BLEProxiMesh:

A Bluetooth Smart Mesh Network for Sensing Indoor Location

by

Swati Sehgal, B.Tech, B. Engg.(Hons)

Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfillment of the requirements for the Degree of

Master of Science in Computer Science

(Mobile and Ubiquitous Computing)

University of Dublin, Trinity College

August 2015

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Swati Sehgal

28^{th} August, 2015

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Swati Sehgal

28^{th} August, 2015

Acknowledgments

I would like to sincerely thank my supervisor Dr Jonathan Dukes, for proposing this interesting project and providing me an opportunity to work on it. This project provided me a perfect platform to pursue my passion in programming and allowed me to use a cutting edge technology like Bluetooth Low Energy. The research guidelines, advice and constructive feedback given by him were of great help to me. This project would not have been possible without his help, encouragement and support.

I would like to thank Dr. Ciarn Mc Goldrick, Course Director, M.Sc. in Computer Science Mobile and Ubiquitous Computing, for his advice throughout the year and Meriel Huggard for her constant reinforcement. I would also extend my sincere gratitude to all the staff of Distributed Systems Group, School of Computer Science and Statistics, Trinity College Dublin.

I thank my parents for allowing me to pursue my study away from India and for all their sacrifices. Special thanks to my friend Kiran Tom Kurian, Department of Engineering, Mechatronics, ITB for his assistance and friendly support. Last, but certainly not least,I would like to express my gratitude to all those who gave me the possibility to work on this project.

Swati Sehgal

University of Dublin, Trinity College
August 2015

BLEProxiMesh:

A Bluetooth Smart Mesh Network for Sensing Indoor Location

Swati Sehgal, M.Sc.
University of Dublin, Trinity College, 2015

Supervisor: Dr. Jonathan Dukes

Abstract

BLEProxiMesh is a low-cost, energy efficient sensor and access network for detecting the presence of people (or things) in a smart building. In particular, the sensor network facilitates the discovery and tracking of tags in a building, with a particular emphasis on security and emergency response applications.

The proposed system consists of Bluetooth Smart Tags and Beacons. Each Tag estimates its own position in the building based on pings received from nearby Beacons using Centroid localization algorithm and communicates its estimated position to the Beacon network through one or more nearby Beacons. This information is replicated throughout the Beacon network using the Trickle algorithm, which has been implemented using the

GAP and GATT profiles of the Bluetooth Smart protocol stack. The system is designed to achieve a low-cost, energy-efficient, easy-to-deploy solution for data propagation and replication that is robust enough to be operational in emergency situations which not only involve power constraints but also power failure in the worst case. The system is dynamic and enables fast detection and propagation of changes in the network. The network is designed not only to provide a sensing platform but also to provide an access network for retrieving Tag information.

A working prototype has been implemented to validate the approach using a small-scale Beacon network. Future work will evaluate the performance of the system at scale.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

There is a lot of planning done and strategies made in order ensure fire safety of a building and more importantly people in the building. There are preparedness plans, evacuation action plans, fire prevention plans, special escape procedures for people with disabilities which vary depending on their disabilities. Organisations spend a lot of resources for training and education of people in the building trying to prepare them for emergency situations. This involves designing maps and printing maps, creation of planning guides, planning and execution of fire drills, purchase and installation of fire alarms and ironically a lot of planning is done on how planning is done for dealing with fire disasters. Clearly, this consumes a lot of money, time and resources but when it comes to an actual occurrence of fire in a building all the training goes in vain and what they basically do is to rush towards the nearest exit they are aware of. All the training, instructions, maps go in vain and at an event of a disaster people just follow their instincts. Hence, all the investment in training and planning end in futility.

Even if all the people in the building are following the instructions that they have learned during training, the loophole lies in the fact that all the fire evacuation plans are static. What if the plan directs all the people to an exit located very close to the place where the fire began. This might make the situation even more hazardous.

Also, during the disaster it is very critical to locate people in the building as there might be children or people with disabilities who need special assistance. Furthermore, in case of severe fire there are immediate casualties affected by burns or inhalation of lethal gases who are incapable of evacuating the building themselves and need to be rescued.

Therefore, it is very important to have a fire emergency response system that is dynamic, convenient for the evacuees to follow, easy for the first hand responders to use and facilitates formation of fast and dynamic plans depending on the situation and location of people at that particular time. The system must have the ability to track, locate and discover people to develop an appropriate evacuation strategy. Since, emergency situation naturally have power constraints or power failure, the system has to robust enough to be able to operate with power constraints.

1.2 Proposed System

The main aim of the project is to use the capabilities of Bluetooth Smart for creating a fire emergency response system that facilitates first hand responders by detecting the presence and location of people in a smart building and allowing them to make a quick informed plan about how the rescue operation has to be performed. Furthermore, position is updated dynamically enabling the functionality of tracking the movement of people. The above mentioned system can be developed by creating a sensing and access network of Bluetooth Smart Tags and Beacons. Beacons are low power fixed devices which advertise their position which are used by Tags for position estimation. Each tag estimates its own position in the building and communicates its estimated position to its nearest beacon. Once a beacon in the network obtains Tag information, it disseminates the obtained information throughout the Beacon network. In order to replicate data between beacons a mesh network has to be created.

Conventionally, Bluetooth Smart devices are designed to be part of star based topology but for data replication to be implemented in the system the devices must have the ability to receive and forward data. Mesh network not only overcomes the problems which arise due to failure of central hub in case of star topology, it also provides a solution that is intelligent and supports distributed self-organization allowing the beacon to operate independently and at the same time having the capability of collaborating with other beacons when required [?]. Apart from the fact that mesh network opens door for many game changing applications in industrial, automotive, mobile, emergency, mesh is considered the key to IoT because of the following reasons :

- The network is self sufficient and has self organisation capabilities
- The network is reliable as it has the abilty to deal with node failure [?].

- The network has no gateway and supports decentralised access as a device can access the network by connecting any node in the network.
- The network allows optimum energy consumption and improved network efficiency[?].
- The network is scalable as nodes can be easily added or removed from the network without affecting the underlying network architecture.

1.3 Approach

Currently, a lot of research work is being done on implementation of mesh network consisting of Bluetooth Smart devices as use of Bluetooth smart in a mesh network can be of unmatched value in the IoT. Also there are commercial and open source projects like Wirepas Pino, CSRmesh, nRF51-ble-bcast-mesh that are trying to target IoT applications using BLE. Wirepas Pino uses the concept of time-slotted multi-access channels for multihop routing while also supporting multi-tree routing and multi-sink support[1]. CSRmesh is based on flood based mesh network and overcomes the need of forming connections. The devices part of CSRmesh passively scan for incoming messages trying to identify if the received message is a command or has to be relayed further and hence the approach taken is completely independent of GATT connections [2]. The project nRF51-ble-bcast-mesh tries to create a mesh network by implementing Trickle Algorithm in the timeslots provided by the Softdevice Multiprotocol Timeslot API which is working at the radio hardware module[3].

The approach adopted in this dissertation is creation of mesh network by implementing Trickle Algorithm over GATT and GAP of the Bluetooth Smart protocol Stack. To the best of the writer's knowledge this is a novel approach adopted for creation of a Bluetooth Smart mesh network. Also, the project performs position estimation using centroid based localisation algorithm but the purpose of position estimation using several beacons is to add more accuracy to iBeacon[4] like position estimate identified by proximity values rather than aiming to achieve extremely accurate x and y coordinates. In fact, proximity sensing is done as part of the setup allowing to progress to the core of the project, which is development and validation of a system level architecture of Trickle based Bluetooth smart mesh network implemented using the GAP and GATT profile.

[1] http://www.wirepas.com/technology/
[2] http://www.aircable.net/online/index.php?route=information/information&information_id=12
[3] https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh/blob/master/docs/how_it_works.adoc
[4] http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/

1.4 Benefits and Features

In the recent years, smart building and smart home market has expanded rapidly. According to the report "Smart Cities Will Include 10 Billion Things by 2020 Start Now to Plan, Engage and Position Offerings" conducted by Gartner, one of the leading IT research and advisory company, smart cities and smart buildings would use 1.1 billion devices out the devices constituting the Internet of Things (IoT) and this number would spiral up to around 10 billion by 2020. The growth of WSN and IoT not only provide the technology for building energy efficient smart home solutions, but also allow to induce intelligence in building and hence has been the major driving force for rapid adoption of smart buildings among consumers. Therefore, there is a huge market potential in the area of smart buildings and products like BLEProxiMesh can be used for applications far beyond fire emergency response system

Also, Location information induces intelligence and context into sensor networks and hence adds value to a lot of applications. Localization using WSN can be particularly useful in routing application to find efficient paths e.g. to locate the closest printer, search and rescue of a person in a disaster scenario. WSN are forseen as a realisable solution for applications that involve habitant monitoring, environment monitoring and tracking [3]. Because of sensor miniaturizing and the ability of sensors to be embedded in wearable devices or even be placed in identity cards, they can be used as "location-tracking tags" [4].

Some of the key features of the BLEProxiMesh are:

1. Low-cost and energy efficient
2. Facilitates discovery, tracking and security of people (or objects)
3. Decentralised access to Tag location information
4. Position estimation of the Tags is done by themselves
5. Quick detection and propagation of change of position in the system
6. Implementation of Mesh opens possibilities of many other applications e.g. Lighting, Heating, Emergency, door locking, window sensing

1.5 Objectives

The main areas of research explored during the dissertation includes mesh Network, Trickle Algorithm, Bluetooth Low Energy and Indoor Positioning. Further, research work was done to identify methods of implementing energy efficient system that support multihop data transfer not only using BLE but also other wireless technologies. Also, gossip style network was studied to gain a better understanding of Trickle Algorithm as it uses the concept of "polite-gossip".

The research question of the dissertation is:
Is the implementation Trickle Algorithm over the GAP and GATT profile of BLE protocol stack successfully able to propagate data throughout the network of BLE Beacons in an energy efficient and cost-effective manner?

1.6 Why Bluetooth Smart ?

There are numerous technologies which can be used for Indoor positioning using WSNs. These include Infrared,Ultrasonic, Wi-Fi, RFID, ZigBee, BLE [5]. As suggested by Zhu et al. [5], BLE is identified as the best choice for indoor positioning due to the reasons explained further. Where on one hand the position estimated using Infrared does not provide very accurate results, the results are extremely accurate using ultrasonic but the cost overcompensates for the obtained results. Because of this and the fact that these technologies suffer from poor signal penetration, both of these are not suitable for use at a broader scale [5]. As far as RFID is concerned, node density plays a key role in the accuracy which is not desirable. Finally, Zigbee is ruled out because of its limited range. Moreover, BLE has a lot of advantages that make it suitable for use for the dissertation.

1. Its setup and operational cost is extremely low and energy efficiency very high. A coin cell battery can be used to power a BLE device for years [5].

2. BLE devices have higher range. Making changes to the transmission power allow the range of the device to be changed depending on the kind of application [5].

3. RSSI gives a measure of the received signal strength and is received during scan for any advertising packet which can be useful for position estimation [6].

4. BLE provides flexibility in using the BLE framework depending on the application rather than having fixed use cases [1].

5. Since BLE is present in smartphones, tabletes etc there is a very lower barrier to adoption for BLE and also because BLE devices are mass produced (to be embedded in smartphones, tablets etc) cost per unit is very low. Therefore, pervasiveness of BLE makes it one of the most suitable technology for IoT.

6. BLE has the feature of frequency hopping spread spectrum that makes communication over BLE fast and reliable [5].

1.7 Internet of Things and Wireless Sensor Network

The future of Internet is envisioned as Internet of Things (IoT). IoT is a concept where a huge network of addressable devices interact for exchanging data. The aim of IoT is to amalgamate various communication technologies with physical infrastructure [7] and induce networking and computational capabilities in objects of everyday use, allowing them to collaborating with other such devices . Devices which are part of IoT not only include conventional computing devices like computers, mobiles, tablets but also unconventional devices like wearables, cars, fridges, toaster, devices for measuring physical attributes of environment like heat, light, pressure.

One of the most important components of IoT are Wireless Sensor Networks (WSN) as they enable intelligent monitoring and management while also allowing dynamic transmission of information. Sensors in WSN not only generate contextual information about the physical world [8] but also process the information and allow information dissemination across the network which makes it extremely useful for infrastructure systems. Furthermore, they also have a distinguishing features like sensing and actuating. Because of the fact that WSNs offer flexibility and are easy to deploy, they have seamlessly becoming part of our daily lives and have fuelled the demand for intelligent applications in the areas of healthcare, logistics, domotics, entertainment,assisted and enhanced-living [8]. WSN have completely remodelled the way internet is perceived and is gaining traction not just from researchers but also from consumers. One of the predominant applications of WSN is home automation and monitoring where multitude of sensors with diverse capabilities collaborate together to achieve a common application specific goal. Since the past decade, there has been a transition from interactive systems to intelligent systems where systems are desired to learn behavioural pattern of the user overtime and perform operations based on their learning. Though, it may sound trivial but a lot of research work is being done to induce learning into the systems.

1.8 Dissertation Structure

- Chapter 1 provides an introduction to the project. It explains the motivation behind the project, the proposed system, the adopted approach and its benefits. This chapter also includes key objectives of the project, motivation of using BLE as the wireless technology and a basic introduction to IoT and WSN.

- Chapter 2 discusses the different areas of the project and research work done in each of those areas.

- Chapter 3 describes the system design which includes functions of various devices used in the system, the system topology and an overview of Trickle Algorithm along with the possible approaches of implementing this algorithm.

- Chapter 4 provides complete detail of the prototype implemented providing information about the hardware devices used, data structures defined for development along with the description of proximity sensing, indoor positioning and Trickle Algorithm implementation. This chapter also provides evaluation of the prototype developed.

- Chapter 5 concludes the dissertation by talking about the developed prototype, future work and the challenges encountered during the project

Chapter 2

State of the Art

2.1 Introduction

The State of the Art lays down the foundation for rest of the dissertation. This chapter acknowledges the work done by the research community on BLE based mesh networks and different approaches of implementing Mesh network which are the core features of the project. This section provides an introduction to Bluetooth Low Energy, describes Energy Efficient systems and explains indoor localisation techniques.

2.2 BLE Based Multihop Networks

The work done of Mikhaylov and Tervonen [10] provides an overview of techniques for multihop data transfer over BLE networks. The authors identify that Bluetooth low energy suffers from the limitation of only supporting single hop data transfer and claim their work to be the first implementation of multihop routing over BLE networks. The authors suggest multihop service, provide its implementation and present the results of the evaluation. They explain the basics of BLE and its protocol stack and then mention about the two possible approaches of implementing multihop routing. First one being, implementation of a multi-hop network as BLE L2CAP traffic type using the reserved CID which would involve transfer of IPv6 packets over BLE and the second one being the development of multihop transfer service on the top of GATT layer[10]. The second approach of development of the a service over GATT was used for implementational purpose.

In the work done by the authors four GATT characteristics of MH transfer service are defined:

- Route_entry (write only): used for transferring data about routes

- Own_addr(read only): used to store the 6 byte address of the current device
- Dst_addr(read-write): 6 bytes final destination address
- Data (write only): used for transferring data

The authors define data structures like SeekTbl, RouteTbl and DataBufwhich which store address of the node route to which has to be found, address of the final destination node and data to be transmitted to the final destination respectively.

The approach adopted in the paper consists of node and data discovery phase which is followed by data transfer phase. During the data discovery phase, the address of destination node (say d) is added to the SeekTbl showing that the current node (say s) is looking for node d, followed by which the current node starts advertising that it is looking for the node d. Any node (say x) containing the node d in their RouteTbl responds and initiates a connection leading to establishment of connection between node s and node x, followed by which the master device (node x) sends the route information contained in its RouteTbl to the slave device (node s) which inturn moves the information about node d from its seekTbl to routeTbl. Following this, data transfer takes place between the nodes in which node s writes the address of node d to dest_addr characteristic of node x and node s gets data from its DataBuf and writes them to the data characteristic of the master. The data transfer goes on till the Master has sufficient resources.

This work done by the authors puts a check mark on the feasibility of multihop data transfer over BLE. Moreover, they talk about the challenges faced (e.g.lack of simulators) and identify that security is a feature that that they intend to deal in the future work. Haatanen[11] suggests a number of approaches of implementing multi hop data transfer using BLE. The author also mentions that since BLE originally uses point-to-point network with a simple one-level star topology, mesh has to be implemented on a higher level where the data transfer takes place in one direction from multiple nodes to a single node acting as a gateway[11]. However, the author fail to provide implementational details of this approach and hence doesn't verify its feasibility.

2.2.1 Commercial Implementations

A low cost, ultra low power and hardware independent mesh networking based platform called Wirepas Pino was launched in October 2014, aiming to provide a low bandwidth multi-hop mesh networking protocol stack solution that is scalable, adaptive "fullyautomatic", "self-optimizing" and self-organising[5].This solution was developed jointly by both Wirepas and Nordic Semiconductors by using their protocol stack and hardware respectively to ensure network self optimization, managing high node densities and numerous nodes forming the network. The main motivation behind the development of this project (which is also very closely related to the topic of the dissertation) is the fact that mesh networking can very significantly contribute to a myriad of IoT applications, overcome the dependence on star based topology where there is always concern of central node being the single point of failure of the whole system and increase the reliability of the network while also making sure that security is not compromised[5]. Furthermore, it facilitates applications that don't need infrastructure and complicated configuration procedures.

CSRmesh is another low power wireless technology for the Internet of Things (IoT) that uses Bluetooth Low Energy. This mesh allows a Bluetooth Smart device (laptop/tab/smartphone) to control a large number BLE enabled devices that form a network. Furthermore, it can be used to build intelligence into the network which open pathways of innovative intelligent home-automation systems e.g. heating, lighting, security by adjusting the lighting, heating according the user specific needs. The major advantage of this over other such home automation solutions are that it allows a device to control the BLE devices from anywhere by adopting a decentralized approach making sure that the the communication is not restricted by range and is accessible from anywhere in the network. For spreading the information in the mesh network, forwarding of packets takes place over multiple hops using an addressing scheme that may be used to refer to individual devices or a group[6]. In 2014, CSRmesh was made opensource giving developers opportunity to develop IoT application using CSRmesh APIs.

2.3 Approaches of implementing Mesh Network

2.3.1 Gossip-based Data Propagation

Since mesh network involves propagation of data throughout the network, literature on gossip-based networks was studied in order to understand how data can be distributed throughout the network in an energy efficient manner. Though, there wasn't any implementation of gossip in a BLE based network, it was identified that Trickle algorithm

[5] https://www.nordicsemi.com/eng/News/News-releases/Product-Related-News/Mesh-networking-platform-uses-N
[6] https://wiki.csr.com/wiki/CSRmesh

(implemented in the project nRF51-ble-bcast-mesh[3], described in more detail in Trickle Algorithm Section) uses a form of polite gossip for propagation of data throughout the network.

Research papers on energy efficient gossip based protocols were referred to develop an understanding of techniques used in gossip based networks to maintain energy efficiency.

Lee at al. [12] propose an energy based gossiping protocol that uses gossip probability that is determined based on the residual energy of a particular sensor leading to energy efficiency for the entire network. According to the authors, the key limitation of AODV is that its sink nodes require a lot of power in the system, which might lead to a situation where energy remaining at the sink nodes becomes excessively low making them unable to transfer data and adversely affecting the overall performance of the network. The authors present an innovative approach aiming to reserve as much energy as possible by preventing unnecessary data forwarding. This is done by periodically checking if the energy remaining in the nodes is greater than a certain threshold. If the energy is greater than the threshold it goes into the active mode where it still participates in gossiping otherwise it goes to sleep mode which correspond to a gossip probability of 0.8 and 0.3 respectively[12]. To reduce the energy consumption, the transmission period and the transmission power is reduced by preventing any redundant data transmission.
For experimental setup NS-2 simulator is used and the simulation is done to measure the average remaining energy, standard deviation of remaining energy, deliver rate of packet in the system in cases where gossiping is used, where AODV is used and the other two cases are gossiping with gossip probability of 0.8 and 0.2. The results expectedly show that AODV consumes the most energy as compared to the other protocols and Gossip with gossip probability of 0.3 consumes the least energy. For the data delivery rate, it is observed that gossip with gossip probability has the best results.

Another approach of energy efficient gossip was described by Hou and Tipper [13] which proposes an energy efficient gossip-based sleep protocol (GSP). Since the sensors are low power devices it is very important to make sure that proper measures are taken to preserve energy of the sensors so that they can last longer. The authors put forward a concept in which nodes sleep intermittently depending on gossip sleep probability. The value of gossip sleep probability must always be sufficient enough to keep the network connected[13]. Each node is attached with a timer and whenever the timer expires, the node contemplates based on a probability value if it should sleep or wake up[13]. The

[3] https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh/blob/master/docs/how_it_works.adoc

authors suggest two versions of GSP synchronous and asynchronous versions of GSP and compare them based on exeperiemnts performed using simulations[13]. The authors explain reactive and proactive routing protocols and also the motivation behind sleep based protocol. This protocol ensures that the energy utilized in the network is optimum and hence ensures energy efficiency in sensor networks. The authors describe that their approach aims to reduce flooding and redundant messages in the network while also briefly introducing the concept of percolation theory often applied to the gossip approach.

The authors explain that in GSP, if gossiping probability is p and gossip protocol aims to spread a message in the entire network, then if nodes sleep with probability equal to (1-p) then, it signifies that none of the are asleep leading to very good network connectivity. Two algorithm GSP1 used specifically for MANETs and GSP2 which is an asynchronous version of GSP1 have been presented and performance of the algorithms is evaluated using simulations. A proptocol like GSP uses information obtained from neighbourhood to make the decision if they should be awake or asleep which lead to optimization of energy consumption.

2.3.2 Bluetooth Scatternet

Bluetooth scatternets are networks formed by configuring nodes which are part of various piconets for relaying data over multiple hops. Petrioli and Basagni[14] provide information about a novel protocol for creation of a scatternet. The authors suggest that their proposed solution is novel because of the following reasons [14]:

- it generates scatternet without the nodes being in the transmission range
- establishes multiple paths between nodes
- identifies that the number of slaves in the scatternet can be much more than 7 for this particular implementation

The approach suggested is experimentally verified and the performance of the network is evaluated. Explanation of concepts like topology discovery, piconet formation and piconet interconnection provide the necessary background on the bluetooth technology

The authors introduce basic concepts of bluetooth device roles i.e. master and slave. Piconet is described to be a single hop network that can have at most 7 active slaves and scatternet is described to be a multihop ad hoc network where devices with multiple roles are part of more than one piconets. Topology used for the project is mesh and hence the scatternet formed is referred to as BlueMesh. The developement consists of two phases of topology discovery and scatternet formation which are described in significant detail in the paper. The authors also describes the role selection process and the gateway selection process along with simulation which is used to measure the following [14]:

- average number of piconets.
- average number of roles assigned to a particular node.
- average number of slaves taking part in a piconet
- average length of the routes.

The authors work provide a very good overview of scatternet formation and validates the approach through simulation.

Work by Wang et al. [15] is another implementation of multihop routing where Bluetooth based networks are used in reliable, cost efficient and flexible manner. The authors mention that in order to achieve distributed decision-making, the network must have the following features:

- a collection of data after regular interval
- communication links participants of the network
- sufficient computing power
- real-time control capability which is required ultimately for construction of wireless ad hoc network

The authors explain basics of Bluetooth technology, piconets, Frequency Hopping Spread Spectrum (FHSS) and how it facilitate the master unit in centrally controlling the time-division multiplexing scheme. The authors aim to develop a scatternet structure that is flatter and less hierarchical. The scatternet formation involves establishing connections followed by which the Bluenet Algorithm is used which consists of three phases. The rules of the algorithm as described by the authors are:

1. *Rule 1*: a piconet should not be formed within a piconet
2. *Rule 2*: only a single connection should be formed to the same piconet for a bridge node [15]

3. *Rule 3*: connection has to be only maintained between master and active slaves while the master tries connect to slaves

The first phase of the algorithm tries to leave out some Bluetooth nodes which can be later on attached to the already connected piconets and finally in the third phase piconets get connected to form a scatternet. Followed by this the paper measures the performance of the created scatternet using average shortest-path length and maximum traffic-flows as parameters.

2.4 Bluetooth Low Energy

Bluetooth Low Energy (BLE), also referred to a Bluetooth Smart is the Bluetooth 4.0 core specification originally designed by Nokia as Wibree and was later adopted by the Bluetooth SIG [1]. It is an emerging wireless technology designed particularly for the purpose of short range communication focussing on applications and devices that require relatively low data transfer and operate at low power rate [16]. Bluetooth Low Energy Contrary to the fact that it appears as an optimized version of the Classic Bluetooth, it is not backward compatible with it and was designed primarily as an extremely low cost, low bandwidth, low power, and low complexity [1] radio standard. The rapid penetration of BLE is due to the fact that its growth is very strongly coupled with the growth of Smartphone and the adoption of BLE by companies in the mobile industry like Samsung and Apple.

The benefits of BLE are utilized in various innovative applications not only because of its low barrier to adoption but also because of its extremely low cost. Since its manufactured on such a large scale, the manufacturing cost per unit is very low. Furthermore, BLE devices can run for years powered by just a single coin cell battery and hence, clearly beats it competitors like Wi-Fi, GSM, Zigbee, etc in terms of energy consumption.“With a relatively easy-to-understand data model, no intrusive licensing costs, no fees to access the core specs, and a lean overall protocol stack, it should be clear why platform designers and mobile vendors see a winner in BLE” [1].

2.4.1 Bluetooth Low Energy Protocol Stack

The design of a completely new protocol stack not only promotes flexibility but also extensibility. Due to the protocol design, devices can be developed to support both classic Bluetooth and Bluetooh Smart (such devices are referred to as Dual Mode or Smart Ready devices) [16].

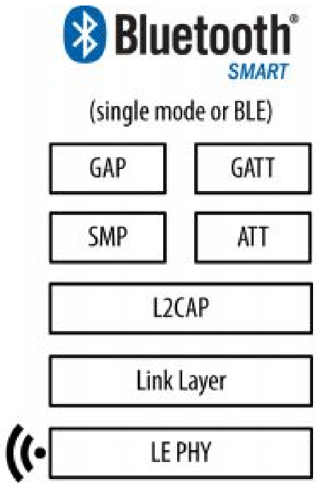

Figure 2.1: BLE Protocol Stack [1].

As described in [1],the BLE protocol stack consists of three parts:

1. **Application**: The highest level of the stack where the logic and the UI are implemented.

2. **Host**: The BLE stack host consists of following layers:

    - Generic Access Profile (GAP)
    - Generic Attribute Profile (GATT)
    - Logical Link Control and Adaptation Protocol (L2CAP)
    - Security Manager Protocol (SMP)
    - Host Controller Interface (HCI)

3. **Controller**: The controller includes the following layers:

    - Host Controller Interface (HCI)
    - Link Layer (LL)
    - Physical Layer (PHY)

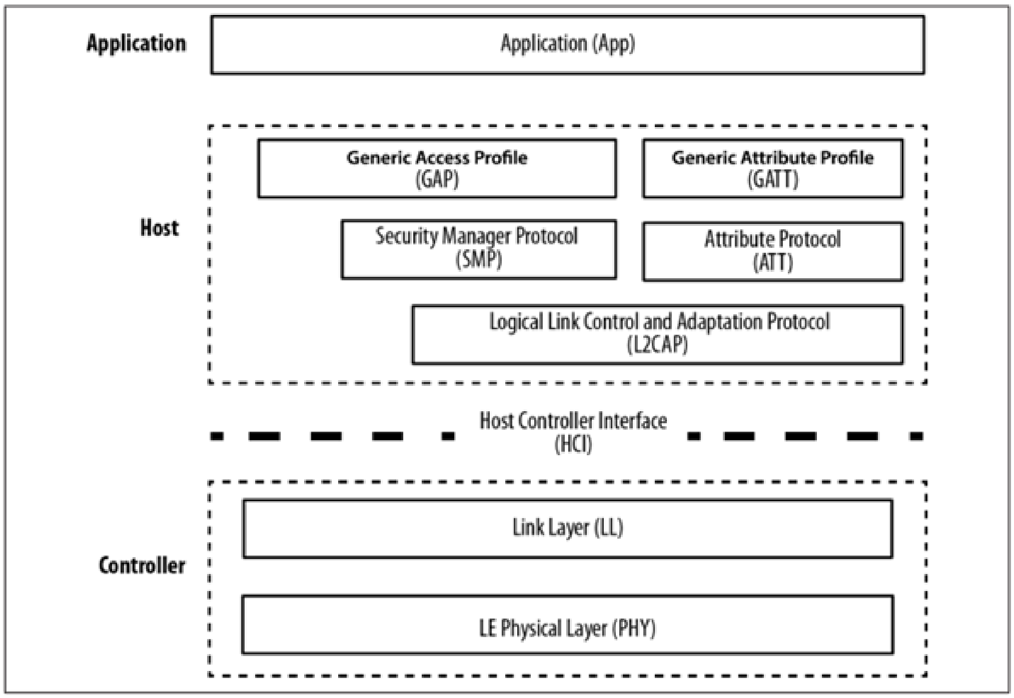

Figure 2.2: BLE Protocol Stack[1].

Each layer has been described in detail as follows:

1. **Physical Layer (PHY)**
   BLE is operational at the frequency of 2.4 GHz in the ISM band and contains 40 channels out of which 37 are data channels and 3 are advertising channels." Device discovery, connection establishment and broadcast transmission" [17] are done using advertisement channels whereas, bidirectional communication is handles by data channels. Frequency hoping spread spectrum is used in order to overcome interference and modulation technique of GFSK is used on all physical channels. The data transfer rate at the physical layer is around 1 mbps.

2. **Link Layer (LL)**
   Link layer is responsible both for advertisement of data using the advertising channels and bidirectional data transfer between devices which happens after connection is established between two devices.

   The roles defined in the Link Layer of the BLE protocol stack are:

   - Advertiser (or Broadcaster)
   - Scanner (or Observer)
   - Master (or Peripheral)

- Slave (or Central)

Advertiser and Scanner are terms used for devices when they are not in an active connection and Master and Slave are terms used for devices when they in a connection [1]. An advertiser broadcasts or advertises data periodically via the advertising channels. A scanner receives data transmitted by advertiser. Another important point to be noted here is that BLE supports two types of scanning:

- *Passive Scanning*: In passive scanning, the scanner tries to read the advertising packets sent by the advertiser and can not request more data.
- *Active Scanning*: In active scanning, the scanner can sends out a *scan request* immediately after advertising packet is received which is a request for more data. This request is handled by the advertiser through a *scan response* which contains additional data. This feature of active scanning increases the amount of data that can be received from the advertiser.

For bidirectional communication, the Central device initiates connection, whereas the Peripheral (which is initially an advertiser) sends advertisement packets and accepts connection initiated by the central device. These different roles allow applications to be created in a more lightweight manner.
Nowadays, devices can be configured to take dual roles of advertiser and scanner simultaneously. Nordic Semiconductors and mbed allow such support with the help of softdevice S130.

3. **Host Controller Interface (HCI)**
Host Controller Interface (HCI) is a standard protocol that allows for the communication between a host and a controller to take place across a serial interface [1]. Commands and events are defined as a part of HCI to facilitate host and controller interaction.

4. **Logical Link Control and Adaptation Protocol (L2CAP)**

L2CAP layer, also part of the protocol stack of the classic Bluetooth has been highly optimised for its use in BLE [16]. The two main functions of L2CAP layer are:

- Multiplexing data between higher and lower layers of the protocol stack and performs encapsulation of the data that makes them suitable for use. [1] [16]

- Though Townsend et al. [1] mention that Fragmentation and reassembly are done in L2CAP layer to ensure that the large packets received from the upper layers are broken down into smaller chunks and later smaller chunks are recombined to be sent back to the upper layer, Gomez et al. [17] mention that these features are not required as the upper layer provide data which does not exceed maximum size of 23 bytes.

L2CAP does not implement any flow control or retransmission mechanism and works on the best-effort delivery approach and plays very important role in two main protocols ATT and SMP [17].

5. **Attribute Protocol (ATT)**
ATT is simple a client/server based protocol that allows two devices to communicate. Server maintains a set of attributes which contains data handled by the GATT [17]. Usually the role of client and server are defined by GATT and has nothing to do with the role of central and peripheral. For communication purpose, the client sends a request to the server asking for using the attributes on the server.Moreover, this protocol allows the server to send notifications and indication to the client [17].

6. **Generic Attribute Profile (GATT)**
Gatt is a framework that defines services and profiles and makes use of ATT to discover services and exchange characteristics between devices [17].

7. **Security Manager Protocol (SMP)**
SMP is responsible for handling all the "key exchange for the purpose of encryption and authentication by using Cipher Block Chaining Message Authentication Code (CCM) algorithm and 128 bit AES block cipher". Several key exchanges need to be done for the connection to be established. SMP provide security in two different modes depending on whether its being used at the link layer or at the ATT layer. It also supports a security mechanism referred to as privacy feature allowing the device to change its private address. Furthermore, other security features supported by SMP include bonding, Pairing and Encryption Re-establishment [1].

8. **Generic Access Profile (GAP)**
The GAP roles (central,peripheral, broadcaster, observer), modes (advertising/ connected) and procedures required for device discoverability and connectivity with other devices is defined by GAP [17]. This also includes security modes and procedures and identification of formats required to manage the data [1].

2.4.2 Bluetooth Low Energy Hardware Configuration

The configuration of the commercially available products based on BLE may be:

- SoC (system on chip) which runs application, host and controller on a single chip
- Dual IC over HCI where one chip runs the application and host and the other runs the controller and the chips communicate through HCI (Host Controller Interface)
- Dual IC with connectivity device where one chip runs the application and the second runs the host and controller and communication between both the chips takes place over a proprietary protocol

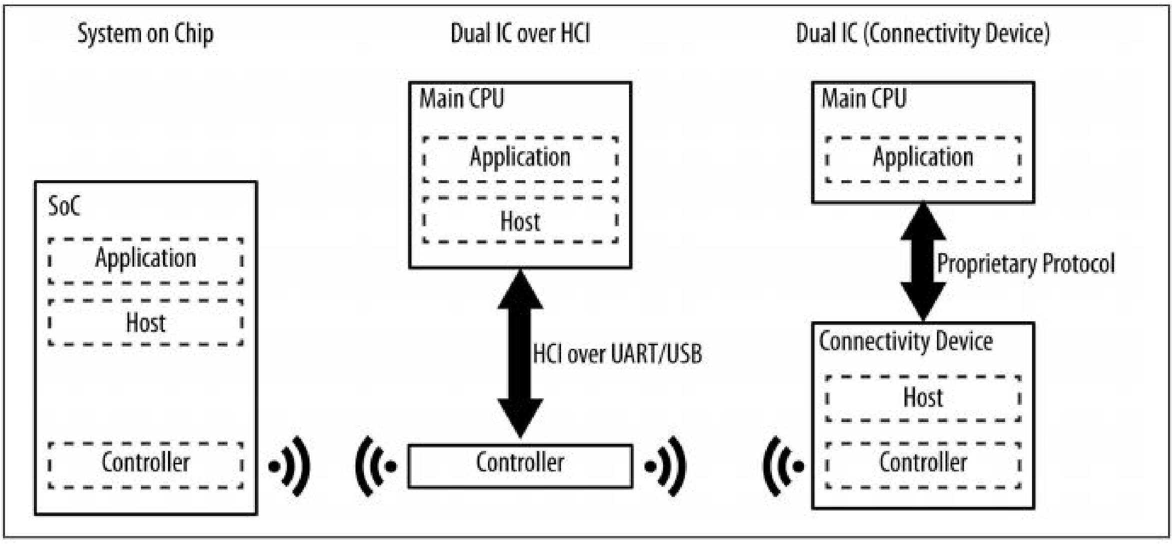

Figure 2.3: Hardware Configuration [1].

2.5 Comparison of BLE with other communication protocols

The work by Siekkinen et al. [18] measures the energy utilization of BLE devices and compares that to the energy consumed of Zigbee devices. The experimental results and analysis done by the authors depict that BLE is much more energy efficient than Zigbee. This comparison is done by measuring the "number of bytes transferred per Joule of energy consumed". The authors claim that they are the first ones to compare the energy consumption of BLE with a competing technology like ZigBee followed by which simple models are built to demonstrate the energy consumption.

For BLE, the network topology chosen is Star topology where the central node is the master may be connected to multiple slaves. The slaves advertise themselves on the advertisement channels and if the Master scans the channels and discovers slaves connection is established between them and data is transferred as events. Devices, routers and coordinator communicating through a CSMA/CA channel and form the ZigBee architecture. The nodes in the network are connected to each other via routers or coordinator.

The parameter measured during the experiment is Energy Consumption along with other tunable parameters. The energy consumption highly depends on the value of the parameters chosen. Energy consumption of the slave which happens due to advertisement depends on both the interval of sending advertisement packets and the number of channels used for this purpose and similarly interval of scanning also directly affects power consumption of the scanning device. Energy consumption of the devices is also monitored when devices are in connection. An innovative approach mentioned in the paper is transfer of IPv6 packets over BLE instead of raw L2CAP frames to avoid data transfer between two slaves via the master. When BLE is compared to ZigBee in a similar scenario, BLE proves to be much more energy efficient. However, due to lack of support of Adaptive Frequency Hopping in the hardware of BLE, the system did not perform robustly and suffered packet loss in the presence of Interference. In case of ZigBee, interference led to increase in transfer time and consumption of more energy.

Zhao et al. [19] also compare BLE with another wireless technology (Wi-Fi in this case). The comparison of both the wireless technolgies was done based on how accurately they can be used for indoor localization making it particularly for the dissertation.The authors discus about BLE in various conditions e.g. indoor/outdoor, line of sight (LOS)/non LOS, experimentally analyse the accuracy of the localization results from both BLE and Wi-Fi and declare BLE as the winner by around 27 percent.Though both these technologies operate at 2.4 GHz ISM band, Wi-Fi gives more RMS error than BLE. Localization(indoor and outdoor) is conducted and comparison between both these technologies in the same environment lead to a conclusion that localization using BLE lead to more accuracy. For the purpose of localization, trilateration is done using the RSSI value which denotes the link quality. The problem in case of localization using Wi-Fi is identified to be the placement of AP not being optimal and its very low scan rate. In the attenuation model, the author describes how RSSI value can be used to determine the distance which is further used for the purpose of trilateration.

$$RSSI(d) = RSSI(d_0) + 10n\log(d/d_0) + X$$

where,

$RSSI(d_0)$	:the RSSI (dBm) at distance d_0
n	:path loss factor,
X	:zero-mean Gaussian noise

The value of RSSI are derived at three different angles of receiver orientation, then the distance is derived from it based on the above formula and compared with the actual distance. Though Wi-Fi gives poor results, it can be seen from the results that in case of BLE, the approximation through the formula is very close to the actual value. This experiment is conducted both indoor and outdoor for both BLE and WIFI in LOS and non LOS setup. In outdoor setup, the path-loss exponent for all three orientations varies between 2.47 and 3.00 for BLE and between 2.63 and 3.00 for WIFI. In indoor setup, the path loss exponent and indoor reception is lower for BLE compared to that in the outdoor setup. In outdoor setup for the Wi-Fi the path loss difference between LOS and non LOS case is very big making the distance obtained from RSSI value very inaccurate. For the indoor localization both static localization and responsive tracking is done (using particle filter) where the author also uses RMS error for various samples of BLE and Wi-Fi signals. For BTLE the RMS error is 3.8m for Wi-Fi RMS error is 5.3m. Finally, the reasons for the better performance of BLE are identified to be

- channel hopping mechanism
- lower transmission power
- much higher sampling rate compared to Wi-Fi.

The authors highlight the positive points of BLE for the purpose of localization and explain both the technologies by comparing them in both LOS and non LOS cases. The research work explains both static and dynamic localization(tracking) and the experimental setup for evaluating the performance of signal strength of RF signals.

2.6 Energy Efficiency Systems

As disccused in **Section 2.3.1**, Lee et al. [12] and Hou and Tipper [13] suggest that energy efficient data propagation is the primary motive of the whole system. Use of gossip probability, timers and checking residual energy of the nodes is done to determine if its eligible to take part in gossip.

As also mentioned in the **Section 2.5**, Siekkinen et al. [18] discuss in detail and provide a comparative study between Bluetooth Low energy and Zigbee in terms of energy efficiency. The energy consumption is similar to that of Zigbee devices during the discovery phase and the reason for that was attributed to high "over the air data rate of BLE". Furthermore, it was noticed that BLE and Zigbee perform almost the same when a single packet per connection event takes place but undoubtedly, BLE outweighs Zigbee where rather than sending single packet multiple packet corresponding to the same connection are send which is clearly attributed to the lower overhead of connection management in BLE.

Shah and Rabaey [20] primarily focus on energy efficient routing that keeps in mind the overall network and not just trying to use the least energy path. The authors explain that most research that has been done on energy efficient routing schemes try to identify the minimum energy path but what has been overlooked is the fact that minimum energy is not synonymous to optimum energy usage as certain parameters like network lifetime and long-term connectivity also need to be taken into consideration. Therefore, the suthors paper introduce a scheme called energy aware routing that does not mind following a slightly less optimal paths if that path leads to overall network efficiency by focussing mainly on routing problem. The energy aware routing protocol is a reactive protocol and aims to find a set of best possible paths and chooses one which is probabilistically the best. Other parameters which are also evaluated include survivability. The authors define sensors, controllers and actuators based in the three main functions performed by sensor networks. The design of the piconode with respect to physical ,media access control and network layers has been explained in detail along with ad-hoc routing schemes which include the description of proactive and reactive routing protocols. The algorithm aims to overcome the problem of find the lowest energy route because of its high implications on the network lifetime. The authors explain the setup phase and the data communication phase along with mathematical explanation. Finally, the paper has a section on the simulation over 76 nodes conducted using OPNET which clearly show that the energy efficiency is very high using this energy aware routing protocol.

2.7 Indoor Positioning Techniques

In the Master thesis by Hasan et al. [16] explain and evaluate different techniques of indoor positioning using Bluetooth Low Energy (or Bluetooth Smart). The authors claim that this is the first project to have researched the use of a new technology for the purpose of indoor localization. The authors explain the basic concepts of Bluetooth and BLE then describe various Signal parameters (Received signal strength indicator, Link quality, Transmitted power level, Inquiry response rate) that can be used for the purpose of indoor localization. It is identified that RSSI is a very significant parameters which is used in many localisation techniques and hence is adopted for this dissertation as well.

The core of the thesis lies in the section where the authors describe various positioning techniques. The techniques that have been used in the implementation of this project are:

- Trilateration
- Filter based technologies
- Fingerprinting
- Cell based positioning,
- Triangulation
- Time of flight (TOF)

Initially the authors gives a general overview of each technique and then in the subsequent section explain it in detail along with the mathematical and theoretical explanation how that technique can be used for the purpose of positioning. Further the authors provide three position estimates namely Least square estimation, Three-border positioning and Centroid positioning and also mention the use of Wi-fi Access points for the purpose of trilateration.

The authors address issues like the use of RSSI and its reliability for the purpose of trilateration and explain how they use the average values of RSSI to make sure that the highly fluctuating readings of the RSSI are smoothed before using its values for position estimation. The authors also mention about the commercially available products like iBeacon, estimote etc that use similar technologies and then explain the experiment conducted by them very comprehensively. They explain the test environment, setup and hardware and also how they applied the algorithms provided in the literature review. Particularly the trilateration and the iterative trilateration has been explained with the help of flowchart hence provides a very good understanding of the implementation done in the project.

The experiments conducted are done using multiple algorithms and the results and compared based on various parameters (e.g. accuracy, precision, calibration complexity, scalability, response time, robustness and adaptability) which are used as evaluation criteria. The authors explain the test setup in terms of hardware and software and also tell about the test environment along with the explanation of the working of the system.

Chapter 3

Design

3.1 Introduction

The diagrams on the next page describe the main components of the proposed system. The system primarily consists of:

1. Tags
2. Beacons
3. Clients

A beacon can be defined as a device that periodically broadcasts its location information in small advertisement packets. The purpose of this device is to provide location based information which helps to contextualize various applications. There are a lot of hardware devices that can be configured to act as Beacons. These devices can range from low-cost, low-power Bluetooth Smart chips to powered Bluetooth Smart ready devices like smartphones, tablets, laptops etc. However, for this project Beacons can be visualised to be devices which are integrated in the building fabric and hence are fixed. Using smart ready or powered devices as beacons adds another layer of complexity of dealing with movement of the beacons in the system which is out of the scope of the project.

Since the scale of the proposed system in BLEProxiMesh can range from a small area in a room to a large building, it would be cost effective if low-cost, low-power beacons are used for deployment. Energy efficient beacons which don't have to be charged regularly are more suitable for the system.The main function of the Beacons in the system is to facilitate proximity sensing of the Tags in the system and enable data replication throughout the beacon network using Trickle Algorithm.

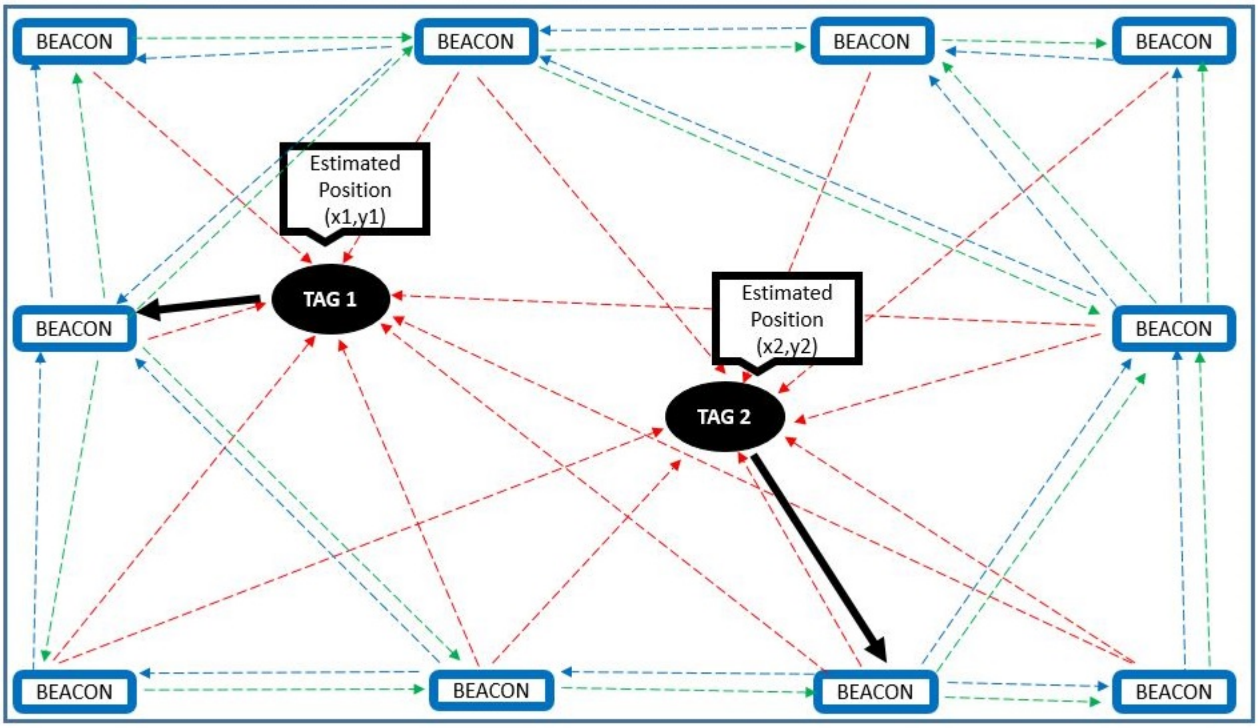

Figure 3.1: The System Design

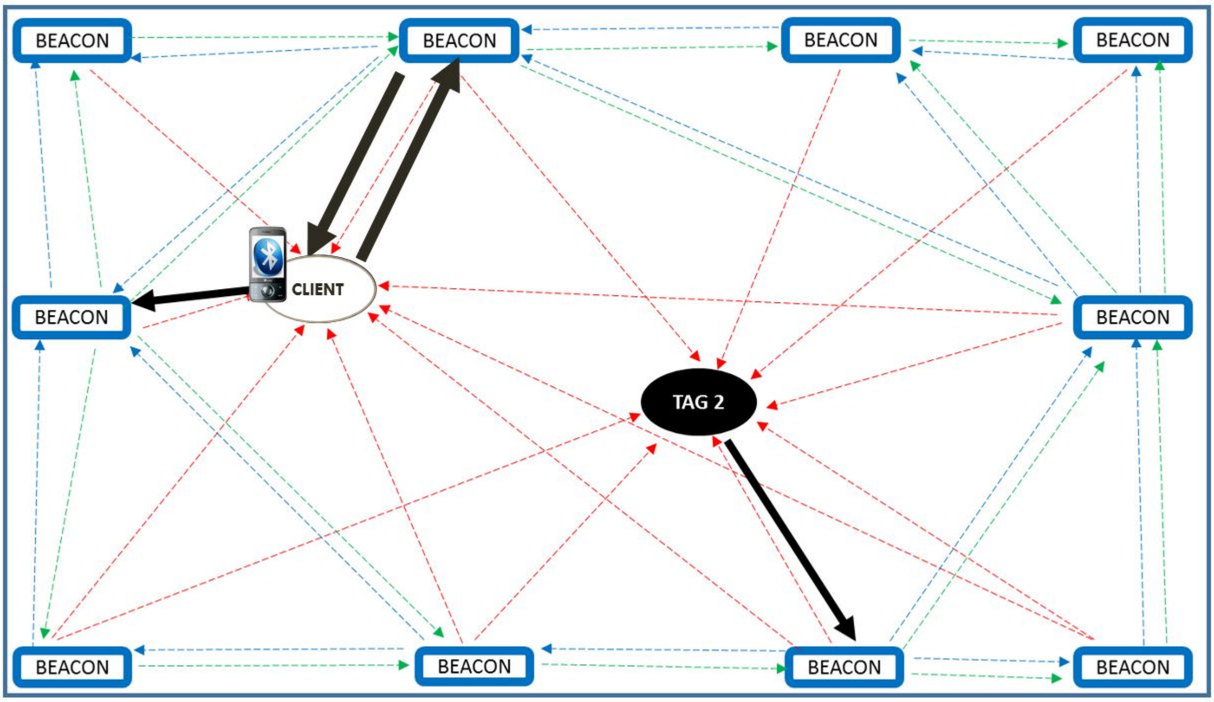

Figure 3.2: Querying Location Information

In the system, Tags are devices that are intended to symbolise/depict people in the building which means that everyone in the building must carry Tags. Therefore, Tags should be light weight Bluetooth Smart devices that are convenient for people to carry. They should have a long battery life so that people don't have to go through the hassle of charging them regularly. There are a lot of commercial products in the market ranging from Bluetooth Smart stickers like StickNFind[7] and Tile[8] which are very coin size devices that can be stuck, attached or tied to anything that people carry with themselves like their key chain, bag or wallet.A commercial product called PROTAG Elite[9] is a credit card size Bluetooth Smart Tag which can be easily slipped into a persons wallet. A very convenient Bluetooth Smart Tag would be if the Bluetooth Smart chip can be embedded or integrated in the ID cards of employees in an organisation or students in an institution. This would allow the card to be used for locating and tracking the movement of people in the building using their ID card. Furthermore, the current NFC based cards require the cards to be brought very close to NFC scanners in order to open doors. If Bluetooth Low energy technology is used instead, Bluetooth Smart ID cards would be detected if they are at close proximity to the door even if the ID card is present in the pocket or bag of the person trying to access the building.

Another possible solution is to use smartphone of the people in the building as their identification tags. However, since BLE is a relatively new technology all the people in the building might not have a smartphone that supports BLE.

Finally, a client in the system is likely to be a powerful Bluetooth Smart device or even a Smart ready device that preferably has a user interface allowing the user of the device to request location information of any Tag in the system through an application running on the device. Hence, a client has the ability of retrieving location information of any Tag in the system.

[7] https://www.sticknfind.com/
[8] https://www.thetileapp.com/
[9] http://theprotag.com/

3.2 Functions of the Devices

The following section describes the functions of a Tag, Beacon and a Client in the system

3.2.1 Beacons

As described previously, in the system Beacons are extremely low power fixed Bluetooth Smart devices which advertise their position to the Tags in the system and eventually use Trickle Algorithm for achieving data replication throughout the network.
The functions of a Beacons in the system are:

1. To create custom advertising data which contains custom advertising structures to advertise their position in the form of (x,y) coordinates in the advertising structure of type "MANUFACTURER SPECIFIC DATA".

2. Creating a service which includes creation of characteristic, assigning UUID to both service and characteristics and adding it to the Bluetooth smart protocol stack

3. Setting up states for both advertising and connection.

4. The beacon which is closest to a particular tag receives the Tag id, Timestamp and (x,y) coordinate corresponding to that Tag. This position is calculated by the tag using Centroid Based algorithm and the RSSI value.

5. It is responsibility of all the beacons to maintain database containing information about all the Tags in the system. This database is maintained in the form of a Queue using Linked List. Since, Linked List support dynamic memory allocation it was better to use it as a data structure instead of an array because Tags and their corresponding information can then be stored on each node of the linked list. Since Beacons are devices with limited memory this decision supports efficient memory utilisation.

6. The Beacons have to ensure that information about all the tags is propagated throughout the network. This is achieved by the use of Trickle Algorithm. All the trickle related parameters like k (threshold), c (counter), t (timer) for each tag is also maintained in the database.

3.2.2 Beacon Custom Advertisement Packet

The project uses the custom advertisement packet which consists of smaller advertising structures which are represented by smaller blocks in the Table below. The advertising packet consists of advertising structures of the type Flags, Complete list 16 bit service, Manufacture specific data, Complete local name, Tx power which are always present in the advertised packet. However, the fifth advertising structure is either of type Trickle_Metadata or Trickle_Fulldata.

AD 0	AD 1	AD 2	AD 3	AD 4	AD 5/ AD 6
Length | Type | Data					

Table 3.1: Custom Advertising Structures within Advertising Packets

Advertising Structure	AD Type	ADStruct Data(L,T,D)
AD 0	FLAGS	2 | 0x01 | 0x06
AD 1	COMPLETE LIST 16BIT SERVICE IDS	3 | 0x03 | 0x00A0
AD 2	MANUFACTURER SPECIFIC DATA	3 | 0x0FF | 0x0303
AD 3	COMPLETE LOCAL NAME	3 | 0x09 | 0x3400
AD 4	TX POWER LEVEL	2 | 0x0A | 0xBC

Table 3.2: Custom Advertising Packet(Detail)

Advertising Structure	AD Type	ADStruct Data(L,T,D)
AD 5	TRICKLE METADATA	L | 0x1F | D
AD 6	TRICKLE FULLDATA	L | 0x13 | D

Table 3.3: Custom Advertising Structures

3.2.3 Tags

As described earlier, Tags are lightweight Bluetooth Smart devices whose location is estimated. These devices also have the ability to access location information of any other Tag in the system. The access is completely decentralised and requires no gateway.
The functions of a Tag in the system are:

1. Scanning the beacons present in its vicinity and getting their position for using in the Centroid based position estimation algorithm.

2. Tags perform the function of RSSI smoothing through sorting and averaging. Since position estimation is done using RSSI value which fluctuates a lot, the average of the strongest signal values is used for position estimation.

3. Tags calculate their distance from all the beacons and store them in ascending order of their distance. This allows them to use the closest beacons for position estimation.

4. Tags are responsible for initiating connection with the nearest beacon.

5. After the connection establishment, Tags send their tag id, timestamp and the calculated position to the nearest beacon through the GATT Characteristic.

6. Tags are also responsible for disconnecting the connection once it knows that its nearest Beacon has received the information it was trying to send. This is important to make sure that all the Beacons are advertising their position which might be needed by other tags in the system for proceeding with their position estimation.

3.2.4 Clients

Clients can be considered to be powerful Bluetooth Smart compatible devices like smartphone or Tablet that can be used to retrieve information from the network by querying a beacon using a tag id.

3.3 System Topology and BLE Roles

3.3.1 Peripheral and Central Role Assignment

There are two possible approaches:

1. Tag-Central Beacon-Peripheral Model
2. Tag-peripheral Beacon-Central Model

Details of both the approaches along with their advantages and disadvantages have been explained below.

1. Tag-Central Beacon-Peripheral Model

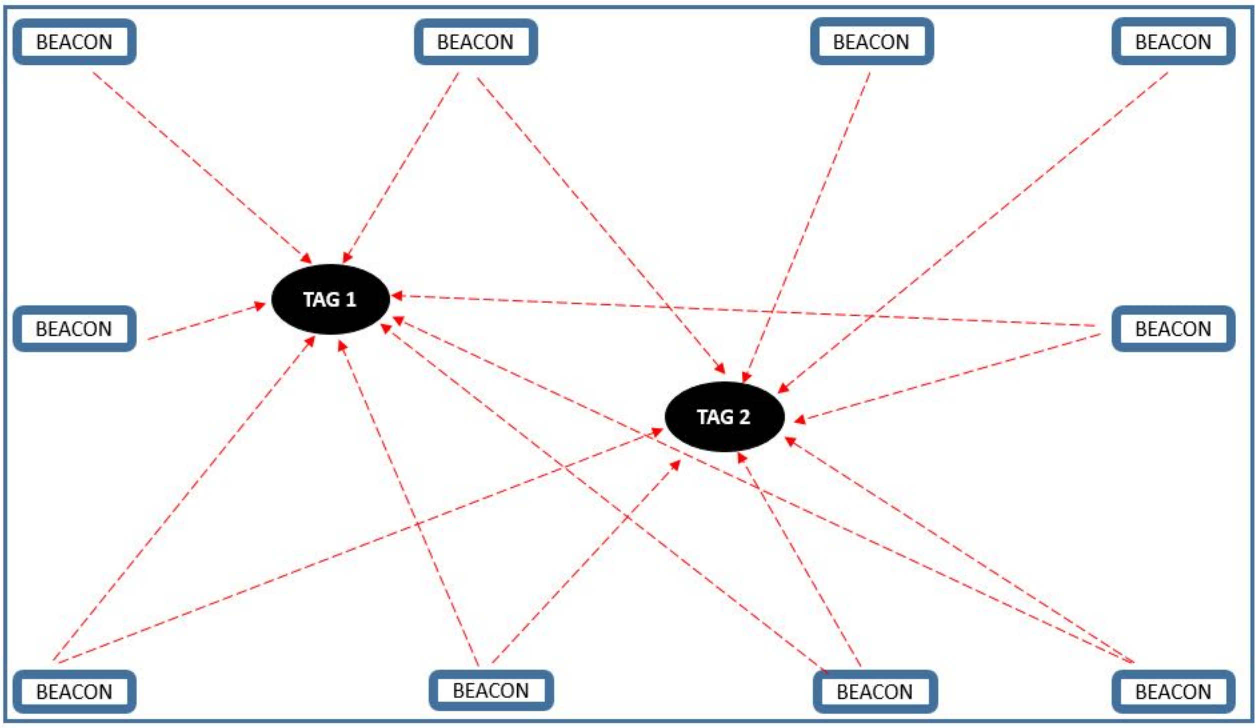

Figure 3.3: Tag-Central Beacon-Peripheral Model

This approach means that before connection establishment between a Tag and a beacon, Tag acts as a Scanner or observer scanning for all the advertisement packets advertised or broadcasted by the Beacons (which act as advertiser/broadcaster) and once connection is established Tag becomes the central and beacon becomes the peripheral.

Advantages of this approach are:

(a) The Tags can receive the beacon position coordinates in the advertising packet which makes the process of position estimation very convenient.

(b) The scanning device receives the RSSI value when an advertising packet arrives which makes it very helpful during position estimation. The transmission power is received as a separate advertising structure in the adversing packet.

(c) Since, every tag calculates its own position it becomes very good from authenticity point of view as it is always ensured that the position estimation corresponding to a particular tag is always generated from a trusted source (the tag itself) leaving very less room for manipulation of data by a malicious node.

(d) It becomes very beneficial for the tag to get an overall picture of the network and at the same time to make sure that the closest beacons (identified based on the signal strength value) are used for position estimation . Because of this, the tags easily maintain information about its distance from all the beacons.

(e) Another functionality that was very effortlessly incorporated using this model was the ability of the tag to measure the distance by which it has moved. Because of this, only if a tag changes position more than a certain threshold, position is updated and sent to the nearest beacon.

The disadvantage of this approach is:

(a) Beacons are continuously advertising in this model even if there is no Tag in their vicinity. Even though beacons are low power device, this is an unnecessary wastage of energy as the devices have to wake up every advertising interval to advertise. One way of overcoming this problem is by having a longer adversing interval but that adversely affects the speed of position estimation.

2. Tag-peripheral Beacon-Central Model

   This approach means that before connection establishment between a Tag and a beacon, the Tag acts as an advertiser or broadcaster sending out advertisement packets to the beacons to make its presence felt to the beacons. The beacons in turn are responsible for scanning or observing the advertisement packets and initiating the connection between Tags and the beacons. This approach is more energy efficient but, leads to some complications which have been explained as disadvantages of this approach

   The advantage of this approach is:

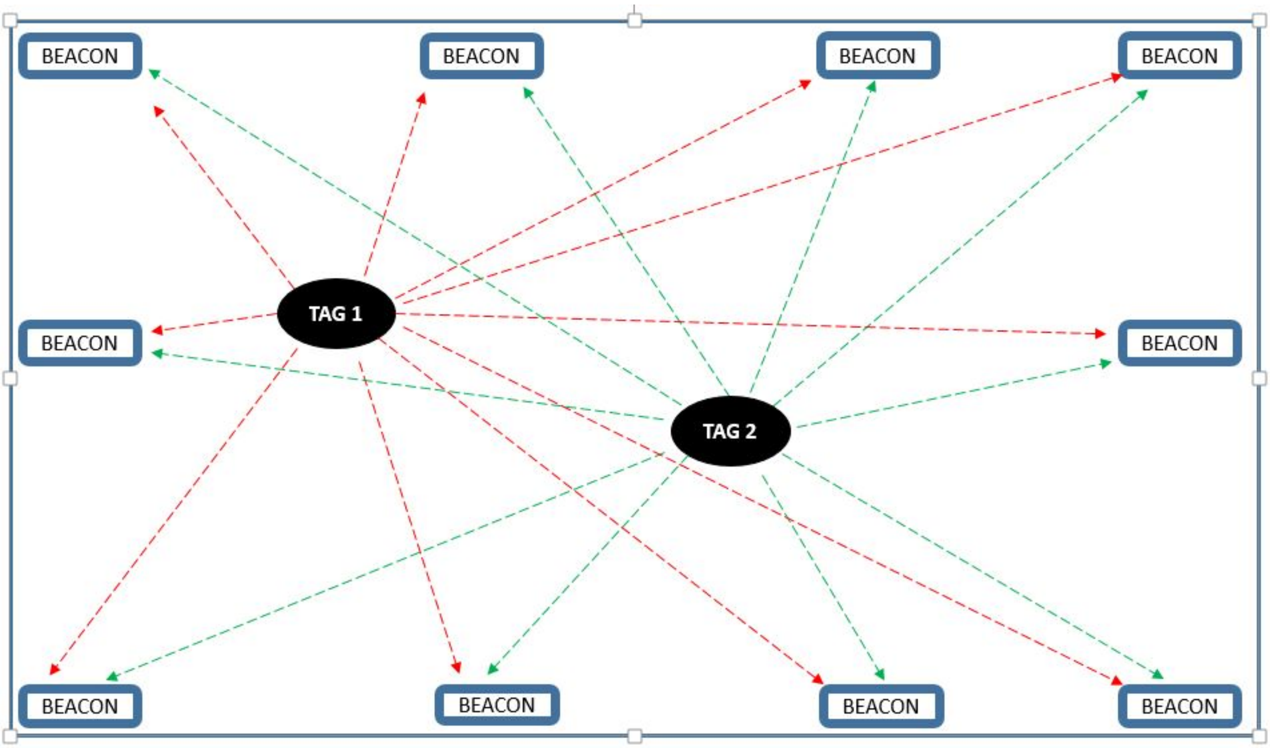

Figure 3.4: Tag-peripheral Beacon-Central Model

(a) Unlike the first approach, beacons don't have to advertise all the time. Only when a tag appears in the vicinity the beacons need to send their coordinates. This makes the whole system more energy efficient.

The disadvantages of this approach are:

(a) Since, the beacons are responsible for initiating the connection, multiple beacons might send connection request at the same time. This might lead to a situation where a tag is bombarded with a lot of connection requests and hence needs have a mechanism to decide which Beacon it should connect to.

(b) The Tag has to make sure that it has received position of all the beacons in the system for accurate position estimation. Unless it receives data from all the nodes, it is unable to sort them based on their distance which might lead to a situation where it is using a farther node as compared to a closer node for estimation. This is because, nodes which are closer to the tag have better signal strength and are naturally suitable to be used for accurate position estimation.

Eventually, decision was taken to use Tag-Central Beacon-Peripheral Model because of it many advantages. However, during the development phase, it was realised that it would be beneficial if the Beacon had a dual role of advertiser and scanner. This would help it to advertise its position which is used by the Tags to perform position estimation, and later on act as a scanner/observer as well which would be required for the implementation of the Trickle Algorithm. Complete implementation of the algorithm is explained in the Implementation section.

3.4 Location Estimation

Centroid based positioning technique was used for position estimation in the system. Hongyu [2] suggests a weighted centroid based approach which makes the process of position estimation which uses geometric operations rather than iterative operations. This techniques was relatively easy to understand and implement and hence was used for position estimation in order to progress quickly to the core of the project where Trickle Algorithm had to be implemented over GAP and GATT.

The approach adopted by the author uses RSSI value for distance calculation and then optimize the algorithm by choosing the nearest nodes for position estimation. Eventually, weights are used to further improve the accuracy of the system. One of the advantage of using RSSI value is that using it neither increases the hardware cost nor does it have any implications on energy of the system.

The author also talks about range-based and range-free positioning methods and explains the difference between them. Range based positioning methods require calculation of distance or the orientation between nodes in order to estimate the position Some of the range based positioning methods mentioned are Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA) and Received Signal Strength Indicator (RSSI) [2].

Steps involved in centroid based algorithm are:

1. Distance calculation is performed using the RSSI value. The formula used for this purpose is:

$$\mathrm{p}_r(\mathrm{d})=\mathrm{p}_r(\mathrm{d}_0)_{dB}+ 10\mathrm{klog}(\mathrm{d}/\mathrm{d}_0) +\chi_\sigma \quad [2]$$

   Here,

   $\mathrm{p}_r(\mathrm{d})$:path loss at distance d
   $\mathrm{p}_r(\mathrm{d}_0)$:path loss at distance d
   χ_σ :Gaussian distribution with mean 0 and variance L

2. As described by the author, if A, B, C are nodes with coordinates (x_A,y_A), (x_B,y_B), (x_C,y_C) respectively and we have to find (x,y) coordinates of point D, the device at point D needs to get the RSSI values from nodes (A, B, C) for calculating distances d_A, d_B and d_C respectively using step 1.

3. Using distance formula, three equations are formed which correspond to the measurement of the distance if each point A, B and C from point D

   The equations that reflect the above statement are:

$$(x - x_A)^2 + (y - y_A)^2 = {d_A}^2$$
$$(x - x_B)^2 + (y - y_B)^2 = {d_B}^2$$
$$(x - x_C)^2 + (y - y_C)^2 = {d_C}^2$$

4. Since in the above equations, it can be clearly noticed (x,y) lie on intersection of three circles with centres (x_A,y_A), (x_B,y_B) and (x_C,y_C) and radii d_A, d_B and d_C. This can be geometrically represented by the following image:

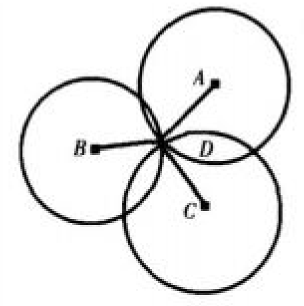

Figure 3.5: Centroid Based localization[2].

5. For finding out the coordinates of point D (x,y) the adopted approach taken is that instead of directly aiming to find the intersection of the circles, first the line of intersection of a pair of circles is found, following which the point of intersection of the lines is found which can be geometrically seen from the following diagram:

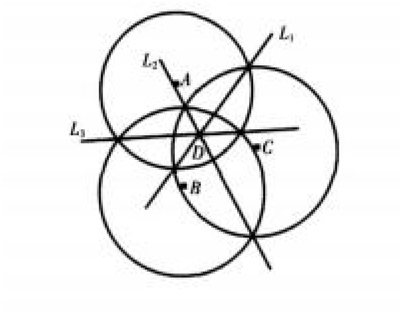

Figure 3.6: Position Estimation Using Centroid based localization[2].

6. As an additional step for optimization purpose, the above mentioned steps are performed using the 4 nearest nodes taking three nodes at a time in order to obtain 4 such points, $D_1(x_1, y_1)$,$D_2(x_2, y_2)$,$D_3(x_3, y_3)$ and $D_4(x_4, y_4)$ [2] and then finding (x,y) by:

$$x = \frac{(x_1+x_2+x_3+x_4)}{4} [2]$$
$$y = \frac{(y_1+y_2+y_3+y_4)}{4} [2]$$

This step adds an additional layer of optimization and make the position estimation more accurate.

Clearly this position has been very mathematically explained which is extremely helpful for developing an understanding, helped to make an effortless transition from theoretical algorithm to code. The fact that this algorithm is fast make it totally apt for use in scenario where the tags/nodes are constantly changing position

3.5 Trickle over Bluetooth Low Energy

3.5.1 Overview of Trickle Algorithm

Trickle is an algorithm that allows energy efficient,simple, reliable and scalable way of information dissemination and propagation by the use of 'polite gossip' policy [21] and an underlying inconsistency model to avoid flooding of data into the network [22]. This algorithm uses concept of epidemic, scalable multicast and broadcasts in WSN for exchange of metadata, as its basis. In this algorithm nodes receiving metadata that they already have remain quiet and the nodes receiving old metadata try to update other nodes by broadcasting updates [21]. Kermajani et al. describe it as a transmission scheduling algorithm primarily used for the purpose of determining whether data should be transmitted or not and when that data needs to be transmitted [23]. The techniques that make it fast and reliable for resolution of data inconsistency are *adaptive transmission period*, *timer based suppression mechanism* and *redundant message suppression mechanism* [23] [24]. The algorithm allows the inconsistencies to be resolved dynamically while also preventing redundant transmission of data when the network is in steady state making it energy efficient.

The motivation of using this algorithm was drawn from nRF51-ble-bcast-mesh[10] which was created in collaboration with The Norwegian University of Science and Technology(NTNU).

Features of this algorithm which make it suitable for use are:

1. **Energy Efficiency and Low Maintenance**: The algorithm controls the rate of data transfer so that devices that are part of the network receive only a small trickle of data sufficient enough to keep all the nodes in the network updated [21]. This is a very important property of the algorithm as it ensures energy efficiency of resource constraint devices. Furthermore, the network prevents data exchange *redundant message suppression mechanism*, if it is in a stable state thereby reducing the cost of maintenance significantly [21][23].

2. **Rapid Data Dissemination**: Any updates generated in the network must be propagated rapidly to all the nodes even if the network is large in scale and contains multiple hops [21].

3. **Scalability**: Since the scale of sensor networks is usually very large,it becomes extremely important to ensure that the algorithm dynamically scales to changes in network density which might arise due to node failure [21].Timer based based suppression mechanism plays a key role and allows communication cost to scale logarithmically with changes in network [24].

3.5.2 Description of Trickle Algorithm

Levis et al. [21] describe the various parameters,variables used in the algorithm and the steps involved in its implementation.

[10] https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh

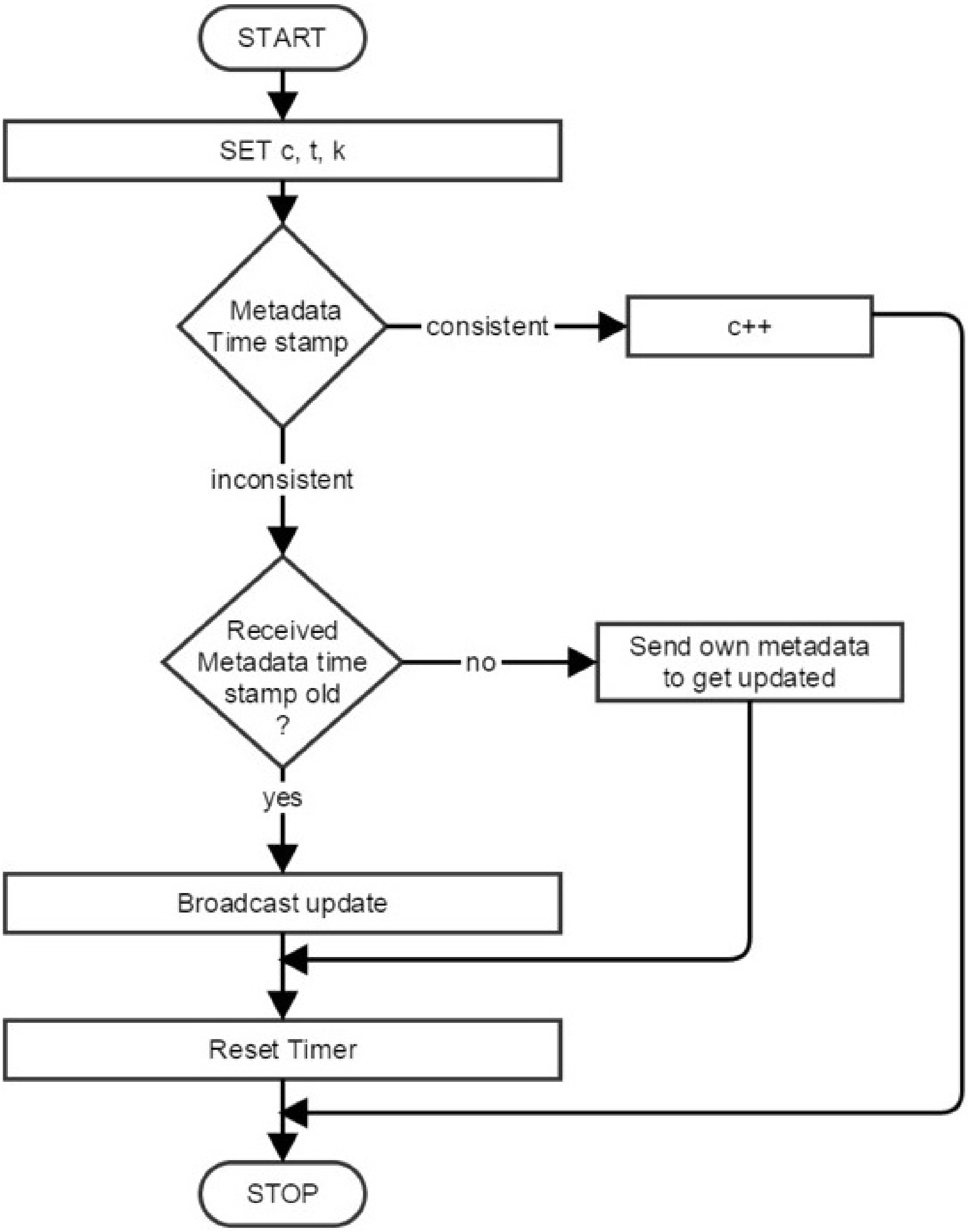

Figure 3.7: Trickle Algorithm

Parameters as described by the Levis et al. [21] in the algorithm are:

τ	:time constant
k	:threshold
t	:time within current interval
c	:counter

As described by Levis et al. [21] the Rules of the Algorithm are:

1. Initialisation of t=rand [0,τ]

2. If a node hears consistent metadata (same as it has), c++

3. If trickle hears inconsistent metadata

   (a) if node with metadata ϕ_x hears ϕ_{x-y}, it broadcasts the data to update rest of the nodes .

   (b) if node with metadata ϕ_x hears ϕ_{x+y}, it broadcasts its own metadata ϕ_x forcing the updated nodes with metadata ϕ_{x+y} to send updates.

4. At time t, if c<k, Broadcast its metadata.

5. When interval expires (t=τ), t=rand [0,τ] and c=0

The above mentioned algorithm suffers from short-listen problem [21] and hence was modified in the RFC6206 [22] by introducing a half interval listen only period which inturn puts a question on trade off between transmission cost and delay at each hop [24].

3.5.3 Trickle Implementation Approaches

Trickle Algorithm is used for the purpose of propagation of information about all the tags throughout the network in an energy-efficient way. It broadcasts metadata and timestamps rather than the whole message to avoid redundant transmissions. According to the algorithm, if a node receives metadata with timestamp that it already has, it stays quiet. However, if it receives metadata from a node that is less updated it sends out broadcast aiming to bring rest of the nodes up to date. In case it receives data from node with less updated than its own it broadcasts its own data hoping that an updated node would send out the updated information. There were three possible ways of achieving this goal.

In each of the below described approaches, the Tag acts as GATT Client and writes value (tag id, timestamp and the position) to characteristics everytime its position is updated.

The position update happens if Tag changes its position more than a certain distance. This distance value can be changed depending on the application.

1. **Using The GATT Service**

   This method allows the Beacons in the system to advertise their metadata (tag ids and their corresponding timestamps) using a custom adversing packet and if an update is needed, full data is sent through the GATT update characetistic which is a write only characteristics. Here, full data includes tag id, timestamp and position. This approach allows the beacons to compare their data with the advertised metadata and only if the beacon identifies that a particular beacon has older metadata, it sends connection request and sends it the updated data.

   This approach requires the Beacons to take the role of both GATT Client and GATT Server. This is due to the fact that once a tag has calculated its position it has to initiate connection with the nearest beacon. In this situation, the Tag acts as the GATT Client and the Beacon acts as the GATT Server. But later on when beacons need to propagate data into the system using Trickle algorithm, if a beacon realises that a particular node is having outdated it needs to connect to that particular node and send it updated full data, which clearly means that the beacon which initiates the connection has to also act as a GATT Server.

2. **Using Advertisement Packet**

   Since it is possible to create custom Adversing structures in advertisement packet, two new advertising structure have to be defined:

   - Trickle_Metadata_type
   - Trickle_Fulldata_type

   The purpose of doing this was so that the scanning device can check the type and see if it has received metadata or fulldata. Any adversing packet would either contain an advertising structure of type Trickle_Metadata_type or Trickle_Fulldata_type. If adversing structure of Trickle_Fulldata_type is obtained, the Beacons adds it to its database if information of that particular tag is not present or an outdated information is present in the table which is recognised using timestamps.

   The only disadvantage of this approach is the it restricts the scalability of the system as if the number pf tags increase beyond a certain point it would be difficult to fit both the metadata and the full data into the adversteiment packet which can contain a maximum of 31 bytes of data.

3. **Using Scan Request and Scan response**

   This technique was required because of the limitation in the previously mentioned approach of limited size of the advertisement packet . Scan response allows additional 31 bytes of data to be sent if scan request is enabled on the scanning device. However,the mbed BLE_API, which was used for development did not support enabling or disabling scan request dynamically. Because of this, if additional data had to be sent as part of scan response, scan request had be always enabled. Problem with this approach arises in cases where the advertising data does not exceed 31 bytes and can fit in the advertising packet leading to an unnecessary overhead of sending the scan response.

   For implementation of Trickle algorithm this approach too custom advertising structures had to be defined.

3.6 Querying Location Information

A custom GATT Service called BLEProxiMeshService was created which had three characteristics. The characteristics of the service and their functions are:

1. **Update**: This characteristic is a Write only characteristic and is used by the Tag to update its position. Once the Tag calculates its position it connects to the nearest beacon and writes (tag_id, timestamp , position) to this characteristic

2. **Control**: This characteristic is a Read Only characteristic and is used either by a Tag or a Client trying to access the network in order to obtain position information of Tags in the system.

3. **Data**: Once a read request is obtained either from the Tag or Client, the position information corresponding to that Tag using this characteristic.

Chapter 4

Prototype Implementation and Evaluation

4.1 Introduction

This chapter provides a detailed description of how the prototype of the proposed system was implemented. It include description of the hardware and software requirements along with explanation of the algorithms implemented in the system which include Centroid based localisation and Trickle Algorithm

4.2 Requirements

4.2.1 Hardware

The hardware devices which were used for the implementation of the project were:

1. nRF51 Development Kit board (PCA10028)[11]

2. nRF51 Dongle-Development and test dongle (PCA10031)[12]

Both these hardware devices were chosen because of following reasons[11][12]:

1. Low-cost devices for BLE using nRF51 Series SoC .

2. Software development toolchains using Keil, IAR and GCC and support for ARM mbed toolchain. ARM mbed leads very quick development because of its cloud based

[11] https://www.nordicsemi.com/eng/Products/nRF51-DK
[12] https://www.nordicsemi.com/eng/Products/nRF51-Dongle

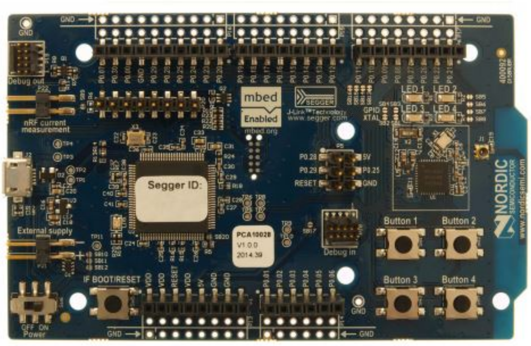

Figure 4.1: [11]

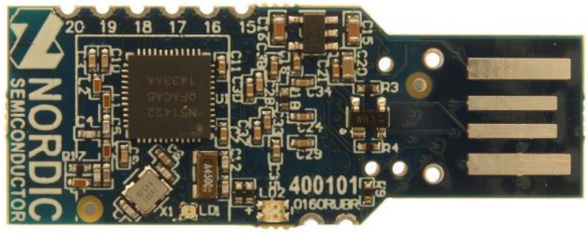

Figure 4.2: nRF51 Dongle (PCA10031)[12]

IDE, open-source software libraries and a large development community willing to support each other.

3. Debugging can very easily done using Segger Jlink tool chain.

4. Testing of the advertisements and connections of a peer device can be done using the Master Control Panel using Master Emulator firmware on the dongle.

5. nRF51 series SOC have already pre-compile and pre-linked binary files for the protocol stack referred to as softdevices.

4.2.2 Software Setup and configuration

Two development platforms that were identified for the chosen hardware along with some details are enlisted in the following section.

1. **ARM Mbed**

   ARM mbed is specially designed for building IoT based application and acts as a one stop shop for all the development needs. It provides support for development and deployment by providing all the necessary programming libraries, tools and frameworks along with C/C++ SDK, HDK and tools to support collaborative development[13].

   The online mbed IDE facilitated rapid prototype development and was very user friendly. Compilation of a project generated a hex file which had to be programmed onto the board. This programming can be simply done by dragging and dropping the hex file onto the drive corresponding to the board/dongle or by using nRFgo studio.

[13]https://developer.mbed.org/

2. **Keil μVision with Nordic SDK**

   An alternative development platform can be using Keil μVision along with Nordic SDK. The latest version of Keil μVision can be downloaded from the official Keil website[14] and the latest version of Nordic SDK can be downloaded from Nordic developers website[15]. The Keil μVision provides the development environment and the nordic SDK provides the necessary drivers, libraries, examples and APIs for the nRF51 soft devices needed for the development process. Along with this, nRFgo Studio has to be used for programming softdevices onto the board followed by which Keil μVision Pack installer can be used for importing sample projects into the workspace. These projects then have to be built and loaded onto the board.

ARM mbed's cloud based online IDE/compiler and huge development community with a lot of open source information were the two main reasons that the project implementation was carried out using ARM mbed.

When development of the project started, ARM mbed provided support for only broadcaster/ peripheral as the BLE_API used S110 softdevices. So, it was planned that peripheral devices would be programmed using ARM mbed and devices which need to be configured for scanning or BLE central role would be programmed using Nordic SDK along with S120 and S130 softdevices. But, because of change in the BLE_API Which extended support for S130 softdevices and GATT Client on this platform, it could be used for programming of central devices and devices playing dual roles too.

4.2.3 Nordic Softdevices

Nordic Semiconductor protocol stack solution referred to as softdevices are precompiled, prelinked libraries that can be programmed into the System on a Chip configuration devices[16].

The Application Programming Interface (API) is composed of C functions making the application completely exclusive of the compiler and linker. Softdevice facilitate the code to be created as ARM Cortex- M0 project allowing access to all the toolchains of ARM Cortex- M0 to be integrated with BLE application[17]. The softdevices provide flexibility

[14] http://www.keil.com/
[15] https://developer.nordicsemi.com/nRF51_SDK/
[16] https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822
[17] Nordic S110 nRF51 SoftDevice Specification v2.0

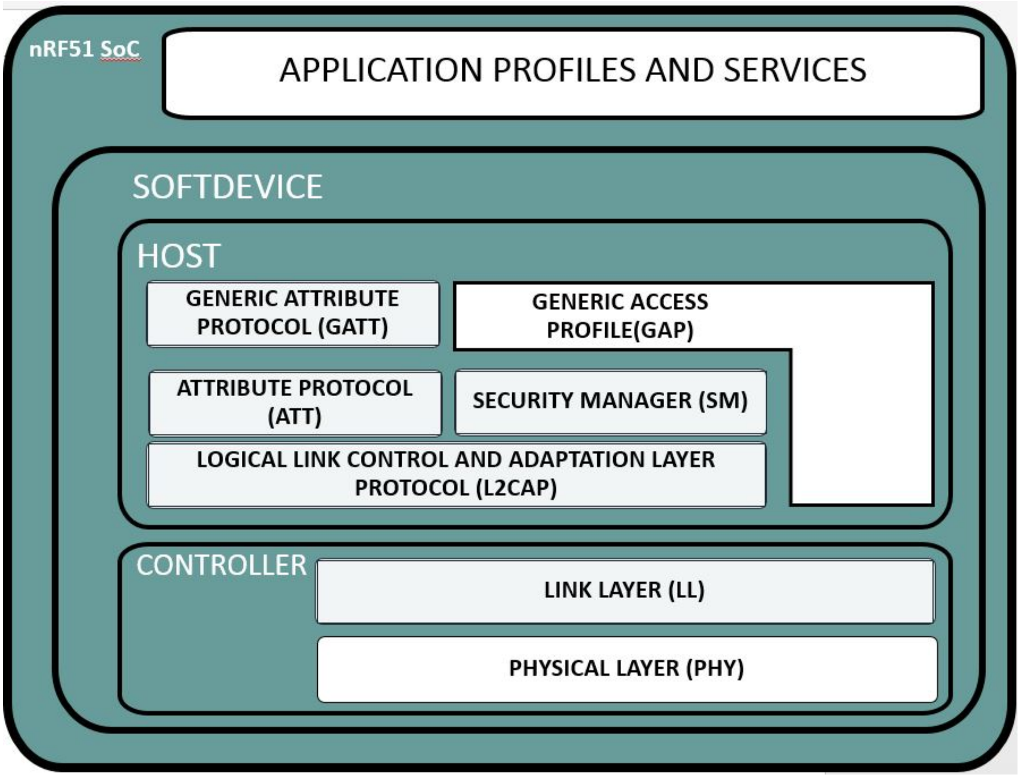

Figure 4.3: [16]

in using API for building application on a SoC configuration by integrating the controller and host of the bluetooth protocil stack[17] [18].The softdevices also have the flexibility of being programmed either during production or development phase.

Nordic provides three softdevices for Bluetooth Smart devices:

1. **S110 Softdevice**
   This is the nordic single-mode protocol stack stack solution for Bluetooth Smart devices that supports concurrent peripheral and broadcaster role[17].

2. **S120 Softdevice**

   The S120 SoftDevice is a Bluetooth Smart (previously called Bluetooth low energy) central protocol stack solution supporting up to eight simultaneous Central role connections[19].

3. **S130 Softdevice**
   S130 is a concurrent multi-link protocol stack solution which supports up to 3 central

[18] Nordic S130 nRF51 SoftDevice Specification v2.0
[19] Nordic S120 nRF51 SoftDevice Specification v2.0

and 1 peripheral simultaneous connections while also being capable of supporting simultaneous observer and broadcaster roles[18].

4.3 Device Data Structures

4.3.1 Beacon

Data structure for Beacon

Data Type	**Variable**
Address_t	Beacon_Address
int8_t	*highest_rssi
int8_t	next_rssi_index
int8_t	current_rssi
int8_t	average_rssi
int8_t	init
float	distance
int8_t	proximity
int8_t	x
int8_t	y

Table 4.1: Beacon Data Structure

The description of elements of the Structure are:

Beacon_Address: Bluetooth Address of the Beacon
***highest_rssi**: Pointer to an array (rssi_array) containing 8 most strongest RSSI values
next_rssi_index: index for the storing the next RSSI value in an array
current_rssi: The most recent RSSI value received
average_rssi: The average of all the elements of the rssi_array
init: Initialising elements of a beacon object if is detected for the first time
distance: Distance of the beacon from the Tag which is maintaining the beacon data
proximity: Proximity of the beacon from the Tag which is maintaining the beacon data. The proximity values are 1, 2, 3 for near, intermediate, far respectively
x: x coordinate of the beacon position
y: y coordinate of the beacon position

An array of the Beacon data structure is created in which number of elements of the array is same as the number of beacons. So, each element of the array referred using index is an object of the Beacon structure. The index is assigned to the beacon right at the beginning while the address of the beacons.

Beacon BeaconObj[NUMBER_OF_BEACONS]

If no advertising packets are received from a particular beacon for more than a certain time, number of elements of the array are decremented by one and hence the data corresponding to that particular beacon is deallocated resulting better memory utilization.

4.3.2 Tag

For Tag data structures defined are:

Data Type	**Variable**
uint8_t	tag_id
uint8_t	version
int8_t	x
int8_t	y
uint8_t	c
uint8_t	t
TagTable	*next

Table 4.2: TagTable Data Structure

Data Type	**Variable**
uint8_t	tag_id
uint8_t	version
MetaData	*next

Table 4.3: Metadata Structure

4.4 Proximity Sensing and Location Estimation

4.4.1 Proximity Sensing

Data Structures that were used were:

1. **rssi_mat**[ ][ ]: 2D matix containing 16 most recent rssi values of all the beacons. It is implemented as a circular buffer
   Number of rows=Number of beacons, Number of columns=16
2. **sort_rssi_array(int beacon_index)**: This function is used to sort one of the rows in the 2D matrix rssi_mat which is specified by the input parameter beacon_index.

This function sorts the specified row of the matrix, stores the values in an array and returns a pointer to the sorted array. Sorting of the RSSI values is done to make sure that the strongest signals received are accounted for the distance calculation and also to avoid excessive fluctuation of RSSI values by sorting the 16 most recent RSSI values obtained from that beacon and then taking an average of the strongest 8 RSSI values to obtain a value which is ultimately used for obtaining distance .

The flowchart below explains how RSSI value obtained from a beacon during scanning is used to populate rssi_mat which is used for sorting

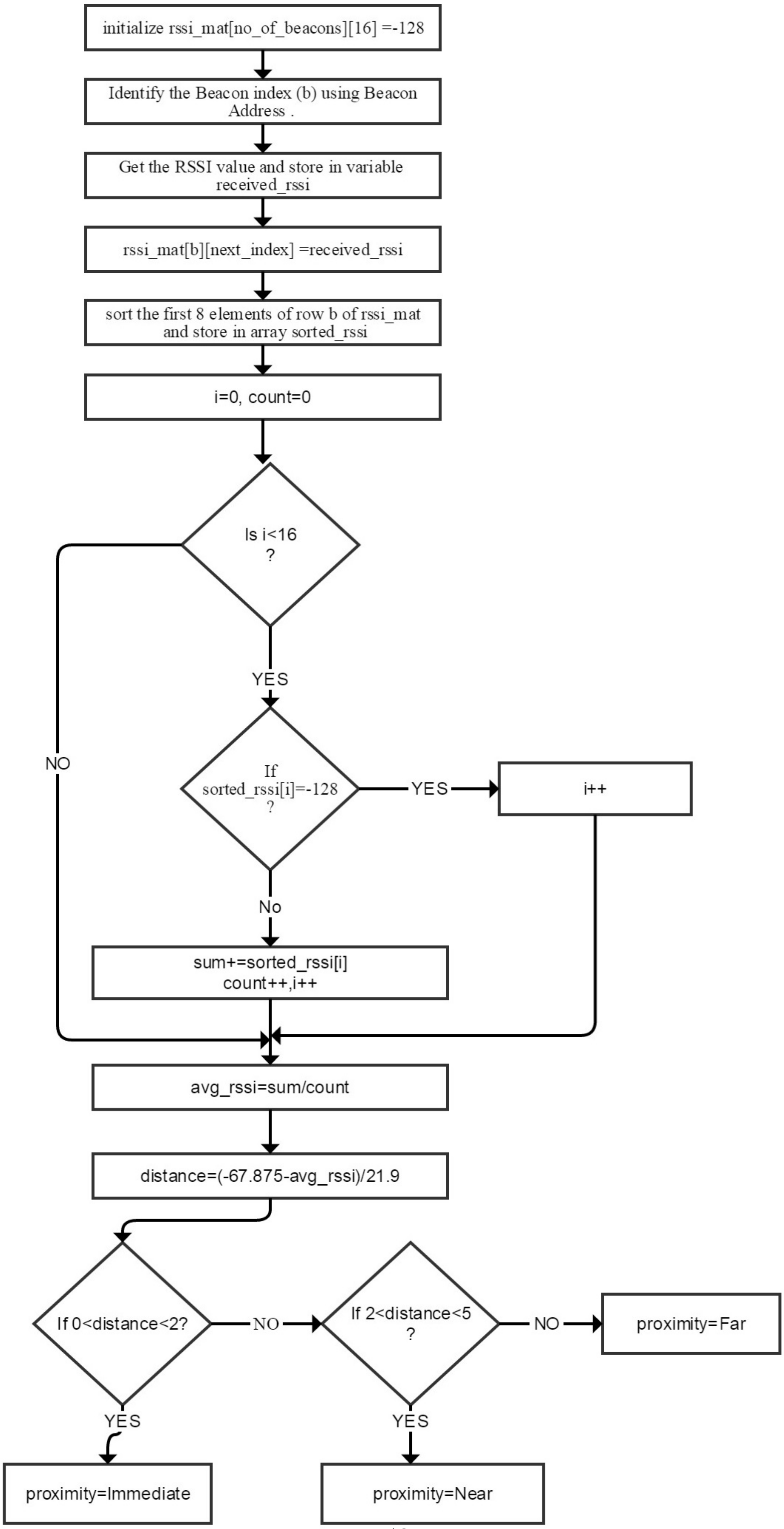

Figure 4.4: RSSI Smoothing and Proximity Sensing

4.4.2 Centroid Based positioning

The centroid based algorithm is applied on the 4 nearest beacons by taking iteratively taking 3 beacons out of the nearest 4 beacons. For every beacon, the distance estimation is done using RSSI value. Since, the Tag lies on locus of points at the calculated distance from beacon position. This locus is a circle where the centre is the beacon position and radius is the distance of the beacon from the tag (estimated using RSSI value). Since, there are three beacons being used at one time it means that the position of the tag is identified by intersection of three such circles, centres of which correspond to the beacon position. In order to calculate the point of intersection of three circles, the approach adopted in [2] first finds the line of intersection of the three circles followed by taking intersection of pairs of lines and finally takes the average of all the points of intersection of the lines obtained generating (x_1, y_1). The above process is repeated till all the possible combination of triples are used and (x_2, y_2),(x_3, y_3),(x_4, y_4) are obtained and finally an average of the points is taken to obtain the final position of the tag. The data structures used for this purpose are:

1. **distance_based_sorted_beacons[NUMBER_OF_BEACONS]**: This array stores the beacon index (index of the Beacon Structure array) in the ascending order of their distance from the tag. So, if beacon identified with index 3 is located closest to the tag then, distance_based_sorted_beacons[0] would be 3.

2. **distance_array[NUMBER_OF_BEACONS]**: This array is used to maintain the distances of beacons based on their beacon indices. e.g. if a beacon with beacon index 2 has distance 23 from the tag then, distance_array[2]=23.

3. **xintercept[4][3]**:A 2D array containing x intercept of all the lines which are formed as intersection of a pair of circles out of the three circles formed in that particular triple beacons chosen out of 4 nearest beacons. xintercept[i][j] stands for x intercept calculated for the ith triple and jth line.

4. **yintercept[4][3]**: A 2D array containing y intercept of all the lines which are formed as intersection of a pair of circles out of the three circles formed in that particular triple beacons chosen out of 4 nearest beacons. yintercept[i][j] stands for y intercept calculated for the ith triple and jth line.

5. **constant[4][3]**: A 2D array containing constant of all the lines which are formed as intersection of a pair of circles out of the three circles formed in that particular triple beacons chosen out of 4 nearest beacons. constant[i][j] stands for y intercept calculated for the ith triple and jth line.

6. **xintersect**[4][**3**]: This 2D array was used to store x coordinate of the point of intersection of each pair of lines using the values of intercepts and constants.

7. **yintersect**[4][**3**]: This 2D array was used to store y coordinate of the point of intersection of each pair of lines using the values of intercepts and constants.

8. **xavg**[**4**]: This array was used to find x coordinate value for that particular triple of beacons. e.g. xavg[1] represents the x coordinate value obtained for the first triple of beacons.

9. **yavg**[**4**]: This array was used to find y coordinate value for that particular triple of beacons. e.g. yavg[1] represents the y coordinate value obtained for the first triple of beacons.

   Finally, tag position coordinates **(x_centroid,y_centroid)** are obtained by finding the average of values in the array xavg and yavg respectively.

4.4.3 Trickle Algorithm Implementation

Section 4.3 describes the TagTable and MetaData data structures which are used in the system.

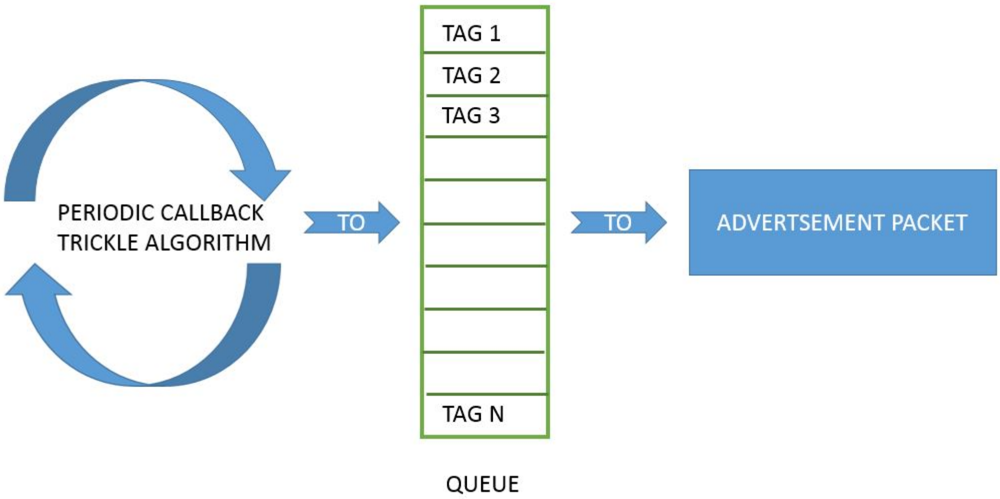

Figure 4.5: Implementation of Trickle

As described in the image above, trickle algorithm consists of three parts:

1. **Periodic Callback**
   The periodic Callback is responsible for checking periodically if c_{tag} <K. If this

condition is sattisfied it places that tag id and version number into the Metadata Data structure.

2. **Updated Tag information received from Tag**
   Any beacon receiving Tag id,version and position from the Tag places it directly to the TagTable DataStructure

3. **Advertisement CallBack**
   This Callback is responsible for periodically accessing data from Queues. It dynamically decides whether data has to be taken from TagTable or from MetaData and subsequently places data into advertisement packet.

Each of the above mentioned components are described by the flowcharts in the following page.

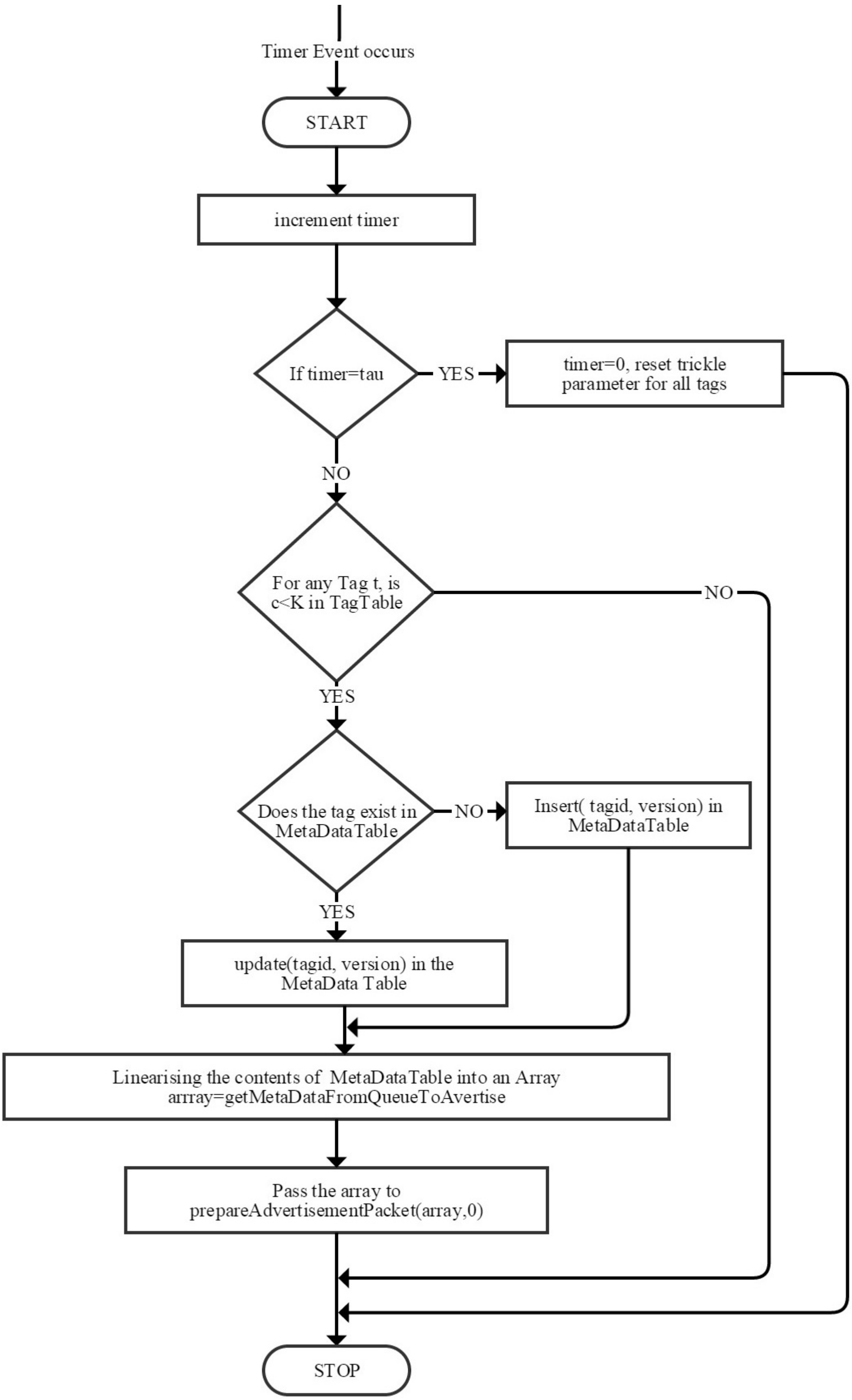

Figure 4.6: Periodic Callback

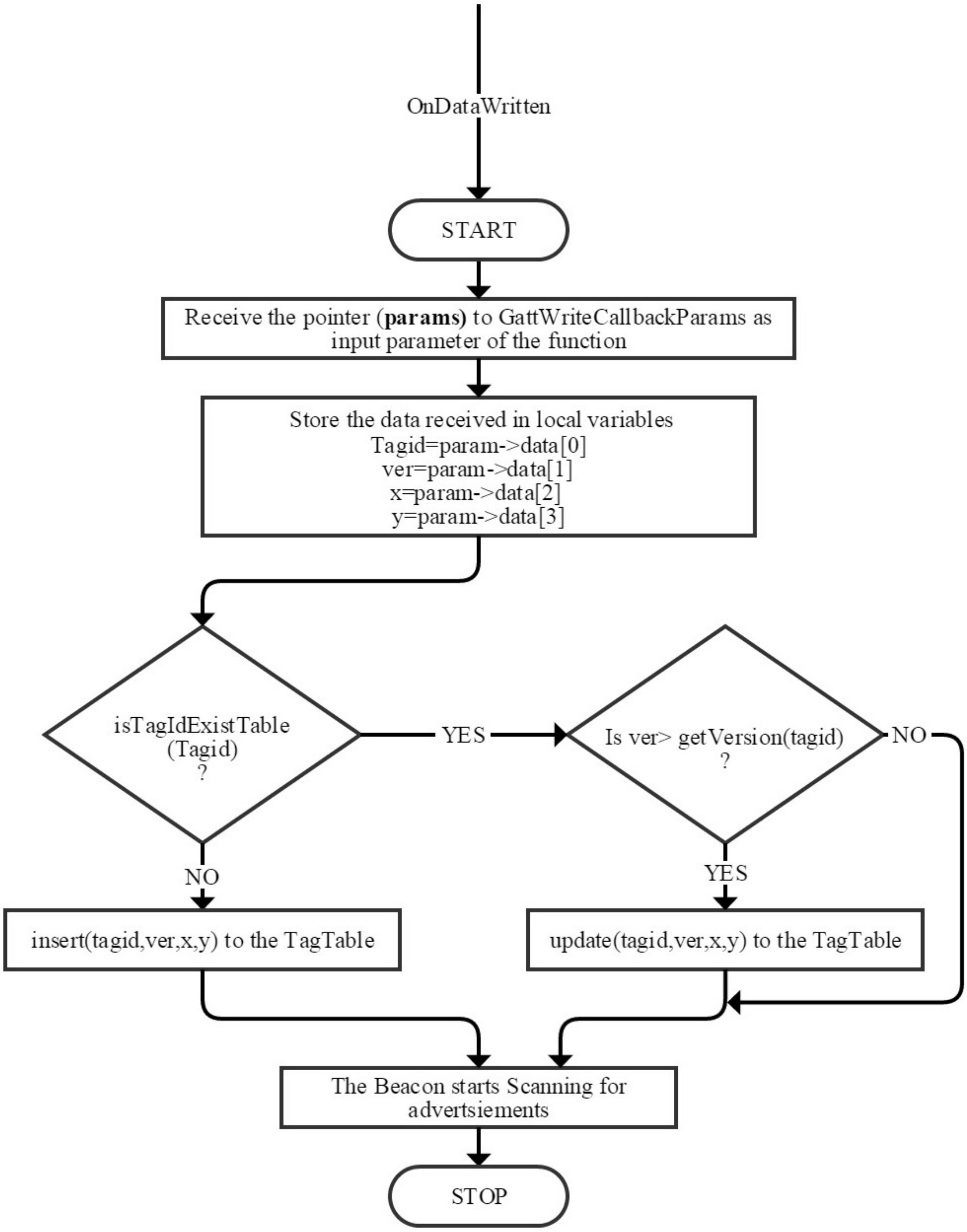

Figure 4.7: Characteristic Value Received

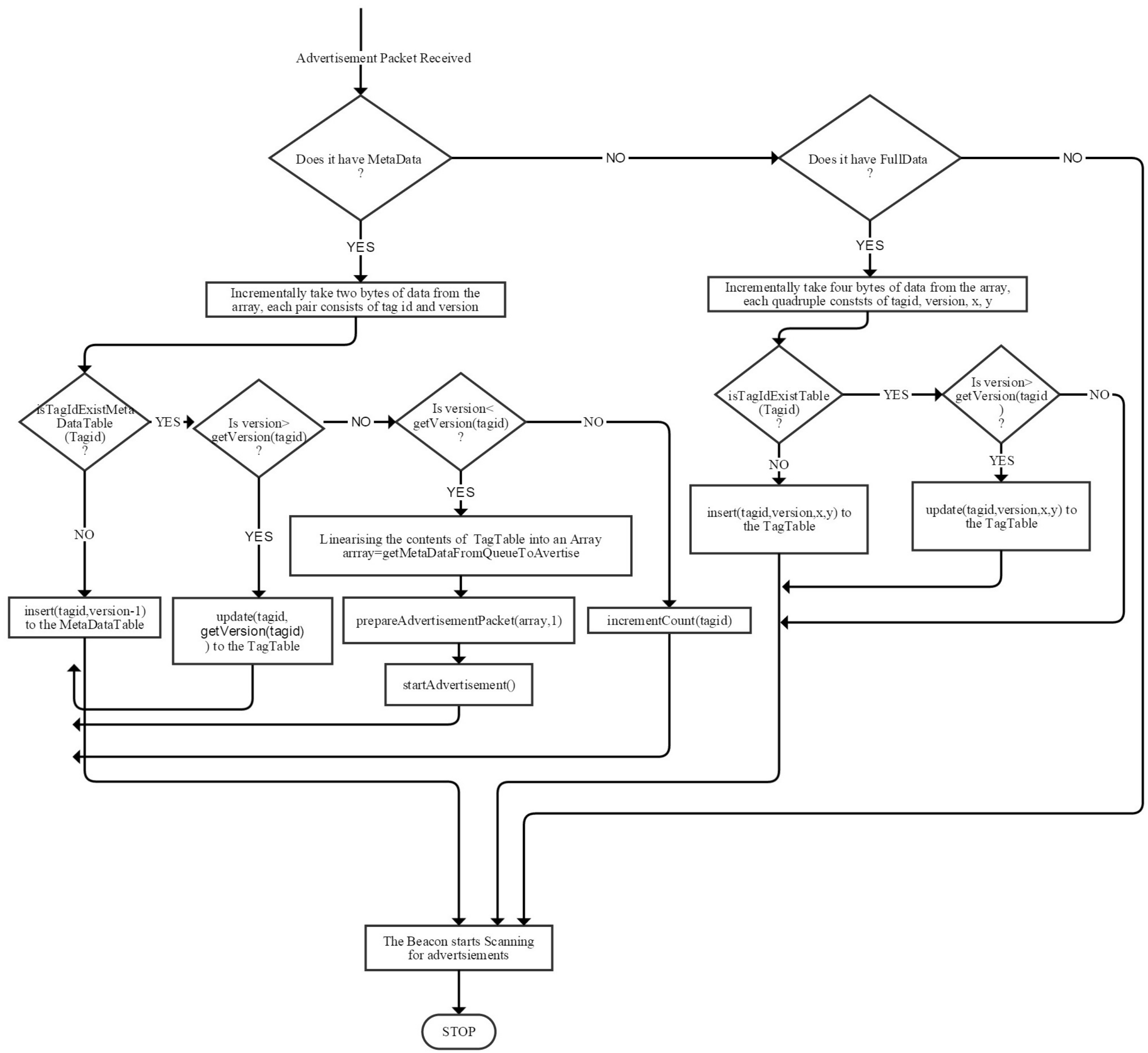

Figure 4.8: Advertisement Callback

Functions implemented for the Trickle Algorithm implementation are:

int8_t getVersion(uint8_t tag_id): This function takes tag_id as input and returns the corresponding version number stored in the TagTable. It returns -1 if the tag_id does not exist.

int8_t getCount(uint8_t tag_id): This function takes tag_id as input and returns the corresponding counter (c) stored in the TagTable. It returns -1 if the tag_id does not exist.

int8_t getTimer(uint8_t tag_id): This function takes tag_id as input and returns the corresponding timer (t) value stored in the TagTable. It returns -1 if the tag_id does not exist.

void incrementCount(uint8_t tag_id): This function takes tag_id as input and increments the counter for that tag_id in the TagTable.

void IncrementTimer(uint8_t tag_id): This function takes tag_id as input and increments the timer for that tag_id in the TagTable.

bool isTagIdExistTable(uint8_t tag_id): This function takes tag_id as input and returns true if the tag_id exists in the TagTable otherwise it returns false.

bool isTagIdExistMetaData(uint8_t tag_id): This function takes tag_id as input and returns true if the tag_id exists in the MetaDataTable otherwise it returns false.

bool isTableEmpty(): This function returns true if the TagTable is empty otherwise it returns false.

bool isMetaDataEmpty(): This function returns true if the MetaDataTable is empty otherwise it returns false.

void displayMetaData(): This function is used to display the contents of the MetaDataTable.

void displayTable(): This function is used to display the contents of all the tags in the TagTable.

void insert(uint8_t id,uint8_t ver): This function is used to insert a node containing tag_id=id and version=ver in the MetaDataTable.

void update(uint8_t id,uint8_t ver): This function is used to update the version of a node with tag_id=id to version=ver in the MetaDataTable.

void insert(uint8_t id,uint8_t ver,int xr,int yr): This function is used to insert a node containing tag_id=id, version=ver and position (x_r, y_r) in the TagTable.

void update(uint8_t id,uint8_t ver,int xr,int yr): This function is used to update the version and position of a node with tag_id=id with values of version=ver and position (x_r, y_r) in the TagTable.

uint8_t* getMetaDataFromQueueToAdvertise(): This function is used to place the data from the MetaDataTable to an array which is placed in advertisement packet.

uint8_t* getPositionDataFromQueueToAdvertise(): This function is used to place the data from the TagTable to an array which is placed in advertisement packet.

4.5 Evaluation

4.5.1 Verifying Proximity Detection

Objective

To test and verify if a Tag is able to correctly detect proximity value depending on its distance from beacons located at different distances.

Methodology

An experiment was conducted by placing five beacons at a distance of 20 cm, 50 cm, 1 m, 2 m and 4 m and the proximity value calculated by the Tag was printed to the terminal and noted.

Result

The results are tabulated in the table below:

No.	Distance	Expected Result	Obtained Result	Result
1	0.2	Immediate	Immediate	Passed
2	0.5	Immediate	Immediate	Passed
3	1	Near	Near	Passed
4	2	Near	Near	Passed
5	4	Far	Far	Passed

Table 4.4: Experiment to Validate Proximity Detection

Discussion

The results clearly indicate that the system is able to perform proximity detection accurately.

4.5.2 Verifying Localisation

Objective

To determine if Centroid based position algorithm is able to accurately calculate position of Tags in the system

Methodology

The 2.5 m X 5.5 m room was used to perform an experiment to see the working of Centroid based Algorithm. Considering the extreme left corner of the room as (0,0), four Beacons were places at the position of (1,1), (2,1),(1,5) and (2,5) and all the beacon were programmed to advertise their respective position as part of the Manufacture specific Advertising structure of their Advertising packet. A Tag scans for advertising packets and estimate its position based on the position of the beacons present in the packet received and the corresponding RSSI value received for that particular advertisement.

The Tag was placed at different positions and the estimated position was compared to its actual position:

Result

Discussion

The results show that the estimated postion using centroid based algorithm are good. In most of the cases an accuracy within a range of 1-2 meters is obtained but some cases

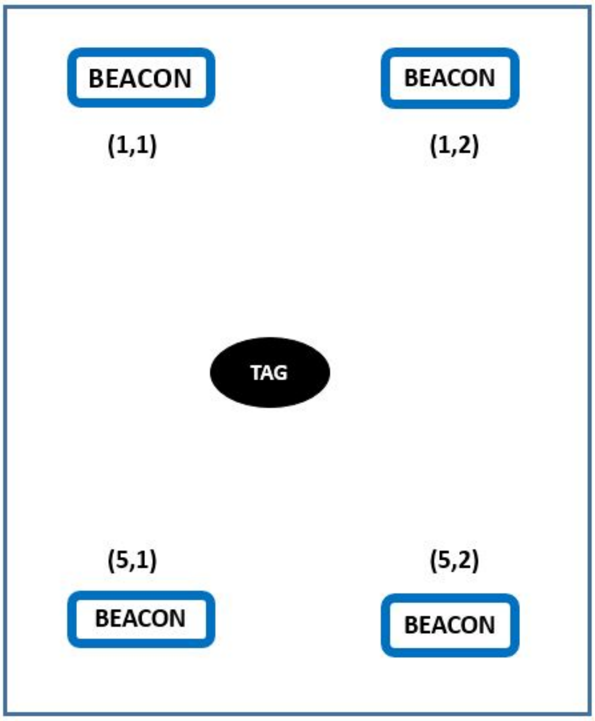

Figure 4.9: Verifying Localisation

don't produce very accurate. The inaccurate result are attributed to obstruction of the signal caused by furniture in the room. Therefore, inaccurate results are obtained in some cases like 4. and 5. whereas relatively accurate results are obtained in rest of the cases.

4.5.3 Trickle Operation

Objective

The objective of this experiment was to test if position data of Tags is replicated over all the Beacons in the network using Trickle Algorithm implemented. It is also important to verify if an update in the Tag position updates all the beacons in the network.

Methodology

Four beacons were arranged in a linear manner to ensure that the first and the forth beacon were not within transmission range. A Tag was introduced into the system and it was checked if the estimated position of the tag is propagated throughout the network by checking if the last beacon is able to receive Tag position. Android Application called Master Control Panel was used to check the Metadata advertised by the Beacons and serial Terminal was used to check the position data propagated to the Beacon.

No.	Tag Actual Position	Estimated Position
1	(1,1.5)	(0.875,2.125)
2	(3,1.5)	(2.175,2.175)
3	(5,1.5)	(4.775,1.125)
4	(3,1)	(0,1.5)
5	(3,2)	(1.275,1.5)
6	(3,2)	(0,1.225)
7	(2,2)	(1.575,1.175)

Table 4.5: Experiment to verify Centroid based Localisation

Followed by this, Tag was moved to another location which led to an update in the position and then it was checked if the last beacon was updated or not.

Result

It was observed using the Master Control Panel application that Position Data was replicated on all the Beacons of the network. All the beacons were adversing the Tag id and the latest Timestamp. Changing the position of the Tag incremented the Timestamp which too after some time updated the Beacons. Using Serial Terminal, the last node which was out of the transmission range was checked and it was observed that it had the updated Tag position.

Discussion

Trickle algorithm was successfully implemented over GAP and GATT of Bluetooth Low Energy. As explained above, the approach was verified using a small Beacon network. Future work would be done to test and evaluate the system on a large scale using Bluetooth Low Energy Simulators.

Chapter 5

Conclusion and Future Work

5.1 Introduction

This is the final chapter that concludes the dissertation by highlighting the features of the developed prototype, the challenges encountered during the project and the future work that can be done.

5.2 The Prototype

A working prototype of a small mesh network of Beacons using Trickle Algorithm was successfully implemented. The novel approach of implementing Trickle over GAP and GATT of Bluetooth Low Energy protocol stack was validated through an experiment and it was concluded that the approach is able to successfully allow information replication throughout a network consisting of Bluetooth Smart devices.

5.3 Future Work

5.3.1 Large Scale Evaluation through BLE Simulator

Even though the approach of Trickle over GAP and GATT was tested on a small network, it is imperative to verify the approach on a large scale using Bluetooth Low Energy Simulators. Because BLE is a relatively new technology, there are only a few simulators available for testing. Mikhaylov [25] presents a simulation tool that enables to simulate Bluetooth Low Energy Network which facilitates more quantitative research on BLE networks. The results derived from the simulator are compared to those obtained by actual BLE hardware devices and it is concluded that the simulator provides quiet realistic results and hence is suitable for simulating large scale networks comprising of BLE devices.

It would be part of the future work to identify more such simulators and select the most suitable one for large scale testing of the system.

5.3.2 Other Indoor Positioning Techniques

The main focus of the project was to develop a system level architecture that facilitated discovery, tracking and provided an approximate estimate of the position of people in a building. Position estimation using Centroid based localisation was done as it was relatively easy to understand, implement and hence, enabled to have the feature of location estimation incorporated into the system at a very early stage providing enough time to focus on implementation of Trickle Algorithm using Mesh. However, it can't be denied that more accurate indoor position estimation would definitely add more value to the project. Hence, more accurate indoor positioning techniques would be definitely explored as part of the future work.

5.3.3 Measuring the Propagation Delay

A very important parameter that needs to be evaluated to quantify the performance of the system is data propagation time i.e. the time required to propagate data throughout the network. This is a very critical parameter that would allow to assess the system performance. The aim would be to reduce the propagation delay in the system in order to promote data replication at a faster rate.

5.3.4 Scalability of the System

The scale of the current prototype is limited because Advertising packets are used for broadcasting Metadata containing Tag id and timestamp and incase it is identified that certain nodes need to be updated, Full_data is also broadcasted in the adverstising packet. This puts a very big limitation on the number of Tags in the system. However, this approach of using the advertising packet was only used because of limitation of the API on the development platform. Initial aim was to use GATT characteristics for implementation of Trickle Algorithm. This limitation can be overcome if the problem in the API is fixed in the future or by using another platform like Keil with Nordic SDK for development.

5.3.5 Degree of Replication

For reducing the energy consumed in the system and for improving the overall network performance, Trickle algorithm can be implemented such that rather than replicating data on every beacon, data is replicated selectively across several beacons. This would not only reduce the propagation delay but also help to reduce the system traffic. However, a very low degree of replication can have an adverse impact on the access network.

Therefore, it is very important to determine an optimal degree of replication which primarily depends on parameters like the network density , number of beacons in the network, frequency of data updates, and frequency of data access.

5.3.6 Security

One of the features of the proposed system is that it allows a client to query position of any other Tag in the system. This have many dangerous implications when viewed from a security point of view. This means that any person who has a Bluetooth Smart device has the ability to locate a specific person (using their Tag id) in the building easily. Hence, making it very easy for an attacker to execute their malicious plans. It is therefore, extremely important to have a layer of security in the system to ensure that it is not misused.

The developed prototype of the system does not take security into consideration but security using confidentiality, Integrity and Authentication or some form of encryption is definitely worth implementing in the near future.

5.4 Challenges

The challenges faced during the implementation of the projects were:

1. The use of cloud based compiler of ARM mbed became a problem because many a times the build server became non functional.

2. Testing and Debugging the hardware for proper functioning was very time consuming due too lack of support of a debugger in ARM mbed compiler and hence there was a lot of dependence on the Serial Terminal for displaying messges trying to identify the flow of the system.

3. Due to the use of different BLE_API for GATT Client and GATT Server it was not possible to configure the Beacons to acquire functionality of both GATT Client

and GATT Server. Though this was a limitation of the development platform, this approach without any doubt could have been used for the implementation of Trickle algorithm.

4. Also, the development platform did not support dynamically switching between active and passive scanning and hence the approach of using Scan request and Scan response for the purpose of Trickle Algorithm could not be done. However, keeping the device in the active scanning mode right from the beginning allowed to add additional 31 bytes of data which made the approach completely analogous to the use of advertising data but with additional 31 bytes of data.

5.5 Final Remark

The aim of creating a Bluetooth Low Energy mesh network was achieved by using the innovative approach of implementing Trickle Algorithm over GATT and GAP. To the best of the writer's knowledge, this is the first time such an approach had been adopted for creation of a mesh network over BLE.

Implementation of Mesh over BLE along with location based service has the capability of bridging the physical and the digital world, opening possibilities to a wide range of applications ranging from BLE Tag based door sensing, entry and navigation using Tags in large exhibitions, object location and organisation in large warehouses using Bluetooth Smart Tags.

This approach therefore, takes another step towards the future of the internet which is envisaged as the Internet of Things.

Appendix A

A.1 Abbreviations and Acronymns

WSN	Wireless Sensor Network
BLE	Bluetooth Low Energy
SIG	Special Interest Group
GSM	Global System for Mobile Communications
ISM	industrial, scientific and medical
GAP	Generic Access Profile
GATT	Generic Attribute Profile
L2CAP	Logical Link Control and Adaptation Protocol
ATT	Attribute Protocol
SMP	Security Manager Protocol
HCI	Host Controller Interface
GFSK	Gaussian Frequency Shift Keying
CCM	Cipher Block Chaining Message Authentication Code
AES	Advanced Encryption Standard
AODV	Ad hoc On Demand Distance Vector
GSP	Gossip-based Sleep Protocol
MTP	Media Transfer Protocol
TOA	Time of Arrival
TDOA	Tme Difference of Arrival
AOA	Angle of Arrival
RSSI	Received Signal Strength Indicator
IoT	Internet of Things
OS	Operating System
SDK	Software Development Kit
HDK	Hardware Development Kit
API	Application Programming Interface

A.2 BLEProxiMeshService

<table>
<tr><th>Overview</th><th>Properties</th><th>Security</th><th>Descrip tors</th></tr>
<tr><td>Name: Position Update
Description: This characteristic is used to send the position calculated by Tag to the nearest Beacon.
Type: org.bluetooth.characteristic.position_update
Requirement: Mandatory</td><td><table>
<tr><th>Property</th><th>Requireme nt</th></tr>
<tr><td>Read</td><td>Excluded</td></tr>
<tr><td>Write</td><td>Mandatory</td></tr>
<tr><td>WriteWit houtResp onse</td><td>Excluded</td></tr>
<tr><td>SignedWri te</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Excluded</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>WritableA uxiliaries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>Extended Propertie s</td><td></td></tr>
</table></td><td>None</td><td>None</td></tr>
<tr><td>Name: Control
Description: This characteristic is used by a tag to query the position of another tag.
Type: org.bluetooth.characteristic.control
Requirement: Mandatory</td><td><table>
<tr><th>Property</th><th>Requireme nt</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Excluded</td></tr>
<tr><td>WriteWit houtResp onse</td><td>Excluded</td></tr>
<tr><td>SignedWri te</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Excluded</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>WritableA uxiliaries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>Extended Propertie s</td><td></td></tr>
</table></td><td>None</td><td>None</td></tr>
</table>

<table>
<tr>
<td>Name: Data
Description: This characteristic is used to send the position corresponding to the Tag id the position of which is requested by another Tag.
Type: org.bluetooth.characteristic.data
Requirement: Mandatory</td>
<td>
<table>
<tr><th>Property</th><th>Requirement</th></tr>
<tr><td>Read</td><td>Excluded</td></tr>
<tr><td>Write</td><td>Mandatory</td></tr>
<tr><td>WriteWithoutResponse</td><td>Excluded</td></tr>
<tr><td>SignedWrite</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Mandatory</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>WritableAuxiliaries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>ExtendedProperties</td><td></td></tr>
</table>
</td>
<td>None</td>
<td>None</td>
</tr>
</table>

Figure A.1: BLEProxiMeshService:Custom GATT Service.

Bibliography

[1] K. Townsend, C. Cufí, and R. Davidson, *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking.* O'Reilly, 2014.

[2] H. Shi, "A new weighted centroid localization algorithm based on RSSI," in *Information and Automation (ICIA), 2012 International Conference on*, pp. 137–141, June 2012.

[3] H. Chen, P. Huang, H. C. So, and K. Sezaki, "Mobility-assisted position estimation in wireless sensor networks," in *14th IEEE International Conference on Parallel and Distributed Systems, 2008. ICPADS '08.*, pp. 607–614, Dec 2008.

[4] K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: Challenges and opportunities," *IEEE Pervasive Computing*, vol. 3, pp. 16–23, Oct. 2004.

[5] Z. C. Z. L. JianYong Zhu, Luo Haiyong, "RSSI based bluetooth low energy indoor positioning," in *2014 International Conference on Indoor Positioning and Indoor Navigation*, (Busan, Korea), 2014.

[6] D. Schwarz, M. Schwarz, J. Stückler, and S. Behnke, "Cosero, find my keys! object localization and retrieval using bluetooth low energy tags," in *RoboCup 2014: Robot World Cup XVIII [papers from the 18th Annual RoboCup International Symposium, João Pessoa, Brazil, July 15*, pp. 195–206, 2014.

[7] "Internet of things:wireless sensor networks," tech. rep., IEC.

[8] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless sensor networks and the internet of things: Do we need a complete integration?," in *1st International Workshop on the Security of the Internet of Things (SecIoT'10)*, (Tokyo ,Japan), IEEE, December 2010.

[9] D. Christin, A. Reinhardt, P. S. Mogre, and R. Steinmetz, "Wireless sensor networks and the internet of things: Selected challenges."

[10] K. Mikhaylov and J. Tervonen, "Multihop data transfer service for bluetooth low energy," in *ITS Telecommunications (ITST), 2013 13th International Conference on*, pp. 319–324, Nov 2013.

[11] M. Haatanen, "Self-organizing routing protocol for bluetooth low energy sensor networks," *Information Technology and Telecommunications*, 2012.

[12] B. Lee, H. K. Song, Y. Suh, K. H. Oh, and H. Y. Youn, "Energy-efficient gossiping protocol of wsn with realtime streaming data," in *Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on*, pp. 219–224, Aug 2014.

[13] X. Hou and D. Tipper, "Gossip-based sleep protocol (gsp) for energy efficient routing in wireless ad hoc networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 3, pp. 1305–1310 Vol.3, March 2004.

[14] C. Petrioli and S. Basagni, "Degree-constrained multihop scatternet formation for bluetooth networks," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 1, pp. 222–226 vol.1, Nov 2002.

[15] Z. Wang, R. Thomas, and Z. Haas, "Bluenet - a new scatternet formation scheme," in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pp. 9 pp.–, Jan 2002.

[16] E. Dahlgren and M. Hasan, "Evaluation of indoor positioning based on bluetooth smart technology," Master's thesis, Department of Computer Science and Engineering (Chalmers), Chalmers University of Technology, Gothenburg, 2014.

[17] J. P. Carles Gomez, Joaquim Oller, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, p. 11734, 2012.

[18] M. Siekkinen, M. Hiienkari, J. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pp. 232–237, April 2012.

[19] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does btle measure up against wifi? a comparison of indoor location performance," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pp. 1–6, May 2014.

[20] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, vol. 1, pp. 350–355 vol.1, Mar 2002.

[21] D. C. S. S. Philip Levis, Neil Patel, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, (Berkeley, CA, USA), pp. 2–2, USENIX Association, 2004.

[22] J. H. O. G. J. K. P. Levis, T. Clausen, "The trickle algorithm," tech. rep., 3 2011. RFC 6206.

[23] H. Kermajani, C. Gomez, and M. Arshad, "Modeling the message count of the trickle algorithm in a steady-state, static wireless sensor network," *Communications Letters, IEEE*, vol. 16, pp. 1960–1963, December 2012.

[24] B. Djamaa and M. Richardson, "Optimizing the trickle algorithm," *Communications Letters, IEEE*, vol. 19, pp. 819–822, May 2015.

[25] K. Mikhaylov, "Simulation of network-level performance for bluetooth low energy," in *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*, pp. 1259–1263, Sept 2014.