COMP4336/9336 Lab 8

Device to device communication 2 (WiFi-Direct)

Lab Objectives

- Learn how to check the availability of WiFi-Direct on the mobile phone.
- Learn how to discover nearby devices with WiFi-Direct accessories.
- Learn how to connect to a nearby device such as a mobile phone or laptop via WiFi-Direct.

Preparation

- *Background*: WiFi-Direct, initially called Wi-Fi P2P, is a Wi-Fi standard that enables devices to connect easily with each other at typical Wi-Fi speeds without requiring any infrastructure such as wireless access point. It can provide a direct communication between two devices for all purposes such as transferring file or Internet connectivity. You can find numerous short introductions and tutorials of WiFi-Direct from the Internet (e.g. http://en.wikipedia.org/wiki/Wi-Fi_Direct).

- *Android facilities to manage WiFi-Direct*: The Android 4.0 (API level 14) or later lets devices with the appropriate hardware to connect directly to each other via Wi-Fi without an intermediate access point. The Wi-Fi P2P APIs consist of the following main parts:

  - **Methods** that allow you to discover, request, and connect to peers are defined in the *WifiP2pManager* class**.**

  - **Listeners** that allow you to be notified of the success or failure of *WifiP2pManager* method calls. When calling *WifiP2pManager* methods, each method can receive a specific listener passed in as a parameter.

  - **Intents** that notify you of specific events detected by the Wi-Fi P2P framework, such as a dropped connection or a newly discovered peer.

    More details:

    http://developer.android.com/guide/topics/connectivity/wifip2p.html

- A u*seful Android class* to do the lab tasks:

  - **WifiP2pManager**

This class provides the API for managing Wi-Fi peer-to-peer connectivity. This lets an application discover available peers, setup connection to peers and query for the list of peers. When a p2p connection is formed over WiFi, the device continues to maintain the uplink connection over mobile or any other available network for internet connectivity on the device. More details:

http://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager.html

- *Some useful notes* which might be helpful to accomplish tasks:

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

```
<uses-permission android:required="true"
    android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:required="true"
    android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:required="true"
    android:name="android.permission.INTERNET"/>
```

    Note: The element types in the manifest are ordered. Uses-permission needs to be first under the tag or outside of the application element otherwise the phone does not allow to access the WiFi-Direct interface and the program will crash.

  - *Checking hardware:*

    With this line of code, you can check whether your device supports WiFi-Direct or not:

```
WifiP2pManager WD_Manager = (WifiP2pManager) getSystemService(Context.WIFI_P2P_SERVICE);
if (WD_Manager.WIFI_P2P_STATE_ENABLED!=2) {//there is no WifiDirect interface}
```

  - *Initiate Peer Discovery:*

    You can use *discoverPeers()* methods from *WifiP2pManager* class to discover the available nearby devices. The function call immediately returns after sending a discovery request to the framework. The application is notified of a success or failure to initiate discovery through listener callbacks onSuccess() or onFailure(int).

  - *Connecting to a peer:* In order to connect to a nearby peer via WiFi-Direct you can use ***connect*** method from *WifiP2pManager* calss.

```
// coonect to device
```

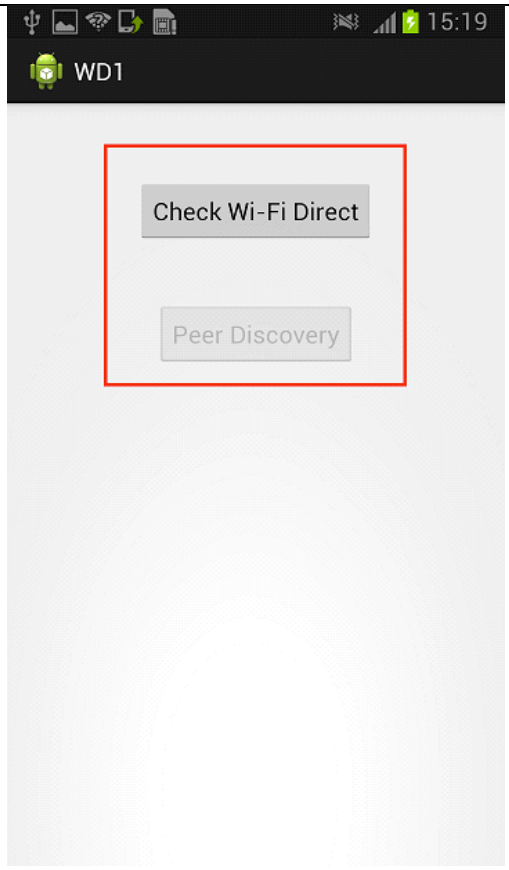
```
WifiP2pConfig config = new WifiP2pConfig();
onfig.deviceAddress = device.deviceAddress;

WD_Manager.connect(wifiDirectChannel, config, actionListener);
```

Lab Tasks

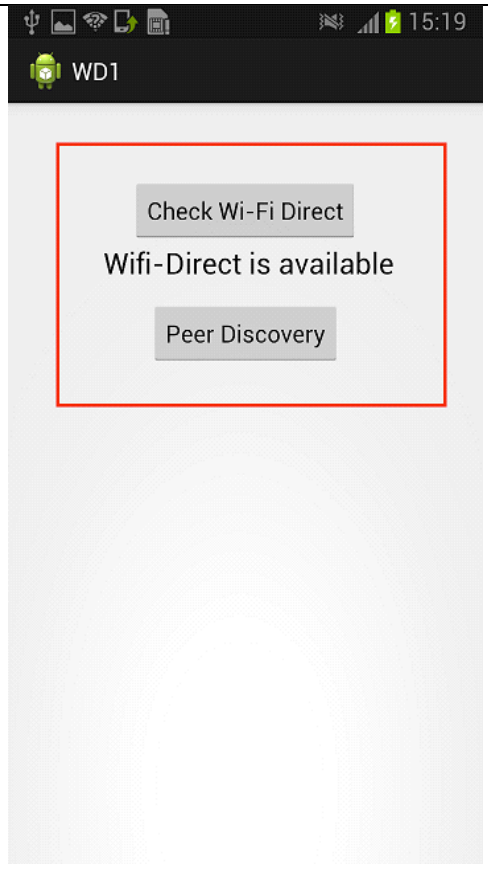
Task1 (0.25 Marks): Check the status of WiFi-Direct
Develop a program to check the availability of **WiFi-Direct** on your mobile phone. The status of the **WiFi-Direct** interface should be displayed in a *text view*.

Task2 (0.5 Marks): Peer discovery
Develop a program to search and discover all nearby **WiFi-Direct** devices and show the device name and MAC address in a *list view*. The program should repeat the discovery process when user clicks the "peer *discovery*" button.

Task3 (0.25 Marks): Connect to a nearby device.
Based on the program developed in task2, add a new functionality to your program in which user can select one of the discovered device and connect to that via WiFi-Direct after the target device confirms your request (A popup should be appeared in the target to get the confirmation of your peer). For that, you can ask one of your classmates to cooperate with you.

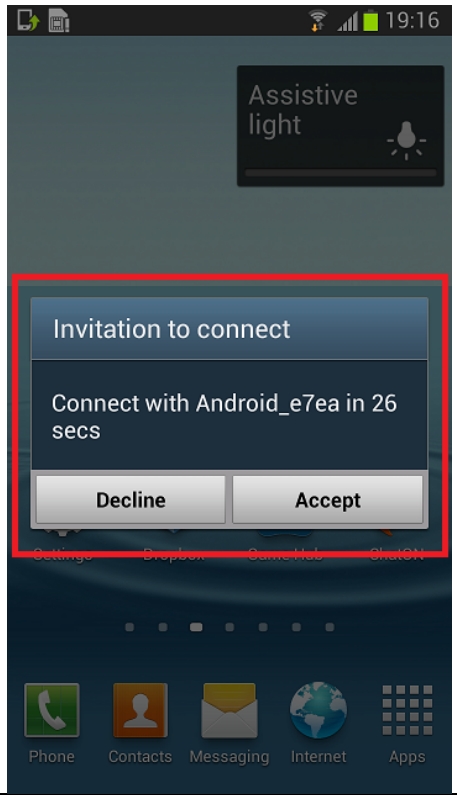


Task1 (1)



Task1 (2)

Task2



Task3