

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Distributed Localization Algorithms for Wireless Sensor Networks: From Design Methodology to Experimental Validation

Stefano Tennina

*Department of Electrical and Information Engineering, Center of Excellence in Research DEWS, University of L'Aquila, 67040 L'Aquila  
Corresponding author  
Italy*

Marco Di Renzo

*Institute for Digital Communications (IDCOM), School of Engineering, College of Science and Engineering, The University of Edinburgh, Alexander Graham Bell Building, The King's Buildings, Mayfield Road, Edinburgh EH9 3JL, Scotland  
United Kingdom (UK)*

Fabio Graziosi and Fortunato Santucci

*Department of Electrical and Information Engineering, Center of Excellence in Research DEWS, University of L'Aquila, 67040 L'Aquila  
Italy*

## Abstract

Recent advances in the technology of wireless electronic devices have made possible to build ad-hoc Wireless Sensor Networks (WSNs) using inexpensive nodes, consisting of low-power processors, a modest amount of memory, and simple wireless transceivers. Over the last years, many novel applications have been envisaged for distributed WSNs in the area of monitoring, communication, and control. Sensing and controlling the environment by using many embedded devices forming a WSN often require the measured physical parameters to be associated with the position of the sensing device. As a consequence, one of the key enabling and indispensable services in WSNs is localization (i.e., positioning).

Moreover, the design of various components of the protocol stack (e.g., routing and Medium Access Control, MAC, algorithms) might take advantage of nodes' location, thus resulting in WSNs with improved performance. However, typical protocol design methodologies have shown significant limitations when applied to the field of embedded systems, like WSNs. As a matter of fact, the layered nature of typical design approaches limits their practical usefulness for the design of WSNs, where any *vertical* information (like, e.g., the actual node's position) should be efficiently shared in such resource constrained devices. Among the proposed solutions to address this problem, we believe that the Platform-Based Design (PBD) approach Sangiovanni-Vincentelli (2002), which is a relatively new methodology for the design of embedded systems, is a very promising paradigm for the efficient design of WSNs.

In particular, the PBD methodology allows to define a standard set of services and interface primitives (called Sensor Network Services Platform or SNSP) that can be made available to an application programmer independently from implementation issues on any (wireless) sensor network platform.

In the depicted context, the present contribution reports our recent research advances along two main directions. Firstly, we exploit the PBD methodology for the efficient design of ad-hoc WSNs with localization capabilities. In particular, the PBD paradigm is used to derive a fully distributed positioning algorithm, and a general protocol architecture for WSNs. Secondly, we validate the suitability of a practical implementation of the proposed solutions onto commercially available WSN platforms, and analyze their achievable performance in realistic propagation environments.

More specifically, the contributions of the present research work are as follows: 1) we will define a PBD-inspired Location Service (LS) along with its parameters and service primitives, which collects and provides network-wide information about the nodes' spatial position, 2) we will introduce a novel iterative positioning algorithm, which is called Enhanced Steepest Descent – ESD Tennina et al. (n.d.), and will show, by using computer-based simulations, that it can outperform other well-known distributed localization algorithms in terms of estimation accuracy and numerical complexity, 3) we will analyze the implementation issues related on mapping the ESD algorithm onto the CrossBow's MICAz sensor node platform, and investigate, via experimental activities, the effect of network topology and ranging errors on the performance of the proposed distributed localization algorithm, and 4) we will test the performance of the ESD algorithm during an extensive campaign of measurements conducted by using the Texas Instruments (TI)/Chipcon CC2431's hardware location-finder engine in a realistic and dynamic indoor propagation environment. We will show that the ESD algorithm can be efficiently used to improve the localization accuracy provided by the CC2431's location-finder engine. Moreover, as a byproduct of this latter experimental activity, we will show that the need of site-specific parameters for the correct operation of the CC2431's location-finder engine may severely reduce the localization accuracy of the system in dynamic environments, as well as propose and validate a simple solution to counteract this problem.

**Keywords:** Platform Based Design (PBD), positioning, ad-hoc wireless sensor networks.

## 1. Introduction

### 1.1 Overview

Wireless Sensor Networks (WSNs) are distributed networked embedded systems where each node combines sensing, computing, communication, and storage capabilities Goldsmith & Wicker (2002). Due to their unprecedented design challenges and potentially large revenues, in recent years WSNs have witnessed a tremendous upsurge in interest and activities in both academia and industry Dohler (2008). In particular, they have become increasingly popular in military and civilian sectors, and have been proposed for a wide range of application domains, e.g., control and automation, logistics and transportation, environmental monitoring, healthcare and surveillance.

In general, WSNs are required to possess self-organizing capabilities, so that little or no human intervention for network deployment and setup is required. A fundamental component

of self-organization is the ability of sensor nodes to “sense” their location in space, i.e., determining where a given node is physically located in a network Bachrach & Taylor (2005); Wang & Xiao (2007). In particular, node localization is a key enabling capability to support a rich set of geographically aware protocols for distributed and self-organizing WSNs Mauve & Widmer (2001), and for achieving context-awareness.

It is well-known Hofmann-Wellenhof et al. (1997), that the Global Positioning System (GPS) can greatly facilitate the task of location estimation by potentially allowing every GPS-equipped receiver to accurately localize itself in any point located on or above the Earth surface. However, GPS-based localization solutions are often considered a non-completely viable and well-suited solution for position estimation in WSNs, as sensor nodes are supposed to operate at low-complexity and low-power consumptions Bulusu et al. (2000). Moreover, GPS-based solutions have the undesirable side-effect that they cannot provide reliable location estimates in indoor environments, and in the presence of dense vegetation Perkins et al. (2006); Savvides et al. (2001). As a consequence of the above, much research has been done in the WSNs community to develop new techniques for localization in those environments where GPS-aided positioning is either unfeasible or does not meet the design requirements and paradigms of networked embedded systems, i.e., the so-called GPS-denied (or GPS-less) environments. The result of this intensive research work has been the proposal of many new solutions (alternative to GPS) to address the problem of distributed network location discovery (see, e.g. Santucci et al. (2006) and references therein). However, in Langendoen & Reijers (2003); Wang & Xiao (2007) the authors have clearly shown that among the existing algorithms none seems to perform better than the others, and claim that the definition of location algorithms with accurate positioning capabilities and low communication and computation costs for GPS-denied environments is still an ongoing area of research at both theoretical and experimental levels.

Furthermore, existing solutions for location estimation have often been obtained without considering the fundamental interactions of positioning algorithms with other entities of the protocol stack: in other words, current solutions do not adopt a methodological view of the whole protocol stack for system optimization. As a matter of fact, the traditional design approach is based upon the ISO-OSI layered model, i.e., the whole system is decomposed in a layered fashion, and the design of each layer follows the *isolation* principle. In general, lower layers are abstracted by means of a set of service primitives, while the higher layers in term of service requirements. This approach greatly simplifies the design task, but may lead to sub-optimal design solutions Kawadia & Kumar (2005). Moving from this consideration, novel design approaches are being developed by several researchers with the aim to design more efficient protocol solutions. Among the various alternatives, cross-layer design methodologies Srivastava & Motani (2005) are receiving a significant interest by the research community. In particular, the cross-layer approach advocates the benefits, in terms of costs and performance, of a joint design of the functionalities at different layers. In fact, it allows to reduce the duplication of functionalities, which may arise when designing each layer in isolation, and provides a joint optimization of system parameters. Nevertheless, cross-layer design is known to raise the design complexity, and to reduce the modularity and thus the re-use of system components Kawadia & Kumar (2005).

## 1.2 Aim and Motivation

In the light of the above overview, the main aim of the present manuscript is threefold: i) to propose the adoption of a novel methodology to design an efficient Location Service for

WSNs, thus overcoming the limitations of current design methodologies based on ISO–OSI and cross–layer paradigms, ii) to introduce a novel atomic localization algorithm with improved performance with respect to current solutions, and iii) to offer a solid proof of concept of the proposed methodologies and algorithms by means of computer simulations and experimental activities conducted with some WSNs testbeds.

### 1.2.1 The Need for A Novel Methodological Approach

In general, the WSN domain presents several challenging problems: it is characterized by hard real-time constraints, it has to be fault tolerant and design–error free, and it has to react to a nondeterministic adversary environment. Although existing cross–layer design paradigms seem to solve the limitations shown by the ISO–OSI approach, we emphasize a methodology that favours re–use at all levels of abstraction to keep the design complexity at a moderate level. The goal is to design a sensor node which is able to reconfigure itself and to form a network without any need for expensive infrastructure.

To meet the above design goals and requirements of WSNs, we adopt a recently proposed design methodology for embedded wireless systems, which is called Platform Based Design (PBD) Sangiovanni-Vincentelli (2002). The basic tenets of this methodology are: i) an orthogonalization of concerns, i.e., the separation of the various aspects of design to allow more effective exploration of alternative solutions, and ii) a meet–in–the–middle process, where successive refinements of specifications meet with abstractions of potential implementations. Basically, orthogonalization of concerns pushes to identify parts of the system which are independent enough (*orthogonal*) to be designed in separate steps. This is the same approach pursued in the traditional ISO/OSI model, where orthogonal functionalities of a network node have been identified and grouped in the well–known 7 layers (application, presentation, session, transport, network, data link and physical layer). Moreover, the meet-in-the-middle process advocates a richer abstraction of a layer, where services are exposed together with a model of cost/performance. The expression *meet–in–the–middle* thus comes from the fact that the design of a layer is neither subject to the higher layer requirements, as in a top–down approach, nor to the lower layer features, as in a bottom–up approach. Instead, service requirements are defined with a notion of the potential capabilities, performance and related costs of the lower layers (called *platforms* in the methodology). In other words, the meet–in–the–middle view of the design process defines an approach that maximizes re–usability and verifiability, while maintaining constraints on performance, cost and power consumption.

Furthermore, in recent years, the adoption of PBD has been proposed for the design of communication protocols Sgroi et al. (2000) and communication infrastructures Pinto (2008); Pinto et al. (2008), with particular emphasis on the challenges of wireless communications da Silva et al. (2000). In Bonivento et al. (2005), the methodology is applied to wireless networked control systems, with the definition of a flow based on three layers of abstractions, which takes into account both the design of the control algorithm and of the distributed architecture, as well as the definition of the control application to be mapped over the network nodes. In Balluchi et al. (2004), the platform–based design approach has been applied to the design of wireless sensor networks, with the definition of a *Network Platform* as a collection of services. Motivated by the above considerations, the first aim of this research work is to show how the PBD tenets can be applied for the design of an efficient distributed Location Service for WSNs.

### 1.2.2 The Need for Improved Localization Algorithms

Although several optimization algorithms for location estimation have been proposed in the literature to date, in Wang & Xiao (2007) the authors have recently shown that each of them exhibits advantages and disadvantages in terms of computational cost, overall accuracy, and suitability to be deployed onto today's available WSNs' devices. Accordingly, one aim of this contribution is to introduce a novel and more efficient (in terms of computational cost and accuracy) optimization algorithm suitable for distributed WSNs localization.

Among the various solutions so far proposed in the literature, many authors agree that a promising approach for distributed sensor node localization is the so-called "recursive positioning methods", see e.g. Santucci et al. (2006); Savvides et al. (2001); Wang & Xiao (2007). Loosely speaking, recursive algorithms are often employed to overcome the limits related to the short-range communication capabilities of sensor nodes, by enabling the position estimation process to be composed by many subsequent steps/phases through which all the sensors in the network localize themselves in a distributed fashion Santucci et al. (2006); Savarese (2002). These techniques have several positive features, e.g., i) they appear to be a good solution for sensor nodes with limited range capabilities, ii) they may efficiently counteract the sparse anchor node problem, and iii) they are distributed by nature. However, they still present several critical design issues, e.g., i) in Savvides et al. (2001) authors have shown that in recursive approaches the positioning error may accumulate along the iterative process, thus severely corrupting the final estimates of sensor nodes located in remote areas, i.e., regions of the network where "startup anchors" (i.e., nodes that are aware of their exact location) are sparse, and ii) in Dulman et al. (2008); Savarese (2002) authors have verified that some bad network topologies may introduce significant errors even with accurate distance estimates. In particular, to cope with error accumulation, accurate optimization algorithms have to be used for position estimation, but typically with high computational costs and long time, which may represent a serious limitation for handling e.g. nodes' mobility.

Motivated by these considerations, the second aim of the present contribution is twofold: i) to propose a comparative study of various optimization algorithms Nocedal & Wright (2006) that can be used for position estimation, and ii) to propose an enhanced version of the classical Steepest Descent algorithm, which we call Enhanced Steepest Descent (ESD), for improving the efficiency of position estimation.

### 1.2.3 The Need for Experimental Analysis and Validation

Although most analysis about the performance of WSNs are often conducted via computer-based (numerical) simulations, such a kind of analysis typically show significant limitations to assess the actual improvement and implementation issues of the proposed solutions when the algorithms need to be implemented onto today's available sensor nodes platforms, and when the WSN needs to be deployed in a realistic propagation environment. A couple of examples of these issues may be as follows: i) most analysis conducted via numerical simulations do not take into account the actual and limited capabilities of commercially available sensor nodes, which often results in the development of novel solutions that, even providing improved performance, are not implementable onto sensor nodes platforms due to their high computational complexity and memory requirements, and ii) numerical simulations typically rely on important assumptions to reproduce, e.g., ranging (i.e., the distance estimation between pairs of nodes) error models and the wireless propagation conditions, which may not represent in a consistent way the actual technique used for ranging computation, as well as the actual characteristics of the wireless propagation channel (e.g., the presence of obstacles,

non-line-of-sight propagation scenarios, and dynamic motion of objects or people around the area of interest), respectively. Actually, in the recent period the problem of understanding the real impact of the assumptions typically done for the analysis of ad-hoc networks via computer-based simulations is receiving a growing attention by the research community. In particular, recent papers, e.g., Newport et al. (2007), have claimed and verified via experiments that wrong or simplistic assumptions of how radios work may result in a completely different behavior and performance between simulation and experimentation. Accordingly, the authors suggest to either use real data as input to simulators or cross-validating simulated results with accurate experimental activities.

Motivated by the above considerations, the third aim of the present contribution is to validate the applicability and efficiency of the proposed PBD methodology by means of a WSN testbed deployed in a realistic propagation environment, as well as to analyze the performance improvement provided by the proposed ESD algorithm via experimental activities. In particular, we will describe two campaigns of measurements aiming at analyzing the achievable performance (i.e., localization accuracy and reliability) of two WSNs testbed platforms implemented using commercially available sensor nodes. The measurement campaigns are performed in two typical GPS-denied environments represented by static and dynamic indoor scenarios. The WSNs testbed platforms are currently available at the Center of Excellence in Research DEWS (University of L'Aquila, Italy) – [www.dews.ing.univaq.it/dews](http://www.dews.ing.univaq.it/dews), and the Networked Control Systems Laboratory (NCSlab) (the Italian node of the European Embedded Control Institute (EECI) at the University of L'Aquila) – [www.eeci-institute.eu](http://www.eeci-institute.eu), and are being extensively used for the analysis and design of WSNs for positioning applications.

### 1.3 Contribution

Motivated by the above considerations, the specific contributions of the present chapter are as follows: i) we will present a PBD-based Location Service for WSNs and define the set of primitives required for its implementation, ii) we will propose a novel distributed optimization algorithm for nodes' position estimation, which is an enhanced version of the classical Steepest Descent and is called ESD, iii) the proposed solution will be compared, via computer-based simulations, with other well-known optimization algorithms available in the open technical literature, and its improved performance in terms of error accuracy, computational complexity (i.e., time required to estimate the final position), algorithm initialization, and network topology will be investigated and discussed, iv) we will show that the ESD algorithm can be readily implemented onto the CrossBow's MICAz sensor node platform Cro (2008), and will substantiate and validate, via experimental activities, the results obtained via simulation when realistic ranging measurements are used at the input of the algorithm, and v) by means of off-line computer simulations performed on real captures acquired with the TI/Chipcon's CC2431<sup>1</sup> testbed Tex (2008) in a highly dynamic indoor environment, we will prove that the ESD algorithm can be effectively used to further refine the position estimated by the CC2431's location engine, thus yielding a non-negligible improvement in estimating the actual position of a sensor node with a modest increment in computational complexity.

---

<sup>1</sup> CC2431 sensor nodes developed by Texas Instruments (TI)/Chipcon are widely recognized as the first commercially available System-on-Chip (SoC) solution with a hardware RSS-based (Received Signal Strength) location-finder engine targeting ZigBee/IEEE 802.15.4 wireless sensor networking applications

### 1.4 Paper Organization

The remainder of this paper is organized as follows. In Section 2 the PBD-based Location Service is described; the functional decomposition into several platforms and their primitives are then provided. By focusing on the positioning algorithm, Section 3 will describe several optimization algorithms for WSNs position estimation, and will introduce the proposed ESD algorithm. In Section 4, simulation results will be presented and commented. In Section 5, the testbed platforms deployed using both Crossbow's MICAz and TI/Chipcon's CC2431 sensor nodes will be introduced, practical implementation issues of the ESD algorithm will be addressed, and experimental results will be discussed either in static and dynamic indoor environments. Finally, Section 6 will conclude the paper.

## 2. Location Service Design for WSNs: A PBD-based Approach

Moving from the basic tenets of the PBD approach described in SgROI et al. (2003), we consider a node architecture as depicted in Fig. 1, where i) an Application Interface (API) exposes the set of relevant services and hides lower networking details; and ii) the Sensor Network Service Platform (SNSP) is a middleware layer of services, which implements the exposed functionalities by resorting to the underlying protocol stack entities. Among the SNSP SgROI et al. (2003),

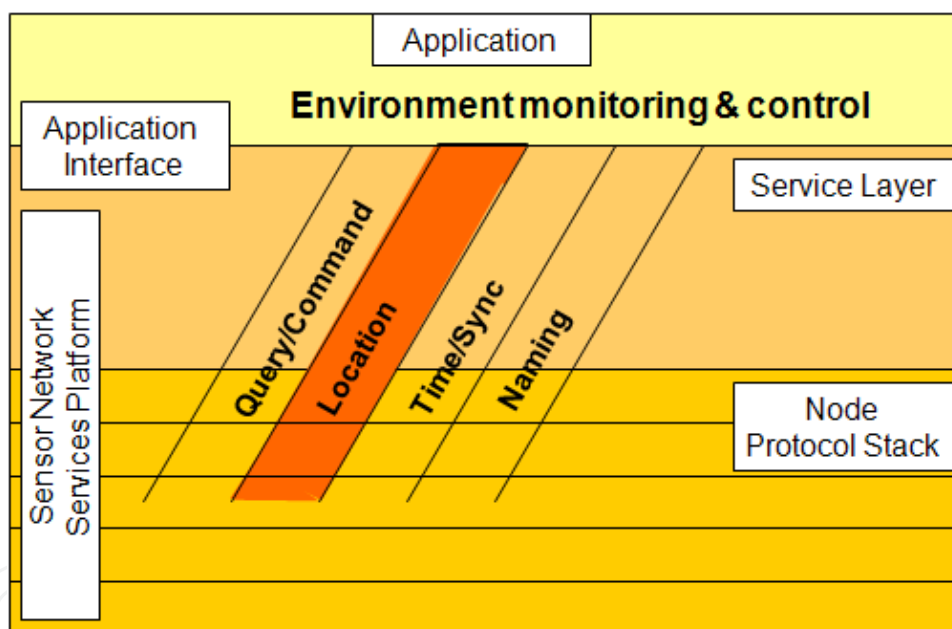


Fig. 1. Sensor Network Service Platform and Application Interface.

the Location Service (LS) collects and provides information about the spatial position of the nodes in the network. A *point location* is defined as a t-ple of values, which identify the position of the node within a reference system. Assuming, e.g., a common 3D cartesian reference system, a location (i.e., node's position) is a *struct* type collecting fields such as: i) the nodes' coordinates ( $x, y, z$ ); ii) a scale factor, which defines the resolution, and iii) the accuracy level yielding the reliability indicator of an estimated position.

Fig. 2 shows the functional decomposition of our designed Location Service into several PBD platforms, each one characterized by the relevant set of primitives (i.e., services) exposed towards the upper layer, hiding lower level details. In other words, in this framework, the level



of details increases when moving from the top to the bottom of the protocol stack. In what follows, we will briefly outline the set of primitives of each defined platform.

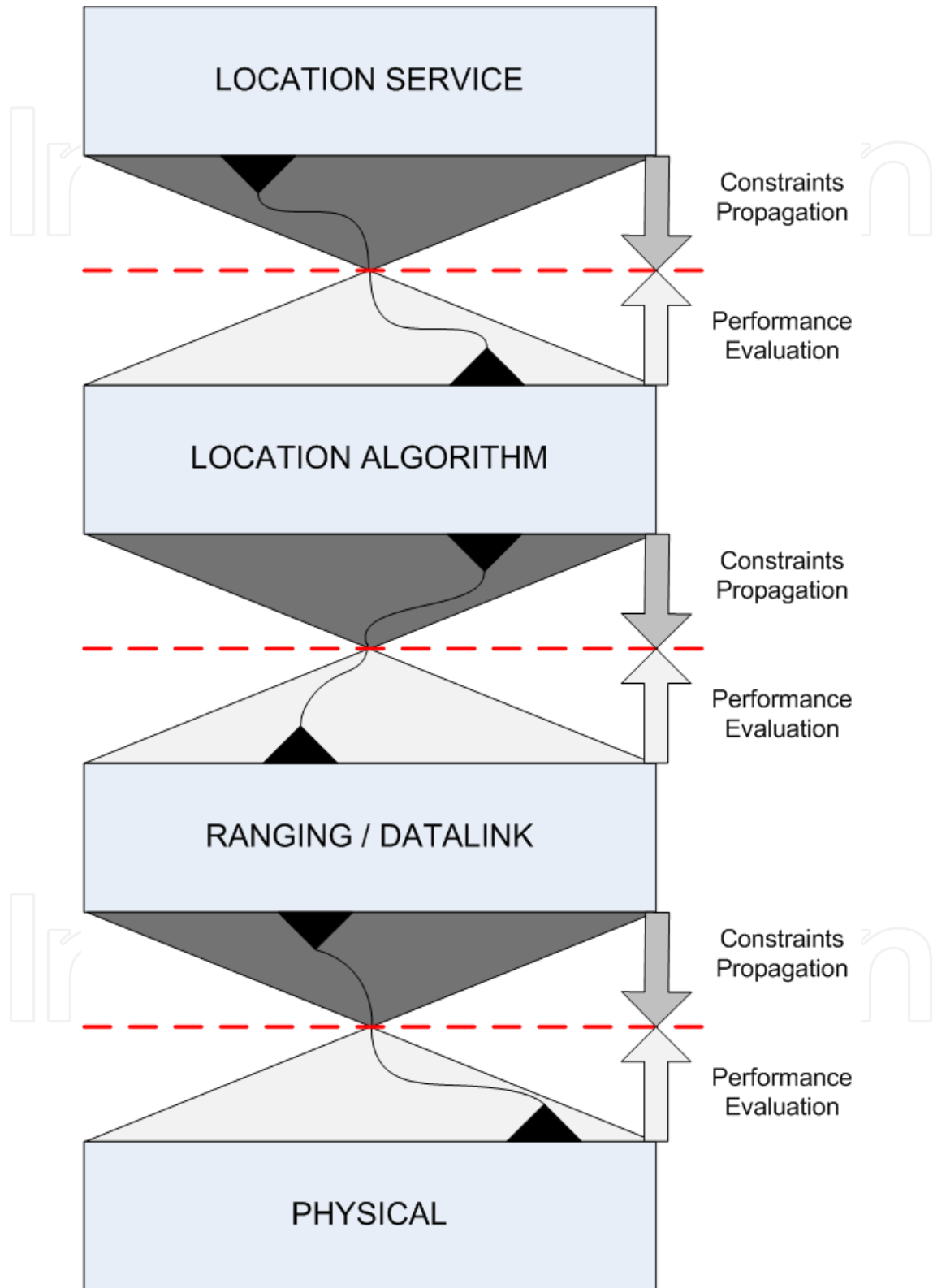


Fig. 2. Location Service Platform Stack.

## 2.1 Location Service Platform

According to the general setup introduced in Sgroi et al. (2003), the following set of LS primitives and related parameters are assumed at the application interface.

- *int* *LSSetup*(*struct resolution \*r*, *struct accuracy \*a*, *struct reference \*rs*, *int Time Tmax*) sets the resolution and the accuracy of location data, the reference system and the maximum time interval for obtaining the location data. A call to this primitive also starts the LS service.
- *int* *LSUpdate*(*struct resolution \*r*, *struct accuracy \*a*, *struct reference \*rs*, *int Time Tmax*). Similar to the *LSSetup()*, but at run-time.
- *struct location* *LSGetLocation*(*int NodeID*) returns the location of the node with *ID = NodeID*.

Accordingly, at the highest layer, we consider a set of primitives which simply consists in the attempt of each node to be aware of its position as soon as it starts operating into the network. Furthermore, this layer imposes a set of requirements which propagate deeply in the stack and which has to be met by the lower levels, i.e., by choosing the proper solutions. In this case, these requirements typically deal with i) the maximum allowable accuracy of the final position estimation of each node, ii) the maximum percentage of nodes allowed to remain unlocalized, and iii) the maximum time required to complete the position estimation algorithm.

## 2.2 Location Algorithm Platform

This is the core platform, where the mathematical model of the positioning algorithm is defined and performance is evaluated in order to meet the previous application's requirements. In this platform, we can consider the class of distributed and cooperative recursive positioning algorithms briefly described in Section 1.2.2. A set of primitives is listed and briefly described in what follows.

- *float distance* *LAGetRange*(*int NodeID*) operates a cooperative ranging<sup>2</sup> procedure between a node and the neighbor having *ID = NodeID*. *NodeID* denotes a node identifier, which is used by a node to identify its neighbors;
- *struct location* *LAInitialEstimation*() returns the initial position estimation according to a predefined criterion. Alternatives for initial estimation include the simple random guess, as well as a smarter, but more complicated, solution like in Savarese (2002);
- *struct location* *LAStep*(*struct location \*arrNeighsLoc*, *struct refinementParameters par*, *int Time T<sub>up</sub>*) proceeds one step ahead with the positioning algorithm once new information about positions of neighbors is collected. It returns the updated position estimation of the present node. When a stop criterion is reached<sup>3</sup>, node starts broadcasting its actual estimation;
- *void* *LABroadcast*(*struct location \*loc*) locally broadcasts the present position and accuracy of the estimate as well;
- *LACoordination*(*struct location \*loc*) is invoked when a node with insufficient connectivity cannot resolve an ambiguity in position estimation and requires cooperation of its neighbors.

---

<sup>2</sup> *Ranging* is the process of estimating the distance between a pair of nodes Tennina et al. (n.d.).

<sup>3</sup> A stop criterion may deal with the fact that the desired accuracy has been reached or that the timeout (*T<sub>up</sub>*) is expired

### 2.2.1 Recursive Positioning Method

As discussed in Section 1, we will consider a recursive positioning method for network location discovery. In particular, the well-known recursive and hierarchical method proposed by Santucci et al. (2006); Savvides et al. (2001) is analyzed for the sake of illustration. The following notation is used: i) a *blind* node is a node not aware of its position, ii) a *startup anchor* is a node aware of its position since the beginning of the location discovery procedure, and iii) a *converted anchor* is a blind node that has estimated its position the location discovery process. The basic version of the algorithm involves the following steps:

- Phase 0: At the beginning, “startup anchors” broadcast their position.
- Phase 1: Blind nodes that are connected (i.e., they are in the neighborhood) to at least four<sup>4</sup> “startup anchors” compute their position.
- Phase 2: Once a blind node has estimated its position, it becomes a “converted anchor” and broadcasts its estimated position to other nearby blind nodes, thus enabling them to estimate their positions.
- Phase 3: This process is repeated until the positions of all the nodes that eventually can have either four “startup anchors” or “converted anchors” are estimated.
- Phase 4: In this phase, an attempt is performed to solve eventual ambiguities for those nodes that do not have sufficient connectivity, by assuming cooperative decisions within the set of neighboring nodes (*LACoordination()*).

As a consequence, depending on the current step of the algorithm, the four anchor nodes with known positions may be either “startup anchors” or “converted anchors”. Of course, differently from “startup anchors”, the position of the “converted anchors” is affected by a certain error. In what follows, we will denote with “reference nodes” both “startup” and “converted” anchors.

### 2.2.2 Position Computation

The recursive positioning method described in Section 2.2.1 requires a technique to compute the location of a blind node from the position of four “reference nodes”, which may be in part “startup anchors” and in part “converted anchors”. In general, the computation of the position of a blind node involves two basic steps: i) measuring the distances between pairs of sensors using the *LAGetRange()* function, and ii) estimating the node’s position via the optimization of a given cost function obtained from the measured distances, using the *LAInitialEstimation()* function first and the *LAStep()* function after.

With regard to position computation from range estimates, in the literature two basic family of algorithms are often considered: i) *triangulation*, which foresees to estimate the position of the unknown node by finding the intersection of four spheres in a three-dimensional environment, and ii) *multilateration*, according to which the estimated position is obtained by reducing the difference between the actual measured distances and the estimated Euclidean distances between blind and reference nodes, i.e., via the minimization of an error cost function. According to Wang & Xiao (2007), the main difference between the two approaches is that multilateration algorithms are more robust to noisy range measurements. Both methods will be analyzed and compared in the present manuscript.

---

<sup>4</sup> In order to compute the position of a blind node we need, at least, ranging measurements from four anchor nodes in a three-dimensional space Tennina et al. (n.d.).

### 2.3 Ranging / DataLink Platform

With regard to distance estimation between pairs of nodes, several methods have been proposed in the literature (see, e.g., Patwari et al. (2005) for a survey). In our design stack, this role is managed by the Ranging/Datalink platform, which joins the design of the ranging capabilities of a node (i.e., which kind of algorithm is used to estimate the mutual distances) with the design of the communication protocol (i.e., which kind of medium access control is adopted and how the message passing is accomplished). A subset of primitives is listed and briefly described in the following.

- *float distance RDLGetRange(object source, float tol, float accuracy)* is the primitive by which a distance measurement (i.e., ranging) is obtained from the measurement of a physical parameter, which can be, e.g., the received signal strength (RSS) or the signal time of arrival (ToA). In addition, this primitive prepares the underlying physical entities for starting ranging operations, which typically require better accuracy if compared to the one adopted for communication purposes;
- *int RDLSetBOParameters(int BO, int  $\alpha$ , int  $\beta$ )* is the primitive which sets the maximum value of the actual BackOff counter, as well as the parameters  $\alpha$  and  $\beta$ , which represent the increasing and decreasing steps for the back off counter, respectively;
- *int RDLIncreaseBOValue(int BO, int  $\alpha$ )* is the primitive that increases the BackOff counter based on a parameter  $\alpha$  and some rules, as Binary Exponential Backoff or Multiplicative Increase Linear Decrease (MILD) Bhargahavan et al. (1994);
- *int RDLDecreaseBOValue(int BO, int  $\beta$ )*. Similar to the previous one, this primitive decreases the BackOff counter based on a quantity  $\beta$  and some rule;
- *int RDLSend(int NodeID, object Data, int Time  $T_{trans}$ )* is a primitive which allows the node to send Data to the neighbor having  $ID = NodeID$ , subject to a timeout ( $T_{trans}$ ) for the transaction;
- *int RDLReceive(object \*Data)* is a primitive alerting the node about the arrival of Data from a neighbor.

As a matter of fact, at this level we have designed a subset of basic communication primitives, i.e., those derived from the class of Carrier Sense Multiple Access (CSMA) with Collision Avoidance (CA) MAC protocols, adopted e.g. in *IEEE Std 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)* (2006), jointly with the definition of some primitives structured to support cooperative ranging in the contest of the mentioned localization application.

### 2.4 Physical Platform

This platform allows physical connectivity among nodes within the transmission range. For positioning purposes, we can define the following service primitives.

- *int PhySetup(int PTX, float  $T_{RES}$ , float  $\delta$ )* sets values of physical parameters, such as transmission power level, maximum time resolution and delay for a synchronization process.
- *int PhyUpdate(int PTX, float  $T_{RES}$ , float  $\delta$ )*. Similar to the *PhySetup()*, but at run-time.
- *object sourceVal PhyGetRange(object source, float  $T_{RES}$ )* is the primitive which supports *RDLGetRange()*. It gets the value of a physical parameter, so that *RDLGetRange()* can convert this value in a distance estimate based on the type of physical parameter and

a mathematical model of conversion. For example, if ranging measure is done by RSS, the source would likely be an acquisition via ADC (Analog to Digital Converter) of the incoming signal strength.

At this interface several other internal parameters can be considered, due to the complexity of the transceiver design. Most of these parameters are also handled by Resource Management Service “which allows an Application or a Service to get or set the state of the physical elements of the hardware” Sgroi et al. (2003). Due to likely tight constraints on spatial resolution for ranging measurements, wideband or Ultra Wide-Band are interesting opportunities for signal design at the physical layer. If compared to other technologies for ranging, e.g., ultrasound *Calamari Project* (n.d.), UWB may provide highest resolution because it relies on very short impulses and large bandwidth, and ranging can be somehow embedded in a synchronization process with tuneable settings<sup>5</sup>.

However, in the present contribution we consider the RSS measurements at the *PhyGetRange()* function (and in turn at the *RDLGetRange()* function), as it is nowadays a measure easily available on many commercial off-the-shelf sensor node platforms, such as the CrossBow’s MICAz and the TI/Chipcon’s CC2431 ones, which are used in our experimental activities and measurements. To be used in practice (see Section 5.1), RSS-based techniques need a calibration phase to estimate the path loss law, a relation between the received signal power and the actual distance between the nodes (by assuming the transmit power is known and fixed). These calibration issues will be analyzed in the present paper, as well as the impact of outdated measurements on the system performance.

### 3. ESD: A Novel Localization Algorithm for WSNs

#### 3.1 Notation

The aim of this section is to introduce a novel localization algorithm for WSNs. To do so, let us first introduce some basic notations useful for analytical formulation. By assuming an area with  $N_A$ ,  $\{A_i\}_{i=1}^{N_A}$ , “startup anchors” and  $N_U$ ,  $\{U_j\}_{j=1}^{N_U}$ , blind nodes, the following notation will be used throughout this chapter: i) bold symbols will be used to denote vectors and matrices, ii)  $(\cdot)^T$  will denote transpose operation, iii)  $\nabla(\cdot)$  will be the gradient, iv)  $\|\cdot\|$  will be the Euclidean distance and  $|\cdot|$  the absolute value, v)  $\angle(\cdot, \cdot)$  will be the phase angle between two vectors, vi)  $(\cdot)^{-1}$  will denote matrix inversion, vii)  $\hat{\mathbf{u}}_j = [\hat{u}_{j,x}, \hat{u}_{j,y}, \hat{u}_{j,z}]^T$  will denote the estimated position of the blind node  $\{U_j\}_{j=1}^{N_U}$ , viii)  $\mathbf{u}_j = [u_{j,x}, u_{j,y}, u_{j,z}]^T$  will be the trial solution of the optimization algorithm, ix)  $\bar{\mathbf{u}}_i = [x_i, y_i, z_i]^T$  will be the positions of the reference nodes  $\{A_i\}_{i=1}^{N_A}$ , and x)  $\hat{d}_{j,i}$  will denote the estimated (via ranging measurements) distance between reference node  $\{A_i\}_{i=1}^{N_A}$  and blind node  $\{U_j\}_{j=1}^{N_U}$ . Moreover, for analytical simplicity, but without loss of generality, we will present the optimization algorithms by assuming  $N_U = 1$  and  $N_A = 4$ .

<sup>5</sup> At the receiver, synchronization can be done by using a correlation mechanism between the received signal and a local signal (template) Stiffler (1968) or a delayed version of the received signal itself (differential receiver Alesii, Antonini, Di Renzo, Graziosi & Santucci (2004); Alesii, Di Renzo, Graziosi & Santucci (2004))

Before going into the details of the novel ESD algorithm, let us also summarize some basic localization methods with the aim to highlight the main advantages and superiority of the proposed solution.

### 3.2 Triangulation Method

In this method, the position of node  $U_1$  is obtained by inferring a geometric triangulation among estimated and actual distances. Accordingly, the unknown position is obtained by finding a solution that simultaneously solve the following set of equations:

$$\begin{cases} (x_1 - u_{1,x})^2 + (y_1 - u_{1,y})^2 + (z_1 - u_{1,z})^2 = \hat{d}_{1,1}^2 \\ (x_2 - u_{1,x})^2 + (y_2 - u_{1,y})^2 + (z_2 - u_{1,z})^2 = \hat{d}_{1,2}^2 \\ (x_3 - u_{1,x})^2 + (y_3 - u_{1,y})^2 + (z_3 - u_{1,z})^2 = \hat{d}_{1,3}^2 \\ (x_4 - u_{1,x})^2 + (y_4 - u_{1,y})^2 + (z_4 - u_{1,z})^2 = \hat{d}_{1,4}^2 \end{cases} \quad (1)$$

This system of equations can be solved using a Least Squares solution, which yields  $\hat{\mathbf{u}}_1 = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ , where matrix  $\mathbf{A}$  and vector  $\mathbf{b}$  can be found in Savarese (2002). In general, triangulation methods may fail to find a solution for the system in (1) when range and reference position estimates are noisy. Multilateration methods are, in general, preferred in this case. The triangulation method will be denoted as the *INV* method throughout the paper.

### 3.3 Multilateration Method

In this method, the position of node  $U_1$  is obtained by minimizing the error cost function  $F(\cdot)$  defined as follows:

$$F(\mathbf{u}_1) = \sum_{i=1}^{N_A} \left( \hat{d}_{1,i} - \|\mathbf{u}_1 - \bar{\mathbf{u}}_i\| \right)^2 \quad (2)$$

such that  $\hat{\mathbf{u}}_1 = \arg \min_{\mathbf{u}_1} \{F(\mathbf{u}_1)\}$ . The minimization of (2) can be done using a variety of numerical optimization techniques, each one having its own advantages and disadvantages in terms of accuracy, robustness, convergence speed, complexity, and storage requirements Nocedal & Wright (2006). Note that as optimization methods are iterative by nature, we will denote with index  $k$  the  $k$ -th iteration of the algorithm and with  $F(\mathbf{u}_1(k))$  and  $\mathbf{u}_1(k)$  the error cost function and the estimated position at the  $k$ -th iteration, respectively. The final estimated position will be denoted by  $\hat{\mathbf{u}}_1 = \mathbf{u}_1(\bar{k})$ , where  $\bar{k}$  is such that:

$$F(\mathbf{u}_1(\bar{k})) < \Phi \quad \text{or} \quad \bar{k} = \text{MAX}_{\text{iter}} \quad (3)$$

with  $\Phi$  being the desired accuracy computed on the error function in (2) and  $\text{MAX}_{\text{iter}}$  being the maximum number of iterations allowed for the algorithm.

Basically, Equation (3) represents the stop criterion mentioned in Section 2.2; then both design parameters  $\Phi$  and  $\text{MAX}_{\text{iter}}$  are application-dependent.

#### 3.3.1 Classical Steepest Descent (SD)

The classical Steepest Descent (SD) is an iterative line search method which allows to find the (local) minimum of the cost function in (2) at step  $k + 1$  as follows (Nocedal & Wright, 2006, pp. 22, sec. 2.2):

$$\mathbf{u}_1(k+1) = \mathbf{u}_1(k) + \alpha_k \mathbf{p}(k) \quad (4)$$

where  $\alpha_k$  is a step length factor, which can be chosen as described in (Nocedal & Wright, 2006, pp. 36, ch. 3) and  $\mathbf{p}(k) = -\nabla F(\mathbf{u}_1(k))$  is the search direction of the algorithm.

In particular, when the optimization problem is linear, in the literature there exist some expressions to compute the optimal step length to improve the convergence speed of the algorithm. On the other hand, when the optimization problem is non-linear, as considered in this contribution, a fixed and small step value is in general preferred, in order to reduce the oscillatory effect when the algorithm approaches the solution. In such a case, we have  $\alpha_k = 0.5\mu$ , where  $\mu$  is the learning speed Santucci et al. (2006).

### 3.3.2 Enhanced Steepest Descent (ESD)

The SD method provides, in general, a good accuracy in estimating the final solution. However, it may require a large number of iterations, which may result in a too slow convergence speed, especially for mobile ad-hoc wireless networks. In order to improve such convergence speed, we propose in this contribution an enhanced version of it, which we call Enhanced Steepest Descent (ESD).

The basic idea behind the ESD algorithm is to continuously adjust the step length value  $\alpha_k$  as a function of the current and previous search directions  $\mathbf{p}(k)$  and  $\mathbf{p}(k-1)$ , respectively. In particular,  $\alpha_k$  is adjusted as follows:

$$\begin{cases} \alpha_k = \alpha_{k-1} + \gamma & \text{if } \theta_k < \theta_{\min} \\ \alpha_k = \alpha_{k-1}/\delta & \text{if } \theta_k > \theta_{\max} \\ \alpha_k = \alpha_{k-1} & \text{otherwise} \end{cases} \quad (5)$$

where  $\theta_k = \angle(\mathbf{p}(k), \mathbf{p}(k-1))$ ,  $0 < \gamma < 1$  is a linear increment factor,  $\delta > 1$  is a multiplicative decrement factor, and  $\theta_{\min}$  and  $\theta_{\max}$  are two angular threshold values that control the step length update.

By using the four degrees of freedom  $\gamma$ ,  $\delta$ ,  $\theta_{\min}$  and  $\theta_{\max}$ , we can simultaneously control the convergence rate of the algorithm and the oscillatory phenomenon when approaching the final solution in a simple way, and without appreciably increasing the complexity of the algorithm when compared to the classical SD method. Basically, the main advantage of the ESD algorithm is the adaptive optimization of the step length factor  $\alpha_k$  at run time, which allows to dynamically either accelerate or decelerate the convergence speed of the algorithm as a function of the actual value of the function to be optimized. In the next sections we will show the performance improvement introduced by this algorithm.

## 4. Proof-of-Concept via Computer-based Simulations

In the frame of PBD approach, performance evaluation is a fundamental concern in the mapping process between *functional description* and *implementation* and it is intended to verify that a solution actually belongs to the design space defined by the *platform*, so that higher layer functional requirements can be met Sgroi et al. (2000). Due to the complexity of network scenario and the need of modeling various components, we have developed a flexible node model. We can test algorithms with a full view of the network while abstracting lower protocol layer (e.g. datalink) details. Furthermore, with the same framework, we can test specific node's behavior by restricting the attention to a reduced number of nodes.

#### 4.1 Atomic Localization

In this section, we will describe some MATLAB simulation results with the aim to assess the performance of the proposed ESD algorithm in several operating conditions and compare its performance with other localization algorithms.

##### 4.1.1 System Setup

The scenario depicted in Fig. 3, is used to have a common reference environment to analyze the improvement provided by the proposed ESD algorithm, and compare several optimization algorithms. For this setup, we assume that the anchor nodes are all “startup anchors”, which allows to investigate the so-called atomic location discovery problem, i.e., only Phase 1 described in Section 2.2.1 is implicitly considered in this system setup.

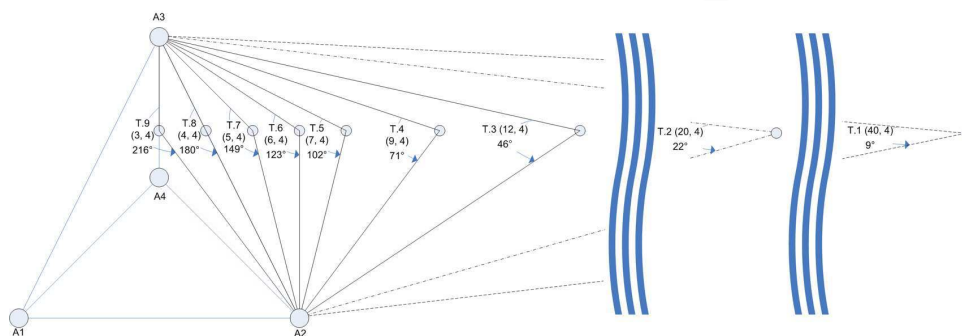


Fig. 3. Reference scenario and network topology (atomic localization step/phase).

In Fig. 3, we have three “startup” anchor nodes  $A_1$ ,  $A_2$ ,  $A_3$ , a non-complanar “startup” anchor node  $A_4$ , and a blind node  $U_1$ , which may be located in one of the positions  $T_h$ , with  $h = 1, 2, \dots, 9$ . In order to analyze the impact of the network geometry/topology on the performance of the optimization algorithms, we have introduced a parameter similar to the so-called geometric dilution of precision factor Savvides et al. (2001). In particular, in every  $T_h$  position the unknown node sees the reference nodes with an increasing angle when moving from  $T_1$  to  $T_9$ : this corresponds to moving from a scenario ( $T_1$ ) with a bad geometry where ambiguities may arise during position estimation, towards a scenario ( $T_9$ ) where the unknown node is surrounded by reference nodes, thus giving an ideally optimal network topology for position estimation, regardless of the specific algorithm Wang & Xiao (2007).

The main parameters used to obtain simulation results are as follows: i)  $\bar{\mathbf{u}}_1 = [0, 0, 0]^T$  m,  $\bar{\mathbf{u}}_2 = [6, 0, 0]^T$  m,  $\bar{\mathbf{u}}_3 = [3, 6, 0]^T$  m, and  $\bar{\mathbf{u}}_4 = [3, 3, 1]^T$  m; ii) the blind node may occupy 9 positions, e.g.,  $\mathbf{u}_1 = [40, 4, 0]^T$  m in  $T_1$  ( $9^\circ$ ) and  $\mathbf{u}_1 = [3, 4, 0]^T$  m in  $T_9$  ( $216^\circ$ ); iii) the ranging error will be modeled as a Gaussian random variable with mean value given by the actual distance between reference and blind nodes and a fixed standard deviation denoted by  $\sigma_R$ , which is supposed to be independent from the actual distance; iv) the position error statistics are obtained by averaging over 2500 realizations of the ranging error for every position of the blind node; v) in order to analyze the effect of both the initial guess and the network topology on the optimization algorithm, 36 starting points uniformly distributed on a circle on the plane  $z = 0$  centered at  $[0, 0, 0]^T$  and with radius 50m are considered; vi) the maximum number of iterations for each algorithm is  $\text{MAX}_{\text{iter}} = 5000$ ; vii) the tolerance on the minimum of the error function is  $\Phi = 0.05$ ; viii) the initial learning speed for SD and ESD is  $\mu = 0.1$ ; and ix) the degrees of freedom for the ESD algorithm are:  $\gamma = 0.1$ ,  $\delta = 1.75$ ,  $\theta_{\min} = 5^\circ$  and  $\theta_{\max} = 30^\circ$ .



Algorithm	Comp. Time (s)	Mean Error (m)	Std. Error (m)
CG <sub>1</sub>	0.0253 (T <sub>1</sub> )	7.47 (T <sub>1</sub> )	6.28 (T <sub>1</sub> )
	0.0090 (T <sub>5</sub> )	1.93 (T <sub>5</sub> )	1.17 (T <sub>5</sub> )
	0.0060 (T <sub>9</sub> )	1.21 (T <sub>9</sub> )	0.56 (T <sub>9</sub> )
CG <sub>2</sub>	0.0255 (T <sub>1</sub> )	7.44 (T <sub>1</sub> )	6.23 (T <sub>1</sub> )
	0.0090 (T <sub>5</sub> )	1.93 (T <sub>5</sub> )	1.18 (T <sub>5</sub> )
	0.0058 (T <sub>9</sub> )	1.21 (T <sub>9</sub> )	0.56 (T <sub>9</sub> )
SD	0.2206 (T <sub>1</sub> )	6.65 (T <sub>1</sub> )	4.14 (T <sub>1</sub> )
	0.0264 (T <sub>5</sub> )	1.93 (T <sub>5</sub> )	1.07 (T <sub>5</sub> )
	0.0115 (T <sub>9</sub> )	1.26 (T <sub>9</sub> )	0.61 (T <sub>9</sub> )
ESD	0.0793 (T <sub>1</sub> )	6.79 (T <sub>1</sub> )	4.12 (T <sub>1</sub> )
	0.0096 (T <sub>5</sub> )	1.93 (T <sub>5</sub> )	1.06 (T <sub>5</sub> )
	0.0058 (T <sub>9</sub> )	1.23 (T <sub>9</sub> )	0.59 (T <sub>9</sub> )
NLS	0.2615 (T <sub>1</sub> )	6.72 (T <sub>1</sub> )	4.12 (T <sub>1</sub> )
	0.0363 (T <sub>5</sub> )	1.92 (T <sub>5</sub> )	1.03 (T <sub>5</sub> )
	0.0202 (T <sub>9</sub> )	1.23 (T <sub>9</sub> )	0.58 (T <sub>9</sub> )
INV	0.0001 (T <sub>1</sub> )	15.67 (T <sub>1</sub> )	9.96 (T <sub>1</sub> )
	0.0001 (T <sub>5</sub> )	3.50 (T <sub>5</sub> )	2.19 (T <sub>5</sub> )
	0.0001 (T <sub>9</sub> )	2.26 (T <sub>9</sub> )	1.36 (T <sub>9</sub> )

Table 1. Comparison of optimization algorithms (CG<sub>1</sub> and CG<sub>2</sub> are the Fletcher–Reeves Polak–Ribière and Hestenes–Stiefel algorithms with secant method Tennina et al. (n.d.).

#### 4.1.2 Numerical Results

In Table 1 we have reported a performance comparison of the optimization algorithms described in Section 3 in terms of computational time, mean and standard deviation of the positioning error. We observe that: i) the positioning error increases when moving the blind node from  $T_1$  to  $T_9$  due to network topology, as expected, ii) the triangulation algorithm (*INV*) provides the worst performance in terms of error accuracy, iii) the ESD algorithm provides the same accuracy as the SD and NLS<sup>6</sup> algorithms, but reaches the final solution faster (this is an important result for, e.g., mobile networks), iv) the ESD performs as well as the CG<sup>7</sup> algorithms in most scenarios, but outperforms them in those network topologies that are prone to ambiguities (e.g., when the blind node is located in  $T_1$ – $T_4$  positions).

Fig. 4 shows the performance of all simulated algorithms with respect to the Cramer–Rao Lower Bound (CRLB) as defined in Dulman et al. (2008). The results are related to a blind node located in position  $T_4$  in Fig. 3, and the horizontal axis shows the starting position used to initialize every algorithm (i.e, initial guess point), which is an important parameter to be

<sup>6</sup> Non Linear Least Square Tennina et al. (n.d.). This is a sophisticated but quite complex solution, because matrix factorization and Hessian computation are required.

<sup>7</sup> Non–Linear Conjugate Gradient Tennina et al. (n.d.). These methods have been used extensively to solve non–linear optimization problems as they do not require matrix storage and need, in general, a smaller number of iterations than SD method.

investigated to analyze the robustness of every optimization algorithm. The results show that: i) the *INV* algorithm provides, on the average, the worst performance, which is also independent from the actual initialization point of the algorithm, ii) CG algorithms are very sensitive to the initial guess point, and in some scenarios the algorithm may fail to converge to the true position of the blind node (our experimental trials show that CG algorithms fail to converge when the initial guess is mirrored by  $180^\circ$  with respect to the true node's position), and iii) SD, ESD and NLS algorithms seem to perform globally better than the other ones, and have similar performance. Moreover, these latter algorithms provide results very close to the CRLB.

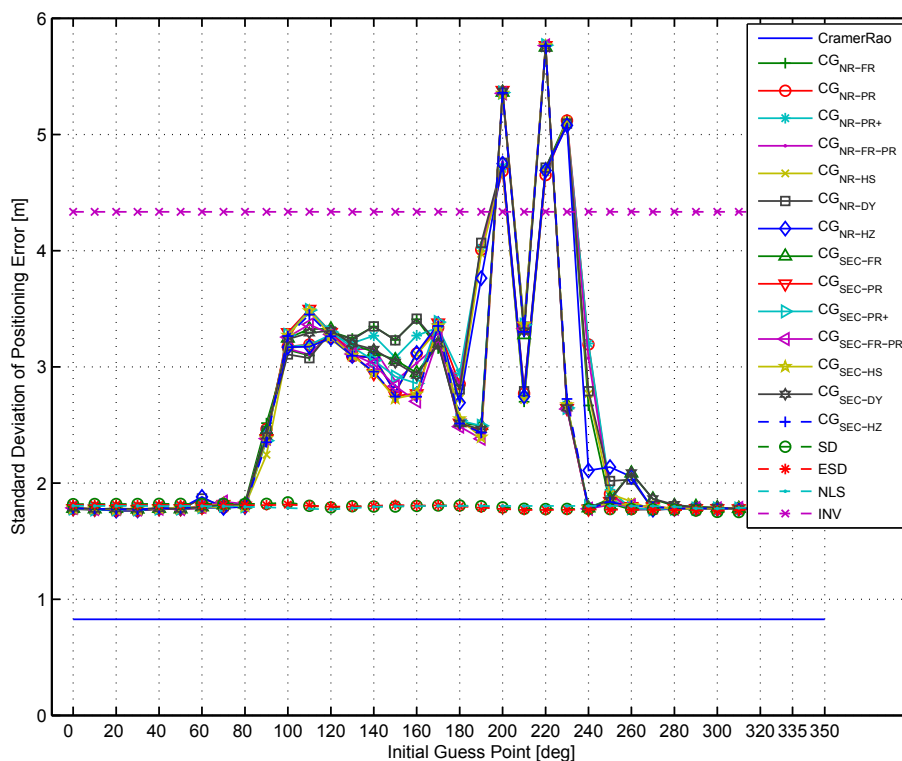


Fig. 4. Performance of the optimization algorithms with respect to the CRLB, and as a function of the initial guess point. The blind node is in position  $T_4$  of Fig. 3.

## 4.2 Network-wide Localization

In this section we extend the results obtained at the atomic level to a network composed by several blind nodes to evaluate the performance of our proposed ESD algorithm, i.e. considering all the phases described in Section 2.2.1.

### 4.2.1 System Setup and Numerical Results

Accordingly, moving from the architectural view of the nodes already presented in SgROI et al. (2003), we developed a node model as shown in Fig. 5, where at the application interface a set of services for implementing e.g. several kinds of control algorithms over WSNs are exposed. By focusing on the Network Platform, i.e. the blocks under such application interface, the introduction of a vertical module should be noted. The vertical nature of this data structure

is specifically intended to let all layers may have access to the information stored within (e.g. distance, position estimation and residual energy of batteries for each neighbor). This structure is intended to be shared also in the simulation code, since various layers use a pointer for access. Performance evaluation at network level has been carried out by resorting to the Discrete Event Simulator OMNeT++ Varga (n.d.), in which the node model shown in Fig. 5 has been implemented.

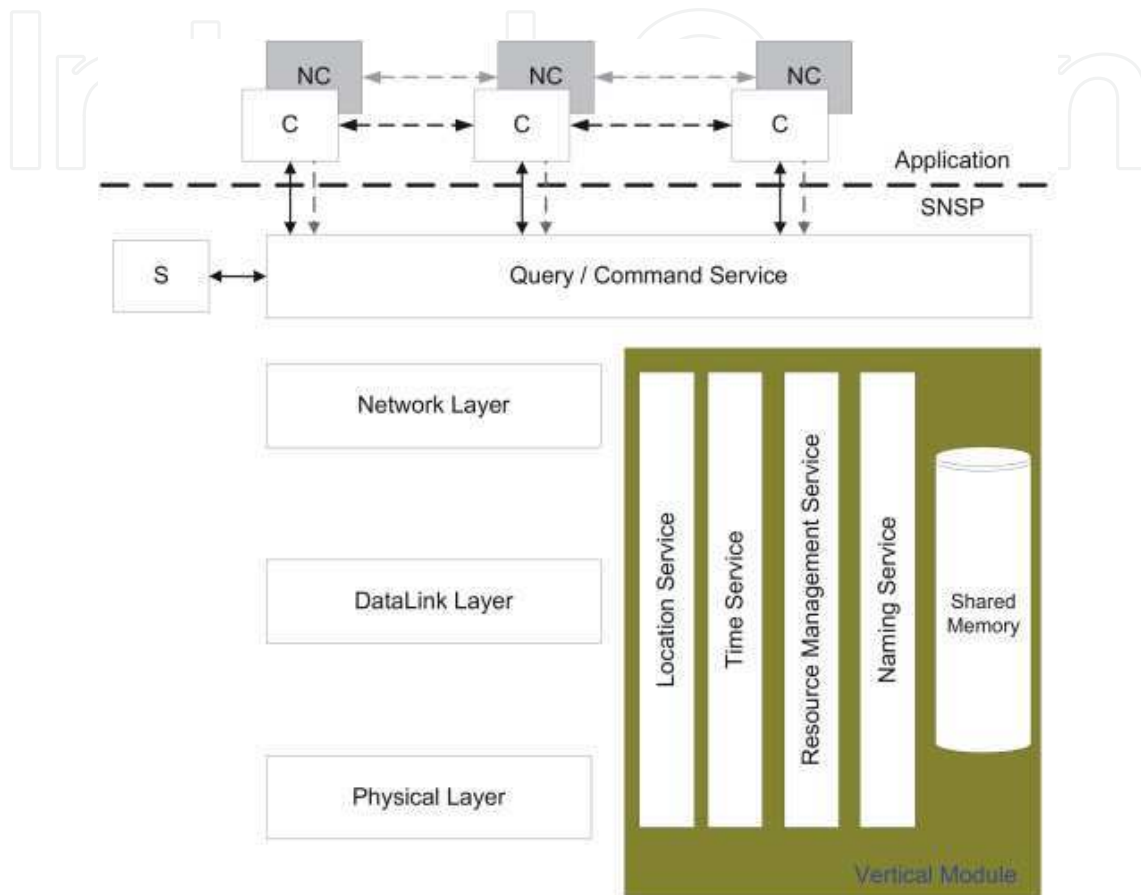


Fig. 5. Reference node architecture Santucci et al. (2006).

As an example, numerical results have been obtained in a network scenario with 100 nodes randomly (uniform distribution) deployed over a squared area with side length equals to  $30m$ . Five anchors are randomly placed along the perimeter of the network area and have a transmission range equal to  $9m$ , as large as those exhibited by normal sensor nodes. Moreover, the error on each distance measurement is modelled as a truncated (between  $-3\sigma$  and  $3\sigma$ ) zero-mean Gaussian random variable, with standard deviation  $\sigma = 0.15m$ . Nodes implement also the CSMA-CA algorithm whose primitives have been briefly depicted in Section 2.3. While previous results showed that the proposed algorithm outperforms in many cases the solutions existent, in Fig. 6 we show that it allows effectively nodes to obtain good final position estimation. As a matter of fact, 83% of nodes has a final position estimation error less than transmission range and 99% of nodes estimate their position with an error less than twice of transmission range. Note that the density of nodes in this simulated scenario compensates for the low number of anchors in the network.

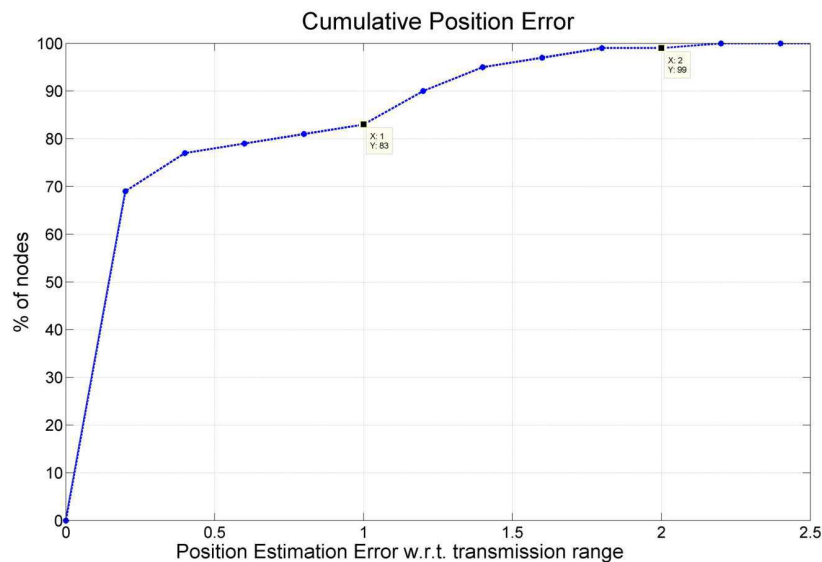


Fig. 6. Cumulative distribution of position error (x-axis scale is normalized to the nodes' radio range). 83% of nodes have a position error equal or less than transmission range, while 99% have a position error equal or less than twice of transmission range.

## 5. Proof-of-Concept via Experimental Testbeds

In order to assess both implementation issues and performance of the proposed ESD algorithm via experiments besides computer simulations, we have implemented a testbed platform by using both CrossBow's MICAz (see Cro (2008)) and Texas Instruments/Chipcon CC2431 (see Tex (2007)) sensor nodes.

### 5.1 Ranging Model

Both sensor nodes platforms use a RSS-based ranging method, and requires a (known) RSS-to-distance calibration curve to estimate the distance between pairs of nodes from a RSS measurement Cro (2008), as follows:

$$d = 10^{\left[\frac{RSS-A}{10n}\right]} \quad (6)$$

where  $d$  denotes the transmitter-to-receiver distance,  $n$  is the propagation path-loss exponent,  $A$  represents the RSS value measured by a receiver that is located 1m away from the transmitter (i.e., reference distance), and RSS is the actual measured value.

In order to estimate this calibration curve, we use the standard procedure described in Aamodt (2008), which consists in deploying a grid of nodes in the area of interest and extracting the desired parameters by post-processing the gathered data. Accordingly, a  $6m \times 10m$  grid of sensor nodes has been deployed in the NCSlab, as shown in Fig. 7. The sensors located in the ground floor are receiver nodes, while transmitter nodes are deployed at the edge of the measurement area, thus yielding a minimum and maximum transmitter-to-receiver distance of 0.5m and 11.7m, respectively. Moreover, the transmitters can be located at different heights with respect to the ground floor (ranging from 5cm to 1.2m). To estimate the calibration curve, the transmitters broadcast packets in a time-scheduled fashion such that collisions are avoided, and the receivers collect RSS values for each received packet, and then send a report to the host PC.



Fig. 7. Deployed testbed using CrossBow's MICAz sensor nodes for ranging calibration.

The RSS-to-distance reference curve in Equation (6) is obtained via a least-squares best linear fitting from several collected RSS values (every receiver node measures RSS values during a 5 minutes acquisition window, resulting in approximately 2000 RSS values). The obtained result is shown in Fig. 8 along with real measurements. Note that, in Fig. 8: i) the RSS values are represented as absolute values in arbitrary units, as provided by the receiver nodes, ii) the distance  $d$  in the horizontal axis is normalized to the reference distance of  $d_0 = 1\text{m}$ , and iii) the computed fitting parameters are  $A = 59.66$  and  $n = 1.84$ . Note that a path-loss exponent smaller than free space propagation is obtained (i.e.,  $n < 2$ ), which is probably due to the fact that the receiver nodes are located very close to ground floor, which provides a strong constructive reflected propagation path in addition to the direct one.

## 5.2 System Setup MICAz

In order to analyze implementation issues of the ESD algorithm, and validate simulative results of atomic localization with experimental activities, we have deployed CrossBow's MICAz sensor nodes with a similar setup as the one shown in Fig. 3. The testbed has been deployed in an empty conference room of our NCSlab.

The main parameters used in this testbed setup are as follows: i) the reference nodes' positions are  $\bar{\mathbf{u}}_1 = [2, 1, 0]^T \text{m}$ ,  $\bar{\mathbf{u}}_2 = [2, 3, 0]^T \text{m}$ ,  $\bar{\mathbf{u}}_3 = [4, 2, 0]^T \text{m}$ , and  $\bar{\mathbf{u}}_4 = [3, 2, 0.5]^T \text{m}$ ; ii) similar to Fig. 3, the blind node may occupy 16 positions, e.g.,  $\mathbf{u}_1 = [3, 10, 0]^T \text{m}$  in  $T_1$  and  $\mathbf{u}_1 = [3, 2.5, 0]^T \text{m}$  in  $T_{16}$ ; iii) the statistics (e.g., mean value) of the positioning error are obtained by averaging over 40 independent runs (i.e., acquisitions) of the algorithm for each blind node; and iv) the maximum number of iterations for the ESD algorithm is 250. Finally, the ranging error is obtained from RSS measurements as described in Section 5.1. In order to compare experiments and simulations in a fair way, computer-based analysis having at the input the

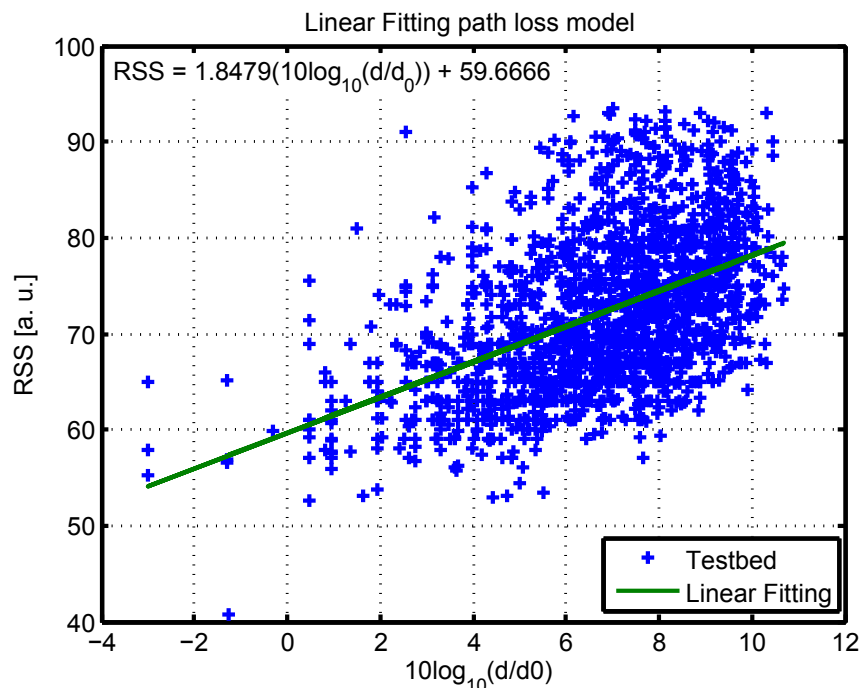


Fig. 8. RSS-to-distance ranging model.

ranging model derived in Section 5.1, and considering real RSS captures from each blind node have been simulated as well.

### 5.3 Results MICAz

In Fig. 9 we have reported the mean value of the positioning error with respect to the angle under which the unknown node sees the reference nodes (i.e., this curve is obtained by averaging over the 40 acquisitions), along with its standard deviation. Superimposed to the experimental results, we have also reported those obtained via computer-based simulations using the same experimental ranging model obtained in Section 5.1, and having at the input the real experimental captures taken with the testbed. The perfect overlap between the two curves substantiates the correct implementation of the ESD algorithm on the CrossBow's MICAz testbed platform using the NesC programming language Gay et al. (2003). This is an important result to use the testbed for further analysis aiming at quantifying, via experimental activities, other important performance indexes, such as power consumptions and complexity, as well as at judging the overall performance of the ESD algorithm.

### 5.4 System Setup CC2431

In order to try to overcome the issues related to the off-line RSS-to-distance ranging model calibration, we have deployed a second testbed in the NCSlab using TI/Chipcon's CC2431 sensor nodes. The goal of this study is to analyze the impact of an erroneous or outdated estimate of the propagation-dependent parameters, propose novel solutions to counteract this problem, and understand if the proposed ESD algorithm can be efficiently used to further refine the position estimation provided by the location-finder engine, available on TI/Chipcon's CC2431 sensor nodes, in a scenario with dynamic changes of the propagation conditions. To do so, and have a sound understanding of the performance of the ESD algorithm in a more

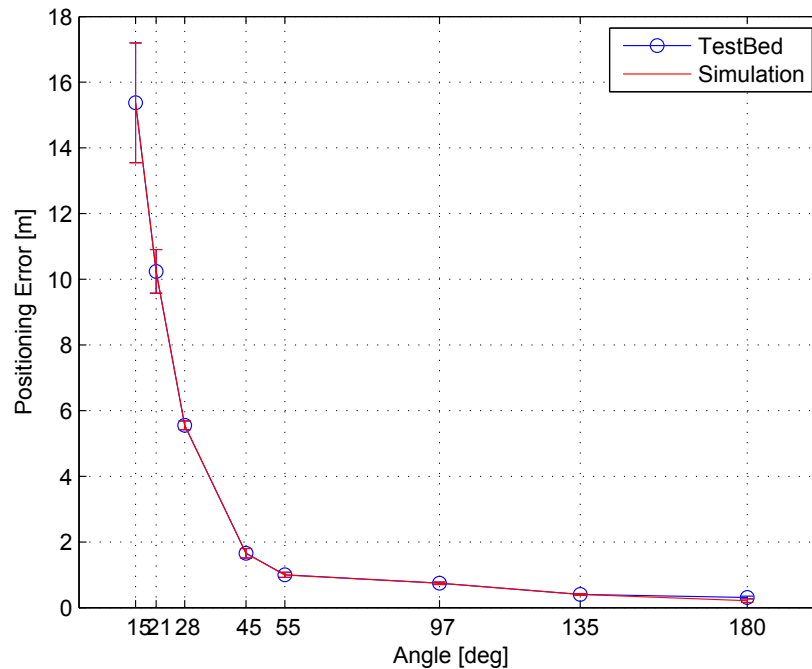


Fig. 9. Mean value and standard deviation of the positioning error: comparison between simulation and experimentation.

realistic scenario than the one analyzed in Section 5.2, we have conducted a campaign of measurements during the opening ceremony day of the NCSlab on March 27, 2008. The event was characterized by a half-day kick-off conference during which the past, present, and future activities of the laboratory were presented. The kick-off conference was attended by several people, and yielded a good occasion to test the performance of the deployed WSN, and, in particular, to test the achievable performance of the TI/Chipcon's CC2431 location engine in a realistic GPS-denied environment, where the propagation characteristics of the radio channel changed appreciably during the event due to the people's movement inside the room (i.e., dynamic indoor environment). The duration of the event was approximately three hours and forty minutes, thus providing enough statistical data to well support our findings and conclusions. The data collected during this measurement campaign have been used as an input to the ESD algorithm and its performance has been quantified via off-line computer-based simulations, while ongoing research activities concern with an efficient implementation of our ESD refinement algorithm onto the TI/Chipcon's CC2431 sensor node platform.

#### 5.4.1 NCSlab Opening Ceremony

The opening ceremony of the NCSlab was characterized by four main phases, which well describe the dynamic nature of the event and, as a consequence, the dynamic nature of the propagation environment to be analyzed. In what follows there is a brief description of each phase:

1. The first phase, which took place before the starting of the ceremony, is characterized by a progressive increase of the number of people inside the room.

2. The second phase, which took place during the development of the ceremony, is characterized by several people (staying either seated or stand) inside the room, and some people coming in and going out the room.
3. The third phase, which took place at the end of the ceremony, is characterized by the vast majority of people staying stand and leaving the conference room.
4. The fourth phase corresponds to the scenario with no people in the room, thus giving a virtually static indoor scenario with almost fixed propagation characteristics.

The WSN's setup used during the event is characterized by the following main setting: i) nine anchor nodes distributed on the room's perimeter (i.e. in direct communication each other) broadcast their position every 800ms on a time division basis in order to avoid collisions, ii) a blind node fixed in the middle of the room estimates its position every 8s, averaging over 10 RSS acquisition per anchor, iii) the anchor nodes are located at 115cm above the ground floor on the top of wood supports, iv) the blind node is located 115cm above the ground floor during the first three phases, while it is 59cm above the ground floor during the last phase. Moreover, four case studies have been investigated and briefly described in the following.

#### **5.4.2 Static Calibration with Measurement Grid – Conference Room Empty (1)**

The first case study is related to a static estimation of the propagation parameters needed by the location engine. As described in Section 5.1, the parameters have been estimated in the conference room when it was empty, i.e., no chairs and desks were in the room, and with a grid of 44 "test" nodes deployed 115cm above the ground floor.

This off-line calibration leads to the definition of a curve similar to the one shown in Fig. 7, but whose fitting parameters for the present testbed platform are  $A = 39.29$  and  $n = 2.23$ .

#### **5.4.3 Static Calibration with Anchor Nodes – Conference Room with Furniture (2)**

The second case study is still related to a static estimation of the propagation parameters needed by the location engine. However, with respect to the first case study, the propagation parameters are estimated in the conference room with furniture. Moreover, similar to the first case study, the propagation parameters are estimated just once, and are not updated during the progress of the opening ceremony.

However, the main difference with the previous case study is that  $A$  and  $n$  are not estimated by resorting to a grid of "test" nodes. In contrast to the usual method described by Aamodt (2008), we let anchor nodes performing an adaptive estimation of the propagation parameters  $A$  and  $n$ , by resorting to the knowledge of their positions, thus their mutual distances, and performing a least-squares best linear fitting of the couples  $(RSS; d)$  of the Equation 6 Tennina et al. (n.d.).

#### **5.4.4 Dynamic Calibration with Anchor Nodes – Continuous Training during the NCSlab Opening Ceremony (3)**

In this third case study, we use the same approach as in Case 2 for the estimation of parameters  $A$  and  $n$ . However, these parameters are not estimated once, but are continuously updated on a regular basis during the whole development of the opening ceremony. In Fig. 10, the estimated propagation parameters are reported as a function of time. These parameters are those estimated by the blind node, and computed as the arithmetic average of those estimated by the anchor nodes. We can readily figure out that there is a significant fluctuation of these parameters during the progress of the conference. This figure qualitatively suggests that using



an outdated estimate for the channel parameters may certainly yield less accurate estimates of the distances and thus of the final position estimation of the blind node.

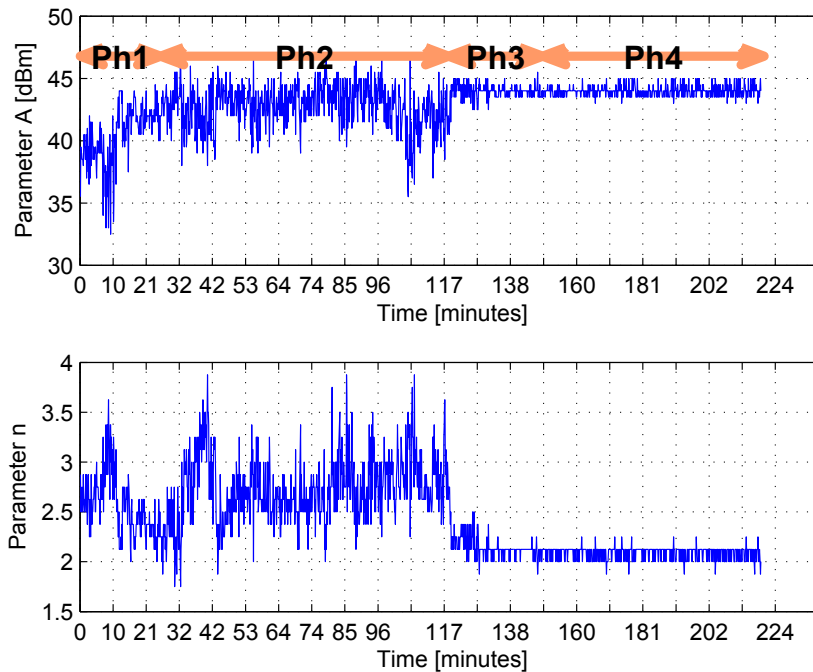


Fig. 10. Estimated propagation parameters during the NCSlab's opening ceremony.

#### 5.4.5 Dynamic Calibration with Anchor Nodes – Off-Line Refinement using the ESD Algorithm (4)

The last case study foresees the same scenario and methods already described in Case 3. However, we introduce a refinement operation to improve the localization accuracy of the system. In particular, the position estimated by the location engine in Case 3 is not considered as the final estimated position of the blind node, but it represents the input for the ESD algorithm.

#### 5.5 Results CC2431

In order to understand the improvement of dynamic updating the channel-dependent parameters, we can look at Table 2. The following conclusions can be drawn. i) For a fixed phase, the performance improves significantly when  $A$  and  $n$  are updated during the progress of the conference (third column). ii) The improvement is more remarkable during phase two, which is a very dynamic phase and where the dynamic adaptation is more important. iii) The continuous training is also beneficial in phases one and three, but the improvement is less evident due to the short duration of these two phases. iv) Apart from the case study described in Case 1 (first column), using the ESD algorithm to refine the estimated position is always beneficial to improve the accuracy. v) The reason why the ESD does not improve the performance in the first case study is due to the fact that the ESD needs the RSSI-to-distance curve to refine the position. Since this curve is not updated continuously in the first two case studies, the algorithm may diverge from the actual solution, as we have in column one. This conclusion is also confirmed by the fact that in an almost static scenario (phase four), the ESD improves the overall accuracy also without updating the channel-dependent parameters. vi) The larger

Phase	Case 1	Case 2	Case 3	Case 4
1	2.69 (3.72)	2.85 (2.27)	2.20	2.00
2	3.04 (5.52)	3.19 (2.97)	1.22	1.03
3	2.77 (3.46)	2.94 (2.26)	2.72	1.93
4	3.04 (1.79)	3.04 (1.32)	2.11	1.28

Table 2. Average positioning error [meters] over the observation time. The value shown into the parentheses in the first two columns represents the improvement that can be obtained refining the search with the ESD algorithm (similar to Case 4).

error that can be observed in column two with respect to column one is probably due to the smaller number of points used to estimate the calibration curve (in both cases  $A$  and  $n$  are not updated during the progress of the conference). However, the difference is in the order of few tens of centimeters, and thus can be acceptable. vii) Finally, we note that, as described in Section 5.4, the results in phase four cannot be directly compared to the results in the other phases as the position of the blind node was different. However, also in this case the accuracy improves when moving from column one to column four.

## 6. Conclusions

In the present chapter, we report our recent research advances along two main directions. Firstly, we adopted the Platform Based Design methodology for the efficient design of ad-hoc WSNs with localization capabilities. In particular, the PBD paradigm has been used to derive a fully distributed positioning algorithm, that we call ESD, and a general protocol architecture for WSNs. The proposed solution has been compared with other well-known positioning algorithms available in the open technical literature, and the improvement provided by the proposed ESD algorithm has been clearly assessed by resorting to computer-based simulations in both network-wide and atomic scenarios. Secondly, we have validated the suitability of a practical implementation of the proposed solution onto commercially available WSN platforms, and analyzed their achievable performance in realistic propagation environments. Results have clearly shown that the ESD algorithm can be actually implemented in CrossBow's MICAz sensor node platforms with a modest computational complexity and with good localization performance even when using RSS-based ranging methods. The experiments conducted with the TI/Chipcon's CC2431 sensor node platform have confirmed this in typical and dynamic environments with any a priori knowledge of channel behavior. Although most of the results described in the present contribution are related to the performance of the localization algorithms in terms of accuracy, robustness, convergence speed, complexity, and storage requirements of static nodes, we are currently deploying a WSN testbed to focus our future analysis on i) the evaluation of the energy consumptions of the ESD algorithm, and its comparison with other positioning algorithms, and ii) the analysis of the performance when this solution is used to track people or objects moving in typical GPS-less environments.

## Acknowledgements

The authors would like to express their gratitude to the research and technical staff of the Center of Excellence in Research DEWS and the Networked Control Systems Laboratory (NC-Slab) for their assistance in conducting the experimental activity. Special thanks go to Alessia D'Alessandro (B.Sc., ECE) and Andrea Scarinci (B.Sc., ECE) for conducting part of the experimental measurements with the TI/Chipcon's CC2431 testbed platform.

## 7. References

- Aamodt, K. (2008). CC2431 location engine, *Application note AN042*. Chipcon products for Texas Instruments, 20 pages.
- Alesii, R., Antonini, F., Di Renzo, M., Graziosi, F. & Santucci, F. (2004). Performance of a Chip-Time Analog Differential Receiver for UWB Systems in a Log-Normal Frequency-Selective Fading Channel, *Wireless Personal Multimedia Communications WPMC04, Abiano Terme, Italy*.
- Alesii, R., Di Renzo, M., Graziosi, F. & Santucci, F. (2004). A Low-Complexity Receiver for Ultra Wide Band Communications, *IEEE Euro Electromagnetics Congress, EuroEM04, Magdeburg, Germany*.
- Bachrach, J. & Taylor, C. (2005). *Handbook of Sensor Networks: Algorithms and Architectures - Localization in Sensor Networks*, Wiley.
- Balluchi, A., Di Benedetto, M., Ferrari, A., Girasole, G., Graziosi, F., Parasiliti, F., Pinto, A., Pesserone, R., Petrella, R., Sangiovanni-Vincentelli, A., Santucci, F., Sgroi, M., Tursini, M., Alesii, R. & Tennina, S. (2004). Platform Based Design: Applications and Flow, *Deliverable DPBD2, Version 1, Project IST-2001-38314 COLUMBUS*.  
**URL:** [http://columbus.gr/documents/public/WPPBD/Columbus\\_DPBD2.pdf](http://columbus.gr/documents/public/WPPBD/Columbus_DPBD2.pdf)
- Bharghavan, V., Demers, A., Shenker, S. & Zhang, L. (1994). MACAW: a media access protocol for wireless LANs, *ACM SIGCOMM*.
- Bonivento, A., Carloni, L. & Sangiovanni-Vincentelli, A. (2005). Rialto: a Bridge between Description and Implementation of Control Algorithms for Wireless Sensor Networks, *ACM International Conference on Embedded Software, EMSOFT'05, Jersey City, NJ, USA*.
- Bulusu, N., Heidemann, J. & Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices, *IEEE Wireless Communications* 7(10): 28–34.
- Calamari Project* (n.d.). <http://www.cs.berkeley.edu/~kamin/calamari>.
- Cro (2008). *CrossBow's Motes*.
- da Silva, J., Sgroi, M., De Bernardinis, F., Li, S.-F., Sangiovanni-Vincentelli, A. & Rabaey, J. (2000). Wireless Protocols Design: Challenges and Opportunities, *8th IEEE International Workshop on Hardware/Software Codesign, S.Diego, CA, USA*.
- Dohler, M. (2008). Wireless sensor networks: the biggest cross-community design exercise to-date, *Bentham Recent Patents on Computer Science*, Vol. 1, pp. 9–25.
- Dulman, S., Baggio, A., Havinga, P. & Langendoen, K. (2008). A geometrical perspective on localization, *1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pp. 85–90.
- Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E. & Culler, D. (2003). The NesC language: a holistic approach to networked embedded systems, *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 1–11. ISBN:1-58113-662-5.

- Goldsmith, A. J. & Wicker, S. B. (2002). Design Challenges for Energy-Constrained Ad-Hoc Wireless Networks, *IEEE Communication Magazine* 9: 8–27.
- Hofmann-Wellenhof, B., Lichtenegger, H. & Collins, J. (1997). *Global positioning system: theory and practice*.
- IEEE Std 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) (2006). IEEE Computer Society.
- Kawadia, V. & Kumar, P. (2005). A Cautionary Perspective on Cross Layer Design, *IEEE Communications Magazine*.
- Langendoen, K. & Reijers, N. (2003). Distributed localization in wireless sensor networks: a quantitative comparison, *IEEE Computer Networks* 43(4): 499–518.
- Mauve, M. & Widmer, J. (2001). A survey on position-based routing in mobile ad hoc networks, *IEEE Networks* 15(6): 30–39.
- Newport, C., Kotz, D., Yuan, Y., Gray, R., Liu, J. & Elliott, C. (2007). Experimental evaluation of wireless simulation assumptions, *Simulation* 83(9): 643–661.
- Nocedal, J. & Wright, S. (2006). *Numerical Optimization*, 2nd edn, Springer.
- Patwari, N., Ash, J. N., Kyperountas, S., Hero III, A. O., Moses, R. L. & Correal, N. S. (2005). Locating the Nodes, *IEEE Signal Processing Magazine* 22(4): 54–69.
- Perkins, D. D., Tumati, R., Wu, H. & Ajbar, I. (2006). Localization in Wireless Ad Hoc Networks, *Resource Management in Wireless Networking*, Springer US, Vol. 16, pp. 507–542.
- Pinto, A. (2008). COSI: COmmunication Synthesis Infrastructure, *GSRC Annual Symposium*.
- Pinto, A., D'Angelo, M., Fischione, C., Scholte, E. & Sangiovanni-Vincentelli, A. (2008). Synthesis of Embedded Networks for Building Automation and Control, *American Control Conference (ACC 08)*, Seattle, Washington.
- Sangiovanni-Vincentelli, A. (2002). Defining Platform-based Design, *EEDesign of EETimes*. (available for download at [www.gigascale.org/pubs/141.html](http://www.gigascale.org/pubs/141.html)).
- Santucci, F., Graziosi, F. & Tennina, S. (2006). Location Service Design and Simulation in Ad-Hoc Wireless Sensor Networks, *International Journal on Mobile Networks Design and Innovation* 1(3/4): 208–214.
- Savarese, C. (2002). *Robust positioning algorithms for distributed ad-hoc wireless sensor networks*, Degree of master of science, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.
- Savvides, A., Han, C.-C. & Srivastava, M. (2001). Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors, *International Conference on Mobile Computing and Networking, Rome, Italy*, pp. 166–179. ISBN: 1-58113-422-3.
- Sgroi, M., da Silva, J., F., D. B., Burghardt, F., Sangiovanni-Vincentelli, A. & Rabaey, J. (2000). Designing Wireless Protocols: Methodology and Applications, *ICASSP*.
- Sgroi, M., Wolisz, A., Sangiovanni-Vincentelli, A. & Rabaey, J. (2003). A service-based universal application interface for ad-hoc wireless sensor networks (draft).  
URL: <http://www.gigascale.org/pubs/435.html>
- Srivastava, V. & Motani, M. (2005). Cross-layer design: a survey and the road ahead, *IEEE Communications Magazine*.
- Stiffler, J. (1968). Rapid Acquisition Sequences, *IEEE Transactions on Information Theory* IT-14(2).
- Tennina, S., Di Renzo, M., Santucci, F. & Graziosi, F. (n.d.). ESD: A Novel Optimization Algorithm for Positioning Estimation of WSNs in GPS-denied Environments – From Simulation to Experimentation, *International Journal of Sensor Networks*. (in press).
- Tex (2007). *CC2431 Data Sheet: System-on-chip for 2.4 GHz zigbee/IEEE 802.15.4 with location engine*. Chipcon products for Texas Instruments, 15 pages.

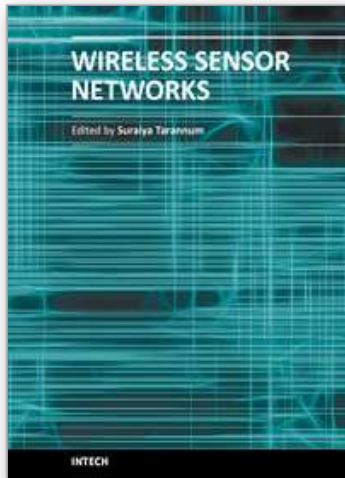
Tex (2008). *TI's website*.

Varga, A. (n.d.). The OMNeT++ Discrete Event Simulation System, *European Simulation Multiconference (ESM), Prague*.

Wang, C. & Xiao, L. (2007). Sensor Localization under Limited Measurement Capabilities, *IEEE Networks* **21**(3): 16–23.

IntechOpen

IntechOpen



## Wireless Sensor Networks

Edited by

ISBN 978-953-307-325-5

Hard cover, 342 pages

**Publisher** InTech

**Published online** 29, June, 2011

**Published in print edition** June, 2011

### How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Stefano Tennina, Marco Di Renzo, Fabio Graziosi and Fortunato Santucci (2011). Distributed Localization Algorithms for Wireless Sensor Networks: From Design Methodology to Experimental Validation, *Wireless Sensor Networks*, (Ed.), ISBN: 978-953-307-325-5, InTech, Available from: <http://www.intechopen.com/books/wireless-sensor-networks/distributed-localization-algorithms-for-wireless-sensor-networks-from-design-methodology-to-experime>

# INTECH

open science | open minds

### InTech Europe

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

INTECHOPEN

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen