

Wireless Sensor Network Localization using Hexagonal Intersection

Eva M. García, Aurelio Bermúdez, Rafael Casado, and Francisco J. Quiles

Instituto de Investigación en Informática de Albacete (I³A)
Universidad de Castilla-La Mancha
02071 Albacete, Spain
{evamaria.garcia, aurelio.bermudez, rafael.casado, francisco.quiles}@uclm.es

Abstract. Device localization or positioning is a key issue in wireless sensor networks applications. One solution widely used to estimate the position of a node in the network consists in using the intersection of coverage areas. For the sake of simplicity, these areas have traditionally been modeled by rectangles, assuming some extra inaccuracy. In this paper we propose a localization algorithm based on hexagonal intersection. Results show that the only substitution of the geometric shape provides better results in the localization of the devices composing the network.

Keywords: Wireless sensor networks, localization.

1 Introduction

A wireless sensor network (WSN) consists of a set of electronic devices able to collect physical information about the environment where they have been disseminated, to process that information, and to transmit it through a wireless medium. Each device (popularly called *mote*) incorporates a microcontroller (integrating a processor and a memory), a radio transmitter and receiver, a battery, and a sensor card, which gives name to the set, and allows to measure a great variety of physical magnitudes (pressure, temperature, humidity, etc.).

Application scope of WSNs is greater every day. These networks are used in fields such as precision agriculture, environmental monitoring and forecasting, military applications, and natural disasters. Practically, all these applications require that each sensor knows their accurate location in the space. In many cases, it is enough to have an approximation.

Usually, it is difficult, maybe impossible, to program this position at each device in advance. In outdoor environments, it is possible to use a satellite positioning system, like GPS (*Global Positioning System*) [5]. However, nowadays it is an unfeasible option to equip every node in network with a GPS receiver, essentially due to size, energy consumption, and price reasons. Hence, other techniques have been gained popularity. These algorithms assume that a small number of devices, so called *anchors* or *beacons*, already know their location. These nodes propagate this

information to the rest of the network. Then, the other nodes can infer their position through a distributed algorithm.

Increasing the mathematical complexity of these algorithms carries advantages and disadvantages. On the one hand, a parallel increment in the accuracy of the estimate is produced. On the other hand, the battery consumption is increased during the process, because of the executed mathematical operations and the amount of performed communications.

A subset of the existing localization algorithms can be classified according to the kind of geometric figure used to represent the possible localization area. Thus, algorithms based on circular intersection are located on one end, providing great accuracy, but requiring the use of trigonometric functions and complex data structures. On the other end, we have algorithms based on rectangular intersection, with very low computational cost and more imprecision in the estimate.

In this work we propose one solution establishing a tradeoff between both points: hexagonal intersection. The use of hexagons provides more accuracy on devices localization. Moreover, extra complexity does not represent an important obstacle, particularly if we consider new generation motes, with higher performance with regard to computation capacity and power consumption.

The rest of this paper is organized as follows: Section 2 presents taxonomy of main kinds of localization algorithms, and focuses on solutions using rectangular intersection. Section 3 describes basics of hexagonal intersection. A comparative performance analysis between both algorithms is performed in Section 4. Finally, in Section 5 some final conclusions and possible future lines of work are offered.

2 Wireless Sensor Network Localization

There is an enormous variety of positioning algorithms in the literature, depending on the exact method to estimate distances between nodes. *Time difference of arrival* (TDOA) and *received signal strength* (RSS) are the most popular methods.

Systems like Cricket [8], Calamari [15], and AHLoS [12] use the technique TDOA. They send out two signals (usually ultra sonic sound and radio frequency signals) propagating with different speeds, and measure the difference in time of arrival. If both signal propagation speeds are known, a distance can be derived from the delta of arrival. The raw difference measurements tend to yield average estimation errors of about 74% [15]. Yet, quite good accuracies can be achieved by post-processing the measured data with techniques like noise canceling, digital filtering, peak detection and calibration. While some authors report average range estimation error of 10% [15], others claim an error of about 1% at a maximum range of 9 meters [9].

While these systems yield low estimation errors they have two limitations that limit their applicability in real world deployments. The first one is their reduced coverage: They are typically able to cover 3-15 meters [11], which is only a fraction of the communication range of radio frequency transmitters. The second and much more important limitation is that they require a separate sender/receiver pair, which implies negative effects on size, cost and energy consumption.

Algorithms based on RSS use radio signal attenuation properties to model the distance between nodes as a function of the *received signal strength indicator* (RSSI). Systems that rely on the RSSI as input parameter such as [10], [2], [1] tend to be quite accurate for short ranges if extensive post-processing is employed, but are imprecise beyond a few meters [6]. At short ranges, distance estimations exhibiting error of about 10% at the maximum range of about 20 meters [16] are feasible. The uncertainty of the radio propagation imposes problems like multipath propagation, fading and shadowing effects as well as obstacles in the line-of-sight. These effects complicate the development of a consistent model. As a result systems relying exclusively on RSSI values remain inaccurate distance estimators.

Regardless of the method applied to estimate distances, the possible position of a node is represented by a circle around the beacon. Then, the intersection point of three circles, by means of a technique called *trilateration*, gives the node location (Fig. 1).

Another localization algorithms use directional antennas and triangulation methods [7]. In [13], a combination of RSS and Bayesian inference is applied.

Apart from the above techniques, there exists a collection of localization mechanisms, called *range-free*, in which it is not necessary to estimate distances to neighbor nodes. A simple option lies in that a node receiving the position of a neighbor, estimates its own one inside the coverage area of the neighbor.

Additionally, the intersection of different coverage areas contributes to reduce the potential area where the device may be located.

Obviously, a realistic approach consists in representing the coverage area by using a circle. However, the drawback of this option is that the result of intersecting two circles is not a regular geometric shape, increasing the complexity of the data structure used to represent it. One example is shown in Fig. 2.

An alternative is the *Bounding-Box* algorithm [14], in which that circular area is represented by means of a square, getting a lighter, but more inaccurate algorithm. Next, we describe the characteristics of rectangular intersection in more detail.

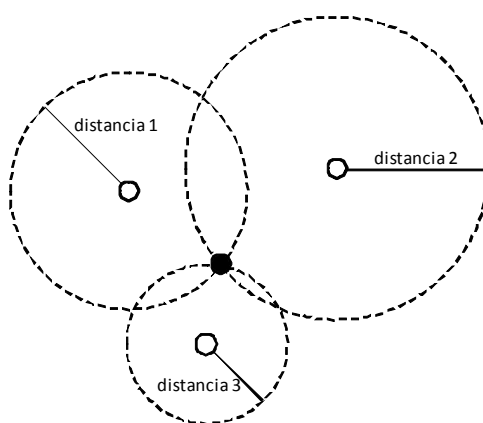


Fig. 1. Trilateration. Distances to three beacons (white nodes) allow a sensor (black node) to determine its location.

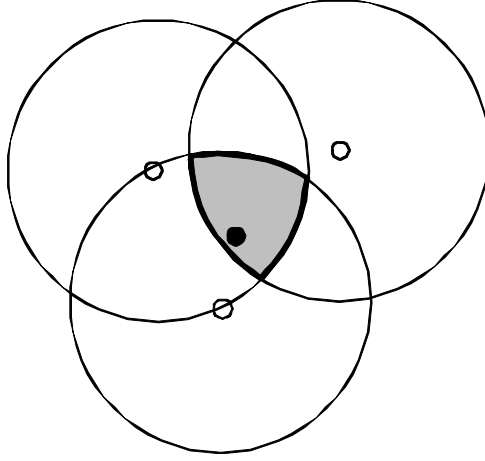


Fig. 2. Circular Intersection. Intersection of coverage areas of the three beacons allows a sensor to fence in the zone in where it is.

2.1 Rectangular Intersection

This technique is based on the idea of connectivity. If two nodes can communicate with each other, it is assumed that one is within a square centered at the other node and with a side equal to twice the radio range. The main advantage of this algorithm is that intersecting squares is a mathematically simpler operation than intersecting circles. The reason is that when intersecting squares, the result is a rectangle (Fig. 3); however when intersecting circles, the intersection is much harder to describe mathematically. Each rectangle can be represented by its upper left and lower right corners. If all neighboring nodes have their centers at coordinates (x_i, y_i) , then the corners of the intersection rectangle can be defined by

$$\begin{aligned} [x_{UL}, y_{UL}] &= \left[\max_{i \in 1..N} \{x_i\} - R, \max_{i \in 1..N} \{y_i\} - R \right] \\ [x_{LR}, y_{LR}] &= \left[\min_{i \in 1..N} \{x_i\} + R, \min_{i \in 1..N} \{y_i\} + R \right] \end{aligned}$$

The second advantage of this algorithm is that it is executed in a distributed fashion. Each node acquires positions of all its neighbors. Then, the squares centered at these neighbors are intersected yielding a final rectangle. The final position estimate is the center of the intersection rectangle.

The disadvantages of this method are the dependence on the convexity of communication regions as well as the need for high connectivity to obtain a refined and accurate estimate.

In the algorithm *N-Hop Multilateration* a phase of refinement is performed [12]. In this way, nodes without an initial estimate, due to the lack of connectivity with one beacon at least, take as reference points the estimated positions of unknown nodes in the previous phase (rectangular intersection). As a result, these nodes without connectivity estimate their position by intersecting the squares of these new reference

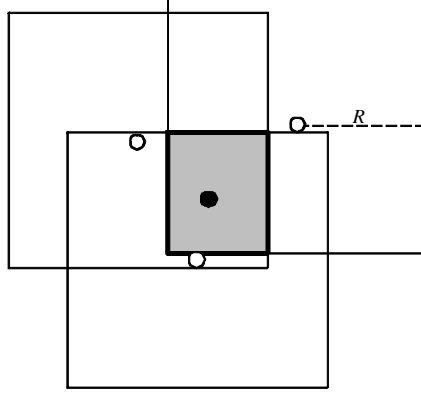


Fig. 3. Rectangular Intersection. Beacons know their exact position. The central node estimates its position from intersection of squares representing beacons range.

points, and send out new information to neighboring nodes. This process is executed in an iterative fashion until a priori established tolerance value is reached.

One further step is taken in [4]. Nodes having a location estimate do not send a pair of coordinates (x,y) , but the upper left and lower right corners of the rectangle Q delimiting its location estimate. When a node receives information from its neighboring nodes, it updates its location using the next formula:

$$Q \rightarrow Q \bigcap_{k \in K} [x_{\min}^k - S, x_{\max}^k + S] \times [y_{\min}^k - S, y_{\max}^k + S]$$

where K is the set of nodes that sense that one performing the location update, S is the sensing range (R was the communication range), and $x_{\min}^k, x_{\max}^k, y_{\min}^k$ and y_{\max}^k give the coordinates of the bounding box for the position of the k^{th} node.

3 Hexagonal Intersection

As we have seen, localization techniques based on rectangular intersection present the advantage of requiring minimum computational capacity at the nodes of the network. However, it is obvious that the error introduced in estimates is greater than using circular range areas. Given that the areas of the circle and the square are

$$A_{circle} = \pi r^2$$

$$A_{square} = (2r)^2$$

then the initial assumption of approximating a circular area by means of the square containing it, involves increasing the working area in a 27.3%.

An intermediate approach with regard to the obtained error and the complexity of the computations may consist in representing coverage areas by using hexagons. In particular, we start from the regular hexagon centered at the circular range and whose apothem is equal to the radio range (Fig. 4).

Bearing in mind that the areas of the circle and the hexagon fit the next formulas:

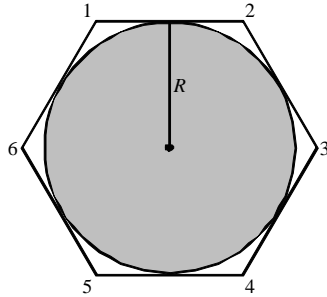


Fig. 4. Range approximation by hexagon.

$$A_{circle} = \pi r^2$$

$$A_{hexagon} = \frac{6r^2}{\sqrt{3}}$$

then, the use of this geometric shape implies an increase of a 10.2% in the localization area.

Although the result of intersecting rectangular areas in an iterative fashion is always a new rectangle, intersecting hexagonal areas will result in irregular polygons with 6, 5, 4, or even 3 sides (Fig. 5). However, these polygons have the property of that each side has a slope of 0° , 60° or 120° . Although they have not necessary six sides, from now on we call these polygons *pseudo-hexagons*.

If a rectangle is determined by the coordinates of two opposite vertices, a pseudo-hexagon can be defined by means of three vertices belonging to different sides. If we number vertices in clockwise, starting with the upper left vertex, we have considered vertices 1, 3, and 5 (Fig. 4).

Conceptually, intersection of rectangles and intersection of pseudo-hexagons are very similar. Intersection of two rectangles A and B consists in comparing each side

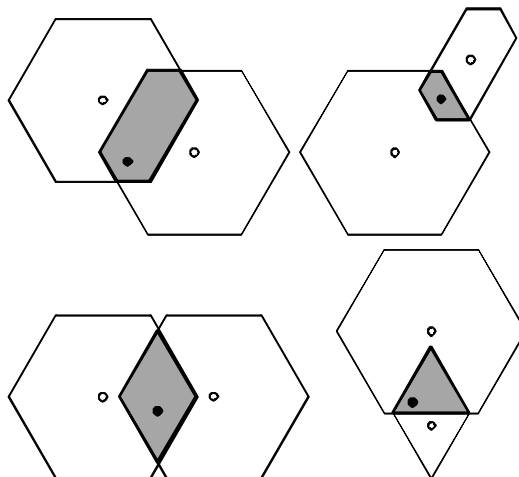


Fig. 5. Hexagonal Intersection.

of A with the corresponding side of B, and choosing those being closer to the center. After that, the computation of the four cut points of the lines defining the selected sides is performed.

The intersection of pseudo-hexagons is carried out in the same way; however, in this case, it is necessary to compare six pairs of sides (instead of four) and to obtain nine cut points.

The main difference between rectangular and hexagonal intersection is not the geometric shape itself, but the fact that the sides are or not parallel to the coordinates axes, because this implies that the computation of cut points is implicit in the case of rectangles.

In an analogous way to rectangular intersection, in order to obtain the intersection of two pseudo-hexagons, we just have to calculate vertices 1, 3, and 5 of the resulting area.

4 Performance Evaluation

To evaluate our proposal, we compare it with the improved rectangular intersection algorithm [4], described in section 2.1. Remember that in this algorithm network nodes send out the two corners determining the rectangle obtained by intersecting the rectangular areas received from other nodes.

4.1 Simulation Methodology

The simulation tool has been implemented in Adobe Flash 8. Fig. 6 shows the aspect of the user interface. The simulator gets the necessary parameters for each execution from a database, which stores simulation results as well.

We have carried out extensive numerical configurations, executing 100 simulations

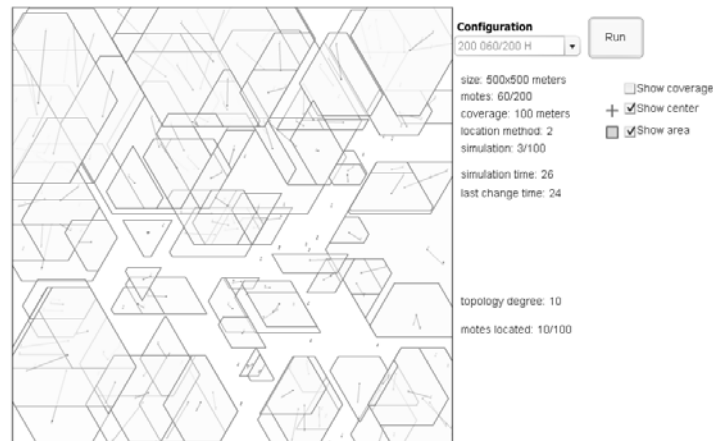


Fig. 6. Sensor Network Simulator showing localization areas and errors.

for each one and obtaining average results. In all cases, the nodes were distributed randomly in a square area of 500×500 meters; the coverage range of the nodes has been set to 100 meters. Although in the rectangular intersection algorithm the communication range and sensing range are different, we have assumed the same value for these two parameters.

For the configurations, we have varied the amount of network nodes (from 100 to 300) and the amount of nodes knowing their accurate location (from 10% to 70%).

We have used two metrics to evaluate our algorithm. The first one determines the distance between the real and estimated positions, denoted as localization error. Also, we compute the size of the final area in which the node estimates its position, denoted as localization area.

4.2 Simulation Results

Fig. 7 shows the error obtained by the rectangular intersection and hexagonal intersection algorithms (identified by “R” and “H”, respectively, in the legend), as a function of the percentage of network nodes which are equipped with a GPS receiver, and considering different amounts of nodes in the scenario.

In order to estimate the error for each node, we have obtained the Euclidean distance from the center of the polygon obtained –rectangle or pseudo-hexagon– by the localization algorithm to the real position of the node. For each simulation run we have computed the average error, and finally each point in the plots corresponds to the average error for a set of simulations. Fig. 7(f) shows the error rate considering both techniques. Each series in this plot has been obtained from the results shown in the previous plots.

We can observe that the error obtained when the hexagonal intersection technique is applied is about a 10% lower. In some cases, we obtain an improvement of a 30%. Additionally, this difference increases as the fraction of nodes equipped with GPS increases.

Fig. 8 shows the size of the localization area obtained by both algorithms, according again to the percentage of network nodes equipped with a GPS receiver, and for different node densities. Fig. 8(f) shows the way in which this area has been reduced. We can see that the area obtained by the hexagonal intersection algorithm is a 12% inferior. In this case, network density and the amount of beacons do not affect in the same way that in the case of the error.

To conclude this evaluation, Fig. 9 compares the error and the localization area obtained by both techniques, but in this case according to the amount of nodes in the scenario. We can observe again that the estimations are better when we increase the number of nodes in the scenario, the portion of beacons, and the hexagonal intersection is employed.

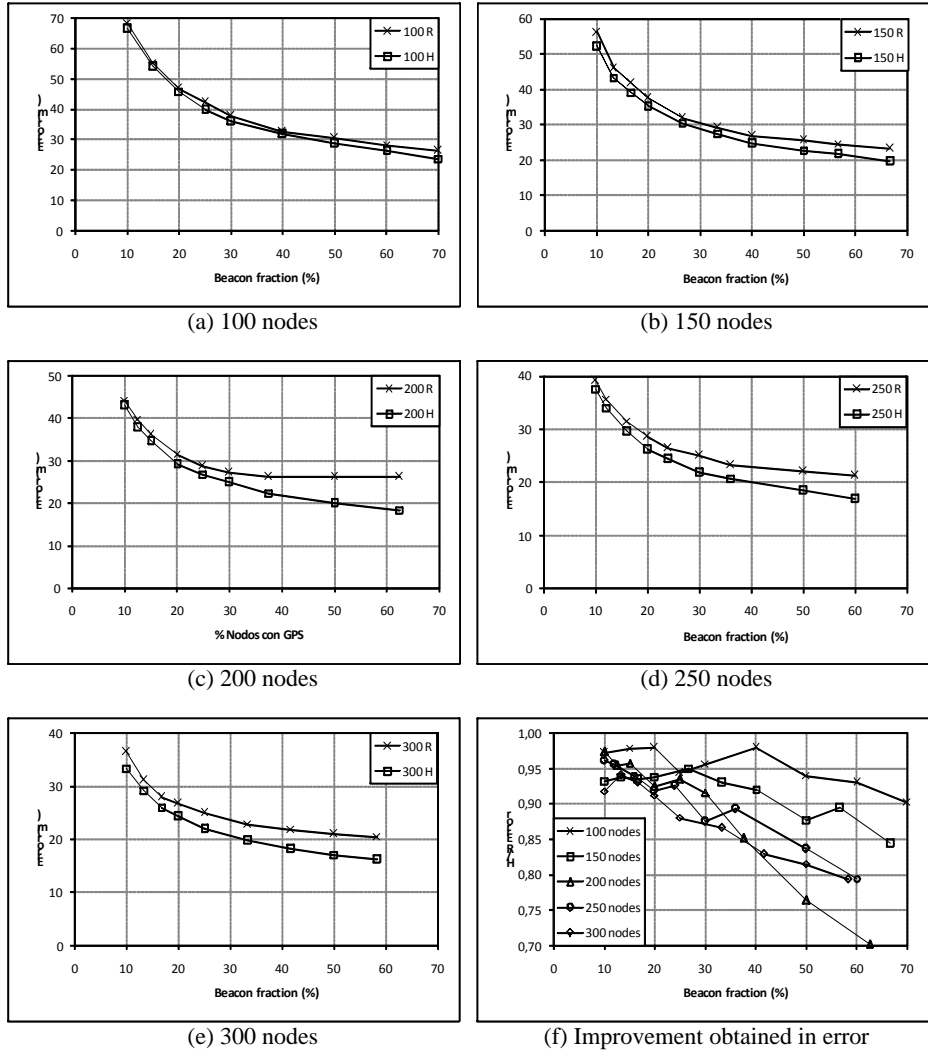


Fig. 7. Error obtained by the localization algorithms, according to beacon fraction.

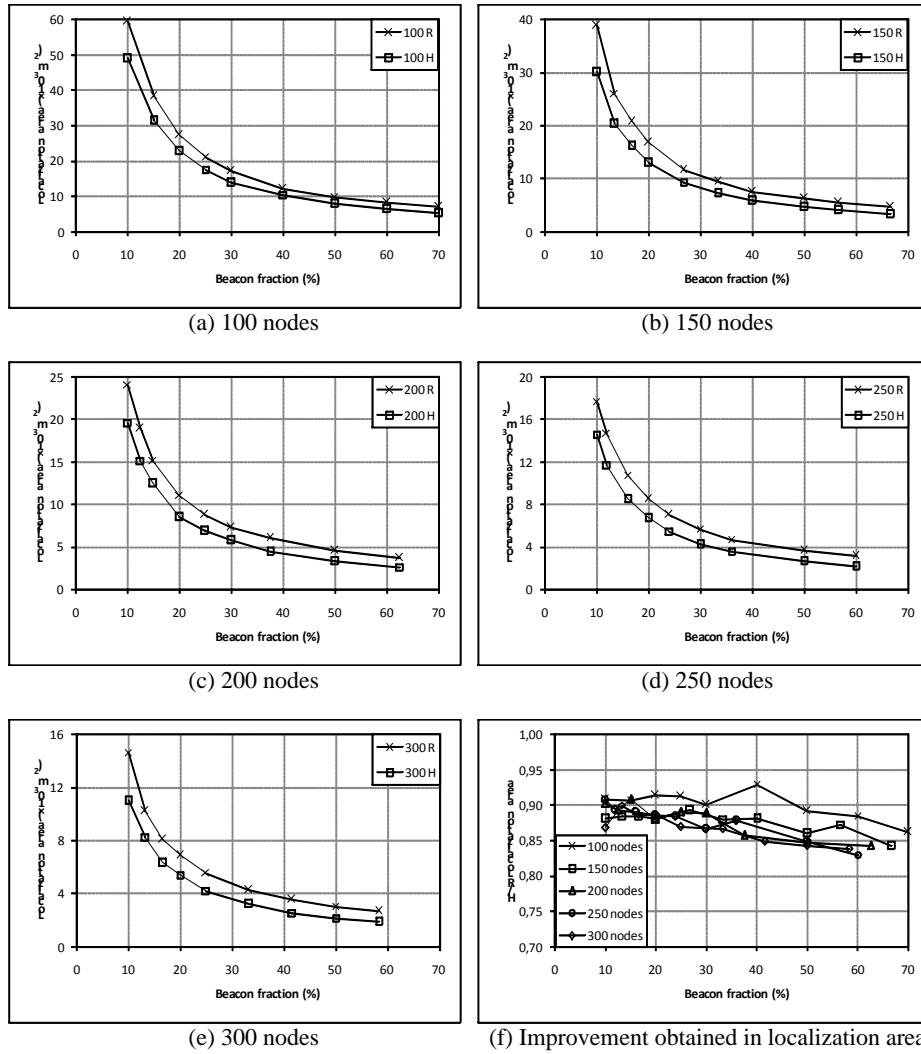
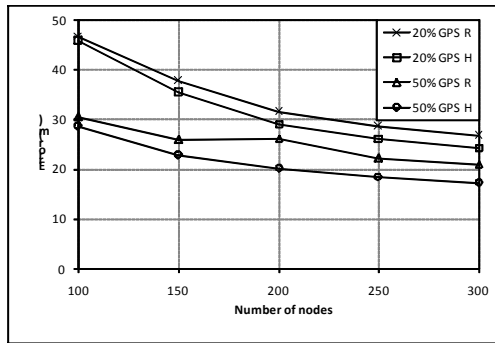


Fig. 8. Area obtained by the localization algorithms according to beacon fraction.

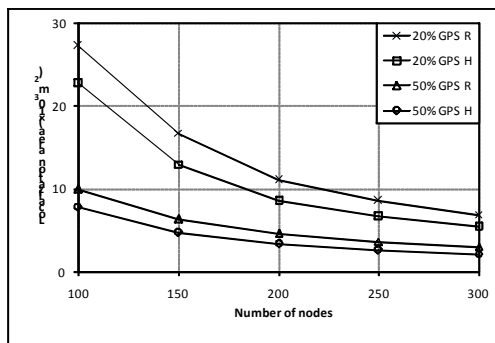
5 Conclusions and future work

In this work, we have shown that the use of a new localization technique called hexagonal intersection contributes to reduce the error obtained by some traditional localization algorithms for wireless sensor networks, based on assuming square coverage areas.

As future work, we plan to implement our proposal in real new generation devices, in which additional complexity is not a problem. Also, we plan to introduce a mobile beacon in the localization system.



(a) Localization error



(b) Localization area

Fig. 9. Error and area obtained by the localization algorithms according to number of nodes.

6 Acknowledgements

This work has been jointly supported by the Spanish MEC and European Commission FEDER funds under grants Consolider Ingenio-2010 CSD2006-00046 and TIN2006-15516-C04-02; by JCCM under grant PBC05-007-1; and by UCLM under grant TC20070061.

References

1. Bergamo, P., Mazzini, G.: Localization in sensor networks with fading and mobility. In Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002.
2. Beutel, J.: Geolocation in a picoradio environment. Thesis, 1999.
3. Buschmann, C., *et al.*: Radio Propagation-Aware Distance Estimation Based on Neighborhood Comparison. In Proceedings of the 4th European conference on Wireless Sensor Networks (EWSN'07), 2007.
4. Galstyan, A., *et al.*: Distributed online localization in sensor networks using a moving target. In Proceedings of International Symposium of Information Processing Sensor Networks (IPSN'04), April, 2004.
5. Hoffmann-Wellenhof, B. Lichtenegger, H., Collins, J.: GPS: Theory and Practice. 4th ed. Springer-Verlag, 1997.
6. Maroti, M., *et al.*: Radio interferometric geolocation. In Proceedings of the 3rd International conference on Embedded networked sensor systems (SenSys'05), 2005.
7. Nasipuri, A., Kai, L.: A directionality based location discovery scheme for wireless sensor networks. In First ACM International Workshop on Wireless Sensor Networks and Applications, September, 2002.
8. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom'00). ACM Press, 2000.
9. Sallai, J., *et al.*: Acoustic ranging in resource-constrained sensor networks. In International Conference on Wireless Networks, 2004.
10. Savarese, C., Rabaey, J.M., Beutel, J.: Locationing in distributed ad-hoc wireless sensor networks. In Proceedings of ICASSP'01, 2001.
11. Savvides, A., Han, C., Strivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom'01). ACM Press, 2001.
12. Savvides, A., Park, H., Srivastava, M.B.: The bits and the flops of the N-hop multilateration primitive for node localization problems. ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), 2002.
13. Sichitiu, M.L., Ramadurai, V.: Localization of wireless sensor networks with a mobile beacon. In Proceedings of MASS, 2004.
14. Simic S.N., Sastry, S.: A distributed algorithm for localization in random wireless networks. 2002, unpublished manuscript.
15. Whitehouse, K., Culler, D.: Calibration as parameter estimation in sensor networks. In proceedings of the 1st ACM international workshop on wireless sensor networks and applications (WSNA'02). ACM Press, 2002.
16. Whitehouse, K., *et al.*: The effects of ranging noise on multihop localization: an empirical study. In the 4th International Conference on Information Processing in Sensor Networks (IPSN'05), 2005.