UNIVERSITY OF CALIFORNIA
Los Angeles

Self-Configuring Localization Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Nirupama Bulusu

2002

The dissertation of Nirupama Bulusu is approved.

Mani B. Srivastava

Stefano Soatto

John Heidemann

Mario Gerla

Deborah L. Estrin, Committee Chair

University of California, Los Angeles

2002

To my parents, and Anu.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGMENTS

What an incredible experience graduate school has been! Working on this dissertation, and graduate school in general, have challenged, rewarded and transformed me in ways I could not even have anticipated when I began almost five years ago. Several people have lent me their support, guidance, and help in this process; and now I get the opportunity to thank them — I hope I can do so meaningfully (and at length, ahem) and without forgetting anyone!

First and foremost, I would like to thank my advisors, Professors Deborah Estrin and John Heidemann. Deborah took me on as a Ph.D student without even meeting me in person; and John, without knowing much about me. It was indeed a stroke of enormous good fortune that led to my working with them in a very young and fertile research area; in fact, I picked up my thesis topic in my very first research meeting with John and Deborah.

As any of her students or colleagues would attest, one of the most unforgettable aspects of working with Deborah is just knowing someone like her. Deborah combines enormous energy, wit and intellect with amazing clairvoyance and human qualities. She has handled my progress as a Ph.D student with tremendous sensitivity. For that and for her endless supply of exciting research problems; for the way she always helped me think about research from a wider perspective; for instilling in me the values of collaboration; and for all her down-to-earth advice and *constant* encouragement, I will be forever grateful. It has been my privilege and honor to be associated with Deborah right from her days as an energetic professor at USC exploring the possibilities of device networks to her new role as the visionary founder-director of a new center for embedded networked sensing at UCLA.

In no less measure, John has guided and influenced me. John is an excellent researcher and maintains the highest standards for himself and his students. John is also an *awesome* advisor — he is extremely smart and attentive, really enjoys advising, is incredibly helpful, responsive, and always has the best interests of his students at heart. While the process of getting a Ph.D is hardly mechanical, it still requires organization and taking a number of small steps. Without John's help, I would have certainly floundered in this regard. John has been extremely generous with his time, and more patient and understanding than I probably deserve. His advice, care and painstaking attention to detail has really helped me sharpen my research, analysis, writing and presentation skills. And yet he has always given me a lot of freedom in my work. My appreciation for John grows each time I interact with him. I am incredibly grateful to John for his collaboration, guidance, wisdom and constant encouragement and support. I hope I can inherit and live up to his high standards!

I thank Professors Mario Gerla, Richard Muntz, Stefano Soatto, and Mani Srivastava for serving on my qualifying exam and dissertation committees. Their suggestions have greatly improved my dissertation work. In particular, I thank Professor Muntz and Soatto for emphasizing the importance of mathematical formulation, and Professor Soatto for pointing out interesting related work in computer vision. Mani approaches even the smallest task with dedication and commitment, and his responsiveness to student email is legendary. Mani's invaluable insight on localization and density-based protocols during my qualifying exam has helped me make significant improvements on this dissertation work.

In addition to my committee members, I have benefitted immensely from my interactions with three professors at USC — Ashish Goel, Ramesh Govindan and Gaurav Sukhatme. Ashish suggested facility location as a possible problem formulation for addressing issues of node placement. His wonderful course on Randomized Algorithms at USC has provided me with a new set of algorithmic tools for networking

research. Ramesh Govindan attended all my presentations and gave me detailed feedback through our interactions on the SCADDS project. I admire Ramesh not only for his ingenious ideas, but also for his rigorous writing standards. Gaurav Sukhatme taught one of the most interesting and engaging courses of my graduate career which exposed me to all the cool things roboticists do! Several of the ideas contained in this thesis sprung up during our interactions on the SCOWR project and in his "Sensing and Planning in Robotics" course during that productive Fall 1999 semester. Gaurav has also been an active mentor and strong well-wisher right since my first year in graduate school. I thank Ashish, Gaurav and Ramesh for all that I have learnt from them and for their support, and look forward to interacting with them in the future.

I thank Verra Morgan, our Computer Science Department's Graduate Assistant; for fending off the bureaucracy and for her consistent helpfulness; Roberta Nelson for ensuring timely paychecks and fee remittances, and Uyen Nguyen for her promptness in processing travel reimbursements and ordering lab goodies!

Graduate school would not nearly half as intellectually stimulating without my fellow students — they make it downright fun! I would like to thank all members of the USC Computer Networks and Distributed Systems Laboratory — Ahmed Helmy, Chalermek Intanagonwiwat, Satish Kumar, Pavlin Radoslavov, Mohammed Rahimi, Reza Rejaieshustari, Puneet Sharma, Fabio Silva, Hongsuda Tangamunarunkit, Wei Ye, Haobo Yu, Ya Xu and Jerry Zhao; and the UCLA Laboratory for Embedded Collaborative Systems — Solomon Bien, Daveey Braginsky, Jamie Burke, Naim Busek, Alberto Cerpa, Jeremy Elson, Deepak Ganesan, Lewis Girod, Ben Greenstein, Vladimir Bychkovskiy, Ani Nahapetian, Denis Pyrelyubskiy, Tom Schoellhammer, Thanos Stathopoulos, Tommy Tran, Hanbaio Wang and Yan Yu. In particular, I thank Jeremy, Lew, Alberto, Fabio, Yan, Vlad, Hongsuda, Pavlin and Deepak for their general attitude of inclusiveness, for putting up with my various quirks and for cheering me up when the

going got tough.

Jeremy Elson developed two crucial components of the experimental testbed — the Radiometrix RPC device drivers and *moted*; without which the experimental results in this thesis would not have been possible. Jeremy also deserves enormous credit for getting the LECS lab functional in a very short time when we moved from USC to UCLA. Time and again, Jeremy has selflessly prioritized the interests of other lab members over his own; and done so with a wonderfully droll sense of humor.

I was greatly assisted in my experimental data collection by two undergraduates — Tommy Tran who developed the mote visualization software and Vladimir Bychkovskiy (now graduate!). I thank them for their efforts, enthusiasm and overall sweet disposition.

My understanding of localization has been enriched by my discussions with Lewis Girod. Lew first came up with the term federated, for referring to the problem of combining coordinate systems. Lew is a consummate researcher; he is critical, extremely dedicated and has tremendous integrity in his research approach. I hope I can emulate Lew's attitude toward research. I also thank Bhaskar Krishnamachari (Cornell) for sharing his analysis of phase transitions.

I was fortunate to spend two summers away from school in industrial internships at Sun Microsystems Ltd. and Xerox Palo Alto Research Center (PARC). At Sun Microsystems, I thank Michael Speer, Charles Perkins and Kuang-Yeh Wang. At PARC, I thank Feng Zhao, Patrick Cheung, John Gilbert, Leonidas Guibas, Jaewon Shin and especially Jim Reich for all that I learnt from them working on the CoSense project. It was fascinating to contrast their information-theoretic perspective on localization, tracking and routing with my networking perspective, and I hope the debate continues.

I have been extremely fortunate to have had two great mentors and role models in my engineering career — Kavitha and Suchitra Raman. They are genuine well-wishers and I am indebted to them for all their guidance through my undergraduate as well as graduate studies — especially, in making *informed* career decisions. Jitender Gill has expanded my horizons, helped me overcome my self-absorption, and has been a true friend over the past five years.

My most important acknowledgement is to my family members, who have filled my non-academic life with happiness and have been my source of resilience in difficult times. Most significantly, I refer to my sister Anupama and her family (Bipin and Megha!), who always make sure that I am having fun; my mother who has never valued anything more than my ultimate happiness and contentment; and my father, who has always lovingly encouraged me to pursue my passions and *always* believed in me, even when I did not. I love them and owe this to them.

Los Angeles, California *Nirupama Bulusu*

October 2002

I gratefully acknowledge the grants that supported my research at UCLA. My research was supported in part by funding from the National Science Foundation (NSF) through grant no. NSF-ANI-9979457 on the SCOWR project, and from the Defense Advanced Research Projects Agency (DARPA) through grant no. DABT63-99-1-0011 on the SCADDS project. Additional funding was provided by the UCLA School of Engineering and Applied Sciences (HS-SEAS), Intel Corporation, and Cisco Systems Incorporated.

VITA

1997	B.Tech. (Computer Science and Engineering), Indian Institute of Technology, Madras (IIT-M), Madras, Tamil Nadu, India.
1997–1998	Graduate Teaching Assistant, Department of Computer Science, University of Southern California (USC), Los Angeles, California.
1998	Summer Intern, Advanced Technology Development Group, Sun Microsystems Incorporated, Menlo Park, California.
1998–2000	Graduate Research Assistant, Department of Computer Science, University of Southern California/Information Sciences Institute (USC/ISI), Marina del Rey, California.
2000	M.S. (Computer Science), University of Southern California (USC), Los Angeles, California.
2000–2002	Graduate Student Researcher, Laboratory for Embedded Collaborative Systems (LECS), Department of Computer Science, University of California at Los Angeles (UCLA), Los Angeles, California.
2001	Summer Intern, Systems and Practices Laboratory (SPL), Xerox Palo Alto Research Center (PARC), Palo Alto, California.
2002	Lecturer, Department of Electrical Engineering, University of Southern California (USC), Los Angeles, California. Taught introductory undergraduate course on computer networks.

PUBLICATIONS

GPS-less Low Cost Outdoor Localization for Very Small Devices. Nirupama Bulusu, John Heidemann, Deborah Estrin. IEEE Personal Communications Magazine, October 2000.

Effects of Detail in Wireless Network Simulation. John Heidemann, Nirupama Bulusu, Jeremy Elson, Chalermek Intanagonwiwat, Kun chan Lan,Ya Xu, Wei Ye, Deborah Estrin, Ramesh Govindan. In Proceedings of the Communications Networks and Distributed Systems Modeling and Simulation Conference (CNDS), January 2001.

Adaptive Beacon Placement. Nirupama Bulusu, John Heidemann, Deborah Estrin. In Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-21), Phoenix, Arizona, USA. April 2001.

Density-adaptive Algorithms for Beacon Placement in Wireless Sensor Networks. Nirupama Bulusu, John Heidemann, Vladimir Bychkovskiy and Deborah Estrin. UCLA Computer Science Technical Report UCLA-CS-TR-010013, May 2001.

Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems. Nirupama Bulusu, Deborah Estrin, Lewis Girod and John Heidemann. In Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 01), July 2001.

Tradeoffs in Location Support Systems: The Case for Quality Expressive Location

Models for Applications. Nirupama Bulusu, Deborah Estrin and John Heidemann. In Proceedings of the Ubicomp 2001 Workshop on Location Modeling for Ubiquitous Computing, Atlanta, Georgia, USA. October 2001.

Using Geospatial Information in Sensor Networks. John Heidemann and Nirupama Bulusu. In Proceedings of the Workshop on Intersections between Geospatial Information and Information Technology, Arlington, Virginia. National Research Council. October 2001.

Scalable, Ad Hoc Deployable, RF-based Localization. Nirupama Bulusu, John Heidemann and Deborah Estrin. In Proceedings of the Grace Hopper Celebration of Women in Computing Conference 2002, Vancouver, British Columbia, Canada, October 2002.

ABSTRACT OF THE DISSERTATION

Self-Configuring Localization Systems

by

Nirupama Bulusu

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2002

Professor Deborah L. Estrin, Chair

Recent technological advances have fostered the emergence of small, low-power devices that integrate micro-sensing and actuation with on-board processing and wireless communications capabilities. Through distributed coordination, pervasive networks of micro-sensors and actuators are expected to revolutionize the ways in which we understand and construct complex physical systems. Fundamental to such coordination is *localization*, or the ability to establish spatial relationships among objects.

In this dissertation, we address the challenges involved in localization for very large, ad hoc deployed sensor networks. Although several localization technologies have been proposed in the past few years, none currently satisfies all our requirements because no single localization system is simultaneously scalable, ad hoc deployable and accommodating of the hardware constraints of very small devices. Our thesis is that all these issues can be solved simultaneously by a *self-configuring* localization system that autonomously adapts to its environmental dynamics. Our approach is based on localized adaptive algorithms that self-configure to exploit both the local processing on each sensor node, as well as the redundancy across densely-deployed sensor nodes.

First, to accommodate device constraints, we adopt a low cost, hardware-independent localization approach for very small devices that leverages the existing radio (RF) com-

munications capabilities of such devices and does not require any other sensors.

Second, to scale to very large sensor networks, we develop a decentralized, self-localization methodology for devices. Instead of relying on a central server to compute their positions, devices themselves perform a localized location computation based on radio connectivity constraints to a small number of nearby beacons (nodes with known positions), obtained by listening to radio broadcast advertisements of beacons.

Third, we need to ensure a uniform localization granularity in dynamic, unpredictable environments with numerous radio propagation vagaries. One solution to this problem is to extensively instrument and model the environment, a priori. Unfortunately, this approach does not scale well. Instead, we advocate and develop a self-configuring mechanism in which beacons themselves measure and adapt to their environment and availability of neighboring beacons.

Finally, we quantitatively analyze the impact of beacon density on localization. We show that proximity based localization using only local information saturates at a threshold beacon density μ_{thresh}. We develop various self-configuring algorithms for incremental beacon placement for sparse beacon deployment. For dense beacon deployment, it is desirable to keep the operational beacon density close to μ_{thresh} to reduce the probability of self-interference amongst beacons and to conserve energy. We develop a parameterized algorithm (tunable according to radio parameters) to adjust the duty cycle of beacons based on the availability of other beacons in the neighborhood to realize a low operational density.

These techniques form the bases of our self-configuring localization system. We have implemented it as a user-level library on two test-beds, Radiometrix RPC-418 radios, and motes with RFM radios. We evaluate and demonstrate the effectiveness of our localization system in terms of the performance of the basic localization algorithms, as well as the beacon placement techniques to adapt it to noisy environments.

CHAPTER 1

Introduction

To begin, begin.

— William Wordsworth

1.1 Motivation: Location-Aware Computing

In the last decade, we have witnessed a burgeoning amount of research and commercial interest in the area of ubiquitous computing. As envisioned by Mark Weiser [Wei93, Wei91] in the early nineties, in ubiquitous computing, various computing elements are so seamlessly integrated into the environment that they will be invisible to common awareness.

Several factors have fueled this vision. Recent advances in CMOS IC, wireless communication, and MEMS technology have led to dramatic reductions in size, power consumption and circuitry cost. Various functions such as sensing and signal processing can now be integrated into a single wireless node. Coordination and communication among such nodes will not only enable seamless computing but also revolutionize information technology, especially applications related to sensing and controlling physical environments. Small active devices or sensors can coordinate to perform larger sensing tasks (*i.e.,* distributed micro-sensing tasks), which could not have been achieved with individual node capabilities.

Potential applications of such sensor networks span many domains: physiological monitoring; environmental monitoring (air, water, soil, chemistry); condition based maintenance; smart spaces; military surveillance; precision agriculture; transportation; factory instrumentation and inventory tracking. Coordinated efforts in exploring such applications have already begun, the Center for Embedded Networked Sensing [CEN] at UCLA, MIT's Project Oxygen [OXY] and Berkeley's CITRIS effort [CIT], the Aware Home at Georgia Tech [AWA] to name a few.

Fundamental to such seamless coordination in these systems is *location awareness*. Localization is a mechanism to establish spatial relationships in these devices. Because these systems are coupled to the physical world, location measures and gives a context to that physical coupling. Many of these envisioned systems are embedded to monitor or control the behavior of physical systems (as compared with strictly virtual information systems), and therefore nodes often need to determine their action based on their physical location (am I the correct sensor to monitor a particular object?).

Networked applications are often implemented in the form of a layered network protocol stack [Zim80] and localization benefits span several layers of the protocol stack. At the application layer, localization is indispensable for context-aware applications that select services based on location [HHS99], and for sensor networks that achieve power conservation by combining data from multiple sensors. At the network layer, location information on a scale with the transmission range can enable geographic routing algorithms that can propagate information efficiently through a multi-hop network [Fin87, KK00].

Several issues render the localization problem more challenging for large scale, densely distributed sensor networks than in many other domains. Sensor networks must satisfy several physical constraints. In order to be untethered and deeply embedded, individual nodes must have a small form factor and provide their own energy.

The system overall must tolerate ad hoc deployment and unattended operation without infrastructure support.

Given such constraints the network designers' goals shift from optimizing channel throughput or minimizing node deployment, to extending system lifetime and robustness in the face of unpredictable dynamics. Moreover, in these extreme contexts, centralized solutions are not applicable because the communications associated with extracting the dynamic system state to a central location, and in a timely manner, will consume excessive amounts of precious energy resources.

Any deployable localization system for sensor networks must scale to large areas, to large numbers of devices, must accommodate the device constraints of very small devices, and must be robust and fault-tolerant even in the presence of significant environment and system dynamics.

Traditional information systems have not had such a location focus, consequently our support for localization has been relatively weak. Presently, the most well known and widely available technology for localization is the Global Positioning System (GPS) [HLC92]. Since its introduction nearly three decades ago, both the applications and scope of GPS usage have exploded. Nevertheless, GPS has several drawbacks which make it ill suited to sensor networks. Firstly, it is not ubiquitously available — GPS does not work indoors, under water, and in very cluttered urban environments. Secondly, it may not be economically viable for sensor networks. While a typical GPS receiver costs around 100 dollars and consumes power on the order of magnitude of Watts, typical sensor nodes are expected to be disposable in the near future. Thus, it does not always meet the operational (for example, low power and low cost [PK00]), environmental (for example, indoors) or cost constraints.

Although a number of localization systems have been proposed in the past few years [HLC92, PCB00, HHS99, BP00b] (these are reviewed in Chapter 2), none currently satisfies the requirements for ad hoc deployment of large scale sensor networks because no single existing localization system is simultaneously scalable, ad hoc deployable and accommodating of the hardware constraints of very small devices.

1.2 The Problem: Environment Dependent Configuration

Not surprisingly, localization has manifested itself as a classical problem in many disciplines, including the autonomous robot navigation problem [HS98] in mobile robotics [TFB01], virtual reality systems [WBV99], air, land and water vehicle navigation in intelligent transportation systems [VOR, HLC92], user location and tracking in cellular networks [RAD] etc.

A key challenge in engineering localization systems for these applications has been *environmental dependence*. To achieve localization in any given environment, several characteristics of the environment need to be taken into account as they can influence the measurements of the sensors used for localization (for example, the temperature and speed of propagation of light in the medium, the parameters of signal attenuation etc.). Since these properties can vary widely from one environment to another, these environment-dependent parameters need to be configured in all nodes in the localization system.

Traditional localization systems address this problem either (i) through extensive environment-specific calibration and configuration of the centrally controlled, tightly coupled localization system [WBV99, BP00b, RAD] or (ii) through sophisticated, memory and compute-intensive probabilistic position-estimation algorithms [TFB01].

Large-scale, densely distributed sensor networks that are closely coupled to the physical world require node localization, but under far severe node-level resource constraints (limited energy, bandwidth, memory and processing) [BHE00].

Consequently, localization systems that can reconcile these needs by necessity must be based on a loosely coupled, distributed systems architecture (as in [BHE00, PCB00, SHS01a, Gir00, HWB00]) that can adapt to the dynamics of its environment, relying neither on any centralized controller nor on sophisticated processing and sensing capabilities on each on every node.

1.3 A Solution: Self-Configuring Localization Systems

We have highlighted the deployment, configuration and operational issues for sensor network localization. Our approach to solving the aforementioned issues for localization is based on "self-configuration".

Our thesis is that all these issues can be solved simultaneously by a distributed localization system that is also *self-configuring*, *i.e.*, it autonomously measures and adapts to its environmental and system dynamics to achieve both environmental independence and robust, unattended system-level operation.

Our approach is based on localized adaptive algorithms that self-configure to exploit both the local processing available on each sensor node, as well as the redundancy available across densely-deployed sensor nodes.

1.3.1 Self-Localization

To accommodate existing device constraints, we adopt a low cost, hardware-independent localization approach for very small devices that leverages the existing radio (RF) communications capabilities of such devices and does not require any other sensors. To scale to very large sensor networks, we develop a decentralized, self-localization methodology for devices. Instead of relying on a central server to compute their positions, client devices themselves perform a localized location computation based on radio connectivity constraints to a small number of nearby beacons (known nodes which are position-aware). These constraints are obtained from listening to radio broadcast advertisements of nearby beacons.

1.3.2 Self-Configuration

As discussed earlier, one of the key challenges in localization is to ensure a uniform localization granularity even in dynamic, unpredictable environments (with numerous radio propagation vagaries and unpredictable terrain). One solution to this problem is to extensively instrument and model the environment, a priori and deploy beacons based on these measurements. Unfortunately, this approach does not scale well for very large numbers of beacons and for very large environments. Instead, to adapt to noisy environments, we advocate and develop a self-configuring mechanism in which beacons themselves measure and adapt to their environment and availability of neighboring beacons. Our self-configuring mechanisms apply to other ad hoc beacon-based localization systems as well [SHS01a].

1.4 Contributions

A number of research activities have laid the groundwork for both low-cost localization [BP00b, CCK01] and scalable localization [HLC92, PCB00]. However, these research efforts are polarized: they either solve the hardware half of the problem (accommodating device constraints) or the networking half of the problem (making localization scalable).

Consequently, none of the proposed systems meet all our requirements, because in each instance, only half of the problem is solved. Our work bridges this gap. We have developed, analyzed, simulated, and refined a comprehensive set of techniques for making localization systems self-configuring.

We account for each component in the overall system — from the network location protocol and position estimation algorithm to the beacon deployment and adaptation strategy — resulting in the design and implementation of a comprehensive system for localization in ad hoc wireless networks.

Ultimately, we believe that high-quality localization will be commonplace. But before this can happen, we must understand, build and deploy localization systems for sensor networks. This dissertation research is one step toward this goal. Our contributions advance the state of the art in localization and energy-conserving protocols — especially in the context of large scale sensor networks — as follows:

- *Localization Methodology.* We have developed a low cost, scalable and energy-efficient RF-based localization methodology. Our localization methodology is novel both in its highly scalable system architecture for self-localization, and in its simple sensing model of RF-proximity through radio connectivity. Furthermore, we have implemented it on two very different experimental platforms, and evaluated its use under a number of settings. We present experimental results to show that our localization methodology works well outdoors.

- *Density analysis.* We formalize the notion of beacon density and analyze the impact of beacon density on the localization quality — both in terms of localization error and system responsiveness. Our analysis leads us to different problem formulations for beacon placement at different densities.

- *Solutions to Beacon Placement.* We make our localization system self-configuring by addressing the following two beacon placement problems.

  - In noisy environments, the existing beacon field infrastructure might prove insufficient to ensure localization quality. We address this issue with both a novel problem formulation and a novel solution approach. We formulate the problem of adaptive beacon placement — deployment of additional beacons at new points to augment an existing infrastructure of beacons. Our solution is novel in that it is empirically determined — based on actual measurements of the terrain rather than on an idealized model. We have developed and evaluated through simulation and experiment, several algorithms for this purpose. We discuss the design space and present our algorithms, evaluation results and future work. GRID is the *first* algorithm that used robotic mobility to heal and self-configure an unattended network [BHE01a]. HEAP is a localized algorithm that automates the placement of new beacons.

- Alternatively, even if beacons were densely deployed to provide redundant coverage, we may not want to keep all of them operational simultaneously. We develop STROBE, an algorithm that exploits deployment redundancy to improve system lifetime, without degrading the quality of localization.

Our solutions to beacon placement not only apply to other localization systems, but also provide a methodology and case study in tuning network density as a function of the level (fidelity) of network service required, that can be applied to several other problems in densely deployed sensor networks.

1.5 Dissertation Overview

The remainder of this dissertation is organized as follows. In the next chapter, we survey related work in the fields of localization at large, and adaptive protocol design in sensor networks.

Chapter 3 precisely defines the network model that we assume for all of our work.

We also present our assumptions of nodes and argue our design principles of localized algorithms.

Chapter 4 describes our research methodology and wireless testbed.

In Chapter 5, we discuss the design, implementation and evaluation of our RF-based localization system based on the principles described in Chapter 3.

In Chapter 6, we discuss the role of self-configuring beacon systems. Chapters 7, 8, 9 explore three different forms of beacon self-configuration. We also present simulation and experimental results that demonstrate the advantages of self-configuring beacon systems.

Finally, in Chapter 10, we identify a number of remaining challenges with our approach, present plans for future work, identify applications that benefit from a localization system such as ours, and new research problems in sensor networks and provide references to our implementation and simulation framework.

CHAPTER 2

Background and Related Work

What we have learned from others, becomes our own by reflection.

— Ralph Waldo Emerson

The real voyage of discovery comes not in seeking new landscapes but in having new eyes.

— Marcel Proust

Localization is by nature an interdisciplinary problem involving several areas of computer science and relevant to many kinds of engineering systems. Consequently, research has proceeded on both the systems and algorithmic fronts in computer science.

In this chapter, we survey background research and work related to our self-configuring localization system framework. Rather than attempt to cover the entire spectrum of research in location-aware computing or even in localization, we will concentrate on areas that are most relevant to the work in this dissertation[1]. In the next few sections, we survey related work in each of the following areas:

localization systems,

[1] Our discussion is not strictly chronological. Since this work was among the earliest efforts to address node localization in ad hoc sensor networks, several of the localization technologies for sensor networks (including Cricket[PCB00, PMB01], [DPG01], [ACZ01], APS [NB00]) and adaptive sensor network protocols (including SPAN [CJM01]) discussed here were developed subsequent to our work.

robust position estimation algorithms,

techniques for localization error reduction,

system deployment techniques, and

adaptive network protocols

2.1 Localization Systems: An Overview

In this section, we review previous work in localization systems for ubiquitous computing and other applications. We review why state-of-the-art developments in localization systems do not meet the requirements and motivate our approach. We focus on:

measurement techniques for obtaining constraints (ranges, angles, proximity) that correlate unknown positions,

system architecture, and

robust position estimation algorithms

2.1.1 Measurement Techniques

Measurement techniques for localization include ranging from radio Time Difference of Arrival techniques [HLC92, WL98, AET, TIM], ranging using radio and sound [WJH97, PCB00], triangulation from camera-images [OK93] and video [RS00], radio signal strength measurements [BP00b, HWB00] and measurements of radio connectivity [BHE00, DPG01]. We can broadly classify them as fine-grained or coarse-grained localization methods, depending on the *granularity* (or error) of the obtained measurements, and also sub-classify them as ranging, directionality, pattern-matching based

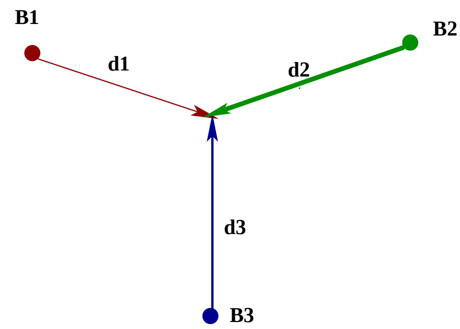


Figure 2.1: Localization from range measurements.

systems, depending on the *type* of measurements.

Fine-grained localization methods typically estimate ranges or angles relative to beacons and compute the location of the unknown node using *trilateration* (position estimation from distance to three points, see Figure 2.1) or *triangulation* (position estimation from angles to three points, see Figure 2.2).

2.1.1.1 Ranging-based Systems

The most popular measurement type is ranging. There are two methods to obtain range measurements — timing and signal strength. We explore both these approaches below.

Timing

The distance between a client node and a beacon may be inferred from the time of flight of the communication signal.

The time of flight may be calculated using the timing advance technique which measures the amount the timing of the measuring unit has to be advanced in order for the received signal to fit into the correct time slot to be in phase with an internally generated signal. This technique is used in the Global Positioning System (GPS) [HLC92] and Pinpoint [WL98], which estimate distance from the radio signal time of flight. GPS measures one-way flight time, whereas Pinpoint's Local Positioning System (LPS) measures round-trip time (thereby eliminating the need for time synchronization).

GPS is a wide-area radio positioning system. In GPS, each satellite transmits a unique code, a copy of which is created in real time in the user set receiver by the internal electronics. The receiver then gradually time shifts its internal clock, until it corresponds to the received code, an event called *lock-on*. Once locked on to a satellite, the receiver can determine the exact timing of the received signal in reference to its own internal clock. If that clock were perfectly synchronized with the satellite's atomic clocks, the distance to each satellite could be determined by subtracting a known transmission time from the calculated receive time. In real GPS receivers, the internal clock is not quite accurate enough. An inaccuracy of a mere microsecond corresponds to a 300 m error.

Pinpoint's 3D-iD system[WL98] is a Local Positioning System that covers an entire three-dimensional indoor space and is capable of determining the 3-D location of items within that space. The LPS subdivides the interior of the building into cell areas that vary in size with the desired level of coverage. The cells are each handled by a cell-controller which is attached by a coaxial cable to up to 16 antennas. It provides an accuracy of 10m for most indoor applications, although some may require accuracy of 2m. The main drawback of this system is that it is centralized and requires significant infra-structural setup.

Active Bat[WJH97], the acoustic range-finder [Gir00], Cricket [PCB00], Cricket Compass [PMB01] and AHLoS [SHS01a] make explicit time-of-arrival measurements based on two distinct modalities of communication, ultrasound and radio, which travel at vastly different speeds ($350m/s$ and $3 \times 10^8 m/s$ respectively), enabling the radio signal to be used for synchronization between the transmitter and the receiver, and the ultrasound signal to be used for ranging. The Active Bat system, however, relies on significant effort for deployment indoors and may not work very well outdoors because they all use a single transmission frequency (40 kHz), and hence there is a high probability of interference from other ultrasound sources.

Signal Strength

An important characteristic of radio propagation is the increased attenuation of the radio signal as the distance between the transmitter and receiver increases. Radio propagation models [Rap96] in various environments have been well researched and have traditionally focused on predicting the average received signal strength (RSSI) at a given distance from the transmitter (large-scale propagation models), as well as the variability of the signal strength in close proximity to a location (small-scale or fading models). In the RADAR system [BP00b, BP00a] (and also in SpotON [HWB00]), Bahl and Padmanabhan suggest estimating distance based on signal strength in indoor environments. They compute distance from measured signal strength by applying a Wall Attenuation Factor (WAF) based signal propagation model. The distance information is then used to locate a user by trilateration. This approach, however, yielded lower accuracy than radio mapping of signal strength corresponding to various locations for their system. Their radio-mapping-based approach is quite effective indoors, unlike ours, but requires extensive infra-structural effort, making it unsuitable for rapid or ad hoc deployment.

2.1.1.2 Signal Pattern Matching

Another fine-grained localization technique is the proprietary Location Pattern Matching technology, used in the U.S Wireless Corporation's RadioCamera system [RAD]. Instead of exploiting signal timing or signal strength, it relies on signal structure char-

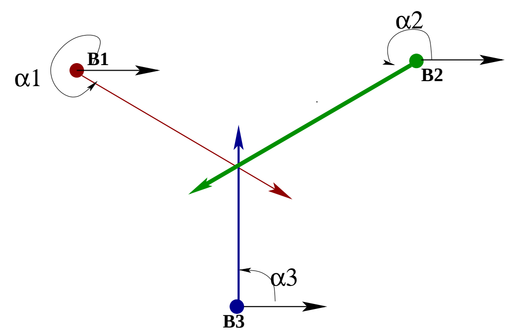


Figure 2.2: Localization from directionality or angle constraints.

acteristics. It turns the multi-path phenomenon to surprisingly good use: by combing the multi-path pattern with other signal characteristics; it creates a signature unique to a given location. The RadioCamera system includes a signal signature database for a location grid of a specific service area. To generate this database, a vehicle drives through the coverage area transmitting signals to a monitoring site. The system analyzes the incoming signals, compiles a unique signature for each square in the location grid, and stores it in the database. Neighboring grid points are spaced about 30m apart. To determine the position of a mobile transmitter, the RadioCamera system matches the transmitter's signal signature to an entry in the database. The system can use data from only a single point to determine location. Moving traffic and changes in foliage or weather do not significantly affect the system's capabilities. The Nibble system[CCK01] similarly estimates location from signal intensity using Bayesian inference algorithms. The major drawback of these techniques, as with RADAR, is the substantial effort needed for the generation of the signal signature database. Consequently, they are not suited for the ad hoc deployment techniques that interest us.

2.1.1.3 Directionality-based Systems

Another way of estimating location is to estimate the angle of each beacon with respect to the client node in some reference frame. The position of the client node can then be computed using triangulation methods.

An important example of directionality-based systems are the VOR/VORTAC stations [VOR], which were used for long distance aviation navigation prior to GPS. The VOR station transmits a unique omnidirectional signal that allows an aircraft aloft to determine its bearing relative to the VOR station. The VOR signal is electrically phased so that the received signal is in various parts of the 360 degree circle. By determining which of the 360 different radials it is receiving, the aircraft can determine the direction of each VOR station relative to its current position.

Small aperture direction finding is another directionality based technique used in cellular networks. It requires a complex antenna array at each cell site location. The antenna arrays can in principle work together to determine the angle (relative to the cell site) from which a cellular signal originated. When several cell sites can determine their respective angles of arrival, the cell phone location can be estimated by triangulation. There are two drawbacks of this approach which make it inapplicable to our application domain. The cost of the complex antenna array implies that it can be placed only at the cell sites. Second, the cell sites are responsible for determining the location of the client node, which will not scale well when we have a large number of such nodes and desire a client-based approach.

Directionality based systems are not very effective in indoor environments because of multi-path effects.

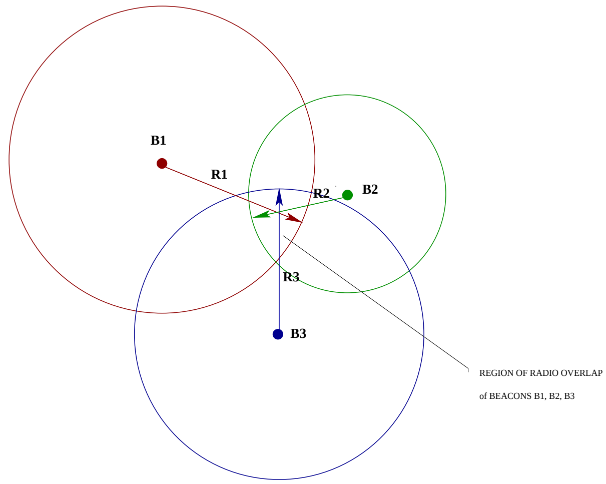


Figure 2.3: Localization from connectivity constraints.

2.1.1.4 Proximity-based Systems

Coarse-grained localization methods estimate unknown node location from proximity to beacons, as in Figure 2.3). One of the earliest such systems was the Active Badge system [WHF92] developed in 1991. Here, each person or object is tagged with an Active Badge. The badge transmits a unique Infra Red (IR) signal every 10 seconds, which is received by sensors placed at fixed positions (beacons) within the building and relayed to the location manager software. The location manager software is able to provide information about the person's location to the requesting services and applications.

Another system based on IR technology was developed by Azuma in 1992 [Azu93] and is the precursor to the current HiBall tracking system at the University of North Carolina at Chapel Hill [WBV99]. This system requires IR transmitters to be located at fixed positions inside the ceiling of the building. An optical sensor sitting on a head-mounted unit senses the IR beacons, and system software determines the position of the person.

Both these IR-based solutions perform quite well in indoor environments, because IR range is fairly small and can be limited to the logical boundaries of a region, such as a room (by walls). On the other hand, the same technique cannot be as effectively applied using radio in indoor environments, because radio propagation in indoor environments suffers from severe multi-path effects that make it impossible to precisely control the radio range. The short range of IR, which facilitates location is also a major drawback of these systems because the building has to be wired with a significant number of sensors. In the few places where such systems have been physically deployed, sensors have been physically wired in every room of the building. Such a system scales poorly, and incurs significant installation, configuration, and maintenance costs. IR tends to perform poorly in the presence of direct sunlight and hence cannot be used outdoors. Another drawback is that it is a tracking system rather than a self-localization system.

More recently, Doherty *et al* [DPG01] has also proposed techniques for localization from radio-connectivity. However, unlike our approach, theirs is centralized and requires that nodes deliver connectivity information to a centralized processor, for solution as a convex optimization problem. Even allowing for a centralized solution, the constraints in the problem are susceptible to errors, since lack of connectivity implies great separating distance, though nearby nodes may just be blocked by an obstacle or intermittent noise. Inaccuracies might cause solutions to oscillate or, worse yet, make the problem infeasible.

2.1.2 System Architecture

Localization systems using similar measurement techniques can differ considerably in their system architecture. For instance, Active Bat, the GALORE Panel [GBE02], and Cricket all use ranging based on acoustic and radio signals, but their system architectures are centralized, hierarchical and decentralized respectively. The design of a localization system is largely influenced by application requirements — such as the requirement for highly accurate or real time position estimation. The system could be either

> *tightly coupled* (uses beacons that are wired to a centralized controller and placed at fixed positions) or
>
> *loosely coupled* (uses beacons that are wireless and coordinate in a completely decentralized manner with no central control).

2.1.2.1 Tightly coupled systems

Several of the traditional and mature localization technologies have a tightly coupled system architecture, motivated by application requirements. These include the Active-Bat system [WJH97] developed for sentient computing applications [ACH01] and the HiBall Tracker [WBV99] designed for virtual reality applications. These applications have high accuracy and real-time tracking requirements.

Problems of time synchronization and coordination amongst beacons are easily resolved because these systems are wired and have a centralized controller. These systems therefore achieve high accuracy. But the drawback is that the centralized position estimation limits the number of devices these systems can simultaneously track (HiBall). Secondly, wiring significantly impedes deployment. A key research challenge in these systems is achieving similar granularity outdoors where deployment cannot be controlled and wiring may be infeasible.

2.1.2.2 Loosely coupled systems

Motivated by deployment concerns, recently proposed localization systems [BHE00], Cricket [PCB00] and AHLoS [SHS01a] are decentralized and completely wireless. They sacrifice the accuracy of tightly coupled systems for ease of deployment, and scalability to large numbers of devices. They rely on a system of beacons, each of which periodically transmits an advertisement containing its position. Clients compute their position based on the advertisements they receive.

Because beacons are wireless and deployed in an ad hoc manner, beacon coverage is not guaranteed. Due to the lack of centralized control, there is no explicit coordination amongst beacons. Thus beacons can contend and self-interfere when emitting a signal (radio, acoustic etc.). These problems need to be addressed for large scale deployment.

2.1.3 Robust Position Estimation Algorithms

Besides ranging techniques and architectural issues in system design, a third component of a localization system is a robust algorithm for position estimation. In this section, we discuss three classes of algorithms for robust position estimation, namely

Monte Carlo Localization

Convex Optimization

Iterative Multilateration

2.1.3.1 Monte Carlo Localization

In the field of mobile robotics, localization has been referred to as "the most fundamental problem to providing a mobile robot with autonomous capabilities" [Cox91]. Environmental obstructions such as walls, moving people and objects, can greatly interfere with the sensing capabilities of a mobile robot. Statistical techniques provide a means to represent uncertainty in sensor measurements. For example, robot localization is just an example of a statistical inference problem on Lie groups [Sri96]. Consequently, the focus here lies on robust position estimation in real-time through probabilistic localization techniques that account for unpredictable sensing error.

An interesting development in probabilistic localization algorithms for mobile robot navigation has been Monte Carlo Localization (MCL) [TFB01]. MCL algorithms represent a robot's belief by a set of weighted hypotheses (samples), which approximate the posterior under a common Bayesian formulation of the localization problem. These algorithms are computationally efficient, versatile, resource-adaptive and robust under a range of circumstances. However, these algorithms have been designed to localize a single mobile robot (with PC-class computational hardware) with respect to its environment. Consequently, they have not addressed issues of scalability or hardware constraints.

2.1.3.2 Convex Optimization

One way to formalize the problem of estimating node positions in a sensor network, is to express relations (angular, range etc.) between different pairs of nodes (known or unknown) as a set of convex constraints. Doherty has proposed convex optimization techniques [DPG01] for solving the position estimation problem in sensor networks in an off-line, centralized manner. The advantage of this approach is that it requires very few references (or beacons) since all system constraints are solved globally. However, this algorithm is not very robust to failures — when there are ambiguities in measurements.

2.1.3.3 Iterative Multilateration

Multilateration is the problem of estimating a node's position from ranges to three or more known nodes (beacons). If not all nodes have ranges to at least three beacons, their positions must be estimated through an iterative process. Savvides *et al* have explored iterative techniques for robust position estimation in sensor networks [SHS01b]. Iterative techniques incur additional energy costs in communication, and are not guaranteed to be completely fault tolerant. Such techniques have also been explored by [ACZ01, NB00].

2.1.4 Summary of Localization Work

In the previous sections, we summarized related work in a number of areas relevant to the localization issues in this thesis, namely: localization systems, algorithms, and sensing techniques.

Most proposed systems have focused on the localization technology, relying either on sophisticated hardware capabilities of devices or on centralized approaches incorporating extensive infrastructure and planning.

Although these research efforts provide a rich spectrum of work to build on, none of these contributions have completely solved the problem of localization in very large, ad hoc deployed sensor networks.

This work proposes a decentralized, low-cost and hardware independent approach can substantially increase the scalability and viability of the localization system, and is described in more detail in Chapter 5. A detailed discussion of various proposed localization systems is given in [BHE00] and [HB01].

2.2 Localization Error Reduction: An Overview

Any deployed localization system is subject to errors from many sources. In this section, we discuss the various sources of localization error, the approaches developed to address them, and their shortcomings.

2.2.1 Sources of Localization Error

Localization error in these systems stems from three main sources. *Beacon unavailability* is caused due to sparse beacon deployment or radio propagation vagaries [Rap96]) that affect the visibility of beacons that should be in range. If the number of beacons is not sufficient, then a position estimate cannot be obtained. *Poor calibration* causes measurement errors whose magnitude varies depending on the ranging technology used, precision of time synchronization and the quality of equipment. *Beacon placement and density* essentially control localization granularity in proximity based localization systems such as [BHE00, DPG01] (see Figure 2.3). The geometric relationship between beacons controls localization granularity in multilateration systems that estimate position from distances to three or more beacons (see Figure 2.1), due to the uncertainty in range measurements [War98].

2.2.2 Error Reduction Approaches

Researchers normally use the following approaches to reduce localization error in their systems.

Geometric and Statistical Tests The first approach is a computational approach and is used in [WJH97, PCB00]. The key idea is to combine simple *geometric consistency checks* and *statistical tests* to identify and eliminate incorrect distance measurements and deal with Gaussian noise in measurements.

Multiple Sensor Modalities Geometric and statistical tests alone cannot eliminate non-Gaussian noise. An example of such noise is the non line-of-sight (NLOS) problem. A second approach to eliminate spurious non line-of-sight readings (non Gaussian error) is to use multiple but orthogonal sensing modalities (for example, acoustic and optical ranging) [GE01a]. However, it cannot be applied to end nodes which must rely solely on radio characteristics to determine location.

Ultra Wide Band (UWB) Ranging A promising new technology on the horizon is ultra-wideband radio ranging [AET, TIM]. Because ultra-wideband signals have much higher signal bandwidth than narrow-band signals, they can penetrate through walls and other obstacles, thereby avoiding non-line-of-sight conditions. The technology however, is still under development and not robust to foliage.

2.2.3 Summary of Error Reduction Work

We have summarized the sources of localization error and two approaches to localization error reduction: namely, geometric and statistical tests and combining measurements from multiple sensor modalities. While these approaches can locally reduce error, to achieve a uniform localization granularity across the terrain, we need to address the complementary problem of infrastructure deployment.

2.3 System Deployment Techniques: An Overview

When localization is accomplished using beacons, the question of where and how many of these beacons should be placed or deployed arises. Several other researchers have also stressed the significance of beacon placement in determining the overall quality of a service such as localization or coverage [PCB00, MKP01].

An ideal placement of beacons should use the fewest possible number of beacons to provide uniform and full coverage of the area of interest. So far, researchers have tried to address beacon (or node) deployment issues using either guidelines (influenced by environment conditions and application requirements) or optimal placement algorithms. In this section, we review these techniques and discuss their drawbacks.

2.3.1 Guidelines

The Cricket Location Support System [PCB00], which is also proximity based, proposes deployment guidelines for beacons in indoor environments based on practical considerations. Whenever a beacon is placed to demarcate a physical or virtual boundary corresponding to a different space, it must be placed at a fixed distance away from the boundary demarcating the two spaces. Such placement ensures that a receiver rarely makes a wrong choice, unless caught within a small distance from the boundary between two beacons advertising different spaces.

When ranging is based on measurements of signal time-of-flight, the transmitter and receiver must have line-of-sight. In non line-of-sight conditions, the signal may take a reflected path, thereby leading to an incorrect range measurement. To maximize the likelihood of line of sight to beacons, the Active Bat system [WJH97] uses ceiling mounted beacons. Both Active Bat and the HiBall Tracker [WBV99] use massive redundancy in beacon deployment to improve position-estimation.

2.3.2 Optimal Placement

Another approach to addressing the deployment problem is to formulate it as an optimal placement problem. Optimal placement problems have been studied in various contexts by researchers including facility location (theory [CGS99]) and pursuit evasion problems in robotics ([GLL99]).

2.3.2.1 Art Gallery and Pursuit Evasion

In robotics, art gallery and pursuit evasion [GLL99] problems have been well studied. In the *art-gallery* analogy, the robot's goal is to move from one position to another to maximize visual coverage of its surroundings, as a human might try to do in a gallery. A complementary set of approaches addresses the *pursuit-evasion* problem in which a robot tries to move so as to evade observation or capture by mobile trackers. However these approaches are based on modeling the environment as a polygon and are best suited for vision-based localization and tracking systems such as [HMS02]. They account for neither the noise nor the wide variety of terrain conditions one would expect to encounter for ad hoc sensor networks.

2.3.2.2 Facility Location

Facility Location [CGS99, SC99] problems are a well known class of theoretical computer science problems and have been the subject of extensive research over the past thirty-five years. In these facility location problems, there is a set of locations, where the cost of building a facility at location i is $f(i)$; furthermore, there is a set of client locations (such as stores) that require to be serviced by a facility, and if a client at location j is assigned to a facility at location i, a cost of $c(i, j)$ is incurred. The objective is to determine a set of locations at which to open facilities, so as to minimize the total facility and assignment costs. Since these problems are NP-hard, it is unlikely that there exist efficient algorithms to find optimal solutions. Instead, the focus has been on designing algorithms that are guaranteed to find solutions within a particular factor of the optimum. Solutions are based on linear relaxations to the natural integer programming formulations that yield extremely good lower bounds.

2.3.3 Offline Analysis

Researchers have recognized that these systems will be deployed at large in an ad hoc fashion, without controlling the placement of each and every node. Instead, they have focused on developing techniques to identify problems in a deployed sensor field. For example, Megeurdicherian *et al* [MKP01] propose solutions to coverage problems in wireless ad hoc sensor networks given global knowledge of node positions using Voronoi diagrams [Aur91] to compute maximal breach paths and find gaps in coverage.

2.3.4 Summary of System Deployment Techniques

The techniques discussed for system deployment are a) not scalable to large sensor networks, b) not suitable for rapid deployment and c) not generalizable to a variety of environments and systems, suffering unknown and unpredictable radio propagation vagaries.

It is virtually impossible to preconfigure to such terrain and propagation uncertainties and compute a satisfying beacon placement to achieve uniform localization granularity across the terrain.

These considerations motivate our work. The focus of this dissertation is the design of beacon placement algorithms based on two complementary distributed and density-adaptive approaches, described in Chapters 7, 8 and 9.

2.4 Adaptive Network Protocols: An Overview

To our knowledge, measurement-based adaptive algorithms for beacon placement have not been previously studied in the networking literature. However, our work has been informed and influenced by a variety of other research efforts in several fields, which we now describe.

2.4.1 Adaptive Protocols in the Internet

The Internet has evolved from a small, well controlled, cooperative experiment into an enormous, chaotic, competitive infrastructure. While this has allowed a system of enormous scale, it has also put a lot of pressure on the underlying infrastructure. *Empirical (or measurement-based) adaptation* has served as a powerful design principle in Internet evolution — for various networking protocols, including the Transmission Control Protocol (TCP) [Jac88], Scalable Reliable Multicast (SRM) [FJL95], and measurement based admission control [JDS95].

TCP adaptively sets its timers or congestion control windows based on round trip time measurements in order to adapt to a wide range of link bandwidths while maintaining high performance [Jac88].

Algorithms in the Scalable Reliable Multicast framework (SRM) [FJL95] dynamically adjust their control parameters based on observed performance within a multicast session. This allows applications using the SRM framework to adapt to a wide range of group sizes, topologies and link bandwidth while maintaining robust and high performance.

The measurement based admission control algorithm described in [JDS95] uses ongoing measurements rather than apriori characterization to determine behavior of existing flows, which enables it to provide predictive service with fairly reliable delay bounds at network utilization significantly higher than those achievable under guaranteed service.

2.4.2 Adaptive Protocols in Sensor Networks

Within the context of unattended ad hoc sensor networks, the design of adaptive algorithms as a self-configuring mechanism is a burgeoning area of research. For instance, the AFECA algorithm proposed in [XHE00], exploits node deployment density and demonstrates adaptive fidelity. It adapts sleep times based on node density, scaling back node duty cycles (and so reducing routing "fidelity") when many interchangeable nodes are present. This allows it to substantially increase the network lifetime. ASCENT [CE02], GAF [XHE01] and SPAN [CJM01] similarly apply the idea of tuning density to trade operational quality against system lifetime for topology maintenance and energy-efficient routing respectively.

2.4.3 Summary of Adaptive Network Protocols

We discussed empirically adaptive protocols in the Internet — to enable congestion control in transport protocols, reliable multicast that scales to arbitrary group sizes and topologies, and admission control; and in sensor networks — for topology maintenance and routing. The novel aspect of the work described in this dissertation is applying the concept of empirical adaptation to *beacon placement*, in order to enable a localization system that can self-configure and cope with a wide variety of noisy environments.

2.5 Summary

In this chapter, we surveyed localization systems, system deployment and localization error reduction techniques. None of the deployment or error reduction techniques apply for really large scale systems. All of the scalable localization systems ignore deployment and maintenance issues.

We describe an radio-beacon based localization methodology in Chapter 5. We also show how these localization systems can be tuned to provide probabilistic guarantees of localization quality.

CHAPTER 3

Network Model

The wireless telegraph is not difficult to understand. The ordinary telegraph is like a very long cat. You pull the tail in New York, and it meows in Los Angeles. The wireless is the same, only without the cat.

— *Albert Einstein*

Sensor networks have vastly different system constraints and performance requirements from traditional packet switched networks such as the Internet and wireless ad hoc networks and require a different network architecture. The work in this dissertation builds heavily on the network model advocated by Estrin *et al* [EGH99] for large, densely deployed sensor networks. In this chapter, we present an overview of this network model to provide context for our self-configuring localization system. Our system leverages upon the following concepts:

1. *Tiered Architectures.* Although Moore's law predicts that hardware for sensor networks will become smaller, cheaper and more powerful, technological advances will never prevent the need to make tradeoffs. Nodes will need to be faster or more energy-efficient, smaller or more capable, cheaper or more durable. Instead of choosing a single hardware platform that makes a particular set of compromises, we believe that an effective design is one which uses a tiered platform consisting of a heterogeneous collection of hardware.

2. *Broadcast Media.* The wireless channel provides a medium for broadcast communication. All nodes within the nominal transmission radius of the transmitter can receive broadcasts.

3. *Multi-hop Communication.* The energy costs of direct long range wireless communication are prohibitive. Long range communication also prohibits spectrum reuse. Therefore, in these systems we use multi-hop communication both to conserve power and improve network capacity. The tradeoff is that the delay is longer.

4. *Localized Algorithms.* A localized algorithm is one in which communication within nodes is restricted to a certain scope. Localized algorithms exhibit good scalability and robustness properties and may be ideally suited for large-scale, multi-hop sensor networks.

5. *Data-centric Communication.* Unlike traditional networks, a sensor node may not need an identity (*e.g.,* an address)[1]. That is, sensor network applications are unlikely to ask the question: What is the temperature at sensor # 27? Rather, applications focus on the *data context* generated by sensors, e.g., what is the temperature in the conference room?

6. *Application-specific In-network Processing.* Traditional networks are designed to accommodate a wide variety of applications. We believe it is reasonable to assume that sensor networks can be tailored to the sensing task at hand. In particular, this means that intermediate nodes can perform application-specific data aggregation and caching, or informed forwarding of requests for data. This is in contrast to routers that facilitate node-to-node packet switching traditional networks.

[1] In some situations, for example, for querying a specific faulty sensor the ability to address an individual sensor is clearly necessary.

In the following sections, we describe each of these architectural components that together form the foundation for our self-configuring localization system.

3.1 Tiered Architectures

Many recent advances in chip integration technology will enable match-box sized sensor nodes equipped with a battery, a power-conserving CPU (several hundred MHz), a memory (several tens of Megabytes) [KKP99].

Although Moore's law [Moo] predicts that hardware for sensor networks will inexorably become smaller, cheaper, and more powerful, technological advances will never prevent the need to make tradeoffs. Even as our notions of metrics such as *fast* and *small* evolve, there will always be compromises: nodes will need to be faster or more energy-efficient, smaller or more capable, cheaper or more durable.

Instead of choosing a single hardware platform that makes a particular set of compromises, we believe that an effective design is one which uses a tiered platform consisting of a heterogeneous collection of hardware. Larger, faster, and more expensive hardware (*sensors*) can be used more effectively by also using smaller, cheaper, and more limited nodes. The smaller devices will trade functionality and flexibility for smaller form factor and power. Alone, they would not be adequate to support our sensor network applications [CEE01]. However, in conjunction with more endowed nodes, they significantly enhance the network's capabilities. There are many possible advantages to augmenting sensor nodes with small form factor tags, such as:

- *Density:* Tags, by definition can be significantly lower cost and therefore can be deployed in larger numbers, more densely, than larger, higher capacity sensor nodes.

- *Longevity:* Tags can be significantly lower power and therefore can be deployed

	1999 (Bluetooth Technology)	2004
Communication	(150nJ/bit) 1.5mW*	(5nJ/bit) 50uW
Computation		~ 190 MOPS (5pJ/OP)

Assume: 10kbit/sec. Radio, 10 m range.

Large cost of communications relative to computation continues

Figure 3.1: Trends in communication costs relative to computation costs (Source: Gregory J. Pottie).

for longer periods of time, or at higher duty cycles, than larger, higher capacity sensor nodes, particularly, if we are able to exploit higher density.

- *Form factor:* Tags are smaller and therefore can be more easily and unobtrusively attached to a wider variety of targets (e.g., for tracking, condition based maintenance, and other logging applications).

3.2 Broadcast Media

The wireless channel provides a shared broadcast communication medium. Wireless channel access can be arbitrated using a multiple access protocol — either contention-based (Carrier Sense Multiple Access) or contention-free (Time Division Multiple Access (TDMA))[Tan96]. All nodes within the nominal transmission radius of the transmitter can receive broadcasts, except in a TDMA-network.

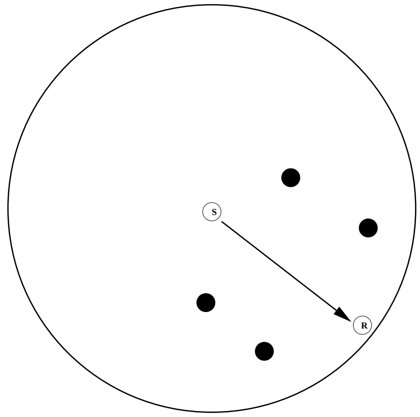


Figure 3.2: An illustration of direct long range communication.

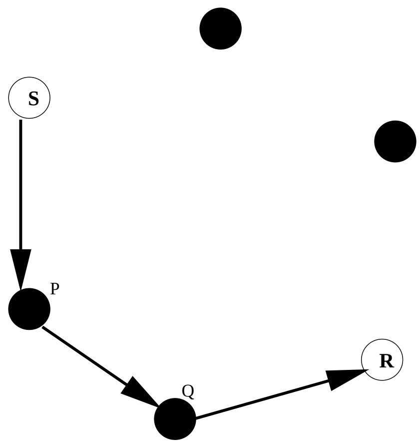


Figure 3.3: An illustration of multi-hop communication.

3.3 Multi-hop Communication

Communications is the dominant consumer of energy in sensor networks [PK00]. This trend is expected to continue in the near future (see Figure 3.1).

The simplest way for a sender S and a receiver R to communicate is through direct-link communication, regardless of the distance between them. This is illustrated in Figure 3.2. The energy cost of direct long range communication is prohibitive (because received signal power attenuates exponentially with distance). Long range communication also prohibits spectrum reuse [PK00].

Therefore, in these systems we use multi-hop communication. In multi-hop communication, a sender S and a receiver R may communicate through intermediate nodes R and Q. This is illustrated in Figure 3.3. Multi-hop communication conserves power [PK00] and also improves the capacity [Sha48] of the wireless network [GK00]. The tradeoff with multi-hop communication is increased latency (due to transmission and processing delays experienced at intermediate nodes) and routing complexity.

3.4 Localized Algorithms

Sensor network coordination applications are better realized using *localized* algorithms. This term means a distributed computation in which sensor nodes only communicate with sensors within some neighborhood, yet the overall computation achieves a desired global objective.

The design rationale for localized algorithms may be explained as follows. Since the sensors themselves are physically distributed it is natural to design sensor networks using distributed algorithms. Furthermore, localized algorithms have two attractive properties. First, because each node communicates only with other nodes in some neighborhood, the communication overhead *scales* well with increase in network size. Second, for a similar reason, these algorithms are *robust* to network partitions and node failures.

3.5 Data-centric Communication Paradigm

Ad hoc networks refer to self-organizing networks of mobile wireless nodes that do not depend on a fixed infrastructure. Several routing protocols have been proposed for ad hoc networks [RT99]. Unlike traditional networks, a sensor node may not need an identity (*e.g.,* an address) [2]. That is, sensor network applications are unlikely to ask the question: What is the temperature at sensor # 27? Rather, applications focus on the *data context* generated by sensors, e.g., what is the temperature in the conference room? Addressing may not even be required for network operation, nodes could just select random transaction identifiers on a per transaction basis, allowing for significant savings in addressing overhead [EE01].

Data is named by attributes and applications request data matching certain attribute values. So, the communication primitive in this system is a request: Where are nodes whose temperatures recently exceeded 30 degrees? This approach decouples data from the sensor that produced it. This allows for more robust application design: even if sensor # 27 dies, the data it generates can be cached in other (possibly neighboring) sensors for later retrieval.

Directed diffusion [IGE00] is an example of a scalable and robust data-centric communication paradigm. By eliminating the *indirection, e.g.* the mapping from a name to a node address to a route, a sensor network can eliminate the maintenance overhead associated with constructing and maintaining these mappings and directory services [HSI01].

3.6 Application-specific In-network Processing

Traditional networks such as the Internet are general-purpose networks designed to accommodate a wide variety of applications. We believe it is reasonable to assume that sensor networks can be tailored to the sensing task at hand. In particular, this means that intermediate nodes can perform application-specific data aggregation and

[2] In some situations, for example, for querying a specific faulty sensor the ability to address an individual sensor is clearly necessary.

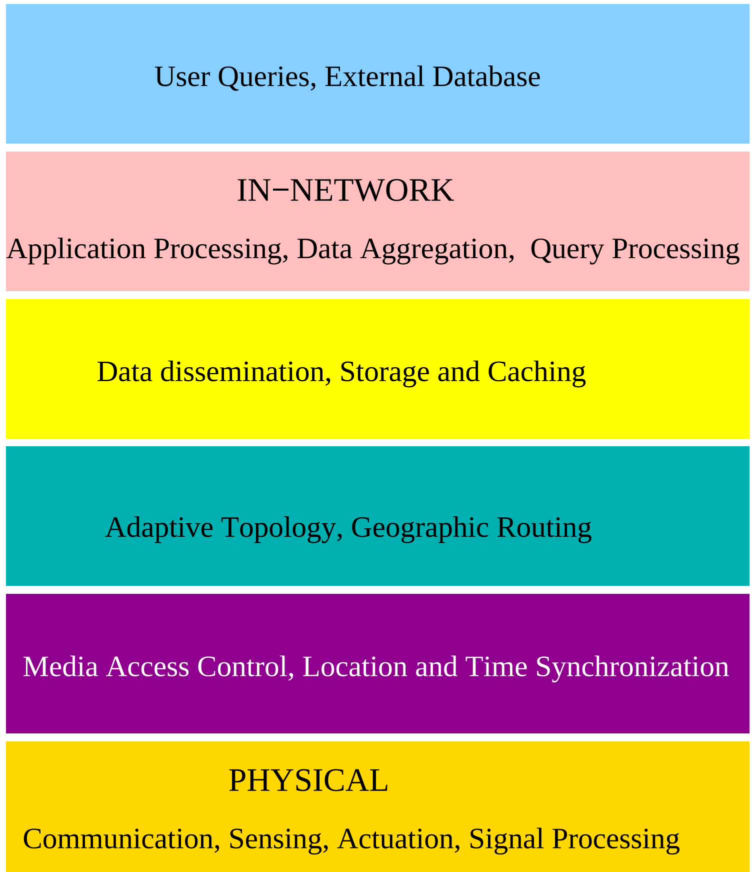


Figure 3.4: Layered communications architecture for sensor networks.

caching, or informed forwarding of requests for data. In densely deployed networks, this could lead to substantial bandwidth and energy savings [IEG02] by exploiting correlations in data. This is in contrast to routers that facilitate node-to-node packet switching in traditional networks without examining the content of data.

3.7 Summary

To close this chapter, we illustrate the communications architecture in Figure 3.4. In summary, tiered architectures allow us to trade form factor against functionality. We leverage this idea in using a beacon-based architecture for localization (Chapter 5). Broadcast communication provides a means for a transmitting node to simultaneously reach several receiver nodes. We exploit broadcast capabilities to efficiently implement our RF-proximity localization system without degrading system responsiveness by having beacons broadcast position advertisements. Multi-hop communication conserves energy and enables spectrum reuse. Localized algorithms enable scalability and robustness besides conserving energy in a multi-hop network. Hence, our algorithms for node localization (Chapter 5) and self-configuration (Chapter 9) are localized algorithms. Data-centric communications simplifies network maintenance and configuration by eliminating the indirection of naming nodes and binding them to node addresses. It also motivates node localization (Chapter 5), since location is a natural way to name data in a physically coupled sensor network. In-network processing conserves bandwidth and energy. We leverage in-network processing to selectively propagate only the data items with maximum utility in the HEAP approach for beacon self-configuration described in Chapter 8. In short, these design principles are requisite to an effective and scalable sensor network, and altogether comprise a solid foundation for our self-configuring localization system.

CHAPTER 4

Experimental Methodology and Wireless Testbed

What we observe is not nature itself, but nature exposed to our method of questioning.

— Werner Heisenberg

It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories instead of theories to suit facts.

— Sherlock Holmes, in The Whole Art of Detection, by Sir Arthur Conan Doyle

Our goal is to enable localization that is scalable, ad hoc deployable and energy efficient and requires minimum configuration. To achieve this goal, we use a combination of analysis, simulation, design, implementation and performance measurement.

In this chapter, we describe the methodology and tools used to perform the research reported in this dissertation. We start by describing our research methodology and the simulation environment we used, followed by the details of the LECS (Laboratory for Embedded Collaborative Systems) and SCADDS testbeds and the different networks (nodes) that form the testbed. We conclude this chapter with a description of our measurement techniques and performance metrics.

4.1 Research Methodology

Our general research methodology involves three phases. In the first phase, we *analyze* the problems that could affect RF-based localization performance by measuring the properties of radio propagation in deployed wireless networks. This analysis is done using different transmission workloads, whose control parameters include the inter-frame spacing, density of nodes, transmit power level, transmitter-receiver distance, area of deployment etc. Using the tools described in Section 4.4, we collect packet-level traces and analyze them in detail. This analysis yields a better understanding of the reasons for the degraded performance, and also gives us data on realistic values for the different parameters in the network elements, such as bandwidths, delays, error and radio connectivity patterns, etc.

The gathered data and intuition on the nature of the problems serve as the starting point for the second phase, *simulation*. Simulation is a powerful approach that enables rapid prototyping of various ideas and solutions, and allows us to explore the design space of parameters and algorithms more thoroughly than in a direct implementation. It also helps us "time travel" and explore possible technology trends that help us predict what performance might be achievable in the future using different solutions. The principal advantage of simulation is in having a controlled environment in which we can analyze problems and design solutions, explore a wide variety of alternatives, and compare them equitably.

After obtaining a good understanding of the problem via measurement and simulation, we move to the third phase, *implementation.* Here, we implement the promising solutions from the simulation phase in our experimental testbed.

After completing an initial implementation, we proceed to the *performance evaluation* phase, which leads back to our analysis phase. Here, we measure the performance of our new algorithms and protocols under different environmental and network conditions, and gather packet-level traces to understand the reasons for observed performance. When we cannot expect to completely explore how the implementation performs under a wide variety of emulated radio propagation conditions. Finally, we compare the results of these measurements to the results obtained from simulation. This helps us tune the implementation to perform well, and also helps us identify shortcomings in the simulations that we correct to accurately reflect reality.

The rest of this chapter discusses our simulation environment, the experimental networks in our wireless testbed, and the measurement techniques and the performance metrics we used.

4.2 Simulation Environment

Simulation is an invaluable tool in networking research and our work is no exception to this general rule. The advantages of simulation are rapid prototyping, being able to vary a wide range of parameters in a controlled fashion repeat-ably, which is not always possible in real implementations.

Selecting the correct level of detail (or level of abstraction) for a simulation is a difficult problem. Choices about detail are particularly difficult for wireless network simulations [HBE01]. In sensor network simulation, these choices include the topology model, radio propagation model, and the power consumption model.

4.2.1 Topology Models

Because packets are broadcast over an unguided wireless medium, the network topology is a function of the radio propagation model and the distribution of nodes.

Nodes were either distributed uniformly or uniformly at random in a square terrain.

There is an edge between two nodes if there is radio connectivity between them.

4.2.2 Radio Propagation Models

A radio propagation model predicts the received signal power as a function of the transmitter and receiver positions. To receive packets successfully, the received signal power must exceed a certain threshold.

In order to evaluate our algorithms under different propagation conditions, we carried out our simulations for

ideal radio propagation conditions

noisy radio propagation and

a terrain based shadowing model based on a bitmap of the environment.

4.2.2.1 Ideal Radio Propagation

The Friis free space model [Rap96] is an ideal radio propagation model, which predicts the received signal power as a deterministic function of the distance d from the transmitter as

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{4\pi^2 d^2 L} \tag{4.1}$$

where P_t is the transmitted power G_t and G_r are the antenna gains of the transmitter and the receiver respectively. $L(L \geq 1)$ is the system loss and λ is the system wavelength. It is normal to select $G_t = G_r = 1$ and $L = 1$ in simulations.

4.2.2.2 Noisy Radio Propagation

To study the impact of noise on our beacon placement algorithms, we model random propagation noise as follows. For each beacon field, connectivity to any beacon B at any given point P is determined based on a noise model.

In our noise model, connectivity to a beacon B exists at a point P, if $distance(P, B) \leq R(1 + u.nf(B))$. $nf(B)$ is the noise factor of the beacon B, and is chosen uniformly between 0 and $Noise$, the maximum noise factor for the field. u is chosen uniformly at random between -1 and 1.

The intent was to create non-uniform propagation noise for the beacons, and to create random regions with higher propagation noise than the rest of the location field. We did this because the impact of noise is less evident when each beacon has an identical propagation field.

4.2.2.3 Terrain Based Shadowing Model

We have ported the terrain based shadowing model from the Arena/ns simulator [YVS01] to our simulations and describe it below.

The basic log-normal shadowing model [Rap96] consists of two parts. The first is a path loss model which predicts the mean received power at distance d, denoted by $\overline{P_r(d)}$ relative to a close-in reference distance d_0. The second is a log-normal random variable[1] that reflects the variation of the received power at certain distance. The overall shadowing model is represented by

$$\left[\frac{P_r(d_0)}{\overline{P_r(d)}}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \tag{4.2}$$

where β is called the path loss exponent. Typically $\beta = 2$ for free space propagation, and lies between 3-5 for outdoor obstructed environments [Rap96]. $P_r(d_0)$ can

[1] It is of Gaussian distribution if measured in dB.

be computed from the Friis free space model [Rap96]. X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB}. σ_{dB} is called the shadowing deviation. Its typical value in outdoor environments is 5-12. The shadowing model extends the ideal circle model to a richer statistical model: beacons can only probabilistically communicate when near the edge of communication range or when there is clutter in between.

The terrain-influenced shadowing model extends the shadowing model in cluttered environments to a terrain based model. By inspecting a bitmap of the terrain, it distinguishes the case when there is clear line-of-sight between two points (good propagation condition) and when there is not (bad propagation condition). Radio propagation conditions can be varied by choosing different path loss exponents (β_1 for good propagation and β_2 for bad propagation respectively), to simulate the significant difference in signal strength and multi-path effects between direct and indirect transmission. The communication range is decided by the transmission power, propagation condition and the receiving threshold.

4.2.3 Energy Consumption Models

Since our goal is to build long-lived localization systems, it is important to model the energy usage of individual nodes. An energy consumption model characterizes the power consumption for each of the various tasks a node may perform.

Typical processing costs are much lower than communication costs [PK00]. Additionally, a task such as transmission of small beacon advertisements is not a computation-intensive activity. Thus, communications is the dominant consumer of energy in these systems. Therefore, our energy model only characterizes the energy usage of the radio transceiver only and does not explicitly model the energy usage of the processor.

Radio Energy Model: A typical radio can operate in any one of four states - Trans-

Table 4.1: Parameters of the energy consumption model.

PARAMETER	VALUE
P_X	660 mW
P_R	395 mW
P_I	35 mW
P_S	0 mW

mit, Receive, Idle and Sleep. We represent power dissipation in each of these states with the symbols P_X, P_R, P_I and P_S respectively. The relative power dissipation in each of these states depends on the radios being used. We choose an energy consumption model to mimic realistic sensor radios [Kai00]. These parameters are also used in [IGE00] and are summarized in Table 9.2.

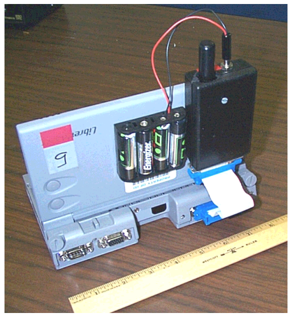
4.3 Wireless Testbeds

While simulations are a valuable design tool, radio propagation in practice deviates a lot from mathematical models and exhibits high spatio-temporal variation. Therefore, experimental data is invaluable to evaluating these systems.

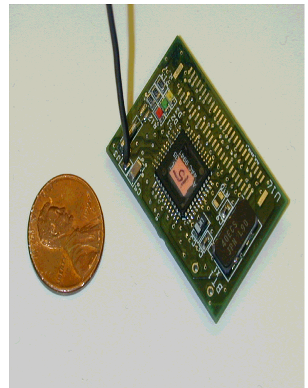
We have implemented a prototype of our localization methodology on two different experimental testbeds.

1. Radiometrix RPC radios connected to laptops via. a serial interface

2. UC Berkeley Rene motes [HSW00], completely integrated with RFM [RFM] radios completely, shown in Figure 4.1.



(1) Librettos with Radiometrix RPC radios



(2) Motes

Figure 4.1: Experimental testbed for the localization methodology.

4.3.1 Radiometrix RPC-418

Our first experimental testbed consists of Radiometrix RPC 418 (radio packet controller) modules connected to a Toshiba Libretto running RedHat Linux 6.0. In our experiments, one of these modules is used as a receiver and four are used as beacons. A 3 inch antenna is used for the experimental purposes. The software for the Radiometrix RPC-418 modules consists of two components.

- *Beacon*: The beacon periodically transmits a packet (every 2 seconds in our experiment) containing its unique ID and position.

- *Receiver*: The receiver obtains its current measured position based on an input from the user. For each measured position, it samples for a time period t determined by the sample size S, and logs the set of beacons it hears from and its current localization estimate.

4.3.2 Berkeley Motes

We have conducted further experiments on very small, embedded devices called motes, developed at the University of California, Berkeley [HSW00], and available from Crossbow Technologies [CRO]. These devices have a RISC-like 8-bit CPU that runs at 4MHz. Motes are equipped with 512 bytes of SRAM, 256 Kbits of EEPROM, and a 916 MHz ISM radio (RF Monolithics TR1000 [RFM]) that can transmit at the rate of 10Kb/s. The transmit power level of the radio can be controlled using a digital potentiometer on the mote.

Motes can be programmed in three configurations: Snooper, Beacon or Logger. A Snooper mote acts as a network interface for a PC via the RS-232 interface and can listen to all transmitted data packets and forward this to the PC. A Beacon mote can be instructed to periodically transmit packets with its ID at a given transmit power level. A Logger mote records all messages sent out by beacons into an EEPROM, and can transfer this information on demand to a Snooper mote connected to a PC.

4.4 Measurement Techniques and Performance Metrics

In this section, we discuss the tools and techniques used to measure and evaluate localization quality and our improvements in these localization systems.

4.4.1 Measurement Tools

While the basic localization software consists of just two components (*beacon* and *client*), the experimental testbed to measure it is actually a lot more elaborate.

We made the following extensions to basic localization software components.

- *Beacon:* It not only transmits periodic advertisements with its position, but it

Table 4.2: Beacon control commands supported by the transceiver.

PARAMETER	DEFINITION
STANDBY	In this mode, a beacon sets its clock rate to the slowest and does not transmit its coordinates. When it receives an ON command, it will enter the NORMAL mode again.
TRANSMISSION POWER CONTROL	It can adjust its own transmission power. (valid in NORMAL mode only)
TRANSMIT RATE CONTROL	It can adjust the rate at which it transmits its coordinates (valid in NORMAL mode only)
RANDOMIZED RESUME	Since a command packet floating on the air can control more than one beacon, each beacon in the STANDBY mode will enter NORMAL mode x milli-seconds after it hears the command, where $x \in (0, y)$ milli-seconds and y is adjustable.

also listens to and responds to commands from a *Beacon Remote Controller*.

- *Client:* It not only estimates its position from the beacon advertisements it receives (the algorithm for doing this position estimation is described in Chapter 5), but also reports its estimated position to the *Beacon Interpretor*.

We developed the following control and visualization tools for experimental control.

- *Transceiver:* The *Transceiver* communicates with the Beacons as well as the Clients. It is needed to send control packets to beacons and to interpret the estimated coordinate sent by the Client. To avoid interference, problems with beacons' life-time, etc. the beacons can be remotely controlled. The commands supported are enumerated in Table 4.4.1.

- *Beacon Remote Controller:* A Java Graphical User Interface (GUI) that lets the users select the desired TRANSMISSION POWER and TRANSMIT RATE settings that beacons operate at. This program will encode the information and send it as a command packet to the beacons. ON and OFF commands can be selected to toggle the beacons between STANDBY and NORMAL modes.

- *Beacon Interpretor:* A Java Graphical User Interface (GUI) that listens for coordinates the Client report and puts them in a table. A user can manually enter the *actual* coordinates of the Client is located and these coordinate will be compared with the *estimated* coordinate to compute the localization error.

- *Visualization:* The visualization program displays all transmitting motes in our laboratory. This is useful to verify whether beacons are transmitting as expected, or if other motes are transmitting and interfering with the experiment.

By allowing us to automatically control and configure the beacons, these tools allow us to experiment rapidly under different beacon density and interference settings without having to reprogram and redeploy all the beacons.

4.4.2 Performance Metrics

The metrics we use to evaluate performance of the localization system include the *mean* and *median localization error* in the terrain. Ideally, these should be close to zero.

4.5 Summary

This chapter described our research methodology, which includes analysis, simulation, design, implementation and performance evaluation. We described the simulation environment and the details of our radio propagation and radio based energy models. We also described our wireless testbed consisting of Radiometrix RPC-418 radios and Berkeley motes. Finally, we defined metrics used to evaluate localization performance in the rest of this thesis.

CHAPTER 5

Localization from Radio Proximity

The amount of misguided ingenuity which has been expended on these two problems of submarine and aerial navigation during the nineteenth century will offer one of the most curious and interesting studies to the future historian of technologic progress.

— George Sutherland, American lawyer, 20th Century Inventions (1900)

Having established our network model and system model based on tiered architectures, we now turn to a detailed discussion of the core contribution of this thesis: our self-configuring localization system architecture. In the next four chapters, we describe the issues in making the localization system self-configuring and the techniques for achieving that self-configuration.

In this chapter, we develop the distributed algorithm, sensing model, and the network protocol that comprise our localization system. Each unknown node can compute its position individually, enabling the system to scale to large numbers of nodes. Moreover, several unknown nodes can compute their positions simultaneously, allowing the system to be extremely responsive and operate in real time. In the next few sections, we describe our localization system in detail, focusing on:

- a detailed overview of our localization system
- radio connectivity inference
- position estimation algorithm based on radio connectivity relations.

Table 5.1: Notation used to describe the RF-based localization methodology.

PARAMETER	DEFINITION
d	Separation distance between adjacent beacons
R	Transmission range of the beacon
T	Time interval between two successive position advertisements transmitted by a beacon
t	client sampling or data collection time
$N_{sent}(i,t)$	Number of advertisements that have been sent by B_i in time t
$N_{recv}(i,t)$	Number of advertisements sent by B_i that have been received in time t
CM_i	Connectivity metric for B_i
S	Sample size for connectivity metric for beacon B_i
CM_{thresh}	Threshold for CM
(X_{est}, Y_{est})	Estimated Location of the receiver
(X_a, Y_a)	Actual Location of the receiver

We then describe its implementation and present a large number of experimental and simulation results under a number of typical network and environment configurations.

5.1 Overview

In this section, we formalize the problem of *node localization*, state our design goals for an ideal solution and describe our network location protocol.

5.1.1 Problem Definition

Formally, the problem of node localization can be stated as follows.

Given:

S, a collection of sensor nodes at positions P_i

M, a set of measurements that establish relations between P_i.

Estimate: $P_i \quad \forall\, i$

5.1.2 Design Goals

An ideal localization solution must meet the following design goals:

- *RF-based:* We focus on small nodes which have some kind of short-range radio frequency (RF) transceiver. Our primary goal is to leverage this radio for localization, thereby eliminating the cost, power and size requirements of a GPS receiver.

- *Client based:* In order to scale well to large distributed networks, the responsibility for node localization must lie with the client node that needs to be localized and not with the beacons.

- *Ad hoc:* In order to ease deployment, we desire a solution that does not require extensive pre-planning or wired infrastructure.

- *Responsiveness:* We need to be able to localize individual nodes within a fairly low response time.

- *Low Energy:* Small, un-tethered nodes have modest processing capabilities, and limited energy resources. If a device uses all of its energy localizing itself, it will have none left to perform its task. Therefore, we desire to minimize computation and message costs to reduce power consumption.

- *Adaptive Fidelity:* Finally, we want the accuracy of our localization algorithms to be adaptive to the granularity of available beacons.

5.1.3 Network Location Protocol Description

Multiple nodes in the network with overlapping regions of radio coverage serve as beacons, labelled B_1 to B_n. Beacons are situated at known positions, (X_1, Y_1) to (X_n, Y_n).

Beacons periodically broadcast advertisement packets (period = T) containing their respective positions. To avoid collisions, beacons randomize their packet transmissions, rather than explicitly coordinate with each other. Furthermore, in any time interval T, each of the beacons would have transmitted exactly one advertisement.

5.2 Sensing

We measure radio connectivity to establish proximity relations between known nodes (beacons) and unknown nodes (clients). In this section, we describe the statistical methodology for inferring radio proximity and nominal radio transmission range. The notation we use to describe our proximity inference and position estimation methods are given in Table 5.1.

5.2.1 Proximity Inference

To drive the radio proximity-based position estimation algorithm, a client node must determine which beacons it has connectivity to, and whether the connectivity is weak or strong.

For a fixed time period t, each client node listens to and collects all the position advertisements that it receives from various beacons. We characterize the information at a client $c \in C$, where C is the set of all clients, for each beacon B_i by a *connectivity* *metric ($CM_{c,i}$)*, defined as:

$$CM_{c,i} \;=\; \frac{N_{recv}(i,t)}{N_{sent}(i,t)} \times 100 \qquad (5.1)$$

Since each client has its own connectivity metrics, we implicitly drop the c subscript.

In order to improve the reliability of our connectivity metric in the presence of various radio propagation vagaries, we would like to base our metric on a sample of at least S packets, where S is the sample size, a tunable parameter of our method (***i.e.***, $N_{sent}(i,t) = S$). Since we know T to be the time period between two successive beacon signal transmissions, we can set t, the receiver's sampling time as:

$$t = (S + 1 - \epsilon)T \qquad (0 < \epsilon \ll 1) \qquad (5.2)$$

Note that a beacon message can be heard by all the clients in its range.

5.2.2 Nominal Radio Transmission Range

The nominal transmission range of a radio cannot be theoretically determined, but must be established empirically. We define the nominal radio transmission range of a beacon *statistically* as the median range with 90% connectivity.

5.3 Position Estimation

In this section, we describe our algorithm for position estimation and discuss its complexity.

5.3.1 Algorithm Description

From the position advertisements that it receives, the client node infers proximity to a collection of reference points for which the respective connectivity metrics exceed a certain threshold, CM_{thresh} (say 90%). We denote the collection of beacons by $B_{i_1}, B_{i_2}, \ldots, B_{i_k}$. The client localizes itself to the region which coincides to the intersection of the connectivity regions of this set of beacons, which is defined by the *centroid* of these beacons.

$$(X_{est}, Y_{est}) \;=\; (\frac{X_{i_1} + \cdots + X_{i_k}}{k}, \frac{Y_{i_1} + \cdots + Y_{i_k}}{k}) \qquad (5.3)$$

We characterize the accuracy of the estimate by the *localization error* LE defined as,

$$LE \;=\; \sqrt{(X_{est} - X_a)^2 + (Y_{est} - Y_a)^2} \qquad (5.4)$$

By increasing the *range overlap* of the beacons that populate the grid i.e., increasing the ratio $\frac{R}{d}$, the granularity of the localization regions becomes finer, and hence the accuracy of the location estimate improves. This is illustrated in Figure 5.1.

5.3.2 Algorithm Complexity

Both the communication and computation complexity for a device to infer its position once are $O(k)$, where k is the number of beacons in radio range.

Because both the computation and communication complexity grow linearly with

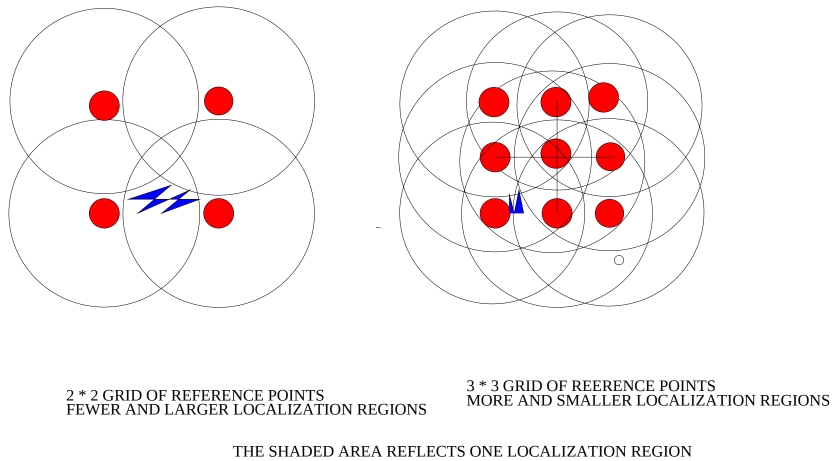


Figure 5.1: Granularity of localization regions vs. range overlap.

the density of the beacon infrastructure, rather than the size of the system, our system is generally scalable.

5.4 Preliminary Measurements

Since our localization method depends on the spherical radio propagation assumption, we checked the validity of our assumption in both outdoor and indoor environments.

In outdoor environments, we evaluated the effectiveness of our idealized radio model by comparing its accuracy to experimental measurements. We evaluated propagation between two Radiometrix radio packet controllers (model RPC-418) operating at 418 MHz. A node periodically sent 27 byte position advertisements; we define a 90% packet reception rate as *connected* and empirically measured an 8.94m spherical range for our simple model.

To evaluate how well our simple model compares to a real-world scenario, we placed a radio in the corner of an empty parking lot (i.e., at the origin $(0, 0)$) and then measured connectivity at 1m intervals over a 10m square quadrant. Figure 5.3

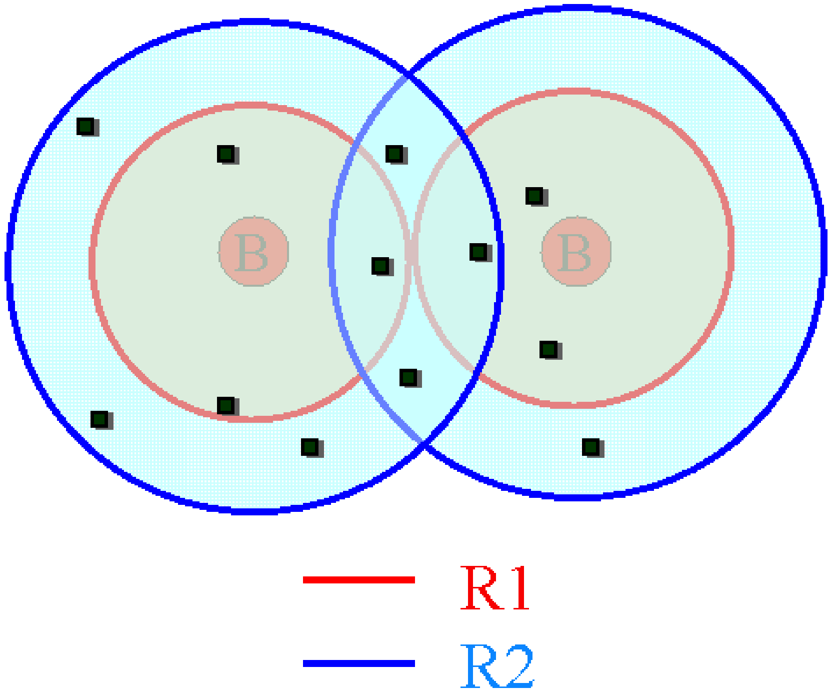


Figure 5.2: An illustration of localization with multiple transmit power levels.

compares these measurements with connectivity as predicted by the model. Among the 78 points measured, the simple spherical model matches correctly at 68 points (an 87% correlation) and mismatches at 10, all at the edge of the range. Error was never more than $2m$. No dead spots were observed.

As expected, our simple, idealized radio model approximation is not appropriate for indoor environments where reflection and occlusion are common. Our indoor measurements of propagation range varied widely from 4.6m to 22.3m, depending on walls and exact node locations and orientations. Furthermore, these measurements were not time invariant. We found that connectivity could vary from 0 to even 100% for the same transmitter receiver positions, at different times of the day.

Hence the idealized radio model may be considered valid for outdoor unconstrained environments only.

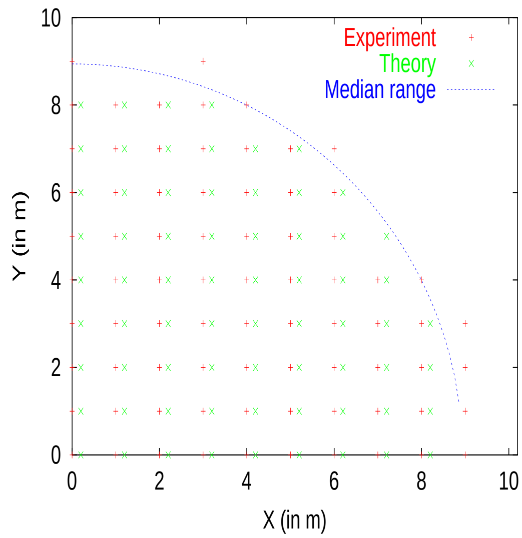


Figure 5.3: 90% connectivity ranges for the beacon (0,0)

5.5 Position Estimation with Multiple Power Levels

To further improve the granularity of beacons, we can leverage the capability of nodes to transmit at software controllable radio power levels. Using a small number of different power levels we can create a number of distinct ranges for each beacon(see Figure 5.4).

As usual, beacons situated at known positions, (X_i, Y_i), transmit periodically with a time period T. However, they cycle through a discrete number of transmission power levels.

Clients listen for a period $t \gg T$ to evaluate connectivity. If the percentage of messages received from a beacon B_i with range r_i in a time interval t exceeds a threshold CM_{thresh}, that beacon is considered connected at r_i. The smallest of all connectivity ranges is considered for each beacon.

When the beacon placement is uniformly distributed, the weighted centroid of the positions of all connected beacons is a feasible solution in the region of connectivity

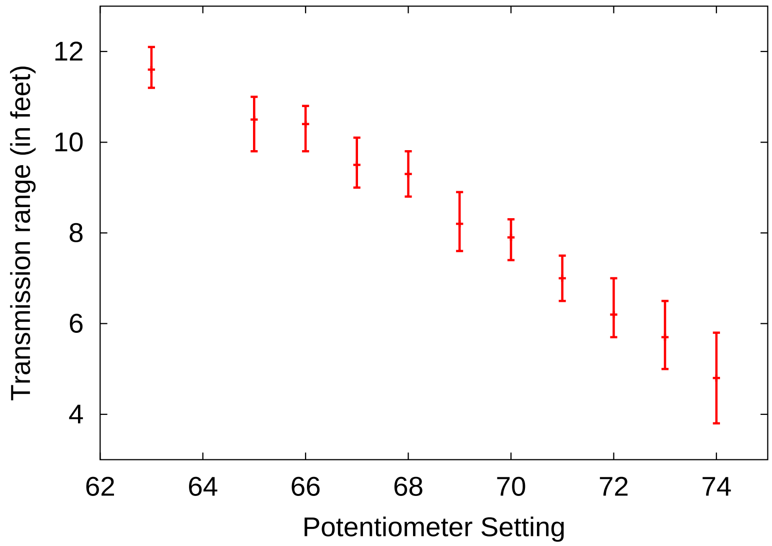


Figure 5.4: Nominal transmission range vs. transmit power setting.

overlap [BEG95]. A client estimates its position (X_{est}, Y_{est}) to be the centroid of the positions of all connected beacons.

$$w_i = \left(\frac{\left(\frac{1}{{r_i}^2}\right)}{\Sigma_{i=1}^{k}\left(\frac{1}{{r_i}^2}\right)}\right) \tag{5.5}$$

$$(X_{est}, Y_{est}) = \left(\Sigma_{i=1}^{k} w_i \cdot X_i, \Sigma_{i=1}^{k} w_i \cdot Y_i\right) \tag{5.6}$$

A more general convex optimization technique to determine a feasible solution can be found in [DPG01] (for non-uniform placement.)

Given the actual position of the client (X_a, Y_a), we can compute the accuracy of the localization estimate or the *localization error* $LE_{\mathcal{B}}(X_a, Y_a)$, which is the distance between the client's estimated and actual positions.

$$LE_{\mathcal{B}}(X_a, Y_a) = [(X_{est} - X_a)^2 + (Y_{est} - Y_a)^2]^{\frac{1}{2}} \tag{5.7}$$

5.6 Experimental Results

5.6.1 Experimental Testbed

Our experimental testbed [TES] consisted of 5 Radiometrix RPC-418 (radio packet controller) modules connected to a Toshiba Libretto running RedHat Linux 6.0. One of these modules is used as a receiver and the rest are used as beacons. A 3 inch antenna is used for the experimental purposes.

The software for the Radiometrix RPC-418 modules consists of two components.

- *Beacon*: The beacon periodically transmits a packet (every 2 seconds in our experiment) containing its unique ID and position.

- *Client*: The receiver obtains its current measured position based on an input from the user. For each measured position, it samples for a time period t determined by the sample size S, and logs the set of beacons it hears from and its current localization estimate.

For our experiment, we placed the 4 beacons at the four corners of a $10m \times 10m$ square, in an outdoor parking lot. This square was further subdivided into 100 smaller $1m \times 1m$ grids and we collected data at each of the 121 small grid intersection points.

5.6.2 Outdoor Results

In this section, we discuss the results obtained from our outdoor experiments. Our experimental parameters were T = 2 seconds, S=20, t=41.9 seconds.

Figure 5.5 shows the areas of connectivity of the 4 beacons in the grid. We see several distinct regions in the grid, based on the areas of overlap. Each distinct region constitutes an equivalence class, defined by the centroid of the beacons in the region.

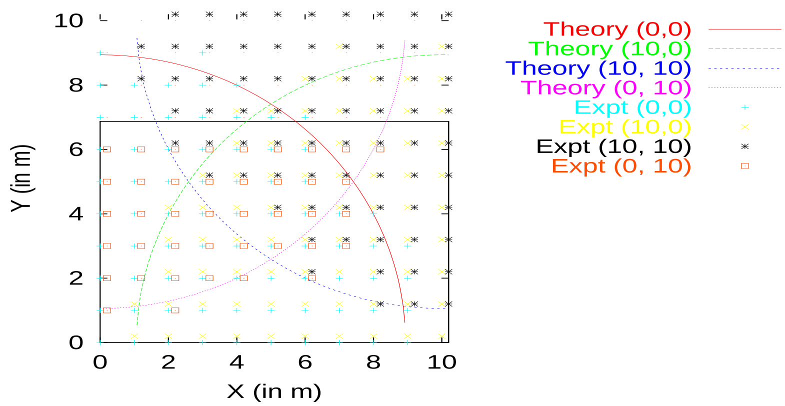


Figure 5.5: Experimental vs. theoretical 90% connectivity ranges for the four beacons.

These can be contrasted with the theoretically predicted overlap regions, also seen in Figure 5.5.

The location estimate at each grid point is the centroid. We use the *localization error* metric defined previously to characterize the performance.

In Figure 5.6, the *localization error* obtained from experiment is plotted as a function of the position. The localization error is lowest at the the position corresponding to the centroid of the region and increases towards the edges of the region. The mean localization error was 1.83 m and the standard deviation was 1.07 m. The minimum error was 0 m and the maximum error was 4.12 m across 121 grid points.

Figure 5.7 shows the cumulative localization error distribution across all the grid points, from both the theoretical model and the experiment. They track each other closely, including plateaus in the error levels, although the spherical model is consistently more optimistic. In our experimental results, for over 90% of the data points, the localization error falls within 3.0 meters i.e within 30% of the separation-distance between two adjacent beacons. This result is based on 4 beacons only. Since we observed a high correlation between our model and experiment, improved granularity can be expected with a higher overlap of beacons.

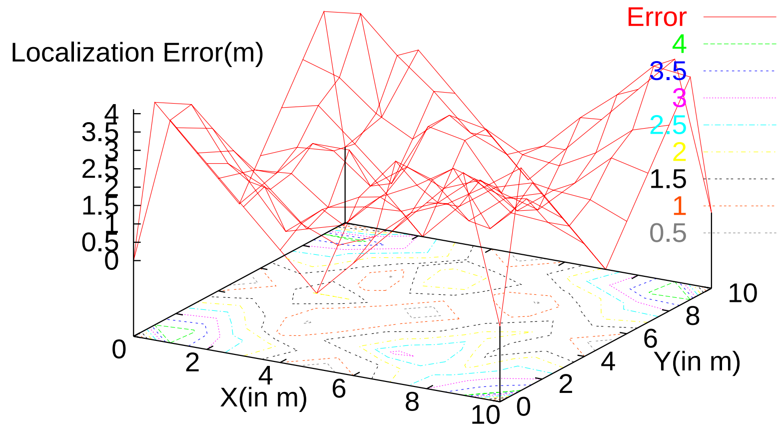


Figure 5.6: Localization error vs. position.

5.7 Detailed Simulations

Based on our validated outdoor model, we performed numerical simulations to predict how the granularity of localization could be expected to improve in our scheme when the overlap of beacons is increased.

In our simulation, we assume an infinite two-dimensional mesh of beacons, with any two adjacent beacons spaced a distance d apart and transmission range R. Our coordinate system is centered at one such beacon, which is assumed to be at $(0, 0)$.

The localization estimate of any point (X, Y) in the mesh can be obtained in two steps.

Step 1: Determine all the beacons which are within range R of (X, Y), by considering all the beacons in the rectangular region defined by $(X - R, Y - R)$ and $(X + R, Y + R)$.

Step 2: Localize (X, Y) to the centroid of the selected beacons and compute the cor-

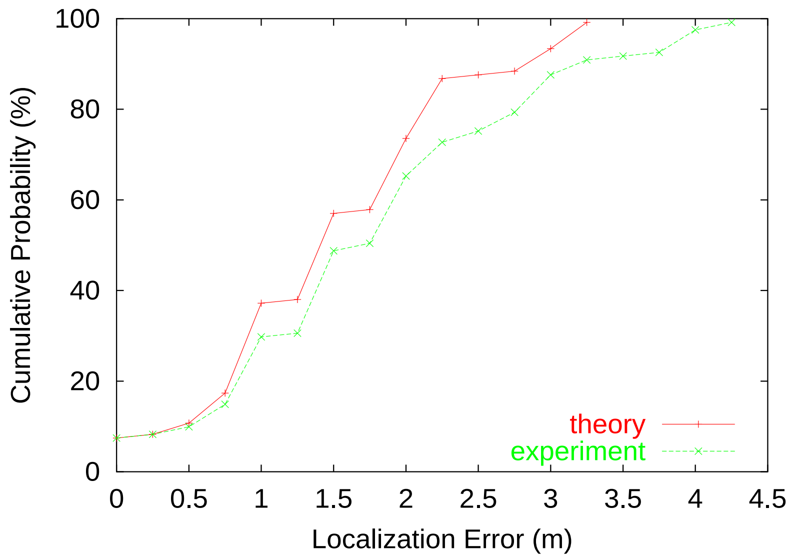


Figure 5.7: Cumulative localization error distribution.

responding localization error.

5.7.1 Localization Improvements with Increased Range Overlap

For a given d, we increase the overlap R/d from 1 to 4. We consider the mean and maximum localization errors of the localization estimates for 10201 uniformly spaced points within one grid in the mesh, for each R/d value. Figure 5.8 presents the *simulation based* scaling result of the localization error behavior. Although the maximum and mean error do not decrease monotonically, non-trivial increments to R/d, (for instance, an increment of 1) lead to lower maximum and mean localization errors on the whole. In particular, the maximum localization error experiences a substantial drop (from $0.5d$ to $0.25d$) when the overlap R/d is increased from 1 to 4.

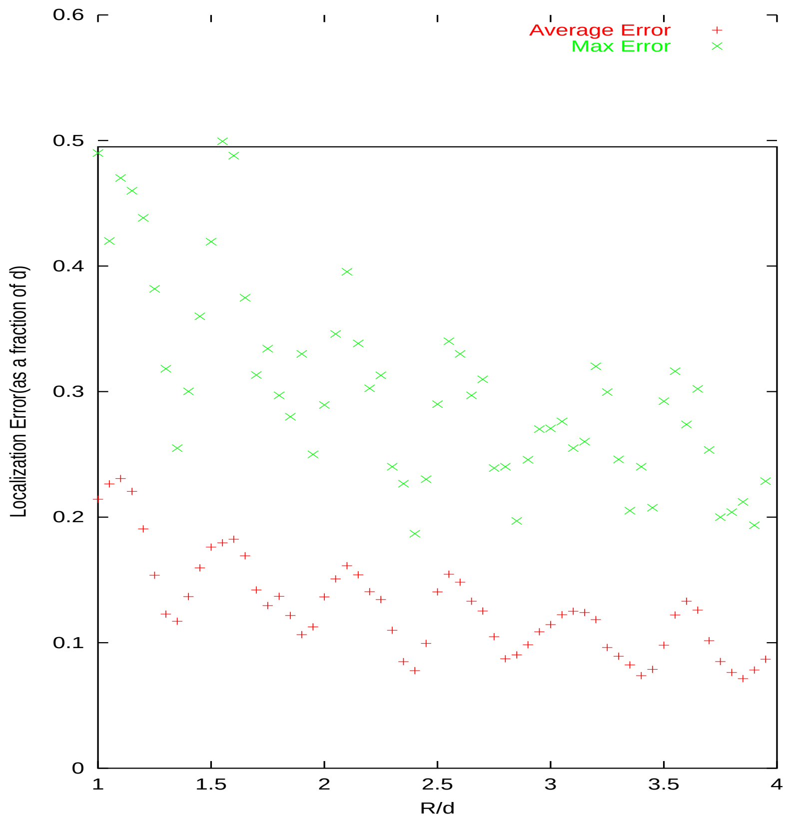


Figure 5.8: Localization error vs. range overlap, R/d. (Simulations)

5.7.2 Localization Improvements with Multiple Power Levels

Figure 5.9 illustrates the improvement in median localization error as the number of transmit power levels increases with ideal radio propagation. We see again that beyond a certain limit, additional transmit power levels do not provide much gains in localization.

5.8 Summary

In this chapter, we addressed the problem of node localization for very small devices that do not have GPS receivers. We explored an RF-based localization methodology in

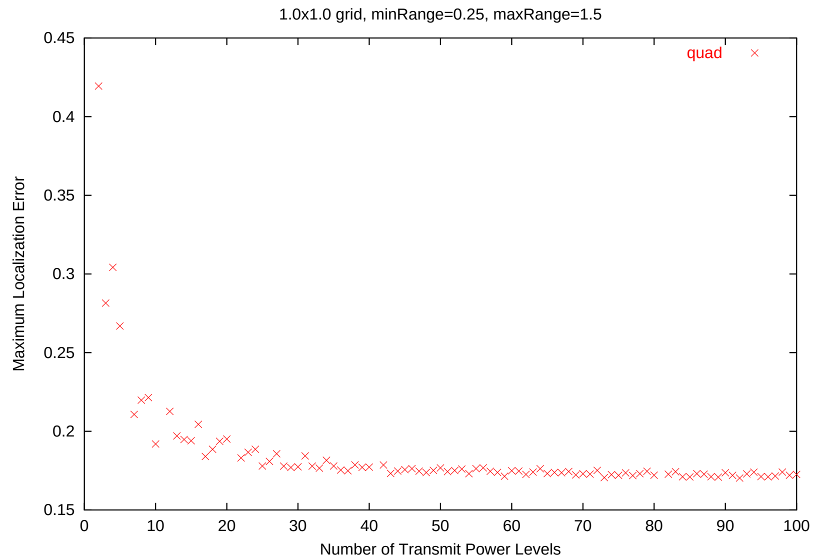


Figure 5.9: Median localization error vs. number of unique transmit power levels.

which the client localizes itself with high confidence (under an idealized radio model) to the weighted centroid of a set of proximate beacons using a connectivity metric. Although our approach uses a very simple radio model, in outdoor environments it correlated very well with reality (87 percent).

Our approach is simple, entirely RF-based, receiver-based and adaptive to the granularity of beacons available. Additionally, it requires no coordination among beacons or sensor nodes. It is therefore potentially scalable to very large distributed networks of devices.

Our experiments have shown promising results with our scheme for a small number of beacons. Our simulation results suggest that the granularity of localization can be further improved by increasing the overlap of beacons and by using multiple power levels.

The localization granularity from our RF-proximity based approach may not be sufficient for certain classes of applications. It is nevertheless useful for several applications with less stringent accuracy requirements. We will explore these in Chapter 10. However, several general problems still need to be tackled for large scale deployment, for example, adapting to noisy environments. We will address these in the next four chapters.

CHAPTER 6

Self-Configuring Beacon Systems

Making a system reliable is not really hard, if you know how to go about it. But retrofitting reliability to an existing design is very difficult.

— Butler Lampson [Lam83]

6.1 Introduction

The localization system we have developed and described in Chapter 5 uses a set of beacons (nodes with known positions) that are spatially distributed throughout the geographical region of interest, so that client nodes (nodes whose positions are known) can localize themselves by listening to nearby beacons. Besides our approach, several other proposed localization systems rely on beacons [PCB00, HHS99, WBV99]. Although such localization systems require an underlying infrastructure of beacons, they have two advantages over localization systems without beacons such as [Nag99], in which clients must establish a coordinate system and locate themselves in that coordinate system solely by communicating with each other. First, having beacons spatially distributed throughout the geographical region lets devices compute their location efficiently in a scalable, decentralized manner without loss of accuracy. Second, even when the application permits off-line, centralized position-estimation algorithms (as in [DPG01]), both the convergence and estimation accuracy can be significantly improved by having some nodes as beacons [DPG01].

In large ad hoc sensor networks that must operate unattended, our primary goal is to make the localization system simple to configure and deploy. In a beacon based localization architecture, this includes two major concerns.

- *Beacon Configuration.* Beacons must know their positions with respect to some coordinate system in order to advertise them. Each beacon needs to be configured with its spatial coordinates during deployment.

- *Beacon Placement.* How many beacons do we need? Where should they be placed? The beacon density and placement are important in influencing the overall localization quality. Uniformly dense placement is good and has its benefits, it is not adequate.

In the following sections, we explore various configuration issues for beacon systems in greater detail and motivate our approach based on self-configuration.

6.2 Automating Beacon Configuration

Automating the process of configuring beacons with their spatial coordinates is important for large scale and highly dense beacon deployment. This includes two issues — establishing a coordinate system (geodetic, Cartesian, polar) and a frame of reference, and estimating beacon locations in that coordinate system.

We can automate the process of assigning beacon coordinates using several techniques. In an outdoor setting, we can assume that beacons will infer their position through GPS [HLC92]. In this case, these positions would need to be transformed from geodetic coordinates (latitude, longitude) to Cartesian coordinates in the frame of reference [HLC92] if desired.

Table 6.1: Notation used to describe beacon systems for localization.

PARAMETER	DEFINITION
B	A beacon device that has knowledge of its position
	(and is presumed to be statically placed.)
C	A client device whose position is unknown
	(and can either be static, portable or autonomously mobile.)
$\mathcal{B}$	An assigned placement of beacons.
N	The total number of deployed beacons.
A	The total area in which beacons are deployed.
$position(B)$ and	The position (X and Y coordinates) of a beacon B.
$(\ X_B,\ Y_B\ )$	
$Range, R$	The nominal transmission range of the beacons, in an area A.

In an indoor setting, we believe that initial beacon placement will be structured. Only a few beacons will need to have their positions assigned manually, the rest can exploit this structure (for example, in a rectangular grid) in beacon placement to infer their coordinates. This is the approach used in fact in the HiBall tracker [WBV99].

6.3 Impact of Beacon Density

To understand the issues involved in beacon placement, we start by considering the impact of beacon density on the quality of localization in these systems. Table 6.3 describes the notation used to describe beacon systems used for localization.

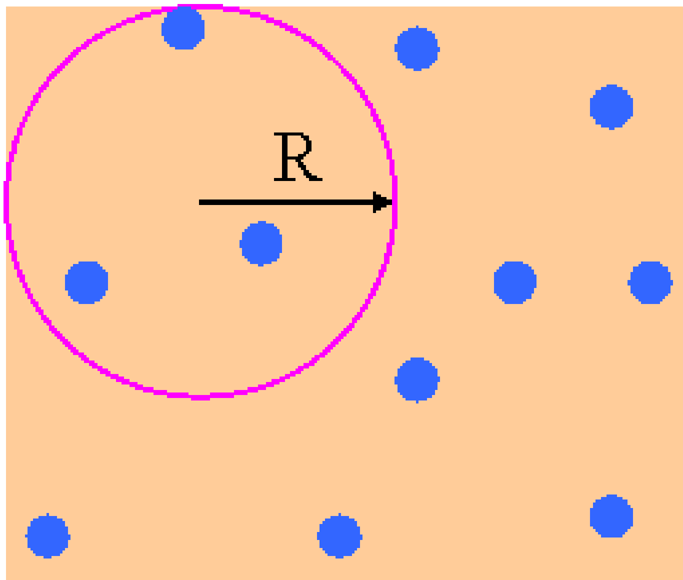


Figure 6.1: Beacons per nominal coverage area is the number of beacons in the circle of radius R (radio range).

6.3.1 Characterizing Beacon Density

How should we define beacon density?

Beacon deployment density ρ , a classical notion, denotes the number of beacons per unit area.

$$\rho = \frac{N}{A} \tag{6.1}$$

However, this definition does not abstract away the effect of the nominal communication (radio transmission) radius on the perceived density. We have come up with a new density metric that encapsulates the effect of the radio transmission range.

Beacons per neighborhood μ (also referred to as beacons per nominal radio coverage area $bpnrca$) denotes the number of beacons that exist in a nominal radio

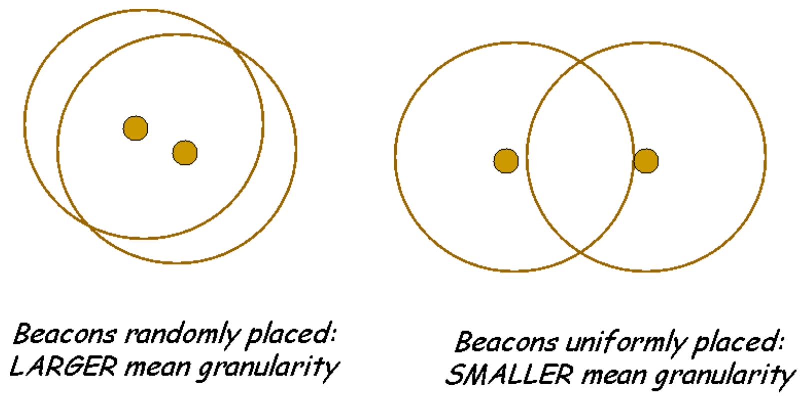


Figure 6.2: Impact of beacon placement on localization granularity.

transmission coverage area ($\pi \cdot Range^2$). This is illustrated in Figure 6.1.

$$\mu \quad = \quad \rho \cdot \pi \cdot Range^2 \tag{6.2}$$

6.3.2 Impact on Localization Granularity

For a given beacon placement $\mathcal{B}$ and a square terrain of area $A = Side \times Side$. Let r denote the ratio $\frac{Side}{step}$. Let us the define a grid of points $step$ units apart, the point $P(k,l)$ as follows:

$$P(k,l) \quad = \quad (k \cdot step, l \cdot step) \quad \forall\, 0 \leq k, l \leq r \tag{6.3}$$

The quality of localization in the terrain can be characterized in terms of statistical metrics such as the mean and median localization error over various points in the terrain, defined as follows.

$$MeanErr(\mathcal{B}) \quad = \quad \frac{\Sigma_{k=0}^{r}\Sigma_{l=0}^{r} LE_{\mathcal{B}}(P(k,l))}{(r+1)^2} \tag{6.4}$$

$$MedianErr(\mathcal{B}) \quad = \quad median(LE_{\mathcal{B}}(P(k,l))) \quad \forall\, 0 \leq k, l \leq r \tag{6.5}$$

Figure 6.3 plots the mean localization error as a function of beacon density. Regardless of actual beacon placement, the localization granularity saturates at a certain

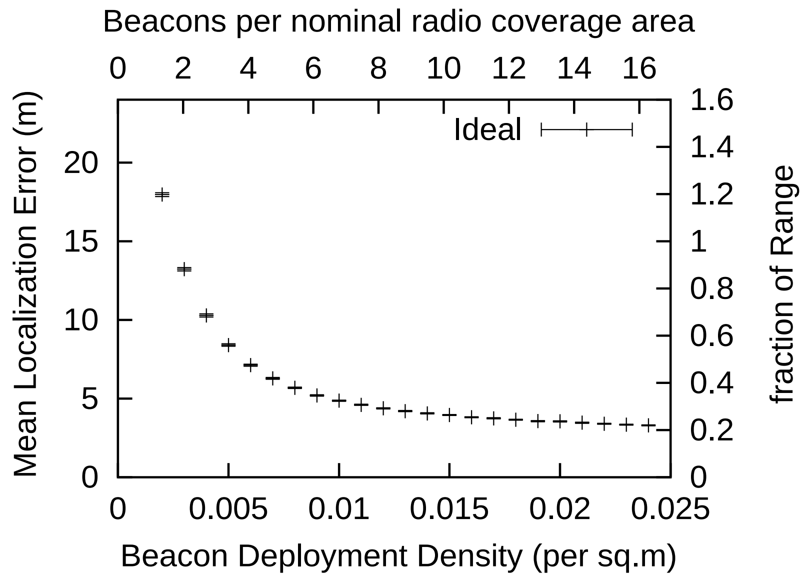


Figure 6.3: Mean localization error vs. Beacons per nominal radio coverage area. (Simulations).

threshold beacon density μ_{thresh}. Localization granularity saturates at a certain number of beacons per neighborhood, around 6 in our case. The graph is based on simulations of 1000 random topologies per beacon density.

6.3.3 Impact on Channel Contention and Self-Interference

Consider a contention-based underlying media access protocol, wherein more than one node may attempt to transmit at the same time *i.e.,* contend for the wireless channel. An example of such a media access protocol designed for wireless sensor networks is SMAC [YHE02].

Let us assume the beacons per nominal radio coverage area is μ. Assume that any given instant, the probability of a beacon transmitting an advertisement packet is p. If T_x is the transmission time of an advertisement packet and the beaconing interval is T

then

$$p = \frac{T_x}{T} \tag{6.6}$$

Let $p_{success}$ denote the probability that the packet is successfully received without any interference. Let $p_{collision}$ denote the probability of collision in the wireless system. We can model the channel contention as follows.

Let X indicate number of beacons that will try to transmit a packet.

$$p_{success} = Pr[X = 1] \tag{6.7}$$

$$= p \cdot (1 - p)^{\mu} \tag{6.8}$$

$$p_{collision} = 1 - p_{success} \tag{6.9}$$

This shows us that the probability of packet collision increases exponentially with the beacon density μ. In order to maintain the same collision probability $p_{collision}$ at a higher beacon density, we need to significantly reduce the packet transmission probability p. Since the transmission time of a beacon advertisement packet T_X is fixed, this means that we must correspondingly increase the beaconing interval T. Since the sampling time of a client for its location computation (defined in Chapter 5) is directly proportional to T, this means that there is a corresponding increase in location computation latency. Thus, we cannot simultaneously increase beacon density and maintain the same system responsiveness for localization.

6.3.4 Two Assertions about Beacon Density

We can make two assertions about beacon density in the context of proximity-based localization systems with localized location computation (Chapter 5).

1. Regardless of actual beacon placement, the localization granularity saturates at a certain threshold beacon density μ_{thresh} (see Figure 6.3, $\mu_{thresh} = 6$ bpnrca).

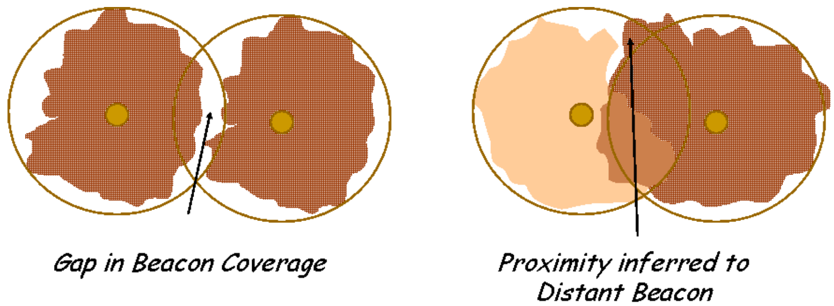


Figure 6.4: Impact of RF propagation vagaries on localization granularity.

2. As the beacon density increases, the probability of collisions among competing beacons vying for the same transmission slot increases. (Chapter 5, see also [PCB00]).

At low and medium beacon densities, the quality of localization suffers due to poor placement of beacons due to various environment and calibration-dependent vagaries which we will discuss in the next two sections. Unfortunately, we cannot predict these problems before hand. These problems cannot be addressed at design-time. This motivates run-time self-configuration of the localization system.

6.4 Impact of Environment

We discussed the characteristics of radio propagation in Chapter 2. In various subtle ways (*e.g.* path-loss, shadowing and multi-path), the environment affects the quality of radio propagation, and consequently localization in an RF-based localization system (see Figure 6.4).

Tolerance of random placement (see Figure 6.2) or high node mobility are not the only reasons to design sensor networks to be self-configuring. Even in cases where they are placed uniformly and do not move, nodes must independently self-organize to coordinate for collaborative sensing functions [SGA00, CHZ02].

The environments in which these systems are expected to operate will be time-varying due to RF propagation vagaries and other environmental dynamics (for evidence of time-varying behavior, see measurements by Zhao [ZGE02, Zha02]). In addition to time-varying components, many characteristics of the environment will be a function of fixed elements, such as trees or hills on a terrain. Although time-varying effects can be analyzed statistically [GKW02], errors and distortions resulting from fixed elements must be compensated by detecting and adapting to these conditions. An approach aimed at characterizing the environment has the potential to improve sensing fidelity as well as energy efficiency. For example, in the multi-modal localization system [GE01b] previously described, nodes could retain long-term information about non line of sight pairs detected when obstructions change slowly.

Savvides *et al* [SHS01a] propose an approach by which nodes in a wireless network can improve the accuracy of their RSSI based location estimates (discussed in Section 2) by dynamically deriving (learning) the surrounding wireless channel properties. The algorithm starts with an initial guess of channel properties[1] and tries to obtain node position estimates through a sequence of successive multilateration. The initial set of position estimates can now be used to obtain an initial estimate of the channel properties by providing two crucial components: (i) A large set of inputs for the estimation of the channel parameters. (ii) A corresponding error variance that is used as a weight for each input in the channel model estimator.

Using these inputs, the channel model estimator can produce a new estimate of the channel properties which can be used in subsequent multilaterations. The process is repeated until the values of the channel model, and consequently position estimates converge to a specified tolerance.

This approach makes it a versatile solution that even without prior calibration can

[1] For instance, parameters such as the additive Gaussian channel noise in the log-normal shadowing model[Rap96].

work in many different settings where the propagation channel properties are different. Furthermore, if the sensors are deployed over a wide area, the signal propagation characteristics may vary widely even across the region of interest. Calculating the propagation characteristics locally yields better accuracy in the node location estimates.

6.5 Impact of Sensor Calibration

As in any sensor system, *calibration* is important to our RF-based localization system. Calibration refers the creation of beacon specific information such that any given beacon can transmit at the same power as the others and can accurately map its transmit power level to a nominal transmission range. Characterizing and accounting for beacon specific variations in this way insulates higher level localization algorithms from hardware dependencies and the details of signal processing.

When beacons are un-calibrated, variations among beacons can cause large variations in perceived beacon density, anisotropic coverage, asymmetric connectivity, etc. These can degrade the performance of the localization scheme.

While simple filtering techniques can eliminate some outliers, beacons must still be able to auto-calibrate and compensate for the perceived differences. Hightower *et al* [HWB00] and the TinyOS project at Berkeley have been studying how to implement self-calibration in sensor nodes. We can potentially leverage their techniques.

6.6 Goals of Self-Configuration

The goal of self-configuration is for beacons to automatically adapt to variations in density, environment and miscalibration. Others have addressed adapting to a fixed environment [SHS01a, GE01b] and to miscalibration [HWB00].

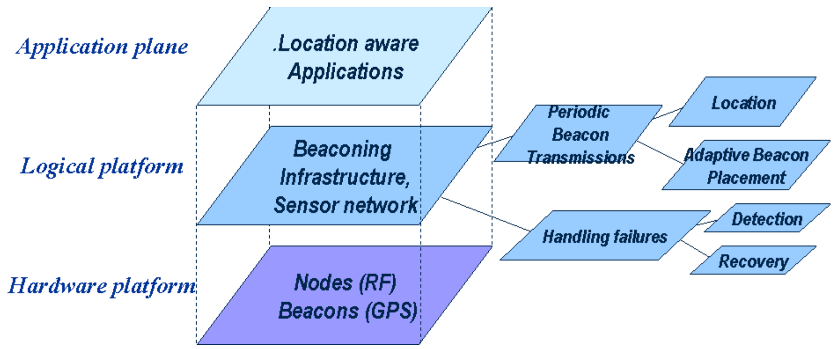


Figure 6.5: A self-configuring localization system architecture.

This dissertation focuses on adapting to beacon density. We have formally introduced the notion of beacon density and shown that the quality of localization can be related to the density of beacons. The deployed density of beacons is however not equal to the actual beacon density.

We have shown that different problems arise at different deployment densities, such that density guides the approach to self-configuration. We discuss the following two forms of self-configuration in this dissertation.

- *At low and medium densities:* Are the deployed beacons enough to guarantee good localization quality throughout the terrain? How do we ensure this? If they are not enough, how can we add beacons to improve the quality of localization.

- *At high densities:* How do we coordinate densely deployed beacons so as to reduce channel contention while best exploiting the spatial diversity and redundancy of densely-deployed beacons?

6.7 Summary

In this chapter, we explored various issues in deploying beacon systems for localization. We studied the impact of beacon density, environment and sensor calibration on localization quality. Our approach was bimodal — we studied these issues to motivate why beacon systems must self-configure. Conversely, we also identified various forms of self-configuration and show how self-configuring beacon systems (organized according to the architecture described in Figure 6.5) can eliminate these deployment issues.

We discuss three different forms of self-configuration in Chapters 7, 8 and 9 respectively.

CHAPTER 7

GRID: Centralized Incremental Beacon Placement

To measure is to know. If you cannot measure it, you can not improve it.

— Lord Kelvin

In Chapter 6, we established that beacon placement would affect the quality of localization. In this chapter, we formalize and address the problem of *adaptive beacon placement*: given an existing field of beacons, how should additional beacons be placed for best advantage?

We develop novel algorithms that permit a person or mobile robot to place additional beacons to incrementally extend an initial beacon field. This allows for *measurement-based* adaptation to terrain conditions. We also evaluate the gains from incrementally improving an RF-based location field using extensive simulations.

If the only way to improve the quality of localization in a region by adding an additional beacon is to place it at a single point in the region, then it is difficult to design algorithms that can identify that point in the presence of so much noise. The design of our algorithms is predicated on the notion of *solution space density* [BEG01]. The efficacy of algorithms (such as our beacon placement algorithms) designed to work in noisy environments is predicated on the assumption that the solution space for the problem must be dense in number of satisfying solutions. We are seeking a reasonable solution, not necessarily a unique optimal solution.

7.1 Motivation

Intuitively, a uniformly dense placement of beacons should suffice to ensure a certain quality of localization. Uniform placement is good, but insufficient due to the following reasons:

- Beacons may be perturbed during deployment. Consider for instance, a terrain comprising of a hilltop. Air dropped beacon nodes will roll over the hill, while lighter sensor nodes may stay atop the hill.

- Even when beacon placement is uniform, noise (in the form of terrain and propagation uncertainties) may affect the visibility of beacons that should be in range. Radio signal propagation in general is significantly affected by multi-path effects, fading, shadowing *etc.* Uneven terrains and obstacles bring in an additional dimension of uncertainty [Rap96].

Very dense placement may not be a viable solution due to several reasons:

- Cost or Power: The cost of the beacons may preclude very dense beacon placement. Power considerations may require that only a restricted smaller subset of beacon nodes be active at any given time so as to prolong system lifetime [EGH99, XHE00].

- Terrain Commonality: Even when cost is not a concern, the environmental or terrain conditions may be such that merely increasing the density uniformly will not overcome the problem. For instance, if the number of air-dropped beacons were doubled, the same situation would persist. Also, the terrain may already have a very high density of beacons (enough to achieve the maximum possible quality of localization under ideal conditions) and hence the new beacons must be added in particular places to cope with noise.

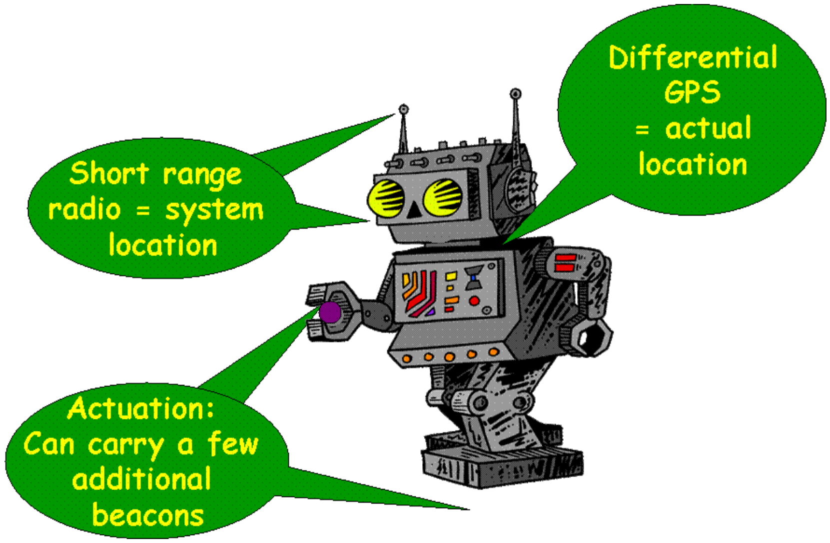


Figure 7.1: Mobile robot capabilities for instrumenting terrain.

- Self-interference: At very high densities, the probability of collisions among signals transmitted by the beacons increases. Therefore even if we had unlimited numbers of beacons, we would like to limit their use.

The fundamental limitation of these two approaches is that they are basically fixed strategies, that do not take into account environmental conditions that cannot be predicted *a priori*. It is virtually impossible to preconfigure to such terrain and propagation uncertainties and compute an ideal (or even satisfying) beacon placement that uniformly achieves a desired quality of localization across the region. Clearly, the beacon placement needs to adapt to the noisy and unpredictable environmental conditions.

7.2 Design Considerations

Given a localization algorithm, one must deploy a field of beacons as infrastructure, and then extend this field if it proves insufficient.

Our approach to incremental improvement of localization through beacon placement is based on *measurement-based adaptation*. By *adaptation*, we mean we are improving the quality of localization by adjusting beacon placement or adding a few beacons rather than by completely re-deploying all beacons. By *measurement-based*, we mean the deployment of additional beacons is influenced by measurements of the operating localization system rather than by careful or complete off-line analysis of a complete system model.

Our general approach is to use a GPS-equipped mobile robot or human to explore the terrain. We assume that the robot (or human) can determine its geographic position using a high precision differential GPS receiver and map it to the local coordinate system. The robot has a short range radio similar to the one used by the sensor nodes, and can thus compute its localization estimate using the connectivity based localization algorithm. Thus it has a means of computing the localization error at any point on the terrain. It also has a capability to carry a certain number of beacons that it can deploy as additional beacons wherever it deems necessary (see Figure 7.1). Therefore, based on its measurements of localization error at different points in the region, it must compute good places to deploy additional beacons (illustrated in Figure 7.2) and deploy them. [1] We define this problem as *adaptive beacon placement*.

7.2.1 Assumptions

The design space of possible robot-based beacon placement algorithms is very large. We have begun with a simple choice: an off-line algorithm with complete terrain exploration and no measurement noise. We use this simple problem to define the problem and preliminary solutions. These solutions can be generalized to other problems.

[1] In general, the SCOWR project [SCO] focuses on incorporating robotic motion and communication into distributed sensing applications (e.g see [MS00]).

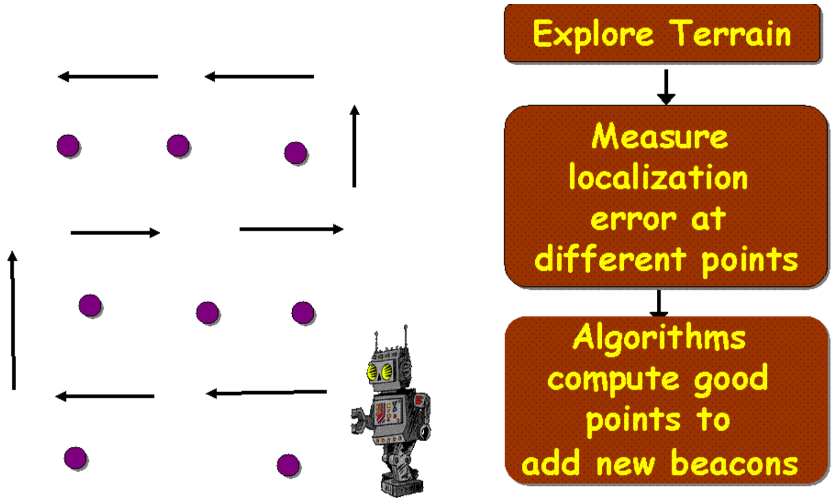


Figure 7.2: The GRID approach to adaptive beacon placement.

7.2.2 Problem Definition: Incremental Beacon Placement

More formally, the problem of incremental beacon placement can be stated as follows.

Given:

τ - A square terrain of side $Side$ and area A.

$\mathcal{B}$ - An initial placement of beacons in area A.

M - A set of localization error measurements at various points in the terrain $step$ meters apart ($step < Side$) recorded by a robot.

R - Nominal transmission range for each beacon.

Find:

C - A candidate point where a new beacon can be added to improve localization.

7.3 Grid Design

Before we describe Grid, we describe two simple and intuitive off-line algorithms, Random and Max, for incremental beacon placement. The goal of all these algorithms is to determine candidate points for placement of an additional beacon, so as to maximize the gains obtained. These three algorithms differ in the amount of global knowledge and processing used to make their decision.

7.3.1 Random

This is the simplest algorithm, which pays no attention to the quality of localization at different areas of the region and simply selects a random point in the region as a candidate point for adding an additional beacon.

Step 1 Select a random point (X_r, Y_r) in the terrain.

Step 2 Add a new beacon at (X_r, Y_r).

We investigate this primarily for comparison with the other algorithms, but also because it is similar in character to uncontrolled airdrop of additional nodes. The complexity of this algorithm is $O(1)$.

7.3.2 Max

The Max algorithm (illustrated in figure 7.3) can be described in three steps:

Step 1 Divide the terrain into $step \times step$ squares.

Step 2 Measure localization error at each point $(i \times step, j \times step)$ in the terrain that corresponds to a square corner. ($0 \leq i, j \leq \frac{Side}{step}$)
Number of data points in the terrain, $P_T = (\frac{Side}{step} + 1)^2$.

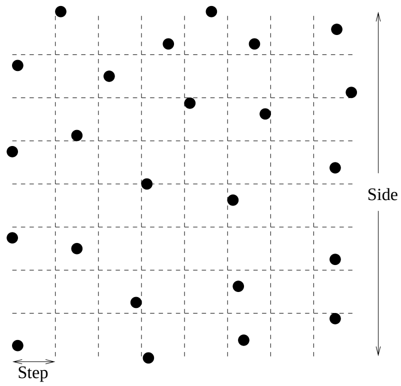


Figure 7.3: An illustration of the Max algorithm.

Step 3 Add new beacon at the point (X_m, Y_m) that has the highest measured localization error among all points.

This algorithm is predicated on the assumption that points with high localization error are spatially correlated. The advantage of this algorithm is that it can be computed in a very straightforward way. However, it may be overly influenced by propagation effects or random noise that may cause very high localization error at one point while the localization error at points very close to it remains low; *i.e.*, it is sensitive to local maxima.

The complexity of the Max algorithm is linear in P_T, the number of data points at which the localization error is measured *i.e.*, $O(P_T)$.

7.3.3 Grid

The Grid approach to determining a candidate point is to compute the cumulative localization error over each grid, for several overlapping grids in the terrain. This is based on the observation that adding a new beacon affects its nearby area, not just the point where it is placed.

The Grid algorithm (illustrated in figure 7.4) consists of the following steps:

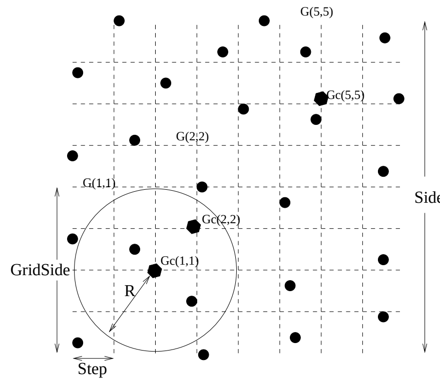


Figure 7.4: An illustration of the Grid algorithm.

Steps 1 and 2 are the same as the Max algorithm.

Step 3 Divide the terrain into N_G partially overlapping grids as follows:

Step 3.1 Each grid has a side, $gridSide = 2R$. Thus each grid encloses the radio reachability region of its center.

Step 3.2 For $1 \leq i, j \leq \sqrt{N_G}$, the grid $G(i,j)$ is defined by its center $G_c(i,j) = (X_c(i,j), Y_c(i,j))$ where
$X_c(i,j) = \frac{gridSide}{2} + \frac{(i-1)(Side-gridSide)}{\sqrt{N_G}-1}$
and $Y_c(i,j) = \frac{gridSide}{2} + \frac{(j-1)(Side-gridSide)}{\sqrt{N_G}-1}$.

Step 4 For each grid $G(i,j)$, compute the cumulative localization error $S(i,j)$ at all the points measured in Step 2 that lie in the grid $G(i,j)$. Number of data points per grid, $P_G = P_T \times \frac{Side^2}{(2.R)^2}$.

Step 5 Add the new beacon at the center $G_c(i,j)$ of the grid $G(i,j)$ with the maximum cumulative localization error.

While the Grid algorithm has the advantage that it can improve many points at once, it is computationally far more expensive than the Max and Random algorithms because it additionally divides the terrain into several grids and computes the cumulative localization error in each grid.

The complexity of the Grid algorithm is linear in the product of N_G, the number of grids considered and P_G, the number of data points per grid at which the localization error is measured., i.e $O(N_G.P_G)$.

Section 7.4 provides a performance comparison of these three algorithms. We note that these are by no means the only possible algorithms, but these are representative of the effectiveness attainable with different degrees of processing.

7.4 Performance Evaluation

In this section, we report on some results from a preliminary performance evaluation of our beacon placement algorithms. We use numeric simulations to explore, in some detail, the implications of several design choices.

7.4.1 Goals, Metrics and Methodology

Our goals in conducting this evaluation study were three-fold:

- Place the performance of Grid and Max algorithms in the context of the Random algorithm. This serves as a sanity check for the intuition behind the Grid and Max algorithms, as also to explore the influence of the level of knowledge on algorithm performance.

- Understand role of beacon density on algorithm performance.

- Understand impact of noise such as propagation losses and terrain features on the beacon placement algorithms.

We choose two metrics to analyze and compare the performance of our algorithms. These metrics are statistics evaluated for the observed localization error at all $step \times step$ square corners obtained by subdividing the region.

Improvement in mean localization error M_1 computes the difference between mean localization error at all measured points in the terrain before and after the beacon node is added. This metric indicates the overall impact of adding a beacon to quality of localization in the entire terrain. For a given beacon placement $\mathcal{B}$

$$\mathit{MeanErr}(\mathcal{B}) = \frac{\sum_{k=0}^{\frac{Side}{step}} \sum_{l=0}^{\frac{Side}{step}} LE_{\mathcal{B}}(P(k,l))}{(\frac{Side}{step}+1)^2}$$

where $P(k,l) = (k \cdot step, l \cdot step)$

$$M_1 = \mathit{MeanErr}(\mathcal{B}_{init}) - \mathit{MeanErr}(\mathcal{B}_{final})$$

Improvement in median error M_2 computes the difference between the median localization error at all the measured points in the terrain before and after the beacon node is added. This metric indicates the improvement due to adding a beacon on the quality of localization at the top 50% of the points with the highest localization error at the terrain.

$$M_2 = \mathit{MedianErr}(\mathcal{B}_{init}) - \mathit{MedianErr}(\mathcal{B}_{final}) \tag{7.1}$$

We study these metrics as a function of beacon density. We consider a square terrain of side 100m. Each node has a nominal radio range of 15m. To study the performance of our algorithms as a function of beacon density, we generate a variety of beacon fields of different densities.

In each of our experiments, we vary the number of beacon nodes from 20 to 240 in increments of 10 beacon nodes. The corresponding beacon density varies from 0.002

Table 7.1: Various GRID simulation parameters.

PARAMETER	VALUE
$Side$	100m
R	15m
$step$	1m
N_G	400

beacons per square m to 0.024 beacons per square m. To put these density values in context, the corresponding number of beacons per nominal radio coverage area (πR^2) varies from 1.41 to 17. For each density, we generate 1000 different beacon fields. Each beacon field is generated by randomly placing the beacons in the $100m \times 100m$ square terrain. The performance metrics, for each algorithm and beacon density, are averaged over the 1000 beacon fields. To characterize the stability of our results, all graphs include 95% confidence intervals. The simulation parameters are listed in Table 7.1.

7.4.2 Impact of Beacon Density

As observed earlier, beacon density has a considerable impact on quality of localization. To quantify this effect, we evaluate the relationship between mean localization error and beacon density. Figure 7.5 graphs mean localization error for varying beacon densities under idealized radio propagation conditions. We see that the mean localization error falls sharply with increasing beacon density, until it reaches a density of 0.01 beacons per square m (approximately 7 beacons per nominal radio coverage area) and saturates at around 4m ($0.3R$). We refer to this density as the *saturation beacon density*. There is little to be gained from deploying beacon nodes at more than this density.

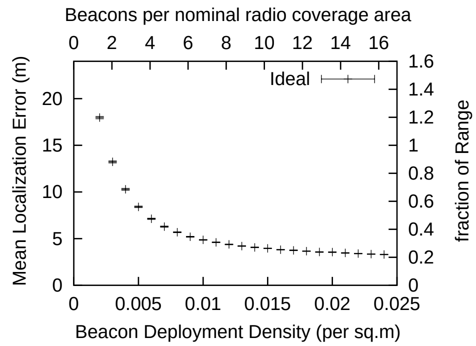


Figure 7.5: Mean localization error vs. beacon density (Ideal)

Our first experiment compares the three algorithms under idealized radio propagation conditions (*i.e.*, perfect connectivity for distances $\leq R$, no connectivity otherwise). The aim of this experiment is to isolate and study the impact of beacon density on the Random, Max and Grid beacon placement algorithms.

Figures 7.6 graphs the improvements in mean and median localization error for various beacon densities. As expected the Random algorithm has the least improvement.

At low densities (≤ 0.005, much below saturation density), the Grid algorithm clearly performs best, with improvements in mean localization error at least twice that of the Max algorithm. Grid achieves such performance because it considers the quality of localization over a grid, and can improve many points at once. The performance of Max is slightly better than Grid for regions of moderate density (0.008 to 0.02 per square m). At these densities, the points with maximum localization error are very loud, and Max suppresses them better. At very high beacon densities(≥ 0.02 beacons per square m), the quality of localization is saturated, and the performance of the three algorithms is about the same.

A similar trend with respect to beacon density is observed for the median localization error, although the improvements in median localization error are relatively more

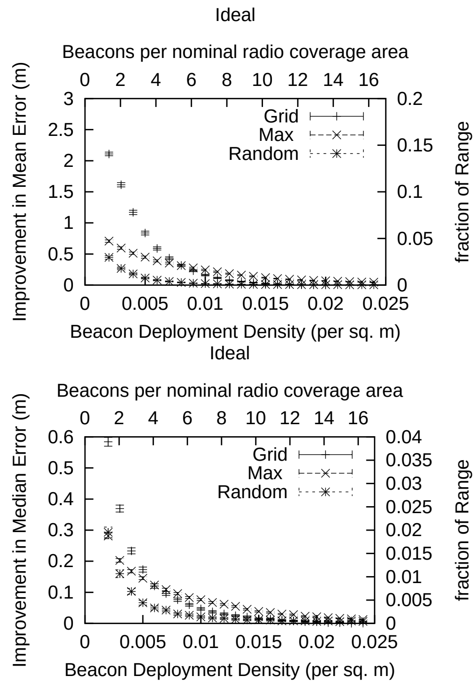


Figure 7.6: Improvement in mean and median errors vs. beacon density (Ideal)

modest (roughly 25% of the improvements in the average localization error with Grid). This is because the algorithms are effective in fixing a few hot spots with high localization error with the addition of a single beacon rather than in improving the localization throughout the terrain.

From our analysis we infer that, at least under idealized conditions, our beacon placement algorithms (Grid and Max) are applicable only to a regime corresponding to low or insufficient beacon density.

7.4.3 Impact of Noise

As stated earlier, idealized radio propagation conditions are rather unrealistic. Random noise can severely affect radio connectivity [Rap96], and thereby degrade the quality of localization. Since this noise cannot be predicted, beacon placement algorithms

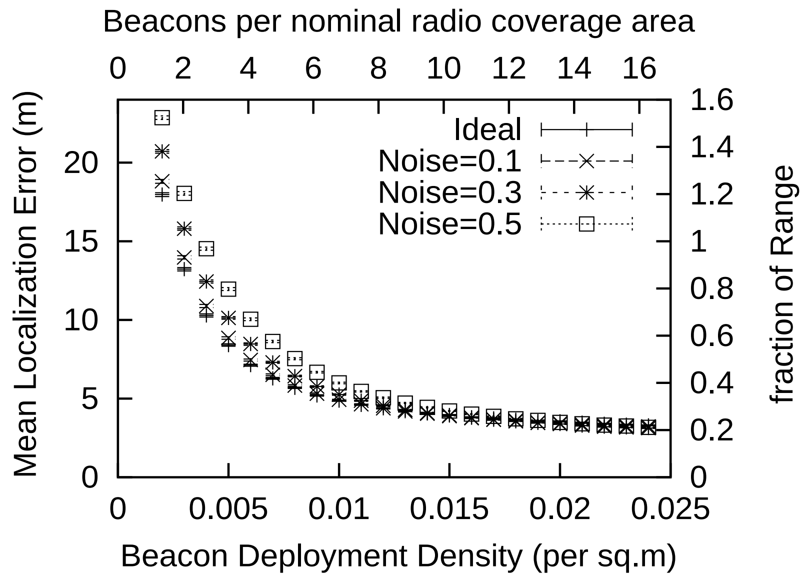


Figure 7.7: Mean localization error vs. beacon density (Noise)

must adapt to it through measurements. To study the impact of noise on our beacon placement algorithms, we model random propagation noise as follows. For each beacon field, connectivity to any beacon B at any given point P is determined based on a noise model. In our noise model, connectivity to a beacon B exists at a point P, if $distance(P, B) \leq R(1 + u.nf(B))$. $nf(B)$ is the noise factor of the beacon B, and is chosen uniformly between 0 and $Noise$, the maximum noise factor for the field. u is chosen uniformly at random between -1 and 1. The intent was to create non-uniform propagation noise for the beacons, and to create random regions with higher propagation noise than the rest of the location field. We do this because the impact of noise is less evident when each beacon has an identical propagation field. Note that this noise model is location based and static with respect to time *i.e.,* not time varying. We use 4 different settings of $Noise$, 0 (corresponding to Ideal propagation), 0.1, 0.3 and 0.5.

To quantify the impact of noise, we evaluate the variation in mean localization error from the ideal case in the presence of noise. Figure 7.7 plots the mean localization error as a function of beacon density for various noise levels. We observe a steady increase

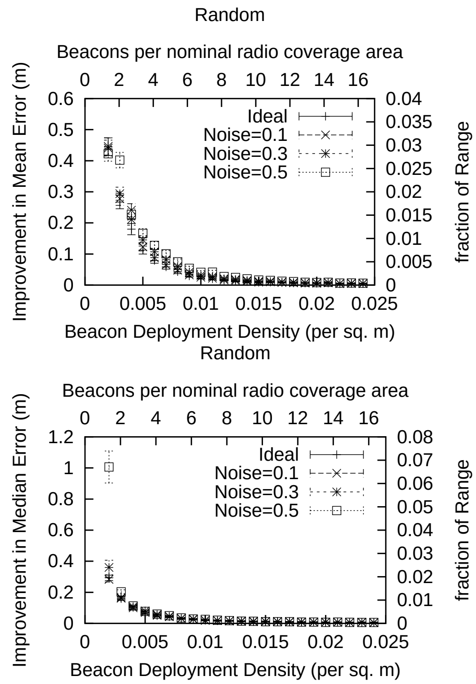


Figure 7.8: Performance of the Random algorithm with Noise

as the level of $Noise$ increases from 0 to 0.5, in both the mean localization error for each beacon density (*e.g.*, from 18m to 23m for 0.02 beacons per square m) and in the saturation beacon density (from 0.01 to 0.015 beacons per square m). The mean localization error follows the same general trend with increasing beacon density with noise as with idealized radio propagation.

Figure 7.8 graphs the improvement in the mean and median localization error when an additional beacon is placed with the Random algorithm, for various beacon deployment densities and noise levels. The gains in both metrics with the Random algorithm are somewhat unchanged with noise. This result is as expected, because noise is not an input in the Random algorithm, which does not make any measurements.

Figures 7.9 and 7.10 graph the improvement in the mean and median localization error when an additional beacon is placed with the Max and Grid algorithms respectively, for various beacon deployment densities and noise levels. We observe that noise

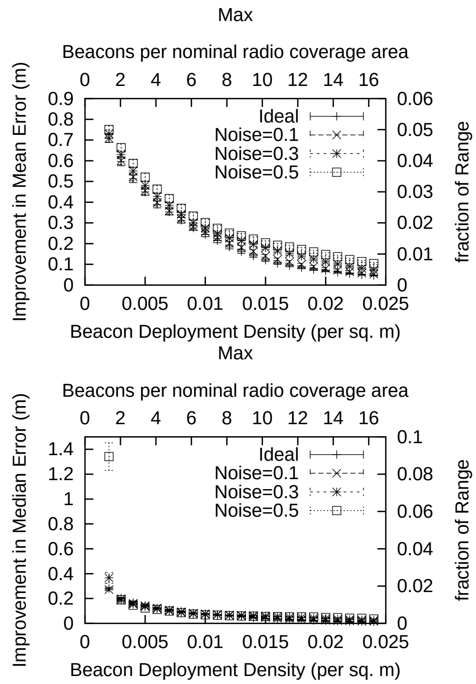


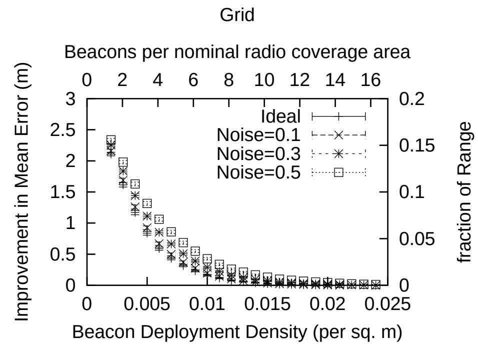
Figure 7.9: Performance of the Max algorithm with Noise

makes regions of moderate beacon densities (0.005 to 0.01 beacons per square m) more amenable to improvement (improvements of 0.5m to 1m in mean error for corresponding increases in mean error of 1m to 3m) with the Grid algorithm, and to a lesser extent with the Max algorithm. The improvements to the median error are relatively unchanged with noise, because as we noted earlier, the focus of the algorithms is on improving a few hot spots.

7.4.4 Summary of Results

There are several lessons that we can draw from this evaluation of or beacon placement algorithms:

- Our beacon placement algorithms are applicable to a regime corresponding to low or insufficient beacon density deployment(≤ 0.01 beacons per square m or



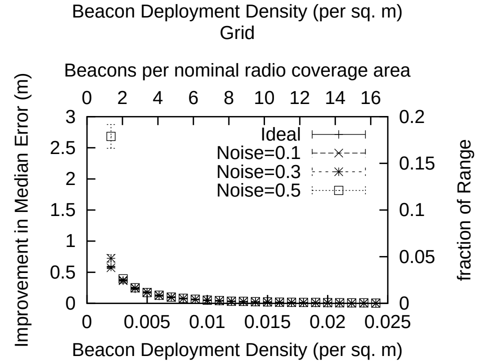


Figure 7.10: Performance of the Grid algorithm with Noise

7 beacons per nominal radio coverage area in the ideal case).

- At low densities, the Grid algorithm has the potential for significant improvements to the mean and median errors compared to the Max or Random algorithms.

- When noise level is increased from 0 to 0.5, there is a steady increase in both the mean localization error (up to 33%) and saturation beacon density (up to 50%).

- The Grid algorithm is clearly superior to Max and Random algorithms even in the presence of noise. The performance of the Random algorithm is unchanged with noise, whereas noise makes even moderate density regions more amenable to improvement with the Grid algorithm.

7.5 Summary

In this chapter, we emphasized the importance of beacon placement in localization approaches and motivated the need for empirically adaptive beacon placement algorithms. We described the notion of solution space density, which forms the basis for our algorithms.

We outlined a general approach for adaptive beacon placement based on exploration and instrumentation of the terrain by a mobile human or robot agent. We designed and evaluated three algorithms based on this approach: Grid, Max and Random. Our algorithms are applicable to a regime of low and medium beacon density deployment. In this regime, Grid clearly outperforms the Max and Random algorithms. In our simulations, we showed that beacon density rather than the noise level has a higher impact on the performance of beacon placement algorithms. When the noise level is increased from 0 to 0.5, there is a steady increase in both mean localization error (up to 33%) and saturation beacon density (up to 50%). The algorithms exhibited the same relative trend in the presence of noise as in an ideal scenario, although noise makes regions of moderate beacon density more amenable to improvement.

Although, we have evaluated our algorithms in the context of beacon placement for RF-based localization, they may generalize to other problem domains where issues of node placement are rather critical: global coverage or universal connectivity in wireless sensor networks, measurement based repositioning of seismic sensor nodes (surface conditions, coupling with the ground are significant influences on the quality of sensing attainable in these nodes). In traditional Internet web caching, the placement of web caches may be done based on analyses of web traffic, web server requests.

The novel aspect of our approach is the emphasis on *empirical adaptation*. The drawback with that approach is that its reliance on a mobile agent to make terrain measurements limits its scalability to large terrains and its applicability when agent-based measurements of localization error at arbitrary positions are not possible.

CHAPTER 8

HEAP: Localized Incremental Beacon Placement

An approximate answer to the right question is worth a good deal more than the exact answer to an approximate problem.

— John Tukey

You only need sit still long enough in some attractive spot in the woods that all its inhabitants may exhibit themselves to you by turns.

— Henry David Thoreau, from the chapter ”Brute Neighbors” in Walden

At low and medium densities, the beacons deployed in an ad hoc manner for localization may not be sufficient to ensure robust localization throughout the terrain. Although GRID serves a critical function in addressing this problem, its drawback is its reliance on a mobile agent to make terrain measurements. This limits its scalability to large size terrains. Furthermore, it cannot be applied when agent-based measurements of localization error at arbitrary positions in the terrain may not be possible.

In this chapter, we describe HEAP, an *adaptive*, *localized* algorithm that enables beacons to select candidate points for incremental beacon placement in order to improve the quality of localization in the terrain.

As in the case of GRID, the goal of HEAP is incremental beacon placement i.e., to discover places to add a few new beacons to maximize improvement in localization, rather than to completely re-deploy the beacon field.

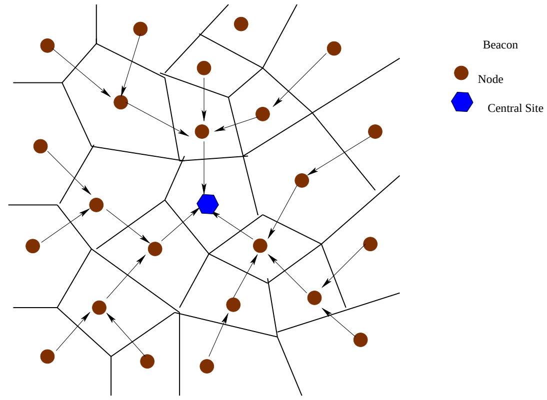


Figure 8.1: Information flow in HEAP.

However, HEAP differs from GRID in two significant respects. First, the system measurements are made by *beacons themselves*, not by an external agent. This means it is applicable when physical measurements of localization error in the terrain are not possible. Second, the HEAP measurements are distributed, not centralized. Consequently, it is more scalable as the number of nodes in the network increases.

In the rest of this chapter, we describe both the network architecture underlying HEAP as well as the algorithms. In the next section, we describe the design of HEAP. We present extensive simulations of HEAP in Section 8.2 and experimental results in Section 8.3 to validate its benefits in a real deployed RF-based localization system. Finally, we conclude.

8.1 HEAP Design

The HEAP approach to incremental beacon placement is based on system measurements. In HEAP, the wireless network consists of three entities: *Node*, *Beacon* and a *Placer*. Beacons exchange neighborhood information with each other to determine suitable candidate points within their local neighborhood (for example, within a region

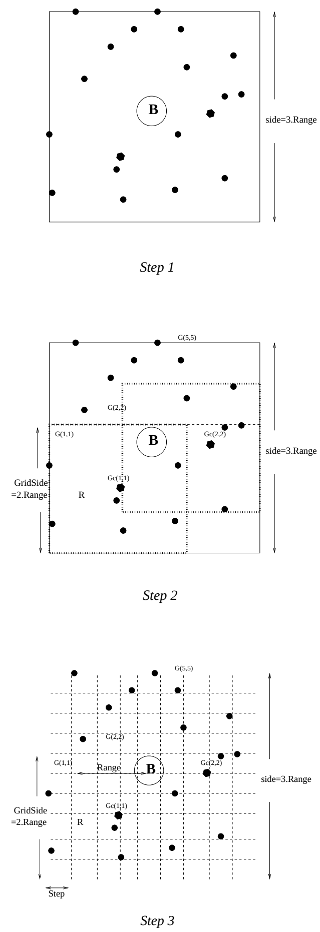


Figure 8.2: Illustration of the HEAP-GRID algorithm.

of radius r around the beacon) for adding new beacons. Because new beacons need to be physically deployed, a controlling agency is needed. Beacons send their candidate points to the Placer via intermediate nodes. The Placer deploys new beacons.

In systems where the Placer is fixed and located far from energy-constrained beacons, hop-by-hop communication rather than direct long range communication to the destination site is preferable for energy-efficiency. Furthermore, it is infeasible to transmit all data across the network, even hop-by-hop. By performing local computation to reduce data before transmission, orders of magnitude energy savings can be obtained [PK00]. Intermediate nodes aggregate and relay data from the beacons to the placer hop by hop.

Information flow in HEAP is illustrated in Figure 8.1. Data is transmitted from beacons to the placer, with in network aggregation at intermediate nodes.

HEAP measurement is a *distributed* algorithm that depends on *in-network processing* to select placement sites. The only centralized part is the placer. It is required only because we assume incremental node deployment from a single agency (we selected this definition for comparison with prior central algorithms such as Grid, described in Chapter 7). A fully distributed variation on the HEAP algorithm would allow a node to deploy additional beacons if improvement passed some threshold.

8.1.1 Algorithms

Information flow in HEAP can be set up using one of several data dissemination mechanisms proposed in the research literature [HCB00, IGE00]. Once this is in place, the three entities Beacon, Node and Placer execute their parts.

Beacon B : A beacon exchanges information and learns to estimate its beacon neighborhood. It then selects a candidate point and sends it to its $parent$ node.

Node N : An intermediate node in the hierarchy receives candidate points from all its neighbor beacons and child nodes. It selects and forwards one of these candidate points to its $parent$ node.

Placer P : The placer receives candidate points from all its neighboring beacons and child nodes. It eliminates any candidate points that do not satisfy constraints and selects good points for adding new beacons.

8.1.2 Neighborhood Estimation

Before a beacon in HEAP can select a candidate point, it needs to estimate its beacon neighborhood to the number of hops appropriate to its candidate point selection algorithm. This is accomplished by executing the algorithm $BeaconNeighborhood\ (\ B, NumHops\ )$ at Beacon B. In this algorithm, beacons include information about other beacons in their neighborhood of a certain scope in their advertisements. A beacon iterates $NumHops$ times, increasing its scope by 1 each time.

Algorithm $BeaconNeighborhood\ (\ B,\ NumHops\ )$

Input: B - A beacon.

$NumHops$ - The Number of hops (or the scope) to which neighborhood must be computed.

Output: A set of all beacons within $NumHops$ of beacon B.

Step 0 $NumPhases \leftarrow NumHops$

Step 1 $Phase \leftarrow 1$

Step 2 $Neighborhood\ (\ 0\ ) \leftarrow position\ (\ B\ )$

Step 3 while ($Phase \leq NumHops$) do

$BROADCAST\ (\ B,\ Phase - 1,\ Neighborhood\ (\ Phase - 1\ ))$

Listen to broadcasts of other beacons' $Phase - 1$ neighborhoods.

$Neighborhood\ (\ Phase\ )$ is the union of all the $Phase - 1$ neighborhoods heard during this period.

$Phase \leftarrow Phase + 1$

Step 4 Return $Neighborhood\ (\ NumHops\ )$

We consider HEAP-GRID, a simple algorithm for selecting candidate points, that extends the basic GRID algorithm proposed in [BHE01a]. We also experimented with HEAP-MAX (see [BHE01b]), the HEAP distributed algorithm with the MAX evaluation function in [BHE01a], but do not report on it here because the HEAP-GRID function gives better performance.

8.1.3 Candidate Point Selection

The HEAP-GRID algorithm for candidate point selection, learns the neighborhood of a beacon, but with a larger scope of 4 hops. This is illustrated in Figure 8.2. Beacon B determines candidates points in its neighborhood, in this case a square of side $3 \cdot Range$ based on the locations of its neighbor beacons. Its approach is to simulate the cumulative localization error over each grid for several uniformly separated points in its neighborhood. It divides the neighborhood into a few square grids, and picks the grid center with the highest error as a candidate point. This is based on the observation that adding a new beacon affects its nearby area, not just the point where it is placed.

Algorithm $HEAP - GRID(B)$

Input: B - A beacon.

Output: A candidate point where a new beacon could be added.

Step 0 $NeighborSet \leftarrow BeaconNeighborhood(B, 4)$

Step 1 $side \leftarrow 3 \cdot Range$

$Q \leftarrow position(B) = (X_B, Y_B)$

Consider a square S with side $side$ and center Q.

Step 2 Divide S into N_G partially overlapping grids as follows.

Step 2.1 $gridSide = 2 \cdot Range$

Let the side of each grid be $gridSide$.

Each grid encloses the radio reachability region of its center.

Step 2.2 For $1 \leq i, j \leq \sqrt{N_G}$, the grid $G(i, j)$ is defined by its center $G_c(i, j)$.

$$\begin{aligned}
G_c(i,j) &= (G_{CX}(i,j), G_{CY}(i,j)) where \\
G_{CX}(i,j) &= X_B - \frac{(side - gridSide)}{2} + \frac{(i-1) \times (side - gridSide)}{\sqrt{N_G} - 1} \\
G_{CY}(i,j) &= Y_B - \frac{(side - gridSide)}{2} + \frac{(j-1) \times (side - gridSide)}{\sqrt{N_G} - 1})
\end{aligned}$$

Step 3 For each grid $G(i, j)$, compute the cumulative localization error $CE(i, j)$ for the grid $G(i, j)$ as follows.

Step 3.1 Divide the grid into squares of size $step \times step$.

Step 3.2 $\forall\ 0 \leq k, l \leq (\frac{gridSide}{step})$,

let $P(k, l) = (P_X(k, l), P_Y(k, l))$ be the point in the region that corresponds to a square corner.

$$\begin{aligned}
P_X(k,l) &= G_{CX}(i,j) - gridSide/2 + k \cdot step \\
P_Y(k,l) &= G_{CY}(i,j) - gridSide/2 + l \cdot step
\end{aligned}$$

Step 3.3 Estimate localization error at each point $P(k, l)$ as follows.

Let χ be the set of all beacons in $NeighborSet$ that are within distance $Range$ of $P(k, l)$.

$$LE(P(k,l)) \leftarrow EstimateLocalizationError(P(k,l), \chi)$$

Step 3.4

$$CE(i,j) \leftarrow \Sigma_{k=0}^{k=\frac{gridSide}{step}} \Sigma_{l=0}^{l=\frac{gridSide}{step}} LE(P(k,l))$$

Step 4 Return $(G_c(p,q), S(p,q))$ of the grid $G(p,q)$ with maximum cumulative localization error as the selected candidate point.

Although HEAP-GRID is by no means the only possible algorithm, it is representative of the effectiveness attainable with a localized algorithm.

8.1.4 Error Estimation

One of the aspects of candidate point selection by a beacon is to estimate localization error at various points based on its knowledge of the beacon neighborhood. This error estimation is the domain-specific part of beacon placement, one can substitute the procedure below for connectivity based localization with other procedures.

Algorithm $EstimateLocalizationError(P, \chi)$

Input: P - A point in 2 dimensional space.

χ - A set of beacons within radio range $Range$ of point P.

Output: An estimate of localization error at point P.

Step 0 $\chi' \leftarrow \{(position(B),\ Range\ ) \mid B \in \chi\ \}$

Step 1 $P_{est} \leftarrow LocalizationFromConnectivity(\chi')$

Step 2 $\epsilon \leftarrow LocalizationError(P, P_{est})$

Step 3 Return ϵ.

8.2 Detailed Simulations

We have used simulations to explore, in some detail, the implications of several design choices in HEAP. In this section, we report on some results from these simulations.

8.2.1 Goals, Metrics and Methodology

Our goals in conducting this evaluation were two-fold: (i) Compare HEAP performance to a completely Random algorithm as well as to a centralized algorithm (GRID) with global knowledge of beacon positions and terrain or connectivity conditions. (ii) Understand the impact of noise caused by propagation losses and terrain features on the beacon placement algorithms.

We choose the same metrics to analyze the performance of our algorithms as those used in Chapter 7. These metrics are statistics evaluated by sampling the localization error at all $step \times step$ square corners obtained by subdividing the region.

Improvement in mean localization error M_1 computes the difference between mean localization error at all measured points in the terrain before and after the beacon node is added. This metric indicates the overall impact of adding a beacon to quality of localization in the entire terrain. For a given beacon placement $\mathcal{B}$

$$\textit{MeanErr}(\mathcal{B}) \quad = \quad \frac{\Sigma_{k=0}^{\frac{Side}{step}} \Sigma_{l=0}^{\frac{Side}{step}} LE_{\mathcal{B}}(P(k,l))}{(\frac{Side}{step}+1)^2}$$

where $P(k,l) = (k \cdot step, l \cdot step)$

$$M_1 \quad = \quad \textit{MeanErr}(\mathcal{B}_{init}) - \textit{MeanErr}(\mathcal{B}_{final})$$

Improvement in median error M_2 computes the difference between the median localization error at all the measured points in the terrain before and after the beacon node is added. This metric indicates the improvement due to adding a beacon on the quality of localization at the top 50% of the points with the highest localization error at the terrain.

$$M_2 \quad = \quad \textit{MedianErr}(\mathcal{B}_{init}) - \textit{MedianErr}(\mathcal{B}_{final}) \tag{8.1}$$

We study these metrics as a function of beacon density. In addition, we assume $step = 1m$.

Table 8.1: Terrain-influenced shadowing model parameters.

PARAMETER	DEFINITION	VALUE
λ	Wavelength	0.333m
β_1	Path loss exponent(unobstructed)	2
β_2	Path loss exponent (obstructed)	4
σ_{dB}	Standard deviation of noise	5
P_t	Transmitted Power	660mW

To understand how HEAP copes with noisy radio propagation, we evaluated HEAP for both (i) ideal radio propagation conditions and (ii) a terrain based shadowing model (uses a bitmap of the terrain). We ported the latter from Arena/ns [YVS01] to our simulations. The experiments were carried out in a simulated square terrain of side 100m. From Figure 8.4 we can see that the environment contains both obstructions and good terrain, so the terrain based propagation model is quite appropriate. The various propagation model parameters we chose is summarized in Table 8.1. The values of β and σ_{dB} are chosen from the ranges of their typical values [Rap96]. The terrain-based shadowing model has different values of β for line of sight and non line of sight respectively. P_t, the transmit power is selected from [Kai00] and P_{thresh}, the receiving threshold is set to be the receive power at the nominal radio range $Range$ using Friis free space model [Rap96]. These do not necessarily reflect the details of real environment but are representative of a range of environments in which our algorithms may be used.

8.2.2 Impact of Beacon Density

To compare the performance of HEAP, our localized algorithm for various beacon densities with GRID, centralized measurement based algorithm described in [BHE01a],

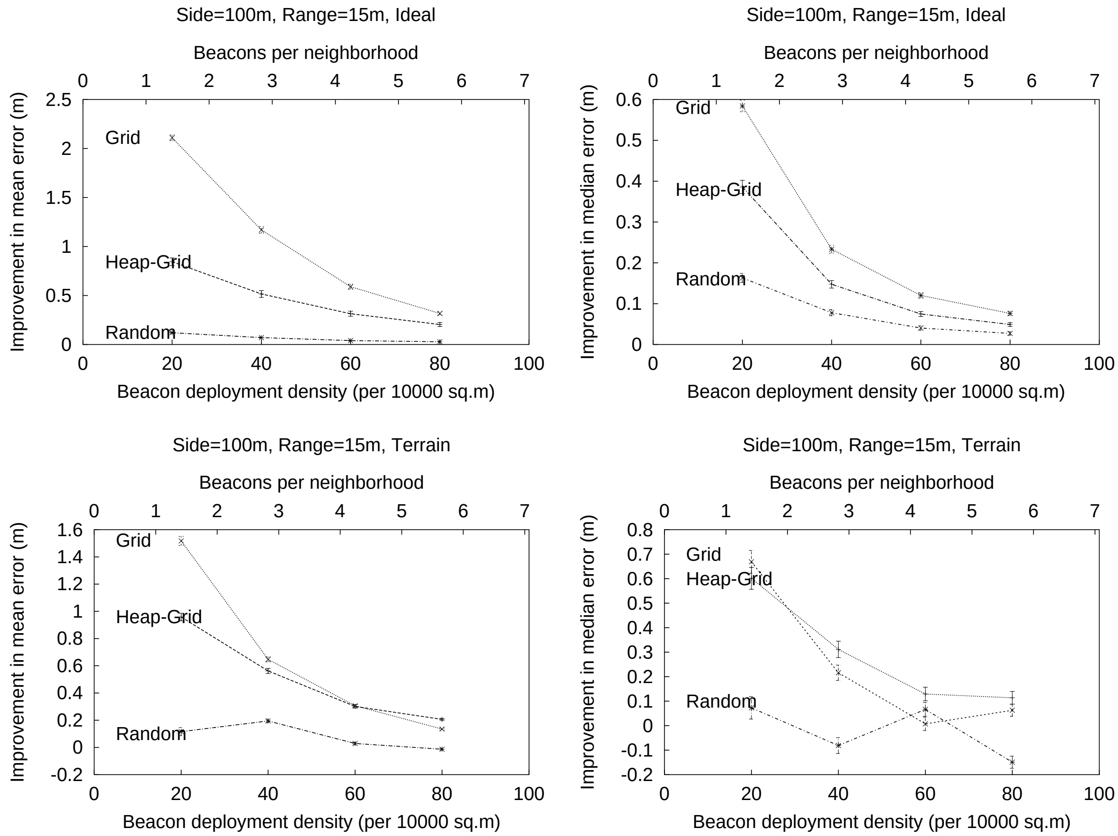


Figure 8.3: Performance comparison of HEAP with centralized algorithms for the mean and median localization granularity metrics.

we conducted the following simulation experiment. We varied the number of beacons, N from 20 to 80 in increments of 20. The nominal radio transmission range of a beacon $R = 15m$. Correspondingly, μ, the number of beacons per nominal radio coverage area ($bpnrca$) varies from 1.41 to 5.64. We generated 1000 different beacon fields per beacon density. Each beacon field is generated by randomly placing the beacons in the $100m \times 100m$ square terrain. Performance metrics for each algorithm and beacon density are averaged over the 1000 beacon fields. To characterize the stability of our results, all graphs include 95 percentile confidence intervals.

Figure 8.3 plots the improvements in the mean and median localization errors as a

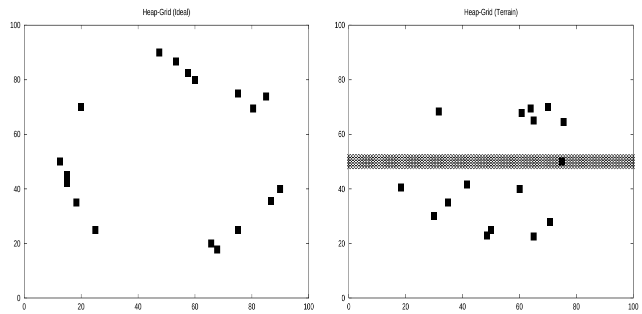


Figure 8.4: HEAP Candidate Point Selection. Ideal case vs. terrain with wall.

function of beacon deployment density for both ideal radio propagation model and the terrain based shadowing model.

With ideal radio propagation, both algorithms perform well for low densities ($< 3\ bpnrca$), but GRID has the potential for significant improvements. For all the three algorithms, the metric M_1 (improvement in mean localization error) decreases rapidly for densities $\geq 3\ bpnrca$, and saturates for densities $\geq 6\ bpnrca$. The gain in median localization error (metric M_2), for GRID relative to HEAP-GRID is considerably lower than metric M_1. Because HEAP-GRID selects candidate points only in the local neighborhood, it is unlikely to identify noisy points as well as the centralized algorithms. Its worst case improvement, and consequently, mean improvement M_1 tends to be much smaller.

The trend exhibited by metric M_2 for ideal radio propagation is further exemplified for the terrain-influenced shadowing model for radio propagation, as Figure 8.3 shows. In the terrain case, for low densities, the total number of noisy points far exceeds the ideal case. GRID which instruments the whole terrain leverages this and posts higher gains in mean error by substantially improving a large number of bad points. HEAP-GRID focuses on moderately bad points and therefore improvement is relatively lower. The median error improvements for the terrain case for HEAP are also much better for higher densities.

Although the gain for HEAP does not equal the centralized algorithm, both are comparable. Moreover, HEAP is distributed and therefore much more scalable.

8.2.3 Impact of Terrain Features

To qualitatively evaluate the effectiveness of HEAP in selecting good candidate points in a noisy terrain, we conducted a second simulation experiment wherein initial beacon placement is always uniform, varying the number of beacons N and the transmission range R. N = 25, 36, 49, 64, 81 and 100. R = 15m, 20m, and 25m. In each case, HEAP is run to determine the candidate points for two scenarios (a) an ideal terrain with no obstacles and (b) a terrain with a wall in the middle shown in Figure 8.4. In Figure 8.4, each point represents a new placed beacon from one simulation run. Candidate points shown are those selected in 20 runs of HEAP with uniform placement for both the ideal case and for a terrain with a wall in the middle. The right plot adds a wall (shown in grey) as terrain. A simple boundary constraint is applied to remove algorithm bias towards candidate points at the corners of the terrain. Candidate points shift closer to the center when there is a wall in the middle.

In the ideal case, HEAP-GRID selects candidate points closer to the periphery of the region enclosed by the boundary constraint. This is because it selects the center of the grid with the highest cumulative localization error, and in the ideal case such grids are more likely to be located at the edges of the terrain (even with uniform beacon placement and the boundary constraint). For the terrain, the candidate points shift closer to the center near the wall. The actual points selected depend on the beacon density, range and positions of the beacons relative to the wall.

Despite having to deal with erroneous information (poor neighborhood approximation, idealized radio model etc.), the HEAP algorithms are able to select candidate points closer to a terrain feature such as a wall. However, such a result may not be

Figure 8.5: Beacon deployment in the UCLA LECS Laboratory.

valid for very small terrain objects, such as foliage.

8.3 Experimental Results

We also evaluated HEAP in a real testbed deployment to verify its effectiveness. We deployed a localization system consisting of 16 beacons in an indoor environment, the Laboratory for Embedded Collaborative Systems (LECS) at UCLA (see Figure 8.5). Beacon motes are attached to the ceiling tiles of a lab with partitions and open space. The configuration in which beacons are placed is shown in Figure 8.6. Each + sign indicates a beacon. 16 beacons are uniformly located in a 24×24 feet square region, with adjacent beacons 8 feet apart. We chose an indoor setting for this experiment because the radio propagation is not ideal indoors due to multi-path effects, and therefore it provides us an interesting test case to study how well HEAP helps the system adapt to its environmental conditions.

We varied the transmission power and frequency settings using software-enabled control commands (see Section 4.4). For each unique setting, we collected the following data:

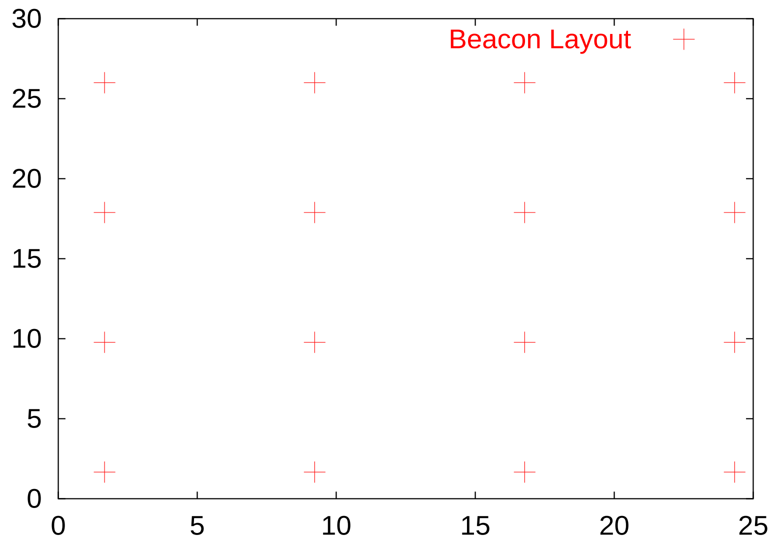


Figure 8.6: The configuration in which beacons are placed in the LECS laboratory.

Table 8.2: Control parameters for the beacon system.

PARAMETER	VALUE
Transmission Power/Potentiometer Setting	75
Beaconing Interval (seconds)	3

- *Beacon Connectivity Measurements:* Each beacon measures its connectivity to other beacons. We obtain the beacon network topology from the connectivity measurements of all beacons.

- *Localization Error Measurements:* We measure localization error at various points in the terrain by walking across the room and collecting data at spacings of 2 feet. For each point, the localization error is averaged over several trials.

We found real experiments to be very valuable. We observed the following for our experiment with 16 beacons:

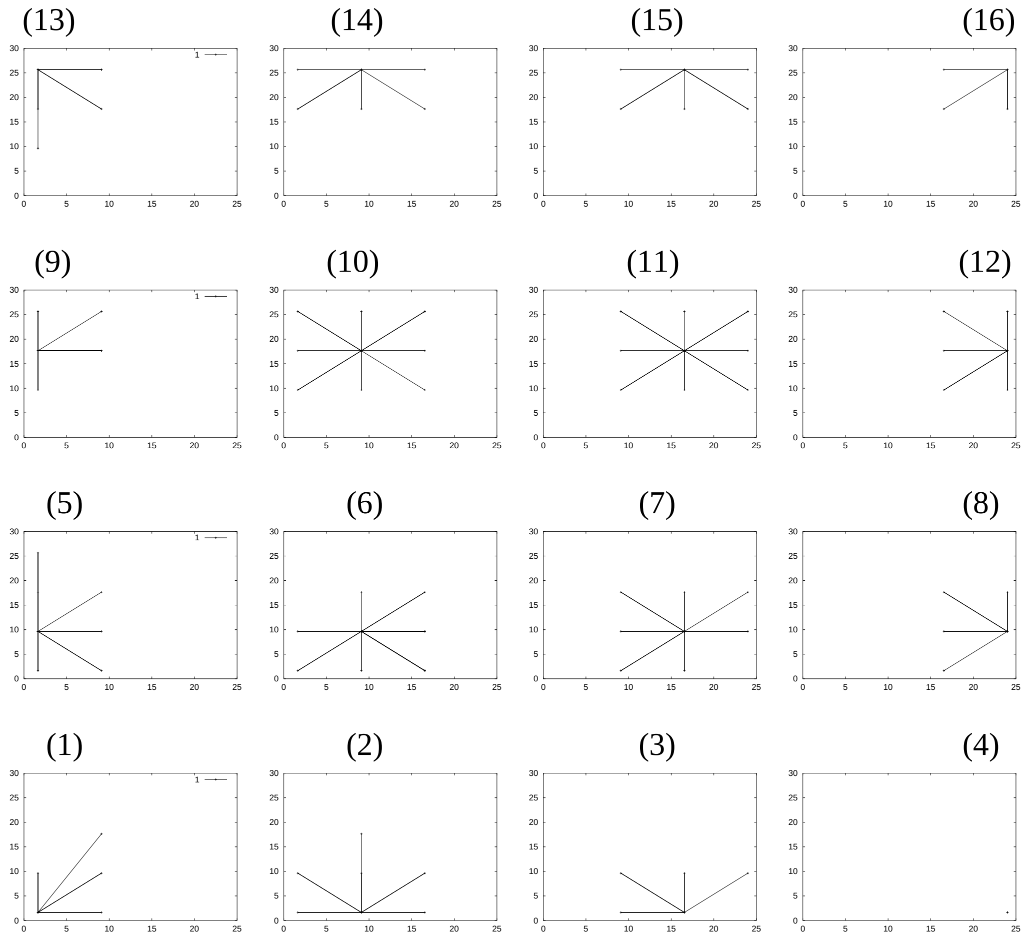


Figure 8.7: Beacon connectivity graph obtained in our experiment.

- At the same physical point, the localization estimate varied over time.
- The connectivity relation between two beacons varies over time.

Because candidate point selection in HEAP is based on beacon connectivity relations, the connectivity graph provides us complete information to emulate the HEAP algorithm. We physically deployed new beacons at candidate points selected by HEAP and recomputed the localization error at various points in the terrain.

Table 8.2 refers to the control settings used for the experiment whose results are shown in Figures 8.7 and 8.8.

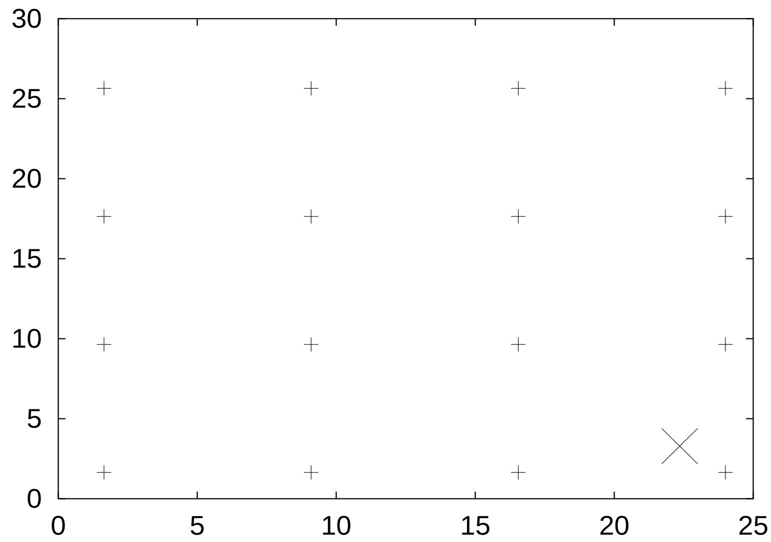


Figure 8.8: Candidate point selected by HEAP.

Figure 8.7 plots the Beacon Connectivity Graph. Connectivity of each beacon is shown in a separate graph and beacons are numbered. Connectivity between beacons is sometimes asymmetric (as in beacons 1 and 6), and some beacons have a greater connectivity degree than others (compare 5 with 9). The corner beacon 4 has no connectivity. We can see that it is asymmetric, some beacons have greater connectivity than the others.This connectivity graph was used to emulate the HEAP algorithm.

Figure 8.8 displays the candidate point selected by HEAP to add a new beacon. The plus (+) sign indicate positions of beacons. The cross (X) sign indicates the position of the candidate point selected by the HEAP algorithm. We can see that the candidate point is very close to the position of the failed beacon (beacon 4 in the lower right corner in Figure 8.7).

We physically added a new beacon at the candidate point. Figure 8.9 plots the cumulative distribution function of the localization error before and after the new beacon was added at the candidate point selected by the HEAP algorithm. While the median

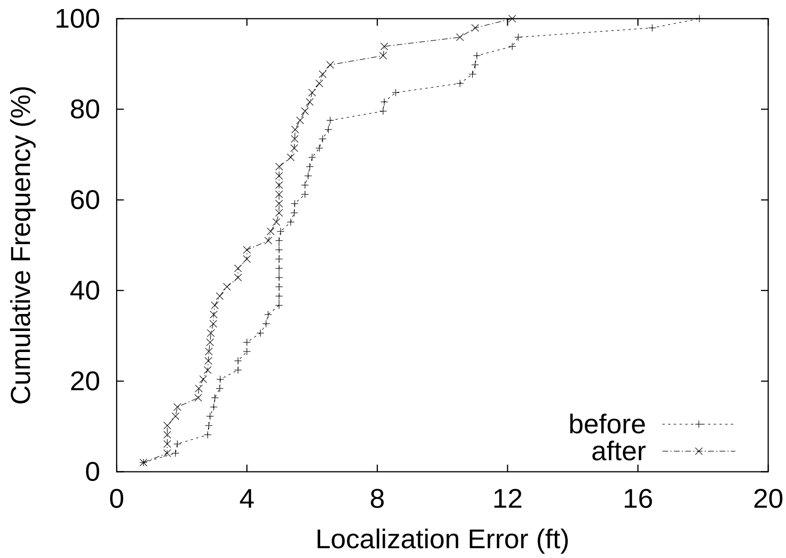


Figure 8.9: Cumulative distribution function (CDF) of localization error (experiment).

error remains the same, there is significant improvement in the 90%ile error (drops almost 50% from 11 feet to 6 feet). This shows us that HEAP can be effective in a real environment.

8.4 Discussion

From our design and evaluation of HEAP, we can draw the following general lessons.

1. Our simulations show that localized and adaptive algorithms such as HEAP are effective in comparison to centralized adaptive algorithms such as GRID in addressing beacon placement. This is because the relevance of information needed by a beacon for algorithmic computation drops as a function of distance or number of hops to the beacon.

2. Our experimental results show that HEAP can benefit a real deployed localization system. This proves that adaptive self-configuration to terrain and envi-

ronment characteristics based only on local coordination among beacons and without a system or terrain model is both feasible and worthwhile.

3. Proximity-based localization has a beacon density beyond which the benefit of additional beacons falls off. This observation suggests the STROBE algorithm targeting high beacon densities, evaluated in the next chapter. More generally, the study of performance as a function of density is important for algorithms involving many nodes.

8.5 Summary

In this chapter, we presented HEAP, our low-complexity algorithm for self-configuration at low and medium beacon densities. HEAP uses the design principle of *localized algorithms* set forth in Chapter 3.

We presented detailed simulations to show that HEAP can achieve results comparable to centralized adaptive algorithms. Finally, we presented experimental results that demonstrated the benefits of HEAP in a real deployed localization system. HEAP is a general framework to select candidate points for adding new beacons. The only aspect of HEAP that is domain specific is the error estimation function described in Section 8.1.4. HEAP could be applied to other localization systems which use beacons.

CHAPTER 9

STROBE: Selectively TuRning Off BEacons

Sleep is the interest we have to pay on the capital which is called in at death; and the higher the rate of interest and the more regularly it is paid, the further the date of redemption is postponed.

— Arthur Schopenhauer

In this chapter, we describe STROBE, an algorithm for rotating functionality in densely deployed beacon networks in order to enable coordination amongst beacons without interference and extend overall system lifetime. We motivate our choice of density-adaptive protocols as a building block, especially in the context of beacon networks for localization in Section 9.1. We present the design of STROBE in Section 9.3 and an analysis of its energy usage in Section 9.4. We present the evaluation of STROBE using simulations and experiment in Sections 9.5 and 9.6 respectively. We present our concluding remarks in Section 9.8.

9.1 Motivation

A key requirement for large scale sensor networks is robust, unattended operation. Here it may not be feasible to improve localization by adding new beacons at empirically determined points, as with HEAP. Instead, we would begin with a very dense beacon deployment initially, and then rotate functionality amongst beacons (by turning them on and off) to maximize lifetime. Note that the *economies of scale* involved in massive deployment provide a sound incentive for deploying beacons very densely initially, instead of deploying a few beacons initially and then replacing them or recharging them whenever they run out of energy. If nodes are cheap enough, predeployment reduces later administrative costs.

The acronym STROBE stands for Selectively TuRning Off BEacons. The goal of the STROBE algorithm is for beacons to cooperatively achieve an adaptive operational density without diminishing the localization granularity.

For beacon deployment densities μ_{actual} much greater than the saturation threshold μ_{thresh}, tuning the operational beacon density can provide several advantages without diminishing localization quality. First, the duty cycles of individual beacons can be reduced without diminishing localization granularity, thus increasing system lifetime. Second, with fewer operational beacons at any instant, the overall number of beacon transmissions are reduced, thereby reducing the probability of self-interference amongst beacons 6.3. Finally, a higher percentage of beacons could remain active in noisier obstructed parts of the terrain, whereas a smaller percentage of beacons may need to be active in unobstructed terrain, achieving similar localization granularity and the adaptive self-configuration that motivates this work.

9.2 Design Considerations

We have made some assumptions in the design of STROBE. We state these assumptions and discuss their implications below.

- Beacons are static and compute their position only once. Therefore, we can ignore both the computational and communication energy for continuous position estimation of beacons (for example, GPS acquisition overhead).

- Clients may be mobile and need to update their positions continuously. Therefore, beacons need to remain active throughout the system lifetime. [1] This motivates the need for a continuous or periodically adaptive algorithm like STROBE.

- The interval between successive beacon transmissions remains fixed during the system lifetime. While this is not inherently necessary, it considerably simplifies our design and analysis.

STROBE must accomplish several goals. First, it must maintain *uniform localization granularity* both across the system and over time. Second, it must maximize *system lifetime* both by minimizing the energy usage at each beacon as well as by load balancing energy usage across the beacons. Third, it must minimize *convergence time* of the beacon infrastructure from an initial state where all the beacons are active to an energy-efficient state, where only the threshold level of beacons needed to maintain the desired localization granularity are active. Finally, after convergence to a steady state, the system should not deviate significantly from it.

9.3 STROBE Design

In this section, we discuss the design of STROBE. We discuss the duty cycle of a beacon in STROBE, and the decision making approach that governs the state transitions in this duty cycle.

9.3.1 STROBE Duty Cycle

Typically, each beacon transmits one position advertisement in a beaconing interval T_B and sleeps for the remainder of the interval. Each position advertisement has four

[1] If the ratio of clients to beacons is very small, then these systems could be triggered.

Beacon Identifier	Beacon Position	Packet Sequence Number	Beacon Status

Figure 9.1: Beacon Position Advertisement Packet Format for STROBE.

fields: beacon identifier, beacon position, sequence number, beacon status. Beacon status is usually set to be UP (see Figure 9.1).

In STROBE, a beacon can be in one of three states: *Voting (V), Designated (D)* and *Sleep (SL)*. The state transition diagram is depicted in Figure 9.2. Beacons can switch from *Voting* to *Designated* or *Sleep* states and vice versa. All beacons start out in the *Voting* state, wherein, a beacon turns on its radio and broadcasts position advertisements every T_B seconds and also listens for advertisements from its neighboring beacons. When a beacon node enters *Voting* state, it sets a timer for T_V seconds. When the timer fires, it evaluates where it should go to sleep based on a decision making process explained in Section 9.3.2. If so, it broadcasts an advertisement with State set to be DOWN and transitions to the *Sleep* state. Otherwise, it transitions to the *Designated* state. A beacon node in sleep state wakes up after a sleep time T_{SL} and transitions back to *Voting* state. A beacon node in *Designated* state periodically advertises at intervals T_B for a time T_D and then transitions back to *Voting* state. A beacon node in *Sleep* state wakes up after a sleep time T_{SL} and transitions back to*Voting* state.

Distinct *Voting* and *Designated* states are necessary in order to avoid the overhead incurred due to receiving advertisement messages from other neighbor beacons when in the Voting state. Three important parameters of STROBE that influence its energy usage and system lifetime are T_V, T_D, and T_{SL}.

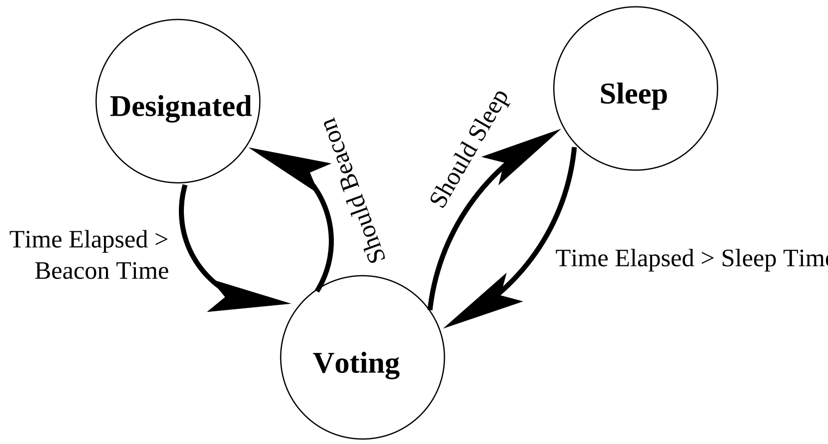


Figure 9.2: State Transition Diagram for STROBE.

9.3.2 Beacon Decision Making

During the *Voting* state, a beacon evaluates ζ, the number of currently active beacons that are its neighbors.

$$\zeta \;=\; |B_{up} - B_{down}| \tag{9.1}$$

where B_{up} is the set of all beacons it heard from whose most recent advertised state is UP and B_{down} those whose most recent advertised state is DOWN.

This means that the number of active beacons in its neighborhood, including itself is $\zeta + 1$. Let μ_{thresh} be the threshold number of beacons in any given neighborhood at which the localization granularity saturates.

If $(\zeta + 1) \leq \mu_{thresh}$, then it has to remain active. If $(\zeta + 1) > \mu_{thresh}$, then its transition probability p to the *Designated* state is given by:

$$p \;=\; \frac{\mu_{thresh}}{\zeta + 1} \tag{9.2}$$

With probability $(1 - p)$ it transitions to the *Sleep* state.

Probability Analysis

Suppose we have $n = \mu_{actual}$ beacons in some area. Only $k = \mu_{thresh}$ need to perform a given task, the rest can go to sleep.

Let each node independently decide to participate in the task with probability p. What should the probability p of a node participating in a given task be?

Let X be the random variable that indicates how many beacons actually participate in the task. The probability distribution function of X is the simple binomial distribution:

$$Pr[X = i] = \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i}$$

Probability that the task is accomplished is $Pr[X \geq k]$.

$$Pr[X \geq k] = 1 - \sum_{i=1}^{k-1} \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i}$$

This equation gives a phase transition at $p = k/n$ [KBW02]. Thus, the actual probability of state transition should be set to slightly higher than k/n.

Note that this is a very simple decision making approach, influenced only by the number of currently active neighbors ζ. It is *memoryless* [Pap91] — state transitions depend only on the current state are not governed by a history of previous state transitions. This makes it really simple to implement and analyze.

More sophisticated approaches could incorporate information such as energy reserve of a beacon and its neighbors, as well as bias a beacon's current estimate of ζ based on a previous history of measurements. However this would require beacons to maintain some additional state, which increases complexity.

Table 9.1: Terminology used in energy analysis of STROBE.

TERM	DEFINITION
P_X	Transmit power of a beacon's radio transceiver
P_R	Receive power
P_I	Idle power
P_S	Sleep power
T_B	Beaconing interval
T_X	Transmit time of a beacon advertisement
Φ	Maximum (Initial) Energy of a beacon node
μ_{thresh}	Threshold beacons per nominal radio coverage area
μ_{actual}	Actual mean beacons per nominal radio coverage area

9.4 Energy Analysis

As we stated in Section 3.4, localized algorithms such as STROBE are sensitive to choice of parameters. To better understand the influence of such parameter choices and to characterize the performance of STROBE, we present a simple model and analysis of energy usage.

Our energy model characterizes only the energy usage of the radio transceiver only and does not explicitly model processor energy. There is a reason for this: Typical processing costs are much lower than communication costs [PK00]. Additionally, transmission of beacon advertisements is not a compute intensive activity.

Table 9.4 summarizes the terminology we use in our energy analysis.

9.4.1 Simple Beaconing

In simple beaconing, each beacon transmits one advertisement in a beaconing interval T_B and sleeps for the remaining part of the interval. Energy consumed by a beacon node per beaconing interval:

$$E_B = P_X \cdot T_X + P_S \cdot (T_B - T_X) \tag{9.3}$$

Power dissipated by a beacon node per beaconing interval:

$$P_B = \frac{E_B}{T_B} \tag{9.4}$$

Lifetime of a beacon node with simple beaconing:

$$L_B = \frac{\Phi}{P_B} \tag{9.5}$$

We observe that the lifetime of any adaptive operational density scheme can never exceed $(\frac{\mu_{actual}}{\mu_{thresh}})L_B$.

We observe that in a realistic engineering design, we would try to keep T_B, the beaconing interval as high as possible. Even when beacons are densely deployed, they will not be deployed at a factor several times higher than μ_{thresh}, so as to minimize costs. The proportion of P_I, P_R, and P_X depends on the specifics of the radio considered. For radios such as the WINS-NG transceiver [Kai00], this ratio is approximately 1:10:20, for WaveLan radios this is measured as 1:1.05:1.6 [SK97].

9.4.2 STROBE

As discussed, a beacon can be in any of three different states in STROBE.

Let d be the mean degree of a beacon node *i.e.*, the number of active neighbors from whom it receives advertisements during the Voting cycle.

The overhead incurred in the *Voting* state is substantial compared to simple beaconing. This means that a node must transition from the *Voting* state to simple beaconing *Designated*, which also justifies the use of 3 states in STROBE.

Without loss of generality, we assume that T_V and T_{SL} are integral multiples of T_B. Thus, the energy consumption in the three states for STROBE is given by:

$$\begin{aligned}
E_V &= \frac{T_V}{T_B} \cdot (P_X \cdot T_X + d \cdot P_R \cdot T_X + P_I \cdot (T_B - (d+1) \cdot T_X)) \\
E_{SL} &= T_{SL} \cdot P_S \\
E_D &= P_D
\end{aligned}$$

The power consumption in the three states of STROBE is given by:

$$\begin{aligned}
P_V &= \frac{(P_X \cdot T_X + d \cdot P_R \cdot T_X + P_I \cdot (T_B - (d+1) \cdot T_X))}{T_B} \\
P_{SL} &= P_S \\
P_D &= P_B
\end{aligned}$$

Let t_V and t_{SL} be the time spent by the beacons in the V and SL states respectively.

Thus, the lifetime of a beacon node in STROBE:

$$\begin{aligned}
L_{STROBE} &= t_V + t_D + t_{SL} \\
\Phi &= t_V \cdot P_V + t_D \cdot P_D + t_{SL} \cdot P_{SL}
\end{aligned}$$

Additionally, from the state transition diagram of STROBE we conclude

$$t_V = T_V \cdot \left(\frac{t_{SL}}{T_V + T_{SL}} + \frac{t_D}{T_V + T_D} \right)$$

Ideally in a system, each node listens and sleeps for the same proportion of time as all the other nodes. Assuming μ_{thresh} beacons per neighborhood are active at any point of time,

$$\frac{t_{SL}}{t_V + t_D} = \frac{\mu_{actual} - \mu_{thresh}}{\mu_{thresh}} \tag{9.6}$$

Substituting for t_{SL} from Eqn. 9.6 in Eqn. 9.6 and setting $T_{SL} = T_D$ for good load balancing, we get

$$\frac{t_V}{t_V + t_D} = \frac{\mu_{actual}}{\mu_{thresh} \cdot \left(2 + \frac{T_D}{T_V}\right)} \quad (9.7)$$

This implies that the best case lifetime of a beacon node in STROBE,

$$\mathcal{L}_{STROBE} = \frac{\mu_{actual}\Phi}{\mu_{thresh}\left(\frac{P_V \cdot t_V + P_D \cdot t_D}{t_V + t_D}\right)}$$

$$\mathcal{L}_{STROBE} = \frac{\mu_{actual} L_B}{\mu_{actual}\frac{\frac{P_V}{P_D} - 1}{2 + \frac{T_D}{T_V}} + \mu_{thresh}}$$

Our analysis of the above equation gives us the insight that T_D = T_{SL} should be set very high compared to T_V. However this disguises one simple fact, setting the ratio T_D/T_V very high may not load balance energy very well.

9.5 Detailed Simulations

We have conducted extensive evaluations of STROBE using simulations. In this section, we discuss our findings.

9.5.1 Goals, Metrics and Methodology

Our goals in evaluating STROBE using simulations are to answer the following questions:

- Is STROBE effective?
- How do various parameters affect its performance?
- How well does STROBE perform compared to the optimal case?

We use several metrics in our evaluation. We study the following metrics as a function of time.

- *% Beacons Active - $P_{active}(t)$:* Percentage of total beacons that are in either *Voting* or *Designated (D)* states at any given instant of time.
- *% Beacons Alive - $P_{alive}(t)$:* Percentage of total beacons that possess energy reserves greater than zero at any given instant of time.
- *Median localization error of the terrain - $MedErr(t)$:* at any given instant is calculated as follows. Divide the terrain into squares of size $1m \times 1m$. Consider all the points in the terrain that correspond to corners of the square. Compute localization estimates at these points based on beacons active at that instant and the corresponding localization errors. The median of these localization errors is approximated to be the median localization error in the terrain.

We use two other metrics.

- *First node death:* Time elapsed since the start before any single node in the terrain runs out of energy.
- *System lifetime:* Time elapsed since the start before the median localization error exceeds a n operational error threshold (for example, $0.4 \cdot Range$).

For our simulations, we choose an energy consumption model to mimic realistic sensor radios [Kai00]. These parameters are also used in [IGE00] and are summarized in Table 9.2.

Table 9.2: Energy consumption parameters used in STROBE evaluation.

POWER DISSIPATION	RADIO OPERATION MODE	VALUE
P_X	Transmit	660 mW
P_R	Receive	395 mW
P_I	Idle	35 mW
P_S	Sleep	0 mW

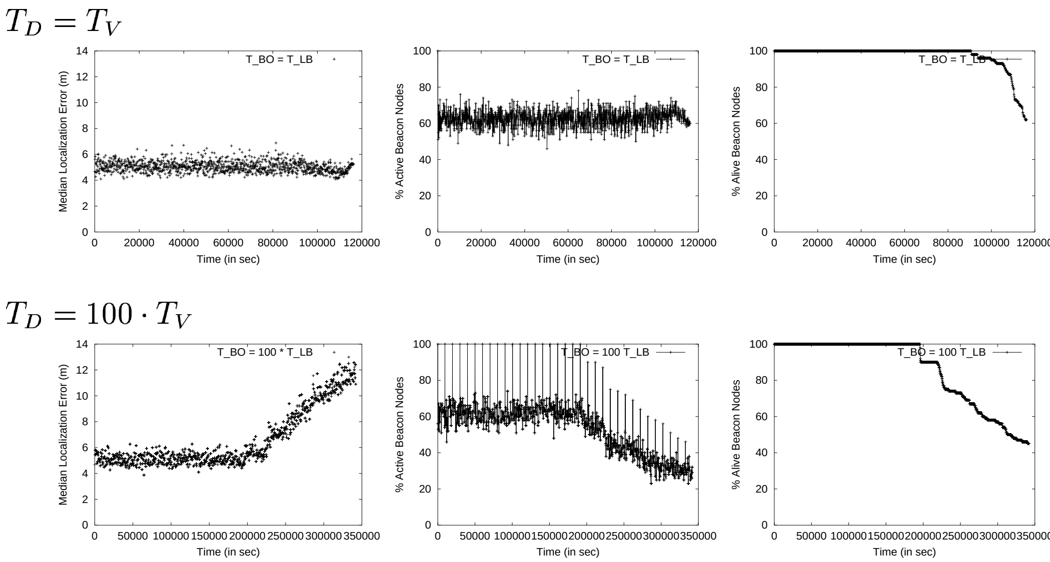


Figure 9.3: STROBE performance for two ratios of $\frac{T_D}{T_V}$.

9.5.2 Sensitivity to STROBE Parameters

To study the sensitivity of STROBE performance to its parameters, especially the rate of adaptation, $\frac{T_V}{T_D}$, we simulated a terrain of area $100m \times 100m$ with 100 randomly placed beacons in the terrain. The nominal radio range is $20m$. Thus the number of beacons per nominal radio coverage area is around 12 ($\mu_{actual} = 12$, $\mu_{thresh} = 6$). Each node has a starting energy of 10000J. Transmit time (T_X) of a beacon advertisement is 0.025 seconds. Beaconing interval T_B is set to be 1 second. T_V is set to be 5 seconds and T_D is varied to be T_V and $100T_V$.

Figure 9.3 compares the performance of the STROBE algorithm for various ratios of $\frac{T_D}{T_V}$ with respect to these metrics: median localization error, percentage of active beacons, percentage of beacons alive at nodes. The simulation terminates when none of the nodes has sufficient energy to either transmit or receive packets.

The top row corresponds to the ration $\frac{T_D}{T_V} = 1$ and the bottom row corresponds to $\frac{T_D}{T_V} = 100$. The simulation parameters are $R = 20$m, $N = 100$, T_B = 1s, $T_V = 5T_B$, Φ = 10000J, and the snapshot period is 100s.

Increasing the ratio $\frac{T_D}{T_V}$ improves the system lifetime. For instance, the first node deaths occur at 90000 seconds and 200000 seconds respectively for values of $\frac{T_D}{T_V}$ set to 1 and 100. It also improves the time duration between the first node death t_F and the last node death t_L. In addition it also minimizes the variations in median error over small periods of time.

The median localization error over time is closely correlated to the percentage of beacons alive. The step wise degradation (*i.e.*, increase) in the median localization error after the first node death mirrors the step wise decrease in the percentage of beacons alive over time. A closer inspection of the terrain snapshots over time reveals that because beacons are distributed uniformly in limited-size terrain, we see boundary conditions at the edges. For boundary beacons, the observed neighborhood size is either close to or less than μ_{thresh}, therefore they all tend to remain active and die first at approximately the same time. The next phase occurs when the next set of beacons that die are the ones that were adjoining the previous boundary beacons and are now the new boundary beacons, leading to a cascading failure of nodes.

9.5.3 STROBE Benefits

Our second simulation experiment demonstrates STROBE benefits for an applicable context (small beaconing interval, high beacon density). We simulate a terrain with

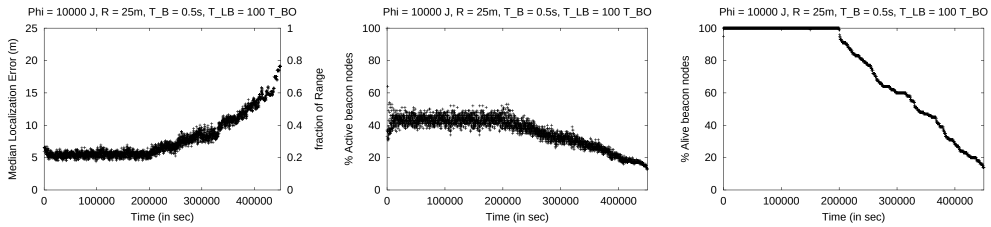


Figure 9.4: STROBE performance for N=100, R=25m, $T_B = 0.5s$, $T_V = 2T_B$, $T_D = 100T_V$, Φ=10000J.

100 beacons distributed uniformly at random in a 100 m $\times 100$ m terrain. The nominal radio range of these beacons is 25m. The corresponding beacons per neighborhood $\mu_{actual} = 19 = 3.1\mu_{thresh}$. We choose a reasonably small beaconing interval, $T_B = 0.5$ seconds. We set the various STROBE parameters as follows: $T_B = 0.5s$, $T_V = 2T_B$, $T_D = 100T_V$, Φ=10000J. The lifetime of a beacon using simple beaconing L_B is

$$L_B = \frac{\Phi}{T_B} \tag{9.8}$$

In this case, $L_B = 300000s$.

The best case system lifetime in STROBE from our previously described energy usage analysis

$$\mathcal{L}_{STROBE} = \frac{\mu_{actual} L_B}{\mu_{actual}\frac{\frac{P_V}{P_D}-1}{2+\frac{T_D}{T_V}} + \mu_{thresh}} \tag{9.9}$$

where μ_{actual} is the actual number of beacons per neighborhood, μ_{thresh} is the threshold number of beacons per neighborhood for localization, P_V and P_D are the mean power dissipated in the *Voting* and *Designated* states respectively.

Figure 9.4 plots the median localization error, percentage of active beacons and percentage of beacons alive as a function of time. Snapshots are taken every 100 seconds. The degradation in median localization error as well as percentage of beacons alive over time is considerably smoother than in our previous simulation experiment. Lifetime of the algorithm using simple beaconing $L_B = 300000s$. In this case, STROBE maintains a median localization error within $0.2 \times Range$ for up to 200000 seconds, $0.3 \times Range$ for up to 300000 seconds, and $0.5 \times Range$ for up to 400000 seconds. Actual system lifetime (L_{STROBE}) is increased to around 450000 seconds or $1.5 L_B$. This is low compared to the best case lifetime predicted by our model substituting $\frac{T_D}{T_V} = 100$ of 850000 seconds $2.8 L_B$. That calculation assumes energy usage can be load balanced effectively across beacons and that beacons are uniformly distributed in the terrain. However, as we have seen boundary nodes tend to die first, causing a cascading effect. To improve further on these lifetimes, beacons could perform edge detection to identify boundary conditions and adjust their beaconing period T_B to be higher compared to other beacons. Alternatively, a higher density of beacons could be deployed near the boundary.

STROBE transitions probabilistically from *Voting* to *Sleep* states, causing a higher percentage of beacons than the threshold percentage to remain active. Leveraging auxiliary information may significantly improve this lifetime.

9.5.4 Summary of Simulation Results

For densely-deployed beacon systems (density above the threshold density), our example shows that a completely localized algorithm like STROBE can extend the system lifetime 1.5 times without diminishing localization granularity with 3.1 times saturation density of nodes. Lifetime gains can be improved further for higher beacon densities and energy dissipation rates in active state, and by augmenting STROBE with boundary detection mechanisms.

9.6 Experimental Results

We have also evaluated STROBE experimentally. This is slightly harder to do because we have to measure both the energy depletion at different nodes over time and the degradation of localization quality at various points across the terrain and over time (which requires manual intervention and is therefore not feasible at a very fine grained time scale). Instead, the methodology we used was experimental emulation.

- We collect real beacon connectivity data and play back this connectivity data in a custom simulator to emulate the beacons' decision making process in STROBE. In modeling the behavior of a localization system, radio propagation is the hardest to model well, and hence it is important to verify it using real data.

- We simulate power consumption over time using a radio energy model. Since radio communication (as opposed to computation) dominates the power consumption of these nodes, this provides us with a good approximation of energy usage. Moreover, by using the same energy consumption model as our simulation, we can also validate the simulation.

- We emulate localization error in our connectivity based localization method using the connectivity data. This allows us to analyze the degradation in localization quality at a very fine-grained time scale.

Figure 9.5 plots the median localization error as a function of time (for both the experiment and simulation). We notice that the system lifetime with experimental emulation is comparable to idealized simulation, but the quality of localization is only slightly worse (20%). Thus, our idealized simulations can be considered a good indicator of STROBE performance. The localization quality is slightly worse in the experimental case because we did not account for link asymmetry in initial design of

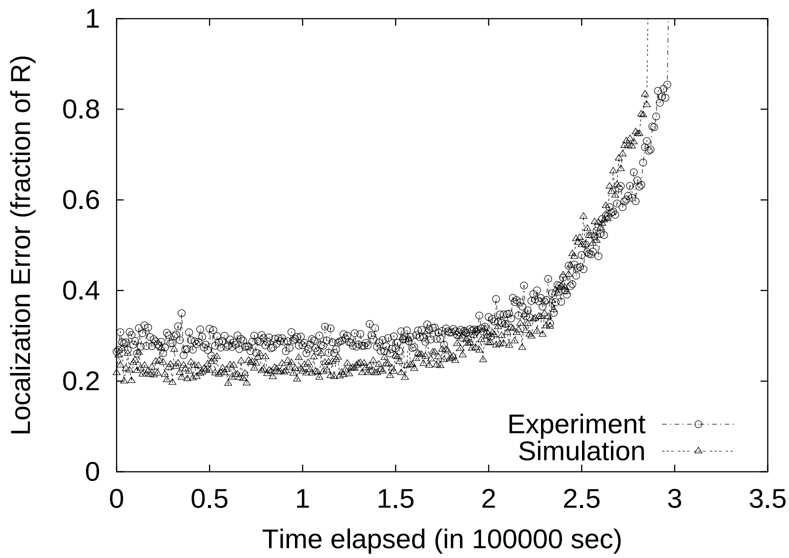


Figure 9.5: Median localization error vs. time. Comparing experimental emulation with the simulation.

the STROBE decision making. Beacons can turn themselves off even when neighbor is a stray far-away beacon. To avoid this, we may need a geographic filtering technique in the beacon decision making process.

9.7 Discussion

We draw two general lessons from our design and evaluation of STROBE.

1. For density regimes above the threshold density, our example shows that a completely localized algorithm like STROBE can extend the system lifetime 1.5 times without diminishing localization granularity with 3.1 times saturation density of nodes. Lifetime gains can be improved further for higher beacon densities and energy dissipation rates in active state, and by augmenting STROBE with boundary detection mechanisms.

2. Adaptation to terrain conditions and node availability invariably has an associated measurement overhead. Therefore adaptive density should be applied only when the benefit of adaptation greatly exceeds its overhead. Examples of this are high density beacon deployment and high energy dissipation in active states. STROBE is not justifiable in contexts when beacons are already operating at a very low duty cycle or when the deployment density is not high enough to provide enough interchangeable beacons.

9.8 Summary

In this chapter, we presented STROBE, our algorithm for self-configuration at high beacon densities. STROBE builds on the observation that proximity-based localization saturates at a certain beacon density to rotate functionality amongst redundant beacons and extend system lifetime.

We described the duty cycle of beacons in STROBE, and its decision making approach. We presented our justification for choosing three states in STROBE and the relative time periods for each state via energy usage analysis.

We presented detailed simulations to show that STROBE (i) converges quickly, (ii) maintains uniform localization quality both across the terrain and over time, and (iii) can significantly extend overall system lifetime.

Like HEAP, STROBE is also a general approach that can be applied to localization systems not based on RF-proximity. The only aspect of STROBE that is domain specific is the decision making function.

Furthermore, our design and evaluation methodology for STROBE can be applied to other networking problems where the measured performance is a function of node density.

CHAPTER 10

Conclusions and Future Work

The outcome of any serious research can only be to make two questions grow where only one grew before.

— Thorstein Veblen

This is not the end. It is not even the beginning of the end. But, it is perhaps the end of the beginning.

— Winston Churchill

We close this dissertation with an enumeration of several remaining challenges to our proposed approach. We then present a number of research problems that may be addressed in future work. Next we describe several cases where our algorithms may potentially impact research areas outside of our self-configuring localization system architecture. Finally, we outline the availability of our reference implementations, simulation scripts, and tools, and conclude.

10.1 Outstanding Problems

Although the self-configuring localization system framework provides a promising foundation for scalable, ad hoc deployable, RF-based localization in unpredictable environments, a number of outstanding problems must be addressed.

Sensor self-calibration

What is the nominal transmission range of a beacon for a given transmit power level? The answer varies depending upon the environment, and for different COTS beacons in the same environment. Sensor and radio calibration remains one of the single biggest problems in the use of sensor networks. Many sensors (seismic, acoustic, radio etc.) apply amplitude based models that relate distance to received signal power for sensing. These can change very easily in different environments and across different COTS sensors. Automating sensor calibration is an important issue that needs to be addressed. Some initial problem formulations and solutions may be found in [WC02, BME02]. Useful ideas for formalizing the calibration problem can be leveraged from the field of computer vision.

Measured power consumption

Power-conservation is the key design factor in sensor networks, and in our algorithms such as STROBE. Although, we have evaluated STROBE using real radio data, we have not measured the real power consumption of the sensor nodes. Instead, we have emulated the radio energy usage. It is increasingly desirable to evaluate these algorithms using real power measurements to completely validate the algorithms. Techniques to measure battery capacity are suggested in [PSS01].

Convergence

Our simulations show that STROBE converges quickly to a stable state (within 6 cycles) — in which only the desired number of beacons are active. Our simulations assume beacons are distributed uniformly at random. However, we have not proved the convergence properties of STROBE theoretically. The convergence properties of distributed self-configuring protocols (such as STROBE [BHE01b], ASCENT [CE02], SPAN [CJM01] and GAF [XHE01]) are not well understood and need to be further analyzed.

10.2 Future Directions

The work in this dissertation motivates some interesting and potentially fruitful areas for future work. Some of these are direct extensions derived from our work in this dissertation and are closely related to our self-configuring framework. Other ideas focus on interesting new areas or novel applications of our framework and motivate research in significantly new directions.

10.2.1 Self-Configuration

There are a number of areas of future work related to self-configuration.

Self-configuring Network Protocols

Lately there has been significant research in self-configuring, density-adaptive network protocols — to establish a connected network topology, beacon systems, and perform adaptive routing. In these protocols, only some nodes in the network must participate to perform the task, the rest can go to sleep. However in sensor networks, some nodes may be participating in several different tasks (routing, topology control, beaconing). A node's decision making process must thus be integrated across all the tasks it is participating in. Implementing such integrated decision-making seems to be an interesting problem to explore in the future.

In STROBE, the decision of a beacon to remain active or sleep is influenced only by requirements to maintain a uniform localization granularity across the system at all times. We may not really need homogeneous localization granularity in the system at all times, especially if the system is event based (as in STEM [STS02]). Instead we may want the beacon network to self-configure in response to application dynamics or events. We have experimented with triggered beacon systems in our laboratory and these can lead to orders of magnitude improvements in energy-conservation.

A Theory of Self-Configuring Networks

So far the focus has been in the development of network protocols for self-configuration. However, the performance of these protocols is extremely sensitive to the choice of parameters. Consequently, it is important to establish a cohesive theoretical foundation for self-configuring systems. Besides our density-based analysis, analyses based on phase transitions [KBW02] and geometric relationship maintenance [Gui02] seem promising approaches in this direction.

10.2.2 Localization

There are a number of areas of future work related to localization.

Robust Position Estimation Algorithms

In our localization methodology, nodes simply inferred position from beacons they were directly connected to. A more challenging problem is to make beacons *themselves* learn their positions without any references, and in a distributed computation. We need robust *distributed* position estimation algorithms when there are very few references. While there have been some algorithmic and system developments lately, they have assumed simple error models [SHS01b] or relied on centralized location computation [GBE02].

Federated Spatial Coordinate Systems

Because no single localization technology works everywhere, one expects the use of several localization technologies for pervasive computing applications, each with their own frames of reference, location granularity and error models. An interesting area of future work is the development of algorithms for establishing federated coordinate systems that can combine location information from several frames of reference and will allow a new class of seamlessly integrated applications.

Location Models

Several location-aware computing applications act not on the physical location (e.g. X, Y, Z coordinates) but logical location [SAW94] (room 210, Building A).

Hightower *et al* [HBB02] have proposed a seven layer *Location Stack*, analogous to the OSI networking stack [Zim80] for organizing functionality in location-aware applications. However, applications can be more robust and adaptive if they are aware of the uncertainty in the location information (for example, the accuracy or update latency in location information) or the costs involved in obtaining that location information (for example, power expended) [BEH01].[1]

Developing location models that present the appropriate abstraction of location information to an application is a challenging problem for the future, especially given the wide range of applications.

[1] This concept can be considered equivalent to the well-known Application Layer Framing (ALF) argument of Clark and Tennenhouse [CT90] for network applications, which states that applications can best decide how to adapt to the network conditions.

Applications

Several technologies currently exist for fine-grained localization. These achieve finer granularity at the expense of scalability [WJH97], form factor or energy expended [PCB00] or responsiveness [GBE02] — that precludes their use for a wide-range of applications where localization granularity requirements are lower — but where node localization must be scalable, responsive, unobtrusive and not waste energy. Our localization system is applicable in these contexts and we enumerate these applications below.

1. *Tag and track:* Really small devices are cheap, low-power, unobtrusive and enable measurement in the physical world. Thus, they are ideal for tagging and tracking the movements of wild animals in biological studies to monitor their behavior, or migration patterns. They could also be used to track the movements of users in a field, such as an auditorium or stadium (for example, tracking sports players around a field). They could be used for search and rescue in ski-hill resorts where one would want to beacon for help, or want to search for a person lost in an avalanche.

2. *Power-conservation in the network:* Localization on a scale with transmission range opens up new ways of power conservation in multi-hop wireless networks.

   It could be used to implement directional broadcasts — when an event is only of interest to nodes located in a certain direction, only broadcast in that direction. It could also be used to implement range-limited broadcasts that conserve power —- a base station estimates the farthest distance it must transmit to cover all nodes, and reduces its transmission power accordingly.

   Once nodes know where they are, they can advertise this information to their neighbors. Distributed nodes can elect leaders based on *best* location. Services needed near a location can be active, while further from a location, nodes can power down.

3. *Approximate Navigation:* Limited mobility sensors such as Robomote [SRS02] could automatically replace sensors that have died. Each failed sensor could pass along its position to the others so that if it goes down, another sensor could be sent automatically to take its place, given the last position it was known to occupy.

10.2.3 HEAP/STROBE Beyond Localization

Several concepts from this dissertation could be applied to areas outside our self-configuring localization system architecture.

Passive Localization and Tracking Systems

We considered self-configuration in a localization system that relies on active beacons and has passive clients (target nodes that need to be located). An alternative kind of localization and tracking systems are sensor networks deployed to passively detect and track objects (such as [CHZ02, WEG03, LWH02, BI98]). Sensors tracking an event can calculate a trajectory and alert nodes in the path of the event to listen, while nodes not in the trajectory can sleep and conserve energy. Chu *et al* [CHZ02] have studied high-level algorithms for self-configuration based on sensor information. These could be combined with our complementary density-based protocols for self-configuration to make the system robust.

Time Synchronization in Multi-hop Sensor Networks

Reference Broadcast Synchronization (RBS) [EGE02] is a recently developed, promising fine-grained network time synchronization scheme wherein a set of receivers are synchronized with each other by listening to the sender's reference broadcast, in contrast to a traditional time synchronization protocol in which a receiver synchronizes with a sender. To achieve network-wide time synchronization in a multi-hop network, broadcast regions of two or more senders must overlap with each other. Techniques like STROBE can be applied to determine which set of nodes should send a reference broadcast to accomplish network-wide time synchronization.

10.2.4 New Research Problems in Sensor Networks

From our experiences with the localization problem, we derive several observations on more general problems encountered in wireless sensor networks — which motivate significantly new areas of research.

Code Construction

During our experiments on radio propagation, a problem we frequently encountered was interference from external sources (such as cell phones and wireless repeaters) and sensor noise — which led to high packet loss in the wireless channel. Such a situation could be severely exacerbated in dynamic, unpredictable environments. Traditional error-correcting codes such as Hamming codes [Ham50] enable reliable communication without retransmission over a noisy channel with a low error rate, and are not equipped to cope with such noise. Enabling reliable communication in rapidly deployable, wireless sensor networks poses the following algorithmic challenge — developing new code constructions that are designed to tolerate extremely large amounts of noise and yet are computationally efficient to be usable in sensor nodes with modest processing capabilities.

Distributed Feature Extraction

Readers may recall our discussion in Chapter 9 where we stated that the performance of the STROBE algorithm could be further improved if beacons on the boundary could perform edge detection and operate conservatively. This is an example of the general problem of distributed feature extraction — how do sensor nodes sharing a common feature (for example, they all lie on the same temperature ISO-contour) collaborate to establish that feature and act accordingly? Techniques from image processing may be fruitfully applied here (as in [GEH02, SE02]).

Random Sampling

Our adaptive, measurement-based algorithms such as GRID (Chapter 7) and HEAP (Chapter 8) use comprehensive measurements of the terrain to diagnose problems. Applying random sampling techniques [MR95, MRL99] to measure only select parts of the terrain or process measurements selectively could considerably improve the computational efficiency of our algorithms. Furthermore, random sampling techniques could also find application in the feature extraction problems discussed above. An initial example of such efforts is [BEG02].

Byzantine fault-tolerance

Wireless sensor networks are vulnerable to a number of failures (such as node failures due to energy depletion) and security threats common to wireless networks (such as spoofing, snooping, and jamming the channel). While recent research has addressed low-level security primitives such as efficiently authenticating broadcast communication [PSW01], higher-level issues such as providing system-wide fault tolerance have not been sufficiently explored. Byzantine fault-tolerance seems to be an appropriate goal for wireless sensor networks, and can be provided through redundant nodes.

An interesting challenge will be building networks that are both *dependable* and *energy-efficient*. For instance, redundant nodes (in a factor of 3) in sensor networks can be used for Byzantine fault-tolerance, as well as to achieve energy conservation by reducing the duty cycles of individual nodes. Can we accomplish both without requiring 9 times the numbers of nodes?

10.3 Availability

All of the software and protocol implementation developed in this dissertation are available on-line at:

`http://lecs.cs.ucla.edu/~bulusu/localization`

10.4 Summary

In this dissertation, we proposed several general techniques to make a localization system self-configuring.

Localization, or the problem of estimating spatial relationships among objects, has been a classical problem in many disciplines, including mobile robotics [TFB01], virtual reality systems [WBV99], navigation systems [VOR, HLC92], and cellular networks [RAD].

A key challenge in engineering localization systems for these applications has been environmental dependence, because the nature of the environment often influences the characteristics of the sensors used for localization.

Traditionally, this has been addressed through extensive environment-specific calibration and configuration of the centrally controlled, tightly coupled localization system [WBV99, BP00b, RAD] and sophisticated, memory and compute-intensive probabilistic position-estimation algorithms [TFB01].

Large-scale, densely distributed sensor networks that are closely coupled to the physical world require node localization, but under far severe node-level resource constraints (limited energy, bandwidth, memory and processing) [BHE00]. Localization systems that can reconcile these needs by necessity must be loosely coupled, distributed systems [BHE00, PCB00, SHS01a, Gir00, HWB00].

We have highlighted the deployment, configuration and operational issues of such a localization system and argued that it must itself *self-configure*, that is, autonomously measure and adapt to the environmental and system dynamics in order to achieve environmental independence and robust, unattended system-level operation.

In this dissertation, we have presented the design and evaluation of algorithms to achieve that self-configuration. Our design process relied on the following four pieces of insight.

First, beacons are one key approach to localization as they can guarantee the convergence and accuracy of loosely-coupled distributed localization systems [SHS01a]. This realization argues for a beacon-based approach to self-configuration.

Second, the localization granularity can be directly related to the beacon density when beacons are distributed uniformly at random. Our simulations show that the localization granularity saturates at a threshold beacon density μ_{thresh}. This realization argues for a density-sensitive approach to self-configuration.

Third, the beacon density is not a homogeneous phenomenon in real environments.

Rather, the beacon density varies throughout the terrain due to deployment perturbations and due to environment-dependent propagation vagaries even when beacons are placed uniformly. Note that this observation implies that just deploying beacons below or at this theoretical density μ_{thresh} will not be adequate. Instead, beacons must themselves establish the density through measurements and suggest local candidate points where new beacons could be added so as to improve the localization quality, as in the HEAP algorithm proposed herein. Care was taken to propagate neighborhood information beyond a single hop, so that beacons can select the candidate points effectively.

Fourth, beacons contend for the wireless channel when they broadcast advertisement packets containing their position. When beacons are deployed at high densities (greater than μ_{thresh}) in order to provide redundancy, the responsiveness and granularity of the system degrades due to the self-interference caused by the channel contention. Instead of having all the beacons simultaneously participate, beacons must explicitly coordinate so that only some of them participate at a time. For a variety of performance reasons, characterizing the measured and threshold density allowed a simple randomized algorithm that achieved the desired statistical behavior in maintaining localization granularity. Careful analysis of energy usage allows us to tune sleep probabilities and periods so as to maximize system lifetime. This idea could be relevant to not only beacons but also to routing [XHE01], media access [YHE02] and topology control [CE02].

Our experimental results show that these various algorithms have significantly improved the performance of the localization system proposed in Chapter 5.

REFERENCES

[ACH01] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. “Implementing a sentient computing system.” *IEEE Computer*, **34**(8):2–8, August 2001.

[ACZ01] Joe Albowicz, Alvin Chen, and Lixia Zhang. “Recursive position estimation in sensor networks.” In *Proceedings of the International Conference on Network protocols (ICNP ’01)*, pp. 35–41, Riverside, California, November 2001. IEEE Computer Society.

[AET] Aetherwire Corporation. *http://www.aetherwire.com*.

[Aur91] Franz Aurenhammer. “Voronoi diagrams — a survey of a fundamental geometric data structure.” *ACM Computing Surveys (CSUR)*, **23**(3):345–405, 1991.

[AWA] Georgia Institute of Technology. “The Aware Home.” *http://www.awarehome.gatech.edu*.

[Azu93] Ronald Azuma. “Tracking requirements for augmented reality.” *Communications of the ACM*, **36**(7):50–51, July 1993.

[BEG95] Marshall Bern, David Eppstein, Leonidas Guibas, Jay Hershberger, Subhash Suri, and J. Wolter. “The centroid of points with approximate weights.” In *Proceedings of the 3rd European Symposium on Algorithms*, volume 979 of *Lecture Notes on Computer Science (LNCS)*, pp. 460–472. Springer-Verlag, 1995.

[BEG01] Nirupama Bulusu, Deborah Estrin, Lewis Girod, and John Heidemann. “Scalable coordination in sensor networks: Self-configuring localization systems.” In *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA)*, Ambleside, Lake District, UK, July 2001.

[BEG02] Solomon Bien, Deborah Estrin, and Ashish Goel. “Data-driven protocols and probabilistic sampling for sensor networks.” *Center for Embedded Networked Sensing (CENS) Opening Ceremony*, October 2002. Poster presentation.

[BEH01] Nirupama Bulusu, Deborah Estrin, and John Heidemann. “Tradeoffs in location support systems: The case for quality-expressive location models for applications.” In Michael Beigel, Phil Gray, and Daniel Salber, editors, *Location Modeling for Ubiquitous Computing - Ubicomp 2001 Workshop Proceedings*, pp. 7–12, Atlanta, GA, USA, September 2001.

[BHE00] Nirupama Bulusu, John Heidemann, and Deborah Estrin. “GPS-less low cost outdoor localization for very small devices.” *IEEE Personal Communications Magazine*, **7**(5):28–34, October 2000.

[BHE01a] Nirupama Bulusu, John Heidemann, and Deborah Estrin. “Adaptive beacon placement.” In *Proceedings of International Conference on Distributed Computing Systems (ICDCS-21)*, pp. 489–498, Phoenix, Arizona, USA., April 2001. IEEE Computer Society.

[BHE01b] Nirupama Bulusu, John Heidemann, and Deborah Estrin. “Density adaptive algorithms for beacon placement in wireless sensor networks.” Technical Report UCLA-CS-010013, University of California at Los Angeles, Los Angeles, California, USA., May 2001.

[BI98] R.R. Brooks and S.S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Prentice Hall PTR, 1998.

[BME02] Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. “Distributed calibration in sensor networks.” *Center for Embedded Networked Sensing (CENS) Opening Ceremony*, October 2002. Poster presentation.

[BP00a] Paramvir Bahl and Venkata N. Padmanabhan. “Enhancements to the RADAR user location and tracking system.” Technical Report MSR-TR-2000-12, Microsoft Research, Redmond, WA, USA, February 2000.

[BP00b] Paramvir Bahl and Venkata N. Padmanabhan. “RADAR: An in-building user location and tracking system.” In *Proceedings of IEEE Infocom 2000*, volume 2, pp. 775–784, Tel-Aviv, Israel, March 2000. IEEE.

[CCK01] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. “A probabilistic location service for wireless network environments.” In *Proceedings of Ubicomp 2001*, pp. 18–24. Springer Verlag, September 2001.

[CE02] Alberto Cerpa and Deborah Estrin. “ASCENT: Adaptive self-configuring network topologies.” In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2002)*, volume 3, pp. 1278–1287, New York, New York, USA, June 2002. IEEE.

[CEE01] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. "Habitat monitoring: Application driver for wireless communications technology." In *Proceedings of ACM SIGCOMM Workshop on Data Communications in Latin America and the Carribean*, pp. 20–41, San Jose, Costa Rica, April "3–5" 2001. ACM Press.

[CEN] University of California at Los Angeles. "The Center for Embedded Networked Sensing (CENS)." *http://cens.ucla.edu*.

[CGS99] Moses Charikar, Sudipto Guha, David Shmoys, and Eva Tardos. "A constant-factor approximation algorithm for the k median problem." In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1–10. ACM, May 1999.

[CHZ02] Maurice Chu, Horst Haussecker, and Feng Zhao. "Scalable information driven sensor querying and routing for Ad Hoc Heterogeneous sensor networks." *International Journal on High Performance Computing Applications*, 2002. To appear.

[CIT] University of California at Berkeley. "The Center for Information Technology Research in the Interest of Society (CITRIS)." *http://citris.berkeley.edu*.

[CJM01] Benjie Chen, Kyle Jamieson, Robert T. Morris, and Hari Balakrishnan. "SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks." In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '01)*, pp. 85–96, Rome, Italy, July 2001. ACM.

[Cox91] Inegmar J. Cox. "Blanche — An experiment in guidance and navigation of an autonomous robot vehicle." *IEEE Transactions on Robotics and Automation*, **7**(2):193–204, 1991.

[CRO] Crossbow Technologies Incorporated. *http://www.xbow.com*.

[CT90] David D. Clark and David L. Tennenhouse. "Architectural considerations for a new generation of protocols." In *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM '90)*, pp. 200–208, Philadelphia, PA, USA, August 1990. ACM Press.

[DPG01] Lance Doherty, Kristofer S.J. Pister, and Laurent El Ghaoui. "Convex position estimation in wireless sensor networks." In *Proceedings of IEEE INFOCOM 2001*, volume 3, pp. 1655–1663, Anchorage, Alaska, USA, April 22-26 2001.

[EE01] Jeremy Elson and Deborah Estrin. "Randomized, ephemeral transaction identifiers in wireless sensor networks." In *Proceedings of International Conference on Distributed Computing Systems (ICDCS-21)*, pp. 459–468, Phoenix, Arizona, USA, April 2001. IEEE Computer Society.

[EGE02] Jeremy Elson, Lewis Girod, and Deborah Estrin. "Fine-grained network time synchronization using reference broadcasts." In *Proceedings of the Fifth Symposium on Operating System Design and Implementation (OSDI 2002)*, Boston, Massachussetts, USA, December 2002. To Appear.

[EGH99] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. "Next century challenges: Scalable coordination in sensor networks." In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*, pp. 263–270, Seattle, WA, USA, August " 15–20" 1999. ACM Press.

[Fin87] Gregory Finn. "Routing and addressing problems in large metropolitan-scale internetworks." ISI Research Report ISI/RR-87-180, Information Sciences Institute, University of Southern California, March 1987.

[FJL95] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. "A reliable multicast framework for light-weight sessions and application level framing." In *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM '95)*, pp. 242–256, Cambridge, MA, USA, August 1995. ACM.

[GBE02] Lewis Girod, Vladimir Bychkovskiy, Jeremy Elson, and Deborah Estrin. "Locating tiny sensors in time and space: A case study." In *Proceedings of the International Conference on Computer Design (ICCD 2002) (invited paper)*, Freiburg, Germany, September 16-18 2002. To Appear.

[GE01a] Lewis Girod and Deborah Estrin. "Robust range estimation for localization in ad hoc sensor networks.", 2001. http://scantron.lecs.cs.ucla.edu/papers/NLOS.ps.

[GE01b] Lewis Girod and Deborah Estrin. "Robust Range Estimation Using Acoustic and Multimodal Sensing." In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2001)*, volume 3, pp. 1312–1320. IEEE, March 2001.

[GEH02] Deepak Ganesan, Deborah Estrin, and John Heidemann. "Dimensions: Why do we need a new data handling architecture for sensor networks?"

In *First Workshop on Hot Topics in Networks (Hotnets-I)*, Princeton, New Jersey, USA, October 2002. To appear.

[Gir00] Lewis Girod. "Development and characterization of an acoustic rangefinder." Technical Report USC-CS-TR-00-728, University of Southern California, April 2000.

[GK00] Piyush Gupta and P. R. Kumar. "The capacity of wireless networks." *IEEE Transactions on Information Theory*, **46**(2):388–404, March 2000.

[GKW02] Deepak Ganesan, Bhaskar Krishnamachari, Alec Woo, David Culler, Deborah Estrin, and Stephen Wicker. "Complex behavior at scale: An experimental study of low-power wireless sensor networks." Technical Report UCLA/CSD-TR 02-0013, University of California at Los Angeles, Department of Computer Science, Los Angeles, California, USA, 2002.

[GLL99] Leonidas Guibas, D Lin, Jean Claude Latombe, S LaValle, and Rajeev Motwani. "Visibility-based pursuit evasion in a polygonal environment." *International Journal of Computational Geometry Applications*, **9**(5):471–494, October 1999.

[Gui02] Leonidas J. Guibas. "Sensing, tracking and reasoning with relations." *IEEE Signal Processing Magazine*, **19**(2):73–85, March 2002.

[Ham50] Richard W. Hamming. "Error detecting and error correcting codes." *Bell Systems Technical Journal*, **29**:147–160, April 1950.

[HB01] Jeffrey Hightower and Gaetano Borriello. "Location systems for ubiquitous computing." *IEEE Computer*, **34**(8):57–66, August 2001.

[HBB02] Jeffrey Hightower, Barry Brumitt, and Gaetano Borriello. "The location stack: A layered model for location in ubiquitous computing." In *Proceedings of the 4th IEEE Workshop on mobile Computing Systems & Applications (WMCSA 2002)*, pp. 22–28, Callicoon, NY, USA, June 2002. IEEE Computer Society Press.

[HBE01] John Heidemann, Nirupama Bulusu, Jeremy Elson, Chalermek Intanagonwiwat, Kun chan Lan, , Ya Xu, Wei Ye, Deborah Estrin, and Ramesh Govindan. "Effects of detail in wireless network simulation." In *Proceedings of the SCS Multiconference on Distributed Simulation*, pp. 3–11, Phoenix, Arizona, USA, January 2001. Society for Computer Simulation.

[HCB00] Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocols for wireless sensor networks." In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS)*, pp. 3005–3014, Big Island, Hawaii, USA, January 4-7 2000.

[HHS99] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. "The anatomy of a context-aware application." In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*, pp. 59–68, Seattle, WA, USA, August 1999. ACM Press.

[HLC92] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. *Global positioning system: Theory and practice, second edition*. Springer-Verlag, 1992.

[HMS02] Andrew Howard, Maja J. Mataric, and Gaurav S. Sukhatme. "An incremental deployment algorithm for mobile robot teams." In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2002)*, volume 3, pp. 2849–2854. IEEE, March 2002.

[HS98] C. Hofner and G. Schmidt. "Path planning and guidance techniques for an autonomous mobile cleaning robot." *Robotics and Autonomous Systems*, **14**:199–212, May 1998.

[HSI01] John Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. "Building efficient wireless sensor networks with low-level naming." In *Proceedings of the Symposium on Operating Systems Principles*, pp. 146–159, Chateau Lake Louise, Banff, Alberta, Canada, October 2001. ACM.

[HSW00] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. "System architecture directions for networked sensors." In *Proceedings of ASPLOS-IX*, pp. 93–104, Cambridge, MA, USA, November 2000. ACM.

[HWB00] Jeffrey Hightower, Roy Want, and Gaetano Borriello. "SpotON: An indoor 3D location sensing technology based on RF signal strength." UW CSE 2000-02-02, University of Washington, Seattle, WA, USA, February 2000.

[IEG02] Chalermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. "Impact of network density on data aggregation in wireless sensor networks." In *In Proceedings of the International Conference on*

Distributed Computing Systems (ICDCS-22), Vienna, Austria, July 2002. IEEE.

[IGE00] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. "Directed diffusion: A scalable and robust communication paradigm for sensor networks." In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pp. 56–67, Boston, MA, USA, August 2000. ACM Press.

[Jac88] Van Jacobson. "Congestion avoidance and control." In *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM '88)*, pp. 314–329, Stanford, CA, USA, August 1988. ACM Press.

[JDS95] Sugih Jamin, Peter B. Danzig, Scott Shenker, and Lixia Zhang. "A measurement-based admission control algorithm for integrated services packet networks." In *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM '95)*, pp. 2–13, Cambridge, Massachussetts, United States, August 1995. ACM Press.

[Kai00] William J Kaiser. "WINS NG 1.0 Transceiver Power Dissipation." *Sensoria Corporation*, 2000.

[KBW02] Bhaskar Krishnamachari, Remon Bejar, and Stephen Wicker. "Distributed problem solving and the boundaries of self-configuration in multi-hop wireless networks." In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS '02)*, pp. 3888–3897, Big Island, Hawaii, USA., January 7-10 2002. IEEE.

[KK00] Brad Karp and H.T. Kung. "GPSR: Greedy perimeter stateless routing for wireless networks." In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pp. 243–254, Boston, MA, USA, August 2000. ACM Press.

[KKP99] Joseph M. Kahn, Randy H. Katz, and Kristofer S. J. Pister. "Next century challenges: Mobile networking for smart dust." In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*, pp. 271–278, Seattle, WA, USA, August " 15–20" 1999. ACM Press.

[Lam83] Butler W. Lampson. "Hints for computer system design." *Operating Systems Review*, **17**(5):33–48, October 1983. *Ninth ACM Symposium on Operating Systems Principles*. Later republished, but with less satisfactory copy editing, in *IEEE Software* 1,1 (January 1984) pages 11-28.

[LWH02] Dan Li, Kerry D. Wong, Yu Hen Hu, and Akbar M. Sayeed. "Detection, classification, and tracking of targets." *IEEE Signal Processing Magazine*, **19**(2):17–30, March 2002.

[MKP01] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani Srivastava. "Coverage problems in wireless ad hoc sensor networks." In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2001)*, volume 3, pp. 1380–1387, Anchorage, Alaska, April " 22-26" 2001. IEEE.

[Moo] Gordon S. Moore. "Moore's Law." *http://www.intel.com/research/silicon/mooreslaw.htm*.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MRL99] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsey. "Random sampling techniques for space efficient online computation of order statistics of large datasets." In *Proceedings of ACM SIGMOD Conference 1999*, pp. 251–262, Philadelphia, Pennsylvania, USA, June 1-3 1999. ACM Press.

[MS00] Maja Mataric and Gaurav Sukhatme. "Embedding robots into the internet." *Communications of the ACM*, pp. 67–73, May 2000.

[Nag99] Radhika Nagpal. "Organizing a global coordinate system from local information on an amorphous computer." A.I. Memo 1666, Massachusetts Institute of Technology Artificial Intelligence Laboratory, August 1999. http://www.ai.mit.edu/publications/.

[NB00] Dragos Niculescu and B. R. Badrinath. "Ad hoc positioning system (APS)." Technical Report LCSR-TR-435, Laboratory of Computer Science Research, Rutgers University, 2000.

[OK93] Masatoshi Okutomi and Takeo Kanade. "A multiple-baseline stereo." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(4):353–363, 1993.

[OXY] MIT Laboratory For Computer Science. "Project Oxygen." *http://www.oxygen.lcs.mit.edu*.

[Pap91] Athanasios Papoulis. *Probability, random variables, and stochastic processes, third edition*. McGraw-Hill, 1991.

[PCB00] Nissanka Priyantha, Anit Chakraborty, and Hari Balakrishnan. "The Cricket location support system." In *Proceedings of ACM MobiCom 2000*, pp. 32–43, Boston, MA, USA, August 2000. ACM.

[PK00] Gregory J. Pottie and William J. Kaiser. "Wireless integrated network sensors." *Communications of the ACM*, **43**(5):51–58, May 2000.

[PMB01] Nissanka Priyantha, Allen Miu, Hari Balakrishnan, and Seth Teller. "The Cricket compass for context-aware applications." In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, pp. 1–14, Rome, Italy, July 2001. ACM.

[PSS01] Sung Park, Andreas Savvides, and Mani B. Srivastava. "Battery capacity measurement and analysis using lithium coin cell batteries." In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 382–387, Huntington Beach, California, USA, August 2001. IEEE.

[PSW01] Adrian Perrig, Robert Szewczyck, Victor Wen, David Culler, and Doug Tygar. "SPINS: Security protocols for sensor networks." In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '01)*, pp. 189–199, Rome, Italy, July 2001. ACM.

[RAD] U. S. Wireless Corporation. "Location pattern matching and the RadioCamera network." *http://www.uswcorp.com/USWCMainPages/our.htm*.

[Rap96] Theodore S. Rappaport. *Wireless communications: Principles and practice*. Prentice Hall PTR, 1996.

[RFM] RF Monolitithics Incorporated. *http://www.rfm.com*.

[RS00] Wasinee Rungsarityotin and Thad E. Starner. "Finding location using omnidirectional video on a wearable computing platform." In *Proceedings of the Fourth International Symposium on Wearable Computers (ISWC 2000)*, pp. 61–68, Atlanta, GA, USA, October 18–21 2000. IEEE Computer Society.

[RT99] Elizabeth M. Royer and Chai-Keong Toh. "A review of current routing protocols for ad-hoc mobile wireless networks." *IEEE Personal Communications*, **6**(2):46–55, April 1999.

[SAW94] Bill Schilit, Norman Adams, and Roy Want. “Context-aware computing applications.” In *Proceedings of the Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 85–90, Santa Cruz, CA, USA, December 1994. IEEE Computer Society.

[SC99] David Shmoys and F. A. Chudak. “Improved approximation algorithms for capacitated facility location problems.” In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. S875–S876. ACM Press, 1999.

[SCO] University of Southern California. “SCOWR: Scalable coordination for wireless robots.” *http://robotics.usc.edu/scowr*.

[SE02] Thomas Schoellhammer and Deborah Estrin. “Distributed pattern matching in sensor networks.” *Center for Embedded Networked Sensing (CENS) Opening Ceremony*, October 2002. Poster presentation.

[SGA00] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J. Pottie. “Protocols for Self-Organization of a wireless Sensor Network.” *IEEE Personal Communications Magazine*, **7**(5):16–27, October 2000.

[Sha48] Claude E. Shannon. “A mathematical theory of communication.” *Bell Systems Technical Journal*, **27**:379–423,623–656, 1948.

[SHS01a] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. “Dynamic fine-grained localization in ad-hoc networks of sensors.” In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, pp. 166–179, Rome, Italy, July 2001. ACM.

[SHS01b] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. “Location discovery in ad-hoc wireless networks.” ACM, 2001.

[SK97] Mark Stemm and Randy H. Katz. “Measuring and reducing energy consumption of network interfaces in hand-held devices.” *IEICE Transactions on Communications*, **E80-B**(8):1125–1131, August 1997.

[Sri96] Anuj Srivastava. *Inferences on transformation groups generating patterns of rigid motion*. PhD thesis, Washington University, St. Louis, August 1996.

[SRS02] Gabe T. Sibley, Mohammed H. Rahimi, and Gaurav S. Sukhatme. “Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks.” In *Proceedings of the IEEE International Conference on Robotics *and Automation (ICRA-2002)*, volume 2, pp. 1143–1148, Washington D.C, USA, May 11-15 2002. IEEE.

[STS02] Curt Schurgers, Vlasios Tsiatsis, and Mani B. Srivastava. “STEM: Topology management for energy efficient sensor networks.” In *Proceedings of the IEEE Aerospace Conference 2002*, volume 3, pp. 1099–1108, Big Sky, MT, USA, March 17-22 2002.

[Tan96] Andrew S. Tannenbaum. *Computer Networks, Third Edition*. Prentice Hall PTR, 1996.

[TES] The SCADDS TestBed. “http://www.isi.edu/scadds/testbed.html.”.

[TFB01] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. “Robust Monte Carlo localization for mobile robots.” *Artificial Intelligence*, **128**(1–2):99–141, Summer 2001.

[TIM] Timedomain Corporation. *http://www.timedomain.com*.

[VOR] VOR (Very High Frequency Omnirange). *http://www.allstar.fiu.edu/aero/VOR.htm*.

[War98] Andrew Martin Robert Ward. *Sensor-driven computing*. PhD thesis, University of Cambridge, Cambridge, UK, August 1998.

[WBV99] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D’nardo Colucci. “The HiBall Tracker: High-performance wide-area tracking for virtual and augmented environments.” In *Symposium on Virtual Reality Software and Technology*, pp. 1–10, December 1999.

[WC02] Kamin Whitehouse and David Culler. “Calibration as parameter estimation in sensor networks.” In *Proceedings of the First ACM International Workshop on Sensor Networks and Applications (WSNA 2002)*, Atlanta, Georgia, USA, September 2002. To appear.

[WEG03] Hanbiao Wang, Jeremy Elson, Lewis Girod, Deborah Estrin, and Kung Yao. “Target classification and localization in a habitat monitoring application.” In *Proceedings of the IEEE ICASSP 2003*, Hong Kong, April 2003. To appear.

[Wei91] Mark Weiser. “The computer for the 21st century.” *Scientific American*, **265**(3):94–104, September 1991.

[Wei93] Mark Weiser. "Some computer science issues in ubiquitous computing." *Communications of the ACM*, **36**(7):75–84, July 1993.

[WHF92] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. "The Active Badge location system." *ACM Transactions on Information Systems*, **10**(1):91–102, January 1992.

[WJH97] Andy Ward, Alan Jones, and Andy Hopper. "A new location technique for the Active Office." *IEEE Personal Communications Magazine*, **4**(5):42–47, October 1997.

[WL98] Jay Werb and Colin Lanzl. "Designing a positioning system for finding things and people indoors." *IEEE Spectrum*, **35**(9):71–78, September 1998.

[XHE00] Ya Xu, John Heidemann, and Deborah Estrin. "Adaptive energy-conserving routing for multihop ad hoc networks." Research Report 527, USC/Information Sciences Institute, October 2000.

[XHE01] Ya Xu, John Heidemann, and Deborah Estrin. "Geography-informed energy conservation for ad hoc routing." In *Proceedings of ACM MOBICOM 2001*, pp. 85–96, Rome, Italy, July 2001. ACM.

[YHE02] Wei Ye, John Heidemann, and Deborah Estrin. "An energy efficient MAC protocol for wireless sensor networks." In *Proceedings of IEEE Infocom 2002*, volume 2, New York, USA., June 2002. IEEE.

[YVS01] Wei Ye, Richard T. Vaughan, Gaurav S. Sukhatme, John Heidemann, Deborah Estrin, and Maja J. Mataric. "Evaluating control strategies for wireless-networked robots using an integrated robot and network simulation." In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2001)*, volume 3, pp. 2941–2947, Seoul, Korea, May 21-26 2001. IEEE.

[ZGE02] Yonggang Jerry Zhao, Ramesh Govindan, and Deborah Estrin. "Aggregate network properties for monitoring wireless sensor networks." Technical Report 02-773, Computer Science Department, University of Southern California, Los Angeles, California, USA., July 2002.

[Zha02] Yonggang Jerry Zhao. "Connectivity study of a CSMA based wireless network." Technical Report 02-774, Computer Science Department, University of Southern California, Los Angeles, California, USA., July 2002.

[Zim80] Hans Zimmerman. “OSI reference model — The ISO model of architecture for open systems interconnection.” *IEEE Transactions on Communications*, **COM-28**(4):425–432, April 1980.