RSSI or Time-of-flight for Bluetooth Low Energy based localization? An experimental evaluation

Davide Giovanelli, Elisabetta Farella
ICT Center - Fondazione Bruno Kessler
{dgiovanelli, efarella}@fbk.eu

***Abstract*—In this paper, we focus on Bluetooth Low Energy (BLE) and in particular on its use for ranging, starting from the observation that data on RSSI in BLE comes nearly for free. The SDK typically provides to developers an easy way to extract this data and use it to implement their algorithms. However, RSSI based localization techniques have known limits. An alternative information to be used in ranging for localization purposes is Time-of-Flight (ToF). Still, this data is not provided by the BLE API, therefore we propose a practical approach for ToF extraction on top of BLE to be used as alternative to or in combination with RSSI. Furthermore, with the paper, we release the sources of the library used to perform the ToF measurement on BLE, that can be used *per se* or as input for a localization algorithm. We tested the measurements indoor and outdoor at different distances, both considering Line-of-Sight free or occluded by user body. We conclude evaluating ranging performance, test repeatability and comparing the obtained results with the popular RSSI based approach.**

***Index Terms*—Localization, Ranging, Bluetooth, Time of Flight, ToF, RSSI.**

I. Introduction

Since its launch in 2010, Bluetooth Low Energy (BLE), or Bluetooth 4.0, an emerging protocol that enables low-power wireless connectivity, has pervasively reached consumer devices such as smartphones, tablets, wearables and laptops. This protocol has established itself not only for communication purposes but also for additional functionalities such as localization and beaconing.

One of the primary aims of fixed Bluetooth beacons is to provide location (or proximity) based services to users in the neighbors of the beacon itself. In fact, since the transmission range is limited (from few meters up to hundred, depending on settings), if a user receives the beacon's packets it means that he/she is in the proximity of the beacon and then he/she may have the access for a specific service in that place. This functionality has been also used at commercial level to find things, by attaching a beacon to keys, wallets, bicycles, etc.

A more complex service that is commonly provided using beacons is indoor localization [1, 2, 3], which aims at replacing the service provided by GPS for indoor environments, where the GPS signal cannot be received. This is frequently based on the Received Signal Strength Indicator or RSSI, from which distance between two beacons can be derived. Unfortunately, RSSI has known limits mainly due to RF signal propagation effects such as reflections, absorption, degradation of measurements accuracy with distance, etc. Anyhow, on top of RSSI several localization approach can be built, e.g. based on triangulation (requiring at least 3 fixed beacons) or fingerprinting. These techniques typically require either complex filtering on measurements to improve accuracy or a preceding off-line phase of data collection to be compared in the on-line phase or for training a machine learning algorithm [4, 5].

ISBN 978-3-903176-03-4 2018 IFIP

Distance between two BLE nodes can be extracted not only from RSSI but also from Time-of-Flight (ToF). However, in existing BLE implementations the ToF measure is not provided and therefore a custom solution must be developed. Being based on a different physical phenomenon with respect to RSSI, it can be an alternative or a companion to RSSI as input for a localization algorithm. Measuring ToF can be challenging due to the clock resolution that would be required. In this work, we describe how to extract this information from a BLE module without changing the standard stack. To partially overcome the issue with clock resolution, we propose to oversample the signal and reconstruct it with advanced filtering. We therefore provide a general method and the corresponding open source library to be used on the nordic nRF52 platform.

Therefore, the contribution of this paper is manifold: we first show how to measure ToF using a commercial out-of-the-box BLE hardware and how to embed this measurement without interfering with the existing BLE stack. Then, we collect and present measurements performed with both ToF and RSSI approach for ranging in various scenarios (i.e. indoor/outdoor and in line-of-sight/non-line-of-sight). We evaluate performance and repeatability of measurements. In section VI, we outline in particular the different behaviors of RSSI and ToF when line of sight (LoS) is occluded by the body carrying the BLE node. Moreover, it is part of this contribution the open source software module for measuring the ToF on BLE using the Nordic NRF52 platform.

II. Background

As mentioned in the introduction, Bluetooth has been exploited both commercially [3] and in research as a suitable technology for localization.

One frequent approach is based on proximity, which requires the presence of several beacon in the surrounding. To this purpose, a building (e.g. the Gatwick airport in London in [6]) can be equipped with a number of Bluetooth beacons, which periodically broadcast their IDs and some other information about their status (battery level, sensor data). A smartphone application can exploit the beacons signals by activating the Bluetooth scanner; in this way, it will receive the packets sent by all the beacons that are in the proximity of the user. The packets are then decoded and used, for example, to grant the access to a location-based service to the user. The proximity is typically based on the RSSI, which can be easly recovered through the BLE API. It is measured in dBm and represents the amount of power detected by the receiver during the reception of the packet. Since the power density decreases with a known law (namely the Friis equation [7]) as the distance from the transmitter increases, it is possible to estimate if the smartphone is close to the beacon or far, just by knowing the RSSI value [8]. Therefore, by knowing the beacons' location, also the user's location can be estimated. This technique is usually known as *model-based-localization*. Under ideal conditions (line of sight, absence of reflections, accurate RSSI sampling) the RSSI-model-based-localization provides in general good results. However, in indoor environments the radio propagation is strongly affected by reflections on walls, ceiling and furnitures. This leads to achieve poor distance estimation performance [8, 9, 10] when one tries to predict the distance between the smartphone and the beacon relying only on the RSSI data and using the physical models. Moreover, since the model has a logarithmic nature, all kinds of noise/error that affect the RSSI value are reflected in the estimation error that grows linearly as the distance between devices increases.

To overcome the effects of reflections and the non-idealities of indoor propagation, *data-based* (or *RSSI-fingerprinting-based*) techniques can be applied [3, 11, 12, 13]. Instead of using a physical model for converting the RSSI to a distance and then estimate the location, these techniques rely on an off-line phase, where a number of RSSI values are collected at known positions using a scanning device (i.e. the device to be localized). Once the labeled data is collected, a machine learning algorithm is trained to output the most probable position given the previously acquired RSSI readings which acts as a fingerprint for the propagation in the environment under observation. One positive aspect of this strategy is that the reference nodes (i.e. the beacons) do not need to be at known position since the machine learning based model just requires the measure to be deterministic. Therefore the only constraints on the beacon is that it must be at fixed position. This technique is more precise with respect to the use of a propagation model, since the trained algorithm inherently takes into account the effects of reflections that rarely are modeled in the physical models.

Coupled with this, there are also tracking algorithms that exploit models representing physical constraint (i.e. a human target cannot jump from one side of the room to the other in few millisecond) for filtering out bad estimations or compensate for artifacts based on the history of the positions [14, 15].

Even if the RSSI fingerprinting is widely used also in commercial products [3, 16] there are some drawbacks that limit its applicability and reliability:

- The RSSI values are strongly affected by the antenna radiation, which is potentially different on each smartphone model, and also by non-line-of-sight conditions [9]
- The off-line phase for collecting RSSI at known positions requires time and is ad hoc for that place
- The off-line phase needs to be repeated if the building structure is changed (furniture, mobile walls) because also reflections change
- It cannot be applied to the class of application that rely on mobile networks [17]
- Sometime large and hard to predict errors can rise [18].

On the other side there are techniques to measure the distance traveled by a radio signal by means of the propagation time between the sender and the receiver. They are called Time-of-Fight (ToF) techniques [19].
Since radio signals travel at a known speed (i.e. the speed of light, approximatively $30\ cm$ every $1\ ns$) it is possible to calculate the traveled distance having the travel time. As a drawback, making a time measurement in the nanosecond scale is a critical task; moreover, the clock drifts on the two sides of the link impose additional limits. For these and other reasons the main technology that is used to perform these measurements is the Ultra Wide Band (UWB) radio.

The UWB radios send information using very short burst (called chip) of electromagnetic perturbation. Because of the short duration of these chips, the reflections on walls or ceiling are received as echoes, and not as a reverberation overlaid to the direct signal, making it possible to increase the reliability of the localization through echo cancellation techniques [20].

The drawback is the need of dedicated UWB hardware that is unavailable on today's smartphones, and since market forecasts envision the Bluetooth technology to become even more pervasive than now [21] in few years, it is worth investigating the use of ToF approach as an alternative or in conjunction with RSSI methods.

It is worth to highlight that, in this paper, our aim is not to provide a final solution for indoor localization. We are instead focusing on ranging between two BLE standard nodes, measured with the approach of ToF. Today's indoor localization systems, based on BLE are in fact ignoring ToF as input to determine proximity, loosing a powerful source of data for localization. This data is of course affected by noise, anyway it still provides consistent information because

the environment impacts in a different manner on ToF with respect to what usually experienced with RSSI.

III. Time of Flight on BLE

In general the Time-of-Flight technique can be applied to localization because the time taken by the electromagnetic wave to propagate from the transmitter to the receiver is proportional to the distance between them. In a localization scenario, once the distance between the target node and at least three reference nodes (anchors) is known, the target node can be localized with respect to the anchors on a two dimensional plane. If the target needs to be localized in a three dimensional space one additional anchor is necessary [22].

One of the main challenge of measuring the Time-of-Flight is the needed time resolution; in fact, the propagation speed of RF signals is approximately 300000 km/s. This means that, in the ideal case, to achieve 30 cm of accuracy in a single measurement, the BLE radio and the timer employed for measuring the ToF should be clocked at 1 GHz, which is almost two orders of magnitude more that what available on the typical BLE SoC. Moreover, to correctly measure the propagation time, a common clock reference should be shared between the transmitter and the receiver, otherwise at the receiver side the trigger for starting the timer would be missing. The synchronization via wireless of a common clock reference in the GHz range is unfeasible on todays low cost SoC. Moreover, the need of being compliant with the BLE further complicates it.

To face these constraints we applied two techniques:

- **Averaging:** A distance estimation is obtained by means of averaging N consecutive ToF measurements. This helps in increasing the accuracy even if the reference timer used for measurement is clocked well below the GHz
- **Two-Way ToF:** We will focus on the Two-Way propagation time. It implies a bidirectional communication and it represents the time taken by the signal to propagate from node A to node B, plus the processing time on node B, plus the propagating time in the opposite direction, from B to A.

The first technique does not require a sophisticated analysis, it only needs a buffer on the SoC memory able to store N timer values. Once the values are collected they are averaged and then the distance is calculated based on the mean value.

Instead, the Two-Way Time-of-Flight measurement implies to have a deep look into how BLE standard works. As per specification, every time two BLE devices communicate over the same RF channel, the radio activity of the two nodes is coordinated to leave a non-overlapping interval of 150 us between the transmission phases of the two radios. This interval is called Inter Frame Space (IFS or T_{IFS}) [23][1] and it applies to all the Bluetooth versions starting from 4.0.

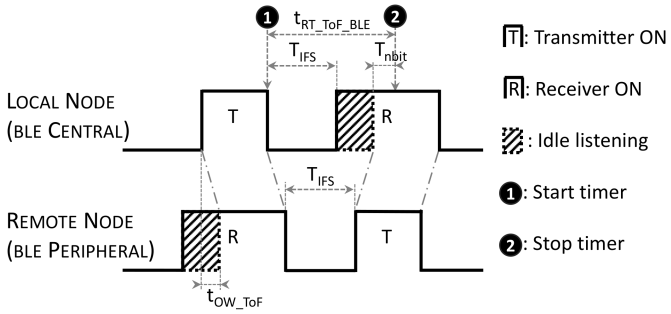


Fig. 1: Connection event representation. The One-Way propagation time t_{OW_ToF} is exaggerated to highlight the details of the measurement. Figure not to scale.

Since this IFS is fixed by the specification, by measuring the delay between the end of transmission to the actual start of reception, it is possible to extract the information regarding the Two-Way ToF (Fig. 1). By halving it we calculate the One-Way ToF, which is denoted with t_{OW_ToF} in Fig. 1.

The IFS has to be respected every time the communication is bidirectional and coordinated by a central device. In the BLE world there are two main cases where this sort of coordinated communication takes place:

- at every connection event after two devices have established a BLE connection
- when a scanner device sends a scan request to an advertiser (or a beacon).

In this paper, we focus on the first one, since it relies on a BLE connection and, as a consequence, data collection will be faster, more reliable and, since the connection data is exchanged using channel hopping over 37 BLE channels, we benefit of channel diversity. The second one, instead, exploits the advertising mode of BLE, therefore it will happen only on the 3 advertising channels [23][2]. Its occurrence is less frequent and less controllable, therefore the data collection will be slower.

IV. Setup

A. Requirements

First of all, we want to test off-the-shelf hardware with its limitations. Since one of the aims of Bluetooth is to be low cost, our solution cannot rely on high-end clocks nor expensive hardware. We also want a solution that does not require the redefinition of the specifications of Bluetooth, therefore our measure will be performed on regular BLE packets, without adding any kind of proprietary transmission.

B. Hardware

We will use two nodes, one will be referred as *local* the other will be referred as *remote*. The local one will perform the measurement and will estimate its distance from the remote

[1] Vol6.B.4.1

[2] Vol6.B.1.4

one, while the remote will not do anything else but behaving like any BLE device (working as a BLE beacon). For the Bluetooth terminology the local nodes will take the Central role, while the remote nodes will take the Peripheral role.

As BLE radio we choose the nRF52840 by Nordic Semiconductor, which is a low cost Bluetooth compatible system on chip (SoC). It includes the radio, the processing unit (MCU) and all the typical peripherals of today's microcontrollers (timer, ADC, UART, etc). Even if the hardware and the libraries used are Bluetooth 5.0 compliant, for this paper, we used only the features already available on the version 4.0 of the specifications.

C. Implementation

The local node will estimate its distance from the remote one by exploiting the Inter Frame Space, as explained in Sec. III. Therefore, it will measure the actual time interval between the end of its transmission (denoted by ❶ in Fig. 1) and the reception of the first symbol transmitted by the remote node (denoted by ❷ in Fig. 1). The resulting time can be described with:

$$t_{TW\_ToF\_BLE} = T_{IFS} + T_{nbit} + T_D + 2\ t_{OW\_ToF} \qquad (1)$$

Where TW in $t_{TW\_ToF\_BLE}$ stands for Two-Way. The firsts three terms of the sum are fixed and they are: $T_{IFS} = 150\ \mu s$, $T_D = 47.65\ \mu s$, $T_{nbit} = n\ \mu s$, while $2\ t_{OW\_ToF}$ is the back and forth propagation time. T_{nbit} is the on-air time for the first $n\ bits$ of the packet, and this fixed delay is due to the fact that the actual reception can be detected by the hardware only after some bit are received. Since Bluetooth 4.0 uses $1\ Mbps$ modulation, each bit is transmitted in $1\ us$. In our case, $n = 40$ because the MCU in use generates an event to stop the timer only after the reception of the preamble and the access address ($40\ bits$ in total). Instead, T_D is a further delay, which is due to the electronics, this value is retrieved from the datasheet of the component. For details on the BLE packet format the reader can refer to the specification document [23][3].

The time counting task is carried out using one of the digital timers included in the SoC. This timer is configured to increment the counter with a $16\ MHz$ cadence and it is automatically started and stopped by the radio peripheral without the intervention of the MCU (this connection exploit the PPI - Programmable Peripheral Interconnect of the nRF52840). This ensures the minimum processing delay on the timer management since the radio and the timer are connected by hardware. The source code of the developed software module used for measuring the ToF can be found in [24].

For the measurement we make use of one timer that is active just for the time necessary for collecting one single ToF measurement ($t_{TW\_ToF\_BLE} \approx 200\ \mu s$). Since its maximum current consumption is $120\ \mu A$ and the maximum duty cycle is 0.027[4], the measure adds, in the worst case, $3.2\ \mu A$ to the average current consumption. This value is comparable with the current consumption of the SoC when it is in sleep mode. Nevertheless, the measure can be carried out only when the device is active and connected to another BLE device: in this state the average current consumption of a BLE central (again at the maximum duty cycle) is lower than $1\ mA$ [25]. Therefore, ToF measure is negligible in terms of consumption compared to the average current for maintaining a BLE connection, being three orders of magnitude lower.

In other words, we are piggybacking the distance measurement on a regular BLE communication, with very low overhead. Of course, if no communication is taking place between the BLE devices, the connection has to be established first and some hundreds of packets has to be exchanged before having some ToF result. The average current for his procedure may consistently vary because, before establishing the connection, the device is requested to go in *scanning* to discover the remote node. Excluding extreme cases with very unfavorable settings, the overall average consumption will be below $1.5\ mA$, with a duration that depends on the number of packets to be collected.

V. Experimental procedure

For each connection event, which includes one Tx phase and one Rx phase happening on the same RF channel (Fig. 1), we store the $t_{TW\_ToF\_BLE}$, the RSSI and the carrier frequency. In fact the Bluetooth frequency hopping scheme requires a channel change (channel hop) at every connection event.

During the tests the nodes where fixed on a stand at $40\ cm$ above the ground and the distance between the two nodes has been varied to investigate how the ToF compares to RSSI for estimating the distance. We collected data from 1000 connection events at: $20\ cm$, $50\ cm$, $1\ m$, $2\ m$ and so on with $1\ m$ step up to $20\ m$.

The tests were repeated indoor and outdoor and, for both environments, we tested the Line-of-Sight (LoS) and the Non-Line-of-Sight (NLoS) conditions. The non-line-of-sight condition is created interrupting the line-of-sight between the two nodes using a man standing with its torso near ($30\ cm$) the local node. This is for simulating the situation where the smartphone's owner stands in between the smartphone itself and the anchor.

Therefore, we collected a dataset that represents four cases: Outdoor LoS, Indoor LoS, Outdoor NLoS, Indoor NLoS. Moreover, the whole dataset has been collected twice, one is for tuning the algorithm parameters (training set) and the other is for testing the performance (test set).

Each test data, which consists of the triplet (*ToF, RSSI, Frequency*), is collected for the same 1000 connection events.

[3] Vol6.B.2.1.2

[4] This value comes from: $\frac{200\ \mu s}{7.5\ ms}$, where $200\ \mu s$ is the time the timer is used and $7.5\ ms$ is the minimum connection interval of the BLE standard [23, Vol6.B.4.5.1].

If a connection events contains CRC errors, its triplet is replaced by the following one. At the end of the test, data is transferred to a PC as text file for further analysis.

VI. Results & Discussion

A. Physical Model

Once the data has been collected and stored in the PC, we used Matlab for visualizing the results and numerically compare ToF and RSSI performance.

First of all, we plotted the mean and the standard deviation with respect to the distance (Fig. 2 shows the RSSI and ToF data for the outdoor line-of-sight test). Then, we used the mean values at each distance to estimate the range based on the acquired data, therefore we trained two physical models that are supposed to represent the behavior of the two quantities. The models are necessary if we want to compare test data since the RSSI and ToF data is in two different measurement units (respectively dBm and ns) then we cannot directly compare the two results, thus we convert both in meters using the aforementioned models. In this paper we do not want to focus on the model itself, instead the purpose is to evaluate the quality of the acquired data in a fair way (that is the reason why we collected the RSSI and ToF for the same packets). In fact, regardless the used technique (fingerprinting or model-based), it is quite oblivious that the less noisy and more repeatable is the algorithm's input data, the more accurate will be the localization.

We used a log-normal model for the RSSI and a linear model for the ToF [8, 26, 27], which are described by the two equations:

$$d^{RSSI} = d_0^{RSSI} 10^{\frac{RSSI_0 - rssi}{10\alpha^{RSSI}}} \quad (2)$$

$$d^{ToF} = d_0^{ToF} + c^{ToF} * ToF \quad (3)$$

Eq: 2 describes the mapping from the $RSSI$ sample to distance, and Eq: 3 equivalently from ToF data to distance. The models parameters ($d_0^{RSSI}, RSSI_0, \alpha$ for Eq: 2 and c^{ToF}, d_0^{ToF} for Eq: 3) are trained with linear regression on the training set and the trained models are visible in Fig. 2 as red traces. The objective of the training is to minimize the sum of the squared error over the training set, therefore we will carry on our analysis comparing results using the RMSE (Root Mean Squared Error). In this regard we want to highlight that the RMSE value tend to be larger than the mean error value. In fact positive and negative errors will compensate each other when the mean is calculated, while they won't compensate in the RMSE calculation. For this we believe the RMSE is more relevant than mean error for evaluating ranging performance.

From Fig. 2 it is clear that the ToF measurements have a large standard deviation; this was expected and it is because the timer runs at $16\ MHz$, which means that for every clock cycle the radio waves travels approximately for $18.8\ m$. Since we are measuring the Two-Way ToF the distance to travel is doubled (back and forth) then every timer tick represent $9.4\ m$ of distance traveled by the radio waves. This $9.4\ m$ is hence the granularity of the distance estimation based on a single ToF sample. Anyway, since the two devices are not clock-cycle synchronized, the drift and the phase noise of the clocks adds a measurement noise that can be considered uncorrelated among different ToF samples. Therefore the clock noise act as dithering noise, which permits to increase the resolution of result over $9.4\ m$ if multiple samples are averaged.

Another important observation is that the slope of linear model used for ToF (red line in Fig. 2-(b)) should be the inverse of the speed-of-light per definition, but actually it is the inverse of a smaller value (approximately $180000\ \frac{km}{s}$). We still miss a satisfactory and complete explanation for this, but we believe it is a second order effect of reflections.

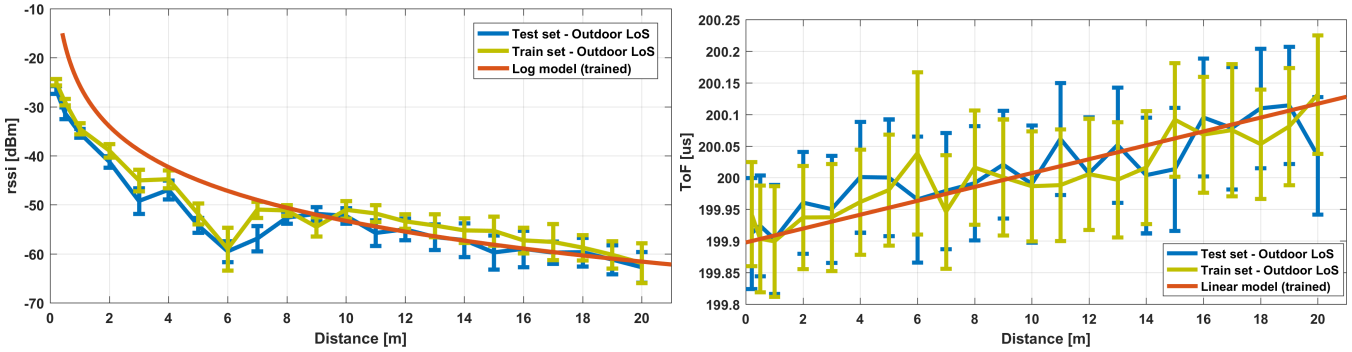
B. Models performance

With the purpose of investigating how the training condition impacts on the estimation error, we trained a different model for each training set (indoor/outdoor, LoS/NLoS) and then we tested them all on all test sets (note that the training and test sets have been collected on different days). Moreover, we created an outdoor generic model, which is trained with the merge of the outdoor LoS and the outdoor NLoS datasets, and we did the same also for the indoor condition. Recapping, we trained six different models (outdoor LoS, indoor LoS, outdoor NLoS, indoor NLoS, outdoor generic, indoor generic) and we tested them in the four cases (outdoor LoS, indoor LoS, outdoor NLoS, indoor NLoS). The resulting test error are reported in Table I where the mean error and RMSE (Root Mean Squared Error) is reported for all combinations of train and test.

The error values in this paper must be considered in the context of ranging and not of localization. In fact, for localizing a target on a plane, the distances from at least three known points are needed. It is clear that if the noise in the three distance estimations is not correlated, the localization error will be reduced with respect to the ranging error from one single beacon.

There are some other important observations regarding Table I:

- (almost) all models have their minimum error when the test data has been collected in the same conditions of train data; this was expected and it means that different datasets collected with the same conditions are similar (i.e. the experiment is repeatable)
- the RMS error obtained using ToF is always less than the one obtained with RSSI. This is highlighted also by the mean values (bottom line of Table I) that represent how a specific model behaves in the general case
- the impact of the LoS or NLoS condition is more severe on the RSSI data
- in some cases the RSSI error tends to diverge (almost up to $60\ m$) while the ToF error remains under control



(a) RSSI data.

(b) ToF data.

Fig. 2: Acquired data: RSSI values (a) and Time of flight (b), plotted with respect to nodes distance in the Outdoor LoS condition. The mean value over 1000 ToF samples is reported, error bars are one standard deviation high. The two colors depict the train and test dataset. The fitted model is in red.

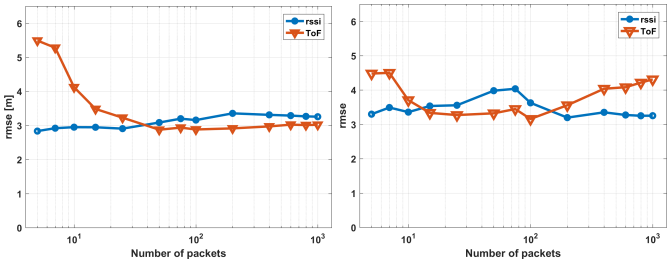
	RMSE	TRAIN CONDITION: outdoor LoS RSSI	outdoor LoS ToF	indoor LoS RSSI	indoor LoS ToF	outdoor NLoS RSSI	outdoor NLoS ToF	indoor NLoS RSSI	indoor NLoS ToF	outdoor generic RSSI	outdoor generic ToF	indoor generic generic RSSI	indoor generic generic ToF
TEST CONDITION	**outdoor LoS**	mean: -1.77 rmse: **3.26**	mean: -0.46 rmse: **3.02**	mean: -4.07 rmse: 4.97	mean: **1.55** rmse: 3.50	mean: 6.17 rmse: 7.65	mean: 3.66 rmse: 4.90	mean: 4.11 rmse: 5.30	mean: 5.19 rmse: 6.44	mean: 3.02 rmse: 4.51	mean: **1.87** rmse: **3.55**	mean: **-1.11** rmse: **4.23**	mean: 4.23 rmse: 5.39
	indoor LoS	mean: **0.57** rmse: 3.79	mean: **0.23** rmse: 3.35	mean: **-1.69** rmse: **4.01**	mean: 2.06 rmse: **3.48**	mean: 7.04 rmse: 8.80	mean: 3.89 rmse: 4.78	mean: 5.94 rmse: 7.58	mean: 5.37 rmse: 6.23	mean: 4.57 rmse: 6.43	mean: 2.20 rmse: 3.76	mean: 3.09 rmse: 4.85	mean: 4.39 rmse: 5.22
	outdoor NLoS	mean: -20.20 rmse: 20.92	mean: -6.00 rmse: 7.40	mean: -21.97 rmse: 22.49	mean: -3.01 rmse: 5.18	mean: **-1.03** rmse: **3.25**	mean: **-0.99** rmse: 4.31	mean: -14.37 rmse: 16.19	mean: 1.98 rmse: 4.76	mean: -9.62 rmse: 10.24	mean: -3.95 rmse: 5.89	mean: -50.84 rmse: 58.91	mean: **-0.32** rmse: 4.20
	indoor NLoS	mean: -6.49 rmse: 7.28	mean: -6.79 rmse: 8.70	mean: -9.09 rmse: 9.67	mean: -3.67 rmse: 5.45	mean: 4.79 rmse: 6.69	mean: -1.60 rmse: **4.09**	mean: **0.91** rmse: **3.64**	mean: **1.53** rmse: **2.89**	mean: **0.16** rmse: **3.85**	mean: -4.72 rmse: 7.21	mean: -8.83 rmse: 10.22	mean: -0.89 rmse: **3.66**
	average	mean: 1.30 rmse: 8.81	mean: 1.42 rmse: 5.62	mean: 1.20 rmse: 10.28	mean: 1.86 rmse: 4.40	mean: 5.31 rmse: 6.60	mean: 2.77 rmse: 4.52	mean: 3.60 rmse: 8.18	mean: 4.17 rmse: 5.08	mean: 2.88 rmse: 6.26	mean: 1.99 rmse: 5.10	mean: 3.09 rmse: 19.55	mean: 3.09 rmse: 4.62

TABLE I: Mean and RMSE (root mean squared error) values obtained training and testing the models with various combinations of the data sets. The minimum Mean and RMSE of RSSI and ToF in each column is highlighted in bold. Unit is m.

C. Dependency on the number of packets

As already mentioned, the resolution is increased by the means of averaging data over 1000 ToF samples, but this implies that an application would have a response latency that is at least the time needed to collect all the packets. Since we collect one sample per connection event and since, as per specification, the minimum time between connection events is 7.5 ms, the minimum expected latency is 7.5 s, which may be too high for some application. There are workarounds for reducing the overall latency without violating the 7.5 ms. Anyway it is important to analyze how the error changes as the amount of samples for the average is reduced.

For doing this we repeated the model testing, using the average of only the first N samples at each position for calculating the distance. We used this technique instead of random sampling over the 1000 samples for obtaining coherent results, since the frequency hopping pattern built in the BLE cannot be randomized. We kept all the 1000 samples for the training phase since it is supposed to be an off-line phase and the latency for this is not considered critical. Two representative results are reported in Fig. 3, which shows how the error changes with respect to the number of averaged samples. Both the errors for ToF and RSSI are reported; the data of the two figures refers to Outdoor LoS and NLoS conditions.

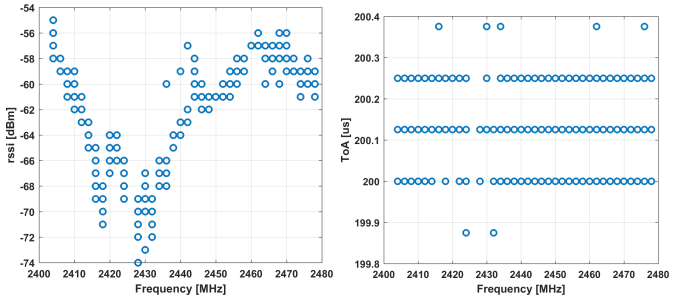


(a) Outdoor LoS

(b) Indoor LoS

Fig. 3: RMSE (Root mean squared error) in function of the number of packets averaged for the outdoor LoS (a) and NLoS (b). Note that X axis is log scale.

We repeated this analysis for all the cases we collected (the 24 combinations of training/testing in Table I) and the behavior is not always the same. In the majority of the cases (roughly 70 %) we obtained a benefit of at least 1 m on the RMSE after averaging, anyway for some case like outdoor NLoS condition reported in Fig. 3-(b), the RMSE is pretty constant after the average, no matter how many packets are considered. For the test cases where the average is effective we did not have relevant decrease of the error over 100 samples averaged, meaning that the time needed for collecting data can be reduced down to 0.75s without sensibly worsen the performance. Moreover the RSSI performance is almost unaffected



(a) RSSI (b) ToF

Fig. 4: Effect of channel hopping on raw (not averaged) RSSI and ToF data. Each figure contains 1000 data points, then each circle may represent multiple overlapped points.

by the amount of samples to average; this is explained by the measurement granularity. As said the single measurement of the ToF has a granularity of $\frac{1}{16\ MHz} = 62.5\ ns$, which is converted in $9.4\ m$. While the RSSI granularity is $1\ dB$; this is less straight forward to convert to meters, but using the model described by Eq.3, within the testing range ($0.2\ m$-$20\ m$), the estimation granularity spans approximatively from $7.5\ cm$ to $1.7\ m$. Now it is clear that, even in the worst case, the single sample granularity of RSSI is smaller than ToF; therefore adding samples to average is an effective technique for ToF data, while it is less effective for the RSSI data. These considerations can be taken into account when the localization latency is a critical aspect of the application.

D. Dependency on the channel

The last observation we want to bring to the reader is the effect of the channel hopping on the acquired data. In Fig. 4 the raw samples collected with the two nodes at $20\ m$ of distance in outdoor LoS conditions are plotted with respect to the used channel. It is immediate to see (Fig. 4-(a)) that the RSSI data is strongly affected by the channel. These results where expected from the two-ray model of propagation [10, 28], since during the outdoor test we had the reflections of the ground and also of a neighboring building. The reported figure has been chosen as representative but at the other distances the channel effect has a different behavior, which is dependent on the surrounding environment. At the same time the ToF shows more constant results (Fig. 4-(b)).

From this result, it is clear that using the RSSI for calculating the distance between two nodes without considering which channel each sample refers to is limiting. Modeling the effects of reflections at different frequencies and in the general case (without prior information on the environment) is a complex task. In contrast, including the channel information in a machine learning algorithm that automatically takes it into account does not add a relevant overhead. In particular, for the fingerprinting techniques giving more data to the algorithm is likely to give better estimation result. Unfortunately, the channel number information is not exposed through the standard Bluetooth API of the most popular mobile devices (Android/iOs smartphones) and also on many embedded platforms, then it is much more challenging for the developer to get that information.

E. Repeatability of the test

Although it is clear from the results that the collected ToF data, even if noisy, can be used as an indicator of distance even for narrow band protocols such as BLE, its usefulness may be questionable if the measure isn't repeatable over different beacon models or over different specimens of the same beacon. With this regard we did a preliminary evaluation testing both the nRF52840 and the nRF52832 with two different SDK (respectively softdevice 140 v5.0.0-2alpha and softdevice 132 v5.0.0) and also another SoC (Texas Instruments cc2650, with BLE stack v2.2) used as BLE beacon, the test has been repeated on multiple specimens of the selected SoC.

What we found is that choosing a different BLE SoC manufacturer has potentially the same impact to changing the SDK, and the effect is a small change in the offset of the measure. This offset should be nominally $T_{IFS} + T_{nbit} + T_D = 199.65\ us$ which equals to Eq:1 when the distance between the two devices is zero. As visible from the figures, obtained results are slightly different (the measured offset is few hundreds of nanosecond longer) and we justify this with the hardware/software implementation of the BLE stack. In fact we are dealing with time resolutions which are in the same order of magnitude of clock interval of the SoC, therefore the hardware/software architecture plays a fundamental role for this kind of measure. Unfortunately this can hardly be calculated analytically since most of the BLE SoC manufacturers give part of the SDK as pre-compiled binaries, then source code cannot be analyzed. Nevertheless the results are stable among specimens of the same device/SDK configuration, then it is reasonable to speculate that the offset can be a constant value statically associated with a specific configuration of SDK-SoC model and broadcasted as part of advertising packet payload, similarly to what already happen for the *calibrated Tx power* field in the Eddystone UID packet [29].

This evaluation is preliminary and is only qualitative, anyway it has been useful to verify that the experiment can be brought in the real world outside the small, controlled and homogeneous testbed we used.

VII. Conclusion

In this work, we investigated if RSSI is the only indicator that can be used for localization based on Bluetooth Low Energy or if ToF can be a viable additional one. Using low-cost off-the-self hardware, we have shown that with BLE the ToF can be measured if a sufficient amount of packets is averaged. We calculated the energy overhead due to the ToF measurement finding that in case the BLE connection is already established (e.g. for data transfer) the overhead for measuring the ToF is negligible. While, if the connection has to be established for the purpose of localization the average

current consumption is higher ($1\ mA$ to $1.5\ mA$), which corresponds to the consumption of a BLE central device sending and receiving empty packets.

Experiments showed that ToF and RSSI have comparable performance for distance estimation in the range $0 - 20\ m$; however, ToF is slightly better in terms of RMSE. ToF performance starts to degrade when the average is calculated on windows below 100 samples. While for RSSI, averaging over multiple samples does not have a particular effect, making it a good choice when the latency is a key point.

As expected, LoS/NLoS condition had an impact on distance estimation with RSSI; this should be taken into account when the device estimating its position is a Smartphone. In such case, the possible proximity to owner's body makes difficult to predict if beacons and smartphone are in LoS. At the same time, the LoS/NLoS has a lighter effect on distance estimation when the ToF is used for ranging.

Another kind of data that is often discarded for the RSSI based localization is the channel (or the career frequency). The Bluetooth standard is based on channel hopping for mitigating the effects of interference and fading, and in this work we have shown the high impact of channels on raw RSSI values. Therefore, to increase algorithms accuracy RSSI data should be always coupled with the channel where available. This observation may be considered by Bluetooth radio manufacturers or by Bluetooth SIG when defining APIs used for controlling the radio.

Results reported in this paper show that ToF alone is not fully reliable for localization; however, it furnishes valuable data at low price, useful to increase the accuracy of a model-based or fingerprint-based algorithm.

Recently, the new version of Bluetooth specification has been released. Version 5.0 includes some interesting features, such as the long range support (up to $1\ km$ in LoS), which is achieved using coded modulations. With such a kind of long range wireless communication, the RSSI based localization will lose reliability because of the logarithmic nature of the Log-normal model of propagation, therefore we expect ToF measurement on Bluetooth 5.0 to gain interest soon.

VIII. Acknowledgements

Work funded by PAT L.P. 6/1999 SpinRetail

References

[1] F. Palumbo, P. Barsocchi, S. Chessa, and J. C. Augusto, "A stigmergic approach to indoor localization using bluetooth low energy beacons," in *12th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Aug 2015, pp. 1–6.

[2] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, no. 5, p. 596, 2016.

[3] Estimote, https://estimote.com/, accessed: 2017-08-03.

[4] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Wireless rssi fingerprinting localization," *Signal Processing*, vol. 131, pp. 235 – 244, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168416301566

[5] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth indoor localization with multiple neural networks," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, May 2010, pp. 295–300.

[6] Nordic Semiconductor, http://blog.nordicsemi.com/getconnected/beacons-go-live-at-londons-gatwick-airport, accessed: 2017-09-01.

[7] J. A. Shaw, "Radiometry and the friis transmission equation," *American journal of physics*, vol. 81, no. 1, pp. 33–37, 2013.

[8] K. Heurtefeux and F. Valois, "Is rssi a good choice for localization in wireless sensor network?" in *IEEE 26th Int. Conf. on Advanced Information Networking and Applications*. IEEE, 2012, pp. 732–739.

[9] A. Awad, T. Frunzke, and F. Dressler, "Adaptive distance estimation and localization in wsn using rssi measures," in *Digital System Design Architectures, Methods and Tools . DSD 2007. 10th Euromicro Conf. on*. IEEE, 2007, pp. 471–478.

[10] C. Sommer and F. Dressler, "Using the right two-ray model? a measurement based evaluation of phy models in vanets," in *Proc. ACM MobiCom*, 2011, pp. 1–3.

[11] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[12] S. Kajioka, T. Mori, T. Uchiya, I. Takumi, and H. Matsuo, "Experiment of indoor position presumption based on rssi of bluetooth le beacon," in *Consumer Electronics (GCCE), IEEE 3rd Global Conf. on*. IEEE, 2014, pp. 337–339.

[13] H. J. P. Iglesias, V. Barral, and C. J. Escudero, "Indoor person localization system through rssi bluetooth fingerprinting," in *Systems, Signals and Image Processing (IWSSIP), 19th Int. Conf. on*. IEEE, 2012, pp. 40–43.

[14] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[15] Y. Jin, M. Motani, W.-S. Soh, and J. Zhang, "Sparsetrack: Enhancing indoor pedestrian tracking with sparse infrastructure support," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[16] IndoorAtlas, http://www.indooratlas.com/, accessed: 2017-08-03.

[17] D. Giovanelli, B. Milosevic, C. Kiraly, A. Murphy, and E. Farella, "Dynamic group management with bluetooth low energy," in *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE, 2016, pp. 1–6.

[18] J. Torres-Sospedra and A. Moreira, "Analysis of sources of large positioning errors in deterministic fingerprinting," *Sensors*, vol. 17, no. 12, p. 2736, 2017.

[19] S. Lanzisera, D. Zats, and K. S. Pister, "Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 837–845, 2011.

[20] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta, "Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization," in *14th ACM Conf. on Embedded Network Sensor Systems*. ACM, 2016, pp. 137–149.

[21] ABI Research, https://www.abiresearch.com/, accessed: 2017-09-22.

[22] Z. Yang, Y. Liu, and X.-Y. Li, "Beyond trilateration: On the localizability of wireless ad-hoc networks," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2392–2400.

[23] Bluetooth SIG, "Core version 4.2," https://www.bluetooth.com/specifications/bluetooth-core-specification, accessed: 2017-08-03.

[24] Davide Giovanelli, https://bitbucket.org/dgiovane/time_of_flight_ble/, accessed: 2018-07-06.

[25] Nordic Semiconductor, https://devzone.nordicsemi.com/power/, accessed: 2018-02-20.

[26] J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang, "Distance measurement model based on rssi in wsn," *Wireless Sensor Network*, vol. 2, no. 08, p. 606, 2010.

[27] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli, "A data fusion technique for wireless ranging performance improvement," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 1, pp. 27–37, 2013.

[28] J. C. Giacomin, L. H. Correia, T. Heimfarth, G. M. Pereira, V. F. Silva, and J. L. De Santana, "Radio channel model of wireless sensor networks operating in 2.4 ghz ism band," *INFOCOMP Journal of Computer Science*, vol. 9, no. 1, pp. 98–106, 2010.

[29] Google, https://github.com/google/eddystone, accessed: 2018-01-29.