

IMPROVING DATA AVAILABILITY IN MOBILE APPLICATIONS
THROUGH ENHANCED COOPERATIVE LOCALIZATION

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Jeffrey Matthew Hemmes

Douglas Thain, Director

Christian Poellabauer, Director

Graduate Program in Computer Science and Engineering

Notre Dame, Indiana

July 2009

The views expressed in this document are those of the author and do not reflect
the official policy or position of the United States Air Force, Department of
Defense, or the U.S. Government.

© Copyright by
Jeffrey Matthew Hemmes
2009
All Rights Reserved

IMPROVING DATA AVAILABILITY IN MOBILE APPLICATIONS THROUGH ENHANCED COOPERATIVE LOCALIZATION

Abstract

by

Jeffrey Matthew Hemmes

It has become commonplace for mobile computing systems to be constructed using low-cost, commodity components for localization. While the expected error in consumer-grade sensors can still be acceptable for localization at human scale, all sensors have fundamental limitations which manifest in different ways in different environments. Cooperative localization techniques can compensate for these hardware limitations, facilitating robust positioning in location-sensitive mobile applications.

Four significant challenges exist for developers of localized mobile systems. First, the nature of connectivity in mobile ad-hoc networks can often be highly sporadic, with frequent disconnects and changing topology. Second, sensor error frequently occurs in unexpected ways, is produced by multiple sensors working in tandem, or exhibits much different behavior depending on the type of sensor. Third, the lack of effective tools for measuring spatial separation between mobile nodes makes implementing existing localization methods difficult. Finally, many applications that rely on localization of remote network nodes can fail without accurate and precise positioning.

The foundation of this work is a rigorous evaluation and discussion of sensor error as encountered in practice. This work examines sensor error, which can

Jeffrey Matthew Hemmes

manifest in unexpected ways outside of controlled environments and applications, focusing on its effect on human-scale localization. Data collected from both empirical measurement and outdoor exercises are used to construct error models which may be used to evaluate new ideas in mobile cooperative computing. A simulation environment for mobile ad hoc networks incorporating various models of localization error is presented.

Next, the utility of sharing location information and error metrics among cooperating users is explored. Two methods are presented which can account for and reduce localization error using shared data, exploiting the independence of error among nodes in close proximity. A scenario-based evaluation approach is used to demonstrate possible techniques for using shared location information. System parameters required for effective utilization such data is also discussed. Simulation trials show that up to 50 percent reduction in overall localization error can be realized in many cases using only commercial-grade sensors.

Finally, the effect of robust localization and error reduction at the application layer is studied. This is examined in the context of a new method for selecting available peers in a mobile network for the purpose of short-term data storage and retrieval. By weighting the utility of each remote node based on error metrics and the confidence level of those metrics, an increase in the effective availability of data in the system can be demonstrated.

CONTENTS

FIGURES	v
TABLES	vii
ACKNOWLEDGMENTS	ix
CHAPTER 1: INTRODUCTION	1
1.1 Mobile Distributed Systems	1
1.2 Building Mobile Systems With Consumer-Grade Components	4
1.3 Why a Cooperative Model?	6
1.4 Potential Applications	7
1.5 Overview of Dissertation	9
CHAPTER 2: RELATED WORK	13
2.1 Sensor Evaluation and Mobile Ad-Hoc Network Testbeds	13
2.1.1 Simulation	14
2.1.2 Emulation	15
2.1.3 Experimental Trials	16
2.1.3.1 GPS Error Analysis	16
2.1.3.2 Dead Reckoning Systems	18
2.1.4 Existing Testbeds and Deployed Systems	19
2.1.4.1 Wireless Sensor Networks	19
2.1.4.2 Mesh Networks	21
2.1.4.3 Mobile Testbeds	22
2.2 Localization Techniques	23
2.3 Mobility	25
2.3.1 Mobility Models	26
2.3.1.1 Random Waypoint	26
2.3.1.2 Random Walk	27
2.3.1.3 Random Direction	28
2.3.2 Mobility Prediction	28

CHAPTER 3: MULTI-SENSOR LOCALIZATION	30
3.1 Introduction	30
3.2 The TeamTrak Mobile Testbed	32
3.3 Experiences With the Global Positioning System	37
3.3.1 Sources of Error	38
3.3.2 GPS Error in Practice	40
3.3.3 Error Metrics	44
3.3.4 Modeling GPS Error	58
3.4 Digital Accelerometer	60
3.4.1 Overview	60
3.5 Digital Compass	63
3.5.1 Overview	63
3.5.2 Sources of Error	64
3.6 Experiences With Dead Reckoning	65
3.6.1 Putting the Pieces Together	65
3.6.2 Dead Reckoning Error in Practice	68
3.7 The TT-Sim Simulator	71
3.8 Conclusion	72
 CHAPTER 4: COOPERATIVE LOCALIZATION	75
4.1 Introduction	75
4.2 Combination Methods	79
4.3 A Cooperative Approach to Localization	82
4.3.1 Single Localized Node	83
4.3.2 Two Localized Nodes	89
4.3.3 More Than Two Nodes	95
4.3.4 Connectivity	96
4.3.5 Combining Positions	101
4.4 Experimental Setup	105
4.5 Results	110
4.5.1 Simple Averaging: 300-Meter Range	111
4.5.2 Simple Averaging: 150-Meter Range	114
4.5.3 Simple Averaging: 125-Meter Range	114
4.5.4 Simple Averaging: 100-Meter Range	116
4.5.5 Simple Averaging: 30-Meter and 15-meter Range	118
4.5.6 Distribution of Error	120
4.5.7 Distribution of Connectivity	122
4.5.8 Inclusive Averaging	124
4.5.9 Selective Averaging	126
4.5.10 Filtered Trilateration	127
4.6 Limitations	138

4.6.1	Stability of Dead Reckoning	138
4.6.2	Trilateration Challenges	139
4.7	Conclusions	140
CHAPTER 5: ON-DEMAND DATA BACKUP AND RECOVERY IN MOBILE SYSTEMS		144
5.1	Context Aware Data Replication	147
5.1.1	Peer Selection	149
5.1.1.1	Initial Scoring	152
5.1.1.2	Availability Estimation	152
5.1.2	Data Management and Recovery	155
5.2	Evaluation	157
5.2.1	Initial Selection	158
5.2.2	Availability	161
5.2.3	Random Waypoints	166
5.2.4	Anisotropic RF Propagation	168
5.2.5	Localization Error	169
5.2.5.1	Weighting Localization Error	173
5.3	Conclusion	174
CHAPTER 6: CONCLUSION		176
BIBLIOGRAPHY		180

FIGURES

1.1	Commercial GPS for Military Use	6
1.2	Overview of Dissertation	10
3.1	TeamTrak Hardware Components	34
3.2	TeamTrak Display Modes	35
3.3	GPS Drift by Number of Satellites Acquired	41
3.4	GPS Drift by Number of Satellites Acquired (Continued)	42
3.5	GPS Error Over Time	42
3.6	Distribution of Error in Garmin and HP GPS Receivers	45
3.7	Measurement Error Cumulative Distribution Functions (Garmin and HP GPS Receivers)	46
3.8	Obstructed Sky View at Data Collection Point	48
3.9	Example of GPS Consistency in Practice	49
3.10	Reliability of Horizontal Error	57
3.11	Observed GPS Drift	59
3.12	Accelerometer Patterns	61
3.13	Effect of Pitch and Roll on Compass Heading	66
3.14	Distribution of Compass Error	67
3.15	Dead Reckoning Versus GPS Accuracy	68
3.16	Dead Reckoning Error In Practice	69
3.17	Dead Reckoning Error In Practice	70
3.18	Simulated Dead Reckoning Error	71
4.1	Averaging Positions (Not in Triangle)	81
4.2	Trilateration	82
4.3	Dead Reckoning Averaging (w/GPS Error)	84
4.4	Dead Reckoning Versus Actual Path (One Node)	86
4.5	Dead Reckoning Averaging (No GPS Error)	87
4.6	Dead Reckoning Versus Actual Path (Two Nodes)	91
4.7	Fixed Separation Between Two Nodes	92

4.8	Averaging Two Dead Reckoning Positions	94
4.9	Wireless Connectivity	98
4.10	Trilateration With Two Nodes	99
4.11	Example Initial Node Placements	109
4.12	Effects of Simulation Time on Mean Position Error	110
4.13	Averaging (300 m Range)	113
4.14	Averaging (150 m Range)	113
4.15	Averaging (125 m Range)	115
4.16	Averaging (100 m Range)	115
4.17	Effects of Simple Averaging	117
4.18	Averaging (30m Range)	119
4.19	Averaging (15m Range)	119
4.20	Distribution of Position Error (Averaging)	121
4.21	Distribution of Connections to Anchors	123
4.22	Inclusive Averaging (100-Meter Range)	125
4.23	Selective Averaging (100-Meter Range)	125
4.24	Trilateration (300 Meter Range)	128
5.1	Candidate Peer Node Selection	153
5.2	Availability With Linear Motion	162
5.3	Availability With Random Walk Mobility Model	162
5.4	Effect of Data Requirement on Success Rate	163
5.5	Selection: Arbitrary Mobility	167
5.6	Selection: Anisotropic RF Propagation	169
5.7	Candidate Selection (With Localization Error)	171
5.8	Selection: Localization Error	172
5.9	Selection: Localization Error (With Weighting)	174

TABLES

3.1	GPS SIGNAL QUALITY SAMPLE SIZES	43
3.2	NUMBER OF SATELLITES VS. GPS ERROR	46
3.3	LOCATION QUALITY/CONFIDENCE VS. HDOP VALUES . .	50
3.4	MEAN, MINIMUM, AND MAXIMUM REPORTED GPS POSITION ERRORS BY QUALITY (HP IPAQ RECEIVER)	52
3.5	HORIZONTAL DILUTION OF PRECISION ($\epsilon = 5, 10, 15, 20, 25, 30$)	52
3.6	HORIZONTAL DILUTION OF PRECISION ($\epsilon = 30, 35, 45, 50$)	54
3.7	EFFECT OF STRIDE LENGTH ESTIMATION ON ACCURACY OF TOTAL DISTANCE ESTIMATION FOR DEAD RECKONING	62
4.1	EFFECT OF POSITION AVERAGING ON PRECISION AND ACCURACY (DEAD RECKONING)	89
4.2	RANDOM WAYPOINTS, WIMAX RANGE, NO DISTANCE ERROR	130
4.3	PARALLEL PATHS, WIMAX RANGE, NO DISTANCE ERROR	131
4.4	RANDOM WAYPOINTS, WIMAX RANGE, WITH DISTANCE ERROR	131
4.5	PARALLEL PATHS, WIMAX RANGE, WITH DISTANCE ERROR	132
4.6	RANDOM WAYPOINTS, WIFI RANGE, NO DISTANCE ERROR	132
4.7	PARALLEL PATHS, WIFI RANGE, NO DISTANCE ERROR . .	133
4.8	RANDOM WAYPOINTS, WIFI RANGE, WITH DISTANCE ER- ROR	133
4.9	PARALLEL PATHS, WIFI RANGE, WITH DISTANCE ERROR	134
4.10	RANDOM WAYPOINTS, BLUETOOTH CLASS 2 RANGE, NO DISTANCE ERROR	134
4.11	PARALLEL PATHS, BLUETOOTH CLASS 2 RANGE, NO DIS- TANCE ERROR	135

4.12 RANDOM WAYPOINTS, BLUETOOTH CLASS 2 RANGE, WITH DISTANCE ERROR	135
4.13 PARALLEL PATHS, BLUETOOTH CLASS 2 RANGE, WITH DISTANCE ERROR	136
4.14 RANDOM WAYPOINTS, BLUETOOTH CLASS 3 RANGE, NO DISTANCE ERROR	136
4.15 PARALLEL PATHS, BLUETOOTH CLASS 3 RANGE, NO DISTANCE ERROR	137
4.16 RANDOM WAYPOINTS, BLUETOOTH CLASS 3 RANGE, WITH DISTANCE ERROR	137
4.17 PARALLEL PATHS, BLUETOOTH CLASS 3 RANGE, WITH DISTANCE ERROR	138
5.1 INITIAL SELECTION (LINEAR MOTION)	158
5.2 INITIAL SELECTION (RANDOM WALK)	160
5.3 AVERAGE TIME BEFORE FAILURE	165

ACKNOWLEDGMENTS

I am thankful for the many opportunities I have had to work closely with truly exceptional people during my time at Notre Dame.

First and foremost, my thesis advisor, Dr. Douglas Thain, has continually provided a challenging, creative, and productive environment in which I and the many other students under his tutelage can effectively carry out our research. He has done so by providing bountiful hardware and software resources, asking demanding questions during whiteboard sessions, and providing valuable opportunities to publish, travel, present ideas, and, perhaps most importantly, independently lead research efforts. I am exceedingly grateful for the many opportunities for professional growth and development Doug has provided.

Dr. Christian Poellabauer has continually challenged and mentored me to see problems from different perspectives, and his patience, encouragement, and terrific advice directly led to acceptance of at least one publication at a very high quality conference. His advice and coaching on writing and speaking effectively have provided a significant positive impact on my professional development, and through his highly effective instruction has been an important role model for my future academic career.

Dr. Surendar Chandra has provided terrific insight into wireless sensor networks, both in his coursework on the same topic and in one-on-one meetings discussing my research. I am grateful very for his instruction, sage advice, and

tough questions.

I am most grateful to the Air Force Institute of Technology and the Department of Computer and Electrical Engineering for their generous support of my education through academic fellowships and travel and book allowances, as well as the highly-sought-after opportunity to attend graduate school while on active duty. In particular, Major Christopher Mayer and Captain Dusti Jayne deserve special thanks. Chris worked with me closely to find just the right opportunity, and Dusti has been instrumental in helping me navigate the complexities of DoD and Air Force regulations and federal law, as well as providing top-notch administrative support at all times.

I would like to thank the U.S. Government, Department of Defense, for their very generous Defense University Research Instrumentation (DURIP) grant, number W911NF-06-1-0120, which allowed me to pursue my work by making possible the acquisition of the TeamTrak hardware.

Last, but certainly not least, I thank my beautiful wife Stacia for her tireless support and unyielding encouragement, without which none of this would have been even remotely possible.

CHAPTER 1

INTRODUCTION

1.1 Mobile Distributed Systems

The last decade has brought forth explosive growth in the use of personal communication devices based on wireless connectivity, which enables mobility and facilitates an “anywhere, anytime” computing capability. These types of systems, particularly those designed for robust commercial applications, are typically not fully autonomous and still require fixed networking infrastructure to varying extents in order to interoperate effectively and provide consistent performance. For example, a standard commercial cell phone cannot work at all unless within sufficient proximity to a cell tower. If such a system is to be used effectively in a very remote geographic area, networking infrastructure must be provided simultaneously, as is commonly the case with deployed military communications systems.

With increasing awareness of the capabilities of distributed systems, a broader segment of the scientific community has become appreciative of the benefits of their use. It is now relatively common to see them deployed as sensor networks used for data collection efforts supporting various research endeavors. Like traditional distributed systems used in data-intensive or computation-intensive scientific applications, sensor networks are generally designed and built with static

architectures and network topologies. However, they are frequently deployed to very remote physical locations in which *a priori* construction of wired infrastructure is either extraordinarily difficult or impossible. The importance users place on maximizing the effective lifespans of these systems operating under austere environmental conditions demands the use of limited computing power and minimal resource consumption. Furthermore, these systems are often deployed on a fairly large scale, composed of perhaps hundreds or even thousands of devices. These requirements and limitations are very important to note, as they frequently force system designers to consider harsh tradeoffs when building sensor networks.

Despite the existence of such an enormous variety of computing systems, the needs of many user communities are not adequately met. Many existing mobile computing platforms are still not directly interoperable, especially in remote environments, which makes data sharing among users working in concert challenging at best. While mobile systems need a capability that can be exploited without assuming pre-existing infrastructure, autonomous operation is not necessarily an overriding requirement, and is, in fact, only one of several key parameters. However, developing capable and reliable systems without reliance on infrastructure is a daunting problem, and one that will be the focus of much of this work. Mobile ad-hoc networks provide a convenient way of getting around these concerns, but typically pose significant reliability and performance problems in practice.

Whether or not a mobile computing device operates independently or as part of a larger distributed system, one of the canonical requirements for mobile applications is to have some notion of locality, i.e., satisfactorily answering the question *where in the world am I?* In sensor networks, localizing sensor output provides context, enabling the transformation of raw data into information that can be

effectively utilized at the application layer. However, localization is as difficult a problem to solve as it is important. The challenges arise from the inherent uncertainties in the sensor data upon which most localization techniques rely as well as the practical difficulties installing beacons or other fixed, ground-based infrastructure components. A large number of system designs incorporate use of the Global Positioning System (GPS), originally developed by the U.S. Department of Defense (Kaplan, 1996), and simply call the problem solved. Depending on the particular application, the use of commercial GPS alone may be sufficient. For instance, for an application that determines the location of a tractor-trailer rig, the magnitude of the approximation error inherent to GPS positions is probably acceptable; if localizing an individual sensor the size of a coin, perhaps not. Determining whether the precision and accuracy of a location approximation is acceptable or not is a fundamental problem that must be addressed in order for mobile applications to exploit locality.

Localization in sensor networks with static topologies has one important advantage over mobile networks: random errors in position can be smoothed out over time by averaging multiple samples. Indeed, averaging samples over a period of several days can yield accuracy to within one or two meters (Wilson, 2001). When localized nodes become mobile, however, that luxury is usually no longer available. One reason for this is the limitation of throughput found in most commercially available GPS receivers. In many types of receivers, position updates are only available at a rate of around once per second. A lag of several seconds which might be required to facilitate averaging makes futile any attempt to accurately estimate a position in this manner, for at least as long as the operator carrying the device is in motion. Furthermore, as users move about, blocking of

GPS signals may preclude collecting additional samples. Clearly, expecting the operator to stop and wait for a sufficient number of samples to be collected is not an acceptable tradeoff.

Finally, the problem of error detection and error handling is exacerbated by some of the practical realities in the current state of system design methodologies. These challenges, along with a few of the benefits, are discussed next.

1.2 Building Mobile Systems With Consumer-Grade Components

Component-based approaches have long been used in the area of systems engineering. Through the use of existing off-the-shelf components, acquired from either commercially available sources or reused from previous internal development projects, system development time and acquisition costs are reduced, oftentimes substantially (Heineman and Councill, 2001). This approach towards systems engineering dovetails nicely with the underlying philosophies of distributed computing. One of the most widely recognized advantages of distributed paradigms such as cluster or grid computing is that by constructing them from lower-cost commodity hardware and software, powerful computing capabilities can be acquired and systems deployed with lower overall acquisition costs and shorter development time than they could be otherwise. In many cases, the resultant cost savings arising from component reuse might mean the difference in whether system development and deployment are even feasible at all.

Distributed systems using traditional wired networks are well understood, as that area has enjoyed an abundance of research over the years. As an emerging technology, the use of mobile networks as distributed systems has been far more limited. Furthermore, the integration of off-the-shelf hardware and software into

mobile sensor networks is a relatively recent trend as well. Thus, the impact of commodity hardware and software integration on the suitability of such systems has not been examined in any significant way.

The challenge of developing with commodity components lies in the fundamental tradeoff that exists between cost and performance. While it might be possible to realize performance gains by incorporating custom-fabricated, high-quality parts, doing so would likely push development costs significantly over budget. More significantly, most sensor devices have fundamental limitations that cannot be overcome simply by throwing money at the problem. GPS is but one example of the latter. To construct a system using any off-the-shelf sensor device, a developer should pay particular attention to both the interfaces provided as well as the failure modes, as the sensor may have been originally developed intended for a much different application, and any new application using data from that sensor should account for these failure modes in some way. This requires a rigorous analysis of sensor data collected under realistic operational conditions, which are based on both the new system requirements and an evaluation of any resultant failure modes.

The TeamTrak mobile ad-hoc network testbed (Hemmes et al., 2007) is the product of our experiences using low-cost commodity hardware in mobile sensor networks. TeamTrak is a testbed implementation for outdoor urban or semi-urban environments, intended as an approximation of personal navigation systems developed for military use. It is indeed a quite reasonable model of such systems; a commercial GPS receiver used by the U. S. Air Force for training purposes is shown in Figure 1.1. Operation of physical implementations of such sensor systems under noisy, uncontrolled conditions presents both system designers and maintain-



Figure 1.1: Commercial GPS for Military Use

ers with unique fault tolerance challenges, many of which are unanticipated and frequently open new avenues for future research. This dissertation is the culmination of several years of experience building, deploying, and evaluating ideas in mobile cooperative computing using the TeamTrak framework. An overview of TeamTrak is presented in Section 3.2 of Chapter 3.

1.3 Why a Cooperative Model?

Cooperative computing is a way for users to work together and share resources with each other in a computing environment that is at once complex, dynamic, heterogeneous, and unreliable. While many works in the literature attempt localization among mobile nodes using distributed approaches, two aspects make this work unique and suitable for a cooperative approach. First, with multiple sensors attached to each node, each with individual advantages and disadvantages, it is quite possible for different users to have position estimates whose quality varies dramatically, even if multiple users happen to be in close proximity to each other.

Different types of sensors fail in different ways, and this observation can be exploited with a cooperative data sharing approach. Secondly, even among nodes using similar sensor hardware for localization, the data quality may vary significantly, again, even when sensors are positioned in close proximity to each other. Consider a case where two nodes are localized using GPS. If one receiver is able to acquire a fix and the second is not (as frequently occurs in practice), the first may have an accurate position estimate that can be used to improve the second, which may only have a rough estimate of position using cached information. Similarly, if both have a position estimate, but one is clearly more accurate than the other, the node with lower quality location data can leverage the higher quality data of the other. Its position can then be improved through averaging or another combination technique.

1.4 Potential Applications

This section describes a sample of real-world applications whose performance and effectiveness could be improved through a cooperative localization technique.

Personal Navigation. The most obvious application of improved localization of mobile computing devices is personal navigation systems which allow users to identify their location in relation to geographic features represented as objects on a map. On a human scale, absolute error, uncertainty, and variability in position are far more pronounced and have more significant consequences than when localizing much larger objects such as vehicles. For instance, localizing an individual operator to a walking path requires much greater precision and accuracy than localizing an automobile to a four-lane highway.

Geographic Forwarding. Many applications or protocols used in mobile ad-hoc networks not only require localization of the nodes, but bad localization can have serious performance implications. For example, routing algorithms based on geographic forwarding can suffer from suboptimal routes, misrouted packets (Abu-Ghazaleh et al., 2005), or, if poor localization is malicious in nature, wormhole attacks (Perrig et al., 2001). Several techniques have been devised to verify location accuracy (Abu-Ghazaleh et al., 2005; Sastry et al., 2003), but these are not general purpose algorithms.

Search and Rescue. Search and rescue operations can be greatly aided if handheld computing devices self localize. This can already be accomplished with cell phones as they handshake with a nearby tower (Sandoval, 2007), but more accurate localization can facilitate speedier rescues. In particular, data from cell handshakes are not readily available to searchers on the ground, nor are such data always available at all.

Sensor Networks. A wide variety of sensor networks has emerged for a broad application space, to include such disparate applications as habitat monitoring, infrastructure protection, and battlefield awareness (Zhao and Guibas, 2004). These types of systems require localized nodes for purposes of providing geographically meaningful data, and many require location information for either routing or collaborative tracking tasks (Bachrach and Taylor, 2005). Inaccurate positioning, particularly when coupled with a relatively low deployment density, could easily cause the system to be ineffective at the task for which it was designed, built, and deployed.

Moreover, many other types of applications which are dependent on data replication, such as peer-to-peer networks, exploit spatial locality to achieve greater

communication efficiency. The work presented in Chapter 5 of this dissertation is one example. All of these types of applications rely on robust localization, yet are frequently deployed to or operated in places lacking infrastructure to easily facilitate positioning. Although reliable localization is generally assumed in many works, that is frequently not the case encountered in practice, which in turn leads to overall poor application performance.

In this dissertation, practical experiences and lessons learned with commodity hardware and software are presented in the context of a mobile ad-hoc network testbed, along with ways in which mobile applications can detect and deal with sensor measurement error. Focus is on the propagation of error when location data is combined from multiple hardware sources using an array of sensors attached to connected devices. The contribution is an exploration of error reduction techniques in unreliable mobile sensor networks and the further exploitation of robust localization information.

1.5 Overview of Dissertation

This dissertation will examine whether the data available in systems composed of commodity hardware and software are sufficient to determine whether error-prone sensor data from various sources can be effectively used alone or in concert to improve positions in real time. Figure 1.2 illustrates the focus of each chapter and its respective place in the architecture of a typical mobile system.

At the device layer, raw sensor data are retrieved from the hardware. This data may include some estimate of localization error as a function of distance. Additionally, such data may be adjusted based on an error model created for each particular sensor. At the next higher layer, location data are shared among

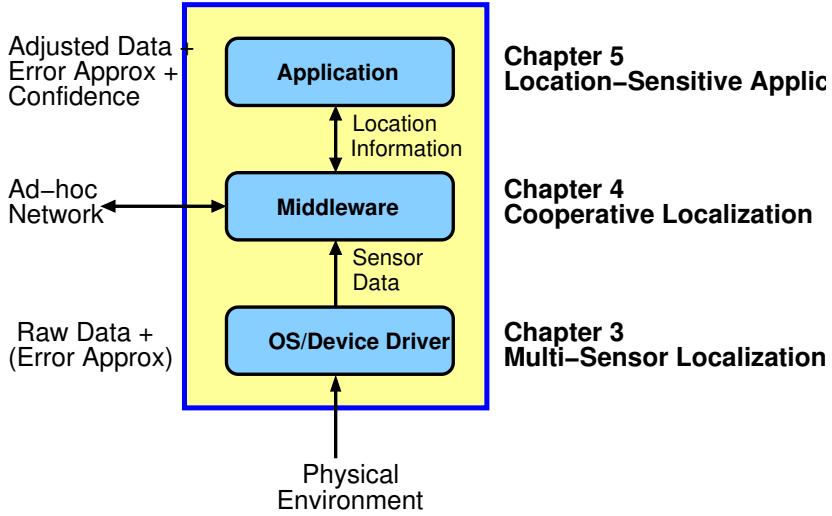


Figure 1.2. Overview of Dissertation

connected nodes and combined to improve accuracy and precision. This corrected data, along with an adjusted error estimate and confidence level, are then funneled to the application layer.

Chapter 2: Related Work. Typically localization in mobile networks or sensor network is accomplished using GPS, with the most common limiting factor its relatively high deployment cost or energy consumption rate, not positioning error. Furthermore, most works in wireless sensor network localization assume a static network topology. This chapter presents an overview of existing research into localization of sensor networks and related areas as pertaining to evaluation of mobile networks, paying particular attention to those addressing localization in the presence of mobility. It also presents the current state of the art in mobility prediction techniques as well, an area expanded on in Chapter 5.

Chapter 3: Multi-Sensor Localization. Much of the study of sensor error involves understanding the “good” cases, i.e., looking at the cases in which sen-

sors perform well, oftentimes with a goal of ensuring the data is reliable enough for a particular application. Ignoring the behavior of error does not help application designers determine how to handle boundary cases, which occur frequently enough in practice to be noticeable by the user. Thus, such error warrants investigation and should be accounted for in system design. Frequently, sensor error manifests in unintuitive ways, and understanding the manner in which common sensors fail in practice helps sensor hardware designers build better devices and application designers build more robust location-sensitive applications. Furthermore, the study of sensor data in this chapter is fundamental to the development of valid error models which constitute a large part of the evaluation efforts in this dissertation.

Chapter 4: Cooperative Localization. The vast majority of localization schemes assume that anchor nodes always have low-error localization, and other nodes estimate their own position (or, alternatively, have their positions estimated for them) based on the anchors' positions. This chapter explores ways in which location data of varying quality obtained by different techniques can be combined to improve the localization of the system as a whole. In this work, the term sensor includes such nontraditional means of detection and measurement as wireless network connectivity. Evaluation of these cooperative approaches is presented through a series of scenarios in which sensor data is shared among collaborating actors, but may or may not be accepted based on the combination method and both the relative quality of the data and the confidence in that quality.

Chapter 5: Location-Sensitive Applications. Chapter 5 demonstrates the importance of robust localization in mobile networks by presenting a method for replicating sensor data which relies on mobility prediction to maximize avail-

ability of directly connected nodes. In this method, mobility prediction is based on a subset of location history, which is then extrapolated and combined with a model of wireless signal propagation in order to estimate a *window of opportunity* in which nodes can exchange data. This chapter explores the semantics of peer node selection, data placement, recovery, and data management and further discusses the effect of localization error on selection and placement, and presents a simple weighting technique to account for such error.

CHAPTER 2

RELATED WORK

2.1 Sensor Evaluation and Mobile Ad-Hoc Network Testbeds

The need for mobile ad-hoc network (MANET) testbeds has gained widespread recognition in the literature only relatively recently, but research into multi-hop wireless networks using physical prototypes has its roots back in the early 1970s with the advent of the ALOHA project at the University of Hawaii (Abramson, 1970). ALOHA was most noteworthy in that it successfully demonstrated the feasibility of broadcast messaging over wireless channels, albeit in a single-hop network that would be considered quite primitive by today's standards.

The first multi-hop ad-hoc network that incorporated some degree of mobility among nodes was introduced only a few years later when DARPA began work on PRNET (Jubin and Turnow, 1987; Kahn et al., 1978). PRNET was originally designed to study the feasibility of packet-switched, store-and-forward wireless communications (Kahn, 1977). Incidentally, in the earliest days of mobile networks, wireless networks were given the term *packet radio networks*, from which the name PRNET was derived.

As work progressed on PRNET over the following decade, it led to the introduction of the SURAN project in 1983 (Beyer, 1990), whose goals were to evaluate more sophisticated routing protocols, allow for far greater scalability than previously realized, and permit a degree of survivability in the presence of electronic

attacks in military-type applications. SURAN was the first prototype of a modern mobile ad-hoc network.

As study of MANETs grew over the subsequent decades, three fundamental approaches to evaluation emerged and evolved. Conceptual ideas can be tested and validated through simulation, emulation, and through scientific experiments using the systems as a whole or their separate, individual components. Each of these approaches has its own strengths and weaknesses.

2.1.1 Simulation

Simulation is arguably the most prevalent approach to evaluating MANETs today, and with good reason. Simulation tools provide a flexible and effective means to quickly evaluate attributes such as scalability and node mobility over great distances which would be very difficult or impossible to evaluate with physical implementations. Simulation environments such as NS-2 (Information Sciences Institute, 2006) and GloMoSim (Zeng et al., 1998) are by far the most commonly used, with rich feature sets and active community support, although legions of simulation tools built specifically for individual projects exist as well. All of these tools, however, suffer from the same fundamental limitation as any other simulator. Results are only as valid as both the assumptions made about the behavior of external factors and the complexity and granularity of the models. Although their discussion pertains primarily to routing protocols, Kiess and Mauve point out that qualitative rankings of systems under evaluation can largely depend on the particular simulator as well as both the validity and complexity of the models used in the evaluation (Kiess and Mauve, 2007).

2.1.2 Emulation

Additional realism can be injected into a test environment through the use of emulation. Emulation simply augments a simulation tool with physical hardware and production software, much like hardware-in-the-loop simulations familiar to designers of embedded systems. Emulation using ad-hoc networks has a fairly lengthy history as well. Much of the SURAN project was evaluated through emulation (Beyer, 1990), and its use has expanded significantly in subsequent decades.

As much of the research into MANET design involves the study of network protocols, a large number of existing emulators were designed and built for such a purpose. There are numerous physical emulation environments in existence which are intended to evaluate network-layer protocols, and do so by modeling the different network layers to varying extents. These emulation tools can be categorized into two groups: those which model only the physical layer, e.g., (Chin et al., 2002; De et al., 2005; Girod et al., 2004; Judd and Steenkiste, 2002; Levis et al., 2003; Raychaudhuri et al., 2005; Sanghani et al., 2003; Vaidha et al., 2004), and those which model both the physical and MAC layers (Chao et al., 2003; Heisenbüttel et al., 2005; Maltz et al., 1999; Matthes et al., 2005; Nordstrom et al., 2005; Zhang and Li, 2002a). While emulation in the latter cases typically involve MAC-layer filtering in a decentralized fashion, packet routing and delivery can also be emulated through the use of a centralized simulator (Ke et al., 2000). WHYNET (Varshney et al., 2007) takes emulation of networking components a step further, and facilitates simulation of software and hardware systems, which may include the entire operating system and applications.

Lastly, emulation can be accomplished through a trace-based mechanism. In

network emulators, a trace-based approach adds realism by modeling packet loss and delay based on the results of prior empirical evaluation of real wireless networks (Liu et al., 2004, 2005; Noble et al., 1997). The use of empirical evaluation and experimentation is particularly relevant to this work, and their application in mobile sensor networks is discussed next.

2.1.3 Experimental Trials

Models developed for simulation and emulation tasks can either be purely theoretical or based on empirical data collected via experiments. An implementation of a mobile network provides a convenient platform to conduct experimental trials and gather empirical data. This data can be used to develop or refine analytical models for use in simulation, which adds to the validity of such models, or as feedback into agile system development processes. Additionally, results of live system tests by themselves can be used for evaluation purposes, augmenting simulation or emulation results. The foundation of any component-based engineering approach, as described earlier in Section 1.2, lies in understanding the capabilities, limitations, and interfaces of individual components. Therefore, this section begins with a discussion of component-wise experimental evaluation.

2.1.3.1 GPS Error Analysis

When approached with the idea of independently evaluating commercial GPS receivers for accuracy, a skeptical observer might ask why the accompanying data sheets are insufficient. The answer is twofold: First, technical data describing GPS receiver performance are not uniformly reported across manufacturers (Buick, 2002). Robust error handling in a system which incorporates more than one

model of GPS receiver hardware, as TeamTrak does, might prove considerably more challenging to design and successfully implement if component performance is measured inconsistently across devices. Second, even GPS receivers of the same model or manufacturer may perform quite differently in different environments or when employed in specialized applications such as agriculture. These specialized applications often have unique requirements and performance demands not likely to be fully anticipated by sensor hardware designers or that paints a pessimistic view of the receiver's capabilities. Examples of such include mounting on rapidly moving vehicles such as aircraft (Shannon et al., 2002; Taylor et al., 2004; Thomson and Smith, 2006) or deployment in dense urban or wooded environments.

In terms of empirically-based error models for GPS, several experimental GPS accuracy evaluations are published in the literature. Additionally, specific guidelines for evaluating GPS accuracy exist (Han et al., 2004; Institute of Navigation Standards, 1997), but many of these analyses were accomplished prior to the U. S. Government's disabling of so-called *selective availability*. Selective availability provided a far less accurate position estimate in commercial-grade receivers, and so many of the earlier published accuracy data is no longer meaningful. Similarly, several evaluations were conducted under much different environmental conditions than are of interest in this work, such as in locations with wide-open sky views (Rupprecht, 2007; Wormley, 2007) or using additional correction techniques such as differential GPS, which maximize the accuracy of GPS, but may not always be representative of real-world conditions. Furthermore, while such experiments yield acceptable statistical models of GPS error distributions, results frequently lack confidence values for specific error values or ranges, which makes designing systems sufficiently robust to account for GPS error somewhat more challenging.

2.1.3.2 Dead Reckoning Systems

Regardless of the accuracy of GPS measurements, cases in which no position is available are frequently encountered. While indoor operation is the canonical example of this used in many works, many other situations arise in which GPS accuracy is poor, to include outdoors. For those cases in which GPS is highly unreliable or unavailable, localization can be accomplished via *dead reckoning*.

The Merriam-Webster dictionary defines dead reckoning as:

The determination without the aid of celestial observations of the position...from the record of the courses [traveled], the distance made, and the known or estimated drift.

and was originally applied to navigation in ships or airplanes. This definition aptly applies to pedestrian navigation as well, hence the more slightly generalized definition used here.

Research into localization using dead reckoning has been ongoing for years. Techniques for implementing dead reckoning systems and their associated challenges have been nicely described in the literature (Amundson, 2006). As the authors point out, using wheeled robots reduces the error in calculating specific movements significantly, so the real challenge with dead reckoning lies in building systems based on effective step detection and stride length estimation for non-wheeled platforms such as human operators, whose steps are not rigidly mechanical in nature. These types of systems have not yet been widely built or explored, particularly those applied to wearable systems.

Dead reckoning systems mounted on a walking platform, despite the inherent difficulty, is not a new concept. The CMU Ambler (Roston and Krotkov, 1991), first developed in the late 1980s, is a six-legged walking robot that relies on the kinematics of its mechanical legs to determine steps taken and detect error condi-

tions such as slips. It had no GPS or other external, on-line means of localization or correction of dead reckoning errors, nor did it use electronic sensors such as accelerometers for motion detection.

Attempts have been made to apply dead reckoning techniques to human operators. NavMote (Fang et al., 2005) is a pedestrian dead reckoning system that relies on a compass/accelerometer pair which operate in tandem to determine changes in position resulting from each step taken by a human operator. NavMote is intended to be used more as a tracking system to provide location data to a centralized command post, rather than providing real-time status displays to the operator. NavMote does not rely on inter-node communications for data sharing, nor does it provide localization by means other than dead reckoning or manual adjustments.

2.1.4 Existing Testbeds and Deployed Systems

Actual implementations of mobile ad-hoc networks have been somewhat limited, particularly those requiring or facilitating human interaction. Generally, focus has been more on autonomous operation of wireless sensor networks in remote locations. Because of the specific application requirements for these types of systems, the body of work in wireless sensor network testbeds intended for human intervention or interactions is far more limited. The remainder of this section briefly surveys some of the works in the area of sensor networks.

2.1.4.1 Wireless Sensor Networks

Wireless sensor networks are nearly always designed and built to target a very specific application domain, e.g., (Arora et al., 2005; Burghardt et al., 2002; Cerpa

et al., 2003; Corr and Okino, 2000; Correal and Patwari, 2001; Delin and Jackson, 2000; He et al., 2004; Marcy et al., 1999; Mason et al., 1995; Odell and Wright, 2002; Yarvis et al., 2002; Zhang and Li, 2002b). A survey by Xu categorized the predominant classes of current sensor network applications as habitat monitoring, environmental observation, and health monitoring (Xu, 2003). Certainly many others exist, a sampling of which will be mentioned in this section.

Among deployed systems designed for habitat monitoring, perhaps the most widely cited is the GDI sensor network, constructed from low-power motes and deployed to Great Duck Island, Maine, for the purpose of monitoring behavioral patterns of storm petrel (Mainwaring et al., 2002). ZebraNet (Juang et al., 2002), as the name implies, uses GPS receivers affixed to zebras to facilitate the study of behaviors such as migration patterns and inter-species interactions. In ZebraNet, sensor data, to include data in addition to GPS location, is shared between connected nodes within wireless range using a flooding mechanism similar to many routing protocols. PODS (Biagioni and Bridges, 2002) is a habitat monitoring system that collects both imagery and weather data for the purpose of understanding why certain endangered plants grow only in specific locations. Similarly, sensor networks exist for herding cattle (Butler et al., 2004) and so on.

Of those sensor networks intended for human interaction, most are designed to target very specific applications such as health monitoring (Falck et al., 2006; Hester et al., 2006; Oliver and Flores-Mangas, 2006) or measuring physical motions of the body for a specific activity such as dancing (Aylward et al., 2006).

However, there are a number of sensor networks that have been built targeting broader application. For example, Eco (Park and Chou, 2006) consists of wearable devices that are extensible through the addition of sensor equipment, but the

functionality is limited by the physical dimensions of each device, which does not permit interaction with the user. Similarly, TeamTrak (Hemmes et al., 2007) is extensible through the use of plug-in commodity sensor hardware with standard interfaces, but provides a capability for interaction not only with a local user, but other users connected via ad-hoc networks.

2.1.4.2 Mesh Networks

In contrast to MANETs, mesh networks are typically deployed in static topologies and generally provide network connectivity over a wide area at a lower cost than would be obtained with multiple wired access points. While mesh network designs usually do not incorporate sensor hardware to any significant extent, they are commonly the most mature implementations of wireless multi-hop networks.

These types of systems are generally intended to provide multi-hop Internet connectivity over a wide area (Robinson et al., 2005). The roofnet project by MIT (MIT, 2008), perhaps the most widely known mesh network, provides such connectivity over the length of several city blocks. Similarly, several metropolitan areas have implemented prototype mesh networks as part of ongoing research efforts. Melbourne, Australia (Melbourne Wireless Committee, 2008) and Leiden, the Netherlands (Wireless Leiden Foundation, 2008), are but two examples.

An evaluation of 802.11a wireless Ethernet performance on a mesh network of Windows XP machines was accomplished in (Draves et al., 2004) and revealed a significant amount of asymmetry (as determined by measured bandwidth) in the bi-directional links on a single hop. Use of this testbed demonstrated that multiple wireless radios using wireless Ethernet on the same node tends to cause interference among the wireless cards, resulting in a significant loss of available

bandwidth. This result was confirmed in (Bruno et al., 2005), using a testbed composed of Linux machines with multiple wireless radios installed on each node.

2.1.4.3 Mobile Testbeds

It is very common for testbeds to involve the use of robots or vehicles to achieve mobility (Castro et al., 1998; Liu and Pang, 1999; Pang and Liu, 2001). Such platforms provide much greater stability than can be achieved with a wearable or handheld device, which makes the use of many types of sensors such as accelerometers and compasses far more straightforward, as they are not subjected to noisy motion patterns to the same extent as human-mounted or human-portable sensors.

Finally, it is important to note that a number of testbeds, such as mLab (Karygiannis and Antonakakis, 2006), have attempted to bridge the gap between simulations and field tests by including wireless network connectivity in the platform. However, these systems are generally installed indoors and rely on simulation to evaluate interactions due to mobility.

Netbed (White et al., 2002) is a mobile wireless network testbed that encompasses a range of evaluation techniques by facilitating evaluation using any combination of simulation, emulation, and experimentation (White et al., 2003). Localization in Netbed is limited to the possibility of using commercial GPS receivers with differential correction, but the test scenarios envisioned by the authors involve both urban and semi-indoor, i.e., inside a vehicle, environments, which may not provide the most accurate positioning. Netbed does not account for localization error in any significant way.

2.2 Localization Techniques

Although localization is a concern primarily in mobile networks, where node movement is not only common, but expected, there has been work done in the area of localization in traditional wired networks, although the approaches taken are much different. A common method for determining location of wired machines is through correlating latency with distance (Ledlie et al., 2007; Wong et al., 2007). Alternatively, desktop workstations may be outfitted with wireless monitors to determine proximity to wireless access points in known locations (Chandra et al., 2007). Note that these approaches are limited to static wired networks, and generally do not work well in mobile environments, due to low precision.

There has been a great deal of work in the area of localization in wireless sensor networks, to include several surveys, e.g., (Langendoen and Reijers, 2003). Much work has been done with standalone dead reckoning techniques, which include systems affixed to human operators such as NavMote (Fang et al., 2005). While dead reckoning techniques comprise a substantial portion of this work, the goal is not simply to build a better dead reckoning system; rather, to understand the use of cooperative data sharing among remote peers to improve location accuracy given estimated positions of varying quality.

Cooperative localization is not a new idea, but the traditional approach assumes that a subgraph of the network topology has good localization a priori through either an active GPS signal or predetermined ground truth location information (Bulusu et al., 2000; Hu and Evans, 2004; Lorincz and Welsh, 2005; Niculescu, 2001; Savarese et al., 2002; Savvides et al., 2002, 2004; Sun and Guo, 2004). The location of such “beacon” nodes is used, along with some method for determining the separation distance between them, to estimate the position of the

others.

A distributed method for localization is presented in (Priyantha et al., 2003), but this method assumes a static configuration in which no physical node locations are known a priori. The focus of their work is more on determining the topology of the network, i.e., relative localization, which can then be translated to a specific coordinate set, and also cannot account for node mobility. Like other related localization schemes this method requires an accurate distance estimation or measurement for each pair of connected nodes. Similarly, convex position estimation techniques can be used (Doherty et al., 2001), but this requires a homogeneous physical implementation. Moore et al (Moore et al., 2004) proposed an algorithm for relative localization without anchors in the presence of noisy distance estimates, but requires significant computational overhead, which may not be suitable for existing hardware.

The Spotlight technique (Stoleru et al., 2005) uses a unique event-based approach to localization in which a base station estimates position based on the time a particular node reported an event's occurrence, but this method is centralized, and therefore not suitable for use in MANETs, and further requires deployment of additional infrastructure, to include an airborne vehicle which signals the nodes on the ground. It is also not clear whether this approach would be suitable for cluttered environments or deployed over a very small geographic area.

Traditional approaches to distributed localization generally make assumptions about the quality of location data. First, instead of assuming nodes are either localized accurately, i.e., via GPS, or not, a range of location qualities, e.g, none, dead reckoning, GPS, fixed, etc., should be assumed. Second, the presence of at least some nodes with high-quality location information, i.e., anchor nodes, is

assumed, particularly for systems deployed indoors and in which all nodes either use dead reckoning or have no initial positioning information. In cases in which GPS is not available for any node, the quality of the location provided by an anchor used to laterate positions of other nodes may be substantially less than that of GPS. Third, targeting work for use in human-scale mobile systems implies that location information may be at best highly dynamic and often unreliable.

Despite the importance of accurately estimating distance between nodes, there are few available options which are both reliable and adaptable to both indoor and outdoor environments. RSSI is a commonly cited approach to estimating distance, but its suitability under noisy outdoor conditions is highly uncertain. RADAR (Bahl and Padmanabhan, 2001) is a system which uses RSSI to estimate distance to within a few meters accuracy, but by design is generally limited to indoor environments. A number of range-free schemes have been devised (He et al., 2003) that rely on a connection-based algorithm for distance estimation. DV-HOP (Niculescu, 2001) is one example that does not require range measurement, but does assume isotropic RF signal propagation and in the worst cases, which are a function of network topology, is subject to substantial error.

2.3 Mobility

It has been observed that node mobility can significantly affect the performance of routing protocols (Bai and Helmy, 2004). Both the importance of simulation and emulation in evaluation of MANETs and the close relationship between localization and mobility dictate the effective use of mobility models. Camp et al surveyed various mobility models used in simulation of wireless networks and organized them into two broad categories: trace-driven and synthetic models (Camp

et al., 2002). Furthermore, Pazand and McDonald presented an overview and critique of such models (Pazand and McDonald, 2007). Rather than discuss all models in depth, the interested reader is encouraged to review the existing work. In this dissertation, focus is limited to the more widely adopted mobility models.

2.3.1 Mobility Models

The most frequently used, random-based mobility model is the Random Waypoint model. This model has been the foundation for many MANET evaluations, primarily due to its simplicity. Two important and frequently used variants of Random Waypoint are the Random Walk and Random Direction models, each of which is discussed separately.

2.3.1.1 Random Waypoint

Perhaps the most commonly implemented mobility model, the Random Waypoint model (Johnson and Maltz, 1996) moves a node at a constant speed to a randomly selected destination point within the boundaries of the simulation field, then pauses for a specified time T_{pause} , then chooses a new waypoint, and so on. The speed of each node is selected randomly over the interval $[0, V_{max}]$. The relationship between these two parameters, T_{pause} and V_{max} , is primarily responsible for the behavior of the system. Setting V_{max} too high relative to T_{pause} causes the system to become highly dynamic and somewhat unstable. To quantify the overall *nodal speed*, which can facilitate comparisons between instances of the model based on the stability of the topology, Johansson et al propose a *mobility metric*,

based on relative speed between nodes, defined as:

$$\bar{M} = \frac{1}{|i,j|} \sum_{i=1}^n \sum_{j=i+1}^n \int_0^t RS(i,j,t) dt$$

where

$$RS(i,j,t) = |\vec{V}_i(t) - \vec{V}_j(t)|$$

and $|i,j|$ is the number of distinct node pairs in the system, n is the number of nodes in the network, and t is the current simulation time (Johansson et al., 1999). The Random Waypoint model is found in the most popular network simulation environments, NS-2 (Information Sciences Institute, 2006) and GloMoSim (Zeng et al., 1998), as well as in the simulation tools developed for the work discussed in Chapters 4 and 5.

2.3.1.2 Random Walk

Random Walk is a widely used mobility model with numerous enhancements (Akyildiz et al., 2000). This model can be thought of as a specific instance of Random Waypoint in which $T_{pause} = 0$ and a new speed and direction selected after a designated time interval, rather than at a specified point. This model is considered *memoryless*; it only requires the last position of a mobile operator and a randomly chosen direction and stride length at the end of each time interval. While this model is conceptually very simple and commonly used, its realism is suspect since in practice nodes tend to make very sudden stops and abrupt changes of direction.

2.3.1.3 Random Direction

The Random Direction model borrows from both the Random Walk and Random Waypoint mobility models in that nodes select a direction and speed randomly, but continue movement in a single direction until reaching the boundary of the simulation field. As in Random Waypoint, a node at a field boundary pauses for some specified time, then randomly chooses a new random speed and direction. This model was developed due to observations of the previous models that demonstrated the distribution of nodes at steady state was highly non-uniform; nodes tend to converge in the center of the simulation field (Royer et al., 2001). The Random Direction model was proposed to alleviate this problem.

2.3.2 Mobility Prediction

While there is an abundance of research in the area of mobile networks, work specifically in the area of mobility prediction is much more limited. Mobility prediction is frequently based on trace data; using the current direction and speed based on previous locations to extrapolate future positions. There are a number of works which attempt to predict future availability by tracking position history, but routing protocols generally do not incorporate system state information into routing determinations, or combine system state with mobility data.

In (Larkin, 2005), mobility is predicted by building and maintaining a table to track estimated periods of connectivity loss, but the focus of that work is limited to disconnections due solely to mobility. While this work bears some similarity to theirs, to include the ability to use the prediction method independent of the routing protocol, their work assumes that availability of nodes does not change due to diminishing system state, and therefore the network is static in terms of

the total number of nodes in the system.

The method presented in this work is most closely related to that proposed by Pascoe (Pascoe et al., 2007). Whereas in their work, the prediction is intended to be used for estimating the amount of overhead incurred in both unicast and multicast routing protocols as routes are broken due to mobility, this method is designed to select a single hop route with the largest predicted availability for purposes of short-term data storage and retrieval of transient data. Furthermore, their work does not assume link breakage due to system failures.

Also closely related is the work proposed by Su (Su, 2000), which uses GPS location information to predict the future location of nodes moving independently. Additionally, parameters such as radio propagation range are known *a priori*. Like other routing protocols, this approach does not account for availability constraints other than mobility or localization error, and does not address the specific problem of data recovery.

Other methods of predicting availability do so by measuring signal strength, with a diminishing signal portending a link disconnection. Given experiences with the directional nature of many wireless antennas, using detected signal strength alone may not be the most appropriate factor for predicting mobility in many types of applications. Examples that employ a signal strength measurement for availability estimation are (Chellappa-Doss et al., 2003) and (Goff et al., 2001).

CHAPTER 3

MULTI-SENSOR LOCALIZATION

3.1 Introduction

Chapter 2 described a wide array of mobile testbeds along with existing sensor-based localization techniques. A number of these testbeds are constructed using commodity hardware and software components. In many of these cases such components are selected for the express purpose of rapid prototyping, which significantly accelerates system development. In recent years, increased use of off-the-shelf hardware and software has become commonplace among system developers, both in research projects as well as production systems built by industry. The rate of adoption is due to the substantial benefits realized in terms of both cost savings and decreased development time. Trends in systems development techniques suggest even greater reliance on the use of reusable components designed and built by multiple vendors for general application, with current practices trending towards a greater focus on system integration, rather than design and implementation from scratch (Rising, 2001). Additionally, the National Science Foundation has long advocated the use of open hardware and software in research testbeds (Aiken et al., 2002).

Despite the obvious advantages of integrating off-the-shelf sensor devices, their use presents some unique challenges as well. Low-cost, mass-produced sensor hardware may be in some cases of inferior quality when compared to specially designed,

custom fabricated hardware, and therefore may provide data of uncertain accuracy (Refan et al., 2003), particularly when applied to new systems for which the sensors are not specifically optimized. However, the feasibility of large-scale deployments commonly envisioned by application designers depends in large part on the use of such hardware. The challenge, then, is presenting applications with data of sufficiently high quality, or at least recognizing cases in which sensor data has significant error, while simultaneously accepting the cost constraints which necessitate the use of commodity sensors.

This chapter examines the data reported by an array of low-cost sensors used in the TeamTrak testbed implementation, a necessary precursor to the distributed localization methods discussed in Chapter 4. Specifically, focus is on collecting and observing sensor data used for localization with the larger objective the characterization of those failure modes which may be encountered under normal operating conditions. Failure modes of interest may result from either known limitations of the hardware, e.g., compass roll, or unexpected but repeatable variations in the sensor data which may be attributed to any number of causes. In turn, error models can be used to improve localization for a single node, and can then be used collaboratively to detect conditions under which sensor data may be unreliable and facilitate correction. The goal of this work is not to build a flawless navigation system per se, as many concurrent works are progressing towards that aim, but instead to understand the nature of sensor error, particularly the more esoteric cases which might significantly affect navigation systems constructed from low-cost components.

3.2 The TeamTrak Mobile Testbed

The overarching purpose of TeamTrak is to evaluate research ideas in mobile distributed computing without reliance on specialized or custom fabricated hardware. The hardware testbed consists of inexpensive commodity equipment to the greatest extent possible. The research prototype consists of 32 Lenovo X41 Thinkpad tablet computers running Windows XP and eight HP iPAQ hx2795b PDAs running Windows Mobile. A standard ANSI Z89.1 Class C safety helmet provides a convenient platform for mounting mobile sensor equipment.

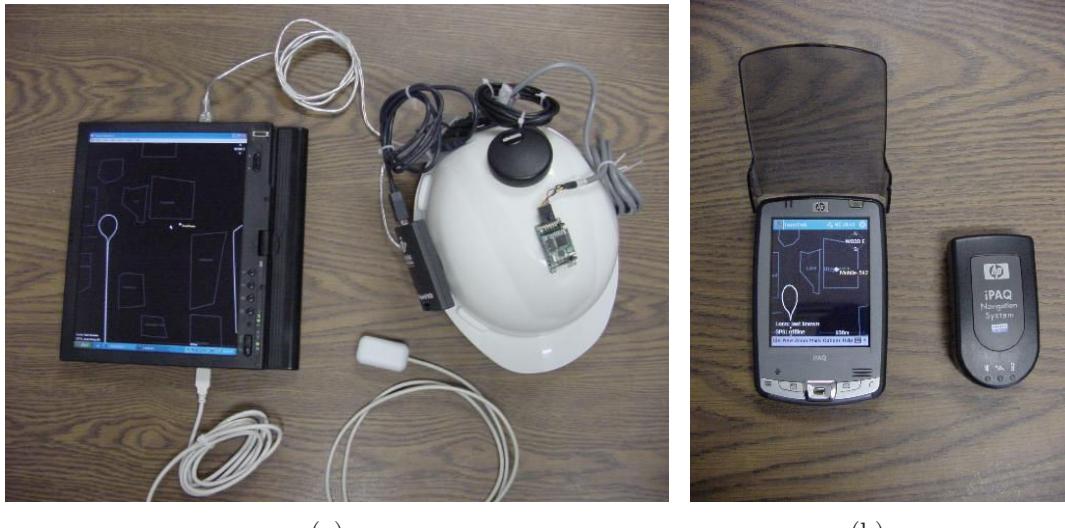
In its current incarnation, the TeamTrak framework incorporates three main types of localization sensors, all obtained from commercial sources and implemented with standard hardware and software interfaces:

- *GPS Receiver:* GPS is the primary means of localization in TeamTrak, but GPS has limitations exhibited in urban environments beyond what is widely published in the literature, and in such cases errors observed in the output of commodity GPS hardware may differ significantly from well-publicized error rates of GPS. These limitations are not inherent to any single manufacturer or model of receiver, as both the Garmin GPS-18 USB and the HP iPAQ BT-308 receivers (Garmin International Inc, 2005; Hewlett-Packard Company, 2003), both of which are part of the baseline TeamTrak implementation, have demonstrated significant positioning error at times, but in practice do so in different ways and to different extents. It should also be noted that urban canyons do not need to be particularly dense in order to experience substantial error.
- *Digital Three-Axis Accelerometer:* In the absence of an accurate GPS signal, or in the event of a low power state where operating a GPS receiver con-

tinuously might be impractical, localization can be accomplished through *dead reckoning*, defined in Section 2.1.3.2. The accelerometer can be used to detect individual footsteps taken by a human operator, and coupled with an estimate of stride length, may be used to determine the incremental distance traveled. When employed in tandem with a digital compass used to measure heading, accelerometer data can be used to determine position. Naturally, this method assumes the process begins from a known initial location, as dead reckoning is only capable of determining changes in position, not absolute location. In this work data from the SparkFun SerAccel v5 digital accelerometer is examined.

- *Digital Compass (Two- and Three-Axis)*: Currently, TeamTrak employs both the PNI V2Xe two-axis digital compass and the OceanServer OS3500 three-axis compass. Mounted on a human operator, reported headings from both compasses are influenced by pitch and roll to varying degrees, and in some cases the effects can be quite dramatic. Pitch and roll variations, which are a direct consequence of mounting on an unstable platform, are far more pronounced when triggered by a human operator as compared to a more stable robot or vehicle, and therefore must be accounted for in any personal dead reckoning navigation system.

The communications medium is wireless Ethernet (IEEE 802.11b) in ad hoc mode and no base station. The standard Windows IP configuration is used: each node detects an available network, then negotiates a link-local RFC 3927 IP address. Although wireless Ethernet is not at all optimized for outdoor peer-to-peer communication, it is supported by standard consumer electronics. Thus, TeamTrak needs no specialized hardware, facilitating the possible addition of a variety



(a)

(b)

Figure 3.1: TeamTrak Hardware Components

of computing devices.

Augmenting the basic hardware platform are sensors connected by either USB or serial port. GPS data are provided through Garmin GPS-18 USB GPS receivers, or in the case of the PDAs, an HP iPAQ BT-308 Bluetooth receiver. In addition to GPS, the platform includes the PNI V2Xe digital 2-axis compass, the Watchport/V2 digital camera, and SparkFun SerAccel v5 digital accelerometer for localization. These are discussed in detail in Chapter 3. The overall system architecture easily allows expansion through additional sensors connected via standard interfaces. Figure 3.1 shows the devices that comprise each TeamTrak node in each configuration.

The software infrastructure consists of two parts: an application and a simple routing protocol for sharing information. The TeamTrak software is a standalone application designed to build and run on multiple platforms. It also includes several display modes, depending on the localization technique used or the specific application scenario. If a GPS receiver is connected and receiving a live signal,

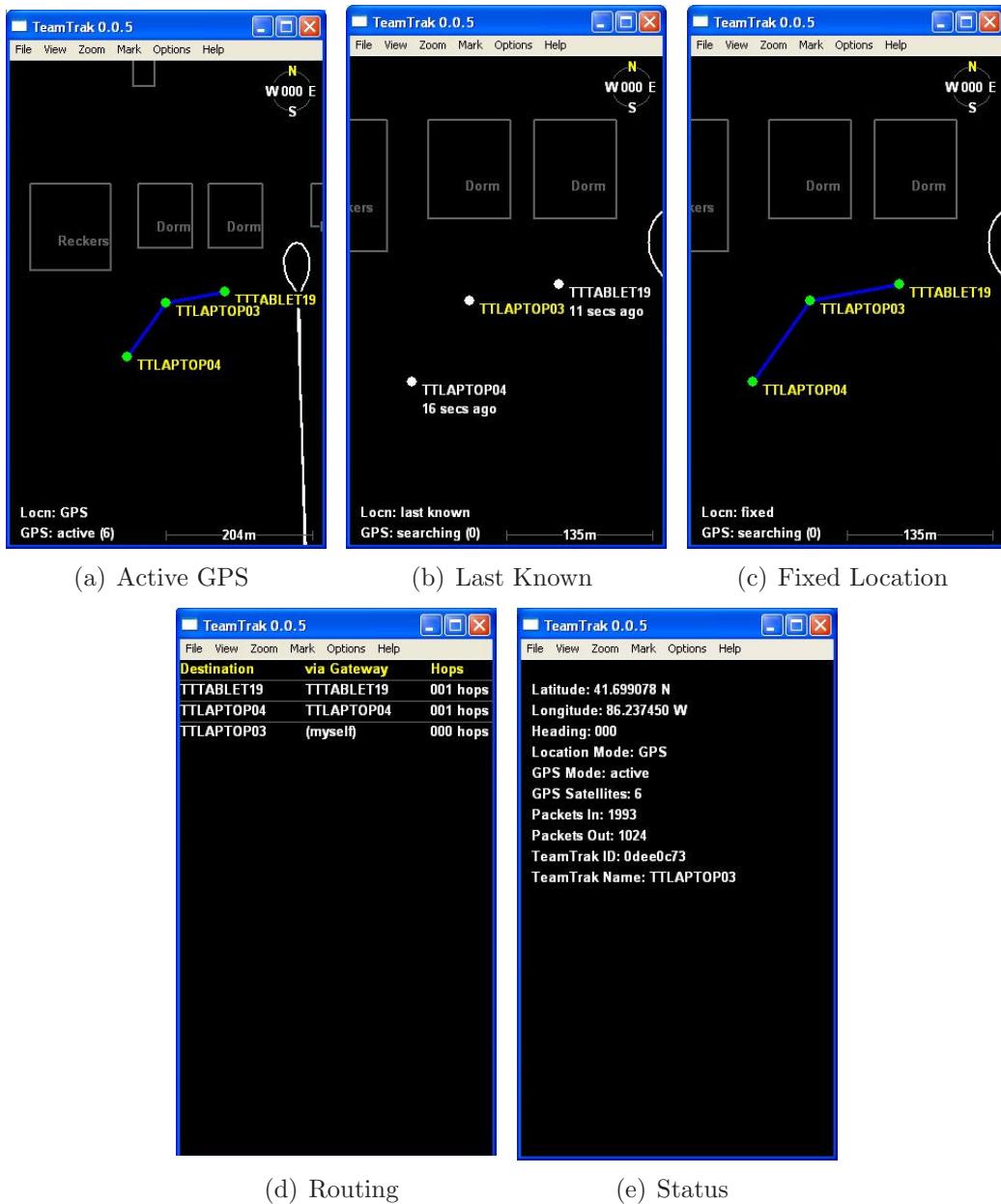


Figure 3.2: TeamTrak Display Modes

the display indicates such in the lower left corner, as shown in Figure 3.2(a). If the GPS signal is lost, the display shows the node at its most recent known location, as shown in Figure 3.2(b). Alternatively, if no GPS signal is available, and dead reckoning localization is being used, the display is updated appropriately. Similarly, as other nodes move and become disconnected, the display will continue to show their last known location, but changes the symbol used to represent them and also indicates the length of time since a packet was last received from each node. Location may also be set manually for testing purposes or correcting GPS error in cases where the exact location of a node is known. In this case, the display indicates the location is fixed, as shown in Figure 3.2(c). Additionally, the display shows compass heading for each node and is capable of displaying live video imagery from the digital camera. Figure 3.2(d) shows the routing table, and 3.2(e) the current status of the local device.

Data sharing in TeamTrak is accomplished through the use of a simple distance-vector routing protocol similar to RIP (Malkin, 1998), but does not employ the multiple packet types found in DSDV (Perkins and Bhagwat, 1994). At 1-second intervals, each node broadcasts the contents of its routing table, including sensor data for each entry, to all other connected nodes. The selection of a proactive routing protocol is primarily for simplicity and is not without some well-known limitations. Stabilizing the routing table initially requires some time as the data must propagate through the network using a flood-like mechanism. This further implies that clearing stale data involves delays as well, which leads to the well-known counting to infinity problem (Leon-Garcia and Widjaja, 2000). The approach taken in the design of TeamTrak for handling stale data is to make it persistent until manually cleared from memory by the operator.

3.3 Experiences With the Global Positioning System

In practice, even the highest quality sensor equipment produces error or has limitations based on environmental conditions. GPS is one example of such a system, although there are certainly many other types of sensor systems that exhibit limitations in different environments. GPS consists of about 34 non-geostationary satellites orbiting at an altitude of 10,900 nautical miles (Uni, 2002). As GPS is an external system on which ordinary users have no influence, accuracy of position is a variable in this work that cannot be directly controlled, and so the objective is to gain practical experience with commercial GPS receivers and to understand the nature of error observed in urban or semi-urban environments. While there are many factors which influence GPS accuracy to varying extents, a significant portion of the error variation in GPS can be attributed to or influenced in some way to the particular geometry of the available satellites used to obtain a fix (Wormley, 2007). However, it is also widely accepted that the quality of the position reported by a GPS receiver is diminished, often quite significantly, when the receiver is in the presence of large structures or natural foliage that even partially block direct view of the constellation, hence the focus on urban environments. The combination of occasional obstructed views and poor geometry can result in variations in positioning error of up to several orders of magnitude.

Despite the extremes in error observed in GPS receivers, typical commercial household GPS receivers are capable of performing quite well, and most are generally known to be accurate to around 15 meters with a 95 percent confidence level (Köhne and Wößner, 2005), although some may be rated slightly better. To cite a specific example, a non-differential Garmin GPS receiver similar to the GPS-18 was empirically evaluated to be accurate to within seven meters with a

95 percent confidence level (Rupprecht, 2007). However, these ratings are determined under ideal environmental conditions, and more significantly, the positions outside the stated accuracy are of interest here, particularly considering that error states can occur over lengthy durations even if the overall accuracy, as measured using thousands of samples recorded over a significant length of time, proves to be within specifications. In other words, error among those 5 percent of positions which fall outside the specified tolerance is substantial enough to be a noticeable annoyance to the user. Finally, given that GPS error is generally modeled as a random variable, it is not sufficient to simply look at raw error values. When studying GPS error, or when using error estimates in applications, it is also very important to obtain a sense of the confidence level in any particular error value or estimate.

3.3.1 Sources of Error

Prior to May 2, 2000, accuracy of commercial GPS receivers was limited by so-called *selective availability*, an artificial and deliberate perturbing of both the L1 time, the time signal used by civilian receivers, and the satellite ephemeris data, which indicate the expected position of the space vehicles based on their orbits. Selective availability was intended to make position approximations in commercial receivers less accurate than military grade GPS. During times of selective availability, commercial receivers could be expected to be accurate only to within approximately 100 meters (Köhne and Wößner, 2005). Disabling selective availability has significantly improved the position accuracy of commercial GPS receivers, but several important sources of error still remain. This section is not intended as a comprehensive discussion of possible sources of GPS error. Instead,

it is more to broadly describe the external conditions under which a GPS receiver may produce erroneous positions.

Atmospheric Effects. While most commercial receivers can compensate for reduced radio propagation speed in the ionosphere and troposphere under normal conditions through the use of internal models, unforeseen cases such as unusually strong solar winds can result in positioning error of magnitudes up to several meters. Furthermore, positioning error can be caused by refraction of the signals attributed to varying concentrations of water vapor in the troposphere. While such conditions affect accuracy to a degree, many commercial receivers are capable of accounting for refractive effects through the use of techniques such as *wide area augmentation* (WAA). The Garmin GPS-18 receiver, for example, is capable of exploiting WAA.

Satellite Geometry. Positions of GPS satellites vary at different times of day and on different days of the month. One consequence of this is variability in the precision of the position estimate. Two satellites which are positioned at a 90-degree angle from the view of the receiver will estimate a more precise and accurate ground position than two satellites aligned in a more linear arrangement. Each signal produces a range of possible positions, and with a sufficiently wide angle, the intersection of multiple signals is much smaller. In practice, however, poor satellite geometry does not actually create error on its own; rather, it effects an amplification of other types of errors.

Multipath Effects. Multipath effects are typically found in urban environments, where radio signals are reflected off of large, solid structures such as large trees and buildings. The error is caused by both the additional time it takes for a reflected signal to reach the receiver as well as the lack of direct visibility

from the receiver to the satellites caused by large obstructions.

Satellite Orbits and Clock Synchronization. GPS satellites require small corrections to ephemeris data in the orbiting space vehicles due to small shifts in their orbits. They also require small periodic clock corrections as well. These updates are managed and loaded manually from the master control facility located at Schriever Air Force Base in Colorado Springs, Colorado. Delays applying such updates could result in positioning errors of as much as several meters.

3.3.2 GPS Error in Practice

Experiences using the TeamTrak system in informal settings revealed the frequency and extent of GPS error among commercial receivers in semi-urban environments. These experiences augmented other, oftentimes humorous, anecdotal observations of GPS error in automotive navigation systems. For instance, while driving to a conference a while back, the onboard GPS system precisely tracked the vehicle down the middle of the Chicago River. Similar experiences with automotive systems showed vehicles traversing cornfields and through dense forests, while actually driving along an interstate highway. While GPS error has certainly been studied before, the intent of this work is to measure GPS error and correlate the data to available metrics which are generated by the receiver and accessible through its software programming interface. The objective is not only to describe and model GPS error in urban environments, but ultimately to apply lessons learned towards building systems that can account for and correct error even when its presence may not be immediately obvious to the user.

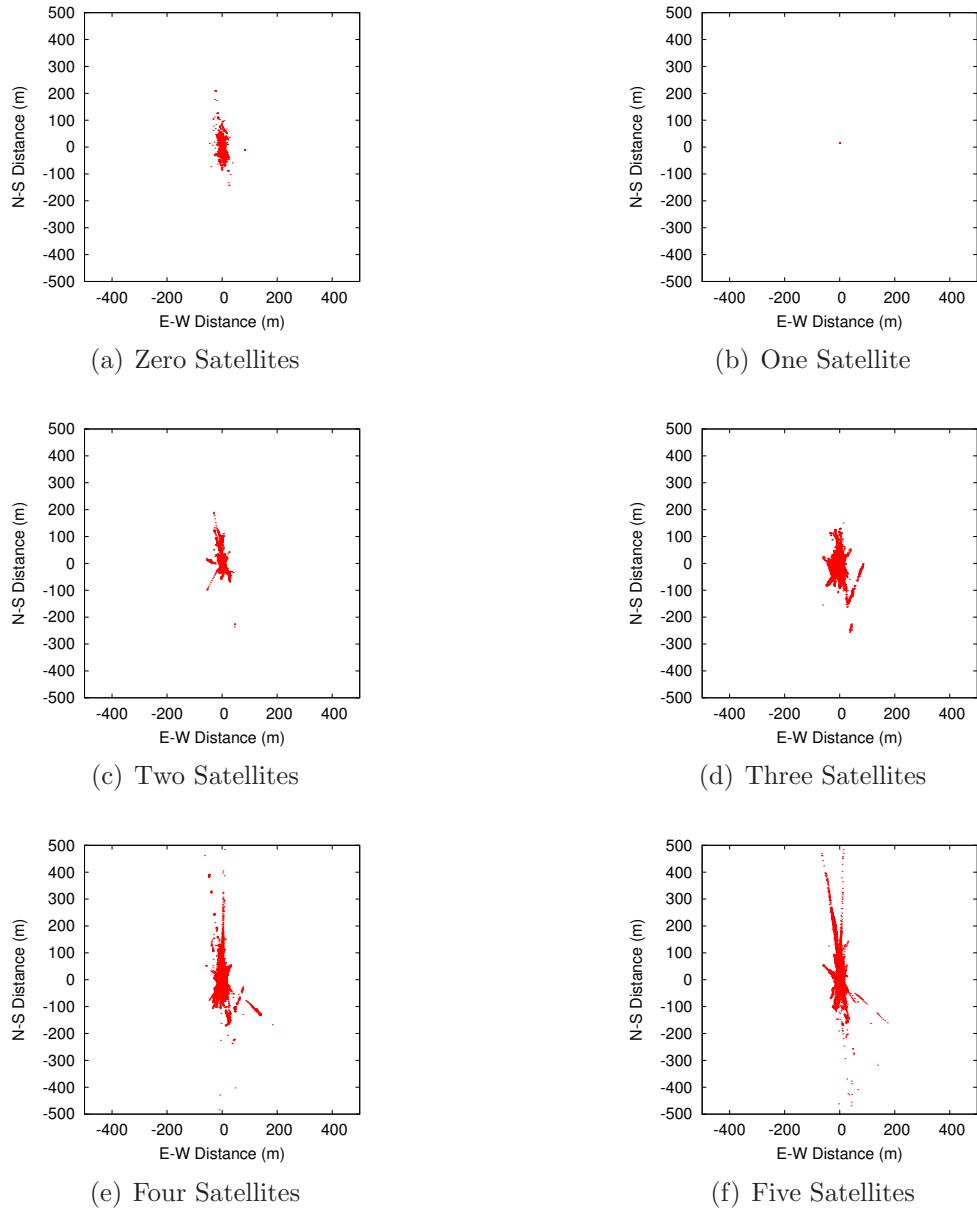
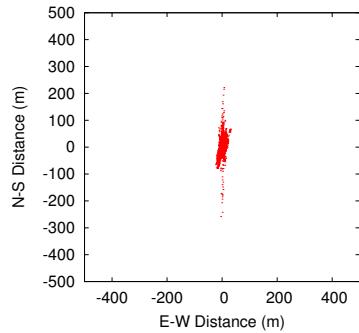
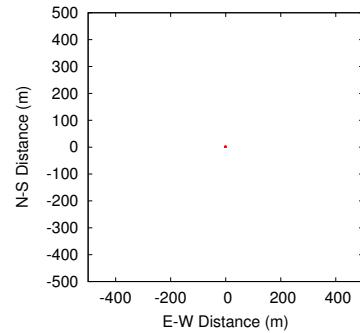


Figure 3.3: GPS Drift by Number of Satellites Acquired



(a) Six Satellites



(b) Seven Satellites

Figure 3.4: GPS Drift by Number of Satellites Acquired (Continued)

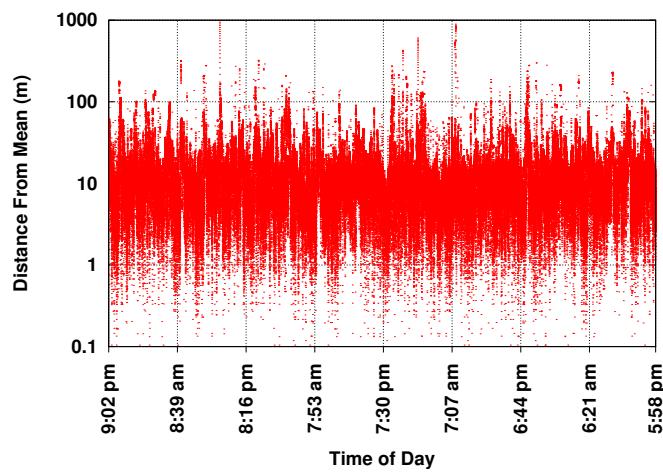


Figure 3.5. GPS Error Over Time

TABLE 3.1
GPS SIGNAL QUALITY SAMPLE SIZES

Number of Satellites	Number of Samples	Number of Satellites	Number of Samples
0	31,301	4	89,843
1	9	5	42,885
2	13,736	6	11,931
3	64,282	7	29

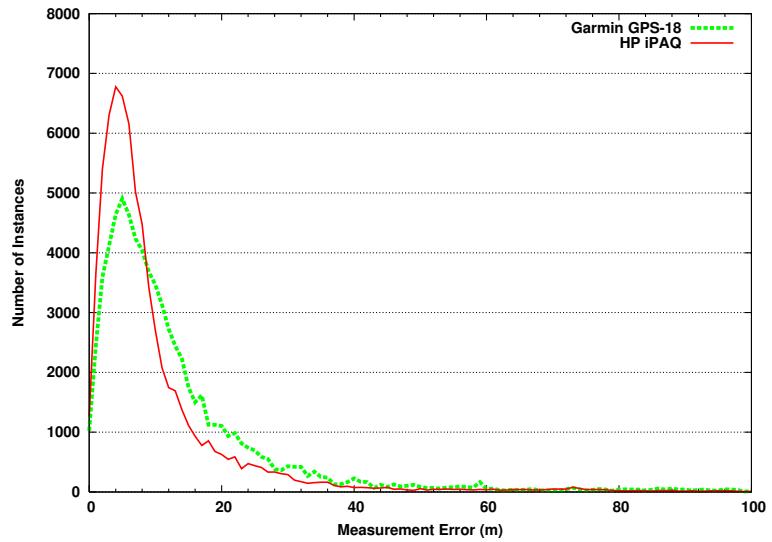
To understand the effect of signal quality on error, which generally manifests itself as a drift in the reported position which may be very gradual or quite abrupt, several experiments were conducted in which a single Garmin GPS-18 receiver was placed outdoors in a fixed location with a partially obstructed view of the sky. At regular, i.e., 1-second, intervals over a period of several days, the reported position was logged along with the number of satellites used to approximate the position for each sample. Figure 3.3 shows the amount of drift for positions reported with between zero and five satellites, respectively. Similarly, Figure 3.4 shows the drift of positions obtained with six or seven satellites. The number of samples used to plot each subfigure of Figures 3.3 and 3.4 is shown in Table 3.1. Drift in this case is determined by computing the Euclidean distance from each sample to the mean position (expressed in latitude and longitude) for the entire data set because the receiver was kept stationary. The true location is assumed to be the mean

of all recorded points, as this is generally accepted as an accurate measurement due to the reduction or elimination of most random error in the samples. A data set collected over a period of several days can yield an approximation of the true position accurate to within one or two meters (Wilson, 2001). Curiously, in this experiment, with a larger number of satellites used to estimate position, i.e., four and five, the absolute difference between sampled positions and the mean was in many cases much larger than that observed with fewer satellites.

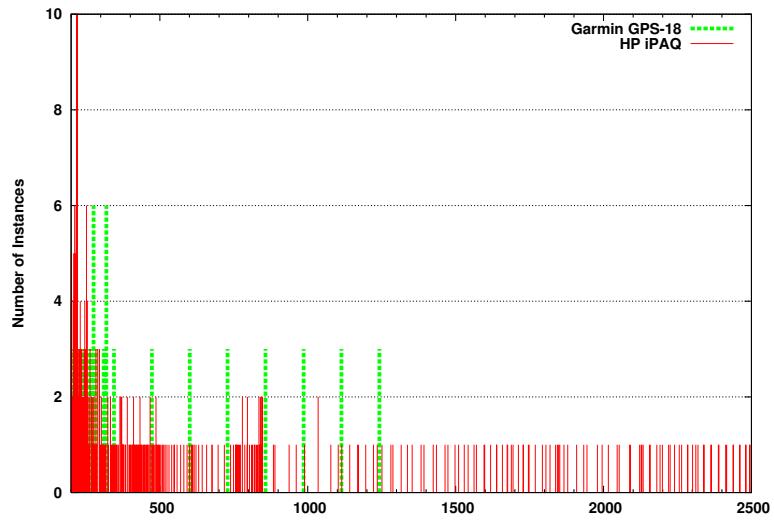
A comparison of the error distribution in two commercially available receivers, the Garmin GPS-18 and the HP iPAQ BT-308, is shown in Figures 3.6(a) and 3.6(b). These figures show the number of instances of each error value observed from each receiver over a total of 72,454 samples. Interestingly, while the HP iPAQ receiver produced more positions accurate to within 10 meters (as illustrated by the higher curve on the top graph, Figure 3.6(a)), it also produced significantly more positions with error greater than one kilometer, as illustrated in Figure 3.6(b). The mean error from the HP receiver is 15.69 meters with standard deviation $\sigma = 79.73$, while the mean error in the Garmin receiver is 14.27 meters with $\sigma = 23.22$. The maximum error observed in the HP was 2,507 meters, while the Garmin's maximum observed error was 1,207 meters. The expected GPS error using the Garmin receiver is 13.32 meters, and the expected error using the HP receiver is 15.69 meters. A CDF for the measurement error in both receivers is shown in Figure 3.7.

3.3.3 Error Metrics

Examining the output from both receivers and their technical data sheets, one may conclude that the following metrics could be useful for error detection and



(a) Measured Error $\leq 100\text{m}$



(b) Measured Error $> 100\text{m}$

Figure 3.6: Distribution of Error in Garmin and HP GPS Receivers

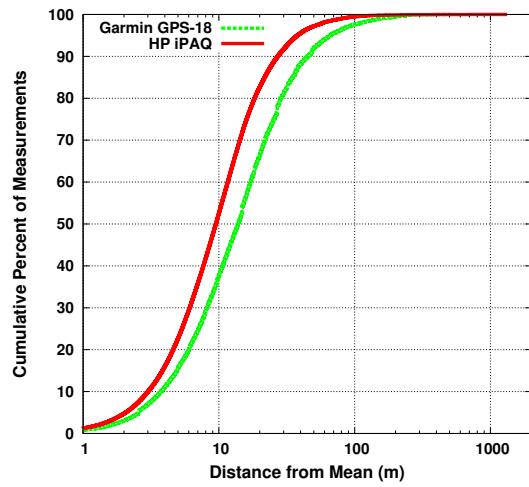


Figure 3.7. Measurement Error Cumulative Distribution Functions
(Garmin and HP GPS Receivers)

correction when building mobile systems that rely on GPS localization:

TABLE 3.2
NUMBER OF SATELLITES VS. GPS ERROR

Number of Satellites	Total Samples	Mean Error (m)	Standard Deviation σ	Max Error (m)
0	38,858	24.17	41.54	1,328
1	9	15.44	0.00	16
2	15,453	21.82	13.31	239
3	86,000	21.82	22.99	839
4	144,585	20.78	18.12	806
5	132,212	17.53	21.82	896

TABLE 3.2

Continued

Number of Satellites	Total Samples	Mean Error (m)	Standard Deviation σ	Max Error (m)
6	97,214	10.92	14.94	575
7	51,466	8.90	12.15	525
8	18,621	7.55	8.63	296
9	4,008	6.58	9.62	301
10	134	3.58	2.84	9

Number of Satellites. It is commonly accepted that the more satellites used to approximate a position, the more accurate the position should be. That notion is intuitive in that with more possible regions used in the estimation, the receiver should be expected to yield a smaller intersection in which the reported location should lie, and thus the resultant position should be both more accurate and more precise. Experience with the commercial GPS receivers used in TeamTrak suggests this is not always the case, particularly in the less-than-ideal environmental conditions of interest in this work.

To evaluate the effectiveness of using number of satellites as a quality metric, GPS data, to include both latitude and longitude as well as the number of satellites acquired for each position sample, were collected over a period of 7 days in total with a stationary GPS-18 receiver placed in a location with a partially obstructed



Figure 3.8. Obstructed Sky View at Data Collection Point

view of the sky, shown in Figure 3.8. In all, 588,560 points were recorded. In a similar fashion to the experiment described in Section 3.3.2, the mean is used as the true location in this experiment. Table 3.2 illustrates the correlation between satellites and GPS error, defined here as the Euclidean distance between each sample position recorded and the overall mean. Note that while there is a linear decrease in both the mean error and the standard deviation σ which correspond to increases in the number of satellites used, there are substantial errors in the positions associated with each satellite count, many of which are on the order of a kilometer. Two exceptions to this observation are the cases in which the number of satellites reported is one, in which case nine data samples is insufficient to draw any meaningful conclusions from, and ten satellites, which can estimate positions very accurately, but is unfortunately a relatively rare case.

To graphically illustrate the relationship between number of satellites and GPS accuracy, Figure 3.3 in Section 3.3.1 shows the variation in reported position for a static receiver as a function of the number of satellites acquired for each position. Even in cases where each position was derived from four or five satellites, the estimated position can vary from the mean by as much as 500 meters or more. Using six satellites, the drift was as large as 300 meters. This data further suggests that the number of satellites alone is insufficient to reliably detect error in GPS.

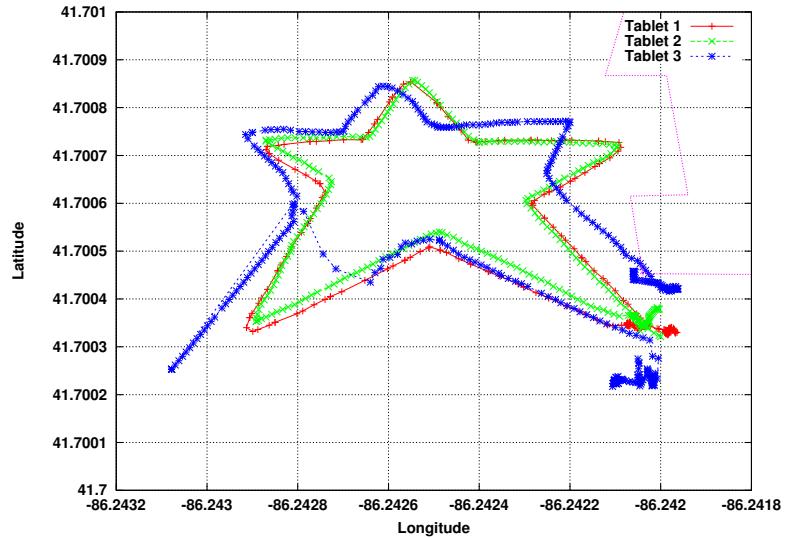


Figure 3.9: Example of GPS Consistency in Practice

Finally, as a notional illustration of the lack of correlation between accuracy and number of acquired satellites, Figure 3.9 shows the paths plotted during an outdoor excursion using the Garmin GPS-18 receivers. In this exercise, three students plotted their location using GPS at one-second intervals while traversing a path shaped like a star (as an aside, the star-shaped path was selected for this exercise as it had more detail which could better reveal inconsistencies in location information between nodes traveling together. The impact of such inconsistency likely would be less apparent than if the students had traversed a simpler, linear path). The machines identified as Tablet 1 and Tablet 2 have GPS positions which are tightly clustered together throughout the duration of the exercise. In this case, Tablet 1's positions were consistently obtained with either nine or ten satellites, and Tablet 2's position using a range between six and nine satellites. The third machine, Tablet 3, never acquired a fix with more than five satellites, and in the case of the positions along the lower left hand corner of the graph

which illustrate a fairly significant deviation, the error resulted when the receiver reported an active GPS fix, but the total number of satellites acquired was zero. Experience has shown this behavior is not unique to either the Garmin receiver or the particular location of the exercise. Active fixes with zero satellites reported also has been observed in the HP iPAQ receiver as well as in other geographic areas as well.

One additional observation that can be made from these results is that multiple receivers positioned in the same physical location generally will not fail in a similar fashion. In fact, the number of satellites acquired, the quality of the position, or even whether a fix is obtained at all is highly inconsistent even among identical model hardware used in the same area at the same time, as the exercise illustrated in Figure 3.9 illustrates. Experience has demonstrated that the GPS error observed among collocated or very closely placed devices occurs independently from each other, and localization accuracy may be quite different from one device to the next. This behavior was observed numerous times during the course of outdoor exercises, when two data collectors physically standing next to one another observed on their display that their nodes were positioned on opposite ends of a quad; a separation distance on the order of 100 meters.

TABLE 3.3
LOCATION QUALITY/CONFIDENCE VS. HDOP VALUES

Quality	HDOP Range
Ideal	0 - 2
Excellent	2 - 4

TABLE 3.3

Continued

Quality	HDOP Range
Good	4 - 7
Moderate	7 - 9
Fair	9 - 21
Poor	21+

Horizontal Dilution of Precision (HDOP). Dilution of precision is a numerical indicator representing both the quality of the satellite geometry and the confidence in such quality. HDOP values range from one (and in some cases less than one), which represents the best possible satellite geometry, to a maximum value which depends on the specific receiver and/or manufacturer. For instance, some Garmin receivers report a maximum HDOP value of nine, while the HP iPAQ receiver reports values up to and including 50. For all receiver models, larger values represent lower quality satellite geometry, which implies a less reliable, less accurate position, while an HDOP value of one is generally accepted as ideal for most commercial applications.

Table 3.3 shows a scale for determining the quality of a given GPS position (Person, 2003). The estimated accuracy for the HP receiver is a function of the HDOP value, which is available directly from the device. In the case of the HP receiver, HDOP values less than one were somewhat frequently observed, and

therefore such values are included in the scale.

TABLE 3.4
MEAN, MINIMUM, AND MAXIMUM REPORTED GPS POSITION
ERRORS BY QUALITY (HP IPAQ RECEIVER)

Location	Mean Error	Standard Deviation σ	Max Error	Min Error
Quality	\bar{x} (m)	(m)	(m)	
Ideal	6.952	0.072	500.34	0.106
Excellent	9.576	0.057	97.33	0.106
Good	13.971	0.128	91.43	0.157
Moderate	18.472	0.344	65.19	0.157
Fair	21.794	0.289	125.96	0.157
Poor	110.10	4.279	2507.4	0.3275

TABLE 3.5
HORIZONTAL DILUTION OF PRECISION ($\epsilon = 5, 10, 15, 20, 25, 30$)

Quality	\bar{x} (m)	$P(\text{error} \leq \epsilon)$		Quality	\bar{x} (m)	$P(\text{error} \leq \epsilon)$
$\epsilon = 5$ m						$\epsilon = 20$ m
Ideal	17.825	51.882		Ideal	17.825	85.046

TABLE 3.5

Continued

Quality	\bar{x} (m)	$P(\text{error} \leq \epsilon)$	Quality	\bar{x} (m)	$P(\text{error} \leq \epsilon)$
Excellent	6.942	40.720	Excellent	6.9422	97.345
Good	12.998	14.838	Good	13.000	80.157
Moderate	17.909	14.240	Moderate	17.909	67.972
Fair	21.113	11.788	Fair	21.113	65.500
Poor	108.71	3.2660	Poor	108.71	30.151
$\epsilon = 10$ m			$\epsilon = 25$ m		
Ideal	17.825	72.533	Ideal	17.825	89.166
Excellent	6.9422	81.977	Excellent	6.942	98.651
Good	12.998	44.023	Good	12.998	90.248
Moderate	17.909	35.521	Moderate	17.909	75.966
Fair	21.113	38.879	Fair	21.113	73.662
Poor	108.71	13.568	Poor	108.71	37.961
$\epsilon = 15$ m			$\epsilon = 30$ m		
Ideal	17.825	78.942	Ideal	17.825	91.506
Excellent	6.9422	93.657	Excellent	6.9422	99.509
Good	12.998	65.436	Good	12.998	96.542
Moderate	17.909	55.267	Moderate	17.909	81.789
Fair	21.113	55.784	Fair	21.113	79.689
Poor	108.71	23.492	Poor	108.71	44.745

TABLE 3.6
HORIZONTAL DILUTION OF PRECISION ($\epsilon = 30, 35, 45, 50$)

Quality	\bar{x} (m)	$P(\text{error} \leq \epsilon)$	Quality	\bar{x} (m)	$P(\text{error} \leq \epsilon)$
$\epsilon = 35$ m			$\epsilon = 45$ m		
Ideal	0.8247	93.337	Ideal	17.825	94.710
Excellent	6.9422	99.769	Excellent	6.9422	99.924
Good	12.998	99.074	Good	12.998	99.836
Moderate	17.909	87.083	Moderate	17.909	94.018
Fair	21.113	84.367	Fair	21.113	88.667
Poor	108.71	48.702	Poor	108.71	57.035
$\epsilon = 40$ m			$\epsilon = 50$ m		
Ideal	17.825	94.151	Ideal	17.825	95.626
Excellent	6.9422	99.892	Excellent	6.9422	99.937
Good	12.998	99.777	Good	12.998	99.918
Moderate	17.909	89.889	Moderate	17.909	96.400
Fair	21.113	87.474	Fair	21.113	89.468
Poor	108.71	53.559	Poor	108.71	59.338

Using the HP iPAQ receiver placed in a stationary location with a partially obstructed sky view, GPS positions were recorded over a period of 24 hours along with the HDOP value associated with each. Because this experiment was conducted in the same physical location as the previous experiment with the Garmin receiver, results from both were compared to the same “true” position, i.e., the mean of all recorded points. Indeed, samples acquired in this experiment were used to further adjust the overall mean by averaging them with the Garmin experiment’s data. As before, in this experiment the average deviation of the reported position from the true position, called \bar{x} here, was determined, along with the maximum and minimum deviations observed in the data set, sorted by HDOP range. Table 3.4 shows \bar{x} as well as the maximum and minimum measured position errors for each range of position quality reported by the HP iPAQ GPS receiver.

Not surprisingly, for increasing ranges of HDOP values (and thus decreasing estimated quality), both the mean error \bar{x} and the standard deviation σ of points associated with each range, increase. However, the maximum and minimum measured errors do not behave exactly as expected. Even with very good satellite geometry, large errors occasionally still occur, as evidenced by, for example, the 500-meter measured error which occurred with a very low HDOP value. Similarly, regardless of HDOP, it is still possible for the receiver to produce a very accurate position, as evidenced by the very small minimum measured errors observed for all HDOP ranges.

Finally, to get a sense of the reliability of HDOP as a quality metric in the iPAQ receiver, data collected from a stationary HP GPS receiver were sorted by HDOP ranges, then the confidence level for each quality indicator was determined. Confidence in this experiment was determined by assessing the probability that

the measured GPS error was less than or equal to a predetermined error tolerance level ϵ . Tables 3.5 and 3.6 indicate the confidence level of the quality indicator for increasing values of ϵ up to and including 50 meters.

Horizontal Error. Many Garmin GPS receivers, to include the GPS-18 model, generate a *horizontal error* metric for two-dimensional fixes, which is an estimate of the accuracy of the position in two dimensions. Horizontal error is generally a function of both horizontal dilution of precision values and root mean square (RMS) errors empirically determined by the manufacturer. Looking at the horizontal error is a convenient, but rough and not necessarily reliable, indicator of the estimated position error. Unlike HDOP, horizontal error values can be examined directly rather than basing the quality on a separate scale. Much like HDOP, however, horizontal error is modeled as a random variable, so its value cannot be guaranteed accurate and must be checked against confidence values to be useful.

Figure 3.10 shows the confidence level for horizontal error reported by the Garmin receiver. In the figure, the error of each reported position was measured and the values sorted by horizontal error in increasing order. The figure shows the probability that the measured error is less than or equal to the estimated horizontal error. Note that most values have a confidence level of between 80 and 90 percent, with the exception of some very small and very large horizontal error values. In most applications, the large values would be, in all likelihood, discarded anyway, and very small values, i.e., less than 10 meters, are probably too unreliable to be used in practice. The anomaly in the graph occurs with a horizontal error value of three meters. This seemingly perfectly accurate metric can be explained by the small number of samples at that value (three in total), all

of which had an associated measured error of less than or equal to three meters. Clearly, the extremely small sample size for that error value would be reason to use caution if modeling error based on a data set such as this.

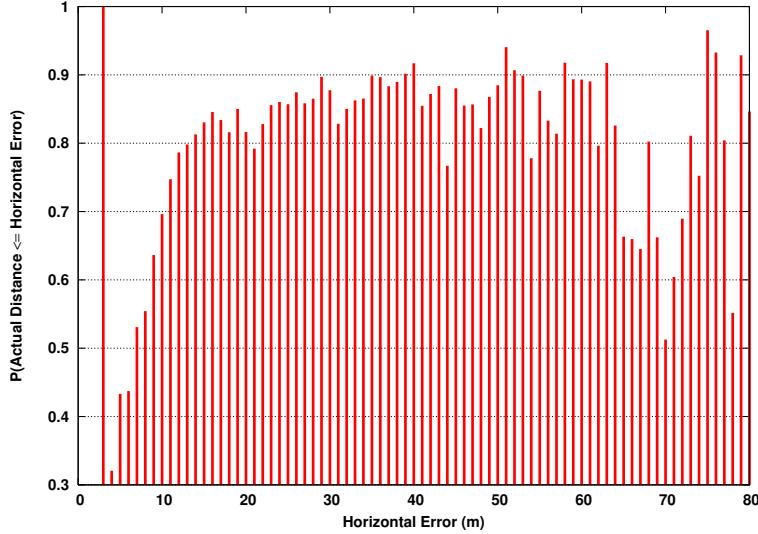


Figure 3.10: Reliability of Horizontal Error

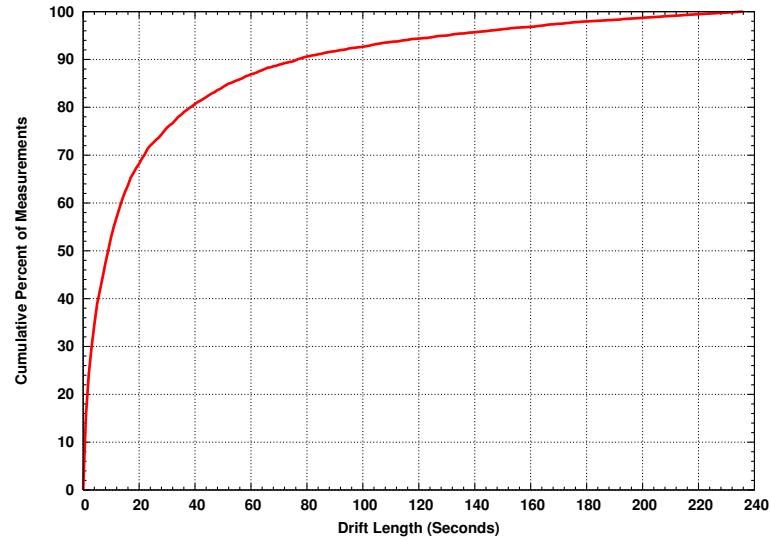
Having examined the errors encountered during practical application of GPS receivers, attention is now turned towards handling localization when a previously-available GPS signal is no longer available at all. To provide continuous localization in such cases, a mechanism has been incorporated into the TeamTrak platform to accomplish dead reckoning. This mechanism requires two independent sensors operating in tandem: a digital accelerometer which functions as a pedometer for both step detection and stride length estimation, and a compass to determine the heading of the human operator.

3.3.4 Modeling GPS Error

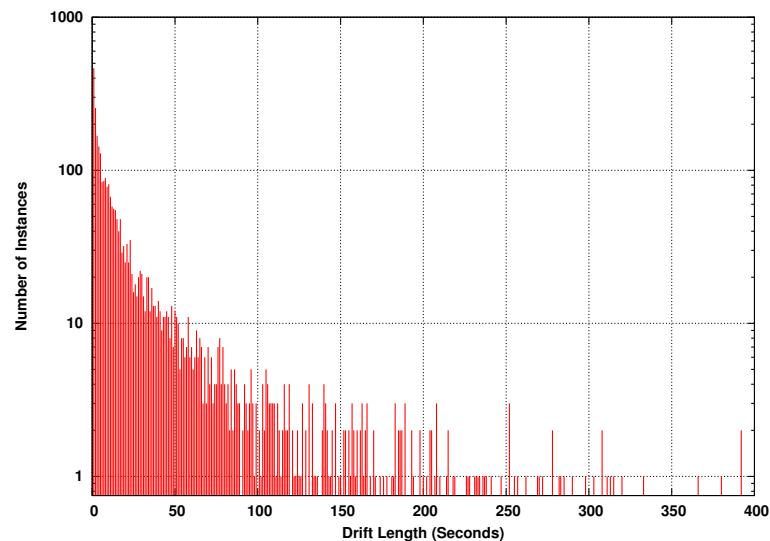
A probabilistic model of GPS error requires measuring two components: the magnitude of the error and the duration of time each particular error magnitude occurs. Both can be modeled with a CDF derived from the empirical data. The first step in modeling GPS error is generation of a uniformly distributed random, floating point number, the value of which corresponds to the cumulative distribution of discrete error magnitudes. In other words, for a randomly generated value X in the interval $[0, 1)$, the error model returns the quantile of order X :

$$F(X) = \inf\{x \in \mathbb{R} : X \leq F(x)\}$$

Once the magnitude of the error is determined, the next step is computing the expected drift length for that particular error value. As with absolute error, the drift length is determined empirically, and both a CDF and a probability density function is computed from the live GPS data. For this model, a drift is considered to be the number of consecutive samples recorded at 1-second intervals whose error deviates from the previous sample by less than one meter. Figure 3.11(a) shows the cumulative distribution for all drift lengths, regardless of associated error value. Similarly, Figure 3.11(b) shows the distribution of all recorded drift lengths. The model of drift length is a conditional probability function. Given an error magnitude E , the probability of a specific drift length at distance E is given by the inverse of the PDF.



(a) GPS Drift Length Cumulative Distribution Function



(b) GPS Drift Length Distribution

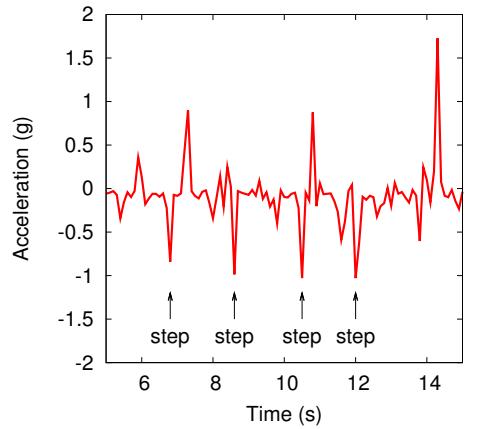
Figure 3.11: Observed GPS Drift

3.4 Digital Accelerometer

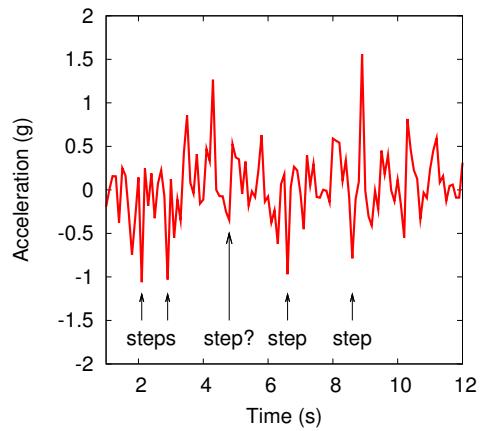
3.4.1 Overview

One inherent limitation of GPS is that it cannot be used in locations lacking a view of the sky, which includes not only indoor environments but other highly shielded areas such as very dense urban centers, under jungle canopies, etc. As a workaround for this limitation, previous work has proposed using accelerometers which are mounted on, and can detect movement of, mobile robots (Liu and Pang, 1999; Pang and Liu, 2001). The accelerometer output can then be integrated to determine current velocity and ultimately change in position. This change in position is then used for fine-grained localization starting from a valid initial reference point. The projects that have implemented such a localization method have demonstrated good results, but their success is due primarily to the relatively consistent acceleration patterns found in robots or wheeled vehicles.

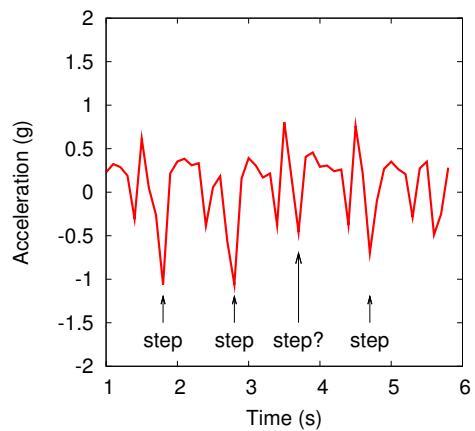
Attempting localization in this fashion for humans is far more challenging, and initial testing of the accelerometer proved to be quite discouraging. Unlike a robot or a wheeled vehicle, humans provide a relatively unstable platform for mounting motion-sensing hardware, and attempting to integrate accelerometer data can lead to very substantial approximation errors, often on the order of kilometers. The difficulty arises from two fundamental problems: first, humans do not provide a solid platform on which a sensor can be mounted, which makes determining the specific axes of motion very difficult. Second, humans while walking exhibit a wide range of “noisy” motions which cause errors to be greatly amplified when integrating raw sensor output for more than a few seconds at a time, even under the most carefully controlled conditions.



(a) Walking (Flat Surface)



(b) Running (Flat Surface)



(c) Stair Climbing

Figure 3.12: Accelerometer Patterns

TABLE 3.7
 EFFECT OF STRIDE LENGTH ESTIMATION ON ACCURACY OF
 TOTAL DISTANCE ESTIMATION FOR DEAD RECKONING

Trial Number	Actual Distance (m)	Estimated Distance (m)	Accuracy (%)
1	72.87	70.81	97.17
2	72.87	72.48	99.46
3	72.87	72.89	100.02

However, despite the inherent limitations of mounting sensor hardware to a human operator, the accelerometer is certainly useful for assisting localization. The device can be used to detect specific motions of the body. Fastened to a person's shoelaces it can be used quite effectively to detect individual steps. Coupled with a compass (discussed in Section 3.5), accelerometers can be used to provide location updates when GPS is not available. The accelerometer functions as a pedometer which detects individual steps taken. Each local minimum in the accelerometer reading represents a step, as illustrated in Figures 3.12(a) through 3.12(c), and by looking at values of jerk, i.e., the first derivative of acceleration, these local minima can be detected effectively. All data read between each local minimum represents a single stride. Using the local maximum and minimum values for the acceleration along the Z (rotational) axis on each stride, the approximate stride length is given as:

$$\text{stride length} \approx k \sqrt[4]{A_{max} - A_{min}}$$

where A_{max} and A_{min} are the local maximum and minimum acceleration values along the Z axis, respectively, and k is an empirically determined constant proportional to the length of the individual operator's leg. Naturally, to use this model, the k value would have to be determined prior to using this dead reckoning system and the system calibrated for the particular user. This method for determining stride length was used in the NavMote project (Fang et al., 2005). Experience has shown this method to be quite effective, provided a suitable value for k is selected. Table 3.7 shows the accuracy of stride length estimation. In these trials, the results achieved over a distance of approximately 73 meters would be acceptable for any application envisioned. Step detection using jerk has been nearly 100 percent accurate consistently as well.

3.5 Digital Compass

3.5.1 Overview

With a suitable model for stride length estimation in place, localization can be accomplished by combining the distance traveled with each step with the heading Θ reported by the compass. Simple trigonometry can be used to determine the change in position along the axes. For navigation purposes in the TeamTrak testbed, a two-dimensional Cartesian plane is assumed. The changes in longitudinal and latitudinal positions, $\Delta x, \Delta y$, respectively, are defined as:

$$\Delta x = \text{stride length} \times \sin(\Theta) \times 1.21 \times 10^{-5}$$

$$\Delta y = \text{stride length} \times \cos(\Theta) \times 9.0 \times 10^{-6}$$

where the constants represent a distance of one meter at the latitude and longitude at which the testbed is employed.

3.5.2 Sources of Error

In the TeamTrak platform, the compass is mounted on a standard ANSI Z89.1 safety helmet worn by the operator. Because of the uncertain motions that occur not only from walking, but from various movements of the head, shoulders, and neck, the compass is subjected to a great deal of pitch and roll variations in practice which must be accounted for in practice. The original implementation of TeamTrak included a non-tilt-compensated, two-axis digital compass. Such a compass, if subjected to any tilt greater than approximately ten degrees in any direction, would experience error on such a magnitude as to be unusable. The nature of the testbed design demanded the use of a tilt-compensated compass, which was acquired and incorporated into the platform.

Compass error evaluation was conducted in two parts. First, the effect of pitch on the reported heading was evaluated. This was done by placing a calibrated compass on a level surface with a fixed, known heading, then rotating the angle of pitch between -90° and 90° while recording the output. Since the compass is tilt-compensated, the effect of pitch on heading proved to be minimal, as shown in Figure 3.13(a). However, the device does not compensate for roll, as the documentation warns. A graph of raw heading data versus roll angle is shown in Figure 3.13(b). Even though the compass does not compensate for roll, the heading can be improved by approximating the effect of roll using a simple sine function and adjusting accordingly. The heading adjustment used to compensate

for roll in this particular compass is:

$$\text{heading} = h - 70 \sin\left(\frac{\pi}{90}r\right)$$

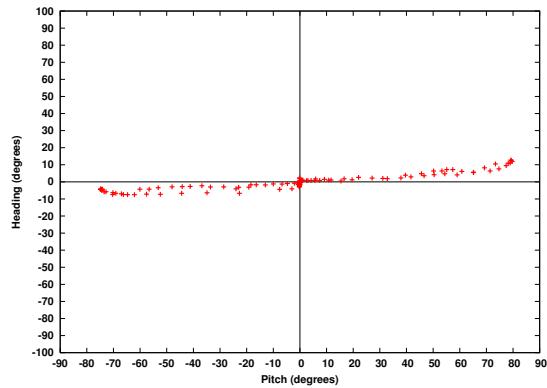
where h is the uncorrected heading reported by the compass, and r is the amount of roll, both expressed in degrees. The adjusted heading is shown in Figure 3.13(c). Compass error follows a normal distribution with a mean of -0.794 degrees and a standard deviation σ of 11.33.

Finally, effectively modeling the compass error, even with tilt compensation, requires understanding the distribution of errors experienced in practice. To accomplish this, a three-axis compass connected to a TeamTrak tablet computer was brought outdoors and fully calibrated in that particular environment. Then, with a fixed, known heading, the data collector shot a simple azimuth and walked in that direction, keeping the compass reasonably steady without being unduly careful not to subject it to pitch and roll variations. The objective of the experiment was to collect data about compass error which might realistically be experienced in practice and characterize the distribution of that error. Figure 3.14 shows the distribution of compass error.

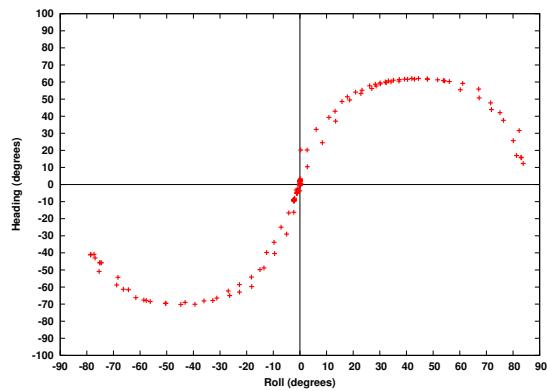
3.6 Experiences With Dead Reckoning

3.6.1 Putting the Pieces Together

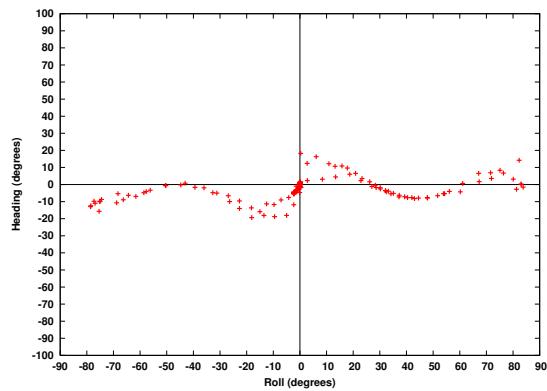
With the error in both the accelerometer and three-axis compass modeled and accounted for, both sensor components can be combined into a single, cohesive dead reckoning system. As alluded to in Section 3.1, the objective here is not to build a flawless personal navigation system. Rather, it is to build a *suitably functional* dead reckoning system that provides continuously available position-



(a) Effect of Pitch



(b) Effect of Roll (Uncorrected)



(c) Effect of Roll (Corrected)

Figure 3.13: Effect of Pitch and Roll on
Compass Heading

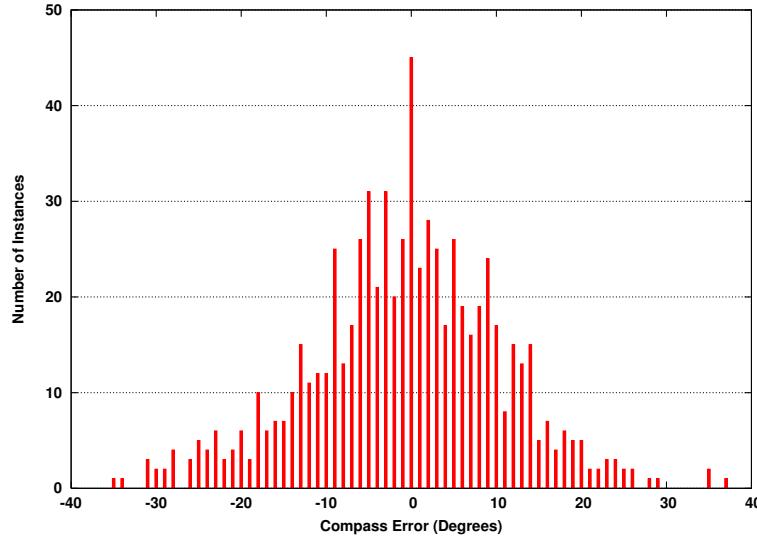


Figure 3.14. Distribution of Compass Error

ing, albeit with possibly significant error, so that the ideas about cooperative localization discussed in Chapter 4 can be fully evaluated, either in practice with a physical testbed implementation such as TeamTrak, or through experimental evaluation by providing parameters which can be applied to modeling and simulation.

With a good empirically determined constant coefficient for estimating stride length as described in Section 3.4, the error in dead reckoning is dominated by compass error and is very much dependent on the quality of the initial starting point. If that point was determined via GPS during a period of substantial drift or even a one-time “jump” in position, then localization, even with perfect dead reckoning, can be no more accurate than the error associated with that drift.

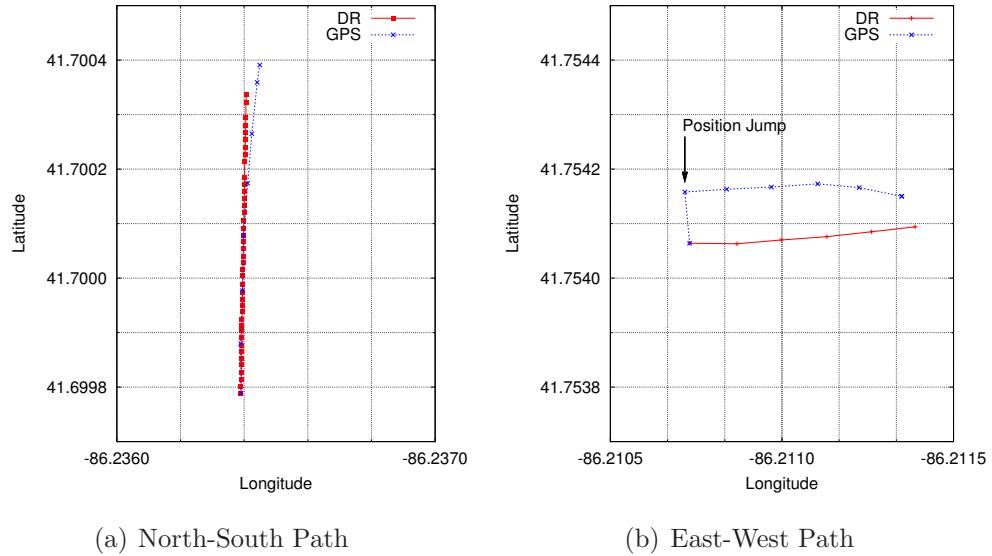


Figure 3.15. Dead Reckoning Versus GPS Accuracy

3.6.2 Dead Reckoning Error in Practice

To illustrate the effect of starting position accuracy on dead reckoning, Figure 3.15 shows the latitude and longitude of two sets of points along the same path. One path was approximated using the two-axis compass, the other using the three-axis compass in dead reckoning. In Figure 3.15(b), low accuracy of the GPS signal used for the starting location results in a drift in position, which in some cases might cause the dead reckoning approximation to be more accurate than the GPS position. The inaccuracy of the dead reckoning in 3.15(b) is also attributable to the use of uncorrected compass heading affected by tilt. In this experiment, the two-axis compass was very carefully positioned and stabilized before each step was taken; obviously, requiring such a task would make this device suitable for use in production systems. The compass used in 3.15(a) is tilt-compensated, and

therefore provides approximated positions of acceptable accuracy, even when less care is taken to manually reduce error in heading.

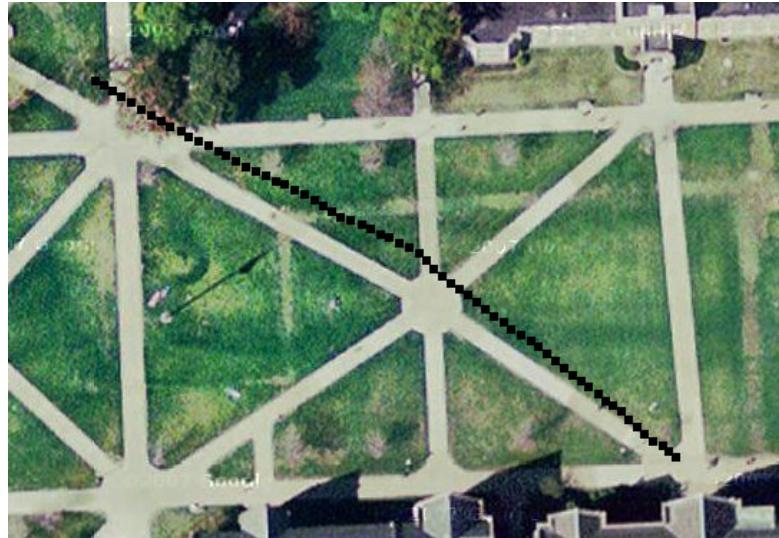


Figure 3.16. Dead Reckoning Error In Practice

To further demonstrate the capability of the dead reckoning system constructed for TeamTrak, two additional exercises were conducted to show the effect of the combined error of both the compass and accelerometer. In the first exercise, the operator simply traverses a straight path beginning from a well-measured starting location. Figure 3.16 shows the path estimated by the dead reckoning system compared to the path along the footpath which was actually traveled. While the map shown in the figure has its own error of unknown magnitude, it is assumed here to be correct. Along this 92-meter path, compass error accounts for at most

one or two meters laterally, and the stride length/step detection system accounts for an error in the total distance traveled of less than 10 meters.

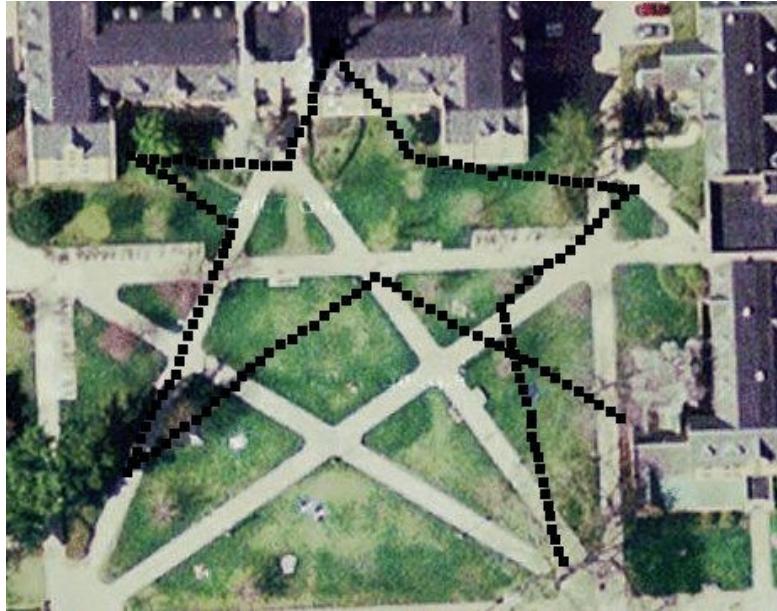


Figure 3.17. Dead Reckoning Error In Practice

Traversing a more complex path results in a more pronounced error. Figure 3.17 shows a path traversed from a starting point measured with multiple independent GPS samples. While it is possible to attain a reasonable degree of accuracy in the absence of GPS, the figure clearly shows that an error which occurs early cannot be eliminated or even reduced with any certainty. Correction of dead reckoning requires an additional localization method whose accuracy does not decay with the number of steps taken. Furthermore, a larger number of turns taken

in the path traversal increases the likelihood of substantial positioning errors.

3.7 The TT-Sim Simulator

While physical testbed implementations are invaluable for gaining practical experience with systems in realistic environments, it is extraordinarily difficult to use them to evaluate certain characteristics such as scalability. Modeling and simulation can effectively substitute for real-world deployments and exercises, but models must be carefully constructed to ensure validity. To the greatest extent possible, simulation parameters in this work are derived directly from empirical data collected in experiments with hardware devices operating under realistic conditions.

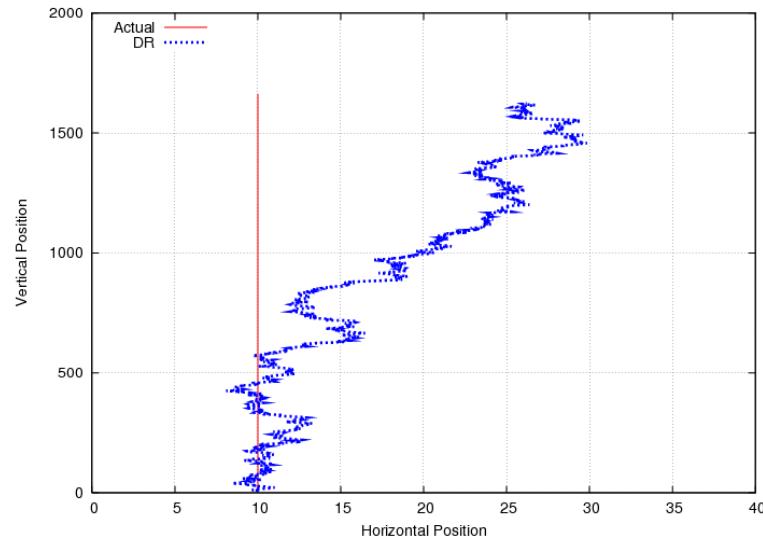


Figure 3.18: Simulated Dead Reckoning Error

TT-Sim is a simulator based on the TeamTrak platform which allows for flexibility in creating evaluation scenarios. The simulator models a collection of mobile TeamTrak nodes operating under conditions in which sensor error is present. Both dead reckoning and GPS error are explicitly modeled using the data described in this Chapter. Figure 3.18 shows a trace of a single node's actual path along with the estimated positions using modeled dead reckoning error as output by TT-Sim. Parameters that can not be empirically measured fully are conservatively estimated to avoid producing overly optimistic results. TT-Sim is used for most of the evaluation tasks in this dissertation.

3.8 Conclusion

The work in this chapter evaluated the measurement error in an array of commodity sensor devices: the Garmin GPS-18 and HP iPAQ GPS receivers, the SparkFun SerAccel 3-axis digital accelerometer, and the OceanServer OS3500 3-axis digital compass, all of which are incorporated into the TeamTrak platform. The purpose of these evaluations is to understand error modes in order to serve two distinct objectives. The first objective was to gain practical experience with sensor error and the limitations of low-cost commodity hardware which would arise in real-world scenarios. This was accomplished through outdoor data collection exercises using the GPS receivers as well as through experimental trials with dead reckoning apparatus. The second objective was to understand and describe the error using empirical data which would provide valid sensor error for constructing models which could be incorporated into simulation efforts.

The accelerometer has provided a surprisingly robust means for detecting individual footsteps when employed as a digital pedometer, at least in the case

when the operator walks with fairly deliberate steps. As mentioned previously, no system is perfect, and successful attempts could be made to defeat the detection mechanisms without a tremendous amount of difficulty. That limitation is well understood, but is beside the point. This system of stride detection and measurement exists only to provide a method for fine-grained navigation in the absence of GPS and a platform for evaluating error correction techniques. That it experiences error is not only expected here, but in some ways even desirable! No sensor system exists without measurement error, and the type of compounding error observed in practice using the stride detection component of the TeamTrak dead reckoning system is representative of that exhibited by many types of commercial pedometers. Having a representative model of this type of error assists in any future evaluation tasks related to dead reckoning techniques in general.

Similarly, evaluation of the digital compass used in the TeamTrak platform has provided insight into the range of error experienced when mounted to a human platform, even when using the built-in tilt compensation for error correction. The use of the compass for dead reckoning shows that overall localization error is far more sensitive to measurement error in the compass heading than it is to that in the stride length estimation. The photo in Figure 3.17 effectively illustrated the effect of such. The difficulty lies in the coupling of the sensitivity of navigation to compass error and the inability to fully correct the error using modeling. For practical purposes, the implication is that dead reckoning systems cannot self-correct and require assistance from external sources. This is certainly not a new observation, but the lack of consistency among GPS localization from one device to another suggests that lack of a GPS fix in one device implies that information to correct positions which are stale or with significant error may still be available

from other remote sources.

With the set of commodity sensors evaluated, the error in each can be modeled such that at least a qualitative assessment of the reliability of sensor data can be completed. Since a single node will always experience measurement error in localization, sometimes significantly, attention now shifts to external sources for correction. External sources include collaborating peers sharing location data, using a set of techniques aptly described as cooperative localization, a topic discussed in detail in Chapter 4.

CHAPTER 4

COOPERATIVE LOCALIZATION

4.1 Introduction

Advances in wireless networking technology and integrated circuit design have opened up vast possibilities for mobile applications and wireless sensor networks (Akyildiz et al., 2002; Committee on Networked Systems of Embedded Computers, 2001; Correal and Patwari, 2001; Feder, 2004; Perkins et al., 2002), yet localization still presents significant challenges to designers of location-sensitive mobile applications. Localization of mobile nodes is frequently accomplished via GPS, but as demonstrated in Chapter 3, GPS positions may be inaccurate, sometimes significantly, under frequently encountered environmental conditions, even when available metrics suggest otherwise. When operated in areas characterized by obstructed or partially obstructed views of the sky, obtaining a fix may be impossible at times. To address this limitation, techniques have been developed or proposed to facilitate correction of GPS error or lack of GPS availability altogether in specific locations using modified or additional receivers. Examples of such techniques include using only those satellites positioned at the highest points in the sky as opposed to those which produce the lowest dilution of precision value, increasing the physical elevation of the receiver, and setting up a fixed offset point at which a reliable GPS fix is available, then computing distance and bearing from each

mobile receiver to that point (Corvallis Technology, Inc., 1996). However, none of these are completely suitable for all applications. In a very remote location, such assistance cannot be generally assumed.

Previous work and experiences with commodity sensor hardware incorporated into the TeamTrak platform demonstrate that even when using an approximate error model to account for and correct measurement error in sensor data, results may still be unreliable at times, particularly when using pedestrian dead reckoning. Any dead reckoning technique requires an accurate initial reference point, which may not always be available, but most importantly, experiences compounding measurement error as a function of the number of steps taken due to both approximation errors and sensor limitations at each step. Although existing individual navigation systems have addressed this limitation by featuring some mechanism for periodic correction (Leonard and Durrant-Whyte, 1991; Liu and Pang, 1999; Pang and Liu, 2001), these techniques generally require preplanned and preinstalled infrastructure, maps, or even manual intervention to correct positioning error. The inherent limitations of available localization techniques and the demonstrated independence and variability of location quality among devices suggest that correction, if at all possible, might be available through other connected nodes sharing location information in a cooperative manner over an ad-hoc, peer-to-peer network.

The work in this chapter is the natural consequence of three observations made during a series of outdoor exercises using TeamTrak as a general-purpose approximation of localized mobile ad-hoc networks:

1. *Sensor error is certain.* Naturally, this observation seems quite obvious. However, it is also the observation that precludes attempts to simply buy

more expensive hardware in the hope that doing so will solve a mobile system’s localization requirements. Fundamental sensor limitations will exist for well into the foreseeable future, regardless of hardware quality or type of sensor, particularly for those sensors whose cost is sufficiently low to feasibly permit deployment on an intermediate to large scale. Therefore, at a minimum some type of software error detection and correction techniques will continue to be a desirable, if not necessary, part of system design.

2. *Even identical sensors experience error independently.* Even with sensor hardware equipment of the identical manufacturer and model, error frequently occurs independently between devices. GPS receivers, for instance, placed in the same physical location at the same time and loaded with identical almanacs will not necessarily acquire the same group of satellites with the same geometry, or may acquire different numbers of satellites, resulting in varying degrees of accuracy. The practical effect of this independence of error is that multiple human operators, each carrying portable GPS receivers, may experience much different positioning even when standing beside one another. If location data were shared between devices, the less accurate position could be adjusted accordingly. Localization through dead reckoning starting at different times and with different GPS fixes will have different estimated error. Similarly, a position estimated through dead reckoning techniques requires periodic correction from a remote source if a local GPS fix cannot be obtained due to the compounding measurement error.
3. *Averaging reduces error.* It is well understood that for stationary receivers, averaging the reported positions over time can improve accuracy substantially over any single position sample by smoothing out the random error

present in any position sample. With a cooperative data sharing approach, averaging positions which represent the same physical point can improve accuracy while reducing the amount of jitter experienced with many localization techniques.

Techniques that exploit differences in location quality among connected nodes through cooperative data sharing have existed for a number of years, but location quality in many works is treated as a Boolean entity; generally speaking, nodes are either fully localized or they are not. Even techniques employing stepwise refinement of positions assume that locations of fixed anchor nodes are always accurate. As previously demonstrated, even if these positions are obtained using GPS, that is not necessarily the case at all times. Mobility and urban environments make nearly certain the likelihood that nodes will have a much wider spectrum of location qualities than would be found under more favorable conditions. Furthermore, depending on the deployment and specific application scenario, the source of localization might not be limited to GPS: dead reckoning, manual configuration, or perhaps some other scheme entirely might be used, each inherently having its own unique failure modes which must be taken into account.

The focus of this chapter is an examination of the requirements for and implications of sharing location data among connected mobile nodes and using such data to adjust other positions. Shared data can be leveraged to improve individual positions as long as a determination can be made as to whether doing so is likely to be beneficial. This determination is based on probabilistic models derived from empirical evidence which allow for an estimation of both the approximate error in a remotely obtained position as well as the confidence level in that error estimate. For background, possible methods of combining shared location data are reviewed,

followed by an exploration of simple scenarios which demonstrate the strengths and weaknesses of each method.

4.2 Combination Methods

- **Assumed Collocation:** This is by far the simplest approach to leveraging shared data. A node that receives location information from another simply localizes itself at that position. A few assumptions are required for this method to be useful. First, as the name implies, both nodes must be at, or very near, the same physical location. Second, this method also requires an assumption that the node whose position is adjusted does not have any reasonable estimate of its own position. If such an estimate were available, it is easy to see why this approach would be less than ideal. Throwing out data, even with a fairly high error magnitude, which suggests spatial separation in favor of this approach would likely be ineffective at significantly improving localization in general. Averaging two separated positions, for instance, reduces the error attributable to spatial separation by as much as half. As co-opting a neighbor’s position is equivalent to averaging when the number of connected neighbors is one, the estimated error magnitude and confidence values associated with the shared position would be retained for both.
- **Averaging:** As an alternative to assuming collocation, multiple position samples can be combined through simple averaging. This method is intended to remove or, at a minimum, smooth out much of the random error which can occur in sensor data, thereby resulting in a more accurate and stable position. Averaging of the x and y coordinates of n position samples is

accomplished independently, i.e.:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

with standard error $\frac{\sigma}{\sqrt{n}}$. Because averaging is intended to remove the random error intrinsic to each data point, positions may be accepted and used without regard to any associated error or confidence levels. However, one of the most significant difficulties that arises when averaging positions lies in the assumption that the node for which other positions are averaged resides within a boundary established by the other points. Without any notion of its own position but with connectivity to nodes which all lie in roughly the same direction, averaging gives poor results. When nodes are connected in an ad-hoc fashion via wireless radios whose antennas lack the ability to determine angle of arrival, the assumption will likely fail, resulting in a wildly inaccurate position. One of the initial experiences with averaging in TeamTrak was a humorous example of what can go wrong. In this case, while waiting for an initial GPS fix, an operator whose tablet was “localized” using cached information (although the position itself happened to be correct) from a previous exercise was surprised to see his location suddenly jump dozens of meters away when connectivity was established to another machine of whose presence he had not been aware. Figure 4.1 illustrates this limitation.

- **Trilateration:** Trilateration is a method for determining relative positions among objects based on their measured or estimated distance and their relative bearing, both requiring prior localization of the reference points.

Using the distance from reference points, or anchors, and the relative bearing to each, determining the position of the unlocalized node is simple. Using multiple reference points at different bearings, the estimated position can be isolated to a single point, assuming no error. The presence of measurement error, both in the anchors' locations as well as in the distance measurement, changes the estimated position from a point to a region. As Figure 4.2 illustrates, the error of the laterated position depends on that of the anchors. In the figure, the shaded region represents the error of the estimated position which may be attributed to combined measurement errors in the anchor positions and in the distance measurements. Because trilateration generally does not permit smoothing out random error to any significant extent, the positions of the anchor nodes can not have large associated error.

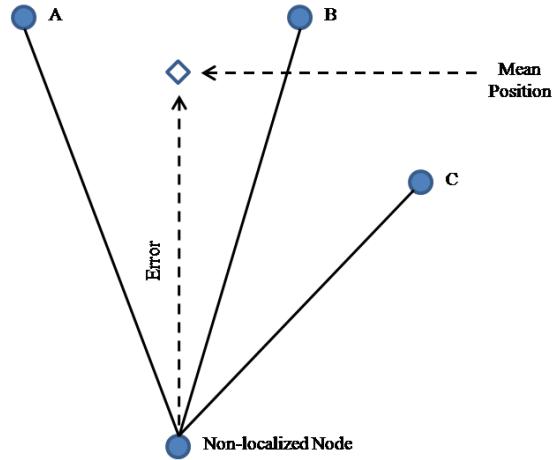


Figure 4.1: Averaging Positions (Not in Triangle)

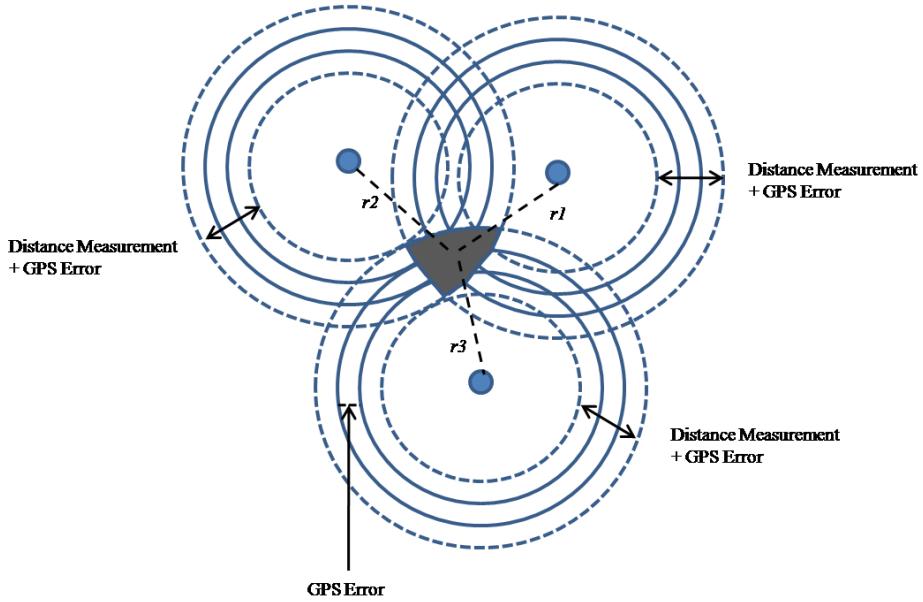


Figure 4.2: Trilateration

4.3 A Cooperative Approach to Localization

Despite the presence of sometimes significant error in GPS, a single sensor-laden node can localize itself with acceptable accuracy in most cases. However, the independence of GPS error and the presence of other localization techniques suggest that connected nodes will have varying degrees of location quality. Ideally, much like the stepwise refinement found in localization techniques for static sensor networks, the objective is to improve lower quality positions using higher quality positions and distance estimations, where applicable. Naturally, it is important to define what is meant by “improve” location quality. Measurement has two components: precision and accuracy. Accuracy refers to the “closeness” of the measurement to ground truth, while precision refers to the repeatability of subsequent measurement, as indicated by the variation among multiple samples. Cooperative data sharing can improve both under most but not all conditions, so

evaluation of this problem requires a scenario-based approach. First, a few simple cases that likely would never occur or be implemented in practice, but are intended for illustrative purposes, are presented. The following subsections describe these simple evaluation scenarios.

4.3.1 Single Localized Node

In this case, a single node has multiple redundant sensors affixed to it, and the error in each sensor's output occurs independently of the others. Such a scenario could occur, for instance in the case of dead reckoning, if one compass were mounted on a helmet, while a second one were mounted on a belt, and so on. In practice, compass error has a more significant impact on localization accuracy than stride length estimation in dead reckoning systems. With a sufficiently well-estimated constant k in the stride length estimation described in Chapter 3, total distance traveled, as defined by the number of steps detected multiplied by the estimated length of each stride, can be as accurate as within roughly three percent or better of the actual distance, assuming careful, deliberate steps and the user keeps the compass pointing in the actual direction of travel at all times. A compass heading error of 30 degrees, for example, would have a far greater effect on accuracy of dead reckoning localization over time than a stride length estimation error of 3 percent over any nontrivial distance, which for most applications is negligible, or at least well within acceptable limits. For this reason, *dead reckoning error*, while the result of error in data from a combination of sensors, is generally dominated by the measurement error in the compass heading that results from internal measurement errors in the hardware as well as any uncompensated effects of pitch and roll.

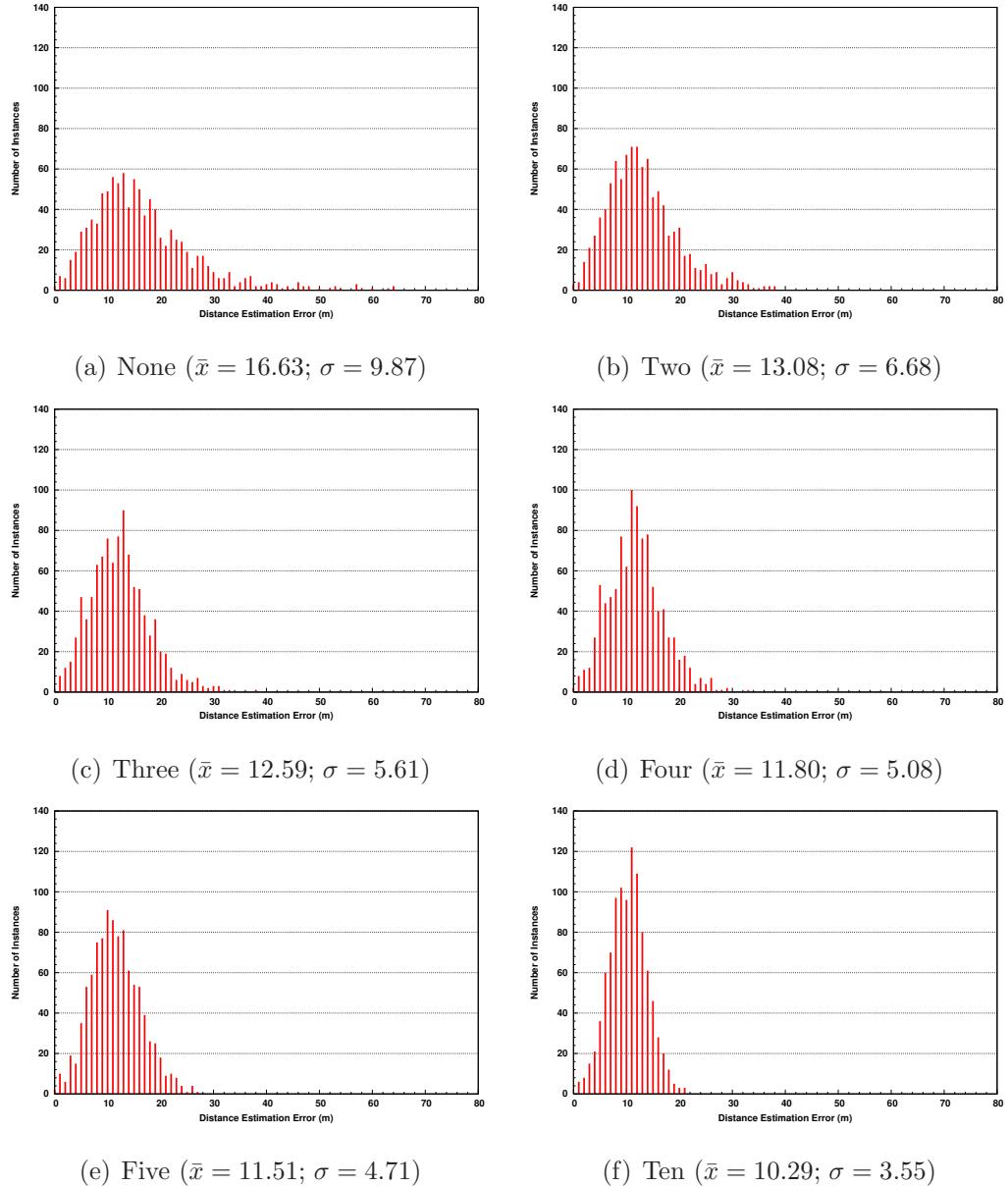


Figure 4.3: Dead Reckoning Averaging (w/GPS Error)

Using a single node removes many of the fundamental challenges arising from cooperative localization models and is described here for simplicity. Namely, distance measurement, an important but notoriously tricky component to implement in practice, is not a factor with collocated sensors. Additionally, all sensor data are measuring the same physical location, as all hardware is mounted on the same device or on the same human operator, so any resultant error can be attributed to random variation and can simply be averaged away. In a case such as this, trilateration is not an appropriate method, as the inter-node distance is zero and relative bearing is undefined. Additionally, assuming a single shared position makes no sense, despite the fact that the sensors are in fact collocated, because in this case multiple samples representing the same point exist, and one position will not be discarded in favor of another. Since all samples are independently measured estimates of the same physical location, the most reasonable cooperative method to use in this scenario is averaging. As mentioned earlier, averaging smooths the random jitter from the position estimates attributable to compass error and provides a more precise measurement as determined by the standard deviation σ of the dead reckoning error.

Figure 4.3(a) shows the distribution of error in the estimated position of the final point of a path of length 340 meters. In this experiment, a simulated node traversed a straight path. At the end of each trial, the Euclidean distance from the last point estimated by the dead reckoning system and the true final location is determined and recorded. The figure shows the distribution of these dead reckoning errors over 1,000 independent trials. The mean dead reckoning error, i.e., distance between estimated and true endpoints, is 16.63 meters with a standard deviation of 9.87. Figure 4.4 illustrates the true path taken and the corresponding

path estimated via dead reckoning for a single trial with no averaging. Each unit along the axes represents a distance of one meter. In each trial, dead reckoning begins with a GPS position determined from a probabilistic model based on data collected experimentally as described in Section 3.3.2 of Chapter 3. Compass error at each step is determined from a model based on the distribution of actual measurement errors observed in practice. In the figure, the solid line represents the actual path taken and the dotted line represents the same path estimated via dead reckoning, to include measurement and approximation errors.

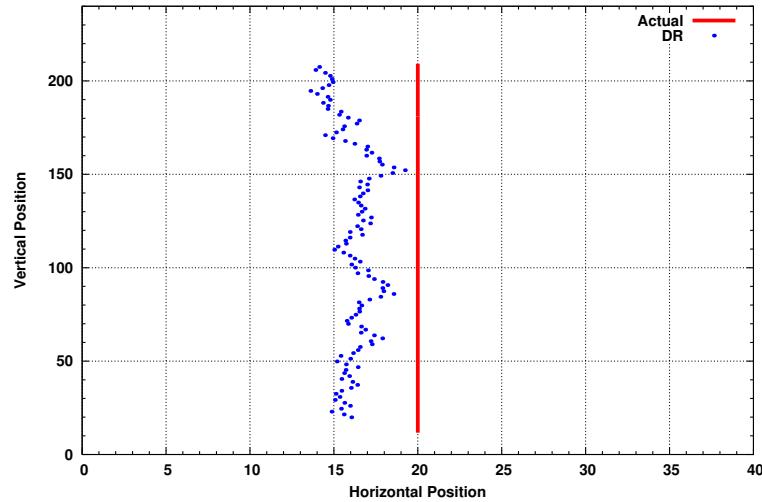


Figure 4.4. Dead Reckoning Versus Actual Path (One Node)

Figures 4.3(b) through 4.3(f) show the effect of averaging a varying number of positions estimated by independent sensors but representing the same physical

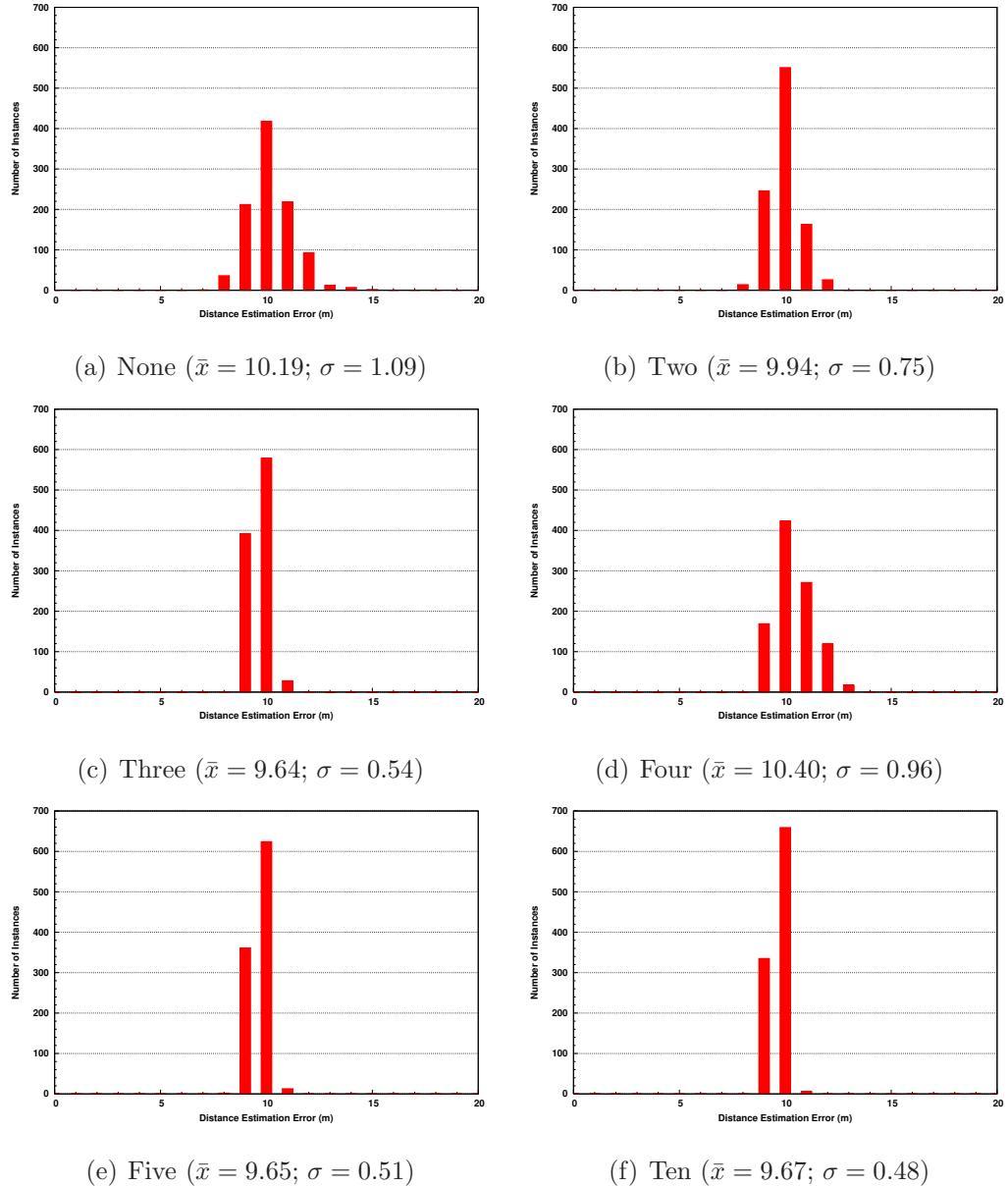


Figure 4.5: Dead Reckoning Averaging (No GPS Error)

location. Not surprisingly, these results demonstrate that averaging such positions reduces mean error and variance, which suggests that sharing data among collocated sensors can improve both precision and accuracy in dead reckoning systems. Note that the error may still be as large as approximately 20 meters when traversing a 340-meter path. This error is due to both the dead reckoning error and the error in the GPS positions which provide the initial starting point.

In both Figures 4.3 and 4.4, the dead reckoning technique begins with a position determined via GPS, which is subject to measurement error of its own. To evaluate the effect of averaging dead reckoning positions without the uncertainty of additional error, the assumption of the accuracy of the starting location is changed to a known, perfectly measured, fixed location. Figure 4.5(a) illustrates the effect of such a perfect starting location on dead reckoning accuracy without averaging. As before, the histogram shows the distribution of error in the final position over 1,000 independent trials. Figures 4.5(b) through 4.5(f) show the effects of averaging positions using a variable number of concurrent, but independent, samples each representing the same physical location. In general, both the mean error and standard deviation σ decrease with an increase in the number of position samples. Table 4.1 lists the mean error \bar{x} and standard deviation σ for each case.

TABLE 4.1
 EFFECT OF POSITION AVERAGING ON PRECISION AND
 ACCURACY (DEAD RECKONING)

Averaged Starting Mean				Averaged Starting Mean			
Samples	Position	Error	σ	Samples	Position	Error	σ
1	GPS	16.63	9.87	1	Actual	10.19	1.09
2	GPS	13.08	6.68	2	Actual	9.94	0.75
3	GPS	12.59	5.61	3	Actual	9.64	0.54
4	GPS	11.80	5.08	4	Actual	10.40	0.96
5	GPS	11.51	4.71	5	Actual	9.65	0.51
10	GPS	10.29	3.55	10	Actual	9.67	0.48

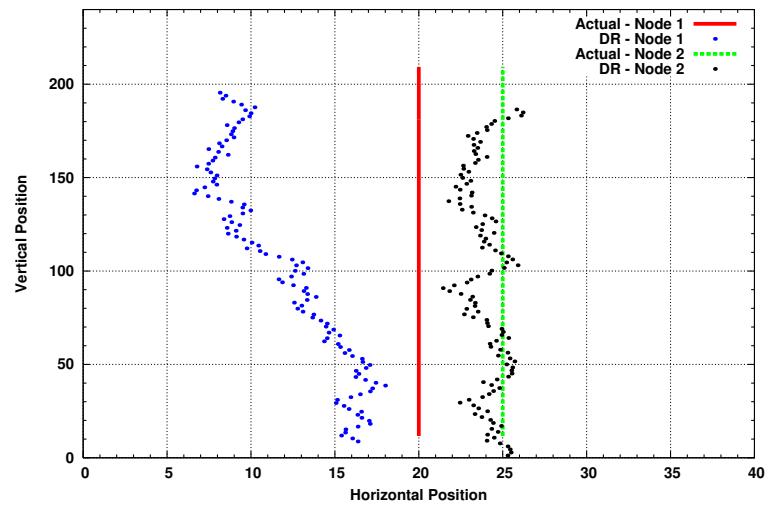
4.3.2 Two Localized Nodes

Evaluating the effects of averaging independent position estimates collocated on a single platform simplifies cooperative localization by assuming away the inherent difficulty in estimating or otherwise accounting for separation distance, or quantitatively evaluating location quality as a function of a random variable. In this section, the effect of sharing data among spatially separated independent nodes is evaluated. In each of these scenarios, the positions are not assumed to be averages of multiple samples. Figure 4.6(a) illustrates an example scenario with two nodes moving along parallel paths, while Figure 4.6(b) illustrates the

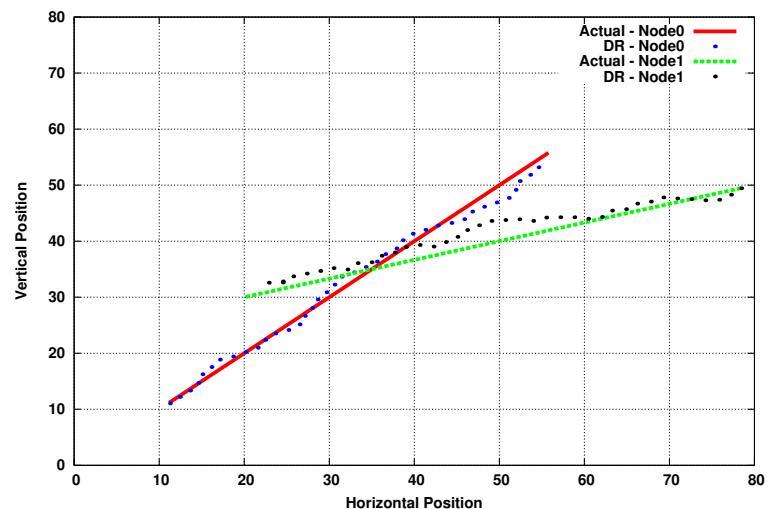
paths taken in a scenario in which assumptions about distance cannot be made. A scenario in which nodes are moving along parallel paths would occur, for instance, when two operators are walking together towards the same destination, a case commonly encountered among military troops, for whom walking alone in relatively high threat areas is not advisable. As noted in practice, two nodes physically present in close proximity may have much different location quality, even using identical localization hardware. In each of the figures, the solid line represents the actual path taken and the dotted lines are the approximated position at each step based on sensor data and illustrate the error associated with each position sample.

In scenarios in which two nodes travel along parallel paths, there are two cases to consider. First, a case in which an estimation of the distance between nodes is unimportant. An example of such a scenario would be two operators walking side by side at the same speed with no more than a meter or so separation between localized devices. For practical purposes, these devices could essentially share the same reported physical location since the granularity of the separation is sufficiently small that the distance would not matter for most applications. Both nodes, then, could share the same estimated position. Detection of remote nodes located within such a small distance could be accomplished using a wireless communication protocol such as Bluetooth.

The case of parallel paths at similar speeds is important for two reasons. First, it maximizes the amount of data which might be shared among nodes over the length of the path as they are always in communication with each other. Second, it permits additional simplifying assumptions that the general case of random motion does not. The possibility of sharing a position is one assumption. In



(a) Parallel Paths



(b) Nonparallel Paths

Figure 4.6. Dead Reckoning Versus Actual Path (Two Nodes)

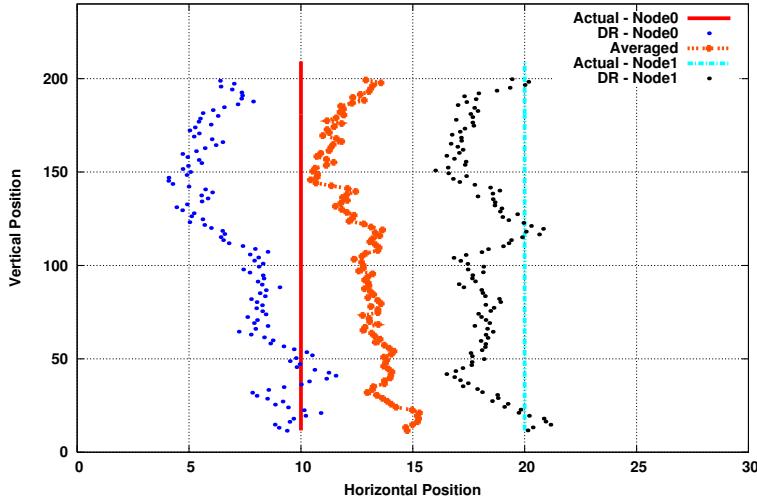


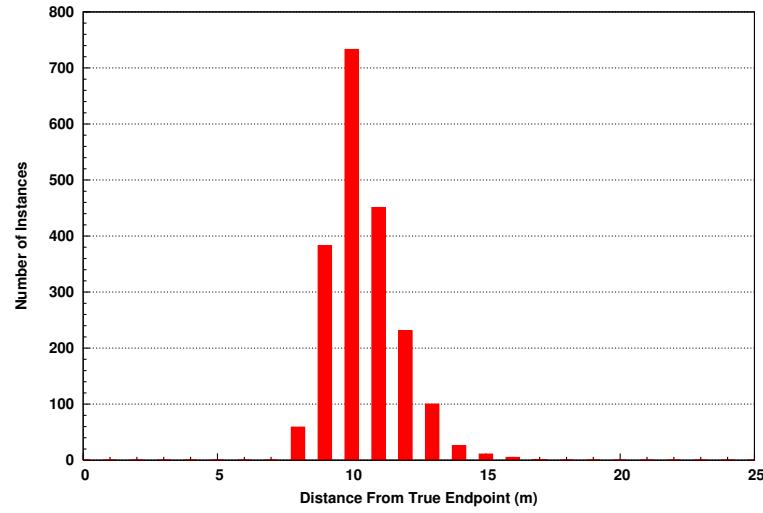
Figure 4.7: Fixed Separation Between Two Nodes

addition, the failure of one node’s sensors does not preclude its localization using cooperative data sharing, since location information may be available from the adjacent node. Since both can share a position at sufficiently close range, averaging shared data provides additional reliability through redundancy and elimination of random error. However, a small spatial separation is very important, as using an averaging technique with positions from both nodes will result in large error without sufficiently proximate placement. Similarly, since trilateration frequently offers little benefit without a sufficient number and placement of anchors, for a case in which two nodes of roughly equal location quality this method would not offer any meaningful improvement.

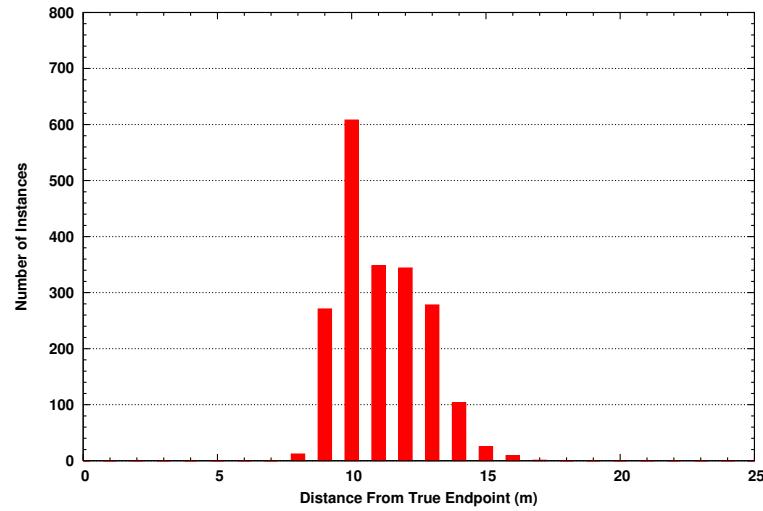
A more interesting scenario arises when operators traversing parallel paths are separated by some nontrivial or variable distance and collocation cannot be assumed. While location information can be shared, averaging might not be inappropriate as the error of the averaged position would be at best half the distance from each node to the centroid. If the separation distance is known, then a sim-

ple lateration scheme might be used to estimate the actual positions of both. Figure 4.7 illustrates such a case. However, this approach generally fails with two nodes because any lateration originating from the centroid simply restores the original error-prone position. Therefore, in this case assuming collocation is inappropriate and averaging positions is ineffective if the centroid is computed using shared dead reckoning information, even if the distance estimation between nodes is perfect. There is not enough available location information for either estimated position to converge on the true position. Dead reckoning error increases at roughly the same rate from the initial position, and there are no anchors present, so sharing data in this case would take increasingly bad positions and make them slightly to moderately worse. This is generally the case in scenarios with two nodes whose positions have roughly equal measurement or estimated error.

As an illustration, consider the graph in Figure 4.8. Figure 4.8(a) shows the distribution of error in the case of two nodes localized using dead reckoning without data sharing for 1,000 independent simulation trials with an actual separation of 10 meters. The starting positions for both nodes are determined with a GPS error model. In this case the mean error \bar{x} is 10.45 meters with a standard deviation σ of 1.30. If both positions are averaged, as shown in Figure 4.8(b), then \bar{x} increases to 11.09 meters and σ increases to 1.56, a six percent increase in \bar{x} and a 15 percent increase in σ , which suggests that in this case averaging reduces both the accuracy and precision of the positions. Therefore, while averaging positions can improve localization by reducing the jitter from compass error, it is only to the point at which separation between nodes becomes sufficiently large that error attributable to the spatial separation of nodes is greater than the tolerance of the application requirements.



(a) Dead Reckoning With No Averaging



(b) Dead Reckoning With Averaged Positions

Figure 4.8: Averaging Two Dead Reckoning Positions

4.3.3 More Than Two Nodes

In scenarios in which the paths traveled by the mobile operators are more random in nature and the number of nodes increases, assumptions used within the previous, much simpler cases no longer apply, and fewer methods may be used for combining positions. The most significant challenge when dealing with arbitrary paths is accounting for spatial separation between nodes with a wireless connection. Unfortunately, the current state of the art in accurately measuring distance using wireless connectivity is quite poor. However, there is some utility in using radio frequency propagation for localization purposes. Consider a case in which a device is completely unlocalized, yet is able to connect to another node which has a GPS fix. Clearly it is reasonable to conclude that the unlocalized node is within some radius of the GPS location, with the radius bounded by the maximum wireless propagation distance in the specific environment. If the application requirement is such that even a very rough approximation of position is acceptable, assuming collocation can be a reasonable approach.

In the absence of a reliable method for estimating distance using wireless connectivity, and alternative approach to dealing with spatial separation is simply to ignore it altogether and average the positions of connected remote nodes. Doing so can be beneficial in cases such as previously described, in which a node is fully unlocalized or has such a substantial position error that even an error on the order of the communication range would be an improvement over its existing position or perhaps even acceptable for the application.

4.3.4 Connectivity

For purposes of cooperative localization, the collocation method may be unacceptable in many cases. As demonstrated in Section 3.3.2 in Chapter 3, error attributable to GPS drift can be as large as several kilometers. Coupled with possibly several hundred meters of wireless radio range, a collocated approach can exhibit very substantial localization error. Using a cooperative approach over non-trivial distances to establish an initial position may in many cases be ineffective in general, at least with a very small number of anchors, and experiences with very small scale experiments have shown that to be the case. However, as before, a poor quality location may be better than none. Instead of further examining collocation, this work explores the use of wireless connectivity in cases in which all nodes have at least some notion of their own location to varying extents, although some of which may be of very poor quality. Of course, it is important to note, although should be obvious, that wireless connectivity is not a singular, monolithic entity. Several types of wireless protocols can be considered for localization purposes, three of which are examined here:

1. **Short Range Connectivity.** At very close range, Bluetooth connectivity can be used to determine proximity between nodes. Since most commercial Bluetooth radios have a maximum effective range of 10 to 15 meters for Class 2 radios or approximately one meter for Class 3 (Bluetooth SIG, 2004), this may possibly be an effective method of detecting nearby localized nodes for purposes of sharing location information. Moreover, short range connectivity may also be used among multiple connected nodes to determine outliers among positions in close proximity.

2. Intermediate Range Connectivity. As shown in Chapter 3, the use of 802.11 wireless Ethernet in ad-hoc mode can be another method for detecting proximity and roughly estimating distance. The effective range of wireless Ethernet varies, but previous experiments in this work show that it can be determined for a particular wireless card with acceptable reliability. In the case of the Intel PRO/Wireless 2915ABG wireless card used in the TeamTrak platform, a maximum consistent wireless range of roughly 300 meters was empirically determined, as shown in Figure 4.9. This figure was constructed using data from an outdoor exercise involving 14 GPS-enabled nodes moving around the Notre Dame campus over a period of an hour. One of the machines consistently experienced substantial GPS error; its data was excluded. The others show the distance and direction at which at least one-way connectivity was established between pairs of nodes. The graph is a composite of all connectivity between all exercise participants.

3. Long Range Connectivity. WiMax (IEEE 802.16 Working Group on Broadband Wireless Access Standards, 2005) is a much more recently introduced technology that, while yet to be widely adopted, potentially can be used for purposes of localization by detecting the presence of nodes that are far more widely dispersed, as it has maximum effective range of as much as several kilometers.

With each type, the establishment of a connection is what facilitates localization, particularly trilateration, since the edge of wireless range can provide a somewhat reliable upper bound on distance estimation. Of course, the presence of substantial obstructions and other factors which affect RF propagation and would commonly be found in urban environments reduce the accuracy of any dis-

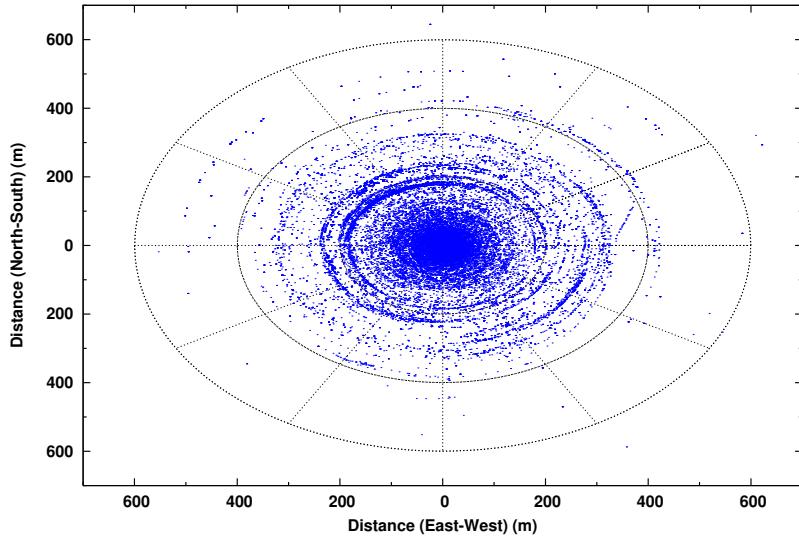


Figure 4.9: Wireless Connectivity

tance estimation, particularly for long range connectivity such as WiMax, but at much shorter ranges such methods can be useful. It is important to note that for trilateration, the distance at which an initial connection between node pairs occurs is what is important; once a connection has been established and there is further mobility within each node's communication range, the distance estimation becomes far less reliable with the number of steps taken by either node's operator.

As an aside, there has been much work towards constructing reliable distance estimation techniques within the range of wireless connectivity, to include methods such as received signal strength, time of arrival (the technique used in the GPS system), time difference of arrival, angle of arrival, and so on. However, none have proved to be very accurate or reliable except under very carefully controlled conditions or, in the case of GPS, augmented with heavy infrastructure, precise clock synchronization, and maintenance tasks infeasible at a much smaller scale.

Existing methods of precision distance estimation over more than trivial ranges using wireless signals are ineffective or impractical for systems such as mobile ad-hoc networks. The best currently available method for gauging distance under outdoor conditions in urban settings is the maximum wireless propagation model for Bluetooth, 802.11, and WiMax, but is largely limited to purposes of detecting errors (and possibly correcting at short range, i.e., Bluetooth Classes 2 and 3). Some research is currently being done with laser range finders, which may not be suitable for production systems, but is sufficiently accurate to evaluate ideas in cooperative localization at this time. However, such work is still in its relative infancy.

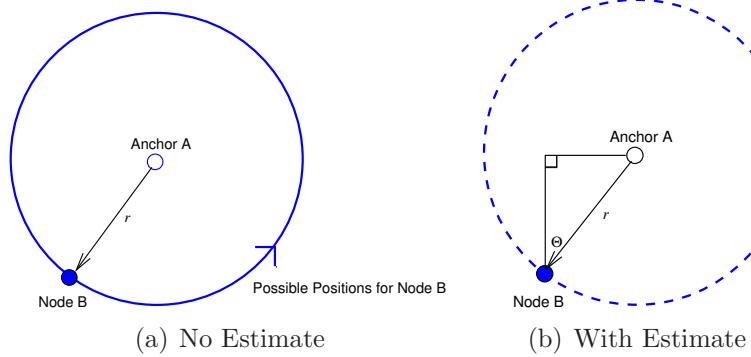


Figure 4.10: Trilateration With Two Nodes

Localization in the case of two mobile nodes at the edge of wireless connectivity can be accomplished through trilateration as long as the node to be further localized has some notion of its own position. Otherwise, the estimated position can only be determined to lie on the edge of a circle (assuming roughly isotropic

RF propagation) at a distance of r , where r is the maximum wireless range of the anchor node. Figure 4.10(a) illustrates the possible range of positions without an initial estimate, while Figure 4.10(b) shows the effect of an initial estimated position. Given the importance of an initial estimate, using a cooperative localization approach, even if it does not produce an ideal result immediately, may produce a sufficient initial estimate for further improvement. Of course, the greater the variability in RF propagation, and thus the more uncertainty in the distance measurement, the more stepwise refinement required.

With an approximated position for the unlocalized node the relative bearing between it and the anchor can be estimated, which can then reduce the number of points at which the adjusted position can lie. The presence of additional anchors further narrows the number of possible points as well. The unpredictable connectivity patterns in mobile ad-hoc networks imply that a desirable number of anchors may not be available at all times. This variability affects the quality of locations of other nodes attempting to localize based on available anchors sharing location data while moving in various patterns.

Throughout this section, a scenario-driven approach to evaluation is used, with each scenario derived from a single, overarching problem domain. The problem statement can be specified as follows:

Situation: Assume n mobile nodes in the field. A subset of nodes have a functioning GPS receiver. Those that do not rely on dead reckoning for localization. The initial reference point for dead reckoning is determined via GPS, but after establishing the initial point, GPS is no longer available to those nodes. Each delta in position computed by the dead reckoning apparatus is affected by both compass error and stride length estimation error at each step. Additionally,

all GPS positions have measurement error, which may be substantial.

Objective: Each node must compute an estimate of its current physical location to display to its human operator. Nodes may combine information collected from multiple remote sources to determine a final result.

Hypothesis: Quality of position estimates from dead reckoning can be improved by sharing information once measurement error grows sufficiently large.

The remainder of this section describes two methods of combining shared location samples based on whether distance measurement is available.

4.3.5 Combining Positions

When using shared location data, ultimately the non-localized or poorly-localized node must combine this information to estimate its own position. Regardless of the type of sensors used for localization or the communication range of the mobile nodes, remote positions are acquired, and these are used to establish or improve its position using one of several methods:

Simple Averaging. With this approach, the mean position of all n nodes within communication range of node \mathcal{N} , as well as \mathcal{N} 's own position, is determined, and this mean position is used as the new “interpolated” position. The average x and y positions, \bar{x} and \bar{y} , respectively, are determined separately:

$$\bar{x} = \frac{1}{n+1} \left(x_{\mathcal{N}} + \sum_{i=1}^n x_i \right) \quad \bar{y} = \frac{1}{n+1} \left(y_{\mathcal{N}} + \sum_{i=1}^n y_i \right)$$

The algorithm for the case of simple averaging connected nodes is as follows:

Averaging (Non-Inclusive). In this case positions of remote peers are averaged as in the simple averaging case, but no local position is included. Such a

Algorithm 1 Simple Averaging

```

1:  $(x, y, e) \leftarrow \text{Location-Use-Best}(\mathcal{N})$ 
2: if Location-Source( $\mathcal{N}$ ) = GPS then
3:   return  $(x, y, e)$ 
4: end if
5: for  $i = 1$  to  $n$  do
6:   if Connected( $\mathcal{N}, i$ ) then
7:      $\bar{x} \leftarrow \bar{x} + x_i$ 
8:      $\bar{y} \leftarrow \bar{y} + y_i$ 
9:      $e_{total} \leftarrow e_{total} + error_i$ 
10:     $count \leftarrow count + 1$ 
11:   end if
12: end for
13: if  $count \neq 0$  then
14:    $\bar{x} \leftarrow \bar{x} / count$ 
15:    $\bar{y} \leftarrow \bar{y} / count$ 
16: end if
17: return  $(\bar{x}, \bar{y}, e_{total})$ 

```

scenario would occur among nodes which are entirely non-localized; for instance, a node with neither GPS capability nor any starting position for dead reckoning.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Selective Averaging. This method only averages the k remote positions whose estimated error is no greater than that of \mathcal{N} . The purpose of accounting for error in such a way is to eliminate very poor positions which might skew the mean position, particularly among a small sample of remote peers.

$$\bar{x} = \frac{1}{k} \sum_{\forall i \mid e_i \leq e_{\mathcal{N}}} x_i \quad \bar{y} = \frac{1}{k} \sum_{\forall i \mid e_i \leq e_{\mathcal{N}}} y_i$$

Filtered Trilateration. Trilateration is a commonly proposed method of localization in sensor networks. The most significant advantage is that it can

provide a very precise and accurate position, assuming the proximity of anchors. An unlocalized node uses the positions of neighboring anchors nodes in conjunction with a distance estimation to estimate its own position. The algorithm used in the evaluation sections of this chapter is as follows:

To filter positions for trilateration from peers in the presence of both localization error and mobility, let $P_{\mathcal{N}}$ be the (possibly empty) set of all location samples of preferred quality and confidence level at Node \mathcal{N} , the node to be localized. Here, τ is the error tolerance and c_{min} is the minimum confidence level for each error value as determined for the application. Positions of lower error and higher confidence are accepted.

$$P_{\mathcal{N}} = \{(x, y, e, c) \mid (x \in \mathbb{R}) \wedge (y \in \mathbb{R}) \wedge (e \leq \tau) \wedge (c \geq c_{min})\} \quad (4.1)$$

and $P'_{\mathcal{N}}$ be the (possibly empty) set of all location samples of arbitrary quality and confidence levels generated by \mathcal{N} as it self-localizes, which may include positions of significant error magnitude or low confidence:

$$P'_{\mathcal{N}} = \{(x, y, e, c) \mid (x \in \mathbb{R}) \wedge (y \in \mathbb{R}) \wedge ((e > \tau) \vee (c < c_{min}))\} \quad (4.2)$$

We take $\ell_{\mathcal{N}}$ to be the locally determined position of highest quality:

$$\ell = \begin{cases} (x_0, y_0, \infty, 0.0) & \text{if } (P_{\mathcal{N}} \cup P'_{\mathcal{N}}) = \emptyset \\ (x_i, y_i, e_i, c_i) \mid \forall i \forall j ((p_i, p_j \in P_{\mathcal{N}}) \wedge (p_i.e = \min\{p_j.e\})) & \text{if } P_{\mathcal{N}} \neq \emptyset \\ (x_i, y_i, e_i, c_i) \mid \forall i \forall j ((p_i, p_j \in P'_{\mathcal{N}}) \wedge (p_i.e = \min\{p_j.e\})) & \text{if } P_{\mathcal{N}} = \emptyset \end{cases}$$

Let L be the set of all location samples among n nodes directly connected to \mathcal{N} :

$$L = \{(x, y, e, c) \mid (x \in \mathbb{R}) \wedge (y \in \mathbb{R}) \wedge (e \in \mathbb{R}) \wedge (c \in [0, 1])\} \quad (4.3)$$

There are five cases which must be considered depending on the estimated error value e and the associated confidence level c of each remote peer node i , where $i = 1..n$.

- Case 1: $e_i \leq \tau$ and $c_i \geq c_{min}$
- Case 2: $e_i \leq \tau$ and $c_i < c_{min}$
- Case 3: $e_i > \tau$ and $c_i < c_{min}$
- Case 4: $e_i > \tau$ and $c_i \geq c_{min}$
- Case 5: $e_{\mathcal{N}} \leq e_i$

Which case represents the minimum threshold for acceptance depends on the application, and different cases can produce much different levels of accuracy. For each of the above cases, we find the set of remote positions R and corresponding distances D among nodes in R which are directly connected to node \mathcal{N} , whose best position is represented by $\ell_{\mathcal{N}}$:

$$R = \begin{cases} \{p \in L \mid (p \notin (P_{\mathcal{N}} \cup P'_{\mathcal{N}})) \wedge (p.e \leq \tau) \wedge (p.c \geq c_{min})\} & \text{if case 1} \\ \{p \in L \mid (p \notin (P_{\mathcal{N}} \cup P'_{\mathcal{N}})) \wedge (p.e \leq \tau) \wedge (p.c < c_{min})\} & \text{if case 2} \\ \{p \in L \mid (p \notin (P_{\mathcal{N}} \cup P'_{\mathcal{N}})) \wedge (p.e > \tau) \wedge (p.c < c_{min})\} & \text{if case 3} \\ \{p \in L \mid (p \notin (P_{\mathcal{N}} \cup P'_{\mathcal{N}})) \wedge (p.e > \tau) \wedge (p.c \geq c_{min})\} & \text{if case 4} \\ \{p \in L \mid (p \notin (P_{\mathcal{N}} \cup P'_{\mathcal{N}})) \wedge (q \in (P_{\mathcal{N}} \cup P'_{\mathcal{N}})) \wedge (p.e \leq q.e)\} & \text{if case 5} \end{cases} \quad (4.4)$$

$$D = \{d \mid r \in R \implies d = \sqrt{(r.x - \ell_N.x)^2 + (r.y - \ell_N.y)^2}\} \quad (4.5)$$

Next, use the points contained in the set R and the pairwise distances to create a set of points \hat{P}_N , with each member representing a possible position of node N . Here, θ_i represents the bearing from N to node i .

$$\hat{P}_N = \{p_i \in R, d_i \in D \mid ([p_i.x - d_i \cos \theta_i], [p_i.y - d_i \sin \theta_i], e_i, c_i)\} \quad (4.6)$$

Then average the positions:

$$p = \begin{cases} \left(\frac{1}{n} \sum_{p \in \hat{P}_N} p[x], \frac{1}{n} \sum_{p \in \hat{P}_N} p.y \right) & \text{if } \ell = (x_0, y_0, \infty, 0.0), \\ \left(\frac{1}{n+1} \left(\ell_N.x + \sum_{p \in \hat{P}_N} p.x \right), \frac{1}{n+1} \left(\ell_N.y + \sum_{p \in \hat{P}_N} p.y \right) \right) & \text{otherwise.} \end{cases}$$

Thus \bar{p} , the estimated position of N , is determined with a simple filtered trilateration algorithm (in this example for Case 1), which is shown in Algorithm 2. Algorithm 3 is used to determine whether correction is required for node N with associated set of positions P determined via local sensors.

4.4 Experimental Setup

Evaluation of the scenarios in this section is accomplished via discrete-event simulation, much like the work in sections 4.3.1 and 4.3.2 using the TT-Sim simulation environment. In this section, the effect of sharing location data among a larger number of nodes moving in random patterns is studied and attempting to improve positions using the combination methods described in Section 4.3.5.

Simulation parameters are determined empirically based on data collected us-

Algorithm 2 Filtered Trilateration

```
1: if Correction-Required( $\mathcal{N}$ ) then
2:   ( $x, y, e, c$ )  $\Leftarrow$  Location-Use-Best( $\mathcal{N}$ )
3:   for  $i = 1$  to  $n$  do
4:     if  $i \neq \mathcal{N}$  and Connected( $\mathcal{N}, i$ ) then
5:        $L[i] \Leftarrow$  Location-Use-Best( $i$ )
6:       if  $L[i].e \geq \tau \wedge L[i].c \geq c_{min}$  then
7:          $R[i] \Leftarrow L[i]$ 
8:          $\theta[i] \Leftarrow \arctan((L[i].y - y) / (L[i].x - x))$ 
9:          $d[i] \Leftarrow \sqrt{(x - L[i].x)^2 + (y - L[i].y)^2} + \epsilon$ 
10:         $R[i].e \Leftarrow |R[i].e| + |\epsilon|$ 
11:      end if
12:    end if
13:  end for
14: end if
15: if  $R = \emptyset$  then
16:   return  $(x, y, e, c)$ 
17: else
18:   for  $i = 1$  to  $n$  do
19:     if  $R[i] \neq 0$  then
20:        $R[i].x \Leftarrow R[i].x - d[i] \cos \theta[i]$ 
21:        $R[i].y \Leftarrow R[i].y - d[i] \sin \theta[i]$ 
22:     end if
23:   end for
24: end if
25:  $\bar{x} \Leftarrow \frac{1}{\#R} \sum_{r \in R} r.x$ 
26:  $\bar{y} = \frac{1}{\#R} \sum_{r \in R} r.y$ 
27:  $e \Leftarrow \frac{1}{\#R} \sqrt{\sum_{r \in R} (r.x - \bar{x})^2}$ 
28:  $c \Leftarrow \min \forall r \in R \mid r.c$ 
29: return  $(\bar{x}, \bar{y}, e, c)$ 
```

Algorithm 3 Correction-Required

```
1: for  $i \in P$  do
2:   if  $\min(P[i].e) \leq \tau$  and  $P[i].c \geq c_{min}$  then
3:     return false
4:   else
5:     return true
6:   end if
7: end for
```

ing TeamTrak discussed in Chapter 3. For GPS error, the magnitude of the error and both the expected length of the drift and the expected direction towards which the position drifts given a specific GPS error are explicitly modeled. The distribution of errors in recorded GPS positions from both receivers was incorporated into a single CDF. The model of GPS error takes the actual position of a node at a given time and perturbs it with an offset based upon the CDF. A pseudo-random number is generated using `drand48()`, the value of which is input to an inverse CDF to obtain the magnitude of the error. Once the error value is determined, a similar approach is used to determine the length of the drift and the bearing of that specific error, again based on empirical data incorporated into probabilistic models.

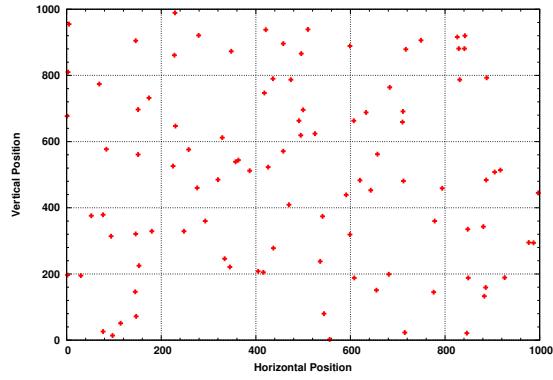
A similar approach is taken for dead reckoning error. Since the distribution of compass error is generally Gaussian, a random number with such a distribution is generated within the range observed in live experiments with three-axis compass hardware. The mean and standard deviation of compass error were empirically determined. Similarly, the error in stride length is determined, and is combined with heading to determine the next measured position (again, independently of the actual position).

The simulation is intended to model a localized mobile ad-hoc network at steady state. The initial placement of nodes is random with uniform distribution across the simulation field. Various network densities were modeled to simulate operations over broader ranges, as illustrated in Figure 4.11. For each evaluation scenario, a high, intermediate, and low density network was modeled. Mobility is modeled using Random Waypoints, a method selected due to its generality, i.e., most real-world scenarios would not involve random patterns of motion and

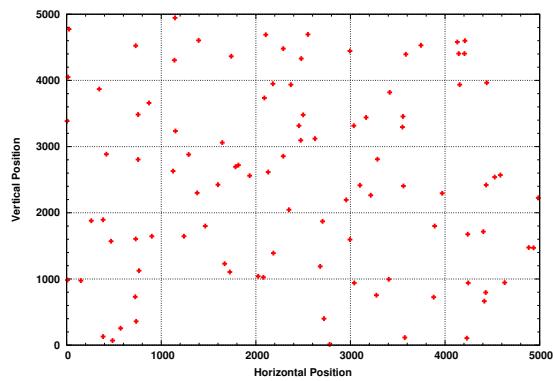
the use of a random model in this work is not likely to produce overly optimistic results. Finally, the speed of each agent in the simulation is random with a mean of 1.56 meters per second, the average walking speed of an adult male, and with normal distribution.

In the cases of trilateration, at each step in the simulation, every node evaluates its own position based on estimated error and the confidence in that error based on probabilistic models. For dead reckoning nodes, the quality of the position decays based on compass error and stride length estimation models. Once a node's position is determined to be either outside the application tolerance, which in these trials is 15 meters, or below a threshold of 70% confidence, it is deemed to require correction. The node's position is estimated based on the positions of and distances to its neighbors, but only using those positions of sufficiently high quality and confidence, regardless of the specific source of localization. With the exception of the results described in Section 4.5.9, for simpler evaluation scenarios, no self-assessment of localization error is accomplished.

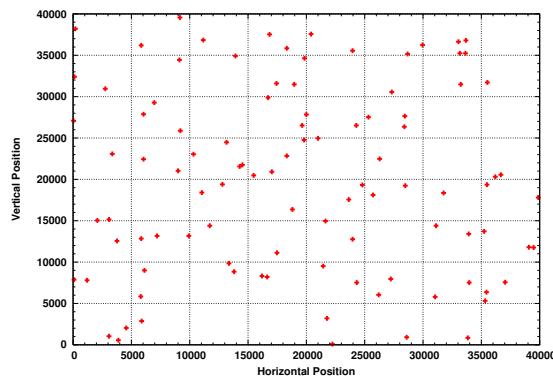
Because dead reckoning error itself is generally a function of the number of steps taken, over time this error can significantly eclipse the measurement error of the initial GPS position. Therefore, running simulations for longer periods of time would result in higher average error, but such error changes proportionately with the length of the simulation. In this paper, experiments simulate motion over a period of 20 minutes, over which our experiences with such systems show is sufficient time for uncorrected dead reckoning error to grow quite large. Figure 4.12 illustrates. In the figure, the length of a simulation was gradually increased, and each curve reflects the mean position error of all points over all steps in the simulation. Increases in the number of steps results in an increase in the mean error,



(a) High Network Density (1 km^2)



(b) Medium Network Density (25 km^2)



(c) Low Network Density ($1,600 \text{ km}^2$)

Figure 4.11: Example Initial Node Placements

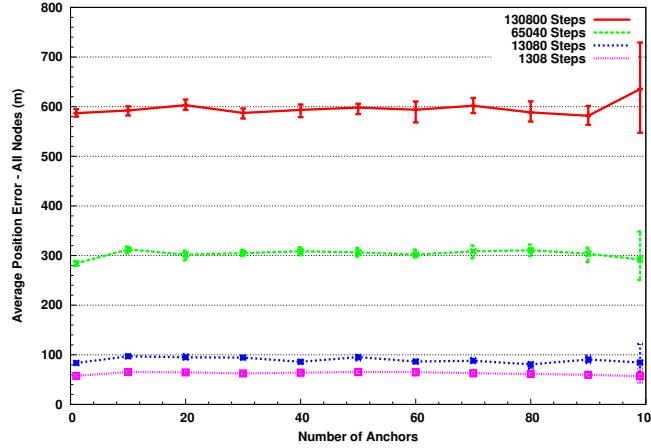


Figure 4.12. Effects of Simulation Time on Mean Position Error

as would be expected. The variability in the case of 99 anchors reflects the variability in dead reckoning error for a single node, which unlike the other cases, does not become smoothed out by averaging over multiple nodes. Since this case only examines dead reckoning error, the number of anchors is irrelevant to the mean error here, but was varied in simulation trials and included in the figure as a basis for comparison in later figures.

4.5 Results

While Sections 4.3.1 and 4.3.2 demonstrated the improvement in location accuracy through averaging independent samples, in those cases there was either no spatial separation or the distance between nodes was both fixed and known. This section examines the effects of combining multiple position samples where there is a variable and non-trivial separation that may or may not be known, and connectivity between any set of nodes may be sporadic. Section 4.5.1 discusses the effects of simply averaging the reported positions of connected nodes, while

Sections 4.5.8 and 4.5.9 discuss variations on simple averaging which exclude some positions based on specified criteria. Finally, Section 4.5.10 discusses the effect on position error using the filtered trilateration technique presented in Section 4.3.5.

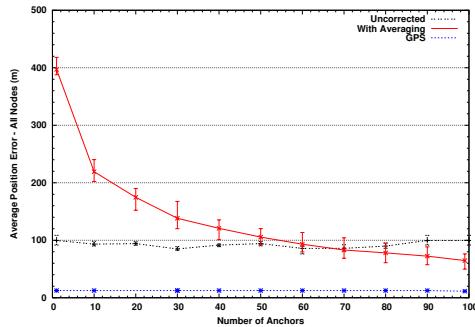
4.5.1 Simple Averaging: 300-Meter Range

In this scenario, any node whose location is determined via dead reckoning attempts to correct its location by averaging its own position with the positions of all nodes directly connected within its communication range without regard to the source of each neighbor's position, i.e., nodes use Algorithm 1 to combine positions and nodes localized via GPS, dead reckoning, or cooperative averaging may be considered anchors. Each position included in the average is weighted equally. Nodes each have a variable communication range between 200 and 300 meters depending on direction, intended to model the anisotropic RF propagation commonly found in wireless radios. At each step, prior to any averaging, a node first localizes itself using the best possible location information using estimated error, which may be either its uncorrected dead reckoning position determined from the sensors or an averaged position from the previous step. With 100 nodes in total, the number of GPS anchors in the system was increased in increments of 10. Rather than consider less interesting cases in which there are zero or 100 anchors, scenarios with either a single anchor or 99 are included. Previous experiences attempting to correct erroneous GPS positions by averaging other GPS positions proved fruitless; therefore the case of averaging in which all 100 nodes are GPS-enabled anchors is not examined, nor is any attempt made to correct one GPS position with another in the cases of simple averaging. If a GPS fix is available, using it uncorrected for localization is probably the better option compared to

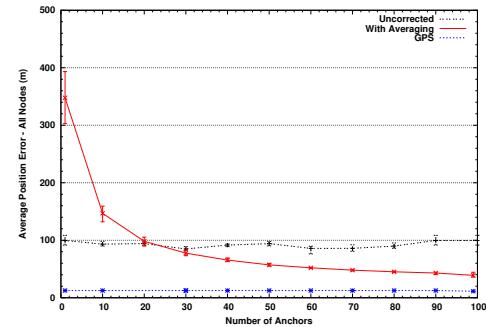
averaging remote positions which may lie in a radius of up to 300 meters. Since the measurement error of each remote position might lie even farther away from the position to be adjusted, a relatively small position error could quickly become quite large. When averaging a GPS position with a dead reckoning position, there is a tendency for a relatively accurate position to be skewed, often dramatically, by another with a high estimated error, and this worsening quickly propagates among the connected nodes in the system.

Figure 4.13 shows the effect of simple averaging for dead reckoning nodes with at least one GPS anchor among them. Each subfigure 4.13(a) through 4.13(c) shows the mean position error of all nodes over all time steps in the simulation over 10 independent simulation runs. In each of the three network densities modeled, increasing the number of anchors in the system reduces the overall average position error, but to realize any improvement in mean error, the ratio of anchors to other nodes would likely need be at least 70 percent, but to achieve significant improvement, that ratio would likely need to be even higher.

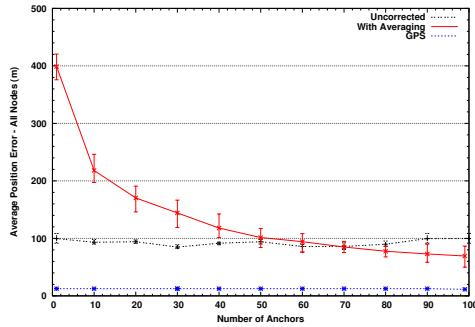
It is important to note that when averaging, the error in the adjusted positions is a function of both the separation between nodes and the dead reckoning error. Even in cases where dead reckoning positions are freshly reset, such as at the beginning of each simulation trial, the localization error after adjustment is highly dependent on the arrangement of connected anchors. Existing localization techniques such as APIT assume an unlocalized node is surrounded by anchors relatively symmetrically. In a case where mobility invalidates such an assumption in general, having very few anchors results in an average position error for dead reckoning nodes that is several times worse than in the case of doing nothing.



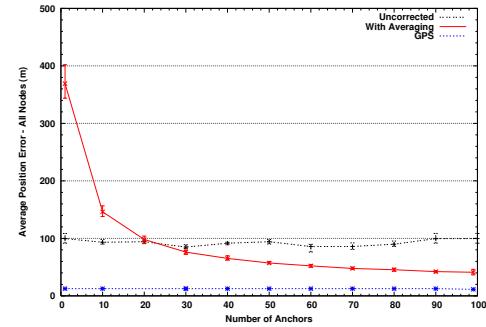
(a) High Network Density



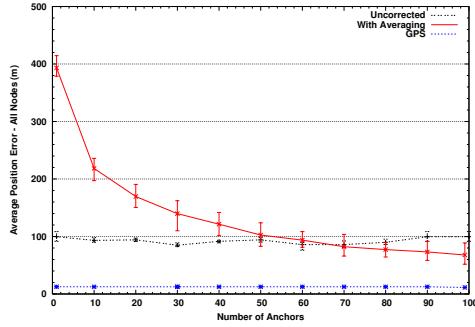
(a) High Network Density



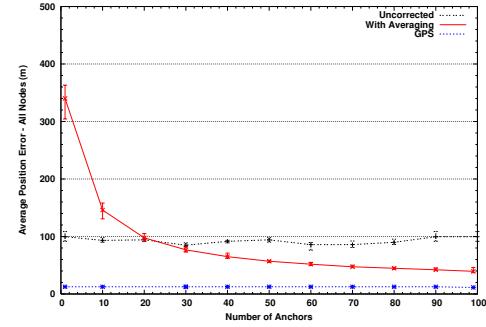
(b) Medium Network Density



(b) Medium Network Density



(c) Low Network Density



(c) Low Network Density

Figure 4.13: Averaging (300 m Range)

Figure 4.14: Averaging (150 m Range)

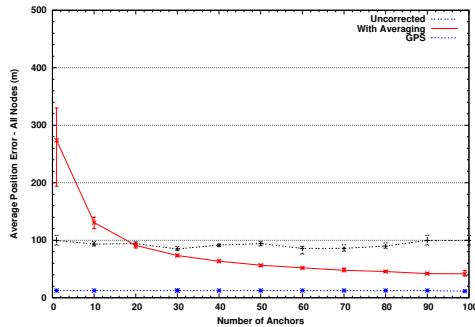
4.5.2 Simple Averaging: 150-Meter Range

Because position error among averaged samples is dependent on both measurement error among each anchor's reported position and the spatial separation between nodes, particularly given an asymmetrical arrangement of anchors, a reduction in such error can be effected in part by reducing the communication range, thereby only considering positions of more proximate nodes.

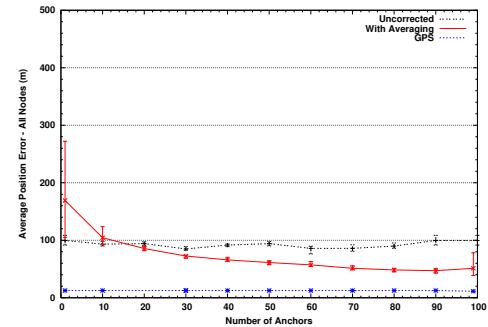
As in the case of 300-meter communication range, Figure 4.14 also shows the effect on the average position error of all nodes among over all time steps, but in this scenario the communication range is reduced to a maximum of 150 meters. In this configuration, while the mean position error in the worst case (a configuration with a minimal but nonzero number of available anchors) is roughly the same as that in the case in which nodes have a 300-meter communication range. However, simply by averaging, the mean position error is reduced to less than half that of uncorrected dead reckoning while using fewer anchors, i.e., in this case 20 as opposed to 50 or more out of 100 nodes in total when sharing location data over a much longer range. As in the previous case, increases in the total number of anchors effect reductions in the mean position error overall. It should be noted that in the worse cases, despite a reduction in the mean position error compared to that of the case of a 200- to 300-meter range, average error remains several times worse than that of the uncorrected case. To provide a meaningful correction scheme, these errors should be reduced further.

4.5.3 Simple Averaging: 125-Meter Range

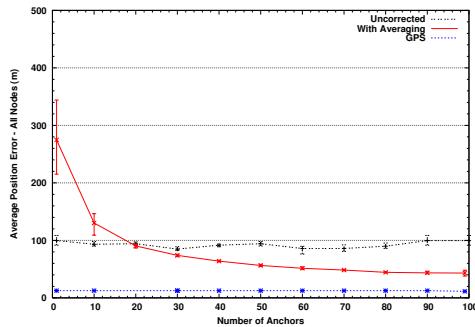
Further scaling back the communication range results in a further reduction of the mean position error in the cases where few anchors exist, as shown in



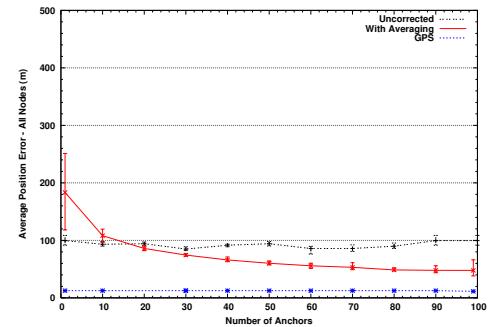
(a) High Network Density



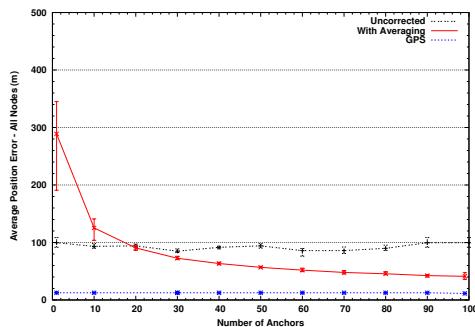
(a) High Network Density



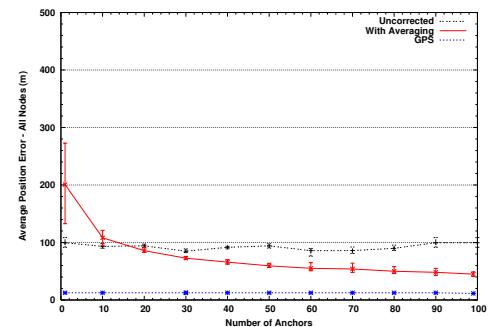
(b) Medium Network Density



(b) Medium Network Density



(c) Low Network Density



(c) Low Network Density

Figure 4.15: Averaging (125 m Range)

Figure 4.16: Averaging (100 m Range)

Figure 4.15, while still retaining the reduction in mean error exhibited in the previous cases. In this simulation scenario, the effects of different network densities still do not show to any significant extent, suggesting that the mean position error is less sensitive to density than it is to communication range, i.e., whether the case is very frequent averaging which occurs in the highly dense network or very sporadic averaging which occurs in the low density network, such adjustments do not affect the mean position error as much as changes over a long distance. This also appears to be the case when connectivity is frequent enough to impact the mean position error, but sufficiently infrequent that the compounding dead reckoning error may grow to be fairly large. With an intermediate node density, a reduction in the communication range to a maximum of 125 meters produces an average position error in the worst case of less than four times the mean in comparison to doing nothing, but begins to realize improvement in position error at or above a roughly 20 percent ratio of anchors to other nodes.

4.5.4 Simple Averaging: 100-Meter Range

In this scenario, the communication range is reduced to no more than 100 meters, and again nodes without local GPS information average their positions with all of their connected neighbors, the effect of which is shown in Figure 4.16. In this configuration, the mean error in the worst case is less than twice that of the uncorrected case, and the variance is less than three times. Increasing the number of anchors in the system results in an improvement in the dead reckoning positions by more than 50 percent in the cases of networks which are relatively saturated with anchors. The majority of the improvement resulting from averaging is due to the prevention of growth in the dead reckoning error over a large

number of steps. The mean error with uncorrected dead reckoning, as shown in Figure 4.12 is dominated by the error towards the end of each simulation, i.e., generally speaking, the larger the number of steps since the last reset, the larger the error. Averaging therefore serves as a reset for dead reckoning, but without accurate ranging techniques the accuracy is sensitive to communication range. To illustrate what happens when positions are averaged, Figure 4.17 shows a trace of a simple scenario with two GPS-enabled nodes and a single dead reckoning node (labeled Node 2). In this case, Node 2 is navigating via dead reckoning starting at coordinates (1000, 0) and at the outset its interpolated position is the same as its estimated dead reckoning position.

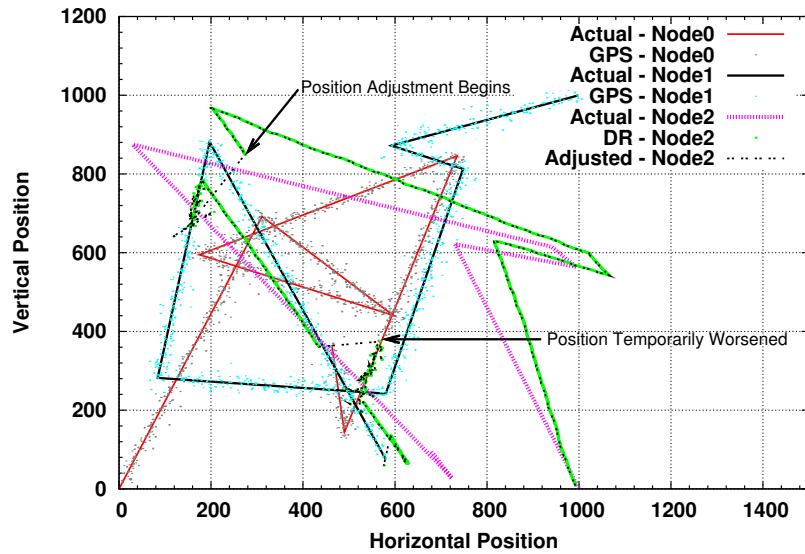


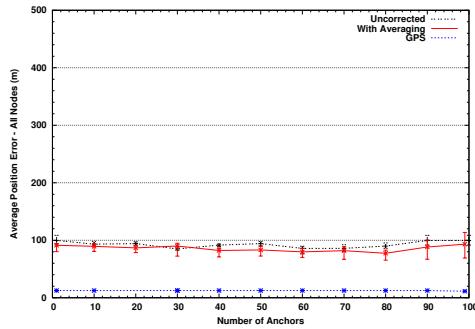
Figure 4.17: Effects of Simple Averaging

Upon establishing communication with an anchor (Node 0), its position is adjusted by simply averaging the two positions together. This causes the adjustment shown by the arrow at the top of the figure. Note that the adjusted position eventually moves much closer to the actual path traveled, shown by the heavy dotted line. This is where the overall improvement is realized. Eventually, Node 2 establishes connectivity with another anchor (Node 1), and in this case the adjusted position is made worse by averaging with the anchor's, as shown by the arrow on the lower right of the graph, but due to both the limited range and the paths traveled, the adjusted position eventually converges closer to the actual path.

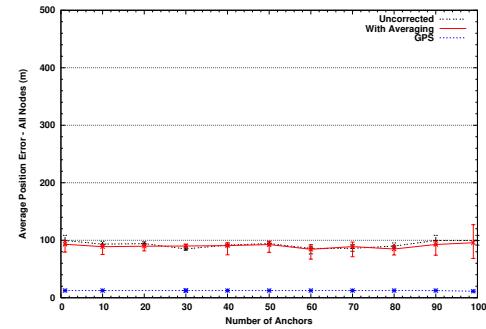
4.5.5 Simple Averaging: 30-Meter and 15-meter Range

The idea behind averaging positions over lower communication range is to reduce the error introduced as a function of the spatial separation between nodes when the geometry of connected anchors is unbalanced. Intuitively, an even shorter range than used in Figure 4.16 should reduce mean position error further, as each position sample represents locations much closer to the same physical location. However, a further reduction in the communication range to within a minimum of 20 and a maximum of 30 meters, produces little benefit, as shown in Figure 4.18. In this case, because the communication range is short, connectivity is far less frequent, and therefore fewer opportunities to correct bad positions exist. The effect is the same as in previous cases with extremely low network density; not enough data is shared to make significant corrections.

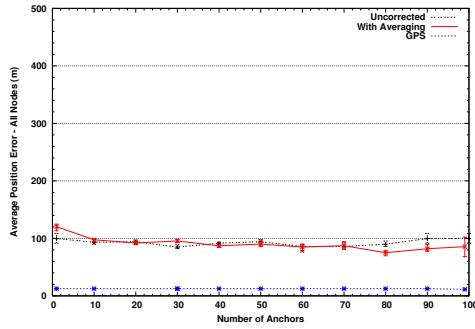
While the mean position error is much better in the cases of low anchor density, i.e., fewer than 20 anchors in total, the improvement at higher anchor densities is not sufficient to justify the added expense of sharing location data. Finally, reduc-



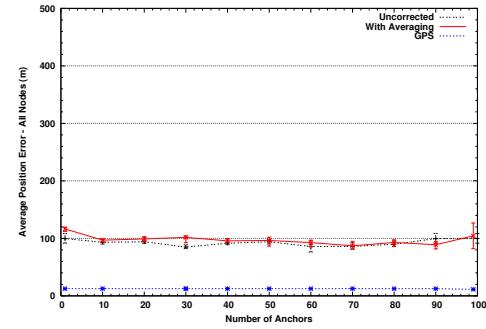
(a) High Network Density



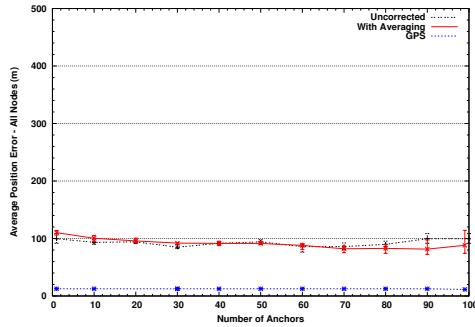
(a) High Network Density



(b) Medium Network Density

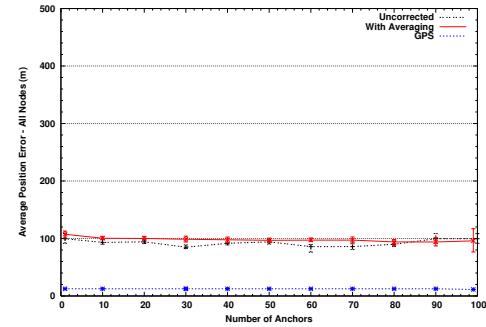


(b) Medium Network Density



(c) Low Network Density

Figure 4.18: Averaging (30m Range)



(c) Low Network Density

Figure 4.19: Averaging (15m Range)

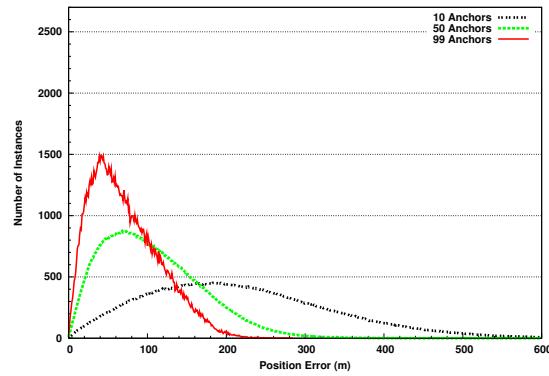
ing the location even further to that of Bluetooth, as shown in Figure 4.19, realizes no significant benefit to improving localization, primarily due to the unlikelihood that in general, two nodes would be sufficiently close to exchange location data.

4.5.6 Distribution of Error

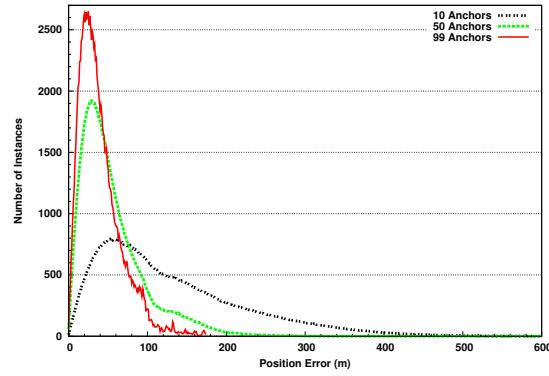
To further evaluate the effectiveness of cooperatively sharing location information and using such to reduce position error, the distribution of measurement error, as determined by the distance from each position estimated via dead reckoning apparatus or through cooperative averaging, to the actual position for each node at each time step, is shown in Figure 4.20. The figure shows the effect of three different communication ranges on the overall distribution of position error: Figure 4.20(a) shows the distribution of errors when nodes are capable of a maximum 300-meter wireless range, while 4.20(b) shows the result with a 100-meter range, and 4.20(c), a 30-meter range. In each case, the error distribution with varying number of anchors is shown, and each histogram is normalized to account for the differing number of nodes whose positions have been adjusted.

In Figure 4.20(a), a significant number of samples exhibit a very large magnitude of error with a low density of anchors. In the case of a 10 anchors among 100 nodes, the distribution is normal, but very flat, indicating a high variance of error with fewer occurrences of each. Increasing the number of GPS-enabled anchors among in the network, reduces both variance and mean error. However, as shown previously in Figure 4.13, in this case the resultant mean error is unacceptably large, particularly in the cases with low anchor density, due primarily to the wide spatial separation, reflected in the error histogram.

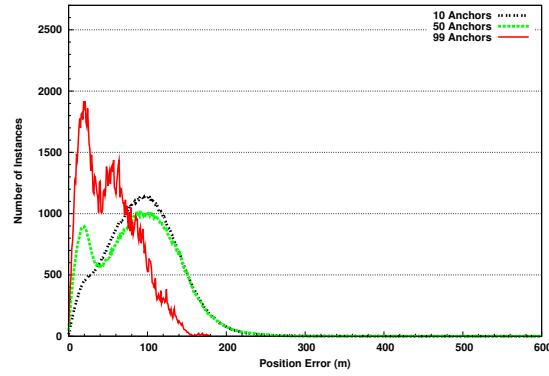
Similarly, with a very short communication range, shown in Figure 4.20(c),



(a) 300-Meter Range



(b) 100-Meter Range



(c) 30-Meter Range

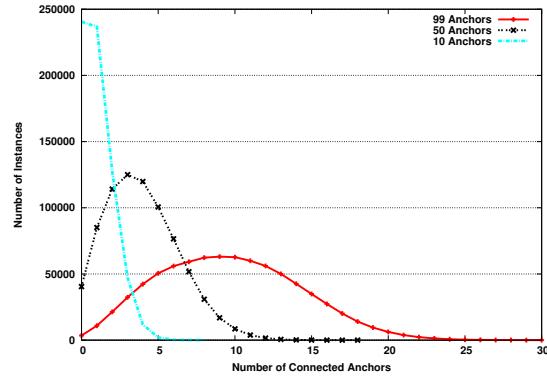
Figure 4.20: Distribution of Position Error (Averaging)

there is relatively little difference in the histograms among all of the anchor ratios due to highly infrequent connections. In the case of 100-meter range (Figure 4.20(b)), the histograms for different anchor ratios are grouped much more closely, suggesting that when using a simple averaging technique, provided a sufficient deployment scale, localization error is less sensitive to changes in the number of anchors than it is to changes in the maximum communication range.

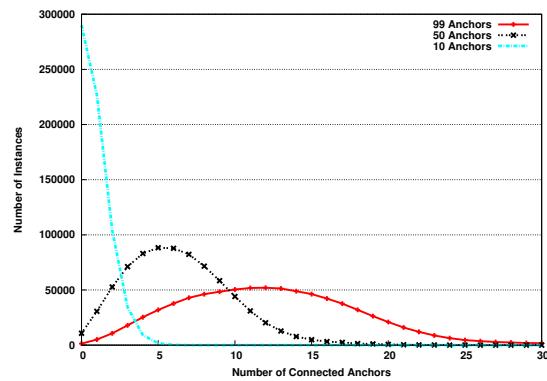
4.5.7 Distribution of Connectivity

As with any distributed localization scheme, accuracy of positions determined via averaging or dead reckoning is greater with a larger number of connected anchors. For these simulations, the total number of connected anchors was recorded at each step among all dead reckoning nodes in cases where the total number of anchors in the system varied from 1 to 99, roughly in increments of 10. Early field tests with averaging locations using TeamTrak produced results not likely to be acceptable in real-world commercial or military applications, due primarily to the limited scale of the tests, which involved a very low number and density of anchors. To get a better sense of how an averaging technique might work at a larger scale, the number of connected anchors at each time step for each node was recorded during simulation trials using 100-meter communication range. In all of these simulations, the actual anchor positions are offset by modeled GPS error, and each graph has a logarithmic scale in the y axis.

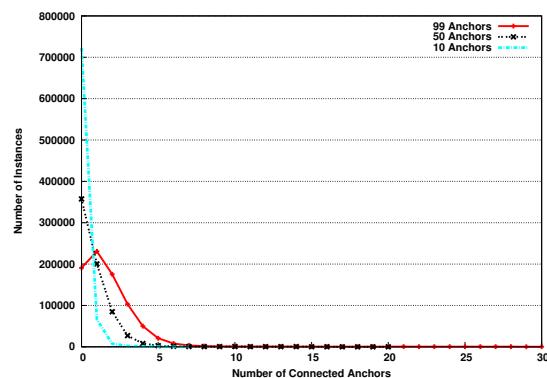
Figure 4.21 is a histogram showing the distribution of anchors in an intermediate density network where the total number of GPS-enabled anchors varies from one, 10, 20, and so on up to 99. For clarity, only select cases are shown in the figures. The figures show that as the total number of anchors in the system or the



(a) High Network Density



(b) Medium Network Density



(c) Low Network Density

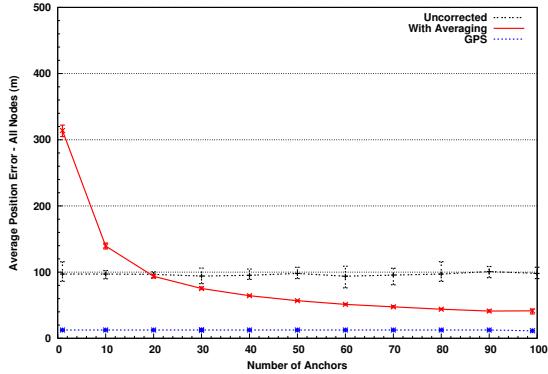
Figure 4.21: Distribution of Connections to Anchors

density of the network increases, the number of time steps in which connectivity to at least one anchor is established similarly increase, suggesting that averaging would be more effective in networks of higher anchor density, whether that density is determined by the ratio of anchors to non-anchors overall or by the relative proximity of all deployed nodes.

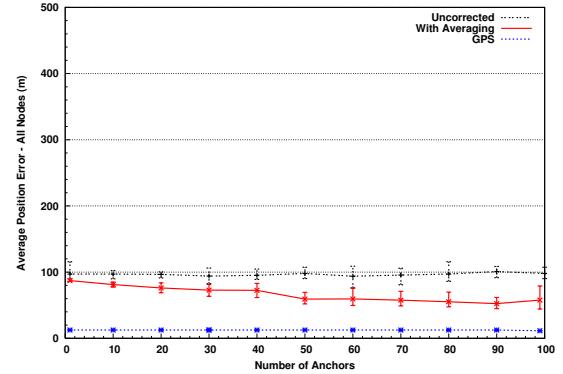
4.5.8 Inclusive Averaging

Having looked at cases in which positions are accepted and averaged without consideration of any additional information, it is also worth examining some alternative schemes for averaging positions, to account for error estimates easily obtainable from connected neighbors. Since the intent of averaging positions is to improve one with a high degree of error, it would make sense to eliminate the position to be corrected and only consider those of its connected neighbors. The risk in this case is where a particular position error would normally be acceptable but is adjusted anyway (due to the assumption that estimating dead reckoning error is inherently difficult and any estimate has a degree of uncertainty. Therefore it would be better to attempt correction whenever possible), it is possible to inject error on the order of the maximum communication range when a connection is initially established.

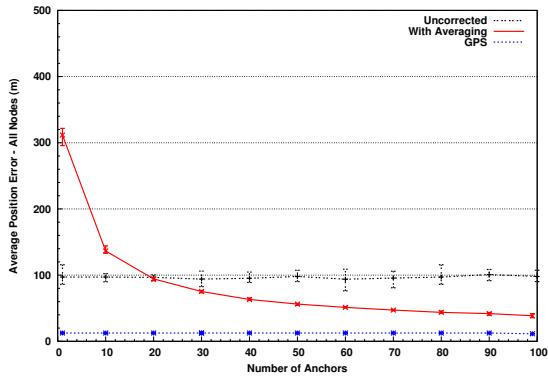
Figure 4.22 illustrates the effect of averaging remote positions with the position to be corrected, which may have a high degree of error. In this case the maximum range is 100 meters. In the steady-state cases illustrated in the figure, the error in the stale dead reckoning positions is significant enough to skew mean error upwards by roughly 100 meters in the worst cases, while showing no appreciable difference in cases with larger numbers of anchors in the system.



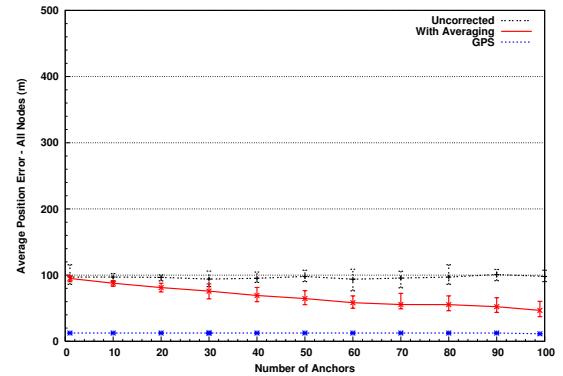
(a) High Network Density



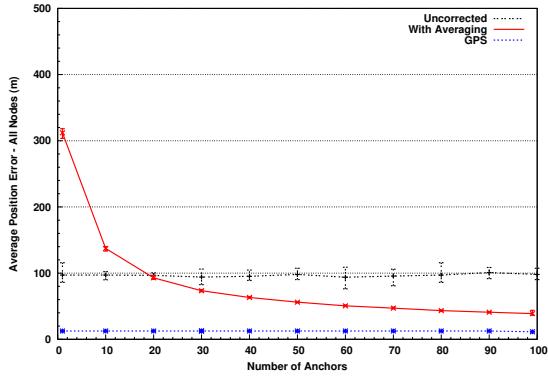
(a) High Network Density



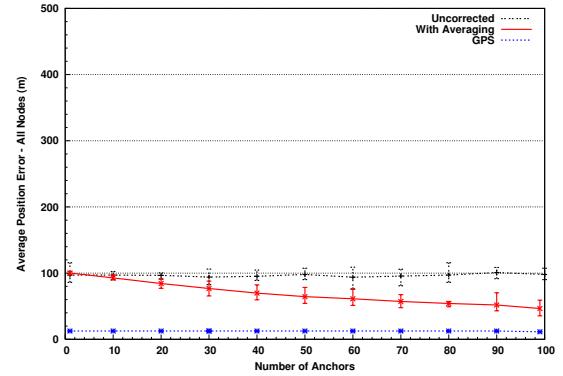
(b) Medium Network Density



(b) Medium Network Density



(c) Low Network Density



(c) Low Network Density

Figure 4.22: Inclusive Averaging (100-Meter Range)

Figure 4.23: Selective Averaging (100-Meter Range)

4.5.9 Selective Averaging

In the cases of simple averaging, in which a position which is known to be bad is thrown out, in this case, dubbed selective averaging, a position that is determined to have a higher measurement error than the position to be adjusted is not considered. Such a case is the logical next step from the simple case, and requires that a node to be localized consider the estimated error of all positions, including its own, which could easily be reported along with the location itself in any data sharing protocol, and only include those whose estimated error is no greater than its own. If a dead reckoning node has a more accurate position than its connected neighbors based on its own estimate of its localization error, no adjustment occurs.

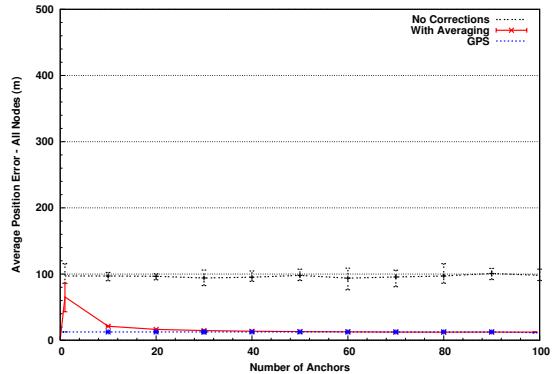
Figure 4.23 shows the effect of such selective averaging in a scenario with 100-meter maximum range. Note that while the mean error in the cases of relatively high anchor density is similar to that of the case of simple averaging, the worst cases is roughly the same as for uncorrected dead reckoning, so this method has the advantage of not making positions worse on the average. By accounting for estimated error, the dramatic increases in mean error which occur whenever one dead reckoning node averages its position with either another or an anchor whose GPS position is drifting substantially can be reduced. A selective averaging technique would most likely be used to correct positions in urban environments where anchors exist but with occasional areas in which GPS fixes are poor or cannot be obtained, so an approach such as this could be effective in the absence of ranging techniques, particularly since this method offers significant improvement without increasing mean overall position error. While some positions may be worsened, the overall net effect is a reduction in dead reckoning error.

4.5.10 Filtered Trilateration

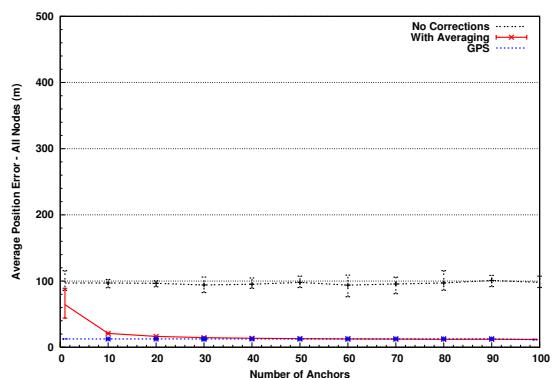
Having examined the effects of variations of range-free averaging techniques, focus now shifts to the use of trilateration, which requires an estimate of both the distance and bearing between a node and multiple anchors. Naturally, since no distance measurement technique for outdoor mobile ad-hoc networks is sufficiently mature and reliable at the present, this work is based largely on an assumption that such a method exists.

Simulation setup is the same as the previous cases, except instead of averaging the positions of neighbors, Algorithm 2 as specified in Section 4.3.5 is implemented. Here, because of the error present in GPS positions as well as in distance measurement, averaging is used to smooth out the random error present in both. Nodes localized with GPS have a mean measurement error of approximately 12 meters based on empirical measurement, and dead reckoning error is essentially unbounded. Distance measurement error follows a Gaussian distribution with σ equal to half of the maximum communication range. As shown in Figure 4.24, by using the filtered trilateration algorithm similar accuracy can be obtained, but these results depend on the existence of a reliable distance determination. To reduce the number of assumptions required in the model, bearing is determined using the simulated GPS measurements at the anchors and the estimated position using dead reckoning at the other nodes.

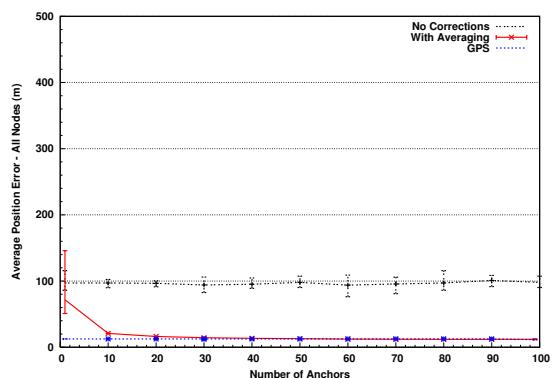
Given the uncertainty of making deterministic correction decisions based on the values of random variables, it is important to look at the effect of implementing an algorithm such as this with a filtering mechanism to rule out the use of positions with large measurement error. Since error metrics are not completely reliable, it is possible to make positions worse based on erroneous data. As long



(a) High Network Density



(b) Medium Network Density



(c) Low Network Density

Figure 4.24: Trilateration (300 Meter Range)

as the number of positions improved is greater than the number worsened, and the mean improvement is greater than the mean worsening, the algorithm can be effective. Tables 4.2 through 4.17 show the percentage of positions in total which are improved and worsened in a number of different cases, with the mean and standard deviations for each.

To understand the effects of this variability, an evaluation using only three mobile nodes is considered, as the additional complexity provided by the interaction of more than two nodes provides more interesting scenarios, but is small enough to enumerate more possible cases. As before, discrete event simulation is used for evaluation with several variable parameters: the number of anchors, i.e., GPS enabled nodes, relative to others, the communication range of each node, ranging from distances commensurate with Bluetooth Class 3 and increasing through that of WiMax, the mobility model, to include both Random Waypoints and parallel motion, as well as the presence of distance estimation error. In all of these trials, a total of three nodes performed as agents. Wireless range parameters were empirically measured, where possible, or taken from published specifications or other empirical data in the literature. Distance estimation error is modeled with a Gaussian distribution with standard deviation one-fourth the wireless range. This is intended to keep error a function of distance, with the extreme cases within approximately one half of the estimated range. The application error tolerance is 15 meters and the minimum confidence threshold is 70 percent. The 15-meter error tolerance was selected as a commonly published 2σ measurement for many GPS receivers, and the minimum confidence level determined for each estimated position error described in Chapter 3.

The following tables show the effects of varying the number of anchors available

in the system in total, relative to the other nodes, as well as the communication range of the nodes. Communication range, as previously demonstrated, has a significant impact on the effectiveness of cooperative localization. Consider the case of long range communication in the presence of distance measurement error, shown in Tables 4.4 and 4.5. In particular, the case of the general mobility model shows that the percentage of points in the system that are actually made worse, as well as the average magnitude of the increased error, are greater than those in the improved cases, which suggests that data sharing in this case has a net negative effect on the quality of positions in the system, and would therefore be ineffective at improving localization. However, looking at cases of shorter range, such as the WiFi range shown in Table 4.8, both the number of points whose error increased and the average change in those points is less than those of the improved cases. This suggests that, at least for relatively short ranges, cooperative data sharing can have a net positive effect on the locations of the nodes.

TABLE 4.2
RANDOM WAYPOINTS, WIMAX RANGE, NO DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	7.026 ($\sigma = 6.71$)	8.35	10.37 ($\sigma = 13.5$)	19.9
2	8.40 ($\sigma = 7.94$)	6.67	18.3 ($\sigma = 14.8$)	17.7
1	6.571 ($\sigma = 6.65$)	5.82	17.468 ($\sigma = 13.10$)	9.77
0	3.885 ($\sigma = 0.00$)	0.01	3.140 ($\sigma = 0.00$)	0.00

TABLE 4.3

PARALLEL PATHS, WIMAX RANGE, NO DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	5.773 ($\sigma = 5.36$)	18.4	4.060 ($\sigma = 4.67$)	18.4
2	6.669 ($\sigma = 5.94$)	22.8	3.309 ($\sigma = 3.80$)	10.4
1	6.396 ($\sigma = 5.94$)	17.96	0.894 ($\sigma = 4.73$)	19.9
0	1.680 ($\sigma = 3.24$)	0.01	0.503 ($\sigma = 0.26$)	38.29

TABLE 4.4

RANDOM WAYPOINTS, WIMAX RANGE, WITH DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	8.181 ($\sigma = 7.85$)	2.77	165.899 ($\sigma = 179.67$)	35.85
2	106.806 ($\sigma = 126.54$)	9.34	155.057 ($\sigma = 170.05$)	26.39
1	157.025 ($\sigma = 184.95$)	10.82	162.954 ($\sigma = 192.05$)	13.46
0	4.895 ($\sigma = 0.00$)	0.01	2.220 ($\sigma = 0.00$)	0.00

TABLE 4.5

PARALLEL PATHS, WIMAX RANGE, WITH DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	5.745 ($\sigma = 5.57$)	19.44	4.854 ($\sigma = 4.42$)	19.52
2	6.348 ($\sigma = 5.82$)	21.16	3.852 ($\sigma = 3.60$)	10.81
1	6.294 ($\sigma = 5.56$)	8.75	4.119 ($\sigma = 3.97$)	5.40
0	10.150 ($\sigma = 0.00$)	0.00	10.260 ($\sigma = 0.00$)	0.00

TABLE 4.6

RANDOM WAYPOINTS, WIFI RANGE, NO DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	5.419 ($\sigma = 5.16$)	6.28	3.580 ($\sigma = 4.89$)	6.21
2	6.276 ($\sigma = 5.55$)	6.75	3.360 ($\sigma = 4.13$)	3.68
1	6.419 ($\sigma = 6.03$)	4.66	1.399 ($\sigma = 3.80$)	6.58
0	1.037 ($\sigma = 2.14$)	0.16	0.950 ($\sigma = 1.47$)	14.3

TABLE 4.7

PARALLEL PATHS, WIFI RANGE, NO DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	5.355 ($\sigma = 5.01$)	0.20	2.635 ($\sigma = 5.49$)	0.20
2	6.675 ($\sigma = 6.05$)	22.8	3.306 ($\sigma = 3.76$)	10.4
1	5.132 ($\sigma = 5.94$)	3.43	1.718 ($\sigma = 4.74$)	7.29
0	1.621 ($\sigma = 3.09$)	0.07	0.503 ($\sigma = 0.23$)	38.0

TABLE 4.8

RANDOM WAYPOINTS, WIFI RANGE, WITH DISTANCE ERROR

#	Improved (mean) (m)	Percent Improved	Worsened (mean) (m)	Percent Worsened
Anchors				
3	6.369 ($\sigma = 5.98$)	5.10	6.596 ($\sigma = 9.87$)	7.94
2	7.571 ($\sigma = 7.07$)	5.33	7.819 ($\sigma = 9.62$)	5.75
1	7.472 ($\sigma = 7.89$)	4.55	3.300 ($\sigma = 10.54$)	5.86
0	6.409 ($\sigma = 5.82$)	4.75	1.355 ($\sigma = 4.47$)	7.48

TABLE 4.9
PARALLEL PATHS, WIFI RANGE, WITH DISTANCE ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	5.637 ($\sigma = 5.41$)	19.00	3.997 ($\sigma = 4.65$)	19.31
2	7.495 ($\sigma = 7.01$)	5.43	7.708 ($\sigma = 9.55$)	5.76
1	6.173 ($\sigma = 5.89$)	18.09	1.091 ($\sigma = 4.83$)	18.25
0	1.619 ($\sigma = 3.07$)	0.08	0.504 ($\sigma = 0.28$)	38.66

TABLE 4.10
RANDOM WAYPOINTS, BLUETOOTH CLASS 2 RANGE, NO
DISTANCE ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	5.419 ($\sigma = 5.16$)	6.28	3.580 ($\sigma = 4.89$)	6.21
2	6.127 ($\sigma = 5.32$)	0.19	1.927 ($\sigma = 4.59$)	0.08
1	6.263 ($\sigma = 5.32$)	0.16	2.258 ($\sigma = 3.97$)	0.66
0	1.504 ($\sigma = 2.69$)	0.00	0.347 ($\sigma = 1.09$)	0.05

TABLE 4.11
 PARALLEL PATHS, BLUETOOTH CLASS 2 RANGE, NO DISTANCE
 ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	0.902 ($\sigma = 2.23$)	0.11	0.886 ($\sigma = 1.52$)	0.12
2	6.434 ($\sigma = 5.47$)	19.2	1.878 ($\sigma = 3.71$)	6.83
1	6.614 ($\sigma = 6.02$)	7.89	2.782 ($\sigma = 4.39$)	3.98
0	1.503 ($\sigma = 2.59$)	0.07	0.413 ($\sigma = 0.38$)	31.9

TABLE 4.12
 RANDOM WAYPOINTS, BLUETOOTH CLASS 2 RANGE, WITH
 DISTANCE ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	4.339 ($\sigma = 3.34$)	0.33	4.496 ($\sigma = 1.55$)	0.27
2	4.994 ($\sigma = 2.94$)	0.30	4.094 ($\sigma = 1.82$)	0.27
1	6.311 ($\sigma = 5.41$)	0.30	5.357 ($\sigma = 2.38$)	0.05
0	3.790 ($\sigma = 0.00$)	0.03	3.190 ($\sigma = 0.00$)	0.02

TABLE 4.13
 PARALLEL PATHS, BLUETOOTH CLASS 2 RANGE, WITH DISTANCE
 ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	5.225 ($\sigma = 5.35$)	7.83	4.175 ($\sigma = 4.01$)	7.92
2	4.918 ($\sigma = 5.03$)	7.13	4.300 ($\sigma = 4.23$)	5.60
1	4.104 ($\sigma = 3.58$)	5.23	3.598 ($\sigma = 3.39$)	3.32
0	9.250 ($\sigma = 0.00$)	0.02	5.910 ($\sigma = 0.00$)	0.02

TABLE 4.14
 RANDOM WAYPOINTS, BLUETOOTH CLASS 3 RANGE, NO
 DISTANCE ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	1.255 ($\sigma = 0.25$)	0.03	0.000 ($\sigma = 0.00$)	0.00
2	1.255 ($\sigma = 0.25$)	0.03	0.000 ($\sigma = 0.00$)	0.00
1	1.130 ($\sigma = 0.00$)	0.02	0.000 ($\sigma = 0.00$)	0.00
0	0.000 ($\sigma = 0.00$)	0.00	0.000 ($\sigma = 0.00$)	0.00

TABLE 4.15
PARALLEL PATHS, BLUETOOTH CLASS 3 RANGE, NO DISTANCE
ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	5.152 ($\sigma = 5.07$)	14.38	4.642 ($\sigma = 4.40$)	14.13
2	5.444 ($\sigma = 5.07$)	12.97	4.251 ($\sigma = 4.23$)	9.98
1	4.537 ($\sigma = 4.07$)	9.18	2.848 ($\sigma = 2.72$)	5.47
0	0.000 ($\sigma = 0.00$)	0.00	0.000 ($\sigma = 0.00$)	0.00

TABLE 4.16
RANDOM WAYPOINTS, BLUETOOTH CLASS 3 RANGE, WITH
DISTANCE ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	1.255 ($\sigma = 0.25$)	0.03	0.000 ($\sigma = 0.00$)	0.00
2	1.255 ($\sigma = 0.25$)	0.03	0.000 ($\sigma = 0.00$)	0.00
1	1.130 ($\sigma = 0.00$)	0.02	0.000 ($\sigma = 0.00$)	0.00
0	0.000 ($\sigma = 0.00$)	0.00	0.000 ($\sigma = 0.00$)	0.00

TABLE 4.17
 PARALLEL PATHS, BLUETOOTH CLASS 3 RANGE, WITH DISTANCE
 ERROR

#	Improved	Percent	Worsened	Percent
Anchors	(mean) (m)	Improved	(mean) (m)	Worsened
3	4.957 ($\sigma = 4.82$)	7.72	4.432 ($\sigma = 4.31$)	7.67
2	5.249 ($\sigma = 5.22$)	7.93	4.432 ($\sigma = 4.33$)	5.90
1	3.722 ($\sigma = 3.22$)	5.85	2.995 ($\sigma = 2.87$)	3.52
0	0.000 ($\sigma = 0.00$)	0.00	0.000 ($\sigma = 0.00$)	0.00

4.6 Limitations

4.6.1 Stability of Dead Reckoning

Figure 3.14 in Section 3.5 shows the distribution of error in a tilt-compensated, three-axis digital compass. This error distribution was obtained through repeated outdoor trials to collect empirical data. It is important to point out that any human-mounted dead reckoning system can be defeated without a great deal of difficulty. In the case of TeamTrak, with its head-mounted compass, simply looking in a direction other than that in which one is traveling is sufficient to skew the heading enough to produce inaccurate results. Compasses mounted elsewhere can be similarly affected with appropriate body movements. All dead reckoning is simulated with the assumption that while taking a step, the operator looks in the direction of travel and does not look down, to the sides, and so forth. Furthermore,

the operator is assumed to travel with a normal walking gait.

4.6.2 Trilateration Challenges

Trilateration presents a number of significant challenges in practice, particularly when localization of mobile nodes at arbitrary distances is required. Using this method in a static system has advantages in two respects. First, an individual node's sensor measurements can be averaged over time to reduce random error, resulting in a more accurate and precise position. Second, distance estimation using wireless signals is a fundamentally difficult problem, and existing implementations experience very high error rates under all but the most carefully controlled conditions at short range. At first, RSSI might appear to be a promising method for estimating distance, with the relationship

$$RSSI = 10 \log \frac{d}{d_0}$$

where $RSSI$ is the signal strength indicator, measured in dBm, d is the estimated distance between radios, and d_0 is a predetermined reference distance. Most work with RSSI is for indoor use, as in practice it experiences very high rates of error at greater distances or in noisy environments. Approximations of range errors observed in practice form the basis for the error model in this work. One alternative approach is DV-HOP, which can determine the average distance between connected nodes, but accuracy is highly dependent on network topology. If there are “folds” in the topology, the estimated error can grow very large. In our experiences with TeamTrak, connectivity can be unpredictable and the types of topologies which would cause DV-HOP to experience large errors are common (Niculescu and Nath, 2003). Another alternative for estimating distance is

using round trip time of packets sent between nodes. However, initial experiments with such an approach outdoors at short range using non-specialized hardware showed no promising results at all.

Probabilistic approaches can account for much of the noise which plagues other approaches, but are still far from ideal. As an example, Madigan et al proposed a probabilistic distance estimation method based on Bayesian graphical models for indoor applications. Even under the relatively friendly conditions found inside structures (as compared to outdoor environments), the authors could attain average accuracy of around seven or eight meters at best (Madigan et al., 2006).

One promising potential alternative to measuring arbitrary distances using range-based techniques is simply using maximum communication range for a given protocol to correct locations immediately upon connection. For instance, connectivity using a Bluetooth Class 2 radio, can narrow down a node's location to within 10 meters of another. If one is well-localized, i.e., has low positioning error with a high confidence level, a node with a higher positioning error can adjust its position accordingly to an accuracy within the 10-meter maximum range. The effectiveness of such an approach would be quite similar to the results shown in Tables 4.10 through 4.17 due to the short ranges involved.

4.7 Conclusions

In this chapter a number of techniques for exploiting location data shared among connected mobile nodes have been presented along with an evaluation of the effects and benefits of such sharing on position error under a variety of constraints and system parameters. Different specific application scenarios and available technologies call for different data sharing, filtering, and combination

techniques. Cooperatively sharing location information can reduce overall localization error even without the ability to accurately measure spatial separation between nodes.

Assuming that multiple nodes are collocated and may use a single location sample for all, collecting and averaging shared positions are appropriate techniques to use when nodes are in reasonably close proximity. This has the advantage of providing location data to completely unlocalized nodes nearby or smoothing out random errors among GPS-enabled nodes, thereby providing a more precise and more accurate position for both. However, it is clear that such an approach breaks down rapidly with increased separation between nodes as well as with significant variations in GPS location quality. For this reason, shorter-range wireless protocols such as Bluetooth may be the most appropriate connectivity for data sharing in such cases. Since many consumer-grade GPS receivers have a 2σ accuracy rating of 15 meters, which is acceptable for most commercial applications, the 10-meter maximum connectivity range of Bluetooth Class 2 radios should not pose any significant concerns in practice.

With arbitrary distances of inter-node separation, averaging positions of connected neighbors can reduce error in dead reckoning localization, provided that such localization is relatively stale. Because using a simple combination method to average position introduces error as a function of such separation, cooperative localization is most effective if the communication range is relatively short, but not excessively so. An overly long range introduces error from wide separation, while an excessively short range does not facilitate connectivity without requiring human intervention. The results show that with the error experienced in the commodity off-the-shelf sensor hardware such as that found in TeamTrak, a range

of approximately 100 meters can reduce mean position error by roughly half in anchor-rich environments, while at the same time minimizing the compounding error resulting from a network with a low anchor density. Furthermore, filtering based on error metrics can reduce the rapid increase in error in low density deployments.

Even better results can be achieved by simply filtering positions using estimated measurement error associated with each position. Excluding positions with large measurement error reduces the likelihood of worsening positions, while simultaneously using anchors with accurate localization to reset dead reckoning. The implication might seem to be that adjusting positions determined via similar means, e.g., dead reckoning versus dead reckoning or GPS versus GPS, would be ineffective, but that is not generally the case. While adjusting GPS positions versus GPS positions with a simple averaging technique frequently worsens localization overall due to the error attributable to spatial separation dominating the 2σ measurement error of GPS, the growth of error in dead reckoning techniques frequently can be sufficiently large and varied across nodes that such an averaging technique may be beneficial in many instances.

Traditional trilateration can be improved to account for measurement error in localization by simply filtering positions using available metrics associated with each position. Excluding positions with large measurement error or low confidence levels reduces the likelihood of worsening positions, while simultaneously leveraging anchors with accurate localization to reset dead reckoning. With filtering, trilateration could improve dead reckoning localization to approximately that of GPS. Although mature technology does not currently exist for these types of applications, with even a reasonably robust distance estimation scheme, whether

from a range-based scheme or simply from using the maximum communication range of the wireless protocol (assuming it can be accurately determined at longer ranges), sharing location data among connected nodes can improve localization accuracy, even in cases where the corrections are intermittent and infrequent. This assumes a filter is in place to exclude remote positions which are clearly unsuitable or whose reliability has a high degree of uncertainty. As better ranging techniques are developed, a more general purpose trilateration scheme that accounts for sensor error, such as that proposed in this chapter, could be feasible.

CHAPTER 5

ON-DEMAND DATA BACKUP AND RECOVERY IN MOBILE SYSTEMS

Mobile computing devices have become increasingly ubiquitous over the last several years. Such devices have been employed in a wide array of applications, exemplified by such disparate information systems as cell phones, UHF radios, and PDAs. While using such devices for recreational purposes is commonplace, critical applications such as coordination of search and rescue and emergency response operations are also becoming widespread. Using such a system for military or paramilitary command and control purposes places a high premium on data availability, security, and internode communication. An abundance of real-world cases demonstrate that without efficient data sharing in a crisis environment, successful mission accomplishment is very difficult, if not impossible.

While mobile information systems enable coordinated operations through real-time communications, the underlying mobile networks which facilitate data sharing are frequently problematic; ad-hoc topologies are inherently unstable, unpredictable, and pose a significant challenge to providing reliable access to distributed data. Network partitions and churn are a certainty in any mobile network of more than trivial size or deployed over more than a very limited geographic area, which implies that despite the continual evolution of robust routing protocols, reachability from one node to any other at a given time is far from guaranteed. While also affected by changing network connectivity, availability of data and services

provided by a particular device is also impacted by the limited computing capacity inherent to mobile computing. In very remote locations without the benefit of fixed, preinstalled infrastructure, mobile computing devices relying solely on battery power will eventually become unavailable and unusable before completion of any likely real-world mission due to limited system resources. Because of the significant likelihood of device and network failures in such an operational deployment scenario and the potential mission criticality of the data stored on such networks, it is necessary to have a reliable means of off-device data backup and retrieval, even if only for short durations.

It has been well established in practice that when it comes to data, more is not necessarily better. In typical mobile networks, devices cannot simply and naively broadcast all data recorded by all participants to all other devices within range and flood the network with large amounts of traffic. An indiscriminate data transfer model such as this has several serious drawbacks. First, such an approach can consume tremendous amounts of bandwidth and storage space, which are limited in the first place. Second, it places an undue burden on all computing devices in the network to process volumes of extraneous data, imposing a severe performance penalty on computing devices and possibly even hindering the effectiveness of human operators, resulting in impaired mission. Third, heavy resource utilization, particularly wireless radio transmission, has substantial system availability costs due to shortened battery life on each mobile device.

Conversely, some preservation of transient data is required. For instance, sensor data which may be considered extraneous at one time may in fact be required later as the situation on the ground changes, necessitating its preservation. Also, the criticality of much of the data in a crisis response environment is a function

of location. For instance, the threat posed by specific hazards may be highly localized and of concern to operators only in the immediate vicinity, and may not be significant enough to warrant a broadcast notification to all users of the system, particularly in light of limited computing resources and available bandwidth, but may still require preservation for reporting purposes and after-action analysis. To ensure such data preservation without the benefit of more robust computing capability, what is needed is a cooperative data sharing protocol for networks of localized mobile nodes. The use of context for data transmission decisions limits the total amount of wireless broadcasting required by each node while at the same time maximizing the pairwise availability between any two adjacent nodes. The term cooperative is used here under the assumption that all users of a system are considered friendly to each other and willingly share available computing resources to ensure successful completion of a common objective. An exploration of the social aspects related to cooperative computing in general are beyond the scope of this work.

Maximizing availability of connected peer nodes is important because of the nature of ad-hoc networks. Links are frequently broken, and outdoor environments can be quite noisy, particularly those in and around populated, urban areas, with both factors contributing to overall unreliable connectivity and low available throughput. Additionally, maximizing availability is important because command and control systems typically employ encrypted communications, which impose additional, oftentimes significant, overhead in any data exchange. While there is an abundance of past and ongoing research in the field of mobile networks, work specifically in the area of availability prediction in such systems is much more limited. While a few works attempt to predict future availability by tracking position

history, routing protocols that incorporate system state of nodes at the next hop into routing determinations, or those which combine system state with mobility are rare or nonexistent.

The objective of the work in this chapter is a new method for context-aware cooperative data transfer for the purpose of short-term data backup and recovery. Context is described by available system capacity, geographical positioning through accurate localization, and the tracking of relative motion patterns among connected peer nodes in mobile ad-hoc networks. First, a method is presented for context-aware peer selection using location-based single-hop routing, to include the initial evaluation process and the final selection through availability prediction, in which availability is a function of expected mobility. Next, several issues related to data recovery are discussed and the effectiveness of the method is evaluated through simulation trials. Results are presented along with some concluding remarks.

5.1 Context Aware Data Replication

In the literature, mobile networks are frequently assumed homogeneous, consisting of a collection of identical hardware platforms with roughly similar capabilities and deployed with each node generally having the same overall capacity. In practice, despite similar configurations, it is common for devices to be deployed with significantly varying states such as battery level or available storage space, and any underlying system configuration would consist of dissimilar hardware, such that each node may be either a laptop computer, PDA, or even a stationary desktop workstation located at a command post, each of which having varying capabilities in terms of processing power, available storage space, and wireless

communication range. Despite differences in capability, what is important is that these devices are connected to an ad-hoc routing network and are able to receive accurate positioning data perhaps, but not necessarily, through a portable GPS receiver, and cooperatively share such information with other nodes.

In addition to the heterogeneity of the overall system, as with any ad-hoc network a deployed system can also be expected to experience a substantial amount of churn as operators enter and leave the network, either by traveling beyond the maximum communication range of individual wireless radios, entering shielded structures, experiencing device failure and network partitions, or as other such events occur. Device failure can occur either through physical destruction of the device in certain circumstances or through gradual loss of battery power. The potential loss of power is significant as continuous operations in very austere locations may preclude swapping batteries when required. In military applications, soldiers laden with 100 pounds or more of gear while simultaneously operating an array of mission-related equipment in hostile areas cannot be reasonably expected to swap batteries in mobile communication devices with any frequency.

The wireless communication range for each mobile device is assumed to be a predetermined system parameter. This is not a new assumption, and was argued previously in (Deng et al., 2004) among others. Despite variations in RF signal propagation which occur in practice, a conservative, but consistently determined, measurement is all that is required, so estimate the approximate wireless range is not unreasonable. For instance, for the Intel PRO/Wireless 2915ABG wireless card factory installed in the Lenovo Thinkpad tablets used in the baseline Team-Trak configuration, a wireless range measurement of approximately 200 meters is shown in Section 4.3.4, Figure 4.9. While there are variations in the communica-

tion range, consistent connectivity is most important, so even given anisotropic radio propagation, longer ranges in different directions can be effectively ignored since they generally do not contribute to consistent connectivity.

The transient data may be arbitrary depending on the specific application and scenario, but generally can be assumed to be logs of events recorded internally by the device or externally by either peripheral sensor devices or the operator. As data is collected, eventually that which is deemed critical may need to be replicated on a neighboring node to ensure persistence. The need for replication may be based on internal factors such as a determination of impending device failure due to critical system state or external factors such as the possibility of device loss or destruction due to entering a hostile area.

5.1.1 Peer Selection

In traditional distributed systems built on fixed infrastructure, it is common practice to maintain a single catalog that describes the available capability of all nodes in the system. An example of such a system is the Chirp personal file server (Thain et al., 2008), which uses a well-known, designated catalog server updated periodically by each individual file server. Clearly, a catalog server in a mobile network would be impossible to implement effectively in such a fashion. A more decentralized approach might be through the use of distributed hash tables, but even that would be impractical in a highly dynamic wireless ad-hoc network. Maintaining static information about the capability and availability of nodes in the system is unworkable in such a dynamic environment, as such information rapidly becomes stale and useless, so one alternative is for each node to simply monitor the state of its peers for as long as they are directly connected and then

select one as needed.

Selection of a peer node depends on several factors. The first step in performing a data transfer is determining the amount of data that must be moved. Once the data size is known, the next step is finding a location with sufficient available space. Because criticality of data depends mostly on the application or situation, the specific data set is best determined with policy, which can simply specify a prioritized list of files that are to be offloaded. To determine the remote storage requirement, the volume of data is measured with an internal resource monitor that periodically records the total size of the files expressed in the policy specification. The resource monitor also checks the current battery state and available local storage space. With this information and knowledge of the system state of neighboring nodes, it is trivial to determine whether its peers have sufficient available storage which may be used as temporary scratch space. Gathering this information requires advertisements of available storage space from each peer node, much like notifying catalog servers in traditional distributed systems, and such advertisements may be included as packet fields in a routing protocol or transmitted separately. In this work the advertisement is part of a distance vector routing protocol implemented within the TeamTrak testbed.

In addition to determining the size of data which must be preserved, a resource monitor task periodically polls the battery level and available local storage space and records it. Such state information is transmitted to its peers as data fields in the packets in the routing protocol, and is used in conjunction with other device state information such as physical location determined via a connected GPS receiver, dead reckoning, or through a cooperative localization technique as described in Chapter 4. All nodes in the system are localized.

As routing packets are received from remote devices, each node constructs a routing table that contains the battery state, available free storage space, and latitude/longitude position of all connected peers. At the time a data replication operation is required or requested, the current view of the network based on the routing table is used to select a peer within range of a single hop. Selecting a closely connected peer is desirable for two reasons. First, it is extremely challenging if not downright impossible to determine with any reasonable degree of certainty the properties of the wireless connection between two arbitrary remote nodes. More importantly, however, in many operational scenarios moving or replicating the data is time-sensitive or time-critical and storing it in a geographically proximate location may be more beneficial to recovery. In other words, placing data at an arbitrary connected node that may only be reachable across several hops may not be advantageous despite the potential availability of greater system capability at that node. This does not take into account the higher probability of failures and retransmissions introduced when moving data over additional, intermediary nodes as described in (Rohner et al., 1998).

Peer selection in this approach consists of a two-part scoring method which can augment existing routing protocols in mobile ad-hoc networks. Whichever routing protocol is employed would need to include, or be modified to include, system state and geospatial location information about each node as described previously. Such information could easily be inserted into many types of routing packets, as long as the mobile devices themselves have some notion of location built in, e.g., are affixed to a GPS receiver.

5.1.1.1 Initial Scoring

In order to find a suitable peer node, it makes sense to quickly eliminate from further consideration those devices which are obviously not appropriate candidates based on their available system capability. For instance, if a peer has insufficient battery power to receive, for example, a 1 MB file transfer, or is unlikely to be able to hold the data for very long, then it should not be considered after the initial examination of its system state, unless the state were to change either through battery replacement if possible or by freeing storage space.

The first step in peer selection, then, is to make a rough initial evaluation of all nodes in the routing table, intended more to eliminate clearly unsuitable nodes than to find an optimal node, and assign a score to each that reflects whether its capability warrants further consideration. Any node with very low battery power or available storage space less than the amount required to complete the transfer is assigned an initial score of zero. Nodes that are more than a single hop away are also scored zero. All remaining nodes are assigned an initial score of one. These nodes are the candidates for possible data transfer.

5.1.1.2 Availability Estimation

When a data transfer is requested or required, the candidate nodes, which are indicated in the routing table with an initial score of one, are then re-scored based on an estimated window of opportunity. The window of opportunity, expressed in units of time, is determined by extrapolating the peer's location history over time to some approximate point at which either connectivity will likely be very poor or the node will move out of wireless range entirely, rendering any further direct data transfer impossible. The window of opportunity is illustrated in Figure 5.1.

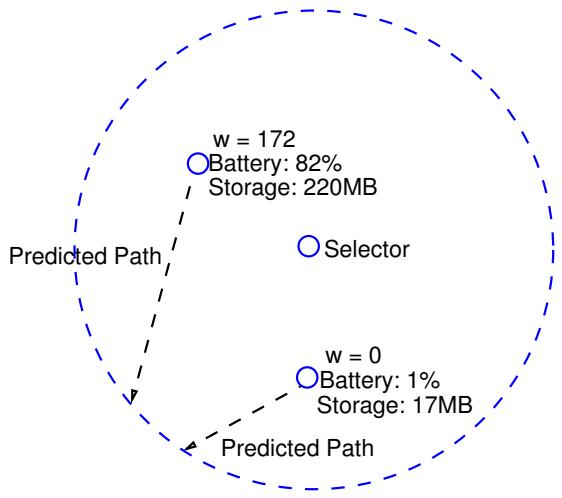


Figure 5.1. Candidate Peer Node Selection

Once the window of opportunity is computed for each possible candidate node, the node requesting the transfer simply performs a greedy selection and chooses the peer with the largest window of opportunity.

This method ensures that nodes near the fringes of the wireless range are less likely to be selected than those nearby if they are moving away from the requesting node, but more likely if they are moving towards it, and that nodes which are stationary relative to the requesting node are the most likely to be selected as their windows of opportunity are considered infinite. Figure 5.1 provides an illustration of the peer selection method. In this scenario, a node with insufficient battery power is eliminated from consideration in the first step. Other candidate nodes are then considered based on the predicted window of opportunity, illustrated with arrows and dotted lines, which indicates the estimated length of time of availability. In the figure, two candidate nodes are in motion, so they have a limited window. Since one of the two candidates has critically low available system resources, the other node is the one selected. If there are multiple nodes with

acceptable resources available, that with the highest score among those would be selected.

A very simple distance equation is used to determine the window of opportunity between the peer's current location and the intersection point of the wireless radio range. Because wireless range is generally not isotropic, any empirically or analytically determined range can be used; as long as the boundaries are roughly known, the intersection points can be computed. For each candidate node c_i ,

$$score_{c_i} = \frac{\sqrt{(x_{int} - x_{i_t})^2 + (y_{int} - y_{i_t})^2}}{t_i - t_{i-1}} \quad (5.1)$$

where (x_{int}, y_{int}) is the intersection point between the candidate node's extrapolated current path and the estimated limit of the wireless radio range. Because only a conservative estimate of availability is required, the communication range is considered to be the maximum range at which nodes have two-way connectivity. For simplicity, in this case an isotropic model of connectivity is sufficient, although it certainly does not need to be. Given the approximate communication range r , the intersection points x_{int}, y_{int} are determined as follows:

$$a = (x_{i_t} - x_{i_{t-1}})^2 + (y_{i_t} - y_{i_{t-1}})^2 \quad (5.2)$$

$$b = 2((x_{i_t} - x_{i_{t-1}})(x_{i_{t-1}} - x_{j_i}) + (y_{i_t} - y_{i_{t-1}})(y_{i_{t-1}} - y_{j_i})) \quad (5.3)$$

$$c = x_{j_i}^2 + y_{j_i}^2 + x_{i_{t-1}}^2 + y_{i_{t-1}}^2 - 2(x_{j_i}x_{i_{t-1}} + y_{j_i}y_{i_{t-1}}) - r^2 \quad (5.4)$$

Because connectivity between node j and each candidate node i is a given, there are always two points of intersection between the path taken by i and the outer range of the wireless radio. The intersection point that lies behind i is ignored. Therefore we have:

$$\mu_1 = -b + \sqrt{b^2 - 4ac} \quad (5.5)$$

$$\mu_2 = -b - \sqrt{b^2 - 4ac} \quad (5.6)$$

with intersection point (x_{int}, y_{int}) , where x_{int} and y_{int} are expressed as:

$$x_{int} = x_{i_{t-1}} + \mu_1(x_{i_t} - x_{i_{t-1}}) \quad (5.7)$$

$$y_{int} = y_{i_{t-1}} + \mu_1(y_{i_t} - y_{i_{t-1}}) \quad (5.8)$$

5.1.2 Data Management and Recovery

The focus of this work is a backup and recovery scheme based on localization for application scenarios in which data replication is necessary or desirable in the short term. Because of the dynamic nature of mobile ad-hoc networks, a distributed storage system for longer-term applications built on mobile networks is not likely to be feasible, although some work has been done to address the fundamental issues associated with data replication in MANETs (Hara, 2005).

Nodes which store data on behalf of another use whatever scratch space is available on its local storage device to do so. Eventually, however, that stored data might need to be either retrieved or purged. The node which originated the data must either fetch the data or inform the peer that any of the replicated files it stores on its behalf no longer need to be maintained. This might result from replacing a battery or the operator's successful return from a potentially hazardous location. In a traditional distributed system build on a wired network with fixed infrastructure, control of replicated copies is not difficult. On the other hand, in a wireless network, particularly a MANET, data recovery is a nontrivial problem.

If a node transfers data just before leaving the network, the only metadata that might be known to the transferring node is the initial location of the transfer, but even that is hardly guaranteed. The node performing the temporary storage may have further transferred its data to a subsequent node. Since most peer-to-peer networks assume either a fairly static locality of data or multiple replicated copies of data, traditional approaches to data recovery are unsuitable for this system.

In most cases, there is only a single replicated copy of data for each node to be managed. For sensor network applications, this data would generally consist of time sequenced sensor data or system logs, largely obviating many of the most significant challenges inherent to effective data management in an unstable environment. Still, effectively managing data is inherently challenging under such conditions. Due to rapidly changing topology and frequent disconnects, providing data for recovery or purging can be at most a best-effort service. For the types of scenarios in which this method might be employed, it is generally more critical simply to ensure data is preserved somewhere than to guarantee a complete restoration on the originating node in a timely manner. The practical implication is the possibility that a redundant copy of data may reside on the network but be unreachable by its originator. Alternatively, while peers which are reachable only through multiple hops are not considered initially for transferring data, it is possible that over time, multiple transfers by several peers or changes in topology due to mobility may position the data more than a single hop away from the originating node.

When remote storage of replicated data is no longer needed, it can either be retrieved by the originator or simply purged. Naturally, a successful restore would result in purging of the replicated copy as well, since generally mobile devices

are resource constrained. In a manner somewhat reminiscent of the Access Log location management method for accessing replicated copies (Hara, 2005), the originator sends a unicast message to the node corresponding to the data items maintained in a table, which requests the particular operation. If the replicated data is available on the node originally selected and there is a connection, even indirect, between it and the originating node, then recovery is simple, assuming the data can be successfully transmitted without disconnects or partitioning.

A node attempting to rejoin the network will check for the presence of the peer to which its data was transferred. If the node is present, then the originator either recovers the data, the replicated copy is deleted, or the node holding the replicated copy forwards the request while informing the originator of the new location. If the replicated copy is unreachable, nothing more can be done until either a route can be established or the mission ends. The tradeoff for ensuring higher availability of data in a mobile network is the possibility of redundant copies, and in the general case, possible inconsistency between such replicated copies.

5.2 Evaluation

This section demonstrates that the context-aware selection method outperforms selection by either random choice or geographic proximity under most conditions with a variety of assumptions. In this work, a selection is considered superior if it remains within wireless range, and thus available, for a longer period of time using various patterns of motion. The effectiveness of this approach was initially evaluated through a series of discrete event simulations using a very simple linear and random movements among remote peers. The Random Walk mobility model was selected because of the obvious difficulty in predicting availability when

nodes make frequent changes of direction. The simulations were conducted in two parts, discussed here separately. The first part is an evaluation of the quality of the initial selection, based on the criteria needed to conduct data transfer. The second part evaluates the quality of the selection over a longer period of time in order to determine whether a context-aware approach yields a selection with a higher average availability, as determined by both system state and connectivity.

Unless otherwise stated, in the initial simulations 50 nodes were placed at random within the node of interest's wireless range of 300 meters, with uniform distribution, and assigned random speeds and directions. Because having nodes remain stationary relative to the node of interest does not yield interesting evaluation results, as the availability of any such node would not be affected by movement, focus of the evaluation tasks is limited to nodes actually in motion. For these simulations, a minimum speed of one meter per second and a maximum speed of four meters per second were selected. This is intended to model a range of speeds of human operators from a slow walk to running quickly.

5.2.1 Initial Selection

TABLE 5.1
INITIAL SELECTION (LINEAR MOTION)

Method	Success		Optimal Selections (%)
	Rate (%)	Failures	
Random	74.27	187.00	10.43
Nearest	84.47	113.00	5.79

TABLE 5.1

Continued

Method	Success	Optimal	
	Rate (%)	Failures	Selections (%)
Context Aware	100.00	0.00	100.00

Table 5.1 compares the ability of the context-aware approach to initially choose an optimal peer node moving linearly with constant speed to that of the random and geographic selection methods. For initial selections, a successful selection is one in which the chosen peer has ample battery life and storage space to complete a 50 MB data transfer and lies within a 300-meter wireless radio range. Because all nodes are initially within range, a selection in this case is considered a failure only when the selected node has insufficient system state. Optimal selections are those with the largest composite evaluation score among all nodes, and for the random and nearest selection methods, an optimal selection is one that was also chosen by the context-aware approach. Context-aware selection will always choose a node with sufficient system state for the size of the data assuming one is available; hence the 100 percent success rate shown in the table. These results are shown for simulations of 1,000 independent runs.

Table 5.2 further compares the context-aware approach to random and geographic selection when peer nodes move randomly in both speed and direction. Other simulation inputs are the same as in the linear model.

TABLE 5.2
INITIAL SELECTION (RANDOM WALK)

Method	Success		Optimal
	Rate (%)	Failures	Selections (%)
Random	73.11	195.00	9.43
Nearest	84.30	114.00	5.64
Context Aware	100.00	0.00	100.00

For both patterns of motion, even using a random approach can produce reasonable results, with a selection success rate of about 74 percent in simulations where the initial system states among nodes are randomly generated with a discrete uniform distribution. As shown in the table, the random selection method generally chooses a higher number of unsuitable nodes compared to selecting the most geographically proximate node, but this is more likely due to the randomly generated system state than fundamental limitations of the selection method. By eliminating the possibility of selecting unsuitable nodes, the context-aware approach does not experience the failures exhibited by the other selection methods. It is always possible in a given scenario for no suitable peers to exist, but no selection algorithm would work in such a case, and as in the case of stationary nodes, is a much less interesting simulation scenario.

Due to the greedy nature of the selection algorithm, it is possible that multiple peers may select a single stationary node for data backup simultaneously. To avoid

a single node becoming a bottleneck for multiple transfers, a policy mechanism can be used to select other nodes in such an event. Similarly, an access control policy could be used to prevent unauthorized placement or retrieval, but such policy and access control schemes have not yet been explored within the context of this work.

5.2.2 Availability

After the initial evaluation and selection, the selected nodes were reevaluated after increasing periods of time to measure the quality of the selection using the length of time connectivity is maintained. While it is certainly true that nodes out of wireless range of a specific device may still be reachable over multiple hops, the likelihood that data transmissions will fail increases significantly with the number of hops (Rohner et al., 1998), so the ideal case would be to select nodes that can later be reached directly if possible, with the goal to find the peer with the greatest availability. This section evaluates the ability of the context-aware approach to do so.

To evaluate the longer-term suitability of the selections, both random and linear motion patterns were simulated among 50 nodes over a period of 700 seconds, and at 5-second increments, reevaluated the availability of the selection. Batteries discharge at a constant rate with a total lifetime of 3 hours. For each selection time, 1,000 independent simulation runs were conducted and the mean success rate for each method was recorded.

Figure 5.2 shows the effectiveness of the context-aware approach as compared to a random or strictly geographic approach when peer nodes move in a constant linear fashion, starting at random locations to simulate a steady state scenario.

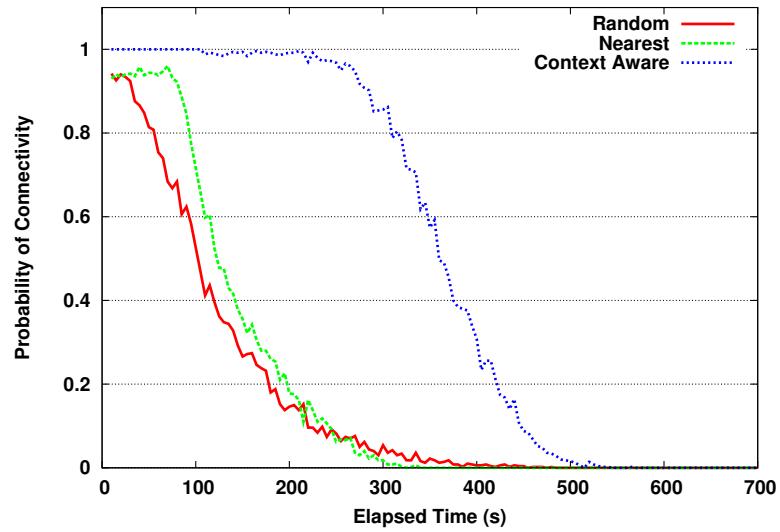


Figure 5.2: Availability With Linear Motion

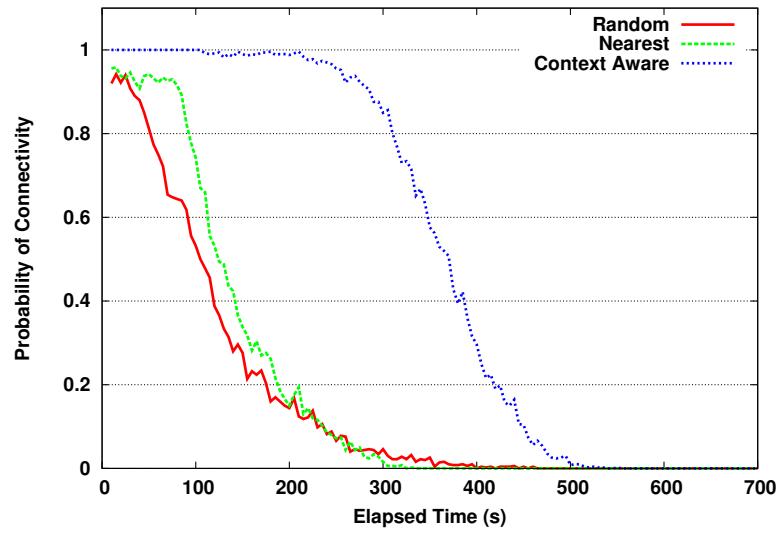


Figure 5.3: Availability With Random Walk Mobility Model

Once the initial selection is made, the simulator attempts to contact the selected node again after a designated time interval. Any selected node with critically low battery power or storage space, or has moved out of wireless range, is considered a bad choice. By selecting nodes with an acceptable system state that also maximize the conservatively estimated availability window, the probability of easily retrieving data is higher than that obtained using the other approaches. Figure 5.3 shows similar effectiveness of the context-aware method using remote peers that move randomly.

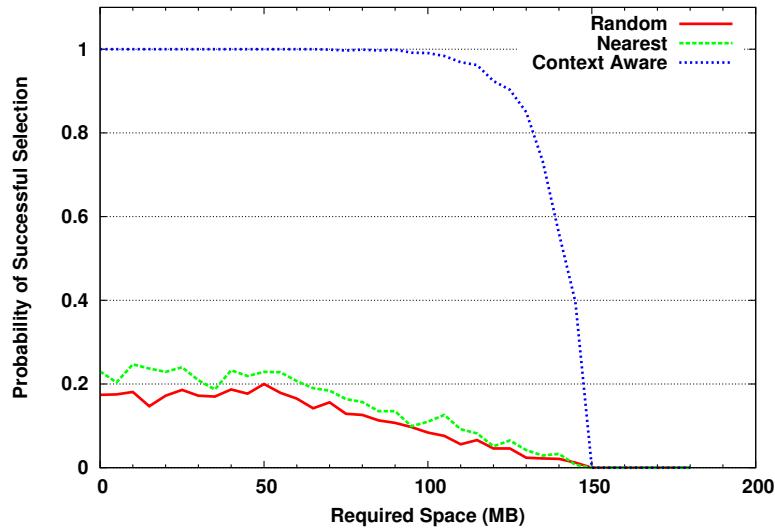


Figure 5.4. Effect of Data Requirement on Success Rate

Figure 5.4 shows the effect of increasing data requirements on the success rate. In these simulations, nodes are assumed to have available storage space ranging

from a minimum of 50 MB to a maximum of 150 MB. As the size of the data requirement increases up to the limit of available storage resources, the success rate for all three methods approach zero, with the context-aware approach having a much higher success rate overall.

Finally, the effect of the number of nodes and the size of the wireless radio range on the average length of time before each approach fails is evaluated. Table 5.3 shows the amount of elapsed time, i.e., the availability period, averaged over 1,000 independent trials for each node/range pair, before each method's selection fails due to insufficient system state or eventual movement out of communication range. In each case, the context-aware method produces a selection with a higher average availability time than the other methods, in some cases by as much as six times that of the random selection.

For very small-scale networks, both in terms of wireless capability and number of nodes, the context-aware method does not provide enough of a benefit to be worthwhile, since availability is extremely limited in any case. The results indicated with a ¹ highlight one example of such. However, as the size and capability grow significantly larger, as shown, for example, with a ², the context-aware approach gives dramatically better results than either nearest-neighbor or random selection. This suggests that as wireless capability increases and ranges grow substantially larger, a context-aware approach for maximizing availability may have substantial benefit for improving reliability of data sharing in mobile networks.

TABLE 5.3
AVERAGE TIME BEFORE FAILURE

Number of Nodes	Selection Method	Maximum Wireless Range (m)						
		10	50	100	500	1,000	5,000	10,000
10	Random	3.2¹	19.7	33.8	147.7	275.9	1,190.3	2,108.2
	Nearest	3.9¹	31.2	50.5	217.7	426.2	1,805.7	2,998.0
	Context	3.9¹	47.6	91.5	439.1	852.2	3,356.1	4,768.6
50	Random	8.6	19.2	32.1	148.8	297.2	1,219.7	1,926.8
	Nearest	10.9	31.0	53.1	221.4	434.8	1,897.5	3,127.4
	Context	11.0	67.7	133.6	633.8	1,227.7	4,377.9	5,523.7
100	Random	9.8	19.8	35.3	146.9	278.4	1,275.0	2,043.0
	Nearest	13.3	30.3	52.8	225.1	429.0	1,872.3	3,143.4
	Context	13.5	75.4	149.6	699.0	1,338.6	4,715.0	5,131.8
500	Random	10.1	19.5	34.5	152.7	303.9	1,198.5	2,020.5
	Nearest	15.8	31.2	21.7	219.2	440.6	1,920.9	3,304.8
	Context	16.4	88.3	172.4	819.8	1,545.8	5,034.0	5,041.0
1,000	Random	10.1	19.9	33.6	153.1	285.5	1,234.4	2,082.1
	Nearest	16.6	31.2	53.2	225.3	435.8	1,846.8	3,195.9
	Context	17.3	92.4	179.7	846.6	1,618.0	5,049.9	4,993.4
5,000	Random	10.0	19.6	34.5	152.8	276.1	1,231.0	2,061.1
	Nearest	17.5	31.3	54.4	227.4	433.3	1,973.3	3,179.1
	Context	18.7	99.1	192.7	903.0	1,706.2	5,038.4	5,188.5

TABLE 5.3

Continued

Number of Nodes	Selection Method	Maximum Wireless Range (m)						
		10	50	100	500	1,000	5,000	10,000
10,000	Random	10.1	19.8	33.1	156.1	291.2	1,233.2	1,903.0²
	Nearest	17.8	31.0	53.1	228.1	432.0	1,978.8	3,319.0²
	Context	19.2	101.1	195.2	919.4	1,735.4	2,853.4	5,356.5²

5.2.3 Random Waypoints

In the previous sections, the selection algorithm was evaluated using simple linear or random mobility models. Additionally, the node making the selection was assumed stationary. The limitation of such models and assumption is that while effective in highlighting the differences in efficacy of the selection algorithms evaluated, neither are likely to occur in practice, and may ultimately be overly simplistic. To further evaluate the selection algorithm under more challenging conditions, the mobility model is changed to Random Waypoints for all nodes in the system, to include the selector. This change in assumption effects a change in the way peers are tracked and scored; it is the relative motion rather than absolute, that affects availability. In addition, sudden changes in direction may occur among any node in the system at any time. For purposes of evaluation, the Random Waypoints model was again selected due to its generality and that it

provides a more difficult scenario than would likely to be encountered in practice.

In this scenario, the random, nearest neighbor, and context-aware selection algorithms were executed among a network of 100 nodes with varying communication ranges and varying network densities. Initial placement of the nodes is random with uniform distribution about the simulation field. Communication ranges vary from 10, 50, 100, 300, 500, 1,000, and 10,000 meters with gradually decreasing densities commensurate with each successive increase in maximum communication range. All nodes in the system are in motion with an average speed of 1.56 meters per second, which is the average walking speed of an adult male.

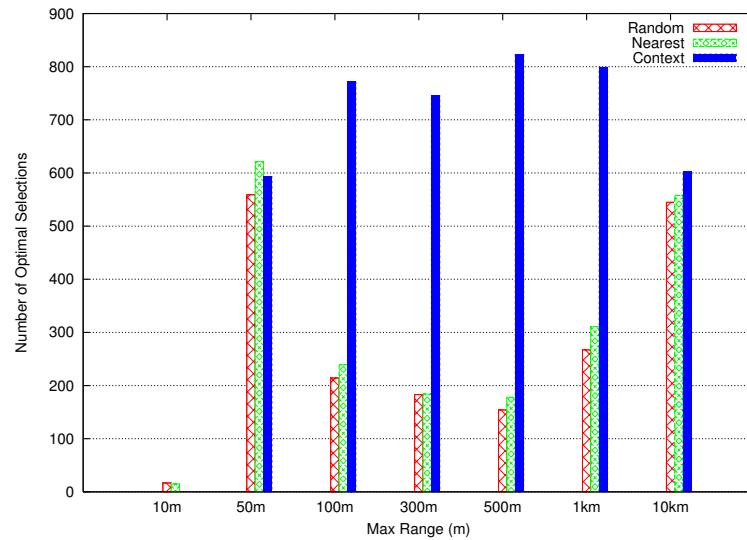


Figure 5.5. Selection: Arbitrary Mobility

Figure 5.5 illustrates the number of selections made by each algorithm over

1,000 independent trials where the selected peer had the longest actual availability starting from time of selection. Availability can be terminated by either moving out of communication range or loss of battery power. In the simulation trials whose results are shown in the figure, GPS is perfect and RF propagation is isotropic. At very short ranges, the likelihood of establishing connectivity is remote (at least with a random model of mobility this is the case; in a real-world scenario operators may congregate together, which would facilitate connectivity). The likelihood of further finding a peer whose system status is sufficient is even less; hence, the very low histogram bars for short ranges. If no peer is found when a transfer is required, the selection fails. Conversely, despite much longer communication ranges, the decreased network density in such cases diminishes the probability of establishing connectivity and thus selecting a suitable peer. Indeed, only the availability of a connected neighbor would be impacted by the increased range. Multiple selection methods choosing the same node explains the similar results for very long ranges; a limited number of peers are available as the nodes are dispersed over a much wider area.

5.2.4 Anisotropic RF Propagation

In addition to the random mobility model introduced in the previous scenario, the effect of anisotropic RF propagation is introduced. In these simulations, the variation in RF propagation is as much as 15 percent of the maximum communication range with propagation in each direction randomly determined based upon the difference between the lower and upper bounds. This parameter was estimated based on empirical tests using wireless Ethernet, but is used here for illustrative purposes rather than providing a rigorously constructed model of wireless com-

munication range, which is beyond the scope of this work. In these simulations, the context-aware selection algorithm generally considers only the lower bound of full connectivity, thereby behaving similarly to the isotropic case with a somewhat shorter communication range. As in the previous section, Figure 5.6 illustrates the total number of selections per algorithm in which the selected peer had the longest actual availability time, determined over 1,000 independent simulation trials. Because the connectivity model is essentially the same as the isotropic case, the results do not substantively differ.

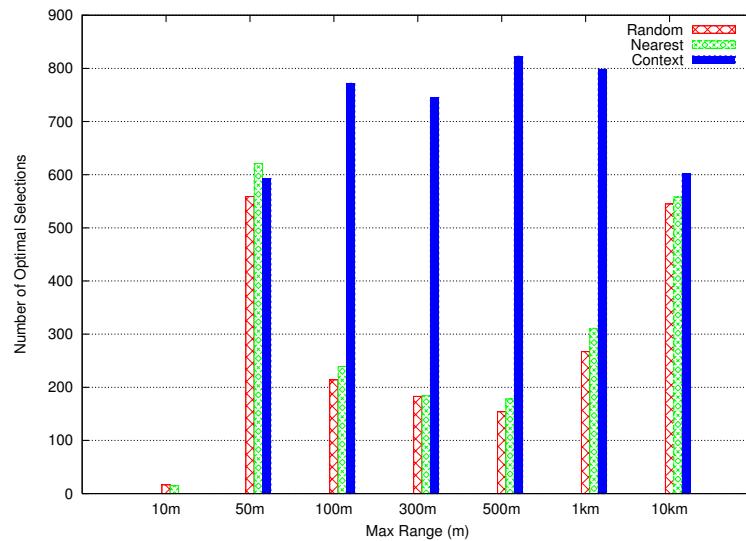


Figure 5.6. Selection: Anisotropic RF Propagation

5.2.5 Localization Error

Finally, in addition to the revised mobility and RF propagation models, the evaluation of the context-aware selection algorithm is now subjected to the addi-

Algorithm 4 Weighted Peer Selection

```
1: for  $i = 1$  to  $n$  do
2:   if Connected( $k, i$ ) then
3:     if (Space-Available( $i$ )  $\geq$  Space-Required( $k$ ))  $\wedge$   $\neg$  State-Critical( $i$ ) then
4:       Score( $i$ )  $\Leftarrow 1$ 
5:     else
6:       Score( $i$ )  $\Leftarrow 0$ 
7:     end if
8:   end if
9: end for
10: if select = true then
11:   for  $j = 1$  to  $n$  do
12:     if (Score( $j$ )  $\neq 0$ )  $\wedge$  ( $j \neq k$ ) then
13:        $v = \frac{1}{\Delta t} \sqrt{((x_{i_t} - x_{k_t}) - (x_{i_{t-1}} - x_{k_{t-1}}))^2 + ((y_{i_t} - y_{k_t}) - (y_{i_{t-1}} - y_{k_{t-1}}))^2}$ 
14:        $\Delta x = (x_{i_t} - x_{k_t}) - (x_{i_{t-1}} - x_{k_{t-1}})$ 
15:        $\Delta y = (y_{i_t} - y_{k_t}) - (y_{i_{t-1}} - y_{k_{t-1}})$ 
16:        $\Delta r = \sqrt{\Delta x^2 + \Delta y^2}$ 
17:        $D = (x_{i_{t-1}} y_{i_t}) - (x_{i_t} y_{i_{t-1}})$ 
18:        $int_x = \frac{D \Delta y - \frac{\Delta y}{|\Delta y|} \Delta x \sqrt{range^2 \Delta r^2 - D^2}}{\Delta r^2}$ 
19:        $int_y = \frac{-D \Delta x - |\Delta y| \sqrt{range^2 \Delta r^2 - D^2}}{\Delta r^2}$ 
20:        $w_i = \frac{c_{i_t} + c_{i_{t-1}}}{e_{i_t} + e_{i_{t-1}}}$ 
21:       Score( $i$ ) =  $w_i \frac{\sqrt{(x_{int} - x_{i_t})^2 + (y_{int} - y_{i_t})^2}}{v}$ 
22:       if Score( $i$ )  $\geq score_{max}$  then
23:         selected =  $i$ 
24:          $score_{max} = \text{Score}(i)$ 
25:       end if
26:     end if
27:   end for
28: end if
29: return selected
```

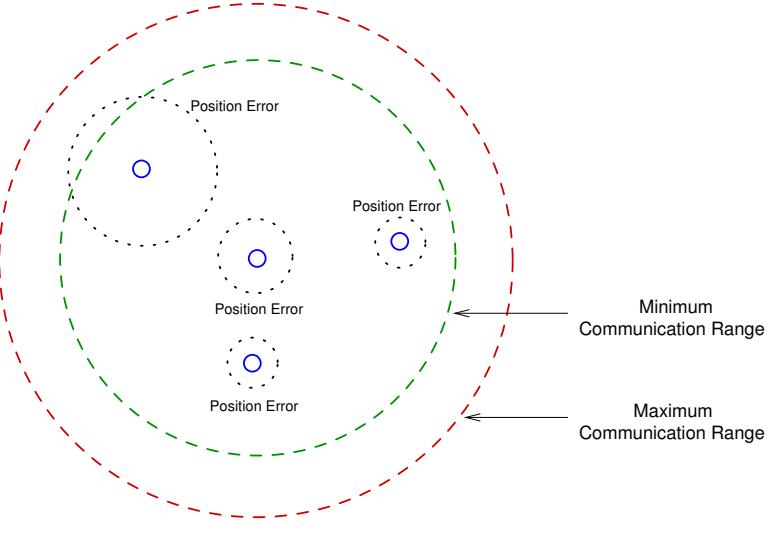


Figure 5.7. Candidate Selection (With Localization Error)

tion of modeled localization error. Each node's position is subject to both error and uncertainty, as illustrated in Figure 5.7. In this particular scenario, GPS error was used as a representative method of producing localization error in all nodes. The introduction of error and uncertainty means that selections must be made based on each neighbor's reported position, which may be subject to substantial error and uncertainty, as in a real-world scenario may be determined from any of several available methods. In this case, only reported positions are evaluated, and all positions are assumed to be error-free for purposes of selection, i.e., no accounting for such error is done yet.

Compared to the previous cases, the introduction of localization error in this case causes the context-aware approach to slightly underperform relative to similar cases with perfect localization. Moreover, for short communication ranges, the random and nearest-neighbor approaches both outperform the context-aware method due to the granularity of GPS error in relation to the short potential

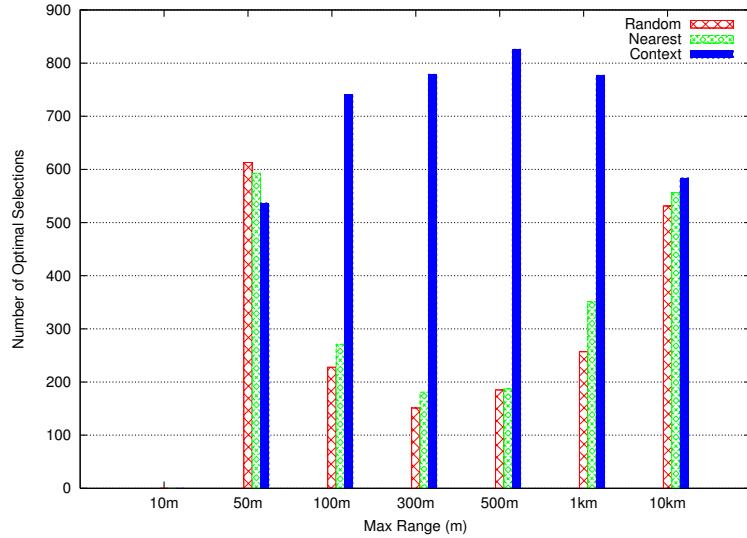


Figure 5.8. Selection: Localization Error

availability times, as shown in Figure 5.8. In such cases, particularly when the positions become jittery, e.g., short GPS drift lengths, but error is pronounced, even a short window of opportunity cannot effectively be determined. However, as the communication ranges increase, the effectiveness of the algorithm similarly increases (assuming a reasonable network density to provide an adequate number of nodes to select from). Even with GPS error, the observed tendency of such error to drift in the same general direction for several sections actually assists the algorithm, and such drift is incorporated in the GPS error model used in this evaluation. If the direction of the error were strictly uniformly distributed and changed at every time step, the algorithm would be far less effective in estimating availability. Note that such a phenomenon is not impossible; a very unfortunate GPS fix with a jittery position could render estimating an availability window with any degree of accuracy much more difficult. Other localization techniques,

e.g., dead reckoning in particular, are subject to far less jitter from one location measurement to the next.

5.2.5.1 Weighting Localization Error

To help mitigate the effects of localization error on the selection algorithm, it is possible to weight the score of each candidate to account for both error and confidence levels. Doing so requires both available metrics from each localization technique as well as a predetermined model of confidence for those techniques. Clearly, a node whose location information has either high error magnitude or low confidence should have a lower score than another whose estimated error is low and confidence level is high. One very simple weighting scheme is to simply multiply each candidate's score by the ratio of confidence to horizontal error, i.e., for node i , the score w_i is modified such that $w'_i = \frac{c_i}{e_i} w_i$.

Therefore, the algorithm with modifications in its entirety is specified in Algorithm 4. In this algorithm, node k is the selector.

To evaluate the effect of this modified scoring method, the maximum actual localization error and the maximum horizontal error metric was increased independently for a system of 100 nodes. As before, the Random Waypoints mobility model was used, as was a 300-meter maximum communication range. Figure 5.9 shows the increase in average availability as a result of weighting each score based on estimated error and confidence. The graph shows an increase in mean availability of as much as 20 percent by simply lowering the score of those nodes whose available metrics suggest poor localization. When localization error is small, however, weighting the score based on estimated metrics produces results very similar to the unweighted case. As either the actual error, the estimated error, or both,

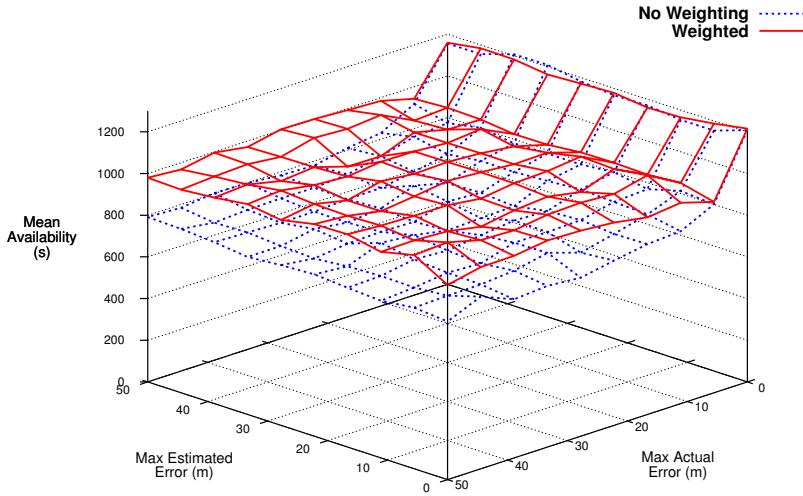


Figure 5.9. Selection: Localization Error (With Weighting)

increase, then increases in mean availability can be realized. Additionally, it is possible that a more sophisticated weighting scheme could produce further improvements.

5.3 Conclusion

In this chapter a method for selecting peers to offload transient data is proposed that accounts for both heterogeneous and dynamic system state among nodes as well as estimated availability through mobility. Previous work has shown that mobility prediction can improve overall availability and link longevity, which is important for data transfer operations, but this work goes a step further and incorporates dynamic state information in the peer selection process. This approach has the advantage of choosing nodes most likely to remain available for recovery

in applications for which transient data has high value. Further improvements can be realized by accounting for localization error in the scoring process.

Simulation results show that this approach improves the success rates of both the initial selection and data recovery. For small scale networks, to include those with very limited wireless range, accounting for predicted availability may be of limited benefit. However, as the network scales, particularly with more powerful wireless radios, using a fully context-aware selection method can significantly increase the availability of temporarily replicated data, which in turn can lead to greater success for operations which depend on the reliable access to and availability of sensor information.

CHAPTER 6

CONCLUSION

Component-based approaches to building systems offer many advantages in terms of cost savings and lower development times, given the availability of commodity hardware components with standard interfaces. However, many of the attributes that provide such advantages simultaneously pose challenges for developers of robust mobile applications. Understanding interfaces to sensor components or even examining metadata from each device is insufficient. For robust mobile applications dependent on location information, error metrics must be explicitly understood and modeled, as many types of sensors, for instance GPS receivers, often exhibit error in unexpected ways. The foundation of this work is an evaluation of an array of existing sensor hardware for location-sensitive applications and mobile ad-hoc networking.

While it is well understood that GPS localization can be very accurate and precise, the most accurate positioning, such as found in military-grade applications, typically requires expensive hardware, laborious offline analysis, etc., and thus is not feasible at a large deployment scale or in real time. The expected error in low-cost, commercial GPS systems is still acceptable for localization at human scale, but the fundamental limitations of GPS which manifest in many environments must be addressed for any general purpose mobile application. Robust

personal navigation in arbitrary environments presents tremendous opportunities for system and application developers.

As application requirements specify fine-grained positioning for human-scale navigation or other location-sensitive applications, robust performance necessitates the use of techniques to augment existing sensors. Of course, this idea is well understood among researchers and developers of mobile sensor networks, but existing correction techniques in general have proven to be inadequate. The example of mobile backup systems shows that even with localization quality generally deemed acceptable, application performance can still suffer if error is not accounted for in some way.

In this work, I presented and evaluated several methods to account for localization error. Starting with a standalone dead reckoning system incorporated into TeamTrak, which can augment GPS in environments in which GPS positioning is inconsistent or error-prone. However, localization via dead reckoning alone is insufficient, as periodic corrections are required, with frequency depending largely on the number of changes in direction as well as the quality of the initial position, which may be quite poor. I presented two methods for providing such correction, depending on available technology.

Whether a reliable ranging technique is available or not, correction of pedestrian dead reckoning systems can be accomplished by cooperatively sharing location information. With no means of measuring spatial separation between nodes, combining positions by simply averaging those of connected neighbors places an upper limit on position error as a function of both communication range and measurement error, which can increase accuracy of positions whose measurement error is effectively unbounded. Therefore, even with a very limited amount of informa-

tion, cooperatively sharing data can effect improvements. Of course, with more information available, better results can be achieved. Averaging positions which are filtered by estimated error can improve positions, while simultaneously avoiding the poisoning effects of highly inaccurate positions. For such simple correction schemes, the extent to which the accuracy of a position is improved is a function of all of the following: measurement error of the positions used for correction, the density and geometry of the anchors, communication range, and the “staleness” of the position to be corrected.

If ranging is available, additional options for correction positions are available. Not only can the accuracy of localization measured via dead reckoning be improved using trilateration, but correcting GPS positions subjected to drift may be possible using other GPS positions. It may also have the effect of reducing jitter. However, doing so requires a filtering mechanism to screen out positions with high error or whose error metric has low confidence. This requires both a model of GPS error as well as a protocol for cooperatively sharing location information. Particularly in the case of GPS-to-GPS error correction, error in position must be accounted for in order to avoid making locations worse, particularly in environments with poor anchor node density or unreliable metrics.

Even with a discipline for effecting improvements in localization using shared sensor data, the presence of (occasionally significant) measurement error is a certainty, particularly due to the effects of mobility and, more significantly, fundamental limitations of localization techniques. This will continue to be the case for the foreseeable future. For applications in which precise localization is a requirement, such error must still be taken into account. One possible way to do this, demonstrated in this dissertation, is by weighting the utility of a localized

node with both the magnitude of the estimated position error and the confidence level in such estimates. Such accounting can facilitate better selection decisions in protocols such as geographic forwarding of routing packets.

While I have showed how significant gains in location quality can be realized through sharing information, this work has only barely scratched the surface of the larger problem of localization in mobile sensor networks. Some challenging, and potentially quite daunting, challenges remain open problems, particularly rigorous evaluation of the error of positions determined via the simplest cooperative localization schemes.

BIBLIOGRAPHY

- N. Abramson. The ALOHA System - Another Alternative for Computer Communications. In *Proceedings of the Fall 1970 AFIPS Computer Conference*, pages 281–285, 1970.
- N. Abu-Ghazaleh, K.-D. Kang, and K. Liu. Towards Resilient Geographic Routing in WSNs. In *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks (Q2SWinet '05)*, pages 71–78, October 2005.
- B. Aiken, V. Bahl, B. Bhattacharjee, B. Braden, M. Crovella, V. Frost, M. Gerla, J. Heidemann, K. Jeffay, J. Kurose, J. Lepreau, R. Mundy, C. Partridge, L. Peterson, R. Rao, D. Raychaudhuri, H. Schulzrinne, J. Shapiro, K. Sollins, J. Turner, A. Vahdat, H. Vin, and E. Zegura. NSF Workshop on Network Research Testbeds, October 2002. URL http://www-net.cs.umass.edu/testbed_workshop/.
- I. Akyildiz, Y. Lin, W. Lai, and R. Chen. A New Random Walk Model for PCS Networks. *IEEE Journal on Selected Areas in Communications*, 18(7):1254–1260, 2000.
- I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks (Elsevier) Journal*, 38(4):393–422, March 2002.
- M. Amundson. Dead Reckoning for Consumer Electronics. In *Proceedings of the 2006 IEEE/ION Position, Location, and Navigation Symposium (PLANS)*, April 2006.
- A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, M. Nesterenko, R. Shah, S. Kulkami, M. Aramugam, L. Wang, M. Gouda, Y. Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker. ExScal: Elements of an Extreme Scale Wireless

- Sensor Network. In *Proceedings of the 11th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, August 2005.
- R. Aylward, S. D. Lovell, and J. Paradiso. A Compact, Wireless, Wearable Sensor Network for Interactive Dance Ensembles. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, April 2006.
- J. Bachrach and C. Taylor. *Handbook of Sensor Networks*, chapter 1. Wiley, 2005.
- P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *IEEE INFOCOM*, April 2001.
- F. Bai and A. Helmy. *A Survey of Mobility Models*, chapter 1, pages 1–30. Kluwer Academic Publishers, 2004.
- D. Beyer. Accomplishments of the DARPA SURAN Program. In *Proceedings of the Military Communications Conference (MILCOM)*, September 1990.
- E. Biagioli and K. Bridges. The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species. *Special Issue in Distributed Sensor Networks for the International Journal of High Performance Computing Applications*, 16(3), August 2002.
- Bluetooth SIG. Specification of the Bluetooth System, November 2004. URL <http://www.bluetooth.com>.
- R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Communications Magazine*, 43(3):121–131, March 2005.
- R. Buick. GPS Guidance - Making an Informed Decision. In *Proceedings of the 6th International Conference on Precision Agriculture, Agricultural Buildings and Structures*, pages 271–284, 2002.
- N. Bulusu, J. Heidemann, and D. Estrin. GPS-Less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications Magazine*, 2000.
- F. Burghardt, S. Mellers, and J. Rabaey. The PicoRadio Test Bed, December 2002. URL http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/Test_Bed.
- Z. Butler, P. Corke, R. Peterson, and D. Rus. Networked Cows: Virtual Fences for Controlling Cows. In *MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*, June 2004.

- T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad-Hoc Network Research. *Wireless Communication and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, 2(5):483–502, 2002.
- J. Castro, V. Santos, and M. I. Ribeiro. A Multi-Loop Robust Navigation Architecture for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, May 1998.
- A. Cerpa, N. Busek, and D. Estrin. SCALE: A Tool for Simple Connectivity Assessment in Lossy Environments. Technical Report 21, Center for Embedded Network Sensing, University of California, Los Angeles, September 2003.
- R. Chandra, J. Padhye, A. Wolman, and B. Zill. A Location-Based Management System for Enterprise Wireless LANs. In *NSDI 07: 4th USENIX Symposium on Networked System Design and Implementation*, June 2007.
- W. Chao, J. Macker, and J. Weston. Naval Research Lab’s Mobile Network Emulator, January 2003. URL <http://downloads.pf.itd.nrl.navy.mil/proteantools/MNE-1.5.pdf>.
- R. Chellappa-Doss, A. Jennings, and N. Shenoy. User Mobility Prediction in Hybrid and Ad Hoc Wireless Networks. In *Australian Telecommunications Networks and Applications Conference*, December 2003.
- K. Chin, J. Judge, A. Williams, and R. Kermode. Implementation Experience with Manet Routing Protocols. *SIGCOMM Computer Communication Review (CCR)*, 32(5):49–59, November 2002.
- Committee on Networked Systems of Embedded Computers. Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers. National Academy Press, 2001, report of the National Research Council Committee on Networked Systems of Embedded Computers, 2001. URL http://books.nap.edu/html/embedded_everywhere.
- M. Corr and C. Okino. Networking Reconfigurable Smart Sensors. In *SPIE Interational Symposium on Enabling Technologies for Law Enforcement and Security*, volume 4232, November 2000.
- N. Correal and N. Patwari. Wireless Sensor Networks: Challenges and Opportunities. In *Proceedings of the 2001 Virginia Tech Symposium on Wireless Personal Communications*, pages 1–9, June 2001.
- Corvallis Technology, Inc. *Introduction to the Global Positioning System for GIS and TRAVERSE*. CMTinc.com, 1996. URL <http://www.cmtinc.com/gpsbook/>.

- P. De, A. Raniwala, S. Sharma, and T. Chiueh. MiNT: A Miniaturized Network Testbed for Mobile Wireless Research. In *Proceedings of INFOCOM 2005*, pages 2731–2742, March 2005.
- K. Delin and S. Jackson. Sensor Web for In Situ Exploration of Gaseous Biosignatures. In *IEEE Aerospace Conference*, volume 7, March 2000.
- J. Deng, Y. Han, P.-N. Chen, and P. K. Varshney. Optimum Transmission Range for Wireless Ad Hoc Networks. In *IEEE Wireless Communications and Networking Conference*, March 2004.
- L. Doherty, L. Ghaoui, and K. Pister. Convex Position Estimation in Wireless Sensor Networks. In *IEEE INFOCOM*, April 2001.
- R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multihop Wireless Networks. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 133–144, September 2004.
- T. Falck, J. Espina, J.-P. Ebert, and D. Dietterle. BASUMA - The Sixth Sense for Chronically Ill Patients. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, April 2006.
- L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie. Design of a Wireless Assisted Pedestrian Dead Reckoning System - The NavMote Experience. *IEEE Transactions on Instrumentation and Measurement*, 54(6):2342–2358, 2005.
- B. Feder. Wireless Sensor Networks Spread to New Territory. *New York Times*, July 2004. URL <http://www.nytimes.com/2004/07/26/business/26sensor.html>.
- Garmin International Inc. *GPS-18 Technical Specifications (190-000307-00)*, June 2005. URL http://www.garmin.com/manuals/425_TechnicalSpecification.pdf.
- L. Girod, J. Elson, A. Cerpa, T. Staphopoulos, N. Ramanathan, and D. Estrin. EmStar: A Software Environment for Developing and Deploying Wireless Sensor Networks. In *Proceedings of the 2004 USENIX Technical Conference*, 2004.
- T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive Routing in Ad Hoc Networks. In *7th ACM/IEEE Conference on Mobile Computing and Networking*, July 2001.

- S. Han, Q. Zhang, H. Noh, and B. Shin. Dynamic Performance Evaluation Method for DGPS Receivers Under Linear Parallel-Tracking Applications. *Transactions of the ASAE*, 47(4):321–329, 2004.
- T. Hara. Data Replication Issues in Mobile Ad Hoc Networks. In *Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA 05)*, August 2005.
- T. He, C. Huang, B. Blum, and J. Stankovic. Range-Free Localization Schemes for Large Scale Sensor Networks. In *Proceedings of Mobicom*, September 2003.
- T. He, S. Krishnamurthy, A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-Efficient Surveillance System Using Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.
- G. Heineman and W. Councill. *Component-Based Software Engineering - Putting the Pieces Together*. Addison-Wesley, 2001.
- M. Heisenbüttel, T. Braun, T. Roth, and T. Bernoulli. GNU/Linux Implementation of a Position-Based Routing Protocol. In *Proceedings of the IEEE ICPS Workshop on Wireless Mesh Networks (MeshNets)*, July 2005.
- J. Hemmes, D. Thain, C. Poellabauer, C. Moretti, P. Snowberger, and B. McNutt. Lessons Learned Building TeamTrak: An Urban/Outdoor Mobile Ad Hoc Network Testbed. In *WASA '07: International Conference on Wireless Algorithms, Systems, and Applications*, August 2007.
- T. Hester, R. Hughes, D. Sherrill, B. Knorr, M. Akay, J. Stein, and P. Bonato. Using Wearable Sensors to Measure Motor Abilities following Stroke. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, April 2006.
- Hewlett-Packard Company. *User's Guide, HP iPAQ Navigation System (348773-001)*, September 2003. URL <http://bizsupport.austin.hp.com/docs/support/SupportManual/c00044146.pdf>.
- L. Hu and D. Evans. Localization for Mobile Sensor Networks. In *Proceedings of Mobicom*, September 2004.
- IEEE 802.16 Working Group on Broadband Wireless Access Standards. EEE Std 802.16e, August 2005. URL http://www.ieee802.org/16/pubs/80216_Cor1.html.
- Information Sciences Institute. The NS-2 Network Simulator, 2006. URL <http://www.isi.edu/nsnam/ns>.

- Institute of Navigation Standards. *Recommended Test Procedures for GPS Receivers*, 1997.
- P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom '99)*, pages 195–206, August 1999.
- D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, 353:153–181, 1996.
- P. Juang, H. Oki, Y. Wang, M. Martonosi, S. Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, October 2002.
- J. Jubin and J. Turnow. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE*, 75:21–32, January 1987.
- G. Judd and P. Steenkiste. Repeatable and Realistic Wireless Experimentation Through Physical Emulation. *SIGCOMM Computer Communication Review (CCR)*, 34(1):63–63, January 2002.
- R. Kahn. The Organization of Computer Resources into a Packet Radio Network. *IEEE Transactions on Communications*, COM-25(1):169–178, January 1977.
- R. Kahn, S. Gronemeyer, J. Burchfiel, and R. Kunzelman. Advances in Packet Radio Technology. *Proceedings of the IEEE*, 66:1468–1496, November 1978.
- E. D. Kaplan. *Understanding the GPS: Principles and Applications*. Artech House, Boston, MA, 2nd edition, February 1996.
- A. Karygiannis and E. Antonakakis. mLab: A Mobile Ad Hoc Network Test Bed. In *3rd IEEE Consumer Communications and Networking Conference (CCNC 2006)*, volume 2, pages 1312–1313, January 2006.
- Q. Ke, D. Maltz, and D. Johnson. Emulation of Multi-Hop Wireless Ad-Hoc Networks. In *Proceedings of the 7th International Workshop on Mobile Multimedia Communications (MoMuC)*, October 2000.
- W. Kiess and M. Mauve. A Survey on Real-World Implementations of Mobile Ad-Hoc Networks. *Ad Hoc Networks*, 5(3):324–339, April 2007.
- A. Köhne and M. Wößner. How GPS Works. An Explanation of the NAVSTAR-GPS System. <http://www.kowoma.de/en/gps/index.htm>, 2005.

- K. Langendoen and N. Reijers. Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 43(4):499–518, 2003. ISSN 1389-1286.
- H. Larkin. *Predicting Connectivity in Wireless Ad Hoc Networks*. PhD thesis, Bond University, 2005.
- J. Ledlie, P. Gardner, and M. Seltzer. Network Coordinates in the Wild. In *NSDI 07: 4th USENIX Symposium on Networked System Design and Implementation*, June 2007.
- A. Leon-Garcia and I. Widjaja. *Communication Networks: Fundamental Concepts and Key Architectures*. McGraw-Hill Higher Education, 2000. ISBN 0-07-022839-6.
- J. Leonard and H. Durrant-Whyte. Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.
- P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 126–137, 2003.
- H. Liu and G. Pang. Accelerometer for Mobile Robot Positioning. In *IEEE Industry Applications Conference*, July 1999.
- J. Liu, Y. Yuan, D. Nicol, R. Gray, C. Newport, D. Kotz, and L. Perrone. Simulation Validation Using Direct Execution of Wireless Ad-Hoc Routing Protocols. In *Proceedings of the Workshop on Parallel and Distributed Simulation (PADS)*, pages 7–16, May 2004.
- J. Liu, Y. Yuan, D. Nicol, R. Gray, C. Newport, D. Kotz, and L. Perrone. Validation of Wireless Models Using Direct-Execution Simulation of Ad-Hoc Routing Protocols. *Simulation: Transactions of the Society for Modeling and Simulation International*, January 2005.
- K. Lorincz and M. Welsh. MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In *Proceedings of the International Workshop on Location and Context-Awareness*, May 2005.
- D. Madigan, P. Krishnan, A. Krishnakumar, and I. Zorych. Location Estimation in Wireless Networks: A Bayesian Approach. *Statistica Sinica*, 16(2):495–522, 2006.

- A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, September 2002.
- G. Malkin. RIP Version 2. Internet Engineering Task Force RFC 2453, November 1998.
- D. Maltz, J. Broch, and D. Johnson. Experiences Designing and Building a Multi-Hop Wireless Ad-Hoc Network Testbed. Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon University, 1999.
- H. Marcy, J. Agre, L. Clare, and N. Romanov. Development Platform for Self-Organizing Wireless Sensor Networks. In *SPIE Proceedings*, volume 3713, July 1999.
- A. Mason, N. Yazdi, K. Najafi, and K. Wise. A Low-Power Wireless Microinstrumentation System for Environmental Monitoring. In *International Conference on Solid-State Sensors and Actuators*, volume 1, June 1995.
- M. Matthes, H. Biehl, M. Lauer, and O. Drobnik. MASSIVE: An Emulation Environment for Mobile Ad-Hoc Networks. In *Proceedings of the 2nd Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*, January 2005.
- Melbourne Wireless Committee. *Melbourne Wireless*, 2008. URL <http://www.melbournewireless.org.au>.
- MIT Roofnet*. MIT CSAIL, 2008. URL <http://www.pdos.lcs.mit.edu/roofnet>.
- D. Moore, J. Leonard, D. Rus, and S. Teller. Robust Distributed Localization With Noisy Range Measurements. In *Proceedings of Sensys*, November 2004.
- D. Niculescu. Ad-hoc Positioning System. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2001.
- D. Niculescu and B. Nath. DV-Based Positioning in Ad-Hoc Networks. *Telecommunications Systems*, 22(1):267–280, 2003.
- B. Noble, M. Satyanarayanan, G. Nguyen, and R. Katz. Trace-Based Mobile Network Emulation. In *Proceedings of the 1997 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 51–61, September 1997.
- E. Nordstrom, P. Gunningberg, and H. Lundgren. A Testbed and Methodology for Experimental Evaluation of Wireless Mobile Ad Hoc Networks, February 2005.

- D. Odell and P. Wright. Concurrent Product Design: A Case Study on the Pico Radio Test Bed. Master's thesis, University of California at Berkeley, 2002.
- N. Oliver and F. Flores-Mangas. HealthGear: A Real-time Wearable System for Monitoring and Analyzing Physiological Signals. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, April 2006.
- G. Pang and H. Liu. Evaluation of a Low-cost MEMS Accelerometer for Distance Measurement. *Journal of Intelligent and Robotic Systems*, 30(3):249–265, March 2001.
- C. Park and C. H. Chou. Eco: Ultra-Wearable and Expandable Wireless Sensor Platform. In *International Workshop on Wearable and Implantable Body Sensor Networks*, April 2006.
- M. Pascoe, J. Gomez, V. Rangel, and M. Lopez-Guerrero. Modeling Duration of Routes in Multihop Ad hoc Networks. In *20th IEEE Canadian Conference on Electrical and Computer Engineering*, April 2007.
- B. Pazand and C. McDonald. A Critique of Mobility Models for Wireless Network Simulation. In *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science*, pages 141–146, July 2007.
- C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, August 1994.
- M. Perkins, N. Correal, and R. O'Dea. Emergent Wireless Sensor Networks Limitations: A Plea for Advancement in Core Technologies. In *Proceedings of IEEE Sensors*, volume 2, pages 1505–1509, June 2002.
- A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom '01)*, July 2001.
- J. Person. Dilution of Precision: Writing Your Own GPS Applications, 2003. URL [{http://www.geoframeworks.com/Articles/WritingApps2_3.aspx}](http://www.geoframeworks.com/Articles/WritingApps2_3.aspx).
- N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-Free Distributed Localization in Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.

- D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusano, H. Liu, and M. Singh. Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005.
- M. Refan, K. Mohammadi, and M. Mosavi. Improvement on a Low Cost Positioning Sensor Accuracy. In *AsiaSense 2003: Asian Conference on Sensors*, July 2003.
- L. Rising, editor. *Design Patterns in Communications Software*. Cambridge University Press, New York, NY, USA, 2001. ISBN 0-521-79040-9.
- J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy. Experimenting with a Multi-Radio Mesh Networking Testbed. In *Proceedings of the 1st Workshop on Wireless Network Measurements (WINMee)*, April 2005.
- C. Rohner, E. Nordström, P. Ginningberg, and C. Tschudin. Interactions Between TCP, UDP, and Routing Protocols in Wireless Multi-hop Ad Hoc Networks. In *4th ACM/IEEE Conference on Mobile Computing and Networking*, October 1998.
- G. Roston and E. Krotkov. Dead Reckoning Navigation for Walking Robots. Technical Report CMU-RI-TR-91-27, Robotics Institute, Carnegie Mellon University, November 1991.
- E. Royer, P. Melliar-Smith, and L. Moser. A Statistical Analysis of the Optimum Node Density for Ad Hoc Mobile Networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, June 2001.
- W. Rupprecht. *Post SA GPS Accuracy Measurements*, May 2007. URL <http://www.wsrcc.com/wolfgang/gps/accuracy.html>.
- G. Sandoval. Lessons Learned from Kim Tragedy, December 2007. URL http://www.news.com/Lessons-learned-from-Kim-tragedy/2100-1028_3-6221778.html.
- S. Sanghani, T. Brown, S. Bhandare, and S. Doshi. ORBIT Measurements Framework and Library (OML): Motivations, Design, Implementation, and Features. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, March 2003.
- N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *Proceedings of the 2nd ACM Workshop on Wireless Security*, September 2003.

- C. Savarese, K. Langendoen, and J. Rabaey. Robust Positioning Algorithms for Distributed Ad-hoc Wireless Sensor Networks. In *USENIX Technical Annual Conference*, June 2002.
- A. Savvides, H. Park, and M. Srivastava. The Bits and Flops of the N -hop Multilateration Primitive for Node Localization Problems. In *First ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.
- A. Savvides, C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Adhoc Networks of Sensors. In *Proceedings of Mobicom*, July 2004.
- K. Shannon, C. Ellis, and G. Hoette. Performance of “Low-Cost” GPS Receivers for Yield Mapping. ASAE Meeting Paper No. 02-1150, 2002.
- R. Stoleru, T. He, J. Stankovic, and D. Luebke. A High-Accuracy, Low-Cost Localization System for Wireless Sensor Networks. In *SenSys ’05: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, November 2005.
- W. Su. Mobility Prediction in Wireless Networks. In *2000 Military Communications Conference*, October 2000.
- G.-L. Sun and W. Guo. Comparison of Distributed Localization Algorithms for Sensor Network with a Mobile Beacon. In *IEEE International Conference on Networking, Sensing and Control*, March 2004.
- R. Taylor, M. Schrock, J. Bloomfield, G. Bora, G. Brockmeier, W. Burton, B. Carlson, J. Gattis, R. Groening, J. Kopriva, N. Oleen, J. Ney, C. Simmelink, and J. Vondracek. Dynamic Testing of GPS Receivers. *Transactions of the ASAE*, 47(4):1017 – 1025, 2004.
- D. Thain, C. Moretti, and J. Hemmes. Chirp: A Practical Global Filesystem for Cluster and Grid Computing. *Journal of Grid Computing*, 2008. doi: 10.1007/s10723-008-9100-5.
- S. Thomson and L. Smith. Dynamic Testing of GPS Receivers on Agricultural Aircraft for Remote Sensing and Variable-Rate Aerial Application. In *Proceedings of the 2006 IEEE/ION Position, Location, and Navigation Symposium (PLANS)*, pages 1067 – 1070, April 2006.
- General Information on GPS.* United States Coast Guard, 2002. URL <http://www.navcen.uscg.gov/gps/default.htm>.

- N. Vaidha, J. Bernhard, V. Veeravalli, P. Kumar, and R. Iyer. Illinois Wireless Wind Tunnel: A Testbed for Experimental Evaluation of Wireless Networks. Wireless working group, University of Illinois at Urbana-Champaign, August 2004.
- M. Varshney, Z. Xu, S. Mohan, Y. Yang, D. Xu, and R. Bagrodia. WHYNET: A Framework for In-Situ Evaluation of Heterogeneous Mobile Wireless Systems. In *WinTECH '07: Proceedings of the the 2nd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, pages 35–42, 2007.
- B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, December 2002.
- B. White, J. Lepreau, and S. Guruprasad. Lowering the Barrier to Wireless and Mobile Experimentation. *SIGCOMM Computer Communication Review (CCR)*, 33(1):47–52, 2003.
- D. Wilson. *GPS Error When Averaging Position*, May 2001. URL <http://users.erols.com/dwilson/gpsavg.htm>.
- Wireless Leiden Foundation. *Wireless Leiden: Mission, Vision and Strategy*, 2008. URL <http://wirelessleiden.nl>.
- B. Wong, I. Stoyanov, and E. Sirer. Octant: A Comprehensive Framework for the Geolocation of Internet Hosts. In *NSDI 07: 4th USENIX Symposium on Networked System Design and Implementation*, June 2007.
- S. Wormley. *GPS Errors & Estimating Your Receiver's Accuracy*, 2007. URL http://edu-observatory.org/gps/gps_accuracy.html.
- N. Xu. A Survey of Sensor Network Applications. *Unknown*, 2003.
- M. Yarvis, W. Conner, L. Krishnamurthy, J. Chhabra, B. Elliott, and A. Mainwaring. Real-World Experiences with an Interactive Ad Hoc Sensor Network. In *Proceedings of the International Workshop in Ad Hoc Networking (IWAHN)*, pages 143–151, August 2002.
- X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks, May 1998. URL <http://pcl.cs.ucla.edu/projects/glomosim>.

- Y. Zhang and W. Li. An Integrated Environment for Testing Mobile Ad-Hoc Networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, pages 104–111, June 2002a.
- Y. Zhang and W. Li. An Integrated Environment for Testing Mobile Ad Hoc Networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 104–111, June 2002b.
- F. Zhao and L. Guibas. *Wireless Sensor Networks*. Morgan Kaufmann Publishers, 2004.

This document was prepared & typeset with L^AT_EX 2_<, and formatted with NDDiss2_< classfile (v3.0[2005/07/27]) provided by Sameer Vijay.