A Localization Framework for Wireless Mesh Networks - Anchor-Free Distributed Localization in the DES-Testbed

Bastian Blywis, Mesut Güneş, Felix Juraschek, and Steffen Gliech
Distributed Embedded Systems, Computer Systems and Telematics
Institute of Computer Science, Freie Universität Berlin, Berlin, Germany.
Email: {blywis, guenes, jurasch, gliech}@inf.fu-berlin.de

***Abstract*—Distributed localization has been in increased focus of research since the emergence of wireless multi-hop networks. One class of localization systems uses *anchor-free* algorithms, where no node in the network has information about its geographic position. None of the nodes acts as anchor and thus the algorithms have very limited requirements for the application in particular scenarios. The *Anchor-Free Distributed Localization-Algorithm* (AFL) by Priyantha et al. is such an algorithm that has been developed for applications in wireless sensor networks. It creates a relative coordinate system in the first phase and uses a mass-spring approach for the optimization of the nodes' locations in the second phase. In this publication we report our experiences implementing and using AFL in the DES-Testbed, an IEEE 802.11 based wireless mesh network. The *Distributed Embedded Systems - Localization Framework for Testbeds* (DES-LOFT) is introduced and discussed. It was developed for the implementation of localization systems as there is currently limited support for wireless mesh networks. DES-LOFT enables a holistic approach and interacts with other frameworks because a localization algorithm is never run independent from other services. Based on a first experiment series, we discuss the applicability of AFL in real world networks, propose improvements, and present preliminary results from experiments.**

***Index Terms*—Wireless Mesh Network (WMN), Anchor-Free Localization, Testbed, Framework**

I. Motivation

Localization systems can be classified based on many properties. Range-based localization measure the distances between neighbored nodes. Some systems use *Time-Of-Arrival*, *Time-Difference-Of-Arrival*, or/and *Receive Signal Strength* (RSS) values as a measure of distance. In contrast, range-free localization [1] tries to solve the task by, e.g., evaluating the hop-distances between nodes. Another classification metric considers the location awareness of nodes. Anchor-free approaches do not assume information for any node, while anchor-based localization relies on a subset of nodes that know their position in a specific coordinate system. Anchor-free localization can be applied in many networks due to its minimal requirements regarding the available hardware and the overall network structure.

The *Anchor-Free Distributed Localization Algorithm* (AFL) [2], [3] by Priyantha et al. is one example for range-based, anchor-free localization. The algorithm was designed for use in *wireless sensor networks* (WSNs) to associate detected events with a particular region. The authors assume that a large number of sensor nodes are randomly but uniformly distributed over the area. By using an available distance measurement approach, the nodes create a relative coordinate system and subsequently execute a distributed and iterative optimization algorithm. In the end, the nodes are positioned in a global embedding which is close to the actual topology, i.e., the distances between each pair of nodes in the created coordinate system resemble the true distances but are stretched or compressed by a factor. The iterative optimization is achieved by a mass-spring approach that has been used in many other fields of computer science, for example: graph drawing [4], particle swarm optimization [5], and motion synthesis [6].

Besides WSNs, anchor-free localization is interesting for many other scenarios; in general it should be able to run in most wireless networks. Indoor localization and navigation, where GPS is not available, is one promising example for future application. IEEE 802.11 is the dominant wireless technology used in todays networks as access points are installed in a large number of buildings and IEEE 802.11 transceivers are included in many laptops, smart phones, or digital music players. Thus the infrastructure is already available but currently unused. Two specific application scenarios can be envisioned. On the one hand, AFL can be run on the infrastructure network in a building, for example, a *wireless distribution system* (WDS) or *wireless mesh network* (WMN) [7]. The localization service can be used to guide mobile nodes to find people (firefighters and paramedics searching for victims), rooms, or devices (printers or mobile robots). On the other hand, when we assume a *mobile ad-hoc network* (MANET) where only limited localization accuracy will be possible, location aware nodes can improve service placement schemes [8] or enable geographic routing [9], [10]. In these general scenarios, a coarse accuracy of about 1 to 2 "normal sized" office rooms is sufficient ($\approx$ 10 meters or less).

In contrast to wireless sensor networks that are often deployed at random in a particular area and have a high node degree, IEEE 802.11 based routers in a WDS or WMN are usually not deployed in a complete random manner and have lower node degrees. Additionally, the hardware of WSNs is most often application specific and uniform while nodes in WDSs and WMNs are more heterogeneous. The only

978-1-4244-5864-6/10$26.00 © IEEE

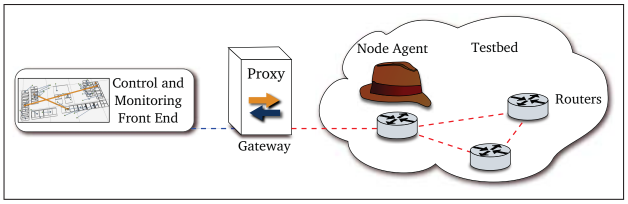


Fig. 1: Architecture overview of the three main DES-LOFT components.

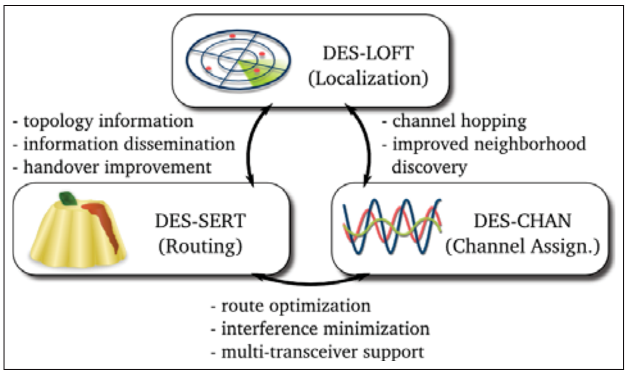


Fig. 2: Cooperation of all three frameworks. The involved components are *DES-LOFT Node Agents*, DES-SERT based routing daemons, and DES-CHAN channel assignment daemons.

common component that might be available on all nodes is the transceiver. This poses a challenge for any localization approach. Unfortunately, there is limited experience and few studies about the applicability of anchor-free localization systems in today's real world networks.

The contribution of this paper consists of two parts. First of all, we introduce the *Distributed Embedded Systems - Localization Framework for Testbeds* (DES-LOFT) that was developed to support the implementation of localization systems in a testbed environment. The cooperation between DES-LOFT and two other frameworks is discussed as localization algorithms are often run on top of other applications and will most likely never run independently. A holistic consideration of individual wireless network services is advocated. Second of all, we report our experiences implementing, running, and evaluating AFL in a WMN testbed. We discuss the applicability in this scenario and propose improvements due to issues that were not reported from simulation-based studies of AFL or equivalent algorithms.

The remainder of the paper is organized as follows. The DES-LOFT localization framework is introduced in Section II. Subsequently, Section III discusses the AFL implementation based on DES-LOFT. Section IV describes the AFL experiments run in a WMN testbed. The results of a first study are presented and discussed. Section V gives an overview of related localization systems. Critique, future work, and improvements for anchor-free localization are discussed in Section VI. The paper closes with a conclusion in Section VII.

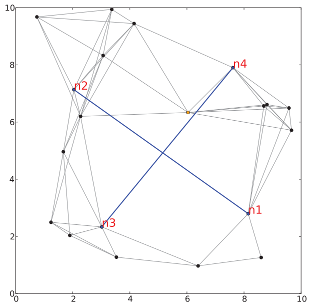
II. A Localization Framework for Wireless Mesh Networks

The *Distributed Embedded Systems - Localization Framework for Testbeds* was implemented as there is limited support for the implementation of localization algorithms given our specific requirements. In general, our software design is driven by two factors: the architecture of the testbed and the application scenarios.

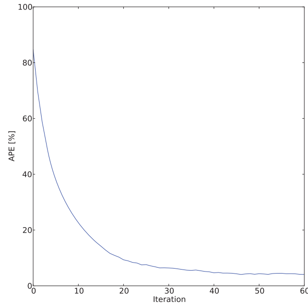
All experiments are run in the DES-Testbed. The Distributed Embedded Systems (DES) testbed [11] at the *Freie Universität Berlin* is a research testbed with the focus on wireless mesh networks and wireless sensor networks. Currently, 100 wireless mesh routers are deployed in three buildings over 4 floors. Each node is equipped with three IEEE 802.11a/b/g network cards. In the default configuration, the network interface cards are configured to orthogonal frequencies in the 2.4 and 5 GHz frequency bands (channels 13, 40, 44). This channel selection tries to minimize interference from other wireless networks in the vicinity. All nodes have an additional Ethernet connection to the central *Testbed Management System* (TBMS) which is used for configuration and monitoring to avoid interfering with the experiments that are using the wireless medium. The TBMS is used to schedule (replications of) experiments in particular time slots and allows autonomous experiment execution or user interactive experiments. In addition to the deployed nodes, several virtual networks are run on a virtualizer which is used for prototype testing and development. DES-LOFT enables to run the localization system on both infrastructures.

Two partially orthogonal application scenarios are targeted: controlled experiments and autonomous localization systems. The former scenario is focused on testbed-based research and development where fine granular control is required over the algorithm under evaluation. In contrast to simulation environments it is much more complicated to configure specific testbed states to study particular scenarios. The latter application scenario is focused on a complete localization system that can be deployed on-demand in any wireless network and runs autonomously with very limited user control; the user shall only use the provided data.

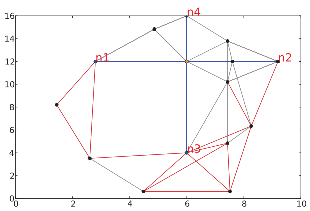
A typical DES-LOFT based localization system is split in several parts and implemented by different components. An overview is shown in Figure 1. All mesh routers run a DES-LOFT *Node Agent* which is a daemon written in ANSI-C. It provides domain specific functions, e.g., for IEEE 802.11 RSSI-based distance measurements, data structures, and the core algorithm itself. Further on, all agents are able to communicate with daemons based on two additional frameworks (see Figure 2). This is motivated by the fact that a localization system often requires crucial data from other sources and may provide data that is of benefit for other network services. The *Distributed Embedded Systems - Extensible Routing-Framework for Testbeds* (DES-SERT) [12] enables the implementation of proactive, reactive, and hybrid routing protocols. Depending on the particular routing daemon, DES-LOFT agents can use the provided topology information, link quality measurements, routing and forwarding services. Neighborhood discovery is another specific example of a service that does not have to be



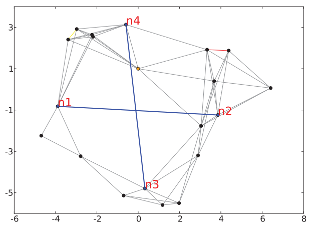
(a) Coordinate system created in phase 1. Nodes n1-n4 are selected to form the axes and the node at (6,6) is the origin.



(b) Average Percentage Error (APE) metric (see Equation 7) as a function of the number of iterations.



(c) Initial embedding of the graph from Figure 4a based on the coordinate system. Red lines mark edges that are more than two times as long as in the original graph (high contracting force).



(d) Mass-spring optimization after 60 iterations. The graph matches the original one, but is translated, rotated, and scaled.

Fig. 4: AFL Example: 20 nodes, random unit-disc graph with a maximum edge length of 4 units, 15% distance measurement error (uniform distribution)

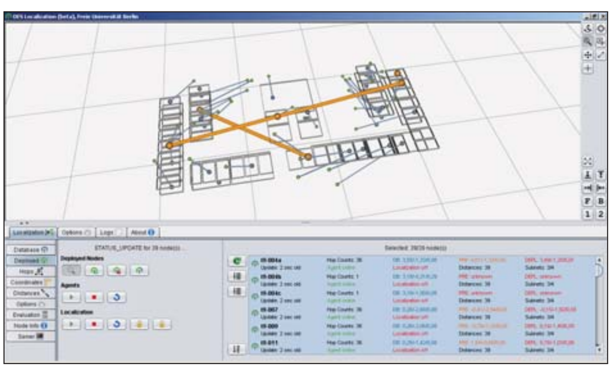


Fig. 3: DES-LOFT control and monitoring interface with an AFL localization experiment showing the axes of a coordinate system. The true and determined positions of all nodes are depicted. The control panel provides localization algorithm specific functions.

implemented in the localization daemon anymore. In return the localization can support route optimization, improve handovers, or enable geographic routing. DES-CHAN is the *Distributed Embedded Systems - Channel Assignment Framework* for the implementation of distributed channel assignment schemes [13]. As the channel assignment configuration can change over time to minimize interference, DES-LOFT can benefit from the data acquired by DES-CHAN as channel assignment daemons may have a more detailed view of the neighborhood; some nodes may otherwise not be determined as neighbors due to the use of orthogonal channels. This holistic approach enables the implementation of routing protocols that exploit multiple network interfaces to minimize interference while handovers and routing can be improved by position information. In the next step, a mobile node tracing component will extend the setup to enable probabilistic route optimization based on statistical data gathered by the three frameworks.

The second DES-LOFT component is the *proxy* application which is run on the testbed gateway server. It monitors the agents run on the routers, starts or stops them, and provides a configuration interface. All data from the agents is cached by the proxy and written to log-files which can be used for later visualization; alternatively the state of the agents can also be visualized in real time. The caching of data shall prevent constant polling from multiple entities which might interfere with an experiment. In addition, agents and proxy communicate over Ethernet to not disturb the experiment due to management packets over the wireless medium.

The DES-LOFT *control and monitoring front end* connects to the proxy which forwards all messages accordingly to the routers (and vice versa) or directly serves data from a cache. It can be run anywhere by the experimenter, even without access to the testbed and use log-files as input. As shown in Figure 3, DES-LOFT provides a 3d view of the testbed with the locations of the routers (true and estimated positions). The control functionality of the front end is always algorithm/protocol specific but customizable. For example, for AFL it is possible to configure some nodes as anchors or to manually assign initial coordinates to all nodes, skipping phase 1 of AFL altogether. A feature reduced front end is available for pure monitoring or playback of logged experiments

Overall the framework enables configuration, execution, and evaluation of localization experiments. It is a tool for holistic research in a testbed environment. DES-LOFT has proved to be suited for the implementation of distributed localization algorithms like AFL.

III. Anchor-Free Distributed Localization in the DES-Testbed

There are several proposed localization systems that apply mass-spring algorithms for anchor-free localization. We selected AFL as an example to test DES-LOFT's features and the integration in the testbed environment. Several issues were encountered during the implementation of AFL due to underspecification or abstract models used in previous simulation-based studies. To highlight selected issues, they are embedded in the text of the next sections as *remarks*. Important differences in the experiment setup are also presented in this way. The DES-LOFT implementation of AFL is called *des-afl* in the following to differentiate it from the AFL specification by Priyantha et al.

A. AFL Description

The *Anchor-Free Distributed Localization* algorithm distinguishes two separate phases: initial fold-free graph embedding and mass-spring based optimization. The authors argue that the fold-free embedding is crucial for the second phase and an improvement over the related work. In phase one, an embedding of the network is determined that is similar to the actual positions of the nodes. Six nodes participate in the creation of a relative coordinate system. The node n_0 starts the whole procedure. It determines the most distant node n_1 based on the hop-count metric. n_2 is the most distant node from n_1 and both represent one axis of the coordinate system. Subsequently, n_3 is determined which has the largest distance from n_1 and n_2. Together with node n_4, which has the largest distance to n_3 and is centered regarding n_1 and n_2, the second axis is formed. Node n_5 is selected as central node to represent the origin. Figure 4a shows an example coordinate system based on a random network. As last step of phase 1, all nodes $n_i \in [1..N]$ are assigned polar coordinates based on the hop distance to n_5 and the angle in the coordinate system:

$$\rho_i = h_{5,i} \times R \tag{1}$$

$$\theta_i = \tan^{-1}\left(\frac{h_{1,i} - h_{2,i}}{h_{3,i} - h_{4,i}}\right) \tag{2}$$

ρ is the radial coordinate, θ the azimuth, and $h_{a,b}$ the hop distance of nodes n_a and n_b. Figure 4c shows an example initial fold-free embedding.

In the second phase all nodes that can communicate are considered to be connected by springs which apply forces to them. These forces depend on the difference between the measured distance to each neighbor and their distance based on the positions in the coordinate system. The mass-spring approach will push and pull the nodes in the coordinate system to positions that minimize the remaining global energy in the network. Updated node positions always shall always lead to a lower energy. If $\overrightarrow{F_i}$ is the total force applied to node i than it will move in the direction of the force by

$$|\overrightarrow{F_i}|/(2m_i) \tag{3}$$

where m_i is the node degree of the particular node. Figure 4c shows the result after 60 iterations with the example graph. The graph matches the original one and Figure 4b shows that the error is reduced over time. Priyantha et al. showed in simulations that the fold-free embedding significantly improved the localization compared to a pure incremental scheme where local-minima were often the result due to folding.

B. AFL Implementation based on DES-LOFT

The central contribution of the AFL algorithm over the related work is the initial fold-free embedding. Despite the goal of a fully distributed algorithm, the first phase requires a specific node in the network or a central entity that controls the whole network.

> **Remark 1**: AFL is not fully distributed as some node has to start phase 1. Nodes have to be selected based on their hop distances to create axes and a node as origin.

This deficiency has already been discussed by Gotsman et al. [14]; they call AFL “far from distributed”. This statement is a little bit exaggerated as AFL just needs to be extended by an election procedure and is otherwise distributed. *des-afl* uses an approach similar to the selection of the root bridge in the spanning tree protocol [15]. When the daemons are started, a contention phase begins where each node periodically floods messages over the network containing a random number. When a node receives a message with a higher number, it ceases its broadcasts. The MAC address serves as a tie breaker when multiple nodes have selected the same random number. The random number is used as primary value to prevent that phase 1 always creates the same coordinate system in subsequent repetitions. The phase continues until only one node is left sending messages. When no other messages are received for a specific time, the node assumes to have won the contention: It becomes the node n_0 for phase 1 of AFL. The node n_0 then selects the most far away node as n_1 based on the hop distance. Although a simple task, a particular problem arises.

> **Remark 2**: AFL does not specify where the hop distances resp. the topology information originate from and how these data are learned by the node(-s).

Of course, there are two simple solutions satisfying the demand for a fully distributed system. The information can either be gathered on-demand by the localization daemon by constructing an adjacency matrix after sending messages to all other nodes or it can be extracted from a routing daemon running a link-state algorithm. In the DES-LOFT implementation, the *des-afl* node agents can either be configured via the front end which can provide deterministic topology information from a database to allow repeatable experiments or acquire the data from our implementation of the *Optimized Link State Routing* (OLSR) protocol [16] based on DES-SERT.

A particular problem with the node selection algorithm became obvious in the first experiments run in the testbed, the virtual networks, and in simple graph based simulations similar to the approach of Priyantha et al. While the first axis is created fine, the second (and third) axis was in several cases not as orthogonal as it could have been, i.e., the angle between two axis was not close to 90°. This resulted in an initial embedding of the nodes which lead to folding in the second phase. The problem is due to the applied heuristic to select candidate nodes for the axes, which especially showed up in networks with low node degree and a non-uniform distribution.

> **Remark 3**: The heuristic to select nodes for the 2nd (and 3rd) axis ignores better candidate nodes.

For example, n_4 the node to complete the second axis shall minimize the distances to n_1 and n_2 and maximize the distance to node n_3:

$$minimize \quad (|d_{1,4} - d_{2,4}|) \tag{4}$$

$$maximize \quad (|d_{3,4}|) \tag{5}$$

The second constraint (5) is the tie breaker when there are multiple candidates. Unfortunately, this approach will ignore nodes that have an inconsiderable larger value for the first constraint but a higher value for the second. This is suboptimal,

Dimensions	Algorithmn	Distance Error	Anchors	#
2d	MIT-Standard	0%	none	1
			5	2
		5%	none	3
		10%	none	4
	MIT-Enhanced	0%	none	5
			5	6
		5%	none	7
		10%	none	8
3d	MIT-Standard	0%	none	9
			5	10
		5%	none	11
		10%	none	12
	MIT-Enhanced	0%	none	13
			5	14
		5%	none	15
		10%	none	16

TABLE I: Overview of all 16 individual AFL experiment configurations used in the testbed. The right-most column contains an ID that is used to refer to particular configurations.

as the initial embedding will often lead to better results, when n_4 is far away from n_3 but not perfectly centered regarding n_1 and n_2. A weighting function is used in *des-afl* that considers both constraints together:

$$maximize \quad \left(\frac{|d_{3,4}|}{|d_{1,4} - d_{2,4}|} \right) \tag{6}$$

An equivalent heuristic is applied for the creation of the third axis. When there are still multiple candidates, which can easily happen as the hop distances are discrete values, a random one is selected from the set.

Another issue might not have shown up in simulations of large networks. The node selection algorithm ignores that nodes can be selected multiple times which again is common in small or non-uniform deployed networks.

> **Remark 4**: The original algorithm, which we call MIT-Standard, does not ensures that different nodes are selected to create the axes.

In the DES-Testbed this happened in a noticeable number of cases, where an axis was created using the same node twice or two axes shared the same node. While the second scenario is not fatal, it often does not result in a fold-free embedding as one or more quadrants of the Cartesian coordinate system are empty and the nodes are crammed together in another one. We modified MIT-Standard to solve these issues and called it MIT-Enhanced. Both variants are used in the experiments for a comparison.

des-afl also applies a jitter for the messages that are periodically sent by the mesh routers. This feature had to be included to prevent a “wobbling“ movement of the routers. In simple graph-based simulations, mass-spring algorithms are often run in a synchronous way, i.e., in each iteration of the algorithm, each node sends its position to all neighbors and all nodes update their positions at the same time. This is not the case in real networks.

> **Remark 5**: Without a jitter for the dissemination of position information, the whole network can be perturbed.

While this was a rare case and over the time the network settled down, it can lead to folding depending on the number and order of received messages from neighbors, e.g., a particular node could be pushed beyond one of its neighbors from which the packet was lost or a large group of nodes ”pushes“ another node. The effect is also heavily dependent on the order in which the localization is activated on the nodes, as well as the packet delivery ratio of the links and the interval to update the position. Unfortunately, there is no specification of the intervals to disseminate the current position and the interval to update it in the original publications [2], [3] and also limited information in the related work where similar algorithms were used, e.g., [10], [17], [18]. We assume that this is due to the above mentioned synchronous approach and the used network model.

> **Remark 6**: Important, performance determining intervals are not specified for AFL.

In real world networks where packets can be lost on the medium, the position update should have an equal or higher value than the position dissemination interval so that one or more messages have been received from each neighbor until the position update.

IV. Experiment Setup and Evaluation

A set of 16 experiments was run in the initial study about the applicability of AFL in real world networks. Table I gives an overview of the configurations. We started with 35 nodes that are deployed in the computer science building as this part of the testbed is the most uniform and dense. This is a much more simple scenario for AFL than starting with the whole network that shows low rigidity because of the links between the three buildings. All routers used their three network interface cards for communication on orthogonal channels to achieve a high node degree. The node degree can not be specified accurately as it depends on the channel, data rate, and external factors but is on average around 7, ranging from 1 up to 23. This numbers seem very high but it has to be considered that several links have a very low packet-delivery-ratio. Compared to the simulation based study of AFL, there are some other differences in the setup.

Priyantha et al. used random networks with 30, 100, and 300 nodes with an average node degree of four, eight, and twelve in a first experiment series to show that fold-free embedding is crucial for a mass-spring based localization. Another experiment series to evaluate the influence of ranging errors used 250 nodes (on average) with different node degrees. Nodes within a particular range R could communicate bidirectionally and without losses. The latter fact is not explicitly mentioned but there is no radio propagation model specified to assume otherwise. In contrast, our network size is fixed and its topology caused by many factors. The mesh routers use multiple wireless network cards and the links have different qualities thus the node degree is very dynamic. The link quality is not constant and links can be unidirectional.

Remark 7: AFL has to deal with packet loss, link asymmetry, and varying node degrees in real world setting. Which was not considered in simulations.

The nodes are distributed over four floors with two larger clusters in the west and east wing. The shape of the building resembles an 'E' (see Figure 3) with its two large atria. This poses a challenge for AFL as the nodes are not uniformly deployed and the graph is not globally rigid. The nodes are also deployed in a three dimensional space. One half of experiments ignores the height and creates a two dimensional coordinate system and the other half tries a three dimensional localization. All nodes participated in both scenarios in the experiments, not just routers on the same floor.

Remark 8: The simulations in [2] considered only nodes deployed in 2d space.

In the testbed, the three dimensional scenario can result in a poor coordinate system as there are only three levels; yet we argue that real world networks will probably never have a perfect cubicle form. The goal of these experiments is to show what AFL can achieve such a scenario.

Each individual experiment is repeated 20 times and run for a duration of 2 min which results in a total experiment time of about 11 hours. The metrics (see Section IV-A) are measured at a sample rate of 30 s. The duration represents only the time of the mass-spring optimization in phase 2. Both coordinate system algorithms were used: standard and enhanced version.

AFL uses the measured distance between any nodes during phase 2 for the calculation of the forces of the springs. Accurate distance measurements without any additional hardware like ultra sonic transmitters and receivers or Time-Of-Arrival or Time-Difference-Of-Arrival capable transceivers are difficult to achieve. We assume in the application scenario where AFL is used in some specific network that no additional hardware besides the IEEE 802.11 transceiver is available. This leaves the receive signal strength as the only option for a range-based localization. Unfortunately, the receive signal strength is well known to allow only limited distance estimation accuracy [19], [20] especially in indoor locations due to multipath propagation, shadowing, and groups of people acting as black bodies [21]. In our first study we focus on the applicability, behavior, and performance of AFL in real world scenarios. The distances are not measured based on the RSSI but known a priori so that one source of error is under full control. We simulate the inaccuracy of measurements and randomness of the path loss by introducing artificial errors. A uniform distribution is assumed for the measured distance as this is on par with the experiment setup by Priyantha et al. The considerations for subsequent studies are discussed in Section VI. Based on simulations and the experiments in the testbed, we expect that AFL will enable localization in the discussed application scenarios even if only few discrete distances can be distinguished, e.g., "near", "far away", or "medium distance".

The timing parameters that were discussed in the previous section are as follows. Each node broadcasts its current position every 1 second. The same interval is used to calculate a new position based on the experienced forces.

A. Metrics

Three metrics are applied to evaluate the experiment data. *Average Percentage Error of Distances* (APE) [22] describes the average error of all node-to-node distances:

$$APE = \frac{\sum_{i,j:i \neq j} |\hat{e}_{ij}| * 100}{N(N-1)} \tag{7}$$

$\hat{e}$ is the normalized error between the true and determined distance of two nodes i and j. N is the number of nodes.

The *Average Euclidean Distance* (AED) [22] metric represents how good the localization matches the real topology by averaging over the absolute difference d_k for each node.

$$AED = \frac{\sum_{k=1}^{N} |d_k|}{N} \tag{8}$$

AED is used only to evaluate the experiments with anchor nodes because there is otherwise no fair measure as the coordinate system is relative.

The *Global Energy Ratio* (GER) introduced in [2] represents the structural error in the determined embedding in regard to the true topology. The authors state that a simple average position error is not sufficient to capture the accuracy of localization algorithms; thus GER considers the average error of all node-to-node distances. Edge length errors as well as structural errors of the graph influence the result.

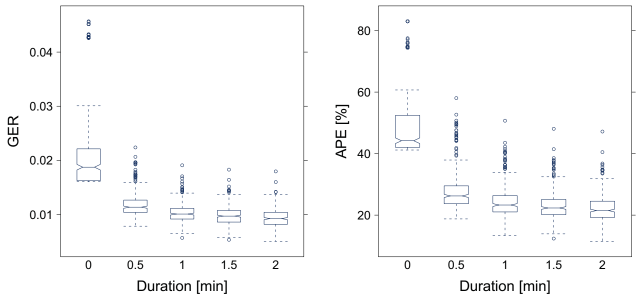
$$GER = \frac{\sqrt{\sum_{i,j:i \neq j} \hat{e}_{ij}^2}}{N(N-1)} \tag{9}$$

Priyantha et al. assume a local or global minimum, when the GER value has not been reduced by more than 0.1% in subsequent iterations. Our experiments run for a fixed time as packets may have been lost on the medium and a local minimum might not have been reached yet.

For all three metrics lower values represent better results. The metrics can be used to evaluate different localization algorithms and parameter settings and show that the localization converges over time but there is one specific downside. All of the metrics have to be calculated in a centralized manner and require global information. Additionally, the true positions have to be known. Thus neither metric can be used by an individual node in a real world application scenario to evaluate the accuracy of its determined position. Further on, the node can not easily tell if the algorithm has terminated as it has to learn about several subsequent APE, AED, or GER values to apply the approach by Priyantha et al.

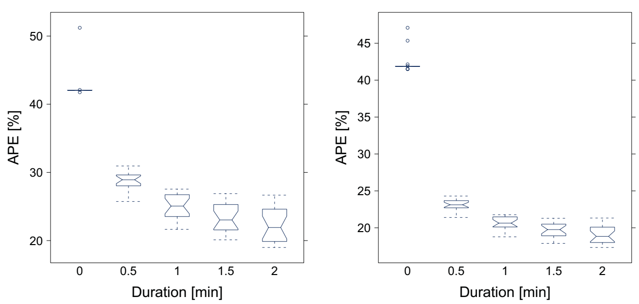
Remark 9: The metrics allow an evaluation but are inadequate for real world use.

A possible solution for this issue is discussed as future work in Section VI.



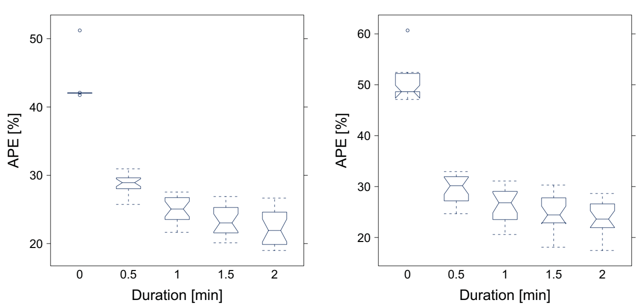
(a) GER as a function of time (b) APE as a function of time

Fig. 5: Box-plots of all measured data points showing the overall behavior of AFL in all 16 experiment scenarios. The AFL algorithm converges in the experiments.



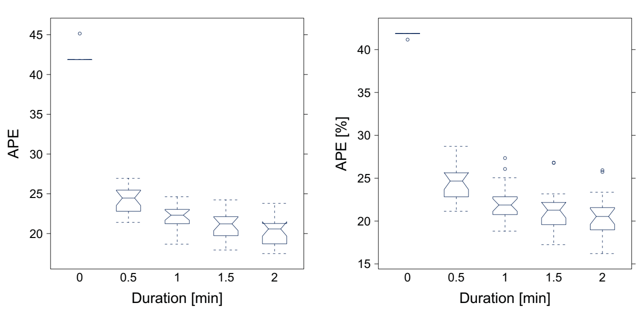
(a) Scenario 1 (MIT-Standard) (b) Scenario 5 (MIT-Enhanced)

Fig. 6: Influence of the coordinate system algorithm for the initial embedding in phase 1 on the overall AFL performance.



(a) Scenario 1 (2d) (b) Scenario 9 (3d)

Fig. 7: Localization in two and three dimensions with the MIT-Standard coordinate system algorithm.



(a) Scenario 7 (5% error) (b) Scenario 8 (10% error)

Fig. 8: AFL with different distance measurement errors and MIT-Enhanced.

B. Evaluation

In total, 1600 data points have been recorded for the 16 individual experiments. Due to the page limit we have selected the most interesting graphs. Figure 5 shows box-and-whisker plots for the GER and APE metric for all accumulated data points independent of the particular scenario. The box marks the upper and lower quartile while the whisker extend up to 1.5 times of the *inter quartile range* (IQR). The median is shown with a 95% confidence interval (notches around the median). This gives an insight in the overall behavior of AFL in the different experiment scenarios. It can be observed that the GER and APE values decrease over time as the nodes optimize their positions and thus the algorithm converges. The graphs let us assume that most significant part of the optimization in phase 2 happens in the first minute (first 60 iterations) while there is less contribution in the second half of the experiment. This has been expected as the total energy in a mass-spring system should decrease reciprocal with the number of iterations.

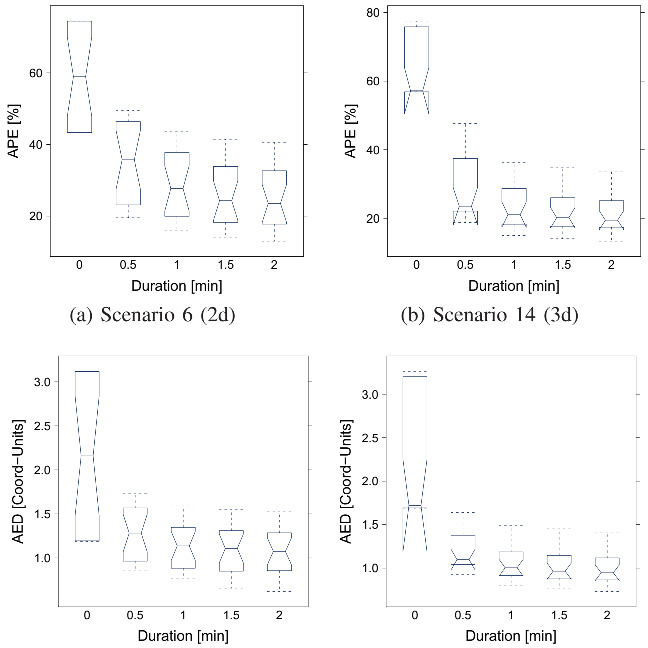
First of all we compare the results of the two discussed coordinate system algorithms. Figure 6 shows the APE metric for scenarios 1 and 5. The enhanced version leads to better results and faster convergence. This could be observed in all experiments and also with the GER metric. Surprisingly, the variance of the data points at the start of the experiment, after the initial embedding, is very low. This was noticed in most but the experiments that used anchor nodes.

The results of the localization in a two- or three-dimensional space are shown in Figure 7. The initial embedding was always worse in 3d regarding the GER and APE values but at the end of the experiments the influence of the additional axis is nearly gone. We assume that the third dimension introduces only a minor error to the localization compared to the error of the mass-spring algorithm.

When the distance measurements have an 5% or 10% error, it does not seem to significantly influence the results as shown in Figure 8. We suspect that many of the errors are clearing each other, e.g, when there is a measurement error of x for the distance to some other node, there is probably another one where the distance has an error of $-x$. Nevertheless, a 20% error remains on average for the node-to-node distances.

As last, we evaluate how well AFL can localize nodes when there are 5 anchor nodes. Figure 9a and Figure 9b show that the APE metric data points have a higher variance compared the equivalent experiments configurations without anchors. The two-dimensional scenario is more affected from this than the three-dimensional localization. The AED values shown in Figure 9c and Figure 9d are given in coordinate system units, where one unit is $11.5\ m$. Considering the goal of a coarse granular distributed anchor-free localization, *des-afl* is close to the desired result.

We learned from subsequent experiments that AFL shows better results, i.e., faster convergence and less chance of folding when node n_5 (the origin) is selected as anchor node. This is not an effect of the known position but that the node does not change its position.



(a) Scenario 6 (2d) (b) Scenario 14 (3d)

(c) Scenario 6 (2d) (d) Scenario 14 (3d)

Fig. 9: AFL with anchor nodes, different dimensions, and MIT-Enhanced. The APE and AED metric are shown as a function of time.

V. Related Work

Several other localization systems have been proposed. We give some examples for systems with focus on mass-spring optimization algorithms. The applicability in our scenario and improvements for AFL are briefly discussed. All addressed papers in this section are ordered by their year of publication.

Bulusu et al. describe a range-free localization system that uses anchor nodes which periodically broadcast their positions [23]. Mobile nodes apply trilateration to calculate positions based on the assumption that the radio ranges are equidistant. Experiments in a testbed setup showed results similar to theoretical considerations but the anchor nodes have to be sufficiently dense. While anchor based approach can achieve a high accuracy, they cannot be easily deployed on demand without prior calibration and position assignment.

Savarese et al. propose a two phase range-based localization system [17]. Initial coordinates are assigned with Hop-TERRAIN, an algorithm similar to DV-Hop [24], based on hop counts to anchor nodes. In the second phase, an iterative optimization of the positions takes place. Nodes with few neighbors or in a "poor constellation" get a low confidence value assigned and thus contribute less to the position optimization of their neighbors. AFL could be improved by evaluation the confidence of position informations from neighbors. A similar classification approach is discussed in the future work section.

MDS-MAP uses *Multidimensional scaling (MDS)* [25] and is a range-free approach that requires only network topology information; further information may be used to refine the localization. The largest linear projection of the adjacency matrix of the network graph is calculated by a specific node and positions assigned appropriately. It is a simple approach but has also problems when the graph is not rigid or sparsely connected, especially in the center. MDS-MAP is comparable to a range-free version of AFL where all nodes are assumed to be deployed with distance 1.

Moore et al. propose localization based on a mass-spring approach to improve the routing in mesh networks [10]. Their virtual location scheme uses $N+1$ dimensions to locate nodes in an N dimensions, stating that the extra degree of freedoms allows to solve local minima. Unfortunately, the interaction of the localization and a specific routing protocol is not discussed in this publication. The proposed system is one of the goals of *des-afl* but it shall be extended by a database of traces to improve the routing for mobile nodes.

Sweeps [26] is a class of algorithms for localization in sparse networks (average node degree < 5), where three nodes have fixed positions. All other nodes run an iterative trilateration algorithm and are initially considered *unlocalized*. Over time, more and more nodes will become *finitely localized nodes*, i.e., a finite set of possible positions has been learned. This is the case, when the trilateration can use positions from two *finitely localized nodes*. When all nodes have only finite possible locations, the number of solutions is reduced. *Sweeps* is a centralized approach and requires a globally rigid network graph to result in a unique localization.

Lederer et al. propose an approach to resolve the problems of MDS-MAP [18]. The algorithm selects nodes on the boundaries, called landmarks, that are used to construct a Voronoi diagram to derive a Delaunay graph which solves the flip ambiguity problem of many localization approaches. The Delaunay graph is rigid if the landmarks are sufficiently dense. The landmarks are assigned positions based on the graph which are than refined by a mass-spring algorithm. All other nodes then apply trilateration to calculate their positions. While an interesting and promising approach, further research is required to study if the assumptions about the graph match with real world networks.

The *collaborative localization scheme from connectivity* (CLFC) for WSNs is another two phase approach [27]. The authors state that CLFC achieves faster convergence than AFL. Initial coordinates are assigned with DV-Hop [24], followed by collaborative refinement. Anchor-nodes periodically flood messages over the network and all nodes learn the minimum hop distance. The average length of all links (called hop correction) is determined by the beacons and broadcasted to their neighbor nodes. The nodes then triangulate their positions with three beacons. As last, a position refinement is run by each node based on the positions of its k-hop neighbors ($k \in \{2,3\}$). An energy value is calculated and a new position determined. In contrast to AFL, CLFC is a range-free approach as no distances are measured. The consideration of k-hop neighbor positions could be one way to solve folding in AFL.

Localization Algorithm Based on a Spring Model (LASM) [28] is another mass-spring based localization system. It is very similar to AFL but relies on anchor nodes and has no initial fold-free embedding. Three called "patches" are proposed for LASM to improve the localization. The first patch handles nodes that have reached a mostly force balanced position (the total force is close to zero) but there is at least one neighbor for which the force is not close to zero. It can be assumed that the node is mispositioned (maybe due to folding). A random position is assigned to resolve the issue. The second patch assigns trust values to each node. Nodes with lower trust contribute less to the mass-spring process. As last, the third patch handles nodes that join or leave the network. Trilateration is applied to assign initial coordinates before the node starts the mass-spring algorithm. Nodes that leave the network of move away require no special handling. The latter will calculate new positions by trilateration. The patches handle scenarios that are not considered in the original AFL specification. Especially the first one might be able to resolve local minima.

VI. Future Work

In our first study we focused on the applicability of the AFL algorithm in real world networks. As we showed, several issues could be resolved, but some work is still required for a complete localization system and there are ways to increase the accuracy.

Many localization approaches have problems in networks that have sparse areas. The DES-Testbed spans multiple buildings and there are less links in between. Further experiments are required how anchor-free localization can be achieved in non-rigid graphs, e.g., using k-hop position information and analyzing the traces of mobile nodes.

Term 3 requires more attention. While it is comprehensible that nodes with a high node degree shall move less compared to sparsely connected ones, this optimization of the mass-spring algorithm has only been empirically determined by Priyantha et al. for graphs with a random uniform distribution. Further on, a node may have n neighbors but might not receive position updates from all of them until a new position is determined. In this case, the position of the node is not updated in the optimal direction or even in the wrong direction. Both scenarios will increase the time until the algorithm converges and increase the chance of local minima.

The neighborhood discovery is a problem itself. Priyantha et al. and other authors assume a graph model where the network topology is obvious and only nodes in a particular range can communicate. In the experiments we were able to observe some cases when a node could suddenly receive a message from a far away node. This raises the question if this node should be considered a neighbor and if it should contribute as much to the total force as neighbors that are close by. In the next version of *des-afl* we will use a sliding window approach where the number of received messages determines a weight for each neighbor to diminish temporal deformations of the network due to this phenomenon.

Inaccurate RSSI-based distance measurement could be improved by considering the number of frames that show a corrupted frame check sequence. Additionally, the number of retransmissions that were necessary to deliver a packet on the data link layer can be incorporated for a more accurate measurement. The required functions are currently being implemented for DES-SERT.

The antenna characteristic should be accounted for in both phases of the algorithm: coordinate system creation and position optimization. Antennas are no isotropic radiators and a common half-wave dipole antenna shows a radiation pattern that is usually lobe-like. Considering that all antennas are orthogonally aligned to the earth surface we still have to care for nodes on different floors. The received energy will be different for nodes that are on the same flow and less for nodes in different floors, even when the distance is exactly the same.

As discussed, AFL requires topology information for the initial embedding. *des-alf* uses the adjacency matrix that is provided by the OLSR daemon. Hop-count was applied as metric to determine the distances of all nodes. This might lead to a false view of the network as several links might show poor packet delivery ratios. The influence of this problem on the result of AFL requires further examination as well as if advanced link/path metrics, e.g., ETX [29] will improve the localization and avoid folding.

As last, for a complete localization system it is required that nodes are able to evaluate the accuracy of the position estimation. As discussed in Section IV-A, none of the three metrics can be calculated by a single node as global information and the true positions are required. At least the first problem could be resolved by a consensus approach [30] giving feedback if the global energy has reached a particular level. Nevertheless, the question remains, how to determine if the mass-spring algorithm has converged to a global energy minimum especially considering the inaccuracy of distance measurements. The classification of the measured distances in few discrete classes, as discussed in Section IV, could ease the distributed convergence detection as smaller errors in the distance measurements will not necessary increase the energy in the network.

VII. Conclusion

In this paper we introduced the DES-LOFT localization framework and discussed the cooperation and integration of multiple services in a wireless network into a holistic system. The Anchor-Free Distributed Localization Algorithm (AFL) was introduced as one example implementation based on DES-LOFT. We highlighted selected issues with the original specification and proposed improvements that enable the application of AFL in real world networks. As a first study in the DES-Testbed showed, AFL can be used as an distributed localization system and the algorithm converges. The accuracy of the anchor-free approach is good enough to provide localization and navigation services in the discussed scenarios using the already deployed infrastructure. Improvements and considerations for subsequent studies and versions of our AFL implementation have been discussed.

Acknowledgment

This work has been partially sponsored by the OPNEX and WISEBED projects of the European Union.

References

[1] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, “Range-free localization and its impact on large scale sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 4, no. 4, pp. 877–906, 2005. [Online]. Available: 10.1145/1113830.1113837

[2] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, “Anchor-free distributed localization in sensor networks,” MIT Lab for Computer Science, Tech. Rep. MIT-LCS-TR-892, April 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.5104

[3] ——, “Poster abstract: anchor-free distributed localization in sensor networks,” in *SenSys ’03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 340–341. [Online]. Available: http://doi.acm.org/10.1145/958491.958550

[4] T. M. J. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement,” *Softw. Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991. [Online]. Available: http://dx.doi.org/10.1002/spe.4380211102

[5] K.-B. Lee and J.-H. Kim, “Mass-spring-damper motion dynamics-based particle swarm optimization,” in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 2348–2353. [Online]. Available: http://dx.doi.org/10.1109/CEC.2008.4631111

[6] J. Christensen, J. Marks, and J. T. Ngo, “Automatic motion synthesis for 3d mass-spring models,” *The Visual Computer*, vol. 13, pp. 20–28, 1997, 10.1007/s003710050086. [Online]. Available: http://dx.doi.org/10.1007/s003710050086

[7] I. F. Akyildiz and X. Wang, *Wireless Mesh Networks*. Wiley & Sons, March 2009.

[8] G. Wittenburg and J. Schiller, “A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks,” in *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom ’08, Middleware Support for Pervasive Computing Workshop)*, Hong Kong, Mar. 2008, pp. 548–553. [Online]. Available: http://dx.doi.org/10.1109/PERCOM.2008.43

[9] R. Jain, A. Puri, and R. Sengupta, “Geographical routing using partial information for wireless ad hoc networks,” *IEEE Personal Communications*, vol. 8, no. 1, pp. 48–57, February 2001. [Online]. Available: http://dx.doi.org/10.1109/98.904899

[10] N. Moore, A. Sekercioglu, and G. K. Egan, “Virtual Localization for Mesh Network Routing,” in *IASTED Conference on Networks and Communication Systems*, 2005.

[11] M. Günes, B. Blywis, and F. Juraschek, “Concept and Design of the Hybrid Distributed Embedded Systems Testbed,” Freie Universität Berlin, Tech. Rep. TR-B-08-10, 2008.

[12] B. Blywis, M. Günes, F. Juraschek, P. Schmidt, and P. Kumar, “DES-SERT: A framework for structured routing protocol implementation,” in *IFIP Wireless Days conference (WD’09)*, Paris, France, 12 2009.

[13] W. Si, S. Selvakennedy, and A. Y. Zomaya, “An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks,” *Journal of Parallel and Distributed Computing*, vol. In Press, Corrected Proof, pp. –, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6WKJ-4XCJ4MF-2/2/695f3370fc36df4a5b96b0de4d2782c1

[14] C. Gotsman, , C. Gotsman, and Y. Koren, “Distributed graph layout for sensor networks,” in *In Proc. Internat. Symposium on Grpah Drawing*, 2004, pp. 273–284. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-31843-9_28

[15] *802.1D, IEEE Standard for Local and metropolitan area networks, Media Access Control (MAC) Bridges*, IEEE Std., Rev. IEEE Std 802.1D-2004.

[16] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626 (Experimental), Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[17] C. Savarese, J. M. Rabaey, and K. Langendoen, “Robust positioning algorithms for distributed ad-hoc wireless sensor networks,” in *ATEC ’02: Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2002, pp. 317–327.

[18] S. Lederer, Y. Wang, and J. Gao, “Connectivity-based localization of large scale sensor networks with complex shape,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 789–797. [Online]. Available: http://dx.doi.org/10.1109/INFOCOM.2008.130

[19] E. Elnahrawy, X. Li, and R. P. Martin, “The limits of localization using signal strength: a comparative study,” 2004, pp. 406–414. [Online]. Available: http://dx.doi.org/10.1109/SAHCN.2004.1381942

[20] N. Patwari, A. O. H. III, J. Ash, R. L. Moses, S. Kyperountas, and N. S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005. [Online]. Available: http://dx.doi.org/10.1109/MSP.2005.1458287

[21] S. Shrestha, S. Shrestha, A. Lee, J. Lee, D.-W. Seo, K. Lee, J. Lee, S. Chong, and N. H. Myung, “A group of people acts like a black body in a wireless mesh network,” in *Proc. IEEE Global Telecommunications Conference GLOBECOM ’07*, A. Lee, Ed., 2007, pp. 4834–4839.

[22] E. Sokolovski, “An Anchor-free Localization Technique for Distributed Networks,” Master’s thesis, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, Lehrstuhl für Informatik IV, Aachen, Germany, December 2007.

[23] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, October 2000. [Online]. Available: http://dx.doi.org/10.1109/98.878533

[24] D. Niculescu and B. Nath, “Ad hoc positioning system (aps),” in *Global Telecommunications Conference*, vol. 5, November 2001, pp. 2926–2931. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2001.965964

[25] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, “Localization from mere connectivity,” in *MobiHoc ’03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2003, pp. 201–212. [Online]. Available: http://doi.acm.org/10.1145/778415.778439

[26] D. K. Goldenberg, B. D. Anderson, P. Bihler, M. Cao, J. Fang, A. S. Morse, and Y. R. Yang, “Localization in sparse networks using sweeps,” in *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2006, pp. 110–121. [Online]. Available: http://doi.acm.org/10.1145/1161089.1161103

[27] J. Wang, L. Huang, X. Li, H. Huang, and J. Li, “A collaborative localization scheme from connectivity in wireless sensor networks,” in *Wired/Wireless Internet Communications*, ser. Lecture Notes in Computer Science, J. Harju, G. Heijenk, P. LangendÃűrfer, and V. Siris, Eds. Springer Berlin / Heidelberg, 2008, vol. 5031, pp. 213–223. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68807-5_18

[28] W. Chen, T. Mei, M. Q. Meng, H. Liang, Y. Liu, Y. Li, and S. Li, “Localization Algorithm Based on a Spring Model (LASM) for Large Scale Wireless Sensor Networks,” *Sensors*, vol. 8, no. 3, pp. 1797–1818, March 2008. [Online]. Available: http://dx.doi.org/10.3390/s8031797

[29] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom ’03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003, pp. 134–146.

[30] F. Fagnani and S. Zampieri, “Randomized consensus algorithms over large scale networks,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 634–649, May 2008. [Online]. Available: http://dx.doi.org/10.1109/JSAC.2008.080506