# 基站描述（二）

## 背景

NIST关于蜂窝数据的描述

1. CellularIdentity information and RF signal strengths measured from cellular network signals are recorded in files whose names end with "Cellular.pbs".  
   The Android operating system continuously scans for cellular signals and a scan cannot be triggered by a user-level application. Therefore, the system periodically (at 1 Hz rate) calls the getAllCellInfo() method, which returns all observed cell information from all radios on the device including the primary and neighboring cells as a list of CellInfo objects that are then written onto the file system as repeated fields of the relevant protocol buffer messages. Samsung Galaxy S6 was different from the other three devices, because the CellInfo objects did not contain valid values. As a work-around solution, we utilized the PhoneStateListener class to get cellular  
   signal related values from the registered network. This allowed us to obtain received cellular signal related information, but only for the network that the device was registered to.  
   Cellular output:The cellular network was continuously scanned by the operating system, and the latest scan results were  
   written into the file system at a 1 Hz rate. The protocol buffer messages contain the information that  
   CellInfo objects contain that are returned by the getAllCellInfo() method of the TelephonyManager class.  
   The name of this data file is { }\_Cellular.pbs.

## 具体API

参考自谷歌API开发文档（<http://www.android-doc.com/reference/android/telephony/TelephonyManager.html>）

public class TelephonyManager：类中有如下方法：

public List<CellInfo> getAllCellInfo ()

可以有效返回下列四个对象

CellInfoGsm, CellInfoCdma, CellInfoLte and CellInfoCdma

lte指4G

wcdma指联通3G

cdma指电信

gsm指移动和联通的2G

基站信息中都包含2个公共方法：

|  |  |
| --- | --- |
| [CellIdentityGsm](http://www.android-doc.com/reference/android/telephony/CellIdentityGsm.html) | [getCellIdentity](http://www.android-doc.com/reference/android/telephony/CellInfoGsm.html" \l "getCellIdentity())() |
| [CellSignalStrengthGsm](http://www.android-doc.com/reference/android/telephony/CellSignalStrengthGsm.html) | [getCellSignalStrength](http://www.android-doc.com/reference/android/telephony/CellInfoGsm.html" \l "getCellSignalStrength())() |

其中CellIdentityGsm中标识基站唯一识别信息：包含

|  |  |  |
| --- | --- | --- |
| int | [getCid](http://www.android-doc.com/reference/android/telephony/CellIdentityGsm.html" \l "getCid())() | CID Either 16-bit GSM Cell Identity described in TS 27.007, 0..65535, Integer.MAX\_VALUE if unknown |
| int | [getLac](http://www.android-doc.com/reference/android/telephony/CellIdentityGsm.html" \l "getLac())() | 16-bit Location Area Code, 0..65535, Integer.MAX\_VALUE if unknown |
| int | [getMcc](http://www.android-doc.com/reference/android/telephony/CellIdentityGsm.html" \l "getMcc())() | 3-digit Mobile Country Code, 0..999, Integer.MAX\_VALUE if unknown |
| int | [getMnc](http://www.android-doc.com/reference/android/telephony/CellIdentityGsm.html" \l "getMnc())() | 2 or 3-digit Mobile Network Code, 0..999, Integer.MAX\_VALUE if unknown |
| int | [getPsc](http://www.android-doc.com/reference/android/telephony/CellIdentityGsm.html" \l "getPsc())() | Integer.MAX\_VALUE, undefined for GSM |

CellSignalStrengthGsm是对应的基站强度信息：

|  |  |  |
| --- | --- | --- |
| int | [getAsuLevel](http://www.android-doc.com/reference/android/telephony/CellSignalStrengthGsm.html" \l "getAsuLevel())() | Get the signal level as an asu value between 0..31, 99 is unknown Asu is calculated based on 3GPP RSRP. |
| int | [getDbm](http://www.android-doc.com/reference/android/telephony/CellSignalStrengthGsm.html" \l "getDbm())() | Get the signal strength as dBm |
| int | [getLevel](http://www.android-doc.com/reference/android/telephony/CellSignalStrengthGsm.html" \l "getLevel())() | Get signal level as an int from 0..4 |

GSM：RSSI in dBm in the range of -51 to -113 or ASU in the range of 0 to 31。ASU 换算关系：2 \* ASU - 113 = dBm 。

UMTS：RSCP in dBm in the range of -25 to -121 or ASU in the range of -5 to 91。ASU 换算关系：ASU - 116 = dBm 。

LTE：RSRP in dBm in the range of -45 to -137 or ASU in the range of 0 to 95。ASU 换算关系：ASU - 140 = dBm 。

CDMA：RSSI in dBm in the range of -75 to -100 or ASU in the range of 1 to 16：

RSSI [dBm] >= -75: ASU = 16,

RSSI [dBm] >= -82: ASU = 8,

RSSI [dBm] >= -90: ASU = 4,

RSSI [dBm] >= -95: ASU = 2,

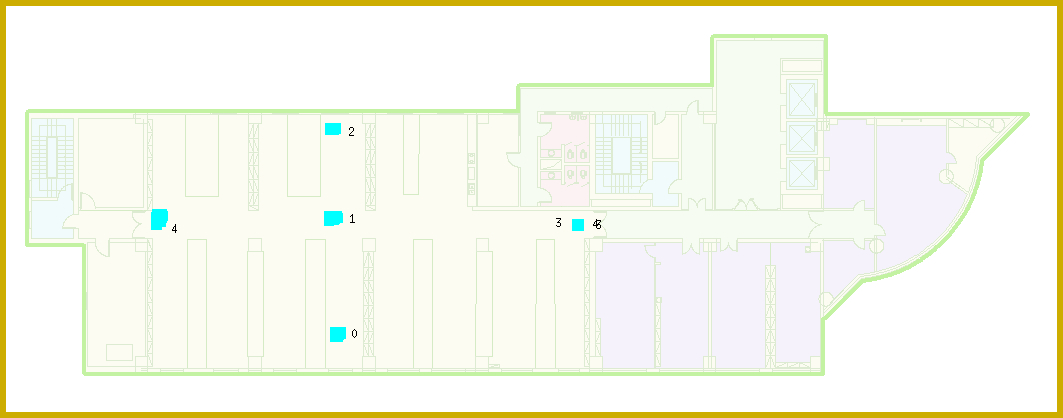
RSSI [dBm] >= -100: ASU = 1

## 开发测试（2天）

安卓测试软件开发完成后，测试采用了vivo和mate2款手机

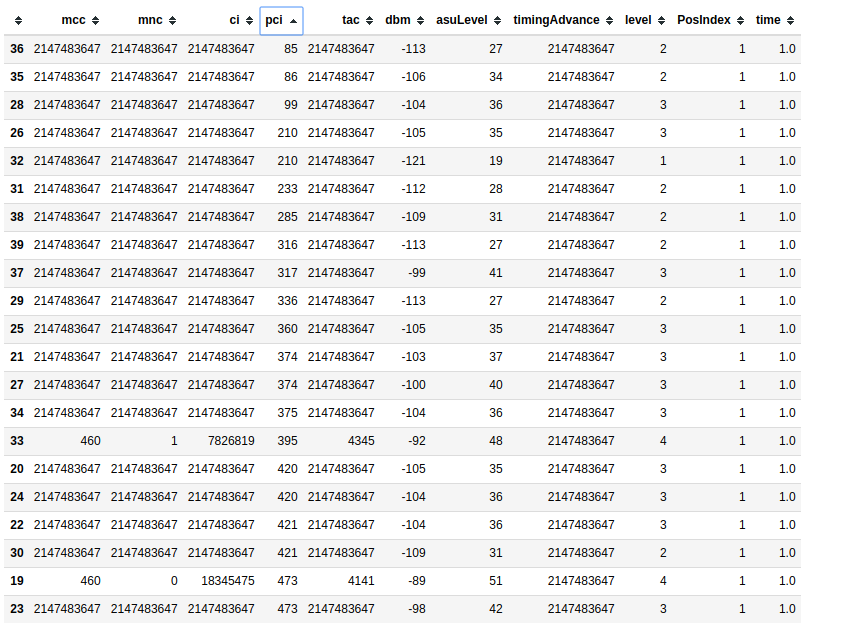
包含了移动和联通2卡

在大厦4层5个控制点，分别测试5次



## 测试结果分析（2天）

### 4.1 基站识别



基站由CI和PCI唯一指定，但是无法识别时会用最大整数代替，也就是2147483647.

这是在大厦的一次测试，21个测试结果中有19个是未识别的

由于采用了双卡双待的手机，所以移动联通各自有一个强度最高的基站信息

后文暂时使用pci来指定唯一，重复的认为是同一个

### 4.2 测试过程及数据分析

其中vivo1，mate1是同时间的两个设备

Vivo2，mate2则是第二天在同一位置采集的坐标

测试结果使用2个指标描述：

1. 准确度：对于同一位置的一次请求，生成的是如4.1的一串数据，将其中待匹配的两个周期的数据中PCI作为多维向量的维度，对应强度值DBM作为多维向量的值，就形成了每个周期自己的多维向量。

匹配相似度就使用了两个周期的多维向量，计算其余弦相似度作为相近程度的描述

而后，将匹配相似度最高的作为匹配位置，同时位置正确的作为匹配正确，准确度即描述描述在25次测试中匹配正确的概率

1. 平均误差：对于每次匹配，将匹配度最高的点（>0.75）对应的位置和待匹配位置求距离，而后将每个设备的25次测试距离求平均，得到匹配的平均误差，平均误差越小，说明匹配越准确。

匹配结果表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Vivo1** | **Mate1** | **Vivo2** | **Mate2** |
| **Vivo1** | **1.0,0.0** | **0.52,6.17** | **0.36,7.48** | **0.4,9.58** |
| **Mate1** | **0.44,7.088** | **1.0,0.0** | **0.36,9.59** | **0.6,5.09** |
| **Vivo2** | **0.44,11.91** | **0.32,13.54** | **1.0,0.0** | **0.12,16.12** |
| **Mate2** | **0.32,7.95** | **0.72,4.07** | **0.28,9.88** | **1.0,0.0** |

\*前后匹配度不相同是因为对于A匹配B是A中的一个周期数据，搜索B中的所有周期数据，匹配得到的最大值，而B匹配A则是B中的一个周期，搜索A中的所有周期，所以并不相同

使用均匀分布，随机匹配10次，得到的结果为：

|  |  |  |
| --- | --- | --- |
| **测试编号** | **准确度** | **匹配误差（m）** |
| **1** | **0.24** | **11.97808159** |
| **2** | **0.32** | **9.608430328** |
| **3** | **0.2** | **12.75665156** |
| **4** | **0.12** | **11.07010455** |
| **5** | **0.2** | **12.58724448** |
| **6** | **0.12** | **13.7357099** |
| **7** | **0.24** | **11.01522153** |
| **8** | **0.16** | **11.5289187** |
| **9** | **0.28** | **11.83117562** |
| **10** | **0.32** | **11.45808159** |
| **平均值** | **0.22** | **11.75** |

### 4.3 测试结果

* **在几乎相同时间采集的相同设备，匹配准确度极高，基站强度及其分布是非常稳定的，（平均准确度1，平均误差0m）**
* **在几乎相同时间采集的不同设备，匹配度会下降（平均准确度0.34，平均误差9.81m）**
* **在不同时间采集（隔日）的相同设备，匹配准确度也会下降,但是比上面的更好（平均准确度0.52，平均误差7.14m）**
* **在不同时间的不同设备，匹配度下降严重，效果最差（平均准确度0.35，平均误差10.16）**
* **随机匹配结果约为平均准确度0.22，平均误差11.76）**

### **4.4 测试结论**

仅针对本次测试的两款安卓设备，在29m\*14m的区域中，双卡手机定位测试效果而言：

* **基站定位短时效果（几小时内）对相同设备而言准确性非常高。**
* **基站定位对于相同设备，即使经过了一段时间后，定位效果仍然可用。**
* **基站定位对不同设备而言，精度下降明显，即基站信号在不同设备之间的变化远超相同设备时间变化带来的变化，基站定位的精度受设备变化影响严重。**
* **在单卡手机上，由于无法搜索非本网络制式内的基站，即联通只能搜索联通，移动只能搜索移动，导致接收的基站数量将进一步减少，匹配度也会进一步下降。**
* **更详细的结果和结论需要更进一步的测试！**

## 下一步工作（待定）

* **测试单卡安卓手机（移动、联通、电信）的定位效果（预期2日）**
* **测试安卓和IOS设备的测试结果（预期5日）**
  + **IOS采集设备的开发（2日）**
  + **同样面积大小的单卡双系统设备的测试（2日）**
  + **结果分析（1日）**
* **测试更大面积（1万平米），更多设备（安卓/IOS,单卡/双卡）的测试结果（预期8日）**
  + **单点单设备5次测试时间约2～5min**
  + **每25㎡测试约1个点**
  + **则1万㎡，测试3个设备，单卡及双卡测试日期约为400\*3/60\*2=40小时约6日**
  + **数据分析、处理及结果分析约2日**
* **更改当前匹配算法及重新测试（>5日）**
  + **调研其他方法**
  + **Demo及测试**
  + **开发及测试**
  + **现场测试**
  + **数据分析及结果处理**
* **......**