Sensor Data Network Using Android Mobile Devices

Chris Jordan G. Aliac
Intelligent Systems Laboratory, Cebu Institute of Technology, Cebu City, 6000 Philippines
chrisjord26@yahoo.com

Abstract. We have an insatiable desire to make sense of the world around us. How do we best observe and record the details of time, nature, location, events, and our own personal experiences? Data collection has always been a major challenge in sensor networks and various techniques have been proposed to enable efficient data collection. One such methodology is the use of mobile elements within the existing infrastructure to enable data collection. Smartphone’s are becoming ever more popular as they become cheaper to buy and very user friendly and easy to write applications. A wide range of these smartphones’ are based on the Android operating system. In this paper, we present a system that deploys lightweight Sensor Networks easily based on widely available Android Devices. An Android application is used to communicate with the on-board sensors of the mobile device, collate the variety of sensor readings to then be displayed on the device and sent to a central server for data logging and analysis. To ensure data reliability and redundancy, backup logs are saved within the sensor devices and then later sent up the server or other live sensor device via SMS message.

1 Introduction

Research into semantic sensor networks has been focusing on the treatment and processing of data aggregated from large networks of sensors, often based on specialized equipments geographically distributed in large areas, [1] discusses a number of challenges related to Semantic Sensor Networks in such scenarios. The challenge we are particularly interested in relates to the ability for rapid development of applications that make use of Sensor Network. We especially look at applications in scenarios where it is needed to set-up networks of simple sensors quickly and easily (e.g., school projects, small experiments). In recent years, the Android platform became a de-facto standard for different types of mobile devices from several manufacturers. The computational power of these devices already allows client processing of small to medium volumes of semantic data [2]. According to Gartner Press Release for November 10, 2010 [3], Android is one of the most popular smartphone platforms at the moment, and the popularity is even rising. It is one of the most open and flexible platforms providing software developers easy access to phone hardware and rich software API. Android mobile devices not shows as a powerful computing and communication mobile device of choice, but it also comes with a
rich set of embedded sensors, such as an accelerometer, digital compass, gyroscope, GPS, microphone, and camera. Collectively, these sensors are enabling new applications across a wide variety of domains, such as healthcare [4], social networks [5], safety, environmental monitoring [6], and transportation [7, 8], and give rise to a new area of research called mobile phone sensing. Until recently mobile sensing research such as activity recognition, where people’s activity (e.g., walking, driving, sitting, talking) is classified and monitored, required specialized mobile devices (e.g., the Mobile Sensing Platform [MSP]) [9] to be fabricated [10]. Mobile sensing applications had to be manually downloaded, installed, and hand tuned for each device. User studies conducted to evaluate new mobile sensing applications and algorithms were small-scale because of the expense and complexity of doing experiments at scale. As a result the research, which was innovative, gained little momentum outside a small group of dedicated researchers. Although the potential of using mobile phones as a platform for sensing research has been discussed for a number of years now, in both industrial [11] and research communities [12,13], there has been little or no advancement in the field until recently.

2 System Overview

![Fig. 1. Data Collection Overall Layout](image)

A sensor network is a data centric network which comprises distributed sites of
information represented by sensor nodes contributing data sensed [14]. They are producers of valuable data that needs to be collected in a cost efficient manner. The cost includes cost of communication, power consumption and time involved in communication. Data collection can be classified into Data Collection using static nodes and Data Collection using mobile elements. Since our focus is on mobile device based data collection, the literature looks into the use of mobile elements for data collection and not into traditional based multi hop and clustering based techniques. The idea of using mobility within sensor network to collect data from sensor nodes has been discussed by a number of researchers [15, 16, 17, 18, 19, 20, 21, 22]. The use of mobility to improve performance in Ad-Hoc networks has been proposed in ZebraNet project and Manatee project [3]. These mobile elements can be introduced or existing mobility in the environment can be used to achieve mobility. The use of mobility can be classified broadly as Random mobility, Predictable mobility and Controlled mobility. Kansal et al. [4, 12] presents a work on using controlled mobility for data collection in sensor networks. Shah et al. [3] and Sushant et al [12] present Data Mules that makes use of random mobility for data collection. Chakrabarti et al. [11] proposes a novel way of using predictable mobility to collect data from sensor networks that are distributed over an area.

2.1 Accelerometer Sensor

As stated in the above literature, Android devices contain several on-board sensors which can be easily acquired by writing an Android application. In this paper, the accelerometer sensor, which is always found on Android phones, will be demonstrated as a potential data on which to acquire. The devices’ accelerometer consists of 3 axes (X, Y, Z). Forces are applied to the sensor which yields results in m/s². Normally gravity acts upon the sensor and depending on the phones position, one of the phones axes will register 9.8 m/s² at stationary non-moving state.

\[ Ad = - \sum F_s / \text{mass} \]

Fig. 2. 3 Axis accelerometer forces
By measuring accelerometer data, can yield a lot of potential applications such as vibration sensing on a specific rigid structure, measuring road conditions traveled by car and measuring seismic activity to name a few.

2.2 Android SDK

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language [23].

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user. Underlying all applications is a set of services and systems, including:
A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser.

- Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data.
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.
- A Notification Manager that enables all applications to display custom alerts in the status bar.
- An Activity Manager that manages the lifecycle of applications and provides a common navigation backstack.

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

- **System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.
- **Media Libraries** - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view.
- **SGL** - the underlying 2D graphics engine.
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.
- **FreeType** - bitmap and vector font rendering.
- **SQLite** - a powerful and lightweight relational database engine available to all applications.

### 2.3 Data Acquisition Infrastructure

In this Sensor network, a server runs as the data acquisition for the different sensors. The System is connected via WIFI connection in which the server opens up a TCP socket port. The Android mobile sensors will act as clients connected to the TCP port. Given the simplicity of the way devices connect in such a way that TCP runs on any platform, servers may appear from different device types of different platforms. Server devices may appear as a desktop computer, a laptop or even a tablet PC. Platforms run from Windows based or even Linux or MAC Systems. In this paper, Windows platform is run using .Net C# as the programming language of choice.
3 Sensor Data Network Potentials

Although the system shows limitations to the number of sensors connected to the system. The architecture can be expanded to form a network of networks. TCP network servers can act as a large sensor array and can communicate to other TCP network servers. It can also be altered to become a Semantic Sensor Data Network by altering ports to Internet (port 80) and establishing Web servers for the TCP servers to send data or even better, make TCP server to WEB servers and alter the Android sensor apps to connect via Internet Web.

3.1 Reliability

Data from sensors must be sent to servers by means of a standard TCP socket. But there will be instances wherein TCP servers will be offline. At this condition, to ensure data will be sent to the server, Data will also be backed-up on the Android mobile sensors. Upon a substantial amount of time will be given for the server to be back online, when the server remains offline, data will be sent to other Android mobile sensors via SMS. A redundant SMS server can also be established online for the mobile sensors to transit data. In a worst-case failed network scenario.

4 System Results

Two Android devices were installed with a written application that acquires data from its accelerometer and connects to a server program running on the WIFI network. Data is transmitted in the form of raw float data from the 3 axis. Data received form the server is displayed onto the UI and options for saving the data is created. A graph is also displayed from the given realtime data in separate axis values.

4.1 Android Application program

The android application program was written using the Eclipse IDE. Since accelerometer is not possible in IDE Android emulator, an actual Android phone was connected and program was tested via USB connections an run on Android debug mode ADB.
4.2 Server Program

The server program was written using Visual C# express edition. Dot Net libraries provide a powerful socket connection library via TCP Listeners. The Server runs on port address 8000 and can easily be ported to port 80 internet connection.
4.3 Data Analysis

Since data is passed to the server as 3 xis data in the form below:

\[
\text{sensor1: X: -1.493966, Y: -0.306454, Z: 9.998084}
\]

Data can be analyzed by the sensor sending the data and the current real-time data by the 3 axis X, Y, and Z. The data shown above tells us that the phone's position is lying flat on its back by the Z data at 9.9 m/s\(^2\). It would lie face first if the data reflect -9.9 m/s\(^2\). Based on initial trials of data logged and saved, sensor data is very sensitive, and a simulated movement test was conducted. Data was taken, and absolute offset from 9.9 m/s\(^2\) was calculated. Results are shown below.

<table>
<thead>
<tr>
<th>Test</th>
<th>Data Logged</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5 to 0.9</td>
<td>Stationary</td>
<td></td>
</tr>
<tr>
<td>1 to 3</td>
<td>Mild movement</td>
<td></td>
</tr>
<tr>
<td>4 to 9</td>
<td>Severe Movement</td>
<td></td>
</tr>
<tr>
<td>Greater than 9</td>
<td>Abnormal movements</td>
<td></td>
</tr>
</tbody>
</table>
Although the table shown on the table does not show conclusive data due to the fact that no basis from actual data was used to conclude remarks. This table shows that raw data results can be easily analyzed for a specific study.

5 Conclusion

Based on the system results, this system can of potential use to different applications for different disciplines. The network system can be easily converted form a basic data sensor network where basic sockets are used on TCP, to a full Semantic sensor web that could gather a large array of data.

References

8. UC Berkeley/Nokia/NAVTEQ, “Mobile Millennium”: http://traffic.berkeley.edu/


