Location Discovery in SpeckNets using Relative Direction Information

Ryan McNally D. K. Arvind
School of Informatics, University of Edinburgh,
10 Crichton St. Edinburgh, EH8 9AB, UK
`http://www.inf.ed.ac.uk`

Abstract

A speck is intended to be a miniature (5X5X5mm) semiconductor device that combines sensing, processing, wireless communication and energy storage capabilities. A specknet is an ad-hoc wireless network of specks. The location of specks in the network is useful in processing information, for reasons ranging from routing data to giving the data sensed a spatial context. This paper presents an algorithm for discovering the location of specks and updating that information in the face of movement. The proposed algorithm exploits the location constraints implied by the sensed directions to a speck's one-hop neighbours in order to compute a likely location. Direction information may be gleaned in a robust manner through the use of free-space optical communications systems. The algorithm is fully distributed, requires no special infrastructural support, has modest requirements in terms of computation and communication and does not rely on range measurement or anchor nodes. The performance of the location discovery algorithm is evaluated in the SpeckSim simulator under a range of adverse conditions.

Key words : networks, localization

1 Introduction

A speck [23] is designed to combine sensing, processing, wireless networking capabilities and a captive power source, all integrated in a minimal volume.

The specknet is a programmable computational network of hundreds of specks - in effect a fine-grained distributed computation platform built on a substrate of an ad-hoc wireless network of resource-constrained nodes. The model of distributed computation takes into account some specific features of specknets, such as the unreliability of wireless communication, and a higher than normal failure rate of specks due to the harsh operating conditions, meagre power supply, and very large volume manufacturing where individual specks cannot be fully tested.

A feature that will be important in many applications of specknets in pervasive and ubiquitous computing is the ability to map the data being sensed, and the information subsequently extracted, to its location within the specknet. In addition location data is very useful when routing messages across the network [1], especially in large networks where conventional routing tables become infeasible on resource-constrained devices.

Developing location algorithms for specknets poses unique problems. The specks will have extremely limited power (~1mW power budget), computation, bandwidth and storage (~2KB), and communication is expensive in comparison to computation (~x10). Any location algorithm should require minimal processing and storage, be robust against unreliable specks and communications, and be fully distributed across the specknet. The algorithm proposed in this paper satisfies each of these constraints at the cost of the requirement that the specks be equipped with appropriate sensory capability.

In the rest of this paper, Section 2 outlines existing algorithms for location discovery and their weaknesses in the face of the requirements of specknets; Section 3 describes the proposed algorithm; Section 4 describes the metric chosen to evaluate the performance of the algorithm and presents the results, with conclusions and possible improvements outlined in Section 5.

2 Related Work

Algorithms for location discovery in sensor networks can be classified according to various factors:

2.1 Infrastructure

Some localisation algorithms require infrastructural support over and above the network nodes themselves. This infrastructure is typically of two forms:

The first is to use a small number of devices separate to the network that enables nodes to locate themselves, as in the Lighthouse [2], Thunder [3] and similar approaches [4][5], and those approaches using mobile robots [6][7]. These algorithms share a common drawback, namely that the operation of the algorithm is entirely dependent on the operation of these scarce resources. If the devices break down, no localisation is possible. In addition, these devices must advertise themselves to the entire network and so are by necessity conspicuous. This requirement may be problematic for some applications where stealthy deployment is valued.

The other approach to infrastructure is to have a subset of nodes in the deployed network that is more capable than normal nodes. These so-called *anchor nodes* have knowledge of their location through some other means, such as GPS or simply explicit programming. Other nodes can then compute their location with respect to these anchor nodes. Such approaches have requirements ranging from only a few anchor nodes [8][9] to requiring that a anchor nodes be present throughout the deployed network [10][11].

The drawback to using anchor nodes lies in their deployment. Algorithms that use few anchor nodes typically operate by having other nodes measure the distance to the anchors and trilaterating their own location. Thus if the anchors are deployed to be almost collinear or coincident, the trilateration against these positions using noisy distance measurements is error-prone and even ambiguous. Algorithms that use many anchor nodes require that they are distributed evenly throughout the deployment area.

In both cases, such careful placing of the anchor nodes is difficult to guarantee in a true ad-hoc deployment.

2.2 Centralisation

Centralised algorithms [12][13][14] operate by gathering information, such as network connectivity or sensor readings, for the entire network onto the central computer, where it can undergo substantial analysis before the computed locations for nodes are disseminated back into the network. These approaches reduce the amount of computation required of the nodes but represent a single point of failure and limit the scalability of the network.

The increased amount of data that must be routed across the network is extremely undesirable due to the relatively high power expenditure of communication when compared to computation. In order to collect the network information at the central computer, the amount of data transfer necessarily increases rapidly closer to the central computer, leading to a problematic disparity in power consumption (and hence node lifetimes) across the network.

Centralised algorithms also suffer in networks with node mobility. The latency burden of information making two full traversals of the network in addition to the processing time can easily render the computed locations irrelevant by the time they are received by the nodes.

2.3 Available data

Algorithms can also be categorised according to what information is available to the nodes. The baseline is usually connectivity (the set of neighbours that a node can contact directly), but nodes may also have distance information about their neighbours, gained through received signal strength information (RSSI) [15][16] or ultrasonic time-of-flight [17].

Using RSSI as a very rough *indicator* of distance is easy to implement and can help refine location estimates and resolve ambiguities. However, it has been shown [18][19] that RSSI is a poor *measure* of distance in the general case, and can vary dramatically with apparently subtle environmental factors that are extremely difficult for nodes to account for.

Using ultrasonics and measuring the time-difference of arrival of a sonic and radio signal can give much greater accuracy when computing inter-node distances. However, ultrasonic systems suffer from relatively high power consumption and physical bulk. Ultrasonic transceivers also suffer from issues of interference, low data rates and interference, reducing their utility in high-density networks [20].

Akella et al.[21] propose an algorithm that uses direction information in addition to range measurement. Direction information can be gathered in a robust fashion by using a free-space optical (FSO) communication system. This system comprises a number of optical transceivers, essentially a light-emitting diode (LED) paired with a light sensor, such that a node can transmit a signal in a particular known direction, and receive signals coming from that direction. The transceivers are arrayed on the node to provide communication capability in any direction.

This system gives the nodes the ability to determine the relative direction to a neighbouring node based on which light sensor received the neighbour's signal. If the neighbour details which transceiver is being used to transmit the received signal, a node is able to compute that neighbour's orientation relative to itself.

The algorithm combines the computed orientation with the range measurement in order to compute the neighbour's relative location. If the node knows its own location, this relative location estimate for the neighbour can be converted to an absolute estimate by simple vector addition, and communicated to the neighbouring node. This neighbouring node is then able to compute absolute location estimates for its neighbours, and so on. In this way, the number of nodes that know their own location can grow from a single elected origin node to encompass the entire network.

The primary problem with this approach is that each relative location estimate has an error factor determined by the fidelity of the FSO communication system. A system with many individual transceivers is able to estimate the direction and orientation of neighbouring nodes with greater accuracy than that with fewer transceivers. As each node's location estimate is based on the location estimate of a single node that is one hop closer to the origin node, the location errors compound as the hop count from the origin node increases. This accumulation of location error severely limits the utility of the algorithm in large networks.

3 Algorithm Description

The algorithm proposed in this paper, dubbed the Sectoring algorithm, requires a similar level of node capability as the FSO algorithm due to Akella et al.[21], namely, the ability to sense the direction of neighbouring nodes. Note that the inter-node ranging ability assumed by the FSO algorithm is not required for the Sectoring algorithm.

The direction sensing capability can be implemented in a robust fashion with the use of optical communications. A node equipped with an array of LED transmitters and a corresponding array of light sensors can make directed transmissions and detect the direction of incoming transmissions. The fidelity of a sensed direction is based purely on the number of transceivers that a node is equipped with: i.e. a node with 8 transceivers can transmit and sense in 8 different directions. Such optical communications systems have been used previously to aid in robot swarm organisation [22].

The fundamental premise of the algorithm is to exploit the direction constraints implied upon a node by its neighbours' communications. A node that knows its location restricts the locations of those neighbours that receive its transmission to lie within the area of effect of that transmission. In contrast to radio communication, the extent of an optical transmission is easy to control with the physical design of the node. Baffles around the LED and light sensor can limit their effect to a known segment of the area around the node. In addition, the area of effect can be easily approximated as a circle segment. The individual constraint on a node's location implied by a single neighbour may not be enough to compute a sufficiently accurate location estimate, but the combination of many constraints implied by many neighbours is likely to increase the accuracy of the location estimate.

The algorithm operates as a feedback loop: each node gathers information about its immediate neighbours' locations and orientations, uses this information to compute a location estimate, and then shares this estimate with its neighbours, thus contributing to their own location estimates. The algorithm does not require any infrastructure external to the network or anchor nodes that have perfect knowledge of their location, and so the resulting coordinate system is consistent internally to the network but will differ by an affine transform to the positions of the nodes as measured by some external observer.

Each node in the network must agree on a common point to use as the origin of the coordinate system. This is achieved by calculating the coordinate system relative to the node with the lowest ID number. The algorithm operates by having every node make periodic transmissions from each of its optical transceivers. Each transmission details:

- The ID of the transmitting node
- The origin ID that the transmitting node is located with respect to
- The computed location of the transmitting node
- The predicted direction of the transmission. This is obtained by adding the orientation of the transceiver onto the computed orientation of the node.

Each node will store the details of recently received transmissions, in addition to the angle of the optical receiver that the transmission was detected on. Each node also keeps track of the lowest origin ID yet encountered. When a transmission is received, the origin ID contained therein is examined to determine the data's validity:

- If the received origin ID is greater than that held, then the transmission is ignored
- If the received origin ID is lower than that held, then all stored data is discarded and the stored origin ID is updated to the new lower value
- If the received origin ID is the same as that held, then the stored data is updated with the received details as normal

This procedure allows the network to converge on using a single reference point for the coordinate system. It is important to note that the node with the lowest ID has no special responsibilities as the origin, and participates in the algorithm as any other node does. All that is required is that the network agree on a common reference point around which to build a coordinate system, and it is of no consequence to the algorithm if the origin node subsequently expires.

At the outset of the algorithm, every node considers itself to be the origin of the coordinate system, i.e.: its location estimate is (0,0), its computed orientation is 0 degrees, and the origin ID field is set to its own ID. In order to predict the area of effect of a transmission, each node must be able to estimate its own location and orientation relative to the rest of the network.

3.1 Orientation Estimation

Nodes can calculate their orientation relative to each neighbour by examining the direction of incoming transmissions and the orientation of the transceivers that they were detected on. As illustrated in Figure 1, node *A* calculates the direction of its transmission *t* by adding the orientation of the transceiver to its own orientation estimate *a*. Node *B* detects the transmission on the transceiver that has relative orientation *r*, and can the calculate its own orientation *b* as 180-*r*+*t* degrees. Each neighbour can imply a different orientation for a node, and so the average of each implied orientation is used to predict transmission directions. This averaging allows the nodes to estimate their orientation with greater accuracy than the fidelity of their FSO system would suggest.

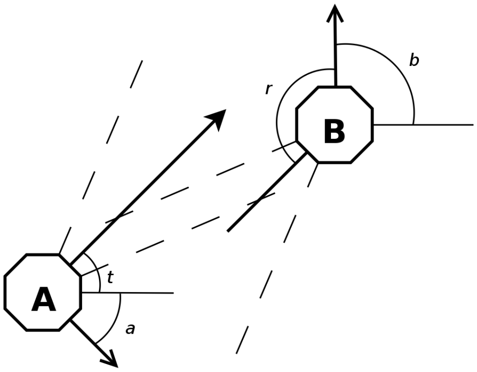

Figure 1: Calculating relative orientation

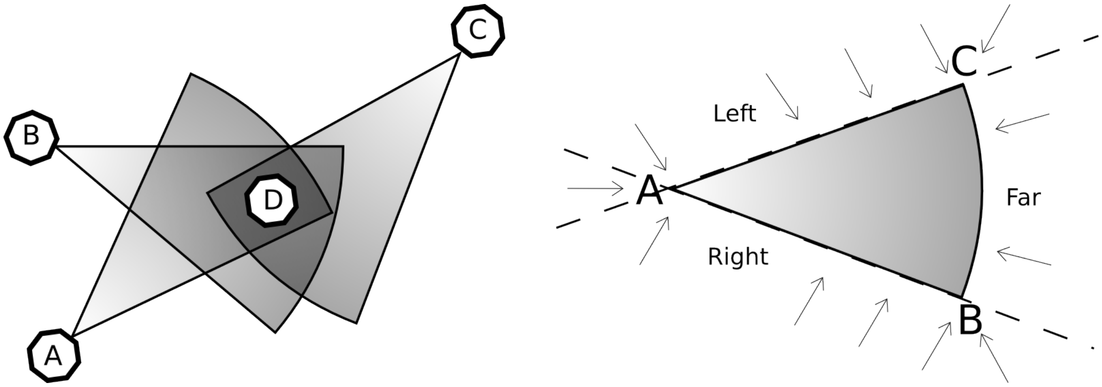

Figure 2: Location estimate constraints and constraint compliance procedure

3.2 Location Estimation

In order to compute a location estimate, each node examines the stored transmission records and constructs a set of circle segments using the received transmission direction information and location estimate taken with knowledge of the divergence angle of the FSO transceivers. The node then calculates a point that lies in the intersection of the circle segments as the location estimate, as illustrated on the left of Figure 2, where node *D* computes a location that lies within the transmissions of nodes *A*, *B* and *C*.

The location estimate is made using an iterative approach. Each circle segment constraint *C* implied by the transmission from a neighbour is compared with the current location estimate *E*. If a node has no existing estimate *E*, it simply takes the centroid of *C* as an initial guess that is then refined. If *E* lies outwith *C*, then the minimum alteration vector *V* is calculated such that *E*+*V* lies within *C*. Alteration vectors are calculated for every neighbours' circle segment constraint, and the average of these vectors is applied to the location estimate.

The procedure to calculate *V* is simple, given the vertices *A*, *B* and *C* of the circle segment. If the current location estimate *E* lies to the right of the line passing through *A* and *B*, then *V* is calculated as the shortest vector between *E* and the line segment *AB*. *V* is calculated similarly if *E* lies to the left of *AC*. If *E* lies to the left of *AB* and to the right of *AC*, the distance between *E* and *A* is calculated and compared with the assumed transmission range. If *E* lies outwith the transmission range, then *V* is calculated as the minimum vector to satisfy the constraint, i.e.: that moves *E* directly towards *A*. The right-hand side of Figure 2 illustrates these tests.

In addition to the rigid constraint that a node's location estimate must lie within the circle segments implied by its neighbours' transmissions, a further elastic constraint is applied to reflect the fact that a node is more likely to lie further from a neighbour, where the circle segment is wider. This is achieved by applying a repulsive force to each node's location estimate in the direction of each neighbour's transmission. The repulsive force is scaled such that it is stronger when two nodes' estimated locations are closer together, and conversely much weaker when the nodes are more distant.

3.3 Summary

The Sectoring algorithm uses a modest level of storage capacity (2 node IDs and 3 numbers with low precision requirements – it was found that using more than 8 bits per number gave negligible performance improvements) per neighbour, and a modest amount of computation (ranging from 18 arithmetic and 2 trigonometric to 24 arithmetic and 4 trigonometric operations) per neighbour in order to compute or refine a location estimate. The sensory requirements for operation are straightforward to implement and robust in the face of environmental effects such as electromagnetic interference and background noise. There is a requirement, however, for deployed nodes to have line-of-sight contact with immediate neighbours.

While the Sectoring algorithm uses a similar level of hardware capability to the algorithm due to Akella et al., it distinguishes itself by exploiting information from multiple sources. In the case of Akella's algorithm each node is only located with respect to one of its neighbours. This reliance on a single source of information leads to the accumulating error that severely limits the scalability of the approach.

When using the Sectoring algorithm, nodes locate themselves with respect to their entire neighbourhood. This fusion of multiple sources of location information, and the feedback process of continually sharing location information with neighbours in order to refine location estimates, eliminates error accumulation and allows the localisation of even very large networks.

4 Algorithm assessment

4.1 Simulation Parameters

Simulations were performed using the SpeckSim simulator to determine the performance of the algorithm. The baseline scenario is 100 nodes distributed randomly over a unit square. Each node is equipped with 8 optical transceivers arrayed to give complete and non-overlapping coverage around the node. Each transceiver has a transmission range of 0.2 units. Each node attempts to transmit once every second. The primary metric shown is the location error: this is simply the distance between a node's computed and physical locations, disregrading a network-wide skew transformation. All simulations were repeated 100 times with different network layouts and communication timing, and the results averaged. Error bars where present indicate the standard deviation of that metric over the repetitions of the simulations.

4.2 Results

Perhaps the easiest way to illustrate the performance of the algorithm is with a diagram of the output. Figure 3 shows the error between each node's actual and computed position, disregarding any network wide skew, after the network has reached a settled state. The light lines denote network links.

The network on the left is comprised of 100 nodes with a communications range of 0.2 units, while the network on the right is double the magnitude, with 400 nodes each with a communication range of 0.1 units. These images demonstrate that location error does not increase with the number of hops to the origin node as in Akella's approach. A feature of note is that neighbouring errors tend to have similar magnitudes and directions, indicating that nodes have located themselves with respect to their immediate neighbours with a high degree of accuracy, but that errors creep in between clusters of tightly-connected nodes.

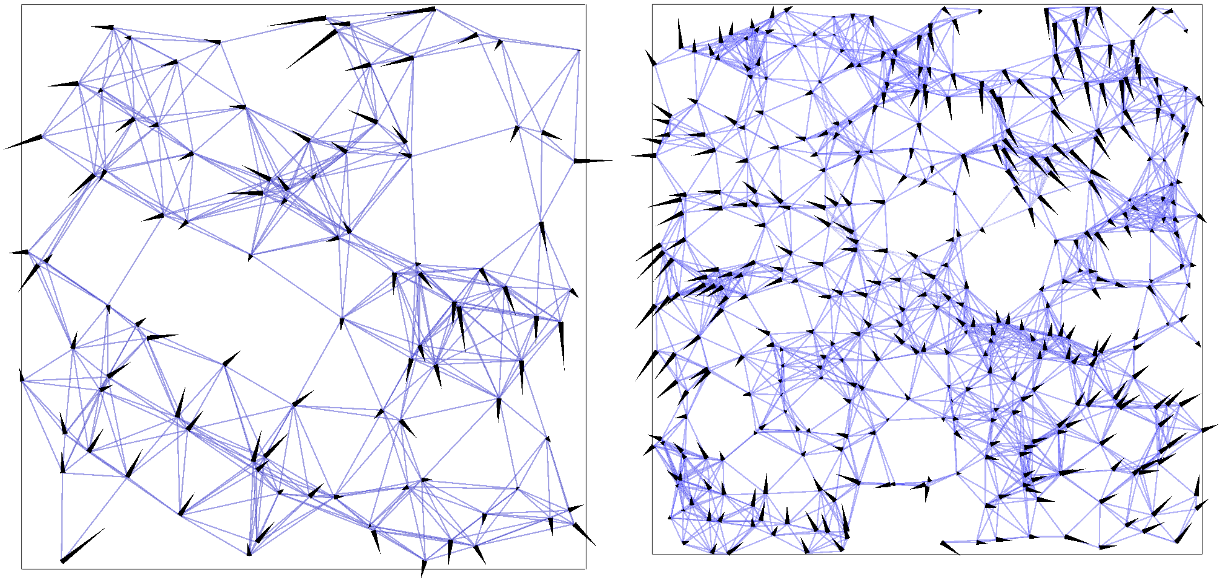

Figure 3: Location estimate error

One of the most important factors for the accuracy of the location estimates is the fidelity of the FSO system. The first graph in Figure 4 charts the magnitude of location error over time, and shows the effect of varying the number of transceivers that each node is equipped with. As can be seen, nodes are able to locate themselves to

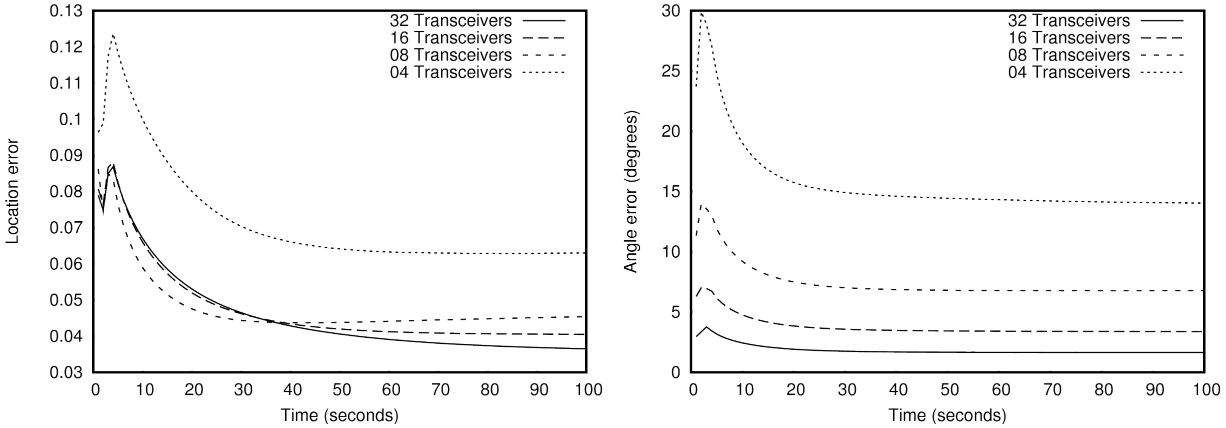

Figure 4: Location error and angle error over time

within 35% of their communication range while having the ability to sense that their neighbours are in one of only four possible directions. Although not shown on the graph for reasons of clarity, the standard deviation of the location error metric for networks of nodes with 8 or more transceivers remains steady at around 0.012 units.

The second graph shows the benefits of combining several sources of angle information. For instance, nodes with only 4 transceivers are only able to sense their orientation to within 45 degrees based on a single source, but can achieve an average accuracy of 15 degrees by combining readings from many neighbours. The second graph also demonstrates the dependence of location error on angle error, with both graphs showing a rise in error as the network converges on using the same seed node, and then a gradual decrease as the system of constraints approaches a globally-satisfied state. As expected, increasing the number of transceivers leads to lower location and angle errors.

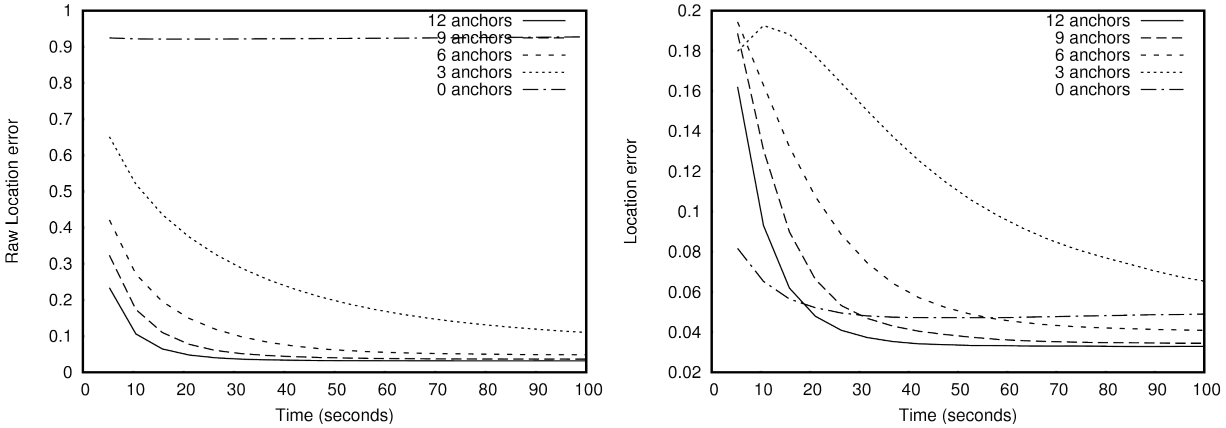

Figure 5: Effects of location anchor nodes

Unlike many localisation algorithms, the Sectoring algorithm does not require anchor nodes in order to compute a consistent location map for the entire network. The algorithm will, however, seamlessly integrate the information from any anchor nodes that may be present such that the location map for the entire network converges on the physical truth of the network deployment. The algorithm operates exactly as before, with the caveat that the anchor nodes turn their receiving hardware off - their location estimate is not based on that of their neighbours. The graph on the left of Figure 5 shows the raw error in the location estimates - simply the average distance between a node's computed and actual locations. The network with no anchor nodes present is steady at a high level of raw error, reflecting the error due to the network-wide skew between computed location map and physical truth, but networks with anchor nodes converge on having a raw error level similar to the corrected error levels shown in Figure 4. The graph on the right of Figure 5 shows the corrected error metric over the same networks, and demonstrates how the presence of the anchor nodes leads to an initially high corrected error metric that falls with time as nodes converge on knowing their true physical locations and the corrective transform approaches the identity matrix.

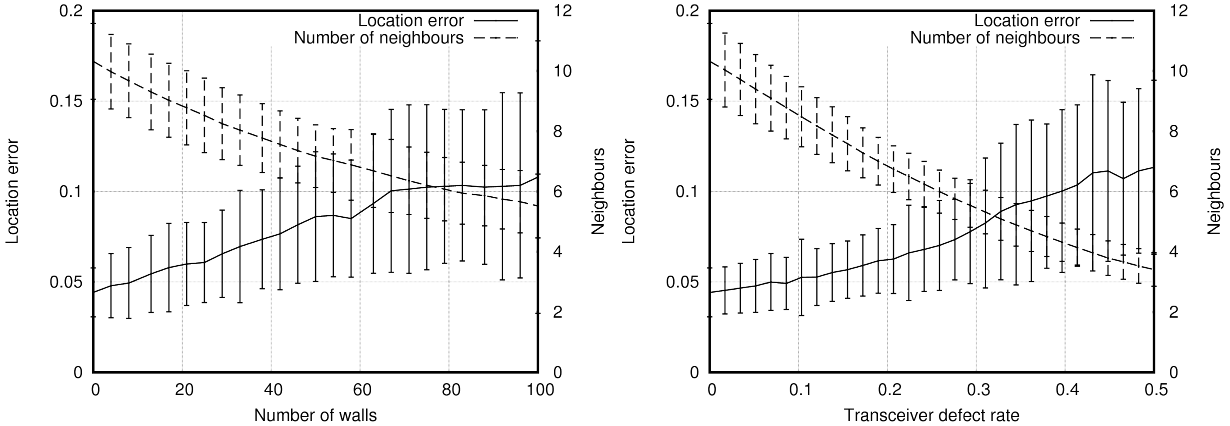

Figure 6: Effects of opaque walls and transceiver defects

It is unlikely that a network will be deployed in an ideal environment, or that large-scale device manufacture is flawless, and so it is important to explore algorithm behaviour under adverse conditions. The left-hand graph in Figure 6 shows the effects of introducing randomly-placed 0.1 unit long walls into the deployment area that block transmissions. The right-hand graph charts location error under various levels of imperfect manufacture. For example, when the “transceiver defect rate” is 0.2, there is a 20% chance that a given transceiver on a node is non-operative, and can neither transmit nor receive data.

These graphs both demonstrate the relationship between location error and neighbourhood size: in general, nodes with more neighbours have tighter constraints on their locations and thus lower levels of error. That this relationship is not precisely mirrored between these graphs is down to the nature of the failed communications. Given three nodes *A*, *B* and *C*, the constraints on *A*’s location are tightest when *AB* and *AC* are orthogonal. If *A* is situated next to a wall, there is 50% less chance of this orthogonal arrangement occurring than if *A* were situated in open space. In contrast, nodes that have faulty transceivers are still able, on average, to communicate in all directions, and so the optimal orthogonal arrangement can still occur.

The graph on the left-hand side of Figure 7 explores the effects of a failure-prone communications channel. For instance, when “RX failure probability is 0.5, there is a 50% chance that the reception of a particular broadcast will fail, and the receiving speck will not get the transmitted data. The shading of the graph gives an indication of location error at that particular level of channel failure and simulation time, with darker shades indicating larger errors. The contour line plots are placed at location error levels of 0.05, 0.06, 0.07 and 0.09 units in order to clarify the shading. As can be seen, increasing levels of channel unreliability only serve to delay a network’s attainment of a low error rate, rather than preclude it. As expected, as the chance of a reception failure approaches 100%, the delay in the network attaining a settled state grows dramatically.

The graph on the right-hand side of Figure 7 shows the cumulative distribution function of location error in 1000 random network layouts of 100 nodes each. From this we can assert that for a randomly deployed network of

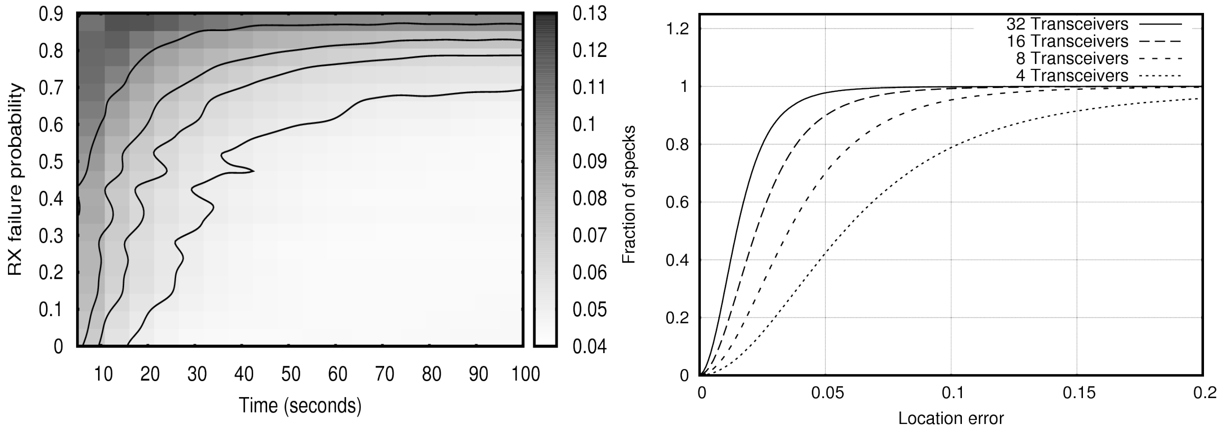

Figure 7: Effects of noisy channel | Location error distribution

nodes equipped with 8 transceivers each, there is a 70% chance that a given node will compute its location to within 0.05 units, or 25% of its own communication range. Indeed, even nodes with only 4 transceivers are able to locate themselves to within 50% of their own communication range with a probability of 80%.

5 Conclusion

We have described a novel algorithm that calculates location information to a high degree of accuracy, is simple to implement, and that requires minimal bandwidth, computation and storage. In addition to performing location discovery, the algorithm's requirements are slight enough that it can be run continually in order to maintain each node's computed location in the face of movement.

The algorithm is entirely distributed, and thus reliable in the face of node failures, and runs on a homogeneous network. In addition, anchor nodes that know their own location can be introduced into the network, which will seamlessly integrate the higher-grade location information and so reduce errors.

Algorithm performance has been shown to scale gracefully in the face of adverse conditions such as grievous node manufacturing defect rates and communication channel faults, both persistent and intermittent.

In network deployments where node complexity and cost is limited, acceptable performance can be obtained with even a minimal number of optical transceivers. This characteristic highlights the value and utility of direction information in network localisation.

Although not addressed in this paper, the performance of the algorithm could be improved, especially in sparse networks, by taking range information into account. Unlike radio communications, the energy of an optical transmission reduces in a predictable manner with increasing distance. In addition, the emission pattern of LEDs is very simple to characterise. This allows nodes to make accurate estimations of inter-node distances based on the received power of a transmission. The range information would further limit the range of possible locations for a node, and thus reduce location error.

References

[1] Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: Mobile Computing and Networking. (2000) 243–254

[2] Römer, K.: The lighthouse location system for smart dust. In: Proceedings of MobiSys 2003 (ACM/USENIX Conference on Mobile Systems, Applications, and Services), San Francisco, CA, USA (May 2003) 15–30

[3] Zhang, J., Yan, T., Stankovi, J., Son, S.: Thunder: Towards Practical, Zero Cost Acoustic Localization for Outdoor Wireless Sensor Networks. ACM SIGMOBILE Mobile Computing and Communications Review **11**(1) (2007) 15–28

[4] Wang, L., Zeadally, S.: Landscape: a high performance distributed positioning scheme for outdoor sensor networks. Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on **3** (2005)

[5] Stoleru, R., Vicaire, P., He, T., Stankovic, J.A.: Stardust: a flexible architecture for passive localization in wireless sensor networks. In: SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems, New York, NY, USA, ACM (2006) 57–70

[6] Sichitiu, M., Ramadurai, V.: Localization of wireless sensor networks with a mobile beacon. Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on (2004) 174–183

[7] Pathirana, P., Bulusu, N., Savkin, A., Jha, S.: Node localization using mobile robots in delay-tolerant sensor networks. IEEE Transactions on Mobile Computing **4**(3) (2005) 285–296

[8] Niculescu, D., Nath, B.: Ad hoc positioning system (APS). GLOBECOM-NEW YORK- **5** (2001) 2926–2931

[9] Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. Lecture Notes in Computer Science (2003) 333–348

[10] Bulusu, N., Heidemann, J., Estrin, D.: Gps-less low cost outdoor localization for very small devices.

IEEE Personal Communications Magazine **7**(5) (October 2000) 28–34

[11] He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-free localization schemes in large scale sensor networks (2003)

[12] L. Doherty, K. S. J. Pister, L.E.G.: Convex position estimation in wireless sensor networks. In: Proc. of the IEEE Infocom. (2001) 1655–1663

[13] Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.: Localization from Mere Connectivity. In: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, ACM New York, NY, USA (2003) 201–212

[14] Moses, R., Krishnamurthy, D., Patterson, R.: A Self-Localization Method for Wireless Sensor Networks. EURASIP Journal on Applied Signal Processing **2003**(4) (2003) 348–358

[15] Moore, D., Leonard, J., Rus, D., Teller, S.: Robust Distributed Network Localization with Noisy Range Measurements. In: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM New York, NY, USA (2004) 50–61

[16] Ma, J., Gao, M., Zhu, Y., Ni, L.: Anchor-free Localization with Refinement in Sensor Networks. submitted for publication, July (2004)

[17] Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-support System. In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, New York, NY, USA, ACM Press (2000) 32–43

[18] Elnahrawy, E., Li, X., Martin, R.: The limits of localization using signal strength: a comparative study. Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on (2004) 406–414

[19] Whitehouse, K., Karlof, C., Culler, D.: A practical evaluation of radio signal strength for ranging-based localization. ACM SIGMOBILE Mobile Computing and Communications Review **11**(1) (2007) 41–52

[20] Hazas, M., Hopper, A.: Broadband ultrasonic location systems for improved indoor positioning. IEEE Transactions on Mobile Computing **5**(5) (2006) 536–547

[21] Akella, J., Yuksel, M., Kalyanaraman, S.: A Relative Ad hoc Localization Scheme using Optical Wireless. Communication Systems Software and Middleware, 2007. COMSWARE 2007. 2nd International Conference on (2007) 1–8

[22] Kelly, I., Martinoli, A.: A scalable, on-board localisation and communication system for indoor multi-robot experiments. Sensor Review (Journal of) **24:2** (April 2004) 167–180

[23] Arvind, D.: Speckled computing. In: Technical Proceedings of the 2005 NSTI Nanotechnology Conference. (May 2005)