COMP4336/9336
Lab - WiFi-1

Lab Objectives

You will learn to work with some useful android Wi-Fi classes and methods. You will develop:

1. A program to scan, monitor and connect to Wi-Fi access points (AP).
2. A program to sort available nearby APs based on their signal strength.

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) 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.

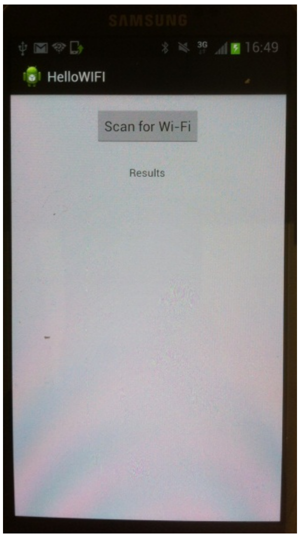
```
//Adding an OPEN network

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();
```

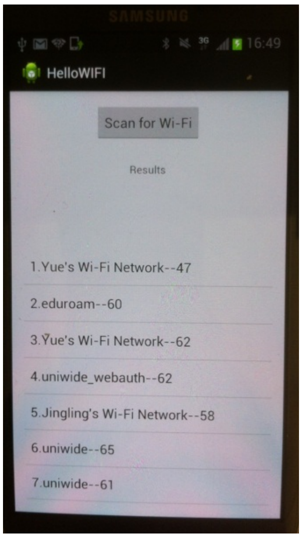
Note: In order to connect to an AP with *WPA* or *WPA2* security mode which needs authentication (providing username and password) such as *AP:uniwide* you have to change above code to set the username/password.

Lab Tasks

TASK1 (0.5 Marks): AP Scanner

Develop a program to scan all the nearby APs and show the ESSID, BSSID, signal strength, the encryption mode in a list view. For the encryption mode, we only distinguish if the AP is completely open (without any encryption) or not. The program should repeat the scanning process when the "SCAN" menu item/button is clicked.





1.Yue's Wi-Fi Network--47

2.eduroam--60

3.Yue's Wi-Fi Network--62

4.uniwide_webauth--62

5.Jingling's Wi-Fi Network--58

6.uniwide--65

7.uniwide--61

A simple view for task1.

TASK2 (0.5 Marks): AP Scanner

In task1, all physical AP with different signal strength and same name have been explored. For example, we have more than 5 AP with name "*uniwide*" and different signal strength. In this task, you have to show only four distinct stronger AP in the output.

TASK3 (1 Marks): Connect to a Specific AP

Based on the program developed in task2, add a new functionality to your program. The users can select one of the APs and connect to the AP by clicking a button (or menu item). When the AP is connected, pop-up a message box displaying the ESSID and IP address.

Note1: As all discovered APs need identity/password for connection, the *AP:uniwide* might be one option for this task.
Note2: Do not put your own username/password on your code. Instead of that try to develop a pop up dialog which will collect username/password for selected AP. In order to develop such dialog, these links could be helpful:
http://developer.android.com/guide/topics/ui/dialogs.html
http://techblogon.com/android-login-registration-screen-with-sqlite-database-example/#