Exact Localization on Resource Limited Sensor Nodes - Making it Feasible -

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

ABSTRACT

The random deployment process and the unpredictable movement of sensor nodes lead to a high demand for an exact and reliable self localization process. Existent methods are mostly not feasible on strongly resource limited sensor nodes with an absolute minimum of energy. Thus, this paper describes the "Distributed Least Squares (DLS)"-algorithm that reduces the needed amount of energy to a minimum by distributing the complex localization process. DLS is a resource-aware localization method that achieves 47% computation savings and 86% energy savings compared to the reference method, a "Fully Distributed Multilateration" on every sensor node. In the end, DLS significantly extends the overall network lifetime.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications

General Terms

Algorithms, Design, Theory

Keywords

wireless sensor networks, localization

1. INTRODUCTION

Wireless sensor networks (WSN) are composed of hundreds of tiny electronic devices, able to sense the environment, compute simple tasks and communicate with each other. Gathered information (e.g. temperature, humidity etc.) are transmitted in a multi hop fashion over direct neighbors to a data sink, where the data is interpreted [1]. With methods such as self configuration and self organization the network reacts to node failures.

Due to the desired node size of only a few cubic millimeters, the dimensions of the communication module and the battery are critical. Consequently, the scarcest resource within a network is the available energy [2]. Therefore, achieving a long lifetime of the sensor network requires low power hardware as well as optimized algorithms.

After deploying the sensor network over an area of interest, initially the sensor nodes carry no position information. Sensor information are only useful if combined with their geographical position. Possible positioning technologies are the Global Positioning System (GPS), the Global System for Mobile Communication (GSM) or soon the European System Galileo [3],[4],[5]. However, these systems are unsuitable for miniaturized sensor nodes and could only be used for a small number of nodes, due to the size of the hardware, the high prices and the high energy requirements. Thus, it is a common technique to integrate an existing localization system on some more powerful nodes, further called beacons. Then, all remaining nodes estimate their own position with measurements such as distances to these beacons autonomously. A node's position is very important, because (i) sensed data without a location where they were gathered are generally useless, (ii) full 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 the position itself is the information of interest.

In this paper we present a new approach to energy-saving determination of unknown coordinates with a higher precision compared to approximate positioning methods [6]. Using the "Distributed Least Squares-(DLS)"-algorithm, all calculations are split between the resource-limited sensor nodes and the high-performance base station.

This paper is structured as follows: In Section 2 we give a basic overview of the methods for localization in wireless sensor networks. In Section 3 we describe the position estimation based on relationships to known points. Next, we present in Section 4 our new DLS-algorithm to split the least squares method with the aim to minimize the load on the sensor nodes. Furthermore, the complexity of DLS is analyzed in Section 5, simulated in Section 6 and discussed in Section 7. We finally conclude the paper with Section 8.

2. RELATED WORK

Considering energy constraints in sensor networks, the group of approximate algorithms consumes less power, but estimates a position with a higher localization error (see Fig. 1). Different approximate (also called coarse-grained) localization approaches exist in the literature [7, 6, 8, 9, 10].

Exact localization of a sensor node features high precision and is based on solving a linear system of equations with coordinates of the beacons and distances to them. With at least three beacons, required in 2-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). The multilateration produces accurate results, however it is complex and resource-intensive and therefore not feasible on resource-limited sensor nodes. Nevertheless, different authors deals with reducing the complexity of these methods [11, 12, 13, 14, 15].

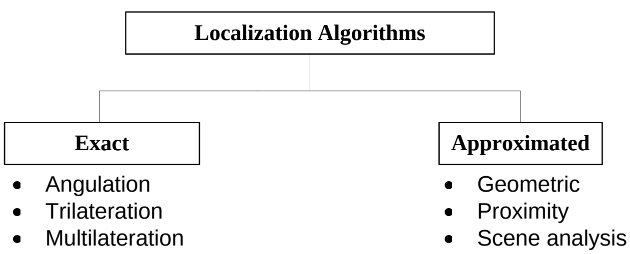

Figure 1: Classification of common localization techniques in sensor networks.

We demand exact localization methods that work on tiny sensor nodes with highly limited energy resources. To achieve this, we transfer the complex calculations such as matrix multiplication or matrix inversion to the base station. Consequently, only simple calculations have to be executed on the sensor nodes. Additionally, we reduce the communication and memory requirements through optimizations of the proposed algorithm.

3. BACKGROUND: MULTILATERATION

Estimating the position of an unknown point $P(x, y)$ requires at least three known points in two-dimensions. With m known coordinates $B(x_i, y_i)$ and its distances r_i to them we obtain:

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \quad (i = 1, 2, \ldots, m) \tag{1}$$

This system of equations must be linearized by using the j'th equation of (1) as the linearization tool. By adding and subtracting x_j and y_j to all other equations this leads to:

$$\begin{array}{l}(x - x_j + x_j - x_i)^2 + (y - y_j + y_j - y_i)^2 = r_i^2 \\ (i = 1, 2, \ldots, j-1, j+1, \ldots, m)\end{array} \tag{2}$$

With the distance r_j (r_i), the distance between the unknown point and the j'th (i'th) beacon, and the distance d_{ij}, the distance between beacon B_i and B_j, this leads, after resolving and simplifying, to:

$$\begin{array}{l}(x - x_j)(x_i - x_j) + (y - y_j)(y_i - y_j) \\ = \frac{1}{2}\left[r_j^2 - r_i^2 + d_{ij}^2\right] = b_{ij}\end{array} \tag{3}$$

Because it is not important which equation we use as a linearization tool, $j = 1$ is sufficient. This is equal to choosing the first beacon and if $i = 2, 3, \ldots, m$ this leads to a linear system of equations with $m-1$ equations and $n = 2$ unknowns.

$$\begin{array}{rcl}(x - x_1)(x_2 - x_1) + (y - y_1)(y_2 - y_1) & = & b_{21} \\ (x - x_1)(x_3 - x_1) + (y - y_1)(y_3 - y_1) & = & b_{31} \\ & & \vdots \\ (x - x_1)(x_m - x_1) + (y - y_1)(y_m - y_1) & = & b_{m1}\end{array} \tag{4}$$

This system of equations can be written in the matrix form $A\mathbf{x} = \mathbf{b}$ with:

$$A = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_m - x_1 & y_m - y_1 \end{pmatrix},$$

$$\mathbf{x} = \begin{pmatrix} x - x_1 \\ y - y_1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \frac{1}{2}\left[r_1^2 - r_2^2 + d_{21}^2\right] \\ \frac{1}{2}\left[r_1^2 - r_3^2 + d_{31}^2\right] \\ \vdots \\ \frac{1}{2}\left[r_1^2 - r_m^2 + d_{m1}^2\right] \end{pmatrix} \tag{5}$$

Now this basic form has to be solved using the linear least squares method. Due to the fact that overdetermined systems of equations with $m >> n$ have no exact solution for $A\mathbf{x} = \mathbf{b}$, we have to apply the L2-norm. This is also called the Euclidean norm, which minimizes the sum of the squares of the residuals:

$$\begin{array}{c} Minimize \\ x \in \Re^n \end{array} \quad ||A\mathbf{x} - \mathbf{b}||_2. \tag{6}$$

A trivial solution of the least squares problem is to reconvert after $\mathbf{x}$. In this case, the unique solution of $A\mathbf{x} \approx \mathbf{b}$ is given by:

$$||A\mathbf{x} - \mathbf{b}||_2 \rightarrow A^T A\mathbf{x} = A^T\mathbf{b}. \tag{7}$$

Solving normal equations is a good choice if the linear system has many more equations than unknowns, i.e. $m >> n$, because after the multiplication $A^T A$ the result is only a quadratic $[n \times n]$-matrix. This simplifies the following computation and makes it easier to be implemented in software.

4. ALGORITHM DESCRIPTION

DLS builds on the mathematical formulations introduced in the background section. By using the linearization tool the matrices in Equ. (5) 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. We assume in the first instance that all sensor nodes can establish communication links between all beacons, then matrix A is the same on every sensor node. Second, vector $\mathbf{b}$ contains distances between sensor nodes and beacons $r_1 \ldots r_m$, which have to be estimated on every sensor node independently. The result is that the normal equations 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: $A_p \cdot \mathbf{b}$, further called the *postcalculation*. Here, the precalculation is executed on one high performance node, which additionally avoids high redundancy, because normally this precalculation has to be executed on all sensor nodes separately. It is very important to emphasize that the precalculation is identical on each sensor node. Thus, it is calculated only once, conserving expensive energy resources. The simple postcalculation is then executed on each 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. 2.

- **Phase 1: Initialization**
  - All beacons send their position $B(x,y)$
  to the base station.

- **Phase 2: Complex Precalculation (central)**
  - Base station builds matrix A and vector $\mathbf{d}_p$.
  - Starting the complex precalculation of matrix A_p.

- **Phase 3: Simple Postcalculation (distributed)**
  - Base station sends matrix A_p
  and vector $\mathbf{d}_p$ to all sensor nodes.
  - Sensor nodes determine the distance to
  every beacon $r_1..r_m$.
  - Sensor nodes receive matrix A_p
  and vector $\mathbf{d}_p$, built vector
  $\mathbf{b}$ and estimate their own position
  $P_{est}(x,y)$ autonomously.

5. THEORETICAL ANALYSIS

5.1 Computation Complexity

At this point it is important to compare the complexity of the normal equations with the postcalculation. In order to define the complexity mathematically, we count the number of floating point operations (flops), which is a commonly used method in literature. It is described in [16] that the total complexity for the least squares method with m beacons and $n = 2$ unknowns is $15m - 5$ flops. That means, with 100 beacons and the summation of x_1 and y_1 we need 1497 floating point operations. Now it can be determined what we save on the sensor nodes without complex precalculations regarding only the remaining postcalculation. On the basis of (7), the base station precalculates $A_p = (A^T A)^{-1} A^T$ and $\mathbf{d}_p = \mathbf{d}^2$. The matrix A_p and vector $\mathbf{d}_p$ are sent to all sensor nodes. Together with the distances $\mathbf{r}$ to all beacons, which every sensor node must determine itself, the postcalculation requires $8m - 11$ flops, which leads with 100 beacons and the summation of x_1 and y_1 to 791 flops. It results that DLS saves 47.16% on the sensor nodes compared to the full calculation.

5.2 Communication Effort

Due to the fact that communication consumes most of the energy, data transfer between sensor nodes must be minimized. Particularly, sending data over long distances stresses the energy capacity of sensor nodes. Communication between base station and beacons is less critical and must be preferred if possible. Therefore, we classify communication in two phases. An uncritical phase, where all beacons send their positions to the base station. This causes no energy loss on the sensor nodes. Additionally, in a critical phase, where the base station sends precalculated information to the sensor nodes that have, in theory, to receive only. Due to errors in the transmission channel and protocols that require acknowledge packets etc., transmitting/sending is never lossless in practice. Furthermore, the base station cannot reach every sensor node in one hop, which demands multi-hopping over some nodes.

Here, we focus on a theoretical examination of the algorithm that is, for the moment, independent of protocol definitions and media access operations. Hence, every sensor node must receive the precalculated matrix A_p and vector $\mathbf{d}_p$ with $[n \cdot (m-1) + (m-1)]$ elements. This results in receiving $(3m - 3)$ elements, which are 1188 bytes with 100 beacons and floating point representation of every element.[1]

[1]On common microcontrollers, that are presently integrated

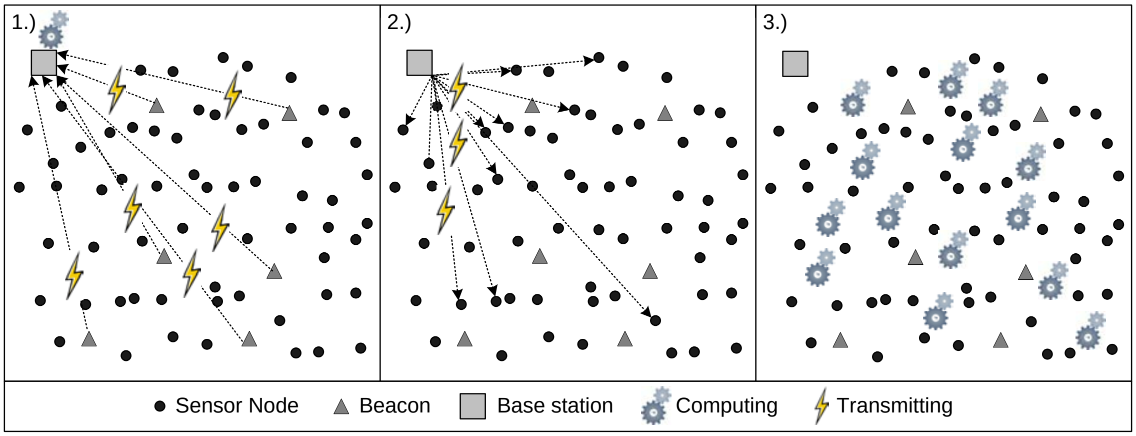

Figure 2: Procedure of the DLS-algorithm, which is divided into three phases.

5.3 Memory Requirement

The reduced calculations must be feasible on sensor nodes with a very small memory, mostly not more than a few kilobytes. In our case, the memory consuming operation is always $A_p \cdot \frac{1}{2} \cdot (r_1^2 - \mathbf{r}^2 + \mathbf{d}_p)$. In the worst case A_p and $\mathbf{r}$ plus $\mathbf{d}_p$ must be stored temporarily in memory before the execution on the sensor node can start. In more detail $[2 \cdot (m-1)] + (m-1) + (m-1) = (4m-4)$ elements must be stored.

6. SIMULATION

In the following, we show simulation results of a packet simulator. Here, DLS was compared to a direct competitor, the "Fully Distributed Multilateration" (FDM), which was among others used by Savvides et al. in [11] in a similar way. FDM is fully distributed, because every sensor node receives beacon positions directly from beacons and executes both the pre- and postcalculation completely independent.

The simulation was performed in J-Sim, a sensor network simulation framework by Tyan et al. [17], in which we added a more complex and therefore more realistic energy model. Our energy model considers the following parameter of energy consumption [2]:

- Power-mode dependent energy consumption with sleep and active mode
- Switching energy from sleep to active mode
- Distance dependent transmission of packets
- Computation complexity of the pre- and postcalculation
- Distance estimations with RSS measurements
- Position estimation via GPS on beacons

The specific energy parameters are based on the MICA2-mote, which is currently the most popular sensor node platform. Beacons have batteries with 21600 Joule, sensor nodes 6650 Joule and base stations are not limited due to access to an infrastructure. The network consists of one base station at position $P(x, y) = 1, 1$ and 315 randomly uniformly deployed nodes, including 15 beacons (with transmission range $60m$) and 300 sensor nodes (with transmission range $20m$) in a sensor field with the dimension $100m \times 100m$. We decided to use no special routing protocols, but restricted flooding to achieve a fair comparison between both algorithms. Restricted means that no more packets are forwarded by a node if all data required for computation have been received.

7. DISCUSSION

Fig. 3 shows the accumulated energy consumption of all nodes in the field. The first position packet for FDM was received at 66,82 seconds simulation time. After $16,20s$ every node received all beacon positions and started the computation process to estimate its position. In this communication phase 224 position packets were sent by the beacons and 52429 packets were received by all nodes ($95,42\%$ by sensor nodes), which is highly energy-intensive.

DLS started phase 1 at $83,32s$ by sending the first position packet. After that, phase 2 began and ended at $90,78s$. At this time the last sensor node received a packet with the precalculated matrix A_p and was therefore able to estimate its position. The whole process required the sending of 240 packets and the receiving of 2320 packets, where $12,93\%$ were received by the sensor nodes only. Summarized, DLS consumed $46,64J$

on sensor node platforms, every element is stored in floating point representation as a 4 byte number.

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

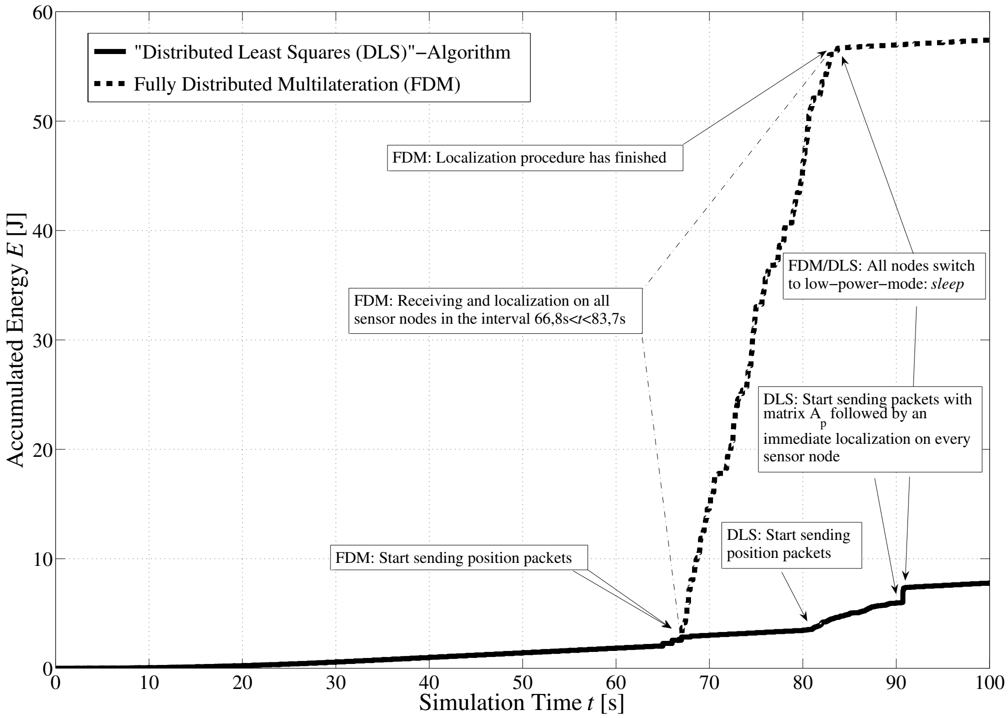

Figure 3: Accumulated energy usage of all nodes in the sensor network.

less energy than FDM, which is equal to 86, 48% savings. This is mainly due to three reasons. First, every sensor node must receive only one packet. After receiving this packet the node switched its transceiver to the very low energy consuming *sleep* state. Second, compared to FDM, a very simple calculation is executed on every sensor node only. And third, instead of computing the matrix A_p on all sensor nodes, which is performed once at the base station, high redundant computations are avoided. Both algorithms estimated the positions with a localization error under 10^{-14}%, reflecting the exact distances used.

8. CONCLUSION

We presented the "Distributed Least Squares"-algorithm (DLS), which allows exact position estimation with minimal energy consumption. This algorithm is based on the least squares method, which is, for many beacons, unfeasible on resource constrained sensor nodes. However, we decreased communication overhead and computation complexity while keeping its high precision. This can be achieved by splitting the linear least squares method into a complex part, precalculated on a high-performance base station, and a very simple post-calculation on every sensor node. Thus, we eliminated redundancy, because normally every sensor node has to process the precalculation. With this approach and based on a network containing 100 beacons we achieved 47.16% savings in computation on every sensor node in comparison to the "Fully Distributed Multilateration" (FDM), as a direct competitor. Moreover, DLS needs only a few kilobytes of memory on the sensor nodes. Finally, we showed in the packet-simulator J-Sim that DLS consumes less energy than FDM - more than 86% savings in total. One next step will be to implement DLS on a sensor network platform and test it in a real world environment.

Acknowledgment

This work was supported by the German Research Foundation under grant number TI254/15-1 and BI467/17-1 (keyword: Geosens).

9. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(3):393–422, 2002.

[2] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. Low-power wireless sensor networks. In *International Conference on VLSI Design*, 2001.

[3] R. Bill, C. Cap, M. Kohfahl, and T. Mund. Indoor and outdoor positioning in mobile environments a review and some investigations on wlan positioning. *Geographic Information Sciences*, 10(2):91–98, 2004.

[4] J. Gibson. *The mobile communications handbook*. CRC Press, 1996.

[5] M. Alouini. *Global Positioning System: An Overview*. California Institute of Technology, 1996.

[6] N. Bulusu. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, 2000.

[7] H. Tian, H. Chengdu, B. M. Brian, S. A. John, and A. Tarek. Range-free localization schemes for large scale sensor networks. In *9th annual international conference on Mobile computing and networking*, 2003.

[8] J. Blumenthal, F. Reichenbach, and Dirk Timmermann. Precise positioning with a low complexity algorithm in ad hoc wireless sensor networks. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 28(2):80–85, 2005.

[9] L. Doherty, K.S.J. Pister, and L. Ghaoui. Convex position estimation in wireless sensor networks. In *20th Annual Joint Conference of the IEEE Computer and Communications*, pages 1655–1663, 2001. Anchorage, AK, USA.

[10] S. Capkun, M. Hamdi, and J.-P. Hubaux. Gps-free positioning in mobile ad hoc networks. *Cluster Computing*, 5(2):157–167, 2002.

[11] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine grained localization in ad-hoc networks of sensors. In *7th ACM MobiCom (Rome, Italy,2001)*, pages 166–179, 2001.

[12] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha. Resilient localization for sensor networks in outdoor environments. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 643–652, 2005.

[13] A. A. Ahmed, H. Shi, and Y. Shang. Sharp: A new approach to relative localization in wireless sensor networks. In *Second International Workshop on Wireless Ad Hoc Networking (WWAN) ICDCSW'05)*, pages 892–898, 2005.

[14] T. C. Karalar, S. Yamashita, M. Sheets, and J. Rabaey. An integrated, low power localization system for sensor networks. In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)*, pages 24–30, 2004.

[15] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks (Elsevier), special issue on Wireless Sensor Networks*, 43(4):499–518, 2003.

[16] F. Reichenbach, A. Born, D. Timmermann, and R. Bill. A distributed linear least squares method for precise localization with low complexity in wireless sensor networks. In *2nd IEEE International Conference on Distributed Computing in Sensor Systems*, pages 514–528. Springer Verlag, Juni 2006. San Francisco, USA.

[17] J-sim: A component-based, compositional simulation environment, http://www.j-sim.org/, 2007.