

Indoor Position System Based on a Zigbee Network

A. Verónica Medina, José A. Gómez, José A. Ribeiro, and Enrique Dorrzoro

Grupo TAIS

Escuela Técnica Superior de Ingeniería Informática

Dpto. de Tecnología Electrónica

Universidad de Sevilla

Avda. Reina Mercedes s/n, 41014 Sevilla, Spain

{vmedina,jgomezdte,josriblim}@us.es, enriquez@dte.us.es

<http://matrix.dte.us.es/grupotais/>

Abstract. TAIS group has developed an indoor position system prototype based on a fingerprint positioning algorithm. The prototype uses IEEE 802.15.4 mote and BitCloud Stack, a full-featured ZigBee Compliant, second generation embedded software stack from Atmel. The design requirements of the prototype were only to determine the actual position in a room of a user in a building, so the prototype accuracy is room accuracy. TAIS group decided to compete in the second edition of EvAAL Competition. This paper presents all the step made to adapt the prototype to the EvAAL environment, the found drawbacks and the obtained results. One of the most important drawback was that the Smart House Living Lab of the Polytechnic University of Madrid has only two rooms, the required accuracy was meters (error less than or equal to 0,5 meters the higher score, higher than 4 meters no score) and the room accuracy was substituted by areas of interest so the behavior of our prototype was going to work was an incognita.

Keywords: IEEE 802.15.4, RSSI, Centroid, Indoor position, ZigBee, WSN.

1 Introduction

Wireless Sensor Networks (WSNs) are present in many applications for solving data acquisition process in researching fields like Ambient Assisted Living [1][2][3][4] or Smart building [5][6][7][8][9]. Depending on its applications, ambient or user sensors and actuators can be used for making decisions. The knowledge of a subject's position is very useful in these kinds of systems because depending on it the decisions to be made are different.

As stated in [11][12], an amount of indoor location tracking systems have been proposed in the literature, based on Radio Frequency (RF) signals, ultrasound, infrared, or some combination of modalities. One of the most popular is using RF signal strength to determine the location of a mobile node applying different strategies. One of those is to use empirical measurements of received radio signals, known as RSSI, Receiver Signal Strength Indicator, to estimate location. By

recording a database of radio “signatures” along with their known locations, a mobile node position can be estimated by acquiring the actual signature and comparing it to the known signatures in the database, also known as fingerprints [10]. There is an intrinsic error in the use of RSS for localization purposes as stated in [16].

We have developed [13] a similar system to the one presented in [12], called MoteTrack, which localized a subject in a room of a building. The prototype uses different motes, Meshnetics’ ones, with other RCB (microcontroller and transceiver) and, also, different software, the BitCloud Stack¹ [15], a ZigBee PRO certified platform [14], an application layer stack, not a MAC (Medium Access Control) layer stack. This prototype was used in a research project² that tried to make an Intelligent Building. The Building had to adapt the environment to make it users feel comfortable by controlling air-conditioning, music, etc. The users of the building had to carry a mote (the mobile mote) and the decision maker software informed the actuator software to change the environments as user requirements using the output of our system (estimated position and sensors information) and other parameters they estimated. The focus of that research project was only an accuracy of room positioning.

In section 2 an overview of the prototype is introduced. Section 3 shows how the prototype is implemented and how the data are collected. The deployment and adaptation to EvAAL is presented in section 4. The results of the EvAAL test in the Living Lab in Madrid is shown in section 5. Finally conclusions are established in section 6.

2 Prototype Overview

In our prototype, a building or other area is populated with a number of Meshnetics’ motes acting as fixed nodes, one of them being the coordinator, C, and a set of mobile nodes, the ones whose position is going to be determined. Each fixed node sends to C periodic beacon messages, beacon 2, which consist of an n-tuple of the format {MobileID, RSSI}, where n is the number of mobile nodes, MobileID is a unique identifier of a mobile node, and RSSI is the signal strength which each fixed node received the last beacon message sent by MobileID node. The beacon message sent by a mobile node is different from the one sent by a fixed node, to differ one from others, the mobile node beacon messages are called beacon 1. Not all fixed motes receive beacon 1 messages, this depend on the coverage area. In this case they send a beacon 2 a zero value in RSSI.

The location estimation of the mobile nodes consists of a two-phase process: an offline collection of reference signatures followed by an online location estimation.

¹ Atmel acquires MeshNetics’ ZigBee Intellectual Properties

² Health Intelligent Technologies Oriented to Health and comfort in Interior Environments (TECNO-CAI) approved project at the fifth call of CENIT program by the Innovation Science Ministry of Spain (CDTI and Ingenio 2010 Program).

2.1 Offline Collection

As in other signature-based systems, the reference signature database is acquired manually by a user with a mobile node and a PC connected to C. Each reference signature, shown as black dots in Figure 1, consists of a set of signature tuples of the form $\{\text{sourceID}, \text{meanRSSI}\}$, where sourceID is the fixed node ID and meanRSSI is the mean RSSI of a set of beacon messages received over some time interval. Each signature is mapped to a known location by the user acquiring the signature database (P1-P5 in Figure 1).

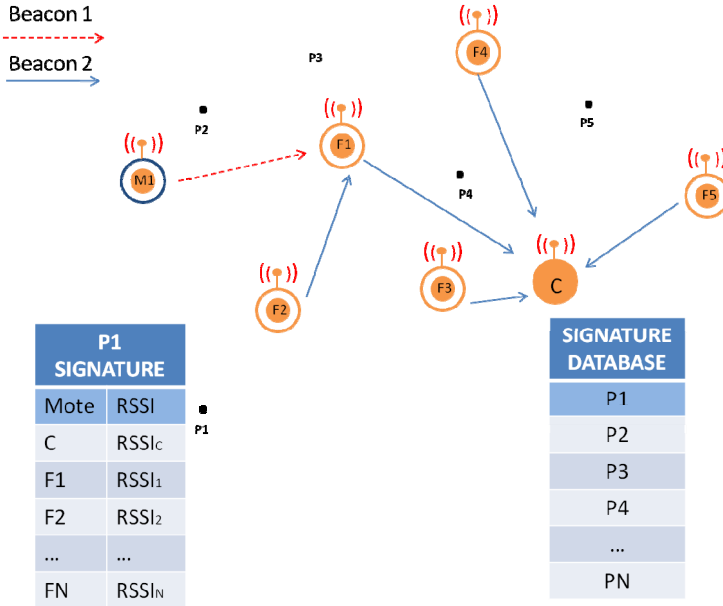


Fig. 1. System Overview. M1 is a mobile node, F1-F5 are fixed nodes, and C is the coordinator, also a fixed node. M1 periodically sends a beacon message, beacon 1, to inform the others nodes that is present. Each fixed node that receives beacon 1, saves the RSSI of that message in a table. Fixed node periodically sends a message to C, beacon 2, to inform about the RSSI that they have received from mobiles node, M1 in this case.

2.2 Location Estimation

Given a mobile node's received signature, s , received from the fixed nodes, and the reference signature set R , the mobile node's location can be estimated as follows. The first step is to compute the signature distances, from s to each reference signature $r_i \in R$. We employ the Manhattan distance metric,

$$M(r, s) = \sum_{t \in T} |RSSI(t)r - RSSI(t)s| \quad (1)$$

where T is the set of signature tuples presented in both signature, $RSSI(i)r$ is the RSSI value in the signature appearing in signature r_i , and $RSSI(i)s$ is the RSSI value in the signature appearing in signature s .

Given the set of signature distances, the location of a mobile node can be calculated in several ways. We consider the centroid of the set of signatures within some ratio of the nearest reference signature. Given a signature s , a set of reference signatures R , and the nearest signature $r^* = \text{argmin}_{r \in R} M(r, s)$, we select all reference signatures $r \in R$ that satisfy

$$\frac{M(r, s)}{M(r^*, s)} < c \quad (2)$$

for some constant c , empirically-determined. The geographic centroid of the locations of this subset of reference signatures is then taken as the mobile node's position. Small values of c work well, generally between 1.1 and 1.2. If $c=1$ the position estimation is the position of the nearest signature saved in the signature database.

3 Implementation and Data Collection

Our system is implemented by using Meshnetics' motes, Meshbean development board. We have used those motes because they have leds, buttons, additional sensors and other sensors can easily be connected to them for the purpose applications of this indoor position system, ambient living and smart buildings, so for prototyping works quite well. They also have a USART (Universal Synchronous/Asynchronous Receiver Transmitter) accessible by a USB connector, so a PC can be connected via a USB port, emulating a COM port, for both programming and receiving information, in our case beacons and sensor values.

A Meshnetics' mote is shown in Figure 2. This mote can have an integrated PCB or an external antenna. This affects only the range of coverage. In the prototype we use the one with external antenna. This mote has a MCU (MicroController Unit) wireless, called ZigBit, a compact 802.15.4/ZigBee module. It integrates both the ATmega1281 microcontroller and AT86RF212 transceiver of ATMEL so the AVR tools are necessary for programming purposes.

The BitCloud Stack has been used for software implementation. A full-featured, next generation embedded software stack from Atmel. BitCloud is fully compliant with ZigBee® PRO and ZigBee standards for wireless sensing and control. It provides an augmented set of APIs which, while maintaining compliance with the standard, offer extended functionality designed.

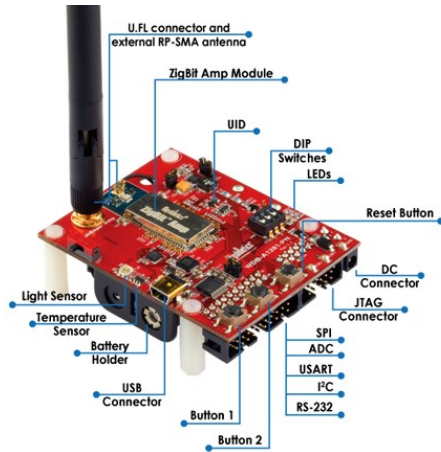


Fig. 2. Meshbeam development board

In ZigBee, there are three kinds of devices, each one having its own purpose:

1. Coordinator (C): A full function device (FFD) that it is in charge of creating the PAN (Personal Area Network) and typically is the point of the WSN to acquire all sensors information from all the other motes to be shown in a computer. The icon used to represent this device is a filled circle, Figure 1 shows one.
2. Router (R): A FFD that it is in charge of routing when the range of coverage requires this capability, so it is possible to have dynamic topologies. The icon used to represent this device is a small filled circle inside a circle, Figure 1 shows six.
3. End device (ED): A reduced function device (RFD) that is always slept (to reduce consumption) and only wakes up to do a specific task, for instance, to send sensor information to the WSN, typically directed toward C. The icon used in Figure 1 is a not filled circle.

A ZigBee WSN is composed of one C, many EDs and many Rs. Each kind of devices can receive what the others transmit if they are in the same range of coverage, because the transmission media is shared by all of them. But not all the received information is processed (the explanation of why this is that way is out of the scope of this paper).

Atmel also provides a demo, called WSN demo, to help development tasks. The demo is prepared to create a dynamic ZigBee topology, controlled by the ZigBee's Network layer. There is no positioning function in this demo so we have to modify the WSN Demo³ source code to do so.

3.1 WSN Demo Modifications

As explained in section 2, to determinate the position, we require two kinds of beacons, beacon1 and beacon2. Beacon 1 is used to inform other devices that a

³ This code was also supplied by Atmel.

mobile mote is present and beacon2 is used to inform C the RSSI value that a fixed mote receives from a mobile one for location estimation. To send both beacons in WSN the information saved in a table at the network layer called neighbor table has been used. This table registers all the FFD (motes that are C or R) that are in the range of coverage of a certain mote and for each one it registers the RSSI value of the received signal from that mote. Periodically, a FFD device sends a Zigbee Network layer message to inform others that it is in the WSN. This message is used by neighbor motes to measure the RSSI value of the received signal and to save it in their own neighbor table. So beacon 1 is automatically sent by the protocol stack. As only FFD can send this kind of message, the mobile motes have to be R, as shown in Figure 1.

The WSN demo code in motes has been changed to send periodically beacon 2 messages, for doing so, a search has to be done in the neighbor table to find out if the mobile mote is in its range of coverage, if so, the beacon 2 is sent to C with the required information as explained in section 3. As neighbor tables are only in FFDs, fixed motes have also to be R.

4 Deployment

The prototype was deployed over half floor of our Department Area, measuring roughly 225 m². The first step we did, was to test how the different kind of materials affect the RSSI value, taking into account those results, and that a mote can cover an area of 4-5 meters, we determined that with a number of 7 fixed motes is enough (Figure 3).

As shown in Figure 3, there are 11 rooms (room number G1.25 is the telecommunications room and doesn't belong to our Department), being six of them double sized than the others and counting the halls as room, numbered from 1 to 3 (left to right). We determined that with two signature points for small rooms and four signature points for big rooms were enough. Halls had more troubles because signals propagated easily so we empirically determined to have a point every 3 meters. In total, we have 46 signature points placed as shown in Figure 3, crosses represent fixed motes, being C the mote called Mote 0, and black dots represent signature points. A similar determination have been done in Living Lab in Madrid, in this case the area is about 100 m² less than our Department Area. Because there were only one room and the bathroom, we estimate that with 5 fixed motes (they are shown in their approximate placement) and 27 signature points should be enough. Figure 4 shows the Living Lab area deployment taking the AOIs (Areas of Interest) into account.

Once all the signature points are determined, next step is to create the signature point database. It is an off-line process where the same Windows application for positioning is used to take signatures information for a period of time. The mean value of each received signature is calculated and saved in the signature database. All the received signatures in each signature points are also saved in a log file to be analyzed afterward in order to change later the signature database if it is necessary.

The required time for this phase depends on the number of samples used to calculate the RSSI mean value and the numbers of signature points to be determined, obviously the larger is the area, the more points are needed.

To separate positioning software application over the different deployment places, a floor plan is included, in charge of binding coordinates and rooms identifications.

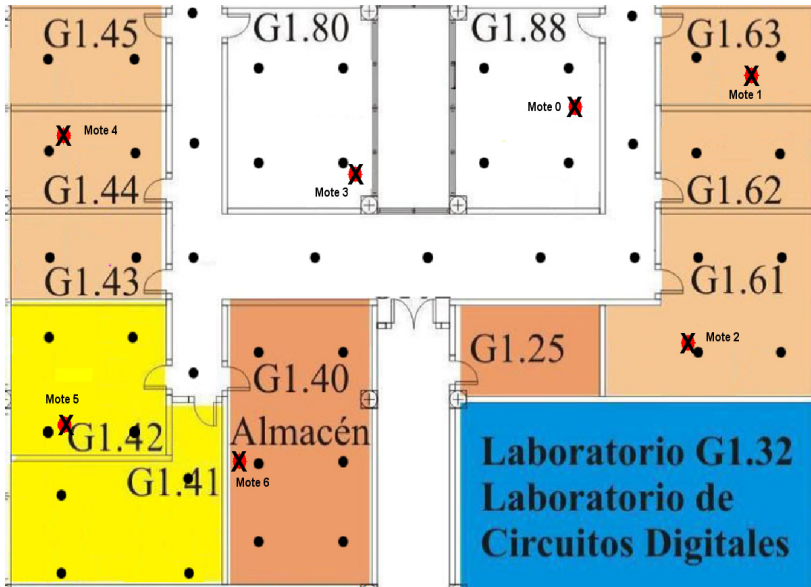


Fig. 3. Signature points. The six fixed motes, labeled as Mote 0 to 6, shown in their real placement.

Once the signature point database is created, the system is ready to determine the online position of a mobile mote. Figure 5 shows the computer software prototype used adapted to Living Lab. As it is shown, we not only determine the position of more than one mobile mote, number 1 to 4, but also the value of two of the sensors that those motes have, light in Figure 5, and temperature. In the online phase, simultaneously, all the acquired information by all the received signatures is saved in a log file in order to be able to make further analysis, and also information like time, the x , y estimated coordinates, battery charge and others extra values extracted through the integrated sensors; in this case light, temperature and battery. The adapted prototype also receives the domotic bus event that the Living Lab equipment generated but this information is not used in the positioning estimation due to time restrictions (the prototype was adapted to EvAAL in one month).

It is also possible to process the acquired information in order to make advices or send it via Internet, for instance to a hospital.

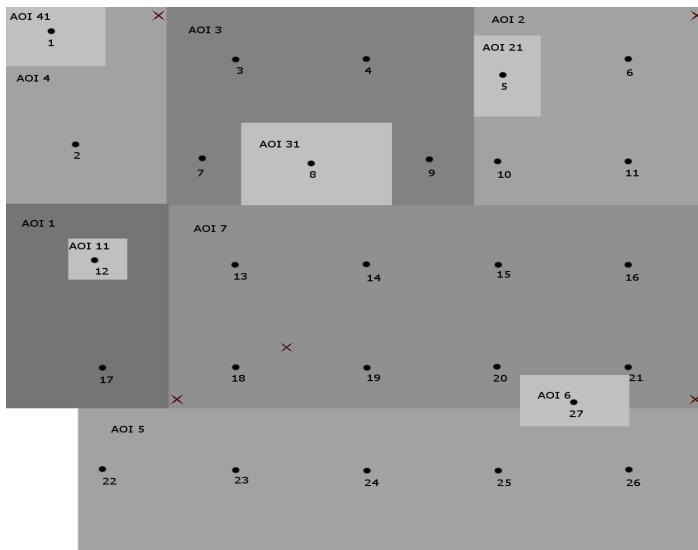


Fig. 4. Living Lab area deployment. AOIs are shown in different gray scales. Fixed motes are the cross dots.

WSN Connection

Refresh COM8 Disconnect Offline Phase
 Update DB
 Erase DB

COM8 opened at 03/07/2012 11:01:22

Log Enable

Fingerprint Sample Connection

Charge

Mobile Notes

Mote 1 Mote 2 Mote 3 Mote 4

Temp (°C) 27
 Light (Lux) 262.50
 Battery (%) 83.254

Fixed Motes

| Temp (°C) | Coord | Fix1 | Fix2 | Fix3 | Fix4 |
|-------------|-------|------|------|------|------|
| 32 | 30 | 28 | 29 | 29 | |
| Light (Lux) | 387 | 392 | 552 | 402 | 225 |
| Battery (%) | 20 | 91 | 92 | 91 | 96 |

Domatic Bus Events

03/07/2012 8:55:01 - Device: switch8 (234.0.345.00) Active = true
 03/07/2012 9:02:59 - Device: bicycle (234.0.345.00) Active = true
 03/07/2012 9:03:59 - Device: switch1 (234.0.345.00) Active = true

Fig. 5. Positioning User Interface adapted to the Living Lab plan view

5 Prototype Tests

Our prototype was tested in order to know if it is possible to determine if a mobile mote is placed in a room, i.e., it doesn't matter exactly where it is inside the room, so the precision required was not very high. This has been this way, because, as mentioned before, the kind of applications for whom our indoor position solution was tested didn't require more precision.

Based on empirical measurements, we determined that the precision of our system is about 77%, i.e., the right room was determined in that percentage being the accuracy among 0 meter to less than 1 meter from the real position. The rest were bad position determination, not the right room, but the accuracy was among 0,5 meters to less than 4 meters from the real position.

All the competitors knew the evaluation criteria before the test day in Living Lab, in our case, we didn't pay attention because we were focused on adapting our prototype to the Living Lab. We didn't even have time left to test correctly the adapted prototype prepared to the EvAAL. For a quick testing, we deployed the Living Lab in two of our Laboratories, marking on the floor the different AOIs as shown in Figure 6. The emulation was not quite good due to the fact that there were a wall between the kitchen and the living-room zone (Figure 7).

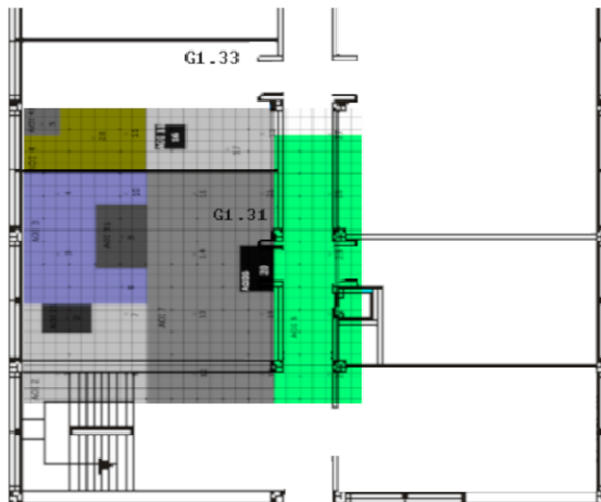


Fig. 6. Living Lab emulation in Electronic Technology Department Labs

The results weren't very good but they were worse than we expected. In our test, the point to point accuracy was 3,1 meters in percentile 75, the result in Living Lab was 4,6 meters. The AOI accuracy result in our labs was 59% being 40% the right AOI and 19% not the right AOI but a subarea in that AOI. In Madrid we got a 20% in AOI accuracy which the right AOI score was 5,5%. So the wall in the labs we tested the adapted prototype made the emulation not as real as we thought. In addition, we forgot to save the log file during the test in Madrid, so we didn't get any sample data to make further studies later on to compare with our test in Seville.

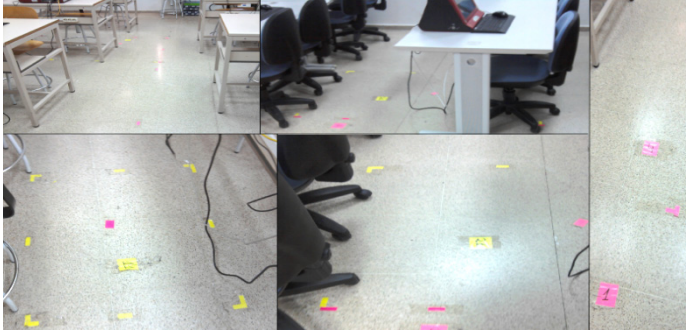


Fig. 7. Living Lab emulation photo

6 Conclusions

In this paper we have presented a prototype for an indoor position system based on BitCloud Stack over a WSN. The system requires to populate an area with a certain numbers of motes depending on the extension, to include the plant view (for instance Living lab one) in the PC application software where the position is going to be determined, and a calibration phase to fulfill the signature database. Once the system is calibrated, it is ready to be used.

As presented, the system could be easily applied to any applications of Assisted Ambient Living or Smart Building adding the specific sensors both to the mobile mote to acquire user information and the fixed mote to acquire environmental information, as all data gathered are received in a central point they should be processed for any purpose and send them via Internet if it is required. There is also a log file available both in the off-line and on-line phase for further analysis of data received.

The prototype has been adapted to the EvAAL competition environment and tested. The results weren't very good but the experience has made us change our mind. Now we know how the goodness and badness of our prototype can be tested and all the things that have to be taken into account to design a system that real users have to use.

Acknowledgments. This work has been carried out within the framework of the research program: (P08-TIC-3631) – Multimodal Wireless interface (IMI) funded by the Regional Government of Andalusia.

References

1. Hristova, A., Bernardos, A.M., Casar, J.R.: Context-aware services for ambient assisted living: A case-study. First International Symposium on Applied Sciences on Biomedical and Communication Technologies. In: ISABEL 2008 (2008)
2. Figueiredo, C.P., Gama, O.S., Pereira, C.M., Mendes, P.M., Silva, S., Domingues, L., Hoffmann, K.-P.: Autonomy Suitability of Wireless Modules for Ambient Assisted Living Applications: WiFi, ZigBee, and Proprietary Devices. In: Sensor Technologies and Applications, SENSORCOMM (2010)

3. Hong, S., De Florio, V., Ning, G., Blondia, C.: Towards. 2008. Building Virtual Community for Ambient Assisted Living. In: 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (2008)
4. Sun, H., De Florio, V., Gui, N., Blondia, C.: PRomises and Challenges of Ambient Assisted Living Systems. In: Sixth International Conference on Information Technology: New Generations, ITNG 2009, pp. 1201–1207 (2009)
5. Martin, H., Bernardos, A.M., Bergesio, L., Tarrío, P.: Analysis of key aspects to manage wireless sensor networks in ambient assisted living environments. In: 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL 2009, pp. 1–8 (2009)
6. Dietrich, D., Bruckner, D., Zucker, G., Palensky, P.: Communication and Computation in Buildings: A Short Introduction and Overview. *IEEE Transaction on Industrial Electronic* 57(11) (2010)
7. Chen, P.-W., Ou, K.-S., Chen, K.-S.: IR indoor localization and wireless transmission for motion control in smart building applications based on Wiimote technology. In: SICE Annual Conference 2010 (2010)
8. Han, C., Chou, P., Duri, S., Hui, L., Reason, J.: The Design and Implementation of a Smart Building Control, pp. 255–262 (2009)
9. Snoonian, D.: Smart buildings. *IEEE Spectrum* 40(8), 18–23 (2003), doi:10.1109/MSPEC
10. Bahl, P., Padmanabhan, V.: RADAR: an in-building RF-based user location and tracking system. In: INFOCOM, pp. 775–784 (2000)
11. Lorincz, K., Welsh, M.: MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 63–82. Springer, Heidelberg (2005)
12. Konrad, L., Matt, W.: MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. To Appear in Springer Personal and Ubiquitous Computing, Special Issue on Location and Context-Awareness (2006) ISSN: 1617-4909, (Print) 1617-4917
13. Medina, A.V., Gómez, I., Romera, M., Gómez, J.A., Dorrónzoro, E.: Indoor Position System based on BitCloud Stack for Ambient Living and Smart Buildings. In: 3rd International ICST Conference on IT Revolutions, Córdoba, Spain (2011)
14. ZigBee, 2009 RF4CE Specification. Version 1.00. ZigBee Document 094945r00ZB (March 17, 2009)
15. Medina, A.V., Gómez, J.A., Rivera, O., Dorrónzoro, E., Merino, M.: Fingerprint Indoor Position System based on OpenMAC. In: International Conference on Wireless Information Networks and Systems, Sevilla, Spain (2011)
16. Potorti, F., Corucci, A., Nepa, P., Furfari, F., Barsocchi, P., Buffi, A.: Accuracy limits of in-room localisation using RSSI. Antennas and Propagation Society International Symposium, 2009. In: APSURSI 2009, June 1-5. IEEE (2009)