# Rethinking Ranging of Unmodified BLE Peripherals in Smart City Infrastructure

Bashima Islam
University of North Carolina at Chapel Hill
bashima@cs.unc.edu

Mostafa Uddin
Nokia Bell Labs
mostafa.uddin@nokia-bell-labs.com

Sarit Mukherjee
Nokia Bell Labs
sarit.mukherjee@nokia-bell-labs.com

Shahriar Nirjon
University of North Carolina at Chapel Hill
nirjon@cs.unc.edu

## ABSTRACT

Mobility tracking of IoT devices in smart city infrastructures such as smart buildings, hospitals, shopping centers, warehouses, smart streets, and outdoor spaces has many applications. Since Bluetooth Low Energy (BLE) is available in almost every IoT device in the market nowadays, a key to localizing and tracking IoT devices is to develop an accurate ranging technique for BLE-enabled IoT devices. This is, however, a challenging feat as billions of these devices are already in use, and for pragmatic reasons, we cannot propose to modify the IoT device (a BLE peripheral) itself. Furthermore, unlike WiFi ranging – where the channel state information (CSI) is readily available and the bandwidth can be increased by stitching 2.4GHz and 5GHz bands together to achieve a high-precision ranging, an unmodified BLE peripheral provides us with only the RSSI information over a very limited bandwidth. Accurately ranging a BLE device is therefore far more challenging than other wireless standards. In this paper, we exploit characteristics of BLE protocol (e.g. frequency hopping and empty control packet transmissions) and propose a technique to directly estimate the range of a BLE peripheral from a BLE access point by multipath profiling. We discuss the theoretical foundation and conduct experiments to show that the technique achieves a 2.44m absolute range estimation error on average.


*MMSys'18, June 12–15, 2018, Amsterdam, Netherlands*

ACM ISBN 978-1-4503-5192-8/18/06…$15.00
https://doi.org/10.1145/3204949.3204950

## CCS CONCEPTS

• **Networks** → *Location based services*;

## KEYWORDS

Bluetooth Low Energy, Internet of Things, Localization, Ranging

**ACM Reference Format:**
Bashima Islam, Mostafa Uddin, Sarit Mukherjee, and Shahriar Nirjon. 2018. Rethinking Ranging of Unmodified BLE Peripherals in Smart City Infrastructure. In *MMSys'18: 9th ACM Multimedia Systems Conference, June 12–15, 2018, Amsterdam, Netherlands.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3204949.3204950

## 1 INTRODUCTION

*Mobility Tracking in Smart Infrastructure.* As the adoption of IoT devices continues, smart city infrastructures are coming to life. The rapid growth of IoT devices and services is impacting healthcare, smart retail store, home automation, parking automation, factory automation, workforce management and consumer electronics. Localizing and tracking IoT devices further enables new applications. For example, in a healthcare facility, a fine-grained localization of IoT devices allows caregivers to observe their patients' movements and whereabouts in real-time and receive notifications in case of emergencies such as when a patient falls, wanders around, enters an off-limit area, or calls for help. Emerging smart retail stores like Amazon Go [5] can track customers solely based on RF signals and reduce an overwhelming use of cameras. In smart parking and metering systems, connected vehicles can be localized using RF signals emitted by their radios. Similarly, in many other domains, an accurate localization of IoT devices solves important open problems such as occupancy detection, accessibility aid for visually impaired people, warehouse automation, and energy and resource management in smart infrastructure.

*The Rise of BLE-Enabled IoT Devices.* Bluetooth low energy (BLE) is becoming the de-facto communication standard for IoT devices due to its low-power physical and data link layer

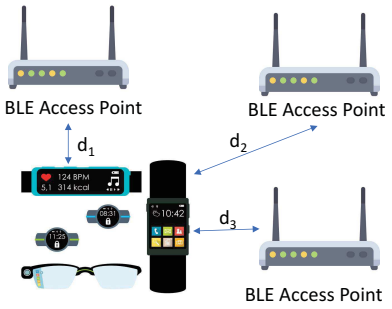


**Figure 1: Network-based BLE Localization: Multiple BLE access points estimate the ranges of the peripherals $d_1, d_2$ and $d_3$, and apply trilateration to localize them.**

operation which is suited to low-power, long lasting, battery-operated devices, and a transaction-oriented light-weight service layer that facilitates rapid development of services on resource constrained devices. By 2020, the global value of BLE devices deployed for IoT services is expected to be as much as $5.57 billion, with most of the value coming from devices in smart home, healthcare, and manufacturing. Although BLE is designed for low bandwidth traffic, in recent years, multimedia content such as images and 3D augmented reality data have been carried over BLE [40–42]. The growth of BLE is increasing in scale and in scope, and it is catalyzing large scale IoT deployments in smart cities.

*BLE Localization Models.* A BLE device acts either as a low-power *BLE peripheral* (e.g., smart watches, heart rate monitors, blood pressure monitors, and weighing machines) or a relatively higher-power *BLE central* device (e.g., a smartphone). State-of-the-art BLE-based localization techniques [26, 44] typically refer to *localizing a central device by using a set of peripherals as anchors.* For example, the use of BLE beacons (e.g., iBeacon [6, 7]), which are placed at predetermined fixed locations as anchor points, to localize people carrying smartphones is an ideal example of such a model. In this model, the infrastructure is low-power, but the mobile device that receives signals from the beacons to estimate its own location is not. We, on the other hand, are interested in localizing low-power, resource-constrained BLE peripherals that may be carried or worn by humans (e.g. wearables), or are attached to mobile platforms such as hospital beds, mobile robots, or drones. Since BLE peripherals are extremely resource constrained, the earlier model where the mobile device computes its own location is not feasible anymore. Furthermore, as most of these peripherals are commercial off-the-shelf devices, we are, in general, not capable of running programs on them. Hence, we investigate a flipped model where *a set of networked BLE central devices (i.e., BLE gateways/access points) are deployed in a smart city infrastructure (like cellular towers or base stations) and they are continuously receiving signals from nearby BLE peripherals and localizing these peripherals in real-time.* A representative setup is shown in Figure 1 where each access point (a BLE central) estimates the range (i.e. the direct point-to-point distance) from the target peripheral. Range measurements at three or more access points are combined to estimate the exact location of the peripheral.

*Rethinking BLE Localization Using Multipath Profiling.* BLE-based localization techniques can be broadly categorized into either fingerprinting [30] or RSSI-based ranging [17, 23, 53]. The downside of fingerprinting is that it requires a significant amount of training effort to characterize an environment and often such a characterization is temporary as the environment changes over time. RSSI-based ranging, on the other hand, suffers from low accuracy as no existing technique has so far been able to deal with *multipath effect* in BLE communication. In this paper, we rethink the problem and propose a completely new approach to BLE localization where we estimate the *multipath profile* (i.e, the propagation delay of different paths) of a BLE communication when signals from a peripheral reaches a central. From the estimated multipath profile, we are able to isolate the direct propagation path (both LoS and NLoS), and hence, estimate the range.

The proposed range estimation process does not require any modifications to the peripheral devices [13, 15, 33]. Unlike existing approaches, it does not require a peripheral device to carry additional tags or beacons [20]. There are some approaches [16, 40] where the knowledge of transmission power carried by a BLE beacon message (a special type of BLE peripheral) is exploited. This does not completely solve the problem since not all BLE peripherals are BLE beacons, and the conversion of a BLE peripheral to a BLE beacon, although possible, requires a complete firmware replacement. Hence, the problem of determining the range of an *unmodified* BLE peripheral is extremely challenging as the only information that is available to the centrals is the received signal strengths (RSSI) of the BLE peripherals. Our proposed technique exploits this RSSI information at different hopping channels in order to obtain the time-of-flight (ToF) of the direct path signal from an IoT device (peripheral) to an access point (central).

In this paper, for the first time, we form a theoretical foundation to infer the frequency response of a baseband signal by measuring the power of a BLE symbol or RSSI measurements at different BLE data channels. An access point, once after extracting the frequency response of a baseband signal, applies inverse discrete Fourier transform (IDFT) to estimate the multipath profile for range/location estimation. To validate our approach, we build a BLE test-bed and evaluate the proposed technique using commercial-off-the-shelf BLE

peripheral devices in both uncontrolled and controlled settings.

*Difference from WiFi Multipath Profiling.* Several recent works [31, 49] on WiFi-based indoor localization have used *Channel State Information* (CSI) at subcarrier level to derive the multipath profile. Subcarrier level CSI provides fine grained information on channel characteristics such as distortion and attenuation of signals which can be exploited to estimate the multipath profile. This is possible with WiFi (802.11a and 802.11ac) as most of the commodity WiFi network interface cards (NIC) are equipped with a channel estimator component that estimates the CSI during demodulating each OFDM subcarriers. Unfortunately, CSI information is not available in any BLE chip as BLE implements a very simple modulation scheme and due to its low-power operation, no BLE chip estimates CSI at the physical layer. Because of the unavailability of CSI information, multipath profiling in BLE is drastically different than WiFi.

*Summary of Contributions.* The contributions of this paper are as follows:

- To the best of our knowledge, we are the first to form a theoretical foundation to estimate frequency response of a BLE baseband signal by measuring RSSI at different BLE channels.
- We develop a technique to estimate multipath profile of BLE signals using commodity hardware, and ultimately measure the time-of-flight of the direct path to estimate the range of an unmodified BLE peripheral .
- We evaluate the proposed BLE ranging solution under different uncontrolled environments (e.g. in line of sight, in non line of sight, and at different locations), quantify the required wait time and effect of interference, and achieve an overall average estimated error of 2.44m (1.5m–1.87m in line of sight).

# 2 BACKGROUND

## 2.1 Bluetooth Low Energy

*BLE Channels.* Bluetooth Low Energy (BLE) is a wireless technology which is especially designed for low power devices that operate in the 2.4GHz ISM band [25]. In this band, BLE has 40 channels, numbered from 0 to 39. Each channel is 2 MHz wide. BLE uses 3 channels (37, 38, and 39) for advertisements, on which, BLE peripherals transmit advertisement packets to announce their presence and to establish connection with a central device. The rest of the channels are used for data transmission between a peripheral and a central device. BLE uses a frequency hopping mechanism to transmit data packets at different channels by using a pseudo-random hopping sequence, which is known to both the peripheral and the central device.

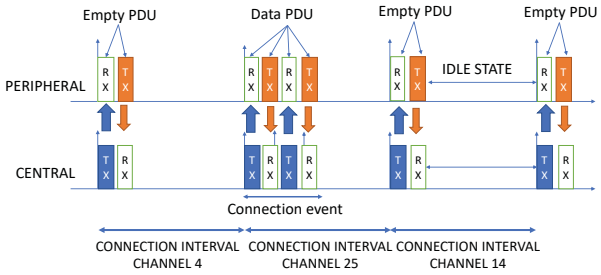


**Figure 2: Connection interval between the peripheral and the central device link-layer connection.**

*BLE Connection.* During connection establishment, a peripheral and a central agree upon the hopping sequence and the *connection interval*. A connection interval is the time between two data transfer or connection events. As shown in Figure 2, at each connection event, a central initiates data transmission. At a connection event, a peripheral replies with a single data packet for each data packet transmission from a central. By default, if there is no data to transmit, both the central and the peripheral transmit a packet that is called an *Empty Link Layer PDU (Protocol Data Unit)*. An Empty PDU transmission allows a peripheral to sync with a central and keep the link layer connection alive. In our approach to BLE ranging, we use RSSI measurements of these Empty PDUs transmitted by a peripheral device at different data channels.

*BLE Modulation and Communication.* BLE transmits data at 1Mbps, with 1 bit per symbol. The physical layer of BLE uses Gaussian Frequency Shift Keying (GFSK) modulation to generate baseband signals from a bit stream of 0s and 1s. In BLE, before applying the GFSK modulation, a bit sequence is transformed into a baseband pulse sequence ($\in \{+1, -1\}$) by using non-return-to-zero (NRZ) line coding. Later, this baseband pulse sequence is passed through a Gaussian filter before modulation to make the baseband pulse transitions (i.e, from +1 to -1, or -1 to 1) smoother. Thus, BLE reduces the interference with neighboring channels at the cost of an increased inter-symbol interference. In the modulation, smooth baseband pulse sequence is mapped to phase deviation as follows:

$$\theta(t) = \frac{\pi h}{T} \int_{-\infty}^{t} \sum_{n=-\infty}^{\infty} x[n] g(\mu - nT)\, \mathrm{d}\mu \qquad (1)$$

where, $x[n] \in \{+1, -1\}$ is the baseband pulse sequence, and $g(.)$ is the Gaussian filter or the *pulse shaping* function [9]. In the above equation, $h$ and $T$ are the modulation index[1], and the symbol period. After modulation, given $f_c$ is the carrier frequency, the BLE passband signal can be described as follows:

[1]The BLE standard defines modulation index in the range [0.45, 0.55].

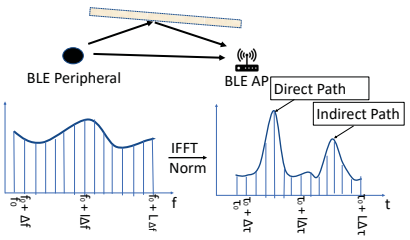


**Figure 3: Technique to transform channel frequency response to multipath profile.**

$$s(t) = A\cos(2\pi f_c t + \theta(t)) \tag{2}$$

In section 4.1, we use Equation 2 to discuss the theoretical foundation of estimated multipath profile for BLE signal.

## 2.2 Multipath Profile

*Multipath Profiling.* Signals from a transmitter to a receiver arrives directly as well as indirectly (e.g. after reflection and scattering). A *multipath profile* provides the propagation delay and the corresponding power strength profiles of the multipath arrivals of a signal. It can be measured directly by detecting multipath signals with different arrival times (i.e., time-of-flight) in the time domain– which requires sophisticated hardware that features high sampling frequency. Another way to obtain the *multipath profile* is to characterize the channel response in the frequency domain and to apply Inverse Fast Fourier Transform (IFFT) on that frequency response.

*Illustrative Example.* Figure 3 shows the process of estimating *multipath profile* in time domain from the channel response in frequency domain. It shows the channel frequency response (left plot), where $f_0$ is the starting frequency, $\Delta f$ is the resolution of frequency sampling, and $L * \Delta f$ is the total bandwidth of the frequency response. After the normalizing and IFFT transformation on the frequency response, we obtain *multipath profile* (right plot), which is a series of signal samples in the time domain with various delays, $\tau_0, \tau_0 + \Delta\tau, ..., \tau_0 + L * \Delta\tau$. Here $\Delta\tau$ is the time resolution in *Multipath profile* that is inversely related to the bandwidth of the frequency response, $\Delta\tau = 1/(L * \Delta f) = 1/B$, here $B$ is the total frequency bandwidth. Note that, as long as the frequency response bandwidth is fixed, the number of equally spaced frequency sampling in the middle has no impact on the time resolution in *multipath profile*. That means, $\Delta\tau$ does not depends on the value of $L$, as long as the total bandwidth $B$ is fixed. Note that, For the given multipath profile in figure 3, the maximum propagation delay of a signal arrival that we can measure is $L * \Delta\tau$.

*Isolating Direct Path.* In the derived *multipath profile*, signal samples having high amplitudes represent different propagation paths of the transmitted signal (both direct and indirect). Among them, the earliest signal samples with a high amplitude (not necessarily highest) represents the direct path delay or time-of-flight. Note that direct path signal can be Line-of-sight (LoS) or Non-Line-of-sight (NLoS). In case of NLoS, the absorption in the obstacles might reduce the the amplitude of the direct path compare to indirect path. This makes it harder to distinguish or detect direct path. Our results also reflects that. Given the direct path propagation delay, we can estimate the relative range between a transmitter and a receiver. For the scenario in Figure 3, we have two spikes in the *multipath profile*. The first (earliest) spike represents the direct signal path and the other spike represents the indirect signal path.

# 3 SYSTEM OVERVIEW

## 3.1 System Architecture

*Networked BLE Access Points.* Our system adopted network infrastructure based localization model as shown in Figure 4. As part of the infrastructure, we have BLE access points at fixed known locations. The role of these access points is to estimate the range (i.e the direct distance) of target BLE peripherals and send the range information to a central server. Using range estimates of a peripheral from multiple access points, the server localizes it by applying standard techniques such as trilateration [23], triangulation [17], or inter Ring Location Algorithm [21]). Since estimating the range is the key to localizing a BLE device, it remains the main focus of this paper.

*Scenarios.* This model is suitable for large scale facilities such as retailer store, smart parking system, and hospitals, where BLE peripheral devices are connected to the network via access points [8]. Unlike previous approaches, our objective is to localize the peripheral devices (such as hear-rate monitoring, blood glucose monitoring, light bulb, motion-sensor, etc.) rather than BLE beacons like iBeacons.

*Assumptions.* We assume each peripheral to be localized is connected to a central device such as BLE gateway or access point in our model. We further assume, besides the connection establishment, BLE access points have the active/passive sniffing capability. In that case, a BLE access point can sniff the Link Layer packets (i.e., empty PDU) of a peripheral, while being connected the peripheral. Prior to communication with a BLE peripheral, first the BLE access point share the link-layer connection information (e.g. hopping sequence, connection interval) with the other access points in vicinity. Thus, they are all able to sniff the transmitted link-layer packets from the peripheral device, and be able

to localize and track it. We further assume that the infrastructure is setup in a way that a peripheral is always heard by at least three access points. Since we adopt a multipath profiling-based localization technique, a line-of-sight (LOS) between a peripheral and an access point is not necessary in our system as long as the BLE signal penetrates the medium.

*Scalability.* In our localization model, each central device/or access points act independently to measures the relative distance of a peripheral device. In addition, in estimating relative range/or location of a peripheral device, we only collect the the empty PDU transmission from that device. In that case, the presence of multiple peripheral devices might effect on the number of empty PDU transmission from a peripheral device, we can successfully collect from. Therefore it might effect on the time to locate a peripheral device, but it will not have any effect on the accuracy of localization. Therefore, with enough number of access point deployment, we can localize large number of peripheral devices.

*Range Estimation Overview.* As mentioned earlier, range estimation is done after estimating the *multipath profile* of a BLE communication, which in turn, obtained by forming a *channel frequency response*. To estimate the channel frequency response, each access point measures the power of a BLE baseband symbol (i.e., RSSI) from the sniffed or captured packets at different frequencies (i.e., BLE channels). This step leverages BLE's frequency hopping design.

After forming the channel frequency response, an access point estimates the multipath profile which provides an estimated time-of-flight (ToF) of the direct path between the peripheral and itself. By multiplying the estimated ToF with the velocity of light, each access point estimates a relative distance (i.e. range) of the peripheral. Each access point sends the estimated range to the central server where rest of the localization happens by combining range measurements from three or more access points.

## 3.2 System Features

The proposed localization system has the following features.

• *Independent of Surrounding Environment.* Many RSSI-based BLE indoor localization techniques use fingerprinting or radio-propagation models [19, 45, 46]. These techniques are dependent on environment parameters, therefore, any change in the environment makes such models partially or fully invalid. In our proposed system, we use multipath profiling technique for range estimation, which is independent of the surrounding environment [50]. In addition, instead of using RSSI based models which are inaccurate and environment dependent, we use RSSI as an indirect measurement of the channel state information.

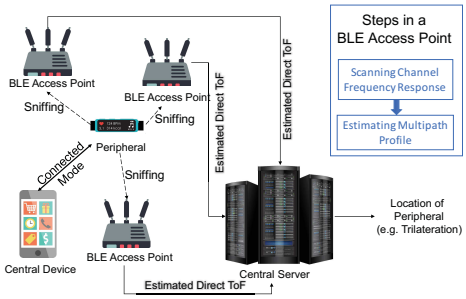


**Figure 4: Network based BLE Localization. Peripheral communicates with a central device in connected mode. Multiple fixed BLE access points sniffs the traffic and estimate the time-of-flight (ToF) of direct path. Then the central server receives the ToF estimations and localize the peripheral.**

• *Unmodified BLE Peripherals.* Most RSSI-based localization systems require changes to a peripheral to inject additional information into the BLE packets such as transmission power and time stamp [34]. Some recent proposals rely on additional sensors such as audio and IMUs of a smartphone [13, 35] to localize the device. Unlike smartphones, BLE peripherals are not always equipped with additional sensors which can be leveraged to enhance the overall location estimation accuracy. Some techniques require hardware modifications such as a directional antenna or an additional BLE beacon tag to the peripheral device [3]. Any such modifications (in hardware or software) are impractical due to the fact that billions BLE-enabled IoT devices are already in use. In our system, we leverage standard BLE protocol implementation without any modification to the peripheral.

• *Available Physical Layer Information.* Even though both BLE and WiFi operate in 2.4GHz band, they have completely different physical layer implementations (e.g. modulation/demodulation, frequency hopping, and bandwidth). Because of this difference, localizing BLE devices is quite different from localizing WiFi devices. For example, WiFi has multiple sub-carriers per channel whereas BLE has no concept of sub-carriers at all. Therefore, unlike WiFi, we do not have access to sub-carrier level channel state information (CSI) of BLE's physical layer from which we could directly estimate the multipath profile. Since estimating CSI is an essential step in WiFi demodulation, CSI comes as a by product of WiFi demodulation. Many WiFi chipsets are engineered to obtain this CSI information. On the other hand, CSI is not available in BLE as BLE uses a low-power, lightweight modulation/demodulation scheme, and thus, no BLE chipset provides CSI. The only physical layer information that BLE chipsets provide is the RSSI measurement per channel. Hence, we devise a range estimation technique for BLE devices that is dependent only on RSSI measurements.

## 4 RANGE ESTIMATION

In this subsections, we explain multipath profile and channel frequency response estimation–which are the two major steps to estimating range of unmodified BLE peripherals.

### 4.1 Estimating Multipath Profile in BLE

*Multipath Profile.* In an indoor environment, a transmitted signal travels on different paths before it reaches a receiver. On each path, transmitted signals experience a different delay, attenuation, and phase accumulation. The ultimate received signals are a combination of these different multi-path signals. Typically, *multipath profile* is a technique to identify these propagation delays of different paths on receiving a signal [50]. This ultimately allows us to estimate the time-of-flight (ToF) of the direct path, i.e. the path with the least propagation delay. In this subsection, we discuss the theoretical foundation of estimating multipath profile for BLE just by using received baseband signal strengths (i.e., RSSI). In addition, we discuss, the process of estimating time of flight (ToF) of the direct path from the derived multipath profile.

*Theoretical Derivation.* Let us assume that a BLE symbol reaches the receiver through $L$ different paths. The received signal from each path corresponds to amplitudes $\{a_i, a_2, \ldots, a_L\}$ and propagation delays $\{\tau_1, \tau_2, \ldots \tau_L\}$. In presence of these multiple paths, the received passband signal $y(t)$ is represented as follows:

$$y(t) = h(t) * s(t) = \sum_{i=1}^{L} a_i cos(2\pi f_c(t - \tau_i) + \theta(t - \tau_i)) \quad (3)$$

where, $s(t)$ is the BLE passband signal for the transmitted symbol, and $h(t) = \sum_{i=1}^{L} a_i \delta(t - \tau_i)$, is channel impulse response with time-invariant assumption, where $a_i$ and $\tau_i$ are the amplitude and the time delay of the $i^{th}$ path. In order to demodulate the received signal, a receiver first multiplies $cos(2\pi f_c t)$ and $-sin(2\pi f_c t)$ to the passband signal $y(t)$ to get the real and imaginary parts of the baseband signal $r(t)$ as follows:

$$r(t) = cos(2\pi f_c t)y(t) - jsin(2\pi f_c t)y(t) \quad (4)$$

Finally, the receiver applies a low pass filter on $r(t)$ to get the following complex baseband signal:

$$V(f_c) = \frac{1}{2}\sum_{i=1}^{L} a_i e^{-j2\pi f_c \tau_i} e^{j\theta(t-\tau_i)} = \frac{1}{2}\sum_{i=1}^{L} a_i e^{-j2\pi f_c \tau_i} v(\tau_i) \quad (5)$$

Here $v(\tau_i) = e^{j\theta(t-\tau_i)}$, is the $i^{th}$ path component of the received baseband signal, and $V(f_c)$ is the ultimate received baseband signal, which is a superposition of multi-path components. During demodulation, each symbol is extracted from this baseband signal $V(f_c)$.

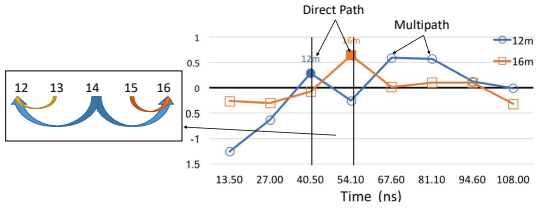


**Figure 5: In BLE, theoretical limit of absolute error in estimated distance is at most 2 m using multipath profiling technique.**

In Equation 5, we see a Discrete Fourier Transform (DFT) relation between the baseband signal $V(f_c)$ and the multipath component $v(\tau_i)$ with various delays $\tau_i$. Therefore, when we apply the inverse discrete Fourier transform (IDFT) on the frequency response of the baseband signal, we obtain the propagation delay of multiple signal paths. Note that the baseband signal value is not often directly accessible in BLE. However, RSSI measurements of a received symbol is proportional to the power of the baseband signal, $V(f_c)$ [51]. Therefore, if we measure the RSSI readings of a symbol at different carrier frequencies $f_c$, uniformly spaced between 2402MHz to 2480MHz, we can infer the frequency response of a baseband signal. Thus, applying inverse Fourier transform on the received RSSI measurement gives us multipath profile for BLE. Once estimating multipath profile, we derive the direct ToF by selecting the earliest time that shows high amplitude value. Then we multiply the speed of light with the direct ToF to estimate the direct distance or range.

*Time and Distance Resolution.* In multipath profile, time-resolution is a key parameter that defines the granularity of identifiability of delays of different multipaths. A higher time-resolution allows us to identify a fine-grained range. According to inverse Fourier transform theory, the time-resolution of a multipath profile is related to the BLE spectrum bandwidth. Such a connection indicates that a wider bandwidth leads to a higher time-resolution for the multipath profile. BLE spectrum bandwidth is limited to 80MHz, which limits the time-resolution to 13.5$ns$ that is equivalent to a distance resolution of 3.75m.

Measuring only the RSSI of the empty PDU packet in the data channels, enforce us to not consider two advertisement channels 37 and 39, which ultimately reduces the frequency bandwidth to 76MHz, and the distance resolution to 4m. Thus, in our multipath profile, each estimated distance becomes a multiple of 4m. In that case, any actual distance that falls between $4 * n$ and $4 * (n + 1)$ (for n=0,1,2,..L), we will observe a higher amplitude either in $4 * n$ or $4 * (n + 1)$ based on its closest proximity in the multipath profile. That means, according to Figure 5, any distance between 12m to 14m will show higher value for distance 12m, and any distance between 14m to 16m will show higher value for distance 16m

in the multipath profile. Thus, we have at most 2m absolute measurement error in the worst case.

For a bandwidth of $B$, two multipaths are indistinguishable if their propagation delays differ by less than $1/B$. In that case, both multipaths are viewed as one multipath component. Therefore, the time resolution $\Delta\tau$ leads to $c/B$ uncertainty in terms of the length difference between non-distinguishable paths, where $c$ is the speed of signal propagation. In BLE, the path length uncertainty is 4m, which limits our BLE ranging up to $\pm 2m$.

## 4.2 Forming Channel Frequency Response

*Necessary BLE Channels.* In order to perform multipath profiling at a sufficient granularity, an access point needs to collect RSSI readings from a peripheral over multiple BLE channels, which need to be uniformly spaced between 2402MHz to 2480MHz. Note that, in forming the channel frequency response from RSSI reading for range estimation, we do not need to sample all BLE channels. Typically, at an indoor environment, the maximum range of a BLE peripheral device is limited to 16-20 meters. Given the available bandwidth, the minimum time resolution in multipath profile can be around 4m as discussed in Section 4.1. Therefore, in addition to the lower and the upper ends of BLE channels' measurements, we only need to measure at least 2-3 equally spaced channels in between the two ends to range a BLE peripheral with maximum possible resolution using multipath profiling.

*Dealing with WiFi Interference.* Since BLE implements an adaptive hopping mechanism to avoid interference with other 2.4GHz communications, in presence of WiFi, a BLE devices will not hop to all channels. However, some lower end and upper end channels (e.g. channel 0 and 36) are not affected by WiFi interference. In this case, in addition to these two ends, if we have enough BLE channel measurements in the middle, WiFi interference will have no impact on the accuracy of range estimation with BLE.

*Required Waiting Time.* According to the BLE standard, at every connection-interval, a BLE peripheral hops to a different channel. In off-the-shelf BLE devices, we find that the connection-interval is between 7.5ms to 120ms depending on the trade-off between data rate and power consumption. In our experiments, we observe that it takes about 2-3 seconds to collect enough number of BLE channel measurements to form the channel frequency response.

*Collected Packet Types.* BLE devices use different transmission powers for transmitting different types of packets. For example, the transmission power of a BLE advertisement packet is different from a BLE data packet. Similarly, BLE data packets have different transmission power levels than


(a) (b)

**Figure 6: (a) Custom BLE access point using three Ubertooth One sniffers and a Raspberry Pi, (b) RF shielded room (Faraday cage) where some of the experiments (controlled) are done.**

empty PDU packet transmissions. Since transmission power affects RSSI readings, when forming the channel frequency response from RSSI readings, we need to make sure that the transmission power of each received empty PDU packet are the same across different channels. Hence, we normalize the RSSI measurements across different channels to make sure that the channel response is invariant to the transmission power. Unfortunately, we do not have any control on changing or knowing the transmission power of the BLE packets from a peripheral device. However, the transmission power of empty PDUs remain the same from a peripheral device across different BLE channels. Therefore, when collecting RSSI readings from a peripheral, we choose empty PDU packets from a peripheral, which is easy to differentiate from a central's empty PDU, and other data PDUs. In addition, empty PDUs are sent at every connection interval (i.e., in milliseconds). Therefore, in a very short time, an access point can collect RSSI readings from a peripheral across multiple BLE channels.

# 5 IMPLEMENTATION NOTES

In this section, we discuss key implementation issues of the system that we think are helpful for anyone who wants to reproduce the results.

*Access Point.* We use off-the-shelf hardware to implement the BLE access points as well as the peripherals. Since our main objective is to measure the range, we passively monitor a peripheral device at an access point and sniff its packets while it is connected to another central device.

To implement the access points, we use Bluetooth Low Energy (BLE) sniffers called the Ubertooth One [4] which are connected to a Raspberry Pi 3 [11] as shown in Figure 6(a). Since we do not know which of the three advertisement channels will be used to establish a connection, we use three Ubertooth Ones to quickly pick up the link layer connection

between a peripheral and a central. Here, each ubertooth one is connected to one of the three advertisement channels.

Ubertooth One is an open source hardware platform for experimenting with 2.4GHz wireless protocols. It is equipped with a RP-SMA RF connector, CC2591 RF front end, CC2400 wireless transceiver, and a LPC175x ARM Cortex-M3 microcontroller. Ubertooth sniffs the 2.4GHz ISM band and receives the signals transmitted by BLE peripherals [12]. Ubertooth by default provides the average RSSI of a packet. In order to collect fine-grained RSSI readings per symbol, we customize the Ubertooth firmware.

Once the Ubertooth picks up the link layer connection, it starts following the hop sequence and collects RSSI values per symbol for empty PDUs.

*Peripheral Device.* As peripheral devices, we use unmodified Lightblue Beans [2] paired with their sister application on an Android smartphone (Nexus 5).

# 6 EVALUATION

In this section, at first, we describe the experimental setup. Then, we evaluate the ranging accuracy of the proposed BLE-ranging method in uncontrolled environment. Next, we quantify the time needed to form channel frequency response both experimentally and with simulation. Finally, we demonstrate the effect of other 2.4 GHz wireless interference on channel frequency response.

## 6.1 Experimental Setup

Indoor scenarios mostly consists of corridors and rooms of different sizes. Thus we choose corridors and rooms of medium size for our experimental evaluation. We use three scenarios (Table 1) for conducting the experiments. For the first scenario, we choose a long corridor in one of the buildings at our campus. We keep the access point and the peripheral in line of sight. As the second scenario, we use the same corridor but place a metallic board between the peripheral and the access point to create a non-line of sight scenario. In both of these scenarios, we vary the distance between the access point and the peripheral from 2m to 16m and collect RSSI values at different distances. Finally, we select a lab room in the department as our third scenario and vary the distance between the access point and the peripheral from 4m to 8m.

As discussed earlier, we can not differentiate between two paths with less than 4m distance between each other. So, we do not collect any data at positions with less than 4m distance from the walls in the direction of the direct path. The data collection process ensures that people do not move between the peripheral and the access point. However, several WiFi access points (10 on the floor) and other wireless devices including WiFi, classic Bluetooth and other BLE devices have been present in the environment. As mentioned earlier, we only log the RSSI values of the empty PDUs from the peripherals. For each distance/ trial points, we collect data for two minutes and we get around 2000 values for each data point. For a trial, we take the mode of the RSSI of the empty PDUs in each channel. We choose mode as it is robust against outliers.

**Table 1: Experimental Scenarios**

| | Environment | Range |
|---|---|---|
| Scenario 1 | Corridor (Line of Sight) | 2m - 16m |
| Scenario 2 | Corridor (Non Line of Sight) | 2m - 16m |
| Scenario 3 | Room (Line of Sight) | 4m - 8m |

## 6.2 Ranging Accuracy

In Figure 7, we show the average absolute ranging error for each scenario. Although the average estimated error is 1.83m for the line of sight scenario in the corridor (scenario 1), it rises to 4m in the non line of sight scenario (scenario 2).In the non-line of sight scenario, the direct path signal experiences a drop in power while penetrating obstacles, and becomes weaker compare to indirect paths, which make it harder to identify. In scenario 3 (room), we achieve the lowest average estimated error of 1.5m.

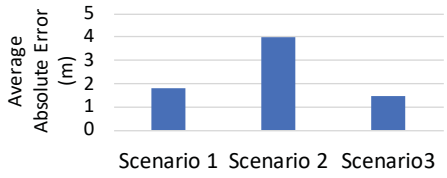


**Figure 7: The average estimated absolute distance error in three uncontrolled scenarios is 2.44m.**

In Figure 8, we show the relationship between the estimated and the actual distances for all three scenarios. Since many data points in Figure 8 coincides and these coincident points are visually represented by a single point. Here, we see that the trend line of the data is a straight line having a near 45 degree angle with the axes. This trend line has a mean squared error of 0.42m.

## 6.3 Channel Frequency Response Formation Time

In Figure 9, we empirically determine the probability of achieving different bandwidths for various amount of waiting time. We avoid collecting data from two bordering channels (channel 37 & 39) as these are advertisement channels

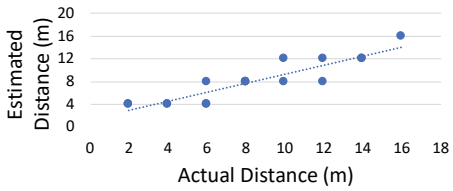


**Figure 8: The scatterplot of actual distances and corresponding estimated distances for all uncontrolled environments shows a linear trend having a mean-squared error of 0.42m.**

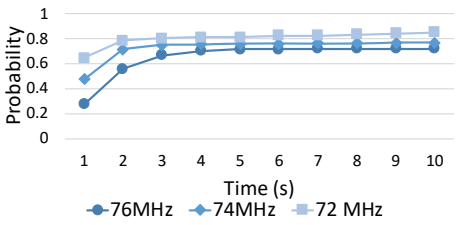


**Figure 9: The probability of achieving bandwidths (76MHz, 74MHz, and 72MHz) increases with waiting time.**

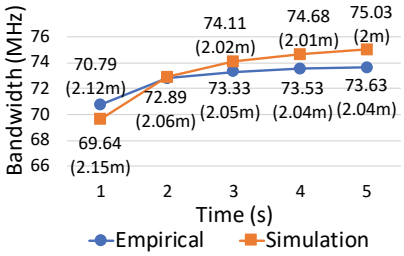


**Figure 10: The expected bandwidth and corresponding theoretical maximum absolute distance error limit in uncontrolled environments for different waiting times for both empirical and simulation results are shown.**

and do not carry empty PDUs. Therefore, we achieve a highest bandwidth of 76MHz as opposed to 80 MHz. We observe that our system is able to achieve 72 MHz bandwidth 79% of the time when it waits for at least 2s to collect RSSI measurements. Similarly, the system achieves the highest 76 MHz bandwidth 70% of the time if it waits for 3s. The corresponding distance errors for each of the three types of bandwidths are mentioned in Table 2.

In Figure 10, we show the expected bandwidth (as well as the distance error) for both experimentally and with simulation. In order to simulate the pseudo random frequency hopping mechanism in BLE, we use an uniformly distributed pseudo random number generator. We choose the connection interval or hopping interval to be a random number between 7.5 ms and 100ms [1]. We use over 1,50,000 data points for the simulation. In Figure 10, we observe that both simulation and data from real experiments achieve an expected bandwidth of around 73MHz in 2 seconds. Although in simulation we observe 75.03 MHz of expected bandwidth in 5s, in real-life data it is 73.63 MHz. From these observations, we conclude that the proposed system achieves an absolute error of 2.06m at most by collecting RSSI values for about 2s to form the channel frequency response.

**Table 2: Theoretical distance resolutions and maximum distance errors for different bandwidths (B)**

| Bandwidth | Distance Resolution | Theoretical Max Distance Error |
|---|---|---|
| 76 MHz | 3.95 m | ± 1.97 m |
| 74 MHz | 4.05 m | ± 2.03 m |
| 72 MHz | 4.16 m | ± 2.08 m |

Note that for simplicity and a low cost implementation, we use off-the-shelf BLE sniffer (Ubertooth One) which is prone to dropping packets [12] as its clock drifts over time and it waits to resynchronize itself. As a result of packet drops, we require more time to receive a desired bandwidth than expected. This is an implementation issue and can be resolved by using a robust software defined radio such as an USRP.

## 6.4 Effect of Interference in 2.4GHz

Different wireless technologies, e.g. WiFi, Zigbee, and classic Bluetooth, coexists in the 2.4GHz ISM band along with the BLE. A major reason for BLE to hop frequencies is to avoid interference with these other technologies. Hence, BLE tends to avoid channels that tends to have a higher traffic [43]. We create such interference scenario and quantify performance of our proposed system under such interferences.

In order to understand the hopping behavior accurately, we use a RF shielded room (a Faraday cage) as shown in Figure 6(b) for these experiments. This room is made with Ferrous walls (including ceiling and floor) with a non-Ferrous layer on top. This combination of Ferrous and non-Ferrous material makes the room free from all types of RF interference from the outside world. To introduce controlled interference inside the room, we set up a WiFi access point that communicates with a laptop on WiFi channel 1. This is the only other signal besides the signals between the BLE peripheral and BLE central. Although the BLE connection avoided overlapped BLE channels with WiFi channel 1, we achieve 72.89 MHz bandwidth in 2 seconds as shown in Figure 11. We further observe that the simulated data achieve 74.96 MHz bandwidth in 5s.

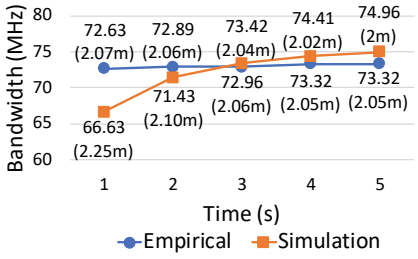


**Figure 11: The expected bandwidth and corresponding theoretical maximum absolute distance error limit in presence of only WiFi channel 1 for different waiting times for both empirical and simulation results are shown.**

Figure 12 shows the result of a similar experiment but this time with only WiFi channel 6 activated. Here, we achieve 71.16 MHz and 72.70MHz bandwidth in 2 seconds for empirical data and simulated data respectively. From these observations, we can conclude that we can achieve at least 71.16 MHz bandwidth in the presence of other wireless signals.

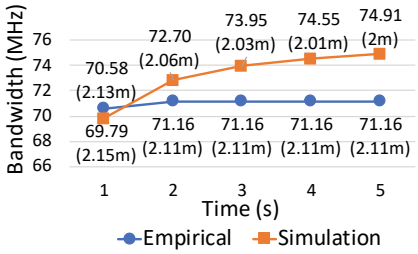


**Figure 12: The expected bandwidth and corresponding theoretical maximum absolute distance error limit in presence of only WiFi channel 6 for different waiting times for both empirical and simulation results are shown.**

## 7 RELATED WORK

*RSSI-based Fingerprinting.* RSSI-based fingerprinting is the most popular localization technique for BLE-enabled devices. Several studies has been performed to compare the performance of BLE and WiFi based fingerprinting [24, 29, 38]. These studies achieved better accuracy using BLE-based fingerprinting compared to WiFi-based fingerprinting. In order to improve the fingerprinting accuracy, filters are proposed to remove the outliers [32]. For further refinement, complex methods e.g. fusion of BLE and WiFi RSSI [48], estimation of the propagation model [28, 36] , consideration of the effect of different channels [37] have been proposed for fingerprinting. However, the requirement of generating radio map for different scenarios makes fingerprinting a costly and impractical solution. Although some works address this challenge by using crowd-sourcing [14] and model based estimation [22], this approach is difficult to deploy as giving incentives to public every time is not feasible. To avoid radio map generation and eliminate the effect of environment change, we use multipath profiling for estimating the distance.

*RSSI-based Ranging.* Another popular localization approach is RSSI-based ranging with BLE beacons (e.g. iBeacon [7], estimode [6]) with algorithms like trilateration, triangulation, inter Ring location etc. In these approaches, multiple BLE beacons (at least three) advertise custom iBeacon packets [10] and a central device (e.g. smartphone) receives the packets to localize itself. This iBeacon packet is different from BLE peripheral packets [10] and contains extra information e.g. transmission power. Such technique consists of two parts - localization and ranging. In order to improve the localization accuracy of trilateration and triangulation, both preprocessing and post processing of the data has been proposed [17, 23].

Ranging is the most important step of the ranging-based approach. RSSI-based regression models and path loss model are the most common techniques for ranging BLE devices [39, 52, 53]. However, these models are highly vulnerable with the change of scenarios and thus requires remodeling for different environments. In order to tackle this problem [19] proposed a solution to estimate the environment using motion sensor of the smartphone. However, all IoT peripherals are not equipped with motion sensors. Moreover, to increase the precision, fusion of WiFi based fingerprinting and BLE based trilateration has been proposed. Even though it increased the accuracy, such technique requires the presence of both BLE and WiFi radio in the IoT device which is not always possible. Besides, the high computational need in the target device for such methods, makes these an inefficient solution for battery powered IoT devices. In our system, we follow network-based localization model, where the calculations are performed in the access points.

*Network-based Localization.* A few works have focused on network-based localization techniques, where several BLE access points localize a BLE enabled device. However, these techniques require special software (e.g., modified transmission packets to carry extra proximity and transmission power information) or hardware (e.g., directional antenna) in the peripheral device [3, 34]. To avoid modifying the transmission packet [20] proposes attaching a BLE beacon to the target BLE peripheral. Even though BLE beacons are inexpensive, attaching beacons to every peripheral is not a viable solution. In our solution, we do not require any special or modified hardware or software in the peripheral devices.

*Using Additional Devices and Sensors.* [19] locates and tracks a BLE beacon using smartphone and improves the accuracy by estimating the environment, mitigating the RSS fluctuation and noise filtering. However, this solution requires additional sensors (e.g. magnetometer and IMU) and specific movement of the smartphone (L-shaped). Many other works have used additional sensors e.g. acoustic sensors [13], IMU [18, 33], magnetometer [15] of the smartphone respectively to improve localization accuracy. But, such techniques require additional hardware which is not a scalable for large scale deployments. Our localization system is not dependent on any parameters (e.g. transmitting signal, data packet format, timestamp and payload information) that may vary from device to device. Moreover, our localization requires no modification (both software or hardware) to the peripheral device.

*CSI-based Solutions.* Some works [31, 49] have used physical layer information of WiFi protocol (e.g. Channel State Information (CSI) [27]) for fine-grained localization. For example, [49] uses CSI to estimate the time-of-flight (ToF), and [31] combines CSI and MIMO setup to estimate the angle-of-arrival (AoA) and ToF. Though these works have achieved sub-meter level accuracy, these methods for WiFi is not directly applicable to BLE. Because, CSI is not available, and its existing estimation technique is not practical for BLE Physical layer implementation.

## 8 DISCUSSION

*Lack of Phase Information.* In our approach, we only estimate the multipath profile, which is the time delay of different paths of the signal. We apply IFFT on the frequency response of a baseband signal to estimate the multipath profile. However, our approach of estimating multipath profiling is limited by the BLE bandwidth spectrum, which have a low time resolution to differentiate multipath signals that are separated by less than 4.16m. In addition, we use frequency response of the signal strength of a symbol (i.e., RSSI), which is an indirect measurement of the power of a baseband signal, to estimate the multipath profile. As we see in equation 5, the baseband signal has both amplitude and phase. Unfortunately, available BLE sniffers (e.g., ubertooth) are unable to extract both amplitude and phase information of a baseband signal. In our approach, we use a coarse information (i.e., RSSI) in estimating the multipath profile. Thus, we loose phase information in our multipath profile estimation for BLE signal. With the phase accumulation information of different paths, one could estimate the travel time of a signal.

*Limitations of Ubertooth.* We have used BLE sniffer (Ubertooth One) to implement our access points for simplicity of implementation. However, being a low cost sniffer, Ubertooth One has some limitations. Due to the clock drift of the system, it fails to stay time synchronized with the BLE peripheral and central. Thus it experiences time shift while hopping and results in missing hops and packets [12]. However, this is an implementation issue and can be solved by either using more powerful BLE sniffers or creating wide-area IoT service utilizing BLE devices at the edge [47].

## 9 CONCLUSION

In this paper, we address the ranging of BLE peripheral devices under a practical constraint of not modifying them in any ways. We leverage the frequency hopping mechanism, modulation-demodulation technique and empty control packet transmission of BLE protocols to solve the ranging problem using multipath profiling. The approach is invariant to the environment and the types of the peripheral devices. Along with providing theoretical formation of our approach, we evaluate our ranging solution, and determine the time needed to perform multipath profiling and the effect of the presence of other wireless signals.

## ACKNOWLEDGMENT

This paper was supported, in part, by NSF grant CNS-1704469.

## REFERENCES

[1] 2016. Connection Parameters of BLE. (2016). https://punchthrough.com/blog/posts/maximizing-ble-throughput-on-ios-and-android
[2] 2016. Light Blue Bean. (2016). https://punchthrough.com/bean
[3] 2016. Quuppa. (2016). http://quuppa.com/
[4] 2016. Ubertooth-One. (2016). https://greatscottgadgets.com/ubertoothone/
[5] 2017. Amazon Go. (2017). https://amazon.com/b?node=16008589011
[6] 2017. Estimode. (2017). https://estimote.com/
[7] 2017. iBeacon. (2017). https://en.wikipedia.org/wiki/IBeacon
[8] 2017. Large Scale Deployment. (2017). https://goo.gl/7mQD6F
[9] 2017. Pulse Shaping. (2017). https://en.wikipedia.org/wiki/Pulse_shaping
[10] 2018. BLE Advertising Primer. (2018). http://www.argenox.com/a-ble-advertising-primer/
[11] 2106. Raspberry Pi. (2106). https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
[12] Wahhab Albazrqaoe, Jun Huang, and Guoliang Xing. 2016. Practical bluetooth traffic sniffing: Systems and privacy implications. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 333–345.
[13] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. 2009. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 261–272.
[14] Philipp Bolliger. 2008. Redpin-adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. ACM, 55–60.

[15] Manoj V Bramhe, Jeetendra Gan, Nayan Ghodpage, Ankit Nawale, and Gurendra Bahe. 2017. Indoor Positioning System using Magnetic Positioning and BLE beacons. *International Research Journal of Engineering and Technology* (2017).

[16] QA Budan, A Naderi, and DL Deugo. 2017. Range of Bluetooth Low Energy Beacons in Relation to Their Transmit Power. *Internet Computing and Internet of Things, 2017 International Conference on* (2017).

[17] Song Chai, Renbo An, and Zhengzhong Du. 2016. An Indoor Positioning Algorithm Using Bluetooth Low Energy RSSI. In *AMSEE*.

[18] Vivek Chandel, Nasimuddin Ahmed, Shalini Arora, and Avik Ghose. 2016. InLoc: An end-to-end robust indoor localization and routing solution using mobile phones and BLE beacons. In *IPIN, International Conference on*. IEEE.

[19] Dongyao Chen, Kang G Shin, Yurong Jiang, and Kyu-Han Kim. 2017. Locating and Tracking BLE Beacons with Smartphones. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. ACM.

[20] Kenneth C Cheung, Stephen S Intille, and Kent Larson. 2006. An inexpensive bluetooth-based indoor positioning hack. In *Proceedings of UbiComp*, Vol. 6.

[21] Rejane Dalce, Adrien Van den Bossche, and Thierry Val. 2014. An experimental performance study of an original ranging protocol based on an IEEE 802.15. 4a UWB testbed (regular paper). In *IEEE ICUWB*.

[22] F Serhan Daniş and Ali Taylan Cemgil. 2017. Model-Based Localization and Tracking Using Bluetooth Low-Energy Beacons. *Sensors* (2017).

[23] Aitor De Blas and Diego López-de Ipiña. 2017. Improving trilateration for indoors localization using BLE beacons. In *Computer and Energy Science, 2017 2nd International Multidisciplinary Conference on*. IEEE.

[24] Ramsey Faragher and Robert Harle. 2014. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation*. 201–210.

[25] Carles Gomez, Joaquim Oller, and Josep Paradells. 2012. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. *Sensors* 12 (2012). https://doi.org/10.3390/s120911734

[26] S Gowrishankar, N Madhu, and TG Basavaraju. 2015. Role of BLE in proximity based automation of IoT: A practical approach. In *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*. IEEE.

[27] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review* 41, 1 (2011), 53–53.

[28] Loizos Kanaris, Akis Kokkinis, Antonio Liotta, and Stavros Stavrou. 2017. Fusing Bluetooth Beacon Data with Wi-Fi Radiomaps for Improved Indoor Localization. *Sensors* 17, 4 (2017), 812.

[29] Eric Kim. 2013. DeepBLE-Localized navigation using Low Energy Bluetooth. *Dept. of CIS* (2013).

[30] Mikkel Baun Kjærgaard. 2007. A taxonomy for radio location fingerprinting. In *International symposium on location-and context-awareness*. Springer, 139–156.

[31] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 269–282.

[32] Pavel Kriz, Filip Maly, and Tomas Kozel. 2016. Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems* 2016 (2016).

[33] William Wei-Liang Li, Ronald A Iltis, and Moe Z Win. 2013. A smartphone localization algorithm using RSSI and inertial sensor measurement fusion. In *Global Communications Conference*. IEEE.

[34] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics* (2007).

[35] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. 2012. Push the limit of WiFi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 305–316.

[36] Jan Pelant, Zdenek Tlamsa, Vlastimil Benes, Ladislav Polak, Ondrej Kaller, Libor Bolecek, Jan Kufa, Jiri Sebesta, and Tomas Kratochvil. 2017. BLE device indoor localization based on RSS fingerprinting mapped by propagation modes. In *Radioelektronika (RADIOELEKTRONIKA), 2017 27th International Conference*. IEEE, 1–5.

[37] Jovan Powar, Chao Gao, and Robert Harle. 2017. Assessing the impact of multi-channel BLE beacons on fingerprint-based positioning. In *IPIN, 2017 International Conference on*. IEEE.

[38] Faragher Ramsey and Robert Harle. 2015. Location fingerprinting with bluetooth low energy beacons. *IEEE Communications* (2015).

[39] Jenny Röbesaat, Peilin Zhang, Mohamed Abdelaal, and Oliver Theel. 2017. An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study. *Sensors* (2017).

[40] Chong Shao, Bashima Islam, and Shahriar Nirjon. 2018. MARBLE: Mobile Augmented Reality Using a Distributed BLE Beacon Infrastructure. In *Internet-of-Things Design and Implementation (IoTDI), 2018 In ACM/IEEE International Conference on*.

[41] Chong Shao and Shahriar Nirjon. 2017. ImageBeacon: Broadcasting Color Images over Connectionless Bluetooth LE Packets. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 121–132.

[42] Chong Shao, Shahriar Nirjon, and Jan-Michael Frahm. 2016. Years-long binary image broadcast using bluetooth low energy beacons. In *Distributed Computing in Sensor Systems (DCOSS), 2016 International Conference on*. IEEE, 225–232.

[43] Sérgio Silva, Salviano Soares, Telmo Fernandes, António Valente, and António Moreira. 2014. Coexistence and interference tests on a Bluetooth Low Energy front-end. In *SAI*. IEEE.

[44] Marco Terán, Juan Aranda, Henry Carrillo, Diego Mendez, and Carlos Parra. 2017. IoT-based system for indoor location using bluetooth low energy. In *Communications and Computing (COLCOM)*. IEEE.

[45] Adel Thaljaoui, Thierry Val, Nejah Nasri, and Damien Brulin. 2015. BLE localization using RSSI measurements and iRingLA. In *Industrial Technology (ICIT), 2015 IEEE International Conference on*. IEEE.

[46] Arvin Wen Tsui, Yu-Hsiang Chuang, and Hao-Hua Chu. 2009. Unsupervised learning for solving RSS hardware variance problem in WiFi localization. *Mobile Networks and Applications* 14, 5 (2009), 677–691.

[47] M. Uddin, S. Mukherjee, H. Chang, and T. V. Lakshman. 2017. BLESS: Bluetooth low energy service switching using SDN. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*.

[48] Andrei Vasilateanu, Nicolae Goga, Laurentiu Guta, Monica N Mihailescu, and Bujor Pavaloiu. 2016. Testing Wi-Fi and bluetooth low energy technologies for a hybrid indoor positioning system. In *Systems Engineering (ISSE), 2016 IEEE International Symposium on*. IEEE, 1–5.

[49] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with a Single WiFi Access Point.. In *NSDI*. 165–178.

[50] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise power delay profiling with commodity wifi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 53–64.

[51] Zheng Yang, Zimu Zhou, and Yunhao Liu. 2013. From RSSI to CSI: Indoor localization via channel response. *CSUR* (2013).

[52] Faheem Zafari, Ioannis Papapanagiotou, Michael Devetsikiotis, and Thomas Hacker. 2017. An ibeacon based proximity and indoor localization system. *arXiv preprint arXiv:1703.07876* (2017).

[53] Yuan Zhuang, Jun Yang, You Li, Longning Qi, and Naser El-Sheimy. 2016. Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors* 16, 5 (2016), 596.