



Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 1083 – 1088

Procedia
Computer Science

The 5th International Symposium on Internet of Ubiquitous and Pervasive Things (IUPPT)

Measuring a distance between Things with improved accuracy

Hosik Cho^{a,*}, Jianxun Ji^b, Zili Chen^b, Hyuncheol Park^a, Wonsuk Lee^a

^a*Software R&D Center, Samsung Electronics, 129, Samsung-ro, Yeongton-gu, Suwon-si, Gyeonggi-do 443-742, Korea*

^b*Samsung R&D China-Beijing, Samsung Electronics, 12A TaiYangGong Middle Road, Chaoyang District, Beijing 100028, China*

Abstract

This paper suggests a method to measure the physical distance between an IoT device and a mobile device (also a Thing) using BLE(Bluetooth Low-Energy profile) interfaces with smaller distance errors. BLE is a well-known technology for the low-power connectivity suitable for IoT devices and also for the proximity with the range of several meters. Apple has already adopted the technic and enhanced it to provide subdivided proximity range levels. But as it is also a variation of RSS-based distance estimation, iBeacon could only provide immediate, near or far status but not a real and accurate distance. To provide the distance using BLE, this paper introduces additional self-correcting beacon to calibrate the reference distance and mitigate errors from environmental factors. By adopting self-correcting beacon while measuring the distance, the average distance error showed less than 10% within the range of 1.5 meters. Some considerations are presented to extend the range to be able to get accurate distances.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Conference Program Chairs

Keywords: accurate distance, self-correcting beacon, RSS-based estimation, distance error mitigation;

1. Introduction

There are many Things already in our current environment at home, office, and streets. Some of them are mobile (smart phones, wearable devices, etc) while some of them are fixed (environmental sensors, appliances, smart TV, etc). The Things might have one or more network interfaces to be interconnected and the BLE could be the most popular technology as most smart phones have one each. The BLE is a suitable connectivity technology among the IoT devices from its nature of low-power operation. Above this, as the BLE uses smaller transmission power than classic Bluetooth, it could be applied to provide proximity between transmitter and receiver¹. Apple's iBeacon is the

* Corresponding author. Tel.: +82-10-9066-2101; fax: +82-31-279-1717.
E-mail address: hosik79.cho@samsung.com.

commercialized specification based on BLE, and it enhanced the BLE protocol to include *txPower* field when a BLE client broadcasts itself for the BLE server scanning or connecting². The server could compare the received signal strength and *txPower* value from the client, and estimate a rough distance between the client and the server. Currently, the iBeacon specification provides three levels of proximity from the estimated distance, which are immediate, near and far status. As the received signal strength would fluctuate with the time and the environment, and as the *txPower* value is determined and fixed by the vendor, it is difficult to provide the accurate physical distance. So, this paper is focusing on the way how to measure the distance between BLE client and server with improved accuracy. General RSS-based distance measuring methodologies will be presented in Section 2. The new system for accurate distance measurement by adopting the self-correcting beacons will be explained in Section 3. Section 4 will show the evaluation result of the proposed system and Section 5 will conclude the paper with further work to extend the range of accurate distance measurement.

2. Background and Motivation

Nowadays, as the popularity of IoT (like smart home) and smart devices like smart phone, smart TV, it becomes a very big demand to get the context of the user for the smart devices can behave differently based on it in IoT. And the location information is a very important part of the context information. So this paper aims to resolve the measuring distance problem to achieve the accurate position in IoT. If the smart devices could get the user's exact position information, their intelligence level will improve greatly, can do the jobs like turning on or off different lights automatically, displaying same contents on different screens, serving the host with water by the home robot and so on.

Currently measuring distances can be done by certain approaches. The following descriptions give some typical possible methods.

1) Time of arrival (TOA): TOA finds the distance between a transmitter and a receiver via a one way propagation time by exploiting the relationship between the light speed and the carrier frequency of a signal³. However, TOA positioning requires very accurate clocks because a 1.0 μ s error in timing equals to a 300 m error in distance estimate⁴. TOA is hard to popularize in the normal devices because the accurate clock will cause big costs. It cannot be used to resolve the actual measuring distances problems.

2) Angle of arrival (AOA): AOA is usually employed as prior-knowledge for the triangulation localization method⁵. So measuring angle is not fit for BLE on the normal devices either.

3) Ultrasound: A mobile node with an ultrasonic sensor measures the distance to a node by exploiting the ultrasonic signal propagation time. However, the transmission range of an ultrasound signal is small as it cannot propagate further than radio frequency wave⁶. It also adds size, cost, and energy supply to each device. Therefore even though ultrasound based localization approach can achieve high accuracy, it is not suitable for IoT environments.

Received signal strength (RSS) based distance estimation is a popular method in wireless sensor networks^{7,8}. Also RSS value of signal is very easy to be captured by the current device like cell phone, which means we do not need extra devices to implement it in the actual environment. As the wireless sensor network nodes are usually assumed to follow IEEE 802.11 or IEEE 802.15.4 standards, previous research would also be well applied to the case of BLE⁹. Table 1 shows the comparison of IEEE 802.11/802.15 series' wireless standards.

Table 1. Comparison of IEEE 802.11/802.15 wireless standards

IEEE wireless standards	Radio Frequency	Data Rate	Modulation & Coding
IEEE 802.11a	5 GHz	54 Mbps	PSK, QAM, OFDM
IEEE 802.11b	2.4 GHz	11 Mbps	PSK, CCK, DSSS
IEEE 802.11g	2.4 GHz	54 Mbps	PSK, QAM, OFDM
IEEE 802.15.1	2.4 GHz	3 Mbps	PSK, FSK, AFH
IEEE 802.15.4	868/915 MHz, 2.4 GHz	40 Kbps 250 Kbps	PSK, ASK, DSSS, PSSS

A radio signal transmitted from an antenna would be propagated through a space experiencing path losses. In this paper, we assume that the signal would follow the log-distance path loss model. The log-distance path loss model is a radio propagation model that predicts the path loss that a signal encounters inside a building or densely populated areas over distance. Log-distance path loss model is formally expressed as:

$$PL = P_{Tx} - P_{Rx} = PL_0 + 10 \cdot \gamma \cdot \log(d/d_0) + X_g \quad (1)$$

Where,

PL is the signal strength after total path loss at the distance d measured in Decibel,

P_{Tx} and P_{Rx} are the transmitted power and the received power respectively,

PL_0 is the signal strength after path loss at the reference distance d_0 measured in Decibel,

d is the length of the path,

d_0 is the reference distance,

γ is the path loss constant or exponent,

X_g is a normal random variable with zero mean reflecting the attenuation caused by flat fading.

In the iBeacon specification, the manufacturer should add $txPower$ value to existing BLE protocols. The $txPower$ value is the received power at the distance of 1 meter. Then, we can replace some variables with the value of $txPower$. When a receiver received a signal with $txPower$ field, the receiver can set;

d_0 to 1 meter,

PL_0 to $P_{Tx} - txPower$

Then, the expression could be

$$PL = P_{Tx} - P_{Rx} = P_{Tx} - txPower + 10 \cdot \gamma \cdot \log(d) + X_g \quad (2)$$

$$P_{Rx} = txPower - 10 \cdot \gamma \cdot \log(d) - X_g \quad (3)$$

The value γ and X_g could be found by empirical measurements. Android beacon library uses following coefficients to calculate distances in indoor environments¹⁰ and we also adopted the same equation.

$$d = (0.89976) * (P_{Rx}/txPower)^{7.7095} + 0.111 \quad (4)$$

Now, the variable is reduced into just two: P_{Rx} and $txPower$. As the $txPower$ value is fixed by the manufacturer, the fluctuation in received signal strength directly affects to the calculated distance. Even if we adopt some filtering algorithms, it is also hard to determine the exact distances.

3. System Design

To calibrate the distance and mitigate the errors, we proposed a self-correcting system by adding extra Thing and placing it on the reference distance. The system consisted of a target beacon, a measuring device and a self-correcting beacon. We want to calculate the accurate distance between the target beacon and the measuring device by installing the target beacon and self-correcting beacon to the fixed position with fixed distance. Fig. 1 shows the installation of the self-correcting system.

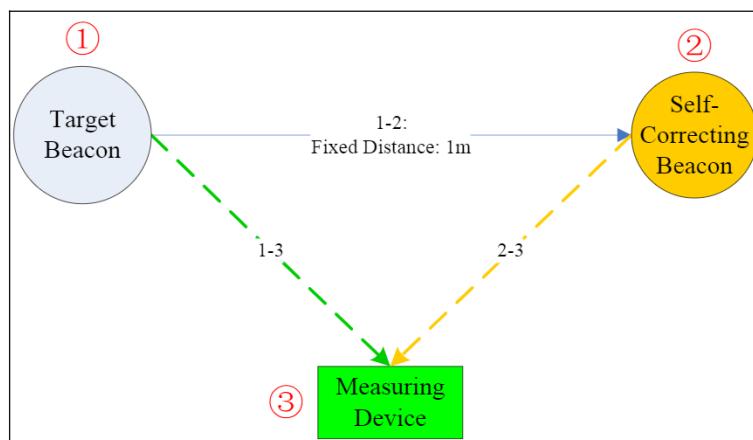


Fig. 1. The installation of the self-correcting system. The measuring device can get the RSS from the target beacon and the scPower from the self-correcting beacon with slight time differences. The measuring device calculates the distance (1-3) based on the scPower value and RSS of the target beacon.

The self-correcting beacon is an extra device to calculate more accurate distance through RSS in the measuring device. The self-correcting beacon receives the signal from the target beacon, and advertises the received signal strength of the target beacon (*scPower*). In the designed system, the measuring device now can utilize the signal strength from the target beacon, *txPower* from the target beacon, and *scPower* from the self-correcting beacon. In the conclusion of Section 2, the main problem is the fixed *txPower* which could not reflect the user's environment. By replacing the *txPower* with the *scPower*, the calculated distance can show more stable and accurate result.



Fig. 2. The system test environment. The test was performed in the office with soft partitions. The target beacon and the self-correcting beacon were fixed with one meter distance. The measuring device moved along with the virtual vertical line from the target beacon with the distance from 0.4 meters to 1.4 meters.

Fig. 2 shows the system test environments. We used an iBeacon-compatible BLE tag from Estimote as target beacon, the other smart phone as measuring device and emulated self-correcting beacon as a smart phone. We set the advertising interval time of the target beacon to 10ms for we can calculate the accurate distance in very short time and the self-correcting beacon manually set to know the MAC address of the target beacon.

4. Evaluations

Based on the previous description to calculate the accurate distance between the measuring device and target beacon, we should get the accurate *scPower* (real RSS of target beacon in 1m, received by self-correcting beacon) and RSS of target beacon (received by the measuring device). But we all know that the RSS of Beacon is always fluctuating because of Gaussian white noise and impact of the environment. So it is a big issue that the *scPower* and RSS of target beacon are both fluctuating, which will make the error of distance increased. But if they have the similar trend simultaneously, we can get more accurate result based on the previous formula (3). To prove this concept we did the following experiment: placing the measuring device 1.5m away from target beacon and correcting beacon 1m away from target beacon. Then we collected the RSS of the target beacon from the measuring device and correcting beacon separately for one minute. Fig. 3 shows the variation trend of (RSS of target beacon in 1.5m, time) and (*scPower*, time), in which the horizontal axis is time, the vertical axis is RSS, the blue line is RSS of target beacon and the red line is *scPower*. From the figure we can see that:

1. The RSS of target beacon is fluctuating;
2. The *scPower* is also fluctuating;
3. They do have the similar trend simultaneously, which means when the *scPower* becomes bigger the RSS of target beacon becomes bigger too.

But there are slight time lags between the RSS and the *scPower* as the self-correcting beacon will receive the RSS firstly from the target beacon and then send the value as *scPower*. As synchronizing the time between the two beacons is uneasy without any precise time module, we used the average value of RSS and *scPower* in 5 seconds

instead of using the real time synchronization. As described previously, we set the broadcasting interval of the beacons to 10ms, which means that the measuring device can collect 500 RSS values and 500 *scPower* values in 5 seconds. The averaged values of RSS and *scPower* also show the similar trend.

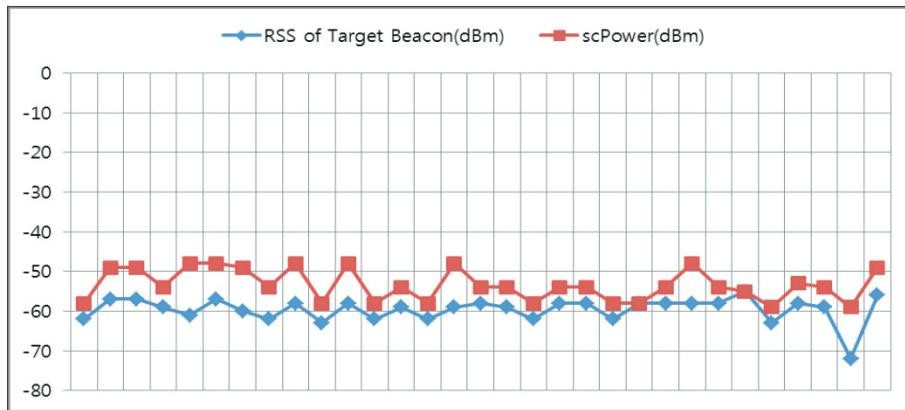


Fig. 3. Comparison of RSS from the target beacon which located in 1.5 meters away and scPower value from the self-correcting beacon which located in 1 meter away from the target beacon. The two values show the similar trend simultaneously according the time goes.

Then we evaluated the accuracy to measure the distances from the proposed self-correcting system. Fig. 4 shows the result comparison of with and without the self-correcting beacon while measuring distance at each reference distances (0.4m, 0.6m, 0.8m, 1.0m, 1.2m and 1.4m). Without the self-correcting beacon, the distance error shows up to 60.3%, in average 46.3%. With the self-correcting beacon, the distance error shows up to 8.1%, in average 4.7%. All the distance errors are the average of the gap between the real distances and the estimated distances.

Table 2. Distance errors of with and without the self-correcting beacon on several reference distances.

Actual distance (meter)	Distance errors with the self-correcting beacon (%)	Distance errors without the self-correcting beacon (%)
0.4	8.1	41.7
0.6	2.7	45.6
0.8	5.0	46.7
1.0	5.3	60.3
1.2	3.1	42.0
1.4	3.7	47.4
Average	4.7	46.3

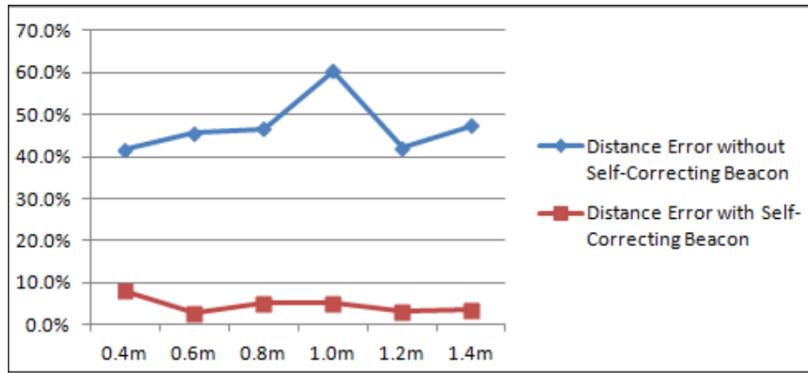


Fig. 4. Distance errors comparison of with and without the self-correcting beacon on several reference distances. It is possible to achieve accurate distances with under 10% distance error when we adopt the self-correcting beacon within 1.5 meters range.

5. Conclusion

In this paper, we propose an accurate distance measurement system between Things having BLE interfaces by adopting a self-correcting beacon. As the system adjusts the white noises and the environmental factors in real time, it can estimate the distances with under 10% distance error within 1.5 meters range of coverage. We also conducted an evaluation for the targets farther than 1.5 meters, but as the distance increases, the distance errors also increased dramatically. In indoor environments, there exists additional signal attenuation errors caused by the multipath signals and it will affect more for long distances than short distances. To extend the coverage of the accurate distance measurement, we are trying to apply multiple model filtering algorithms to track a single target in wireless sensor networks. We also hope the multiple model filtering could help to mitigate the additional errors for longer distances and to give more accurate distance measurements.

Acknowledgements

This work was supported in part by the Geo-spatial Intelligence project of Intelligence Solution Team, Software R&D Center, Samsung Electronics.

References

1. Bluetooth Specification Version 4.2, <https://www.bluetooth.org/en-us/specification/adopted-specifications>.
2. Andy Cavallini, iBeacon Bible 2.0, <https://meetingofideas.files.wordpress.com/2014/06/ibeacon-bible-2-0.pdf>.
3. Huseyin Akcan, Vassil Kriakov, Herve Brönnimann, and Alex Delis, "GPS-Free node localization in mobile wireless sensor networks", Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access, 2006.
4. I. Jami, M. Ali, R. F. Ormondroyd, "Comparison of Methods of Locating and Tracking Cellular Mobiles", IEE Colloquium on Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications, 1999.
5. Xu Huang, M. Barralet, and D. Sharma, "Behaviors of antenna polarization for RSSI location identification", International Conference on Networks Security, Wireless Communications and Trusted Computing, April 2009.
6. Woo-Yong Lee, Kyeong Hur, and Doo-Seop Eom, "Navigation of mobile node in wireless sensor networks without localization", IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, August 2008.
7. Praveen Kumar, Lohith Reddy and Shirshu Varma, "Distance measurement and error estimation scheme for RSSI based localization in Wireless Sensor Networks," Wireless Communication and Sensor Networks (WCSN), 2009.
8. Qian Dong and Waltenequs Dargie, "Evaluation of the reliability of RSSI for Indoor Localization," Wireless Communications in Unusual and Confined Areas (ICWCUC), 2012.
9. Jan Magne Tjensvold, "Comparison of the IEEE 802.11, 802.15.1, 802.15.4 and 802.15.6 wireless standards," September 2007.
10. android-beacon-library, <https://github.com/AltBeacon/android-beacon-library>.