On Improving the Precision of Localization with Minimum Resource Allocation

Frank Reichenbach, Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering,
University of Rostock,
Rostock-Warnemuende, Germany
Email: {frank.reichenbach,dirk.timmermann@uni-rostock.de}@uni-rostock.de

***Abstract*— Autonomous localization of nodes in wireless sensor networks is essential to minimize the complex self organization task and consequently enhancing the overall network lifetime. Recently, precise localization algorithms are impeded by multi path propagation of signals originated by reflections at walls or other objects in the environment. However, the localization error can be reduced by applying statistical methods that consider the mass of input information provided in large WSNs. In this paper we present the "Iterative DLS"-algorithm (iDLS), a new localization method that reduces the error of the initial position estimate by more than 46% and finally allows a precision of $3m$ (fieldsize: $100m \times 100m$) at noisy input. This is achieved by extending the known "Distributed Least Squares"-algorithm by a refinement-phase, where sensor nodes provide their initial position to neighbors. This algorithm places an absolute minimum of computational requirement on the resource constrained sensor nodes.**

I. Introduction

A Wireless Sensor Network (WSN) consists of a large number of tiny wireless devices, able to sense the environment, compute simple tasks and communicate with each other [1]. Due to the desired node size of only a few cubic millimeters, the most limited resource is the available energy [2]. Therefore, achieving a long lifetime of the sensor network requires low power hardware as well as optimized algorithms. The stochastical deployment process of sensor networks, e.g. distribution from an aircraft, leads to unknown positions at initialization. For several reasons, a node's position is very important. (i) Sensed data without a location where they were gathered are generally useless. (ii) Fully covered sensor networks enable energy aware geographic routing. (iii) Self configuration and self organization are key mechanisms for robustness and can be easily realized with position information. And (iv), in many applications such as target tracking, the position itself is the information of interest.

With respect to the demanded miniaturized size of a sensor node and its resource limitations, commercialized positioning technologies like the "Global Positioning System" (GPS [3]) or, from 2011, the European System "Galileo" [4] can not generally be used on all nodes. Therefore, it is a common practice to integrate an existing localization system on some more powerful nodes, further called beacons or reference points. All remaining nodes estimate their own position with range measurements to these beacons autonomously and become also reference points. As already known, the propagation characteristics of radio signals can vary with changes in the environment, which strongly influences the precision of range measurements.

So far, only beacons are considered in the localization process. But in large WSNs the number of beacons is only a small percentage of all nodes in the network. Thus, after the first position estimation sensor nodes can provide these coordinates to all other nodes as well. To reduce the negative influence of ranging errors, these coordinates are well suited to be included in a further localization, which is called *refinement*. Above all, this approach is promising for the following network constellations:

- Static networks, where the node's movement is slow or close to zero.
- Networks with only some static, but additionally some extremely mobile beacons. This can be for example a network without permanent beacons in the network, but a plane (here the beacon) that overflies the network and provides position information for only a short period of time.

For these networks, we developed the "Iterative Distributed Least Squares"-algorithm (iDLS) that is based upon the traditional DLS-algorithm. DLS features high precision at minimal power consumption and was introduced and verified in [5] and [6]. This paper presents iDLS that refines the first position estimate to decrease the localization error. This is achieved, while costs and resources for computation and communication are kept to a minimum.

This paper is subdivided as follows. In Section II we explain general localization methods in WSNs. After introducing the basics of 2D-positioning in Section III, we focus on the iDLS-algorithm in Section IV. Next, in Section V extensive simulations regarding the communication and computation overhead of iDLS are shown. Moreover, iDLS is compared with its direct competitor, the "Iterative Multilateration". Finally, this paper is concluded in Section VI.

II. Related Work

A. General Classification

Most of the *approximate* localization algorithms need only few resources, but estimate a position with a relatively high localization error. Different approximate (also called coarse-grained) localization approaches exist in the literature (see

1095-2055/07/$25.00 ©2007 IEEE

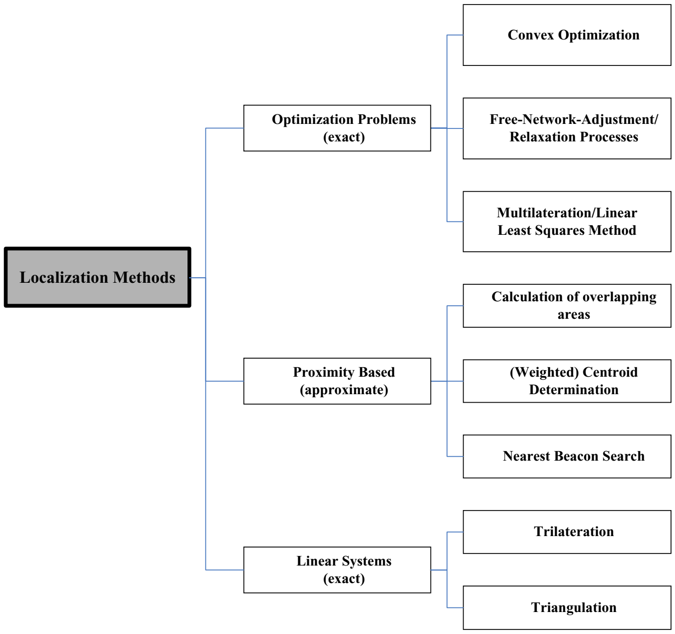

Fig. 1. Classification of the basic localization techniques in sensor networks.

Fig. 1). For example, Tian et al. completely avoid distances in their approach [7]. To do so, triangles are combined with all beacon coordinates in the field. Then every sensor node uses a "Point in Triangulation"-test (PIT) to determine on which triangle surface it is. In this PIT-test only neighbor relations are used. A following intersection test with all filtered triangles shrinks the region where the sensor node is probably located. In another approach by Bulusu et al. every sensor node maintains a list with all beacons in transmission range. The beacons are deployed on an equidistant grid of points [8]. With all received positions, sent by beacons, sensor nodes calculate the centroid as its own position. This approach was extended with distances in form of weights, which further improved the precision [9]. Lastly, a very simple idea is to take the nearest beacon position as position estimate.

In contrast, *exact* (or fine-grained) localization of a sensor node features high precision and is based on solving either (i) linear systems with only the minimum required number of beacons or (ii) a high amount of beacons and distances to them, which leads to optimization problems. In more detail, with at least three beacons (in two-dimensions), sensor nodes estimate their positions via trilateration. More beacons than required result in an over determined system of equations that must be solved with e.g. a least-squares method (multilateration). Although a multilateration produces accurate results, it is complex and resource-intensive and consequently not feasible on resource-limited sensor nodes. Nevertheless, Savvides et al. describe different approaches (Atomic, Iterative, Collaborative Multilateration) for fine-grained localization in wireless sensor networks [10]. Kwon et al. presented a distributed solution using least squares, whereby errors in acoustic measurements can be reduced [11]. Ahmed et al. published a new approach to combine the advantages of absolute and relative localization methods [12]. Moreover, Karalar et al. developed a low-energy system for positioning using least squares, which can be integrated on individual sensor nodes [13]. Further approaches introduce constraints [14] or a relative coordinate system [15]. Zang et al. completely avoid noisy distances [16]. A general overview of distributed positioning systems is given by Langendoen and Reijersin [17].

B. Iterative Methods in Detail

There are different definitions of the term *iterative*. On the one hand, a system of equations, built from beacons in the network, can be solved by iterative methods like splitting techniques. Here, the mathematical method to estimate an initial position itself is meant to be iterative. But in this paper we focus on the process that follows after the initial position estimate. In this process initial positions of sensor nodes are also considered in localization with the aim to improve the precision. In a step by step refinement more and more sensor nodes broadcast their initial position to their neighbors, which include them in the localization process.

Savvides et al. proposed in [10] the "Iterative Multilateration". In this algorithm a distributed "Atomic Multilateration" is started with all beacons in neighborhood. If a node estimated a position in this first phase, then this position is sent to all other nodes. A node that receives a position from a neighbor, immediately repeats the multilateration with this new position included. This process continuous until all nodes are implied. In an approach by Savarese et al. in [18] the first position is determined by multilateration combined with hop-counts to all beacons in the network. The number of hops to every beacon is basis for finding an absolute distance by the "DV-Hop"-method. After this "Hop-Terrain"-phase, a "Refinement"-phase follows, where all neighbor positions within one-hop away are included in a triangulation. This is repeated iteratively until the difference between the old and the new position estimate is marginal. In addition to both approaches above, there exist algorithms that are based on building relative coordinate systems. Although here the idea is to completely avoid beacon nodes, this technique is also iterative [19].

Most of the before mentioned approaches assume that every sensor node has enough resources to estimate its position by multilateration with many reference points. However, multilateration with many reference points can become highly complex. For example, with 100 reference points already $0.8kB$ of $2kB$ of internal memory[1] are reserved [6], which is equal to 40% of the total memory[2]. Moreover, precalculated information can be used in a refinement, instead of repeating the multilateration in every step, which wastefully consumes valuable memory. In this paper we show a resource aware localization algorithm that considers the issues above and achieves highest precision at minimal computation effort and memory usage.

[1]We assume floating point representation of an element that allocates 4 Byte memory.

[2]For a typical sensor node platform like the Chipcon CC1010.

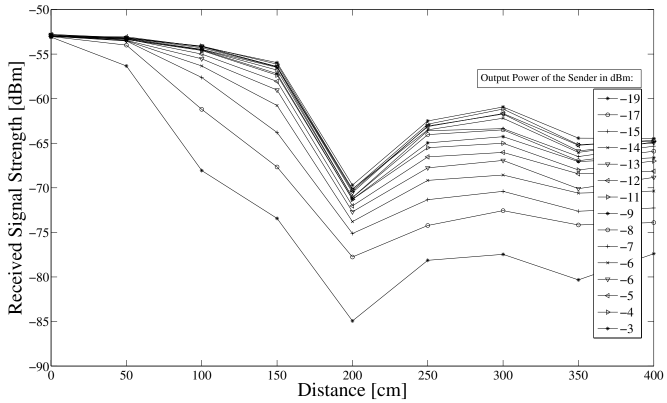

Fig. 2. Indoor RSS-measurements between a sending and a receiving sensor node at different distances. Every curve represents an output power level at the sender (Chipcon CC1010). Obtaining distances from these curves is highly defective, which increases the localization error.

III. Background: Localization Primitives

A. Distance Estimation Techniques

With only few exceptions localization methods require precise range or distance measurements. These distances are gathered by different measuring techniques like e.g. time of arrival (ToA), received signal strength (RSS), and differences of the signal's phase. Signals can be either radio, infrared or ultra sound. If all these measurements would be error-free, the coordinates can be simply computed by a trilateration with three beacons. But, multi-path effects lead, especially indoor, to high measurement errors. Fig. 2 illustrates characteristic signal strength curves of a Chipcon CC1010 sensor node platform. Although determining distances from these curves is highly defective, reading the received signal strength is supported by almost every transceiver-hardware that makes this to a beneficial and cost efficient solution.

Due to relatively short transmission ranges of all nodes, which are demanded in WSNs, multi-hopping mechanisms may be used to establish links between nodes that are not direct neighbors. Without unconnected nodes in the network this allows establishing a communication between all nodes in the network. By hopping over multiple nodes more than one distance measurement must be taken into account. Different techniques to solve this problem, such as DV- [20] or amorphous-hopping [21], exist in literature. If obstacles increase the number of hops on the route and prevent the shortest path discovery, distances are also highly defective. For this problem the literature offers obstacle avoidance algorithms like in [22].

B. Basics: "Distributed Least Squares"-Algorithm

In the following, we explain DLS, which is used to produce an initial estimate. DLS starts with the creation of an over-determined system of non-linear Euclidean distances of the form $(x - x_i)^2 + (y - y_i)^2 = r_i^2$ with $i = 1, 2, \ldots, m$ (where m is the number of beacons, (x, y) is the required position, (x_i, y_i) is the position of beacon i and r_i is the distance between them). This system of equations is linearized with the method described in [23]. This leads to the form $A\mathbf{x} = \mathbf{b}$, where A is the coefficient matrix, $\mathbf{b}$ is the right side vector and $\mathbf{x}$ is the solution vector. By applying the Least Squares Method we obtain the known Normal Equation $\mathbf{x} = \left(A^T A\right)^{-1} A^T\mathbf{b}$.

The matrices in this equation have two important benefits. First, all elements in the coefficient matrix A are generated by beacon positions $B_1(x, y) \ldots B_m(x, y)$ only. By assuming that we can establish communication links between all sensor nodes and all beacons (e.g. by multi-hop-techniques), then matrix A is the same on every sensor node. Second, the vector $\mathbf{b}$ contains the distances between sensor nodes and beacons $r_1 \ldots r_m$ that must be estimated on every sensor node independently. Given these facts, the Normal Equation can be split into two parts - a more complex part, the *precalculation*: $A_p = \left(A^T \cdot A\right)^{-1} \cdot A^T$ and a simple part: $A_p \cdot \mathbf{b}$, which is further called the *postcalculation*. Here, the precalculation is executed on one powerful base station, which additionally avoids high redundancy, because this precalculation would normally be executed on all sensor nodes separately. But it is very important to emphasize that the precalculation is identical on every sensor node. Thus, it is calculated only once, conserving expensive energy resources. The simple postcalculation is then executed on every sensor node with its individual distance measurements to all beacons. This approach complies with two important design strategies for algorithms in large sensor networks - a **resource-aware** and **distributed** localization procedure. Finally, this can be achieved with less communication overhead required for other exact algorithms.

At this point, we briefly describe the algorithm process. DLS is divided into three phases, which are shown in Fig. 3. In phase 1, all beacons send their position $B_i(x, y)$ hop-by-hop over their beacon neighbors to the base station. Then, in phase 2, the base station starts generating the initial matrices and computes A_p. The result is sent over beacons to all sensor nodes, which in phase 3 estimate their position after measuring the distances to all beacons and executing the postcalculation. For the new refinement-phase the following mathematical techniques must be described.

C. An Extension: Adding a New Position Estimate

If a new position must be included in the localization process, there exist an incomplex approach - the "Recursive Least Squares Method" (RLSM) [24]. The mathematical formulation of this problem leads to adding a new line $\mathbf{w}$ to A, which results in $\tilde{A}$. If the covariance matrix $C = (A^T A)^{-1}$ is known, then an updated covariance matrix can be calculated with the "Sherman-Morrison"-formula [25]: $\tilde{C} = C - \frac{1}{1+\mathbf{w}^T\mathbf{u}}\mathbf{u}\mathbf{u}^T$, where $\mathbf{u} = C\mathbf{w}$. With that new covariance matrix the new position $\tilde{\mathbf{x}}$ can be determined $\tilde{\mathbf{x}} = \mathbf{x} + \tilde{C}\mathbf{w}\left(\mathbf{k} - \mathbf{w}^T\mathbf{x}\right)$, where $\mathbf{k}$ is the new line in $\mathbf{b}$. To summarize, with both, the precalculated matrix A_p from DLS and the new covariance matrix C, the initial estimate and additionally a refinement can be accomplished, which leads to the algorithm details of iDLS.

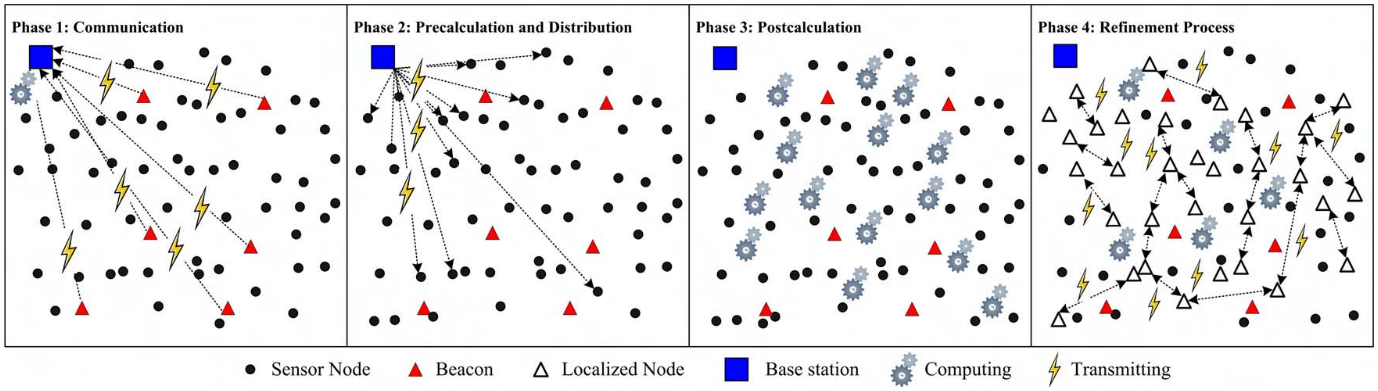

Fig. 3. Overview about the phases of the iDLS-algorithm.

IV. "ITERATIVE DLS"-ALGORITHM

A. Algorithm Description

On basis of DLS a refinement-phase is added (see Fig. 3). In this phase all sensor nodes, that already estimated an initial position, send this position via an info-packet to all neighbors in transmission range. To avoid a complete new multilateration, which wastes resources and energy, there exist the incomplex RLSM that was briefly explained in Section III-C. The complete iDLS-algorithm is given as follows:

- **Phase 1: Initialization**
  - All beacons send their position $B(x, y)$ to the base station.
- **Phase 2: Precalculation (central) and Distribution**
  - Base station builds matrix A and vector $\mathbf{d}_p$.
  - Starting the precalculation of matrices A_p, C. - Base station sends matrices A_p, C and vector $\mathbf{d}_p$ to all sensor nodes.
  - Sensor nodes measure distances to all beacons.
  - Sensor nodes receive matrices A_p, C and vector $\mathbf{d}_p$
- **Phase 3: Postcalculation (distributed)**
  - Sensor nodes build vector $\mathbf{b}$ and estimate their own position.
- **Phase 4: Refinement (distributed)**
  - Localized sensor nodes send their position estimate to its neighbors, which measure the distance to them.
  - If a new position was received, RLSM is executed.

This is repeated until all received info-packets are processed. Although we limit the info-packets only to the neighbors in one-hop range, it is also possible to forward the packets to neighbors more hops away. It must be noted that this process significantly increases the communication overhead at minimal precision improvement.

B. Complexity, Communication and Required Resources

From here on, we compare iDLS to its direct competitor, the "Iterative Multilateration" (iMUL), which was explained in Section II-B.

Complexity: In contrast to iMUL with a complexity of $15m - 5$ (see [5]), where m is the number of reference points, the process of adding a new position with RLSM needs constant 50 floating point operations (flops). For example, with 100 reference points the refinement-process of iDLS needs ca. 90% less computation effort than iMUL. Lastly, iDLS is highly scalable because the number of reference points does not increase the complexity of one refinement-step.

Communication: Beside A_p[3] and d_p with $3m-3$ elements, additionally the covariance matrix C with exact four elements must be transfered in phase 2. Moreover, in the refinement-phase info-packets with two elements must be sent.

Memory: It must only be stored the current covariance matrix with four elements and the current position with two elements. Again, with 100 reference points less than 2% memory must be reserved compared to iMUL (iMUL needs $4m - 4$ elements to be stored, see [5]).

The energy consumption of iDLS is mainly effected by the number of iterations, because in every iteration step an info-packet must be received, the RSS must be measured and an update must be computed. This will be examined in the following Section.

V. SIMULATION AND DISCUSSION

Simulations were performed in J-Sim, a sensor network simulation framework by Tyan et al. [26], in which we added a more complex and more realistic energy model respectively. Our energy model considers (i) power-mode dependent energy consumption with sleep and active mode, (ii) switching energy from sleep to active mode, (iii) distance dependent transmission of packets, (iv) computation complexity, (v) distance estimations with RSS measurements, and (vi) position estimation via GPS on beacons[4]. The specific energy parameters are based on the MICA2-mote, which is currently the most popular sensor node platform. Moreover, info-packets are not forwarded. That means, only direct (one-hop) neighbors are considered in the refinement process for both, iDLS and iMUL. All nodes were uniformly distributed over the sensor field. To simulate defective distances, all exact distances were

[3]Matrix A_p is only important for the initial estimate and can be deleted later.

[4]Details of our energy model or the source code can be obtained from frank.reichenbach@uni-rostock.de.

falsified. That means, distances are stochastically generated on basis of a Gaussian-distribution with the exact distance as mean value and a variance of 10. The following configuration was used:

Parameter	Value
Sensor field dimensions [m]	100×100
Number of sensor nodes	300
Number of beacons	10
Transmission range (sensor nodes) [m]	15
Transmission range (beacons) [m]	75
Simulation time *t* [s]	200

iDLS: Fig. 4 shows the accumulated energy consumption of all nodes in the field. There, iDLS started phase one at $65s$ by sending the first position-packet by a beacon to the base station. After that, phase 2 began where the first DLS-packet from a beacon was received by a sensor node at $76.8s$. Next, starting from $126.8s$, every sensor node sent an info-packet for refinement of its neighbors. This refinement process ended at $139.9s$ with receiving 14131 packets in total.

iMUL: The first info-packet was sent by a beacon at $65.9s$ simulation time. Afterward, all beacons had sent their packets, starting from $118.5s$ sensor nodes sent and received neighbor positions. In this communication phase 16437 info-packets were received by all sensor nodes. Fig. 5 shows all sent and received packets of the sensor nodes. The sensor nodes received fewer packets with iDLS, because in the first phase of iDLS traffic occurs between beacons, only.

In comparison, iDLS consumed not significantly less energy than iMUL. The reason is the high energy consumption of communication, which is outstanding in relation to computation energy. However, compared to iMUL, a very simple calculation is executed on every sensor node only. In contrast, iMUL needed to compute always a complete multilateration, which can be high complex with many reference points and a complexity of $15m - 5$, particularly at the end of refinement, where ca. 55 nodes were included. In contrast, the RLSM needed constant 50 flops. The refinement process of both algorithms leads to a high communication overhead, which dominates the overall consumption.

Both algorithms produced an average error of $\approx 3.05m$ (see Fig. 6). There is no difference, due to the same statistical basis both algorithms use. However, the error of iDLS was increased significantly from $5.71m$ before and $3.05m$ after refinement, which is an improvement of at least 46.58%. Obviously, the error decreased fast at the beginning of the refinement and slower at the end, where it converged at a lower bound. Nevertheless, the energy consumption in the refinement-phase is relatively constant. Consequently, there exist a trade-off between the number of considered info-packets and the localization error. Thus, the refinement process consumed $7.8J$ of the networks total energy. To summarize, iDLS achieved the same precision and energy consumption, but used minimal resources on the sensor node, which makes iterative localization principally feasible.

VI. Conclusion

We presented iDLS, a precise localization algorithm that improves the initial position estimate by 46.6% by considering neighbor information. Although, distances are highly defective, iDLS achieved a small localization error of 3%. In a comparison of iDLS with an "Iterative Multilateration", it was shown that iDLS requires 98% less memory and 90% less computation effort, whereas not more energy was consumed or the precision decreased. However, a refinement-phase is never for free. This leads to a trade-off between energy consumption and precision. In the end, the individual application decides the affordable degree of precision.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, vol.2, pp. 102-114, 2002

[2] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, A. Chandrakasan, "Low-Power Wireless Sensor Networks", *International Conference on VLSI Design*, pp. 205-211, 2001

[3] J. Gibson, "The Mobile Communications Handbook", *CRC Press*, 1996

[4] R. Bill, C. Cap, M. Kohfahl, T. Mund, "Indoor and Outdoor Positioning in Mobile Environments, A Review and some Investigations on WLAN Positioning", *Geographic Information Sciences*, volume 10, number 2, pp. 91-98, 2004

[5] F. Reichenbach, A. Born, D. Timmermann, R. Bill, "A Distributed Linear Least Squares Method for Precise Localization with Low Complexity in Wireless Sensor Networks", *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'06)*, pp. 514-528, 2006

[6] F. Reichenbach, A. Born, D. Timmermann, R. Bill, "Splitting the Linear Least Squares Problem for Precise Localization in Geosensor Networks", *4th International Geographic Information Science (GIScience 2006)*, LNCS 4197, pp. 321-337, 2006

[7] T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in *In Proceedings of the 9th annual international conference on Mobile computing and networking, MobiCom '03,* San Diego, CA, USA, 2003, pp. 81-95.

[8] N. Bulusu, GPS-less low cost outdoor localization for very small devices, in *In IEEE Personal Communications Magazine,* 2000, vol. 7, pp. 28-34.

[9] J. Blumenthal, F. Reichenbach, D. Timmermann, Precise Positioning with a Low Complexity Algorithm in Ad hoc Wireless Sensor Networks, in *PIK - Praxis der Informationsverarbeitung und Kommunikation,* 2005, vol. 28, pp. 80-85.

[10] A. Savvides, C.C. Han, M.B. Srivastava, "Dynamic Fine Grained Localization in Ad-Hoc Networks of Sensors", *7th ACM MobiCom (Rome, Italy,2001)*, pp. 166-179, 2001

[11] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, G. Agha, "Resilient Localization for Sensor Networks in Outdoor Environments", *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pp. 643-652, 2005

[12] A. A. Ahmedm, H. Shi, Y. Shang, "SHARP: A New Approach to Relative Localization in Wireless Sensor Networks", *Second International Workshop on Wireless Ad Hoc Networking*, pp. 892-898, 2005

[13] T. C. Karalar, S. Yamashita, M. Sheets, J. Rabaey, "An Integrated, Low Power Localization System for Sensor Networks", *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 24-30, 2004

[14] L. Doherty, L. Ghaoui, and K. Pister, Convex position estimation in wireless sensor networks, in *Proceedings of the IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01),* 2001, pp. 1655-1663.

[15] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad hoc networks, in *In Cluster Computing,* 2002, vol. 5, pp. 157-167.

[16] L. Zhang, Q. Cheng, Y. Wang, S. Zeadally, "Landscape: A High Performance Distributed Positioning Scheme for Outdoor Sensor Networks", *Wireless And Mobile Computing, Networking And Communications*, pp. 430-437, 2005

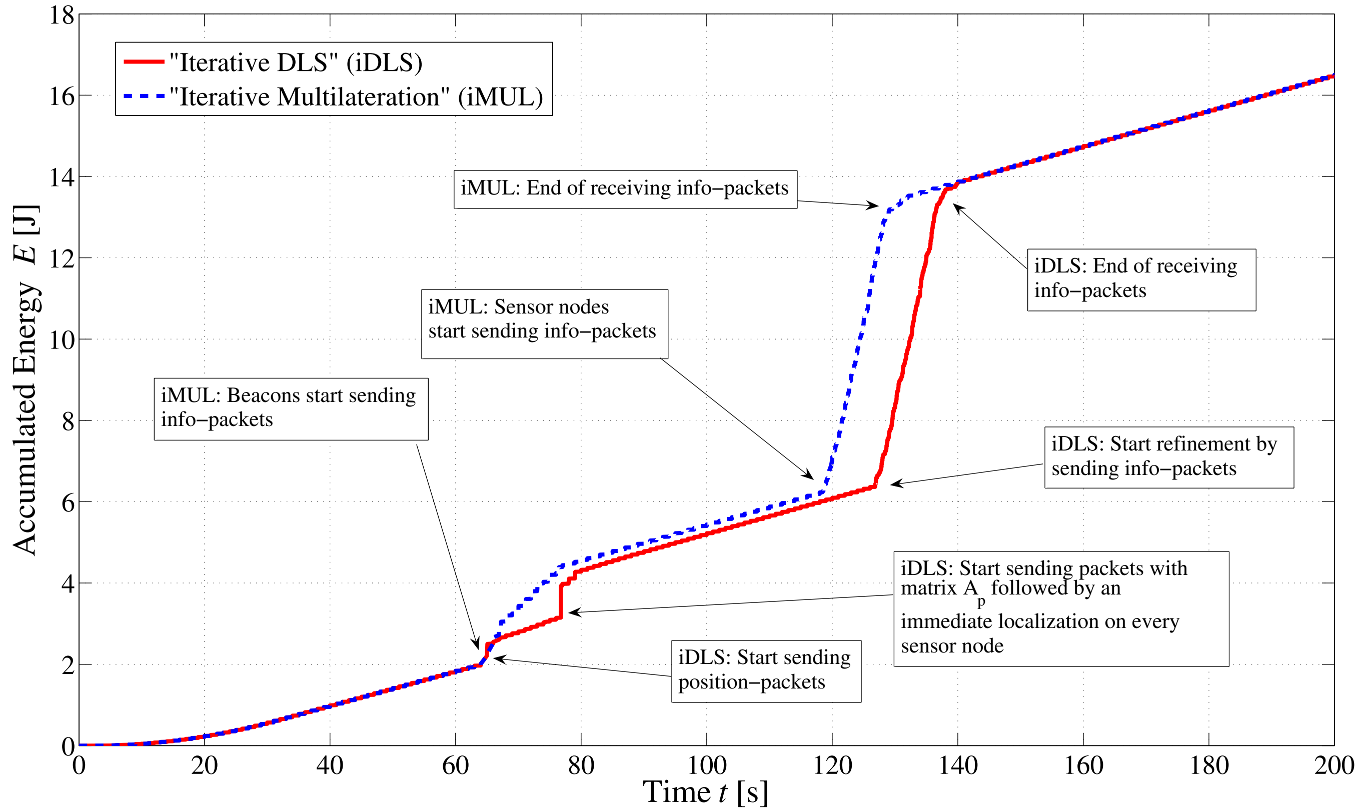

Fig. 4. Accumulated energy of all nodes in the network for "Iterative DLS" and "Iterative Multilateration".

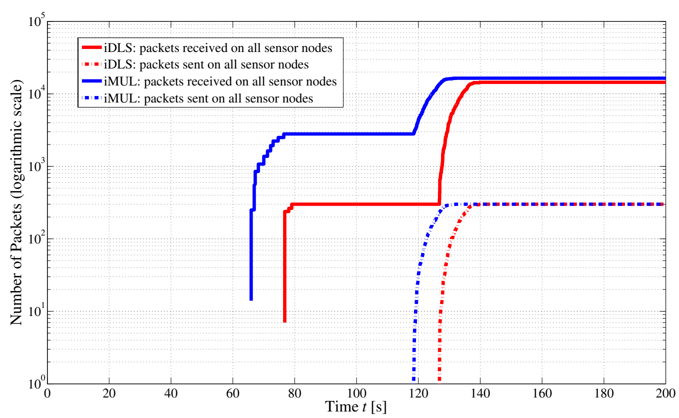

Fig. 5. Communication traffic of all sensor nodes (logarithmic) in the network for "Iterative DLS" (red) and "Iterative Multilateration" (blue).

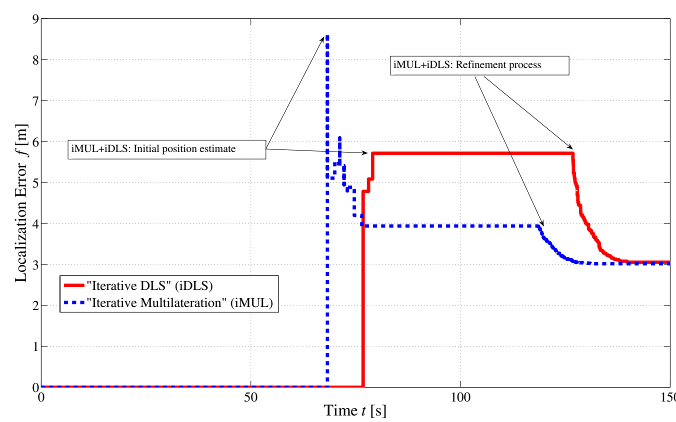

Fig. 6. Absolute localization error over simulation time. After the first position estimate (high peak), the refinement process starts.

[17] K. Langendoen. N. Reijers, "Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison", *Computer Networks (Elsevier), special issue on Wireless Sensor Networks*, volume 43, number 4, pp. 499-518, 2003

[18] C. Savarese, J. Rabaey, K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-hoc Wireless Sensor Networks", *USENIX Annual Technical Conference*, pp. 317-327, 2002

[19] Oh-Heum Kwon, Ha-Joo Song, "A New Map Stitching Method for Anchor-free Localization in Wireless Sensor Networks", *International Conference on Computer and Information Technology*, pp. 236, 2006

[20] D. Niculescu, B. Nath, "Ad hoc positioning system (APS)", *IEEE Global Telecommunications Conference (GLOBECOM 2001)*, pp. 2926-2931, 2001

[21] L. Kleinrock, J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or why Six is a Magic Number", *National Telecomm Conference*, pp. 4.3.1-4.3.5, 1978

[22] Frank Reichenbach, Ralf Salomon, Dirk Timmermann, "Distributed Obstacle Localization in Large Wireless Sensor Networks", *International Wireless Communications and Mobile Computing Conference (IWCMC 2006)*, pp. 317-1322, 2006

[23] W. Murphy, W. Hereman, "Determination of a Position in Three Dimensions using Trilateration and Approximate Distances", *Technical Report MCS-95-07*, 1995

[24] G. H. Golub, C. F. Van Loan, "Matrix Computations", *The Johns Hopkins University Press*, 1996

[25] Ake Björck, "Numerical Methods for Least Squares Problems", *Mathematics Society for Industrial and Applied Mathematics*, 1996

[26] Hung-ying Tyan, "J-Sim", *http://www.j-sim.org/*, 2006