Testbed Implementation and Refinement of a Range–Based Localization Algorithm for Wireless Sensor Networks

Riccardo Crepaldi, Paolo Casari, Andrea Zanella, Michele Zorzi
DEI — University of Padova — Via G. Gradenigo, 6/B, I–35131 Padova (PD), Italy
{firstname.lastname}@dei.unipd.it

ABSTRACT

Many wireless sensor networks (WSN) applications require a more or less accurate degree of knowledge of the sensing nodes' positions. This information can be even crucial in rescue scenarios, battlefield surveillance, object movement tracking and so on. Furthermore, some routing protocols for WSNs may require geographic data to perform handshakes and forwarding operations. In our paper, we deploy and study ROCRSSI+, a refinement of a localization algorithm based on received signal strength measures. We then evaluate the algorithm by simulation and by implementation on an indoor testbed, identifying which impairments could hamper its functioning, and illustrating how to compensate for them.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are recently experiencing a steeply increasing level of popularity. Potentially, WSNs could achieve seamless data gathering and environmental monitoring, providing end users with the capability to remotely acquire and process any kind of environmental information. It is foreseen that in the future, hundreds to thousands of such tiny devices could be spread across a certain area, possibly without control on their final positioning. These sensors will then start communicating through the wireless channel, exchanging information and/or responding to external queries, without need for supervision or central administration.

Many applications for WSNs, either commercial or military, could require consciusness of a node's position. For instance, when tracking the movement of wild animals moving over a selected territory, the entity gathering data needs to know where sensors are reporting from. When controlling the air conditioning system inside a building, automatic monitoring of a circumscribed area could be needed. If a house should collapse, e.g. due to earthquakes, already in–place sensors could be reached through their known position and interrogated, e.g., in order to catch human lifesigns or to locate sensible equipment.

Also, location information provides a means to put geographic MAC and routing protocols at work. For example, Geographic Adaptive Fidelity (GAF) [12] requires a virtual grid setup as a routing substructure, and to set this grid without external intervention requires knowledge of the sensors' locations. Geographic Random Forwarding (GeRaF) [14,15] assumes that each node knows its own location and the location of the final data destination (the sink) and uses them both for advancing by selecting the most convenient relay in a contention–based fashion. GEAR [13] uses location information to direct query messages toward the intended network zones only, without having to flood them throughout the whole network.

Mobility 06, Oct. 25–27, 2006, Bangkok, Thailand.

For the explained reasons, a means for each sensor of calculating its own location is a crucial component of any WSN. This capability is available for free through the Global Positioning System (GPS), as well as the european GALILEO [1], whose satellites are undergoing deployment and will be completely set up in the forthcoming years. Nonetheless, GPS receivers may prove to be very energy–demanding and even difficult to integrate in a small sized wireless sensor. Furthermore, the GPS system doesn't provide reliable results in an indoor environment. GPS–free automatic localization methods are also available, and are typically based on exchanging some amount of information among location–aware and inquiring nodes. Section 2 is meant to resume some of these approaches, highlighting the most significant for our work. The remainder of this paper is oganized as follows. In Section 3 and 4, we carry out a thorough description of the background of our localization algorithm, specifying its differences with available approaches. In Section 5, we report issues and problems arising in the implementation of our algorithm on real WSNs, and describe how to solve them. Finally, we conclude our paper in Section 6.

2. RELATED WORK

The literature on localization algorithms is typically classified according to two main criteria. *Centralized algorithms* rely on a single node with greater computational resources to gather data from the whole network and estimate the position of every other node. On the contrary, *distributed algorithms* have nodes exchange signaling messages, so that each can autonomously estimate its own position, and possibly propagate it to its neighbors. Centralized algorithms can exploit the higher knowledge of a center node to favor precision, but may require a significant amount of signaling and computational effort at that node, especially when every single sensor must be provided precise positioning information. Achieving it distributely requires instead less messaging and may complete faster, especially in denser networks.

Another important classification criterion distinguishes among *range–based* and *range–free* algorithms, the first class including those methods that estimate distances based on some measured quantity. The translation from a physical quantity to a distance is called a *ranging* method. For example [5], range estimates may be obtained from the absolute Time of Arrival (ToA) or Time Differece of Arrival (TDoA), which require perfect synchronization among nodes

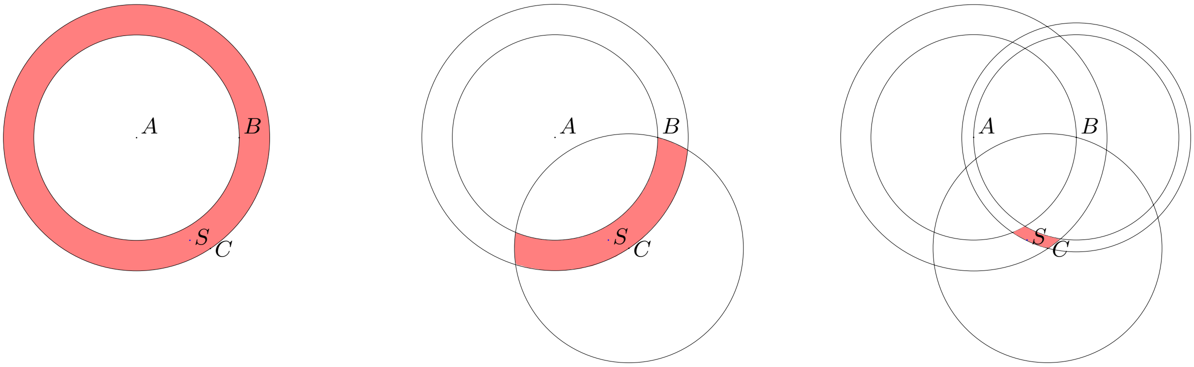

Figure 1: Behavior of the ROCRSSI localization algorithm with 3 beacons.

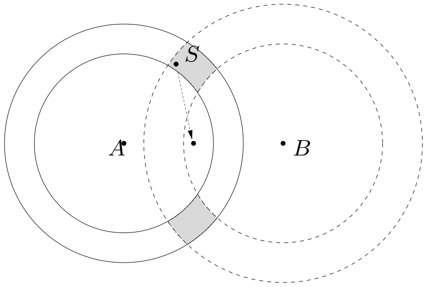

Figure 2: The problem of disjoint areas in ROCRSSI.

and a very fast hardware clock, and infer distance from propagation time. Also, other methods use the Angle of Arrival (AoA) of the signal, but require multi–antenna processing, and may experience decreasing precision at greater distances. All these methods are hence infeasible for sensors, which embed a small processing capability and typically cheap hardware. Another quantity which can be considered is the average received signal power, generally available as a Received Signal Strength Indicator (RSSI). Among all, this is the most suitable method for sensors, since RSSI is usually a default output of embedded node radios. *Range–free* methods are also available that directly operate on the same quantities described before, avoiding the arise of errors in the conversion to distance estimates. In any case, localization algorithms need one or more nodes, commonly referred to as beacons, which know their position with very high accuracy.

Lateration [7] is a range–based algorithm that has any node compute its own position by intersecting at least three circles, each centered on a different beacon and with radius equal to the estimated distance between the beacon and the node itself. Ideally, the intersection should be a single point on a surface, but due to channel and environment impairments, this intersection as a matter of facts identifies an area where the node is likely to be found.

Cooperative Location Sensing [6] is similar to lateration, but finds the intersection among rings instead of circles. The width of each ring is determined by the standard variation of the ranging error.

Min–Max [8, 10] finds the intersection of squares circumscribed to circles identified in the same way as lateration. This method guarantees more robustness to the effects of multipath fading at a very low computational cost. Its main drawback is an intrinsically scarce accuracy. For this reason, Min–Max could be used first to find a rough position estimate, before refining it with more complex algorithms.

With RADAR [3], a number of base stations collect RSSI values from any unknown node and estimate the node position based on this information and a previously derived model for the propagation environment. RADAR finds its best use indoor.

Ring Overlapping based on Comparison of RSSI (ROCRSSI) [4,9] is the start point of our approach, so we will describe it thoroughly in Section 3.

3. ROCRSSI

ROCRSSI is a range–free distributed algorithm, which operates based on the comparison of RSSI values. Namely, each node receives a signal from neighboring beacons and uses it to bound iteratively the area where the node is placed. Its functioning is summarized in Fig. 1.

At first, each beacon transmits at random, and all other beacons evaluate the received signal strength (indicated with p_{AB}, where A is the receiver and B is the transmitter). The sensed values are stored in a table. Once this step has been completed, all beacons transmit their own table to the other nodes. Each node compares the RSSI obtained from a beacon signal with the RSSI measures reported in the packet received from the beacon, eventually determining a circular ring they belong to. This operation is repeated for every beacon node, and the derived areas are intersected. The outcome of this operation is an estimation of the zone where the node is placed, with a further refinement for every new table received. Each time a table is evaluated, one of the following cases applies:

- $p_{AB} > p_{AC} > p_{AS}$ or $p_{AC} > p_{AB} > p_{AS}$. In this case the node assumes to be placed internal to the circle centered in A with radius equal to $\min\{\overline{AB}, \overline{AC}\}$.
- $p_{AB} > p_{AS} > p_{AC}$ or $p_{AC} > p_{AS} > p_{AB}$. In this case the node assumes to be placed inside the circular ring centered in A with inner radius $\overline{AB}$ and $\overline{AC}$.
- $p_{AS} > p_{AB} > p_{AC}$ or $p_{AS} > p_{AC} > p_{AB}$. In this case the node ignores the comparison with the RSSI table broadcast by the current beacon, because, along with the assumptions in [4], the area outside the datermined circles is assumed to be infinite. This would result in no improvement to the current position estimate.

After having received a table from every neighboring beacon, a node has determined the best (i.e., smaller) intersection of rings it is contained within, and calculates the barycenter of this intersection as its position measurement.

This algorithm presents three main drawbacks. First, channel variations may influence RSSI measures. If A and B are two beacons, with $\overline{AS} > \overline{BS}$ but the RSSI corresponding to the transmission of A is greater, S may identify a wrong area which could have no intersections with other previous or further areas. Indeed, channel fluctuations affects all ranging methods based on RSSI or received power. A feasible countermeasure is to quantize the neighboring area, dividing them into smaller cells. Each cell is assigned a counter, which is increased every time the cell belongs to an identified area. At the end of the entire process, the union of cells with greater count value is assumed to approximate the area where the node is located. A second problem also arises from ranging errors, which may eventually result in the calculation of a non-connected intersection of surfaces, as depicted in Fig. 2. In this case, the surface barycenter could result external to the calculated surface.

A third, more subtle, inefficiency, comes from assuming that nodes are located inside an unbounded area. As already explained, this event forces to neglect every measure detecting the node farther than any beacon (the last of the three aformentioned cases). In order to reduce this loss of valuable information, we have made some assumptions and further refinements that have led us to the definition of ROCRSSI+.

4. ROCRSSI+

With ROCRSSI+, the sensor deployment area is bounded, and has a square shape. With this assumption, every time a node finds itself outside the range of a beacon as read from the received table, it can assume to be placed outside the beacon range, but inside the deployment area. This ameliorment is depicted in Fig. 4. With ROCRSSI, node S finds itself outside the range of beacons A and B. Without the refinement allowed by the ongoing measurement, the computed surface intersection overlaps with the coverage area of node C, hence the best position estimate made by node S becomes the position of node C. With ROCRSSI+, instead, the only area outside the coverage of A and B is considered, and the position estimation of S is affected by a smaller error, resulting in the calculation of point E.

With the aim of improving further the performances of ROCRSSI+, we have also considered the CS 252 refinement [11]. Basically, it is based on the iterative minimization of an objective function linking location estimates and positioning errors, under the assumption that the distance between any node i and any beacon j can be modeled as a gaussian random variable, with average value equal to the measured distance.

Before implementing ROCRSSI+ on real sensor motes, we have tested it through a MATLAB simulation. We have considered network with 6×8 nodes in a grid topology, with nearest neighbors 1 m apart. This configuration is the same we will use next for real world evaluations. We have run ROCRSSI+ over considering a simplified propagation model, whereby a node can communicate with all nodes no farther than a predefined coverage range. Due to our particular setting, the network was fully connected, or in other words, any sensor can hear any other.

Simulation results are given in Fig. 3, which depicts the average localization error (in cm) as a function of the number of nodes acting as beacons. The figure compares ROCRSSI with ROCRSSI+ both in a grid and in a random placement, and shows that the best performance is obtained through ROCRSSI+, as expected.

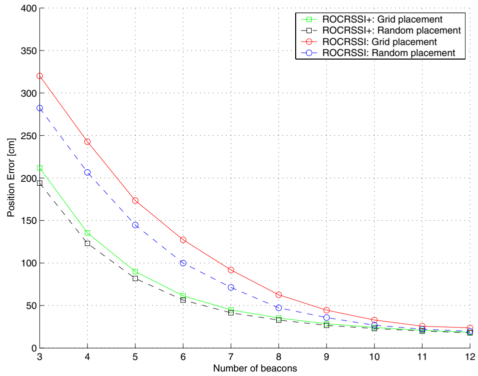

Figure 3: Simulation results for ROCRSSI+.

5. TESTBED DEPLOYMENT AND IMPLEMENTATION OF ROCRSSI+

The whole implementation has been conducted on EyesIFXv2 nodes by Infineon Technologies, equipped with a TDA5250 radio chip and a MSP430 microcontroller. Each node transmits through an external quarter-wavelength antenna.

All nodes run the Operating System TinyOS [2], and the necessary extensions to drive the localization algorithm. USB cables are connected to all sensors using cascaded hubs, and are used to supply power, program nodes, and gather performance metrics. A schematic representation of this network setup is given in Fig. 5.

In our tests, 48 nodes, displaced in a 8 x 6 grid, have been deployed both in an indoor and in an outdoor environment, in order to test ROCRSSI and ROCRSSI+ under different propagation conditions. The grid is 9 m $\times$ 10 m, the nodes are 1.25 m far over one direction, 1.5 m over the other. During the outdoor tests, all nodes were placed 1 m above the terrain, whereas during the indoor tests, all nodes were attached to a grid of cables laid 80 cm below the ceiling of our laboratory. Both indoor and outdoor tests have been repeated as long as needed to get statistically meaninful results.

The average measured localization error is shown in Fig. 6 for both ROCRSSI and ROCRSSI+ as a function of the number of beacons with and without the CS 252 refinement. This error is the mean of tens of experiments as described before. As seen from the picture, the best localization performance is obtained using ROCRSSI+ along with CS 252, as expected. On the contrary, ROCRSSI alone undergoes the worst localization errors.

As regards the indoor environment, the results are given in Fig. 7. In this case, both algorithms experience a larger localization inaccuracy, even if having the same order of magnitude obtained for outdoor results. These errors are much less sensitive to the increase of beacons , and are mainly due to the harsher propagation environment. A laboratory room is jam-packed with microwave reflection sources, like walls, tables, steel boards, computer cases, and similar objects. The overall received signal strength is therefore determined by the superposition of multiple contributions, because electromagnetic waves may constructively or destructively combine, altering the sensed RSSI value. Correspondingly, this has an impact on localization performance, because ROCRSSI+ has to operate on estimated surface intersections that do not approximate reality to a

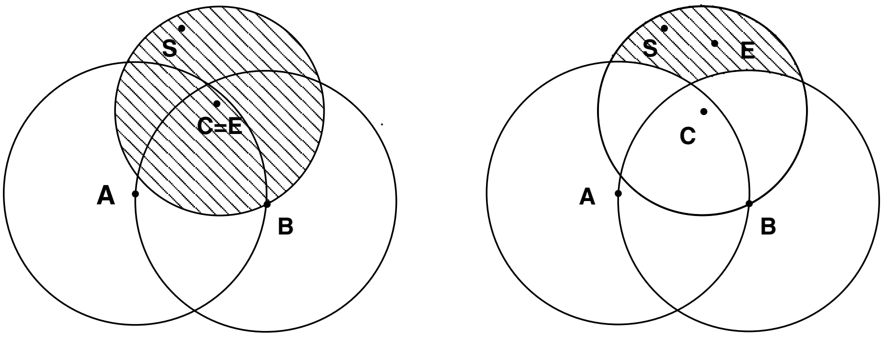

Figure 4: Difference between ROCRSSI and ROCRSSI+.

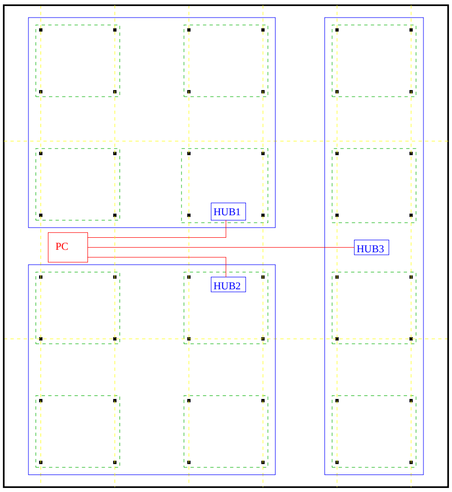

Figure 5: Schematic representation of our testbed deployment.

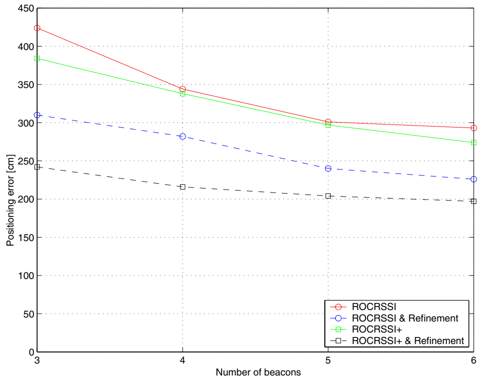

Figure 6: Localization error as a function of the number of beacons in an outdoor scenario.

sufficiently high degree.

Interesting to note, CS 252 gives ROCRSSI+ a better edge on ROCRSSI in the outdoor environment. This fact is based upon the less harsh propagation conditions experienced outdoor, that allow the gaussian approximation for node–beacon distances to be more accurate, instead of being disrupted by faded or interference-affected RSSIs.

Indeed, even if encouraging, the results obtained from a real testbed implementation are affected by a very large error, both indoor and outdoor. While on one hand confirming that real testbed implementation is needed for assessing the performances of any algorithm, these inaccuracies call for a better understanding of the source of such impairments, before putting ROCRSSI+ to work. To this extent, we have conducted some extensive measurement campaign to thoroughly characterize the power received by the EyesIFXv2 radio chip, and its translation into an RSSI value.

Even if giving detailed link level results is out of the scope of this paper, it is nonetheless interesting to highlight a potentially hampering hardware impairment. A test has been conducted to investigate the relationship between the distance of a node and the RSSI value output by the radio chip, as this is the main brick the whole localization algorithm is built upon. In an outdoor environment, we have placed one fixed transmitter, and 5 different nodes used as simultaneous receivers. All nodes have been deployed in turn in the same places, at a distance of $1, 2, \ldots, 10$ m, and the RSSI values measured have been evaluated and averaged over many transmissions at each distance. All nodes where placed 1 m over a metal grid, used as a ground plane.

The results of this experiment are depicted in Fig. 8. Each curve gives the average RSSI value measured by the corresponding node as a function of distance. Even if the trend of the curves is almost the same, they are all affected by a constant offset depending on the considered nodes, which causes RSSI measures to diverge from the correct value. From a practical point of view, this offset has to be compensated for, before having meaningful positioning estimations.

In order to corroborate this deduction based on field experiments, we have also conducted some MATLAB simulations. We have considered a larger network, with 121 nodes arranged in both a grid and a random (connected) topology over an area of 100 m $\times$ 100 m.

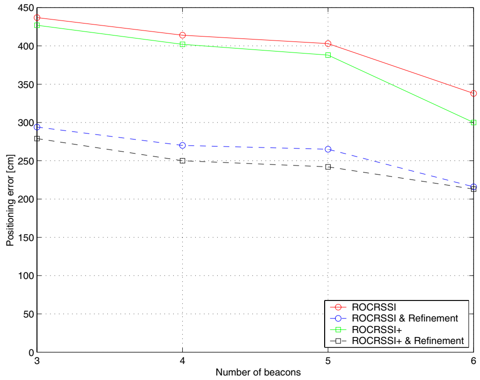

Figure 7: Localization error as a function of the number of beacons in an indoor scenario.

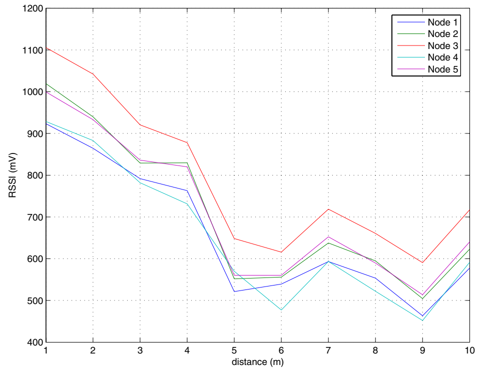

Figure 8: RSSI versus distance for 5 different nodes in an outdoor environment.

Each node has been assigned a random offset uniformly chosen in the interval $[-100\text{mV}, +100\text{mV}]$ which affects all RSSI measures. The simulation results (shown in Fig. 9) are aligned with the field experiments output, and show a substantial increase in the average positioning error, even with a great number of in-range beacons, both for a grid and a random topology. Furthermore, the algorithm offering the best performance is again ROCRSSI+.

These considerations suggest that for improving the performance of ROCRSSI+ and any other localization algorithm, sensor calibration is needed to detect RSSI measurement offsets which need compensation. In our case, we have used the extensive field evaluation to assess the value of this offset on a per node basis, and showed that in fact it has a significant impact on performance.

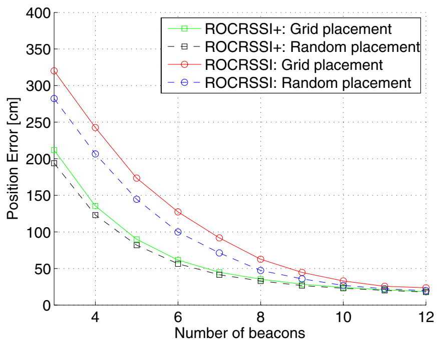

Figure 9: Simulation of the localization error as a function of the number of beacons, both in a grid and random topology.

6. CONCLUSIONS

In this paper, we have described a testbed implementation of a beacon–based localization algorithm for Wireless Sensor Networks. We have first focused on the numerical simulation of the algorithm itself, namely ROCRSSI+, which showed to offer significantly good results, in terms of increasingly accurate position estimates for a greater number of position–aware beacons. We have then deployed this algorithm over an indoor 48-nodes network. We have tested the implementation of ROCRSSI+ and shown that its unexpectedly low convergence was due to the misevaluation of received signal strengths, which proved to be very different among different nodes. We then paved the way for eliminating this problem through RSSI offset compensation.

7. ACKNOWLEDGEMENTS

The authors wish to thank Matteo Andretto and Sebastiaan Blom for their help in implementing the first version of ROCRSSI+.

8. REFERENCES

[1] The Galileo Project – GALILEO Design consolidation, European Commission, 2003. `http://www.esa.int/`.

[2] TinyOS: An open source operating system for WSN. www.tinyos.net.

[3] P. Bahl and V. N. Padmanabhan. RADAR: an in–building RF–based user location and tracking system. In *Proc. IEEE INFOCOM*, pages 775–784, 2000.

[4] T. H. Chong Liu, Kui Wu. Sensor localization with ring overlapping based on comparison of received signal strength indicator. In *Proc. IEEE MASS04*, Fort Lauderdale, FL, Oct. 2004.

[5] F. Koushanfar and S. Slijepcevic and M. Potkonjak and A. Sangiovanni–Vincentelli. *Ad Hoc Wireless Networking*, chapter Location discovery in ad hoc wireless sensor networks. Kluwer Academic Publisher, 2003. X. Cheng, X. Huang and D. Z. Du, *eds*.

[6] C. Fretzagias and M. Papadopouli. Cooperative location–sensing for wireless network. In *Proc. IEEE PerCom*, 2004.

[7] J. Hightower and G. Borriello. A survey and taxonomy of location system for ubiquitous computing. Technical report, University of Washington, CSE Dept., Seattle, WA 98195, Aug. 2001. UW-CSE 01-08-03.

[8] K. Langendöen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.

[9] C. Liu and K. Wu. Performance evaluation of range–free localization methods for wireless sensor networks. In *Proc. IEEE IPCCC*, Phoenix, Arizona, Apr. 2005.

[10] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proc. ACM Int'l. Workshop on Wireless Sens. Nets. and Apps.*, pages 112–121, Atlanta, GA, Sept. 2002.

[11] X. Nguyen and T. Rattenbury. Localization algorithms for sensor networks using RF signal strength CS 252 Class project. Technical report, University of California at Berkeley, may 2003.

[12] Y. Xu, J. Heidemann, and D. Estrin. Geography–informed energy conservation for ad hoc routing. *Proc. 7th ACM MOBICOM*, pages 70–84, 2001.

[13] Y. Yu, D. Estrin, and R. Govindan. Geographical and energy–aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical Report 010023, UCLA Comp. Sci. Dept., May 2001.

[14] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance. *IEEE Trans. Mobile Comput.*, 2(4):349–365, 2003.

[15] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Trans. Mobile Comput.*, 2(4):337–348, 2003.