

# Accurate Anchor-Free Node Localization in Wireless Sensor Networks

Adel Youssef, and Ashok Agrawala  
{adel, agrawala}@cs.umd.edu  
Department of Computer Science  
University of Maryland  
College Park, MD 20742

Mohamed Younis  
*younis@cs.umbc.edu*  
Department of Computer Science and  
Elect. Eng.  
University of Maryland  
Baltimore, MD 21250

CS-TR-4426

UMIACS-TR-2004-56

July 2004

## Abstract

There has been a growing interest in the applications of wireless sensor networks in unattended environments. In such applications, sensor nodes are usually deployed randomly in an area of interest. Knowledge of accurate node location is essential in such network setups in order to correlate the reported data to the origin of the sensed phenomena. In addition, awareness of the nodes' positions can enable employing efficient management strategies such as geographic routing and conducting important analyses such as node coverage properties. In this paper, we present an efficient anchor-free protocol for localization in wireless sensor networks. Each node discovers its neighbors that are within its transmission range and estimates their ranges. Our algorithm fuses local range measurements in order to form a network wide unified coordinate systems while minimizing the overhead incurred at the deployed sensors. Scalability is achieved through grouping sensors into clusters. Simulation results show that the proposed protocol achieves precise localization of sensors and maintains consistent error margins. In addition, we capture the effect of error accumulation of the node's range estimates and network's size and connectivity on the overall accuracy of the unified coordinate system.

**Keywords:** Wireless sensor networks, Anchor-free localization, Network bootstrapping, Relative positioning, Node discovery protocols.

## 1. Introduction

In recent years there have been major advances in the development of low power micro sensors. The emergence of such sensors has led practitioners to envision networking of a large set of sensors scattered over a wide area of interest [1][2][3][4][5][6]. A typical architecture of a sensor network consists of many sensing devices that are capable of probing the environment and reporting the collected data, using a radio, to a command center [7][8]. Sensor networks can serve many civil and military applications such as habitat monitoring, disaster management, combat field surveillance and security. In such applications, sensors are often left unattended and are significantly constrained in the amount of available resources such as energy, storage and computational capacity. These constraints make the design and operation of sensor networks considerably different from contemporary wireless networks, and necessitate the development of resource conscious protocols and management techniques.

In applications with unattended and/or remote deployment of sensors, nodes are usually thrown or randomly spread in cheer numbers over the area that is to be probed [1][9][10]. Without node localization, such a non-deterministic placement of nodes can hinder the effectiveness of sensors networks and in many cases it can diminish its value. For example in target tracking applications the sensor readings have to be correlated to the sensor position in order to locate the target [11]. In addition, many network protocols such as geographic addressing and routing [12][13] and power-controlled MAC protocols [14][15] take advantage of knowing the node's location to enhance network performance. Moreover, estimating nodes positions is a key to studying the coverage properties of the network and predicting the quality level at which a sensor network would serve its application [16]. Awareness of the sensors' location can also enable the selection of a subset of the nodes to be active and turning off the disengaged sensors for increased network lifetime [17].

Accuracy and scalability are the two main goals that should be considered while designing a localization protocol for sensor networks. Most of the localization algorithms reported in the literature have focused on using a number of specialized nodes that know their positions. Such specialized nodes are usually referred to as *anchors nodes*. The rest of the nodes try to estimate their location by exchanging information to collectively determine their distances to the anchors. Most of the anchor-based algorithms require a high percentage of anchor nodes in order to reach an acceptable accuracy [30]. Besides, the inclusion of a GPS receiver on each node is not practical due to the increased sensor complexity, energy consumption, form factor and cost, in addition to the sensitivity of GPS receivers to line of sight conditions. Moreover, most of these algorithms suffer from scalability problem. For example, the solutions proposed in [18][20][21][22][25][27][28] assume that anchors' positions are flooded into the network, which implies a scalability issue since a network-wide flood will become much too expensive for large networks. Also the localization systems discussed in [23][26] requires centralized computations. Although, a centralized localization algorithm can achieve a higher accuracy compared with a fully distributed one, it is not scalable.

In this paper, we propose an anchor-free locally-centralized localization protocol that can determine the position of sensor nodes consistently with low error margins. We assume that there is no anchor nodes with know position. The network is divided into clusters each with its own gateway node. Each gateway is responsible for building a local relative map corresponding to its cluster using intra-cluster nodes' range measurements. We formulate an optimization model to minimize the cumulative errors that may affect the accuracy of the established relative coordinate system. The gateways collaboratively combine their local maps to obtain the global relative topology of the network. A major motivation for our approach is that we believe that locally centralized algorithms scale well with increased network size and are robust to network partitioning and node failure. Yet, they can achieve acceptable accuracy compared to a centralized approach. A locally centralized algorithm should be a good compromise between accuracy, communication overhead, depending on the size of the cluster and the location of the cluster head. While we give an overview of the entire protocol, the focus of this paper is on the accuracy of the intra-cluster location discovery. The accuracy of the proposed scheme is evaluated through simulation. The results confirm the high accuracy of the positions estimated by our approach and capture the impact of the different parameters, such as cluster size and connectivity on the performance.

The remainder of this paper is organized as follows. In the balance of this section we describe the considered system model and discuss related work. Section 2 describes our localization approach. Validation and performance evaluation of our approach can be found in Section 3. Finally, the paper concludes in Section 4 with a summary and an outline of our future research plan.

### 1.1. System Architecture

A set of sensors is spread throughout an area of interest to detect and possibly track events/targets in this area. A unique ID is assigned to each node prior to deployment. There are two kinds of nodes in the network: sensor nodes and gateway (sink) nodes. Each sensor node is equipped with data processing and communication capabilities. All non-gateway nodes have an identical transmission range. All communication is over wireless links. A wireless link is established between two nodes only if they are in the transmission range of each other. Each sensor node is capable of estimating the distance to neighboring nodes that are within its transmission range using Time of Arrival (TOA) technology[20]. Sensor nodes are assumed to be stationary and are divided into clusters. Each cluster has only one gateway node. Sensor nodes are assigned to clusters based on their proximity to the gateways. The gateway nodes are assumed to be less-energy constrained and they are capable of long-haul communications compared to sensor nodes. Gateways are assumed to be in the communication range of one another and they communicate with the task manager node via Internet or satellite. The gateway is responsible for organizing the activities at sensor nodes to achieve a mission, fusing data collected by sensor nodes and interacting with command nodes. Gateway nodes share relevant data among themselves in order to form a global view of the network topology. The architecture is depicted in Fig 1.

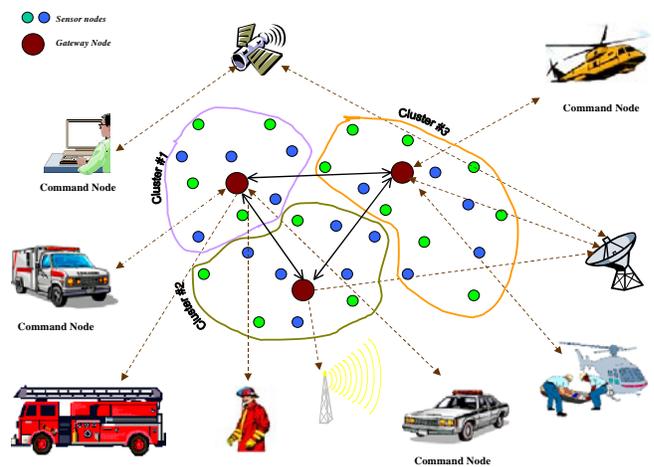


Fig. 1: Multi-cluster sensor network architecture

### 1.2. Related Work

Node localization has been the topic of active research and a number of systems have been proposed over the past few years. Many of those systems fall into one of three classes or a combination of them. The first class includes *range-free* algorithms, which assume that there is no distance/angle information available at each node [18][26][27][29]. Hence, they try to use the basic proximity information available at each node, i.e. which nodes are nearby. For example, the GPS-less system [29] estimates the position of a node as the centroid of positions of connected anchor nodes while the APS system, described in [27], uses the number of hops between two nodes as

an estimate of the distance between these two nodes. In general, range-free techniques suffer from lack of accuracy. The second class employs a number of specialized, *anchor*, nodes that know their positions usually using GPS [20][22][24][25][28]. The rest of the nodes try to estimate their positions relative to these anchors. For example, in the iterative multilateration method [28], an unknown node that is connected to at least three anchors estimates its position by solving a system of equations. Once a node estimates its position it becomes an anchor and assists other unknown nodes in estimating their positions by propagating its own location estimate through the network. Most of the methods in this class require a high percentage of anchor nodes. Langendoen et al. [30] showed that with anchor density of 20%, we could have an accuracy of 25% of transmission range, which falls short from the required inaccuracy in many applications.

The third class of localization systems tries to compute nodes' positions without the use of anchor nodes (i.e. *anchor-free*)[19][23][31]. In this case, instead of computing absolute node positions, the algorithm estimates nodes' positions relative to a coordinate system established by a reference group of nodes. Relative positioning can be sufficient for many applications to work efficiently, for example, location-aided routing [12][13]. Moreover, a relative coordinate system can still be transformed to absolute coordinate system by using only three anchor nodes in case of 2-D (or four anchors in case of 3-D). Schemes in this class can be range-free or range-based.

The multi-dimensional scaling (MDS) described in [19] is an example of the range-free anchor-free methods where each node computes a local map for nodes that are within 2 hops using mainly node connectivity. Then all the nodes in the network communicate with each other to merge these local maps together to form a global map. Besides the scalability problem in finding the global map, MDS-based methods are often good at finding the general layout of the network, but not the precise locations of nodes. The reported inaccuracy is 40% of transmission range, which is too low. An optional refinement phase was suggested to enhance the position accuracy using least squares minimization. The inaccuracy was shown to drop to 31% of the transmission range. Compared to the MDS-based techniques, our algorithm achieves a higher accuracy with less computational overhead.

On the other hand range-based anchor-free methods such as the Self-Positioning Algorithm (SPA) [31] find relative positions in the network using distance measurements between the nodes. Each node builds its own local coordinate system, estimates the positions of one-hop neighbors using triangulation and broadcasts this information to all the nodes in the network to build a global network coordinate system. In addition to the large communication overhead needed to build the global network topology, the achievable accuracy is still limited.

Instead of forming a local coordinate system at each node like SPA, we build a cluster-wide coordinate system only at each gateway node. In this case we gain the following benefits: (1) since the gateway node is assumed to be less-resource constrained, we can use non-linear optimization techniques to estimate the nodes' position more accurately; (2) the communication overhead to build global network topology is reduced since only gateway nodes communicate with each other. While other anchor-free mechanisms consider nodes that 1 or 2 hops away, we estimate the position of nodes that are within  $k$ -hops from the gateway node since the gateway serves the entire cluster. We refer to  $k$  as the cluster radius and it is a parameter in our algorithm. A problem that occurs here is the error accumulated in the node position as it becomes multihop away from the gateway node. One of the contributions of this paper is to show how the error accumulates, as the node becomes  $k$ -hop away from the gateway node, and what factors affect this error accumulation. We also try to answer the following questions: (1) what is the effect of selecting the local coordinate system on the accuracy of the estimated position and computational overhead? (2) how accurate is the estimated position using triangulation only versus using triangulation plus non-linear optimization.

## 2. Anchor-free Localization Protocol

The anchor-free localization problem in ad-hoc sensor networks can be formalized as follows: "Given an ad-hoc network where each node knows the distance measurements, perhaps with some high margin of error, between nodes that are within its listening range, the objective is to construct a local map with accurate relative node

positions.” In this section we present a novel protocol for efficiently and accurately determining node’s position and forming a relative coordinate system. In the next subsection, we present an overview of the protocol. The detailed position estimation algorithm is presented in section 2.2.

## 2.1. Protocol Overview

The proposed location discovery protocol consists of three phases: network-bootstrapping (*NB*), local position discovery (*LPD*), and global localization (*GL*). This paper is mainly dedicated to report on the *LPD* phase, describing detailed mathematical formulation and validating the accuracy of the proposed algorithm. In other words, we focus on the intra-cluster part of our localization scheme. In this section we briefly describe all phases and elaborate on the *LPD* phase in the next subsection.

*The Network Bootstrapping (NB) Phase:* The main tasks performed in the bootstrapping phase are: node discovery, range estimation, and cluster formation. At the end of this phase, all gateway and sensor nodes within  $k$ -hops, where  $k$  is the cluster radius defined in more details in section 3, are made aware of the presence of each other. We follow a technique similar to TinyOS beaconing [32] with a goal to build a breadth first spanning tree rooted at the gateway node such that there is at least one route from each sensor node to a gateway. Each gateway node broadcasts a node discovery message. The message contains three fields: the gateway ID, the sender ID and hop count to the gateway. All nodes receiving the message record the hop count and the sender (the gateway in the first round of broadcast) and rebroadcast the node discovery message after changing the sender’s field and incrementing the hop count. If a node receives multiple node discovery messages from the same gateway, it designates as a parent the neighbor that is on the path with the minimum number of hops to this gateway.

Recall that each sensor node is capable of estimating the distance to neighboring nodes that are within its transmission range using Time of Arrival (TOA) technology [20] or Radio Signal Strength (RSS) [28]. In this paper, we assume that the TOA method is used to estimate the distances between nodes. After building the spanning tree, each sensor node reports its distance estimates to the gateway. All distance reports received or generated by a node are forwarded to its parent until they reach the gateway node. It should be noted that some nodes may receive node discovery messages from more than one gateway, we will refer to those nodes as *boundary nodes*. Boundary nodes are essential for the global localization phase as we will discuss later. A boundary node should store the gateway ID and the ID of the neighbor sensor (parent) on the path with least number of hops to this gateway, for all the gateways it hears from. Boundary nodes will be affiliated with more than one cluster for the purpose of localization.

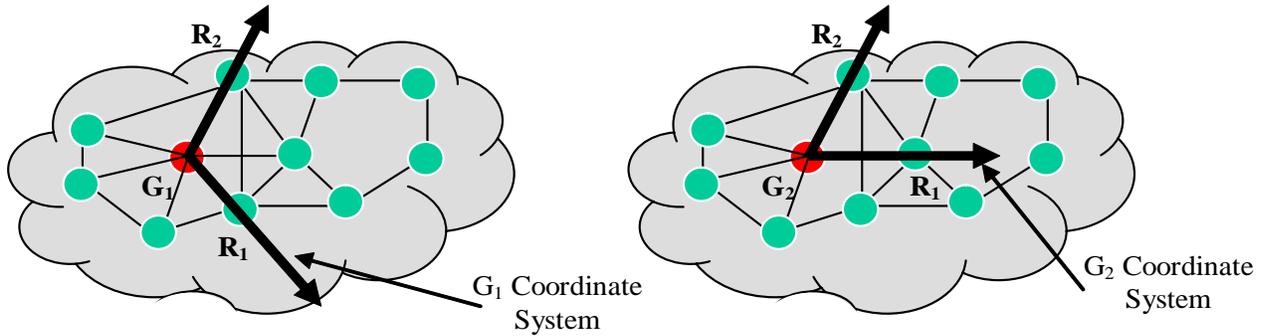
*The Local Position Discovery (LPD) Phase:* The goal of the *LPD* phase is to build a local map at each gateway using the range measurements made by the nodes of the cluster in the *NB* phase. The idea is somewhat similar to the GPS-free localization of [31], which forms a local coordinate system at each node using distances to its next-hop neighbors. However, in our approach we build the coordinate system only at the gateways nodes using distances between nodes that are  $k$ -hop away from the gateway node, where  $k$  is an input parameter to the algorithm. Not only our approach will alleviate the overhead at the sensor node but also enhance the accuracy of the estimated position. As we will show in subsection 2.2 and validate in section 3, we perform optimization to counter the cumulative errors in range measurements at the node level.

The gateway assumes that it is located at the origin of the cluster coordinate system and selects one neighbor node to form the  $x$ -axis, we call this neighbor node the first reference point  $R_1$ . A second reference point ( $R_2$ ) is used to resolve reflection. The  $y$ -axis is selected to be perpendicular to the  $x$ -axis in the direction of  $R_2$ . Then using a combination of triangulation and trilateration techniques, we try to find an initial position estimate ( $P_0$ ) for the nodes located within the cluster using the received measurements of inter-node distances. Finally, the gateway refines the initial position estimate by formulating a least-squares metric and solving it using non-linear optimization techniques. The details of the position estimation algorithm are presented in section 2.2.

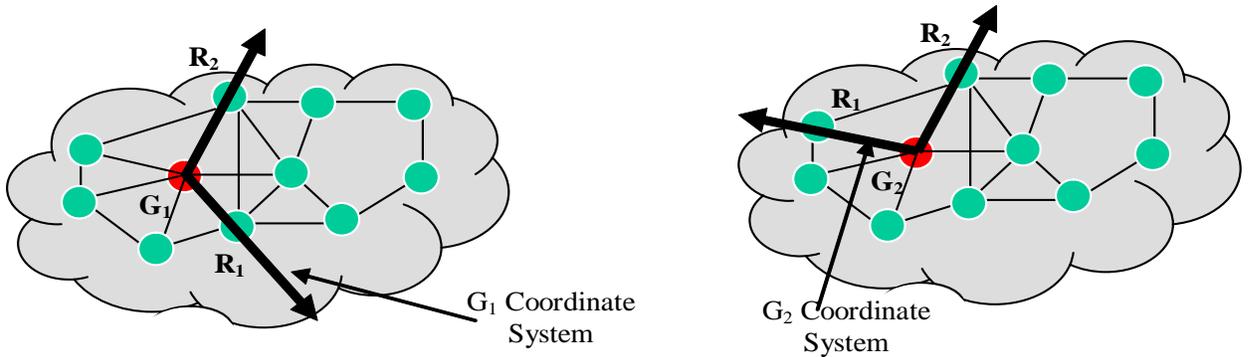
*The Global Localization (GL) Phase:* In the *GL* phase, the gateways collaborate to obtain a global map of the network. This phase can be optional if a global view of the network is not needed, e.g. when the gateways do not perform joint application tasks. After forming a cluster-level map during the *LPD* phase, the gateways positions

are set to  $(0, 0)$  according to the coordinates system of their respective clusters. The axes of local cluster-level maps usually have different directions. Two local maps have the same direction if their  $x$ -axes are pointing in the same direction and similarly for the  $y$ -axes (and  $z$ -axes in case of 3-D). A global coordinate system can be built from the local maps available at each gateway using simple matrix rotations, translations, and mirroring.

We briefly describe how to adjust the directions of the local maps of the gateway nodes to obtain the global topology of the network using boundary nodes. Let  $G_1$ , and  $G_2$  be two gateway nodes. There are two possible scenarios depending on whether mirroring is needed or not. In the first scenario, shown in Fig. 2, only rotation and translation are needed in order for the two gateways to have the same coordinate systems. In the second scenario, shown in Fig. 3, rotation and translation are not enough. The coordinate system of  $G_1$  needs to be mirrored around one of its axes after the rotation.



**Fig. 2:** Only rotation and translation are needed to map from  $G_1$  coordinate system to  $G_2$  coordinate system



**Fig. 3:** Rotation, translation, and mirroring are needed to map from  $G_1$  coordinate system to  $G_2$  coordinate system

To be able to perform such transformation, there must be at least three boundary nodes that belong to both clusters (i.e. within  $k$ -hops from both gateways). Since range measurements are typically inaccurate, we do not expect to find a transformation that maps the node coordinates of one cluster exactly into the measured coordinates of these nodes in the other cluster. Instead we formulate and solve another optimization problem by minimizing the sum of the squares of the residual errors as follows:

Let  $G_1$  and  $G_2$  be two adjacent clusters that have  $m$  common boundary nodes and  $m \geq 3$ . Let  $v_i(G_1)$  and  $v_i(G_2)$  be the coordinates of boundary node  $i$  in  $G_1$  and  $G_2$  respectively, where  $i = 1..m$ . The objective is to find the transformation matrix  $M_{12}$ , that maps node coordinates of  $G_1$  into  $G_2$ , and minimizes the following error function:

$$\min E(M_{12}) = \sum_{i=1}^m \|v_i(G_2) - M_{12}v_i(G_1)\|^2$$

Where

Where  $t_x, t_y$  are the translation transformation and possibly mirroring with the transformation,  $r_i$  is the rotation following properties:

$$M_{12} = \begin{bmatrix} r_1 & r_2 & t_x \\ r_3 & r_4 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} |r_1| = |r_4| \\ |r_2| = |r_3| \\ r_1 r_4 - r_2 r_3 = -1 \text{ for mirroring, } 1 \text{ otherwise} \end{cases}$$

The solution of the above minimization problem was shown by Horn et al [33] to take  $O(m)$ .

## 2.2. Positioning Estimation Algorithm

As indicated earlier we focus in this paper on the *LPD* phase of the localization protocol. In this section we present a novel algorithm for establishing a local coordinate system for each cluster. The algorithm can achieve fairly high accuracy through mitigating the effect of implicit errors in the node's range estimate via minimizing the cumulative error and involving nodes that can reach the gateway over multi-hop.

Recall that at the end of the *NB* phase the gateway, receives node-to-node range measurements for each node within  $k$ -hops from the gateway. In the *PLD* phase the gateway builds a local coordinate system as follows:

Let  $(x_i, y_i, z_i)$  be the 3-D position of node  $i$ , where  $i = 0, 1, \dots, n-1$  and  $P$  be an  $(n \times 3)$  matrix representing all node positions such that:

$$P = \begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_{n-1} & y_{n-1} & z_{n-1} \end{bmatrix}$$

Let  $P(i)$  be a function that returns the position of node  $i$  such that:  $P(i) = [x_i, y_i, z_i]$

Assume that  $D$  is an  $(n \times n)$  matrix representing the inter-node distance measurements. These measurements could have an error. Let  $D_p = D(P)$  be a vector function that returns an  $(n \times n)$  matrix representing the inter-node distances given a node position estimate  $P$ , where

$$D_p[i, j] = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, i, j = 0, 1, \dots, n-1$$

Given a certain position estimate  $P$ , define the following vector error function  $E$ :

$$E(P) = \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} (D_p(i, j) - D(i, j))^2 \quad \text{Eq. (1)}$$

The objective is to find the value of  $P$  that minimizes this error function.

Let  $\nabla E(P)$  denotes the gradient of  $E$  at  $P$ . The gradient is calculated as

$$\nabla E(\mathbf{P}) = \begin{bmatrix} \frac{\partial E}{\partial x_0}(\mathbf{P}) & \frac{\partial E}{\partial y_0}(\mathbf{P}) & \frac{\partial E}{\partial z_0}(\mathbf{P}) \\ \frac{\partial E}{\partial x_1}(\mathbf{P}) & \frac{\partial E}{\partial y_1}(\mathbf{P}) & \frac{\partial E}{\partial z_1}(\mathbf{P}) \\ \vdots & \vdots & \vdots \\ \frac{\partial E}{\partial x_{n-1}}(\mathbf{P}) & \frac{\partial E}{\partial y_{n-1}}(\mathbf{P}) & \frac{\partial E}{\partial z_{n-1}}(\mathbf{P}) \end{bmatrix} \quad \text{Eq. (2)}$$

Therefore,  $\nabla E$  is an  $(n \times 3)$  matrix function of  $\mathbf{P}$ .

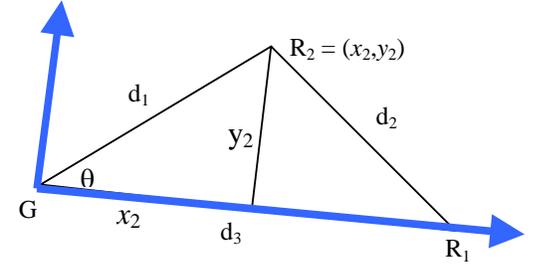
The proposed algorithm consists of two stages: initialization and processing. In the initialization stage, an initial relative position  $P_0$  is estimated. Then in the processing stage, we iteratively improve the initial position estimate. For simplicity, we shall use 2-D coordinates in the following analysis. However, the technique can be easily extended to the 3-D case.

### 2.2.1 The Initialization Stage

The objective of this phase is to find an initial relative position estimate ( $P_0$ ) for all the nodes within the cluster. We use triangulation techniques as discussed in [33], however we extend it to work for nodes that are  $k$  hops away from the gateway node. The initialization phase works as follows:

#### 1- **Selecting a local coordinate system:**

The gateway node ( $G$ ) assumes itself at the origin. Then another node  $R_1$ , that is within the transmission range of  $G$ , is selected to form the positive  $x$ -axis. Finally, a third node that is within the transmission range of both  $G$  and  $R_1$ , but not co-linear with them, will be positioned in the upper half-plane. Thus, the placement of  $R_1$  has the effect of fixing a particular rotational orientation, while the placement of  $R_2$  locks in a particular reflective orientation.



**Fig. 4:** Using the law of cosines to calculate the  $(x_2, y_2)$  position of a node

Hence, the position of the nodes  $G$  and  $R_1$ , is given as follows:

$$P_0(G) = (0,0)$$

$$P_0(R_1) = (D(G, R_1), 0)$$

The coordinates of  $R_2 = (x_2, y_2)$  can be computed with respect to the  $x$ -axis formed by  $\overline{GR_1}$  using the law of cosines as follows (Fig. 4):

$$\cos(\theta) = \frac{d_1^2 + d_3^2 - d_2^2}{2d_1d_3}$$

$$x_2 = \frac{d_1^2 + d_3^2 - d_2^2}{2d_3}$$

$$y_2 = \sqrt{d_1^2 - x_2^2}$$

Where  $d_1 = D(G, R_2)$ ,  $d_2 = D(R_1, R_2)$ ,  $d_3 = D(G, R_1)$ .

- 2- In a similar way, we can compute the initial position estimates for all the nodes  $i$  that are within the transmission range from both  $G$  and  $R_1$  as follows:

Calculate two candidate positions for the node  $i$  ( $(x_i, y_{i1})$  and  $(x_i, y_{i2})$ ):

$$x_i = \frac{d_{i1}^2 + d_3^2 - d_{i2}^2}{2d_3}$$

$$y_{i1} = \sqrt{d_{i1}^2 - x_i^2} \text{ and } y_{i2} = -\sqrt{d_{i1}^2 - x_i^2}$$

Where  $d_{i1} = D(G, i)$ ,  $d_{i2} = D(R_1, i)$ ,  $d_3 = D(G, R_1)$  and  $y_{i1}$  is in the direction of the positive  $y$ -axis and  $y_{i2}$  is in the direction of the negative  $y$ -axis. Then use the distance between node  $i$ , and  $R_2$  in order to choose between the two candidate positions.

- 3- After step 2, all the nodes that are adjacent to both  $G$  and  $R_1$ , have an initial position estimate. We will call those nodes with known position estimate *covered nodes*.
- 4- Now for nodes that are not adjacent to  $G$  and  $R_1$  (i.e. not covered yet), we iteratively estimate their position using the newly covered nodes. If an uncovered node has three or more adjacent covered neighbors, we use multilateration to estimate the uncovered node position and add it to the set of covered nodes.
- 5- We repeat step 4 till all nodes are covered or we have no more nodes with three or more know distances to covered nodes.

Of course, as the node moves away from  $G$ , the error in estimating its initial position in step 4 will increase because of error accumulation. However, the non-linear optimization in the next phase should limit this error. In section 3.2.1, we shall revisit this issue in light of the simulation results. It worth mentioning here that selecting the local coordinate system ( $G, R_1, R_2$ ) has a major effect on the accuracy of the estimated nodes' positions. In section 3.2.2, we discuss three different methods for selecting the nodes ( $G, R_1, R_2$ ) and the accuracy achieved by each method.

### 2.2.2 The Processing Stage

The processing phase iteratively uses gradient descent method to refine the initial position estimates  $P_0$ . The gradient  $\nabla E(P)$ , given in Eq. 2, has the property that when it is evaluated at any position estimate  $P$ , it points in the direction of travel from  $P$  that will maximally increase the error (i.e., uphill). Therefore, to decrease the error  $E$  described in Eq. 1, the value of  $P$  should be slightly changed in the opposite direction (i.e.,  $-\nabla E(P)$ ). The new value of  $P$  at iteration  $j$  is calculated as follows:

$$P_j = P_{j-1} - \lambda_j \nabla E(P_{j-1})$$

Where  $P_0$  is provided by the initialization phase. At each iteration,  $E(P_j) < E(P_{j-1})$  as long as the parameter  $\lambda_j$  is small enough. The non-linear error function given by Eq. 1 has many local minima; so selecting the initial position estimate  $P_0$  affects the accuracy of the estimated position significantly as well as the convergence latency. We will discuss this in more details in the results section. The terminating condition for the iterative minimization process is when the maximum change in any node position is  $\leq \eta$ , where  $\eta$  is the desired position accuracy.

The complexity of the LPD phase can be computed as follows:

- 1- We need  $O(n)$  iterations to calculate  $P_0$ , where  $n$  is the number of nodes in the cluster.
- 2- At each iteration,, the processing stage takes  $O(n^3)$  to calculate the gradient and  $O(n^3)$  to calculate  $\lambda_j$ . Hence, the complexity of the processing phase is  $O(ln^3)$ , where  $l$  is the number of iterations needed to reach a minimum of the error function  $E$ . In section 3.2.3, we will discuss in more details the performance of the processing phase measured in terms of number of iterations along with the factors affecting the performance.

## 3. Validation and Performance Evaluation

We have validated our intra-cluster position estimation algorithm using simulation. There are five parameters used in our simulation:

- 1- *Cluster size (N)*: the number of nodes in the cluster including the gateway node.
- 2- *Cluster radius (k)*: the maximum number of hops between any node in the cluster and the gateway node. Increasing the cluster radius will automatically increase the cluster size.

- 3- *Node connectivity*: the average node degree in the cluster.
- 4- *Transmission range (R)*: all the non-gateway nodes have the same transmission range. Two nodes are neighbors to each other if they are within the transmission range of each other. Increasing the transmission range of the node will increase node connectivity.
- 5- *Range error*: this is the measurement error associated with each distance between any two nodes. This is dependent on the technology used for distance estimation (TOA, AOA, RSSI). In this paper, we assume that the TOA method is used; hence we assume Gaussian range error with zero mean and variance  $\sigma^2$ .

We also consider the following two performance metrics:

- 1- *Accuracy*: the accuracy of the estimated positions is measured in terms of the median error between the estimated positions and the true node positions. The error is scaled as a percentage of transmission range  $R$ .
- 2- *Convergence latency*: the number of iterations taken till the processing phase terminates (i.e. a minimum for the error function  $E$  is reached)

The overall goal of the following experiments is to quantify these metrics and qualify the impact of the various parameters. Mainly, we are interested in answering the following questions:

- *Q1*: What are the factors (cluster size, cluster radius, connectivity, etc.) that affect error accumulation, as the node becomes  $k$ -hops away from the gateway node?
- *Q2*: Does selecting the local coordinate system  $(G, R_1, R_2)$ , as described in section 2.2.1, affect the accuracy of the estimated position and the convergence latency of the optimization? If so, how to select the local coordinate system? In the simulator, we are trying three different criteria for selecting the local coordinate system.
- *Q3*: If a good local coordinate system were selected, would the initial estimates  $(P_0)$  be close enough to the positions resulting from the optimization? In other words, what added accuracy do we gain by conducting the optimization?

### 3.1. Experiments Setup

The *LDP* algorithm was implemented using MATLAB 6.1 release 12.1. All experiments were performed over 200 different topologies representing different cluster sizes ( $N$ ) ranging from 20 to 60 nodes. For each topology, the transmission range of each node ( $R$ ) was varied in order to achieve different node connectivity levels ranging from 6 to 14. The cluster radius ( $k$ ) ranges from 2 to 5 depending on the cluster size and node connectivity. The nodes were randomly placed according to a uniform distribution on a 100x100 area. The inter-node distance measurements were perturbed with a Gaussian random noise with zero mean and variance  $\sigma^2$ , where  $\sigma^2$  ranges from 0 to 2.

As discussed in section 2.2, the local coordinate system is determined by assuming the gateway node  $G$  at the origin and selecting two reference nodes  $R_1, R_2$  that are adjacent to each other and to  $G$  and the three nodes are not collinear with each other. The three nodes together form a triangle as shown in Fig. 4. We will refer to this triangle by  $\Delta(G, R_1, R_2)$  which in turn refers to the local coordinate system. It turns out that the selection of this triangle affects the overall accuracy of the estimated node positions. Notice that if there are  $n$  nodes in the cluster including the gateway node  $G$ , then there is a maximum of  $C_2^{N-1}$  different possible triangles that can be formed, i.e. different coordinate systems with origin at gateway node. Of course, the actual number is less than this since not all nodes can be selected as  $R_1$  and  $R_2$ . We have tried three different methods to select one coordinate system as follows:

- Lowest Aspect Ratio (LAR): Select  $R_1, R_2$ , such that the triangle  $\Delta(G, R_1, R_2)$  has the lowest aspect ratio among all different candidate triangles. This method usually selects nicely shaped triangles and avoids skinny triangles. The aspect ratio is calculated as the ratio between the circum radius and inner radius of the triangle.

- **Maximum Equilateral Triangle (MET):** This method is similar to LAR but takes the side length of the triangle into consideration. In this case, we search for all approximately equilateral triangles. Then we select the one with maximum side length. An approximately equilateral triangle is a triangle with aspect ratio close to 2.
- **Minimum Initial Error (MIE):** In this method we try all different candidate local coordinate systems. For each coordinate system, we calculate the initial position estimate  $P_0$  as described in section 2.2.1. Then we pick a coordinate system that gives an initial position estimate  $P_0$  with minimum error function  $E(P_0)$  given by Eq. 1. The intuition behind this method is to choose an initial position estimate  $P_0$  such that the error function at this position  $E(P_0)$  is as close as possible to 0.

In each of those methods, we compare between the accuracy of the initially estimated node positions and the one obtained after performing non-linear optimization. Our objective is to find whether and when the non-linear optimization is justified. This can be useful to trade-off complexity and accuracy.

### 3.2. Simulation Results

Reflecting on the goal of the experiment listed earlier, we group the presentation of the simulation results into three sets. The first sets reports the achievable accuracy of our algorithm and captures the effect of number of hops considered ( $k$ ), node degree and range error. The second set studies the effect of the selection of the local coordinate system on the accuracy of the estimated positions and how the network size impacts it. The last set is dedicated to the convergence latency and the added value of the optimization. We basically compare the effect of  $k$ , node degree and range error on the convergence latency. In addition, we compare the quality of the optimized position estimate to that of the initial estimates under the different methods for picking the local coordinate system.

#### 3.2.1 Achievable accuracy

Figures 5 and 6 show how the accuracy of the estimated position is affected by the node connectivity and the cluster radius ( $k$ ). The effect of the range error is also captured in both charts. The median error is scaled as a percentage of the transmission range ( $R$ ). From Fig. 5, it can be concluded that increased node degree has a very positive impact on the overall accuracy so much that range errors become very insignificant. These results confirm the effectiveness of our approach since we take advantage of the measures about the different neighbors in minimizing the cumulative range errors. It is also clear from the figures that increasing the connectivity over 10, will lead to an accuracy of less than 10% of transmission range. Fig. 6 shows that an increased value of the cluster radius worsens the accuracy. This is very much expected since the further the node is, the higher the accumulative range error becomes. Combing the findings of figures 5 and 6, it can be concluded that increasing

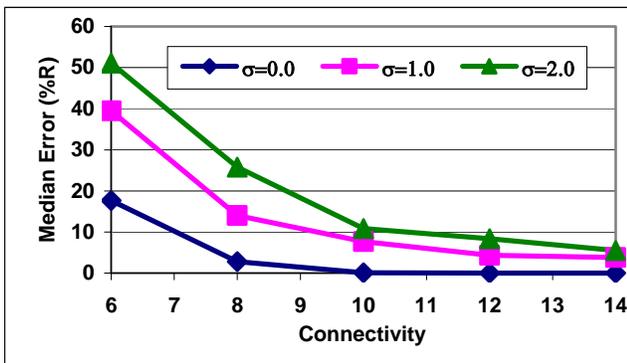


Fig. 5: Impact of node degree on accuracy of estimated positions in the presence of a varying error range ( $=50, k=5$ )

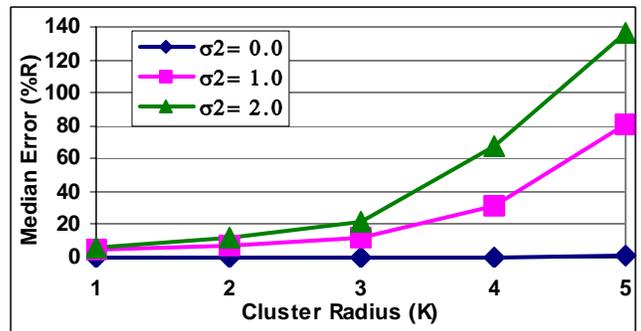
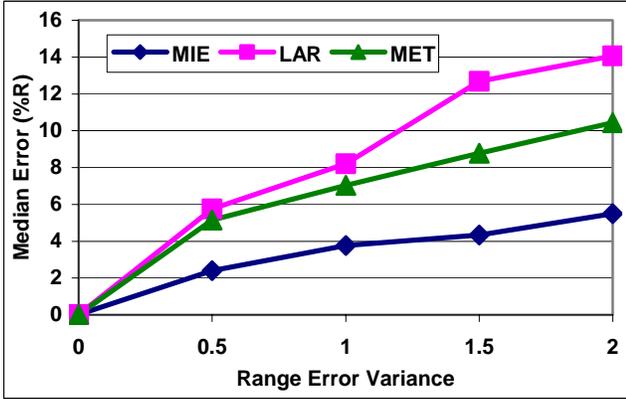
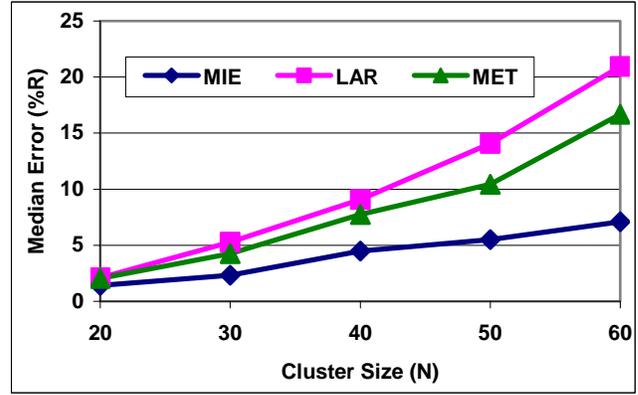


Fig. 6: Relationship of accuracy of estimated positions and number of hops considered “ $k$ ” under different range errors ( $N=50, \text{connectivity} = 6-14$ ).



**Fig. 7:** Effect of error range on accuracy of estimated positions under different local coordinate systems (connectivity=14, cluster size =50)



**Fig. 8:** Accuracy of estimated positions under different local coordinate system (connectivity=14, range error variance=2)

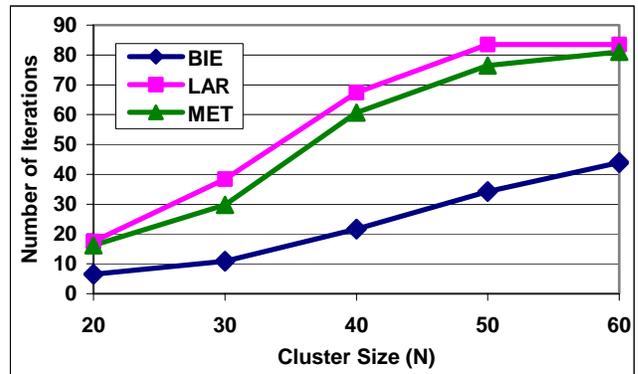
the radius of the cluster is not advisable unless the network connectivity is high in order to maintain high accuracy. This conclusion is actually consistent with the intuition that increasing the number of clusters is advisable for largely partitioned networks to ensure high level of connectivity.

### 3.2.2 Selection of a local coordinate system

The effect of the selection of a local (intra-cluster) coordinate system on achieved accuracy is captured in figures 7 and 8. In general, the experiments clearly indicate that selecting the local coordinate system is one of the most important factors affecting the accuracy of the estimated final nodes' positions. The accuracy obtained if we use MIE is almost double the accuracy obtained using LAR. The gap actually increases as the cluster size increases. From both figures, one can confirm that the MIE approach, which corresponds to minimum initial error  $E(P_0)$ , performs very well. It is also interesting to note that the closeness to the performance of the LAR and MET methods, which is mainly due to the high similarity between the two methods. In the most part MET leads to slightly better accuracy because it considers triangles with large side length. Increasing the side length of the triangle reduces the effect of the error introduced by the range-estimation technology used (TOA).

Fig. 7 indicates that the effect of range error is mostly symmetric on all methods of picking the local coordinate system, with an order of magnitude increase of error variance approximately worsening the accuracy by factor of 2. It is worth noting that the effect of the cluster size is little for small networks size and becomes very noticeable when the network grows in size.

Fig. 9 reflects another implication of the local coordinate system, which is the convergence latency of the optimization. Not only has the MIE method performed well, as demonstrated by figures 7 and 8, it actually expedite the convergence latency of the optimization. Again the increase in cluster size makes the difference between the other methods insignificant in terms of the complexity of the optimization. It is worth noting that the number of iterations increases linearly with the growth in network size demonstrating the scalability of our approach.



**Fig. 9:** The number of iterations for reaching optimal positions estimates as a function of network size (connectivity=14, range error variance=2)

### 3.2.3 Optimization factors

In these set of experiments we have tried to study the factors that impact the convergence latency to optimal position estimates and qualify the value of conducting optimization. Fig. 10 captures the effect of the connectivity and range errors on the convergence latency using the MIE method to select the local coordinate system. Recall that Fig. 9 has demonstrated a linear growth in number of iteration when the network size increases. Fig. 10 shows that a high node degree makes the effect of range error on the convergence latency mute. In addition, it seems that for low range error the node degree does not impact the latency of convergence much. In Fig. 11, we try to show how much accuracy we gain by solving the non-linear optimization problem. The figure compares between the accuracy of the initial position using just two different methods (MIE, MET) and the accuracy of the position obtained before performing optimization (MIE-OPT, MET-OPT). Clearly, when using the MIE method, the accuracy of the initial position is very close to the accuracy of the optimal positions. Moreover, the initial position estimated ( $P_0$ ) using MIE (i.e. MIE-OPT) is more accurate than the position estimated using MET even after performing optimization. Hence, we may want to avoid the computational overhead in the optimization problem and go with the initial position. This gives the application a trade-off between computational power and accuracy. We noticed also that as the node degree increases, the accuracy gained by the optimization process decreases.

## 4. Conclusion and Future Work

Recent advances in micro-electro-mechanical systems technology, and wireless communications have led to the development of inexpensive, low-power micro sensor nodes. Sensor nodes can be deployed by dropping from a plane, delivered in an artillery shell, rocket, or missile. With the ad-hoc and random deployment one cannot accurately predict or plan the placement of each sensor node a-priori. However, knowledge of node location can be an essential requirement for many applications and can also be invaluable for many optimized network management. In this paper, we have proposed a scalable GPS-free location discovery protocol that can perform precise localization of sensor nodes. The proposed approach is cluster based and does not require extensive infrastructure support. We pursue an anchor-free methodology to generate a relative network topology at the cluster level and combines local cluster-based maps into a global network-wide coordinate system.

In the paper we have mainly focused on the intra-cluster node's localization and presented and an effective algorithm for achieving high accuracy while minimizing the involvement of sensor nodes. Simulation results have confirmed the quality of the estimated positions and qualified the impact of the different parameters. Our experiments have concluded that the network connectivity, measured in terms of node's degree, have a very dominant effect on the estimation accuracy. We have also analyzed the effect of selecting the initial local coordinate system and how it significantly affects the accuracy of the results and the convergence latency of the optimization process. We have also shown that we can avoid the computationally expensive optimization problem by spending some time in selecting the coordinate system.

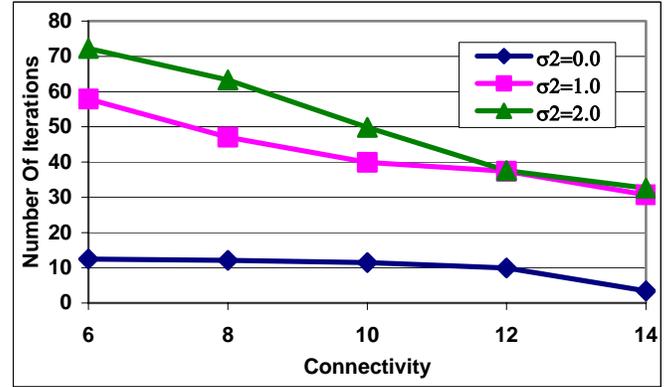


Fig. 10: The number of iterations for reaching optimal positions estimates as a function of node degree for varying range errors

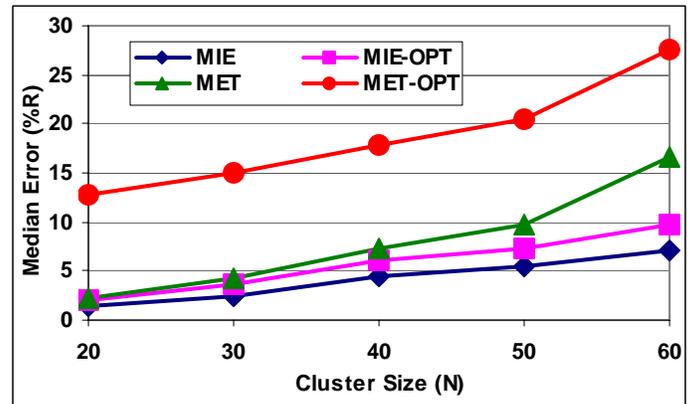


Fig. 11: Impact of optimization on accuracy of estimated positions using different local coordinate systems (connectivity=14, range error variance=2)

We are currently working out and validating the detailed clustering and global localization algorithms. In the future, we aim to analyze the error accumulation and how to limit it. There are two types of error accumulation: intra-cluster error accumulation during the *LPD* phase as  $k$  increases; and inter-cluster error accumulation during the *GL* phase. In this paper we have focused on the first type of errors. We intend to extend the scope to cover inter-cluster error accumulation. As another extension, we also plan to consider the case of homogeneous network where all nodes are equal (i.e. there are no gateway nodes). In this case, a clustering algorithm is needed in order to select a set of cluster heads, which cover the entire network such that each sensor node belongs to at least one cluster.

## References

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, 38(4), pp. 393—422, 2002.
- [2] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless microsensor network models," *ACM Mobile Computing and Communication Review (MC2R)*, 6(2), pp. 1—8, 2002.
- [3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, 43(5), pp. 51--58, 2000.
- [4] J. Rabaey, M. Ammer, J. Silva jr., D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer*, 33(7), pp. 42—48, 2000.
- [5] R. H. Katz, J. M. Kahn and K. S. J. Pister, "Mobile Networking for Smart Dust," in the *Proceedings of the 5<sup>th</sup> Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, August 1999.
- [6] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in the *Proceedings of the 5<sup>th</sup> IEEE/ACM Annual Conference on Mobile Computing and Networks (MobiCOM'99)*, Seattle, WA, August 1999.
- [7] R. Min, et al., "Low Power Wireless Sensor Networks," in the *Proceedings of International Conference on VLSI Design*, Bangalore, India, January 2001.
- [8] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Comm.*, 7(5), pp. 16--27 2000.
- [9] A. Mainwaring, et al., "Wireless Sensor Networks for Habitat Monitoring," In ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, GA, September 2002.
- [10] H. Wang, et al., "Target Classification and Localization in Habitat Monitoring," in the *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, Hong Kong, China. April 2003.
- [11] R. Burne, et. al, "A Self-Organizing, Cooperative UGS Network for Target Tracking," in the *Proceedings of the SPIE Conference on Unattended Ground Sensor Technologies and Applications II*, Orlando Florida, April 2000.
- [12] I. Stojmenovic and X. Lin, "Loop-free Hybrid Single-path Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 10, pages 1023-1032, October 2001.
- [13] J. C. Navas and T. Imielinski, "Geographic Addressing and Routing," In Proc. Of the ACM MobiCom, Budapest, Hungary, September 1997.
- [14] E. Jung and N. Vaidya, "A Power control MAC protocol for Ad Hoc networks," in the *Proceedings of the 8<sup>th</sup> ACM/IEEE Conference on Mobile Computing and Networks (MOBICOM'02)*, Atlanta, GA, September 2002.
- [15] J. Monks, V. Bharghavan, and W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in the *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, April 2001.
- [16] S. Meguerdichian, et al., "Coverage Problems in Wireless Ad-hoc Sensor Networks," in the *Proceedings of IEEE INFOCOM*, Ankorange, Alaska, April 2001.
- [17] A. Buczak and V. Jamalabad, "Self-organization of a Heterogeneous Sensor Network by Genetic Algorithms," *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et. (eds.), Vol. 8, pp. 259-264, ASME Press, New York, 1998.
- [18] X. Ji, "Sensor Positioning in Wireless Ad-hoc Sensor Networks with Multidimensional Scaling," in the *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.
- [19] Y. Shang, and W. Ruml, "Improved MDS-Based Localization," in the *Proceedings of IEEE Conference on Computer Communication (INFOCOM)*, Hong Kong, March 2004.
- [20] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "TPS: A Time-Based Positioning Scheme for Outdoor Sensor Networks," in the *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Hong Kong, March 2004.

- [21] T. He, et al., "Range-Free Localization Schemes for Large Scale Sensor Networks," in the *Proceedings of the 9<sup>th</sup> IEEE/ACM Annual Conference on Mobile Computing and Networks (MobiCOM'03)*, San Diego, California, September 2003.
- [22] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AoA," in the *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 2003.
- [23] Y. Shang, et al., "Localization From Mere Connectivity," in the *Proceedings of ACM MOBIHOC 2003*, Annapolis, MD, June 2003.
- [24] C. Savarese, J. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," *USENIX Technical Annual Conference*, Monterey, CA, June 2002.
- [25] A. Savvides, H. Park, and M. B. Srivastava, "The Bits and Flops of The n-hop Multilateration Primitive for Node Localization Problems," in the *Proceedings of the first ACM international workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002.
- [26] L. Doherty, K. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in the *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, Alaska, April 2001.
- [27] D. Niculescu and B. Nath, "Ad-hoc Positioning system," in the *Proceedings of IEEE Global Communication Conference (Globcom'01)*, San Antonio, TX, November 2001.
- [28] A. Savvides, C. C. Han, and M. Srivastava, "Dynamic Fine-Grained Localization in Ad-hoc Networks of Sensors," in the *Proceedings of the 7<sup>th</sup> IEEE/ACM Annual Conference on Mobile Computing and Networks (MobiCOM'01)*, Rome, Italy, July 2001.
- [29] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low-cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, 7(5):28–34, October 2000.
- [30] K. Langendoen and N. Reijers, "Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison," *Computer Networks (Elsevier), special issue on Wireless Sensor Networks*, August 2003.
- [31] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free Positioning in Mobile Adhoc Networks," in Hawaii International Conference on System Sciences (HICSS-34), pages 3481–3490, Maui, Hawaii, January 2001.. Giordano, I. Stojmenovic, Lj. Blazevic, Position based routing algorithms for ad hoc networks: A taxonomy, in: 'Ad Hoc Wireless Networking', X. Cheng, X. Huang and D.Z. Du (eds.), Kluwer, December 2003.
- [32] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," in the *Proceedings of 1<sup>st</sup> IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, May 2003.
- [33] B. Horn, H.M. Hilden and S. Negahdaripour, "Closed form solution of absolute orientation using orthonormal matrices", *Journal of the Optical Society of America*, Vol. 5, pp. 1127–1135, 1998.