https://doi.org/10.1016/j.comcom.2020.04.041

Decentralized Adaptive Indoor Positioning Protocol Using Bluetooth Low Energy

Yik Him Ho and Henry C. B. Chan

Department of Computing, The Hong Kong Polytechnic University

Abstract

Previous indoor positioning research has mainly been focused on using Wi-Fi and RFID. In recent years, researchers began to study using Bluetooth 4.0 and Bluetooth Low Energy (BLE) for indoor positioning purposes. In general, positioning techniques based on received signal strength indicator (RSSI), such as signal propagation and fingerprint, is commonly used in wireless/mobile networks. These techniques have certain limitations and tradeoff in terms of accuracy, ease of implementation and practical application/deployment. For example, both methods require a training process before deployment. In this paper, we present a decentralized BLE-based positioning protocol that does not require training before deployment. The training process can automatically be done on-the-fly by the anchor nodes. While the anchor nodes are broadcasting, they also scan for signals emitted by other anchors. This collaborative communication process exchanges location information and signal strength measurements between each anchor. This process builds a signal-to-distance reference list for the target node to estimate physical distance in a more accurate way. Experimentation in a real indoor environment shows that the proposed collaborative positioning method can achieve an error of 1.5 meters on average. This is generally applicable for most indoor positioning applications for locating people. Furthermore, its implementation is simple and practical, because it does not require training before positioning estimation and is adaptive to environmental changes.

Keywords: Bluetooth Low Energy, indoor positioning

1. Introduction

GPS (Global Positioning System) is currently the most commonly used localization and positioning technology, for games, entertainment, transportation, logistics, emergencies, etc. As GPS relies on the use of satellites, it cannot be applied effectively in an indoor environment because the satellite signals are blocked by walls and other obstacles. Therefore, there is a need to study indoor positioning systems.

Research on indoor positioning technologies has been conducted for more than two decades. Most focus on Wi-Fi and RFID. However, compared with GPS, it seems that a robust implementation in the consumer market (e.g., indoor navigation by end-users using smartphones) is still uncommon. One of the reasons is related to the difficulty in deploying Wi-Fi- and RFID-based positioning methods directly to smartphones because of various limitations, such as hardware modification, to smartphones. This results in a gap between research and application.

Until a few years ago, with the introduction of Bluetooth 4.0 specification and Bluetooth Low Energy (BLE), there was a growing interest in studying BLE for positioning purposes. In fact, the BLE-based positioning technique is a promising technology for positioning service because of its benefits of lower cost, ease of deployment, availability in mobile phones, etc.

1.1. Positioning approach

In general, to determine the position of a node, distance calculation must be performed first. Common to all wireless technologies, there are basically three major approaches to estimate the distance between two nodes, namely time-based, channel state information (CSI)-based (CSI is also referred to as phase information) and received signal strength indicator (RSSI)-based. The time-based method relies on using the time-of-flight (i.e., signal traveling time) to estimate the distance. To do this accurately, precise and synchronized clocks are required. The phase-based method has attracted a great deal of attention in recent years. Although this method can achieve high accuracy rates (the distance error can be reduced to the centimeter range), it is not easy, cost-effective or practical to implement in today's smartphones. RSSI-based methods make use of signal strength for distance estimation, as signal strength has a negative correlation with distance in general. The RSSI-based method is currently the most commonly used method for Wi-Fi, RFID or Bluetooth due to its ease of implementation.

Research on indoor positioning using different wireless technologies has a history of nearly two decades. [1] is one of the representative works in this area. The paper categorizes indoor positioning methods into two approaches, which are later referred to as signal propagation approach and fingerprinting approach. Generally speaking, the objectives of an indoor positioning system are to (1) determine the position of an object accurately e.g., a person, a smartphone, an Internet of Things (IoT) sensor, etc., and (2) develop a system that has high scalability and is easy to implement. In the past, most research typically focused on the first objective, while the second one was sometimes ignored. This is also one of the reasons why there are not many popular indoor positioning systems in the consumer market.

As mentioned, there are two key approaches for indoor positioning, namely, signal propagation and fingerprinting. The signal propagation method, which is sometimes referred to as simple trilateration, seeks to convert signal strength measurement into physical distance based on a model, such as using Log-distance path loss model in Equation 1 [2] [3].

$$RSSI@d = RSSI@d_r - 10n \log \frac{d}{d_r} + X \tag{1}$$

In theory, signal strength and distance have a negative correlation. If the correlation relationship is known, distance can be determined based on signal strength. However, this assumption is only valid in an ideal environment (e.g., vacuum space) without any interference. In a real-world environment, especially for an indoor environment, the relationship depends heavily on different environmental factors. The solution to this problem is to use a training phase to estimate the environment-based parameters for determining the relationship or model. After calculating an estimated distance, one can then estimate the location accordingly. In general, the estimation procedure is often referred to as trilateration. Compared to fingerprinting, the advantage of the propagation-based technique is that it requires less training. Hence it is more scalable, although the accuracy is lower than that of fingerprinting. By means of a training phase, fingerprinting seeks to create a signal database at different points-of-interest (PoI) in the indoor environment, so that position estimation can be conducted by finding the database record with the highest similarity. In general, the fingerprinting method can achieve higher accuracy at the expense of a time-consuming training phase (i.e., to generate the signal database).

The aforementioned positioning methods require a training phase in order to perform position estimation in an indoor environment. However, this is a time- and effort-consuming process. Furthermore, these methods have low scalability, as periodic re-training is often required. To address this issue, this paper proposes a decentralized, adaptive BLE-based positioning protocol for indoor positioning estimation. Although it can also be classified as a signal propagation method, it does not require prior manual training of the parameters. In particular, training will be done on-the-fly automatically and it can self-adjust to environmental changes over time.

1.2. Major contributions

The main advantages and contributions of this work are as follows:

- The proposed method does not require a manual training stage before deployment, while at the same time it can achieve comparable accuracy.
- A centralized medium for storage and processing is not required.
- The implementation is practical and can be deployed to smartphones relatively easily. No modification to current smartphones is required.
- Although modification to the beacons (i.e., reference nodes) is needed, we found that it can easily be achieved based on our testing. It also provides insights to manufacturers to design better beacon hardware to achieve higher positioning accuracy.

The rest of the paper is organized as follows. Section 2 explains the advantage of using BLE over other wireless technologies. Section 3 summarizes selected recent BLE-based related work. Section 4 explains the challenges in indoor positioning. Section 5 introduces our proposed decentralized BLE positioning protocol. Section 6 and explains the adaptability of the signal propagation model to improve distance estimation and the positioning algorithm respectively. Section 8 presents the experimentation setup and results. We also discuss the effect on positioning accuracy by various factors. Section 9 presents the conclusion.

2. Comparison of BLE positioning with other positioning technologies

In this section, we first give an overview of various positioning technologies using Wi-Fi, RFID, 4G/5G, and BLE. The aim is to compare BLE positioning with other positioning technologies to evaluate its advantages. In the next section, we shall discuss the related work on BLE positioning.

2.1. Wi-Fi

Wi-Fi is one of the most widely used technologies for indoor positioning. As Wi-Fi signals are regularly broadcast by access points (APs), user terminals can easily make use of the Wi-Fi signals for positioning purposes. Common positioning techniques include RSSI, time-of-arrival and fingerprint [1]. Angle-of-arrival can also be used, based on the MUSIC protocol [4], for example. Furthermore, public fingerprint databases are also available for positioning purposes (e.g., [5]). In addition to the traditional device-based approach, there has also been considerable interest in studying device-free approaches (e.g., by monitoring reflected signals), such as MaTrack (Dynamic-MUSIC) [6] and a behavior-based approach using Wi-Fi channel-state information [7].

Due to Wi-Fi's popularity, the major benefit of Wi-Fi positioning is that it can be provided through existing infrastructure (i.e., without additional investment) [8]. However, Wi-Fi APs are primarily designed for communications rather than positioning purposes (i.e., with a different design goal). Furthermore, while there are many Wi-Fi APs, they may be owned or managed by different organizations or people. That means, some APs may be changed (e.g., signal power may be adjusted) or sometimes disabled, resulting in a less certain configuration. This makes the development of a collaborative positioning method more difficult. Compared to BLE, if an organization wants to deploy a large number of Wi-Fi APs for positioning purposes, the cost will be much higher. Although most smartphones can support Wi-Fi, some mobile platforms restrict the use of Wi-Fi scanning for positioning purposes. For example, as reported by [9], Apple's iOS currently only allows developers to develop mobile apps to scan the RSSI of a connected AP, but not nearby (unconnected) APs. In other words, it may be practically difficult to implement certain Wi-Fi positioning methods.

2.2. RFID

RFID is another commonly used technology for indoor positioning. A RFID system can be passive or active and has three key components: RFID reader, antenna and RFID tag. In general, RFID positioning relies on the use of RSSI, angle-of-arrival, phase-of-arrival, fingerprint, time-of-flight and reference tags. For active RFID systems, LANDMARC [10] is one of the earliest and most widely cited active RFID indoor positioning systems based on the use of reference tags. Researchers have also proposed many improved versions of LANDMARC, such as 3-D LANDMARC [11], and LANDMARC with adaptive kNN algorithm [12]. Over the years, various active RFID indoor positioning systems have been developed, such as [13], [14], and [15]. For example, iLocate [15] uses virtual tags and frequency hopping to enhance accuracy. For passive RFID positioning systems, they are typically used for local area tracking such as equipment [16], robot [17] and autonomous vehicle [18]. Available approaches include the use of RSSI (e.g., [19]) and phase-of-arrival (e.g., [20]). By utilizing the phase information, it is possible to develop a highly accurate passive RFID system, such as Tagoram [21].

The advantages of RFID positioning are that it is lightweight, low cost and well suited for tracking assets with passive RFID tags. However, additional infrastructure is often required for its implementation. Furthermore, as most smartphones do not support RFID, it is difficult to deploy RFID indoor positioning for consumers or mobile users in general.

2.3. 4G/5G

Today, GPS is the most commonly used positioning technique used with mobile phones or terminals. However, it cannot be used effectively in the indoor environment. The 4G/LTE standard specifies three core positioning methods, namely: enhance cell ID, Assisted Global Navigation Satellite Systems (A-GNSS) and Observed Time Difference of Arrival (OTDOA) [22] [23]. It also specifies the LTE Positioning Protocol to support the aforementioned positioning methods. However, the aforementioned positioning methods are more suitable for outdoor positioning in general.

With the recent development of 5G, researchers have started to study its potential for indoor positioning. As suggested by [24], some 5G features may improve the accuracy of indoor positioning. For examples, small cell networks and mmWave enhance signal reliability and quality, and enable more accurate time-of-arrival (ToA) estimation. Also, massive Multiple-Input-Multiple-Output (MIMO) antennas provide higher signal-to-noise-ratio and thus ToA estimation uncertainty can be reduced. In addition, with the help of MIMO antenna arrays, beamforming enables position estimation using a more accurate angle-of-arrival (AoA) and angle-of-departure (DoA).

2.4. BLE

While BLE is a relatively new indoor positioning technology, there has been growing interest in studying BLE positioning methods because of their potential and advantages. Currently, most BLE positioning methods typically use RSSI and fingerprinting [9] [25] (please see the related work section). The introduction of the AoA function in the new BLE 5.1 specification may complement existing methods. However, the AoA application programming interface remains unavailable.

As discussed in [9], BLE advantages include better power efficiency for mobile devices and faster scanning responsiveness. Furthermore, because BLE beacons are cheap and easy to deploy (about $3 USD per beacon), they can be densely deployed to enhance positioning accuracy. In addition, as most current smartphones support BLE, many positioning-based applications can be developed and easily introduced. However, as most BLE beacons are battery powered, there is a need to monitor the beacons and change the batteries periodically.

In summary, compared to other indoor positioning technologies, BLE positioning offers several advantages. First, BLE is lightweight, cost-effective and energy-efficient. Second, BLE beacons can easily be deployed in an indoor environment. Last but not least, most smartphones are equipped with BLE, so it is particularly well suited for supporting indoor position-based mobile applications. To contribute to the development of BLE positioning, we present an innovative collaborative positioning method for BLE in this paper.

3. Related Work

In this section, we discuss the related work on BLE positioning, which can be classified into three main approaches: fingerprint, signal propagation and proximity. Although considerable research has been done on indoor positioning in general, there has been considerable interest in recent years in researching BLE RSSI-based positioning techniques. [26] presented a kNN and weighted kNN fingerprinting approach with both Chebyshev and Euclidean metrics. [9] proposed a BLE-fingerprinting-based positioning method. In this work, during a training phase, a signal fingerprint database is constructed. Position estimation is performed by comparing received signals (i.e., by a mobile phone) and the signal fingerprint map based on Euclidean distance and Bayesian estimator. The paper also discussed the advantages of using BLE-based positioning techniques as compared with WiFi-based techniques such as fast scanning, more energy-efficient and easy to implement. [27] also presented a fingerprinting approach. They incorporated a method named eight-neighborhood template matching (ENTM) to generate templates with RSSI values for unknown points to improve the positioning accuracy. According to [9] [28] [29], position accuracy can be further enhanced if the channel information is also provided in the fingerprint database, and the channel information is included during similarity measurements. However, this information is not commonly available in today's smartphones (i.e., by the mobile phone operation system) [30]. [25] presented another fingerprinting method with an autoencoder. Autoencoder is based on a type of unsupervised neural network, which can be used in the training phase (i.e., generation of fingerprints) to eliminate noise, and to better handle RSSI fluctuations and the loss of beacon information. Generally, after generating a fingerprint database, the user position can be estimated based on the similarity between real-time measurements and training records in the database. Typically, methods such as nearest neighbor (NN), k-nearest neighbor (kNN) and weighted kNN. [28] can also be used for reference for more distance and similarity metrics.

The signal propagation model with a trilateration approach is one of the most commonly used signal-based positioning methods. [31] compared this approach for four wireless/mobile technologies, namely Wi-Fi, BLE, Zigbee, and LoRaWAN with the focus on IoT and positioning. In general, the signal propagation model with a trilateration approach can achieve higher scalability. However, it requires a significant amount of training effort to obtain the model parameters to estimate the distance accurately. In other words, it is heavily dependent on the quality of the model parameters. One of the most widely used models is the log-distance model [3]. Regression methods can be used for training the model parameters. Based on the signal propagation approach, [32] presented a BLE-based positioning scheme based on optimization methodology and filtering mechanism (e.g., Gaussian filter). [33] studied an online self-calibration method for regular and dynamic updating of model parameters to enhance positioning accuracy. Using a Kalman filter, [34] presented a weighted centroid localization scheme with the aiming of tackling attenuation and noise issues. Position estimations are carried out based on the importance of the beacons (i.e., according to their signal strengths). Positions are estimated based on weightings assigned to the beacons in accordance with their signal strengths. [35] also proposed a method to enhance positioning accuracy by including the channel information in the broadcasting packets (i.e., for frequency diversity), employing Kalman filtering and using weighted trilateration. [36] presented three different regression methods to determine model parameters for proximity-based positioning (which will be discussed in the next paragraph). Note that these regression methods can also be applied to trilateration. After estimating the physical distances between the target node and the reference nodes, the position can then be determined by means of trilateration. However, the trilateration problem cannot ideally be solved by linear algebra (i.e., perfect equations cannot be determined due to noise). Instead, different optimization methods, such as least-square estimation, should be used.

Apart from trilateration, location can also be estimated using a proximity-based method. For example, iBeacon, proposed by Apple Inc., was one of the first proximity-based positioning techniques. In this approach, instead of providing an exact location, the output is the proximity of a region or PoI. The corresponding positioning techniques are also different. With the aim of generating proximity-based positioning reports, [36] and [37] presented a framework based on beacon deployment information and an RSS-based model. [38] presented a non-static positioning method using time series proximity reports, which take into account the motion of mobile terminals. Based on the motion model, particle filtering and smoothing methods can be used to estimate a user/terminal position based on previous measurements (i.e., also taking movement into account). [39] employed a compressive sensing technique for proximity detection. It seeks to tackle the problem of incomplete/missing signals when beacons are sparsely placed. [40] presented an improved method with a generalized similarity filter to further enhance proximity detection. For the proximity-based approach, [41] discovered an interesting finding - that positioning accuracy may in some cases actually be degraded with an increasing number of beacons (i.e., dense deployment). To achieve better positioning accuracy under dense deployment, it also proposed an adaptive scanning method with a heuristic algorithm.

Table 1 summarizes these selected related works. In general, the baseline accuracy should be approximately one to two meters on average, with 90^{th} percentile ranging from two to four meters. This will be used as the baseline for the proposed method.

4. Challenges in indoor positioning

4.1. Distance measurement

Currently, signal propagation and the RSSI fingerprinting method are two of the most commonly used tech-

Table 1: Summary of different positioning techniques

Work	Approach	Model	Major Contribution
[9]	Fingerprint	Gaussian Process regression, Bayesian Likelihood	Compared various fading mitigation schemes, scanning window mechanisms, advertising periods, beacon density, fingerprint dimensionality and transmission power.
[25]	Fingerprint	Studied Autoencoder, kNN	Positioning in a 3-D space as compared to a normally 2-D space.
[26]	Fingerprint	kNN and weighted kNN	Presented a BLE-based kNN and weighted kNN fingerprinting approach with both Chebyshev and Euclidean metrics.
[27]	Fingerprint	Eight-Neighborhood Template Matching (ENTM)	According to the authors, the proposed algorithm was able to achieve better positioning result than kNN and weighted kNN method.
[28]	Fingerprint	Channel and orientation fingerprinting, Weighted kNN	Compared 38 different distance and similarity metrics for fingerprinting matching.
[31]	Signal propagation	Log distance path loss model, Trilateration	Compared Wi-Fi, BLE, Zigbee and LoRaWAN using signal propagation approach.
[32]	Signal propagation	Log distance path loss model with Gaussian filter and piecewise fitting, Taylor series expansion based positioning	Introduced different techniques to improve positioning accuracy and training such as Gaussian filter, piecewise fitting, unified sampling and device-oriented training model. Used enhanced methods for positioning, such as weighted sliding windows and distance weighted filter based on the trilateral relations theorem. It is suggested to use active learning to regularly adjust the pre-trained model, to further enhance positioning accuracy.
[34]	Signal propagation	Log distance path loss with Kalman filter and moving average, Weighted centroid	Presented Kalman filter and weighted centroid location estimation technique to tackle attenuation and noise problems
[35]	Signal propagation	Broadcasting channel information, Kalman filter and weighted trilateration	Employed frequency/channel diversity, Kalman filtering and weighted trilateration to enhance positioning accuracy.
[36]	Proximity	Gaussian process regression (GPR) based RSS model, Cramer-Rao bound (CRB) and Barankin bound (BB)	Proposed advanced Gaussian process regression (GPR) RSS models and introduced Cramer-Rao bound or Barankin bound to minimize positioning errors.
[38]	Proximity	Motion model and particle smoothing algorithm	Used the motion of the devices for proximity positioning instead of static positioning.
[39]	Proximity	Similarity filter, compressive sensing	Investigated a positioning system with sparse beacon deployment and designed a compressive sensing approach.
[41]	Proximity	Adaptive scanning heuristic algorithm with differential evolution	Presented a positioning system with dense beacon deployment and designed a dynamic and adaptive scanning technique.

niques to estimate indoor position in an indoor environment. As mentioned above, it is time-consuming to construct a signal fingerprint database for the indoor environment, therefore the signal propagation-based method is relatively more scalable. However, to accurately estimate the position is still challenging, mainly due to the nonlinearity and fluctuation of measurements. Therefore, it is hard to estimate the distance between two nodes based on RSSI. For example, although the widely known log distance model, (i.e., Equation 1, [3]) is commonly used, the result is not very accurate, because it is highly dependent on the quality of the training of the parameters (i.e., the path-loss exponent n and the reference RSSI: $RSSI$ at d). Also, factors including noise, reflection, multi-path effect, etc. will also affect the quality, resulting in a very fluctuated measurement. Figure 1 shows the measurements of a BLE beacon's RSSI from an Android phone at different distances. This shows that (1) the RSSI range at different distances is always overlapping. In other words, given a measurement, there might be more than one possible distance, and (2) the RSSI and distance are not always negatively correlated. Some of the measurements at 3.5 m are actually larger than the measurements at 1.5 m.

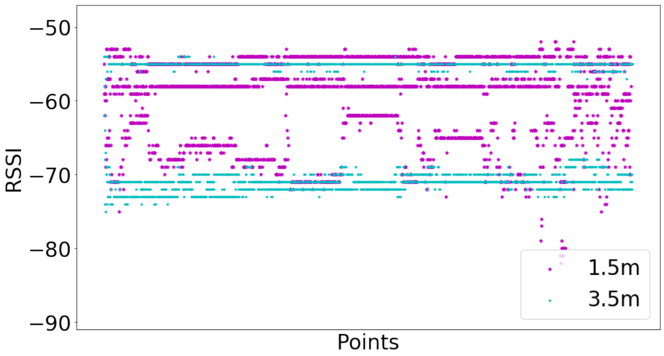


Figure 1: Measurements taken from two beacons at different distance by the same device

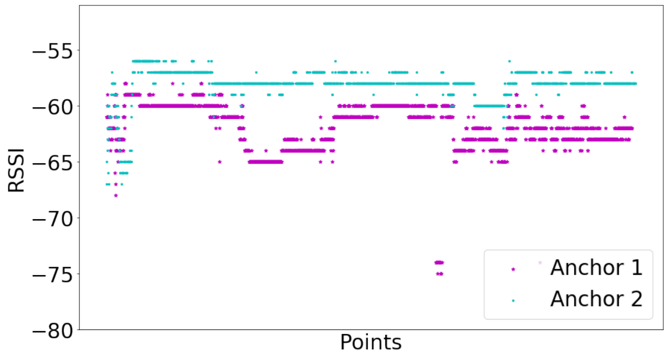


Figure 2: Measurements taken from two broadcasting anchors measured by the same device at the same distance

4.2. Training

Although signal propagation is said to be more scalable, training of the model's parameter is still required. For example, with reference to Equation 1, the RSSI value at a reference distance d_r, namely $RSSI@d_r$, and the path-loss exponent n are three unknown variables. In order to determine the values, taking extensive measurements before deployment is required. Another challenge is that the path-loss exponent is often dependent on different devices, different environments, or even different anchor placements. In other words, sometimes training the variables per environment and per device is required. This leads to an increasing amount of effort to accurately estimate distance.

4.3. Device heterogeneity

In a BLE positioning system, any BLE equipped sensors can potentially become the anchor nodes. However, they are not necessarily identical. For example, they may be equipped with different chips designed and produced by different manufacturers. The different materials of the enclosure will also differ. These factors may affect the traveling of the signal, and thus the distribution of measurements. Figure 19 is an extreme example, when a mobile phone took several measurements from two different broadcasting nodes in the same room. This shows that the distribution of measurements can vary considerably, even in the same room and with the same measurement device. Also, the broadcasting interval and broadcasting power of the beacons may be different. All of these factors will affect accuracy when using RSSI to estimate physical distance on the scanner. One of the ways to minimize the effect of device heterogeneity is to carry out training per anchor, in addition to the above-mentioned training. Similarly, this requires a significant amount of effort, and is thus not scalable.

4.4. Requirement of centralized storage medium

Generally, most current methodologies based on Wi-Fi, RFID or iBeacon require a centralized medium (e.g., a server or the local storage of the target node) to store the location information of the deployed anchors. For example, the iBeacon architecture suggests that the location of beacons should be stored in a server. Once the scanner has received the signal, it looks up the required information, such as the coordinates of the broadcaster, the broadcasting settings, etc., based on the broadcaster ID, to carry out a further calculation. The drawbacks are that it requires additional resources to maintain a server. Also, the scanner must be connected to the Internet. An illustration is shown in Figure 3. In this paper, we propose a decentralized BLE positioning protocol that does not require a centralized means to store and update information. By encoding the information into the advertising data, the mobile terminals can easily receive information through the advertising packets. We explain our approach in detail in the next section.

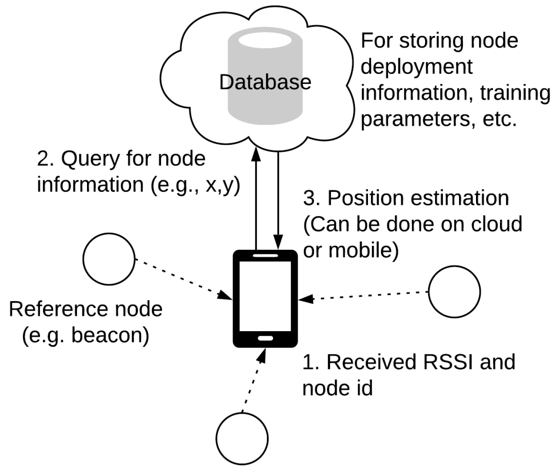


Figure 3: Typical cloud-based positioning architecture

5. Decentralized BLE positioning protocol

In the paper, we have proposed a decentralized BLE positioning protocol. We aim to solve the above-mentioned challenges, while at the same time improve the positioning accuracy. In this section, we explain the components of our proposed protocol.

First, our proposed protocol does not require a centralized medium to store the information of the anchors (e.g., coordinates). Instead, we encode the required positioning data to the BLE advertising packet emitted by the anchor nodes. The information required is:

- the unique identifier of the anchor, either the mac address or a custom-defined unique ID
- position of anchors in the local 2D coordinate system (e.g., x and y or north and east)
- the reference signal strength at different distances, such as RSSI at one meter (rssi@1m), RSSI at four meters (rssi@4m), etc.

5.1. Anchor node

The positioning system consists of a number of anchor beacons. These BLE beacons, also called anchor nodes, are placed at a predefined location and will not be moved frequently. In our proposed method, anchor nodes are responsible for both broadcasting and scanning. We assume that the anchor node has a minimum level of computational resources to store a small amount of data and execute simple programming code, such as storing an array of numbers, calculating a mean value and executing a for-loop code. Although most commercially available BLE beacons today cannot do scanning, based on our investigations, we believe this functionality can be enabled at the software level. Also, some beacons available on the market are equipped with decent hardware. For example, the Estimote Mirror beacon is equipped with a quad-core 64-bit 1.2 GHz CPU and 1 GB memory. In other words, this assumption is highly feasible and easily achievable.

First, the anchor node should broadcast its unique identifier, location (i.e., x and y coordinate), a reference RSSI value $RSSI@d_r$ and the reference distance d_r. The ID and the location of the anchor are set before deployment. $RSSI@d_r$ and d_r will be updated automatically. The anchor node will also broadcast the reference RSSIs of other anchors obtained by continuous scanning. As other anchors are also broadcasting, every anchor should receive the signal from the other anchors. When the anchor receives the signal, it calculates the signal strength average for each anchor node. The details of this updated process are explained in Section 6.

5.2. Target node

The target node, also known as the receiving node, is the object where its location is unknown, and thus requires the location estimation. In general, the target node has the ability to scan for nearby BLE signals and estimate its location by positioning algorithms. A smartphone carried by a human, a robot or an IoT sensor are examples of a target node. In our study, we focus on the application of locating humans, where the target node will be the smartphone.

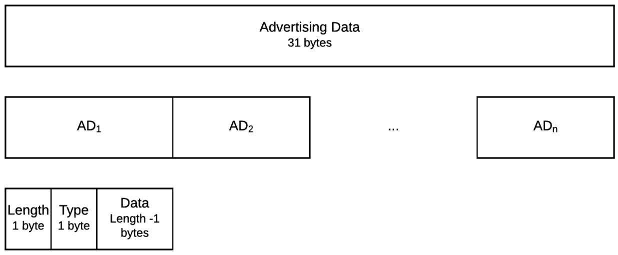


Figure 4: BLE advertising packet structure according to the specification [42]

5.3. BLE advertisement packet

As mentioned, we encode the necessary information for positioning in the broadcasting packet, instead of using a central medium. Before we encode the data into an advertising packet, we need to calculate the size we can use. According to the Bluetooth 4.0 specification, the BLE advertising data can hold up to 31 bytes of data. The advertising data (AD) is constructed by numbers of AD structure data, namely $AD_1, AD_2, ..., AD_n$. The total number of AD in one packet varies, because the data each AD holds may be different. For each AD, the first byte is the length value of that AD data, the second byte is the data type indicator, and the remaining bytes are the actual data. The Bluetooth standard requires that all advertising packets should have AD_1 as the 'Flags' AD data. The Flags AD type indicates its connection capability to others, such as 'Limited Discoverable Mode', 'BR/EDR Not Supported', etc. Because Flags AD data is 3 bytes long, the remaining spaces are 28 bytes. The BLE advertising data structure is shown in Figure 4.

5.4. BLE Indoor Positioning Service

We encode the information we have into BLE Indoor Positioning Service (IPS). Bluetooth SIG has released a series of service specifications for different types of applications. The list of adopted services can be found at [https://www.bluetooth.com/specifications/gatt/services]. Services are sets of characteristics (namely, properties) defined for a commonly known way of information exchange. For example, Heart Rate Service includes these characteristics: Heart Rate Measurement, Body Sensor Location, and Heart Rate Control Point. If a heart rate sensor follows this specification, a software developer can develop a mobile app to connect to these sensors by following the specification, without the need to study the vendor-specific API. At the same time, vendors do not need to design their own set of API for every sensor. By pre-defining the set of properties and standardizing them as specifications, multi-vendor interoperability can be achieved.

Our positioning approach also follows the BLE service specification. Local coordinates are put into the BLE Indoor Positioning Service (IPS). According to the IPS specification [42], the characteristics defined for IPS are shown

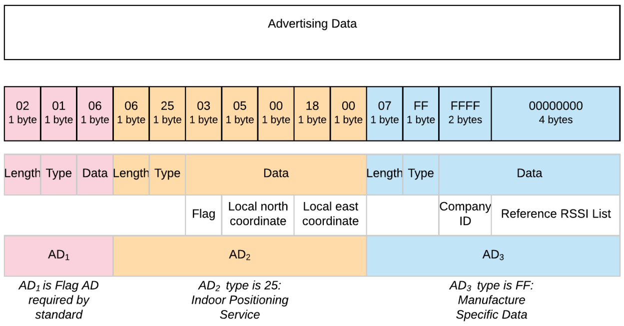


Figure 5: BLE advertising packet example with reference to the specification [42]

in Table 2. This specification defines the data structure required for estimating the location. For example, global and local coordinates can be encoded in the BLE package so that maintaining a centralized medium is not needed to store the information.

Based on the specification, at this stage, only local coordinates are needed and will be broadcast. The result AD packet of Indoor Positioning Service broadcast will be 7 bytes long. For example, if the local XY coordinate of an anchor is (5,24), the AD packet will be: 06 25 03 00 05 00 18.

5.5. Reference RSSI as additional data

In addition to the coordinates, the target node needs the reference RSSI for estimating the physical distance. Although Tx power, also referred to as signal strength at one meter, is predefined by the BLE specification for a similar purpose, we have found that sometimes this value is not accurate. The original design of Tx power is measured and calibrated before deployment. However, as the indoor environment changes rapidly, and the quality of signals is easily affected by these changes, pre-configured Tx power is often not an up-to-date reference point to estimate distance.

Therefore, we encode a set of information called Reference RSSI List to the BLE packet for target nodes to calculate distance. The details of this information are explained in the next section.

Instead of putting the value as Tx power, we put this information as 'Manufacturer Specific Data' (AD type value being 0xFF). As a result, we have 21 bytes (31-3-7, length - AD_1 - IPS) to encode our positioning data. An example of our proposed advertisement package is shown in Figure 5.

6. Measuring and calculation of Reference RSSI

As mentioned above, our proposed method does not require training before deployment and is able to automatically achieve a continuous update process. In this section, we explain this methodology in detail. When a target node estimates the position, a reference data point is required to estimate the physical distance based on RSSI. Although in general a signal propagation model can be used, a training process is also required to train the parameters of the model. In this paper, we propose the use of the list of RSSI-distance reference pairs for a target node to estimate physical distance.

6.1. Advertising RSSI-distance reference data from anchor nodes

RSSI-distance reference pair is a key-value pair, for example, $(-63.8 : 2\text{m})$, $(-68.9 : 3\text{m})$. These data are constructed by scanning the signals from other anchors. For example, when any given three anchors are deployed, namely a_1, a_2 and a_3, all anchors should continuously perform both scanning and broadcasting. For a_1, it should receive the signal strength from a_2 and a_3. Because the packets include the RSSI and the location of the broadcasters (a_2 and a_3), a_1 can easily calculate the physical distances between itself and other anchors by calculating the Euclidean distance. Based on the physical distance from other anchors and the measured RSSI, an RSSI-distance pair is formed. For each anchor, there should be $n-1$ number of RSSI-distance pairs, where n is the total number of anchors seen by the receiver. For instance, a_1 should have the pairs measured from a_2 and a_3, a_2 should have the pairs measured from a_1 and a_3 and a_3 should have the pairs measured from a_1 and a_2. This list is encoded as Manufacturer Specific Data' (AD type value being 0xFF) as explained above.

6.2. Measuring RSSI-distance reference data by the target node

When the target node is scanning, it should receive information from the anchors, as well as the list of reference RSSI. After combining the list from all anchors, the target node should have $(n-1) \times n$ pairs. With reference to the example in the previous section, the target node should receive six pairs of reference RSSI. A graphic illustration is shown in Figure 6.

7. Positioning Algorithm

Among all anchors in $a_1, a_2, ..., a_n$, where $n > 3$, every anchor a_i will measure the RSSIs from other anchors, namely $\boldsymbol{m}$, where $\boldsymbol{m}$ is the array of measurements containing all measurements except for the scanner itself (i.e., i). All anchors will broadcast these measurements (i.e., $\boldsymbol{m}$) to the target node.

7.1. Distance estimation

In order to estimate the position, the target node needs to estimate the physical distance between it and the anchors based on the RSSI. Instead of using a propagation model, we propose to utilize the measurements by the anchors to help with the distance estimation. As mentioned,

Table 2: Characteristics defined for IPS according to the specification [42]

Characteristic	Description
Indoor Positioning Configuration	This indicates whether the anchor will broadcast the positioning information. Note that if broadcasting is unavailable, a mobile node should set up a connection with the anchor to obtain the information.
Latitude	An anchor must provide this compulsory information. Based on WGS84 datum, latitude should range between -90 and 90.
Longitude	An anchor must provide this compulsory information. Based on WGS84 datum, longitude should range between -180 and 180.
Local North Coordinate	This optional information can be used to provide coordinate information on a local system/map. The value should range from -32767 to 32767 decimeters.
Local East Coordinate	If Local North Coordinate is provided, this coordinate information (sometimes called x-coordinate) is also required. Similar to Local North Coordinate, it ranges from -32767 decimeters to 32767 decimeters.
Floor Number	If required, this indicates the floor number of a location.
Altitude	This optional number shows the altitude of a node in decimeters from 0 to 65535.
Special integer	This is an optional 8-bit integer: bit 0: Stationary or mobile bits 1-3: Update time bits 4-6: Precision
Location Name	This optional information (e.g., human-readable location) shows the corresponding name of the node, including where the node is placed (e.g., the name of a shopping mall).

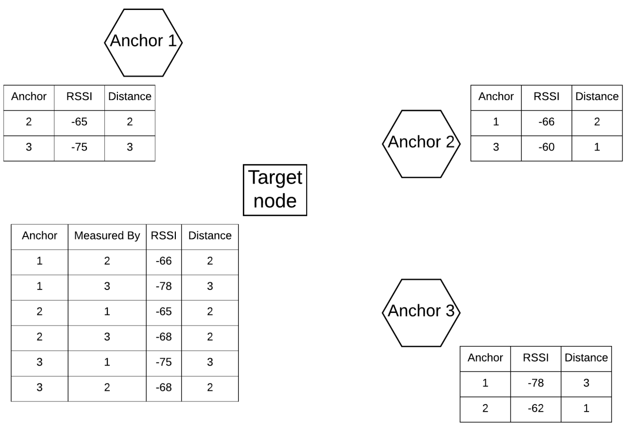


Figure 6: Example of RSSI-distance reference list

the anchors not only broadcast its deployed location, but also the list of RSSI-distance pairs taken by anchors when scanning other anchors. After the target node scans the packets from anchor nodes, it will combine the list, as shown in the table next to the target node in Figure 6.

Physical distances between the target node and anchors are then estimated with the help of the combined reference list. There are different possible ways to utilize the list. We tested two approaches, namely curve fitting and interpolation.

7.1.1. Curve fitting with least square

Curve fitting with least square technique is employed to the aforementioned log distance model (Equation 1). The measured RSSI and the known distance (i.e., the table from the target node in Figure 6) are the input, while n will be estimated by curve fitting technique using Python SciPy Trust Region Reflective algorithm. Also, the bounds of n are set from 1 to 10.

After n is estimated, the physical distance is estimated based on the measured RSSI from the target node.

We also tested a case if $RSSI@d_r$ and d_r are also unknown, letting the curve fitting algorithm find these parameters together with n, and thus estimate the distance. The details and results are presented in Section 8.3.

7.1.2. Interpolation

The input of interpolation is the same as the curve fitting method (i.e., the table from the target node in Figure 6). Instead of using the log distance model, distance is estimated based on linear interpolation and extrapolation. The physical distance is estimated based on the measured RSSI and two nearest known RSSI-distance pairs from the reference list, assuming these two points form a straight line.

The overall distance estimation algorithm is described in Algorithm 1.

7.2. Positioning estimation

In this paper, we use the positioning algorithm as in our previous publication [43]. The major difference is that the training of the value to discretize RSSI into a different bucket is no longer needed. Basically, this is a distance-based technique instead of a fingerprint-based technique. Based on the RSSI-distance reference list, the target node is able to estimate the physical distance between it and

Algorithm 1 RssiToDistance by curve fitting

Input: RSSI measurement $RSSI$, RSSI measurement at 1 meter $RSSI@1m$, Log-distance propagation model $logDistFunc$, Array RL of reference RSSI-distance measurements $[(RSSI, d), ...]$

Output: Estimated distance d

n = CurveFit($logDistFunc$, Array of RSSIs in RL, Array of distances in RL, lowerBound = 1, upperBound = 10)

d = logDistFunc($RSSI$, n, $RSSI@1m$)

return d

{$CurveFit$ is the curve fitting function to find the best n by putting multiple records of RSSI and distance into $logDistFunc$.}

{$logDistFunc$ is the Logarithmic Attenuation model as shown in Equation 1. Note that n is the only unknown in curve fitting.}

each anchor based on RSSIs. The target node uses the physical distance and the location of each anchor (as found in the advertising packet) to calculate the estimated circle of each anchor. By calculating the combination of circles from all anchors, and iteratively eliminating the points, the estimated region is calculated, as shown in Figure 7 and [43]. For error calculation, we output the midpoint of the region.

The overall positioning algorithm is described in Algorithm 2.

Algorithm 2 Positioning

Input: Array B of beacon-RSSI pair $[(b_1 : RSSI), (b_2 : RSSI), ...]$, Array K of generated point-of-interest

Output: Estimated x and y

```
B = sorted desc B based on RSSI
for b in B do
    R = [ ]
    d = CovertRssiToDistance(RSSI)
    for k in K do
        if (CheckIfWithinCircle (k, b, d)) then
            Add k into R
        end if
        K = R
    end for
end for
return MidPoint(R)
```

{$RssiToDistance$ is the function to convert RSSI into distance explained in Section 7.1 and 1}

{$CheckIfWithinCircle$ check if point k is within the circle with centroid of b_x and b_y and radius of d}

{$MidPoint$ return the mid-point of the polygon from the list of points R}

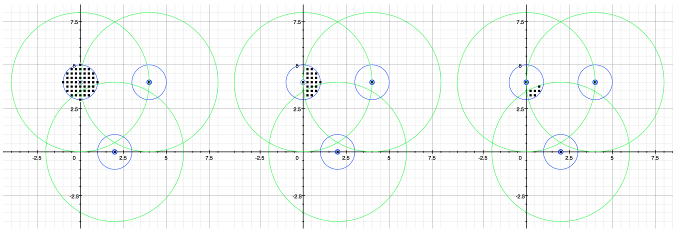
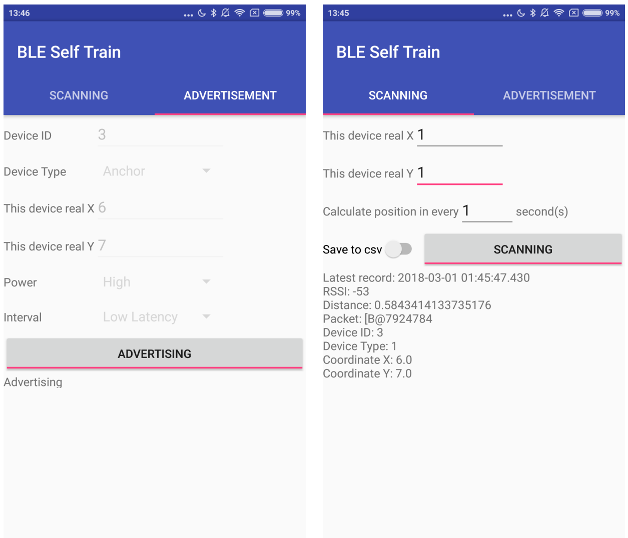


Figure 7: Calculating the estimated position based on iteration and elimination as stated in [43]



(a) Broadcasting (b) Scanning

Figure 8: Android app for experiment

8. Experimentation and Results

We conducted experiments to investigate the accuracy and practicality in different indoor environments using a custom-built Android application. Currently, it is difficult to customize a commercial-off-the-shelf BLE beacon, therefore we built an Android application and turned the Android phone into an anchor to test our method. The application is capable of both broadcasting and scanning BLE packets. The anchor mode is able to scan the packet by other anchors and calculate the mean of RSSI and the physical distance. It then encodes this data based on the above-mentioned structure and converts it into a byte array to broadcast it.

We deployed the anchor nodes at pre-defined locations and it broadcasted and scanned at the same time. After anchor nodes receive a packet from other anchors, it is temporarily stored in the memory. After a certain threshold, such as ten seconds or receiving at least ten packets from the same anchor, it will calculate the mean value of RSSI. The application then turns the data into byte arrays, encodes into the broadcasting packets, and broadcasts these together with its own location.

The target node will scan for the BLE packets. After receiving the packet, it decodes the messages and extracts

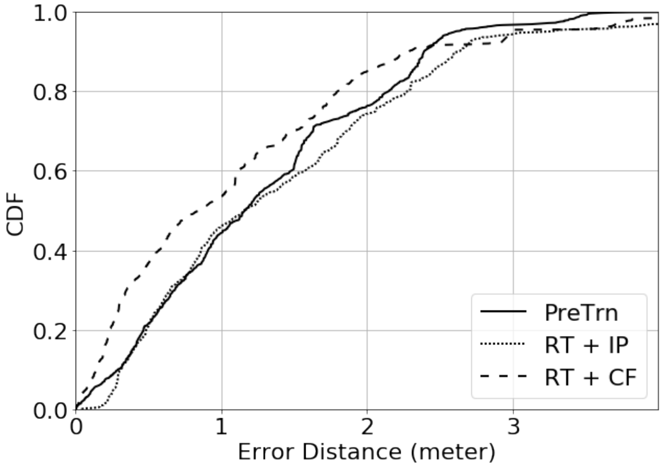


Figure 9: CDF of physical distance estimation error of different methods. PreTrn = Log model with pre-trained parameter. RT = Real-time training. IP = Interpolation. CF = Curve Fitting.

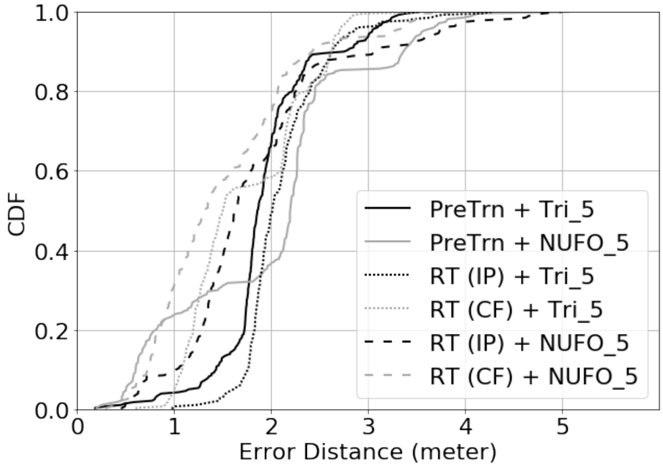


Figure 10: CDF of position estimation error of different methods. PreTrn = Log model with pre-trained parameter. RT = Real-time training. IP = Interpolation. CF = Curve Fitting. Tri = Positioning using trilateration. NUFO = Positioning using NUFO algorithm.

the needed information (i.e., the location of the broadcaster, the RSSI-distance reference list, and the RSSI) for calculation. The target will also store the data temporarily and calculate the RSSI mean value every two seconds, thus estimating the position every two seconds.

As the main focus of the proposed method in this paper is its real-time training capability, we first compare the distance estimation result. Ideally, if the physical distance estimation based on RSSI is accurate, the position estimation will be accurate. Figure 9 shows the cumulative distribution function of the error of the pre-trained method and real-time method. This result shows that for most results, the real-time method is slightly better, but there are also some extreme errors. However, given the fact that the real-time method only uses a few records measured in a very short period of time and then estimates the distance, and does not require extensive training effort, the result is promising.

We conducted experiments in classrooms and offices by placing a different number of beacons in different locations and estimate the position of the target node. We compared the pre-trained and real-time methods with trilateration and the method in our previous publication in [43] as mentioned above.

8.1. Positioning Accuracy

The 25^{th}, 50^{th},75^{th},100^{th} percentile and mean of the positioning error are shown in Table 3. The cumulative distribution function of the error is shown in Figure 10. In the legend, 'RT' means our proposed real-time training method and 'PreTrn' means the pre-trained method. 'Tri' is the trilateration method with least square estimation, and 'NUFO' is the method in our previous publication. The last number indicates the number of anchor nodes. We can see that the real-time method (RT + NUFO) achieves the best result. Approximately 80% of the results achieve an error distance of less than two meters and the average error is 1.5 meters.

Table 3: Percentile and mean of error distance in meters of different methods

	25^{th}	50^{th}	75^{th}	100^{th}	Mean
PreTrn+Tri_5	1.74	1.86	2.07	3.53	1.94
PreTrn+NUFO_5	1.14	2.2	2.36	5.07	1.98
RT(IP)+Tri_5	1.85	2.01	2.27	4.28	2.12
RT(CF)+Tri_5	1.22	1.5	2.19	3.95	1.75
RT(IP)+NUFO_5	1.35	1.63	2.16	4.99	1.84
RT(CF)+NUFO_5	0.93	1.32	2.01	4.13	1.51

It is also noteworthy that our proposed real-time curve fitting method can also improve the positioning result using trilateration (RT (CF) + Tri), compared to the model with a pre-trained parameter (PreTrn + Tri) because the distance estimation is improved as mentioned in Figure 9

We also compared the effect of the number of anchors. Figure 13 and Table 6 show that with more anchors the positioning accuracy increases when using our proposed approach. This is because, with more anchors, there are more reference distance-RSSI pairs, and thus more data can be used for curve fitting. For example, for three anchors there will be at most six pairs. For five anchors, there will be 20 pairs. This gives more measurements for the target node to estimate the parameters with curve fitting, as a result of helping the target node to estimate the distance based on the RSSI more accurately.

We also compare our proposed method with a basic fingerprinting approach. For the proposed method, the best option (i.e., RT (CF) + NUFO) was used. Essentially, we obtained the fingerprints (i.e., average RSSI based on signal reception from the anchors/beacons) for the 40 sample points where the experimental measurements were conducted (i.e., an ideal case).

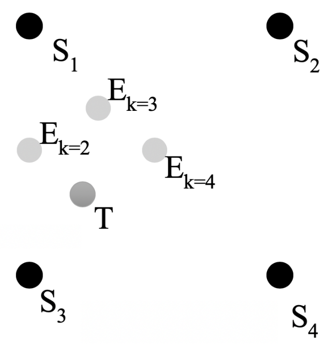


Figure 11: Example of fingerprinting using the kNN algorithm

Based on the fingerprints and measured RSSI for a target point (T), the kNN method is used to estimate the position of T. Let us explain the commonly used approach (e.g., see [1] and [26]) with an example as shown in Figure 11 and Table 4.

Table 4: RSSI for the fingerprinting example

	B1	B2	B3	B4	B5
S1	-74.2	-75	-74.5	-73.5	-79.25
S2	-60.9	-72.8	-77.1	-64.5	-68.1
S3	-76.25	-80.3	-76.3	-70.1	-78
S4	-64.5	-65.5	-69.8	-76.25	-71.1
T	-77.3	-79.5	-75.1	-70.9	-76.2

In this example, there are four sample points S1, S2, S3 and S4. Their fingerprints (i.e., average RSSI) from the five beacons B1, B2, B3, B4 and B5 are shown in Table 4 (e.g., the average RSSI for S1 from B1 is -74.2). The measured RSSI for T is also shown in the last row of the table. To determine the similarities between A and B based on Euclidean distance of A and B, the following function $d(A, B)$ is defined:

$$d(A, B) = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + ... + (A_n - B_n)^2}$$

where A_n and B_n represent the respective RSSI measurement from beacon n. Based on the above table and formula, it can be found that

- $d(T, S1) = 6.8$
- $d(T, S2) = 20.6$
- $d(T, S3) = 2.66$
- $d(T, S4) = 21.04$

That means, the nearest neighbor of T is S3, followed by S1 and S2. For the kNN method, if we choose $k = 1$, the estimated position will be the position of S3. If we choose $k = 2$, the estimated position is $E_{k=2}$ or the midpoint of S3 and S1. If we choose $k = 3$, the estimated position is $E_{k=3}$ (i.e., the mean coordinates of S1, S2 and S3).

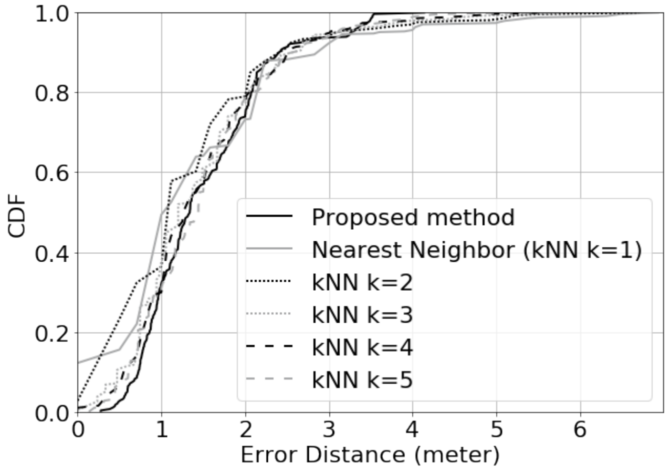


Figure 12: CDF of position estimation error of the proposed method and fingerprinting with kNN algorithm

Table 5: Comparison of positioning errors in meters between the proposed method and the fingerprint method using the kNN algorithm

	25^{th}	50^{th}	75^{th}	100^{th}	Mean
Proposed method	0.93	1.34	2.01	4.15	1.51
Nearest neighbor (kNN $k = 1$)	1.0	1.12	2.24	6.95	1.5
kNN $k = 2$	0.71	1.12	1.8	6.71	1.39
kNN $k = 3$	0.75	1.2	1.89	5.79	1.46
kNN $k = 4$	0.9	1.27	1.9	5.59	1.49
kNN $k = 5$	0.85	1.44	1.91	5.75	1.5

To compare the fingerprint method with the proposed method, we have repeated the previous position estimation experiments using the fingerprint method with different values of k. For the proposed method, the best option (i.e., RT (CF) + NUFO) was used in the experiments. Figure 12 shows the CDF of the fingerprint method in comparison with the proposed method (with the best option). It can be seen that they have similar performance, compared with the proposed method. Table 5 indicates that $k = 2$ provides the best performance. The table also shows the proposed method can achieve position estimation accuracy that is similar to the fingerprint method, which requires training of data. This indicates that the proposed method should be effective. While the fingerprint method with $k = 2$ can achieve slightly better results, the training effort for the fingerprint approach is much higher. For example, based on our experiment, it took around one hour to construct a fingerprint database for the 40 sample points, including data pre-processing. The proposed method does not require a manual training effort. Furthermore, it was found that the fingerprint approach had more extreme errors.

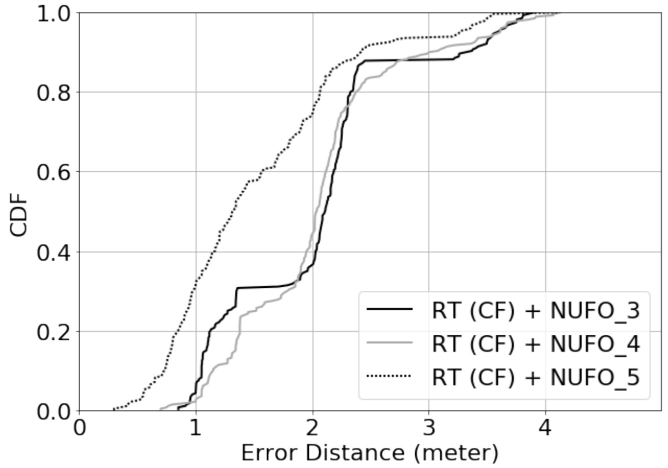


Figure 13: CDF of error distance of different numbers of anchors. The last digit at the legend indicates the number of beacons used.

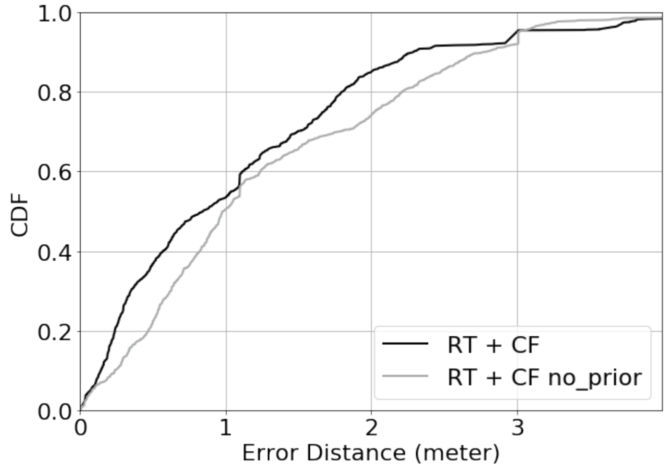


Figure 14: Comparison of CDF of error of distance estimation curve fitting presented above. CF no_prior = curve fitting without prior knowledge

Table 6: 25^{th}, 50^{th}, 75^{th} , 100^{th} percentile and mean error distance in meters of different number of anchor

No. of anchors	25^{th}	50^{th}	75^{th}	100^{th}	Mean
3	1.3	2.11	2.3	3.91	2.02
4	1.52	2.04	2.28	4.12	2.05
5	0.93	1.32	2.01	4.13	1.51

Table 7: Running time of different methods

Method	Average running time (milliseconds)
PreTrn+Tri	9.93
PreTrn+NUFO	21.65
RT(CF)+Tri	45.6
RT(CF)+NUFO	61.54

8.2. Running time

Although our proposed methods achieved higher positioning accuracy, the running time is relatively slower than other methods. As shown in Table 7, our proposed method requires the longest running time among all tested methods. Compared to the pre-trained method, the real-time methods require about 30 to 40 milliseconds more to be computed for both trilateration and NUFO positioning. However, despite the fact that the running time is longer, we believe it has less effect on the user experience. It is because normally a one to two second response time for a smartphone-based indoor positioning system is fast enough (i.e., updating the positioning every one to two seconds), and 40 milliseconds difference in one second is generally not noticeable.

8.3. Number of unknown parameters in curve fitting

The above-mentioned curve fitting method tries to find the value of the path loss exponent, i.e., n in Equation 1. This method assumes the reference RSSI is known, i.e., $RSSI@d_r$. For example, the commonly used approach is to measure RSSIs at one meter away from a beacon and use the average of the measurements as $RSSI@1$. However, this results in an additional effort to find the value. Therefore, we tested the accuracy of the curve fitting method if we have no prior information on the reference RSSI. In other words, d_r and $RSSI@d_r$ are not a measured value, but the unknown parameters to be found by curve fitting together with n.

Figure 14 compares the error of distance estimation between curve fitting with one parameter (i.e., n) and curve fitting with three parameters, namely CF no_prior in the graph (i.e., n, d_r and $RSSI@d_r$). Figure 15 compares the error of positioning between these two methods. Results show that if d_r and $RSSI@d_r$ are known beforehand, the estimation results are better. One of the reasons is that it is more difficult for the curve fitting algorithm to find three unknown parameters compared to one unknown parameter. For example, the Python function occasionally reported that the parameters could not be estimated and optimized. Therefore, although finding d_r and $RSSI@d_r$ by curve fitting might be possible, it is recommended that these two values be measured before deployment. This can be done by a one-off manual measurement, or it is also possible to use one of the anchors for the measurement of $RSSI@d_r$, i.e., putting one anchor at one meter away from another anchor to measure continuously and broadcast to the target node. Also, exploring other curve fitting algorithms to optimize three unknown parameters is also worth studying.

8.4. Deployment

It is expected that the proposed positioning system can be deployed at a reasonable cost and effort. Basically, simple and lightweight programs can be embedded in the beacons to implement the required protocols. Specifically, we need to configure each beacon to provide its fixed coordinates through the broadcasting packets. The deployment

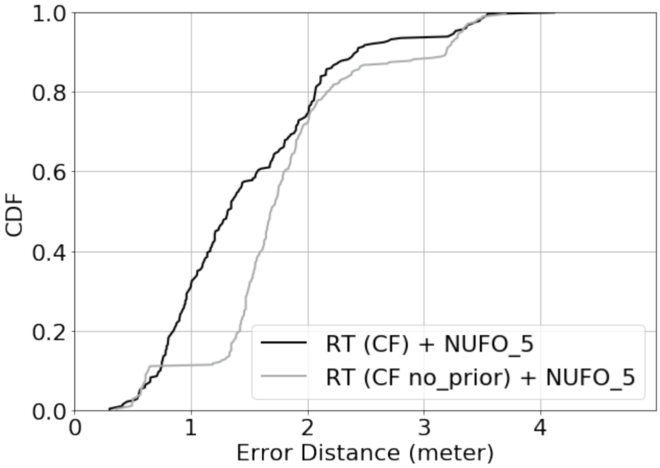


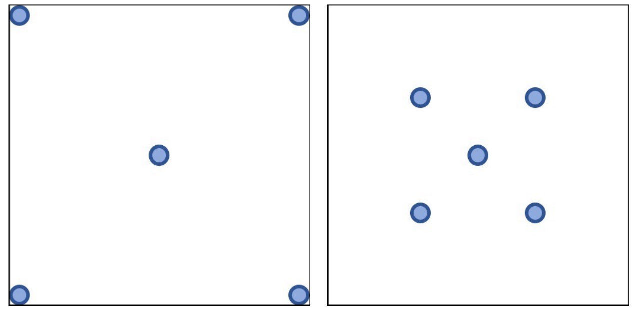
Figure 15: CDF of error distance between two curve fitting methods. CF = curve fitting presented above. CF no_prior = curve fitting without prior knowledge.

effort should be similar to existing BLE beacon-based positioning techniques (e.g., iBeacon positioning system), as the beacon coordinates should still be recorded and stored (e.g., in a database). Note that each beacon only needs to perform simple computations and send a small amount of data through the BLE packets. The relatively intensive processing is performed at the mobile terminals. There can be various applications. For instance, to provide position-based service at a shopping mall, the beacons with the algorithm can be deployed throughout the shopping mall. Customers can then use the position-based service through a mobile app.

We have also conducted further experiments to study the beacon deployment configuration using the best option of our proposed method (i.e., RT (CF) + NUFO). In the original experiments, the beacons were deployed or placed at the four corners and one in the middle of the testing area (i.e., evenly spread out). We have also conducted additional experiments using different placement/deployment configurations (see Figure 16). For alternative configuration 1 (shrink), the beacons were placed closer to one another, while keeping the same distance between each. For alternative configuration 2 (divided), the beacons were divided into two subgroups. For alternative configuration 3, beacons were placed in a more random manner.

Table 8: Comparison of positioning errors in meters with different beacon configurations

	25^{th}	50^{th}	75^{th}	100^{th}	Mean
Default configuration	0.93	1.34	2.01	4.15	1.51
Configuration 1	1.58	1.8	2.21	4.1	1.91
Configuration 2	1.65	2.15	2.76	4.81	2.12
Configuration 3	1.34	2.02	2.29	4.2	2.03



(a) Default configuration (b) Alternative Configuration 1

(c) Alternative Configuration 2 (d) Alternative Configuration 3

Figure 16: Different deployment configurations

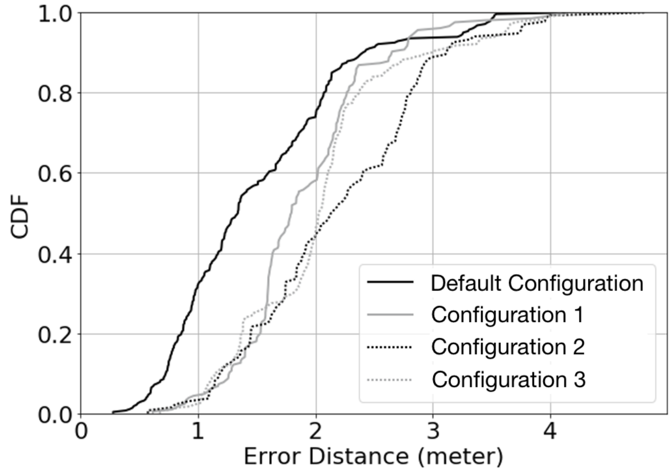


Figure 17: CDF of position estimation errors with different beacon configurations

Figure 17 shows the CDF for the position estimation result. It can be seen that the default configuration provides the best result, but the four curves are basically similar. Based on our experience in running the experiments, a high degree of accuracy can generally be achieved by spreading the beacons out evenly to provide good coverage. While the scope of this paper is not to study deployment configuration, future work can be conducted based on this paper's framework. In future, when beacons can be densely deployed, implementation of the proposed method can be greatly facilitated. A cluster of densely deployed beacons can also provide more anchors or reference points to enhance position accuracy.

8.5. *Multiple users*

Like GPS, the proposed method makes use of distributed processing, hence performance should not depend on the number of users (i.e., it is a scalable solution). Basically, through the collaborative algorithm, the beacons can learn from each other and then broadcast that information to the users of the system. Each user then utilizes the information to estimate his/her position using his/her mobile phone (i.e., in a distributed manner).

We have conducted experiments to test the proposed method in a multi-user environment. As before, for the proposed method, the best option (i.e., RT (CF) + NUFO) was used. For each experiment, there were five users (i.e., mobile phones) in the testing area, placed in random positions. Each mobile phone computed its position individually and simultaneously based on the proposed method. In the experiments, the mobile phones were placed in more than 20 positions in the testing area (i.e., many different configurations were tested). Based on the experiments, the positioning errors of each user were determined.

Figure 18 and Table 9 show the experimental results. It can be seen that the results for all users are similar. This indicates that as expected, the proposed method works well in a multi-user environment because of its distributed processing nature.

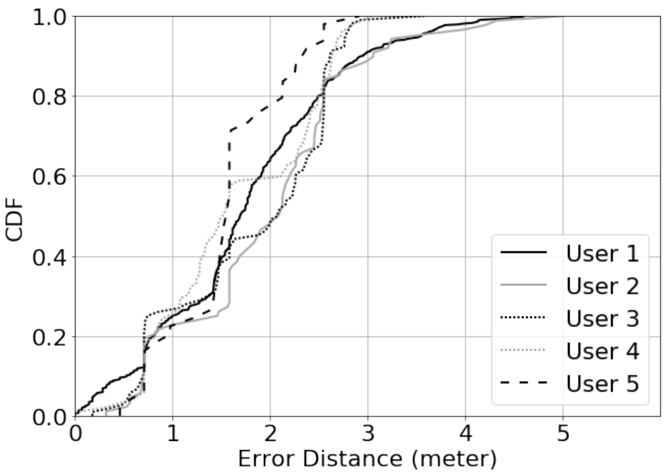
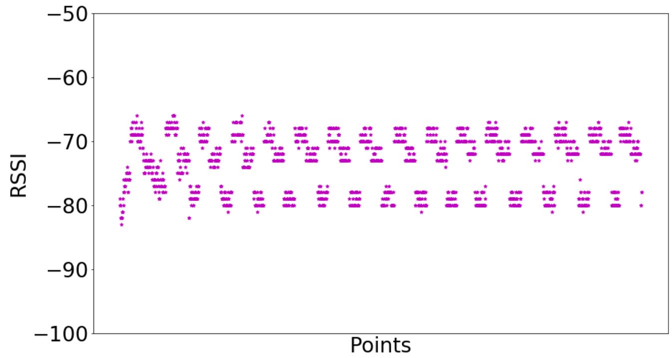


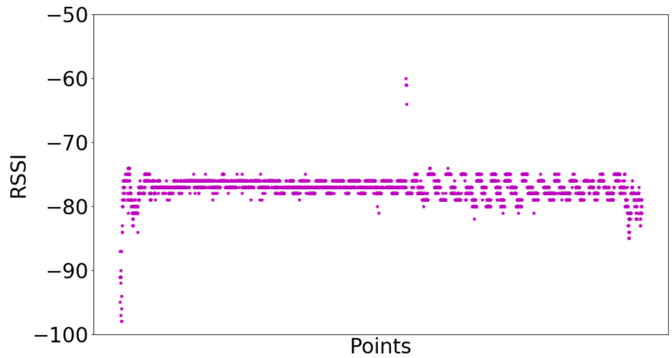
Figure 18: CDF of position estimation errors for multiple users

Table 9: Comparison of positioning errors in meters for multiple users

	25^{th}	50^{th}	75^{th}	100^{th}	Mean
User 1	1.01	1.69	2.35	4.61	1.74
User 2	1.47	2.09	2.51	5.02	1.97
User 3	0.82	2.08	2.55	3.58	1.81
User 4	1.01	1.52	2.41	3.04	1.66
User 5	1.39	1.58	1.86	2.92	1.55



(a) RSSI samples by user 2's device



(b) RSSI samples by user 5's device

Figure 19: RSSI measurements at 1.5 m from a beacon

Table 10: Measurement differences between user 2 and user 5

	Mean RSSI	Standard deviation	Average number of measurements per second
User 2's device	-73.3	4.29	6.98
User 5's device	-77.14	1.76	7.03

However, based on our observation and experience, some mobile phones may achieve better results than others because of better hardware (e.g., BLE chip and antenna) so as to alleviate the fluctuation of RSSI as well as the effect of noise. To further analyze the effect of device differences, we conducted measurements by all the devices used in the multi-user experiment simultaneously at 1.5 m apart from a beacon. As shown in Figure 19 and Table 10, user 5's device (i.e., the best-performing device) received signals with less fluctuation and smaller standard deviation when compared to user 2's device (i.e., the worst-performing device). We believe that a more stable measurements by a better device will have a better positioning accuracy. This is effectively similar to GPS – while each mobile phone receives the same GPS signals for distributed processing, some mobile phones may achieve better positioning accuracy due to their use of better hardware and software.

9. Conclusion

Using BLE for positioning has several advantages, such as lower deployment cost, and no platform restrictions for broadcasting and scanning BLE data. We believe that using BLE for positioning is a more effective solution in the end-user market. In this paper, we propose a decentralized positioning protocol that does not require a centralized server or manual training process. Based on modifications of the beacons, the beacons will broadcast and scan simultaneously to automatically achieve a training-like process on-the-fly. This also ensures that the parameters of the signal propagation model are up-to-date, thus ensuring the estimation result will be accurate. Our proposed protocol provides a solution to the major challenges of indoor positioning, namely difficulties in estimating the distance based on RSSI, the need for training and the need for re-training over time. Also, our proposed protocol is designed and tested specifically for BLE and smartphones. These show that our proposed method is simpler and easier to implement and more practical for mass deployment compared to other wireless methods, such as Wi-Fi and RFID. Based on experimentation in a real indoor environment, we can conclude that our proposed method is also accurate, achieving an accuracy of approximately 1.5 m (with 90th percentile below 3 m in general). This is generally applicable in most indoor positioning systems when locating humans. At the time of writing, we see that some recently-released smartphones are already equipped with Bluetooth 5. This provides us with opportunities to study the advantages of using Bluetooth 5 with this method in the future.

References

[1] P. Bahl, V. Padmanabhan, Radar: an in-building rf-based user location and tracking system, in: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 2, 2000, pp. 775–784 vol.2. doi:10.1109/INFCOM.2000.832252.

[2] J. B. Andersen, T. S. Rappaport, S. Yoshida, Propagation measurements and models for wireless communications channels, IEEE Communications Magazine 33 (1) (1995) 42–49. doi:10.1109/35.339880.

[3] A. Goldsmith, Wireless Communications, Cambridge University Press, 2005. doi:10.1017/CBO9780511841224.

[4] M. Kotaru, K. Joshi, D. Bharadia, S. Katti, Spotfi: Decimeter level localization using wifi, SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 269–282. doi:10.1145/2829988.2787487.

[5] E. S. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, J. Huerta, Wi-fi crowdsourced fingerprinting dataset for indoor positioning, Data 2 (4) (2017). doi:10.3390/data2040032.

[6] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, H. Mei, Dynamic-music: Accurate device-free indoor localization, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 196–207. doi:10.1145/2971648.2971665.

[7] Z. Wang, K. Jiang, Y. Hou, W. Dou, C. Zhang, Z. Huang, Y. Guo, A survey on human behavior recognition using channel state information, IEEE Access 7 (2019) 155986–156024. doi:10.1109/ACCESS.2019.2949123.

[8] S. He, S. . G. Chan, Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons, IEEE Communications Surveys Tutorials 18 (1) (2016) 466–490. doi:10.1109/COMST.2015.2464084.

[9] R. Faragher, R. Harle, Location fingerprinting with bluetooth low energy beacons, IEEE Journal on Selected Areas in Communications 33 (11) (2015) 2418–2428. doi:10.1109/JSAC.2015.2430281.

[10] L. Ni, Y. Liu, Y. C. Lau, A. Patil, Landmarc: indoor location sensing using active rfid, in: Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on, 2003, pp. 407–415. doi:10.1109/PERCOM.2003.1192765.

[11] M. A. Khan, V. K. Antiwal, Location estimation technique using extended 3-d landmarc algorithm for passive rfid tag, in: 2009 IEEE International Advance Computing Conference, 2009, pp. 249–253. doi:10.1109/IADCC.2009.4809016.

[12] K. Han, S. H. Cho, Advanced landmarc with adaptive k-nearest algorithm for rfid location system, in: 2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content, 2010, pp. 595–598. doi:10.1109/ICNIDC.2010.5657852.

[13] A. R. Jimenez Ruiz, F. Seco Granja, J. C. Prieto Honorato, J. I. Guevara Rosas, Accurate pedestrian indoor navigation by tightly coupling foot-mounted imu and rfid measurements, IEEE Transactions on Instrumentation and Measurement 61 (1) (2012) 178–189. doi:10.1109/TIM.2011.2159317.

[14] C. Huang, L. Lee, C. C. Ho, L. Wu, Z. Lai, Real-time rfid indoor positioning system based on kalman-filter drift removal and heron-bilateration location estimation, IEEE Transactions on Instrumentation and Measurement 64 (3) (2015) 728–739. doi:10.1109/TIM.2014.2347691.

[15] D. Zhang, L. T. Yang, M. Chen, S. Zhao, M. Guo, Y. Zhang, Real-time locating systems using active rfid for internet of things, IEEE Systems Journal 10 (3) (2016) 1226–1235. doi:10.1109/JSYST.2014.2346625.

[16] A. Shirehjini, A. Yassine, S. Shirmohammadi, Equipment location in hospitals using rfid-based positioning system, Information Technology in Biomedicine, IEEE Transactions on 16 (6) (2012) 1058–1069. doi:10.1109/TITB.2012.2204896.

[17] E. DiGiampaolo, F. Martinelli, Mobile robot localization using the phase of passive uhf rfid signals, IEEE Transactions on Industrial Electronics 61 (1) (2014) 365–376. doi:10.1109/TIE.2013.2248333.

[18] E. DiGiampaolo, F. Martinelli, A passive uhf-rfid system for the localization of an indoor autonomous vehicle, IEEE Transactions on Industrial Electronics 59 (10) (2012) 3961–3970. doi:10.1109/TIE.2011.2173091.

[19] S. Saab, Z. Nakad, A standalone rfid indoor positioning system using passive tags, Industrial Electronics, IEEE Transactions on 58 (5) (2011) 1961–1970. doi:10.1109/TIE.2010.2055774.

[20] A. Buffi, P. Nepa, F. Lombardini, A phase-based technique for localization of uhf-rfid tags moving on a conveyor belt: Performance analysis and test-case measurements, IEEE Sensors Journal 15 (1) (2015) 387–396. doi:10.1109/JSEN.2014.2344713.

[21] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, Y. Liu, Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices, in: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, MobiCom '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 237–248. doi:10.1145/2639108.2639111.

[22] R. S. Campos, Evolution of positioning techniques in cellular networks, from 2g to 4g, Wireless Communications and Mobile Computing 2017 (2017).

[23] Spirent, An overview of lte positioning.
URL https://www.spirent.com/assets/wp/wp_lte_positioning_overview/

[24] Y. Lu, P. Richter, E. S. Lohan, Opportunities and challenges in the industrial internet of things based on 5g positioning, in: 2018 8th International Conference on Localization and GNSS (ICL-GNSS), 2018, pp. 1–6. doi:10.1109/ICL-GNSS.2018.8440903.

[25] C. Xiao, D. Yang, Z. Chen, G. Tan, 3-d ble indoor localization based on denoising autoencoder, IEEE Access 5 (2017) 12751–12760. doi:10.1109/ACCESS.2017.2720164.

[26] Y.-C. Pu, P.-C. You, Indoor positioning system based on ble location fingerprinting with classification approach, Applied Mathematical Modelling 62 (2018) 654 – 663. doi:https://doi.org/10.1016/j.apm.2018.06.031.

[27] M. Li, L. Zhao, D. Tan, X. Tong, Ble fingerprint indoor localization algorithm based on eight-neighborhood template matching, Sensors 19 (22) (2019). doi:10.3390/s19224859.

[28] G. D. Blasio, A. Quesada-Arencibia, C. R. García, J. C. Rodríguez-Rodríguez, R. Moreno-Díaz, A protocol-channel-based indoor positioning performance study for bluetooth low energy, IEEE Access 6 (2018) 33440–33450. doi:10.1109/ACCESS.2018.2837497.

[29] S. Ishida, Y. Takashima, S. Tagashira, A. Fukuda, Proposal of separate channel fingerprinting using bluetooth low energy, in: 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), 2016, pp. 230–233. doi:10.1109/IIAI-AAI.2016.22.

[30] J. Powar, C. Gao, R. Harle, Assessing the impact of multi-channel ble beacons on fingerprint-based positioning, in: 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2017, pp. 1–8. doi:10.1109/IPIN.2017.8115871.

[31] S. Sadowski, P. Spachos, Rssi-based indoor localization with the internet of things, IEEE Access 6 (2018) 30149–30161. doi:10.1109/ACCESS.2018.2843325.

[32] Z. Jianyong, L. Haiyong, C. Zili, L. Zhaohui, Rssi based bluetooth low energy indoor positioning, in: 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2014, pp. 526–533. doi:10.1109/IPIN.2014.7275525.

[33] G. Anagnostopoulos, Deriaz, Konstantas, Online self-calibration of the propagation model for indoor positioning ranging methods, in: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016, pp. 1–6. doi:10.1109/IPIN.2016.7743644.

[34] S. Subedi, G. R. Kwon, S. Shin, S. seung Hwang, J.-Y. Pyun, Beacon based indoor positioning system using weighted centroid localization approach, in: 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), 2016, pp. 1016–1019. doi:10.1109/ICUFN.2016.7536951.

[35] V. Cantón Paterna, A. Calveras Augé, J. Paradells Aspas, M. A. Pérez Bullones, A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering, Sensors 17 (12) (2017). doi:10.3390/s17122927.

[36] F. Yin, Y. Zhao, F. Gunnarsson, F. Gustafsson, Received-signal-strength threshold optimization using gaussian processes, IEEE Transactions on Signal Processing 65 (8) (2017) 2164–2177. doi:10.1109/TSP.2017.2655480.

[37] F. Yin, Y. Zhao, F. Gunnarsson, Proximity report triggering threshold optimization for network-based indoor positioning, in: 2015 18th International Conference on Information Fusion (Fusion), 2015, pp. 1061–1069.

[38] Y. Zhao, C. Fritsche, F. Yin, F. Gunnarsson, F. Gustafsson, Sequential monte carlo methods and theoretical bounds for proximity report based indoor positioning, IEEE Transactions on Vehicular Technology 67 (6) (2018) 5372–5386. doi:10.1109/TVT.2018.2799174.

[39] P. C. Ng, L. Zhu, J. She, R. Ran, S. Park, Beacon-based proximity detection using compressive sensing for sparse deployment, in: 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017, pp. 1–6. doi:10.1109/WoWMoM.2017.7974317.

[40] L. Zhu, R. Ran, P. C. Ng, J. She, Using generalized similarity filter to enhance proximity detection for sparse beacon deployment, in: 2017 International Conference on Information and Communication Technology Convergence (ICTC), 2017, pp. 433–438. doi:10.1109/ICTC.2017.8190754.

[41] P. C. Ng, J. She, S. Park, High resolution beacon-based proximity detection for dense deployment, IEEE Transactions on Mobile Computing 17 (6) (2018) 1369–1382. doi:10.1109/TMC.2017.2759734.

[42] Bluetooth Special Interest Group, Indoor Positioning Service, Bluetooth® Service Specification.

[43] Y. H. Ho, H. C. B. Chan, Blueprint: Ble positioning algorithm based on nufo detection, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1–6. doi:10.1109/GLOCOM.2017.8254483.