Chapter XIV
Evaluation of Localization Algorithms

Michael Allen
Coventry University, UK

Sebnem Baydere
Yeditepe University, Turkey

Elena Gaura
Coventry University, UK

Gurhan Kucuk
Yeditepe University, Turkey

ABSTRACT

This chapter introduces a methodological approach to the evaluation of localization algorithms. The chapter contains a discussion of evaluation criteria and performance metrics followed by statistical/empirical simulation models and parameters that affect the performance of the algorithms and hence their assessment. Two contrasting localization studies are presented and compared with reference to the evaluation criteria discussed throughout the chapter. The chapter concludes with a localization algorithm development cycle overview: from simulation to real deployment. The authors argue that algorithms should be simulated, emulated (on test beds or with empirical data sets) and subsequently implemented in hardware, in a realistic Wireless Sensor Network (WSN) deployment environment, as a complete test of their performance. It is hypothesised that establishing a common development and evaluation cycle for localization algorithms among researchers will lead to more realistic results and viable comparisons.

INTRODUCTION

Evaluating the relative performance of localization algorithms is important for researchers, either when validating a new algorithm against the previous state of the art, or when choosing existing algorithms which best fit the requirements of a given WSN application. However, there is a lack of unification in the WSN field in terms of localization algorithm evaluation and comparison. In addition, no standard methodology exists to take an algorithm through modelling, simulation and emulation stages, and into real deployment. As a result it can be hard to quantify exactly *how* and under what circumstances one algorithm is better than another. Moreover, deciding what performance criteria localization algorithms are to be compared or evaluated against is important for the success of the resulting implementation given that different applications will have differing needs.

Since localization algorithms are expected to be used in real applications, it is not conclusive to verify their performance in simulation only. The authors here argue that algorithms should be emulated (on test beds or with empirical data sets) and subsequently implemented in hardware, in a realistic WSN deployment environment, as a complete test of their performance.

In this chapter, performance evaluation metrics are discussed alongside three criteria – localization accuracy, cost, and coverage. Given that WSNs are typically constrained in terms of node/network lifetime and per-node computational resources, addressing these constraints leads to trade-offs in the performance of localization algorithms. For example, if maximising localization accuracy is the foremost priority, specific hardware may have to be added to each sensor node, increasing node size, cost and weight. Conversely, if the hardware available is already determined, then the application expectations with respect to performance criteria (such as accuracy) must be adjusted accordingly.

The chapter is structured as follows: a discussion of the various performance criteria and evaluation metrics that are readily used in the analysis of localization algorithms is first presented. Next, representative topologies that affect performance criteria are given, followed by simulation models and parameters that affect the performance of localization algorithms. A case study is presented, outlining an acoustic monitoring sensor network with high accuracy constraints enforced by application requirements. This case study is contrasted with an example where scalability and longer network lifetime are required at the expense of complexity and localization accuracy. Finally, the chapter closes with a brief discussion on the development cycle of a localization algorithm, from simulation to real deployment.

It should be noted that although this chapter makes particular emphasis on simulation and comparison of range-based localization algorithms, many of the metrics and techniques described are applicable to other approaches, such as Angle of Arrival (AoA) based algorithms, for example.

EVALUATION CRITERIA

Whilst the intuitive measure of the performance of a localization algorithm may be to show how well it can estimate positions of nodes compared to the known ground truth (to the degree of accuracy required by the WSN application, as discussed further below), localization algorithms are also subject to the general constraints of wireless networked sensing. It follows that a broader set of evaluation criteria for localization algorithms are needed (and are useful to both developers and users of localization algorithms), examples of which are accuracy, cost, coverage, robustness and scalability. These criteria reflect the constraints already mentioned - computational limitations, power constraints, unit cost and network scalability.

Some evaluation criteria are binary in nature: algorithms either have a specific property or they do not (for example, they are self-configuring or not; they are anchor free or not). Classifications and binary criteria can be used by researchers to narrow the set of existing algorithms to evaluate against, or to choose from. For example, one may only consider distributed, anchor-free, range based localization algorithms, immediately limiting the number of algorithms to compare to. Some evaluation criteria and trade-offs however, need quantification and qualification. These are described below in more detail, and questions are posed that might be useful to the algorithm or WSN application designer in establishing a given algorithm's performance.

Scalability

Can the localization algorithm scale from less than ten nodes to hundreds, or even thousands? Moreover, is it necessary from the WSN application standpoint for the algorithm to hold this property?

A centralised localization algorithm will typically aggregate all input data at a central, more capable sink to carry out processing; this represents a single point of error, and potential bottleneck for network communication. In contrast, a distributed localization algorithm's execution is shared throughout the network with no reliance on a central sink. However, centralised algorithms are conceptually simple and easier to implement in cases where it is known that the network will be small and will not increase. By comparison, distributed algorithms are harder to develop and deploy, but may be advantageous for researchers if the network does not have a simple logical topology (i.e. a tree of nodes sending data to a sink), and will need to support a large number of nodes (tens to hundreds). Theoretically, scalability is an important general consideration; however in actual deployment for specific applications this is not necessarily an overriding one (primarily due to the relatively small numbers of nodes that will be deployed and the amount of effort it takes to deploy them).

Accuracy

How well do the positions estimated by the algorithm match the known, ground truth positions? How well has the WSN application been specified in terms of its minimal localization accuracy needs?

One may think that positional accuracy compared to ground truth is the over-riding goal of a good localization algorithm. On reflection, this is largely application-dependent - different WSN applications will have different requirements on the resolution of the accuracy. Consider a tracking application – the estimated positions of nodes in the network directly affect the accuracy of the tracking. The granularity of the required accuracy may be a ratio of the inter-node spacing. For example, if the average node spacing is 100m, up to 1m error may be acceptable. However, if the average node spacing is 0.5 m, the same error level is clearly unacceptable.

Resilience to Error and Noise

How well can the localization algorithm deal with errors and noise in the input data?

It is important to understand how well the localization algorithm will perform without an accurate or full set of input data. Some algorithms, for example classical multi-dimensional scaling, used by Shang et al (2003) assume measurements from every node to every other for the localization algorithm to converge, which is an overbearing assumption given the realities of most deployment environments.

Evaluation should show how measurement noise, bias or uncorrelated error in the input data affects the algorithm's performance, and also establish the number of nodes that can actually be localized. Errors in measurement are particularly important to consider when adapting a localization algorithm that assume 2D to work for 3D applications (a common assumption in the research community). For example, a simple multilateration computation in 3D is far more sensitive to noisy/inaccurate measurements than its 2D counterpart, due to the extra degree of freedom (the Z axis). Convergence in 3D may then result in flips and reflections of the estimated coordinate, as observed by Allen et al. (2006) and shown in Figure 1.

Coverage

How much of the network can be localized by the algorithm, given a specific network topology/deployment?

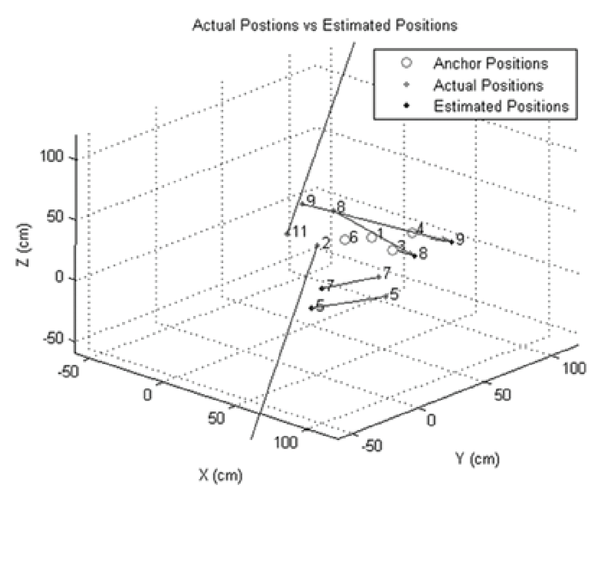
Some algorithms may have problems localizing the whole network if nodes do not have enough neighbours ("enough" is specific to the details of the algorithm) in terms of connectivity or distance constraints/estimates. Coverage may relate to the physical network density, i.e. one may be more likely to get 100% localization coverage in a densely deployed network. In addition, it is worth considering how easy it is to add another node to the network after the initial localization algorithm has completed.

Cost

How expensive is the algorithm in terms of power consumption, time taken to localize a node, communication and pre-deployment set-up (i.e. need for, and number of anchors)?

There are several parameters which one could classify as "individual costs", such as per-node hardware or software cost, power consumption required to complete node localization, time taken to converge on a network wide localization solution, and amount of communication required (messages/data transmitted). An algorithm which can minimise several cost constraints is likely to be desirable if maximising network lifetime is a primary deployment goal. For example, an algorithm may focus on

Figure 1. Using multilateration to estimate unknown positions in 3D with noisy range measurements



minimising communication and complex processing to achieve quick convergence, but at the expense of the overall accuracy.

Discussion

Clearly, the perfect localization algorithm would provide suitably accurate results (relative to the scale requirements of the application), in a simple and decentralised way, with low communication and processing overhead, whilst allowing incremental addition of nodes, and requiring zero anchor nodes. However, all existing and possibly future algorithms will most likely have to trade these criteria off against one another. For example, is it better to increase memory footprint and processing time for accuracy, or use a simpler algorithm and reduce positional estimation accuracy and time taken? Deployment practice and expertise indicate that trade-offs are best resolved when intimately related to the specifics of the class of applications that the particular WSN is deployed to address. The quantitative measurements required to understand these trade-offs are described in the next section.

LOCALIZATION ALGORITHM EVALUATION, COMPARISON AND METRICS

In order to address quantitatively how well a localization algorithm might perform against the criteria described in the previous section, a set of metrics are available. This section breaks down accuracy, cost and coverage and describes well known metrics or common measures used in their evaluation.

Accuracy Metrics

The basic goal of the localization accuracy metric is to show how well matched the ground truth and estimated positions are. Accuracy is likely to be related to measurement noise, bias, accuracy and precision in the input data provided to the localization algorithm. The accuracy metrics described below are separated into those which use ground truth as comparison, and those which do not. Throughout this section, it is assumed that a WSN is composed of n sensor nodes deployed over a given area.

Metrics with Ground Truth

Globally, the positions determined by a localization algorithm represent a geometrical layout of the physical positions of the sensors. This layout must be compared to the ground truth, or known layout of the sensors. It is important therefore that not only the error between the estimated and real position of each node is minimised, but also that the geometric layout determined by the algorithm matches well the original geometric layout.

Mean Absolute Error

The simplest way to describe localization performance is to determine the residual error between the estimated and actual node positions for every node in the network, sum them and average the result. Broxton et al (2006) do this using the mean absolute error metric (MAE), which, for each of n nodes in the network, calculates the residual between the node's estimated $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ and actual (x_i, y_i, z_i) coordinates. This is shown in (1):

$$MAE = \frac{\sum_{i=1}^{n} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}}{n} \tag{1}$$

The resulting metric represents the average positional error in the network, aggregating individual residual errors into one statistic. The MAE computation has much similarity to root mean square (RMS) error, a commonly used calculation to measure the difference (or residual) between predicted and observed values. Slijepcevic et al (2002) also note that whilst knowing the mean absolute error is important in some cases, it is also beneficial to know the maximum error exhibited in the position estimation, as shown in (2).

$$MAX\_ERROR = \max_{i=1..N} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{2}$$

FROB

A slightly different approach is taken by Efrat et al (2006) who use the FROB (Frobenius) metric. In this case, the residual error between all *n* nodes in the network is calculated. It is assumed that the estimated and actual inter-node distances have already been determined. The Frobenius metric is shown in (3), where $\hat{d}_{ij}$ and *dij* are the estimated and ground truth distances respectively and *n* is the number of nodes in the network.

$$FROB = \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\hat{d}_{ij} - d_{ij})^2} \tag{3}$$

FROB essentially determines the RMS of the total residual error, which represents the global quality of the localization algorithm.

GER and GDE

As discussed briefly at the start of this section, it is important for the accuracy metric to reflect not only the positional error in terms of distance, but also in terms of the geometry of the network localization result. If only average node position error is used, there is no sense of the *correctness* of the relative geometry of the network – it is entirely possible that for a given localization result the average error metric is low, but the actual layout created by the algorithm does not match well the physical layout of the network. This problem was identified by Priyantha et al (2003), and addressed by defining the Global Energy Ratio (GER) metric, shown in (4).

$$GER = \frac{1}{n(n-1)/2} \sqrt{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{\hat{d}_{ij} - d_{ij}}{d_{ij}} \right)^2} \tag{4}$$

The distance error between nodes $(\hat{d}_{ij} - d_{ij})$ is normalised by the known distance between the two nodes (d_{ij}), making the error a percentage of the known distance. (One should notice the similarity between GER and FROB – note that FROB does not normalise the distances, and takes the RMS). Ahmed et al (2005) note that the GER metric does not exactly reflect RMS error. They address this by defining an accuracy metric which better reflects the RMS error calculation, called Global Distance Error (GDE), shown in (5).

$$GDE = \frac{1}{R}\sqrt{\frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(\frac{\hat{d}_{ij} - d_{ij}}{d_{ij}}\right)^2}{n(n-1)/2}} \tag{5}$$

GDE takes the RMS error over the network of *n* nodes and normalises it using the constant *R*. In Ahmed et al's context, *R* represents average radio range, meaning the localization results are represented as a percentage of the average distance nodes can communicate over.

ARD

Gotsman and Koren (2005) derive another quality metric called Average Relative Deviation (ARD), shown in (6). ARD is simply the normalised average of the estimate, rather than the RMS error.

$$ARD = \frac{2}{n(n-1)}\sum_{i<j}\frac{\left|\hat{d}_{ij} - d_{ij}\right|}{\min(\hat{d}_{ij}, d_{ij})} \tag{6}$$

The individual distances are normalised in this case by the shorter of the two distances (either the estimated or ground truth), which may not always be the known distance (as in GDE and GER).

BAR

The BAR metric by Efrat et al (2006) is a measure of how well the estimated positions of nodes that sit on the boundary of the localized network match the actual positions. It is in essence the sum-of-squares normalised error taken from matching the estimated boundary with the actual boundary. This metric may be useful as an alternative to GER, in cases where the topology formed does not seem to match well the actual topology, even though the distance error metrics indicate it should. In this case, the average error is not helpful, as the metric can be diluted by high error variance across the network. BAR is used as the minimisation metric for the Iterative Closest Points (ICP) algorithm which matches estimated and actual boundary points. A BAR metric is computed for each iteration of the ICP algorithm (Zhang, 1992), giving a measurement of how well the two boundaries match. When the change in the BAR metric is negligible, ICP has determined the best alignment possible. The BAR metric therefore represents how well the outer geometry matches, and can potentially give insight into where the problems lie for a particular localization algorithm.

Girod (2005) uses a similar technique to compare the shape of a localized network, irrespective of translation, scale and rotation. He defines a four-step approach influenced by the Procrustes method of characterising shape, and uses it to measure estimation fit with ground truth. Firstly, a scaling factor between the real and estimated topologies is established; the maps are then translated and scaled relative to the origin, which is defined as the node closest to the centroid of the estimated topology. The estimated topology is then rotated according the angular offsets between nodes, and finally translated by the average distance between estimated and ground truth points. Average error can now be taken using any of the metrics we have previously described (MAE, GER, GDE, etc). Whilst both Girod and Efrat's approaches take into account the shape of the network, BAR needs only a subset of the nodes on the boundary to contribute towards the computation; Girod's method uses all nodes in the network.

Metrics without Ground Truth

The accuracy metrics above rely on prior knowledge of the actual node position and physical network topology in order to evaluate the localization quality and error. In realistic, un-positioned WSN deployments, this information is not known, and so measurement of error must be determined relative to what information *is* available. For example – if we assume a range-based localization algorithm where nodes measure distance between one another and their positions are estimated based on this information, a metric not using ground truth must compare the measured ranges with the ranges derived from the estimated positions. Unlike the ground truth metrics above, this means that only the actual measured distances can be compared with localization derived ranges. Toward this aim, Girod (2005) defines an average distance error metric, shown in (7). In this case, the estimated distance between two nodes i and j is subtracted from the observed range R_{ij} between them.

$$error = \frac{2}{n(n-1)} \sum_{i,j,i<j} R_{ij} - \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2} \tag{7}$$

Başaran (2006) suggests a FROB similar metric, called SPFROB (shortest-path FROB), based on the shortest path between two nodes, rather than Euclidean distance. This metric is potentially useful for multi-hop localization algorithms which infer distance by the estimated shortest path from a node to an anchor, such as the ad-hoc positioning system (APS) by Niculescu and Nath (2001).

Cost Metrics

Cost metrics relate to how "expensive" it is for localization to be carried out. These costs are related to the traditional constraints of wireless networked sensing devices – low power operation, low computational capability, and redundancy through scale and density. Cost is an important trade-off against accuracy, and is often motivated by realistic application requirements, which are discussed in more detail at the end of the metrics section. As such, cost metrics are typically used to evaluate the trade-offs that are not addressed by positional error and coverage. Several common metrics are described below, along with how they may be determined.

Anchor to Node Ratio

Minimising the number of anchors in the network is desirable from an equipment (cost, power usage) or deployment point of view. For example, using anchors that can estimate position through the Global Positioning System (GPS) will require extra hardware which is both expensive and power-hungry, thus limiting the node lifetime. Similarly, pre-defining anchor positions may be hard if the supposed deployment mechanism is random placement (i.e. nodes being thrown from a vehicle). The *anchor to node ratio* is simply the number of anchors in the network divided by the number of nodes. This metric will typically be used to investigate the trade-off on the accuracy of the localization algorithm, i.e. as the percentage of anchors decreases, how does this affect the accuracy, and the percentage of nodes that can be localized? In anchor based localization algorithms, one must also consider the placement and density of anchors – this is discussed in the Coverage Section further in the Chapter. When using few anchors, Nagpal et al (2003) find that dense networks (on average 15 neighbours per node) are required to provide relatively accurate localization results, but that this accuracy is bounded by the method used to estimate inter–node distance (in their case radio range).

Communication Overhead

Since radio communication is assumed to be a large consumer of power relative to the overall consumption of a wireless sensor node, minimising communication overhead is paramount in maximising the potential network lifetime. Communication overhead will most likely be measured either by actual power consumed or number of packets transmitted to achieve the localization goal. For example, Langendoen et al (2003) use the average number of packets sent per node; power consumption can be derived from this if one knows the cost of sending a single packet (as is discussed in more detail in Modelling Section). This metric will typically be evaluated with respect to the scaling of the network – how does communication cost increase as the network increases in size?

Power Consumption

The proportion of available power that a node spends on localization can affect its lifetime (and the network lifetime). Power consumption will be a combination of the power used to perform local operations and the power used to send and receive messages associated with localization. The more complex the local processing for localization is, the longer it will take the node to process. As above, this metric will also typically be evaluated with respect to scaling of the network – how does power consumption increase as the network increases in size?

Algorithmic Complexity

Standard notions of computational complexity in time and space (i.e. big O notation) can be used as comparison metrics for the relative cost of localization algorithms. For example, as a network increases in size, a localization algorithm with $O(n^3)$ complexity is going to take a longer time to converge than an $O(n^2)$ algorithm. The same is true for space complexity – as the number of nodes increases, the amount of RAM needed (either per node, or centrally) is going to increase at a particular rate; algorithms which require less memory (comparatively) at a given scale may be preferable. This may help motivate a trade-off between centralised and decentralised algorithms – i.e. the centralised approach might be better in some cases if the per-node memory footprint becomes too large as the network scales (this would obviously be offset with the communication overhead).

Convergence Time

Measuring the time taken for both initial measurement gathering and localization algorithm convergence can both provide important comparison metrics. Time taken will most likely be evaluated against network size. For example – how does time taken to gather measurements or localize the network increase as the network increases in size? On the other hand, even for applications with fixed numbers of nodes, a network that takes a long time to localize may be useless if the application requires rapid deployment and processing immediately related to node positions, such as tracking of a moving target. Similarly, if one or more of the nodes in the network are mobile, the time taken to update position may not reflect the current physical state of the network – i.e. positional information may have become stale.

If the localization algorithm is based on non-linear optimisation, there may also be a trade-off to be made between accuracy and convergence time – the extra time taken and energy expended to get a slightly more accurate solution may not be beneficial.

Hybrid Metrics

Hybrid metrics encourage the evaluation of trade-offs in localization algorithms by combining several individual metrics into one composite metric. The way in which the metrics are combined will vary from one hybrid metric to another – one such example is the performance cost metric by Ahmed et al (2005).

Performance Cost Metric (PCM)

The performance cost metric (PCM) is a simple hybrid metric where performance cost C and localization error GDE are weighted by a parameter α, as shown in (8). This weighting is determined by the relative importance the evaluation wishes to place on the relevant components of the metric.

$$PCM = \alpha(GDE) + (1-\alpha)C \qquad (8)$$

Here, GDE (Global Distance Error) localization accuracy metric is a variant of GER, as described in the previous section. The cost aspect C of the PCM metric is described by the average per-node energy required to complete the localization (although one could imagine it being any quantitative cost metric). In deciding whether to use hybrid metrics instead of individual performance metrics, researchers should establish whether the values determined by the hybrid approach represent a fair or meaningful comparison.

Coverage Metrics

Some localization algorithms may not be able to localize all of the nodes in the network. Coverage is simply a measure of the percentage of nodes in the deployed network that can be successfully localized, regardless of the localization accuracy (which is described by previous metrics). However, density of deployment, as well as placement of anchors can have effects on coverage results for different localization algorithms. The effects and their evaluation and are discussed in the following subsections.

Density

The specific approaches that localization algorithms take can directly affect coverage. This can have different implications for anchor based and anchor free localization algorithms. For example, the robust localization algorithm proposed by Moore et al (2004) is an anchor free localization algorithm, based on range estimates between nodes. In order for a node to be considered a candidate for localization, there must be sufficient range estimates between the node and its neighbours to satisfy certain *rigidity* constraints (to protect against positional ambiguities which adversely affect localization results). If the density of the deployment is low, it may be impossible to localize many nodes. Figure 2 shows the relationship between node density, number of anchors and localization error for a multi-hop localization algorithm with random topology (Basaran et al, 2008). As the average node density increases, neighbor nodes generate more information, which can potentially improve localization performance with respect to localization error. Localization algorithms focusing on denser networks should bear in mind that radio traffic, number of message collisions and energy consumption of the nodes will also increase with the increasing average node density in the sensor network.

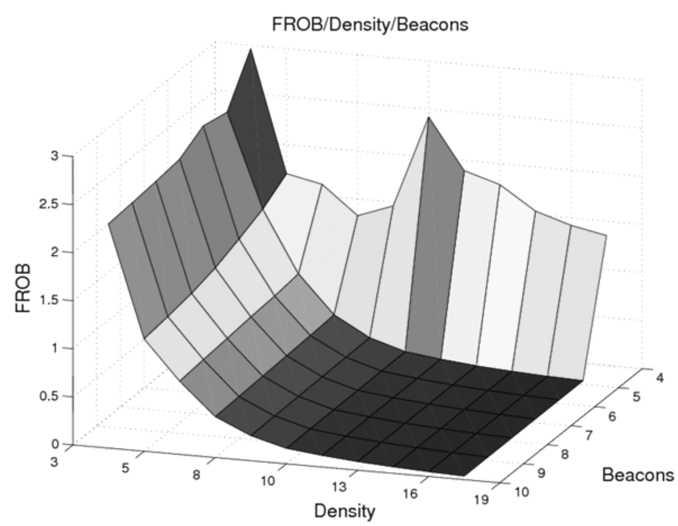
In anchor free localization algorithms, density is measured simply by the average number of neighbours a node has, as in AFL by Priyantha (2003) and the robust localization algorithm proposed by Moore (2004). Density can be used to determine the *minimum* neighbour density required for 100% localization coverage, or for an acceptable level of accuracy.

With reference to anchor based localization algorithms, Bulusu et al (2001) investigate the effects of anchor placement on localization (discussed further below), evaluating mean and median error improvement against anchor density (or *degree*) per square metre, given a random placement strategy. Similarly, in work on partially localizable networks, Goldenberg et al (2005) examine the percent of localizable nodes in the network as the number anchors increases. The authors measure anchor density in terms of average anchors a node has either in its effective radio communication or measurement region.

Anchor Placement

The position of anchors in the network may have a considerable impact on localization error, especially if the localization algorithm assumes that anchors are uniformly or randomly positioned in fixed locations. Assumptions about a pre-defined anchor placement scheme do not take into account environmental factors, terrain (that can affect placing of anchors), and signal propagation conditions, as well as optimal anchor placement. The geometry of anchor nodes with respect to any un-localized nodes in the network can have a varying effect on the accuracy of resulting position estimates. This effect is notably observed in GPS systems, where positional accuracy is seen to decrease when GPS satellites are closer to one

Figure 2. The change in the Frobenius error on a random grid topology for different node densities and increasing number of anchors



another. The Geometric Dilution Of Precision (GDOP) metric is used in GPS systems to describe the geometric “strength” of the GPS satellites’ current positions with respect to the target, and thus can give an indication of whether the accuracy is likely to be good (a small value), or bad (a large value).

Savvides et al (2005) use a GDOP metric to investigate anchor placement in WSNs, using the metric to find the ideal anchor geometry. They conclude that a convex hull of anchors surrounding un-localized nodes is the most favourable configuration for minimising the effects of geometry on localization accuracy.

It is worth noting that some localization algorithms iteratively localize nodes. As a consequence, geometrically significant nodes (i.e. nodes that might allow others to be localized) may not themselves be localized, which could result in low a coverage percentage. Complimentary to this position, Bulusu et al (2002) hypothesise anchor placement needs to be *adaptive* in the face of noisy and unpredictable environmental conditions, proposing and evaluating two simple, mobility based proximity algorithms for incremental anchor placement.

Mobile anchors (or beacons) could also potentially be used to supplement coverage or reduce the number of anchors necessary. A mobile beacon based Bayesian approach to localizing network nodes has been proposed by Sichitiu (2004), and mobility models for simulation are discussed in the Models section of this Chapter.

Evaluating Coverage

In evaluating coverage performance for localization algorithms, researchers must be prepared to try various placement scenarios/strategies for nodes and anchors, as well as various densities. One can evaluate how the accuracy improves as either the number of anchor nodes or neighbours per node increases. Bulusu et al (2002) note that increasing the anchor density does not necessarily guarantee more accurate localization or better coverage; there is essentially a “saturation point” after which no additional gains in accuracy can be made. This is supported by the results shown in Figure 2. Therefore, localization algorithms should be investigated not only with respect to the fewest anchors that can be used, but also the point at which anchors give little or no improvement.

In addition, excessively noisy, biased or missing input data may cause the localization algorithm to behave in unpredictable ways, and may reduce coverage. Therefore as part of understanding coverage, a localization algorithm should also be evaluated with respect to its resilience in the face of varying amounts of measurement noise, as in Langendoen (2003).

Discussion

Accuracy, cost and coverage represent trade-offs for localization algorithms. This is a consequence of localization algorithms usually needing to be optimised toward a set of specific constraints, such as low power operation, speed of localization, scalability or a maximum positional error. A good understanding of trade-offs is important in the context of localization, as it is in general for WSN application design. For example, deploying a network with a large number of anchors is expensive, and requires a large amount of careful placement, especially to guarantee coverage. However, in attempting to minimise or remove entirely the need for anchors, a localization algorithm may compromise its accuracy and simplicity; anchor-free localization algorithms are frequently centralised (even the robust localization algorithm proposed by Moore et al (2004) requires a central phase), and framed as non-linear optimisation or minimisation problems, such as Girod et al (2006), Gotsman and Koren (2005). These approaches may not be tractable to run directly on resource constrained nodes.

It has been shown that accuracy metrics based on average position error may not capture the accuracy of the layout geometry. This is especially true for anchor-free localization algorithms. It has also been shown that the cost of a localization algorithm can take many forms, and can be highly dependent on the application requirements the WSN is designed and deployed to address. Coverage is greatly affected by placement of nodes in the network, be they anchors or regular nodes.

In creating new metrics for algorithm comparison, the designer must carefully consider the performance metrics that need to be addressed. Hybrid metrics can be useful if more than one metric must be analysed at the same time, and it makes sense to evaluate them together. Otherwise, using individual metrics to isolate specific aspects of localization performance is a fine way to evaluate and compare localization algorithms.

The first step toward fully evaluating a localization algorithm is to use the metrics presented in this section and apply them in simulation, along with relevant parameters that best represent the WSN application scenario. These matters are addressed in the next section.

EVALUATING LOCALIZATION PERFORMANCE: REPRESENTATIVE TOPOLOGIES AND SIMULATION MODELS

Evaluation and comparison of localization algorithms can be performed at various scales and using various metrics, as discussed in the previous section. Because real life deployments are expensive and difficult to scale to large numbers, simulation is a relatively easy and highly available tool to validate the performance of localization algorithms. It allows comparative performance evaluation for different environmental models and requirements imposed by the application domain. It also allows researchers to test the robustness of localization algorithms against variable conditions such as ranging error, various network topologies, anchor densities, and numbers of nodes. Simulations can also allow individual characteristics of algorithms to be isolated and evaluated by factoring out or simplifying real-world effects. However, statistical models used in localization simulations can make unrealistic assumptions about the ranging characteristics of deployment environments, which may result in misleading or incorrect results that only come to light during final deployment.

Measuring the performance of a localization algorithm via simulation requires a simulation environment and input parameters that are derived either from statistical models or empirically. The accuracy and achievable precision of localization algorithms strongly depends on the accuracy of the models used in the derivation of these input parameters, making this one of the over-riding limitations of simulation. There are a number of general purpose and localization specific simulators that can be used to evaluate and compare algorithms, including ns-2, OmNet++, and RiST (Reichenbach 2006). Some of these simulators have support for mobility and mobile radio communication, which can aid localization simulation.

This section presents some commonly used component models and building blocks for localization techniques. First, representative network topologies are introduced. This is followed by a presentation of a set of models for: inter-node ranging, noisy radio communication links, and energy consumption. The potential effect of ranging irregularities on localization performance is discussed, as well as other parameters that affect performance (such as node density, anchor/beacon placement, and mobility).

Topologies

Defining ground truth node deployment topologies in simulations can play an important role when comparing the performance of localization algorithms. For example, uniform grid, C-shape and ring-shape topologies can induce effects on localization algorithms that compromise their accuracy. There are essentially two main categories of sensor network topology, *even* and *random*. Even topologies distribute sensor nodes (and anchors) over the deployment area in an exact grid, whilst random topologies perturb individual nodes positions on the grid with random noise (with some predetermined range and variance). Figure 3 shows examples of both topologies. The results collected from the exact grid topology (Figure 3.a) are useful because they are visually simple – it is clear to see deviations in position estimation caused by the localization algorithm. Random topologies, however, better reflect the deployment scenarios in real-world environments (nodes cannot necessarily be placed uniformly). This is also because sensor networks may be deployed in locations where manual placement is either limited (e.g. in a thick forest) or almost impossible (e.g. inside a volcano). In these cases, it is generally assumed that nodes are randomly dropped from some deployment vehicle, and uniform placement cannot be guaranteed.

For these reasons, random network topologies are generally more popular for involved simulation and comparison studies. Topologies can be further sub-classed (*regular* and *irregular* topologies) according to the regularity of their placement densities and shapes.

Regular Topologies

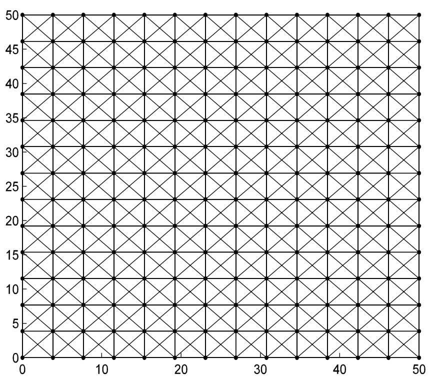
In regular topologies (such as those shown in Figure 3), nodes are typically uniformly distributed over an area as a grid. This has the advantage that average node density in each part of the deployment area is relatively consistent. Many well known multi-hop localization algorithms, such as APS by Niculescu and Nath (2001), estimate the shortest-path distances (in terms of actual distances or number of hops) between sensor nodes and derive an overall Euclidean distance from this to estimate position. Such algorithms, when evaluated in simulation using regular topologies, may appear to be highly accurate, or at least have bounded error. However, this is not sufficient to prove the general effectiveness of a localization algorithm; regular topologies do not necessarily accurately reflect realistic deployment scenarios due to the variety of geographical factors that may restrict placement of sensor nodes.

Irregular Topologies

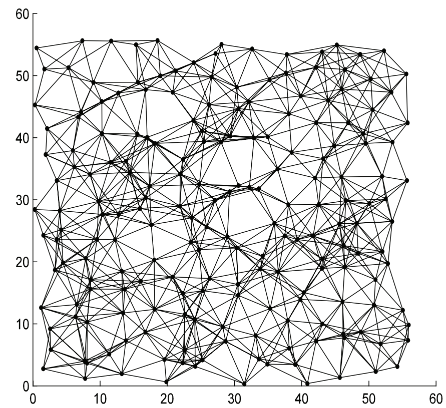
In these topologies, the shortest-path distances between nodes can deviate greatly from the actual Euclidian distances between nodes, and individual node density in a region may deviate greatly from the average density of the WSN. C-shaped, L-shaped and ring-shaped topologies are typical irregular topology examples, and represent irregular deployment configurations that applications may find themselves constrained by. Therefore, such topologies are generally employed to compare and stress various attributes of localization algorithms. In Figure 4 two types of C-shape topologies are presented. Note that in Figure 4, the difference between the Euclidian distance and the shortest-path distance between certain nodes can be large. As a result, individual errors in the localization algorithm may accumulate, resulting in large overall localization errors.

These simple topologies may be combined to generate either larger or more complex sensor network topologies. Obviously, a localization algorithm is more robust and generally usable when it generates accurate results for these types of topologies.

Figure 3. Even and random topology examples



a) Even grid



b) Random uniform

Third Dimension in Topologies

Today, most of the localization studies tend to ignore or trivialize the third dimension in topology setup and simulation. However, the third dimension is unavoidable in most real-life deployment scenarios and, unfortunately, introduces additional complexities to the localization algorithms, as examined by Ghosh (2007). For instance, in a network deployed on a hill or mountain, geographical obstacles hinder the radio communication among nodes. In such scenarios, a node may experience better packet reception but worse transmission rates compared with nodes on higher ground. This increases the percentage of asymmetric links, which may therefore affect communication and ranging assumptions.

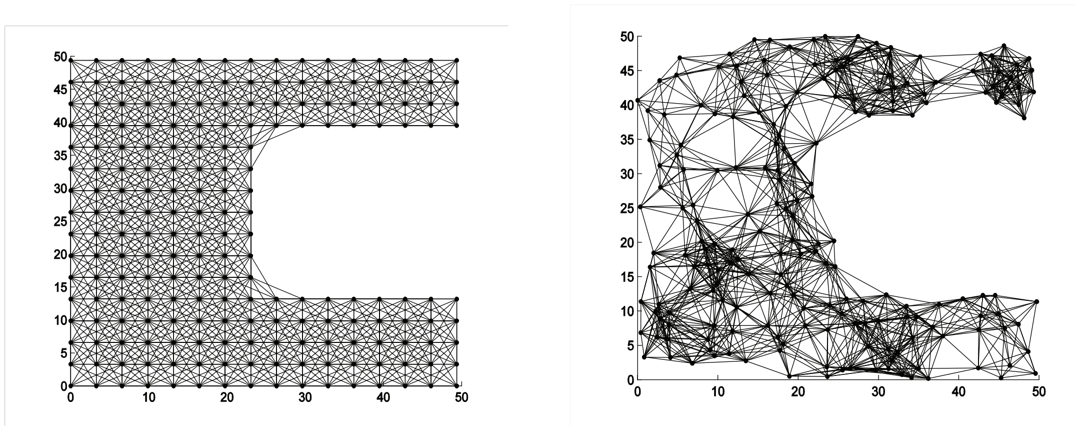
Ranging Models

Ranging is the process of estimating the inter-node distance or angle using one or more modalities (for example signal strength or acoustic time of flight). In simulation, a widely used ranging modality is radio signal strength, but other modalities include acoustic time of flight (ultrasonic or audible) and ultra-wide band (UWB), as discussed by Yu (2004). All ranging techniques approximate the distance between nodes, therefore error related to measurement accuracy, multi-path effects and non-line of sight is expected.

Noisy Disk Model

An accurate, sensible ranging model is a critical aspect of a range or angle-based localization algorithm. The noisy disk, where a node can emit ranging signals to all neighbors within a maximum range R (the radius of the disk), is a commonly used ranging model in simulation. The model has two components: noise and connectivity. The noise component indicates the distribution of error, which is added to the actual distance (e.g., Gaussian, uniform) to form the estimated distance. The connectivity component indicates the maximum distance d_{max} between two nodes at which a distance estimate can be obtained.

Figure 4. Example irregular topologies



a) Even C-shape *b) Random uniform C-shape*

For example, using Gaussian noise (with variance σ), the Noisy Disk defines the distance estimate $\hat{d}_{ij}$ between nodes *i* and *j* in terms of the true distance d_{ij} as:

$$\hat{d}_{ij} = \begin{cases} N(d_{ij}, \sigma) & d_{ij} \le d_{\max} \\ \textit{undefined} & \textit{otherwise.} \end{cases} \tag{9}$$

The Noisy Disk model with no noise component (i.e., it only models the connectivity between nodes) is also known as the Unit Disk model. In simulations, range data used in distance estimations are usually generated from a parametric function of a theoretical propagation model. The level of the detail of the propagation model used is particularly important. Insufficient details can produce unrealistic error in estimations. Unfortunately, even detailed theoretical models may be significantly different from estimations made in real life deployments because of hardware and environmental ranging irregularities.

Whitehouse et al (2005) have proposed an alternative simulation technique – *Statistical Emulation*, where data for simulation is generated by randomly drawing measurements from an empirical data set (discussed further in the range irregularity section). Reichenbach et al (2006) have also proposed a tool using defective observations as input thus enabling more realistic simulations. Theoretical propagation models are briefly discussed below.

Radio Propagation

In radio models, the received signal strength is usually represented with the following formula, measured in decibels:

Received Signal Strength = Sending Power – Path Loss + Fading (10)

The *Sending Power* of a node is determined by the battery status and the type of transmitter, amplifier and antenna. *Path Loss* describes the signal's energy loss as it propagates to the receiver. Path loss can be calculated using different physical models. The "Free Space Model" assumes the ideal propagation condition: that there is only one clear line-of-sight path (LOS) between the transmitter and receiver with no obstacles nearby to cause reflection or diffraction. The path loss is modeled as being proportional to the square of the distance between the transmitter and receiver, and also proportional to the square of the frequency of the radio signal. This model accounts for the propagation distance between sender and receiver using a fixed formula for signal loss, and does not include hardware specific factors such as the gain of the antennas used at the transmitter and receiver, nor any loss associated with mechanical imperfections.

The 'Two-Ray-Ground Reflection Model' considers antenna orientation and distance from ground for both the transmitter and receiver, performing detailed radio ray tracing to estimate reflection of signals. This model is known to give more accurate predictions at a long distance than the free space model. However it does not perform as well at short distance due to the oscillation caused by the constructive and destructive combination of the two rays. In the case where distance between nodes is small, the free space model may be preferred.

The effects of reflection, diffraction and scattering of signals as they hit obstacles will influence the free propagation of signals, leading to observation errors at the receiving node. These effects cause an exponential decay on the signal strength with respect to distance. Signal strength is also assumed to be log-normally distributed for a given distance *d*. The log-normal shadowing path loss model which is the most commonly used radio propagation model in WSN simulations is given as follows:

$$P_r(d) \big/ P_r(d0) = -10\beta \log(d / d0) + X \tag{11}$$

Where $P_r(d)$ is the received power for distance d and $P_r(d0)$ is the received power for a reference distance $d0$. β is the path loss exponent (rate at which signal decays). X is a Gaussian random variable with zero mean and standard deviation σ. β and σ are obtained through curve fitting of empirical data. To approximate a communication link with the shadowing model, Ramadurai and Sichitiu (2003) suggest a simple approach to calculate the distance using a certain radio propagation model and introduce a random error E to the calculated distance.

Acoustic Ranging

A useful model for acoustic ranging error (audible or ultrasonic) proposed by Girod (2005) can be given as follows:

$$R_{ij} = \left\| d_{act} - d_{est} \right\| + X_{ij} + N_{ij} \tag{12}$$

In (12), d_{act} is the actual distance between nodes i and j, d_{est} is the estimated distance and X_{ij} is a Gaussian random variable with zero mean and standard deviation σ. N_{ij} is a fixed bias, which is present only when line-of-sight is blocked. This model represents the basic error components that one finds in acoustic ranging – a non-line of sight bias component and a Gaussian error component. X_{ij} can be reduced by repeated observations but N_{ij} needs to be filtered at higher layers.

Ultra Wide Band Ranging

An UWB radio ranging system has the ability to resolve multi-path components of the wireless propagation channel with extremely high time resolution. A standardized UWB channel model for IEEE

802.15.3a is claimed to best match the empirical measurements (Lee, 2002; Yu, 2004; Forrester, 2003). Additionally, Shah et al. (2005) has shown UWB ranging can be used to devise algorithms robust for both LOS and NLOS environments. The impulse response of the UWB channel model is:

$$h(t) = X\sum_{l=0}^{L}\sum_{k=0}^{K}\alpha_{k,l}\,\delta(T - T_l - T_{k,l}) \tag{13}$$

where $\alpha_{k,l}$ and $T_{k,l}$ are the multipath gain and delay of the k^{th} ray in the l^{th} cluster, respectively. T_l represents the delay of the l^{th} cluster and X indicates the log-normal shadowing effect. Detailed distribution functions of different variables in can be found in Forrester (2003).

Range Irregularities

Aforementioned range models assume circular propagation ranges, whereas in reality propagation ranges tend to have an irregular shape. Range irregularity is one of the main sources of asymmetric links in WSNs. Irregularities are caused by three main factors, relative to the ranging model: device properties, propagation medium and environmental factors. Device properties include the antenna type, the transmission power, antenna gains, receiver sensitivity, receiver threshold and the Signal-Noise Ratio (SNR). Propagation medium properties include the medium type and background noise and environmental factors include attributes such as the temperature of the environment and obstacles within the deployment area (Zhou et al., 2004). As an example, the radiated pattern of the inverted-F antenna installed in the widely-used ChipCon CC2420 radio (Andersen, 2007) is very obviously non-isotropic. Therefore, it is clear that simple radio models that assume a perfect, spherical radio range cannot accurately predict or describe real-world radio characteristics.

Range irregularity models aim to reduce the discrepancy between the simulation and real-world results; two such examples are described below.

Statistical Emulation: Acoustic or Radio Ranging Irregularities

Whitehouse and Culler (2006) identified four different types of empirical ranging irregularities arising from empirical ultrasound/radio range data, which they use to extend the Noisy Disk model (as mentioned previously in this chapter). They define the ranging irregularities as:

- **Extreme overestimates:** An excess of range estimates that are larger than the true distance by more than two standard deviations.
- **Extreme underestimates:** An excess of range estimates that are smaller than the true distance by more than two standard deviations.
- **Long-range proficiency:** The existence of range estimates between nodes farther than nominal range d_{max}.
- **Short-range deficiency:** The existence of range failures between nodes closer than nominal range d_{max}.

The authors then study five stages of ranging models, each incorporating more ranging irregularity detail than the previous one:

- **Model 1)** Noisy Disk (No irregularities)
- **Model 2)** Model 1 + Extreme Overestimates
- **Model 3)** Model 2 + Extreme Underestimates
- **Model 4)** Model 3 + Long-range proficiency
- **Model 5)** Model 4 + Short-range deficiency

Whitehouse and Culler use their empirical ranging data to generate ranging irregularities in simulation, proposing a technique they call *Statistical Emulation*. The authors find that small variations in ranging model can cause large variations in localization error for several algorithms.

RIM: Radio Irregularity Model

Zhou et al (2004) establish a radio model for simulation, called the Radio Irregularity Model (RIM). From experimental results, they assign the following properties to radio sensing hardware:

- **Non-isotropic:** The radio signal from a transmitter has different path loss in different directions.
- **Continuous variation:** The signal path loss varies continuously with incremental changes of the propagation direction from a transmitter.
- **Heterogeneity:** Differences in hardware calibration and battery status lead to different signal sending powers, hence different received signal strengths.

RIM enhances radio models by approximating these three main properties of radio signals. To reflect the two main properties of radio irregularity, namely non-isotropic and continuous variation, Zhou et al (2004) adjust the previously mentioned path loss formula of (10) with two new parameters: the Degree of Irregularity (DOI) and Variance of Sending Power (VSP). DOI is the maximum received signal strength percentage variation per unit degree change in the direction of radio propagation. When the DOI is set to zero, there is no range variation, and the communication range is a perfect sphere. When it is increased, the communication range becomes more and more irregular. The path loss formula is adjusted as follows:

$$Path\ Loss_{DOI} = K_i\ Path\ Loss \tag{14}$$

K_i is a coefficient to represent the difference in path loss in different directions, and α is a random number between -1 and 1, which is generated according to the Weibull distribution (Devore, 1982). Specifically, K_i is the i^{th} degree coefficient, which is calculated as follows:

$$K_i = \begin{cases} 1 \quad , \quad i = 0 \\ K_{i-1} + \alpha \quad DOI \quad , \quad 0 < i < 360 \quad \Lambda \quad i \in N \end{cases} \tag{15}$$

where $K_0 - K_{359} \leq DOI$

Based on (15), 360 K_i values for 360 different directions can be generated by randomly fixing direction as the starting direction represented by i=0. The second parameter, Variance of Sending Power (VSP) is defined as the maximum percentage variance of the signal sending power among different devices. The signal sending power is adjusted as follows:

$$Sending\ Power_{VSP} = Sending\ Power\ (1 + \alpha\ VSP) \tag{16}$$

In (16), Zhou et al (2004) assume that the variance of sending power follows a Normal distribution, which is broadly used to measure the variance caused by the hardware, and α is a random number between 0 and 1. With these two new parameters, *DOI* and *VSP*, the RIM model is formulated as follows:

$$Received\ Signal\ Strength = Sending\ Power_{VSP} - Path\ Loss_{\ DOI} + Fading \tag{17}$$

The authors implement the RIM model in GloMoSim, discovering that the radio irregularity has a greater impact on the routing layer than the underlying link layer.

The ranging models presented in this section affect the accuracy of the estimated distances between nodes. However, as previously discussed in this chapter, other characteristics of localization algorithms should be evaluated, such as running time, coverage, total energy or communication cost.

Communication Models

Bartelli et al. (2007) stated that many recently proposed localization algorithms have both distributed and range based characteristics. For these classes of localization algorithms, there is a dependency on the reliable communication of local neighborhood information in the network. Therefore, simulation and emulation evaluations of these algorithms require an adequate link abstraction. For example, a node running a distributed localization algorithm may want to collect neighborhood information in order to determine its relative position. Other nodes will most likely be performing the same tasks, causing simultaneous packet transmissions, and therefore collisions. Because the communication overhead of a localization algorithm affects both the running time and energy cost, it is important to model links well in evaluating these metrics. For example, an algorithm which generates a lot of traffic will most likely cause problems in a large network, and may significantly reduce the network lifetime unless properly coordinated.

Packet Reception Ratio (PRR), which is a function of the distance between transmitter and receiver, can be used to model the link, as described by Zuniga et al (2005). An alternative is to use a statistical model. A commonly used packet loss abstraction for wireless link layer simulation is a two state Markov model called the Gilbert-Elliott channel. The loss process is determined by the current state of a discrete time stationary binary Markov process. It is assumed that no packets are lost in a 'good state' S_g while all packets are lost in the 'bad state' S_b. The stationary probability of a channel being in the bad state is:

$$P(S_b) = \alpha / (\alpha + \beta) \tag{18}$$

where $\alpha = P_{gb}$ and $\beta = P_{bg}$ denote transition probabilities between S_g and S_b, and vice versa, respectively. Thus, the average packet error probability of the channel is:

$$P_s = P_b P(S_b) + P_g (1 - P(S_b)) \tag{19}$$

where P_b and P_g are the error probabilities in bad and good states respectively. The state transition diagram for a Gilbert-Elliot channel is given in Figure 5. (18) may also be used to model instant node failures in a localization simulation similarly, where S_b denotes the failure state of a node and S_g denotes the non-failure state. In the failure state, all packets sent to the node are lost regardless of the wireless channel state.

Power Consumption Model

Measuring the energy cost of a localization algorithm relies on the battery model used. A commonly used model is referenced by De Marco (2006) – when the sensor transmits k bits, the radio circuitry consumes $kP_{Tx}T_B$ energy, where P_{Tx} is the power required to transmit a bit which lasts in T_B seconds. By adding the radiated power $P_t(d)$, the energy cost E_{Tx}

$$E_{Tx}(k,d) = kP_{Tx}T_B + P_t(d) \tag{20}$$

The model is completed in (21) by adding the term E_{rx} for the reception of packets as well as transmission:

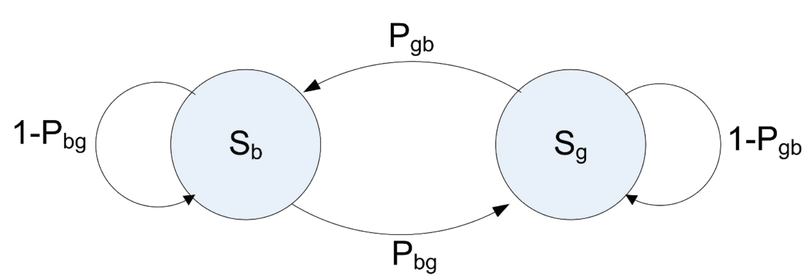
$$E(k,d) = E_{Tx}(k,d) + E_{Rx}(k,d) = kP_{Tx}T_B + P_t(d) + kP_{Rx}T_B \tag{21}$$

P_{Rx} is the power required to correctly receive (demodulate and decode) one bit. In addition to this, the energy consumption model for a single sensor can be enhanced by considering a duty cycle, which may be useful for extremely low power localization algorithms. In this model, a node can be in three operational states each draining different amounts of energy from the battery; *active state* in which the node is either transmitting/receiving/sensing data, *idle state* in which the receiver is on and the node is waiting for an activity to be triggered and *sleep state* in which the node cannot take part in any network activity (from Chiasserini and Garetto 2004). Incorporating these models can help researchers account not only for the number of packets sent by a particular localization algorithm in simulation, but also the power consumption of the packet transmission, and implications of duty cycling.

Simulating Mobility

As mobility can have an impact on the whole network performance in various ways, the model used in simulating the behavior of mobile beacons throughout the simulation is also important. The performance

Figure 5. The Gilbert-Elliott channel model



of the algorithm can vary significantly with different mobility models used for mobile entities. It is even possible to get different performance when the same mobility model is used with different parameters. Widely used wireless network simulators such as ns-2, GloMoSim, Qualnet, or Opnet support various mobility models. However, it is important to use a mobility model that most closely matches the expected real-world scenario. In a localization simulation, if the expected real-world mobility scenario is unknown then researchers should experiment with several of the available mobility models. Camp (2002) suggests that if an entity mobility model is desired, this can be well modeled by the Random Waypoint Mobility model, the Random Walk Mobility model or the Gauss-Markov Mobility model.

Discussion

This section has described various models that can be used for the evaluation of localization algorithms in simulation. However, simplistic assumptions made by these models can affect the overall performance and realism of a simulation. Using models derived from empirical results may be useful in addressing these issues, by informing statistical models. In order to evaluate individual aspects of localization algorithms, researchers may wish to iteratively add complexity to simulations. For example evaluation may start with simple radio and ranging models as well as simple deployment topologies, and progress to adding more complex power measurements and a variety of empirically modeled ranging modalities. This allows not only isolation of relevant components (accuracy being paramount initially), but a more thorough validation at several levels. This moves the simulation into a more realistic domain, preparing researchers to implement and evaluate the localization algorithm with realistic hardware in real environments. In the next section, a case study is presented showing how the real-life, application related requirements on localization can lead to tensions in performance criteria.

CASE STUDY

Given that requirements of WSN applications vary to a large degree with respect to localization, it is not easy to choose a general, representative case study for localization algorithm performance evaluation. The case study chosen here describes a localization system developed by Girod et al (2006). The system is anchor-free, highly accurate over relatively large distances (~4cm 2D positional error at tens of metres), and requires low node densities. The system was deployed and demonstrated with 10 nodes across an 80 x 50m area.

The platform requirements and localization techniques employed are described in this section, and the constraints associated with the localization algorithm together with the trade-offs in performance toward fitting application requirements. A different application is considered as a counter point to illustrate the variety of application specific demands that affect localization and localization algorithms. The reader should note that in this section, we refer to the localization of a WSN as *self-localization*, and the localization of non WSN events of interest as *source localization*.

Acoustic Source Localization

The Acoustic Embedded Networked Sensing Box (ENSBox) platform designed by Girod et al (2006) is a capable platform to support distributed acoustic sensing (see Figure 6). Acoustic sensing applications remain a persistent challenge in wireless sensing as they imply high data rates and are a rich source of challenging problems relating to source and self localization, for example Ali (2007) and Allen (2008). Relatively high processing power is needed in order to locally process data and reduce network overhead.

The primary motivator for the design of the ENSBox was a class of scientific localization applications – namely the source localization of animals and birds, based on their vocalizations. Such applications enable species census, classification and behaviour studies to be performed. For source localization, the ENSBox node employs four microphones per node as a local array, in a tetrahedral configuration over a $12cm^2$ area. A network of ENSBoxes allows the use of beam crossing techniques for source localization, where nodes individually estimate direction of arrival (DoA) of animal vocalizations using the time-difference of arrival (TDoA) of the signals at each microphone. This is possible as the acoustic signals are coherent across the node's array of four microphones. Network wide position estimations can then be made by triangulating the DoA observations. (The lack of coherency of wide-band acoustic signals over tens of metres means that it is difficult to reliably determine the 'start' of the signal at each node, meaning the use of TDoA for position estimation will not yield accurate enough results).

Using the Acoustic ENSBox for source localization requires self-localization to be performed to a high accuracy. Girod envisages the network being used to surround a target at 30-50m spacing (target to node). As such, he sets the self-localization accuracy requirements to be ±0.5m positional estimation and ±1 degree orientation estimation. This is sufficient to keep the source localization positional error within the bounds of the state of the art (±2.5 degrees for the most comparable system to an ENSBox network). Because DoA estimates are used in source localization, it is important that the geometry of the physical topology estimated by the self-localization algorithm be consistent with the actual physical topology. Actual distance error is not as important, given that for DoA triangulation, it is the angles between nodes that must be accurate, hence the topology need only be correct to a scaling factor. The average node density requirements of such an application are not clear, although a minimum of three nodes is required to remove ambiguity of the location of the acoustic event in two dimensions. Given

Figure 6. The Acoustic ENSBox's compact microphone array. The four microphones are arranged in a tetrahedral configuration over a $12cm^2$ area.

that nodes can potentially be deployed over any terrain, it is important that the localization algorithm works well in 3D as well as 2D.

The self-localization solution the Acoustic ENSBox employs is based on acoustic time of flight (ToF) and direction of arrival (DoA) estimation, employing an iterative non-linear least squares minimisation multilateration algorithm (NLLS). The ENSBox nodes are equipped with omni-directional speakers to emit pseudo-noise ranging chirps from; nodes chirp at known times in a sequence, and estimate ToF ranges from each other. DoA estimates are based on an approximate six way cross correlation of the ranging chirp across the four audio channels. The ToF and DoA estimates are used as constraints in the NLLS algorithm, which is carried out in a centralised manner – all nodes report ranges to one elected leader, who performs the localization computation.

The NLLS self-localization algorithm works best when its system of equations is over constrained (that is, there are many range and angle measurements per node). This means that erroneous measurements can be removed at certain points during the position estimation process through outlier rejection procedures. These rejections are based on heuristics such as residual error between two nodes' range estimates and residual error between estimated position and estimated range. Node orientations are iteratively estimated between NLLS iterations by averaging the error between observed DoA and angle based on the NLLS result. Convergence is assumed when residual error for different aspects (yaw, pitch, roll, range) falls below an empirically determined threshold. Sometimes, this means that under constrained systems do not converge.

Girod notes that raw residual error is not sufficient to detect outliers from the linear system formed as part of the NLLS localization algorithm. Therefore, in order to remove outliers, the localization algorithm makes use of studentized residuals (where residual error is divided by an estimate of its standard deviation), a common method of detecting outliers in statistics. Outlier detection is performed after the algorithm has converged, so that the most outlying residual can be removed as a constraint from the linear system. This will potentially enhance the overall localization result, and can be iterated while the algorithm still converges. Girod observes that: 1) average residual error itself is not a good metric to determine a bad fit of coordinates (when ground truth is not available), and in his experiments, 2) that there was not an obvious relationship between average residual error and average positional error. However, there was seemingly a relationship between under-constrained nodes and positional error, pointing to a potential metric which can account for average residual error and under-constrained nodes, although average constraint density is not likely to be sufficient on its own.

Over several experiments in different, semi-obstructed environments, ten nodes were localized with an average 2D error of between 4.4cm and 11.1cm over an 80 x 50m area. The average 3D error was between 26.0cm and 57.3cm – this difference was due to a lack of variation in the Z axis for localization experiments. In practice, it is sometimes possible to make use of a 2D solution by adjusting the pith and roll of the nodes such that their local arrays are approximately planar. This is useful if the user's confidence in the 3D solution is low.

Evaluation

The self-localization system alone will now be examined with respect to the performance criteria established earlier in this chapter – scalability, accuracy, cost and coverage. It has been established at the start of this section that the dominating requirements with respect to the application under discussion here are geometrical accuracy and robustness to ranging error, and that the system meets these requirements by taking advantage of the hardware required for the application. In terms of scalability, although the algorithm is anchor free, the processing it performs is centralised, and comes at a large computational cost. The assumption in this case is that the number of nodes deployed will not be so large to take an unreasonable amount of time for the algorithm to converge on a solution (order of minutes). The algorithmic complexity in this case is $O(N^3)$, which precludes the use of this algorithm for large networks.

In terms of cost, the localization system is expensive – requiring high sample rate audio. The platform has plentiful resources (64MB RAM, 400 MHz ARM CPU), use of which comes at the expense of a shorter battery life. The hardware expense is understandable in the context of the application – acoustic source localization requires multiple microphones, computationally expensive signal processing techniques and data sampled at high rates. The components that aid the localization – time synchronisation and node-to-node state sharing – require constant communication (at least 1 packet every 4 seconds per node), which is not conducive to low power operation. The system is highly accurate, more than meeting its positional requirement in 2D (worst case 10cm error) and just going over 0.5m error in 3D, due to the local array configuration (as previously noted). Special care is given to robust behaviour, but the cost for this is a high number of measurements for each node – the localization algorithm requires an over constrained linear system to remove outliers. In a topology where the number of range measurements per node is low, outliers are likely to become difficult to remove, or even identify; this is likely to be encountered in larger networks. Because the localization algorithm is computed centrally with all measurements, coverage is either 0 or 100%; the algorithm either converges on a result or it does not. This is clearly a problem for scalability.

To conclude, in maximising the accuracy and resilience to measurement noise, the localization system becomes constrained in scalability and unconstrained in cost (power usage, message sending, and computational complexity). This is intuitive if one imagines the criteria in tension – one cannot be maximised without affecting the others. This would seem to limit the generality of the localization approach, but one could argue that any self-localization motivated by a specific application (rather than application class) will make similar optimisations to maximise performance.

Counterpoint

As a simple, brief counterpoint, and with the aim to bring the points discussed so far into a sharper focus, the requirements of a different application are compared to see to what extent the previous localization procedure would suit them. In this motivating example, a WSN network is deployed over a forest in order to monitor it for potential fire events. Nodes in the network acquire temperature and humidity data as part of the calculation of the Fire Weather Index, to help predict dangerous areas for fires. This prediction is intended as an "early warning" system and the network will localize areas in the forest which are highly likely to have fires (as well as detecting fires when they occur).

Forest fires usually occur in summer, and it is envisaged that the network will be deployed before and removed (or replaced) after summer, hence needing a minimum lifetime of at least 6 months continuous operation. Since the deployment area is not likely to be dangerous at deployment times, it can be assumed that nodes will be manually deployed, but that terrain surveying processed are too expensive for the size of the network. The individual constraints on the localization performance are discussed below.

Scale and Density

This network is likely to be far larger than the acoustic sensing network, in terms of number of nodes required and area to be covered. The network is required to be dense in terms of communication – between 10 and 20 neighbours on average is ideal to ensure reliable multi-hop communication paths and allow for duty cycling. Deploying 20 nodes over every 100m by 100m area is likely to be sufficient to maintain at least an average degree of 10 per node.

The fire must be related to an actual physical position, so there must be at least some nodes in the network which are GPS-enabled. However, it is unreasonable for each node to be equipped with GPS, as there is a strong likelihood that it will be rendered useless under the forest canopy. Therefore nodes equipped with GPS could be deployed around the edges of the forest, acting as anchors when required. These nodes would not necessarily have to have the same sensing capabilities as the general network, and as such could be used only when required for localization.

Cost

Network life-time must also be maximised, meaning that radio communication must be kept to a minimum. Ideally, nodes will be duty cycled to take advantage of the deployment density. Additionally, when considering concrete solutions to the forest fire application, hardware cost becomes a factor, meaning that it is not only the power consumption cost that must be considered, but also the per-unit cost. The overall cost of the network will limit how many nodes can be purchased, and so accurate ranging hardware will most likely have to be traded off for simpler, cheaper ranging approaches which are not extra to the application functionality of the system, such as RSSI ranging for example.

Accuracy

Fire event localization is unlikely to be performed in the same way as acoustic localization. The resolution requirement of a fire's geographical location is related to the type of material on the forest floor and how flammable it is. Estimating the fire position could be as coarse as the nearest 100m, and still acceptable.

Coverage

Attaining 100% coverage is important for this application. If any nodes exist in the network which are capable of flagging fire events, but that have not been localized, the network is not meeting its application goals.

Summary

To summarise, the overriding constraints in this application are cost (node price and power consumption), network lifetime and scalability. In order to meet these constraints, it is likely that the network will have to compromise on accuracy. This accuracy trade-off is likely to be manifested in a simple ranging mechanism – highly accurate ranging approaches such as audible acoustic or ultrasonic time of flight represent an extra expense which the nodes cannot justify. In this case, a distributed algorithm would seem to be the best approach. It would not have to be anchor free, although the anchor density would most likely be low, and at a low duty cycle.

It is clear that the approach that the Acoustic ENSBox network uses would not work for the forest fire application – the hardware is too heavy weight to deal with the constraints of the application, and the battery life is not suitable for a long-lived application. The NLLS localization algorithm is too specific to apply to this network, where no angle of arrival measurements could be taken. Also, the computation of the algorithm is not scalable without modification.

A LOCALIZATION ALGORITHM DEVELOPMENT CYCLE

The development and evaluation of a localization algorithm should be considered in its entirety – this implies theoretical modelling and simulation as well as real-life validation of the algorithm. Each stage of the development should characterise and validate a specific aspect of the algorithm. Simulation validates how the algorithm can operate under controlled, simulated conditions – this verifies that the algorithm *functions* correctly. Emulation verifies that the algorithm can work correctly using *empirical data* that reveals conditions which are hard to simulate. Realistic validation shows that the algorithm can work in target environments and with the hardware platforms which are being targeted to support it.

Whitehouse et al (2004) propose that whilst simulation is different from real-world performance, one would expect it to be *indicative* (within some error bound of empirical results) and *decisive* (an algorithm which performs best in simulation should perform best in reality). Therefore, when one is evaluating a localization algorithm against others, one must make sure it performs better in both simulation and realistic deployment.

The verification and validation of a localization algorithm at each of the four stages (modelling, simulation, emulation and deployment) becomes more expensive in terms of (at least) time and cost as we approach real-life deployment. The value of simulation/emulation comes forth with respect to scalability and low cost of entry for researchers – there are no embedded hardware requirements.

Simulation

Researchers can use simulation to simplify some of the difficulties of real deployment (time synchronisation, for example) such that any algorithmic flaws can be isolated at an early stage. For this reason, it is not sensible to try to start with in-situ deployment without simulation verification. Environments such as Matlab, ns-2, OmNet++, Ptolemy and EmStar would be used to simulate the performance of localization algorithms. Different simulation environments allow lesser or greater control over node and network parameters relevant to localization. Simulators such as ns-2 and OmNet++ aim to provide the user with accurate models of wireless propagation and protocol performance, providing a high level language in which to implement simulations. Their wide academic use is desirable for consistency between institutions in a way custom simulators cannot guarantee. Custom simulators can be designed in a variety of languages (Java, C and its variants). Ptolemy provides a hugely powerful framework for modelling, simulation and design of embedded systems using graphical techniques to create state machines, akin to Matlab's Simulink. Development frameworks like EmStar allow researchers to develop end-to-end wireless sensing systems, allowing the same code to be used for simulation, emulation and deployment. Hardware specific simulators, such as TOSSIM and AVRORA can be used when accurate profiling is required (in power consumption analysis, for example). There also exist localization specific simulators, such as Silhouette by Whitehouse (2004, 2006) and SeNeLEx, RiST by Reichenbach (2006).

These environments do not need to be used in isolation, of course – measurements and observations derived from one could be used as set-up parameters in another, or to help inform custom simulation software.

Emulation

Using empirical data to inform simulation parameter values, rather than purely calculating them (for example, ranging or communication data) represents an addition to the realism of a simulation. Empirical data sets can capture some of the environment-specific effects that simple models cannot. The *Statistical Emulation* method proposed by Whitehouse et al (2004), is an example of gathering a data trace in-situ, and using it to power a realistic localization simulation (thus making it an emulation). Part of the challenge of performing this type of emulation is gathering a data set which represents the environment in sufficient detail. Whitehouse (2004) gathered range data using 20 ultrasound enabled nodes that have been arranged in such a way that all ranges between 0.5m and 4.5m (at 0.25cm intervals) can be measured. This captures environmental specific problems, such as non-estimates (range could not be measured) and node-to-node ranging variations (induced by electronic or mechanical differences between nodes). Whitehouse uses this range data set in his Matlab based Silhouette localization software to investigate its effects on the performance of several localization algorithms, comparing the results with pure simulation and finding a disparity between the two. Similarly, real connectivity data can be gathered from a test bed and pushed into a simulation, creating an emulated system.

One of the most powerful emulation frameworks to date is EmStar (Girod et al., 2007). EmStar allows the user to perform simulation, emulation and real deployment using the same framework. This means code developed and simulated can be cross-compiled and tested on real embedded hardware. This approach is advantageous as there is a reduction in the amount of porting required. EmStar allows network connectivity to be emulated in real-time using test bed data, making it a powerful tool for transitioning to real hardware from simulation through emulation.

Real Life Deployment

The strength of using test beds lays in actually being able to run algorithms on real hardware, and gather non-simulated data. This can be particularly useful for testing radio communication, for example. However, creating localization test beds can often be difficult because algorithms are affected by environmental context. Ranging mechanisms will most likely work differently indoors and outdoors, for example if signal strength is being used to determine range or location. Evaluating an algorithm on a test bed in a different environment than the application targets may give an incorrect indication of the algorithm's performance.

Real life deployment of a localization algorithm on hardware in an indicative environment (i.e. similar to where the real network will be deployed) is the most important evaluation of a localization algorithm. Unfortunately, it is also the most time consuming, costly, and error prone aspect of localization evaluation. An in-situ evaluation of a localization algorithm will most likely be as demanding as a real deployment of the network in terms of planning, deployment equipment and time taken to deploy.

The deployment phase of localization algorithm evaluation is also the most error prone and unpredictable, so researchers should have a detailed plan of how and what data needs to be gathered. The aim should not be to perform a large amount of testing, but to have well directed and easily planned experimentation. Software will most likely need to be adapted to work correctly in the field, and worst case scenarios (what to do if pretty much everything fails) should be planned for. Several days should be set aside for deployment, with the understanding that the likelihood is high that things will *not* work as expected first time.

CONCLUSION

When evaluating localization algorithms, it is difficult to separate the issues arising from actual deployments from theoretical drawbacks and constraints of various algorithms. From a theoretical perspective, it is desirable to have an algorithm that is independent of the ranging technique used and platform capability, as well as being robust to the deployment environment and generic with respect to application requirements.

Given that a WSN is deployed for some realistic, physical monitoring and processing aim, the localization algorithm designer should always have some set of motivating applications in mind, throughout the design process. These can be general classes of applications such as tracking and location awareness or very specific, clearly specified applications such as forest fire monitoring and animal call localization. Different applications will place different weightings on the various criteria discussed at the start of this chapter – scalability, accuracy, coverage and cost.

In conclusion, evaluating localization algorithms is not to be underestimated by researchers. In order to fully evaluate a localization algorithm, its performance must be tested in simulation, emulation and realistic environments. Both the design and development process for new localization algorithms and the process of selecting a "best fit" algorithm for a particular application requires consideration of the trade-offs between accuracy, cost, coverage and scalability the localization system needs to achieve. Although simulation is the least costly and most used tool for evaluating algorithms within the WSN domain, with respect to localization researchers must be aware of the limitations of purely simulated models, especially for radio communication and inter-node distance estimation.

The use of metrics to describe the quality of localization is important for all evaluation criteria, but possibly most notably for accuracy evaluation. Using Euclidean error is the simplest, but not always the most telling way of measuring how well a localization solution "fits" ground truth. Also, when ground truth is not available, an equivalent metric must be found which tells the user how well the localization estimate matches the initial constraints (such as inter-node spatial estimates).

Considering the domain's state-of-the-art, being able to instantiate a specific localization algorithm is still not an easy thing to do. Even after choosing a localization algorithm that is most suitable for the motivating application, it is likely that researchers will still have to implement it on specific hardware (with relevant ranging measurement mechanisms, if applicable) before being able to evaluate its performance.

REFERENCES

Ahmed, A. A., Shi, H., & Shang, Y. (2005). SHARP: A New Approach to Relative Localization in Wireless Sensor Networks. *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)* (pp 892-888).

Ali, A. M., Yao, K., Collier, T. C., Taylor, C. E., Blumstein, D. T., & Girod, L. (2007). An empirical study of collaborative acoustic source localization. *In Proceedings of the 6th international Conference on information Processing in Sensor Networks (IPSN '07)*, (pp 41-50).

Allen, M., Girod, L., Newton, R., Madden, S., Blumstein, D.T., Estrin, D. (2008). VoxNet: An Interactive, Rapidly-Deployable Acoustic Monitoring Platform. *In Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN '08)*, (pp. 371-382).

Allen, M., Gaura, E., Newman, R., Mount, S., (2006). Experimental Localization with MICA2 Motes. *In Proceedings of NSTI Nanotech 2006*, (pp. 435-440).

Andersen, A., (2007). 2.4 GHz Inverted F Antenna. *Texas Instruments Design Note DN0007*

Başaran, C. (2007). A Hybrid Localization Algorithm for Wireless Sensor Networks. *Master's Thesis, Yeditepe University,* Turkey.

Başaran, C., Baydere S., & Kucuk, G. (2008). RH+: A Hybrid Localization Algorithm for Wireless Sensor Networks, *IEICE Transactions on Comm*., *E91-B*(No.06), 1852-1861.

Battelli, M., & Basagni S. (2007). Localization for wireless sensor networks: Protocols and perspectives. *In Proceedings of IEEE CCECE 2007*, Vancouver, Canada, April 22-26 2007, (pp. 1074-1077).

Bergamo, P., & Mazzini, G. (2002). Localization in Sensor Networks with Fading and Mobility, *In Proceedings of IEEE PIMRC 2002*, Sept 2002, Lisboa, Portugal, (pp. 750-754).

Broxton, M., Lifton, J., & Paradiso, J. A. (2006). Localization on the pushpin computing sensor network using spectral graph drawing and mesh relaxation. *SIGMOBILE Mob. Comput. Commun. Rev. 10*(1) (Jan. 2006), 1-12.

Bulusu, N., Heidemann, J., & Estrin, D. (2001). Adaptive beacon placement. *21st International Conference on Distributed Computing Systems (ICDCS-21),* (pp. 489-498).

Bulusu N., Heidemann J., Bychkovskiy V., & Estrin D. (2002). *Density adaptive beacon placement algorithms for localization in ad hoc wireless networks.* UCLA Computer Science Department Technical Report UCLA-CS-TR-010013, July 2001.

Camp T., Boleng J., & Davies V. (2002). A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, *2*(5), 483-502.

Cerpa, A., Wong, J. L., Kuang, L., Potkonjak, M., & Estrin D. (2005). Statistical Model of Lossy Links in Wireless Sensor Networks. *International Conference on Information Processing in Sensor Networks*.

Chiasserini C.-F., & Garetto, M. (2004). Modeling the Performance of Wireless Sensor Networks. *IEEE INFOCOM* 2004, (pp. 220-231).

Efrat, A., Erten, C. Forrester, D., Iyer, A., & Kobourov, S.G. (2006). Force-Directed Approaches to Sensor Localization. *Proceedings of the 8th SIAM Workshop on Algorithm Engineering and Experiments (ALENEX)*, (pp. 108-118).

Forester, J. (2003). Channel modeling sub-committee report final. *IEEE802.15-02/490r1-SG3a*, Feb 2003.

Ghosh, A., Wang, Y., Krishnamachari, B., & Hsieh, M. (2007). Efficient Distributed Topology Control in 3-Dimensional Wireless Networks. *4th Annual IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2007*, 2007, (pp. 91-100).

Girod, L., Lukac, M., Trifa, V., & Estrin, D. (2006). The design and implementation of a self-calibrating distributed acoustic sensing platform. In *Proceedings of the 4th international Conference on Embedded Networked Sensor Systems* (Boulder, Colorado, USA, October 31 - November 03, 2006). SenSys '06. ACM, New York, NY, (pp. 71-84).

Girod, L. (2005). A Self-Calibrating System of Distributed Acoustic Arrays. *Ph.D. Thesis, UCLA, USA*.

Girod, L., Ramanathan, N., Elson, J., Stathopoulos, T., Lukac, M., & Estrin, D. 2007. Emstar: A software environment for developing and deploying heterogeneous sensor-actuator networks. *ACM Trans. Sen. Netw., 3*(3), (Aug. 2007), 13.

Goldenberg, D. K., Krishnamurthy, A., Maness, W. C., Yang, Y. R., Young, A., Morse, A. S., & Savvides, A. (2005). Network localization in partially localizable networks. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* IEEE 1(13-17) (pp. 313-326).

Gotsman, C., & Koren, Y. (2005). Distributed Graph Layout for Sensor Networks. *Lecture Notes in Computer Science*, 3383/2005, Springer Berlin/Heidelberg, (pp. 273-284).

Heidemann, J., Bulusu, N., Elson, J., Intanagonwiwat, C., Lan, K., & Xu, Y., et al. (2001). Effects of Detail in Wireless Network Simulation. *Proceedings of the SCS Multiconference on Distributed Simulation*. Phoenix, Arizona, USA, USC/Information Sciences Institute, Society for Computer Simulation. January, 2001, (pp. 3-11).

Krishnamurthi N., Jay Yang S., & Seidman M. (2004). Modular Topology Control and Energy Model for Wireless Ad Hoc Sensor Networks. *Proceedings of OPNETWORK '04*.

Langendoen, K., & Reijers, . (2003). Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw., 43*(4) (Nov. 2003), 499-518.

Lee, J. Y., & Scholtz, R. A. (2002). Ranging in a dense multipath environment using an UWB Radio Link. *IEEE J. Select. Areas Commun*., *20*, December 2002, (pp. 1677-1683).

Lin G., Noubir G., & Rajaraman R. (2004). Mobility Models for Ad hoc Network Simulation. *IEEE INFOCOM* 2004, (pp. 463).

De Marco, G., Yang, T., & Barolli, L. (2006). Impact of Radio Irregularities on Topology Tradeoffs of WSNs. *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA'06)*, (pp. 50-54).

Moore, D., Leonard, J., Rus, D., & Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems* (Baltimore, MD, USA, November 03 - 05, 2004). SenSys '04. ACM, New York, NY, (pp 50-61).

Nagpal, R., Shrobe, H., & Bachrach, J. (2003). Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. *In the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), Palo Alto, April, 2003, published as Lecture Notes in Computer Science LNCS 2634.*

Niculescu, D., & Nath, B. (2001). Ad hoc positioning system (APS). *Global Telecommunications Conference, 2001. GLOBECOM IEEE*, *5*, 2926-2931.

Priyantha, N. B., Balakrishnan, H., Demaine, E., & Teller, S. (2003). Anchor-Free Distributed Localization in Sensor Networks. *LCS Tech. Report #892, MIT, USA.*

Ramadurai V., & Sichitiu M. L. (2003). Simulation-based Analysis of a Localization Algorithm for Wireless Ad-Hoc Sensor Networks, *OPNETWORK* 2003.

Reichenbach F., Koch M., & Timmermann D. (2006). Closer to Reality - Simulating Localization Algorithms Considering Defective Observations in Wireless Sensor Networks. *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC'06),* 2006, (pp. 59-65).

Savvides, A., & Garber, W. L. (2005). An Analysis of Error Inducing Parameters in Multihop Sensor Node Localization. *IEEE Transactions on Mobile Computing, 4*(6) (Nov. 2005), 567-577.

Shah, S. F. A., & Tewfik, A. H. (2005). Enhanced position location with UWB in obstructed LOS and NLOS multipath environments. *In Proc.of European Signal Processing Conf. (EUSIPCO).*

Shang, Y., Ruml, W., Zhang, Y., & Fromherz, M. P. (2003). Localization from mere connectivity. *In Proceedings of the 4th ACM international Symposium on Mobile Ad Hoc Networking & Computing* (Annapolis, Maryland, USA, June 01 - 03, 2003). MobiHoc '03. ACM, New York, NY, (pp. 201-212).

Sichitiu, M. L., & Ramadurai, V. (2004). Localization of wireless sensor networks with a mobile beacon. *In Proc. of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004),* Fort Lauderdale, FL, Oct. 2004, (pp. 174-183).

Slijepcevic, S., Megerian, S., & Potkonjak, M. (2002). Location errors in wireless embedded sensor networks: sources, models, and effects on applications. *SIGMOBILE Mob. Comput. Commun. Rev., 6*(3) (Jun. 2002), 67-78.

Wang, Y., Li, F., & Dahlberg, T. A (2006). Power Efficient 3-Dimensional Topology Control for Ad Hoc and Sensor Networks. *IEEE Global Communications Conference*, (pp. 1-5).

Whitehouse, K., & Culler, D. (2006). A robustness analysis of multi-hop ranging-based localization approximations. *In Proceedings of the Fifth international Conference on information Processing in Sensor Networks* (Nashville, Tennessee, USA, April 19 - 21, 2006). IPSN '06. ACM, New York, NY, (pp. 317-325).

Whitehouse, K., Karlof, C., & Culler, D. (2007). A practical evaluation of radio signal strength for ranging-based localization. *SIGMOBILE Mob. Comput. Commun. Rev., 11*(1) (Jan. 2007), (pp. 41-52).

Whitehouse, K., Karlof, C., Woo, A., Jiang, F., & Culler, D. (2005). The Effects of Ranging Noise on Multihop Localization: An Empirical Study. *IPSN '05,* (pp. 73-80).

Yu, J., & Oppermann, I. (2004). UWB positioning for wireless embedded networks. *In Proc. of IEEE Radio and Wireless Conference*, (pp. 459–462).

Zhang, Z. (1992). Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision, 13*(2), 119-152.

Zuniga, M., & Krishnamachari, B. (2004). Exploring the predictability of network metrics in the presence of unreliable wireless links. *SenSys 2004*, (pp. 275-276).