Estimating Distances Using Neighborhood Intersection

Carsten Buschmann, Dennis Pfisterer, Stefan Fischer
University Lübeck, Institute of Telematics
Ratzeburger Allee 160
23538 Lübeck
{buschmann | pfisterer | fischer}@itm.uni-luebeck.de

Abstract

Information about distances to other nodes in wireless sensor networks has proven advantageous not only for location discovery but is also helpful for context aware applications in general. In this paper we present a novel approach to distance estimation that does neither depend on special hardware nor on unreliable measurements of physical wireless communication properties. Instead, it is inspired by the observation that distant nodes have fewer neighbors in common than close ones and calculates distances from intersection cardinalities of sets of adjacent nodes. We discuss related work and present the new approach in detail including its mathematical foundations. A simulative performance analysis comprising different scenarios shows that our scheme yields competitive results.

1. Introduction

In the area of pervasive computing, many applications need information about their context to function efficiently, i.e. they need to know about their environment and other devices in the network. This is especially true for wireless sensor networks [1]. They consist of tiny, battery-powered computers that are equipped with sensors. Wireless ad-hoc techniques enable applications where hundreds of sensors are dispersed for monitoring physical phenomena like temperatures within an area of interest in near future. Such networks are envisioned to consist of cubic millimeter sized devices enabling smart dust or smart paint [2].

A configuration of devices prior to their usage is unmanageable [1]. Not only does the number of devices inhibit manual configuration or intervention during operation, sensor networks are designed to operate in inaccessible regions and are e.g. foreseen to be deployed by dropping them out of an airplane. Because devices may be unreachable, they must be able to (cooperatively) acquire the necessary information by themselves.

Proximity is one of the most important context modalities. It is not only an intrinsically interesting property of environmental elements and other wireless nodes, but also the key foundation for location estimation schemes [3]. These algorithms assume that a small number of devices, so called anchors, do already know their location. The other nodes try to estimate their distances to the anchors and then infer their position through multilateration.

Many different systems for distance estimation (often also called ranging) have been developed. Nearly all of them employ a sender-receiver-scheme: one node emits some kind of signal, the other uses a special receiver to measure physical signal properties like attenuation or time of flight. These techniques will be discussed in detail in the next section. If no direct distance estimation is possible, multi hop accumulation of distance estimates along the route from anchor to the node is a common workaround (compare e.g. DV-Distance in [4]).

In this paper we present the neighborhood intersection distance estimation scheme (NIDES). This novel approach to distance estimation does not rely on special hardware or unreliable measurements of physical signal properties. Instead, it computes the distances from intersection cardinalities of sets of adjacent nodes. In other words, its operation is based on the observation that nodes that are located close to each other share many of their neighbors whereas distant nodes share only few neighbors.

The remainder of this paper is structured as follows. In the next section we present related work. We review different distance estimation techniques and discuss their advantages and disadvantages. In Section 3 we introduce the neighborhood intersection distance estimation scheme. Then, we elaborate on its mathematical foundations. A discussion of properties and factors of influence follows. In addition, we point out how NIDES can be implemented as a distributed network protocol. Section 4 reports the results of the simulative performance evaluation. The paper is concluded by a summary and directions for future work.

2. Related Work

A number of techniques to acquire distance estimates between nodes have been developed in the past. Their underlying principle is to measure physical properties in order to derive distances. Given a model that correlates the

1-4244-0681-1/06/$20.00 ©2006 IEEE

physical measurements to distances, each node can estimate the distance to other nodes. The accuracy of the proposed techniques depends on the characteristic and suitability of the utilized model. Besides theoretical considerations focusing on the accuracy of the approach, applicability in real world deployments plays a major role. Issues like special requirements to the hardware, form factor, cost, vulnerability to environmental fluctuations and metering precision are equally vital for the selection of a distance estimation technique. In the following, we will provide an overview of existing distance estimation systems, describe their properties and pinpoint problematic issues.

One of the first approaches for distance estimation in ubiquitous computing systems was proximity detection. Systems that use this kind of distance estimation technique are the Active Badge [5] and the Hybrid indoor navigation system [6]. They both use infrared light to periodically transmit beacons with unique identification. The recipients of these beacons - either the fixed infrastructure or the mobile devices - deduce proximity to the beacon since the range of infrared signals is limited to a few meters of distance. With increasing number of beacons a fine grained resolution is achievable. The downside of this distance estimation technique is the very high deployment effort and the need for a backbone infrastructure.

To enable infrastructure-less distance estimations, a broad range of techniques have been proposed, deployed and researched in the past, which can be classified into the following three categories: approaches based on (differential) time of flight, radio signal strength and connectivity.

They rely on different hardware and measurement techniques to autonomously estimate distances to other nodes leading to different accuracy characteristics and requirements.

The measurement of the *time of flight (ToF)* of a signal is a robust method to estimate distances. However, measurements require a tight time synchronization of sender and receiver. Systems like Calamari [7], Cricket [8], AHLoS [9] and others [10, 11, 12, 13] use a technique called ”differential time of arrival” (DTOA) to avoid complex time synchronization. They send out two signals propagating with different speeds and measure the difference in time of arrival. If both signal propagation speeds are known, a distance can be derived from the delta of arrival. The majority of the proposed schemes use acoustic or ultrasonic sound and radio frequency transmissions as signaling technologies. The raw difference measurements tend to yield average estimation errors of about 74% [7]. Yet, quite good accuracies can be achieved by post-processing of the measured data with techniques like noise canceling, digital filtering, peak detection and calibration. While some authors report average range estimation errors of 10% [7], others claim an error of about 1% at a maximum range of 9 m [12]. In [11] the authors report the increase of the maximum range to 12 m with an error of 0.5%.

While these systems yield low estimation errors they have severe limitations that confine their general applicability in real world deployments. DTOA systems inherently require an extra actuator and detector pair which increases cost, size and energy consumption of the hardware platform. Furthermore, the use of acoustic and ultrasonic sound implies additional constraints. Locally fluctuating temperatures can cause a different speed of sound, obstacles in the line-of-sight render the distance estimation impossible and the angle dependent irradiation limits the omni-directional use of the system. The maximum range within which a distance estimate can be obtained is therefore bounded by the range of the acoustic and ultrasonic transmitters. They are typically able to cover 3–15 m [9] which is only a fraction of the communication range of radio frequency transmitters.

To overcome the limitations of systems using (differential) time of flight, different schemes have been developed using the radio interface itself to infer distances to other nodes. These approaches can be divided into two categories: range-based and range-free schemes. Range-based distance estimates use radio signal attenuation properties to model the distance between nodes as a function of the *received signal strength indicator (RSSI)*. Systems that rely on the RSSI as input parameter [14, 15, 16] tend to be quite accurate for short ranges if extensive post-processing is employed, but are imprecise beyond a few meters [17]. At ranges of 20 m and below, simple radio chips lead to errors of 50% to 100% while more sophisticated ones enable distance estimates exhibiting errors of about 10% after calibration [11]. The uncertainty of the radio propagation imposes problems like multi-path propagation, fading and shadowing effects as well as obstacles in the line-of-sight. These effects complicate the development of a consistent model [9]. As a result, systems relying exclusively on RSSI values remain mediocre distance estimators [18]. An improvement is presented in [17] where *radio interferometry* techniques are used to achieve an average localization error of 3 cm and a range of up to 160 m with a largest error of approximately 6 cm (which is about 0.04%). However, radio interferometry seems to be susceptible to the effects of shadowing, reflection and multi-path propagation. The downsides of the approach are the high computational complexity of the algorithm and that it requires special features of the radio chip. In addition the accuracy is achieved by long lasting measurements, which renders the approach impractical for mobile scenarios. The so-called *range-free distance estimates* can only measure distances well above the communication range because they rely on hop-counts [4, 14, 19, 20]. They inherently assign the same distance value to multiple nodes despite differences in distance. Hence, the resulting estimates are rather coarse grained. Below the communication range, they are limited to qualitative proximity detection.

In [21], Cartigny et al. present the idea to detect far away neighbors by comparing neighborhood lists. However, by determining the percentage of shared neighbors they do not estimate distances. Instead, that fraction influences the probability of forwarding a received flooding message. The fewer neighbors the sender and the receiver share, the higher the forwarding probability is. Thus, distant nodes have a higher probability to forward packets, increasing the coverage per packet sent. However, the authors limit usage of the neighborhood list to forwarding purposes.

3. Neighborhood-Based Distance Estimation

In contrast to most of the aforementioned ranging techniques, the neighborhood intersection distance estimation scheme (NIDES) does not require any specific hardware. Instead it makes use of the broadcast property of the wireless communication facility that is available anyway.

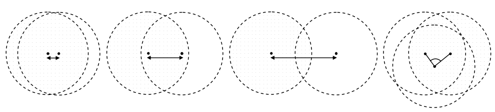


Figure 1. Neighborhood intersection at different distances and angles.

The basic idea behind NIDES is straightforward: if one assumes that the communication range of a node is represented by a circle with radius r around it, then the distance between two nodes can be inferred from the intersection of the circles (see Figure 1). If the nodes are very close to each other, the intersection area has nearly the size of the full circle (a) whereas longer distances lead to smaller intersections (b). Note that in (c), the distance is bigger than r and hence the nodes are not adjacent, i.e. they cannot communicate with each other directly. When a third node is taken into account, even angles between nodes could be determined (d).

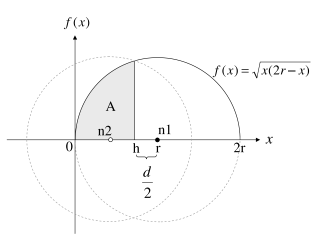


Figure 2. Mathematical foundation of NIDES

What is needed for distance estimation is a functional relation that takes the cardinality of the intersection as its input and yields the distance as its output. Figure 2 depicts the two nodes n1 and n2 and their communication ranges, where n1 wants to calculate its distance to n2. Let us assume that the solid-line half circle represents the communication range with radius r of n1. The node n1 is indicated by the black dot at $(r, 0)$. In this idealized plot the communication range is given by the graph of the circle function

$$f(x) = \sqrt{x(2r - x)}. \quad (1)$$

The area A (shaded grey) is one quarter of the intersection area of the two communication circles of n1 and n2 and is described by

$$\begin{aligned} F(x) &= \int f(x)\,\mathrm{d}x \quad (2) \\ &= \frac{r^2 \arcsin(\frac{x-r}{r}) + (x-r)\sqrt{x(2r-x)}}{2} + c \end{aligned}$$

which is f's antiderivative. We have determined $c = \frac{\pi r^2}{4}$ by setting $F(0)$ to 0. As for distance estimation, we have the situation that the size of the grey area A is known and that h is to be inferred. In other words, we want to deduce the value for x that corresponds to a certain integral value. Unfortunately F cannot be solved to x. Hence we have approximated F by a Taylor approximation T of degree 3 around $x = r$:

$$T(x) = -\frac{2x^3 - 6rx^2 - 6r^2x + r^3(10 - 3\pi)}{12r}. \quad (3)$$

Now T can be solved to x, which yields three solutions of which the relevant one is

$$h = r + 2\sqrt{2}r \sin\left(\frac{\arcsin\left(\frac{3\sqrt{2}(4A - \pi r^2)}{16r^2}\right)}{3}\right). \quad (4)$$

Now the distance between the two nodes can easily be calculated using

$$d = 2(r - h). \quad (5)$$

Like this, the distance between adjacent nodes can be inferred from their communication range r and A which represents one quarter of their communication circles' intersection area. We premise that the range r (or at least an upper bound) is known a priori and will be the same for all devices. Hence we treat it like a constant here. A is approximated using the heuristic that the size of A is proportional to the cardinality of the neighbor set intersection of the two nodes.

The underlying assumptions are that the wireless nodes are spread out locally uniformly over the considered area and that the network is sufficiently dense. We will discuss the consequences of both local and global density fluctuations in Section 4.2.

To clarify the underlying heuristic, imagine the nodes were positioned along a grid. Then each node would represent a quadratic area around it. This basic idea of each node representing a certain area also works even if nodes are randomly distributed.

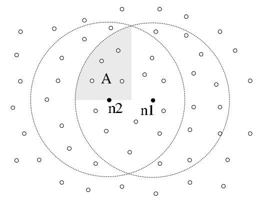


Figure 3. Relation between the intersection area and the number of shared neighbors.

Hence, the fraction of its neighbors that a node shares with a particular neighbor can be used to approximate the size of the area A to

$$A = 0.25\frac{s}{n}\pi r^2 \quad (6)$$

where n is the total number of neighbors the node has and s is the number of neighbors shared with the other node.

Figure 3 illustrates this process of approximating A from the neighborhood set intersection. The total number n of neighbors of n1 is 24, while the number s of shared neighbors is 13. Hence $13/24$ of the circle area intersect, A is a quarter of that and in this example evaluates to 4254.24 if $r = 100$. The resulting value for the d is 73.66 while the actual distance is 73. Note that the two participating nodes themselves are not counted when calculating the fraction because otherwise the obtained results are biased.

Like this, every node can estimate the distances to its neighbors if it knows its own neighbors as well as the neighbors of its neighbors. To make sure all nodes have these pieces of information, each node broadcasts two data packets. Due to length limitation of this paper we will only consider distance estimation based on single-hop-neighborhoods here (c.f. Figure 1 a and b).

First, all nodes send out a 'hello' packet that enables nodes in range to take notice of them. The sending nodes are inserted into the receivers' neighbor lists. Finally, all nodes have built up their list from the hello packets received. This list is kept in memory and will be augmented with information about the neighbor distances in the second step. For fast access, the neighbor list is kept sorted by the node IDs. Unsigned 16 bit values should usually be sufficient to uniquely identify all neighbors. If memory is extremely scarce, neighbor addresses can be easily mapped to at least locally unique (maybe actually 8-bit) identifiers. Even in dense networks with 50 neighbors, this list requires only 100 bytes of memory.

In a second data packet, each node broadcasts its neighbor list. When an adjacent node receives it, it can calculate the number of shared neighbors s. Because both the received and the local list are sorted, it is enough to iterate once through both lists.

In order to calculate the fraction of shared neighbors, the number of total neighbors n is required. If each node would use the length of its local neighbor list as n, two nodes can end up with different estimated distances to the other node. The reason is that they can have a different number of neighbors. Hence, we propose to employ the average neighbor count, i.e. the mean length of the received and the local neighbor list, in order to calculate the total number of neighbors n. Thus two nodes always estimate the same distance to each other regardless of the estimation direction. In addition, fluctuations in the node density and hence in the number of neighbors can be partially compensated for, leading to increased estimation accuracy.

Now the receiver can calculate the distance to the sender, and store the value in a distance list at the index of the sender. Note that it is not required to store the received neighbor list permanently; it can be disposed after the distance has been computed. Only the local neighbor list and the corresponding distances must be kept in memory. To represent the distances, an unsigned 8 bit value should be sufficient.

The most complex calculations required to calculate the distance d from the two neighbor lists are $\sin$ and $\arcsin$, which can be implemented by simple table-based approximations. Already linear interpolation from very simple tables with only 10 entries yields sufficient accuracy. Hence, it is possible to run these calculations on mobile, resource-constrained devices such as PDAs, wireless sensor nodes or other pervasive computing hardware.

Probably it makes sense to repeat the exchange of neighborhood lists from time to time. Periodic estimation enables the scheme to even handle mobility. The faster nodes move, the faster distances and the topology change. As a result, neighborhood information must be exchanged more frequently to make sure that the estimates are not outdated.

In some cases, nodes will already have neighbor lists available although NIDES did not explicitly exchange hello messages. Examples for other software that collects neighbor information are routing protocols such as AODV [22], protocols for topology control like SPAN [23] or clustering schemes. When neighbor lists are available from other protocols, the step of broadcasting hello messages can be omitted. Nevertheless, the transmission of the list in the second packet is still required.

4. Simulative Evaluation

To evaluate the distance estimation accuracy of NIDES we ran an extensive set of simulations in different scenarios. We decided not to simulate the actual exchange of data packets because we wanted to lay the focus of our work on effects that directly influence the distance estimate rather then on networking aspects. For this reason we chose SHAWN [24] for our simulations. It can analyze neighborhood relationships very efficiently and was developed for algorithmic simulations rather than network stack simulations.

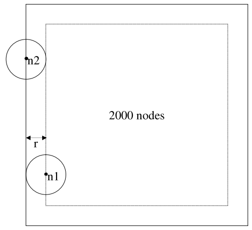


Figure 4. Simulation scenario.

We used the simulation scenario depicted in Figure 4. When analyzing the estimation accuracy, we only considered nodes that were located inside the dotted inner rectangle. The width of this inner area is $2r$ smaller than the simulation area, where r is the communication range. Thus the full communication range of the considered nodes resides within the simulation area (c.f. node n1). This avoids edge effects that would be caused by missing neighbors for node n2 on the left. We will drop this constraint in Section 4.4 when analyzing the effects of network boundaries.

In order to obtain simulation results for different network densities (i.e. different average neighborhood sizes) we calculated the size of the inner area depending on the desired network density so that a fixed number of nodes resided inside. The size of the full simulation area and the overall node number can then be calculated accordingly. Nodes were spread randomly over the full simulation area, i.e. their x and y coordinates where taken from a uniform random distribution. The simulations presented in Section 4.4 are exceptional in that they feature non-uniform node distributions.

For all simulations, the communication range was set to 10. However, the obtained results are valid for arbitrary communication ranges.

We iterated over all pairs of adjacent nodes inside the inner rectangle, estimated their distances using NIDES, and compared these estimates to their real Euclidean distances. To obtain statistically sound results, we averaged the results of 100 simulations with the same parameter set.

4.1. Influence of Network Density

Figure 5 shows an interquartile diagram of the estimation error over the average neighborhood size. The error is expressed as a fraction of the communication range r.

In an interquartile diagram, each bar describes a value distribution where only 25% of the values lie above the upper bound and below the lower bound of the bar. Figure 5 also includes the mean absolute error and the median. For the earlier the absolute error values were summed up for all considered distances and divided by their number, while for the latter a value is selected so that half of the errors are bigger and the other half is smaller. Hence the median indicates whether the estimates are biased.

The spread of the interquartiles decreases with increasing network densities, i.e. neighborhood sizes. As expected, the estimates get more exact with bigger neighborhoods.

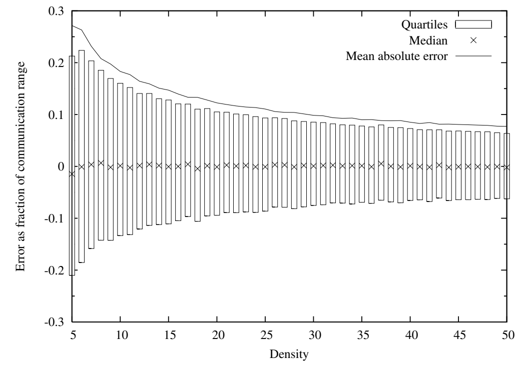


Figure 5. Estimation error distribution over density.

With many neighbors, the interquartile of the estimates lies within $0.06r$ around the correct distance (i.e. 50% of the estimates feature an error of less than 6% of the communication range), the mean absolute error is $0.075r$.

For densities above 10 the mean absolute error is below 18% of the communication range.

Note that the median error is always very close to zero which means that NIDES has no tendency to systematically over- or underestimate distances.

Research has shown that mobile ad-hoc networks tend to partition with high probability if the density, i.e. the average neighborhood size, is below 10 [25]. Others proved that the density required for full connectivity depends on the network size. The more nodes a network consists of, the higher the required density is [26]. For a network comprising 300 nodes, a density of about 13 is needed. Some authors propose to employ a higher node density than actually needed and always put a certain fraction of nodes to sleep in order to save energy and prolong the network lifetime while keeping up coverage [23, 27]. In such a case, also nodes that would usually be sleeping could be woken up and be used for distance estimation in order to increase estimation accuracy.

All in all, we consider an average neighborhood size of 15 to be a reasonable choice for wireless sensor networks. With 15 neighbors, the mean error is about $0.15r$ with an interquartile spread of about 0.2. This means that half of the estimates exhibit an error of 10% of the communication range or less. If not indicated differently, the simulation results presented in the remainder of the paper feature a density of 15.

4.2. Influence of Density Fluctuations

As mentioned above, we consider 15 to be a reasonable density for wireless sensor networks and thus conducted most of our simulations with this density. However, if the average network density is set e.g. to 15 it does not mean that all nodes have 15 neighbors. Even though the nodes are uniformly distributed over the simulation area, den-

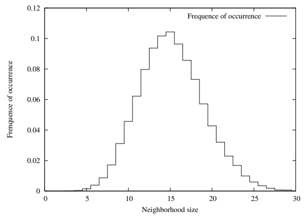


Figure 6. Density histogram, average density 15, uniform distribution

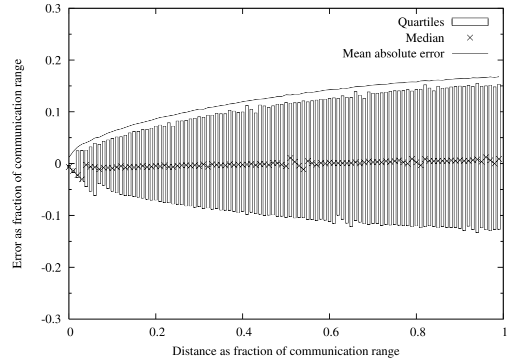


Figure 8. Estimation error distribution over distance.

sity fluctuations occur. Figure 6 shows a histogram of the neighborhood sizes for the simulations in Figure 5 where the density was set to 15 . Neighborhood sizes between 3 and 32 occur, the average is 15.2. These fluctuations are due to irregular node placement. They are actually not unwelcome because they reflect varying neighborhood sizes that occur in real life despite of uniform node placement.

However, these kinds of fluctuations are part of the simulation results presented here, and contribute their share to the spread of the estimation error. Because neighborhood based distance estimation assumes a uniform node distribution, density fluctuations can lead to distances that are over- or underestimated. In general, three different cases can be distinguished. Figure 7 illustrates these three cases with extreme examples. The arrow points from the node that estimates the distance to the one that the distance is estimated to.

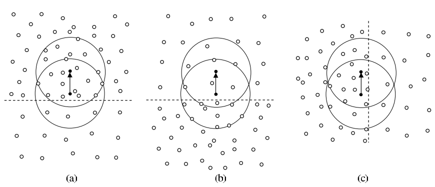


Figure 7. Different kinds of density fluctuations.

If the density in the intersection area is higher than in the rest of the communication area (a), the estimated distance will be too short because the fraction of neighbors in the intersection area is too high to represent its spatial extent correctly. If the density in the intersection area is lower than in the rest of the communication area (b), the estimated distance will be too big. This means that density variances along the estimation axis (indicated by the arrow) lead to estimation errors, while density fluctuations orthogonal to the estimation axis (c) do not influence the estimation accuracy. Unfortunately, such density fluctuations can so far not be detected and thus not be compensated.

4.3. Influence of the Euclidean Distance

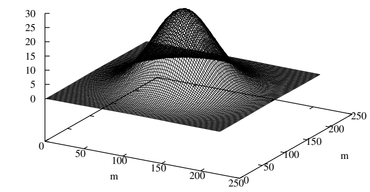
We then studied the dependency of the estimation accuracy on the actual distance between the two involved nodes. Figure 8 depicts an interquartile diagram of the error over the Euclidean node distance. Both distance and error are expressed as a fraction of the communication range.

The mean absolute error and the interquartile range increase with the real distance between the two estimating nodes. The reason for that is the decreasing absolute number of shared neighbors. Hence density fluctuation gains importance, and leads to an increased error spread. However, the interquartile spread stays below $\pm 0.15r$ and the mean absolute error is always less than $0.16r$. If these values are averaged over all distances, a mean error of $0.15r$ and an interquartile range of $\pm 0.1r$ result (as indicated in Figure 5 at x=15).

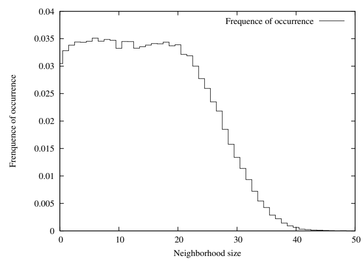
Note that there are more estimates between nodes that are far away than between close nodes. The reason is the circular nature of the communication area. There are only very few neighbors close to a node, but many that are further away. Their number increases in a linear fashion over distance. Hence the bigger errors yielded by estimates to far away nodes weigh heavier when the average error over all different distances is computed. This results in average errors close to those for distances close to r.

The median is not centered for very short distances. It runs on a line that is defined by $g(x) = -x$ instead. This indicates that half of the estimates are 0 even though the nodes have a certain distance, i.e. their displacement is underestimated exactly by their Euclidean distance. The reason is that nodes that are very close to each other have a certain probability of sharing all neighbors. It is due to the fact that the fractions of shared neighbors are discrete and only certain values occur (e.g. $\frac{15}{15}$ and $\frac{14}{15}$, but nothing in between) while the real node distance is continuous. This effect occurs again at about $0.55r$, $0.8r$ and $0.95r$ where different distances must be mapped on the same fraction.

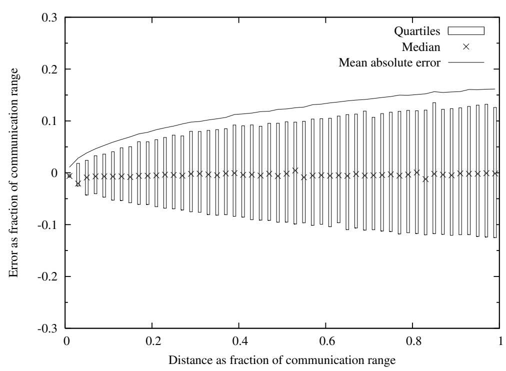
The effect depends on the node density: the higher the density, the smaller the effect (because the denominator increases). However, it never significantly influences the estimation accuracy.



(a) Node distribution over the simulation area.



(b) Density histogram



(c) Estimation error distribution over distance

Figure 9. Gaussian node distribution and resulting error characteristic.

4.4. Influence of Non-Uniform Node Distributions

As stated in the introduction, sensor networks are envisioned to be scattered over an area of interest, e.g. by dropping them out of an air plane. With this kind of adhoc deployment, uniform node distributions are rather unlikely. Instead, Gaussian distributions with a high network density in the center of the deployment area that decreases toward the edges are probable.

Thus we conducted simulations with non-uniform node distributions. An example of a Gaussian node distribution over the simulation area is depicted in Figure 9(a). It shows the node density at every point of the simulation area and results from averaging 100 simulations runs. As for the results discussed above, the average node density is set to 15. The resulting distribution of neighborhood sizes can be inspected at the histogram depicted in Figure 9(b). Neighborhood sizes between 0 and 25 have about the same occurrence probability around 3.3%.

This means that a Gaussian distribution of nodes over the simulation areas leads to an increased spread of occurring neighborhood sizes compared to Figure 6 especially towards the low end.

To explore how this shift in neighborhood size probabilities influences the estimation accuracy, we followed the simulation procedure described at the beginning of this section except for that we did not exclude the boundaries of the simulation area. This means that opposite to all other simulation results described in this paper, the simulations presented in this subsection also include the nodes at the border of the network. Because the network density decreases from the center of the simulation area to the edges, the network fades out slowly.

The resulting error characteristic is presented in Figure 9(c). Again, the interquartile diagram shows the estimation error over distance, both are expressed as fractions of the communication range. Hence the data can easily compared with those resulting from uniform node distributions in Figure 8 in Section 4.3.

The results stemming from Gaussian node distributions closely resemble those obtained with uniform distributions. The error decreases even a bit because of the increased density in the center of the simulation area. The median runs along the x-axis, so obviously the non-uniform distribution does not lead to biased results.

This clearly shows that NIDES is well applicable in real world scenarios that include varying network densities in different parts of the network. The extremely low density at the network edges leads to a decreased accuracy of NIDES in those areas, but does not influence the overall performance.

5. Conclusion and Future Work

In this paper we presented the neighborhood intersection distance estimation scheme (NIDES). This novel approach computes distance estimates from intersection cardinalities of sets of adjacent nodes. In other words, its operation is based on the observation that nodes that are located close to each other share many of their neighbors whereas distant nodes share only few neighbors. We showed how an equation can be derived that takes the fraction of shared neighbors as its input and yields the estimated distance between two nodes. It can easily be computed even on very resource-constrained devices such as wireless sensor nodes.

Through simulations we assessed the ranging accuracy of NIDES. While the scheme gets more accurate with increasing network density, it achieves an average error of only about 15% of the communication range if nodes have about 15 neighbors.

Thus NIDES categorizes between existing approaches for distance estimation. It achieves a higher accuracy than estimates based on RSSI, and is less susceptible to reflections, fading and multi-path propagation. Estimates based on differential time of flight feature a higher accuracy than NIDES but demand for specific hardware such as ultra sound emitters and receivers. These have negative effects on cost, size and power consumption, which seems disadvantageous especially for wireless sensor networks. In addition, these have only a very limited maximum range compared to NIDES. Radio interferometry features promisingly low errors but brings in specific requirements to the RF chip. Also, this approach requires complex run-time computations and ranging takes very long, two disadvantages that can be omitted when neighborhood intersection is used.

In the future, we plan to extend NIDES to consider two hop neighborhoods. We expect this step to further enhance estimation accuracy. Furthermore, it enables nodes to sort neighbors with regard to angle, which opens up new vistas for location estimation protocols. In addition, angles along multi hop paths can be estimated.

References

[1] I. F. Akyildiz, Y. S. W. Su, and E. Cayirci., "Wireless sensor networks: A survey", *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart Dust: Communicating with a Cubic-Millimeter Computer", *Computer*, vol. 34, no. 1, pp. 44–51, 2001.

[3] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison", *Comput. Networks*, 2003.

[4] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)", in *Proceedings of GLOBECOM, San Antonio, November 2001.*, 2001.

[5] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system", *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.

[6] A. Butz, J. Baus, A. Krüger, and M. Lohse, "A hybrid indoor navigation system", in *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, 2001, pp. 25–32, New York, NY, USA. ACM Press.

[7] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks", in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 59–67, New York, NY, USA. ACM Press.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system", in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 32–43, New York, NY, USA. ACM Press.

[9] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors", in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 166–179, New York, NY, USA. ACM Press.

[10] L. Girod and D. Estrin, "Robust Range Estimation Using Acoustic And Multimodal Sensing", in *IEEE/RSI Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001.

[11] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The Effects of Ranging Noise on Multihop Localization: an Empirical Study", in *The Fourth International Conference on Information Processing in Sensor Networks (IPSN "05)*, 2005.

[12] J. Sallai, G. Balogh, M. Maroti, A. Ledeczi, and B. Kusy, "Acoustic Ranging in Resource-Constrained Sensor Networks", in *International Conference on Wireless Networks*, 2004.

[13] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application", in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 59–68, New York, NY, USA. ACM Press.

[14] N. Bulusu, J. Heideman, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices.", *IEEE Personal Communications*, 2000.

[15] C. Savarese, J. Rabaey, and J. Beutel, "Locationing in distributed ad hoc wireless sensor networks", in *Proc. 2001 Int"l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)*, 2001.

[16] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks", in *HICSS*, 2001.

[17] M. Maroti, P. Vlgyesi, S. Dora, B. Kusy, A. Nadas, A. Ledeczi, G. Balogh, and K. Molnar, "Radio interferometric geolocation", in *SenSys "05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005.

[18] E. Elnahrawy, X. Li, and R. P. Martin, "The Limits of Localization Using Signal Strength: A Comparative Study", in *EEE SECON*, 2004.

[19] L. Doherty, K. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks", in *Proceedings of the IEEE InfoCom*, 2001.

[20] N. Sundaram and P. Ramanathan, "Connectivity based location estimation scheme for wireless ad hoc networks", in *Proceedings of IEEE Globecom*, Nov. 2002.

[21] J. Cartigny and D. Simplot, "Border Node Retransmission Based Probabilistic Broadcast Protocols in Ad-Hoc Networks", in *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9*, 2003, p. 303, Washington, DC, USA. IEEE Computer Society.

[22] C. Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing", Request for Comments 3561, Network Working Group, Internet Engineering Task Force, July 2003.

[23] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", *Wirel. Netw.*, vol. 8, no. 5, pp. 481–494, 2002.

[24] A. Kröller, D. Pfisterer, C. Buschmann, S. P. Fekete, and S. Fischer, "Shawn: A new approach to simulating wireless sensor networks", in *Design, Analysis, and Simulation of Distributed Systems 2005, Part of the SpringSim 2005*, Apr. 2005.

[25] H. Hellbrück and S. Fischer, "Towards Analysis and Simulation of Ad-Hoc Networks", in *Proceedings of the 2002 International Conference on Wireless Networks (ICWN02)*, June 2002, pp. 69–75, Las Vegas, Nevada, USA. IEEE Computer Society Press.

[26] F. Xue and P. R. Kumar, "The number of neighbours needed for connectivity of wireless networks", *IEEE Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[27] J. Deng, S. H. Yunghsiang, W. B. Heinzelman, and P. Varshney, "Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks", in *4th Workshop on Applications and Services in Wireless Networks (ASWN 2004)*, 2004, pp. 99–108.