Occupancy Detection via iBeacon on Android Devices for Smart Building Management

A. Corna, L. Fontana, A. A. Nacci and D. Sciuto
Politecnico di Milano

{andrea.corna, lorenzo2.fontana}@mail.polimi.it - {alessandro.nacci, donatella.sciuto}@polimi.it

Abstract— **Building heating, ventilation, and air conditioning (HVAC) systems are considered to be the main target for energy reduction due to their significant contribution to commercial buildings' energy consumption. Knowing a building's occupancy plays a crucial role in implementing demand-response HVAC.**
In this paper we propose a new solution based on the iBeacon technology. This solution is different from the previous ones because it leverages on the Bluetooth Low Energy standard, which provides lower power consumption. Moreover, the iBeacon protocol can be used both on iOs systems and Android ones, making this new approach portable. Differently from our previous work based on iOS devices, in this paper we focus on an Android based solution with the aim of increasing the accuracy of the location and the energy efficiency of the entire system. We increased the accuracy by 10% and the energy efficiency by 15%.

Keywords—smart buildings, indoor location, iBeacon, energy efficiency

I. Introduction

Smart buildings, places where sensors and actuators make the location more intelligent, are becoming more and more relevant and they are the natural evolution of today's constructions, especially for increasing the user comfort and safety and to obtain a more efficient consumption of the energy [7]. In such context, it is fundamental to know the position of the occupants within the building. For instance, in this way, it is possible to avoid energy wastes using the HVAC system only when needed. Another possible use case that benefits of the occupancy information is the efficient management of the lighting system; within a smart building that is aware of the user position, it is possible to turn on and off the lights according to the actual needs, increasing the building efficiency.

As discusses in a prior work we have done [8], many different works [10] have tackled the problem of deriving the building occupancy status using many different technologies (infrared sensors [9], RFID [10, 11] ultrasound pulses [12], GSM [13], WiFi [14,15,16,17,18] and standard Bluetooth [19, 20, 21]) However, despite numerous research works have been conducted to find a cheap, simple, power-efficient and reliable solution to this issue, the problem is still open and an optimal solution, satisfying all the three constraints, has still to be found. With this work, we propose the Apple iBeacon technology [1] as a possible solution to detect the number of users in a room, and how it can be used to gather information about their movements (thus identifying and tracking them) inside the building even if it has not been developed to solve the occupancy detection problem but to enable the design of indoor proximity systems. In a previous work [7], we made a similar study using Apple mobile devices (iPhone and iPad). Differently, here we want to port and improve the same methodology on Android devices, since they represent a huge part of the smartphones market. As it will shown in the next section, the porting of such technology on Android devices is challenging due to some restrictions of the underlying operating system. With respect to the previous work, we increased the accuracy of the classification algorithm we use for the occupancy information from 80% to 90%.

II. The Proposed Occupancy Detection System

As said in the previous section, with this work we analyzed the possibility of using the iBeacon protocol to implement a reliable and low cost occupancy detection system. The iBeacon technology [1] is based over the Bluetooth Low Energy (BLE) [22]. Its protocol contemplates two main components: transmitters with a *Universally Unique Identifier* (UUID) and receivers periodically scanning signals in the air in order to detect particular iBeacon packets. The iBeacon protocol allows the implementation of two main functionalities: region monitoring and ranging. The ***monitoring*** notifies a listener application every time a receiver gets close to a transmitter with a specific UUID. The ***ranging*** provides an approximation of the distance from the iBeacon transmitter using the information of the TX Power field. The philosophy behind this system is quite straightforward: we envision the users with their smartphones (or smart things in general) within a smart building that is instrumented with low cost Bluetooth 4.0 antennas. When a user enter a room that is iBeacon enabled, the room advertise itself to the user; consequently, the user smart-device detects the advertisement and sends this information to the Building Management System (BMS) [23].

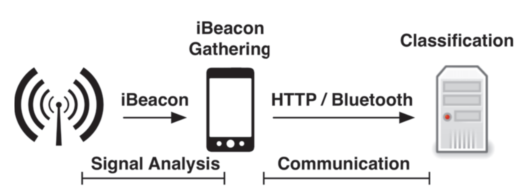

Figure 1. Main aspects of the proposed solution

To implement such functionalities, we have created an architecture composed of three main components: the (1)

beacon transmitters, devices within the rooms sending uniquely identified iBeacon packets to a (2) ***client mobile application*** installed on the occupants smartphones; this app is able to detect beacons produced by the building and sends this information to a (3) ***building remote server*** (the BMS) through an HTTP request or a Bluetooth connection. On this server, some classification algorithms are in charge of extrapolating the occupancy data from the detected packet information. Figure 1 shows the aforementioned architecture that highlights two aspects: the need for an accurate signal analysis and the need for an energy efficient communication between the devices and the remote server [1].

III. Signal Analysis

Similar to other high frequency signals that are transmitted through air, Bluetooth is affected by humidity, presence of other signals and many other environmental factors [2]. Therefore, different tests have been performed to evaluate the fluctuation of the signal received. Tests consisted in positioning the device at a given distance D from the transmitter, after a suitable calibration, and registering the detected signals. Figure 2 shows the recorded values detected with $D = 2\ mt$ with a Samsung S3 mini.

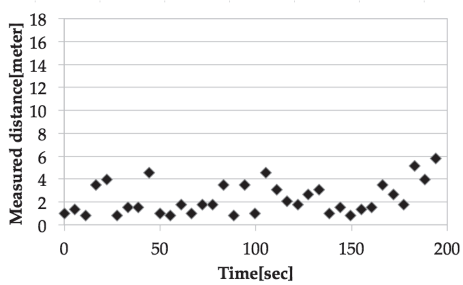

Figure 2. Signal Evaluation with 5 seconds scan period

It can be observed that there is a large variability of the estimated distance between the transmitter and the Android based receiver. This lack of accuracy is caused also by a limit of the Android operating system since its BLE APIs allows only a single signal strength measurement per scan[2], differently from iOS where it is possible to get many measurements for each broadcast advertisement by the transmitter.

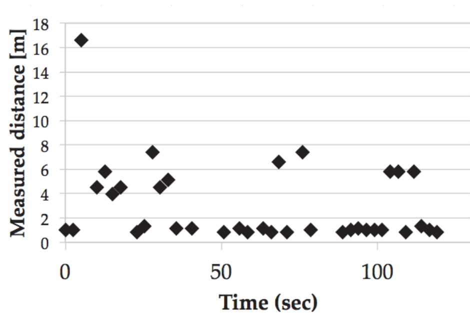

Figure 3. Signals collected with 2 seconds scan period

To understand this concept, let us consider an example: having a scan period of two seconds and an iBeacon generator that transmits thirty times per second, an Android device that scans for ten seconds gets only five samples (despite the rate of the transmitter the device will get one sample per scan and so ten divided by two samples). On the contrary, an iOS device receives three hundred samples because inside each scan it can collect more than one sample. As a consequence, the iOS distance measurements result to be more accurate, since it is allowed to work on a higher number of recorded data. Another important problem we faced during the implementation of our prototype system is that the adapter sometimes looses some samples due to bugs in the software stack. In order to cope with the aforementioned problems, we increased the scan period to collect more sample obtaining more accurate distance estimations (Figure 3).

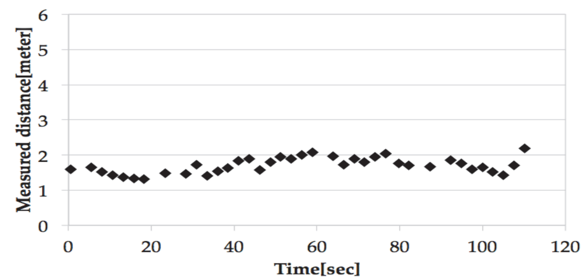

Figure 4. Signal static evaluation (Coeff = 0.65)

Unfortunately, increasing the scan period, the estimation phase takes a longer time, causing the application to be less reactive to distance changes by the user. In order to obtain a low latency but good distance estimation at the same time, we implemented a custom distance estimation algorithm. With the proposed algorithm, we can solve the problem of beacons' losses since we remove the beacon information only after the second consecutive loss, otherwise its value is maintained. Moreover, we solve also the problem of the fluctuation of the signal since we consider that if at a certain time T the device is in a position P, at time T+1 the position will be P+ΔP, and ΔP depends on the speed and on the time interval. With these assumptions, the older position has a role in determining which will be the current one that can be estimated as follows:

$$p_i = p_{i-1} * coefficient + (1 - coefficient) * v_i$$

where p_i is the result of the computation of the value related to a single beacon, p_{i-1} the value of the signal history and v_i the new measurement. So the older position will influence the current one with a given probability, the next one with a lower probability and so on.

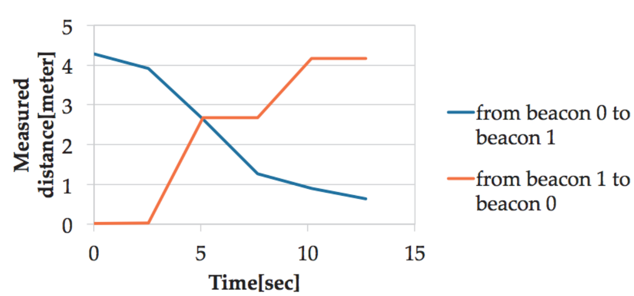

Figure 5. Signal dynamic evaluation (Coeff = 0.65)

1) All the implementation details and the source code are available online: http://energybox.necst.it/bluesentinel

2) The scan period is the time used to collect samples for estimating the distance

Increasing the coefficient makes the signal more stable and less affected by peaks but on the other hand it becomes less responsive to movements. To determine the best trade-off for this coefficient some dynamic tests have been performed by moving the device from one transmitter to another at a speed of 1 - 1.5 m/s and registering the responsiveness to the fluctuation of the signal. After some parameters tuning we found that 0.65 is a good trade off between stability and responsiveness as shown in Figure 4 and Figure 5.

IV. ALGORITHMS FOR INDOOR OCCUPANCY

Given the information provided by the transmitter after the signal analysis, it is necessary to determine the position of the user. However, it is not easy to model the radio propagation in indoor places because of the different factors that can affect the signal [2]. Previous studies [3] have developed different algorithms to estimate the position of a user through a set of information; they can be classified in 3 main categories: *Triangulation*, *Proximity* and *Scene Analysis*. Triangulation has been discarded because it requires very stable and accurate input data [10] and due to the signal fluctuation we decided to not use this technique. In our previous work [8] we used the *Proximity Technique*; this technique uses the strongest signal received from a grid of transmitters, each of which associated with a particular location, in order to determine the position of the user. The results of our first work were encouraging (we reached an accuracy of the 84%) but in this paper we try to increase also the classification accuracy. For this reason, we propose the adoption of the ***Scene Analysis*** technique [10]: it is a pattern recognition method that uses the characteristics of the location to make a classification. More in detail, the approach compares the observed characteristics to pre-stored characteristics for each pattern to determine a match. In our implementation, the relevant feature considered is the detected distance from the different iBeacon transmitters inside the room. First, a data collection phase is needed, requiring an operator that walks around the building collecting samples (beacon identifiers and their detected distances). These samples are then associated with the specific room and sent to the server that stores them in the database. After this phase the server creates a supervised machine-learning model based on all the samples. When a user enters the building the application will send to the server the list of all the beacons detected at a certain instant and their respective distances.

Number of samples	561
Correct classified samples	529
Incorrect classified samlpes	32
Accuracy score	0.9428
Min absolute errore score	0.6576
Min squared error	0.6576

(a)

True Positive Rate	84.1%
False Positive Rate	20%
Precision	94%
Recall	94%
Fi-score	94%
Support	561

(b)

Outside	Inside
227	19
13	292

(c)

Figure 6. Experimental results

The server using the pre-computed model can estimate the user's location. Our implementation used Support Vector Machines (SVM) [5] with the Radial Basis Function kernel, as suggested by [4]. To test the accuracy of this solution we have created a testing application and we asked a user to move within a house and to indicate its actual location.

Part of the collected data was then used to build the aforementioned SVM model (*training set*), while another part was used to test its behaviors (*testing set*). As result we have obtained an accuracy of about the 94% (Figure 6), increasing the accuracy of about 10% from previous work. From the confusion matrix (Figure 6.c) the number of false positive, detection of the user inside the room while he was outside is slightly higher than the number of false negative, detection of the user outside the room while he was inside, is about the same. This result is good since it is better to have false positive than a false negative because false negatives are a problem in terms of user comfort and safety.

V. MOBILE DEVICE ENERGY CONSUMPTION AND COMMUNICATION INFRASTRUCTURE

In Section IV, we anticipated that we envisioned two different ways to send the beacons received by the smartphones to the building server: a first one based on Wi-Fi and a second one based on Bluetooth. In our previous work with iOS devices, we discovered that an architecture based on the Wi-Fi protocol is very expensive from the energy consumption point of view [8]. As known, having energy efficient applications is crucial on mobile devices since the battery is a very limited resource [24]. For this reason, we focused our attention also on the measurement of the energy consumption caused by our app on the Android device. In this work, we performed the measurements with the Wi-Fi communication channel (the same used on iOS) and also with an alternative channel based on Bluetooth.

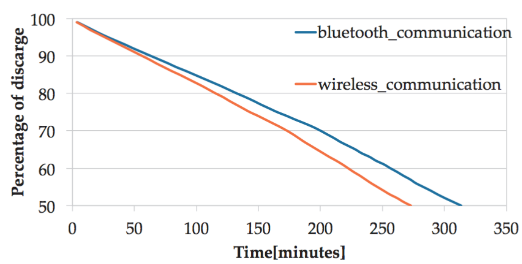

Figure 7. Consumption with http communication

In this last case, a Bluetooth connection is established between the smart device and the beacon transmitter when a beacon is received. To develop this second solution we have created a Bluetooth server in the iBeacon transmitter (that is thought to be not-battery based) that retransmits the information received to the central server using HTTP requests. Implementing this new Bluetooth based solution, we discovered that both implementations have pros and cons: the Wi-Fi is more reliable and stable but forces to keep on the wireless adapter that has a high power consumption. On the other hand, the Bluetooth one is more energy efficient, but it's less stable than the Wi-Fi solution due to bugs in the BLE Android API. In order to understand the energy consumption of our system, we measured the energy consumption of our app using the ***MPower App*** application [24] we have developed. This application is a background service that logs the battery status is a very energy efficient way in order influence the least possible the battery behavior and it is able to model the energy profile of a device. Figure 7 shows the average of 10 measurements performed on a Samsung Galaxy S3 Mini with Android 4.1. As expected, the Wi-Fi solution is more expensive in terms of energy consumption compared to the second one. Using the Bluetooth based architecture we obtained an energy saving of the 15%. As a drawback, in order to support the Bluetooth architecture, a more complicated antenna board is required. As last consideration, with our app installed, the battery lifetime of the mobile device is around 10 hours.

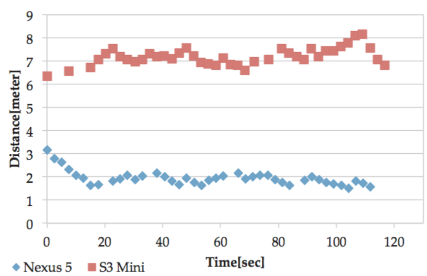

Figure 8. Differences in the received signal strengths

We underline also that the strength of the signal received from an iBeacon antenna, considering the same transmitter and the same distance, changes significantly between different devices. Figure 8 shows an example of two smartphones, a Nexus 5 and S3 mini, positioned at the same distance. A possible solution to this problem might be to collect experimental information on the power strength received by different devices and using them to tune the information that is provided to the server during the setup phase.

VI. Concluding remarks and future developments

With this work we aimed at evaluating the possibility of using iBeacon on Android devices as a suitable technology for the occupancy detection in a smart building. The paper has shown the major challenges in using such technology on the proposed architecture: a big effort has been put in the signal stabilization, on the classification algorithms and on the energy efficiency on the mobile device used to sense the environment. On the classification algorithms side, we have increased the accuracy from 84% to 94%. On the application energy efficiency, proposing an alternative communication pattern via Bluetooth, we obtained a 15% improvement. We believe this is a good starting point for further developments on the different components of the proposed solution. In particular, signal accuracy is variable, but this would require a modification to the Android kernel to provide more samples and achieve the same level of accuracy of the iOS devices. Google announced the release of Android L OS by the end of September, that promises to correct some of the bugs related to Bluetooth present in Android 4.4 and permits to generate beacon packets from the device [6]. With this new support more solutions become possible, with an improvement of the information provided by the devices.

VII. Aknowledgments

This work was partially supported by the Joint Open Lab "S-Cube" - Telecom Italia S.p.A. - Innovation division, Italy.

References

[1] Apple: iBeacon for developers: https://developer.apple.com/ibeacon/

[2] Enterprises, L.: Make your wi-fi hi-fi: The "truth" about wireless signal interference

[3] Liu, H.L.H.D.P.B.J.: Survey of wireless indoor positioning techniques and systems. Technical report, IEEE (2007)

[4] Bolliger, P.: Redpin adaptive, zero configuration in- door localization through user collaboration. Tech- nical report, Institute for Pervasive Computing ETH Zurich, Switzerland (2008)

[5] Laura Auria, R.A.M.: Support vector machines (svm) as a technique for solvency analysis. Technical report, Deutsche Bundesbank, Hannover; German Institute for Economic Research, Berlin. (2007)

[6] Networks, R.: Making an iBeacon with android l

[7] Buildings energy data book. Technical report, US Department of En- ergy, August 2012.

[8] G. Conte, M. De Marchi, A. A. Nacci, V. Rana, D. Sciuto: BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system; 1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings (BuildSys) 2014

[9] WANT, Roy, et al. The active badge location system. ACM Transactions on Information Systems (TOIS), 1992, 10.1: 91-102.

[10] HAHNEL, Dirk, et al. Mapping and localization with RFID technology. In: Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. IEEE, 2004. p. 1015-1020.

[11] NI, Lionel M., et al. LANDMARC: indoor location sensing using active RFID. Wireless networks, 2004, 10.6: 701-710.

[12] WARD, Andy; JONES, Alan; HOPPER, Andy. A new location technique for the active office. Personal Communications, IEEE, 1997, 4.5: 42-47.

[13] OTSASON, Veljo, et al. Accurate GSM indoor localization. In: UbiComp 2005: Ubiquitous Computing. Springer Berlin Heidelberg, 2005. p. 141-158.

[14] JIANG, Yifei, et al. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM, 2012. p. 441-450.

[15] BOLLIGER, Philipp. 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, 2008. p. 55-60.

[16] MELFI, Ryan, et al. Measuring building occupancy using existing network infrastructure. In: Green Computing Conference and Workshops (IGCC), 2011 International. IEEE, 2011. p. 1-8.

[17] BALAJI, Bharathan, et al. Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems. ACM, 2013. p. 17.

[18] NGUYEN, Nam Tuan; ZHENG, Rong; HAN, Zhu. UMLI: An unsupervised mobile locations extraction approach with incomplete data. In: Wireless Communications and Networking Conference (WCNC), 2013 IEEE. IEEE, 2013. p. 2119-2124.

[19] PEI, Ling, et al. Inquiry-based bluetooth indoor positioning via rssi probability distributions. In: Advances in Satellite and Space Communications (SPACOMM), 2010 Second International Conference on. IEEE, 2010. p. 151-156.

[20] ANASTASI, Giuseppe, et al. Experimenting an indoor bluetooth-based positioning service. In: Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on. IEEE, 2003.

[21] Jin-Shyan Lee; Yu-Wei Su; Chung-Chou Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE , vol., no., pp.46,51, 5-8 Nov. 2007

[22] Bluetooth Smart - bluetooth.com/Pages/Bluetooth-Smart.aspx

[23] A. A. Nacci, V. Rana, D. Sciuto: A Perspective Vision on Complex Residential Building Management Systems, The 12th IEEE International Conference on Embedded and Ubiquitous Computing (EUC) 2014

[24] A. A. Nacci, F. Trovo', F. Maggi, M. Ferroni, A. Cazzola, D. Sciuto, M. D. Santambrogio: Adaptive and Flexible Smartphone Power Modeling, Mobile Networks and Applications, MONET, Springeì