**WiFiDirectActivity.java**

**package** xyz.fnplus.echo;  
  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.net.wifi.WifiConfiguration;  
**import** android.net.wifi.WifiManager;  
**import** android.net.wifi.p2p.WifiP2pConfig;  
**import** android.net.wifi.p2p.WifiP2pDevice;  
**import** android.net.wifi.p2p.WifiP2pManager;  
**import** android.net.wifi.p2p.WifiP2pManager.ActionListener;  
**import** android.net.wifi.p2p.WifiP2pManager.Channel;  
**import** android.net.wifi.p2p.WifiP2pManager.ChannelListener;  
**import** android.os.Bundle;  
**import** android.provider.Settings;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.Toast;  
**import** xyz.fnplus.echo.config.Configuration;  
**import** xyz.fnplus.echo.ui.DeviceDetailFragment;  
**import** xyz.fnplus.echo.ui.DeviceListFragment;  
**import** xyz.fnplus.echo.ui.PromptPasswordFragment;  
**import** xyz.fnplus.echo.wifi.WiFiBroadcastReceiver;  
**import** xyz.fnplus.echo.wifi.WiFiDirectBroadcastReceiver;  
  
  
**public class** WiFiDirectActivity **extends** AppCompatActivity  
 **implements** ChannelListener, DeviceListFragment.DeviceActionListener {  
  
 **public static final** String ***TAG*** = WiFiDirectActivity.**class**.getSimpleName();  
 **private** WifiP2pManager **manager**;  
 **private boolean isWifiP2pEnabled** = **false**;  
 **private boolean retryChannel** = **false**;  
  
 **private final** IntentFilter **intentFilter** = **new** IntentFilter();  
 **private final** IntentFilter **wifiIntentFilter** = **new** IntentFilter();  
 **private** Channel **channel**;  
 **private** BroadcastReceiver **receiver** = **null**;  
  
  
  
  
 WifiManager **wifiManager**;  
 WiFiBroadcastReceiver **receiverWifi**;  
 **private boolean isWifiConnected**;  
  
 **public boolean isVisible** = **true**;  
  
 **public void** setIsWifiP2pEnabled(**boolean** isWifiP2pEnabled) {  
 **this**.**isWifiP2pEnabled** = isWifiP2pEnabled;  
 }  
  
  
  
  
  
  
 @Override **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***main***);  
  
 **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**);  
  
 **if** (Configuration.***isDeviceBridgingEnabled***) {  
  
 **wifiManager** = (WifiManager) getApplicationContext().getSystemService(Context.***WIFI\_SERVICE***);  
  
  
  
 **if** (!**wifiManager**.isWifiEnabled()) {  
  
 Toast.*makeText*(getApplicationContext(), **"wifi is disabled..making it enabled"**,  
 Toast.***LENGTH\_LONG***).show();  
 **wifiManager**.setWifiEnabled(**true**);  
 }  
  
  
 **receiverWifi** = **new** WiFiBroadcastReceiver(**wifiManager**, **this**, **this**.**isWifiConnected**);  
  
 **wifiIntentFilter**.addAction(WifiManager.***SCAN\_RESULTS\_AVAILABLE\_ACTION***);  
 **wifiIntentFilter**.addAction(WifiManager.***NETWORK\_STATE\_CHANGED\_ACTION***);  
 **wifiIntentFilter**.addAction(WifiManager.***WIFI\_STATE\_CHANGED\_ACTION***);  
  
 registerReceiver(**receiverWifi**, **wifiIntentFilter**);  
  
  
 **this**.connectToAccessPoint(**"DIRECT-Sq-Android\_ca89"**, **"c5umx0mw"**);  
  
 }  
  
 **final** Button button = (Button) findViewById(R.id.***btn\_switch***);  
  
 button.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 Intent i = **new** Intent(getApplicationContext(), MessageActivity.**class**);  
 startActivity(i);  
 }  
 });  
 }  
  
  
  
 @Override **public void** onResume() {  
 **super**.onResume();  
 **receiver** = **new** WiFiDirectBroadcastReceiver(**manager**, **channel**, **this**);  
 registerReceiver(**receiver**, **intentFilter**);  
 **this**.**isVisible** = **true**;  
 }  
  
  
 @Override **public void** onPause() {  
 **super**.onPause();  
 unregisterReceiver(**receiver**);  
 **this**.**isVisible** = **false**;  
 }  
  
  
  
 **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**) {  
  
 **wifiManager**.startScan();  
 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** 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() {  
 }  
  
 @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();  
 }  
 });  
 }  
 }  
 }  
  
 **public void** displayConnectDialog(String ssid) {  
  
 PromptPasswordFragment ppf = **new** PromptPasswordFragment(**this**, ssid);  
 ppf.show(**this**.getFragmentManager(), ppf.getTag());  
 }  
  
 **public void** connectToAccessPoint(String ssid, String passphrase) {  
  
 Log.*d*(WiFiDirectActivity.***TAG***, **"Trying to connect to AP : ("** + ssid + **","** + passphrase + **")"**);  
  
 WifiConfiguration wc = **new** WifiConfiguration();  
 wc.**SSID** = **"\""** + ssid + **"\""**;  
 wc.**preSharedKey** = **"\""** + passphrase + **"\""**; *// "\""+passphrase+"\"";* wc.**status** = WifiConfiguration.Status.***ENABLED***;  
 wc.**allowedGroupCiphers**.set(WifiConfiguration.GroupCipher.***TKIP***);  
 wc.**allowedGroupCiphers**.set(WifiConfiguration.GroupCipher.***CCMP***);  
 wc.**allowedKeyManagement**.set(WifiConfiguration.KeyMgmt.***WPA\_PSK***);  
 wc.**allowedPairwiseCiphers**.set(WifiConfiguration.PairwiseCipher.***TKIP***);  
 wc.**allowedPairwiseCiphers**.set(WifiConfiguration.PairwiseCipher.***CCMP***);  
 wc.**allowedProtocols**.set(WifiConfiguration.Protocol.***RSN***);  
 *// connect to and enable the connection* **int** netId = **wifiManager**.addNetwork(wc);  
 **wifiManager**.enableNetwork(netId, **true**);  
 **wifiManager**.setWifiEnabled(**true**);  
  
 Log.*d*(WiFiDirectActivity.***TAG***,  
 **"Connected? ip = "** + **wifiManager**.getConnectionInfo().getIpAddress());  
 Log.*d*(WiFiDirectActivity.***TAG***,  
 **"Connected? bssid = "** + **wifiManager**.getConnectionInfo().getBSSID());  
 Log.*d*(WiFiDirectActivity.***TAG***, **"Connected? ssid = "** + **wifiManager**.getConnectionInfo().getSSID());  
  
 **if** (**wifiManager**.getConnectionInfo().getIpAddress() != 0) {  
 **this**.**isWifiConnected** = **true**;  
 Toast.*makeText*(**this**, **"Connected!!! ip = "** + **wifiManager**.getConnectionInfo().getIpAddress(),  
 Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 Toast.*makeText*(**this**, **"WiFi AP connection failed... ip = "** + **wifiManager**.getConnectionInfo().getIpAddress()  
 + **"("** + ssid  
 + **","** + passphrase  
 + **")"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
}

**MessageActivity.java**

**package** xyz.fnplus.echo;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** xyz.fnplus.echo.router.AllEncompassingP2PClient;  
**import** xyz.fnplus.echo.router.MeshNetworkManager;  
**import** xyz.fnplus.echo.router.Packet;  
**import** xyz.fnplus.echo.router.Sender;  
**import** xyz.fnplus.echo.wifi.WiFiDirectBroadcastReceiver;  
  
  
  
**public class** MessageActivity **extends** Activity {  
 **public static** AllEncompassingP2PClient *RECIPIENT* = **null**;  
  
 **private static** TextView *messageView*;  
  
  
  
 @Override **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***message***);  
  
 *messageView* = findViewById(R.id.***message\_view***);  
  
 **final** Button button = findViewById(R.id.***btn\_send***);  
 **final** EditText message = findViewById(R.id.***edit\_message***);  
  
 **this**.setTitle(**"Group Chat"**);  
  
 button.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 String msgStr = message.getText().toString();  
 *addMessage*(**"This phone"**, msgStr);  
 message.setText(**""**);  
  
  
 **for** (AllEncompassingP2PClient c : MeshNetworkManager.*routingTable*.values()) {  
 **if** (c.getMac().equals(MeshNetworkManager.*getSelf*().getMac())) **continue**;  
 Sender.*queuePacket*(**new** Packet(Packet.TYPE.***MESSAGE***, msgStr.getBytes(), c.getMac(),  
 WiFiDirectBroadcastReceiver.*MAC*));  
 }  
 }  
 });  
 }  
  
  
 **public static void** addMessage(String from, String text) {  
  
 *messageView*.append(from + **" says "** + text + **"\n"**);  
 **final int** scrollAmount =  
 *messageView*.getLayout().getLineTop(*messageView*.getLineCount()) - *messageView*.getHeight();  
  
 **if** (scrollAmount > 0) {  
 *messageView*.scrollTo(0, scrollAmount);  
 } **else** {  
 *messageView*.scrollTo(0, 0);  
 }  
 }  
}

**Packet.java**

**package** xyz.fnplus.echo.router;  
  
  
**public class** Packet {  
  
  
 **public enum** TYPE {  
 ***HELLO***, ***HELLO\_ACK***, ***BYE***, ***MESSAGE***, ***UPDATE*** }  
  
  
  
 **private byte**[] **data**;  
 **private** TYPE **type**;  
 **private** String **receiverMac**;  
 **private** String **senderMac**;  
 **private** String **senderIP**;  
 **private int ttl**;  
  
  
 **public** Packet(TYPE type, **byte**[] extraData, String receiverMac, String senderMac) {  
 **this**.setData(extraData);  
 **this**.setType(type);  
 **this**.**receiverMac** = receiverMac;  
 **this**.setTtl(3);  
 **if** (receiverMac == **null**) **this**.**receiverMac** = **"00:00:00:00:00:00"**;  
 **this**.**senderMac** = senderMac;  
 }  
  
  
 **public** Packet(TYPE type2, **byte**[] eData, String receivermac, String senderMac, **int** timetolive) {  
 **this**.setData(eData);  
 **this**.setType(type2);  
 **this**.**receiverMac** = receivermac;  
 **if** (**receiverMac** == **null**) **this**.**receiverMac** = **"00:00:00:00:00:00"**;  
 **this**.**senderMac** = senderMac;  
 **this**.**ttl** = timetolive;  
 }  
  
  
 **public byte**[] getData() {  
 **return data**;  
 }  
  
  
 **public void** setData(**byte**[] data) {  
 **this**.**data** = data;  
 }  
  
  
 **public** TYPE getType() {  
 **return type**;  
 }  
  
  
 **public void** setType(TYPE type) {  
 **this**.**type** = type;  
 }  
  
  
 **public static byte**[] getMacAsBytes(String maca) {  
 String[] mac = maca.split(**":"**);  
 **byte**[] macAddress = **new byte**[6];  
 **for** (**int** i = 0; i < mac.**length**; i++) {  
 macAddress[i] = Integer.*decode*(**"0x"** + mac[i]).byteValue();  
 }  
 **return** macAddress;  
 }  
  
  
 **public static** String getMacBytesAsString(**byte**[] data, **int** startOffset) {  
 StringBuilder sb = **new** StringBuilder(18);  
 **for** (**int** i = startOffset; i < startOffset + 6; i++) {  
 **byte** b = data[i];  
 **if** (sb.length() > 0) sb.append(**':'**);  
 sb.append(String.*format*(**"%02x"**, b));  
 }  
 **return** sb.toString();  
 }  
  
  
 **public byte**[] serialize() {  
  
  
 **byte**[] serialized = **new byte**[1 + **data**.**length** + 13];  
 serialized[0] = (**byte**) **type**.ordinal();  
  
 serialized[1] = (**byte**) **ttl**;  
  
 **byte**[] mac = *getMacAsBytes*(**this**.**receiverMac**);  
  
 **for** (**int** i = 2; i <= 7; i++) {  
 serialized[i] = mac[i - 2];  
 }  
 mac = *getMacAsBytes*(**this**.**senderMac**);  
  
 **for** (**int** i = 8; i <= 13; i++) {  
 serialized[i] = mac[i - 8];  
 }  
 **for** (**int** i = 14; i < serialized.**length**; i++) {  
 serialized[i] = **data**[i - 14];  
 }  
 **return** serialized;  
 }  
  
  
 **public static** Packet deserialize(**byte**[] inputData) {  
 TYPE type = TYPE.*values*()[(**int**) inputData[0]];  
  
 **byte**[] data = **new byte**[inputData.**length** - 14];  
 **int** timetolive = (**int**) inputData[1];  
 String mac = *getMacBytesAsString*(inputData, 2);  
 String receivermac = *getMacBytesAsString*(inputData, 8);  
  
 **for** (**int** i = 14; i < inputData.**length**; i++) {  
 data[i - 14] = inputData[i];  
 }  
 **return new** Packet(type, data, mac, receivermac, timetolive);  
 }  
  
  
 **public** String getMac() {  
 **return receiverMac**;  
 }  
  
  
 **public void** setMac(String mac) {  
 **this**.**receiverMac** = mac;  
 }  
  
  
 **public** String getSenderMac() {  
 **return this**.**senderMac**;  
 }  
  
  
 **public** String getSenderIP() {  
 **return senderIP**;  
 }  
  
  
 **public void** setSenderIP(String senderIP) {  
 **this**.**senderIP** = senderIP;  
 }  
  
  
 @Override **public** String toString() {  
 **return "Type"** + getType().toString() + **"receiver:"** + getMac() + **"sender:"** + getSenderMac();  
 }  
  
  
 **public int** getTtl() {  
 **return ttl**;  
 }  
  
  
 **public void** setTtl(**int** ttl) {  
 **this**.**ttl** = ttl;  
 }  
}