Proximity Data Gathering For Android Through Bluetooth

CSE 237B
Project Report

Submitted to:
Prof. Rajesh Gupta

Submitted by:
Ashwini Pullabhatla
Hayden Gomes

Department of Computer Science Engineering
University of California, San Diego
ABSTRACT
This project intends to implement a proximity data gathering system for the Android cellphone. This system is able to be extended to provide information for a variety of applications. As a demonstration of the system's versatility, the project's example application uses the system for basic social networking and time management scheduling. Bluetooth was selected as the wireless data harvesting method because of its presence in many new technologies including most cellular telephones. Bluetooth also has the advantage of being short range which will aid in restricting detections to a close proximity.
INTRODUCTION

Proximity detection is the ability to detect objects that are a relatively small distance away from the detector. Mobile phones have been using this spatial property for many applications. For instance, the background light of your mobile screen will dim when the proximity sensor in your mobile sees that the phone is closer to our ear. For the purpose of this project, we will be looking at proximity detection on a slightly larger scale by attempting to detect and identify devices within a range of meters rather than millimeters. Proximity information at these distances has a wide range of applications. A rather straight forward application is social networking by using people's devices to identify them and recommend adding people of reoccurring proximity to join your social network. Another possible application is a time management scheduler. In today's busy world it is useful to keep a time schedule either on a day-to-day basis or weekly or monthly. By placing known Bluetooth devices or dongles everywhere that a person may visit throughout their day, the collected proximity information can then be used to determine how much time the person spent in each location.

For the purpose of this project, Bluetooth was chosen for the communication medium that would be used to discover these devices. Bluetooth is the name for a short-range radio frequency (RF) technology that operates at 2.4 GHz. The effective range of Bluetooth devices can vary anywhere from 6 to 100 meters, however most Bluetooth devices including cellular phones, headsets, and computer peripherals are Class II 2.5mW with an effective range of 32 feet (10 meters). Bluetooth transfers data at the rate of 1 Mbps, which is from three to eight times the average speed of parallel and serial ports, respectively. Bluetooth has a number of properties that lead to this selection. It is present in a large variety of new technologies. Most importantly, it has become an unofficial standard in cellular phones. As more and more people carry these devices everywhere they go, their phones will provide a noninvasive way of tagging their identities for the purpose of social networking. Bluetooth is also a fairly short ranged communication medium. This limitation will help to limit the discovered devices to those in a close proximity.

We based our project to achieve two co-ordinates of the social networking circle, Meet and Influence. This project will guide the user in two ways: friend-finding and time-schedule monitoring.
DESIGN
This project involves two goals. Firstly, developing the proximity data collection system for Google's Android G1 development phone. This system is broken into four components, triggering mechanics, controlling the Bluetooth, storing the data, and alerting the parent application. When one of the triggering mechanics occurs, it will start the system scanning. After the scan is complete, all of the discovered devices will be identified. Without pairing, Bluetooth is able to determine a visible device's address, name, major and minor class, and the signal strength of the communication between the devices. This information will then be stored in a database. Finally, the parent application that is using the system needs to be alerted that a scan was completed and that new data is now available.

The second goal of this project is to develop an Android application that would demonstrate the functionality of the the system. The first part of the application is a time schedule for the mobile user. This will allow the user to check how much time he or she has spent in a particular location. With this information the user will be able to rework their habits to make their daily activity to reflect the schedule they want. For instance, most of today's jobs are highly competitive and the manager always needs to keep a track of how much time each employee has spent doing a particular task. The employee needs to submit a weekly status report stating the time spent doing various tasks throughout the week. Using this application, the user can have graphical data showing how much time he or she spent debugging an issue in lab, having meetings in the conference room, or working at their desk.

Secondly, our application will work to naturally develop your social network instead of manually having to instigate the communication/friend connection. It will scan all the Bluetooth devices and at the end of the day it will suggest us the top most frequently occurring devices. We call this friend-finding as it guides us to expand our social network by suggest new friends based on who we frequently are around.

IMPLEMENTATION
While Google has recently released Android version 2.0 which comes with a full Bluetooth API, the G1 does not support this version. Thus an experimental Bluetooth API for Android version 1.6 has to be used for this project. This API was developed by Stefano Sanna, an Italian computer scientist who has recently been focusing his work on developing Android applications. In addition to properly implementing this API package, the collection system implements two triggering mechanisms, a SQLite database, and a Java listener class.

The first is manually triggering by the parent application calling startScan( ). Application developers can use this to create their own triggering mechanics. However for the purpose of this project, this trigger has been directly linked to a menu button, allowing the user to initialize the scan simply by pushing it. The second trigger provided by the system is a timer that will start a scan every time the configured amount of time has passed. For the purpose of this application, the timer was fixed at scanning for devices every 30 minutes.
The SQLite database structure used in this system is based on the database in the Android notepad tutorial. Each entry in the database includes the device's Bluetooth address, the owner assigned name, the device class, the device type as indicated by the class, a discovery timestamp, and a Boolean indicating whether or not the device should be graphed.

The Java listener interface BluetoothScanListener provides a framework for the application developer to construct a listener. It contains methods that are triggered when the phone's Bluetooth is enabled, disabled, starts scanning, and finishes scanning. This can be useful for an application to know when to refresh the information it is displaying for a new scan, or when a scan cannot occur because the Bluetooth has been turned off.

**RESULTS**

We have successfully built the first version of our application on Android G1. Screenshots of our activity are shown below. The first image shows the device list tab. This tab displays all information stored in the SQLite database. Clicking any entry will update the "graph device" field of all entries for that device. The second image shows the application menu. The menu includes a "Toggle Bluetooth" button that will turn on/off the device's Bluetooth when clicked, a "Clear Data" button that deletes all entries in the devices table of our SQLite database, and a "Scan for Devices" button, which is the previously mentioned manual trigger mechanism.
The third image shows the top list tab. This is our application's implementation of the friend-finder for social networking. Here it displays the top 4 most discovered devices and their corresponding "Count" values, the number of times the device was discovered. The final image shows the graph tab. This is our time-schedule monitoring system which plots the amount of time spent around all of the devices selected in the device list tab. The Y axis is in hours, which is calculated based on the assumption that scans were triggered every 30 minutes.

FUTURE WORKS
In this project we have investigated an innovative vision of mobile apps based on proximity detection. Through this work, new interesting aspects of the original vision were revealed, leaving plenty of room for more research and development. We look forward to continuing/extending this work as follows.

- **Social network contact recommendation engine** - Applying our current friend finder system to one of the many popular social network applications for Android. This would require connecting device Bluetooth addresses with accounts on the network so that the discovered devices can be linked to their owner's accounts.

- **Advanced triggering mechanisms** - There is room for a variety of triggering mechanisms including GPS triggered, motion triggered, and time of day triggered. By
including more options for triggering the scan, we would be increasing the functionality of
the system allowing developers to apply it to even more creative applications.

n **Integrate GPS with data collection** - As a way of expanding the information
collected at each discovery time, the GPS coordinates could be stored with each entry in
the database. This would allow applications to provide a location along with each device
that could be useful in determining who the discovered device belongs to.

n **Upgrade friend-finder criteria** - The current friend-finder system looks for which
devices the user is around the most. This could be adapted to limit friends to specific types
of devices, say cellular phones, or only counting the number of unique times the friend
was nearby. This means if a device is around for 4 scans in a row, it would get 1 added
to it’s count. This may help eliminate friend suggestions for people who were in the same
room as you for a long time, but only once, such as strangers in a movie theater.

n **Upgrade time schedule calculation method** - The current time schedule calculation
is based on the assumption that scans happen every 30 minutes. As other triggering
mechanisms are implemented, this assumption will become increasingly inaccurate.
Therefore a new calculation that takes advantage of the timestamps associated with every
discovery should be used to determine how much time a device has been around.

CONCLUSION
We were Successful in implementing the proximity data collection system on the Android
platform using an experimental API. Also, were able to produced an example application which
demonstrates two of the potential uses of such a system. While there is plenty of room for
improvement and future development, we hope that our work has thus far shown that proximity
driven applications is possible using Bluetooth and that we have provided a good framework for
future applications to be built upon.

REFERENCES
1. Lin, Leonard. “Proximity and Relative Location”.

   wireless signal strengths”.


4. Sanna, Stefano. "Experimental Bluetooth API".