Lab 8 Measurement of WiFi Signal Strength

1.The Purpose

Get used to the WiFi system and accomplish the sampling and measuring of WiFi signal strength through programming in Android on smartphone.

2.The Main Content

Researchers are generally required to complete mobile terminal scanning to indoor WiFi Routers, especially in indoor positioning, WiFi access point selection and other popular fields. As the result of the dynamic characteristics of the wireless channel, the wireless signal received by mobile terminals is always unstable. As shown in Figure.1, the signal measured at a settled position from a certain router is distributed in a specific distribution(not a fixed value). Therefore, the signal should be measured repeatedly, though the position is fixed. Then, the result needs to be analyzed as random data in a certain distribution.

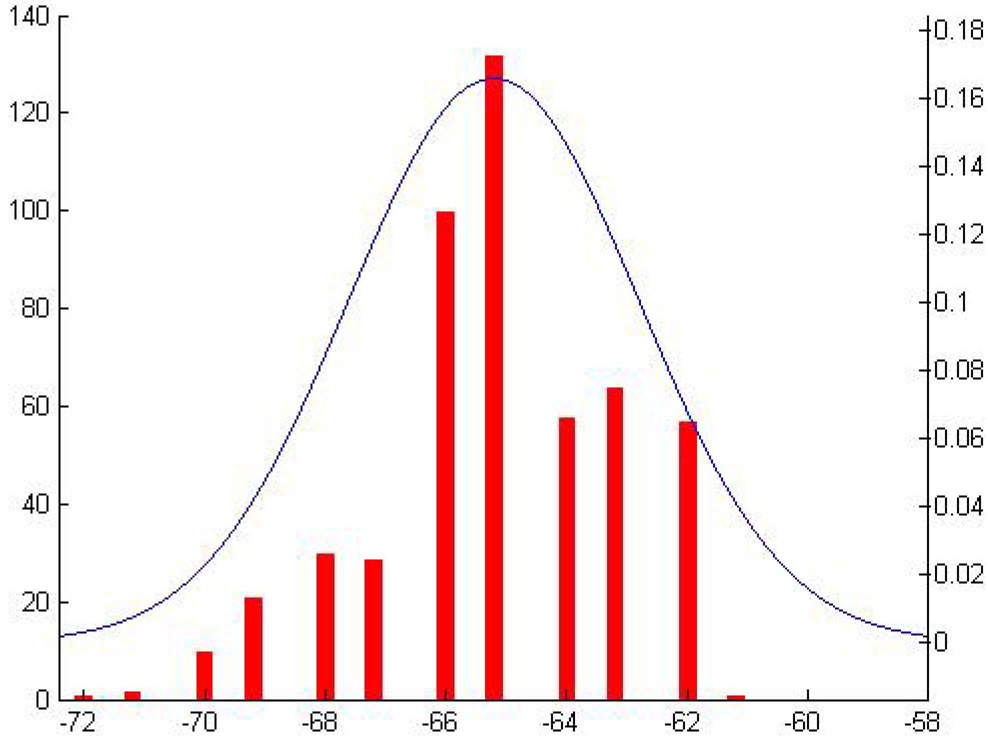

Figure 1 WiFi Signal Strength Distribution

Eclipse, etc is used as the compiler in the experiment to write the Android program. It will scan the WiFi Routers(named “IWCTAP1”, “IWCTAP2”, ... ,”IWCTAP10”) settled around the mobilephone continuously, record the signal strength, display the result on the mobilephone screen and write it to the SD card of the phone by the WiFi system of the Android.

Reference code is given:

(1) MainActivity

```
package com.example.adr_client;

import java.util.Vector;
import com.example.adr_client.R;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {

  private SuperWiFi rss_scan =null;
  Vector<String> RSSList = null;
  private String testlist=null;
  public static    int testID=0;//The ID of the test result
  @Override
     public void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main);

          final EditText ipText = (EditText)findViewById(R.id.ipText);//The
textlist of the average of the result
          final Button changactivity = (Button)findViewById(R.id.button1);
//The start button
          final Button cleanlist = (Button)findViewById(R.id.button2);//Clear
the textlist

      rss_scan=new SuperWiFi(this);
      testlist="";
      testID=0;

          changactivity.setOnClickListener(new Button.OnClickListener(){
           public void onClick(View v) {
               testID=testID+1;
               rss_scan.ScanRss();
               while(rss_scan.isscan()){//Wait for the end
               }
               RSSList=rss_scan.getRSSlist();//Get the test result
                final EditText ipText = (EditText)findViewById(R.id.ipText);
```

```
   testlist=testlist+"testID:"+testID+"\n"+RSSList.toString()+"\n";
                        ipText.setText(testlist);//Display the result in the textlist
                 }
                });

                cleanlist.setOnClickListener(new Button.OnClickListener(){
                 public void onClick(View v) {
                        testlist="";
                        ipText.setText(testlist);//Clear the textlist
                        testID=0;
                 }
                });
         }
   }
```

(2)SuperWiFi

```
      package com.example.adr_client;

      import java.io.File;
      import java.io.FileOutputStream;
      import java.io.IOException;
      import java.io.RandomAccessFile;
      import java.sql.Date;
      import java.text.SimpleDateFormat;
      import java.util.Iterator;
      import java.util.List;
      import java.util.Vector;
      import android.content.Context;
      import android.net.wifi.ScanResult;
      import android.net.wifi.WifiManager;
      import android.util.Log;

      public class SuperWiFi extends MainActivity{//The class of the parameters
of WiFi

            static final String TAG = "SuperWiFi";
            static SuperWiFi wifi = null;
            static Object sync = new Object();
            static int TESTTIME=25;//Number of measurement
            WifiManager wm = null;
            private Vector<String> scanned = null;
            boolean isScanning = false;
            private int[] APRSS=new int[10];
            private FileOutputStream out;
```

```
        private int p;

        public SuperWiFi(Context context)
        {
            this.wm = (WifiManager)
context.getSystemService(Context.WIFI_SERVICE);
            this.scanned = new Vector<String>();
        }

        public void ScanRss(){
            startScan();
        }

        public boolean isscan(){
            return isScanning;
        }

        public Vector<String> getRSSlist(){
            return scanned;
        }

        private void startScan()//The start of scanning
        {
            this.isScanning = true;
            Thread scanThread = new Thread(new Runnable()
            {
                public void run() {
                    scanned.clear();//Clear last result
                    for(int j=1;j<=10;j++){
                        APRSS[j-1]=0;
                    }
                    p=1;
                    //Record the test time and write into the SD card
                    SimpleDateFormat formatter = new SimpleDateFormat
("yyyy年MM月dd日       HH:mm:ss");
                    Date curDate = new Date(System.currentTimeMillis());
//Get the current time
                    String str = formatter.format(curDate);
                    for(int k=1;k<=10;k++){
                        write2file("RSS-IWCTAP"+k+".txt","testID:
"+testID+" TestTime: "+str+" BEGIN\n");
                    }

                    while(p<=TESTTIME)//Scan for a certain times
                    {
                        performScan();
                        p=p+1;
                    }
                    for(int i=1;i<=10;i++){//Record the average of the result
                        scanned.add("IWCTAP"+i+"= "
+APRSS[i-1]/TESTTIME+"\n");
                    }
                    for(int k=1;k<=10;k++){//Mark the end of the test in the
file

    write2file("RSS-IWCTAP"+k+".txt","testID:"+testID+"END\n");
                    }
                    isScanning=false;
                }

            });
            scanThread.start();
        }

        private void performScan()//The realization of the test
        {
            if(wm==null)
                return;
            try
            {
                if(!wm.isWifiEnabled())
                {
                    wm.setWifiEnabled(true);
                }

                wm.startScan();//Start to scan
                try {
                    Thread.sleep(3000);//Wait for 3000ms
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }

                this.scanned.clear();
                List<ScanResult> sr = wm.getScanResults();
                Iterator<ScanResult> it = sr.iterator();
                while(it.hasNext())
                {
                    ScanResult ap = it.next();
                    for(int k=1;k<=10;k++){
                        if (ap.SSID.equals("IWCTAP"+k)){//Write the result to
the file
                            APRSS[k-1]=APRSS[k-1]+ap.level;

    write2file("RSS-IWCTAP"+k+".txt",ap.level+"\n");
                        }
                    }
                }
                //this.isScanning=false;
            }
            catch (Exception e)
            {
                this.isScanning = false;
                this.scanned.clear();
                Log.d(TAG, e.toString());
            }
        }

        private void write2file(String filename, String a){//Write to the SD card
            try {
                  File file = new File("/sdcard/"+filename);
                  if (!file.exists()){
                      file.createNewFile();}
                // Open a random filestream by Read&Write
                RandomAccessFile randomFile = new
RandomAccessFile("/sdcard/"+filename, "rw");
                // The length of the file(byte)
                long fileLength = randomFile.length();
                // Put the writebyte to the end of the file
                randomFile.seek(fileLength);
                 randomFile.writeBytes(a);
                 //Log.e("!","!!");
                  randomFile.close();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
        }

    }
```

(3)Layout File

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
     >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="Scan" />

    <EditText
        android:id="@+id/ipText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/button1"
        android:layout_marginTop="14dp"
        android:ems="10"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/ipText"
        android:layout_marginLeft="15dp"
        android:layout_toRightOf="@+id/button1"
        android:text="clean" />

</RelativeLayout>
```

3.Questions

(1)Why is necessary to record all the measured value rather than only the average value?Please give your own explanation.

(2)Besides the WiFi signal strength, what other information of the Routers can be got in the test?

(3)Why does the scanning need to be operated in thread”scanThread”?

4.Additional Questions

Design an indoor positioning system based on WiFi signal strength by using the signal strength measuring function of Android mobilephone. Any positioning algorithm is available.