CHAPTER 1: INTRODUCTION

Bluetooth Low Energy (BLE) or Bluetooth Smart is an emerging technology that enables transmission of data over short distances. As the name suggests, BLE ensures low energy consumption and cost, besides offering a communication range similar to that of its predecessor, Regular or Classic Bluetooth. Majority of mobile operating systems such as Android, iOS, Windows Phone, BlackBerry, Linux, and Windows 8 natively support BLE [5].

The iBeacon is simply Apple's label for their Bluetooth LE beacons technology standard. While Apple has been aggressive in this industry since early 2013, it's worth noting that BLE technology works with Android devices from 4.3 and above. Mobile devices use ranging to estimate the distance from a beacon. A beacon doesn't provide a GPS coordinate, instead it estimates your proximity. So if your Mobile App detects multiple beacons, it can approximate where you are based on the distance to each beacon [8].

iBeacons are small wireless sensors that can be attached to any location or object. They broadcast tiny radio signals which the smartphones can receive and interpret, unlocking micro-location and contextual awareness. iBeacons are small but low-energy bluetooth (BLE) devices that we can be used for many problems like tracking and positioning indoors. A beacon is a low-cost piece of hardware that uses the Bluetooth Low Energy (often referred to as BLE or Blue Smart) technology to transmit a signal. They are small enough to attach to a wall or countertop and they transmit messages or prompts directly to mobile devices [8].

Estimote Beacons and Stickers are small wireless sensors that you can attach to any location or object. They broadcast tiny radio signals which your smartphone can receive and interpret, unlocking micro-location and contextual awareness. Technically speaking, Estimote iBeacon is a super small computer. It has a powerful 32-bit ARM® Cortex M0 CPU with 256kB flash memory, accelerometer, temperature sensor and what is most important – 2.4 GHz Bluetooth 4.0 Smart (also known as BLE or Bluetooth low energy)

bidirectional radio. It's called low-energy because it can run up to 2+ years with a single coin battery depending on the signal strenth and how frequent they broadcast information around. Using the compatible application installed on them, smartphones can easily determine their proximity to nearby objects and locations. This proximity information can be exploited to build a new generation of magical mobile applications that connect the real world with the smartphones [4, 6].

1.1 Main objectives of this project

I. To develop an understanding of working and applicability of BLE-enabled iBeacons.
II. To understand the Estimote SDK for Android in order to get started with Custom Android App Development.
III. To develop a Custom Android App that interacts with iBeacons via BLE. The App must offer Distance Estimation & Notification functionalities, besides automating the process of Recording the Distance & RSSI Readings for iBeacons.
IV. To record the Distance & RSSI Readings for Different Configurations of iBeacons using our Custom Android App
V. To devise a Localization Algorithm for iBeacon networks
VI. To propose Security Solutions for iBeacons

1.2 Need of this project

This project can be used by the organizations and entities those are planning to deploy iBeacons in their real-life applications. Our Custom built Android App, proposed Localization Algorithm and proposed security solutions, all together make this project highly relevant and useful for anyone planning to deploy and leverage the BLE-enabled iBeacons to upgrade their business. Further, this project can be benefitted from by the application developers who are enthusiastic regarding the development of iBeacon-enabled mobile applications.

1.3 Motivation behind this project

Bluetooth Low Energy (BLE), due its low energy consumption is being widely considered as the key enabler in the Internet of Things (IoT) ecosystem, thus connecting

billion of devices worldwide. Further, majority of mobile operating systems such as Android, iOS, Windows Phone, BlackBerry, Linux, and Windows 8 offer native support for BLE. Therefore, BLE-enabled iBeacons have a variety of applications in the field of retail, healthcare, education, transportation, hospitality, etc.

1.4 Application of this project

The iBeacons can be deployed to enable micro-location aware services including indoor positioning, in-store marketing and delivery of contextual aware information. Further, the iBeacon-enabled mobile applications can be used in several verticals such as retail, education, transportation, healthcare and tourism.

1.5 Thesis outline

This thesis mainly focuses on the working and applicability of BLE-enabled iBeacons, besides proposing a localization algorithm and security model for iBeacon networks. The first chapter introduces the concepts of Bluetooth Low Energy (BLE) and iBeacons, besides discussing the need, motivation and application of this project.

The second chapter presents a detailed study of Bluetooth Low Energy (BLE), the technology working behind the iBeacons. Further, this chapter contains a comparison of BLE and traditional Bluetooth.

The third chapter conducts a literature review, discussing the trends and directions of research being conducted in the domain of iBeacons. The fourth chapter provides a detailed study of iBeacons, including its working, functions, technical details, configuration settings, advantages and limitations.

The fifth chapter specifically focuses on working of Estimote iBeacons. The sixth chapter contains a comparison of iBeacon technology with other location-based technologies such as NFC and GPS.

The seventh chapter deals with development of Custom Android App that interacts with iBeacons. The eighth chapter provides the Distance & RSSI Readings for Different Configurations of iBeacons using our Custom Android App.

The ninth chapter deals with localization in wireless sensor networks, focusing on issues in localization algorithms and ranging techniques available for wireless sensor networks. The tenth chapter proposes a localization algorithm applicable in iBeacon networks.

The eleventh chapter discusses security concerns surrounding iBeacons, and then proposes several security solutions in order to achieve enhanced security in iBeacon networks. The twelfth chapter concludes the thesis and discusses the future scope of iBeacon technology.

Last of all, bibliography includes the references of different literatures those assisted in forming a sound understanding of the topic and timely completion of the project.

CHAPTER 2: LITERATURE REVIEW

Research is being extensively carried out to explore the viability of deployment of Bluetooth Low Energy (BLE) enabled iBeacons in several verticals including retail stores, airports and conferences. Maria Varsamou and Theodore Antonakopoulos have proposed that iBeacons can be deployed for achieving indoor positioning [7]. In this work, they have presented an Android-based application that analyzes iBeacon networks and determines the best signal map by collecting statistics related to spatial and temporal variation of RSSI values corresponding to an iBeacon network. Further, they demonstrated the application of presented Bluetooth Smart analyzer to measure the radiation pattern of iBeacon devices. Based upon experimental data, they conclude that the proposed methodology offers a useful and reliable tool for analyzing iBeacon devices and networks.

In today's modern world, shopping has assumed the importance of being an essential daily activity for a vast majority of people. But, their busy work schedule has reduced the time available to go out for shopping. This necessitates the need for simpler and quicker ways to do their shopping. Common difficulties associated with shopping include difficulty in finding relevant shops inside a shopping mall, people having to travel a long distance even without knowing the exact details and availability of the required items, difficulty in finding required items within a retail store. With the objective of overcoming these difficulties, a fully functional shopping mall application has been proposed in [11]. The application provides detailed information about all the shops inside a shopping mall, a map of shopping mall, list of all available items and customer wish lists. It offers an Android-based mobile application developed using a server side module which acts as the main database server and connects with shop owners and customers. This work exploits wireless communication technology, namely Bluetooth in order to derive the horizontal and vertical location coordinates of the customers.

In their work [13], Markus Köhne and Jürgen Sieck introduce the concept of an iBeacon, followed by a discussion of its working and possible real-life applications of iBeacon-

enabled location based services. Further, the authors discuss the restrictions that come with an iBeacon and suggest some improvements in order to remove these restrictions. Also, the authors analyze the deployment of iBeacons in luggage tracking and thus, evaluate its possibilities and restrictions. This work demonstrates the use of iBeacons in location-based service by an iOS-based application.

The advent of Bluetooth Low Energy (BLE) has provided new opportunities for indoor positioning applications. It supports small, low-cost, portable, battery-powered iBeacons, offering several advantages over WiFi. However, differing use of the radio bandwidth by the iBeacon technology poses some new challenges as well. In their work [12], Ramsey Faragher and Robert Harle explore the application of BLE-enabled iBeacons in BLE fingerprinting by deploying 19 iBeacons distributed across a $\sim$600 m^2 testbed with an objective to position a consumer device. The authors demonstrate the high susceptibility of BLE to fast fading, besides coming up with a strategy to mitigate this fading and quantify the true power cost of continuous BLE scanning. Further, the authors investigate the choice of key parameters in a BLE positioning system, such as iBeacon density, broadcasting power, and broadcasting frequency. Also, the authors present a quantitative comparison of BLE fingerprinting with WiFi fingerprinting. In this work, the results encourage the deployment of BLE beacons for indoor positioning.

In his work [9], Matthew S. Gast explores the usefulness of high-precision location information for mobile application developers, since it allows the smart devices to interact with the outside world. The author demonstrates how to achieve high accuracy with iBeacons. Further, the author discusses the development of applications using BLE-enabled iBeacons, so as to enable proximity and micro-location based services. Also, the author explores the use of APIs and built-in OS features to enable iBeacon-powered applications.

In recent years, the concept of smart buildings has gained much attention and is proving to be an effective solution to order to overcome the problem of huge power consumption of residential and commercial buildings. In this context, determining the exact location of users inside the buildings emerges as a much required feature to optimize the behavior of the building itself. In this work [15], the authors present BLUE-SENTINEL, an accurate

and power efficient method that identifies the occupants of each room of a smart building leveraging mobile devices as source of information. The proposed strategy solves the occupancy detection problem with a reasonably good accuracy and this is done by exploiting BLE-enabled iBeacon technology. Apart from being a power-efficient solution, the iBeacon technology ensures a good level of compatibility and portability, supporting both Android and iOS-based smart devices. Further, the authors validate the proposed solution in a real environment using a prototype system released as open source, thus demonstrating this technology to be a suitable solution to the occupancy detection problem inside a smart building.

Development of accurate indoor positioning solutions still remains an active but challenging research area. Several indoor localization solutions have been recently proposed, some of which even require additional infrastructure such as customized RF hardware. BLE-enabled iBeacon is one such emerging technology that is capable of offering an effective solution for accurate and scalable indoor localization. In this work [14], the authors present a suite of localization tools developed using the iBeacon standard, thus examining BLE's viability for application in indoor positioning solutions. This work demonstrates an average position estimation error of about 0.53 meters.

Discussing the real-life deployment of iBeacons, the Downtown app [18] needs a special mention, which can be used at Coupa Cafe located in downtown Palo Alto, California and other venues around the Bay Area. Here, the iBeacons have been installed on each table, thus the customers can conveniently place their orders with just a few taps on their smartphones, instead of having to wait in a long queue. Similarly, when it comes to payments, it requires just one click without having to wait for the billing.

Rover, a 500 Startups company, associated with developing a software platform for iBeacons, assisted the Toronto's craft beer capital, Bar Hop, to develop their own iBeacon-enabled mobile application. Using the application, users receive promotional offers as soon as they enter the venue. The functionality that makes it distinct than other existing coupon based iBeacon applications is that the users can the option to mark each offer as 'favourite' or 'not relevant'. This allows the Bar Hop to know about a user's likings, thus improving its own offerings [18].

CHAPTER 3: BLUETOOTH LOW ENERGY (BLE)

Bluetooth Low Energy (BLE) or Bluetooth Smart is a wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group (Bluetooth SIG) and is used for transmitting data over short distances. As the name suggests, it has been designed for low energy consumption and cost, while offering a communication range similar to that of its predecessor, Classic or Regular Bluetooth. Bluetooth Smart was originally introduced under the name Wibree by Nokia in 2006 [19]. Later on, it was merged into the main Bluetooth standard in 2010 during the adoption of the Bluetooth Core Specification Version 4.0. Mobile operating systems including Android, iOS, Windows Phone, BlackBerry, Linux, and Windows 8 natively support Bluetooth Smart. As per estimates put forward by Bluetooth SIG, more than 90 percent of Bluetooth-enabled smartphones will support BLE or Bluetooth Smart by 2018 [27].

3.1 How is BLE different from Traditional Bluetooth?

TABLE I: COMPARISON BETWEEN CLASSIC BLUETOOTH AND BLUETOOTH SMART

Technical Specification	Classic Bluetooth technology	Bluetooth Smart technology
Power Consumption	Higher energy consumption as compared to Bluetooth Smart technology	Bluetooth Low Energy, as the name implies, ensures lower energy consumption as compared to Regular or Classic Bluetooth. BLE-enabled device can last up to 2 to 3 years on a single coin cell battery.
Cost	Higher cost as compared to Bluetooth Smart technology	BLE is 60-80% cheaper as compared to traditional Bluetooth

Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
Active slaves	7	Not defined; implementation dependent
Latency	100 ms	6 ms
Minimum total time to send data	100 ms	3 ms [24]
Peak current consumption	<30 mA	<15 mA
Applications	Bluetooth is preferred for more complex applications that require consistent communication and greater data throughput.	BLE is ideal for simple applications that involve small periodic transfers of data.

3.2 How does BLE communication work?

This section describes the working of Bluetooth Low Energy (BLE) communication [21]:

- BLE communication primarily consists of “Advertisements”, i.e. small packets of data, those are broadcasted at a regular time interval by iBeacons or other BLE-enabled devices via radio waves.
- BLE Advertising is a one-way communication method. iBeacons that want to be discovered can “Advertise” or broadcast packets of data in fixed time intervals. These data packets are meant to be received by BLE-enabled devices such as smartphones, where they are used in several smartphone applications to trigger events such as push notifications, specific actions, or prompts.

- Apple's iBeacon standard suggests an optimal broadcasting interval of 100 ms. The more the broadcasting frequency, the more the energy consumption, but more frequent broadcasting ensures faster discovery by smartphones and other devices.
- BLE has a broadcast range of up to 100 meters, which make iBeacons an ideal choice for indoor positioning and location tracking.

3.3 Technical Details of BLE

This subsection deals with technical details of Bluetooth Low Energy (BLE) [24]:

- **Data Transfers**: Bluetooth Low Energy (BLE) transmits short data packets, ranging from 8 octets up to 27 octets those are transferred at 1 Mbps.
- **Frequency Hopping**: BLE employs the adaptive frequency hopping, that is common to all versions of Bluetooth technology in order to minimize interference from other technologies operating in the 2.4 GHz ISM Band.
- **Host Control**: BLE places significant intelligence in the controller, which allows the host to sleep for longer time periods and be woken up by the controller only when the host needs to trigger some specific action. This ensures huge energy savings as the host generally consumes more power than the controller.
- **Latency**: BLE is capable of supporting connection setup and data transfer as low as 3ms, thus allowing an application to establish a connection and transfer authenticated data within few milliseconds during a short communication period before quickly aborting the connection.
- **Range**: Increased modulation index provides a range of over 100 meters for BLE.
- **Robustness**: BLE employs a strong 24 bit CRC on all packets being transmitted, thus maximizing robustness against interference.
- **Security**: BLE implements full AES-128 encryption using CCM that provides strong encryption and authentication of data packets.
- **Topology**: BLE uses a 32 bit access address for every packet corresponding to each slave, thus allowing billions of devices to be simultaneously connected.

3.4 Advantages of BLE

- **Low power consumption**: Due to its innovative design, BLE consumes only a fraction of power of traditional Bluetooth. The power efficiency of BLE makes it the most suitable choice for devices needing to run off a tiny coin-cell battery for long time periods, extending up to 3 years [21, 24].
- **Connecting the Internet of Things**: Initially, Classic Bluetooth provided the means for device communication and manufacturers built hub devices like PCs, smartphones, and tablets to take advantage of these connections. According to ABI Research, the Internet of Things (IoT) ecosystem is slated to explode with a projected 30 billion devices by 2020. Analyst firms across the world recognize BLE as a key enabler in the Internet of Things [5].
- **Better Lifestyle**: The BLE- enabled devices are expected to make our lives better while allowing us to benefit from an ecosystem of BLE-enabled applications [5].
- **Application-friendly technology**: BLE is indeed an application-friendly technology supported by every major operating system. The technology offers flexible development architecture for developing applications in order to bring the everyday objects like heart-rate monitors, shoes and fridges into the connected world and have them interact with BLE-enabled applications.
- **Other advantages**: Additional benefits offered by BLE include lower implementation costs, enhanced range and multi-vendor interoperability [5, 24].

CHAPTER 4: ALL ABOUT iBEACONS

4.1 What are Beacons and iBeacons?

The term “Beacon” and “iBeacon” are often used interchangeably. A beacon is a low-cost piece of hardware that uses the Bluetooth Low Energy (BLE) or Bluetooth Smart technology to transmit a signal. These devices are small enough to be easily attached to an object or wall and they transmit identifiers directly to mobile devices [8, 21].

The term “iBeacon” is the Apple’s branding or label for their BLE-enabled beacons technology standard [8]. iBeacon is a protocol that was introduced by Apple at the Apple Worldwide Developers Conference in the year 2013 [20]. Several vendors have since started manufacturing iBeacon-enabled hardware transmitters, called ibeacons. While Apple has been aggressively working in this vertical since early 2013, but it’s pertinent to note that BLE technology is even compatible with Android devices starting from 4.3 and above.

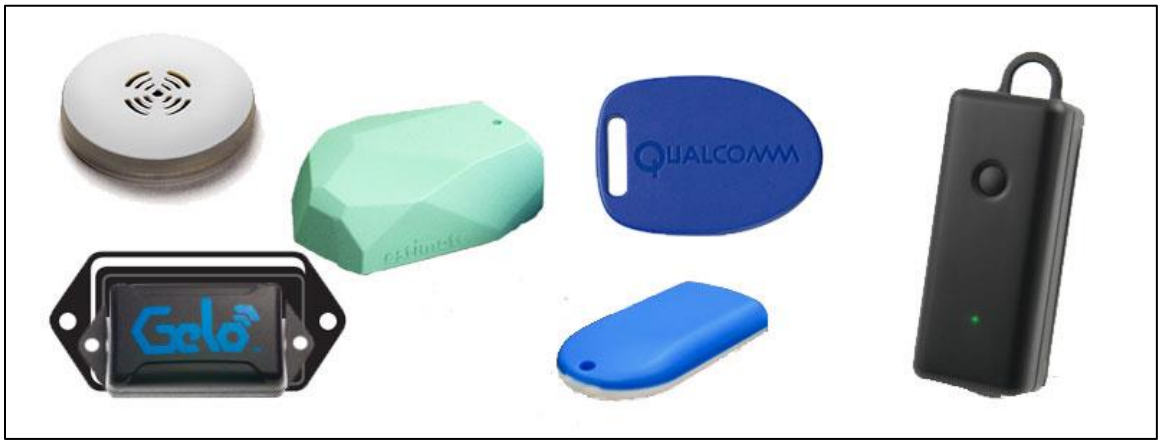

Fig. 1. iBeacon Devices

iBeacons are a class of BLE-enabled beacon devices that broadcast their location to nearby BLE-enabled mobile devices. The iBeacon technology allows smartphones, tablets and other devices to perform meaningful location-aware actions when in proximity to an iBeacon device. iBeacon uses BLE proximity sensing to broadcast a

universally unique identifier that can be used either to determine the device's physical location or to initiate a location-based service such as a push notification [29].

An iBeacon is defined by the following beacon properties [45]:

- **UUID (or proximity UUID)**: 128-bit unique identifier
- **Major**: 16-bit unsigned integer to differentiate among iBeacons within the same proximity UUID
- **Minor:** 16-bit unsigned integer to differentiate among iBeacons with the same proximity UUID and same major value.

The UUID may be specific to the organization (e.g. Heathrow Airport, London), whereas the Major number may denote specific subspace (e.g. Heathrow Terminal 3) and the Minor number may denote specific location (e.g. security, entry gate, restroom) [8].

One major application of iBeacons is to distribute context-aware information at a specific point-of-interest, for example, at a retail store, or library. This may be somewhat similar to GPS-enabled geopush technology, but has much increased precision and deals in "micro-location" (few metres down to a few centimetres) instead of GPS "location".

Another application of iBeacons is that of indoor positioning system which enables location tracking for smart phones by estimating their location. This suggests that an iBeacon can enable the smartphone's software to find its relative location to that iBeacon. Retail stores may deploy the iBeacons for delivering context-aware information such as special offers and deals to the customers, besides enabling mobile payments through point of sale systems [30].

iBeacon differs from most of the other location-based technologies as the iBeacon device is a one-way transmitter that transmits to the receiving device such as smartphone, and this necessitates the development and installation of a specific application on that device to enable its interaction with the iBeacons. Mobile devices use ranging to estimate their relative distance from an iBeacon. An iBeacon estimates your proximity instead of providing a GPS coordinate. Thus, if the specific application on your smartphone detects multiple iBeacons, it can approximate your location based on distance to each iBeacon.

4.2 How do iBeacons use BLE communication?

Apple has standardized the advertising format for BLE-enabled iBeacons. Under this format, an advertising packet broadcasted by an iBeacon consists of four major pieces of information [21]:

- **UUID**: This is a 128 bit or 16 byte string used to identify a large group of related iBeacons.
- **Major**: This is a 16 bit or 2 byte string used to distinguish among different subsets of iBeacons within the larger group.
- **Minor**: This is a 16 bit or 2 byte string used to identify individual iBeacons.
- **T_x Power**: This is used to determine proximity (relative distance) of listening device from the iBeacon. It can be defined as the strength of the radio signal exactly 1 meter from the listening device. This is often used as a baseline to provide an approximate distance estimate.

4.3 What is a beacon region?

Apple's Core Location framework, which includes the iBeacon standard, introduces the term beacon region. It is pertinent to note that iBeacon-enabled applications installed on smartphones can detect beacon regions, instead of individual beacons. Contrary to the common perception, a beacon region is not defined by geographic properties, such as GPS coordinates. Instead, it is defined by beacon identifiers: proximity UUID, Major and Minor. Thus, it can be inferred that the physical representation of a beacon region is the cumulative range of all the iBeacons present in that region. A beacon region may be defined in following three ways [26]:

- **With only UUID**: It consists of all iBeacons having a specific UUID. For example, a region defined with default Estimote UUID would consist of all Estimote iBeacons with unaltered UUID.
- **With UUID and Major**: It consists of all iBeacons having a specific combination of UUID and Major. For example, all Estimote iBeacons having default UUID and Major set to 1.

- **With UUID, Major and Minor**: It consists of only one iBeacon For example, iBeacon with default Estimote UUID, Major set to 1 and Minor set to 1.

Monitoring and ranging are two methods available for compatible mobile applications to scan for beacon regions, and thus detect iBeacons. Monitoring enables triggering relevant notifications or specific actions on applications based on entering and exiting a beacon region, whereas ranging method measures proximity to iBeacons in this beacon region [26].

4.4 Functions of iBeacons

An iBeacon network consists of one or more iBeacon devices that transmit their own universally unique identification number to the devices present within the range. Application on the receiving device may look up the iBeacon using its identification number and then perform various associated functions, such as delivering context aware information or notifying the user. Receiving devices may also connect to the iBeacons to retrieve values from iBeacon's GATT (generic attribute profile) service. iBeacons do not transmit any information to receiving devices other than their own unique identifiers to convey their identity. However, application on mobile devices can use identifiers transmitted by iBeacons to trigger some specific actions or push notifications.

4.4.1 Monitoring

Region monitoring is a term used to describe a BLE-enabled device's usage and detect when a user is in the vicinity of iBeacons. This functionality is used to display alerts or deliver context aware information as a listening device (and user) enters or exits an iBeacon region. Consider a departmental store that is identified by a specific set of proximity UUID and major value. Different sections of that store would be further differentiated by different minor values. An app can monitor region defined by that proximity UUID and major value to provide location-aware information as per different minor values [31].

Fig. 2. iBeacon: Monitoring

Monitoring a beacon region enables the app to know when the listening device enters or exits the range of iBeacons defined by the beacon region. Consider, for example, a museum with an audio guide application, two entrances and an iBeacon installed at each of those two entrances. The application monitors the region encompassing both of the iBeacons to get notified whenever the user enters or exits the museum. If the user has been detected to have entered the museum, then it may be presented with an app notification reminding that it can use the audio guide application to learn more about exhibits available in the museum. Moreover, this all may happen even if the user's phone is in their pocket in locked mode. This means that region monitoring can even function in the background of the listening device because region monitoring works even when the application is not running. This suggests that all what is required to activate region monitoring is the application installed on the mobile device.

Generally, region monitoring is limited to 20 regions and has different delegates to notify listening application and user of entry or exit in the beacon region. In case of iOS, region monitoring also allows the locked phone (or closed application) to react to the entry in a region. Coming to the technicalities related to region monitoring, iOS limits the number

of regions an app can simultaneously monitor to 20, but the number of iBeacons that can be monitored by a single region is almost unlimited (4,294,836,225 iBeacons). Further, since region monitoring works even if the application is not running on phone, it requires the user to grant the app permission to "Access Your Location Even When You Are Not Using the App" [23].

4.4.2 Ranging

Ranging functionality enables the application to know the relative distance between a listening device and iBeacons. Ranging returns to the listening device, an array of all discovered iBeacons along with their properties such as UUID, etc. [34]. Standard beacons have an approximate range of 70 meters whereas long-range beacons may reach up to 450 meters. The distance between listening device and transmitting iBeacon is generally categorised into 3 different ranges:

- **Immediate**: Within a few centimeters
- **Near**: Within a couple of meters
- **Far**: Greater than 10 meters

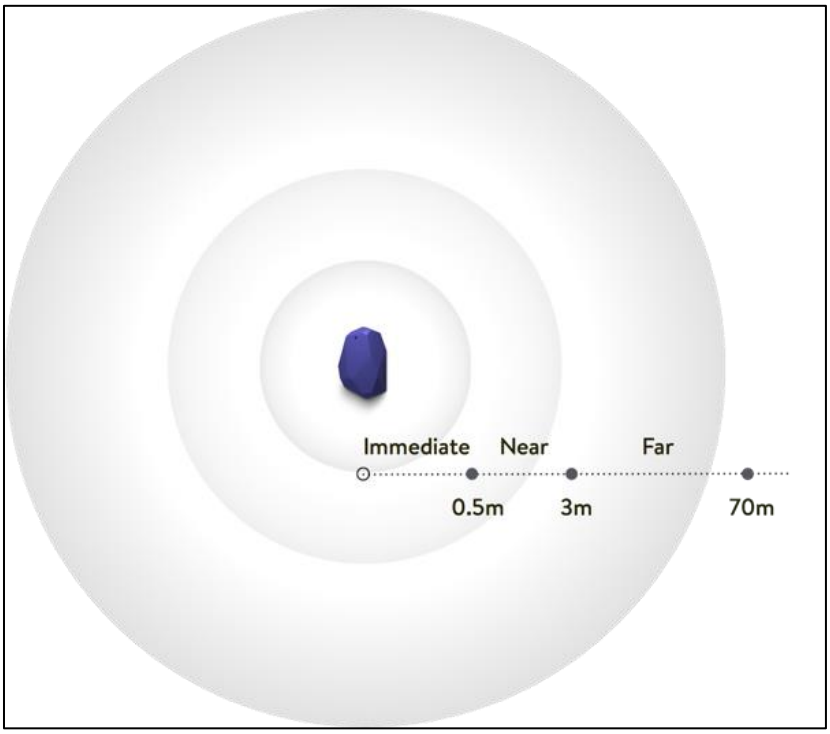

Fig. 3. iBeacon: Ranging

Consider for example an indoor location app of a retail store. Using the ranging technique, the app can determine which department such as grocery, clothing, footwear, accessories you're nearest to. Based on proximity, the app can be used to deliver location-based information about special offers or discounts.

As ranging depends on detection of radio signals, location or proximity estimates may vary depending on the positioning of iBeacons and whether a user's mobile device is in bag, pocket or in-hand. Clear line of sight between an iBeacon and mobile device will certainly deliver better results that may be achieved by ensuring that the iBeacons are not hidden between shelves. Further, in order to achieve consistent ranging, it is recommended that the user is holding the device in-hand (which means that the app is currently running) besides that the app is working in the foreground. Generally speaking, ranging is designed to be used for apps running in the foreground [23, 31].

Contrary to monitoring, ranging works only when the user is actively using the application on mobile device. This suggests that the application needs permission to "Access Your Location While You Use the App" [23].

4.4.3 iBeacon Settings

Configuration settings of an iBeacon include broadcasting power, advertising interval, etc. Both of these configurable have an effect on the battery life of the iBeacon. The frequency of the iBeacon transmission can be altered by varying the advertising interval. Although the iBeacons come with predefined settings, they can be configured by the developer. Further, Major and Minor values can also be changed for iBeacons. These values can be exploited if user wants to connect to specific iBeacons or if user wants to work with more than one iBeacon at once. Typically, multiple iBeacon deployment at any venue shares the same UUID, and uses the major and minor value pairs to segment and distinguish subspaces within that venue. Thus, the Major of all the iBeacons in a specific venue can be set to the same value and the Minor values can be used to distinguish and identify a specific iBeacon within that venue.

4.5 Technical details of iBeacons

Bluetooth low energy (BLE) devices may operate in an advertisement mode to notify nearby BLE-enabled devices of its presence [33]. An iBeacon is a BLE-enabled device that transmits an advertisement which follows a strict format, including an Apple defined iBeacon prefix, followed by a variable proximity UUID, a major value and a minor value [23]. An example iBeacon advertisement frame may look like: fb0b57a2-8228-44 cd-915a-94a142ba1307 Major 1 Minor 2, where fb0b57a2-8228-44 cd-915a-94a142ba1307 is the UUID. Mobile devices running the Android Operating System prior to version 4.3 can only receive iBeacon advertisements but cannot emit iBeacon advertisements, whereas Android version 4.3 and above can receive as well as emit iBeacon advertisements. Further, Android 5.0 ("Lollipop") has offered support for both central and peripheral modes [37].

4.6 Compatible devices

- iOS devices with Bluetooth 4.0, including iPhone 4S and iPad (3rd generation) and iPad Mini (1st generation) and iPod Touch (5th generation)
- Android Operating System 4.3 and above (e.g. Samsung Galaxy S4, Samsung Galaxy Note 3, HTC One, Google Nexus 7 2013 version, Nexus 4, Nexus 5)
- Windows Phone with the Lumia Cyan update or above (Windows Phone 8.1 is not supported)
- Macintosh computers with OS X Mavericks (10.9) and Bluetooth 4.0

4.7 Broadcasting range and proximity

iBeacon can be thought of as small lighthouse tower that's deployed at a fixed location and broadcasts its presence at regular time intervals to all the ships (smartphones) in range. The broadcasting range extends from 2 inches to approximately 70 metres. However, the exact range depends upon the surrounding environment that often includes several objects causing interference for radio signals transmitted by iBeacons. BLE-enabled iBeacons transmit the same type of radio waves as 2.4 Ghz Wifi routers. Thus, the radio signals can be interfered, diffracted or absorbed by objects or water [6].

Mobile devices such as smartphones present within the radio range can receive the Bluetooth radio signals transmitted by iBeacons, even without initial pairing and estimate their relative distance from the iBeacons by measuring the received signal strength (RSSI). This suggests that the closer the listening device is from the iBeacon, the more is the signal strength. Depending upon configuration, mobile devices can probe the signal once every second or even 5 times a second. The more is the advertising interval of iBeacons, the more stable the signal is and the more responsive the mobile application is, thus much better customer experience can be achieved [6].

4.8 Unique identifiers of iBeacons

Smartphones can receive and interpret signals emanating from more than one iBeacon simultaneously. Suppose, if there are three or more iBeacons within radio range, then smartphones can calculate their distance to each iBeacon and thus, leverage this information to estimate their relative location or proximity [6].

Further, iBeacons can be uniquely identified as each iBeacon broadcasts its universally unique identifiers. Every identifier (ID) is 20 bytes long and is divided into three parts: proximityUUID (16 bytes) + major number (2 bytes) + minor number (2 bytes). An example ID: c1d350a010a702415add1040f3813c301004. This may be thought of as an IP address of each iBeacon device [6].

4.9 Context and micro-location from the cloud

Knowing the unique ID of an iBeacon and the approximate distance (based on RSSI) to the iBeacon, micro-location and context can be easily derived. Based on micro-location, different actions can be triggered on user's smartphone. Since an iBeacon has limited flash memory that is incapable of storing anything except for its identifiers (ID), relevant and context-aware content (such as image of a shirt) needs to be fetched and retrieved from local storage or cloud storage [6].

4.10 Geofencing

This is of much interest as applications installed on smartphones can subscribe in the operating system (both Android and iOS) in order to constantly listen for particular identifiers. As the smartphone enters the range of an iBeacon, the application gets notified about this, even when the application is not currently running or the smartphone is locked [6].

4.11 Monitoring distance and triggering events

The compatible application installed on smartphone constantly monitors the signal strength or RSSI of iBeacons, which is compared to three predefined ranges that divide the radio range or area around an iBeacon into three zones: far, near and immediate. The application is also notified about the zone where the phone is currently present. Whenever the smartphone (user) enters a distinct zone, the context changes and the OS notifies the application about this event [6].

4.12 Advantages of iBeacon Technology

iBeacon, being a proximity technology that uses Bluetooth Smart or Bluetooth Low Energy, is best suited for providing indoor positioning based applications that require smart devices to perform actions based on relative proximity to iBeacons. Thus, this technology should be deployed if distribution or restriction of technology is required based on location of user. Following are some of the benefits of deploying iBeacon technology [32, 41]:

- **Accuracy**: Since iBeacon is a micro-location or proximity based technology, it is most suitable for accurate indoor positioning. Whenever a listening device senses an iBeacon in its vicinity using Bluetooth Smart, corresponding actions can be triggered via the compatible application installed on the device.
- **Privacy**: iBeacons do not constantly track the location of users, instead simply logging only when users come into radio range of an iBeacon, thus users can feel confident that their privacy is not being compromised and that the iBeacon network is not tracking them.

- **Integration**: iBeacon being a native Apple technology, scales effortlessly and integrates seamlessly via compatible application installed on smartphones.
- **Affordability**: Even a smart device such as an iPad can be easily configured as an iBeacon via a number of freely available applications, hence the technology is considered to be accessible, affordable and scalable.
- **Usability**: iBeacons use Bluetooth Smart or Bluetooth Low Energy (BLE), hence there is minimal impact on the users' devices and their battery life as BLE minimizes energy consumption. Since, most of the modern smartphones have been equipped with BLE, the iBeacons can easily connect with smartphones without requiring any additional hardware.

4.13 Limitations of iBeacons

- Most applications that use iBeacons are relevant to specific locations. This needs to be kept in mind while designing an iBeacon-powered application. The application developer who writes the code and configures the iBeacons should be aware of exact usage and venue where the application is to be used. If it is not possible for the application developers to conduct on-site testing, then a detailed description and photos of the venue must be sent to them [18].
- There are significant limitations to the placement of iBeacons as well. For example, museums tend to be strict regarding beacon placement. Some of the concerns that need to be considered are: crowd in the museum at peak times and existence of obstacles that an iBeacon's signals need to pass through. These concerns are indeed crucial for smooth interaction with iBeacon-enabled applications. However, there is no universal set of rules regarding how to position iBeacons while they are being deployed in a specific venue. Each use case and venue is different, and hence the exact positioning of iBeacons shall vary accordingly [18].
- Spotting a device's proximity to an iBeacon with accuracy is not easy. In other words, accurate location estimation of a device via iBeacon-enabled application is a challenging task. The estimation of exact location of a device is termed as localization. Devising a localization algorithm for iBeacons is a research problem that shall be worked upon in our project. The broadcast radius of a BLE-enabled

iBeacon typically forms a “zone.” A smartphone can use the BLE signal to determine its relative distance from the iBeacon. Although, the iBeacon-enabled application installed on the smartphone can notify a user’s entry or exit from a beacon region or say, zone without a directional antenna, it can’t determine whether the smartphone (and user) is located behind or in front of the iBeacon [39].

- Locked phones have limited effect on users’ beacon experience - After the iOS 7.1 update, it is evident that iBeacon is the sole BLE-enabled service that remains active even when a phone is locked. But, Apple has rendered iBeacon very limited in capabilities when a phone is in locked state. This can be supported by stating that instead of recognizing more specific ranges like immediate, near, and far, the iBeacon just tracks a locked phone’s entry into a zone, suggesting that the application developers have only one “entry” point to trigger a relevant action [39].

CHAPTER 5: WORKING WITH ESTIMOTE iBEACONS

5.1 What are Estimote iBeacons and Stickers?

Estimote iBeacons and Stickers are small Bluetooth Low Energy (BLE) enabled wireless sensors that can be attached to any object or location. These devices broadcast radio signals at regular fixed time intervals which can be received and interpreted by smartphones, thus enabling micro-location and contextual awareness based services. They are certified as iBeacon compatible and in a sense, can be considered as tiny computers. Estimote iBeacons are switched on, when they arrive and can't be switched off as there is no power on/off switch. Estimote claims their iBeacons to be ready for real-life deployment at scale [4]. Besides Estimote, there are several other vendors such as Roximity [22] those are aggressively working in manufacturing and deployment of iBeacons.

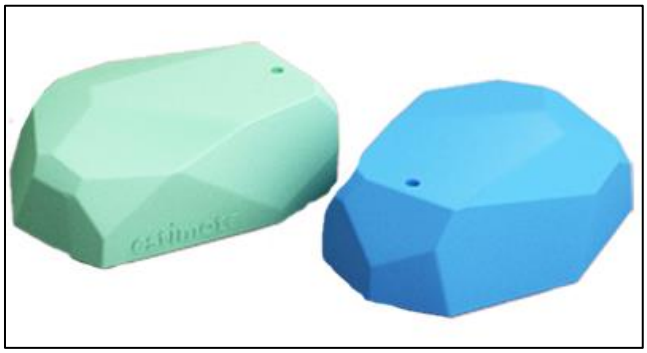

Fig. 4. Estimote iBeacons

Technically speaking, Estimote Beacon can be viewed as a super small computer. It has a powerful 32-bit ARM® Cortex M0 CPU along with 256kB flash memory, temperature sensor, motion sensor and most importantly, 2.4 GHz Bluetooth Low Energy (also known as Bluetooth 4.0 Smart) module, powered by a lithium coin battery. Bluetooth Smart is a completely redesigned Bluetooth standard developed by Nokia™ that is now implemented in all modern smartphones like Samsung™ Galaxy S III or Apple iPhone™ 4S, and other smart devices like Google Glass™ or Fitbit trackers. It is termed as low-

energy because it can run up to 2+ years with a single coin battery, actual lifetime depending upon broadcasting power (signal strength) and advertising interval [6].

5.2 How do Estimote iBeacons work?

Estimote Beacons broadcast radio signals at regular time intervals. BLE-enabled mobile devices in range receive these signals and compatible applications installed on smartphones can interpret and respond to these signals in form of push notification or other corresponding action. Using the Estimote SDK, an application on smartphones can estimate their relative location or proximity to nearby objects and locations, thus recognizing their ownership, type, temperature, motion and approximate location. Estimote SDK and Estimote Cloud grant full access to the metadata, including object type, beacon ownership, beacon configuration settings, and near-precise location, thus enabling remote fleet management and security services built on top of the beacons [4].

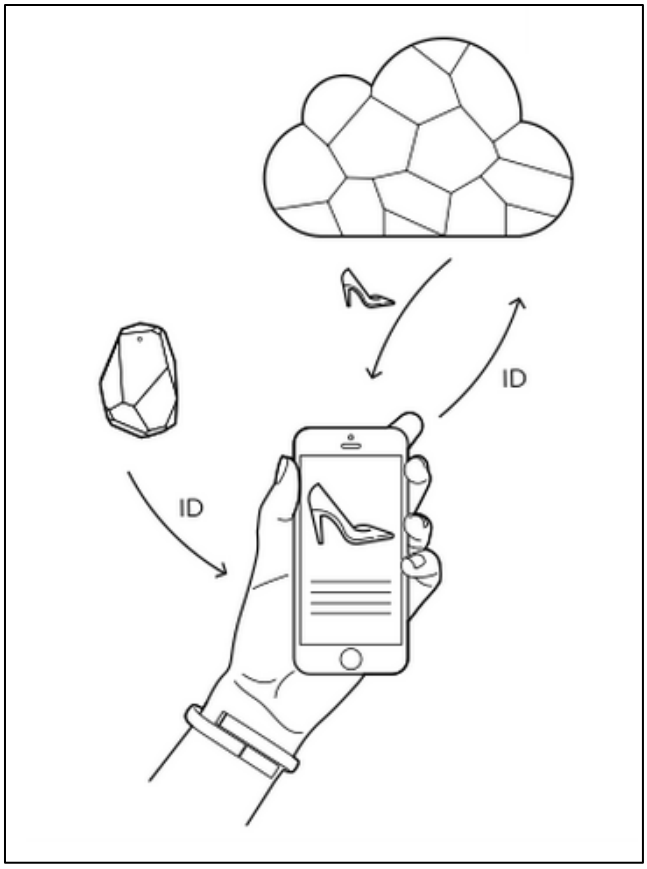

Fig. 5. Working of Estimote iBeacons

5.3 Compatibility with Estimote iBeacons

There are some minimum software and hardware requirements that need to be met in order for Estimote iBeacons to interact and integrate seamlessly with the BLE-enabled mobile device [40].

- **Hardware Requirements**: Bluetooth® Smart or Bluetooth Low Energy (BLE) technology
- **Software Requirements**: Apple iOS 7.0 or later version / Android 4.3 or later version. Additionally, the mobile device's bluetooth must be activated to receive signals transmitted by iBeacons. Further, Bluetooth Smart support was recently added to Windows Phone 8.1, but Estimote currently does not offer tools supporting that platform.

5.4 Ideal placement of Estimote iBeacons

Ideal placement for iBeacons is when deployed above people and objects because it provides a clear line of sight to the users. Bluetooth Low energy or Bluetooth Smart radio signals, like other radio signals, are susceptible to diffraction, interference, and absorption and multipath. These interference types can be minimized by placing the iBeacons above people and objects [44].

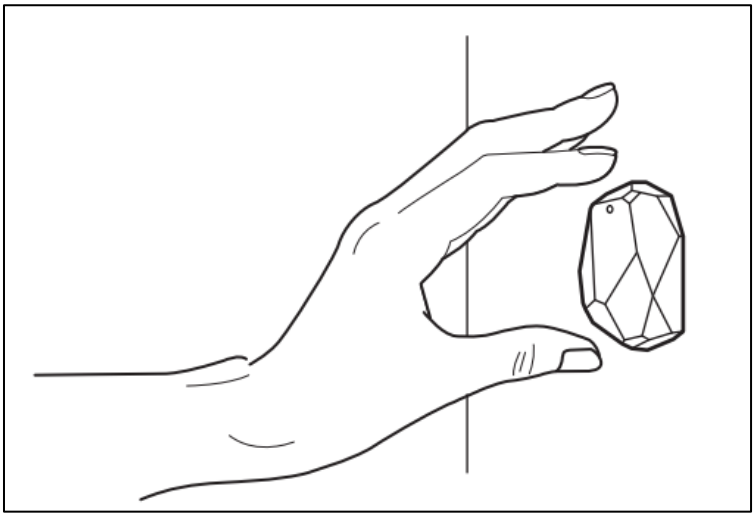

Fig. 6. Ideal Placement of Estimote iBeacons

Further, the data scientists have determined that the most effective iBeacon alignment is vertical position, with the tiny dot facing upwards, as shown in the figure. In this position, the iBeacon is able to transmit the radio signal in the most effective manner. Also, some users have reported that placing the iBeacons on ceilings ensures much improved signal stability [44].

5.5 How to modify Estimote iBeacon's identifiers?

Each Estimote iBeacon comes with a universally unique identifier (ID) that can be modified using Estimote App, Cloud or SDK. It can be divided into three parts: proximityUUID, Major, and Minor. The proximityUUID shall remain fixed whereas the other two are randomized. Default proximityUUID for Estimote iBeacons: B9407F30-F5F8-466E-AFF9-25556B57FE6D [42].

Thus, an iBeacon's ID can be changed by altering Major and Minor values. This can be easily done using the Estimote App through the following procedure [42]:

- First of all, the user needs to open the Estimote Official Mobile App
- Then, the user needs to access the 'Beacons' section in the Estimote App and select an iBeacon for which Major and/or Minor values need to be changed.
- Further, the user needs to authenticate by logging into its Estimote account via Estimote App.
- Then, the user needs to tap on Major or Minor and input the new value.

In order to change iBeacon's UUID, the user needs to follow this procedure [42]:

- First of all, the user needs to open the Estimote Official Mobile App
- Then, the user needs to access the 'Beacons' section in the Estimote App and select an iBeacon for which ProximityUUID value need to be changed.
- Further, the user needs to authenticate by logging into its Estimote account via Estimote App.
- Then, the user needs to tap on UUID, after which the user may either manually enter the new value or generate a new UUID through Estimote Cloud. It is important to

note that if someone else has already claimed a specific UUID for their iBeacons, then the user won't be allowed to use the same UUID.

5.6 Role of Triggers in Estimote iBeacons

Apart from the basic original methods of interacting with Estimote iBeacons available in Core Location framework, namely monitoring and ranging, another feature termed as "Triggers" has been introduced in Estimote SDK. It is a new, high-level API that lets the user forget about all the technical complexities involved in monitoring and ranging and rather focus on when the mobile device enters the range of an Estimote Sticker, thus enables the App to know about this event. If we put it in other words, "Triggers" allow the users to enrich their Apps using logic based on data broadcast by Estimote Stickers such as orientation, motion, temperature, and time. Currently, "Triggers" are only available for Estimote Stickers, but in the near future, support would be added for Estimote Beacons to Estimote SDK as well [47].

5.7 Updation of Estimote iBeacon's firmware

In order to access the latest Estimote features such as power management and secure UUID mode, the user needs to update the Estimote iBeacon's firmware through the following procedure [52]:

- First of all, the user needs to open the Estimote Official Mobile App
- Then, the user needs to access the 'Beacons' section in the Estimote App.
- After the iBeacon is placed next to the mobile device, it needs to be selected within the Estimote App.
- Then, the user needs to login into its Estimote account via Estimote App if it has not been done initially.
- Further, the user needs to click on 'Operating system' tab. This is where the user can update its iBeacon's firmware.
- The firmware update may take a few minutes, until it is done.

5.8 What is Estimote Cloud?

Estimote Cloud is a web-based platform based on cloud technology that has been developed for remotely managing Estimote iBeacons. This acts as a one-stop backend for managing the entire iBeacon fleet. Further, Estimote Cloud consists of various tools for analytics, remote ownership transfer, remote settings adjustment, remote fleet management, besides offering content management component that enables display of data and management of content linked to Estimote iBeacons [35].

5.8.1 Dashboard

Estimote Cloud account consists of a dashboard that displays each Estimote iBeacon associated to that account, along with its name, color and settings such as Proximity UUID, Major, Minor, broadcasting power, advertising interval, hardware and firmware version [45].

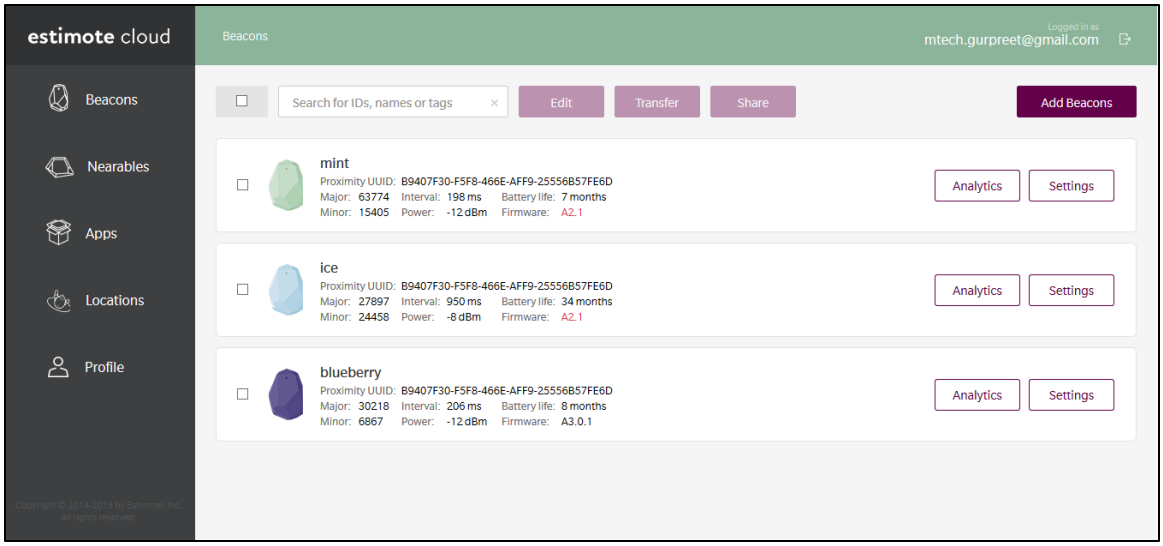

Fig. 7. Estimote Cloud: Dashboard

5.8.2 Fleet Management

Using Estimote Cloud, iBeacon settings can be remotely accessed and changed. Estimote Cloud acts as a remote fleet management tool which allows remote access to the iBeacons' settings. After successful login at http://www.cloud.estimote.com, the user can view a list of all iBeacons assigned to its Estimote Cloud account. Then, the user needs to

go to the Settings screen for an individual iBeacon, in order to access its unique ID, battery level, firmware version and other configuration settings such as Broadcasting Power and Advertising Interval [48].

If the user wishes to change its iBeacons' settings remotely, it just needs to check (select) the iBeacons it wishes to access in the dashboard and then click on the *Edit* button. Then, the user can alter the values for Broadcasting Power, Advertising Interval, Proximity UUID, Major and Minor. Further, the user can also enable or disable Secure UUID and Power Modes, and can even add tags for the iBeacons [48].

After the changes have been confirmed, they will be displayed in Estimote Cloud's dashboard as *pending*. In order to apply those changes to the iBeacons, the user just needs to log in to its account via Estimote mobile application and enter into the range of the iBeacons. This will ensure that the changes are automatically applied to the iBeacons. Moreover, this would take not more than 10 seconds, but the application needs to remain active i.e. run in the foreground. After the changes are applied to the iBeacons, they will be automatically reflected in the Estimote Cloud account [48].

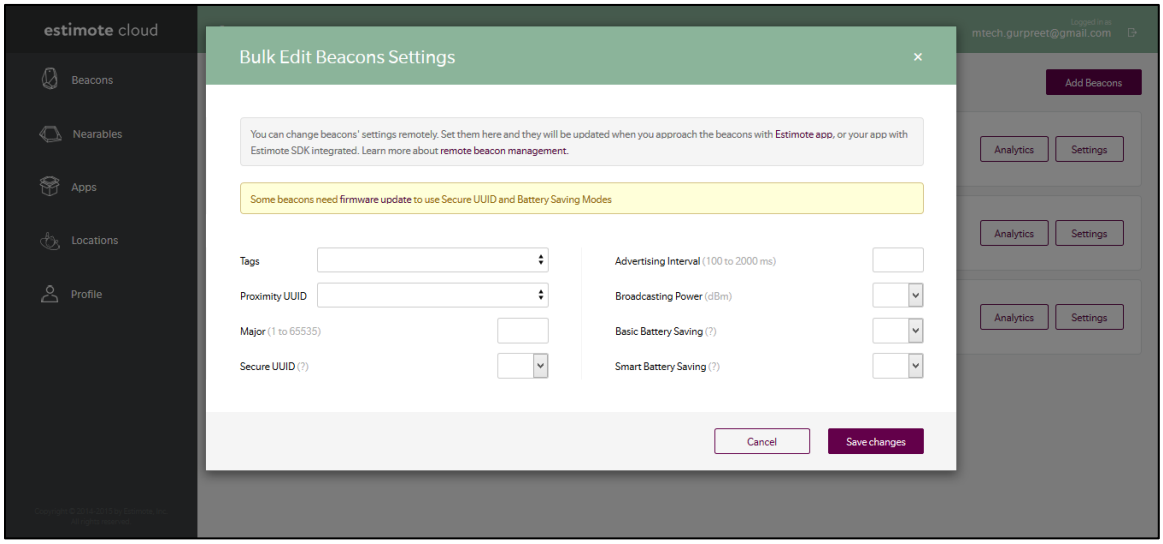

Fig. 8. Estimote Cloud: Fleet Management

5.8.3 iBeacon Analytics

Estimote Cloud offers base line smart analytics to enable tracking of iBeacon activity. Analytics lets the user know the number of times the mobile devices have entered the region of a specific iBeacon. A recent update to Estimote Analytics enables the users to track events for entire regions of iBeacons and gives access to a range of new RESTful API methods to access the data [55].

By default, analytics is disabled and needs to be enabled by the user. It is important to note that the user can collect and access analytics only for the iBeacons those are owned and assigned to its Estimote Cloud Account. The full list of iBeacons may be downloaded by the SDK during App ID/Token setup, and cached on the device. If the user adds more iBeacons to its Account, then it needs to make sure that the App re-performs the setup procedure [55].

Now, beacon region monitoring may be initiated for all the regions that the user wishes to track. Every time an “enter” event is reported, it is immediately recorded in Estimote Cloud Account. Estimote Analytics works only with beacon regions those are defined with proximity UUID, Major and Minor. Beacon regions that omit Major or Minor do not trigger a ping [55].

In order for the user to check the Analytics data, after the Analytics has been enabled in the App, the user can check analytics data of an iBeacon by accessing the Estimote Cloud Account at http://www.cloud.estimote.com and clicking on the “Analytics” button corresponding to that iBeacon [55].

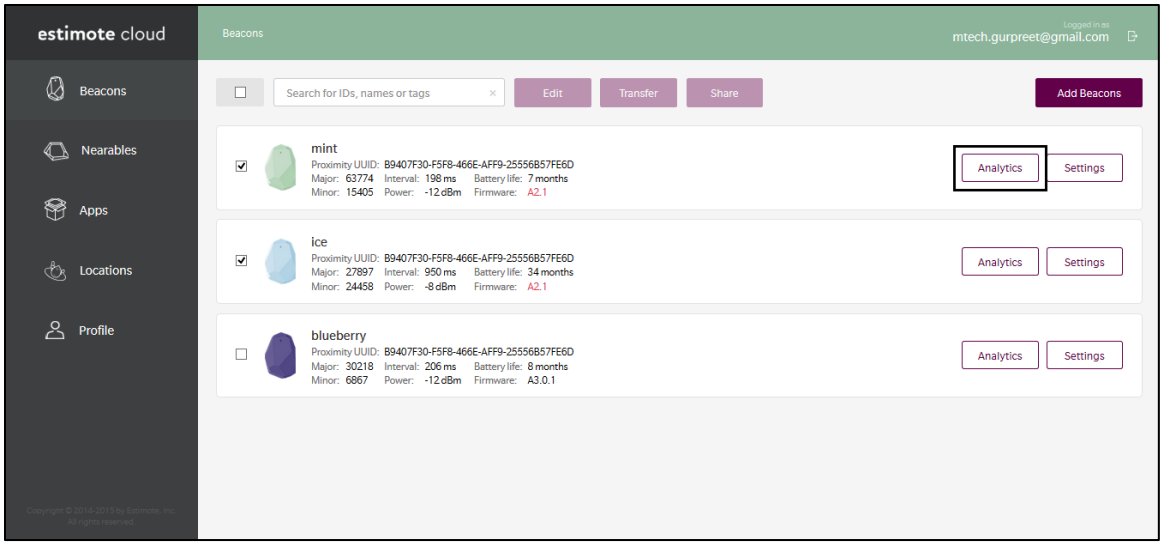

Fig. 9. Estimote Cloud: iBeacon Analytics

The user can view a graph listing the number of pings where the X axis represents days, default range being two months that includes the current and the previous months. In this graph, the number of pings is the total number of times; the devices with compatible Apps using Analytics entered within this iBeacon's range. The date on the graph corresponds to the time zone of user's iBeacon. It is essential to be note that that Estimote Cloud Analytics will record a ping only if the user has an active Internet connection on the mobile device [55].

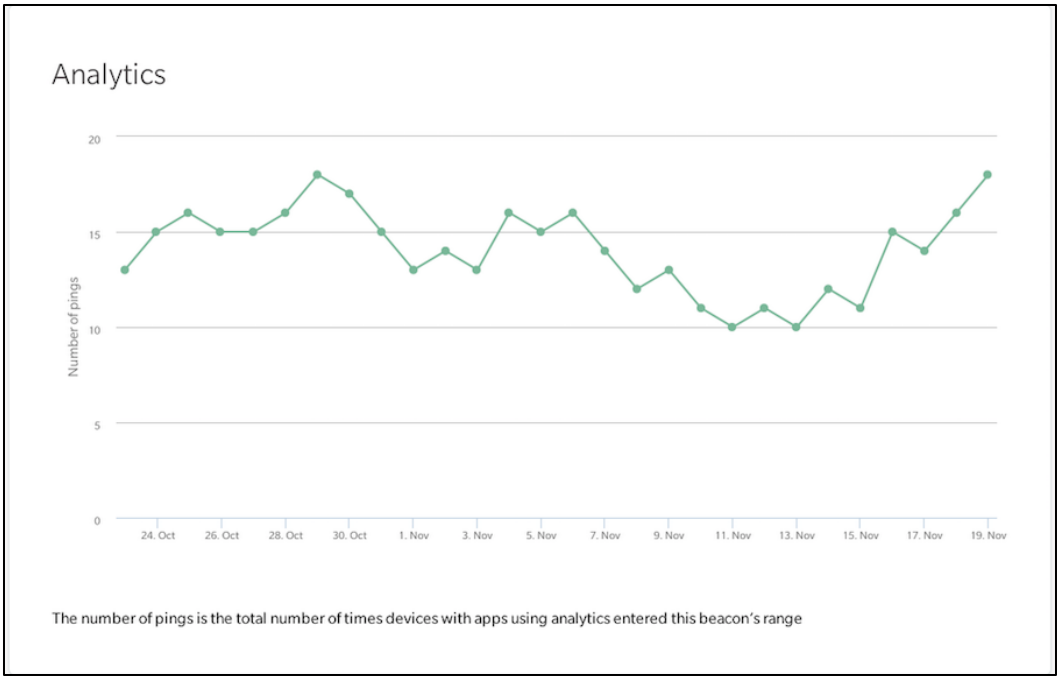

Fig. 10. Estimote Cloud: iBeacon Analytics in Graphical Form

5.8.4 iBeacon Sharing

Shared infrastructure is the key to many iBeacon deployments. An Estimote user can share its iBeacons with other Estimote users. Following are the requirements to enable iBeacon sharing between Estimote users [53]:

- App ID of the application with whom the user wants to share its iBeacons.
- iBeacons shall have Secure UUID feature enabled because without Secure UUID enabled, everyone can read the UUID/Major/Minor identifiers of the user's iBeacons. Thus, sharing something that is already publicly available does not make sense. Secure UUID makes the iBeacon's UUID encrypted and this ensures that only those applications (and people) the user wants to share its iBeacons with are able to decrypt it and thus, access the original UUID. Further, even though the iBeacons are being shared, the third-party application can only range and monitor for the shared iBeacons, but can't view or alter their settings.

In order to share the iBeacons using Estimote Cloud, the user needs to adhere to the following procedure:

- The user needs to login into its Estimote Cloud account and select the iBeacons that have the Secure UUID feature enabled. Such iBeacons can be easily recognized by locating the padlock icon.
- Then, the user needs to click on the “Share Beacons” button.

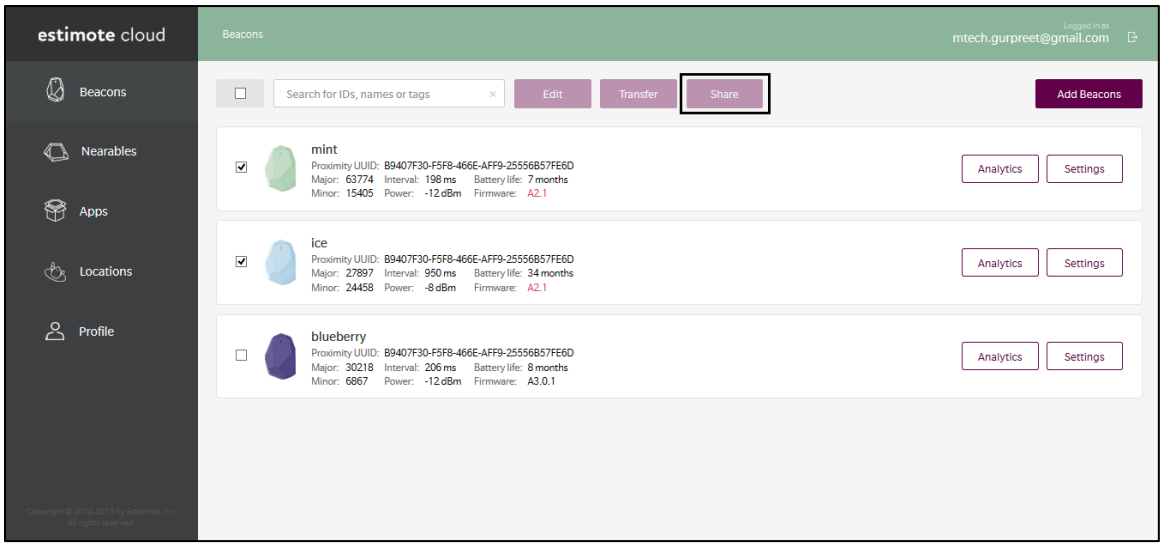

Fig. 11. Estimote Cloud: iBeacon Sharing

- Then, the user needs to enter the App ID of the application whom you wish to grant access to your iBeacons and click on “Share” button.
- Now, the application having the submitted app ID will be able to monitor and range for the shared iBeacons. After the user completes the iBeacon sharing process, the Apps screen that is available by clicking on “Apps” bookmark in the left panel, will start displaying a list of third party apps. The user can access this section to enable or disable the iBeacons for a specific application. The user can disable the iBeacons by changing the settings of a third-party application.

5.8.5 iBeacon Ownership Transfer

Estimote Cloud offers an easy and quick way to transfer the ownership of iBeacons. Transfer of iBeacon ownership is required in two situations [57]:

- A person has ordered the iBeacons, but some other person is using them.

- A person is using the iBeacons, but does not have access to them because they have been assigned to another Estimote Cloud account.

In order to transfer the ownership of one or more iBeacons using Estimote Cloud, the user needs to follow these steps [57]:

- The user needs to login into its Estimote Cloud account at http://www.cloud.estimote.com
- Then, the user needs to select the iBeacons it wishes to transfer.
- Further, the user needs to make sure that all of those iBeacons have the Estimote default Proximity UUID (B9407F30-F5F8-466E-AFF9-25556B57FE6D) as only those iBeacons having the default Proximity UUID can be transferred.
- Then, the user needs to click on 'Transfer Ownership' button.

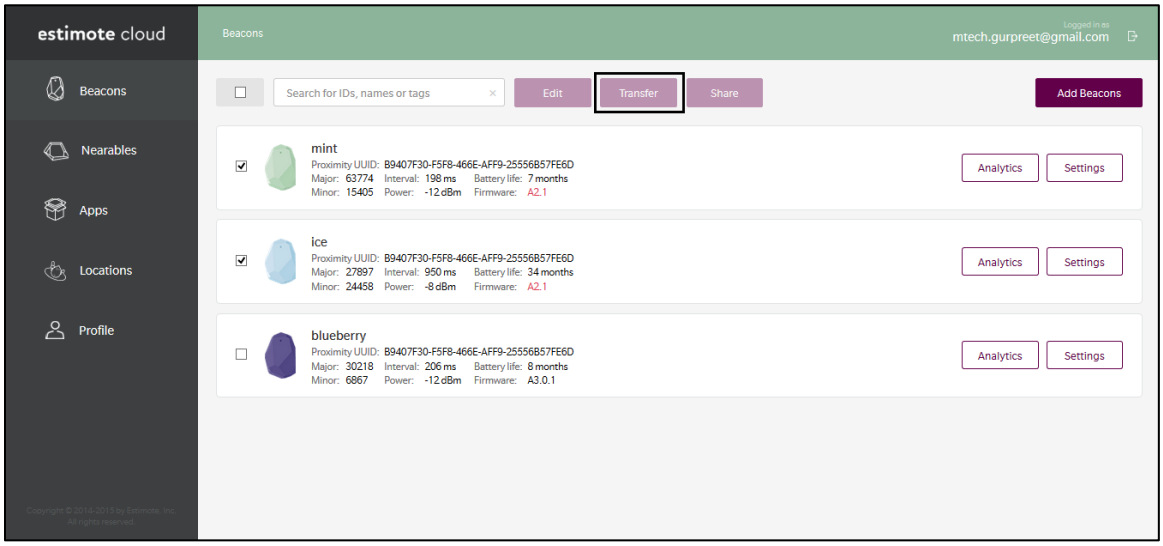

Fig. 12. Estimote Cloud: iBeacon Ownership Transfer

- In the last step, the user needs to type the email address of the new owner so as to transfer the ownership of selected iBeacons. Now, the new owner can access these iBeacons through own Estimote Cloud account.

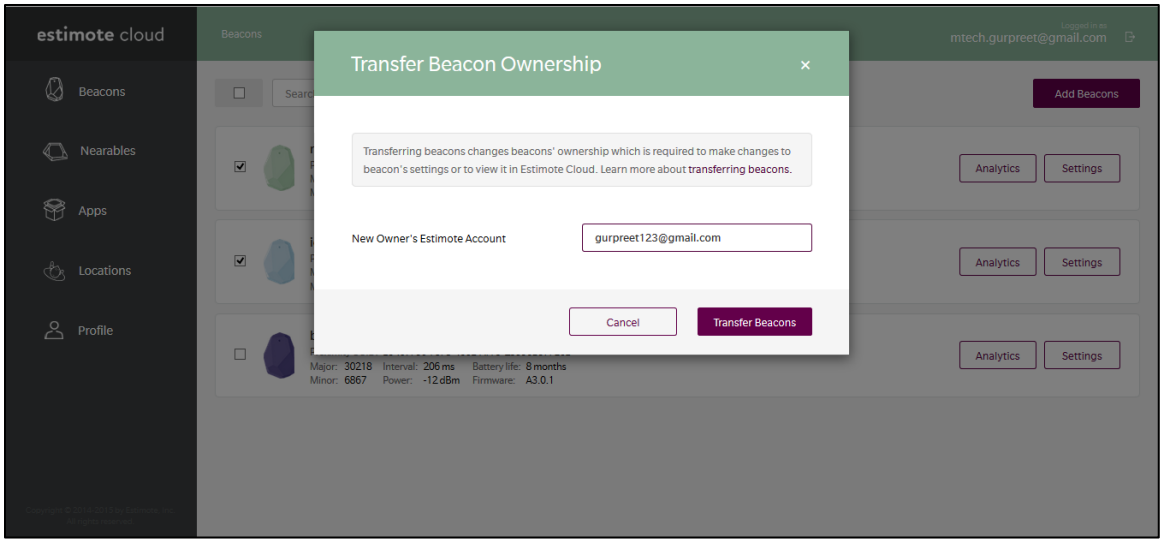

Fig. 13. Estimote Cloud: iBeacon Ownership Transfer to Another Estimote Account

5.8.6 iBeacon Naming

Estimote Cloud enables the user to assign a name to each of the owned Estimote iBeacons and this makes their management much easier. This can be done by clicking on Beacon Settings adjacent to iBeacon the user may wish to assign a name to, followed by choosing a name for that iBeacon. For example, 'library' or 'gate 1' [45].

5.8.7 API Access

If the user wishes to create its own custom application using Estimote SDK, it just needs to use the API token that would enable the application to connect to an iBeacon. The API token can be found by visiting the *Apps* section [45].

CHAPTER 6: COMPARISON OF iBEACON WITH NFC & GPS

iBeacon technology differs from other location-based technologies such as Near-Field Communications (NFC) and Global Positioning System (GPS) in that it offers some additional functionalities [60]:

- iBeacon technology has Bluetooth Low Energy (BLE) as its underlying communication technology. iBeacons are designed to be low-energy devices and hence, they can run for 2+ years on a single coin-sized battery.
- iBeacons don't calculate location, instead they calculate proximity of the listening device. This means that instead of mapping a geographic location, the iBeacons can be attached to objects that either move or are moved.
- iBeacons are affordable, easy to install and have a long lifetime.
- Unlike NFC, iBeacons work at distances that can range as far as 80 metres.
- iBeacons work indoors unlike GPS which doesn't work well inside. Thus, iBeacons are best suited for indoor positioning.
- Most importantly, iBeacon technology is being promoted by Apple Inc. that has paved the way for huge-scale adoption of iBeacon over NFC or GPS.

6.1 Comparison between iBeacon and NFC

iBeacon and NFC are increasingly being postured as competing technologies, wherein one will render the other as useless. However, they both connect the physical and digital worlds, but they have several fundamental differences in how they work and the type of interaction offered.

- iBeacons have a range up to more than 80 metres whereas NFC has a very limited range that is even less than 4 cm (1.57 inches).
- NFC can be either passive or active. When working in passive mode, the power is sent from the reader device. Although Passif, an entity bought by Apple Inc. has

worked on reducing the energy consumption, a battery is still required inside iBeacons.

- iBeacons are based on BLE technology. Most of the modern smartphones are offering support for Bluetooth Low Energy (BLE) whereas various mobile phone vendors such as Apple are omitting the support for NFC standard, paving the way for advancement of iBeacon technology over NFC.
- Both, iBeacons and NFC support in-store mobile payments. Moreover, iBeacons having much larger range as compared to NFC can be better suited for mobile payments as the customers can pay for their shopping bill while being at a distance from the iBeacons [62].
- iBeacons are small BLE-enabled wireless sensors that interact with smartphones, while NFC uses short-range radio waves to enable the information exchange between two devices placed at a short distance, without the need for battery power [62].
- In order to take advantage of NFC technology, the user needs a device that contains an NFC chip. Apple's latest smartphones do not support NFC, nor do several other smartphone vendors. Even if the user has an NFC-enabled phone, retail stores need to deploy the required NFC infrastructure in order to complete transactions. Further, adoption of NFC has been very slow as it means huge investment for businesses. On the other hand, iBeacons only require the listening devices to be BLE-enabled, which most of the smartphones already are. Businesses can easily purchase iBeacons and this would just cost a fraction of what it would to deploy NFC infrastructure [58].
- Another limitation to the NFC standard is that it is solely involved in payments-related solutions. There can be no further interaction between the business and customers. On the other hand, iBeacon has changed the user's shopping experience. Retailers can easily place iBeacons throughout their store and deliver product information and promotional offers onto customers' smartphones [58].
- iBeacon can also be used for indoor mapping purpose. For example, a museum can place iBeacons around different galleries and exhibits and thus, guide the visitors through the museum. Further, information related to exhibits or location of galleries can be delivered using iBeacons-powered smartphone application [58].

- iBeacons can push content, such as promotional offers, location data, or personalized suggestions based on a customer's preferences or shopping history. In case of NFC, it is the customer who must initiate the interaction [62].
- NFC supports encryption, and its susceptibility to hacking is minimal because of close proximity between two communicating devices [62].
- In case of NFC marketing, customers make the choice to engage or not. For example, if a restaurant fixes an advertising banner that needs to be tapped in order to receive a discount coupon, the consumer needs to make the choice to engage. On the other hand, iBeacon is a more aggressive marketing strategy where the promotional content is pushed to BLE-enabled devices within range. However, application of iBeacon or NFC depends upon the specific marketing campaign [64].

6.2 Comparison between iBeacon and GPS

- Being low-energy, iBeacons are also generous to the user's mobile phone battery. Unlike GPS, they don't drain out a user's phone battery [60].
- iBeacons are best suited for indoors unlike GPS which doesn't work well inside. Thus, iBeacon is the most suitable choice for indoor positioning [60].
- Earlier, retailers used to rely upon GPS coordinates to send push notifications whenever the users were in vicinity of their stores. But, this did not prove to be accurate as push notifications would sometimes appear even when a user was just entering the neighborhood. With the advent of iBeacons, it is now possible to easily target multiple indoor locations, which GPS certainly fails to do [58].
- GPS deals in "location" whereas BLE-enabled iBeacons deal in "micro-location". Micro-location capabilities of iBeacons allow the retailers and businesses to take advantage of in-store marketing, which in turn redefines the user's shopping experience. Micro-location targeting when combined with GPS-based macro-location targeting, can prove to be a game changer in marketing and customer engagement strategies [58].
- While GPS may enable location-enabled services (LBS), iBeacons have the potential to enable micro-location and contextual aware services. In case of iBeacons, the smart device can tell the user's location with reliable accuracy, say within a hundred

yards. Further, this location information can be leveraged to push notifications or deliver relevant information. But, in case of GPS technology, location tracking is not considered to be reliable and accurate for being deployed in micro-location based services such as in retail stores and museums [63].

CHAPTER 7: CUSTOM ANDROID APP DEVELOPMENT

One of the unique features of mobile applications is location awareness. Mobile users carry their smart devices with them wherever they go, thus adding location awareness to any application offers the users a more contextual aware experience. The location APIs available in Google Play services facilitate addition of location awareness to an application, thus enabling automated location tracking, geofencing, and activity recognition. These location APIs are preferred over the Android framework location APIs for enabling location awareness in an application [17].

7.1 Estimote SDK for Android

The Estimote SDK for Android is a library that allows the interaction of Android – based devices with Estimote iBeacons. The Estimote SDK system requires Bluetooth Low Energy (BLE) support and Android 4.3 or above. Estimote SDK offers the following functionalities [25]:

- **Beacon Ranging**: This is used to scan Estimote iBeacons and optionally filter them using their properties.
- **Beacon Monitoring:** This is used to monitor beacon regions for those listening devices that have entered or exited a beacon region.
- **Beacon Characteristic Reading and Writing**: This may be used to read, access or change beacon characteristics such as proximity UUID, major & minor values, broadcasting power, advertising interval.

7.2 Features available on our Custom App

Our custom-built iBeacon-enabled android app offers the following features:

7.2.1 iBeacon Scanning

Our custom app has been designed to scan for iBeacons in the vicinity of the mobile device. Our app allows interaction of mobile device with iBeacons via Bluetooth Low Energy (BLE) communication as shown in Fig.

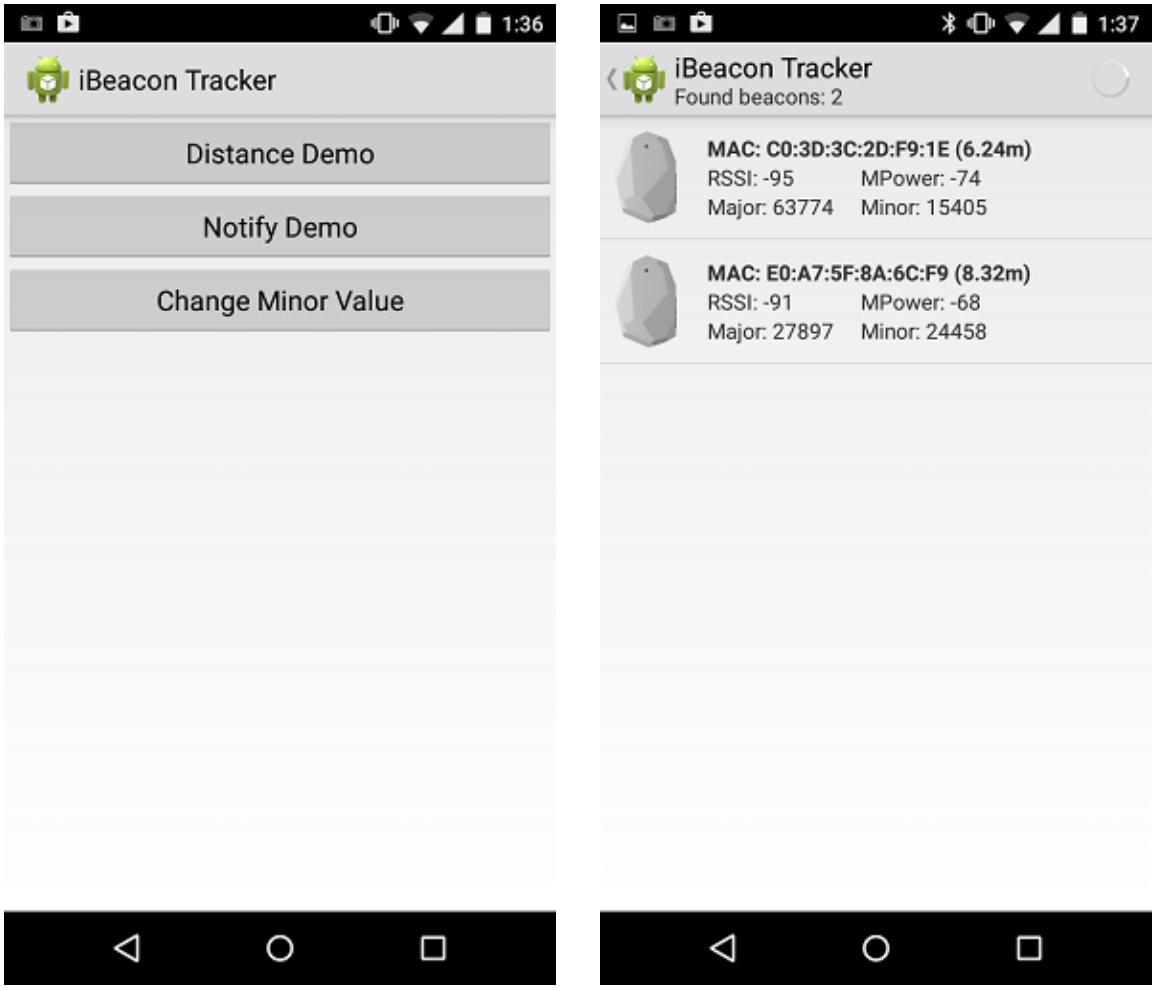

Fig. 14. Custom App: iBeacon Scanning

7.2.2 Process Automation

Our custom app automates the process of recording of Distance & RSSI Readings, where an iBeacon automatically interacts with the mobile device for ‘N’ number of times as specified by the user. Further, using our app, the users can record ‘N’ Readings for Distance & RSSI as shown in Fig.

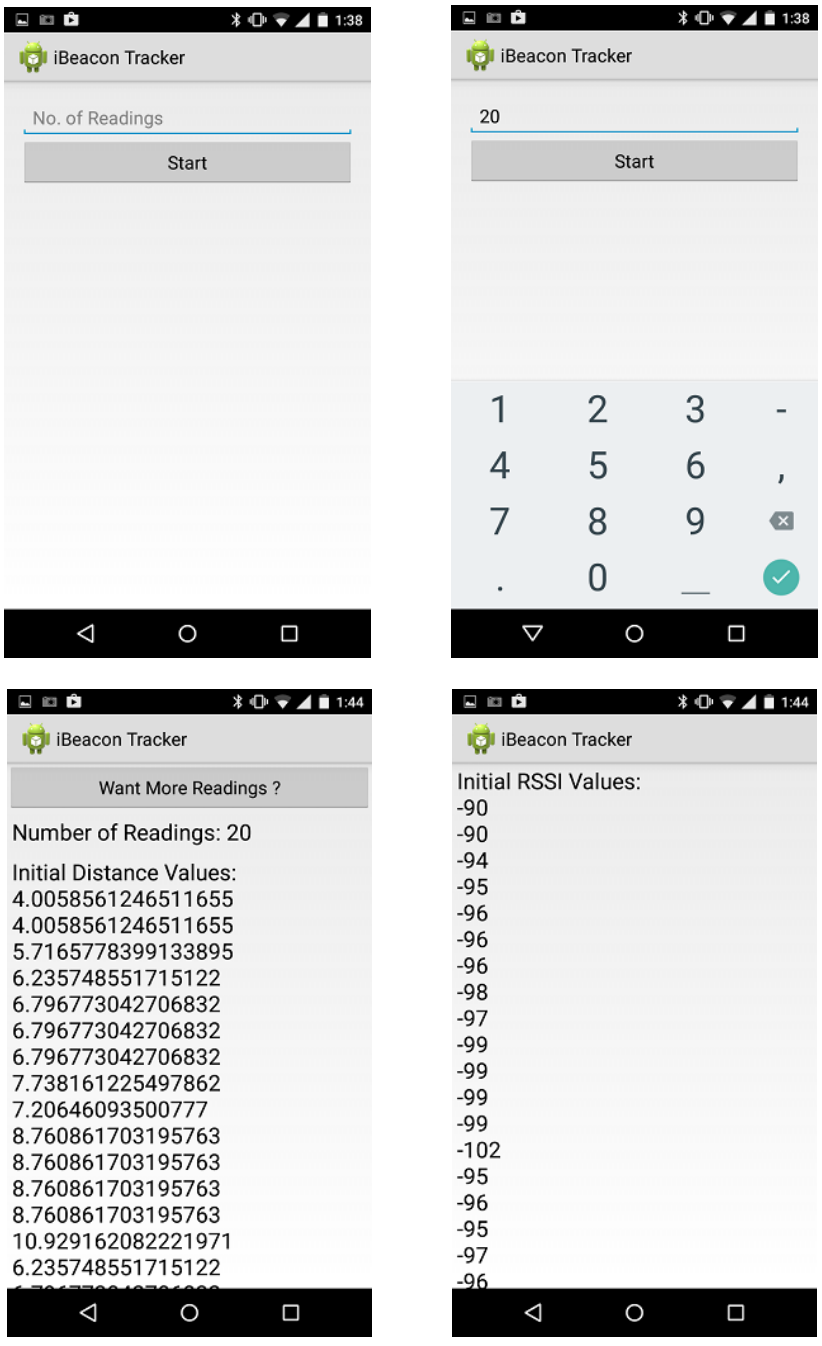

Fig. 15. Custom App: Process Automation

7.2.3 Distance Estimation (Localization)

Our custom app estimates the distance between an iBeacon and the mobile device as shown in Fig. using our localization algorithm, iBLA that has been discussed later in chapter 9.

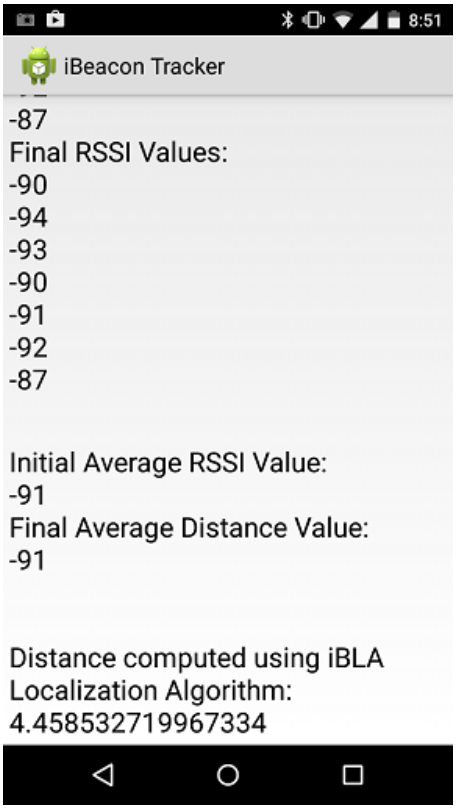

Fig. 16. Custom App: Distance Estimation (Localization)

7.2.4 App Notification

Besides distance estimation, our custom app provides Notification functionality, wherein the mobile app gets notified about the entry or exit of the mobile device from the radio range of an iBeacon device as shown in Fig.

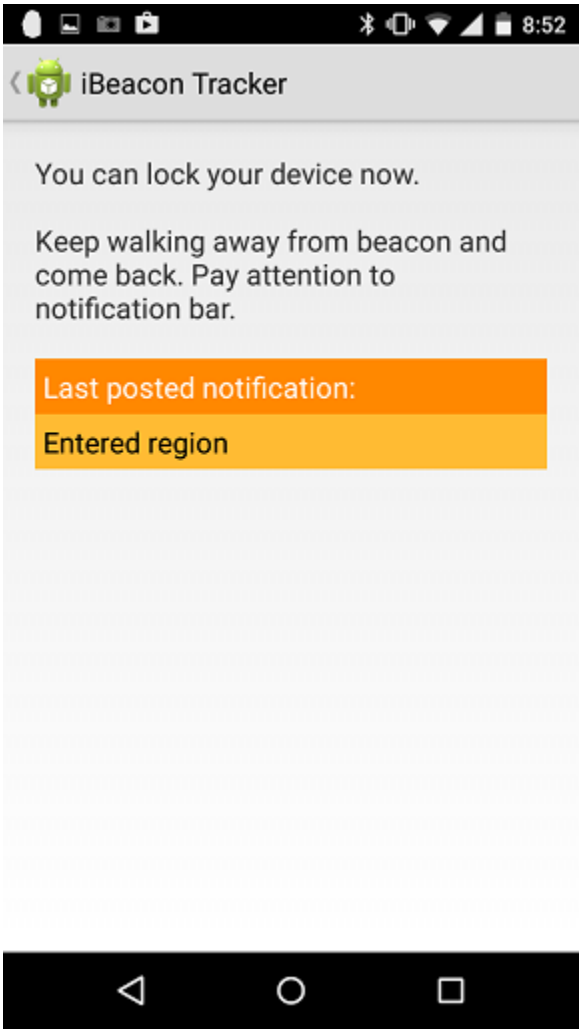

Fig. 17. Custom App: App Notification

7.2.5 iBeacon Characteristics

Our custom app lets the users change the Minor value (a part of iBeacon's identifier) of an iBeacon device to any other value as desired by them as shown in Fig.

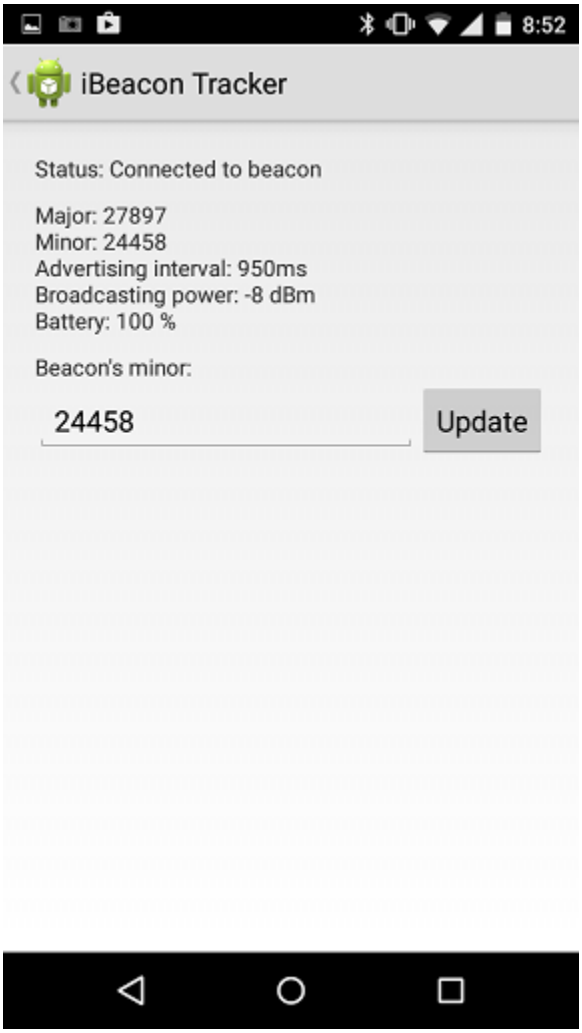

Fig. 18. Custom App: iBeacon Characteristics

7.2.6 Share & Save Reading Sets

Our custom app provides an option to share the Reading Sets via Gmail or any other email service provider as shown in Fig. Also, the app lets the users save their Reading Sets to their own mobile device's storage as shown in Fig.

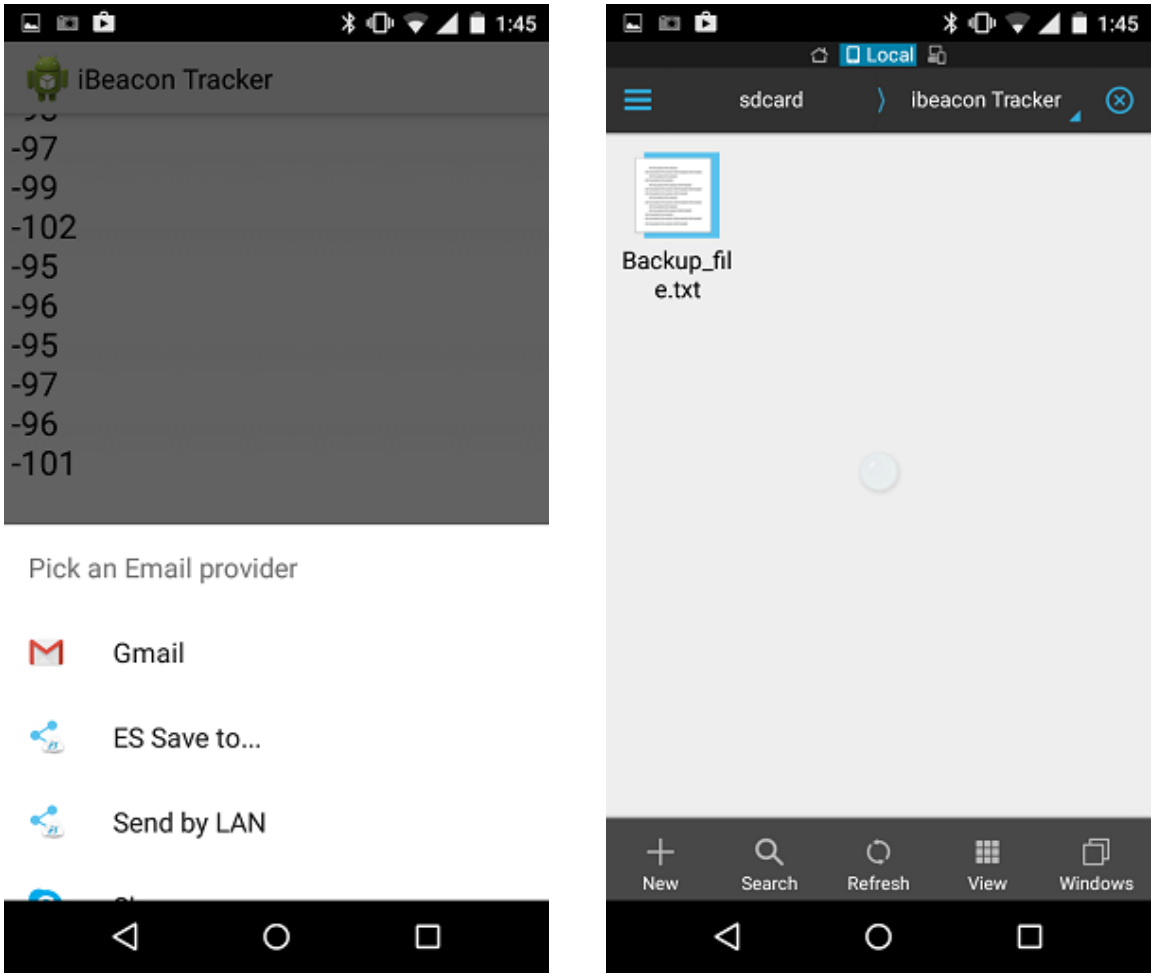

Fig. 19. Custom App: Share & Save Reading Sets

7.3 Reading Sets & Graphs

TABLE II: READING SET: RSSI VS REAL DISTANCE FOR CONTINUOUS TIME INTERVALS

Experiment	RSSI, (Averaged over 20 Readings) (-dBm)	Real Distance (m)
Avg RSSI after 20 scans at 300 ms interval 1	-56.8	0.2
Avg RSSI after 20 scans at 300 ms interval 2	-61.5	0.4
Avg RSSI after 20 scans at 300 ms interval 3	-81.3	2
Avg RSSI after 20 scans at 300 ms interval 4	-86.9	4

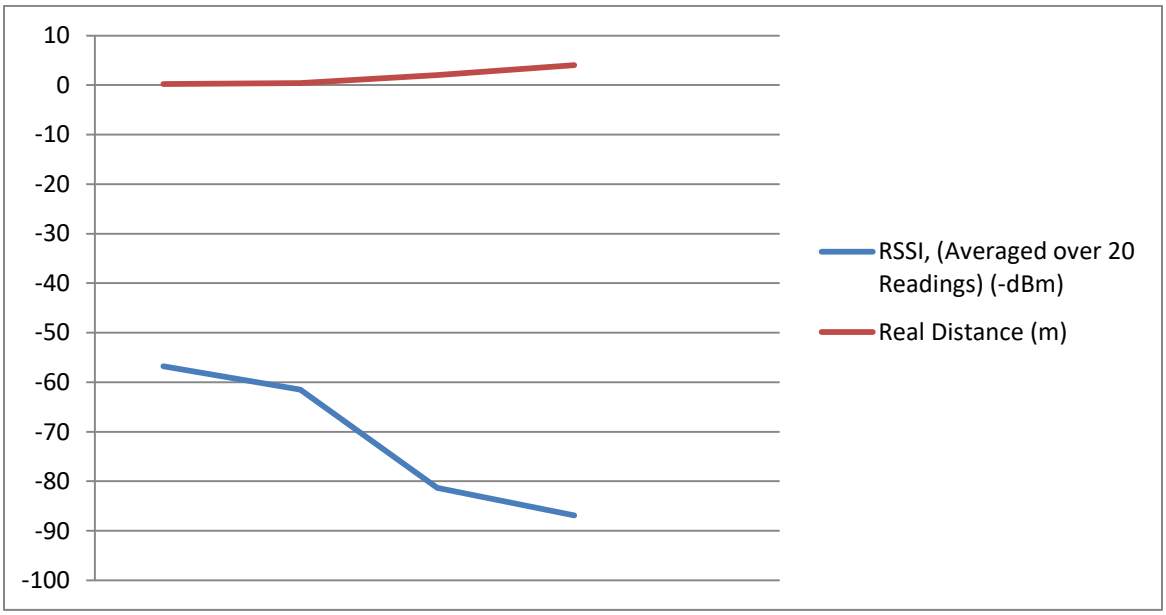

Fig. 20. Graph: RSSI vs Real Distance for Continuous Time Intervals

TABLE III: READING SET: ESTIMATED DISTANCE VS REAL DISTANCE

Average Distance after 20 Readings at 200 ms interval (4 Results) (m)	**Average of 4 Results (m)**	**Real Distance (m)**
0.4, 0.37, 0.27, 0.49	0.38	0.3
0.9, 0.6, 0.4, 1.5	0.85	2
1.75, 1.7, 2, 1.7	1.78	4

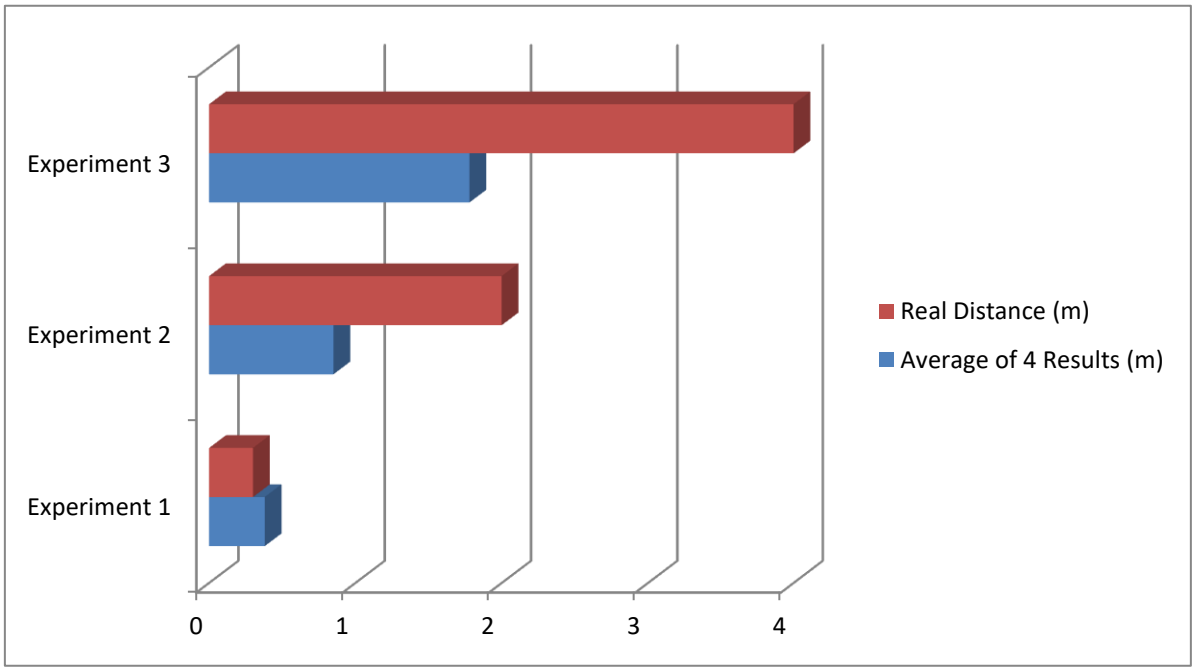

Fig. 21. Graph: Estimated Distance vs Real Distance

TABLE IV: READING SET: RSSI VS REAL DISTANCE FOR NON-CONTINUOUS TIME INTERVALS

Real Distance (m)	RSSI (Averaged over 20 Readings at 950 ms interval) (-dBm)
0.1	-56
0.2	-58.5
0.4	-60.15
0.5	-62.5
0.75	-67.95
1	-71.95
2	-82.9
4	-89.7

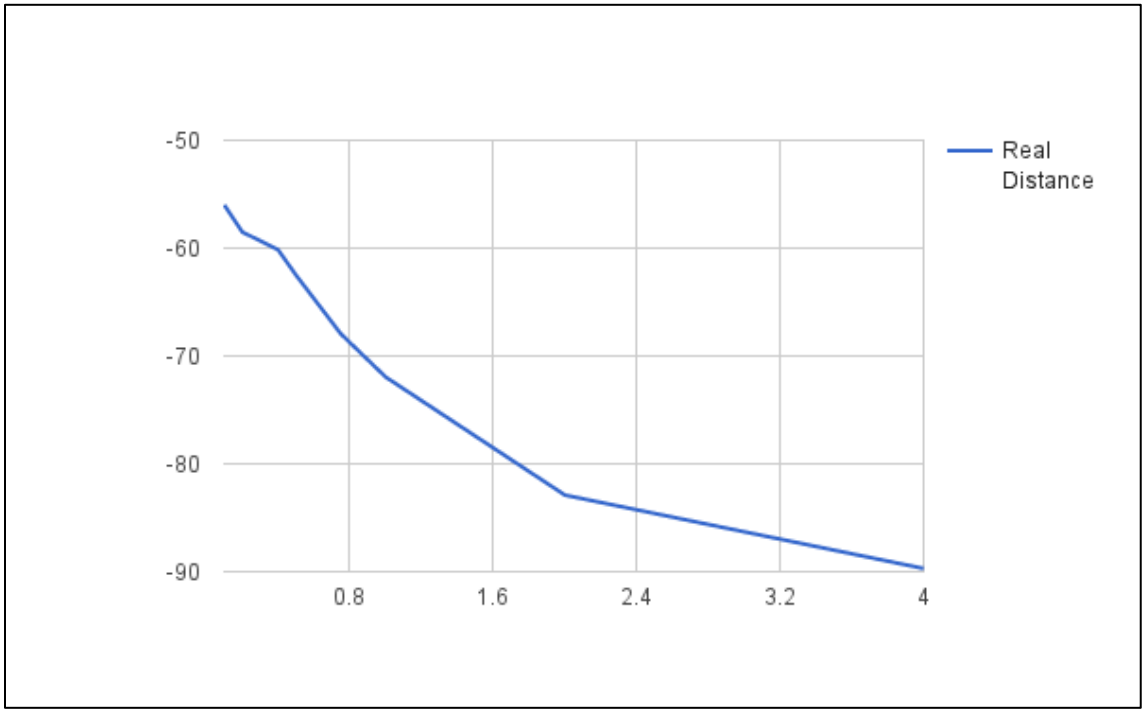

Fig. 22. Graph: RSSI vs Real Distance for Non-Continuous Time Intervals

CHAPTER 8: LOCALIZATION IN WIRELESS SENSOR NETWORKS

A wireless sensor network (WSN), sometimes referred to as a wireless sensor and actor network (WSAN) are spatially distributed autonomous sensors used to monitor physical or environmental conditions, such as temperature, pressure, sound etc. and to cooperatively pass on their data through the network to a specific main location. Modern wireless sensor networks are bi-directional, thus enabling control over sensor activity as well. Initially, the development of wireless sensor networks was motivated by military applications such as battlefield surveillance whereas today these networks are being deployed in a range of industrial and consumer applications, such as industrial process monitoring, machine health monitoring, etc [1] .

The wireless sensor network (WSN) is built of "nodes" ranging from a few to several hundreds or thousands, where each node is connected to one or many other sensors. Thus, the “node” is the building block of any WSN. Each node consists of several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensor nodes and an energy source, usually a battery. A sensor node might vary in size ranging from that of a shoebox to the size of a grain of dust. Similarly, the cost of sensor nodes is variable, ranging from a few hundreds to thousands of rupees, depending upon their associated complexity. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as memory, computational speed, and energy and network bandwidth. Further, the topology of the WSNs might vary from a simple star network to an advanced multi-hop wireless mesh network. Also, the propagation technique employed between the hops of the WSN may be routing or flooding [69, 70]. The main characteristics of a WSN include:

- Energy consumption constraints on sensor nodes
- Ability of WSNs to cope with node failures (also known as resilience)
- Mobility of WSN nodes

- Heterogeneity of WSN nodes
- Scalability of WSN to large scale deployment in industrial applications
- Ability of WSN to withstand harsh environmental conditions
- Cross-layer design support in WSN

Wireless sensor networks (WSNs) have recently gained a lot of attention due to its wide range of applications such as object tracking, warehouse inventory, environmental monitoring, and health care. In all these applications, physical data is continuously collected by the sensor nodes in order to facilitate application-specific data processing and analysis. We are particularly interested in location-aware wireless sensor networks since applications tend to focus more on events associated with a geographical location or region as compared to that on raw data available on sensor nodes.

8.1 Issues in Localization algorithms

In this subsection, we discuss the issues related to localization algorithms:

8.1.1 Resources constraints

Sensor nodes are typically heavily resource-constrained. This renders them unfit for performing large computations. Further, their low battery capacity implies that sensing, communication, and processing must be reduced so as to increase the lifetime of WSNs. A wireless sensor network typically consists of hundreds or even thousands of sensor nodes in sensing or tracking applications where localization is not the primary objective of WSNs. Hence, the energy demand and associated cost of localization must be minimized with permissible localization error. Thus, researchers must focus upon design of localization algorithms with minimum communication and computational overheads as well as maximum localization accuracy.

8.1.2 Radio propagation

A number of factors including free space loss, scattering, reflections, and penetration affect the radio signal propagation. Environmental obstacles such as walls, objects, furniture and specific environmental irregularities such as sand and grass further enhance

the uncertainty of radio signal propagation. Since, range based localization algorithms depend upon radio range to estimate distance, thus all these issues need to be resolved in real-life deployments.

8.1.3 Node density

The accuracy of majority of localization algorithms depends upon node density. Seed based localization algorithms perform badly when the seed density is low in a part of the WSN. Similarly, radio hop count based algorithms [38] require high node density for accurate hop count-distance approximation. Many WSNs not being spatially homogeneous, node density tend to be low in the sparse parts of the sensor networks. Mobility as well can lead to low node density in some parts of the networks. Thus, localization algorithms must be able to perform well in a variety of node densities.

8.2 Ranging Techniques in Wireless Sensor Networks

In this section, the most popular techniques for distance estimation in WSNs have been discussed.

8.2.1 Received signal strength indicator (RSSI)

RSSI is a measurement of the power at which a packet is received. In other words, it equals the transmitter power minus the radio path loss (path attenuation). It can also be defined as the strength of the received radio signal. When a radio signal propagates through space, it decays in power with increasing distance. Therefore, if the transmitter power is known, then the RSSI of a received radio signal can be employed to estimate the distance travelled by the signal [59]. In practice, range estimation based on RSSI is inaccurate due to path loss being influenced by many factors, such as interference, reflection, refraction, diffraction, and absorption [65]. Moreover, in indoor environments, walls and other obstacles such as furniture have a negative impact on the distance estimation or ranging accuracy. However, most of the recently manufactured wireless sensors determine RSSI values, and hence no additional hardware needs to be deployed in order to estimate the received signal strength.

8.2.2 Time of arrival (ToA)

This technique uses the speed of a radio signal and its travelling time to compute the distance travelled by the signal. Each wireless sensor sends a packet to its neighbours, and upon receiving the packet, the neighbours send packets back to the transmitting sensor. Since the signal propagation speed is determined, a sensor node can easily compute its distance to another neighbor by simply multiplying the signal speed with half the round trip time of the radio signal [61].

8.2.3 Time difference of arrival (TDoA)

In this technique, each sensor node contains two types of transceivers, typically, a microphone and a speaker, and a wireless transceiver that employ two distinct signals with altogether different speeds [61, 68, 10]. A sensor node first transmits a radio signal, and then transmits an acoustic signal after waiting for a fixed time interval t_{delay}. A neighbouring sensor node, on receiving the radio signal, records the received time of the radio signal t_{radio}, then activates the microphone to listen to the acoustic signal $t_{acoustic}$ and then again records the received time of the acoustic signal. Then, the receiver sensor node computes the pair-wise distance d using these different time values.

$$d = (v_{radio} - v_{acoustic}) * (t_{acoustic} - t_{radio} - t_{delay})$$

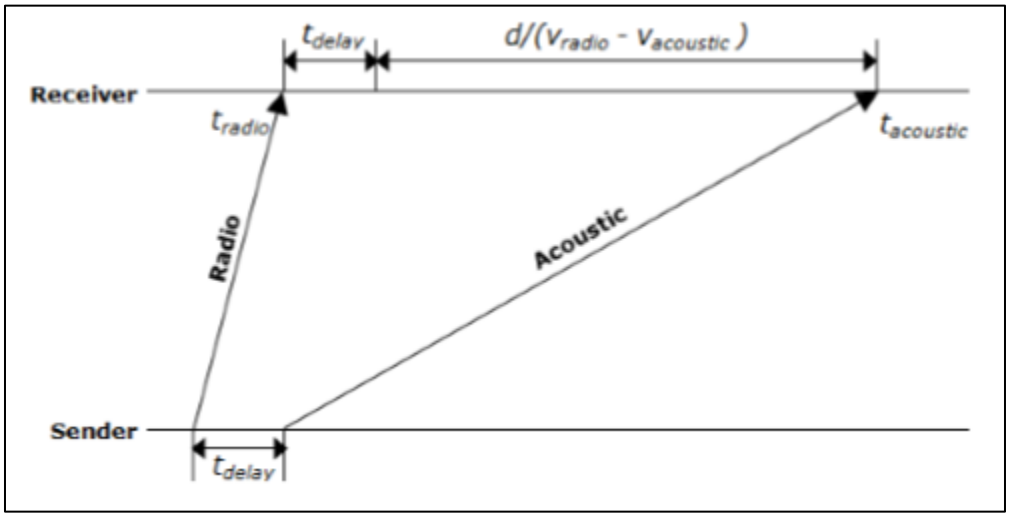

Fig. 23. Time Difference of Arrival (TDoA)

The precision of this technique depends upon various conditions. An echo-free environment and a direct line of sight between the transmitter-receiver pair, both ensure better performance. Further, the speed of the acoustic signal depends upon the air humidity and temperature, which may cause errors in the distance estimation. Moreover, this technique requires each sensor node to be equipped with an extra transceiver, and this has a bearing on its size and cost.

8.2.4 Angle of arrival (AoA)

This technique detects the direction of a radio signal incident on an antenna array. Hence, this technique requires each sensor node to be attached with several spatially separated antennas. In this technique, TDoA at individual element of the antenna array is first measured, and then AoA is computed using these delays [67]. The limitation of this technique is that AoA hardware is bigger and more expensive as compared to that of TDoA. Further, placement of the spatially separated antenna array around the sensor node may not be possible because of the comparatively smaller size of sensor nodes.

8.2.5 Ranging using RSSI

Several techniques have been proposed that are based on received signal strength (RSSI). For example, triangulation technique uses the distances of an object from three fixed seeds (nodes) to compute the location of that object within a building [65]. Ohta et al. [66] estimates the distance of the target node based on the RSSI from different static sensors deployed across the sensor network. Also, maximum-likelihood estimation and minimum mean square error are used to estimate the location of the target node. All of these techniques mentioned above require static seed nodes in order to localize the target node. However, mobility of the nodes also needs to be considered.

CHAPTER 9: LOCALIZATION IN iBEACON NETWORKS

In this section, we propose a localization algorithm for iBeacon network deployed within a rectangular region with both types of nodes, seeds as well as non-seeds. iBeacon Localization Algorithm (iBLA) largely consists of three steps: initialization, sampling, and filtering. We assume that time is split into discrete time steps. In the initialization step, candidate samples for each non-seed node are randomly selected from the deployment plane. Further, we assume that each time step has two phases. In the first phase, all seeds broadcast their exact locations and each non-seed broadcast a set of possible locations, an estimated single location, computed using the possible locations, and a quality measure of the estimate, computed at previous time step, to its neighbours. During the second phase, each non-seed node constructs a sample area using the received information and its own set of locations, computed at previous time step. Further, candidate samples are determined using the sample area and each candidate sample is allotted a weight based on the received information. In the filtering step, samples with weight 0 (zero) are discarded and the weighted average of the filtered samples is the estimated location that is eventually broadcasted to the next time step of the non-seed in the current time step. The filtered samples are the set of possible locations to be broadcasted in the next time step. A quality measure associated with every location estimation output is also computed, that is broadcasted in the next time step. Further, we discuss in detail, some major aspects of the proposed algorithm.

9.1 Constraints

Constraints are of two types, positive and negative. If a node is within the radio range of another node, then it may be inferred that they must be within distance r of each other. This forms an example of a positive constraint. Similarly, if two nodes are not neighbours but are within two-hop of each other, the distance between them must be greater than r. This forms an example of a negative constraint, where each node lies outside a circle of radius r and centered at the other node’s location.

We assume that all nodes, including seeds and non-seeds are deployed uniformly in a rectangle planar area. In order to enhance the simplicity, we assume that the radio propagation is isotropic. Further, we assume that every node knows its maximum speed referred by V_{max}. For a sensor node, we assume S to be the set of its one-hop seed neighbours, T to be the set of its two-hop seed neighbours, and NS to be the set of its one-hop non-seed neighbours.

9.2 Distance computation using RSSI

Signal strength is severely affected by multipath and interference caused by obstacles. Thus, to achieve reliable upper and lower bounds on distance estimation using RSSI is an arduous and complex task. In an attempt to model the radio signal propagation, several experiments were carried out on our sensor hardware in different environmental conditions such as hall, large open space, class room etc. Our experimental results reveal that the standard path loss model [59] does not fit our measured data. Therefore, we utilize our experimental data to get upper and lower bounds on the distance between the nodes. In our experimental setup, we allow several packets to be transmitted between an iBeacon-listening device pair and record the average RSSI for those packets in units of dBm. We varied the distance between the iBeacon and the listening device in the range of 0.1-70 metres. We took 20 sets of received signal strength measurements for each distance. We used the experimental data to compute a distribution D(x) of measured RSSI values for a range of values for actual distance x. We considered computing the upper and lower bound on each distance from the distribution D. However, this approach resulted in lower efficiency. In our simulations, we generated RSSI values at listening devices by sampling from the distribution D(x) for each actual distance x. It was observed that the RSSI measurements varied depending upon the environment.

9.3 Bounding box creation

As in monte carlo localization (MCL) algorithm [67], there are two areas that are considered in the localization algorithms, candidate sample area and the valid sample area. The candidate sample area is built based upon the previously computed set of locations and maximum speed is used to get candidate samples as in Figure 3.2. Thus the

candidate sample area increases with the increase of maximum speed Vmax. The valid sample area formed adhering to the constraints is used to reject the invalid candidate sample areas. With increasing seed density, the valid sample area decreases because of imposition of more severe constraints. The sampling efficiency is defined as the fraction of candidate samples those are valid and this decreases when increase in maximum speed or seed density, thus increasing the associated computational costs. In the approach used by the monte carlo localization boxed (MCB) algorithm [2], the candidate sample area is reduced by creating a bounding box using the one-hop seeds as in Figure 3.3. Further, upper and lower annular bounds on each distance from one-hop seeds give a positive and a negative constraint as shown in Figure 3.5. Hence, upper bound reduces the bounding box as it replaces the entire radio range.

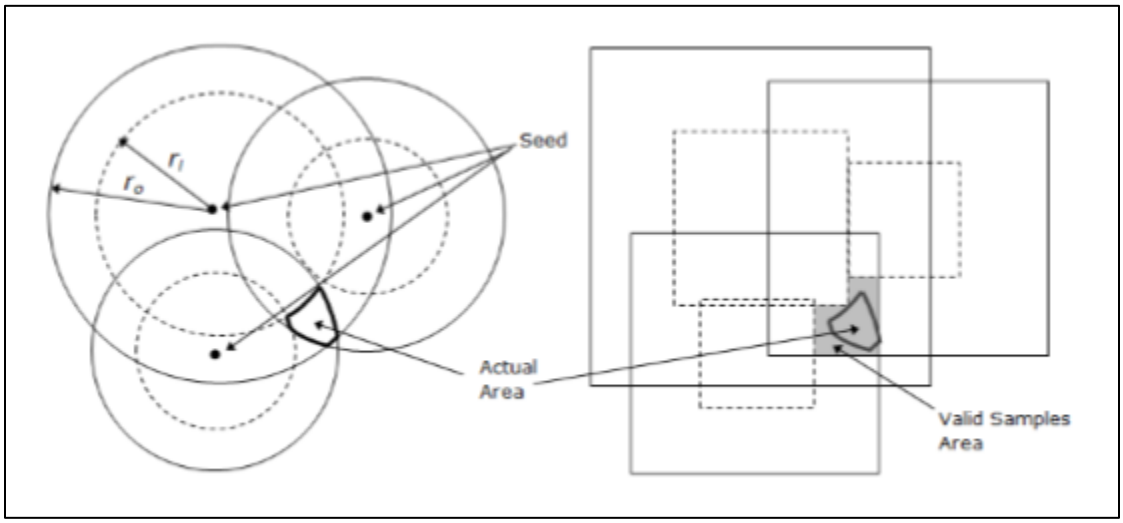

Fig. 24. Bounding Box Creation

As done in WMCL, negative information from two-hop seeds, non-seed neighbours, previous location estimates, and maximum error in both axes are used to reduce the bounding box. It was studied that extra constraints derived from range information improve the accuracy in case of RMCB but only at the cost of lower sampling efficiency. As one-hop seed neighbours also contribute to negative constraints, we exploit these constraints to further reduce the bounding box, in order to improve the sampling efficiency. In order to use the negative information, we choose squares that properly fit inner circles, formed by lower bounds as depicted by dashed squares in Figure 3.5). After

the exclusion of negative region, the valid sampling area (the shaded region) becomes much smaller [16].

9.4 Sampling

We take candidate samples from the area formed by the intersection of the angular areas. However, the intersection of angular areas may result in non-convex areas or disconnected pieces. Accurate modeling of the intersection is a computationally challenging task. Therefore, we first construct a bounding box that contains the intersection, then using the range information, we reduce the bounding box in order to improve the sampling efficiency and then the candidate samples are drawn uniformly at random within the valid sampling area.

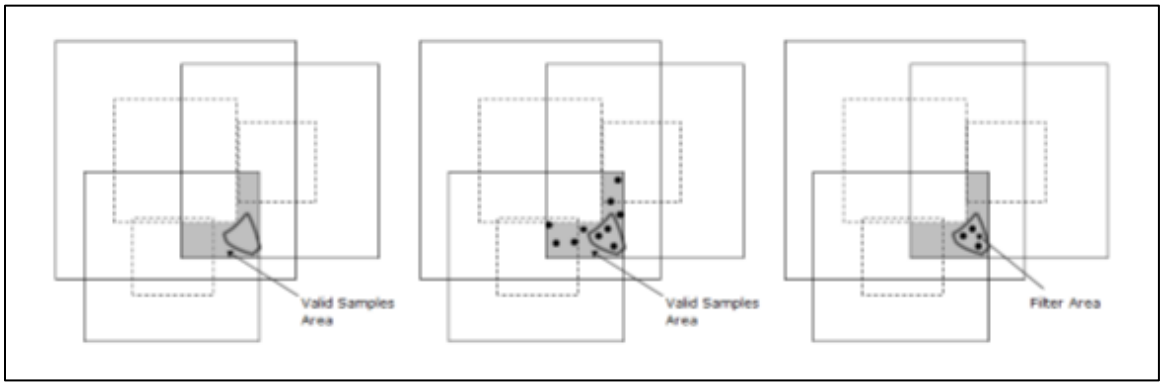

Fig. 25. Sampling

If all the available negative constraints are used, we get a very complex area which is hard to model with good accuracy due to resource constraints and other factors. Further, we employ a heuristic that chooses a seed node whose annulus has the minimum intersection area with the outer bounding box, which improves sampling efficiency and reduces computation. In order to reduce computation complexity, we approximate the area of intersection by using squares to approximate the circles and the area between the squares to approximate the annulus. Hence, we achieve a modified area within the bounding box with a rectangle removed as shown in the left of Figure 3.6.

In the sampling process, we used a simple programming trick to sample uniformly at random using the difference of two rectangles (modified area) without increasing

computational complexity. This technique ensures improved sampling efficiency without negatively impacting the localization accuracy.

9.5 Filtering & weighing the samples

After performing the sampling process, each candidate sample is tested against the available constraints. Samples that do not meet any one of the constraints are rejected or say, filtered out. For example, the samples that are not within ri and ro (lower and upper bounds on distance) from one-hop seeds are removed. Similarly, samples that do not have distance between r and 2r from two-hop seeds are also removed. Thus, the remaining samples are viewed as the possible locations of a non-localized non-seed n as shown in the right of Figure 3.6.

In order to compute error in x and y direction, the smallest axis-aligned rectangle that encloses all the filtered samples: (Xmin, Xmax, Ymin, Ymax) is chosen. If location estimation is (Xe, Ye), the maximum error in the X-direction will be (Xe − Xmin, Xmax− Xe) and the maximum error in the Y-direction will be (Ye − Ymin, Ymax − Ye) as shown in Figure 3.7.

Further, the filtered samples are assigned weights as done in WMCL. It is pertinent to note that each localized non-seed node broadcasts its location estimation and the set of filtered locations, along with the localization error. A non-localized non-seed node takes advantage of neighbour locations and error estimates, and the set of filtered samples of the node itself computed in the previous time step in order to compute and assign weight for each of the samples. For each filtered sample s, a non-seed node n computes weight based on the received information from each one-hop non-seed node k whose error is less than the error of the node n in the previous time step. The sample s is allotted a partial weight w(s,k) as the fraction of samples of k that are within distance $r + V_{max}$ of s. Using the information from all the one-hop non-seeds, the final weight of the sample s is computed as $w(s) = \prod_k w(s,k)$.

9.6 Steps involved in Algorithm

Algorithm iBLA (x)

(∗ runs at each non-seed node x∗)

1. **if** *x* has no neighbours
2. **then** Cannot localize isolated node.
3. **else** create outer bounding boxes B_y for each constraint *y*
4. Create a bounding box *B* from B_y following WMCL
5. Trim and reduce the box *B* by employing the negative constraints
6. Find the seed neighbour z whose annulus has the minimum intersection area with *B*
7. $B \leftarrow B \setminus B_z$
8. Sample *Z* points from box *B*
9. Filter the points employing the neighborhood and non-neighbourhood constraints
10. Compute weight of each sample employing the neighbour error information and previous location samples
11. Compute weighted average of filtered points referred to as u_x by using the weights computed in the previous step
12. Compute maximum distance, v_x from u_x to any filtered sample point
13. Return location u_x and localization error estimate v_x

CHAPTER 10: SECURITY OF IBEACONS: CONCERNS AND SOLUTIONS

10.1 Security Concerns for iBeacons

- **iBeacon Spoofing**: By design, the iBeacon advertisement frame is visible. This means that the interested parties can capture, copy and reproduce the iBeacon advertisement frames at different physical locations. This suggests that the listening devices can capture and reproduce the iBeacon frame. As of Feb 2014, successful spoofing was reported for iBeacons deployed at Apple store [28]. This shall not be viewed as a serious security threat to the iBeacon technology, but application developers must keep this in mind while designing their iBeacon-enabled applications.
- **iBeacon Piggybacking:** The ability to scan for iBeacons and map an entire iBeacon environment allows the competitors or rival organizations to utilize the micro-location information and deliver their own content. This may be explained with the help of an example, Store A may gather all iBeacon IDs from its competitor store, set its own application to listen to those IDs, and deliver its own content, thus making people to leave the competitor store [43].
- **Physical Theft & Damage**: iBeacons deployed at a target venue such as retail store or museum are very much prone to physical theft or damage. iBeacons, if stolen, misplaced, disabled or damaged, can render the application useless for the target customers.
- **Physical Exchange of iBeacons**: Instead of physical damage or theft of iBeacons, someone with malicious intent can tamper with the smooth functioning of the iBeacon-enabled smartphone application even by physical exchange of iBeacons. This may result in irrelevant or misleading content being delivered onto customers' smartphones, which, in turn, shall damage the reputation of the organization.
- **iBeacon Hacking:** If someone with malicious intent changes the Major or Minor value or other characteristics of an iBeacon, that iBeacon device is said to have been

"hacked". Further, any application that has been configured to work with specific iBeacon will stop working [46].

- **iBeacon Impersonation:** Besides being susceptible of hacking, an iBeacon can also be impersonated. Malicious attackers can configure a "fake" iBeacon to impersonate another iBeacon belonging to a retail store, potentially gaining access to deals, promotions, discount vouchers, and other contextual aware information connected with the iBeacon being impersonated [46].

10.2 Security Solutions for iBeacons

Despite all the attention that the iBeacons are receiving from across the technology industry, iBeacons are yet to become market-ready products because businesses have several security concerns regarding iBeacons, especially when a team managed to crack the CES 2014 Scavenger Hunt, a beacon powered app without even being physically present at the venue [46]. Therefore, when it comes to developing apps for micro-location based solutions, a security model should be incorporated that addresses the security concerns and risks involved such as beacon spoofing and piggybacking. It is pertinent to note that the use case shall inform the approach to security. Additionally, the most advanced security solutions include multiple layers of security those are built according to the specific use case.

- **Authorized Access Prevention**: The use of credentials to allow access for changing settings is crucial. Just making it difficult to discover how to connect to the iBeacon to change its settings is not sufficient any more. Instead, a complex password with sufficient length should be employed to prevent people from trying to guess the password. Further, a mechanism that detects unauthorized authentication attempts and initiates progressively longer timeout periods between consecutive authentication attempts must be put in place in order to prevent brute force password attacks [49].
- **Unique Passwords**: All iBeacons should not have the same or similar passwords. Additionally, organizations should have the ability to set unique passwords for each

iBeacon and the iBeacons should not be deployed with default vendor password [49].

- **Encrypted Channel**: All authentication strategies should be implemented over an encrypted channel and passwords should not be extractable from the physical iBeacon itself. Since, Bluetooth pairing to an iBeacon is usually done using Secure Simple Pairing that does not require an out-of-band password, thus it is recommendable to perform initial pairing in an area that is free from unauthorized scanning of the Bluetooth network [49].
- **Provide a mechanism for updates**: Since, security risks and vulnerabilities can be discovered at any given time, it becomes highly essential to receive timely security notifications and firmware updates to ensure the security of iBeacons. Further, iBeacons must be capable of updating their software themselves.
- **Reduce Attack Surface Area**: Since iBeacon is a broadcast only protocol, an iBeacon must be administered it is not acting as a beacon or over a different communication channel, such as another Bluetooth radio. iBeacons should not allow service connections when deployed and broadcasting as an iBeacon. This reduces the attack surface area since an attacker is not able to connect to the iBeacon, thus making it difficult to enact automated attacks [49].
- **Physical Security**: iBeacons are always susceptible to theft, damage, tamper or physical exchange. Thus, iBeacons should be deployed in physically secured environments. Moreover, iBeacons can be placed with other infrastructure equipment that is securely placed in cabinets, behind counters, or physically attached and secured. While attaching to walls may simplify initial deployment of iBeacons, but it makes them an easy target for damage and theft in many non-secured environments [49].
- **GPS Authentication**: The application needs to employ additional authentication measures to ensure whether the detected listening device is actually within the radio range of the iBeacon. This may be achieved by using GPS coordinates of the detected listening device. GPS authentication avoids the inherent risks of beacon spoofing.

- **Web-Based Security**: Another way of enhancing the iBeacon security is by actually delivering the beacon profiles to the compatible application using a web service via a content management system. This ensures that if the iBeacon gets spoofed, it can be easily detected and monitored [50].
- **Encryption of iBeacon identifiers:** As already stated, protection of iBeacons' identifiers is central to any security model that is deployed for iBeacon networks. Thus, to enhance the security of iBeacons, the Major and Minor values (payload) of iBeacon advertising frame must change frequently. Moreover, the payload should be encrypted in order to avoid reverse engineering attempts made on the Major and Minor values. This security strategy has been further enhanced by Sonic Notify which employs a randomly timed encrypted payload that changes i.e. the advertising frame cannot be easily read along with a synchronized rolling UUID and payload. This strategy has been found to be more secure because it is partly server based. enterprise caliber software platform to simultaneously change the value of the UUID along with that of the payload [46].
- **Interspersal of Non-iBeacon Information:** In order to solve the problem of beacon spoofing, BlueCats has adopted a unique strategy where in BlueCats iBeacons intersperse their own proprietary non-iBeacon information into the stream of iBeacon advertisements. Further, any device running BlueCats SDK can utilize this non-iBeacon information to verify that they are actually listening to a BlueCats iBeacon, and not from a spoofed iBeacon [43].
- **Three-Tier Password System**: This 3-tier security strategy has been deployed by Kontakt.io. In the first layer of security, a password-protected Content Management System (CMS) account ensures authorized access to the iBeacons. Further, each iBeacon has two passwords: a Master password that is employed for authorizing major changes such as updating the firmware, and a Beacon password that authorizes changes to the properties of the individual iBeacon. The beacon hijacker or piggybacker requires all three passwords in order to be able to hijack the iBeacons. Furthermore, even if someone is able to extract the iBeacon identifiers and try to clone them, they would still not be able to alter the triggered content without having access to the associated CMS [51].

- **Cloud-Based Authentication:** iBeacons being deployed with a weak security model can be hacked and their UUID, Major and Minor can be easily changed. Thus, iBeacons need to be protected by cloud-based authentication layer. Estimote offers two layers of security: cloud-based authentication that protects iBeacons from unauthorized access and Secure UUID (UUID rotation) that prevents piggybacking on the iBeacon network [54]. iBeacons are secured using cloud-based authentication where in the user needs to connect to Estimote Cloud in order to confirm the ownership of those particular iBeacons, thus protecting the iBeacons from unauthorized access [50].

- **Secure UUID:** It is important to note that protection of iBeacons' identifiers is the key to any security model that is deployed for iBeacons. Secure UUID is one such security measure that prevents beacon piggybacking. Secure UUID employs a mechanism known as UUID rotation. Estimote iBeacons with Secure UUID feature enabled on them periodically change their iBeacon's ID (UUID, major and minor) so that it broadcasts unpredictable and encrypted values. In Secure UUID mode, iBeacons change their ID at a fixed time interval in a pattern that is solely based on a key, which, in turn is safely stored in the iBeacon and cannot be read without access to the associated Estimote Cloud account. This means that even if someone manages to determine the ID's of your iBeacons, they will not be able to piggyback on them, as those ID's keep on rotating to some random value based on the key. Moreover, it is not possible to predict the future ID values without access to the associated Estimote Cloud account. Estimote SDK detects whether Secure UUID is enabled on the iBeacon and automatically handles the decryption task [50, 56].

CHAPTER 11: FUTURE SCOPE

This thesis undertakes a successful attempt to exploit the potential of Bluetooth Low Energy (BLE) for enabling location aware services on handheld devices using BLE-enabled iBeacons. To begin with, an Android-based custom application was developed that interacts with Estimote iBeacons, offering a range of novel functionalities. Further, the issues associated with localization algorithms for wireless sensor networks were investigated. Thereafter, a thorough analysis into the localization aspect of iBeacon networks was conducted. Consequently, a localization algorithm (iBLA) applicable to iBeacon networks was devised. Experimental results demonstrate that the proposed localization algorithm provides a reliable and accurate framework for localization and tracking of iBeacons.

The demonstrated application works currently only for Android-based devices. In the next stage of development, we intend to extend our application to iOS platform as well. Although, the application performs fairly well with iBeacons, better hardware and firmware shall certainly enhance the level of accuracy achieved in localization and tracking of iBeacons. Further, the localization algorithm can provide more reliable and accurate results, if the iBeacons are deployed along with other location-based technologies such as GPS, QR code and WiFi depending upon the exact use case.

REFERENCES

[1] F. Akyildiz and I.H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," in Ad Hoc Networks, vol. 2, no. 4, pp. 351-367, Oct. 2004.

[2] L. Yang and S. Liu, “An Energy-Saving Method for Erasure-Coded Distributed Storage System,” in *Proceedings of the 4th International Conference on Computer Engineering and Networks*, Dec. 4, 2014, pp. 271–278.

[3] M. Rudafshani and S. Datta, ”Localization in wireless sensor networks,” in *Proceedings of the 6th international conference on Information processing in sensor networks*, pp. 51-60.

[4] Estimote, Internet: http://estimote.com/

[5] Bluetooth, “Bluetooth Smart Technology: Powering the Internet of Things,” Internet: http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx

[6] Estimote, “API Documentation,” Internet: http://estimote.com/api/

[7] M. Varsamou and T. Antonakopoulos, ”bluetooth smart analyzer in iBeacon networks,” in *Proceedings of 2014 IEEE Fourth International Conference on Consumer Electronics*, Sept. 7-10, 2014, pp. 288-292.

[8] Jibestream, “Wayfinding, iBeacons, and The Customer Experience Revolution, “ Internet: http://www.jibestream.com/blog/wayfinding-ibeacons-and-the-customer-experience-revolution

[9] M.S. Gast, ”Building Applications with iBeacon: Proximity and Location Services with Bluetooth Low Energy,” in *Building Applications with iBeacons.* Sebastopol, CA, USA: O’Reilly Media, 2014.

[10] J. Bachrach and C. Taylor.”Localization in Sensor Networks”, in *Handbook of Sensor Networks: Algorithms and Architectures,* I. Stojmenovi Ed. Hoboken, NJ, USA: Wiley, 2005.

[11] K.M.D.M. Karunarathna, H.M.D.A. Weerasingha, M.M. Rumy and M.M. Rajapaksha, ”A Fully Functional Shopping Mall Application - SHOPPING EYE,” in *Proceedings of 2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, Nov. 18-20, 2014, pp. 292-296.

[12] R. Faragher and R. Harle, ”Location Fingerprinting with Bluetooth Low Energy Beacons,” *IEEE Journal on Selected Areas in Communications*, vol. PP, pp. 1, May 6, 2015.

[13] M. Kohne and J. Sieck, “Location-Based Services with iBeacon Technology,“ in *Proceedings of 2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, Nov. 18-20, 2014, pp. 315-321.

[14] M. Martin, B.J. Ho, N. Grupen and S. Munoz, ”An iBeacon primer for indoor localization: demo abstract,” in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014, pp. 190-191.

[15] G. Conte, M.D. Marchi, A.A. Nacci and V. Rana, ”BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system,” in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014, pp. 11-19.

[16] M.T. Adnan, S. Datta and S. MacLean, “Efficient and Accurate Sensor Network Localization,” Internet: http://cs-conferences.acadiau.ca/JPUC_SI/2012/Adnan/journal-camera-final.pdf

[17] Android, “Making Your App Location-Aware,” Internet: https://developer.android.com/training/location/index.html

[18] Estimote Blog, “UX for the real world: how should you think about building apps in the age of beacons and Internet of Things," Internet: http://blog.estimote.com/post/101427444795/ux-for-the-real-world-how-should-you-think-about

[19] Howstuffworks, “Is Wibree going to rival Bluetooth?,” Internet:www.howstuffworks.com/wibree.htm

[20] Apple Inc., ”iOS: Understanding iBeacon,” Internet: https://support.apple.com/en-gb/HT202880

[21] ibeaconinsider, “What is iBeacon? A Guide to Beacons,” Internet: http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/

[22] Roximity, Internet: http://www.roximity.com

[23] Estimote Community, “What are region Monitoring and Raning?,” Internet: https://community.estimote.com/hc/en-us/articles/203356607-What-are-region-Monitoring-and-Ranging-

[24] Bluetooth, “The Low Energy Technology Behind Bluetooth Smart,” Internet: http://www.bluetooth.com/Pages/low-energy-tech-info.aspx

[25] Estimote Community, “Estimote Android SDK,” Internet: https://community.estimote.com/hc/en-us/articles/201991918-Estimote-Android-SDK

[26] Estimote Community, “What is a beacon region?,” Internet: https://community.estimote.com/hc/en-us/articles/203776266-What-is-a-beacon-region-

[27] Bluetooth, “Skyrocketing demand for Bluetooth appcessories for latest phones,” Internet: www.bluetooth.com/Pages/Mobile-Telephony-Market.aspx

[28] Stackoverflow, “How to prevent spoofing of iBeacons?,” Internet: http://stackoverflow.com/questions/21955246/how-to-prevent-spoofing-of-ibeacons

[29] AppleInsider Forums, “Inside iOS 7: iBeacons enhance apps' location awareness via Bluetooth LE,” Internet: http://forums.appleinsider.com/t/158126/inside-ios-7-ibeacons-enhance-apps-location-awareness-via-bluetooth-le

[30] Technopark Living, “iBeacon- The game changer in InStore Navigation,” Internet: http://technoparkliving.com/2013/12/ibeacon-the-game-changer-in-instore-navigation/

[31] GitHub, “Estimote / Android-SDK,” Internet: https://github.com/Estimote/Android-SDK

[32] JAMFsoftware,” Five benefits of using iBeacon to distribute or restrict technology,” Internet: http://www.jamfsoftware.com/blog/five-benefits-of-using-ibeacon-to-distribute-or-restrict-technology/

[33] Blog of Adam Warski, “How do iBeacons work?,” Internet: www.warski.org/blog/2014/01/how-ibeacons-work/

[34] Stackoverflow, “iBeacon in the background - Use cases,” Internet: http://stackoverflow.com/questions/19477044/ibeacon-in-the-background-use-cases/19485055#19485055

[35] Estimote Community, “What is Estimote Cloud,” Internet: https://community.estimote.com/hc/en-us/articles/203854516-What-is-Estimote-Cloud-

[36] PennState, Department of Computer Science and Engineering, “Location-Aware Wireless Sensor Networks,” Internet: http://www.cse.psu.edu/pda/SDB/

[37] Mobilepaymentstoday, “SDK will let Android users join the iBeacon party,” Internet: www.mobilepaymentstoday.com/article/221089/SDK-will-let-Android-users-join-the-iBeacon-party

[38] R. Nagpal, H. Shrobe and J. Bachrach, ”Organizing a global coordinate system from local information on an ad hoc sensor network,” in *Proceedings of the 2nd international conference on Information processing in sensor networks*, 2003, pp. 333–348.

[39] Beaconstac Blog, “5 Common Misconceptions around Beacons (and the Reality),” Internet: http://blog.beaconstac.com/2015/01/5-common-misconceptions-around-beacons-and-the-reality/

[40] Estimote Community, “Which devices are compatible with Estimote Beacons?,” Internet: https://community.estimote.com/hc/en-us/articles/201027523-Which-devices-are-compatible-with-Estimote-Beacons-

[41] Twocanoes Blog, “The next iBeacon frontier: Mobile Device Management,” Internet: http://blog.twocanoes.com/post/102894026424/the-next-ibeacon-frontier-mobile-device

[42] Estimote Community, “How do I modify UUID, major and minor values?,” Internet: https://community.estimote.com/hc/en-us/articles/200868188-How-do-I-modify-UUID-major-and-minor-values-

[43] Bluecats Support Center, “Secure Mode and iBeacon + Secure Mode,” Internet: http://support.bluecats.com/customer/portal/articles/1525982-secure-mode-and-ibeacon-secure-mode

[44] Estimote Community, “What is ideal Estimote Beacon placement?,” Internet: https://community.estimote.com/hc/en-us/articles/202041266-What-is-ideal-Estimote-Beacon-placement-

[45] Apple Developer, “Region Monitoring and iBeacon,” Internet: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/LocationAwarenessPG/RegionMonitoring/RegionMonitoring.html

[46] Beekn, “Can Your Estimote Be Hacked? Yes It Can (For Now),” Internet: http://beekn.net/2014/01/can-estimote-be-hacked/

[47] Estimote Community, “What are Triggers and how can I use them?,” Internet: https://community.estimote.com/hc/en-us/articles/205662627-What-are-Triggers-and-how-can-I-use-them-

[48] Estimote Community, “Can I access beacon settings remotely?,” Internet: https://community.estimote.com/hc/en-us/articles/203950176-Can-I-access-beacon-settings-remotely-

[49] Twocanoes Blog, “iBeacon Security Part 2: Beacon Privacy and Security,” Internet: http://blog.twocanoes.com/post/87320735638/ibeacon-security-part-2-beacon-privacy-and

[50] Estimote Blog, “iBeacon security: understanding the risks," Internet: http://blog.estimote.com/post/104765561910/ibeacon-security-understanding-the-risks

[51] kontakt.io, “How we protect our Beacons / iBeacon against piggybacking and hijacking,” Internet: http://kontakt.io/blog/protect-beacons-ibeacon-piggybacking-hijacking/

[52] Estimote Community, “How to update beacon's firmware?,” Internet: https://community.estimote.com/hc/en-us/articles/204040406-How-to-update-beacon-s-firmware-

[53] Estimote Community, “How to share your beacons?,” Internet: https://community.estimote.com/hc/en-us/articles/204576027-How-to-share-your-beacons-

[54] Estimote Community, “Do Estimote Beacons incorporate any security mechanisms?,” Internet: https://community.estimote.com/hc/en-

us/articles/201371053-Do-Estimote-Beacons-incorporate-any-security-mechanisms-

[55] Estimote Community, “Does Estimote offer Analytics for beacons?,” Internet: https://community.estimote.com/hc/en-us/articles/204231993-Does-Estimote-offer-Analytics-for-beacons-

[56] Estimote Community, “How Secure UUID works?,” Internet: https://community.estimote.com/hc/en-us/articles/204233603-How-Secure-UUID-works-

[57] Estimote Community, “How to transfer beacons and manage ownership settings?,” Internet: https://community.estimote.com/hc/en-us/articles/204233613-How-to-access-beacons-and-manage-ownership-settings-

[58] PassbookReady, “Why iBeacon is better than NFC and GPS,” Internet: http://passbookready.com/ibeacon-vs-nfc-vs-gps/

[59] T. S. Rappaport, *Wireless communications principles and practices*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[60] Beekn, “iBeacon and Brands: 5 Approaches to Engaging Customers,” Internet: http://beekn.net/2013/12/ibeacon-brands-5-approaches-engaging-customers/

[61] N. Priyantha, A. Chakraborty and H. Balakrishnan, ”The cricket location-support System,” in *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking*, 2000, pp. 32–43.

[62] GetElastic, “Mobile Commerce: iBeacon vs. NFC [Infographics],” Internet: http://www.getelastic.com/mobile-commerce-ibeacon-vs-nfc-infographics/

[63] Co.Labs, “Why Apple's iBeacon Is Better Than NFC,” Internet: http://www.fastcolabs.com/3017263/why-apples-ibeacon-is-better-than-nfc

[64] RapidNFC, “What Is The Difference Between iBeacon and NFC ?,” Internet: http://rapidnfc.com/blog/100/what_is_difference_between_ibeacon_and_nfc

[65] P. Bahl and V.N. Padmanabhan, ”Radar: an in-building RF-based user location and tracking system,” in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000, vol. 2, pp. 775–784.

[66] M. Sugano, T. Kawazoe, Y. Ohta and M. Murata, ”Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard,” *Wireless and Optical Communications*, 2006.

[67] D. Niculescu and B. Nath, “Ad hoc positioning system (APS) using AOA,” in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003, pp. 1734–1743.

[68] A. Savvides, C.C. Han and M.B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 166–179.

[69] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. Hoboken, NJ, USA: Wiley, 2010, pp. 168–183, 191–192.

[70] K. Sohraby, D. Minoli and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. Hoboken, NJ, USA: Wiley, 2007, pp. 203–209.