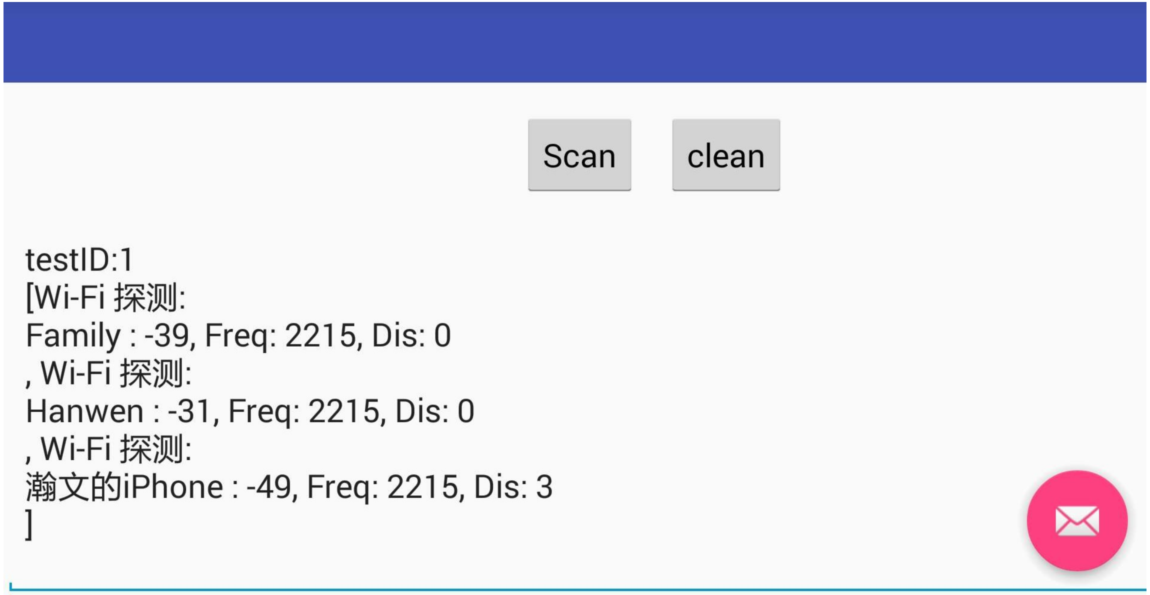


```
testID:1
[Wi-Fi 探测:
Family : -56, Freq: 2215, Dis: 6
Wi-Fi 探测:
Hanwen : 0, Freq: 0, Dis: 2147483647
Wi-Fi 探测:
瀚文的iPhone : 0, Freq: 0, Dis: 2147483647
]
```

We can find out that the distance between my Android phone and my settled routine is 6m, which is almost exact. And for the other inactive Wi-Fi source, I give a maximum distance to represent not available.

3.4.2 Case 2

In case2, I activate all Wi-Fi sources.

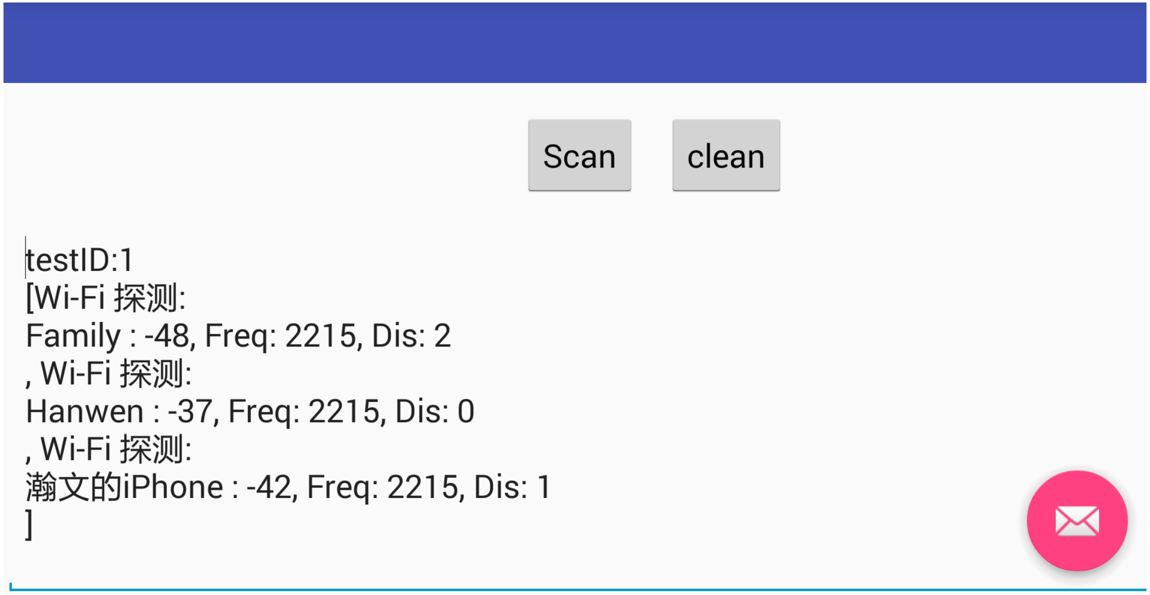


```
testID:1
[Wi-Fi 探测:
Family : -39, Freq: 2215, Dis: 0
, Wi-Fi 探测:
Hanwen : -31, Freq: 2215, Dis: 0
, Wi-Fi 探测:
瀚文的iPhone : -49, Freq: 2215, Dis: 3
]
```

And my Android phone is near these two new-added sources. The distance is 0, which is also correct.

3.4.3 Case 3

In case3, I activate all Wi-Fi sources.



```
testID:1
[Wi-Fi 探测:
Family : -48, Freq: 2215, Dis: 2
, Wi-Fi 探测:
Hanwen : -37, Freq: 2215, Dis: 0
, Wi-Fi 探测:
瀚文的iPhone : -42, Freq: 2215, Dis: 1
]
```

I change the location of my iPhone and Android phone. Because the limit of data cable between Android phone and my laptop, the real distance still no more than 1, and it shows 0 in the screen which is correct. As for my iPhone, I throw it on the other side of sofa in my room. In this case, I create 1 meter interval and my program makes sense again.

3.5 Conclusion

By my method and demo experiment, I find my program which has the ability to calculate approximate distance. With the distance from several known Wi-Fi sources, it is easy to find location of current phone by these known locations.