

# iBeacon: An Approach to Finding People in Crowds

Samuel Bae

Department of Computer Science  
Binghamton University

sbae1@binghamton.edu

Danny Yu

Department of Computer Science  
Binghamton University

dyu6@binghamton.edu

Julian Yuen

Department of Computer Science  
Binghamton University

julian@binghamton.edu

## Abstract

This paper presents the application of low-powered bluetooth technology as an approach to sensor detection. One such use would be locating people in crowded areas. Apple's Bluetooth low energy (BLE) protocol, iBeacon, was considered to be a possible solution for this problem. This low-powered Bluetooth transmitter was paired with an iOS application with a Ruby on Rails backend server to achieve our goals. However, what we thought to be easily configurable options turned out to be the complete opposite. Unfortunately, iBeacon turned out to be insufficient for low-power use and other means of low-powered transmitting should be explored. Although we have a working product of iBeacon technology, our anticipated goal was not reached.

## 1 Introduction

What are iBeacons? [1] Apple released a technology standard called iBeacon that allowed mobile applications to listen to other similar beacon signals and communicate with them. iBeacon can be said to be a positioning system that locates other beacons on a local scale and it is run on bluetooth low energy.

Bluetooth low energy (BLE) is a wireless network protocol that transmits data over short distances. Such communications typically do not exceed 30 meters. BLE is designed to be a low-energy equivalent to its predecessor, the Bluetooth. The cost of BLE is 60-80% cheaper than regular bluetooth and is preferred for simpler applications related to communication. As with the original bluetooth, bluetooth must be turned on in order for BLE to function.

iBeacon is also better than the global positioning system (GPS) on a micro level. While GPS uses the macro-location targeting, iBeacon operates on a micro scale. Moreover, GPS is rendered useless underground whereas iBeacon functions properly. The accuracy of iBeacon compared with the GPS

is also relatively more accurate due to its micro scale. Walls and obstructions affect GPS signals more so than iBeacons. Most importantly, iBeacon consumes less energy than GPS.

## 2 Purpose

Our problem of locating people in large crowds is important because GPS does not accurately pinpoint the location of the person you are trying to find. Some extreme examples that use this application would be concerts or finding your lost child during a large gathering. In such cases, our application would provide a seamless solution for finding a specific person in a crowd.

Currently, the only form of communication in large crowds is to call, text, yell, or look for the person. Our proposed approach was to use iBeacon so that people can search and be searched by others. Each individual is assigned a unique identification that iBeacon would be able to decipher and essentially connect people together.

This work heralds the age of mobile technology as a mini GPS. As long as the person you are attempting to search is within 30 meters, our product would be able to detect and lead you towards the signal. The product shall hopefully take off and become popular for all ages.

## 3 Related Work

On March 12, 2014, Philipp Kindt, Daniel Yunge, Robert Diemer, and Samarjit Chakraborty published a paper detailing a precise energy model for the BLE protocol. [2] They hoped to present a more accurate energy estimation for BLE because as of yet, none existed. Their optimization included their own algorithm that estimated the energy usage based on numerical integration and examining different values through the Gaussian distribution. This work showed us how we can improve the energy usage of BLE if iBeacon was configurable.

We took their findings and tried to use it to our advantage over the course of our project. We looked for potential energy savings through transmission costs but the paper argued that it was inconsequential. Since the slave, the detector, consumes more energy than the master, the emitter, we realize that we would have to fix the polling of the detector in order to minimize energy consumption. In addition, adding as much payload to the packet per event is desired so if possible, we would send as much data as we can for the signal in order to save as much energy as possible.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Binghamton University, May 17, 2014, Binghamton, New York.  
Copyright © 2014 ACM 000-0-0000-0000-0 ...\$10.00

## 4 Application

The crux of our application combines both the iOS application with the Ruby on Rails server to bring forth a complete product that would both emit and detect iBeacon signals. In essence, users can log in and register their unique iBeacon identification key through the Rails server. iBeacon will talk with this backend server using a one-on-one connection. As for the iOS application, it will either emit or detect the iBeacon signals accordingly. The Rails API would be able to communicate with the iOS application and determine the specific profile to search and/or identify.

### 4.1 Ruby on Rails API Backend Server

In order to get data, the iOS application will talk to the API to retrieve data. The capabilities of this server include the creation of users and well as signing in as a user. An authenticity token will be returned during the sign-in for authenticating the user. Furthermore, the user can request information from the API using said authenticity token. This API will be talking with the iOS application to send it the unique identification keys or keep track of account profiles.

#### 4.1.1 Source Code

The following figure is an example of the Ruby on Rails API.

### 4.2 iOS Application

The iOS application connects with the Rails API and will provide the graphical user interface for creating accounts and signing in. The main focus of the iOS application is to emit signals for detection by other users and detect signals to identify the approximate distance to the person you are trying to find. The color of the background changes depending on the distance between the two phones using the application. If multiple iBeacons are detected, the detector can choose which iBeacon signal color to examine.

#### 4.2.1 Source Code

What follows is an example of the Objective C code used to write the application.

## 5 Field Results

As you can see from our finished product, we are failures but we tried our best. Developers cannot easily configure iBeacon to their satisfaction. The only potential configurations that were possible were turning on and off manually and sleeping and waking up. Our final product only took color into consideration and only four colors were used to avoid complexity.

Our prospects for creating a compass or directional arrow were shot down because the iBeacon search cannot discover the direction the signal is coming from. However, it is possible to store data of direction change as detection occurs but this is beyond our capabilities.

## 6 Future Work

As we near the end of our paper, we would like to discuss some possible future work for this application. If multiple beacons were in the vicinity, trilateration/triangularization would seem feasible. Also, in the future, perhaps iBeacon would allow more exact configurations than the absolute basic options that are provided at this time.

## 7 Conclusions

In conclusion, our approach was sound but after realizing that iBeacon cannot be configured, our efforts were put to waste. On the bright side, we learned the basics of approaching a project for the first time without any prior knowledge of iOS application development or Ruby on Rails experience. Taking that into consideration, our project was successful in teaching us real-world application development.

## 8 Acknowledgments

We would like to thank Professor Ting Zhu of Binghamton University for his approval for this project. The paper he sent us allowed us to fully understand how estimations of BLE is still an ongoing issue. We would also like to thank the Internet for providing us with everything we needed to know in order to undertake this project.

## 9 Timeline

Research and software design took approximately one week. The creation of the backend server including user data and authentication API took about two weeks. The iOS application development took two weeks as well. Finally, the testing of our product took one week.

## 10 Roles of Each Person

Danny Yu was in charge of the Ruby on Rails backend server. He researched the topics relating to communicating with the iOS application and worked on the API.

Samuel Bae wrote the code for the iOS application. He connected the iOS application with the Ruby on Rails backend and designed the general layout of the graphical user interface.

Julian Yuen wrote this document in  $\text{\LaTeX}$  and helped with the Rails server and iOS application where necessary. More specifically, he learned basic Ruby on Rails and Objective C just to understand the inner workings of the application. He also served as leader of the team and gave directions as well as set deadlines.

## 11 References

- [1] iBeacon.com. What is ibeacon? what are ibeacons? March 2014.
- [2] P. Kindt, D. Yunge, R. Diemer, and S. Chakraborty. Precise energy modeling for the bluetooth low energy protocol. March 2014.