Localization Methods in Wireless Sensor Networks

by
Florian Walbroel

Written Report of the Seminar Communication Theory

27th September 2017

Lehrstuhl für Theoretische Informationstechnik
Betreuer: Jose Angel Leon Calvo, M.Sc.
Prof. Dr. Rudolf Mathar

List of Contents

1 Introduction **5**

2 Theoretical Background **7**
2.1 Communication Technologies . . . 8
2.2 Network Topologies . . . 8
2.3 Data Processing . . . 9
2.3.1 Multisensor Data Fusion . . . 10
2.4 Localization Methods . . . 11
2.4.1 Distance Measurements . . . 11
2.4.2 Multilateration . . . 11
2.5 Targeting Tracking . . . 13

3 Localization Methods **15**
3.1 Multilateration GNA and GDM Comparison . . . 15
3.2 Ad Hoc Positioning System . . . 15

4 Conclusion **23**
4.1 Outlook . . . 23

Algorithms **24**

1 Introduction

Wireless Sensor Networks have emerged in the recent years due to advancements in technology and, thus, decreasing hardware costs. They are thought to be one of the most promising technologies of the future [1].

WSNs are used for a vast range of application domains, including infrastructure or environmental monitoring, crime prevention and health monitoring via body area networks. This wide range of application domains was made possible by the advances of technology in the recent years. The ever decreasing chip sizes and power consumption of *Processing Elements* PEs, which though still provide a lot of computational performance, allowed the desgin of small and integrated sensor nodes for effordable prices, as with the availability of the technology, the cost is decreasing. As WSNs often consist of a huge number of sensor nodes, the price for a single node has a huge impact for the possiblity of the implementation as well as for the scalabity of the network.

Thus WSNs are therefore increasingly applied to a lot of problems of the daily life, e.g., SmartMeters measuring the power consumption are planned by the EU to be installed in every household until the year 2020. But also in the domain of autonomous driving, WSNs are thought to play a roll, as every vehicle can be viewed as an individual moving sensor nodes and the total of all vehicles on the street as a huge network.

In this report a brief introduction into the different topics of WSNs is given. Further localization methods for sensor nodes in the case that not all nodes are equipped with a global positioning module are evaluated. A brief introduction into *Multilaterion* is given and the performance of the *Gauss-Newton-Algorithm* and *Gradient-Descent-Method* is compared. Also a soley software based solution for distance estimation, the *Ad-Hoc Positioning System*, is presented.

The rest of the report is structured as followed. In Chapter 2 background information on the different aspects of WSNs like comunication technologies and network topologies are covered. In Chapter 3 two examples for localization methods are examplary discussed and evaluated. The report is concluded in Chapter 4.

2 Theoretical Background

A *Wireless Sensor Network* (WSN) is a set of wireless sensor nodes. In general each wireless sensor node has the same basic structure. It is equipped with a *processing unit* (PU), a radio interface, a set of sensors, a memory and a power supply. The hardware installed in a sensor node defines the type of the node. WSNs with only the same node type are called homogenous and with different types of nodes heterogenous [1].

In general there are two types of applications for WSNs: monitoring and tracking. Originally the development of WSNs was motivated by military applications such as battlefield surveillance, though a lot of civil applications have emerged in the recent years. Leading application domains of WSNs include military, crime prevention, environment, health, agricultural, industry and infrastructure.

The PU is the core of the wireless sensor node. Due to power and size constraints, commonly microprocessors like the *ATMEL ATmega128* [2] or *ARM Cortex M3* [3] are used. The PU determines all of the actions performed by the node, e.g., starting a radio transmission or reading in sensor data. The installed sensor types depend on the field of application of the WSN, e.g., for agricultural WSNs temperature, humidity and light sensors would make sense, whereas in the field of smart cities sensors measuring dust particle concentration or CO2 levels would be reasonable.

Due to hardware limitations the nodes of a WSN are constrained by different internal factors. Usually the power supply of a node is covered by a battery which has a limited life time, generating different hindrances in the network. One is that the computional power of a sensor node is a constraint, as power consumption of a chip is a function of the operating frequency and voltage. It also introduces a restraint for the radio interface, as every radio transmssion, especially long range transmissions, consumes power as well. The radio interface and transmission protocol, on the other hand, introduce another constraint, since the bandwidth as well as the transmission range is limited, only a specific amount of data can be processed. Further, in order to propagate information through the network the sensor nodes have to be in range of each other. Though, not only internal factors limit the usage of the sensor nodes, they are also affected by external factors. Especially when installed outdoors nodes have to be resistant against vast environmental influences, e.g. changes in temperature and high humidity. Moreover, it has to be considered that despite all efforts some nodes may fail. The WSN has to continue operating even though a subset of the nodes has failed. The kind of deployment has to be considered since for large WSNs it might not be feasible to install every node individually. One method, especially used for environmental monitoring, is the deployment by plane. When not every node has an uplink some kind of data propagation and collection has to be implemented.

Regarding the constructing of a WSN for a specific application all of these factors have to be considered and thus different challenges have to be faced in order to increase the life time of the network.

This chapter is structured as follows: In Section 2.1 different communication standards and protocols are presented. Section 2.2 deals with different topologies of WSNs and how they can affect the network performance. Section 2.3 discusses the processing of the data collected by the WSN. Different methods for the localization of nodes in a WSN are presented in Section 2.4, Section 2.5, deals with target tracking via WSNs.

2.1 Communication Technologies

As most aspects of a WSN the communication protocol and hardware depends on field of application. Since different applications have different requirements for e.g., latency or connection security. The most common communication technology for WSNs are radio transmitters operating in the ISM bands, i.e. the *Industrial, Scientific and Medical* (ISM) radio bands. These radio frequency bands were originally internationally reservered for the use of *Radio Frequency* (RF) electromagnetic fields for industrial, scientific and medical purposes other than communications [1].

For the field of WSNs the standard IEEE 802.15.4 was introduced by the IEEE 802.15.4 Working Group for data communication devices operating in *Low Rate Wirless Personal Area Networks* (LR-WPANs). The standard is especially designed for short-range, low-power, low-data-rate and low-cost wireless sensor communication [4]. It targets wirless sensor applications which use short ranged communication in order to maximize their battery lifetime. The standard only defines first two layers, the *Physical* (PHY) and *Medium Access Control* (MAC) layers, based on the well known OSI model. The rest of the upper layers of the protocol stack are defined by other architectures, e.g., *ZigBee*. Representative for the most common WSN communication standards and technologies *ZigBee* [5] and *Bluetooth Low Energy* [6] are here examplary presented.

ZigBee operates in frequency ranges between 868 MHz and 915 MHz or at 2.4 GHz. It has a maximum data rate of 250 kBps, with a radio range of 100 m. Due to its low power consumption the battery life time of the operating device can hold from days up to years (depending on the capacity of the battery and how frequently communication is taking place). Its main contribution is to provide an IPv6-based wireless mesh networking (see *Mesh Topology* in 2.2) solution. It enriches the IEEE 802.15.4 standard by adding network and security layers as well as an application framework which allows IPv6 networking. The *ZigBee* communication standard is mostly targeted for SmartMeters or SmartGrid devices.

Bluetooth is a wireless technology for devices with a short-range and cheap hardware. It was itended to replace the cables in *Wireless Personal Area Networks* (WPAN), i.e. for the communication between two devices, e.g., a smart phone and a *Bluetooth* speaker. From *Bluetooth*, *Bluetooth Low Energy* (BLE) evolved. BLE as well as *Bluetooth* operates at a frequency of 2.4 GHz in the ISM band, though BLE uses a doubled channel width of 2 MHz and thus the half of the number channels *Bluetooth* does. It enables low-power communication and an enhanced radio range of up to 200 m with a data rate of 1 MBps. BLE is suitable for devices which operate only on coin-cell batteries for month or even years.

2.2 Network Topologies

There are different topologies for WSNs. The topology depends, again, on the application domain of the network. It has an effect on different aspects of WSN, e.g. the throughput capacity of the sensor nodes can be effected as well as the routing abilities. The three most typical topolgies are briefly summarized below.

The most simple topology for WSNs is a *Star Topology* which is illustrated in Figure 2.1a. Here the (normal) sensor nodes, n_i in the figure, communicate only with a central node or data fusion center, m in the figure. This central node receives and processes the data from the sensor nodes. A typical WSN with a *Star Topology* would be a *Body Area Network* where sensors are located all over the body collecting data and transmitting it to a main device. For example a motion sensor in a fitness bracelet which transmits all its data to the smart phone of a user where it is centrally processed. A *Star Topology* is preferred when the area to be covered is small and/or a low latency is required by the WSN application.

In an *Peer-to-Peer Topology* only the direct communication between two sensor nodes is supported. The *Peer-to-Peer Topology* topology can be seen in Figure 2.1b, it is suitable for networks which need a large coverage area and where latency is not a critical issue for the WSN application.

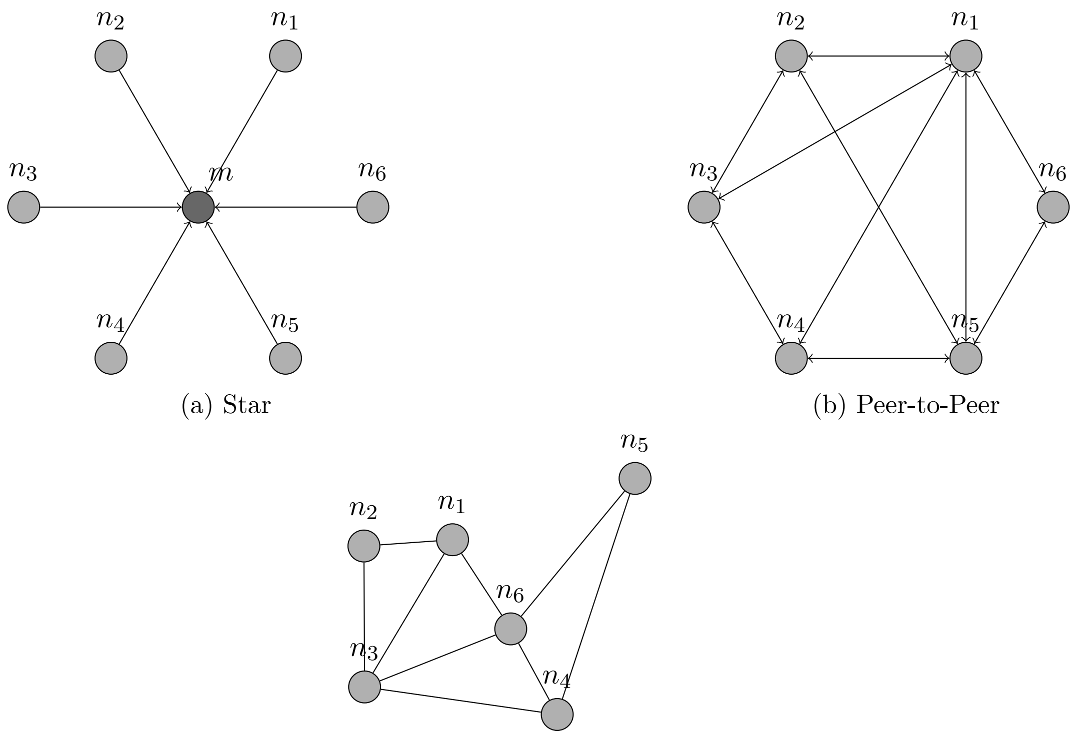
Another topology for a WSNs would be a *Mesh Topology*, here every node can communicate with its neighbour nodes, that are the nodes which are within the range of the radio signal of the sending node. The *Mesh Topology* allows efficient and dynamic routing through the sensor nodes depending on the routing algorithm and whether the location of every sensor nodes is known. Therefore *Mesh Topologies* are quite reliable as nodes failures can be easily compensated through a new routing path in the network (as long as sufficient nodes are remaining active). An example for a *Mesh Topology* can be seen in Figure 2.1c.

As mentioned before, the kind of network topology interferes with the communication protocol and has to be considered already in the early design stages. Not every communication protocol supports every network topology. For example the *Bluetooth* and *Bluetooth Low Energy* communication protocol only allow a *Peer-to-Peer* topology of the network, whereas the *ZigBee* communication protocol supports a *Peer-to-Peer, Mesh* and *Star Topology* of the network.

The network topology also has an impact on the throughput capacity, in [7] different topogies of WSNs with and without obstacles are compared. It was found that for networks without obstacles an uniform distribution with regularly distributed helping nodes have some advantages in improving the throughput capacity of the network. When dealing with a network with obstacles the communication and, thus, the throughput capacity of the network is constrained, the overall throughput capacity is bounded by the bottlenecks that are created near the obstacles. In such a case the optimal distribution of the sensor nodes in the network is found by an algorithm which was proposed in [7]. The algorithm divides the area with respect to the obstacles into areas for which an optimal solution at the bottle necks is found.

2.3 Data Processing

In order to use the data gathered by the sensor nodes, some kind of data processing is needed. Redundant or invalid data of the sensor nodes has to be eleminated and some kind of data compression might be useful. Additionally, when not every sensor node is equipped with an uplink the data has to be communicated through the WSN to a central data fusion point which collects all of the data collected by the sensor nodes. For example WSNs which are deployed by plane, e.g. for environmental monitoring, this can be done by flying over the deployment area and collecting the data by communication with the sensor nodes. Since the number of the sensor nodes may lay in the hundreds or thousands



(a) Star

(b) Peer-to-Peer

(c) Mesh

Figure 2.1: Topologies

a data fusion is necessary, as it is not feasible to evaluate the data gathered by the every sensor node individually.

2.3.1 Multisensor Data Fusion

Multisensor Data Fusion descibes the technique of combining/fusing the data of individual sensor nodes in order to enhance the overall information content. This is done naturally by humans and animals, where all of the senses smell, touch, vision and hearing are combined in order to improve the overall sensing abilities. Especially vivid is the example of the human vision, the 2D sensor data of the two sensors, i.e. the eyes, is combined in order archieve a 3D representation of the environment.

The application domain of multisensor data fusion is widespread reaching from military applications like automated target recognition to civil applications like monitoring manufacturing processes. Techniques for multisensor data fusion are drawn from a wide range of areas including artificial intelligence, pattern recognition, statistical estimation and other areas. [8]

Multisensor data fusion brings mutliple benefits. The first and most basic one is that it enables a human to analyse the gathered data by reducing the total amount of data to a minimum. Another benefit is of statistical nature, measurement inaccuracies and thus the variance of the gathered data is reduced when combining data from the same kind of source, e.g., the average over the temperature data gathered by all of the sensor node might yield a better representation of the system temperature. When combining the sensor data of multiple types of sensors, the amount of information can be increased drastically. Imagine a sensor which only allows to determine the position of an object and another sensor which only allows to determine the speed of an object. Combining the data gathered by both sensors leads to a more accurate description of the object, i.e. its current position as well as its velocity.

Therefore multisensor data fusion has to be considered especially for large WSNs where the data of individual sensor nodes is not managable.

2.4 Localization Methods

In serveral applications of WSNs it is important, that a single sensor node knows its own position, e.g. for target tracking where the communicating sensor nodes needs to know its own position in the WSN grid in order to communicate the position of the target in the grid correctly or for an efficient routing of data through the network.

It might not be the preferred solution to install a global propositioning module in all of the nodes of the WSN in order to determine the position of each node, for serveral reasons: *limited power* - when the battery capacity is limited, providing power for a GPS module in the sensor node might not be possible; *size* - if the size of the sensor node is constrained there might not be enough space; *inaccassibility* - the GPS signal might be obstructed, e.g. if the sensor nodes are deployed indoors; *imprecsion* - GPS has a confidential interval of 8 m; *cost* - for large networks it might just be to expensive [9, 10]. Therefore different approaches on the localization of sensor nodes in a WSN are taken.

2.4.1 Distance Measurements

Most localization methods rely on distance measurements between sensor nodes. There are serveral options for measuring the distance between two radio stations. Commonly used methods include *Time-of-Arrival*, *Time-Difference-of-Arrival* and measurment of signal strength [11].

Time-of-Arrival (TOA) or sometimes called *Time-of-Flight* (TOF) describes the method calculating the distance between two radio stations directly using the time of arrival. When the velocity of the signal is known and the clocks of the two radio stations are synchornized the travel time of the signal can be used to determine the distance. This method is for example used by GPS.

When the signal strenght at the sender is known the signal strength at the receiver can be used to estimate the distance between both.

2.4.2 Multilateration

Multilateration (MLAT) is a navigation technique to determine the position of an object based on distance measurements between the object and at least three anchor points. As for a *relative localization* in a plane at least three points are needed to determine the basis vectors which then can be used for spanning the plane, i.e. given p_0, p_1, p_2 the basis is $B = \{v_1, v_2\}$ with $v_i = \frac{p_i - p_0}{\|p_i - p_0\|_2}$. For an *absolute localization* the absolute positions of at least three anchor points a_0, a_1, a_2 are needed. Using distance measures to the initial/anchor points the positions of the remaining points can be calculated. This situation is illustrated in Figure 2.2.

Two different algorithms that can be used for MLAT, which are also disscussed in [9], are briefly summarzied below and additionally examplary compared in Section 3.1. The first

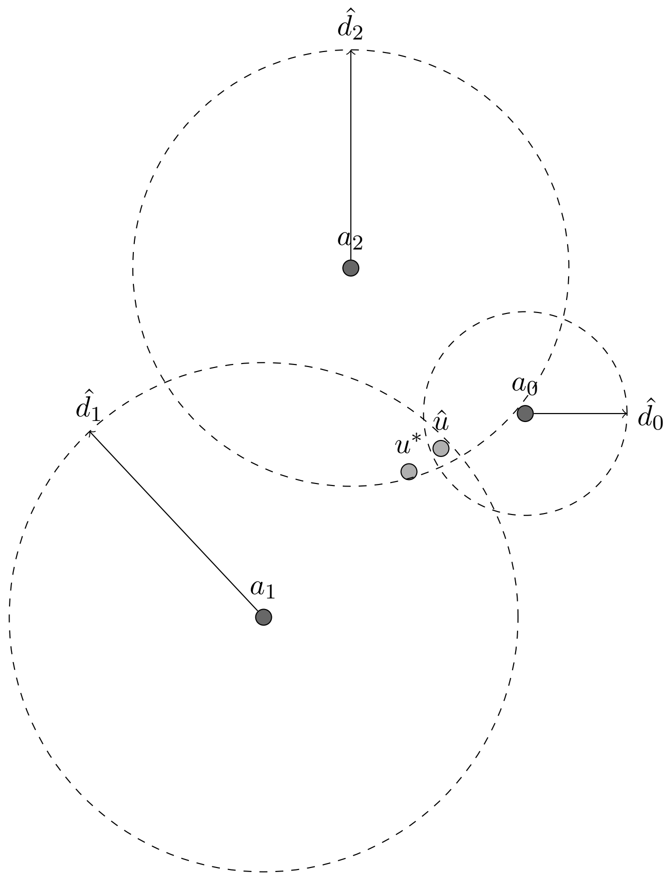


Figure 2.2: Multilateration scenario with 3 anchor points.

is the *Gauss-Newton-Algorithm* (GNA). The second algorithm is based on the *Gradient-Descent-Method* (GDM) which was modified in order to increase the speed of convergence.

As in [9] the notation for the sensor node position is $\mathbf{u}^* = [x^*, y^*]$ and the notation for the estimated sensor node position is denoted by $\hat{\mathbf{u}} = [\hat{x}, \hat{y}]$. Further, the positions of the n anchor nodes are given by $\mathbf{a_i} = [x_i, y_i]$. The multilateration can be calculated in the weigthed least-squares sense by:

$$\hat{\mathbf{u}} = \underset{\mathbf{u}=[x,y]}{argmin} \left\{ \sum_{i=1}^{n} w_i^2 f_i^2(\mathbf{u}) \right\} \tag{2.1}$$

where

$$f_i(\mathbf{u}) = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \hat{d}_i \tag{2.2}$$

and $\mathbf{w} = [w_1, ..., w_n]$ denote the reliability weights associated with the measured distances, though here neglected, i.e. $w_i = 1, \forall i = 1, ..., n$.

Gauss-Newton-Algorithm

The *Gauss-Newton-Algorithm* is an iterative optimisation algorithm, which can be described as:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{\Delta}^{T(k)} \tag{2.3}$$

with the increment $\mathbf{\Delta}^{(k)}$ given as:

$$\mathbf{\Delta}^{(k)} = -\left(\mathbf{J}^{T(k)}\mathbf{W}\mathbf{J}^{(k)}\right)^{-1}\mathbf{J}^{T(k)}\mathbf{W}\mathbf{f}^{(k)} \tag{2.4}$$

where $\mathbf{f}^{(k)} = \left[f_1(\mathbf{u}^{(k)}), ..., f_n(\mathbf{u}^{(k)})\right]^T$ with $f_i(\mathbf{u}^{(k)})$ as defined in 2.2. $\mathbf{J}^{(k)}$ is the Jacobian matrix of $\mathbf{f}^{(k)}$, i.e.:

$$\mathbf{J}_{i1} = \frac{\partial f_i}{\partial x} = \frac{x - x_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \tag{2.5}$$

and

$$\mathbf{J}_{i2} = \frac{\partial f_i}{\partial y} = \frac{y - y_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \tag{2.6}$$

$\mathbf{W}$ is the weight matrix, i.e. $\mathbf{W} = \mathbf{diag}(w_1, ..., w_n)$.

The stop criterion for algorithm can be determined by a maxmimum number of iteration, though here the result may be unprecise, or by a threshold τ for the increment, i.e. $\left\|\mathbf{\Delta}^{(k)}\right\|_2 < \tau$.

Gradient-Descent-Method

As the *Gradient-Descent-Method* is an iterative algorithm as well, where each iteration can also be described like in equation 2.3. Although here the increment $\mathbf{\Delta}^{(k)}$ follows the negative gradient of the direction of 2.1, i.e.:

$$\mathbf{\Delta}^{(k)} = -\alpha\mathbf{J}^{T(k)}\mathbf{W}\mathbf{f}^{(k)} \tag{2.7}$$

with the step size α. Which was proposed in [9] as:

$$\alpha = \frac{\eta}{\sum\limits_{i=1}^{n} w_i} \tag{2.8}$$

where η is an optimized constant which was found to be $\eta = 1.5$ regardless of the number of anchor nodes.

2.5 Targeting Tracking

One main application domain of WSNs is target tracking. It deals with the localization of targets in an area using a WSN.

There are different possibilities and methods for target tracking. In general two cases can be distingushed: The first cases is to use the data sensed by the nodes in order to determine the location of a predefined target. The second is that the target itself is a (moving) sensor node, which shall be localized.

Especially in military application WSN are used for detection of military targets. One example is *PinPtr* an ad-hoc counter-sniper system which was developed in order to localize shooters in a urban terrain [12]. It uses highly redundant acoustic sensing by wireless sensor nodes with known positions in order to determine not only the shooter's location but also the trajectory of the bullet. *PinPtr* is a good example for the complexity of a WSN as it relies on mesh networking, data fusion as well as an ad-hoc localization of the nodes. But also in civil application target tracking is used, for example in environmental monitoring. *ZebraNet* is tracking system developted to track animal migrations [13]. Here special collars equipped with a low-power GPS system are used to monitor the movement of Zebras. In [14] an approach on detecting and localizing targets by a network of *Unmannedd Arial Vehicals* (UAV)s and *Unmanned Ground Vehicles* (UGV)s is implemented.

3 Localization Methods

3.1 Multilateration GNA and GDM Comparison

In the following the two methods GNA and GDM are here examplary compared by number of iterations needed to reach a guven minimum error rate, for different counts of anchor nodes. In both cases the reliabilites weights have been neglected, i.e. $w_i = 1 \forall i = 1, ..., n$.

Figure 3.1 shows the results for the GNA. Figure 3.2 the results for the GDM. Both algorithms were evaluated for different configurations, that is a different number anchor nodes. Further, each configuration was repeated for a set of 2000 random cases. All distance measurements were annotated with a random measurement error of $\pm$ 10 %. In both figures the relative error, i.e. the distance error compared to maximal possible distribution of the nodes, is plotted over the number of iteration the respective algorithm was run.

As can be seen the GNA method achieves a better result than GDM regardless of the configuration, i.e. the number of anchor nodes. Though it is has to be kept in mind that even though the number of iterations is the same, the computational complexity of the algorithms differ as can be seen from equations 2.4 and 3.2. Two more matrix multiplications have to be computed for every iteration. Though, the inversion can be neglected as it is computationally fast for $2x2$ matrices. This has to be considered if one wants to implement these method in wireless sensor node with constraint computational power. Though, both algorithms are suitable for an implentation in a sensor node of a WSN.

3.2 Ad Hoc Positioning System

As mentioned earlier it can be insufficient to have a GPS module installed in every sensor node of the WSN. Further the radio ranges the sensor nodes are limited, e.g. due to limited battery capacities. If not every node of the network can communicate with an anchor node and, thus, cannot measure the distance to it, a different approach has to be taken in order to detemine the location of the sensor node. In [10] an *Ad Hoc Position System* of sensor nodes is proposed, which deals with this scenario. The algorithm proposed is similar to *Distance Vector* (DV) routing.

The basic idea is to view the WSN as a graph $\mathbf{G} = \{V = N \cup A, E, W\}$ with the set of vertices V consisting of the set of (normal) nodes N and the set of anchor nodes A, i.e. $A \cap N = \emptyset$. E denotes the set of edges between these vertices and W the set of the weights of the edges, i.e. the distances between nodes. If the distance between all (anchor) nodes are known then the *plane* topology can be reconstructed, that is the relative position of all nodes to each other. Further, the absolute position of every node $v_i \in V$ be can determined if for some (anchor) nodes $a_j \in A$ the absolute position is known. The scenario is illustrated in Figure 3.3.

Representative for all of the different propagation methods the first and most basic one the so called *Distance-Vector-Hop* (DVH) propagation method is examplary described

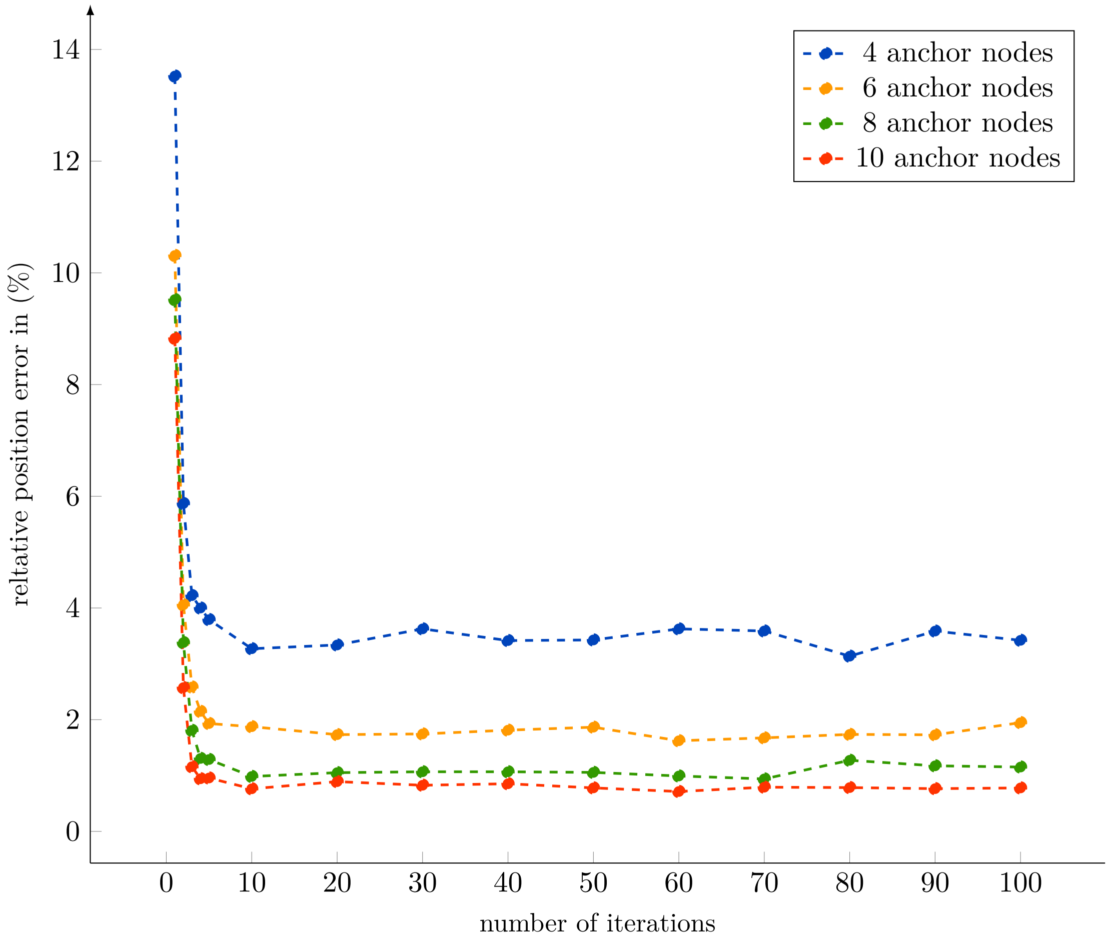


Figure 3.1: GNA results

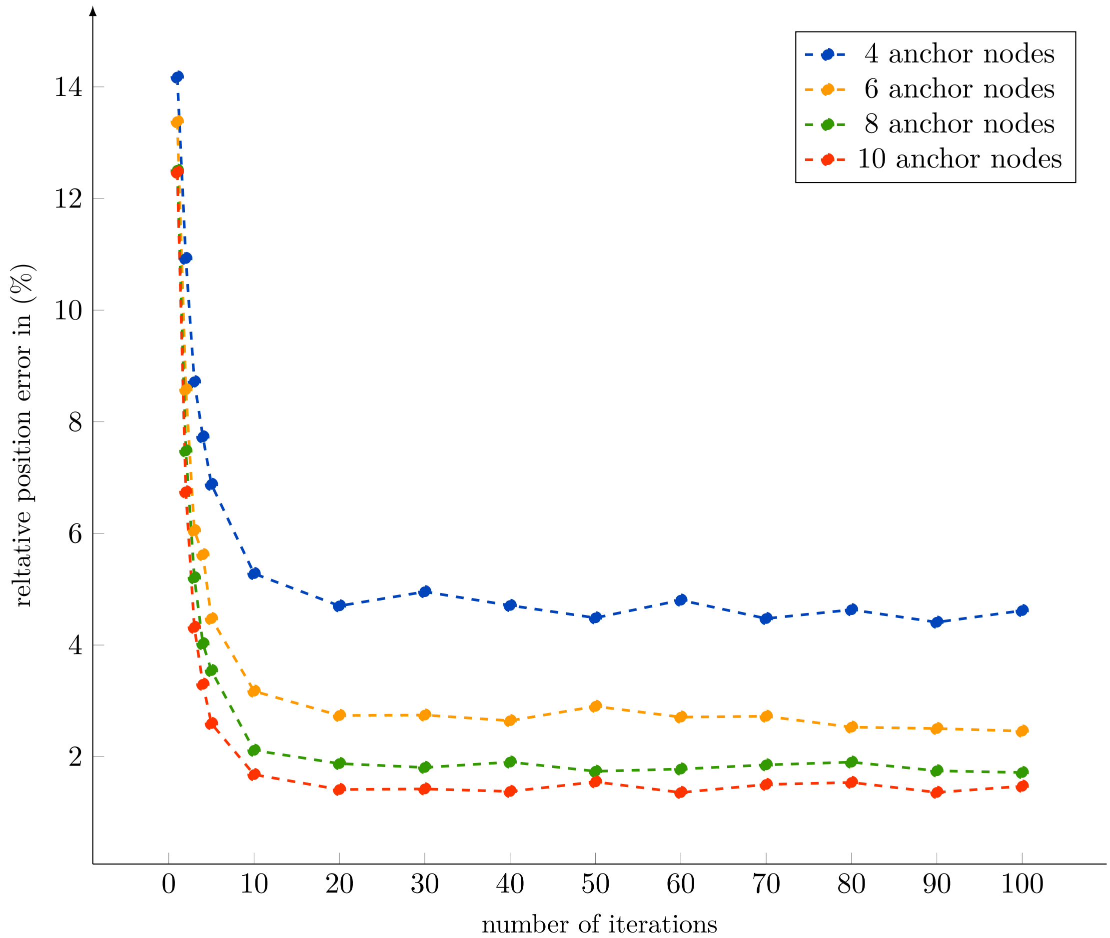


Figure 3.2: GDM results

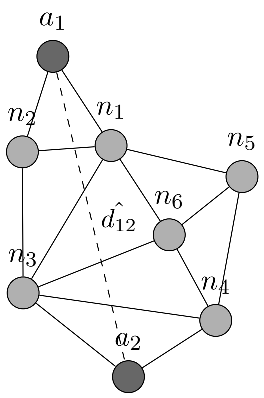


Figure 3.3: APS scenario with 2 anchor nodes.

and evaluated. Though, the algorithm was improved here at the expanse of the number of radio transmissions in order to achieve a more accurate estimation of the node position. Especially for an anisotropic distribution the estimation error could be reduced by a factor two compared to [10].

The DVH propagation method employs a distance vector exchange so that all nodes $n_i \in N$ get the absolute position of every anchor node $a_j \in A$ and the respective distance to a_j in hops. The goal is to estimated the average size of a hop c as precisely as possible. Using the estimated hopsize c in combination with the absolute position of every anchor node a_j and the respective distance in hops h_j a node can determine its own position using e.g. GDM or GNA.

Note that for the DVH propagation method no direct measurement of the distances between single nodes is necessary. It relies solely on the number of hops between the nodes and the absolute positions of the anchor nodes. Therefore, is this method not affected by measurement errors that might occur when measuring the distances between nodes using the radio interface. Further, no implementation for such a method like TOA, is needed for the nodes.

In the following the procedures performed by the nodes in order to implement the DVH are described. The goal state is that every node $n_i \in N$ knows the absolute position of every anchor node $a_j \in A$, the number of hops to a_j as well as the estimated hopsize c. For that each node n_i maintains a table with entries $e_j = \{X_j, Y_j, h_j, d_j\}$ for every anchor node a_j where $[X_j, Y_j]$ is the position of a_j, h_j the distance from the node n_i to a_j in hopes and d_j the estimated distance between n_i and a_j using the estimated hopsize, i.e. $d_j = h_j * c$. Similiarly every anchor has a table for all other anchor nodes. Information in the WSN is propagated using two different kinds of data packets: The first data packet contains a table entry for an anchor node a_j, further referred to as P_{e_j}. The second data packet is hold the hopsize c_j estimated by an anchor node a_j, further reffered to as $P_{c_{j_l}}$, with l denoting the lth hopsize packet of a_j.

The algorithm works as follows: Initially every anchor node broadcasts its position as P_{e_j} into the nework. The procedure for a node receiving a packet with a table entry is listed in Algorithm 1. Once a node receives a packet P_{e_j} it checks if it has matching table entry e_j, if not it updates its table and broadcasts the packet with an updated hop count P_{e_j} to the network. If a node receives a redundant packet, i.e. a node has an entry e_j and receives P_{e_j}, it drops the packet.

Algorithm 1 Node receive P_{e_j}

Input: New Packet P_{e_j}, Last Correction c_{k_m}, Landmark Table $\tau = \{e_1, ..., e_n\}$

$e_j \equiv \{X_j, Y_j, h_j, d_j\} \leftarrow \mathsf{get\_entry}(P_{e_j}, c_{k_m})$
if $e_j \in \tau$ **then**
  $\mathsf{drop\_packet}(P_{e_j})$ ▷ Drop redundant packet.
  return
end if
$\tau \leftarrow \mathsf{add\_entry}(\tau, e_j)$ ▷ Update table
$e_j \leftarrow \{X_j, Y_j, h_j + 1, d_j\}$ ▷ Update entry that is to be broadcasted.
$P_{e_j} \leftarrow \mathsf{get\_packet}(e_j)$
$\mathsf{broadcast}(P_{e_j})$
return

When an anchor node a_k receives a packet P_{e_j}, i.e. $k \neq j$, it first acts in the same way as a normal node does (by forwarding it), but additionally it calculates an update of its estimated hopsize c_{k_l}:

$$c_{k_{l+1}} = \frac{\sum_j \sqrt{(X_k - X_j)^2 + (Y_k - Y_j)^2}}{\sum_j h_j}, \quad k \neq j \tag{3.1}$$

The new estimated hopsize $c_{k_{l+1}}$ packed into $P_{c_{j_{l+1}}}$ is then broadcasted by the anchor node to the network. Instead of propagating only one correction through the network like it is described in [10], an anchor node updates and broadcasts its correction for every new packet P_{e_j} it receives. Thus, resulting in a more accurate estimated hopsize and distance for every node. The procedure for a node receiving a packet with a correction is listed in Algorithm 2. A nodes accepts and forwards a packet $P_{c_{j_l}}$ only if two conditions hold:

The first is it has to be a packet from the anchor node a_k it has received an initial estimated hopsize from, i.e. $k = j$. This guarantees that a node only accepts estimations of the hopsize from the closest anchor node. Thus, it ensures that the local topology has a bigger influence.

The Second condition is that the index l of the packet has to be greater than the index m of the last packet $P_{c_{jm}}$ the node has received, i.e. $m < l$. This condition prevents the propagation of redundant packets in the network and therefore enables controlled flooding of the information through the network.

The node can then use the received estimated hopsize c_{j_m} to determine the estimated distances d_j to all anchor nodes, i.e. $d_j = c_{j_m} * h_j$. The estimated distances to the anchor nodes can than be used to estimate the position of the node using e.g. the methods described in [9].

A simulation of a WSN for different configurations was done in order to evaluate the results of the modified algorithm. Two different scenarios are covered:

The first scenario, illustrated in Figure 3.4 is an isotropic distribution of the nodes in the WSN, i.e. the nodes are distributed randomly around fix points in the scenario map.

The second scenario is an anisotropic distribution of the nodes in the WSN, i.e. the nodes are distributed randomly (by an uniform distribution) all over the scenario map.

The simulation results can be seen in Figure 3.6. The configuration of the scenarios is taken from [10]. It defines a total of 100 nodes, with a node degree of 7.6, i.e. the

Algorithm 2 Node receive $P_{c_{jl}}$

Input: New Packet $P_{c_{jl}}$, Last Correction c_{k_m}, Landmark Table $\tau = \{e_1, ..., e_k\}$

if $c_{k_m} = None$ **then**
  $c_{k_m} \leftarrow c_{jl}$ ▷ Initialize last correction to first received.
end if
if $j \neq k$ **or** $l \leq m$ **then**
  drop_packet($P_{c_{jl}}$) ▷ Drop invalid or redundant packet.
  return
end if
$c_{k_m} \leftarrow c_{jl}$
for $e_j \equiv \{X_j, Y_j, h_j, d_j\} \in \tau$ **do** ▷ Update distances
  $d_j = c_{k_m} * h_j$
end for
update_position(τ) ▷ Calculate position using e.g. GDM or GNA.
broadcast($P_{c_{jl}}$)
return

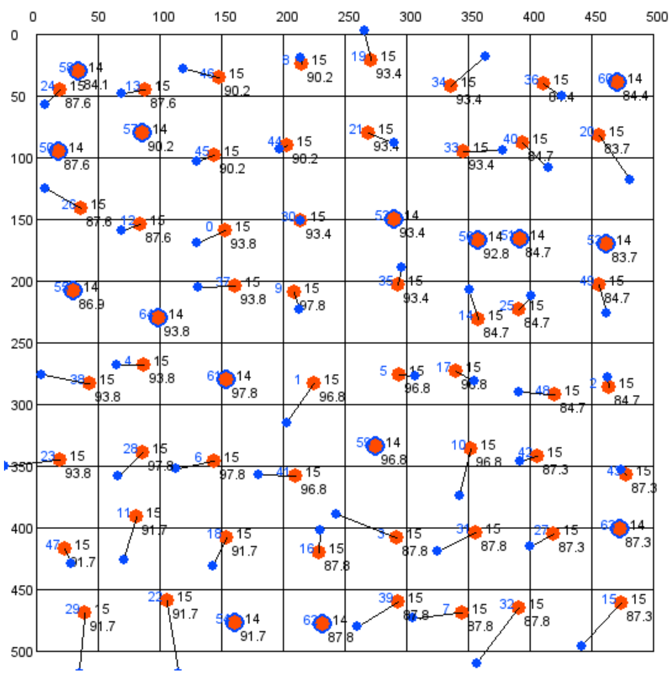


Figure 3.4: APS simulation with iostropic node distribution.

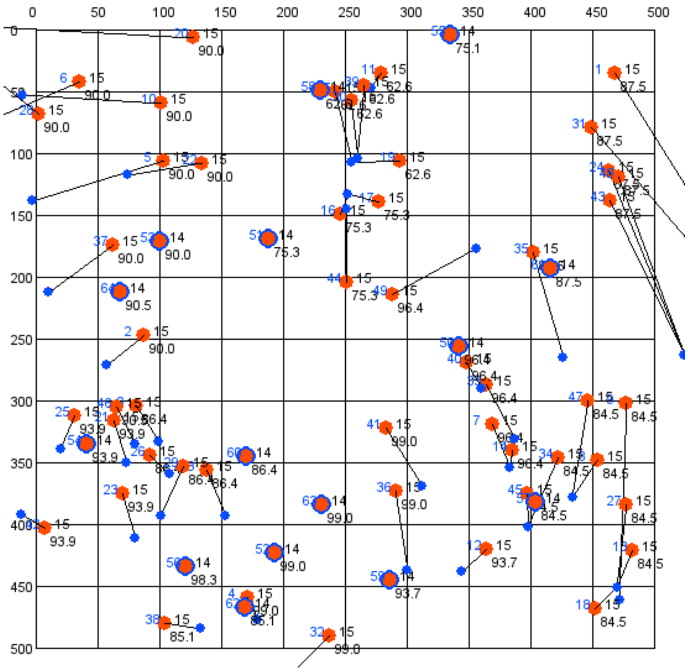


Figure 3.5: APS simulation with uniform node distribution.

number of nodes a single node has a connection to. The estimation error is given in percent relative to the radio radio range of a single node. It is assumed that the radio ranges of all nodes are the same and are circular. Further, the simulation is event based, i.e. nodes are scheduling actions like broadcasts at discrete timing points in the future. This allows a pseudo parallel execution of the different procedures. Every node which is in radio range of a transmission receives the data transmitted without any losses or time delay. Processing of the received data packets is again scheduled at a discrete timing point in the future.

It can be seen that for more the isotropic scenario the error can be reduced to about 25 %. For the anisotropic the average estimation error is significantly higher with an error of at least 50 %. The results show how big the impact of the distribution is. With an increasing variance of the hop size the estimation results get more inaccurate.

In conclusion the simulation results show that even with such a simple algorithm it is possible to achieve a quite accurate position estimation. Further, as this is the most basic procedure it is also the most inaccurate one. The results from [10] show that more accurate results can be achieved if the algorithm is expanded and the *Distance-Vector-Distance* or *Euclidean* propagation methods are used.

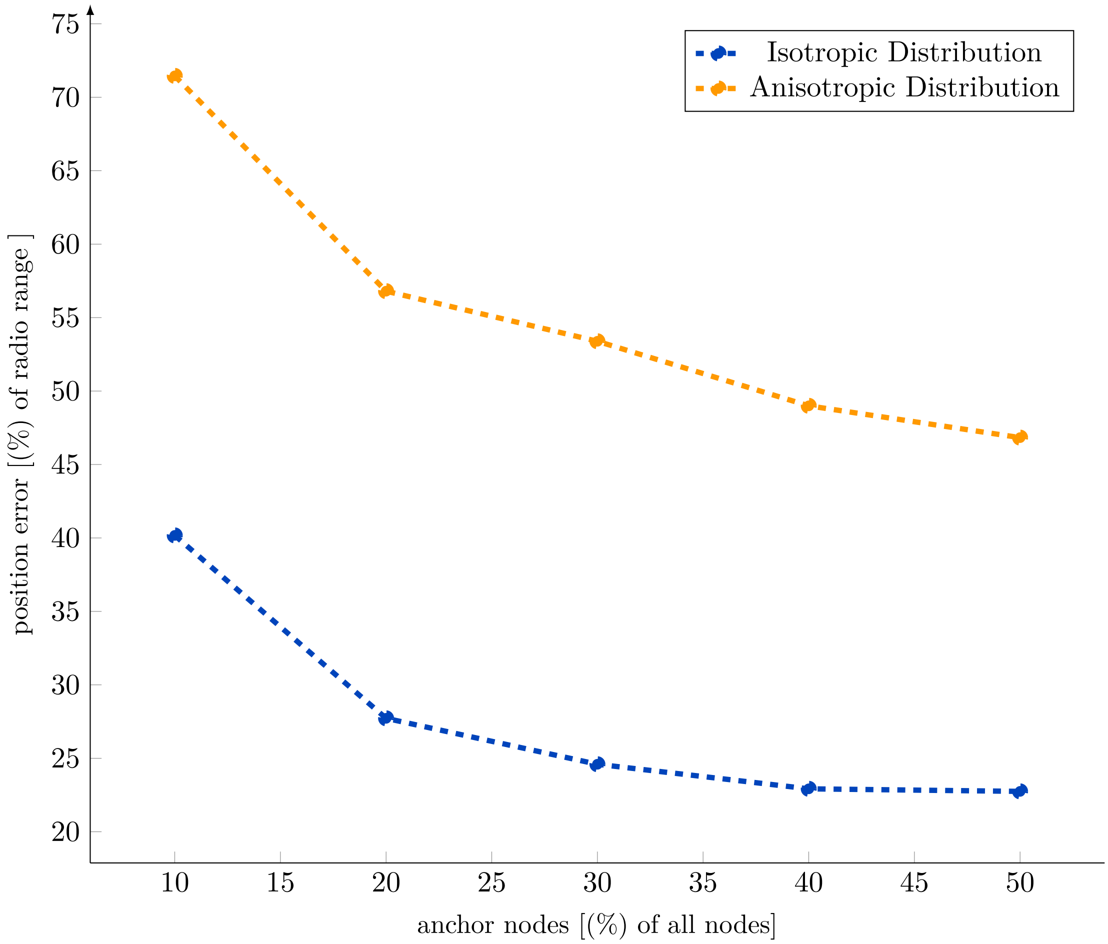


Figure 3.6: APS simulation results

4 Conclusion

This report is only a brief peek into the field of WSNs and methods used to localize sensor nodes in such a network. The, in this report briefly discussed, topics include communication technologies, network topologies, as well as data processing. For the topic of localization methods two examples from resarch are taken, and presented and evaluated.

For the case that not all nodes in a WSN are equipped with a global positioning module alternatives are presented. For that a brief introduction into the *Multilaterion* problem is given and two algorithms which solve the problem are presented, namely the *Gauss-Newton-Iteration* and the *Gradient-Descent-Method*. Further the results presented in [9] are verified as well as analysed.

A software approach for a MLAT based location estimation, namely the *Ad-Hoc Positioning System*, is given. It can be seen that MLAT soley based on a distance estimation using exchanged data packets already achieves quite accurate results. For the most basic APS implementation, the *DV-Hop* method, the location estimation error can be reduced to about 35 % on average relative to the radio range of a single node. The results show that even without specialized hardware accurate location estimations are possible.

4.1 Outlook

Location estimation is not only a topic in WSNs, therefore synergies between WSNs and other existing and emerging technologies exist [1]. The location methods discussed in this report can give a base line for further exploration and research not only in the field of WSNs. A verification of the other methods used by the APS which are presented in [10] and [11] needs to be done. Future research could address an improvement of the APS in order to reduce the location error as well as the data traffic in the network.

Algorithms

1 Node receive P_{e_j} 19
2 Node receive $P_{c_{j_l}}$ 20

Literature

[1] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68(1):1–48, 2014.

[2] ATMEL. ATMEL ATMega128, ATMEL ATMega128L Summary. [Online] `http://www.atmel.com/Images/2467S.pdf` (accessed 2017/01).

[3] ARM. ARM Cortex-M3 Technical Reference Manual. [Online] `http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337e/DDI0337E_cortex_m3_r1p1_trm.pdf` (accessed 2017/01).

[4] IEEE 802.15.4 Working Group. Standard for Information Technology Part 15.4: Wirless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), 2006.

[5] ZigBee Alliance. ZigBee. [Online] =www.zigbee.org.

[6] Bluetooth Low Energy. Bluetooth Low Energy. [Online] `www.bluetooth.com/Pages/Low-Energy.aspx`.

[7] Sen Yang, X. Wang, and Luoyi Fu. On the topology of wireless sensor networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 2095–2103, March 2012.

[8] D. L. Hall and J. Llinas. An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE*, 85(1):6–23, Jan 1997.

[9] C. Müller, D. I. Alves, B. F. Uchôa-Filho, R. Machado, L. L. de Oliveira, and J. B. S. Martins. Improved solution for node location multilateration algorithms in wireless sensor networks. *Electronics Letters*, 52(13):1179–1181, 2016.

[10] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 5, pages 2926–2931 vol.5, 2001.

[11] D. Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 3, pages 1734–1743 vol.3, March 2003.

[12] M. Maroti, G. Simon, A. Ledeczi, and J. Sztipanovits. Shooter localization in urban terrain. *Computer*, 37(8):60–61, Aug 2004.

[13] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. Hardware design experiences in zebranet. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 227–238, New York, NY, USA, 2004. ACM.

[14] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *IEEE Robotics Automation Magazine*, 13(3):16–25, Sept 2006.