

# COMP4336/9336

## Lab - WiFi-2

### Lab Objectives

You will learn more about wifi using android Wi-Fi classes and methods. You will learn about:

1. Frequency of wifi connection
2. IEEE 802.11 standard and its various version
3. Bit rate of wifi connection

### Preparation

1. *Wi-Fi Background:* The IEEE 802.11 standard is free to download from IEEE's website. You are encouraged to read through the standard if you have time (but it is a 500-page document). Besides the full documents of the standard, you can find numerous short introductions and tutorials of 802.11 from the Internet (e.g. [http://www.sss-mag.com/pdf/802\\_11tut.pdf](http://www.sss-mag.com/pdf/802_11tut.pdf) ) or the computer networking books published in recent years. Read some of them and make sure you understand the following concepts: AP, ESSID, BSSID, association, beacon, signal strength, channel, authentication process.
2. In Android development, you can programmatically configure the wireless connectivity of you android devices. Let's find out how to do that, Android comes with a number of system services that are always on, and running and readily available for Android Developers. In Android Development, like all other systems services, provides an API for managing all aspects of the Wi-Fi Connectivity. WifiManager (package name: android.net.wifi) is the class which provides the primary API for the WI-FI services. An operation for network connectivity can be done using the ConnectivityManager class (package name: android.net).
3. The following classes are useful for you to do the lab tasks:

- **WifiManager**

This class is most important class to manage the Wi-Fi connections. It provides the primary API for managing all aspects of Wi-Fi connectivity. Get an instance of this class by calling `Context.getSystemService(Context.WIFI_SERVICE)`. You can use its. More details:

<http://developer.android.com/reference/android/net/wifi/WifiManager.html>

- **ScanResult**

Describes information about a detected access point. In addition to the attributes described here, the supplicant keeps track of quality, noise, and maxbitrate attributes, but does not currently report them to external clients.

For example, the attribute level in this class provides the detected signal level in dBm. You can use this attribute to indicate the strength of signal. More details:

<http://developer.android.com/reference/android/net/wifi/ScanResult.html#level>

- **WifiConfiguration**

This class representing a configured Wi-Fi network, including the security configuration.

- **WifiInfo**

Describes the state of any Wi-Fi connection that is active or is in the process of being set up. You can benefit from LINK\_SPEED\_UNITS to obtain link speed in Mbps.

- **WifiEnterpriseConfig**

You can find more details are available in:

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

#### 4. Some notes about connecting to a Wi-Fi AP which are helpful to accomplish tasks:

- *Enabling Wi-Fi Interface:* The Wi-Fi interface should be enabled before testing your program. To enable Wi-Fi manually, go to *Setting -> Wi-Fi -> Enabled*. You can do that via application by using `setWifiEnabled` from WifiManager class.

- *Take required permissions:* You need to add some privileges to the *Manifest File* in order to allow the application access to the Wi-Fi:

```
<uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"/>

<uses-permission
android:name="android.permission.CHANGE_WIFI_STATE"/>
```

- *Getting the WI-FI System Service:*

```
WifiManager mWifiManager = (WifiManager)
getSystemService(WIFI_SERVICE);
```

- *Enabling and disabling the WI-FI in:*

```
if (!mWifiManager.isWifiEnabled())  
if (mWifiManager.getWifiState() != WifiManager.WIFI_STATE_ENABLING)  
mWifiManager.setWifiEnabled(true);
```

- *Discover nearby access points:*  
You can use `startScan/getScanResults` methods from `WifiManger` class to discover the available access points.
- *Creating and managing Wi-Fi Connections:*

As we described in previous section, for creating configurations Android has provided a class named `WifiConfiguration`.

```
//Adding an OPEN network  
  
WifiConfiguration config= new WifiConfiguration();  
  
config.SSID="\"SSID_OF_NETOWRK\"";  
  
config.allowedKeyManagement.set(KeyMgmt.NONE);  
  
config.status=WifiConfiguration.Status.ENABLED;  
  
int netId=mWifiManager.addNetwork(config);  
  
mWifiManager.saveConfiguration();  
  
mWifiManager.reconnect();
```

Note: In this lab we using lab 02 program to connect *uniwide*. The following tasks should be done when your device is connected to *uniwide*. Before doing the task it is recomanded to have a look on the IEEE 802.11 in wikipedia: [https://en.wikipedia.org/wiki/IEEE\\_802.11](https://en.wikipedia.org/wiki/IEEE_802.11)

## Lab Tasks

### **TASK1 (0.5 Marks): Availability of 5 GHz connection on the phone**

Wifi is working with two frequencies 2.4 and 5 GHz. Find the method to check the availability (support) of 5 GHz wifi on your device. Show it as a text on the screen.

### **TASK2 (0.5 Marks): The frequency and bit rate of wifi connectin**

Check the frequency and bit rate of current connection and show it in GHz on the screen.

### **TASK3 (1 Marks): Check wifi protocol version**

Is there any way to find which 802.11 protocol version is used in your device connection to uniwide, for example (802.11a, c, ac, ...)? How about the connection modulation? You can get help from [Wikipedia](#). You should discuss and answer above questions with the tutor to get mark.