Android 4.0系统WiFiDirect功能详解

作者：aikongmeng　来源：安卓中文网　发布时间：2012/4/27 17:42:19

Android 4.0引入了一项很重要的技术就是 WiFiDirect (WiFi直连) ，它可以让WiFi设备无需热点即可实现两个WiFi设备的P2P数据交换。使用最新的Android 4.0 SDK，最低API Level 14才支持此项技术，在SDK的例子中我们可以看到很多界面用到了Android 3.0时代的Fragment容器。

首先我们需要实现android.net.wifi.p2p.WifiP2pManager.ChannelListener 接口来获取支持WiFi直连的Android设备。

```
public class WiFiDirectActivity extends Activity implements ChannelListener, DeviceActionListener {

public static final String TAG = "wifidirectdemo";

private WifiP2pManager manager;

private boolean isWifiP2pEnabled = false;

private boolean retryChannel = false;

private final IntentFilter intentFilter = new IntentFilter();

private Channel channel;

private BroadcastReceiver receiver = null;

public void setIsWifiP2pEnabled(boolean isWifiP2pEnabled) { //设置一个标记是否启用WiFi直连

this.isWifiP2pEnabled = isWifiP2pEnabled;

}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

//注册所需要处理的action，比如WiFi连接、状态改变等。
```

```
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION);
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION);
intentFilter.addAction(WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION);
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION);
    manager = (WifiP2pManager) getSystemService(Context.WIFI_P2P_SERVICE);
    channel = manager.initialize(this, getMainLooper(), null); //实例化WifiP2pManager对象
    }
    @Override
    public void onResume() {
    super.onResume();
    receiver = new WiFiDirectBroadcastReceiver(manager, channel, this);
    registerReceiver(receiver, intentFilter);
    }
    @Override
    public void onPause() {
    super.onPause();
    unregisterReceiver(receiver);
    }
    public void resetData() {
    DeviceListFragment fragmentList = (DeviceListFragment) getFragmentManager()
    .findFragmentById(R.id.frag_list);
    DeviceDetailFragment fragmentDetails = (DeviceDetailFragment) getFragmentManager()
    .findFragmentById(R.id.frag_detail);
```

```
if (fragmentList != null) {
fragmentList.clearPeers();
}
if (fragmentDetails != null) {
fragmentDetails.resetViews();
}
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
MenuInflater inflater = getMenuInflater();
inflater.inflate(R.menu.action_items, menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {
case R.id.atn_direct_enable:
if (manager != null && channel != null) {
startActivity(new Intent(Settings.ACTION_WIRELESS_SETTINGS));
} else {
Log.e(TAG, "channel or manager is null");
}
return true;
case R.id.atn_direct_discover:
if (!isWifiP2pEnabled) {
```

```
Toast.makeText(WiFiDirectActivity.this, R.string.p2p_off_warning,
Toast.LENGTH_SHORT).show();
return true;
}
final DeviceListFragment fragment = (DeviceListFragment) getFragmentManager()
.findFragmentById(R.id.frag_list);
fragment.onInitiateDiscovery();
manager.discoverPeers(channel, new WifiP2pManager.ActionListener() {
@Override
public void onSuccess() {
Toast.makeText(WiFiDirectActivity.this, "Discovery Initiated",
Toast.LENGTH_SHORT).show();
}
@Override
public void onFailure(int reasonCode) {
Toast.makeText(WiFiDirectActivity.this, "Discovery Failed : " + reasonCode,
Toast.LENGTH_SHORT).show();
}
});
return true;
default:
return super.onOptionsItemSelected(item);
}
}
@Override
```

```
public void showDetails(WifiP2pDevice device) {
DeviceDetailFragment fragment = (DeviceDetailFragment) getFragmentManager()
.findFragmentById(R.id.frag_detail);
fragment.showDetails(device);
}
@Override
public void connect(WifiP2pConfig config) {
manager.connect(channel, config, new ActionListener() {
@Override
public void onSuccess() {
// WiFiDirectBroadcastReceiver will notify us. Ignore for now.
}
@Override
public void onFailure(int reason) {
Toast.makeText(WiFiDirectActivity.this, "Connect failed. Retry.",
Toast.LENGTH_SHORT).show();
}
});
}
@Override
public void disconnect() {
final DeviceDetailFragment fragment = (DeviceDetailFragment) getFragmentManager()
.findFragmentById(R.id.frag_detail);
fragment.resetViews();
manager.removeGroup(channel, new ActionListener() {

@Override

public void onFailure(int reasonCode) {

Log.d(TAG, "Disconnect failed. Reason :" + reasonCode);

}

@Override

public void onSuccess() {

fragment.getView().setVisibility(View.GONE);

}

});

}

@Override

public void onChannelDisconnected() {

// we will try once more

if (manager != null && !retryChannel) {

Toast.makeText(this, "Channel lost. Trying again", Toast.LENGTH_LONG).show();

resetData();

retryChannel = true;

manager.initialize(this, getMainLooper(), this);

} else {

Toast.makeText(this,

"Severe! Channel is probably lost premanently. Try Disable/Re-Enable P2P.",

Toast.LENGTH_LONG).show();

}

}
```

```
@Override
public void cancelDisconnect() {
if (manager != null) {
final DeviceListFragment fragment = (DeviceListFragment) getFragmentManager()
.findFragmentById(R.id.frag_list);
if (fragment.getDevice() == null
|| fragment.getDevice().status == WifiP2pDevice.CONNECTED) {
disconnect();
} else if (fragment.getDevice().status == WifiP2pDevice.AVAILABLE
|| fragment.getDevice().status == WifiP2pDevice.INVITED) {
manager.cancelConnect(channel, new ActionListener() {
@Override
public void onSuccess() {
Toast.makeText(WiFiDirectActivity.this, "Aborting connection",
Toast.LENGTH_SHORT).show();
}
@Override
public void onFailure(int reasonCode) {
Toast.makeText(WiFiDirectActivity.this,
"Connect abort request failed. Reason Code: " + reasonCode,
Toast.LENGTH_SHORT).show();
}
});
}
}
}

}
```