IET Research Journals

Research Paper or Case Study

IET Journals
The Institution of Engineering and Technology

A lightweight localization approach for WSNs: performance analysis and validation

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Marco Santic[1], *Luigi Pomante*[1*], *Claudia Rinaldi*[1]

[1] *Center of Excellence DEWS, University of L'Aquila, L'Aquila, Italy*
* *E-mail: luigi.pomante@univaq.it*

Abstract: Wireless Sensor Networks (WSNs) have become very popular in recent years due to their wide applications spectrum. One of the main issues arising while conceiving with WSNs functionality is the opportunity of developing a proper localization algorithm that has to be tailored basing on hw/sw nodes limitations. This paper proposes a localization algorithm for WSNs based on Received Signal Strength (RSS) measurements whose results, although comparable with existing localization algorithms, proves here to be of practical implementation on devices with very low computational capabilities as well as efficient in real-time usages. Extensive simulations and indoor/outdoor experimental measurements have been performed to verify the effectiveness of the proposed algorithm, taking also into account implementation issues on commercial sensor nodes.

1 Introduction

Wireless Sensor Networks (WSNs) have known an incredible growth in the last decades due to their wide and innovative range of applications (e.g. envinromental monitoring, surveillance, localization, etc.). The individual nodes in a WSN are inherently resource constrained: they have limited processing speed, storage capacity, and communication bandwidth. This aspect has to be taken into account while designing algorithms for WSNs. This paper mainly focuses on localization algorithms that enable a variety of location-based services, such as small vehicle and people tracking in indoor/outdoor environment, patient monitoring for health care purposes, automation and control in domotic technologies. A multitude of different approaches and techniques, that are tailored or adaptable to a WSN, has been investigated in literature [1]. The indication of Received Signal Strength (RSS), within other possible techniques, is a common and coarse approach to locate objects in different environments (indoor and outdoor) [2][3][4][5], where nodes of a wireless sensor network compose a communication infrastructure. A network composed by some fixed nodes placed in certain positions, called *anchors* or *beacons* and one or more fixed/mobile nodes, called *blind*, constitute a typical scenario. The blind nodes have to gather some informations from the anchors, seen as reference nodes, to localize themself. Among all the possible information obtainable, the use of the RSS and all the techniques close to it, poses a very low demand on the complexity of the hardware, since it is a parameter that is commonly obtainable in radio tranceivers and also leads to a low complexity in the software; on the other hand it implies also errors, since the signal propagation is often unpredictable, especially in buildings and indoor scenarios, being prone to shadowing, refractions, reflections and signal scattering [6]. Comparing to the most developed approaches, this work can be related to those that use *fingerprinting* methods [7][8][9] and *lateration* methods [10][11], see in figure 1 a block scheme for a graphical comparison. Fingerprinting, also known as scene analysis [1], refers to the type of algorithms that first collect features (e.g. RSS fingerprints) of a scene and then estimate the location of a target by comparing run-time measurements with off-line collected features [7]. Such methods require thus an off-line training phase; some different works are targeted on pre-computation of fingerprints and use models that sometimes achieve high level of complexity [8]. Lateration, instead, requires two main steps: *ranging* (or *distance estimation*) and *lateration* (or *position estimation*); a common approach is often based upon the relationship between the receiver-transmitter distance and the signal strength. Works trying to overcome errors due to basic log-distance models are exposed in [11], where linear regression is investigated, while [12] and [13] use polynomial regression; these approaches are used to better fit the relation between RSS and distance. Once a set of distances from known positions (where the anchors are placed) is available for the blind node, it is possible to obtain the position of blind node, in two or three dimensions, solving a determined or overdetermined linear system; many works have investigated how to solve such an equations system, and techniques as *linear least square* (LLS), *non-linear least square* (NLS) [11] and *weighted centroid localization* (WCL)[10] are some of them. The technique investigated in this paper can be classified as a range-free method, since it overcomes the two lateration method steps. Experimental results that are taken as reference in this work come from [14], where also other free-range algorithms, such as *Ring Overlapping Circle* RSSI [15], are taken into account. The localization algorithm proposed and implemented in this paper (that is an extended version of [16][17]), if compared to other algorithms, aims to achieve a coarser precision, but it is tailored to require very few computational resources and to keep mathematical operations as simple as possible. The algorithm reckons on a grid of positions, considered candidate, and performs an evaluation of them, computing for each a quality indicator that is derived from radio signal propagation equations. Using a grid, with different grid steps is possible to work with more or less candiate positions, having a trade off between precision and execution time; moreover, relating the propagation conditions, there are not specific assumptions.

This approach shows similarities with fingerprinting techniques, since it uses a cost function on a discrete set of positions. However, it does not need any off-line phase (pre-computation or measurements)[8], but it is still able to have benefits from the advanced refinement techniques, such as LLS or WCL, that can be further applied if needed. In fact, the proposed approach can also be used as the lightweight preliminary phase for other approaches, providing a sufficient precise starting point for an iterative or multi-step localization technique. Different WSNs topologies are suitable for the proposed algorithm and it is worth noting that the localization can be done by the blind node or the anchor nodes, being the scenario simmetrical with respect to RSS, and are appliable all the definitions of positioning as in [1].

This paper is structured as follows: Section 2 introduces a novel approach to localization, starting from the considerations and the constraints related to commercial nodes for WSNs; Section 3 illustrates and discusses about the feasibility analysis and computational cost. Section 4 focuses on the description of the experimental setup,

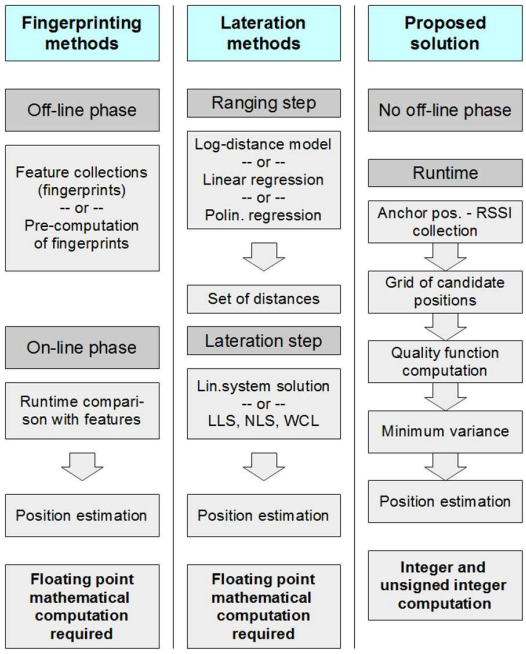

Fig. 1: Graphical comparison of algorithms

where data coming from real scenarios are used to evaluate the effectiveness of the proposed solution, first by means of an off-line execution, then with on-line processing directly on sensor nodes. Results of experiments are presented and discussed in Section 5. Finally, conclusion and future works are discussed in Section 6.

2 Proposed solution: lightweight localization algorithm

In particular circumstances, the deployment and the set-up of a localization technique can be really challenging. The most important requirements for a localization algorithm to be executed by means of a WSN are:

- easy deployment,
- low computational costs.

The satisfaction of these two properties is strictly related to the accuracy the algorithm can guarantee and a trade off is always needed. Too often techniques and approaches proposed in literature require a heavy computational effort, involving a variety of mathematical operations in floating point. Such solutions are not always feasible for the implementation on a device equipped with a minimal computational core. Moreover, it is worth noting that, often, complex mathematical approaches are belittled if there is an underlying RSSI-based ranging approach. With these considerations in mind, this section introduces the *Lightweight Localization* (LwL) algorithm, an innovative solution that explores the possibility of a grid scan, for the evaluation of the best candidate position estimation, given a quality metric derived from the theoretical signal propagation model.

The scenario assumes the presence of a variable number of *anchor nodes*, i.e., nodes whose positions are known and sharable with the other nodes of the network, and one or more *blind nodes*, whose positions are unknown and have to be determined on the basis of anchors positions and RSSI values coming from anchor nodes. While a blind node is mobile, anchor nodes are assumed to occupy only fixed positions.

RSSI (RSS Index or Indicator) based localization approaches rely on the assumption that, when a couple of transmitter-receiver is in free space, the received power is inversely proportionate to the transmitted one by an amount depending on the square of the distance. This allows distance estimation or, at least, point/region characterization. So, the LwL algorithm assumes that a blind node is able to receive beacons from anchor nodes. So it can collect information about the RSSI values (associated to the beacons) and about anchor nodes positions (i.e., data contained in the beacon itself). Such data will be used for position estimation. The main features of the algorithm are described in the remainder of this section.

Since the ideal received power P_{RX} is proportional to the transmitted power P_{TX}, the transmitter-receiver distance d, and the path-loss exponent (PLe) α, [18], it can be written:

$$P_{\mathrm{RX}} \propto P_{\mathrm{TX}} \cdot d^{-\alpha} \tag{1}$$

This equation has the same form of the freespace path loss model, two-ray model, Hata model, and the COST extension to the Hata model, [19].

Under the hypothesis that all the anchor nodes adopt the same transmission power, it is possible to assume the following:

$$P_{\mathrm{RX}} \cdot d^{\alpha} = constant \tag{2}$$

where, based on previous assumption, the relation is valid for all the anchor nodes. The *constant* in (2) summarizes other terms, such as common gains or common attenuations, that are not numerically important in their value. Just for convenience, in the following, a slightly different formula, taken from another frequently used model of the relationship between signal strength and distance [20], [21], is adopted:

$$P_{\mathrm{RX}} \propto P_{\mathrm{TX}} \cdot \frac{1}{1+d^{\alpha}} \tag{3}$$

In this way, zero-distance issues can be avoided and the expression (3) can be written as:

$$P_{\mathrm{RX}j} \cdot (1 + d_{i,j}^{\alpha}) = K_{i,j} \tag{4}$$

Then, by supposing to place a blind node in several positions i, each one with a distance $d_{i,j}$ from an anchor j, and by applying equation (4), it is possible to evaluate the products $K_{i,j}$ and then to compute a quality indicator q_i for each "supposed position":

$$q_i = \sigma_i^2 = \mathrm{Var}[K_{i,j}] \tag{5}$$

The quality indicator is based on the variance since, theoretically, if the supposed position corresponds with the correct one, all the products in equation (4) would give the same result (i.e., the variance would be zero). Thus, the best supposed position $q_{\tilde{i}}$ (i.e., more representative of the real blind node position) will be the one with the lowest variance, i.e. the one that better fits:

$$q_{\tilde{i}} = \min_i(q_i) \tag{6}$$

However, since the received powers $P_{\mathrm{RX}j}$ are represeted by $RSSI_j$ values expressed in dBm, they must be linearised. For this, an offset (e.g. the minimum $RSSI_j$ belonging to the considered set) is subtracted from them and the final result is truncated to an integer. Such a policy does not affect the validity of the results, given that adding a constant value to a random variable does not change the variance, i.e. the expectation increases by the same amount. Moreover, as a positive effect, it simplifies the algorithm implementation on resource-constrained devices (e.g., those without a floating point unit). Finally, to mitigate the effects of fast fluctuations in RSSI values, a moving average filter has been applied to RSSI values associated to each anchor. The buffer depth of such a filter can be set accordingly to several environmental parameters. Once the buffer is full, a new "localization run" can be performed at the reception of each new RSSI value, or after a given time interval, as needed by the final application.

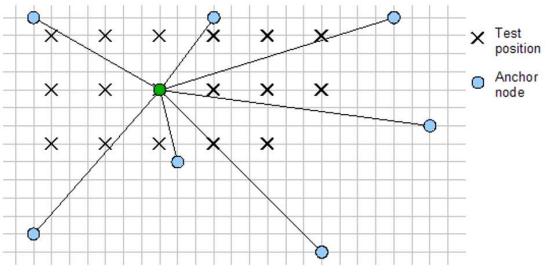

Fig. 2: Grid of the searching area

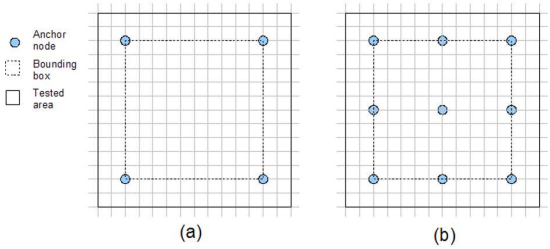

Fig. 3: Grid and anchor displacement in simulation scenarios

From the description above, it should be clear that LwL is based on a greedy brute-force approach that follows the KISS (Keep It Simple and Short) principle, [22], [23]. Equation (4) is evaluated on all (or a portion) of crossing points characterizing the grid that divides the horizontal plane of the space where the blind node can move. This space is usually bounded by anchor nodes, as shown in figure 2. Subsequently, basing on the variance, the best position is chosen as already discussed for equation (5). As the searching area becomes wider, or even three-dimensional, it is possible, under particular circumstances, to keep low the number of test points (see Section 3.2).

Since RSS-based localization algorithms rarely provide results with a precision greater than 0.5 meters, [4], [24], [25], the grid can be built by using a quite wide step (i.e., from 0.5 up to 5 meters). In this way, it is possible to reduce the number of supposed positions, thus lightening the approach as needed. In fact, it is worth noting that, once a position has been identified by means of a wide step (i.e., reduced precision), the same algorithm can be applied by considering a sub-grid built by using a smaller step, just around the previously estimated position, to improve the localization/tracking precision.

3 Feasibility analysis

3.1 Simulation of the lightweight localization algorithm

To analyse the feasibility of the proposed LwL algorithm, as a first approach, different simulation scenarios have been tested in *Matlab*.

A first couple of scenarios, consisting of a box area of 10m x 10m wide, with respectively 4 and 9 anchors (as depicted in figure 3.a and 3.b) placed on the same plane, have been simulated. The propagation formula for *Free Space* (see [26]) has been applied while recording a matrix of ideal RSSI levels for an area +/- 20% of the bounding box along x and y axes; the dimension of the matrix has been changed according to three different grid steps, chosen as illustrated in table 1.

Every point (or position) of the space matrix holds 4 or 9 RSSI values, respectively for scenario a and b. Also a matrix of distances (the term in round braces of equation (4)) was precomputed to quicken the computation. A preliminary test consisted of choosing randomly a target position from the 19881 available (for the grid step

Table 1 RSSI Matrix dimensions.

Grid step (m)	X pos. = Y pos.	# Points
0.10	141	19881
0.25	57	3249
0.50	29	841

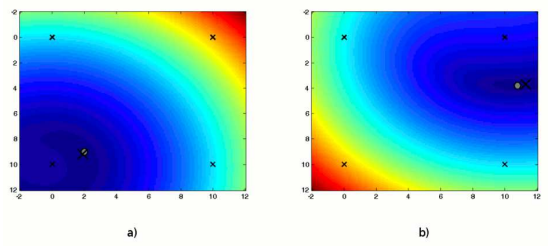

Fig. 4: Chosen/estimated position and quality metrics repr., 4 anchors

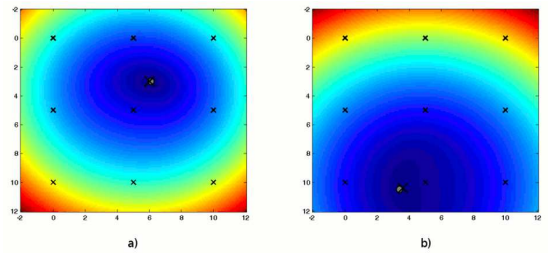

Fig. 5: Chosen/estimated position and quality metrics repr., 9 anchors

of 10cm) and, trying to apply the quality metric described before, estimate the position between all candidates or a subset of them. In figure 4 and figure 5 it is possible to see some runs for 4 and 9 anchors scenarios; the little "x" are the anchors, the grey dots are the chosen positions, the big "X" are the estimated positions.

The coloured areas are the representation of the quality metrics, i.e. the variance in expression (5): blue colour represents low values, red represents high values. The estimated position is clearly in the middle of the darker blue area. The following observations arise from the previous figures: the quality function is a convex function, but it has not a perfect radial symmetry; the estimated position is not the same of the chosen one. Actually this seems to derive from the symmetry of the anchors positions, other than the location of the blind node; the second issue derives from the choice of the quality function, as explained hereafter. Since expression (4) is a rude simplification derived from the propagation model, it is not exact and a non-zero lower bound for the error exists. To estimate such an error an exhaustive tests campaign for all the points of the different configurations has been done; with reference to figure 6, a coloured representation of the minimum error for each point is depicted.

It is clear that the central area is not prone to errors, while there are some zones close to the anchors that lead to a systematic error (in particular this is due to the term "+1", that avoids divisions by zero in equations (3) and (4)). However the 90% of the whole area has an error of less than a meter, and, if considering the only bounding box and tot +/- 20% extra area, 98% of the points have an error less than 1 meter. Some bar graphs are illustrated in figure 7.

Figure 8 illustrates the same minimum error test for the 9-anchor scenario, for the grid steps of 0.5m, 0.25m and 0.1m respectively in sub-figures a), b) and c); the error distribution is clearly the same, just the "resolution" of the map seems to be better. In figure 8.d, the scale is reduced, in order to enhance the error zones. In figure 9 the error distributions for 9-anchors scenario are shown.

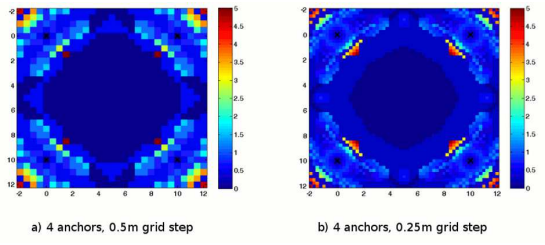

Fig. 6: 4 anchors minumum error representation [meters].

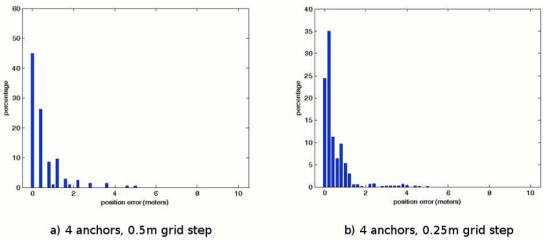

Fig. 7: 4 anchors minimum error distribution.

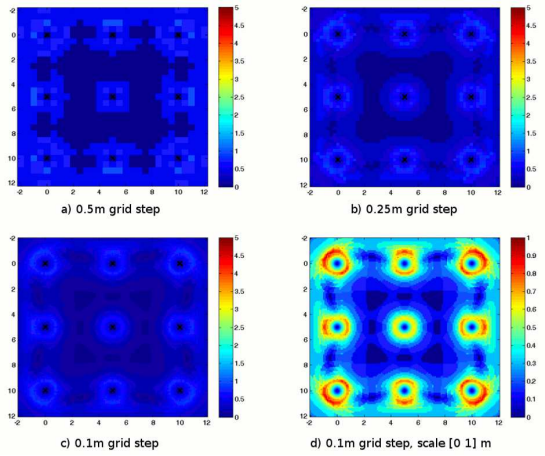

Fig. 8: 9 anchors minimum error representation.

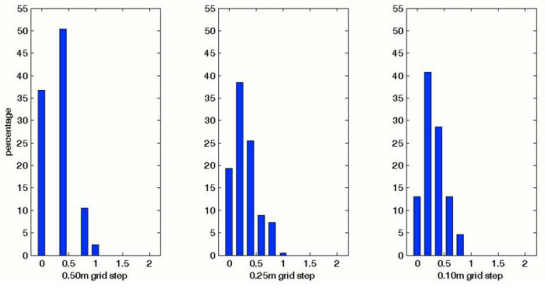

Fig. 9: 9 anchors minimum error distribution.

To simulate a noisy environment, a white gaussian noise, characterized by a mean of 0 dBm and a standard deviation of 4 dBm, has been added to the ideal values of RSSI matrices. This noise has been added to the whole matrix, then the test procedure has been repeated to evaluate the errors. Instead of choosing a position and repeating the test each time with a different realization, each point

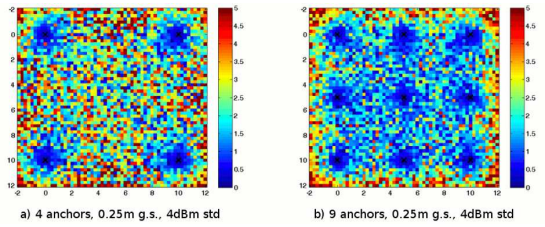

Fig. 10: AWGN to the ideal RSSI (4 dBm STD).

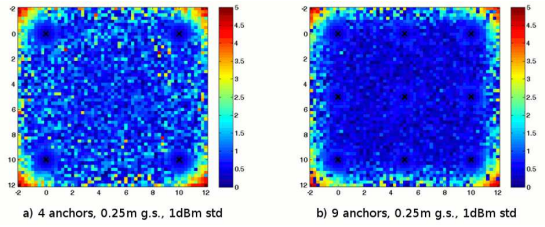

Fig. 11: AWGN to the ideal RSSI (1 dBm STD).

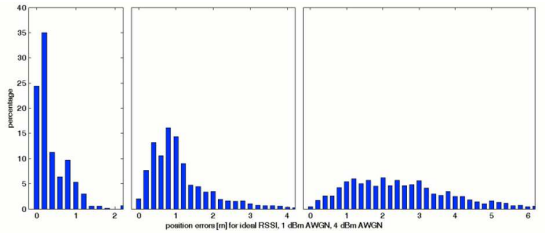

Fig. 12: 4 anchors error distribution for 0, 1, 4 dBm STD.

was estimated, making possible a representation similar to the one of minimum error (see figure 10 and 12).

The referred figures are representative if the overall result is considered, but to evaluate the effect of an averaging technique for the RSSI values, like the buffer depth parameter explained in the previous paragraphs, it is possible to reduce the standard deviation to 1 dBm, obtaining results shown in figure 11.

3.2 Computational complexity considerations

As seen before, there is a minimum error that cannot be avoided, even reducing the grid step; with the presence of noise and of other interferences of a real scenario, the results can only be worse. For these reasons, keeping the grid too dense is not a good choice, since the final result can not be improved, while the computational cost can easily increase. In two dimensions the cost is $O(a \cdot n^2)$, where a is the number of anchors considered and n is the number of points for each bounding box side; in three dimensions, the cost is $O(a \cdot n^3)$. It is worth to keep in mind that one of the main goals of the proposed solution is a reduced computational cost and this should be obtained by working with a very limited n. The quality function is a convex function (figure 13) and the search of the minimum does not necessarily imply the exhaustive evaluation of all points; it can be simplified by the search on the two axis in some different ways, as well as, of course, by involving more sophisticated techniques. In the following some very simple techniques (to obtain a close position) and their results are reported; they all have a linear cost of $O(a \cdot n)$, but it can be further improved.

Once the bounding box is identified, a first possibility is to independently search along its sides for the smaller values of the quality function, then consider the found points as the blind node possible position coordinates; this approach is referred as X0-Y0, because it

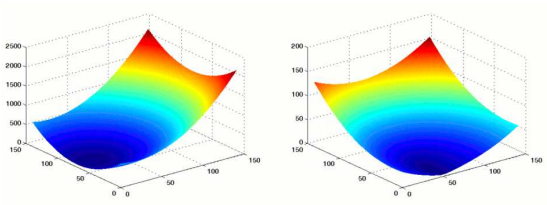

Fig. 13: 3D representation of the quality function.

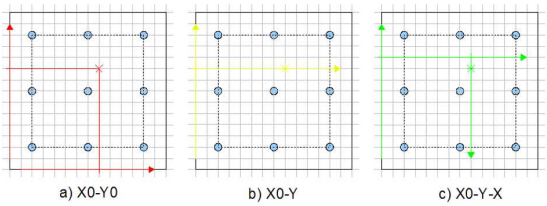

Fig. 14: Possible simple scans for quality function.

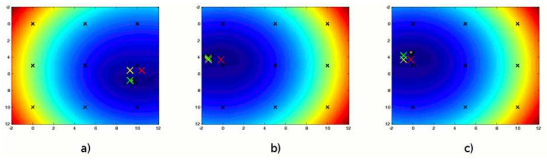

Fig. 15: Estimated positiond with basic searches.

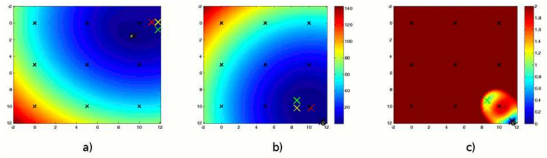

Fig. 16: Near border effect for basic searches.

analyses the two axis independently along the border of the area, as shown in figure 14.

A second possible approach is to search along an axis, let it be Y ($x = 0$), and then along the X by fixing y at its best value ($y = Y$) as illustrated in figure 14b. This is referred as X0-Y. For a "best value" closer to the real minimum, it is finally possible to repeat a search along the first axis (X again), and the approach is called X0-Y-X; figure 14c depicts such case. It is worth noting that a multitude of smarter and more efficient approaches exist to find the minimum, but these reported are the very basic, taking into account not only the computational cost, but also the data structures and the conditional statements needed to execute them. In figure 15 and figure 16 a comparison of the results obtained by applying different techniques is illustrated for different runs: as before, the coloured area represents the quality function values, the grey dot represents the chosen position, the black X represents the estimated position with a "full search" on the grid, the red, yellow and green Xs represent respectively the estimated positions by mean of X0-Y0, X0-Y and X0-Y-X approaches. In particular, by looking at figure 16, it has to be noticed that the "near-border" effects induce to a greater error, and that, outside the bounding box of the anchors, the quality function no longer has only a global minimum (figure 16c is a scale zoom-in of figure 16b).

Again, it is possible to have an a-priori idea of the precision that can be achieved by each of the proposed basic approaches. In figure 17 errors are represented on coloured maps with the scale of [0, 5] meters as for the previously illustrated precision maps. Finally, figure 18 depicts the error distributions for the basic search approaches.

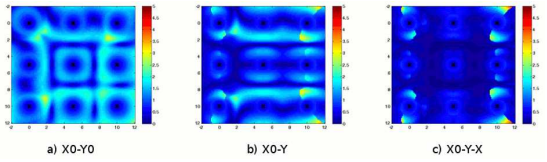

Fig. 17: Precision maps for basic searches.

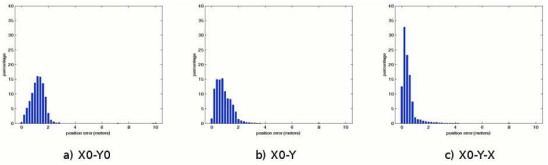

Fig. 18: Error distribution for basic searches.

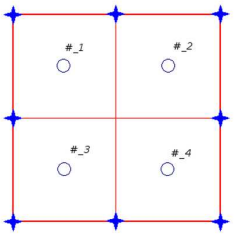

Fig. 19: Simple scenario: anchors as blue stars, tests location as circles.

4 Experiments and validation

To perform a meaningful validation, data coming from real scenarios have been exploited to evaluate the effectiveness of the proposed approach. In the first case the algorithm is applied by means of an off-line execution on a PC platform. In the second case, sensor nodes are used to perform on-line processing directly on the field. So, this section focuses on the description of the experimental setup and related implementation issues, while the analysis of the results is presented in Section 5.

4.1 Off-Line Analisys of real data

4.1.1 Simple scenario: A first set of, both indoor and outdoor, tests has been carried on by using *Memsic Iris* sensor nodes [27], [28] working at 2.4GHz. In such tests, 8 beacons-transmitter anchor nodes have been placed on the corners and along the sides of a box (see figure 19) and the blind node has been placed in the center of each quarter outlined by the anchors while collecting data.

This scenario has been deployed at different scales, with box sides of respectively 2.0m, 3.5m, 6.0m, 10.0m and 20.0m. The last 2 scenarios have been applied only outdoor. RSSI values have been collected from a total of 32 locations, each of them with 800 recorded beacons.

4.1.2 Complex scenario: Another scenario used for validation is related to a real home building, the apartment instrumented in *Casa+* project [29], where 6 anchor nodes belonging to an already deployed WSN infrastructure working at 868MHz were used to

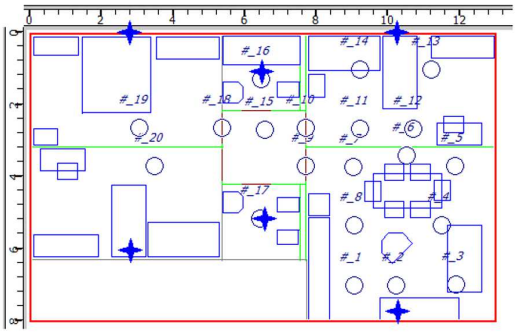

Fig. 20: Complex scenario - real building: anchors as blue stars, test locations as circles.

Table 2 Statistical information from one of the simple scenarios.

Node	count	perc.	Mean	Std
1	101	12.6%	13.39	0.51
2	100	12.5%	27.00	0.00
3	100	12.5%	16.71	0.60
4	99	12.4%	27.84	0.37
5	100	12.5%	8.52	1.40
6	100	12.5%	11.86	0.40
7	100	12.5%	9.61	0.56
8	101	12.6%	16.71	0.88

collect data related to a blind node. 20 different locations were considered as shown in figure 20. For each location, 100 RSSI values have been recorded.

Also if the number of considered positions is far from being exhaustive, this scenario is very important to evaluate the effects of a real environment on the simple hypothesis adopted in this work.

4.1.3 Algorithm implementation: To perform an off-line analysis of collected real data, the proposed localization algorithm has been implemented in C language to parse recorded RSSI data and provide results for different parameters settings.

Moreover, also statistical information about input data (i.e., total number of RSSI values per anchor, mean value and standard deviation) have been evaluated by means of the same SW application. An example of such statistical information is given in table 2.

The searching area has been defined 4 times larger than the anchors bounding box one and the algorithm has been applied by also varying some relevant parameters:

- *grid step*: this parameter determines the grid density
- *buffer depth*: the number of RSSI values for statistical information evaluation
- *PLe alpha* of equation (4)

With respect to the path loss exponent, the goal of the experiments has been to evaluate the best precision reachable by means of a fixed $alpha$, while expecting at the same time the typical curves provided in the literature, i.e., a *PLe* value mainly in the 2.0-3.5 interval.

It is worth noting that, in order to run the localization as many times as possible, the results, discussed in the next section, have been obtained by setting the averaging buffer (i.e., *buffer depth*) to 1. This will not allow to avoid errors due to fast fading and shadowing of the signal, but it concretely reduces processing times while performing the exploration on *grid step* and *PLe*.

4.2 On-line Analysis of real-data

To further validate the proposed approach, the next step was to implement the algorithm, previously simulated by means of real

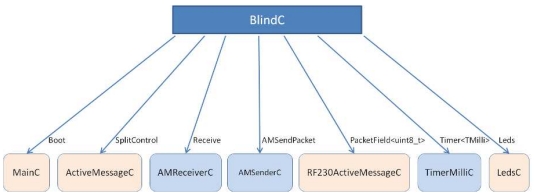

Fig. 21: Blocks scheme of the blind node application.

Table 3 Components interfaces.

Component	Interface
MainC	Boot
ActiveMessageC	SplitControl
AMReceiveC	Receive
AMSenderC	AMSendPacket
RF230ActiveMessageC	PacketField<uint8_t>
TimerMilliC	Timer<TMilli>
LedsC	Leds

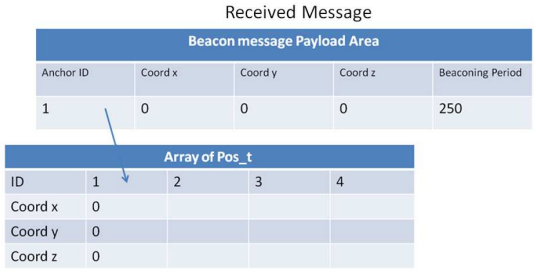

Fig. 22: Coordinates storage.

scenario measurements, onto a real platform, i.e. *Memsic MicaZ* sensor nodes [27], [28], in order to evaluate its performances on the field. For this, the C implementation of the localization algorithm, has been re-designed and coded by using the *nesC* programming language, the one used for the adopted sensor nodes.

4.2.1 Implementation for physical target devices: The *Operating System* adopted for the considered nodes is *TinyOS_2.x*, which is a BSD-licensed embedded operating system, designed for low-power resource-constrained wireless devices and, thus, one of the most used in the wireless sensor network domain [30]. TinyOS is made of *components*. Each component is independent and can be connected to other ones by means of *interfaces*. Components can be of two types, *Modules* and *Configurations*, and a TinyOS application is given by a set of modules and configurations, called *Components Graph*. The component graph of the application executed by the blind node is shown in figure 21, while table 3 lists the involved components and the related interfaces.

The main data structures used in the application are:

1. *Beacon_msg*, sent by the anchor nodes and contains *anchors ID*, *position* and *beacon transmitting period*;
2. *Pos_t*, that contains the *position estimated by the blind node* and the *anchors position*.

The implemented algorithm is composed of seven steps.

1. *Variables initialization*: when the blind node is in *Powered On* the *Boot.booted()* event is signalled. Its management procedure initializes the global variables and call the *AMControl.start()* command, that is in charge of powering on the radio and verifying if it is ready to work. If all is ok, the command returns a value corresponding to *SUCCESS* and the event *AMControl.startDone()* is signalled.

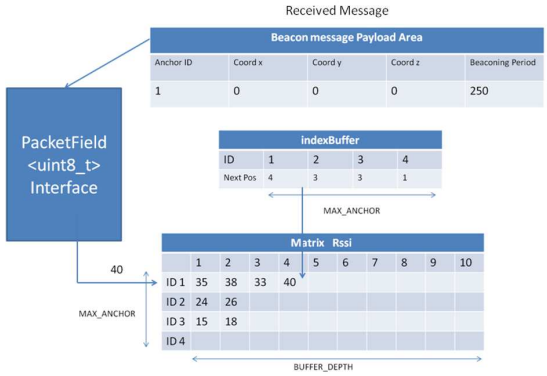

Fig. 23: RSSI value storage.

2. *Beacon_msg reception*: when a beacon is correctly received, the event *Receive.receive(...)* is signalled and properly managed.
3. *Research of min and max coordinates*: for each received beacon, the received anchors coordinates are compared with the stored ones in order to possibly update max and min values on the X and Y axes. In this way, a quadrilateral is defined, representing the bounding box of the anchors used to build a grid with the selected step. Anchor coordinates are stored, depending on the associated ID, in an array of struct of type *Pos_t*, see figure 22.
4. *RSSI buffer filling*: for each received message, the blind node collects the received power strength by means of the *getRssi()* command, and associates such a value to the anchor ID. The RSSI value is then stored in a bidimensional array where each row is managed as a circular buffer of *BUFFER_DEPTH* elements, see figure 23.
5. *Anchors positions and buffers copy*: a boolean array with dimension *MAX_ANCHOR* is used to check if all rows of the buffers array are full (it is a one-time condition needed to start the localization task, i.e., once verified for the first time, the buffers will be updated but always kept full). When the *Next_Pos* field in the *indexBuffer* reaches the *BUFFER_DEPTH*, the realted row of the boolean vector, *bufferFull*, is set to TRUE to state that the buffer is full. *Next_Pos* is then reset to 1, in order to store new incoming RSSI values in a circular way. When all the buffers are full, the function *Timer-Calc.startPeriodic(LOC_PERIOD)* is called, which activates a periodic signalling of the *fired()* event associated to the *TimerCalc* interface. In such an event handling, a copy of buffers matrix and anchors position is created. This is done because of the dynamic nature of such data structures, that are continuously updated, and a blind node position estimate evaluation on a non-static copy of the matrix can lead to unstable results. The same motivation holds for anchors position copy. Thanks to such an approach, original data structures are kept updated also during the localization task computation. This is possible thanks to the preemptive nature of nesC events management with respect to nesC tasks execution.
6. *Data processing*: processing of data contained in the data structures copies (in event *TimerCalc.fired()*) starts with the post of the *CalcPos()* task (code snippet in Appendices). It is effectively the main element of the localization algorithm. *CalcPos()* task is summarized by the following steps.
(a) For each anchor: computation of mean RSSI, in order to reduce fluctuations.
(b) Definition of the supposed positions grid.
(c) For each position of the grid: computation of the distance from each anchor.
(d) Linearization of RSSI performed by means of a look-up table.
(e) For each supposed position: evaluation of quality function Q as defined in equations (4) and (5). Such positions are then stored in a proper array.
(f) Identification of the *Best_Position* for the blind node, among the supposed ones, as the one with the lowest variance.
7. *Best_Position transmission*: the *Best_Position* is sent to the sink node (and then to the PC) by means of the *AMSend.send()* command.

Table 4 Scenario dimensions.

Scenario #	Dimensions	Type
1	3 x 3 m	Indoor
2	5 x 5 m	Indoor
3	5 x 5 m	Outdoor
4	20 x 20 m	Outdoor

It is worth noting that, in this case, the knowledge of the position on the PC-side is used for validation purposes. In fact, depending on the final application, a node could also use the position information for its own purposes without the need of sending it.

After these seven steps, the radio packet containing the localization message, sent by the blind node, is received by the sink node, that is connected to the PC and that has a running application called *Basestation*.

5 Discussion

Following the model and the assumptions discussed in Section 3, experimental setups in real scenarios have been done as described in Section 4. For the off-line analysis (see Section 4.1), a sensor node has been used to collect RSSI data. These data have been processed off-line by a PC running firstly a C-language based implementation of the localization algorithm.

In the tests related to both simple and complex scenarios, every output record keeps evidence of real positions, estimated positions and the number of occurrences in that position. So, it has been possible to evaluate the final precision and accuracy of results, by averaging mean errors for all scenarios for each pair of grid steps and PLe.

With 0.25 and 0.5 m grid steps, the best result is a mean error of 0.81 m by using a PLe of, respectively, 3.0 and 2.8. With a 1.0 m step, the best results are obtained with a PLe value close to 3.6, with a mean error of 1.00 m. Standard deviations related to these errors (i.e., precisions) are between 0.29 m and 0.36 m. A good range of values for PLe, taking into account that the results refer to both indoor and outdoor scenarios of different extension, is the one between 2.4 and 3.6. It is interesting to note, as represented in figure 24a, that the curves representing errors and standard deviation follow the expected trend (but with different minima), decreasing in the above mentioned interval and then rising for PLe values too high or too low.

The complex scenario gives results with an average error of 2.18 m for a grid step of 0.25 m and 0.5 m, and an average error of 2.22 m on a grid of 1.0 m. Such values have been obtained with a PLe in the range of 2.4 to 2.8. It is worth noting that the precision is not better on a more dense grid, while computational effort with a 1.0 m resolution is 16 times lighter than the 0.25 m one. Graphs of the results are shown in figure 24b. They substantially confirm the findings about the trend of the performance w.r.t. the PLe and grid step.

For the on-line analysis based on a real implementation (see Section 4.2), localization data have been sent to a sink node and consequently to a PC for validation purposes. The application running on the anchor nodes is the same as the one used previously for off-line analysis.

For preliminary validation purposes, four different static scenarios (table 4), each one with 4 (*MAX_ANCHOR*) anchor nodes at different positions, have been considered; for each scenario, 3 different blind node positions have been taken into account. Finally, for each position, a set of about 60 localization messages has been collected. Table 5 summarizes the overall results, providing the mean error for each scenario and, also, quantifying the computational time for the run of the localization algorithm. Such results can be compared to those coming from the work in [5], where similar scenarios have been used to gather, with a more complex computation, a position every 2 seconds with a mean error below 2 meters.

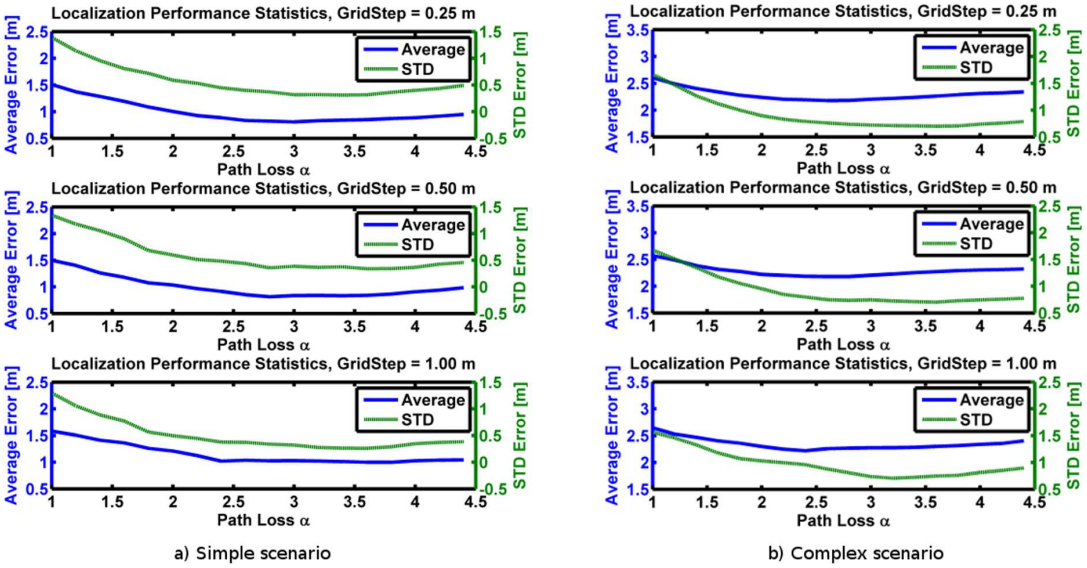

Fig. 24: Simple scenario: average errors and STD.

Table 5 Mean error and computation time per scenario.

Scenario #	Mean Error [m]	Computational time [ms]
1	0.93	84.7
2	1.48	225.5
3	1.32	225.5
4	3.87	3161.5

6 Conclusions and future works

This work has led to design, implementation and validation of a lightweight localization algorithm for WSNs. Known approaches for localization services have been analysed and a proper selection of them has been then integrated and adapted to the requirements of the considered domain. The proposed approach has been firstly analysed by means of simulations, in order to quantify the reachable accuracy, then it has been preliminary validated by means of the execution with real data and by implementing it on real devices.

It is worth noting that this is the first attempt in such a direction and further improvements are still needed. In particular, an "independent axes approach" should be considered (see Section 3.2) to further reduce computation time and also a dynamic anchor management to avoid as much as possible working on too large grids (in fact, as shown in table 5, in such a situation both the mean error and the computation time tend to be too large). By combining both the approaches, meaningful improvements are expected. A further aspect to be investigated is the integration of the localization service in a mobile agent Middleware, such as Agilla2 [31], that would facilitate the deployment in case of heterogeneous hardware. Finally, there is the clear need for more validation activities, especially considering mobile blind nodes in a changing anchors environment. However, the first validation results are encouraging and justify further efforts in such a direction.

7 Acknowledgments

This work has been partially supported by the Artemis ECSEL-JU 2015 SAFECOP project (ID 692529) and FP7-TRANSPORT CASPER project (ID 218564).

8 References

[1] Hui Liu; Darabi, H.; Banerjee, P.; Jing Liu; , "Survey of Wireless Indoor Positioning Techniques and Systems," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.37, no.6, pp.1067-1080, Nov. 2007

[2] H.P. Mistry and N.H. Mistry, "RSSI Based Localization Scheme in Wireless Sensor Networks: A Survey", 2015 Fifth International Conference on Advanced Computing & Communication Technologies, Haryana, 2015, pp. 647-652.

[3] S. Tennina, R. Gomes, M. Alves, V. Ciriello, and G. Carrozza, "The dark side of demmon: what is behind the scene in engineering large-scale wireless sensor networks". In: Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems. ACM, 2011. p. 41-50.

[4] S. Tennina, M. D. Renzo, F. Graziosi, and F. Santucci. "Esd: a novel optimisation algorithm for positioning estimation of wsns in gps-denied environments - from simulation to experimentation". Int. J. Sen. Netw., 6(3/4):131-156, Nov. 2009.

[5] S. Tennina, F. Tarquini, R. Alesii, F. Graziosi, F. Santucci and M. Di Renzo. "Automatic Personal Identification System for Security in Critical Services: Two Case Studies based on a Wireless Biometric Badge". InTech, Recent Application in Biometrics, 2011, ISBN 978-953-307-488-7.

[6] M. Kochlan and J. Micek, "Indoor propagation of 2.4GHz radio signal propagation models and experimental results," The 10th International Conference on Digital Technologies 2014, Zilina, 2014, pp. 125-129.

[7] Alhmiedat, Tareq & Samara, Ghassan & O. Abu Salem, Amer, "An Indoor Fingerprinting Localization Approach for ZigBee Wireless Sensor Networks". Eur. J. Sci. Res. 105. 2013

[8] Bosisio, A.V.; , "Performances of an RSSI-based positioning and tracking algorithm," Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on , vol., no., pp.1-8, 21-23 Sept. 2011

[9] X. Wang, J. Qiu, S. Ye and G. Dai, "An advanced fingerprint-based indoor localization scheme for WSNs," 2014 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, 2014, pp. 2164-2169.

[10] Fink, A.; Beikirch, H.; , "Analysis of RSS-based location estimation techniques in fading environments," Indoor Positioning and Indoor Navigation (IPIN), 2011 Int. Conf. on , vol., no., pp.1-6, 21-23 Sept. 2011

[11] Jie Yang; Yingying Chen; , "Indoor Localization Using Improved RSSBased Lateration Methods," Global Telecommunications Conference, GLOBECOM 2009. IEEE , vol., no., pp.1-6, Nov. 30 2009-Dec. 4 2009

[12] Wang, Y., et al. "An indoors wireless positioning system based on wireless local area network infrastructure." 6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services. No. 54. 2003

[13] Varshavsky, Alex, et al. "Calibree: Calibration-free localization using relative distance estimations." Pervasive Computing (2008): 146-161

[14] X. Luo et al., "Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites", Adv. Eng. Informat. (2010)

[15] Liu, Chong, et al. "Range-free sensor localisation with ring overlapping based on comparison of received signal strength indicator." International Journal of Sensor Networks 2.5 (2007): 399-413

[16] L. Pormante, C. Rinaldi, M. Santic and S. Tennina, "Performance analysis of a lightweight RSSI-based localization algorithm for Wireless Sensor Networks," International Symposium on Signals, Circuits and Systems ISSCS2013, Iasi, 2013, pp. 1-4.

[17] A. Falcone, L. Pomante, C. Rinaldi and M. Santic, "Performance analysis of a lightweight localization algorithm for WSNs in a real scenario," 2015 International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, 2015, pp. 1-4.

[18] John G. Proakis, Northeastern University Dimitris K Manolakis, Massachusetts Institute of Technology, Lincoln Laboratory "Digital Signal Processing", Edition 4th, 2006, Published by Pearson Higher Ed USA.

[19] Andrea Goldsmidth, "Wireless Communications", Stanford University, 2005.

[20] Simon, G.; Volgyesi, P.; Maroti, M.; Ledeczi, A.; , "Simulation-based optimization of communication protocols for large-scale wireless sensor networks" Aerospace Conference, 2003. Proceedings. 2003 IEEE, vol.3, no., pp 3 1339- 3 1346, March 8-15, 2003.

[21] Hai Van Luu and Xueyan Tang, "An efficient algorithm for scheduling sensor data collection through multi-path routing structures". J. Netw. Comput. Appl. 38 (February 2014), 150-162.

[22] Richard Korf, "The Complexity of Brute Force Search". Technical Report, Department of Computer Science, Columbia University. 1984

[23] G. Bendall and F. Margot, "Greedy Type Resistance of Combinatorial Problems", Discrete Optimization 3 (2006), 288 - 298.

[24] Boukerche, A.; Oliveira, H.A.B.; Nakamura, E.F.; Loureiro, A.A.F.; , "Localization systems for wireless sensor networks," Wireless Communications, IEEE , vol.14, no.6, pp.6-12, December 2007.

[25] Hao Guo; Kay-Soon Low; Hong-Anh Nguyen; , "Optimizing the Localization of a Wireless Sensor Network in Real Time Based on a Low-Cost Microcontroller," Industrial Electronics, IEEE Transactions on , vol.58, no.3, pp.741-749, March 2011.

[26] H. T. Friis, "A Note on a Simple Transmission Formula," in Proceedings of the IRE, vol. 34, no. 5, pp. 254-256, May 1946.

[27] MTS420/400 environmental sensor board. Datasheet on MEMSIC company webpage.

[28] Iris Wireless Measurement System. Datasheet on MEMSIC company webpage.

[29] Alesii R., Graziosi F., Marchesani S., Rinaldi C., Santic, M. and Tarquini F. (2013, July). "Short range wireless solutions enabling ambient assisted living to support people affected by the Down syndrome". In EUROCON, 2013 IEEE (pp. 340-346). IEEE.

[30] TinyOS official website: http://www.tinyos.net/.

[31] L. Corradetti, D. Gregori, S. Marchesani, L. Pomante, M. Santic and W. Tiberti, "A renovated mobile agents middleware for WSN porting of Agilla to the TinyOS 2.x platform," 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, 2016, pp. 1-5.

9 Appendices

9.1 Abbreviations

The following abbreviations are used in this manuscript:

WSN	Wireless Sensor Network
RSS	Received Signal Strength
LLS	Linear least square
NLS	Non-linear least square
WCL	Weighted Centroid Localization
LwL	Lightweight Localization
PLe	Path Loss exponent

9.2 TinyOs code

In the following a code snippet for localization task is shown.

```
//-----------------------------------------
//        LOCALIZATION TASK
//-----------------------------------------

task void CalcPos(){
  uint8_t i, j;
  uint32_t average=0, sum=0;
  uint16_t xsp, ysp, devstd=0, min_ds=65535;

  /* Task is executing... */
  //call Leds.led0On();

  /* RSSI Buffer averages */
  for(i=0;i<MAX_ANCHOR;i++){
    average=0;
    for(j=0;j<BUFFER_DEPTH;j++) average+=buffer[i][j];
    /* NEXT TIME: consider >> x with BUFFER_DEPTH = 2^x */
    average/=BUFFER_DEPTH;
    avg_RSSI[i]=average;
  }

  /* For each supposed position... */
  for(xsp=min_x; xsp<=max_x; xsp=xsp+step)
    for(ysp=min_y; ysp<=max_y; ysp=ysp+step){
      /* Pseudo distance (1 + d^2) */
      for(j=0;j<MAX_ANCHOR;j++){
        dist[j]= 1 +
                 (xsp-apCalc[j].x)*(xsp-apCalc[j].x) +
                 (ysp-apCalc[j].y)*(ysp-apCalc[j].y);
      }

      /* Quality function*/
      for(j=0;j<MAX_ANCHOR;j++)
        qual[j] = LUT[avg_RSSI[j]] * dist[j];

      average=0;
      for(j=0;j<MAX_ANCHOR;j++) /* NEXT TIME: merge two "for" */
        average+=qual[j];
      average/=MAX_ANCHOR;

      sum=0;
      for(j=0;j<MAX_ANCHOR;j++) sum+=(qual[j]-average)*(qual[j]-average);
      devstd=sqrt(sum/MAX_ANCHOR); /* NEXT TIME: avoid sqrt */

      /* Check for min devstd */
      if(devstd<min_ds){
        min_ds=devstd;
        best_pos.x=xsp;
        best_pos.y=ysp;
      }
    }
...
}
```