ABSTRACT
Wireless Sensor Networks (WSNs) are among others used to sense the environment. Applications supported by a WSN very often require among others accurate timing as well as the position of each WSN node. This paper presents a survey of different positioning algorithms used in wireless sensor networks. Also a comparison of these solutions is provided, using a set of WSN characteristics.

Keywords
Wireless sensor networks, location, positioning algorithms, sensor nodes, ad hoc networks

1. INTRODUCTION
Wireless Sensor Networks (WSN) have become an important and growing area, because small, low power sensors with embedded processors and radios have become available [EGPS01]. A wireless sensor network (WSN) is a collection of devices equipped with a processor, having sensing and communication capabilities and being able to organize themselves into a network created in an ad-hoc manner. Wireless Sensor Networks can be used in a large variety of applications and are currently used in military, environment, health, commercial and home applications.

This paper presents a survey of different positioning algorithms used in wireless sensor networks and at the same time it compares these algorithms by using a set of WSN characteristics.

The second section of this paper explains what wireless sensor networks are, their challenges, and the methods for estimating the position of a node. The third section lists commonly used and recently proposed positioning algorithms, which are compared in a later section. In the fourth section positioning algorithms used in other wireless networks are analyzed to find key characteristic of wireless sensor networks. The criteria to compare the various methods are listed in section 5. In section 6 the listed positioning algorithms used in wireless sensor networks are compared based on the characteristics listed in section 5. The last section contains the conclusions and gives current and future recommendations for using positioning algorithms in different environments.

2. WIRELESS SENSOR NETWORKS
Wireless Sensor Networks (WSNs) consist of a large number of sensor nodes deployed in a wireless environment. A sensor node is a device equipped with processor and communication capabilities, which could have several sensors attached to monitor the environment.

There are several application areas where WSN are deployed. In military application it can be used for e.g. monitoring friendly forces, equipment and ammunition, battlefield surveillance. Also health-care can benefit from WSN by monitoring patients at home or in hospitals, which triggers and event when the state of the patient changes. In environmental applications WSN are used for animal tracking, monitoring conditions of crops and livestock, etc.

Figure 1: Wireless Sensor Network Topology [SRB01]
A sensor node needs to use the received and/or stored information in order to compute its position. In order to retrieve useful information each sensor has to know its (relative) position and the accurate current time, see [RASP+00]. Most of the wireless sensors do not know their position and have to calculate their relative position using information obtained from other sensors in the network.

After gathering sensor input a location can be computed using various algorithms. Every algorithm has its own (dis) advantages. The characteristics of a WSN decide which method is most efficient in a particular setup. The current state of the art of getting the position of a sensor in WSN is captured in this paper. The most used algorithms will be analyzed and compared using a set of WSN characteristics.

2.1 Challenges
A WSN can consist of a large number of nodes. It is (almost) impossible to configure all node positions manually and therefore the position of nodes in such a network has to be calculated by the nodes themselves.

After a WSN is deployed no maintenance is applied to the nodes. A node is useless after it is out of energy, so nodes have to be energy efficient to expand the lifetime of the node.

The environment of the wireless sensor network is unknown by the nodes and can contain obstacles such as walls and rocks. This can lead to inaccurate topological range information and/or the topological fragmentation of the network.

2.2 Sensing Position
Many applications make use of the Received Signal Strength (RSS) of the radio frequency transmission, which is a cheap, but not very accurate method. Other methods, such as Time of Arrival, Time Difference of Arrival and Angle of Arrival can also be used as input method. These methods have their own strong- and weaknesses.
2.2.1 Time of Arrival
A commonly used technique for obtaining range information via signal propagation time is Time of Arrival. Time is synchronized between all nodes. Distance estimations can be made by monitoring the arrival times of the signals that are sent from other nodes. This technique is also used in GPS systems, which requires expensive hardware and is not energy efficient [HHB+03].

2.2.2 Time Difference of Arrival
Another technique is the Time Difference of Arrival (TDOA). TDOA determining a node’s position by comparing and calculating the difference in time required for a node’s signal to reach different nodes. The difference in the times of arrival of signals at two separate radio antennas can be used to estimate the direction of arrival of the signal. Multiple measurements can also be used to compute the distances to other nodes. “Like TOA technology, TDOA also relies on extensive hardware that is expensive and energy consuming, making it less suitable for low-power sensor network devices”, from [HHB+03].

2.2.3 Received Signal Strength
Received Signal Strength Indicator (RSSI) is a technique using theoretical or empirical models for translating signal strength into distance estimates. This technique does not require any expensive hardware, but range estimates can be inaccurate, which is caused by background interference, irregular-signal propagation and multi-path fading [DUL05].

2.2.4 Angle of Arrival
Angle of Arrival is another technique to estimate distances between nodes using relative angles between neighbors. The angles between nodes in combination with a single distance estimation can be used to estimate the distances with other nodes. Similar to TOA and TDOA, AOA requires additional expensive hardware.

Most techniques above need additional expensive hardware, which makes them unsuitable for wireless sensor networks. RSSI is a commonly used technique, which does not need expensive hardware, but in some cases leads to inaccurate range estimations.

2.3 Computing Position
Nodes in WSN need to compute their position based on their own and other node’s sensors location. The calculation of the position of a node can be done using multilateration, trilateration and triangulations, which are mathematical functions using the received signal strength, time (difference) of arrival and/or angle information as input.

A positioning algorithm generally computes the position in a two-phase approach. In the first phase an initial position is determined by all nodes. In the refinement phase, which is an incremental phase, the found position will be corrected using input from other nodes. The refinement phase will lead to more accurate node positions.

The location of a node can be computed by a central unit or in a distributed manner [ONLB05]. A central unit generates more accurate positions, because all the available information gathered from nodes can be used in the computation. A central unit needs more processing power and thus more energy capacity than regular nodes. Distributed methods are more robust, because such a method will work even when the network is split in multiple segments. Also distributed methods are more scalable, because they distribute the task over the total network.

In the rest of the section the three methods of computing a position are analyzed in more detail. The last two presented methods need at least three nodes with known positions, where as triangulation can execute using information from at least two nodes. The accuracy increases when more nodes are available, because the errors in the measurements are averaged out.

2.3.1 Triangulation
Triangulation [ONLB05] is based on directional communication in stead of distance information and therefore it uses Angle of Arrival as input method. Besides the angles between nodes this algorithm requires to know at least one distance to another node (Figure 2).

![Figure 2: Triangulation [ONLB05]](image)

2.3.2 Trilateration
Trilateration [ONLB05] uses TOA to estimate the distance between known nodes. Each known node broadcasts their position and time, which can be illustrated as three circles. The intersection of three circles is the estimated position of the node (Figure 3).

![Figure 3: Trilateration [ONLB05]](image)

2.3.3 Multilateration
Multilateration [ONLB05] positions a node by computing the time difference of arrival of a signal emitted by three or more nodes. This method minimizes the differences between measured distances and estimated distances. Multilateration is more accurate than triangulation, because it is easier to measure the time accurately than it is to form a very narrow beam required by triangulation (Figure 4).

![Figure 4: Multilateration [ONLB05]](image)

3. POSITIONING ALGORITHMS IN WSN
Positioning algorithms can be categorized in centralized, distributed and localized.
Using a centralized algorithm the positioning information is sent by all sensor nodes to a central point where the computations are performed. After the calculations are performed the central point sends the position back to each node. The disadvantages of this method are huge communication costs and a single point of failure in the network. Also the central point requires a lot of computational power.

A distributed algorithm eliminates the problem of a single point of failure by distributing the task of calculating the positions to several nodes, which calculate it for the nodes around them.

Nodes in a localized system calculate their position themselves using only local data sensed from nodes nearby.

Centralized systems are the most accurate, because have access to all information before calculating the locations of nodes, but require a lot of resources and rely on a single point of failure. Localized systems are more scalable, than centralized or distributed, because each node calculates its own position on the available information, but it leads to less accurate positions.

Another categorization in positioning algorithms can be made: range-free and range-based. Range-free localization uses absolute point-to-point estimates, where range-based uses angle estimations for calculating the position of a node. Range-free localization is preferred in WSN, because range-based systems require expensive hardware elements [HHB+03].

### 3.1 Convex Optimization

Convex optimization is a centralized method using linear programming (LP) and semi definite programming (SDP). A linear program optimizes for a linear objective over a set of constraints [DPG01]. A proximity constraint exists when one node can communicate with another. The constraints restrict the feasible set of unknown node positions. These constraints can be modeled in a linear program (LP) what has the following form:

\[
\begin{align*}
\text{Minimize} & \quad c^T x, \\
\text{Subject to} & \quad Ax < b,
\end{align*}
\]

A semi definite program (SDP) is a generalization of an LP and has the following form:

\[
\begin{align*}
\text{Minimize} & \quad c^T x, \\
\text{Subject to} & \quad F(x) = F_0 + x_1 F_1 + \ldots + x_n F_n < 0, \\
& \quad Ax < b, \\
& \quad F_i = F_i^T
\end{align*}
\]

"The first inequality represents a linear matrix inequality (LMI) on the cone of positive semi definite matrices, i.e. the eigenvalues of \( F(x) \) are constrained to be non positive," from [DPG01]. Efficient computational methods for convex programming problems are available. A quadratic inequality is transformed into an LMI for a two-dimensional problem. These LMI can be stacked to form one large SDP.

Transmitters can vary the output power so that the radius of the signal determines an upper bound constraint for each connection (Figure 5). Errors in distance estimations arise when two nodes can not communicate because of physical barriers. The angle of arrival can be used to estimate a distance between nodes. This information can be added as a constraint to the node. Introducing multiple types of constraints makes the possible size of the region smaller and the position estimate becomes more accurate, but requires more computational power. According to the simulations of [DPG01], convex optimizations perform well when more nodes with known positions are added.

The convex optimization provides very accurate node locations, but requires a lot of computational power for the central node. The position calculation in larger networks is more difficult because the complexity of the computations is increased quadratically or more [DPG01]. A solution for this is splitting up the network in several segments, but a trade-off (accuracy vs. scalability) has to be made.

### 3.2 Multi Dimensional Scaling

Multidimensional Scaling (MDS) determines the position of a node using already available information such as distances and connection constraints [JZ04]. MDS generates relative maps that can be made absolute when 3 or more anchor nodes are available. "MDS captures the inter-correlation of high dimensional data at low dimension in social science, to compute the local maps of adjacent sensors with high error-tolerance", from [JZ04]. The relative maps are pieced together to get the approximation of the physical position and to form a full map of the sensor network. The distance between two nodes \((i \text{ and } j)\) is estimated using the function given in Figure 5.

\[
d_{ij}(X) = \left( \sum_{a=1}^{m} (x_{i,a} - x_{j,a})^2 \right)^{1/2}.
\]

Figure 5: MDS estimation function [JZ04].

It starts with a rough estimate and is made more accurate by minimizing the function, given in Figure 6, in several iterations.

\[
\sigma(X) = \sum_{i<j} w_{ij} (d_{ij}(X) - \delta_{ij})^2.
\]

Figure 6: MDS minimize function [JZ04].

The minimum is reached when the gradient is equal zero. It uses a quadratic function and therefore scales better than the convex optimization method.

### 3.3 Lighthouse System

The Lighthouse System [RM03] is a centralized system, which uses a parallel beam rotating at a constant speed to detect the nodes in its environment. The base station or lighthouse has to be in visible range with all the nodes (Figure 7).

![Figure 7: Lighthouse System Principles [RM04]](image-url)
Each node has to be equipped with a photo detector and a clock in order to send the lighthouse the time the node received the signal, so it can compute its position. The amount of time it takes for a node to receive the signal from the lighthouse determines the distance between the node and the lighthouse.

### 3.4 Ad Hoc Positioning System

The Ad Hoc Positioning System is a hybrid of two methods: distance vector routing and beacon based positioning.

Distance Vector (DV) routing is used for forwarding information hop by hop from each anchor in the network. Each node will calculate its position based on the position of anchors and distance estimates. In DV routing, each node keeps track of a routing table, which contains the distances from itself to all possible destinations. Each entry consists of the distance to a node, the address to communicate with it (like an IP address) and the next node in the path. A node broadcasts this information to its neighbors to increase the accuracy of the estimated position and to update its routing table with changes and estimation updates [PK01].

It can make use of several sensing methods, such as RSSI and AOA [NN03]. Its objectives are to broadcast discovery packets only when necessary, detecting neighbors, general topology maintenance and to disseminate information about changes in local connectivity to neighboring nodes that are interested in this information. Node A in Figure 8 is only directly connected with one beacon. It can calculate its position when surrounding nodes start behaving like anchor nodes.

A beacon based positioning system, such as GPS, contains anchor nodes that send their position to other nodes. The other nodes can compute its position using triangulation or trilateration. Nodes that calculated their position start behaving like anchor nodes by sending their position to neighbor nodes. A node, with no direct connected anchors, can compute their position by using nodes that started to act like an anchor node. This way less anchor nodes are needed which makes this system more scalable.

### 3.5 Sequential Monte Carlo Approach

The Sequential Monte Carlo Localization (MCL) method provides simulation based solutions to estimate the posterior distribution of nonlinear discrete time dynamic models [HE04]. The method updates a set of weighted samples recursively and distributes them to its neighbors. Re-sampling techniques are used to eliminate the normalized weights with small importance. In the first step a prediction of node positions is used to eliminate the normalized weights with small importance. In the first step a prediction of node positions is made. In the second stage the algorithm reduces the solution space by filtering the estimates. This method works well with moving nodes and is range-free. The MCL method has been successfully applied in robot localization, for which it was developed. At startup nodes positions are unknown, except for the anchor nodes. The algorithm works as follows: In the initialization phase a node has no knowledge of its location. An iterative phase updates the estimation by setting Lt based on Lt-1 from the previous step and new observations, ot (Figure 9).

\[
L_t = \begin{cases} 
|L_t| & \text{while (size (L_t) < N)} \\
R = \{U'_i \mid i \text{ is selected from } p(U'_i | L_{t-1}), i \in L_{t-1} \text{ for all } 1 \leq i \leq N \} \\
R_{filtered} = \{U'_i \mid i \in R \text{ and } p(U_i', o_t) > 0 \} \\
L_t = \text{choose (} L_{t-1} \cup R_{filtered} \text{) }
\end{cases}
\]

### Figure 9: MCL localization algorithm [HE04]

By applying this method a trade off has to be made between the costs of communications and deployment and the accuracy of the estimated locations. Using mobility of nodes accuracy can be improved and costs reduced [HE04].

### 4. POSITIONING ALGORITHMS IN OTHER NETWORKS

In order to better understand positioning algorithms suitable for WSN, positioning algorithms used in other types of networks will be analyzed. These methods however share the way a position is calculated by using multilateration, trilateration or triangulation, but cannot be applied in WSN because of scalability concerns of the method and the need for an external infrastructure and/or the high level of energy consumption.

#### 4.1 GPS

The Global Positioning System uses satellites which transmit radio signals. The infrastructure of this system requires 24 satellites orbiting in space to cover the whole earth, so that each receiver can ‘see’ at least four signals. It calculates the position using multilateration. The accuracy of a L1 GPS receiver is about 10-20 meter. Using a form of differential correction, like the Wide Area Augmentation System (WAAS) does, the accuracy can be increased to 3-5 meter [EWP+04].

#### 4.2 GSM

The Global System for Mobile Communications (GSM) can also determine a nodes position by using multilateration and propagation time, TOA, TDOA and/or carrier phase measurements [DMS98]. A GSM node does not calculate a position itself, but the network logs information about the cell locations of GSM devices.

#### 4.3 WLAN

The authors of [HLIN01] present a local positioning system based on WLAN. The RSSI values provided by the WLAN adapter are calculated into distances using a propagation model. The Extended Kalman Filter [HFLR+04] is used to reduce inaccuracies in the perceived RSSI values and maintains the state estimate.

The Access Points have a known position and are used as anchors by the triangulation method. Such systems can be used to track PDA's and laptops in an office building using the existing WLAN technology. While it uses other hardware than WSN, the used method is very similar to WSN positioning algorithms. The tested setup makes use of PDA's computational power and WLAN technology, so energy consumption was not an issue for this research.

#### 4.4 BlueTooth

The authors of [HNS03] propose a positioning method for BlueTooth devices. It is based on RSSI and uses triangulation to refine the position found in the first run. BlueTooth can communicate with several devices at once, which makes...
gathering position information of other nodes easier and effective. Also the proposed system can cope with moving targets. The drawback of a BlueTooth based positioning system is that it requires relatively a lot of power, which makes it unsuitable for WSN.

5. CRITERIA
The goal of the comparison is to find which WSN positioning solution is most suitable for WSN scalable up to very large networks. These networks must be easily deployable, as cheap as possible and have an acceptable lifetime and performance. To find out what positioning algorithms perform best in certain circumstances a set of criteria will be used. This section lists and explains the criteria.

5.1 Energy consumption
The amount of power consumed by a certain solution is of great importance for the lifetime of the system. The power consumption of a node depends on the power consumption of the sensor, processor, and communication capabilities. Using additional hardware mostly leads to higher energy consumption. The rating for power consumption is: less is better.

5.2 Self-organizing
The amount of nodes in a WSN can be quite large. The amount of used anchor nodes is an important factor in comparing the different solutions, because the position needs to be set for anchor nodes. The less anchor nodes a solution needs for calculating nodes positions with an acceptable accuracy the better its rated.

5.3 Scalability
Another important issue is the scalability of the algorithm. The number of nodes used in a WSN may be hundreds, thousands and even millions depending on the application. An algorithm for such application should be able to scale when the number of nodes increases and still perform well. This criterion also looks if an algorithm can cope with moving nodes and the high node density a WSN can have.

5.4 Robustness
Not only is the lifetime of a node dependent on the amount of power consumption, but a node also has to cope with environmental (weather) conditions. Nodes can go defect during a (rough) employment phase. The positioning algorithm must be able to cope with malfunctioning nodes and a changing topology. The better a solution can cope with these changes in its environment the higher rating it gets.

5.5 Performance
The positioning algorithms will also be compared based on the performance of a WSN. This will look at how many data can be collected within a certain time. Can the solution also keep the performance up while the network topology changes?

5.6 Completeness
Another issue is the completeness of the solution. Is it just a theory or is it already applied in practice? This criterion rates a solution based on its maturity and usability. How fast can the solution be used in practice?

5.7 Need for modifications in standards
Some solutions require modifications in existing protocols to make it work, while others can be deployed using existing standards. The less modifications a solution needs the better rating a solution gets.

5.8 Complexity
The last criterion is the complexity of system, which looks at the additional functions that have to be supported by the nodes. Nodes with additional functionalities are harder and more expensive to produce. This is important because the amount of nodes can be very large.

6. Comparison
In this section the location algorithms are compared based on the criteria defined in section 5.

6.1 Convex Optimization
By applying the convex optimization algorithm an optimal solution can be calculated, because of the amount of information available. The level of robustness is quite high also because of the amount of information available. The drawback of this method is that the computations increase quadratic which makes this method not energy efficient. The computations are done centrally and therefore the solution is not very scalable.

6.2 Multi Dimensional Scaling
The MDS method is an iterative algorithm which increases the accuracy of the estimation for each run, but consumes more energy. According to Ji and Zha [JZ04] Multi Dimensional Scaling can be used to create a robust solution which can deal with complex terrain and anisotropic network topology. The solution is very effective and efficient and even with a few anchor nodes available according to the authors. It supports on demand positioning estimations.

6.3 Lighthouse System
The lighthouse system was one of the first projects in this research area. This method depends on direct communication with nodes, which makes it difficult to deploy such network in hostile environments. A single lighthouse can see many nodes, depending on the range of the lighthouse. The system can use several input methods, which reduce the cost for applications where accuracy is not the primary issue.

6.4 Ad Hoc Positioning System
The Ad Hoc Positioning System is a distributed method where makes it more scalable than central methods such as the lighthouse and convex optimization. It can also make use of multiple input methods to increase performance and reduce costs when this is not a high priority. Using multiple input methods increase energy consumption because the computation is more complex. Deployment of such a system is difficult because the anchor nodes need to be placed uniform in the network to get acceptable accuracies.

6.5 Sequential Monte Carlo Approach
The Sequential Monte Carlo approach has proven its use already in robot localization. It can cope with a low anchor density, irregular network transmissions and with hardware limitations. The solution supports movement of nodes, which can actually increase the accuracy of estimations, because it can
measure the distances between nodes from different angles to refine the node positions.

6.6 Summary
Table 1 presents the algorithms with rankings assigned to the different criteria in order to get a good overview of the algorithms. Each criterion gets a rating of good, fair or bad, where good is highest, fair is medium and bad is bad. When no information is available to rank a criterion, no rank is given. The final score indicates which solution is the best one for a large wireless sensor network that is robust, self-organizing, energy efficient and has an acceptable performance.

Table 1: Comparison

<table>
<thead>
<tr>
<th></th>
<th>CO</th>
<th>MDS</th>
<th>LH</th>
<th>AHP</th>
<th>MCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Consumption</td>
<td>B</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Self-Organizing</td>
<td>F</td>
<td>G</td>
<td>B</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Scalability</td>
<td>F</td>
<td>G</td>
<td>B</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Robustness</td>
<td>G</td>
<td>G</td>
<td>B</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>Performance</td>
<td>G</td>
<td>G</td>
<td>-</td>
<td>G</td>
<td>-</td>
</tr>
<tr>
<td>Completeness</td>
<td>G</td>
<td>F</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Needs for modifications in Standards</td>
<td>-</td>
<td>F</td>
<td>G</td>
<td>-</td>
<td>G</td>
</tr>
<tr>
<td>Complexity</td>
<td>G</td>
<td>F</td>
<td>G</td>
<td>G</td>
<td>F</td>
</tr>
</tbody>
</table>

The Ad Hoc Positioning System has the best ranking (table 1). This does not mean it is the best solution for every wireless sensor network. The best option for a wireless sensor network depends on the importance of the criteria. Optionally a weight can be assigned to the criteria to get a better total ranking. The Ad Hoc Positioning System is ranked as best because it is self-organizing and very scalable.

7. CONCLUSIONS AND FUTURE WORK
Locating sensor nodes in a Wireless Sensor Network is a difficult task because of the uncertainty of the received information. In this paper several positioning algorithms have been analyzed and compared using the characteristic of WSN.

The accuracy of the computed position of a node depends on the received data from the sensor. Many applications use the Received Signal Strength of Radio Frequency to compute a position, which is cheap but not very accurate. Other sensing methods can be used to increase accuracy, but require additional hardware which makes the solution more expensive.

Energy consumption is probably the most important factor in most wireless sensor networks, because it determines the lifetime of the network. The amount of computations and communication transmissions needed to estimate all nodes positions determines the amount of energy needed. Generally a trade-off between energy consumption and accuracy has to be made.

The accomplished comparison of the various positioning algorithms has been based on a literature study. Future studies based on e.g., simulation experiments could give a better view on the quantitative comparison between the different positioning algorithms.

ACKNOWLEDGMENTS
We thank the Association of Computing Machinery for making available a proceedings template [ACM] that we could use for the TC&CR Conference.

Also I want to thank Georgios Karagiannis for his valuable feedback and also thanks to Jeroen Logtenberg, Teake Blom and Luke Niesink for valuable peer feedback in the process.

REFERENCES


[DUL05] S.O. Dulman, "Data centric architecture for wireless sensor networks", Ipskamp PrintPartners, Enschede, 2005


international conference on Mobile computing and networking, ACM Press, September 2004


