4-2016

Ibeacon based proximity and indoor localization system

Faheem Faheem
Purdue University

Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Faheem

Entitled
IBEACON BASED PROXIMITY AND INDOOR LOCALIZATION SYSTEM

For the degree of Master of Science

Is approved by the final examining committee:

Baijian Yang
Co-chair

Ioannis Papapanagiotou
Co-chair

Thomas J. Hacker

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Baijian Yang

Approved by: Jeffrey L. Whitten 4/22/2016

Head of the Departmental Graduate Program Date

IBEACON BASED PROXIMITY AND INDOOR LOCALIZATION SYSTEM

A Thesis

Submitted to the Faculty

of

Purdue University

by

Faheem

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2016

Purdue University

West Lafayette, Indiana

Dedicated to Baba Jan!

ACKNOWLEDGMENTS

I would like to thank Dr. Ioannis Papapanagiotou for helping me start this research. If it were not for him to respond to my email about two years ago, I would have never joined Purdue. He was always prompt in responding to my queries and providing me with the support that I needed to proceed with this research. His energy, work ethic and demanding personality motivated me to work harder and improve myself as a student as well as researcher. I learned a lot from him. I would also like to thank Dr. Baijian Yang for his support and guidance throughout the course of this thesis. Despite the busy schedule, he shared his experience and assisted me with my research. I would also like to express my sincere gratitude to Dr. Thomas J. Hacker for helping me out with thesis. His research experience and approach to tackle different research problems have certainly served as a guideline for me.

I would like to thank everyone in the High Performance Computing (HPC) Lab at Purdue University. Specifically, Saurav Nanda, Jason St. John, Zhihao Yao, and Kelu Diao who helped me a lot with my research and studies. I am fortunate to have found such great friends and amazing lab colleagues. They shared their knowledge and were willing to help me whenever I needed their help. Saurav Nanda was a mentor to me who I always relied on for guidance. Jason facilitated all of us whenever we needed any thing in the lab. Zhihao's research insight and expertise always helped me find the missing piece of the puzzle. Kelu's mathematical genius and joyous personality made the mathematics part of my research fun. I can't thanks them all enough for the amazing two years.

I would also like to thank my teachers from Pakistan including Sir Akhtar, Sir Munir, Sir Hanif Rasool, Sir Zafar Iqbal, Dr. Gul Muhammad Khan, and Dr. Sahibzada Ali Mahmud. If it had not been for them, I would have never made it this far. I would also like to thank all my friends back home whom I have had good time with. I would especially like to thank Mr. Muhammad Irfan and Arif Ali, friends who have been no less than brothers to me. Thanks for all your support, motivation, and patience. I would also like to thank Aaria for her support and help.

I would like to thank Mummy and Papa for their support, love, and care. Had it not been for their numerous sacrifices, none of us (my siblings and I) would have ever made it this far. No matter how tough things would get, I always knew that they were there for me and everything would be fine at the end of the day. Particularly, mummy who has always believed in me and made sure that I work hard and do well in my studies and life as a person. I would like to thank all my siblings Najim, Yasamin, Diva, Samim and Kalsoom for all the love, care as well as fights. Despite all the differences, I love you all. I would also like to thank my nephews Mustafa, Ali and niece Aaliyaah for being a part of my life. The smile on their faces and their joy means a lot. I would like to thank Bibi Jan and Baba Jan for their support, love, care and inspiration. Nothing would have been possible without them and their support. I would like to thank Anisa, Shereen Mama, Khaista Mama, Shahla, Faiza, Humayun, Anu, Akbar Mama, Watan, Laila, and Wara Yasamin for their support and love. Last but not the least, I would also like to thank all my maternal cousins (would have listed your names, but I am sure that will take two pages).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

BLE	Bluetooth Low Energy
EKF	Extended Kalman Filter
IoT	Internet of Things
KF	Kalman Filter
KFPF	Kalman Filter enhanced Particle Filter
LBS	Location Based Services
NSF	National Science Foundation
PBS	Proximity Based Services
PF	Particle Filter
PFEKF	Particle Filter-Extended Kalman Filter
RSSI	Received Signal Strength Indicator
SRA	Server-side Running Average
SKF	Server-side Kalman Filter

GLOSSARY

Beacon: BLE enabled device that utilizes iBeacon protocol.

Geofencing: The process of creating a virtual fence around any entity that helps in detecting the entrance and exit of other entities into the virtual fence (Zafari et al., 2016) is termed as geofencing.

iBeacon: Apple's proprietary protocol that allows the BLE enabled devices to transmit certain messages which are used for proximity based services.

Internet of Things: The ability to connect different entities to each other using the Internet is known as Internet of Things (IoT) (Evangelatos, Samarasinghe, & Rolim, 2012).

Location Based Services: Services that are provided to a tenant based on his location are known as location based services.

Micro-location: The process of locating any entity with an accuracy as high as 10 cm is known as micro-location (Zafari et al., 2016).

Smart Buildings: Buildings that have the capability to provide low cost services such as heating, air conditioning, illumination, ventilation, sanitation, security and a wide range of other services to the users without having an adverse effect on the environment (IBE, 2008).

ABSTRACT

Faheem M.S., Purdue University, May 2016. iBeacon Based Proximity and Indoor Localization System. Major Professors: Ioannis Papapanagiotou and Baijian Yang.

User location can be leveraged to provide a wide range of services in a variety of indoor locations including retails stores, hospitals, airports, museums and libraries etc. The widescale proliferation of user devices such as smart phones and the interconnectivity among different entities, powered by Internet of Things (IoT), makes user device-based localization a viable approach to provide *Location Based Services* (LBS). Location based services can be broadly classified into 1) Proximity based services that provides services based on a rough estimate of users distance to any entity, and 2) Indoor localization that locates a user's exact location in the indoor environment rather than a rough estimate of the distance. The primary requirements of these services are higher energy efficiency, localization accuracy, wide reception range, low cost and availability. Technologies such as WiFi, Radio Frequency Identification (RFID) and Ultra Wideband (UWB) have been used to provide both indoor localization and proximity based services. Since these technologies are not primarily intended for LBS, they do not fulfill the aforementioned requirements. Bluetooth Low Energy (BLE) enabled beacons that use Apple's proprietary *iBeacon* protocol are mainly intended to provide proximity based services. iBeacons satisfy the energy efficiency, wide reception range and availability requirements of LBS. However, iBeacons are prone to noise due to their reliance on Received Signal Strength Indicator (RSSI), which drastically fluctuates in indoor environments due to interference from different obstructions. This limits its proximity detection accuracy.

In this thesis, we present an iBeacon based proximity and indoor localization system. We present our two server-based algorithms to improve the proximity detection accuracy by reducing the variation in the RSSI and using the RSSI-estimated distance, rather than the RSSI itself, for proximity classification. Our algorithms *Server-side Running Average* and *Server-side Kalman Filter* improves the proximity detection accuracy by 29% and 32% respectively in contrast to Apple's current approach of using moving average of RSSI values for proximity classification. We utilize a server-based approach because of the greater computing power of servers. Furthermore, server-based approach helps reduce the energy consumption of user device. We describe our cloud based architecture for iBeacon based proximity detection.

We also use iBeacons for indoor localization. iBeacons are not primarily intended for indoor localization as their reliance on RSSI makes them unsuitable for accurate indoor localization. To improve the localization accuracy, we use Bayesian filtering algorithms such as Particle Filter (PF), Kalman Filter (KF), and Extended Kalman Filter (EKF). We show that by cascading Kalman Filter and Extended Kalman Filter with Particle Filter, the indoor localization accuracy can be improved by 28% and 33.94% respectively when compared with only using PF. The PF, KFPF and PFEKF algorithm on the server side have average localization error of 1.441 meters, 1.0351 meters and 0.9519 meters respectively.

CHAPTER 1. INTRODUCTION

This chapter describes the relevant background, along with significance and purpose of this study. The chapter contains the research question and discusses the scope, assumptions, limitations and delimitations of this study.

1.1 Background

The concept of Internet of Things (IoT) relies on the inter-connectivity of most of the devices or 'things' that improves the automation of different services, resulting in higher user satisfaction. The interconnectivity among different entities allows better networking and assists in leveraging novel technologies for an enhanced life standard around the world. Smart buildings are part of the global movement towards 'smart' entities and aim to fully automate buildings so that the energy efficiency is maximized, the tenant's service level is enhanced, and disaster's are properly managed (Zafari et al., 2015, 2016). The deployment of IoT based systems within smart buildings can improve the overall performance of smart buildings and assist in achieving energy goals in a much efficient and effective way. Location of an entity or a user is of primary importance in smart buildings as well as in IoT based services, as both IoT and smart buildings can provide a number of services to the user based on his location.

Service that are provided to the user based on location are known as *Location Based Services*. The first generation of Location Based Services (LBS) did not garner significant interest due to the network centric approach (Zafari et al., 2015, 2016). However, the second generation of LBS, due to its user centric approach, is successfully penetrating the market especially after the widespread use of mobile devices and smart phones. LBS are classified into:

- Localization: In localization, a user's exact location is obtained to provide different services. The accuracy requirements for such services is high
- Proximity: An estimate of user's proximity to an entity is used to provide effective services to the user. Such services are known as proximity-based services (PBS). The accuracy requirement for proximity is not as stringent as for localization.

LBS are primarily intended to be provided in indoor environments such as hospitals, stadiums, retail stores, libraries, and museums. Technologies, such as Global Positioning System (GPS), are much more suitable for outdoor localization. However, the presence of obstructions and the occurrence of different physical phenomena in the indoor environment makes localization a challenging issue. For GPS, the localization error is about 10 meters. However, indoor environments require the error to be limited, ideally below one meter. The basic requirements for LBS are high accuracy, energy efficiency, wide reception range, cost and availability. Over the last couple of decades, numerous technologies and techniques have been used in the literature such as WiFi, Radio Frequency Identification, and Ultra Wideband with the intention to increase the accuracy of indoor localization and proximity based service. However, there is still no wide-scale adoption of one particular technology or technique, due to the inability to fulfill the aforementioned requirements for LBS in indoor environments. The *iBeacon* protocol, proposed by Apple in 2013, allows BLE enabled minuscule devices known as iBeacons/beacons to provide context-aware proximity based services. iBeacons satisfy the energy efficiency, wide reception range and availability requirements for LBS. However, it violates the cost and accuracy requirements. While cost of the iBeacons will likely reduce, due to the growing interest in the technology, in this thesis, we address the accuracy problem of the iBeacons for proximity detection and indoor localization purposes. The study used different Bayesian filtering algorithms for improving the accuracy of the beacon-based indoor localization and proximity system.

1.2 Significance

Indoor localization is an important part of the future IoT and smart building systems. The user's position, once obtained through indoor localization, can be leveraged to provide a wide range of services such as context aware solutions, targeted advertisements, tenant assistance, automated energy management systems, disaster management and interactive environmental elements etc. For example, a student in library may not be aware of the location of the particular book, which he intends to read. Through localization, his current position will be obtained and then he would be continuously guided towards his destination. Similarly, a smart office building would track the location of the tenant, and turn on the lights, air conditioner and the user's computer when he is in close proximity of his office.

Received Signal Strength Indicator (RSSI) based localization is cost efficient. However, the fluctuation of RSSI, due to the multipath fading, in an indoor environment is a challenge in itself. The BLE enabled iBeacons that utilize the iBeacon protocol are energy efficient and can run for years using a battery cell. Their basic purpose is to provide proximity based services through geofencing i.e., when a specific user is in the vicinity of the beacons, beacons can communicate certain information to the user. Such services are finding wide scale adoption, and numerous companies and brands have adopted beacons in their stores and outlets to increase the sales and to maximize the profits.

However, iBeacons are prone to noise due to the reliance on RSSI and cannot be used for highly accurate LBS. To solve this problem, we utilized Bayesian filtering algorithms that have been widely used in the literature for enhancing the accuracy of indoor localization systems and have provided encouraging results (Gustafsson, 2010; Gustafsson et al., 2002; Guvenc, Abdallah, Jordan, & Dedoglu, 2003; Lee, Oka, Pollakis, & Lampe, 2010; Subhan, Hasbullah, & Ashraf, 2013). The use of Bayesian Filtering algorithms improved the proximity detection and indoor localization accuracy of the iBeacons and resulted in an accurate iBeacon based system that could be used for efficient LBS.

1.3 Problem Statement

Proximity and indoor localization services can provide a wide range of services. Different technologies including UWB, WiFi, and RFID have been used for providing such services. However, most of these technologies are not primarily intended for proximity and indoor localization services. Therefore, these technologies do not fulfill the primary requirements of LBS. iBeacon, based on BLE, is a promising technology for PBS. However, it utilizes RSSI values to provide location-based services. The RSSI values are prone to noise and interference that drastically affects the performance of iBeacon-based proximity detection system. If the accuracy of the iBeacon based PBS is improved, it can be used in hospitals, malls, stadiums and numerous other indoor environments to enhance user experience. Although iBeacons are not primarily intended for indoor localization, an improvement in the accuracy of the iBeacons will also make it suitable for indoor localization. Hence, iBeacons with an improved accuracy can be used to provide LBS, i.e. proximity and indoor localization services.

1.4 Research Question

- Will the use of Kalman filter and Extended Kalman filter in cascade with Particle Filter improve the localization accuracy of an iBeacons-based indoor localization system when compared with use of only Particle filter?
- Is it possible to improve the proximity detection accuracy of an iBeacon-based proximity detection system using our proposed Server-side algorithms when compared with the current approach used by Apple?

1.5 Scope

iBeacons use RSSI values for indoor localization and proximity based services. The problem with RSSI based localization and proximity based services in general (e.g., fluctuation due to multi-path fading) and more particularly the reliance of the beacons on 2.4 GHz Bluetooth technology adversely affects the accuracy of the beacon based proximity and indoor localization system. The literature contains a number of different filtering algorithms that are designed to filter out the noise and improve the performance of the sensors (iBeacons in our case) (Bergman, 1999; Gustafsson, 2010; Gustafsson et al., 2002; Guvenc et al., 2003; Lee et al., 2010; Paul & Wan, 2009; Savic, Athalye, Bolic, & Djuric, 2011; Subhan et al., 2013). Usually filters that consider the noise to be non-Gaussian and the system as non-linear produce the best localization accuracy possible. This research study tracked a user equipped with an iPhone and a prototype iOS application within an indoor environment by using our iBeacon-based localization and proximity system. For obtaining an accurate indoor localization system, different Bayesian recursive filtering algorithms were applied to reduce the RSSI variation and sensor noise. Particle Filtering performs best for indoor localization (Gustafsson, 2010; Gustafsson et al., 2002; Lee et al., 2010). However Kalman filters, and Extended Kalman Filter (EKF) will be cascaded with Particle Filters in order to verify whether it would result in any significant performance improvement over a single particle filter system. For obtaining an accurate proximity detection system, server-based algorithms are used. Also, the topology of the beacons was changed to understand the effect of the beacon's location on localization accuracy. The study also describes an architecture for iBeacon based proximity services and showcase two different use cases.

1.6 Assumptions

The assumptions of this study are

- Beacons/iBeacons work with Apple devices running iOS 7 or later as well as Android devices running 4.3 or later versions of Android. These operating systems have the capability to listen to the messages transmitted by beacons.

It is assumed that any user who wants to utilize the services has a device that runs any of the aforementioned operating systems.

- An iOS application developed specifically for iBeacon based services is used in this study. It is assumed that any potential user who wants to use the services will have the iOS application installed on the device.

- iBeacons are Bluetooth Low Energy (BLE) based devices. It is assumed that for a potential user to utilize the service, the Bluetooth service should be enabled on the user device.

- The iBeacons transmit messages in a specific format containing certain beacon related information. It is assumed that the beacon related information is hard-coded into the application (as per Apple's Application Store Policy (Zafari & Papapanagiotou, 2015)).

- The iBeacons transmit messages at random periodic intervals. It is assumed that the beacons are transmitting at highest possible frequency for the purpose of this research study.

1.7 Limitations

The limitations of this study are:

- Received Signal Strength Indicator (RSSI) will be used to determine the position of a user.

- The value of RSSI will fluctuate due to multi-path propagation and interference from devices working at 2.4GHz.

- The average of RSSI values, obtained through Apple's CoreLocation framework will be used.

1.8 Delimitations

The delimitations of this study are:

- Only one user will be tracked for the purposes of this research study.
- The user will be tracked within a pre-defined indoor area whose dimensions are hard-coded in the iOS application as well as server side implementation.
- The user's position will be tracked in a 2-Dimensional (x, y) Cartesian plane while the z-axis will be ignored for the purpose of this study.

1.9 Summary

In this chapter, we discussed the background, significance, problem statement, research questions, scope, assumptions, limitations, and delimitations of this research study. In next chapter, we provide a detailed review of the literature relevant to this research study.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

The mammoth growth in the field of Information Technology (IT) and networking over the last couple of decades have resulted in a wide range of effective and efficient services that aim to enhance the overall user experience around the world. The ever increasing reliance on technology, the need for better connectivity, and automated systems have served as the impetus behind the advent of the IoT.

Connecting different 'things' can provide a number of useful services without the need for human intervention as described in Zafari et al. (2016). Such autonomous connectivity can serve as an asset in various other novel concepts, of which smart buildings is one. Smart buildings, like IoT, are intended to improve the service level to its tenants while mitigating the possible problems such as disasters and pollution caused by energy consumption etc. The use of LBS is in-line with goals of IoT and smart buildings. Indeed, a user's location can be leveraged to further improve the effectiveness of IoT and smart buildings. Various automated systems can perform their tasks much more efficiently, once they know the user's location by turning *on* or *off* various appliances and equipment based on the user's location.

This chapter provides an overview of IoT, smart buildings, iBeacons, proximity-based services, localization and Bayesian Filtering. The chapter also discusses relevant literary work related to proximity based services and localization and focuses on the past work done for improving the accuracy of indoor localization.

2.1 Internet of Things

Kevin Ashton was the first one to use the term 'Internet of Things' in the context of supply chain management (Ashton, 2009). However, it gradually

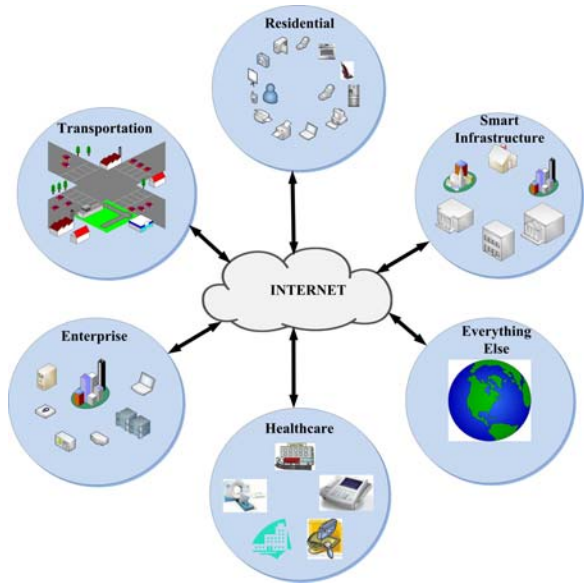

Figure 2.1. IoT: Connecting everything through the Internet

encompassed a wide range of services including transportation, health care, and e-commerce. The core idea of IoT is that different 'entities' or 'things' such as smart phones, senors, actuators or physical items embedded with sensors are connected to each other. The connectivity is leveraged to automate systems and provide better services (Giusto, Lera, Morabito, & Atzori, 2010). IoT is gaining the attention of researchers and different industries around the world. IoT relies on the core principal of connectivity, networking, and sensor networks. Another important field that IoT is linked with is the 'Big Data' (i.e., equipping every single 'thing' with sensors will generate huge amount of data). Figure 2.1 (taken from our papers Zafari et al. (2015, 2016)) presents an insight into the wide range of services that IoT can provide.

Various wired and wireless standards exist that can support the interconnection of devices (Vermesan & Friess, 2013). Table 2.1 (taken from our papers Zafari et al. (2015, 2016)) highlights some of the renowned wireless standards. IoT can provide both residential, and commercial services and application such as e-marketing, e-health (Atzori, Iera, & Morabito, 2010),

Table 2.1 *Wireless Technologies used in IoT*

Technology	Range	Frequency	Throughput
Bluetooth	$\sim$ 100m	2.4 GHz	24.0 Mbps
ZigBee	$\sim$ 100m	2.4 GHz	2-250.0 Kbps
IEEE 802.11a	5000m outdoor	5 GHz	54 Mbps
IEEE 802.11b	140m outdoor	2.4 GHz	11 Mbps
IEEE 802.11g	140m outdoor	2.4 GHz	54 Mbps
IEEE 802.11n	250m outdoor	2.5/5 GHz	600 Mbps
IEEE 802.11ac	35m indoor	5 GHz	1.3 Gbps
IEEE 802.11ad	couple of meters	60 GHz	4.6 Gbps
WiMAX	depends on cell	2.3, 2.5 and 3.5 GHz	365 Mbps downlink, 376 Mbps uplink
LTE	depends on cell	2600 MHz	300 Mbps downlink, 75 Mbps uplink
LTE-Advanced	depends on cell	4.44-4.99 GHz	1 Gbps downlink, 500 Mbps uplink
UWB	10-20m	10.6 GHz	Upto 480 Mbps
RFID	upto 200m	upto 10 GHz	Upto 1.67 Gbps
WiFi Direct	upto 200m	2.4/5 GHz	Upto 250 Mbps

intelligent transportation systems (Bayram, Michailidis, Papapanagiotou, & Devetsikiotis, 2013; Bayram & Papapanagiotou, 2014; Weiland & Purser, 2000), intelligent car parking guidance systems (Faheem, Mahmud, Khan, Rahman, & Zafar, 2013), logistical automation and a wide range of other services. IoT is poised to penetrate deep into the consumer market and a wide range of industries such as appliances, furnitures, and food packaging are going to utilize IoT by the year 2025. Because IoT is in its relative infancy, it faces certain challenges such as device smartness, privacy, security, device interoperability, energy consumption, network addressing the device processing ability (Atzori et al., 2010). The growing interest in IoT has served as the impetus for research initiatives around the world. In 2005, the European Commission started IoT technologies related initiatives (Yan, Zhang, Yang, & Ning, 2008). Similarly the National Science Foundation (NSF) in USA included IoT as a part of the cyber physical systems.

IoT has a lot of potential and it can be applied to a number of applications. The current services provided by IoT are limited and in future, its application would penetrate into offices, residential areas, industrial complexes, shopping malls, research labs, hospitals, libraries, gyms and many more. IoT, in particular, can help the transition towards 'smart' entities such as smart nations, smart cities, smart buildings etc. Solutions that can improve the performance of the aforementioned smart entities will certainly provide great revenue and would assist in raising the standard of living for the tenant. The user would be able to interact with smart environments through IoT, hence providing a better experience to the tenant. Atzori et al. (2010) presented a detailed survey related to IoT. The paper takes the complexities involved in IoT, due to the wide range of technologies, techniques and devices, into account. Various IoT paradigm related visions are discussed. The current challenges of IoT are highlighted and effective solutions are proposed. Miorandi, Sicari, De Pellegrini, and Chlamtac (2012) also presented a survey of different technologies, possible applications, and the challenges of IoT. Because IoT relies on connectivity of different devices, security and privacy are the main concerns with the widescale adoption of IoT. The IoT architecture must be resilient to external and internal threats by providing mechanisms for data authentication, client privacy and access control (Weber, 2010). Furthermore, there is a need for legislation that can handle the security and privacy issues. Such initiatives will certainly bolster the adoption of IoT. Atzori et al. (2010), Miorandi et al. (2012), and Weber (2010) presented a detailed discussion on the possible security threats to IoT and each proposes general solutions.

Zafari et al. (2016) provide a detailed survey of techniques and technologies that can support indoor localization for IoT equipped smart buildings. Novel concepts such as micro-location and geofencing are discussed in the context of IoT and smart buildings. The current and envisioned services that indoor localization can provide to the IoT equipped smart buildings, are discussed. Their paper addresses numerous challenges that the adoption of micro-location and indoor localization technologies face when deployed in IoT equipped smart buildings. The challenges surrounding IoT can be solved by extensive research. Therefore, the research community can play their role in addressing these challenges, particularly the security and privacy concerns.

2.2 Smart Buildings

Buildings that have the capability to provide low cost services such as illumination, heating, air conditioning, and security, without affecting the environment adversely are known as smart buildings (IBE, 2008). This definition of a smart building by the Institute of Smart Buildings (IBE, 2008) is pretty restricted that is, it does not highlight a wide range of other features of smart buildings including automation of the entire system, connectivity among appliances, or energy generation capability. Just like IoT, the motive behind smart buildings is to provide better services to the user. For example, when an employer enters the lobby of his office building, the temperature and lighting in his cabin are automatically adjusted as per the employer's choice, the computer is turned on so that once the user gets there, and he can start working immediately without wasting any time (Snoonian, 2003).

Smart buildings leverage the interconnectivity of various automation systems for better disaster management and emergency services. In case of a fire, the sensors can alert the ventilation system to turn off the fans so that the smoke cannot spread to other parts of the building. The damage as a result of the 2001 attack on Pentagon could have been much more catastrophic in the absence of the advanced automation system (smart building) (Snoonian, 2003). Smart buildings rely on intelligently designed systems and utilize different interconnected subsystems. These interconnected systems facilitate the sharing of valuable information. This results in optimized building performance and an improved interaction between the tenants and building as well as between two different smart buildings. One of the important characteristics of smart buildings is that it generates its own power through a number of renewable resources and does not rely on the power grids. Indeed, the building grid can produce extra energy that can be sold to the adjacent buildings and power grids. The basic characteristics of a smart building are:

- Interconnectivity of business systems.
- Provision of technology to the tenants.
- Connectivity with other smart buildings and smart grids.
- Energy Efficiency.

Figure 2.2 shows the components of a smart building. Different components such as smart appliances, smart meters, building automation etc. are all interconnected and work under the umbrella of smart buildings.

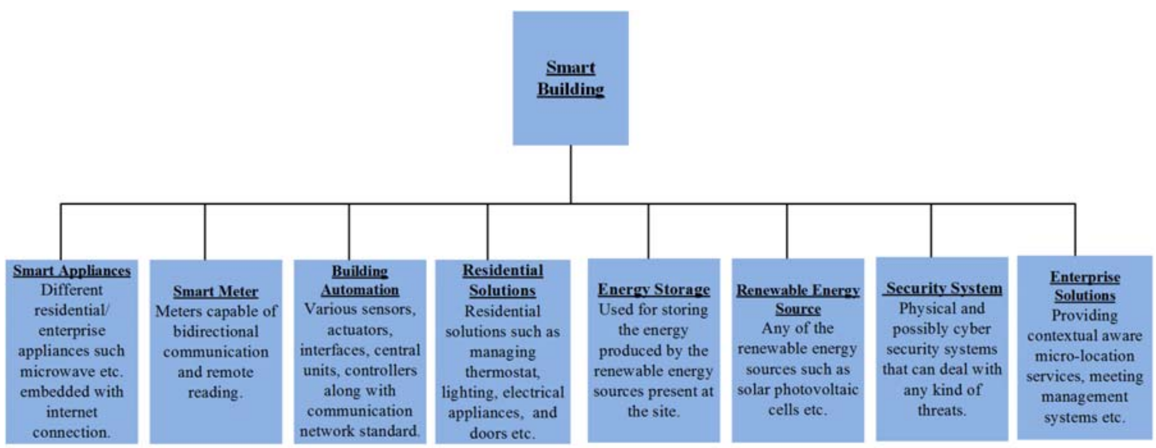

Figure 2.2. Components of a Smart Building.

The research community has worked on improving the architectural design of smart buildings that would allow incorporating technologically advanced systems into the present architecture. The two most famous open communication standards for building automation are

- Building Automation and Control Networks (BACnet): BACnet is a communication only standard and is concerned with the electrical and mechanical systems. BACnet can also work with certain hardware as well. BACnet defines the level of priority for accomplishing the assigned tasks such as which certain signals would take priority in the event of a disaster.

- Local Operating Networks (LonWorks): LonWorks combines hardware and communication and has been widely used in both transportation and utility industry. Recently, LonWorks has been used in buildings. LonWorks relies on different channels assigned to specific signals.

The use of such standards causes a decrease in operation costs (Snoonian, 2003). Wireless Sensor Networks can assist the automation and control processes in smart buildings. The utilization, interconnection and collaboration of the sensors is also bringing IoT into the limelight. Schor, Sommer, and Wattenhofer (2009) utilized wireless sensor networks in smart buildings for enhancing building energy efficiency without affecting the service level. A web services based approach is used to integrate various sensors and nodes into an IP-based network. The proposed approach enables automatic service discovery by implementing a representational state transfer (REST) based application programming interface (API) for accessing the sensor node services. Kleissl and Agarwal (2010) discussed smart buildings in the context of cyber-physical systems and present an analysis of the opportunities provided by jointly optimizing the energy consumption of smart building, both by the processing equipment and occupants. The energy use of different buildings is thoroughly investigated based on which energy saving strategies are proposed for smart buildings. Because energy efficiency is one of the core impetuses behind smart buildings, most of the literature revolves around energy efficiency of smart buildings. Majumdar, Albonesi, and Bose (2012) proposed energy aware meeting scheduling algorithms for smart buildings that can assist in making the building energy efficient. The proposed algorithms take the meeting related parameters into account to schedule the meeting in a room that optimizes the energy efficiency based on the number of occupants, meeting requirements, and time conflicts. The algorithms range from greedy to heuristic ones resulting in a 70% lower energy consumption.

Han, Gao, and Fan (2012) proposed a technique for determining the indoor environment quality (IEQ) and occupancy in a smart building. Statistical estimation methods are used to enhance the measurements obtained from a distributed network of sensors. Data such as carbon dioxide (CO_2) concentration, and relative humidity are obtained through different sensors. The occupancy is modeled using Autoregressive Hidden Markov Model (ARHMM) on the basis of sensor measurements. The proposed model represents the occupancy and IEQ with an average accuracy of 80.78%. Such a model can facilitate the services provided by smart buildings. Firner, Moore, Howard, Martin, and Zhang (2011) proposed *Octopus*, which is an open-source system for supporting data management and fusion for the IoT based applications. The system takes into account the reliance of the smart buildings on sensor data that has to be associated with physical objects. Meyer and Rakotonirainy (2003) presented a detailed survey of the research and strategies that can enhance the diffusion of the IT into smart buildings and homes so that it could improve the overall services provided to the tenants.

Despite the abundance of literature on smart buildings, there is still need for further research that could explore the applications of location based technologies in smart buildings and how context-awareness can further improve the services provided to the tenants. The incorporation of IoT into smart buildings and leveraging localization for providing a wide range of services is yet to be thoroughly explored.

2.3 iBeacons

iBeacon is Apple's proprietary protocol that enables BLE enabled devices (known as beacons) to transmit messages that can be used for proximity-based, context aware solutions. *iBeacon* is the protocol while the devices are known as beacons. However the term iBeacon is also used for the devices. The protocol was included as part of iOS 7.0 in World Wide Developer Conference (WWDC) in 2013 and is part of all the subsequent iOS versions released by Apple. While Android 4.3 and later versions do support iBeacon protocol, Google recently launched its own beacon based platform known as *Eddystone*.

iBeacon requires that the Bluetooth on the user device to be turned on. The beacons are passive devices that can only transmit specific messages at periodic intervals. The user device receives these messages and utilize the signal strength of the transmitted messages to approximate the user's proximity to a specific iBeacon. Due to reliance on BLE technology, they are energy efficient and the devices can run on single coin battery for years. The range of the iBeacons depend on the transmission power. The higher the transmission power, the higher the range and energy consumption. Due to energy constraints and the performance issues, it is preferred to keep the range of the iBeacons low. Also, the higher the transmission frequency, the better will be the performance and lower will be the energy efficiency. Each iBeacon message contains:

- Universally Unique Identifier (UUID): It is a 16-byte mandatory field. The UUID, as per Apple's Application Store policy, must be hard-coded into the application. Conventionally, UUID for the iBeacons of a specific company or store is kept the same. For example, the UUID values of the iBeacons utilized in a store X through the world would have the same UUID.

- Major value: The major value is 2-byte optional field. For a store X in city Y, the major value of all the iBeacons would be same as per the convention.

- Minor value: The minor value is also a 2-byte optional field. It is used to classify the department Z of store X in city Y.

The iBeacons/beacons specifically perform the task of:

- Distance Measurement: The RSSI values from the iBeacons are used to calculate the distance of the user to a specific iBeacon. RSSI values also highlight the accuracy of the obtained estimation results.

- Ranging: The process of classifying the proximity of the user to a specific beacon in any of the four ranges listed in Table 2.2 is known as ranging.

Apple's *CoreLocation* framework averages the RSSI values and then reports them in order. The moving average is intended to reduce the fluctuations in the RSSI.

Table 2.2 *iBeacon Proximity Zones*

Range	Distance
Immediate	Less than a meter.
Near	Between 1-3 meters.
Far	Greater than 3 meters
Unknown	Either user is not in the range of beacon or there is a need for initiating ranging.

Despite its relevant infancy, the technology has attracted wide-scale attention. Numerous companies have started providing beacon based services. Table 2.3 (quotations from our papers (Zafari et al., 2015, 2016)) lists some of the companies that provide beacon hardware or beacon based services.

To benefit from beacon-based services, the user device bluetooth should be turned on and a specific application equipped with beacon service should be installed on the device. The user device is then capable of receiving messages from the beacon. In a typical iBeacon system, the user device communicates with the iBeacons and obtains UUID related information. It forwards that to a server, which communicates with the database for beacon authentication or any other information. The server might then respond back to the user or contact an actuator, based on user's location as shown in Figure 2.3. A detailed discussion on beacons is presented in Zafari et al. (2015, 2016) while recommendations on the deployment of beacons are presented in Zafari and Papapanagiotou (2015).

Table 2.3 *Beacon based Service Providers (Quotations from Zafari et al. (2015, 2016))*

Company	Service Description
Gimbal	"*Gimbal's beacons provides a context-aware advertising platform.*"
Onxy Beacon	"*Onxy beacon offers a beacon management system that is cloud based and helps in building micro location enabled applications for the beacons.*"
Swirl	"*Swirl uses iBeacons and provides end-to-end mobile marketing platform within a store. It is an enterprise grade platform that helps to create, manage and optimize the LBS based mobile marketing.*"
Sonic Notify	"*Sonic Notify uses beacons to provide enterprise proximity solutions.*"
Estimote	"*Estimote utilizes beacons for creating new and contextually rich mobile services.*"

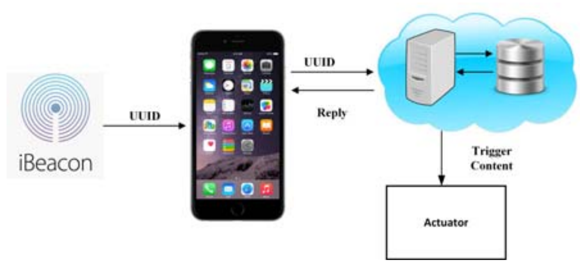

Figure 2.3. Working principal of the beacon

2.4 Proximity Based Services

The type of services that are provided to a user based on proximity to any other entity is known as proximity based services (Mascetti, Bettini, Freni, Wang, & Jajodia, 2009). Rather than the exact position of a user, proximity based services require a rough estimate of the user's location. It is a special type of LBS and has attracted the attention of different industries. Different technologies such as Wi-Fi, Bluetooth, Zigbee, LTE, and RFID can be used for proximity based services. Mascetti et al. (2009) discussed the privacy concerns affiliated with proximity based services. Different preservation techniques that can serve as a trade off between privacy and the quality of service are proposed. The feasibility of the proposed approaches is highlighted through theoretical and experimental results. Meier, Cahill, Nedos, and Clarke (2005) discussed proximity based services in the context of ad-hoc networks. The areas or proximities where specific services are provided by the service providers are advertised, to which the clients can subscribe. Once the clients reach an area that offers the services of their interest, they are notified about the offer using ad-hoc networks. Treu and Küpper (2005) addressed the privacy challenges of proximity-based services. Different techniques are proposed that can allow the user to trust proximity-based services. Users or peers that share the same geographical location and time are matched using a central server, while simultaneously keeping the user privacy intact. The proposed technique utilizes a combination of cryptography and '*k-anonymity protect*' for privacy protection.

2.5 Localization

The process of locating the position of any entity is known as localization. Once the entity/user's position is obtained through localization, a number of different services can be provided. As mentioned earlier, the first generation of LBSs were network centric (i.e., the basic motive behind them was to assist the network and enhance the network and system performance). Because there was no incentive for the user in such a system, the first generation of LBS did not attract significant research as the user's participation is mandatory. The second generation of LBS is user-centric and is intended to provide a wide range of services to the user, based on location. Therefore, LBS has found its use in a number of different industries.

To utilize LBSs, the first step is to determine the user's location with high accuracy. The user can be in an outdoor or indoor environment. GPS is one of the most widely used outdoor localization systems that can attain an accuracy as high as 10 meters (LaMarca et al., 2005; Zafari et al., 2016) and it can provide a number of solutions, with navigation being one of the most widely used ones. Outdoor localization has been widely researched in the context of WSNs (Sichitiu, Ramadurai, & Peddabachagari, 2003), robotics (Kais, Dauvillier, De La Fortelle, Masaki, & Laugier, 2004; Lingemann, Surmann, Nüchter, & Hertzberg, 2004), asset tracking, and navigation etc. Though there is still room for improvement of outdoor localization, this research work is mainly focused on indoor localization. Interested readers are referred to other studies (Bulusu, Heidemann, & Estrin, 2000; Kais et al., 2004; Lingemann et al., 2004; Sichitiu et al., 2003) for further discussion on outdoor localization.

Research related to indoor localization has seen an increase in interest over the last couple of decades (Liu, Darabi, Banerjee, & Liu, 2007). Numerous techniques have been proposed for indoor localization. In the following sub-section, we discuss some of the indoor localization techniques.

2.5.1 Indoor Localization Techniques

A number of different technologies can be used for indoor localization. Some of the most widely used technologies are Bluetooth, Ultra-wide Band (UWB), Magnetic Field Mapping, RFIDs, WiFi, and Zigbee. Table 2.4 lists different technologies and their accuracy in indoor localization. Different localization techniques can be applied to these technologies to formulate an indoor localization system. Based on the technique used, indoor localization algorithms and measuring principles are divided into a) Triangulation b) Scene Analysis and c) Proximity. In the subsections that follow, each is discussed in detail.

2.5.1.1. Triangulation

In triangulation, the location of the target is estimated using three different dimensions (Liu et al., 2007). The two main derivations of triangulation are:

Table 2.4 *Positioning technologies*

Technology	Accuracy
UWB based localization	0.1m
Magnetic Field Mapping	0.1m-2m
Hybrid of RF and IR, or RF and Ultrasonic technologies	1m
WLAN	3m
Zigbee	3m
GPS	10m outdoor

- *Lateration:* In lateration, the distance between the object and different references is measured to obtain an estimate of the target's location. Lateration is also known as range measurement technique.

- *Angulation:* In angulation, the angles formed between the target and the reference points is used to estimate the target's position.

Both lateration and angulation are discussed in detail below.

2.5.1.1.1. Lateration

There are different types of lateration that are listed below.

a) *Time of Arrival (TOA):* In TOA, the distance between the target and the reference points is directly proportional to the propagation time of the signal between the two points (Liu et al., 2007). For locating an object in a two-dimensional environment, a TOA based system requires the signals at least from three reference points. Figure 2.4 depicts TOA based object localization. X, Y, and Z are the reference points with respect to which the user location is to be tracked. The one way propagation time between the reference point and target is used as the indicator of the distance between the two. TOA requires precise time synchronization between the system transceivers (Liu et al., 2007). The transmitted signal contains a mandatory time stamp that helps in determining whether the signal is line of sight (LOS) or reflected/diffracted. The position of the target can be found through a number of different methods. The simplest method is to use principles of geometry to find the points where all the TOA circles intersect. The least squares algorithms (Fang, 1990; Kanaan & Pahlavan, 2004) constitute the alternate method for TOA based localization, in which the position of the target is obtained by minimizing the sum of the squares of the non-linear cost function (Liu et al., 2007). Least square algorithms based method assumes that any target present at (x_0, y_0) transmits a signal at time t_0 so the J reference nodes present at $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots.., (x_J, y_J)$ receive the signal at time $t_1, t_2, t_3, \ldots..., t_J$. The cost function (Liu et al., 2007) is shown in Equation 2.1

$$C(x) = \sum_{l=1}^{J} {\beta_l}^2 c_l(x)^2 \qquad \text{(Eqn. 2.1)}$$

The β_l depends on the signal reliability received at unit l while $c_l(x)$ can be calculated as shown in Equation 2.2

$$c_l(x) = v(t_l - 1) - \sqrt{(x_l - x)^2 + (y_l - y)^2} \qquad \text{(Eqn. 2.2)}$$

v denotes speed of light that is 3.0 x 10^8 $meters/second$ while $\mathbf{x} = (x, y, t)^T$. The function can be obtained for any measuring unit $l = 1, 2...., J$. $c_l(x)$ becomes zero for specific values of x, y and t. The minimization of $C(x)$ results in the estimated location.

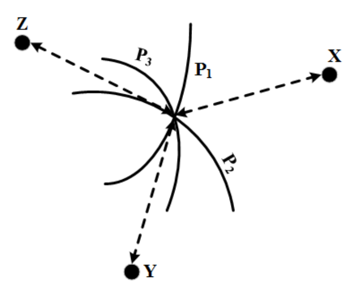

Figure 2.4. TOA/RTOF based localization of object.

Chan, Tsui, So, and Ching (2006) proposed a residual test that can mitigate for the localization error caused by the non-line of sight (NLOS) measurements. The proposed test identifies the LOS reference nodes and their quantity. The identified LOS reference nodes or base stations are used for localization. The principle behind the residual test is that for measurements which only consist of LOS, the normalized values of the residuals will follow a central Chi-square distribution. However when NLOS measurements are present, then the normalized residuals follow a non-central Chi-square distribution. The simulation results highlight the accuracy of the proposed residual test as it can identify the correct number of LOS reference nodes with an accuracy over 90%. Xu, Ding, and Dasgupta (2011) proposed two novel convex optimization methods based on semi-definite programming (SDP) relaxation for the purpose of direct localization. Their proposed method reduces the susceptibility of the three decibel (dB) noise enhancement and does not need prior information regarding the transmission time of the signal. The results show the improvement in performance of the TOA based localization in the presence of noise. Humphrey and Hedley (2008) used super-resolution algorithms for analyzing the TOA measurements and compare it with the traditional inverse Fourier transform. The results show that the Fourier transform based approach is better than the super-resolution algorithms. It is also shown that most of the TOA information stored in the signal exists in a couple of coefficients on the impulse response's leading edge. Based on the findings, the authors propose using a pattern matching technique for obtaining TOA as it outperforms other contemporary techniques.

b) *Time Difference of Arrival (TDOA):* TDOA can be used to obtain the estimated position of the target by examining the time difference between the arrival of a signal at different reference points (Liu et al., 2007). TOA differs from TDOA as TOA relies on absolute time of arrival while TDOA deals with the difference of time of arrival. For TDOA measurements, the transmitter have to lie on the hyperboloid while the range of the measuring units must have a constant difference. The hyperboloid can be mathematically expressed as shown in Equation 2.3

$$R_{l,m} = \sqrt{(x_l - x)^2 + (y_l - y)^2 + (z_l - z)^2} - \sqrt{(x_m - x)^2 + (y_m - y)^2 + (z_m - z)^2}$$
(Eqn. 2.3)

The (x_l, y_l, z_l) and (x_m, y_m, z_m) represent the stationary receivers l, m and x, y, z represent the target's coordinates. Equation 2.3 can be solved by using a Taylor-series expansion and then formulating an iterative algorithm. Alternative methods to obtain solution are non-linear regression and correlation techniques (Liu et al., 2007). Figure 2.5 shows how TDOA based measurements can be leveraged to obtained the target's location in a two-dimensional space. The measurements at point (X, Y, Z) result in two different hyperbolas that provide the target W's location.

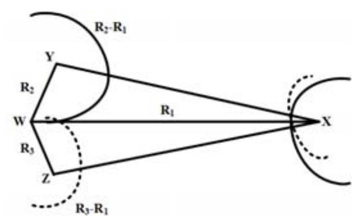

Figure 2.5. TDOA based localization of object.

Gustafsson and Gunnarsson (2003) attempted to solve the uncertainty in TDOA measurements through the use of a Monte Carlo based method for localization and a gradient search algorithm that utilizes a non-linear least squares framework. The Monte Carlo based method can be extended for a dynamic framework that includes the transmitter's motion model. Yang, An, Bu, and Sun (2010) proposed a novel algorithm for TDOA-based source localization where the signal is received at spatially distributed sensors. The algorithm uses the constrained total least-squares (CTLS) technique, while a numerical solution is obtained through a Newton method based iterative technique. Perturbation analysis is used to derive the bias and covariance of the proposed algorithm. Simulation results highlight the fact that despite lower computation cost, the proposed algorithm results in sufficient localization accuracy. Furthermore, the algorithm is robust to large measurement noise when compared to contemporary algorithms. Young, Keller, Bliss, and Forsythe (2003) discussed TDOA based localization for ultra-wideband (UWB) system. Cross correlation based TDOA method is used in combination with a novel technique that combines TDOA estimates from a number of antenna pairs in order to estimate the transmitter's location. Zhang, Kuhn, Merkl, Fathy, and Mahfouz (2006) also used TDOA for UWB based localization. The system results in sub-centimeter accuracy and complies with the FCC's UWB regulations. The method also achieves better ranging measurements. Using the time-domain measurements, multi-path signals can be suppressed that can further improve the accuracy to sub-millimeters (mm).

c)*Received Signal Strength (RSS):* Multi-path effects impact the performance of TDOA as well as TOA, since they rely on signal's arrival that is affected by refraction, reflection and diffraction (Liu et al., 2007). This affects localization accuracy in an indoor environment. An alternative to both TOA and TDOA is using the strength of the signal to estimate the distance of an entity to any reference point that is called Received Signal Strength (RSS). RSS based methods rely on calculating the propagation induced signal path loss. The literature contains various theoretical and empirical models for estimating the distance between the reference point and target using RSS. Figure 2.6 depicts how RSS can be used for object localization. PL_1, PL_2 and PL_3 are the path loss.

Drastic shadowing and multi-path fading that exist in indoor environments can affect the reliability of the path-loss models. The parameters used in the path-loss model vary from site to site, so obtaining a generalized model with specific set of values is not favorable. The performance can be significantly improved by utilizing pre-measured RSS contour values centered at the receiver. Various other alternatives can also improve the performance of the RSS based systems such as utilizing multiple measurements at different reference points or fuzzy logic algorithms (Teuber, Eissfeller, & Pany, 2006).

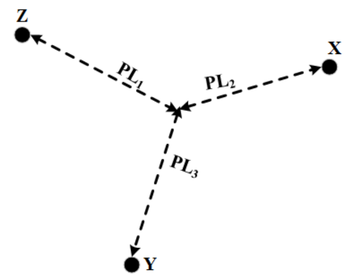

Figure 2.6. RSS based localization of object.

Kaemarungsi and Krishnamurthy (2004) discussed RSS properties of IEEE 802.11b wireless network interface cards (NICs). The paper presents a thorough analysis of the data that assists in understanding the basic features of location fingerprinting. The authors have also compared a Gaussian model with RSS based indoor localization systems in terms of accuracy in order to verify how closely can the Gaussian model fit the measured data. Kaemarungsi (2006) focused on location finger printing that is based on RSSI of WLANs interfaces. The paper also takes into account RSSI's underlying mechanism to dissect it from indoor localization perspective. Extensive experiments are performed in order to analyze the RSSI distribution from five IEEE 802.11b/g WLAN interface. The goal of the paper is to model indoor localization system based on location fingerprinting. Seshadri, Zaruba, and Huber (2005) used a probabilistic method for global localization in an indoor environment using minimum infrastructure. In global localization, the device does not know its initial position and initiates one from scratch. RSSI from wireless NICs is used for localization. Bayesian filtering is applied on samples obtained through Monte-Carlo for obtaining location and orientation estimates. The accuracy of the proposed method is highlighted using simulations.

Gustafsson (2010); Gustafsson et al. (2002); Guvenc et al. (2003); Lee et al. (2010); Subhan et al. (2013) presented a detailed discussion on Bayesian filtering techniques that can assist in addressing the challenges affecting the performance of RSSI based indoor localization. While particle filters (PF) have proven to be better than Kalman Filters (KF) and Extended Kalman Filters (EKF) in an indoor environment, this research study aims to study whether using them in cascade (with different cascade arrangements), that is utilizing particle filters in combination with KF or EKF, can improve the performance of a PF-based localization system.

d)*Round Trip of Flight (RTOF):* In RTOF, the round-trip flight time of a specific signal sent from the transmitter to a specific entity is used for localization purposes (Liu et al., 2007). Figure 2.4 depicts both TOA and RTOF. RTOF, compared to TOA, does not require strict clock synchronization. Both TOA and RTOF use the same method for measuring the range. A radar can serve as measuring unit and the target respond back to the radar signal. The measuring unit measures the round trip time and its performance can be affected if the measuring unit is not aware of any possible delay at the target. If the transmission time is significantly more than the delay in long or medium range, then the delay can be ignored. However, delay must be taken into account for short ranges. Modulation reflection can be used for short range systems (Günther & Hoene, 2005). Similar positioning algorithms can be used for both RTOF and TOA.

2.5.1.1.2. Angulation

The fundamental type of angulation technique is known as Angle of Arrival (AoA). In AoA, the target is located by obtaining the intersection of different angle direction lines formed between the circular radius of the base station and the target (Liu et al., 2007). Figure 2.7 shows how angulation provides the location of the target. In order to track the position of the target in two dimensions, AoA requires two reference points and the angles formed between the reference points and the target. An array of antennas or directional antennas are suitable for AoA based approaches.

AoA can be used for three-dimensional (3D) localization even by utilizing three measuring units. Also, there is no need for timing synchronization in AoA based localization. However, hardware complexity and performance degradation with an increase in the distance between the target and reference points, is one of the significant problems with AoA. Furthermore, measuring the angles can be a tedious task in indoor environments due to various noise and multi-path fading phenomena.

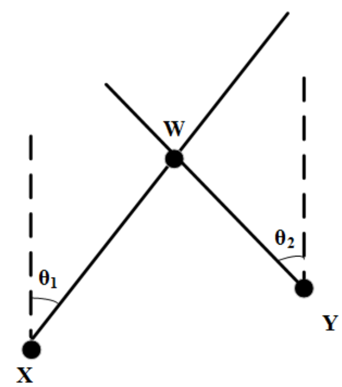

Figure 2.7. AOA based localization of an entity at point W.

Peng and Sichitiu (2006) utilized AOA information between adjacent nodes for localization. They propose a novel scheme for orientation and localization that takes the beacon information from multiple hops away into account. The measurements are assumed to be noisy. The simulation results show that even in the presence of noisy measurements and small number of beacons (reference nodes), the proposed system can accurately locate an entity. Niculescu and Nath (2003) used AOA ability of nodes in an ad-hoc network for localization. The communication among the adjacent nodes is leveraged for obtaining the location of the nodes (not necessarily reference nodes) with AOA capability. Two different algorithms based on Distance Vector (DV), DV-Bearing and DV-Radical are proposed that can provide absolute position and orientation of the nodes without the need for any added infrastructure.

2.5.1.2. Scene Analysis

In scene analysis, the fingerprints of the experiment space are collected. During the online stage, the fingerprints are matched against the obtained measurements to locate the user. The most widely used form of scene analysis is RSS based scene analysis. For localization, there is need for location fingerprinting in which the fingerprint of feature of a signal at a specific location is obtained. The two different stages of fingerprinting are:

- Offline: Site survey is conducted to obtain the RSS values at different locations within the site. The reference points or base stations are placed in the entire site. RSS values at different (x, y) coordinates from different base stations or access points are collected.

- Online: During the runtime or online stage, the obtained RSS values are compared with the offline values and are compared in order to obtain an accurate estimate of the location.

RSS based fingerprinting due to its reliance on RSS is prone to the problems that RSS faces in an indoor environment (i.e., the RSS values can fluctuate due to diffraction, reflection and scattering). There are five different fingerprinting based localization algorithms proposed in the literature: a) Probabilistic methods b) k Nearest Neighbors (kNN) c) Neural Networks(NN) d) Support Vector Machines (SVM) e) Smallest M-vertex polygon (SMP). Liu et al. (2007) provided a detailed discussion on these methods.

2.5.1.3. Proximity

Proximity based algorithms are intended for providing information related to symbolic relative position (Liu et al., 2007). These algorithms use a dense antenna grid and the position of every antenna is used. When a target is detected by any antenna, it is assumed to be co-located with that antenna. Proximity based methods are much easier to implement and a number of different mediums can benefit from such systems. RFID and IR based systems usually use proximity based algorithms for localization.

2.6 Bayesian Filtering

There are a number of scientific problems that require estimating a time varying system's state using a set of noisy measurements (Arulampalam, Maskell, Gordon, & Clapp, 2002). Bayes theory can be applied to such systems to estimate the state. To use Bayesian filtering, we require at least two models as described by Arulampalam et al. (2002).

1. System Model: This model describes the variation of the state with time. The system model relates the state y_i with the process noise and previous state.

2. Measurement Model: The measurement model relates the noisy measurements with the state.

In such an approach for estimating the dynamic state, the posterior probability density function (pdf) of the state is constructed using the set of all possible information including the measurements. The pdf contains all the statistical information available, so it can be considered to be the complete solution to the state estimation problem. In most of the problems, there is a need to estimate the state whenever a measurement is obtained. For such problems, we need a recursive filter that requires sequential processing of data. Recursive filters require the prediction and update stage. The state is usually affected by a number of unknown disturbances, so during the prediction stage, the pdf is translated, deformed and spread. In the update stage, the measurements are used for modifying the prediction pdf. This is where Bayes theorem comes into play as the knowledge or information about the state is updated using the obtained information.

According to the system model, state y_i at time i is a function of the state at $i-1$ and the process noise m_{i-1} as described by Djurić et al. (2003) and given in Equation 2.4.

$$y_i = f_i(y_{i-1}, m_{i-1}) \qquad \text{(Eqn. 2.4)}$$

$f_i : \Re^{n_y} \text{ x } \Re^{n_m} \rightarrow \Re^{n_y}$ is the non-linear function (can be linear as well) for the state y_{i-1} as described by Arulampalam et al. (2002). It maps the current state in the presence of process noise to the next state. $\{m_i, i \in \aleph\}$ indicates independent, identically distributed (i.i.d) process noise sequence. The state and process noise vector dimensions are represented by n_y and n_m respectively while the set of natural numbers is represented by $\aleph$. Similarly, according to the measurement model, the obtained measurement x_i is a function of the state y and measurement noise n at time i as described by Djurić et al. (2003) and given in Equation 2.5.

$$x_i = h_i(y_i, n_i) \qquad \text{(Eqn. 2.5)}$$

$h_i : \Re^{n_y} \text{ x } \Re^{n_n} \rightarrow \Re^{n_x}$ can be either a linear or non-linear function. Just like f_i, h_i also depends on laws of physics/motions and maps the current state and measurement noise to obtained measurements. $\{n_i, i \in \aleph\}$ represents the i.i.d measurement noise sequence. n_x and n_n represent the dimension of measurement and measurement noise vectors respectively.

This, in Bayesian terminology, can be interpreted as repeatedly calculating certain belief in the state y_i at particular time instant i taking various values in the presence of measurement data $x_{1:i}$. Therefore, it is necessary to calculate the probability density function (pdf) $p(y_i|x_{1:i})$. As described by Arulampalam et al. (2002), initial pdf $p(y_o|x_0)$ is assumed to be equivalent to state vector's $p(y_0)$. $p(y_0)$ is also known as the prior and it is assumed to be available. Based on the information provided, the pdf $p(y_i|x_{1:i})$ can be recursively obtained using a prediction and update stage.

In the prediction stage as described by Arulampalam et al. (2002), the Chapman-Kolmogorov equation (see Equation 2.6) is used for obtaining the state's prior pdf at time i provided that the pdf $p(y_{i-1}|x_{1:i-1})$ is available.

$$p(y_i|x_{1:i-1}) = \int p(y_i|y_{i-1})p(y_{k-1}|x_{1:i-1})dy_{i-1} \qquad \text{(Eqn. 2.6)}$$

At time i, the measurement x_i are obtained from the sensors which can then be used to update the prior using Bayes' rule as shown in equation 2.7 and described by Arulampalam et al. (2002). The normalizing constant in equation 2.7 is further explained in equation 2.8.

$$p(y_i|x_{1:i}) = \frac{p(x_i|y_i)p(y_i|x_{1:i-1})}{p(x_i|x_{i-1})} \qquad \text{(Eqn. 2.7)}$$

$$p(x_i|x_{i-1}) = \int p(x_i|y_i)p(y_i|x_{i-1})dy_i \qquad \text{(Eqn. 2.8)}$$

During the update stage, x_i is utilized for modifying the prior density that results in the required current state's posterior density. The recursive process described in Equations 2.6 and 2.7 results in an optimal Bayesian solutions. However, such recursive propagation of posterior density cannot be obtained analytically. Therefore, different algorithms such as KF, EKF, and PF can be used to obtain a solution. Below, the theory related to Kalman Filters, Extended Kalman Filters and Particle Filters is presented.

2.6.1 Kalman Filter

The underlying assumption of Kalman filters is that the posterior probability density is Gaussian at every step that can be characterized by a mean (μ) and variance (σ^2) (Arulampalam et al., 2002). For a Gaussian $p(y_{i-1}|x_{1:i-1})$, $p(y_i|x_{1:i})$ is also Gaussian under certain assumptions such as

- $h_i(y_i, n_i)$ is a linear function of the state y_i and measurement noise n_i

Table 2.5 *Kalman filter parameter notation*

Symbol	Meaning
y	State vector
x	Measurement/observation vector
F	State transition matrix
P	State vector estimate covariance or Error covariance
Q	Process noise covariance
R	Measurement noise covariance
H	Observation matrix
w	Process Noise
v	Measurement Noise

- $f_i(y_{i-1}, m_{i-1})$ is a known linear function of the y_{i-1} and process noise m_{i-1}
- Both the process noise m_{i-1} and measurement noise n_i follow a Gaussian distribution with known parameters.

Based on the above assumptions, Equation 2.4 and 2.5 can be written, as described in Arulampalam et al. (2002)

$$Y_i = F_i Y_{i-1} + m_i \tag{Eqn. 2.9}$$

$$X_i = H_i Y_i + n_i \tag{Eqn. 2.10}$$

Both F_i and H_i are known matrices that define the linear functions. The matrix F is known as the transition matrix while H_i is known as measurement matrix. The covariances of the process and measurement noise are Q_{i-1} and R_i. Table 2.5 contains the notations for different parameters used in the *prediction* and *update* steps of Kalman filters.

During the prediction phase, the state estimate $\bar{y}$ and state vector estimate $\bar{P}$ covariance is calculated as given in Equations 2.11 and 2.12 and described by Guvenc et al. (2003).

$$\bar{Y}_i = FY_{i-1} \tag{Eqn. 2.11}$$

$$\bar{P}_{i-1} = FP_{i-1}F^T + Q \qquad \text{(Eqn. 2.12)}$$

The Kalman gain K weights the observations based on which the estimate is updated. Kalman gain is one of the most important factors that dictates the performance of the filter. It is mathematically represented as given in equation 2.13. It is the measure of range to which the update values depend on the obtained measurements. The higher the value of measurement noise covariance R, the lower the reliance will be on measurements for updating the state vector and error covariance. This is because the increase in R causes a decrease in the Kalman gain. Similarly a higher value of Q means a higher value of Kalman gain, which would increase the weight of the observation when the estimate is being updated. In the update step, Kalman gain, the state and state vector estimate covariance are updated as discussed by Guvenc et al. (2003).

$$K_i = \bar{P}_{i-1}H^T(H\bar{P}_{i-1}H^T + R)^{-1} \qquad \text{(Eqn. 2.13)}$$

$$\bar{Y}_i = \bar{Y}_{i-1} + K_i(X_i - H\bar{Y}_{i-1}) \qquad \text{(Eqn. 2.14)}$$

$$P_i = \bar{P}_{i-1}(1 - KH) \qquad \text{(Eqn. 2.15)}$$

This whole process is iterative and the output from the update in iteration k is the input to the predict step in iteration $k+1$ as seen in Figure 2.8.

2.6.2 Extended Kalman Filter

Kalman filters relies on the assumption that the functions given in equations 2.9 and 2.10 are linear. However if the assumption does not hold (if functions are in fact non-linear), then a local linearization of the equations can approximate the non-linearity condition. This is the core essence of Extended Kalman Filters (EKF) as they locally linearize non-linear functions by taking the

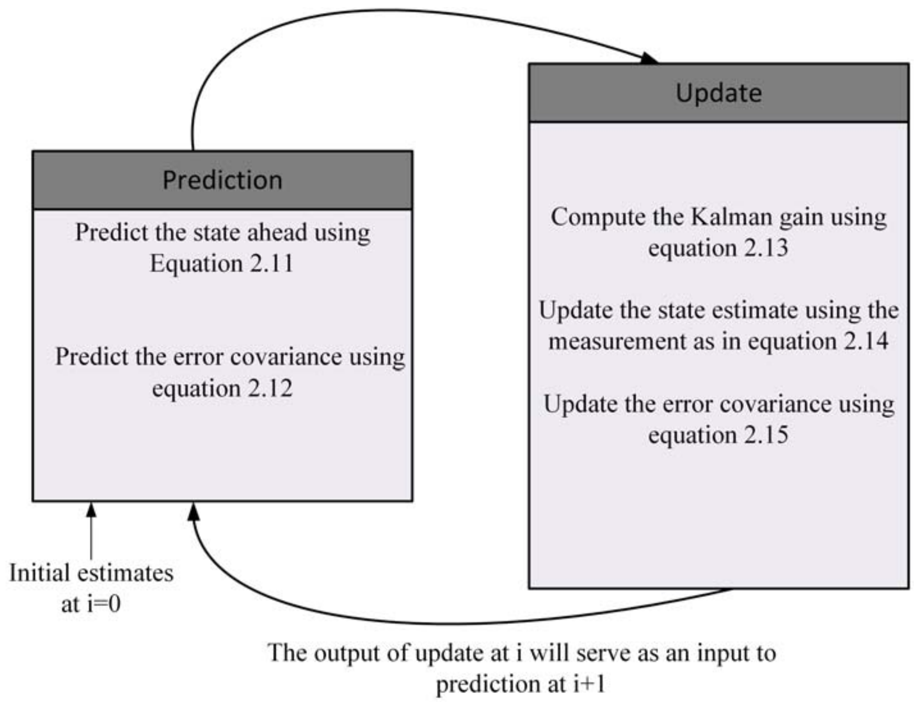

Figure 2.8. Prediction and update steps in Kalman Filter

Jacobian of the non-linear $f(.)$ and $h(.)$ functions listed in Equations 2.4 and 2.5 respectively. Let $\bar{F}$ and $\bar{H}$ are the locally linearized functions obtained through the Jacobian of $f(.)$ and $g(.)$ functions. Before discussing Jacobian, it is worth mentioning that EKF is based on this approximation that the probability $p(y_i|x_{1:i})$ can be approximated using Gaussian as given in Equations 2.16-2.18 by Arulampalam et al. (2002)

$$p(y_{i-1}|x_{1:i-1}) \approx \mathcal{N}(y_{i-1}; m_{i-1|i-1}, P_{i-1|i-1}) \qquad \text{(Eqn. 2.16)}$$

$$p(y_i|x_{1:i-1}) \approx \mathcal{N}(y_i; m_{i|i-1}, P_{i|i-1}) \qquad \text{(Eqn. 2.17)}$$

$$p(y_i|x_{1:i}) \approx \mathcal{N}(y_i; m_{i|i}, P_{i|i}) \qquad \text{(Eqn. 2.18)}$$

where $\mathcal{N}(y; m, P)$ is a Gaussian Probability Density and has arguments state y, mean m and covariance P. Similarly (taken from Arulampalam et al. (2002))

$$m_{i|i-1} = f_i(m_{i-1|i-1}) \qquad \text{(Eqn. 2.19)}$$

$$P_{i|i-1} = Q_{i-1} + \bar{F}_i P_{i-1|i} \bar{F}_i^T \qquad \text{(Eqn. 2.20)}$$

$$m_{i|i} = m_{i|i-1} + K_i(x_i - h_i(m_{i|i-1})) \qquad \text{(Eqn. 2.21)}$$

$$P_{i|i} = P_{i|i-1} - K_i \bar{H}_i P_{i:i-1} \qquad \text{(Eqn. 2.22)}$$

The Jacobian for both $f(.)$ and $g(.)$ functions is given below

$$\bar{F}_i = \left.\frac{df_i(y)}{dy}\right|_{y=m_{i-1|i-1}} \qquad \text{(Eqn. 2.23)}$$

$$\bar{H}_i = \left.\frac{dh_i(y)}{dy}\right|_{y=m_{i|i-1}} \qquad \text{(Eqn. 2.24)}$$

The predict and update steps for EKF are, as described by Guvenc et al. (2003).

- Predict:

$$\bar{Y}_{i-1} = FY_{i-1} \qquad \text{(Eqn. 2.25)}$$

$$\bar{P}_{i-1} = FP_{i-1}F^T + Q \qquad \text{(Eqn. 2.26)}$$

- Update:

$$K_i = \bar{P}_{i-1}H^T(H\bar{P}_{i-1}H^T + R)^{-1} \qquad \text{(Eqn. 2.27)}$$

$$\bar{Y}_i = \bar{Y}_{i-1} + K_i(X_i - H\bar{Y}_{i-1}) \qquad \text{(Eqn. 2.28)}$$

$$P_i = \bar{P}_{i-1}(1 - KH) \qquad \text{(Eqn. 2.29)}$$

The rest of the process is exactly same as Kalman Filters.

2.6.3 Particle Filter

Particle filters, based on the Sequential Importance Sampling (SIS) and Bayesian theory (Djurić et al., 2003), is widely used for indoor localization and tracking (Gustafsson, 2010; Gustafsson & Gunnarsson, 2003; Gustafsson et al., 2002). It is indeed a method of implementing a recursive Bayesian filter by Monte Carlo (MC) simulations. The basic idea behind particle filters is that the posterior probability distribution is represented using a set of weighted random samples that are used for computing the estimates (Arulampalam et al., 2002). The increase in the number of samples cause the filter to perform optimally. For understanding the algorithm in detail, let $\left\{y_{0:i}^k, w_i^k\right\}$ be the set of random measures that characterize the posterior pdf $p(y_{0:i}|x_{1:i})$. $\left\{y_{0:i}^k, k = 0, ...N_s\right\}$ is the set of support points whereas the weight are given by $\left\{w_i^k, k = 0, ...N_s\right\}$. $y_{0:i}\,\{y_j, j = 0, ...i\}$ is the set of the states up to i. The weight are normalized using $\sum_{min} w_i^k = 1$. After the normalization, the posterior density at i is approximated, as given by Arulampalam et al. (2002), using

$$p(y_{0:i}|x_{1:i}) \approx \sum_{k=1}^{N_s} w_i^k \delta(y_{0:i} - y_{0:i}^k) \qquad \text{(Eqn. 2.30)}$$

Equation 2.30 is the discrete weighted approximation of the true posterior probability distribution $p(y_{0:i}|x_{1:i})$. Importance sampling (Bergman, 1999) is used to choose the weights associated with each particle (Arulampalam et al., 2002). For importance sampling, assume that $p(y) \propto \pi(y)$ is the probability density from which drawing particles is tedious. However, $\pi(y)$ can be evaluated for it. Let $y^k \sim d(y)$ where $k \in [1, ..., M_s]$ be the samples generated from the proposal $d(.)$ known as importance density. Then the probability density $p(.)$ can be approximated as given by Arulampalam et al. (2002)

$$p(y) \approx \sum_{k=1}^{M_s} w^k \delta(y - y^k) \qquad \text{(Eqn. 2.31)}$$

where as the normalized weight of the k^{th} particle can be obtained using equation 2.32

$$w^k \propto \frac{\pi(y^k)}{d(y^k)} \qquad \text{(Eqn. 2.32)}$$

If the samples $y^k_{0:i}$ are taken from the importance density $d(y^k_{0:i}|x_{1:i})$, then the weights used in equation 2.30 are given by

$$w^k_i \propto \frac{p(y^k_{0:i}|x_{1:i})}{d(y^k_{0:i}|x_{1:i})} \qquad \text{(Eqn. 2.33)}$$

Due to the sequential nature of the process, at every single iteration, there could be samples that approximate the $p(y_{0:i-1}|x_{1:i-1})$, and the goal is to approximate $p(y_{0:i}|x_{1:i})$ with new samples. The importance density must be chosen for factorizing such that

$$d(y_{0:i}|x_{1:i}) = p(y_i|y_{0:i-1}, x_{i:i})d(y_{0:i-1}|x_{1:i-1}) \qquad \text{(Eqn. 2.34)}$$

Then the samples $y^k_{0:i} \sim d(y_{0:i}|x_{1:i})$ can be obtained by incorporating the existing samples $y^k_{0:i-1} \sim d(y_{0:i-1}|x_{1:i-1})$ into the recently obtained state $y^k_i \sim d(y_i|y_{0:i-1}, x_{1:i})$. The weights must be updated using the update equation that can be derived by first representing $p(y_{0:i}|x_{1:i})$ in terms of $p(x_i|y_i)$, $p(y_i|y_{i-1})$ and $p(y_{0:i-1}|x_{1:i-1})$ that can be mathematically given as

$$\begin{aligned} p(y_{0:i}|x_{1:i}) &= \frac{p(x_i|y_{0:i}|x_{0:i-1})p(y_{0:i}|x_{1:i-1})}{p(x_i|x_{1:i-1})} \\ &= \frac{p(x_i|y_{0:i}|x_{0:i-1})p(y_{0:i}|y_{0:i-1}|x_{1:i-1})}{p(x_i|x_{1:i-1})} \\ &\quad \times p(y_{0:i-1}|x_{1:i-1}) \end{aligned} \qquad \text{(Eqn. 2.35)}$$

$$\begin{aligned} &= \frac{p(x_i|y_i)p(y_i|y_{i-1})}{p(x_i|x_{1:i-1})}\, p(y_{0:i-1}|x_{1:i-1}) \\ &\propto p(x_i|y_i)p(y_i|y_{i-1})p(y_{0:i-1}|x_{1:i-1}) \end{aligned} \qquad \text{(Eqn. 2.36)}$$

The weight Equation in 2.33 can be updated by substituting Equation 2.34 and 2.36 into it, so Equation 2.33 will become

$$\begin{aligned} w_i^k &\propto \frac{p(x_i|y_i^k)p(y_i^k|y_{i-1}^k)p(y_{0:i-1}^k|x_{1:i-1})}{d(y_i^k|y_{1:i-1}^k,x_{1:i})d(y_{0:i-1}^k|x_{1:i-1})} \\ &= w_{i-1}^k\frac{p(x_i|y_i^k))p(y_i^k|y_{i-1}^k)}{d(y_i^k|y_{1:i-1}^k,x_{1:i})} \end{aligned} \qquad \text{(Eqn. 2.37)}$$

Also the importance density function would be only dependent on y_{i-1} x_i if $d(y_i|y_{1:i-1}, x_{1:i}) = d(y_i|y_{i-1}, x_i)$. This assists when only a filtered estimate of $p(y_i|x_{1:i})$ is needed at each time stamp. In such cases, only the state y_i^k would be stored. The weights are then modified into

$$w_i^k \propto w_{i-1}^k\frac{p(x_i|y_i^k))p(y_i^k|y_{i-1}^k)}{d(y_i^k|y_{i-1}^k,x_i)} \qquad \text{(Eqn. 2.38)}$$

while the filtered posterior probability density becomes

$$p(y_i|x_{1:i}) \approx \sum_{k=1}^{N_s} w_i^k\delta(y_i - y_i^k) \qquad \text{(Eqn. 2.39)}$$

The aforementioned algorithm (SIS) is a recursive algorithm in which the weights and support points are recursively propagated with the reception of every single measurement. The aforementioned three Bayesian algorithms can greatly help in localization purposes. Particle Filters are optimal for indoor localization since they assume the system to be non-linear and noise to be Non-Gaussian which is a realistic assumption for indoor environments.

2.7 Summary

In this chapter, we provided a detailed discussion and review of the literature relevant to IoT, smart buildings, iBeacons, proximity based services, localization and Bayesian Filtering. We presented an in-depth into the basics of IoT, its architecture and technologies. We also discussed smart buildings and highlighted how IoT and localization can be leveraged to improve the performance of smart buildings. We also presented a primer on iBeacons that discussed the basics of the protocol and can help to build upon for further analysis. We thoroughly investigated different techniques that are used for indoor localization purposes. We also showcased the basic theory behind such techniques and presented literature that utilizes these techniques. We also presented the basics of Bayesian filtering and explained Kalman Filters, Extended Kalman Filters and Particle Filters. In the next chapter, we provide the framework and methodology used in the research study.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

This chapter provides the framework and methodology used in the research study.

3.1 Hypotheses

H_{0_1}: The use of our proposed *Server-side Running Average* (SRA) and *Server-side Kalman Filter* (SKF) algorithms does not improve the proximity detection accuracy when compared with iBeacon's current proximity detection approach.

H_{α_1}: The use of our proposed *Server-side Running Average* (SRA) and *Server-side Kalman Filter* (SKF) algorithms improves the proximity detection accuracy by at least 10% when compared with iBeacon's current proximity detection approach.

H_{0_2}: The use of a Kalman Filter or an Extended Kalman Filter in cascade with Particle Filter does not significantly reduce the average localization error of an iBeacon based indoor localization system when compared with using only Particle Filter.

H_{α_2}: The use of a Kalman Filter or an Extended Kalman Filter in cascade with Particle Filter significantly reduces the average localization error of an iBeacon based indoor localization system when compared with using only Particle Filter.

3.2 Our Approach

In this research study, we used iBeacons for both proximity detection and indoor localization purposes. We present different approaches that improve the performance of an iBeacon-based proximity detection and indoor localization system. In earlier chapters, we discussed the basics of iBeacons, indoor localization, proximity based services and Bayesian filtering. In this chapter, we highlight the prototype iOS application that we developed for iBeacon based proximity and localization. Then we present our approach for improving the proximity detection accuracy of iBeacon-based proximity system. We also present an insight into how we utilize iBeacons for indoor localization.

3.2.1 Our iOS application

We developed a prototype iOS application that can communicate with iBeacons as well as a server. The application was developed in *Objective C*. The application has four tabs, three of which provide indoor localization, geofencing and proximity based services while the fourth one contains general information. Figure 3.1 shows our prototype application. The right side of Figure 3.1 shows the proximity tab while the left side highlights the indoor localization feature. The application is capable of receiving RSSI values from the iBeacons and then using them for indoor localization or proximity based services as well as communicating with the server.

3.2.2 iBeacon-based Proximity Detection

We highlight the limitations of the current iBeacon based proximity detection. Then we present server-based approaches that utilize moving average and Kalman filtering to improve the proximity detection accuracy of iBeacon-based proximity detection system. We compare our approach with Apple's current

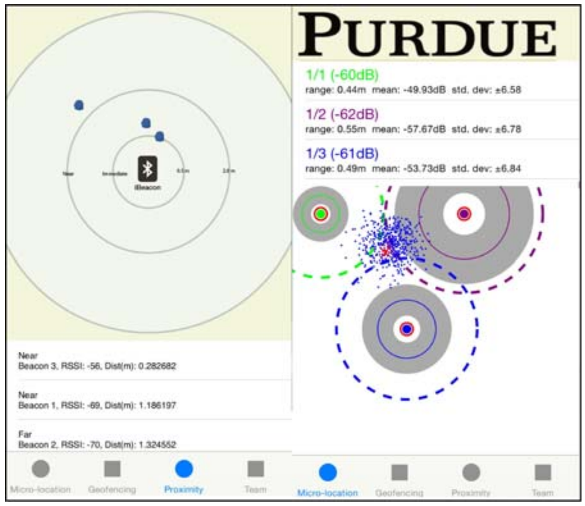

Figure 3.1. iPhone Application Prototype.

approach used for iBeacon based proximity detection and show that our approach resulted in significant improvement.

3.2.3 iBeacon-based Indoor Localization

For indoor localization, we used Particle filters primarily to locate the user using RSSI values. PF, as mentioned earlier, is widely preferred for indoor localization. To improve the performance further, we used KF and EKF for RSSI in cascade with PF to improve the localization accuracy. Below, we highlight our approaches for indoor localization.

3.2.3.1. Particle Filtering on the user device

The iOS application running on the user device was equipped with particle filter. iBeacons were placed in an indoor environment. The user was tracked continuously and the application showed him his approximated position. The particle filter was inputted with the unfiltered RSSI values. RSSI values were converted into distance between the user device and the beacons using path-loss model (Kumar, Reddy, & Varma, 2009) given in equation 3.1. Path-loss model is widely used in RSSI based localization. Through extensive experiment, the different parameters in equation 3.1 were caliberated to enhance the localization accuracy.

$$RSSI = -10nlog(d) + C \qquad \text{(Eqn. 3.1)}$$

where n represents the path-loss exponent, d represents the distance and C is the reference RSSI value at a distance of 1m (Kumar et al., 2009).

3.2.3.2. Particle Filtering on the server side

The user device running the iOS applications collected the RSSI values from different iBeacons and forwarded it to a local Apache Tomcat Server that ran particle filtering algorithm using the unfiltered RSSI values. The output of the algorithm was the estimated user location.

3.2.3.3. Cascaded Filters on the server side

Particle filtering was the main algorithm used for localization. However, for the cascaded filter part; the RSSI values were filtered through KF so that the fluctuation in the RSSI could be reduced. The reduced fluctuation in the RSSI values improved the localization accuracy and reduced the variation of the observed positions. Also, the Extended Kalman Filter (EKF) was used to enhance the measurements obtained from the Server-side Particle Filtering algorithm and improve the localization accuracy.

3.3 Data Sources

Based on the apparatus discussed above, the following data was collected.

- Actuators status (boolean) based on the user's movements in and out of the geofence.
- User's actual position in the (X, Y) grid.
- User's estimated position in the (X, Y) grid

3.4 Data Analysis

The data for both the tasks was analyzed differently.

3.4.1 Proximity based service

The actual proximity zone of the user was compared with the estimated proximity zone. If the actual and estimated proximity zones were same, there was no proximity detection error.

3.4.2 Indoor Localization

After completing all the experiments as described above for the indoor localization system, the data related to the user's original location and estimated location obtained from our proposed model was compared using equation 3.2. The average localization error was calculated as the difference between the actual location (X, Y) and the estimated $(X_{<est>}, Y_{<est>})$. n, equal to 11 in our experiments, was the total number of points used as the actual position while $(X_{<est>})$ and $(Y_{<est>})$ was calculated using the average of 10 measurements (estimates) for a particular point.

$$< Error >= \frac{\sum_{i=1}^{n} \sqrt{(X_i - X_{<est>})^2 + (Y_i - Y_{<est>})^2}}{n} \qquad \text{(Eqn. 3.2)}$$

The average localization error, standard deviation of error was calculated for experiments in different scenarios with various number of particles, and iBeacons. The parameters of KF and EKF were altered to improve the accuracy of the algorithm. A plot of the localization error versus the number of beacons, localization error versus number of particles, and arrangement of filters was also obtained. The average error was used to evaluate the validity of the hypotheses stated for this study.

3.5 Threats to validity

The possible threats to the validity are:

Environmental Noise: Different parameters such as the variables in the path-loss model were optimized for the environment in which the experiments were conducted for this research study. The variables vary for other environments that can affect the results. Similarly, the variation in environment noise such as presence of obstacles, or interference can affect the performance of the system.

iBeacon Type: There are number of beacons available whose performance vary. For this study, we used Gimbal series 10 beacons. The performance of the system can vary for other beacons.

Beacon Transmission Power: The beacons can be configured to transmit at a certain power. The beacons used in this study were optimized to transmit at a certain power (0 dBm) that suited the experimental setup and environment.

Beacon Message Transmission Frequency: The interval (frequency) between the transmission of simultaneous messages from the beacons can be altered. The higher the frequency, the better will be the performance. Changing the message transmission frequency can affect the results.

Number of Beacons: The number of beacons for indoor localization depend upon factors such as transmission frequency, power, area of the room (for experiments) etc. Varying the number of beacons can affect the performance and serve as a threat to the validity of the results in this study.

Topology of the Beacons: Beacons can be arranged in a number of topologies depending on the environment and noise. The topology used for this research study may not be ideal for some other environment.

Parameters of the filter: The filters used in this research study were optimized for this research study and environment. They necessarily might not hold true for other experimental setups and scenarios.

Internet/Network Connectivity: The mobile application needs to contact the server through the Internet/network. Using Internet or network that might have problems serve as threat to the validity of this research study.

3.6 Hypothesis Validation

- Two samples based large sample test was used.
- The average error was used as the test statistic.
  - $\bar{x}$ was the average error of the system only using Particle Filter.
  - $\bar{y}$ was the average error of the system using Cascaded Filters.
- Let μ_1 be the true average error of system using only particle filter, μ_2 be the true average error of system using cascaded filters and Δ_0 be $\mu_1 - \mu_2$ then
  - H_0: $\Delta_0 = .25 \times \bar{x}$
  - H_α: $\Delta_0 > .25 \times \bar{x}$ i.e. rejection region was $Z \geq Z_\alpha$ where $\alpha = 0.05$

$$Z = \frac{\bar{x} - \bar{y} - \Delta_0}{\sqrt{\frac{S_1^2}{m} + \frac{S_2^2}{n}}} \qquad \text{(Eqn. 3.3)}$$

where m and n were the number of samples in sample 1 (only particle filter) and sample 2 (cascaded filters).

- If $Z \geq 1.645(Z_\alpha = 1.645$ for $\alpha = 0.05)$ then the null hypothesis was rejected.

3.7 Summary

In this thesis, we provided the framework and methodology used in the research study. We also presented the hypotheses and our approach for proximity detection and indoor localization. We also listed the data sources, data analysis, the threats to validity and the method for hypothesis validation. In the next chapter, we provide the detailed results for the beacon based proximity services using the proposed cloud based architecture.

CHAPTER 4. IBEACON BASED PROXIMITY DETECTION

This chapter provides our server based approaches used for improving the proximity detection accuracy of an *iBeacon*-based proximity detection system. The current approach utilized by Apple's *CoreLocation Framework* is used as the benchmark. The chapter also describes our cloud-based architecture for end-to-end iBeacon-based proximity services.

4.1 Proposed Optimization for Proximity Detection

iBeacons are the industry standard for providing PBS. It is energy efficient with wide reception range and is readily available in most of the user devices. However, cost and proximity detection accuracy is the major challenge. The cost will reduce with the increase in the adoption of the iBeacons. Below we discuss the current approach utilized for iBeacon based PBS. We highlight its limitations that result in lower proximity detection accuracy. Then, we present our *Server-side Running Average (SRA)* and *Server-side Kalman Filter (SKF)* that improves the proximity detection accuracy by 29% and 32% respectively.

4.1.1 Current Approach

RSSI values from the iBeacons are received by the user device. Apple's *CoreLocation Framework* enables the application on the user device to report a moving/running average of the RSSI value every one second. Since the RSSI values are highly fluctuating, the running average is used to reduce the fluctuation and improve the performance. Based on the value of the running averaged RSSI, Apple then classifies the user's location in any of the aforementioned four zones. In case

Figure 4.1. Current approach used by Apple's CoreLocation Framework.

the user device detects that the user moved from one zone to another, the notification does not take place instantaneously. The device waits for some time in order to ascertain that the user indeed moved to a different zone and it is not because of a fluctuating RSSI value. Figure 4.1 shows the current approach used for iBeacon based proximity detection by iOS.

The problem with the current approach is that the RSSI values fluctuate drastically in the indoor environment despite using moving average. Furthermore, Apple uses a generic model in which the user's proximity to a certain beacon is estimated based on the RSSI values. The underlying assumption is that the channel characteristics of the indoor environment are uniform so an RSSI value higher than say 'x' indicates that the user is in 'immediate' zone while an RSSI value lower than 'x' indicates that the user is in 'near' region. However, this might not be the case in highly noisy or noise-free environments. For example, in highly noisy environment, a user in 'immediate' zone might receive an RSSI value lower than 'x' (due to noise) so the current approach would misclassify him in 'near' zone. This affects the proximity detection accuracy and proximity based services. To account for these limitations, below we describe our server based algorithms that leverage the superior processing power of servers for efficient and effective proximity detection.

4.1.2 Server-side Running Average (SRA)

Our first algorithm, Server-side Running Average intends to address problem of using a generic model for proximity classification using RSSI values, through the use of path-loss model obtained distance as given in Equation 4.1. In Equation 4.1, n is the path loss exponent, d is the distance while C is the RSSI value at a reference distance of 1 meter.

$$RSSI = -10nlog(d) + C \tag{Eqn. 4.1}$$

Using experiments, we obtained a path-loss model (discussed later) to compute distance using RSSI values. Our prototype iOS application received the RSSI values from the beacons and forwarded it to the server that obtained an estimated distance between the user and iBeacons using the path-loss model and RSSI values. The estimated distance was used to classify the user in any of the aforementioned four zones. In order to account for the drastic RSSI fluctuation, our algorithm SRA classified a user in any proximity zone only if three consecutive samples classified him/her in that zone. Algorithm 1 shows the SRA.

Algorithm 1 Server-side Running Average

```
procedure SERVER-SIDE RUNNING AVERAGE
    Obtain a path-loss model using site survey
    while User device receives RSSI from iBeacons do
        Report RSSI values from user device to server
        Obtain distance from RSSI values using path loss model.
        Classify proximity zone based on distance
        if The user is classified as being present in the same zone by three
           consecutive measurements obtained from beacons then
            Accept the proximity estimate
        else Reject the proximity estimate
```

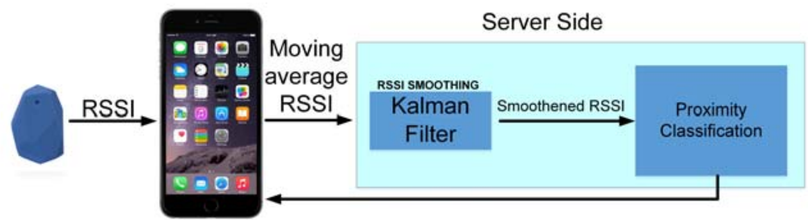

Figure 4.2. Server-side Kalman Filter for proximity classification

4.1.3 Server-side Kalman Filter (SKF)

Our second algorithm, Server-side Kalman Filter utilizes Kalman filtering for reducing the fluctuation in the RSSI. The user device obtains the RSSI values from the iBeacons and forwards it to a server. At the server, Kalman filter is used to reduce the RSSI variation. The smoothed RSSI values are then converted into distance using the obtained path-loss model. The computed distance is used to classify the user in any proximity zone. Just like SRA, SKF only classifies a user in any particular zone if three consecutive samples classify the user in that zone. Algorithm 2 contains the pseudocode for SKF. Figure 4.2 shows the proposed Kalman Filter based approach. Below we described the Kalman filter, used in our experiments, in detail.

Algorithm 2 Server-side Kalman Filter

```
procedure SERVER-SIDE KALMAN FILTER
    Obtain a path-loss model using site survey
    while User device receives RSSI from iBeacons do
        Report RSSI values from user device to server
        Pass RSSI through Kalman Filter
        Obtain distance from filtered RSSI values using path-loss model.
        Classify proximity zone based on distance
        if The user is classified as being present in the same zone by three
          consecutive measurements obtained from beacons then
            Accept the proximity estimate
        else Reject the proximity estimate
```

4.1.3.1. Parameters of the Kalman Filter

Prior to explaining the experimental results, we discuss the Kalman filter parameters for RSSI smoothing used in our iBeacon-based proximity system. We have already discussed the theory related to Kalman Filter in Chapter 2. Here we discuss KF in context of our experiment. Our state Y_i consists of the current RSSI value y_i and the rate of change of RSSI Δy_i given below.

$$Y_i = \begin{bmatrix} y_i \\ \Delta y_i \end{bmatrix}$$

The current value of RSSI y_i depends on the previous RSSI y_{i-1} plus the rate of change of RSSI Δy_{i-1} and the process noise m_i^y, so we can write Equation 2.9

$$\begin{bmatrix} y_i \\ \Delta y_i \end{bmatrix} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{i-1} \\ \Delta y_{i-1} \end{bmatrix} + \begin{bmatrix} m_i^y \\ m_i^{\Delta y} \end{bmatrix} \qquad \text{(Eqn. 4.2)}$$

Hence transition matrix F is given by

$$F = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}$$

The parameter δt depends on the environment. Through trial and error, we obtain the optimal results at δt equal to 0.2. Similarly the Equation 2.10 can be written as

$$\begin{bmatrix} x_i \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y_i \\ \Delta y_i \end{bmatrix} + \begin{bmatrix} n_i^y \end{bmatrix} \qquad \text{(Eqn. 4.3)}$$

So our observation matrix H is given by

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Parameters P, Q and R, obtained through trial and error, used in the experiments are given below.

$$P = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} R = \begin{bmatrix} 0.10 \end{bmatrix}$$

The Kalman Filter was then used for smoothing RSSI as discussed in Algorithm 2.

4.1.4 Experimental Setup, Results and Discussion

We first obtained the path-loss models for two distinct environments i.e. Environment 1 was 11m $\times$ 6m and environment 2 was 8m $\times$ 4m (environment 2) in dimension. The iBeacon was placed in a stationary position and we noted the RSSI values at different distances starting from zero meters up to seven meters. We used iPhone 6s plus running our prototype iOS application to collect RSSI measurements. We collected 22 samples at each distance and averaged that. Then we plotted the distance vs. RSSI values and used matlab's curve fitting function to obtain path-loss exponent n and the RSSI value at reference distance (1 meters in our experiments) C in Equation 4.1. Figure 4.3 and 4.4 show the fitted curve for distance vs RSSI in environment 1 and environment 2 respectively. Based on the fitted curve for environment 1, the n in Equation 4.1 was 0.9116 with 95% confidence bounds between (0.8272, 0.996) and C was -62.78 with 95% confidence bounds between (-64.07, -61.05). The R^2 value for the fitted curve was 0.9915. For environment 2, the path-loss exponent n equals to 1.246 with 95% confidence bounds between (1.139, 1.354) and C equals to -60.95 with 95% confidence bounds between (-62.24, -59.66) while the R^2 value for the fitted curve is 0.9926. Hence for environment 1, Equation 4.1 with above values values become Equation 4.4. Because the goal is to obtain distance using RSSI values, we rearrange Equation 4.4 into Equation 4.5. Similarly for environment 2, Equation 4.6 is rearranged into Equation 4.7 to obtain the distance from the beacons using RSSI values.

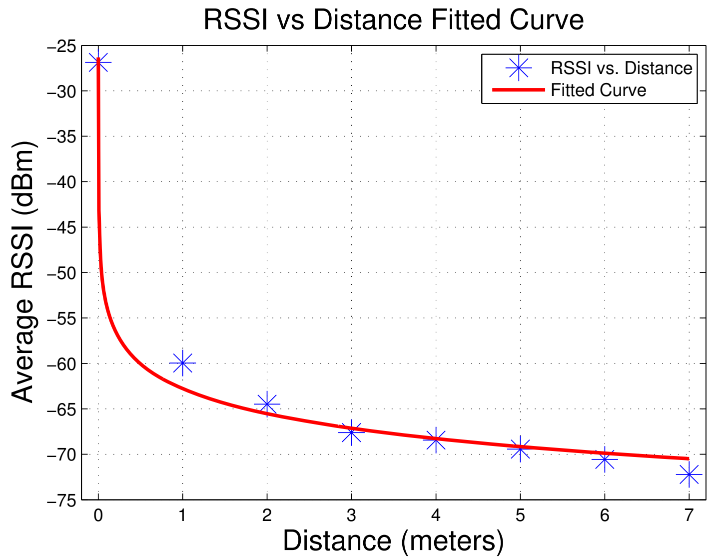

Figure 4.3. Curve fitting for RSSI values at distances from 0 to 7 meters in Environment 1

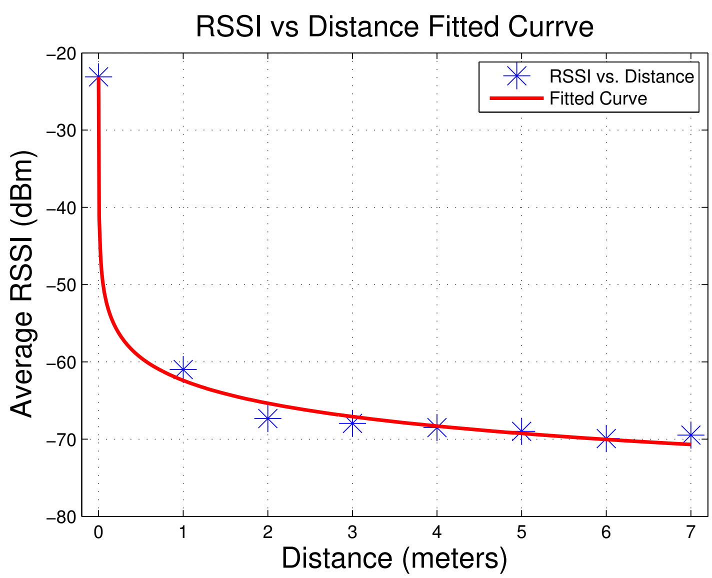

Figure 4.4. Curve fitting for RSSI values at distances from 0 to 7 meters in Environment 2

$$RSSI = -10 \times 0.9116 \times log_{10}d - 62.78 \qquad \text{(Eqn. 4.4)}$$

$$d = 10^{(\frac{62.78+RSSI}{-9.116})} \qquad \text{(Eqn. 4.5)}$$

Table 4.1 *An insight into the estimation error of the fitted curve for environment 1*

Average RSSI	Actual Distance (m)	Computed Distance (m)	Error
-26.8692	0.0001	0.0001	0
-59.9565	1	0.4901	0.5099
-64.4782	2	1.5357	0.4643
-67.6086	3	3.3861	0.3861
-68.4347	4	4.1717	0.1717
-69.4347	5	5.3705	0.3705
-70.5652	6	7.1452	1.1452
-72.2173	7	10.8457	3.8457

Table 4.2 *An insight into the estimation error of the fitted curve for environment 2*

Average RSSI	Actual Distance (m)	Computed Distance (m)	Error (m)
-23.1034	0.0001	0.0009	0.0008
-61	1	1.0093	0.0093
-67.3448	2	3.2601	1.2601
-67.9655	3	3.6563	0.6563
-68.5	4	4.0359	0.0359
-69	5	4.4266	0.5734
-69.9310	6	5.2576	0.7424
-69.4827	7	4.8396	2.1604

$$RSSI = -10 \times 1.246 \times log_{10} d - 60.95 \qquad \text{(Eqn. 4.6)}$$

$$d = 10^{\left(\frac{60.95+RSSI}{-12.46}\right)} \qquad \text{(Eqn. 4.7)}$$

Tables 4.1 and 4.2 highlight the accuracy of the obtained path-loss models. The tables list the average RSSI values, the actual distance at which they were obtained and the computed distances using the path loss model along with the error which is the absolute of difference between the actual distance and the computed distance. The models have an average computation error of 86.14 cm and 67.98 cm in environment 1 and environment 2 respectively. It is evident from Table 4.1, Table 4.2, Figure 4.3, and Figure 4.4 that the obtained path-loss models can accurately estimate distance using RSSI values.

Once we obtained the path-loss models, we evaluated our proposed algorithms and the current approach used by Apple. We put a Gimbal (Gimbal, n.d.) iBeacon in the aforementioned environments. We used a core-i5 Macbook-pro with 8 gigabytes of RAM, running Apache Tomcat 8.0 and Java 1.8 as our server for running SRA and SKF algorihtms. The equipment related information can be found in Table 4.3.

Table 4.3 *Summary of Device Parameters*

Device	Apple iPhone 6s plus
Wireless Interface	Bluetooth V4.2 / 2.4GHz
Operating System	iOS 9.2
Beacons	Gimbal Series 10
Gimbal range	50 meters
Transmission Frequency	100 ms
Major Value	Yes
Minor Value	Yes

We put the iBeacon in a fixed position and evaluated the proximity at a number of distances. Since the ‘unknown’ zone has no practical use, we tested our algorithms only in the ‘immediate’, ‘near’, and ‘far’ zones. In every zone, we evaluated the user’s proximity using the algorithms at two different distances. Table 4.4 shows the various distances in different zones where the algorithms were tested. We collected 20 RSSI samples at each physical location while every single RSSI sample was a moving average of 10 RSSI values (iOS reports RSSI after 1 second while our iBeacon transmitted after 100ms so 1s/100ms = 10). We took 40 samples in every zone (20 samples per distance $\times$ 2 distances in every zone) resulting in 120 samples (3 zones $\times$ 40 samples in every zone) for each environment.

We used a *confusion matrix*, a popular method used for performance evaluation of different classification models (Patro & Patra, 2015), to evaluate the performance of our algorithms. Using confusion matrix, we compared our estimated or classified proximity zone with actual proximity zones. The parameters of confusion matrix, as discussed by Fawcett in (Fawcett, 2006) are summarized in

Table 4.4 *Distances used in experiments*

Distance (m)	Proximity
0	Immediate
0.6	
1.8	Near
2.4	
4.3	Far
5.5	

Table 4.5 *Different parameters used in confusion matrix*

Parameter	Description
True Positive (TP)	When the user is in zone ‘x’ and is classified in zone ‘x’
True Negative (TN)	When the user is not in zone ‘x’ and is not classified in zone ‘x’
False Positive (FP)	When the user is not in zone ‘x’ but is classified in zone ‘x’
False Negative (FN)	When the user is in zone ‘x’ but is not classified in zone ‘x’
Precision/Positive Prediction Value (PPV)	The fractions of samples classified in zone ‘x’. Mathematically, precision = $\frac{TP_i}{TP_i+FP_i}$ where i is any zone.
Sensitivity/Recall	The fraction of samples correctly classified in zone ‘x’. Mathematically, sensitivity = $\frac{TP_i}{TP_i+FN_i}$. The higher the sensitivity, the better will be the algorithm.
Specificity	The fraction of samples correctly classified in any zone other than zone ‘x’. The higher the specificity, the better will be the algorithm. Mathematically, specificity = $\frac{TN_i}{TN_i+FP_i}$.
Fall out/False Positive Rate (FPR)	Mathematically, FPR = 1-specificity = $\frac{FP_i}{FP_i+TN_i}$. The lower the FPR value, the better will be the algorithm.
False Negative Rate (FNR)	Mathematically, FNR = $\frac{FN_i}{FN_i+TP_i}$. The lower the FNR value, the better will be the algorithm.
False Discovery Rate (FDR)	A good indicator for conceptualizing the rate of type I error. Mathematically, FDR = 1 - sensitivity = $\frac{FP_i}{FP_i+TP_i}$. The lower the FDR value, the better will be the algorithm.
Accuracy	The fraction of samples correctly classified. Mathematically, accuracy = $\frac{TP+TN}{TP+TN+FP+FN} \forall$ zones.

Table 4.5 in context of this thesis. Various statistical metrics that are obtained

through confusion matrix for the current approach as well as SRA and SKF are given in Table 4.6 and 4.8 for both environment 1 and environment 2 respectively. It is evident that both SRA and SKF outperform the current approach. The higher values of both true positives and the true negatives in the table, for the SRA and SKF, indicate that both these algorithms can accurately detect when a user is in any particular zone or not. Also, the lower value of false positives and false negatives mean that our algorithms do not misclassify a user in any other zone. The higher sensitivity values in the 'immediate' and 'near' zones for both SRA and SKF in contrast with the current approach indicates that our approach detects users in these zones with higher accuracy than the current approach. This is also verified by lower FDR, FNR, and FPR values. The higher FNR value for the current approach in the 'immediate' and 'near' zone means that the current approach misclassifies most of the samples when the user is in the 'immediate' or 'near' zones. This is because of the fact that current approach does not take the environment into consideration when classifying user proximity and classifies the user in the 'far' zone due to the environmental noise (noise reduces the RSSI values so the user is misclassified in the 'far' zone despite being in 'immediate' or 'near' zone). This is also the reason that the current approach seems to have higher accuracy in the 'far' zone. However, the higher accuracy in the far region for the current approach is due to the inherent flaw in the current approach, also highlighted by the high FDR value in the 'far' zone for the current approach.

Table 4.7 and 4.9 show the proximity detection error of the current, SRF and SKF algorithm at different distances and in different zones in both environment 1 and environment 2 respectively. The current approach has 95% and 75% proximity detection error at 0.6m in environment 1 and environment 2 respectively, that indicates that out of 20, only 1 sample in environment 1 and 5 samples in environment 2 were accurately classified by the current approach. In contrast, our SRA and SKF algorithms accurately classify all the 20 samples. It can be seen that our algorithms SRA and SKF have zero proximity detection error in the 'immediate'

Table 4.6 *Statistical metrics for the current approach, SRA and SKF in Environment 1*

Metric	Immediate			Near			Far		
	Current	SRA	SKF	Current	SRA	SKF	Current	SRA	SKF
True Positive	21	40	40	18	38	39	40	33	38
True Negative	80	78	79	61	73	78	58	80	80
False Positive	0	2	1	19	7	2	22	0	0
False Negative	19	0	0	22	2	1	0	7	2
Precision	1	0.952	0.975	0.486	0.844	0.951	0.645	1	1
Sensitivity	0.525	1	1	0.45	0.95	0.975	1	0.825	0.95
Specificity	1	0.975	0.987	0.762	0.912	0.975	0.725	1	1
Fall out	0	0.025	0.012	0.237	0.087	0.025	0.275	0	0
FDR	0	0.047	0.024	0.513	0.155	0.048	0.354	0	0
False Negative Rate	0.475	0	0	0.55	0.05	0.025	0	0.175	0.05

Table 4.7 *Comparison of proximity detection error of SRA and SKF in comparison with current approach in Environment 1*

Actual	Distance (meters)	Error at Different Distances (%)			Error in Different Zones (%)		
		Current	SRA	SKF	Current	SRA	SKF
Immediate	0	0	0	0			
	0.6	95	0	0	47.5	0	0
Near	1.8	10	10	5			
	2.4	100	0	0	55	5	2.5
	4.3	0	5	0			
Far	5.5	0	30	10	0	17.5	5

Table 4.8 *Statistical metrics for the current approach, SRA and SKF in environment 2*

Metrics	Immediate			Near			Far		
	Current	SRA	SKF	Current	SRA	SKF	Current	SRA	SKF
True Positive	25	40	40	16	37	38	40	39	40
True Negative	80	80	80	65	79	80	56	77	78
False Positive	0	0	0	15	1	0	24	3	2
False Negative	15	0	0	24	3	2	0	1	0
Precision	1	1	1	0.516	0.973	1	0.625	0.928	0.952
Sensitivity	0.625	1	1	0.4	0.925	0.95	1	0.975	1
Specificity	1	1	1	0.812	0.987	1	0.7	0.962	0.975
Fall out	0	0	0	0.187	0.012	0	0.3	0.037	0.025
FDR	0	0	0	0.483	0.026	0	0.375	0.071	0.047
False Negative Rate	0.375	0	0	0.6	0.075	0.05	0	0.025	0

Table 4.9 *Comparison of proximity detection error of SRA and SKF in comparison with current approach in environment 2*

Actual	Distance (meters)	Error at Different Distances (%)			Error in Different Zones (%)		
		Current	SRA	SKF	Current	SRA	SKF
Immediate	0	0	0	0			
	0.6	75	0	0	37.5	0	0
Near	1.8	20	0	0			
	2.4	100	15	10	60	7.5	5.0
	4.3	0	5	0	0		
Far	5.5	0	0	0		2.5	0

zone. This means that all the 40 samples are accurately classified in the immediate. Similarly in the 'near' zone, both SRA and SKF outperform the current approach. As discussed earlier and highlighted by high FNR values of the current approach in the 'immediate', and 'near' zone as well as high FDR value in the 'far' zone, the high accuracy of the current approach in the far zone is not because of the superior performance of the approach but rather the tendency of the approach to misclassify everything in the far zone. SKF is the optimal of all the algorithms and outperforms SRA as well due to the use of Kalman Filters. SRA is simpler but less accurate when compared with SKF. Figure 4.5 and 4.6 show the error of all the three algorithms in the three different zones of environment 1 and environment 2 respectively. As mentioned earlier, both SRA and SKF have a zero proximity detection error in the 'immediate' region. Figure 4.7 highlights the overall accuracy of the algorithms. The current approach has an overall accuracy of 65.83% in environment 1 and 67.5% in environment 2. The SRA achieved a proximity detection accuracy of 92.5% in environment 1 and 96.6% in environment 2. SKF achieved proximity detection accuracy of 97.5% in environment 1 and 98.3% in environment 2. Hence SRA outperforms current approach by 26.7% in environment 1 and 29.1% in environment 2 while SKF outperforms current approach by 31.6% in environment 1 and 30.8% in environment 2. The lower noise level in environment 2, is highlighted by lower C in Equation 4.6 in comparison with the C value for environment 1. This is why the performance of all three algorithms is better in environment 2, compared to environment 1.

In the following section, we discuss our cloud and iBeacon-based architecture for proximity based services.

4.2 iBeacon Deployments for Proximity-Based Services

iBeacons are predominantly used for proximity based context aware solutions. Using the geofencing capability of the beacons, they can be used to

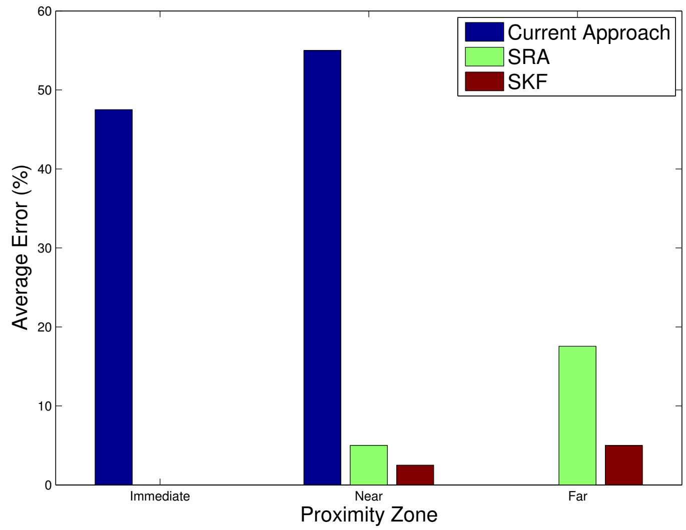

Figure 4.5. Error in different proximity zones for the models in Environment 1

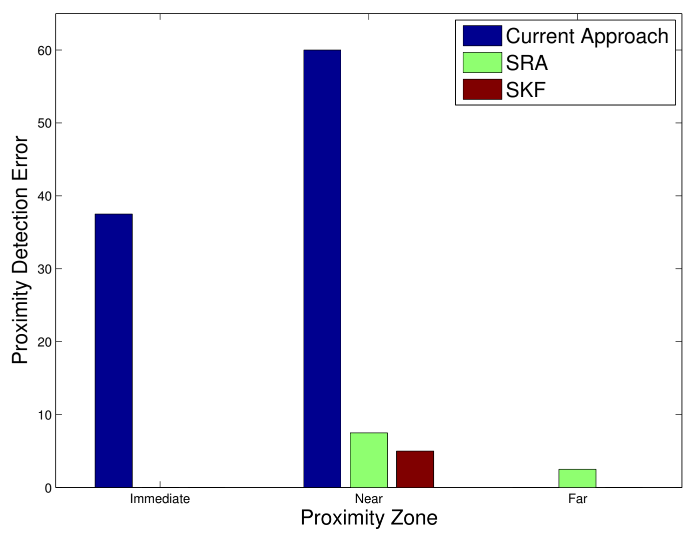

Figure 4.6. Error in different proximity zones for the models in Environment 2

automate various systems and make the environment interactive. We present our cloud-based architecture that leverages iBeacons for geofencing capabilities. The architecture consists of:

- Beacons

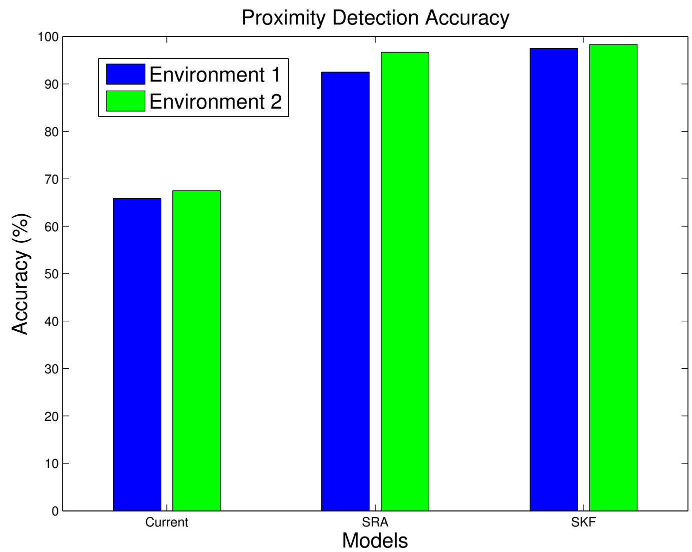

Figure 4.7. Average Proximity Detection Accuracy of three algorithms

- User Device
- Cloud Computing
- Actuator/End Device

We already discussed iBeacons in detail in chapter 2. The interested readers can read Zafari and Papapanagiotou (2015); Zafari et al. (2015, 2016) for further details on iBeacons. Below, we discuss other components of our architecture.

4.2.1 User Device:

As discussed earlier, the user device must be BLE enabled and should run either the iOS 7.0 (or later) or the Android 4.3 (or later) version operating systems. Applications that are beacon capable must be developed to receive the messages transmitted by the beacon. The UUID values of the beacons are to be hard coded into the application as per the requirements of Apple Application Store. A single user device can listen to more than four billion beacons at a time. However, the number of regions or geofences that the application running on the user device can monitor is limited to 20. For this research study, we developed a prototype iOS application in Objective C that is capable of providing indoor localization, geofencing and proximity based services.

4.2.2 Cloud Computing:

Our architecture uses cloud services for providing iBeacon authentication as well as geofencing services. When the user device obtains the UUID, major and minor values of the beacon for the first time, it contacts an Apache Tomcat Server residing on Amazon's Web Services (AWS) and forwards the information using JavaScript Object Notation (JSON) document, sent over the wireless network (WiFi or 4G/LTE) to AWS. For security purposes, the study uses a MySQL database on the server that contains the information about all the known beacons such as the UUID, major value, minor value, the name and the purpose of the beacon. After the information is received from the user device, it is compared with the stored information in the database. If there is a match, then the beacons are authenticated and server notifies the device about the use of the beacon and its purpose. Whenever the user device enters or exits the geofence of any beacon, it notifies the server that can trigger content based on the user's proximity to the beacon or device.

4.2.3 Actuator/End Device:

There are a wide range of devices that can serve as actuators or end devices based on the application and requirements. It can be a ubiquitous household device such as microwave oven, television, computer monitor etc. or an application specific device such as the interactive media present in museum or libraries that tend to engage the users and improve the service level. The actuator is controlled by the user's location and is triggered by the server based on the user's entrance or exit from a geofence. This study used Belkin's *WemoSwitch* as an end device that would turn *on* and *off* based on the user's location. We also utilize the interactive environment of *James Hunt Jr. Library* at North Carolina State University (NCSU) as an end device to verify the effectiveness of the used beacon based proximity service.

The proposed architecture for beacon based proximity can use any server, cloud computing service or database and does not necessarily have to be Apache Tomcat, AWS or MySQL. The beacons were placed close to the end device/actuators for creating a geofence. The beacon related information was stored in the database and the server was deployed on AWS and an *Elastic Bean* instance was run. The user carrying a device with beacon application installed and Bluetooth enabled move around in the ambiance. The events were triggered based on the location and the actuators acted in a pre-defined manner. Below, we present the two different environments in which our architecture was used.

4.2.3.1. Deployment 1

Deployment 1 involved the utilization of a user's location in a 11m $\times$ 6m area to automate an electronic device through a Wemo Switch. The Belkin Wemo Switch is a Wi-Fi based switch that is primarily used to control the appliances over the internet or using smart phones. The Wemo Switch API allows an iOS, Android or any other device to interact with the switch and control its operation. Figure 4.8 shows a schematic of how the system accomplished its task. Once the beacons were verified from the server and the database, the proximity of a device to a beacon was used as an indicator whether to trigger the server to either turn 'on' or 'off' a specific electric device. The Wemo switch has a specific Internet Protocol (IP) address that the server can communicate with. Once the user was within the 'immediate' or 'near' vicinity of the appliance, the mobile phone informed the server which sent the 'on' command to the wemo switch. The switch turned on which in turn activated the appliance as well. As soon as the user moved into either 'far' or

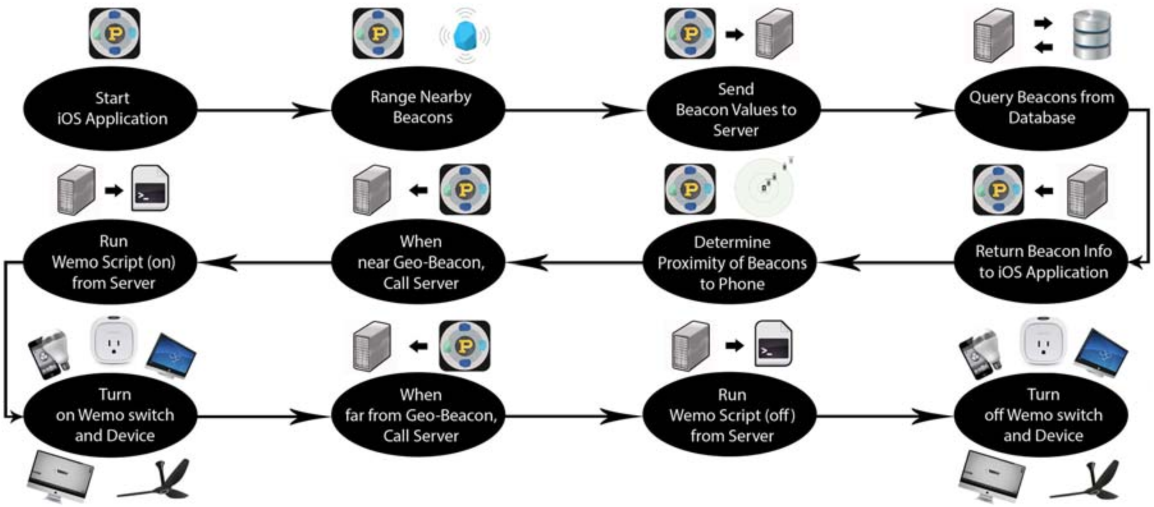

Figure 4.8. End-to-End IoT Solution Implementation

'unknown' region, the user device asked the server to turn off the appliance. Although we used a Wemo switch, numerous other devices and APIs can be communicated using our architecture for creating a geofence and then handling the appliances accordingly. A video demonstration of our implementation can be found at our Youtube Channel (2015a).

4.2.3.2. Deployment 2

In the second deployment, we installed 8 beacons in the James B. Hunt Jr Library (Libraries, 2015) in North Carolina State University (NCSU). The library built at a cost of about $115.2 million is known for its interactive and high-tech environment. There are a number of large screens and devices used to enhance the user's experience and provide efficient, effective and context-aware location based services to the library tenants. The library has an in-house RESTful web-service that allows to listen to HTML 'POST' and 'DELETE' messages and de-serialize JSON or XML based content using basic authentication. To interact with their web-service to trigger content on the display devices, we used the proximity to the beacons to either 'POST' or 'DELETE' from their web-service. After the iOS application was launched and the beacons were ranged, the server and the database were consulted for the beacons. The server responded back with the information about the beacons and the display device with which it is related. When a user was in either the 'immediate' or 'near' vicinity of the beacon affiliated with a particular device, the iOS application sent a specific message containing the user's location and affiliated display to the Tomcat server deployed on cloud. The Tomcat server then authenticated with the RESTful web-service of the library and after successful authentication, posted location aware content to the web-service that showed up on the display devices. The content appeared as long as the user stayed within the 'immediate' or 'near' vicinity or when the time stamp expired on the server. Currently the RESTful API uses a time stamp of two minutes. After the user moved into either a 'far' or 'unknown' region, the user device notified the Tomcat server, which sent a delete message to the web-service to remove location based context aware content from the web-service and the screen. Using a brute force approach i.e. a large number of beacons in the area, we improved the performance of the system to trigger content on a number of different display devices that are currently located in the library. The larger space, greater number of devices and larger number of beacons presented a challenge that the proposed architecture handled efficiently and effectively. The promptness, and accuracy of all these different tasks makes the proposed architecture a viable approach for deployment in even more complex scenarios and highlights the potential for industrial implementations and deployments. A visual depiction of the prototype can also be found on our YouTube Channel (2015b).

4.3 Summary

In this chapter, we discussed the current approach used by Apple for iBeacon based proximity detection. We also described the inherent limitations in the current approach, and discussed our algorithms SRA and SKF that improved the proximity detection accuracy by 27% and 32% respectively when compared with the current approach. We highlighted our cloud and iBeacons-based architecture for proximity based services and highlighted two different deployments using our architecture. In this next chapter, we present our iBeacon-based indoor localization system.

CHAPTER 5. IBEACON BASED INDOOR LOCALIZATION

This chapter provides an iBeacon-based indoor localization system. The chapter discusses the experimental setup, the obtained results and also validates the hypothesis of this thesis. While we use the server side filtering algorithms for hypothesis testing, we also include the results of mobile phone-based particle filtering for indoor localization that has been published in IEEE Globecom 2015 (Zafari & Papapanagiotou, 2015).

5.1 Particle Filtering on the User Device

We placed a number of beacons in an 11m × 6m space that, due to the presence of obstacles, replicates a real world scenario for iBeacon based indoor localization. We also tested our proposed system in a 1m × 1m space. We utilized our developed iOS application that obtained the RSSI values from the iBeacons and used particle filtering algorithm for indoor localization. During the experiments, we altered the number of iBeacons and particles to locate the user (Zafari & Papapanagiotou, 2015). Table 5.1 lists the device related information. Figure 5.1 shows the average localization error in 1m × 1m are for varying number of iBeacons. We increased the number of iBeacons until the addition of further iBeacons did not improve the results or in worst case, affected the localization accuracy due to the interference among iBeacons. In 1m × 1m environment, we started with 3 iBeacons and added iBeacons to improve the overall localization performance. The addition of sixth iBeacon adversely affected the localization accuracy due to the saturation of the space with iBeacons resulting in interference among the iBeacons. Table 5.2 and 5.3 highlight the average localization error and standard deviation of the localization error for varying number of particles and iBeacons respectively. The lowest localization error is observed for 4 iBeacons and 1000 particles. Figure 5.2 shows the average localization error Vs. number of iBeacons in 11 m × 6m area. As in the case of 1m × 1m space, we started with three iBeacons in the space and kept increasing the number of iBeacons until it enhanced the localization accuracy. The addition of the 8^{th} iBeacon into the space, adversely affected the average localization error which is why we did not add any more iBeacons. Table 5.4 and 5.5 show the average localization error and standard deviation of localization error respectively for a number of particles and iBeacons in 11 m × 6m area. The minimum localization error of 0.97 meters was obtained with 1000 particles and 5 iBeacons.

Table 5.1 *Experimental Deployment*

Device	Apple iPhone 4s
Wireless Interface	Bluetooth V4.0 / 2.4GHz
Operating System	iOS 8.1
Beacons	Gimbal Series 10
Gimbal range	50 meters
Transmission Frequency	100 ms
Major Value	Yes
Minor Value	Yes

Table 5.2 *Localization error (meters) for a number of particles and iBeacons in 1m × 1m environment with particle filter on the user device*

Particles	Beacons			
	3	4	5	6
400	0.308	0.290	0.303	0.301
600	0.356	0.308	0.312	0.302
800	0.396	0.301	0.302	0.310
1000	0.384	0.276	0.298	0.316
1200	0.400	0.299	0.293	0.318
1400	0.403	0.289	0.307	0.315
1600	0.385	0.314	0.306	0.316
1800	0.407	0.298	0.299	0.291
2000	0.411	0.312	0.300	0.304

The computational overhead of calculating the particles on the end user's device may result in draining the battery faster. Also, the limited processing power

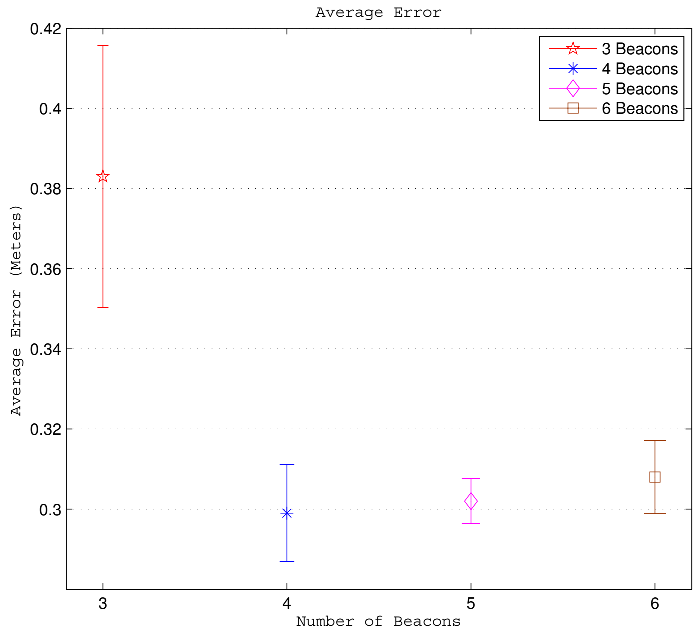

Figure 5.1. Average error vs number of particles for different number of iBeacons for 1m × 1m.

Table 5.3 *Standard Deviation of Localization error for a number of particles and iBeacons in 1m × 1m environment with particle filter on the user device*

Particles	Beacons			
	3	4	5	6
400	0.065	0.158	0.162	0.156
600	0.232	0.172	0.155	0.171
800	0.262	0.170	0.151	0.164
1000	0.273	0.178	0.164	0.152
1200	0.254	0.167	0.159	0.167
1400	0.249	0.167	0.163	0.156
1600	0.238	0.164	0.146	0.149
1800	0.245	0.172	0.175	0.181
2000	0.253	0.161	0.143	0.181

of the device can affect the performance of the algorithm. Therefore, we utilize a local server to run the particle filtering algorithm for computing the user position

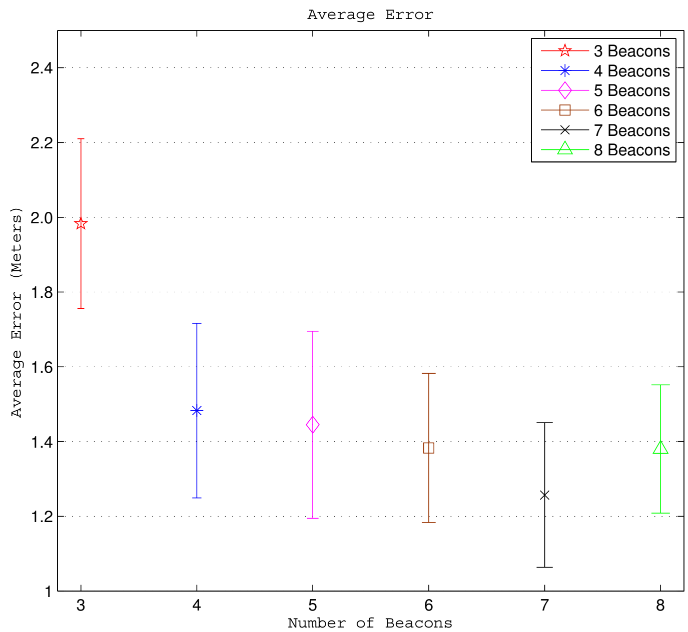

Figure 5.2. Average error vs number of particles for different number of iBeacons for 11m × 6m environment.

Table 5.4 *Average error vs number of particles for different number of iBeacons for 11 m × 6m environment with particle filter on the user device*

Particles	Beacons					
	3	4	5	6	7	8
400	2.195	1.486	1.720	1.590	1.385	1.492
600	2.167	1.074	1.159	1.422	1.200	1.595
800	2.152	1.729	1.721	1.598	1.345	1.623
1000	1.736	1.802	0.975	1.284	1.220	1.432
1200	1.843	1.678	1.531	1.126	1.008	1.275
1400	2.262	1.507	1.575	1.639	1.442	1.180
1600	2.049	1.251	1.455	1.149	1.339	1.245
1800	1.774	1.368	1.540	1.208	1.362	1.168
2000	1.668	1.451	1.328	1.430	1.017	1.411

that is discussed in detail in the next section. The device related information for the following sections can be found in Table 4.3.

Table 5.5 *Standard Deviation of Error vs number of particles for different number of iBeacons for 11 m × 6m environment with particle filter on the user device*

Particles	Beacons					
	3	4	5	6	7	8
400	0.333	0.913	1.005	1.288	0.641	1.399
600	0.996	0.600	0.868	1.164	0.850	1.176
800	0.394	1.041	0.825	1.045	0.672	1.378
1000	0.441	1.082	0.885	1.189	1.144	0.815
1200	0.473	1.034	0.692	0.890	1.020	1.009
1400	0.261	0.733	0.890	1.660	1.042	0.763
1600	0.295	0.640	0.724	0.943	0.637	0.681
1800	0.669	0.698	0.872	0.948	1.000	0.815
2000	0.561	0.458	0.919	1.296	0.977	0.962

5.2 Particle Filtering on the Server Side

The RSSI values from the iBeacons were collected by the user device that reported it to a server. The server utilized particle filtering algorithm to computer user's position. As in the case of running particle filtering on the user device, we varied the number of iBeacons, and the number of particles. We conducted the experiments in a 7 m × 6m space that contained a number of obstacles. We used the particle filtering algorithm to obtain an estimate of the user location and compare it with the original user location to calculate the localization error in accordance with Equation 3.2. We started with 3 iBeacons and kept increasing the number of iBeacons until the addition of further iBeacons did not bring any significant improvement in the localization accuracy or in worst case, affected it adversely. During the course of experiments, as expected, the portion of the experimental space that was behind obstacles and had obstructions had higher localization error in contrast with comparatively less occupied space. Figure 5.3 shows the average localization in meters for different number of iBeacons. It can be seen the average localization error is high for 3 iBeacons. However the addition of further iBeacons improves the performance. An average accuracy of 0.969 meters was obtained with 7 iBeacons, but the addition of the 8^{th} iBeacon deteriorated the results. This is due to self interference among the iBeacons. Table 5.6 shows the average localization error for various number of iBeacons and particles with particle filter algorithm running on the server. The lowest average localization error of 0.859 meters was obtained with 7 iBeacons and 1200 particles. Table 5.7 shows the standard deviation of average localization error for various number of iBeacons and the particles with particle filter algorithm running on the server. In the next section we showcase the results for a cascaded filter arrangement of Kalman Filter followed by Particle filter that we termed as KFPF.

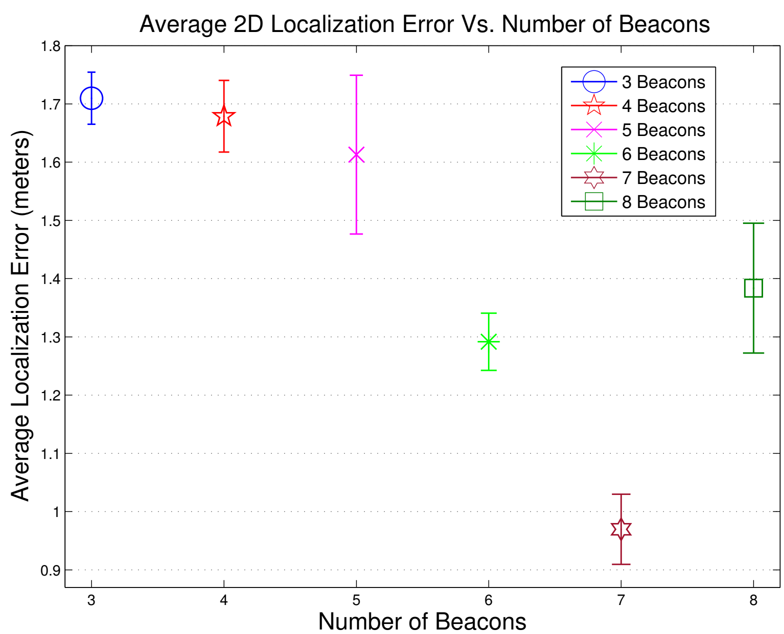

Figure 5.3. Average error vs number of Beacons in 7m $\times$ 6m environment with particle filter on the server side.

5.3 Kalman Filter-Particle Filter Cascade on the Server Side

As described in chapter 3, one of the fundamental challenges of RSSI-based localization is the problem of RSSI fluctuation due to the presence of obstacles in the environment. The RSSI values greatly vary despite no difference in the position

Table 5.6 *Average error vs number of particles for different number of iBeacons for 7 m × 6m environment with particle filter on the server side.*

Particles	Beacons					
	3	4	5	6	7	8
400	1.776	1.566	1.672	1.405	0.916	1.247
600	1.768	1.788	1.693	1.307	1.063	1.395
800	1.733	1.639	1.545	1.314	0.954	1.545
1000	1.696	1.658	1.551	1.243	0.955	1.438
1200	1.724	1.688	1.548	1.267	0.859	1.506
1400	1.665	1.704	1.502	1.252	0.995	1.346
1600	1.701	1.703	1.547	1.259	0.959	1.448
1800	1.643	1.647	1.932	1.300	1.017	1.304
2000	1.681	1.715	1.526	1.278	1.010	1.225

Table 5.7 *Standard deviation of average error vs number of particles for different number of iBeacons for 7 m × 6m environment with particle filter on the server side*

Particles	Beacons					
	3	4	5	6	7	8
400	0.743	0.586	1.027	0.628	0.431	0.483
600	0.641	0.881	1.084	0.589	0.533	0.641
800	0.565	0.669	0.821	0.456	0.386	0.762
1000	0.668	0.721	0.773	0.486	0.428	0.701
1200	0.719	0.748	0.742	0.493	0.382	0.664
1400	0.724	0.561	0.624	0.470	0.391	0.403
1600	0.570	0.879	0.984	0.530	0.454	0.654
1800	0.498	0.605	0.674	0.639	0.392	0.647
2000	0.557	0.655	0.925	0.512	0.339	0.585

of the user as the radio frequency waves can bounce of the walls and various obstruction resulting in interference or noise. To improve the performance of the localization system, we first reduce the RSSI fluctuation using Kalman filter. The filtered RSSI values are then input into PF for locating the user as shown in Figure 5.4. Since the Kalman filter used in Chapter 4 provided improved results in terms of RSSI smoothing and improve the proximity detection, we use the Kalman Filter with same parameters here and model the RSSI as done in Chapter 4. Once the RSSI values were filtered by Kalman Filter, the particle filter on the server side

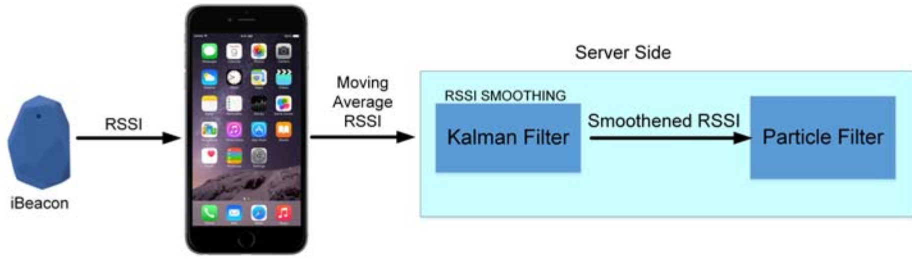

Figure 5.4. Kalman filter enhanced RSSI values for server side particle filter-based localization.

estimated the user's location. Figure 5.5 shows the average localization error with KFPF algorithm for a number of beacons. As done in the previous case, we started with 3 beacons and added beacons till the addition of further beacons did not improve result or in worst case affected it adversely. For the KFPF algorithm, we obtained the best results with 7 iBeacons. Table 5.8 shows the average localization error of the KFPF algorithm for different number of beacons and particles in 7 m × 6m area. The lowest localization error is obtained with 7 beacons and 2000 particles. In comparison with Table 5.6, the average localization error with KFPF algorithm is less. Table 5.9 shows the standard deviation of the localization error for KFPF algorithm for different number of beacons and particles in our experimental area.

Table 5.8 *Average error vs number of particles for different number of iBeacons in 7 m × 6m environment with KFPF algorithm.*

Particles	Beacons					
	3	4	5	6	7	8
400	1.307	1.091	1.147	0.836	0.916	0.812
600	1.484	1.219	1.013	0.902	0.985	0.849
800	1.442	1.238	1.055	0.778	0.736	0.782
1000	1.472	1.276	0.939	0.837	0.748	0.796
1200	1.412	1.450	1.090	0.806	0.709	0.804
1400	1.390	1.371	1.064	0.845	0.724	0.781
1600	1.449	1.376	1.023	0.861	0.742	0.849
1800	1.500	1.295	1.149	0.821	0.714	0.786
2000	1.452	1.193	1.240	0.816	0.708	0.819

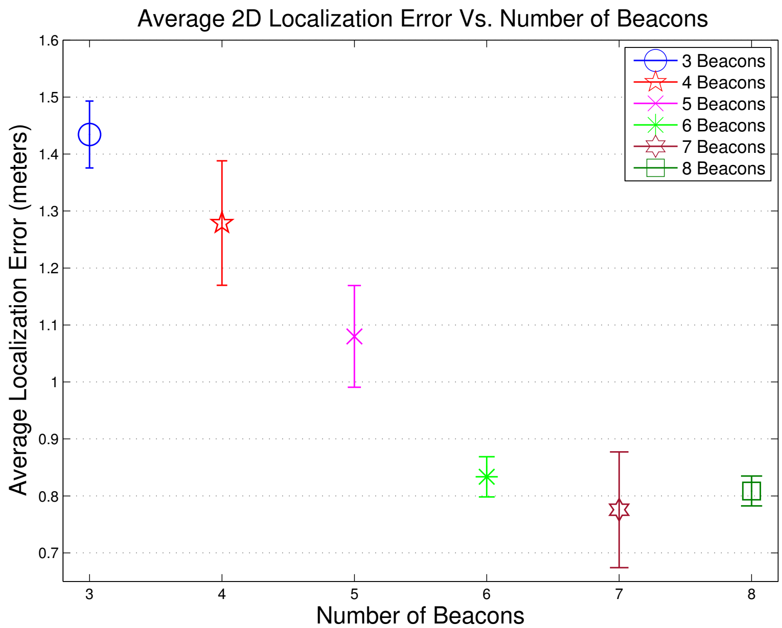

Figure 5.5. Average error vs number of Beacons in 7m × 6m environment with our KFPF algorithm on server side.

Table 5.9 *Standard deviation of localization error vs number of particles for different number of iBeacons in 7 m × 6m environment with KFPF algorithm.*

Particles	Beacons					
	3	4	5	6	7	8
400	0.680	0.555	0.585	0.394	0.431	0.556
600	0.745	0.425	0.532	0.369	0.502	0.705
800	0.756	0.424	0.488	0.384	0.470	0.537
1000	0.826	0.500	0.512	0.331	0.501	0.598
1200	0.827	0.513	0.524	0.341	0.528	0.650
1400	0.702	0.463	0.591	0.387	0.513	0.547
1600	0.773	0.481	0.524	0.410	0.465	0.598
1800	0.833	0.520	0.494	0.394	0.368	0.507
2000	0.696	0.381	0.663	0.352	0.382	0.496

5.4 Particle Filter-Extended Kalman Filter Cascade on the Server Side

The RSSI model used in the previous section and in Chapter 4 for Kalman filter relates the current RSSI values to the previous RSSI values and rate of RSSI change in a linear way. Therefore, we cannot apply Extended Kalman Filter to RSSI filtering (*Even if we do, the Extended Kalman Filter will be equivalent to Kalman Filter*). Therefore, we first use the Particle filtering algorithm to estimate the user's location and then use EKF to reduce the fluctuation in the user position's x and y coordinate as the fluctuating RSSI values also cause the drastic fluctuation in the estimate of the user's location.

The RSSI values from the iBeacons are obtained by the user's device that forwards it to a local Apache Tomcat Server. The particle filtering algorithm running on the server estimates the user's location (The particle's with the highest probability are used to obtain the estimate of the user's location). The PF estimated x and y coordinates are used as input into the EKF algorithm. Below we present the mathematical model for the EKF model.

5.4.1 Mathematical Model for EKF

We use the widely used Position-Velocity (PV) model (Khan, Khan, Khan, & Khan, 2014; Khan, Sottile, & Spirito, 2013) for EKF modeling. Our state Y_i consists of the current x coordinate, y coordinate, the horizontal velocity V_{xi} component, and the vertical velocity V_{yi} component.

$$Y_i = \begin{bmatrix} x_i \\ y_i \\ V_{xi} \\ V_{yi} \end{bmatrix}$$

The state equation for the PV model as given by Khan et al. (2014, 2013) is given below by Equation 5.1.

$$\begin{bmatrix} x_i \\ y_i \\ V_{xi} \\ V_{yi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \delta t & 0 \\ 0 & 1 & 0 & \delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ V_{xi-1} \\ V_{yi-1} \end{bmatrix} + \begin{bmatrix} m_i^x \\ m_i^y \\ m_i^{V_{xi}} \\ m_i^{V_{yi}} \end{bmatrix} \qquad \text{(Eqn. 5.1)}$$

where the matrix given below is the process noise matrix.

$$\begin{bmatrix} m_i^x \\ m_i^y \\ m_i^{V_x} \\ m_i^{V_y} \end{bmatrix}$$

Hence the Jacobian matrix F is given by

$$F = \begin{bmatrix} 1 & 0 & \delta t & 0 \\ 0 & 1 & 0 & \delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The parameter δt is the time interval in which the velocity is constant. Since we obtain the measurements after every 1 second, we used δt =1. Similarly the measurement model in Equation 2.10 can be written as

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ V_{xi} \\ V_{yi} \end{bmatrix} + \begin{bmatrix} n_i^x \\ n_i^y \end{bmatrix} \qquad \text{(Eqn. 5.2)}$$

where the Jacobian matrix H is given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Parameters P, Q and R used in the experiments (chosen after trial and error) are given below.

$$P = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} Q = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix} R = \begin{bmatrix} 0.10 & 0 \\ 0 & 0.10 \end{bmatrix}$$

It is worth mentioning here that the use of Jacobian has linearized the state and observation models. The Extended Kalman Filter involves recursively predicting and updating the state vector as discussed in detail in Chapter 2.

The above PFEKF model was used during the experiments to obtain the user's location. Figure 5.6 shows the average localization error for different number of beacons in the 7m $\times$ 6m environment with server side PFEKF algorithm. We started with 3 beacons and kept adding on more beacons until we reached 8 beacons. We stopped at 8 beacons for sake of comparison with our PF and KFPF-based results. It can be seen that in comparison with both PF and KFPF, the PFEKF has a lower average localization error. However, the performance characteristic is somewhat different. For example, the average localization error for 4 beacons, is slightly more with PFEKF than 3 beacons and we obtain the lowest possible error with 6 beacons. This is because EKF is not an optimal filter and might not always result in optimal model. However, due to modeling the localization problem as a non-linear problem and the use of Jacobian matrix, the overall localization error is much less for server side PFEKF model in comparison with PF and KFPF. Furthermore, we also have a better performing system with just 6 beacons, hence the system is less costly when compared with a 7 beacon system. Table 5.10 shows the average localization error vs. different number of particles for a number of particles with server side PFEKF algorithm. The optimal result is obtained with 6 beacons and 1200 particles. Table 5.11 shows the standard deviation of error vs. different number of particles for a number of particles with server side PFEKF algorithm.

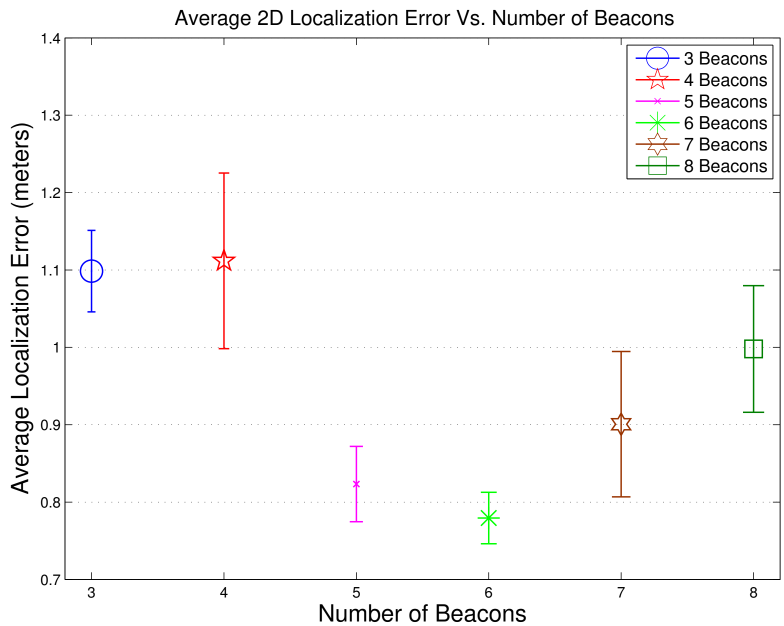

Figure 5.6. Average error vs number of Beacons in 7m × 6m environment with our PFEKF algorithm on server side.

Table 5.10 *Average error vs number of particles for different number of iBeacons in 7 m × 6m environment with server side PFEKF algorithm.*

Particles	Beacons					
	3	4	5	6	7	8
400	0.995	0.874	0.892	0.836	1.032	0.969
600	1.039	1.044	0.838	0.796	1.057	1.086
800	1.120	1.246	0.797	0.765	0.750	1.120
1000	1.118	1.222	0.822	0.756	0.920	1.015
1200	1.079	1.209	0.867	0.720	0.889	1.007
1400	1.112	1.111	0.827	0.793	0.849	1.062
1600	1.155	1.128	0.800	0.760	0.878	0.932
1800	1.153	1.109	0.845	0.788	0.862	0.902
2000	1.116	1.064	0.722	0.800	0.869	0.889

Table 5.11 *Standard deviation of error vs number of particles for different number of iBeacons in 7 m × 6m environment with server side PFEKF algorithm.*

Particles	Beacons					
	3	4	5	6	7	8
400	0.659	0.385	0.536	0.403	0.480	0.619
600	0.579	0.468	0.512	0.384	0.478	0.749
800	0.507	0.648	0.485	0.331	0.576	0.447
1000	0.607	0.717	0.471	0.280	0.505	0.524
1200	0.713	0.751	0.614	0.310	0.608	0.598
1400	0.682	0.628	0.559	0.424	0.567	0.532
1600	0.707	0.734	0.423	0.391	0.606	0.634
1800	0.594	0.813	0.479	0.346	0.561	0.482
2000	0.583	0.637	0.359	0.398	0.570	0.517

5.5 Comparison among server side PF, KF-PF and PF-EKF

In this section, we present a comparison of the different algorithms used in this work for indoor localization. Through the results obtained as a result of the experiments, we show that the cascaded filters significantly improve the indoor localization accuracy when compared with particle filters.

Figure 5.7 compares the average localization error of PF with KFPF in 7m × 6m environment for different number of beacons. From the figure, it is evident that the KFPF cascaded filter approach improves the localization error in comparison with the PF algorithm on the server side. This, as described earlier, is due to the reduction in RSSI fluctuation by Kalman filter that assists particle filter in localization. The performance of both PF and KFPF improve with the addition of beacons with the best results obtained with 7 beacons for both algorithms. The addition of 8^{th} beacon affected the localization error. The box-plot validates our hypothesis and shows that KFPF significantly improves the localization error when compared with PF on the server side. In our experimental setup, the average localization for PF and KFPF on server side was 1.441 meters and 1.035 meters respectively. Hence the KFPF algorithm improved the average localization accuracy by 28% in comparison with the PF on the server side.

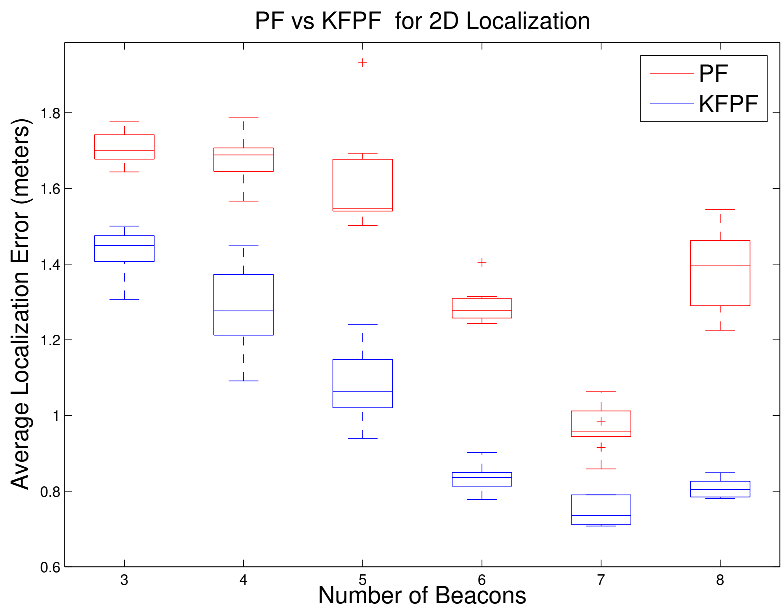

Figure 5.7. Average 2D localization error of PF vs. KFPF in 7m × 6m environment for different number of beacons

Figure 5.8 presents the boxplot highlighting the average 2D localization error of PF and PF-EKF in a 7m × 6m environment for different number of beacons. It is evident from this figure that PFEKF improves the localization error when compared with the PF on the server side. This is because the use of EKF that takes into account the horizontal and vertical velocity component V_x and V_y and reduces the fluctuation in the obtained x and y coordinate. The best localization accuracy is attained with 6 beacons for PFEKF that shows the cost efficiency of the system as we can achieve higher localization accuracy with lower number of beacons. The addition of 7^{th} beacon deteriorates the performance of the PKEKF, however it improves the performance of the PF algorithm. The addition of 8^{th} adversely affects the performance of both PF and PFEKF algorithm indicating that the self-interference can affect the localization error. Another reason for such fluctuating behavior is the unreliability of RSSI due to the presence of varying noise. As

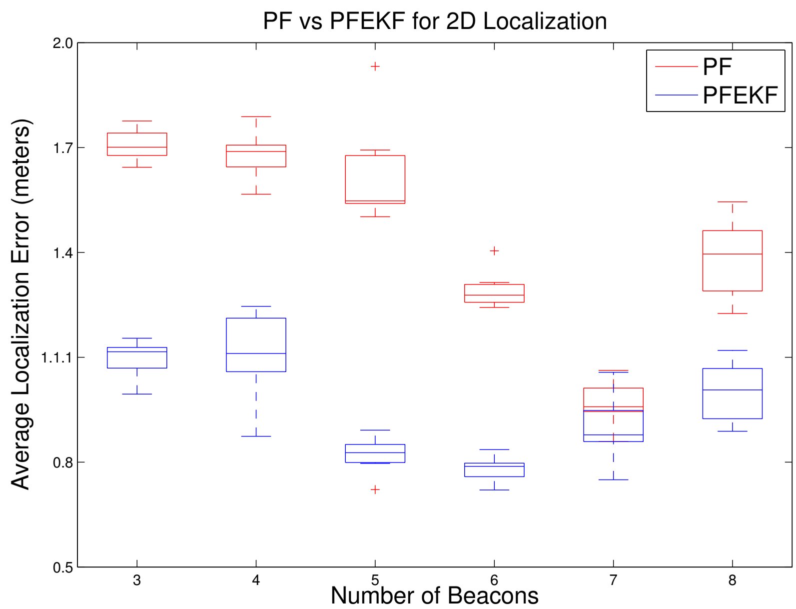

Figure 5.8. Average 2D localization error of PF vs. PFEKF in 7m × 6m environment for different number of beacons

indicated by our experiments, the PFEKF had an average localization error of 0.952 meters that is a 33.94% over the PF server side algorithm.

Figure 5.9 compares the cascaded filters KFPF and PFEKF in terms of the average 2D localization error in 7m × 6m environment for different number of beacons. It is evident from the figure, that both the algorithms have potential for indoor localization. For 6 and lesser number of beacons, the PFEKF slightly outperforms KFPF while KFPF has better localization accuracy for 7 and 8 beacons. As highlighted by Arulampalam (Arulampalam et al., 2002), the EKF algorithm might not always converge to the optimal point. Therefore, we believe that the better performance of KFPF for 7 and 8 beacons is because the EKF module in the PFEKF algorithm did not converge to the optimal point. As indicated by our experiments, the PFEKF improved the average localization error by 8.039% when compared to KFPF.

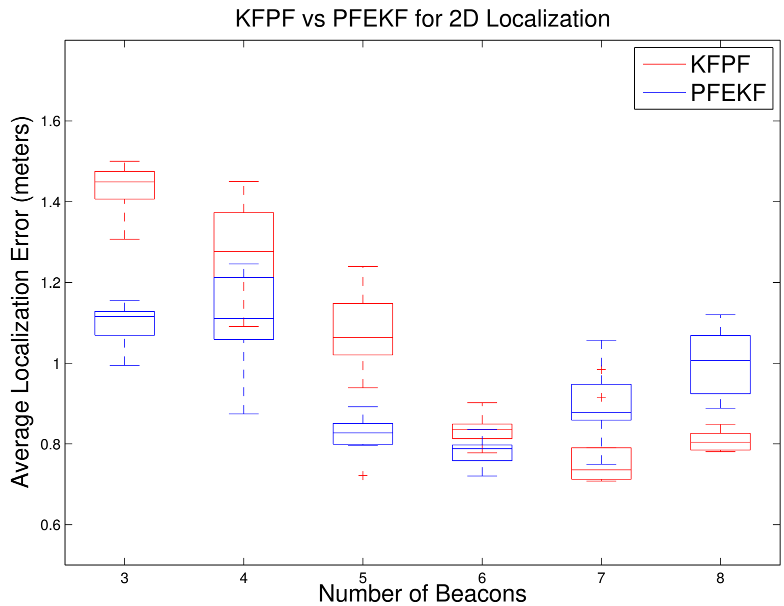

Figure 5.9. Average 2D localization error of KFPF vs. PFEKF in 7m × 6m environment for different number of beacons

We also use two samples based large sample test to validate our hypotheses as explained in Chapter 3. Let $\bar{x}_{PF}, \bar{x}_{KFPF}$, and $\bar{x}_{PFEKF}$ be the average localization error of Server-side PF, KFPF and PFEKF system respectively. Let S_{PF}, S_{KFPF}, and S_{PFEKF} be the standard deviation of the PF, KFPF and PKEKF system respectively. The number of samples m and n in Equation 3.3 be 5940. Below we calculate the Z-value to evaluate if we can reject the null hypotheses.

For comparison PF and KFPF, we put the values in Equation 3.3 and calculate the value of Z as given in Equation 5.3.

$$Z = \frac{1.44 - 1.035 - 0.36}{\sqrt{\frac{0.87^2}{5940} + \frac{0.699^2}{5940}}} = 3.11 \qquad \text{(Eqn. 5.3)}$$

Since Z=3.11$\geq$ 1.645, hence the null hypothesis can be rejected and the alternative hypothesis will hold true. Similarly for the case of PF and PFEKF, we put the values in Equation 3.3 and calculate the value of Z as given in Equation 5.4.

$$Z = \frac{1.44 - 0.9519 - 0.36}{\sqrt{\frac{0.87^2}{5940} + \frac{0.596^2}{5940}}} = 9.357 \qquad \text{(Eqn. 5.4)}$$

As Z=9.357$\geq$ 1.645, hence the null hypothesis can be rejected and the alternative hypothesis will hold true.

From the above results, it is clear that the cascaded approach is better when compared with the PF on the server side. Our results show that iBeacons have the potential to be used for indoor localization which answers our research question. While the iBeacons without any filtering algorithm have huge localization error, the use of combination of signal processing filters can improve the localization accuracy and make it viable for indoor localization. Cascaded filters showed promise for indoor localization and verified our hypothesis that cascaded filters can significantly improve the localization error when compared with PF on the server side. While we used the iBeacons to highlight the effectiveness of proposed cascaded approach, various other RSSI based localization technologies and techniques can benefit from the cascaded filters. RSSI based localization is a challenge, however, a combination of filtering and indoor localization algorithms can help us obtain an effective and accurate indoor localization system. In the next section, we discuss the effect of topology on the performance of the iBeacons.

5.6 Topology of the Beacons

During the course of experiments conducted for this work, we altered the topology of the beacons to analyze how it affects the localization accuracy. We found out that the position of the beacons can drastically affect the localization accuracy. While the positioning of the beacons greatly depend on the environment and the

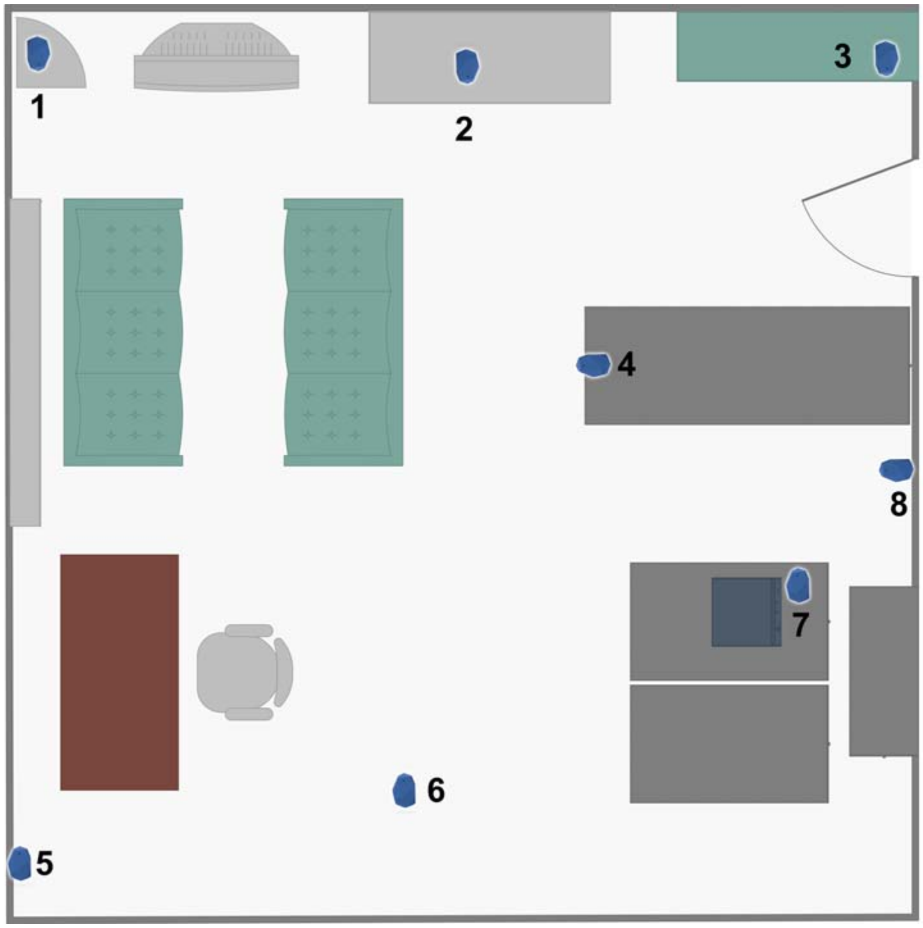

Figure 5.10. The position of the beacons in our experimental setup

room where they are placed for localization, an important point to consider is that the beacons should be placed at suitable heights rather than placing them at lower heights as that drastically affects the localization accuracy and makes the measurement more erroneous. There is no generic model for placement of the beacons as the rooms and deployment sites might vary and we have to alter the number of beacons and position of the beacons based on the number of people, obstacles, and size of the room. But there are some general guidelines that we obtained for deploying beacons.

- Do not place iBeacon close to any obstruction or interfering devices.

- Place beacons at reasonable height so that obstructions can be avoided but do not place them too high as that affects the localization particularly in 2D environment because the RSSI value would decrease with an increase in the height of the beacons (the distance between the user and beacons increases).
- At least 3 iBeacons are required for accurate indoor localization.
- Keep the beacons away from obstacles.
- If any particular location in the room has high localization error, deploy beacons there to improve the localization accuracy.

From the experiments, we conclude that the deployment of beacons should be properly planned as per the environment and deployment site. Since the characteristics and layout of the deployment space vary from one environment to another, it is fundamental to first do a site survey and analyze the environment and then deploy the beacons in a topology which would maximize the localization accuracy. We utilized the same approach for our experiments. Figure 5.10 shows the topology of the beacons that we used. Beacon number 4, 7 and 8 are placed in the locality with high localization error (due to presence of obstructions). This improves the localization accuracy and reduces the localization error in that specific part of the room.

5.7 Summary

In this chapter, we discussed an iBeacon based indoor localization system, and presented results for a user device running particle filters for indoor localization. We also highlighted the experimental setup and results obtained by using PF, KFPF and PFEKF for indoor localization on the server side. Through experimental results, we showed that the use of cascaded filters improve the localization accuracy when compared with a single particle filter. The use of KFPF improved the localization accuracy by 28% when compared with PF while the

PFEKF improved the localization accuracy by 33% when compared with PF. We also provided a discussion on how the positioning of beacons can affect the indoor localization accuracy.

CHAPTER 6. SUMMARY

Apple's BLE-based iBeacon is a technology primarily intended for proximity based services. However, due to its reliance on RSSI, it is prone to noise. Therefore, in its current form, it is not only inviable for indoor localization, but it also results in poor proximity detection accuracy due to the inherent flaws in the proximity detection mechanism currently used by *Apple's CoreLocation Framework*.

In this thesis, we presented an iBeacon-based accurate proximity and indoor localization system. We proposed two server-side algorithms, *Server-side Running Average* and *Server-side Kalman Filter*, for proximity detection that improved the proximity detection accuracy by 29% and 32% when compared with the current approach used by Apple's CoreLocation Framework for proximity-based services. SRA achieved a proximity detection accuracy as high as 96.6% while SKF achieved a proximity detection accuracy as high as 98.3%. The proposed algorithms leverage the high processing power of servers to improve the performance of the proximity detection system and also reduce the energy consumption on the user device by transferring the proximity related calculations to a server. We also presented our cloud-based architecture that can be used to provide proximity based services. While our proposed algorithms and architecture have been tested with the iBeacons, we believe that the proposed approach can also be used by other RSSI based approaches.

We also presented our iBeacon based indoor localization system and leveraged Bayesian theory for making iBeacons viable for indoor localization. Since particle filters have proven to be the optimal Bayesian filters for indoor localization, we showed how PF can be used on the user devices as well as a server to accurately estimate a user's location. We also showed that the use of multiple filters in cascade outperform particle filter (only) used for indoor localization. In our first approach, KFPF, we used Kalman filters to smooth the RSSI values that are reported by the user device (from beacons) to a server. The filtered RSSI values are then used for localization using particle filters. The experimental results show that the use of Kalman filter in cascade with Particle filters improved the localization accuracy by 28% when compared with only particle filters used for indoor localization. We also used a cascade of PFEKF where PF was used for indoor localization and then EKF was used to enhance the measurements obtained through PF. Experimental results showed that PFEKF improved the localization accuracy by 33.94% and 8.039% when compared with PF and KFPF respectively. While PFEKF improves the localization accuracy, it might not also result in the optimal solution. We also discussed how topology affects the localization accuracy. Through experiments, we found out that topology and position of the beacons can drastically affect the localization accuracy. Therefore, it is important to conduct a site survey before the deployment of the beacons. The optimal topology and positioning of the beacons vary from one place to other due to the variation in the channel and environment characteristics which is why every environment should be dealt with separately and use of generic model might no always yield the best possible localization accuracy.

To conclude, iBeacons can certainly be used to provide an accurate indoor localization and proximity system. However, there is need to leverage different signal processing techniques that can account for the environmental noise and improve the system performance. With out proposed system, we achieved a localization error as low as 70 centimeters.

LIST OF REFERENCES

LIST OF REFERENCES

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Ashton, K. (2009). That internet of things thing. *RFID Journal*, *22*, 97–114.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, *54*(15), 2787–2805.

Bayram, I. S., Michailidis, G., Papapanagiotou, I., & Devetsikiotis, M. (2013). Decentralized control of electric vehicles in a network of fast charging stations. *Globecom 2013-Symposium on Selected Areas in Communications (GC13 SAC)*.

Bayram, I. S., & Papapanagiotou, I. (2014). A survey on communication technologies and requirements for internet of electric vehicles. *EURASIP Journal on Wireless Communications and Networking*, *2014*(1), 1–18.

Bergman, N. (1999). Recursive bayesian estimation. *Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation*, *579*.

Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, *7*(5), 28–34.

Chan, Y. T., Tsui, W. Y., So, H. C., & Ching, P. C. (2006). Time-of-arrival based localization under NLOS conditions. *IEEE Transactions on Vehicular Technology*, *55*(1), 17–24.

Channel, I. P. Y. (2015a). *Microlocation and Geofencing - Purdue University.* https://www.youtube.com/watch?v=QCtc7z6PC70. ([Online; accessed 01-Jan-2016])

Channel, I. P. Y. (2015b). *Proximity based interaction with iBeacons at NCSU Hunt Library.* https://www.youtube.com/watch?v=AwckTkpN4-Y. ([Online; accessed 01-Jan-2016])

Djurić, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. *IEEE Signal Processing Magazine*, *20*(5), 19–38.

Evangelatos, O., Samarasinghe, K., & Rolim, J. (2012). Evaluating design approaches for smart building systems. In *IEEE 9th International Conference on Mobile Adhoc and Sensor Systems (MASS)* (pp. 1–7).

Faheem, F., Mahmud, S., Khan, G., Rahman, M., & Zafar, H. (2013). A survey of intelligent car parking system. *Journal of Applied Research and Technology*, *11*(5), 714–726.

Fang, B. T. (1990). Simple solutions for hyperbolic and related position fixes. *IEEE Transactions on Aerospace and Electronic Systems*, *26*(5), 748–753.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861–874.

Firner, B., Moore, R. S., Howard, R., Martin, R. P., & Zhang, Y. (2011). Poster: Smart buildings, sensor networks, and the internet of things. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems* (pp. 337–338).

Gimbal. (n.d.). *Context aware platform.* http://www.gimbal.com/. ([Online; accessed 15-Sept-2014])

Giusto, D., Lera, A., Morabito, G., & Atzori, L. (2010). *The Internet of Things.* Springer.

Günther, A., & Hoene, C. (2005). Measuring round trip times to determine the distance between WLAN nodes. In *Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems* (pp. 768–779). Springer.

Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, *25*(7), 53–82.

Gustafsson, F., & Gunnarsson, F. (2003). Positioning using time-difference of arrival measurements. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 6, pp. VI–553).

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., & Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, *50*(2), 425–437.

Guvenc, I., Abdallah, C., Jordan, R., & Dedoglu, O. (2003). Enhancements to RSS based indoor tracking systems using kalman filters..

Han, Z., Gao, R. X., & Fan, Z. (2012). Occupancy and indoor environment quality sensing for smart buildings. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (pp. 882–887).

Humphrey, D., & Hedley, M. (2008). Super-resolution time of arrival for indoor localization. In *IEEE International Conference on Communications* (pp. 3286–3290).

IBE. (2008). *What is a Smart Building.* http://www.institutebe.com/. ([Online; accessed 02-Sept-2014])

Kaemarungsi, K. (2006). Distribution of WLAN received signal strength indication for indoor location determination. In *1st International Symposium on Wireless Pervasive Computing* (pp. 6–pp).

Kaemarungsi, K., & Krishnamurthy, P. (2004). Properties of indoor received signal strength for WLAN location fingerprinting. In *The first Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (pp. 14–23).

Kais, M., Dauvillier, S., De La Fortelle, A., Masaki, I., & Laugier, C. (2004). Towards outdoor localization using GIS, vision system and stochastic error propagation. In *2nd International Conference on Autonomous Robots and Agents.* (pp. 198–205).

Kanaan, M., & Pahlavan, K. (2004, March). A comparison of wireless geolocation algorithms in the indoor environment. In *IEEE Wireless Communications and Networking Conference* (Vol. 1, p. 177-182). doi: 10.1109/WCNC.2004.1311539

Khan, R., Khan, S. U., Khan, S., & Khan, M. U. A. (2014). Localization performance evaluation of extended kalman filter in wireless sensors network. *Procedia Computer Science*, *32*, 117–124.

Khan, R., Sottile, F., & Spirito, M. A. (2013). Hybrid positioning through extended kalman filter withinertial data fusion. *International Journal of Information and Electronics Engineering*, *3*(1), 127.

Kleissl, J., & Agarwal, Y. (2010). Cyber-physical energy systems: Focus on smart buildings. In *Proceedings of the 47th Design Automation Conference* (pp. 749–754).

Kumar, P., Reddy, L., & Varma, S. (2009). Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks. In *Fifth IEEE Conference on Wireless Communication and Sensor Networks (WCSN)* (pp. 1–4).

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., ... others (2005). Place lab: Device positioning using radio beacons in the wild. In *Pervasive computing* (pp. 116–133). Springer.

Lee, K.-C., Oka, A., Pollakis, E., & Lampe, L. (2010). A comparison between unscented kalman filtering and particle filtering for RSSI-based tracking. In *7th workshop on Positioning Navigation and Communication (WPNC)* (pp. 157–163).

Libraries, N. (2015). *The James B. Hunt Jr. Library.* https://www.lib.ncsu.edu/huntlibrary. ([Online; accessed 01-Jan-2016])

Lingemann, K., Surmann, H., Nüchter, A., & Hertzberg, J. (2004). Indoor and outdoor localization for fast mobile robots. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.* (Vol. 3, pp. 2185–2190).

Liu, H., Darabi, H., Banerjee, P., & Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *37*(6), 1067–1080.

Majumdar, A., Albonesi, D. H., & Bose, P. (2012). Energy-aware meeting scheduling algorithms for smart buildings. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings* (pp. 161–168).

Mascetti, S., Bettini, C., Freni, D., Wang, X. S., & Jajodia, S. (2009). Privacy-aware proximity based services. In *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware.* (pp. 31–40).

Meier, R., Cahill, V., Nedos, A., & Clarke, S. (2005). Proximity-based service discovery in mobile ad hoc networks. In *Distributed Applications and Interoperable Systems* (pp. 115–129).

Meyer, S., & Rakotonirainy, A. (2003). A survey of research on context-aware homes. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers* (Vol. 21, pp. 159–168).

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516.

Niculescu, D., & Nath, B. (2003). Ad hoc positioning system (APS) using AOA. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM.* (Vol. 3, pp. 1734–1743).

Patro, V. M., & Patra, M. R. (2015). A novel approach to compute confusion matrix for classification of n-class attributes with feature selection. *Transactions on Machine Learning and Artificial Intelligence*, *3*(2), 52.

Paul, A. S., & Wan, E. A. (2009). RSSI-based indoor localization and tracking using sigma-point kalman smoothers. *IEEE Journal of Selected Topics in Signal Processing*, *3*(5), 860–873.

Peng, R., & Sichitiu, M. L. (2006). Angle of arrival localization for wireless sensor networks. In *3rd annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON)* (Vol. 1, pp. 374–382).

Savic, V., Athalye, A., Bolic, M., & Djuric, P. M. (2011). Particle filtering for indoor RFID tag tracking. In *IEEE Statistical Signal Processing Workshop (SSP)* (pp. 193–196).

Schor, L., Sommer, P., & Wattenhofer, R. (2009). Towards a zero-configuration wireless sensor network architecture for smart buildings. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings* (pp. 31–36).

Seshadri, V., Zaruba, G. V., & Huber, M. (2005). A bayesian sampling approach to in-door localization of wireless devices using received signal strength indication. In *Third IEEE International Conference on Pervasive Computing and Communications* (pp. 75–84).

Sichitiu, M. L., Ramadurai, V., & Peddabachagari, P. (2003). Simple algorithm for outdoor localization of wireless sensor networks with inaccurate range measurements. In *International Conference on Wireless Networks* (Vol. 2003).

Snoonian, D. (2003). Smart buildings. *IEEE Spectrum*, *40*(8), 18–23.

Subhan, F., Hasbullah, H., & Ashraf, K. (2013). Kalman filter-based hybrid indoor position estimation technique in bluetooth networks. *International Journal of Navigation and Observation*, *2013*.

Teuber, A., Eissfeller, B., & Pany, T. (2006). A two-stage fuzzy logic approach for wireless LAN indoor positioning. In *IEEE/ION Position Location Navigation Symposium* (Vol. 4, pp. 730–738).

Treu, G., & Küpper, A. (2005). Efficient proximity detection for location based services. In *Workshop on Positioning, Navigation and Communication (WPNC).*

Vermesan, O., & Friess, P. (2013). *Internet of things: converging technologies for smart environments and integrated ecosystems.* River Publishers.

Weber, R. H. (2010). Internet of things–new security and privacy challenges. *Computer Law & Security Review*, *26*(1), 23–30.

Weiland, R. J., & Purser, L. B. (2000). Intelligent transportation systems. *Transportation in the new millennium.*

Xu, E., Ding, Z., & Dasgupta, S. (2011). Source localization in wireless sensor networks from signal time-of-arrival measurements. *IEEE Transactions on Signal Processing*, *59*(6), 2887–2897.

Yan, L., Zhang, Y., Yang, L. T., & Ning, H. (2008). *The internet of things: from RFID to the next-generation pervasive networked systems.* CRC Press.

Yang, K., An, J., Bu, X., & Sun, G. (2010). Constrained total least-squares location algorithm using time-difference-of-arrival measurements. *IEEE Transactions on Vehicular Technology*, *59*(3), 1558–1562.

Young, D. P., Keller, C. M., Bliss, D. W., & Forsythe, K. W. (2003). Ultra-wideband (uwb) transmitter location using time difference of arrival (tdoa) techniques. In *Conference Record of the Thirty-Seventh Asilomar Signals, Systems and Computers.* (Vol. 2, pp. 1225–1229).

Zafari, F., & Papapanagiotou, I. (2015, Dec). Enhancing ibeacon based micro-location with particle filtering. In *2015 ieee global communications conference (globecom)* (p. 1-7). doi: 10.1109/GLOCOM.2015.7417504

Zafari, F., Papapanagiotou, I., & Christidis, K. (2015). Micro-location for internet of things equipped smart buildings. *arXiv preprint arXiv:1501.01539*.

Zafari, F., Papapanagiotou, I., & Christidis, K. (2016, Feb). Microlocation for internet-of-things-equipped smart buildings. *IEEE Internet of Things Journal*, *3*(1), 96-112. doi: 10.1109/JIOT.2015.2442956

Zhang, C., Kuhn, M., Merkl, B., Fathy, A. E., & Mahfouz, M. (2006). Accurate UWB indoor localization system utilizing time difference of arrival approach. In *2006 IEEE Radio and Wireless Symposium.*