ABSTRACT

PENG, RONG. Localization and Mobility Modeling in Wireless Ad Hoc Networks. (Under the direction of Dr. Mihail L. Sichitiu).

The availability of wireless devices and their increasing ability of sharing information has stimulated the emergence of the wireless ad hoc networks. In this dissertation, we focus on two topics related to wireless ad hoc networks. One is the localization in the static wireless sensor networks and the other one is the mobility modeling in the mobile ad hoc networks.

Localization is a fundamental service for many applications of wireless sensor networks. In the first part, we consider a distributed, probabilistic localization approach, which is suitable for systems with inaccurate range measurements and a small number of beacons. Two solutions are provided: an RSS-based solution and an angle-based approach corresponding to the measurements used between neighbor nodes. The basic idea of these approaches is to restrict the possible locations of the nodes by using probabilistic constraints. The proposed probabilistic approach is evaluated through simulations based on real-world and simulated measurements; the results are compared with the Cramer-Rao lower bound and other RSS-based and angle-based localization algorithms. The results show that, with inaccurate range measurements, and a small number of beacons, the proposed probabilistic approach outperforms existing methods and approaches the optimum bound.

Due to the scarcity of real mobile ad hoc deployments, most protocol evaluations are carried out through simulations. One of the core components or network simulations is the mobility model, which characterizes the mobility patterns of the mobile devices in a network and uses such patterns to reproduce trajectories of the mobile devices accurately. In the second part of this dissertation, we study several real world traces including wireless LAN and bus traces. The statistical properties of several features that primarily determine the behavior of the mobile users are extracted and thoroughly analyzed. Using the extracted statistical properties and the statistical similarities of those features, we propose a simple generalized mobility generator that can generate both realistic and diversified synthetic traces.

Localization and Mobility Modeling in Wireless Ad Hoc Networks

by

Rong Peng

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Electrical and Computer Engineering

Raleigh, North Carolina

2007

Approved By:

Dr. Arne A. Nilsson	Dr. Do Young Eun
Dr. Mihail L. Sichitiu Chair of Advisory Committee	Dr. Rudra Dutta

Dedication

To the most important people in my life, Mom, Dad, Rui and Huanhuan.
Thank you for your patience, understanding and endless love.

Biography

Rong Peng was born in 1978 to a loving family. She spent her childhood and most of her teen years on the campus of Tsinghua University, Beijing, China, where her parents work. From 1996 to 2001, she attended Beijing University of Technology, from where she received her B.S. degree in the Electronic Engineering. After that, she moved to the United States and spent two delightful years in Newark, Delaware. In 2003, she received her M.S. degree in Electrical Engineering from the University of Delaware. Then Rong moved to Raleigh, North Carolina to pursue a PhD in Electrical Engineering at North Carolina State University. While at NC state, she joined the Wireless Ad-hoc and Local Area Networks Research (WALAN) lab under the direction of Dr. Mihail L. Sichitiu. She focused two research projects involving wireless sensor networks and mobile ad hoc networks. Her research interests include sensor networks, mobile ad hoc networks, distributed systems, system modeling, simulation and performance evaluation.

Acknowledgements

First I would like to disclose my sincere gratitude to my advisor, Dr. Mihail L. Sichitiu. I thank him for leading me into this wonderful world of wireless networking. Dr. Sichitiu has provided me great guidance and direction for my research. He has been a profound encouragement and inspiration for my works. I am grateful that he gave me enough freedom in selecting my topics and for trusting me with my work. I thank Dr. Sichitiu for his patience with me and for being not only my advisor but also my friend. Finally, I want to add that he has been an excellent example to me as an engineer and a researcher. I also want to thank all of my committee members, Dr. Arne A. Nilsson, Dr. Do Young Eun and Dr. Rudra Dutta.

I want to express my deepest appreciation to my parents for their ceaseless patience, never-ending encouragement, and unconditional support. They have taught me to be a genuine person and to always appreciate everything in this world. I am grateful that they have taught me to never give up and to always be confident with myself. There can never be enough words to express how much I appreciate them for their trust in me, their patience and everything else they have done for me. I also want to mention that they both have been very great career role models for me. Without their support, I could have never gotten this far. I also want to thank my sister and my nephew for always being by my side and making me laugh.

I would like to thank my previous advisor Dr. Xianggen Xia at University of Delaware. Under his guidance, I have built a solid theoretical background in wireless communications, which has greatly benefited later in my research.

I would also like to thank my lab mates, Jangeun Jun, Suyoung Yoon, Manjunath Prabhu, Chen Zhao and Mahesh Aia, especially Chen Zhao, who has provided abundant help on collecting data, sharing ideas and programming in the research of mobility modeling. Thank Yuh-Ming Chiu for offering useful theoretical advice.

Lastly, I want to thank all of my friends, Shanshan Zhou, Ting Yi, Miao Yu, Peng Xu, Xingbin Wang and Peter Lee for being there for me, for helping me and for making my life in Raleigh a colorful experience.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The availability of wireless devices and the increasing ability of sharing information has stimulated the emergence of the wireless *ad hoc networks*. An ad hoc network is a self-configuring network system consisting of a collection of wireless devices. These devices are equipped with wireless transceivers that enable them to exchange information. Different from infrastructure-based networks, such as cellular networks, ad hoc networks do not rely on any fixed networking entities. This property allows a wireless ad hoc network to be set up anywhere and anytime irrespective of the existence of wired or wireless infrastructure.

As indicated in [1], the wireless transmission power falls off roughly as the fourth power of the distance. The direct communication is carried between neighbor nodes that are within range of each other. In order to reach a node out of the transmission range, one or more intermediate nodes are used to relay or forward the packet from the sender toward the receiver. Thus, part, or all, nodes in an ad hoc network are required to act as routers and relay packets for other nodes. Since it usually takes multiple transmissions for a message to reach the receiver, this kind of communication is referred to as *multi-hop* communication.

Ad hoc wireless networks have been used in several areas, such as wireless mesh networks and vehicle ad hoc networks. According to the degree of mobility of the

wireless nodes in an ad hoc network, ad hoc networks can be subdivided into two classes: stationary and mobile. In stationary ad hoc networks the position of a node does not change once it has become part of the network; in mobile ad hoc networks, nodes are allowed to move.

A lot of research has been carried out in the field of ad hoc networking technology. The research covers many topics in different types of ad hoc networks and applications. In this dissertation, we focus on two challenging topics. The first one is the localization in the stationary wireless sensor networks and the second one is mobility modeling in the mobile ad hoc networks. In the following sections of this chapter, we briefly introduce background information on wireless sensor networking, localization in stationary wireless sensor networks, mobile ad hoc networking and mobility modeling in mobile ad hoc networks.

1.1 Wireless Sensor Networks

Recent advances in low power sensing and microcontrollers enabled a new wide area monitoring paradigm commonly known as wireless sensor networking [2–4]. The emerging of this paradigm significantly improved the way we interact with the the physical world. As one of the first real world examples of pervasive computing systems [5], wireless sensor networks (WSNs) allow for inexpensive, high resolution monitoring of temperature, sound, light, pressure, motion, pollutants, etc, of large geographical areas.

A wireless sensor network usually consists of a large number of sensor nodes (stationary or mobile) deployed over an area to be monitored. Each sensor node, (also known as a 'mote'), is a self-contained, battery-powered device that is capable of sensing, communication and some level of computation and data processing. Due to the large amount usage of sensors in the network, it is important to keep each node small and inexpensive. This strictly restricts its resources in terms of energy, memory, processing speed and bandwidth. The sensor nodes operate in a distributed and collaborative manner. *Distributed* refers to two aspects:

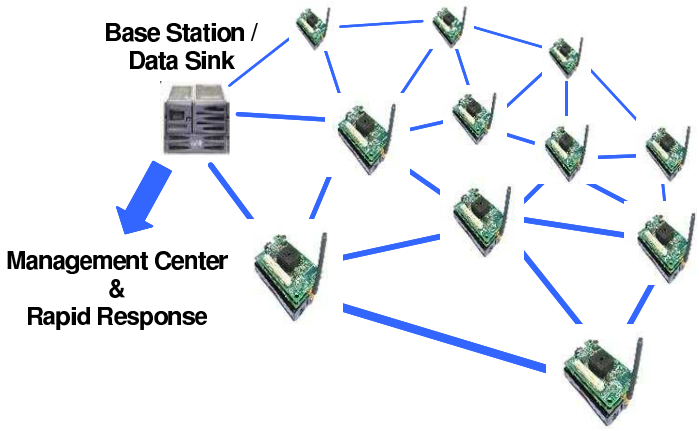


Figure 1.1: A typical wireless sensor network.

- The nodes are spatially distributed;
- The resources and the simple processing functions are spread among the nodes, instead of being aggregated at a single location, in order to reduce the communication between the nodes.

Collaborative means that the data collected from multiple nodes is sometimes exchanged among nodes and/or fused.

As mentioned before, the power required for reliable transmission over a certain distance increases dramatically with the distance, which leads to a major consumption of the energy. Due to the strict energy budget in the WSNs, the transmission ranges of the nodes are usually small. Data collected at the sensor nodes flows across the network towards the base station or data sink using multi-hop communication. The base station links a sensor network via a long-haul network (e.g., the Internet) to the top management center for further response, as shown in Fig. 1.1. The base stations generally have more resources (storage, energy) and the abilities to handle more complicated data processing.

The architecture and the function of a stationary WSN is very similar to the ancient Chinese defense system. Beacon towers sent out alarm signals (beacon fires, flags, smoke and drums) on the observations of enemy activities. The signals usually pass through several towers before reaching the higher command center. In contrast

to the ancient systems and other wireless systems, WSNs require no, or minimal human attendance. Considering of the amount of the sensor nodes utilized and the large coverage of the geographical area, as well as the deployment in some hostile environment, configuring each sensor node manually is impractical. New nodes may be added to the network to expand the network or replace the failed nodes (out of battery or broken). The network needs to be reconfigured periodically in order to keep functioning correctly. All these conditions require a WSN to self-configure. This way, the networks can be deployed with minimized or no human effort and work unattended for a long duration.

Although still in its early stage, WSNs have been used in diverse domains by taking advantages of their low cost and pervasiveness. Some WSN applications include:

- **Military monitoring**. WSNs can be deployed on the battlefield for unmanned surveillance. Sensors can detect and track the movement of vehicles and personnel. For instance, they can be used to localize a sniper.

- **Environmental monitoring**. WSNs can be deployed over a large area to monitor environmental variations. For example, the sensors can be buried beneath the earth surface, they transmit their observations of temperature, pressure, vibration and other related information to a base station located at the surface. The base station relays such observations to the research center for further analysis in order to detect any potential earthquake and locate its source. Other examples include water and air pollutant detection, glacier behavior monitoring, forest ecophysiological monitoring, etc.

- **Structural monitoring**. The sensors are deployed in buildings, bridges, ships and aircrafts to monitor the health status of their structures. By observing the structural strain response or vibration to ambient or forced stimuli [6], the presence of an existing or a potential damage can be inferred and located.

- **Habitat monitoring**. Sensors are used in the field to provide monitoring of the plants and wildlife animals. Long-term observation and tracking can be offered for the researches on the distribution and the migration pattern of any

specific wildlife object. For example, the ZebraNet [7] project at the Princeton University, attached wireless sensors to zebra collars. The data collected by the sensors allows biologists to perform researches on long-range migration, inter-species interactions and nocturnal behavior. Another interesting project, Great Duck Island project [8, 9], was performed by the University of California, Berkeley, the Intel Research Berkeley laboratory and the College of the Atlantic. Wireless sensors networks were deployed on the Great Duck Island, Maine, to monitor the microclimates in and around nesting burrows used by one particular kind of seabird. This allows the biologists to study the environmental factors that affect their nesting habits.

- **Traffic monitoring**. Wireless sensor networks installed along a highway or in a city can be used to assist traffic control in following ways: monitoring traffic volume, vehicle velocity, and road condition; detecting and locating congestion, accident, car failure and illegal driving behavior; providing assistance to drivers and helping drivers to avoid congestions and potential accidents.

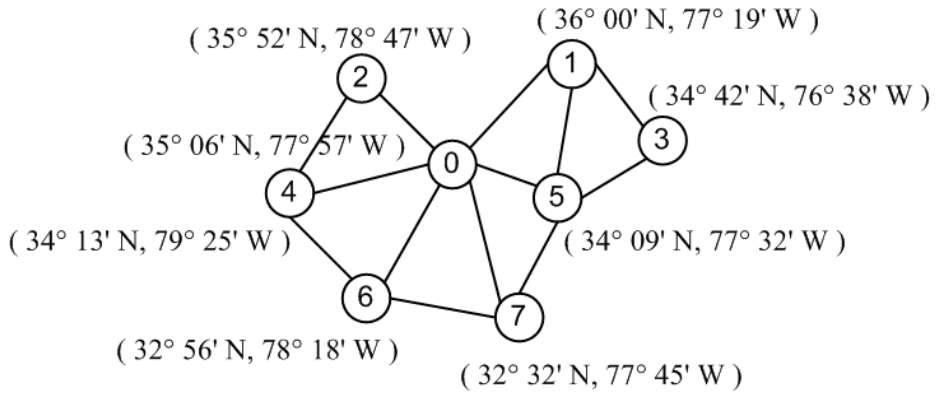
- **Emergency management**. The applications in this category usually require immediate responses from the management center. Typical applications include pre-hospital and in-hospital emergency medical care, search and rescue and disaster recovery. Wireless sensor networks deployed in the area are responsible for detecting the existence of a disaster or a victim and provide their location. The management center, based on the information reported by the sensors, can respond quickly in order to control the disaster or rescue the victim.

1.2 Localization in Stationary WSNs

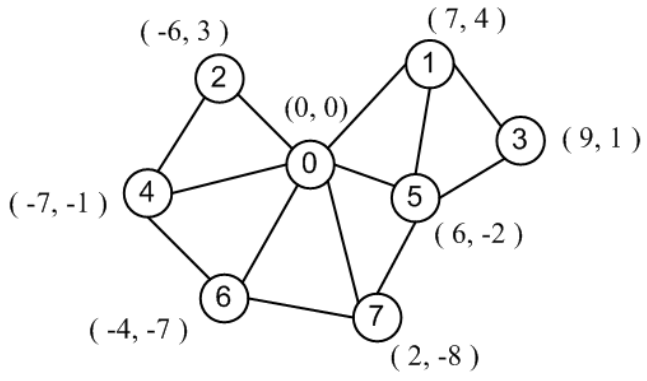
Many of the WSNs' applications mentioned in the last section either require, or benefit from a localization service that provides the exact or approximate position of each sensor node. Simply knowing *where* an event occurs is a requirement for many applications. For instance, tracking, as a classical WSN application, requires good

localization (and time synchronization). However, there are several design challenges that must be overcome for any WSNs localization algorithm to become practical. Some of these challenges can be caused by the wireless communication channel and others by the constrains of the WSN itself.

1.2.1 Localization



(a)



(b)

Figure 1.2: Absolute position vs. relative position.

Localization in WSNs is informally defined as the determination of the physical position of each sensor node. The position of the sensors can be either absolute or rel-

ative [10]. In an absolute position system, all sensors have absolute coordinates, such as latitude, longitude and altitude. In comparison, in the relative position system, a sensor's position is described by the direction, distance or coordinates with respect to another object. For example, as shown in Fig. 1.2(a), an absolute position system represents the positions of all sensors using latitude and longitude; in Fig. 1.2(b), assuming that node 0 is at origin, each sensor determines its coordinate according to its relative position with respect to node 0. Under certain condition, absolute positions can be transformed into relative positions and the other way around.

In many WSN applications, the data collected become meaningless without sensor nodes' positions. However, position information is usually not available when the network is deployed. Due to the large amount of the nodes, a wireless sensor network may be deployed by dropping nodes from a helicopter or by spreading nodes over the field by an unmanned vehicle. Such methods result in a random position distribution of the nodes over the deployment area. Therefore, special hardware or techniques need to be embedded into the sensor nodes in order to realize the localization.

Perhaps the simplest method of providing localization is to equip every sensor node with a Global Positioning System (GPS) receiver. However, this solution is unsuitable for large sensor networks due to the increasing in cost, size and energy. A GPS receiver is very expensive compared to the production cost of the sensor nodes (target prices for sensor nodes are around $1), which become a serious burden when numerous untethered sensors are required. Sensor nodes are supposed to be small for easy deployment. Embedding GPS device into a sensor would enlarge the size of the sensor significantly. Energy is another consideration because the power consumed by GPS can drain the battery very fast. In addition, GPS works efficiently only outdoors with a clear view of the sky, which prevents the use of WSNs indoors or in the presence of obstacles. Another trivial method of localization is hand-placing each sensor and manually recording its position. This is a tedious and error prone approach impractical for large sensor networks and many of the proposed WSN applications.

The most common alternative for localization involves utilizing a limited number of nodes (perhaps the base stations) that are aware of their positions (either from GPS or by manual placement). These nodes are referred to as *beacons*. The rest

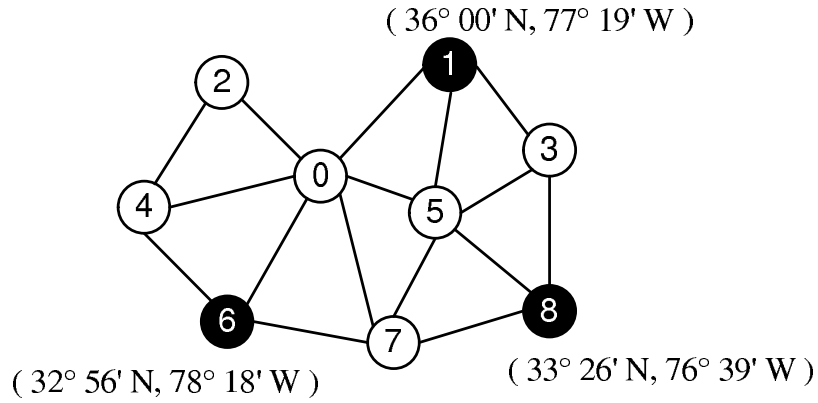


Figure 1.3: Beacons and unknowns.

of the nodes are referred to as *unknowns* and utilize beacons' positions to localize themselves. As shown in Fig. 1.3, nodes 1, 6, 8 are beacons that are aware of their geographical positions while the rest nodes are unknowns.

Depending on the measurement method, localization schemes can be classified in two categories:

- **Range free or proximity based**. This class of approaches infer constraints on the proximity to beacon nodes. These approaches, although usually attractively simple (because they do not require any additional hardware and most require only simple operations), have inherently limited precision.

- **Range based**. These approaches rely on range measurements to compute the positions of the unknown nodes. Most approaches in this category assume exact or almost exact range measurements and are shown to perform very well when this assumption is satisfied.

The majority of the existing approaches are range-based. The range measurements can be either distances or angles between sensor nodes, which are then utilized to perform trilateration (or multilateration) or triangulation to compute the positions. The range measurements are obtained indirectly by following methods:

- **Received Signal Strength (RSS)**. The power strength of a transmitted signal decreases as a function of the distance. Given the transmitter power, the path

loss model, and the received power of a signal, it is possible to estimate the distance between a transmitter and receiver pair. Using RSS to obtain distance is the easiest to implement because RSS information is available in practically all transceivers suitable for wireless networks. The main drawback of RSS-based methods is their relatively large inaccuracy compared with other methods.

- **Time of Arrival (TOA)**. The propagation time of a signal is proportional to the distance between the transmitter and the receiver. Because a slight measuring error in time can cause a large error in distance due to the high signal propagation speed, this method requires that the clocks at the transmitter and the receiver are precisely synchronized and with high resolutions. Sometimes, ultrasound with slower propagation speed is used in order to alleviate the affect of time measurement errors.

- **Time Difference of Arrival (TDOA)**. This method compares the arrival times of several received signals that are transmitted at exact same time from different transmitters. It can also use the arrival times of the same signal received at several receivers. Considering the differences in arrival times, the differences in distances between several transmitter to a receiver (or a transmitter to several receiver) can be determined. In the first case, with two transmitters at known locations, a receiver can be located onto a hyperboloid according to geometric regulations. A well-known application of such a mechanism is GPS.

- **Angle of Arrival (AOA)**. Special antennas are used to measure the angles of incident signals. This technique is used in the VOR/VORTAC aircraft navigation system. The VOR station radiates two signals periodically, one omnidirectional signal and one "beamed" signal rotating with constant speed. This allows an aircraft to determine its bearings with respect to the station. Angles are then used for triangulation to calculate positions.

Although the last three methods require specialized hardware, they can provide better accuracy in comparison to the RSS method.

1.2.2 Related Work

There has been a significant research activity stimulated by the increasing demand for accurate position information in the WSN in recent years. In this section we provide a brief literature review.

Range-free schemes

The active badge system [11] is an indoor range-free system using infrared (IR) for signaling between the sensors and the badges worn by personnel. The location of a badge can be found given the positions of the sensors. In [12], a node localizes itself at the centroid of the intersection of the transmission coverage regions of the beacon nodes. A set of connectivity constraints are built in [13], which are used to discover the location by convex optimization. In [14], an area-based range-free localization scheme is presented; the position uncertainty of an unknown is reduced by using the triangles formed by all the beacons that can be heard by that unknown.

Range-based schemes

RADAR [15] is a range-based indoor localization system that measures RSS at all positions in the entire building and records the RSS into a database during the calibration phase. In the localization phase, the location and the orientation of a user are determined by finding the best match of a set of RSS measurements in the database. The Cricket indoor localization support system [16] utilizes a combination of RF and ultrasound measurements to provide location information to users. In APS [17, 18] a range-free (the DV-hop) and two range-based (the DV-distance and Euclidean) methods are used to obtain distance estimates between unknown nodes and beacons; the distances are then used to locate the unknown nodes by trilateration. Multilateration [19, 20] (a least square scheme) can also be used to determine the positions of the unknown nodes given (potentially incompatible) distance measurements to several beacons. In [19], both signal strength and time difference of arrival (TDOA) are considered. In [21], network connectivity is used for the initial

position estimates and triangulation is used to refine the estimation. An approach for finding the *relative* position of the nodes in the absence of GPS assistance is presented in [22]. Another relative localization method is proposed in [23]: robust quadrilaterals are used to reduce the effect of flip ambiguities caused by inaccurate distance measurements. A scheme tolerant to anisotropic network scenarios [24] uses multidimensional scaling (MDS) to calculate the relative positions of the unknown nodes; the absolute positions are determined using coordinate alignment techniques. With the assistance of a moving target, a deterministic constraint-based localization approach is proposed in [25] using bounding rectangles and negative information. In [26, 27] we demonstrated the feasibility of RSS-based probabilistic localization on a small outdoor testbed with iPAQs and 802.11 wireless cards.

Directionality-based schemes using the angle of arrival (AOA) between neighbor nodes are proposed in [28–30]. In [28], the AOA measurements are exchanged between neighbor nodes, the relative AOA with respect to each beacon (even hops away) can be calculated based on geometric relations among the nodes. This enables the triangulation for the localization. Beacon nodes with rotating antennas are considered in [29]. By detecting the phase difference between the beacon signals sent out from different beacon nodes, an estimation node determines its angular bearings with respect to the beacon nodes and calculates the location according to such bearings. In [30], the localization was solved by using a semidefinite programming relaxation based method.

1.2.3 Challenges and Motivations

It is well known that all wireless transmissions suffer from the randomness, caused by unpredictability of the wireless channel. This poses serious challenges to the design of localization algorithms. Most localization algorithms for WSN rely on measuring the power strength, time, time difference or angle information between the nodes. All these measurements can be noisy, and can significantly decrease the performance of any range-based localization methods. For example, in an RSS-based localization system in Fig. 1.4(a), B_1, B_2, B_3 are beacons with known positions, if the RSSs of the

signals sent from B_1, B_2, B_3 and received at U_1 are measured correctly, they can be mapped to exact distances of d_1, d_2, d_3 that represent the distances between beacons and the unknown. Therefore, given d_1, d_2, d_3, unknown U_1 can be localized at the intersection of three circles, which have radius of d_1, d_2, d_3, respectively. However, if the RSS measurements are noisy, the estimation could deviate from its original position, or the intersection of three circles could be empty and no estimation could be obtained, as shown in Fig. 1.4(b). The discussion above also holds true for other measurements (TOA, TDOA, AOA). Thus a good localization solution to wireless sensor network has to be tolerant to the measurement errors.

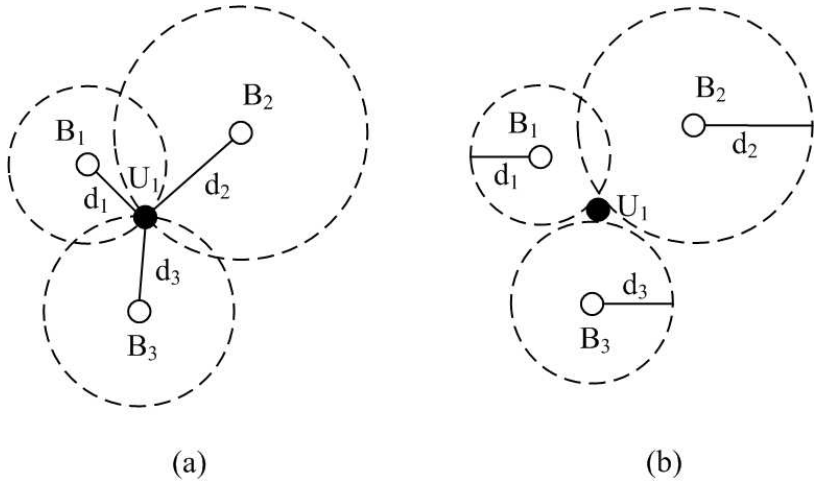


Figure 1.4: The impact of noisy measurements on localization.

In Fig. 1.4, if the distances to beacons are given, for each unknown, it has to be in the transmission range of at least three beacons to achieve localization. Otherwise, the position of that unknown is not unique. Similarly, given the angles an unknown bearing to the beacons, the minimum number of beacons required is two when the orientation of that unknown is available; and three when the orientation of that unknown is unavailable. In a wireless sensor network with many unknowns and a small number of beacons, such requirements are very hard to fulfill. Most unknowns cannot directly hear from enough beacons to perform localization. A possible alternative is to use position information from beacons that are multiple hops away.

Concluding the above discussion, our motivation is to design localization ap-

proaches that are suitable for wireless sensor networks and can achieve good accuracy and precision despite inaccurate range measurements and a small number of beacons.

1.2.4 Contribution

The first part of this dissertation is dedicated to the study of localization techniques suitable for the stationary wireless sensor networks. It mainly focuses on the development of an accurate and robust approach for the distributed localization considering range measurement errors and a small number of beacons. Based on different measurements used, the major contribution of this work can be divided into:

RSS-based probabilistic localization

RSS-based probabilistic localization proceeds in three phases [31, 32]: the first phase involves modeling the uncertainties of range measurements; in the second phase, a set of probabilistic constraints are computed and combined to produce initial position estimates; in the final phase, negative constraints are used to refine the initial estimates. In the design of this scheme, we achieved the following tasks:

- Generalized mapping model between RSS and distance.

  We introduced a generalized closed form probabilistic model which maps each RSS to a log-normal distribution of the distance. The randomness of the wireless channel has a significant impact on the performance of any range-based localization approach. This model characterizes the randomness in outdoor environments with no major obstacle. Experiments show that the model fits very well with the results obtained from real world measurements.

- Modified probabilistic localization using positive constraints.

  Based on the original probabilistic localization proposed in [33], we suggested a modified version that eliminates all redundancies and dependencies among the positive probabilistic constraints. In this way, the constraints can be combined

easily and the computation load can be reduced. To further reduce the computational complexity, which is relatively expensive in the proposed localization scheme, we propose a method that takes advantage of using fast Fourier transform (FFT) such that the computation times for localization can be decreased dramatically.

- Uncertainty reduction using negative constraints.

  Localization using only positive constraint can generate substantially large estimation errors when using information propagated from beacons far away. In order to reduce the errors and narrow down the uncertainties introduced in positive constraint localization, we introduced probabilistic negative constraints to refine the initial estimation. Negative constraint, as shown later, can significantly improve both accuracy and precision for the localization.

AOA-based probabilistic localization

AOA-based localization addresses on localization techniques based on angle information between the neighbor nodes. AOA-based method shares similarity with RSS-based method, they both utilize positive probabilistic constraints to define the position distribution of an unknown [34]. During the development of the scheme, we implemented:

- Extended localization scheme using angle measurements.

  RSS-based localization method is extended in order to use the angle measurements under the assumption that all sensor nodes are capable of detecting the arriving angle of an incident signal. We used a Gaussian model to characterize the angle measurement errors cause by both the wireless channel and the measuring device/method. Positive constraints are used to constrain the possible positions of an unknown. The proposed method can achieve much better accuracy and precision than existing angle-based approach under the condition of inaccurate measurements and insufficient beacons.

- Complexity reducing method to discover the orientations.

  When the orientations of the unknowns are unavailable, the computational time is significantly increased. We proposed a novel approach to discover the orientations by computing a position probabilistic function on each possible orientation. These functions are combined later to determine the orientation. By doing so, the computational complexity can be kept asymptotically in the same order as the localization algorithm when the orientations are known.

1.3 Mobile Ad Hoc Networks

Simply speaking, mobile ad hoc networks (MANETs) are ad hoc networks with mobile nodes, such as laptop and PDA. A typical MANET is shown in Fig. 1.5. The nodes in a MANET are allowed to move randomly and organize arbitrarily. Also, the nodes in a MANET can join and leave the network scene at anytime. Nodes' movements and activities may cause the formation of new links and the breakage of existing links, and thus cause the network topology to vary frequently and unpredictably. Therefore, in addition to being decentralized and self-configuring, a MANET should also be self-organizing and self-healing to assure robustness to variation of the network topology.

MANETs have shown significant potential in military and commercial applications. One classic application is to enabling communications among a group of soldiers on battlefields when it is impossible to rely on a fixed infrastructure. Another example is a vehicular ad hoc network (VANET), which is used to establish communication among vehicles (and between vehicles and roadside equipment) in order to provide passengers information and entertainment and to achieve smooth traffic flow on the road. In addition, MANETs can be used to provide communication for first responders. For example, when the entire communication infrastructure destroyed, a MANET can be set up quickly to transparently substitute the infrastructure-based network.

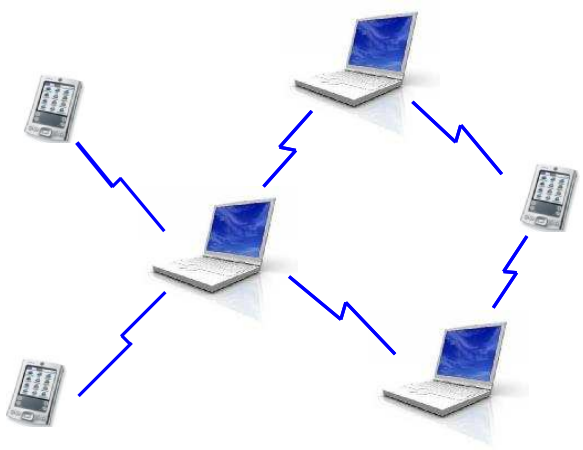


Figure 1.5: A typical mobile ad hoc network.

1.4 Mobility Modeling in MANETs

For a pair of nodes to communicate in a multi-hop communication network, they can either communicate directly if they are within transmission range of each other, or a route is established through a series of intermediate nodes otherwise. In multi-hop communication, any node on a routing path is important. Therefore, the highly dynamic network topology of the MANETs can strongly influence the performance of routing (and other networking) protocols. Understanding the nature of the mobility patterns and how they impact the performances of protocols is essential for designing practical MANET routing protocols.

1.4.1 Mobility Modeling

Due to scarcity of real MANET deployments, most evaluations on the routing protocols have been done through simulations. Simulation has the advantage of allowing for the creation of hypothetical scenarios, replicable experiments and the ability to quickly explore a large design space. Mobility modeling, as a core component in simulation, characterizes the mobility properties of the mobile devices in a network and uses such patterns to reproduce trajectories of the mobile devices.

Existing mobility models can be classified into three categories. The first one,

referred to as the *stochastic (synthetic)* model, includes the most commonly used random waypoint (RWP) model and the random direction model. They use simple statistical distribution, such as random destinations and uniformly distributed speed, to generate the nodes' trajectories. However, these models, while easy to build, are completely unrealistic, since in real world, a node's movement rarely follows simple distributions and the statistical characteristics of the nodes' movements are different from scenario to scenario. The second category, referred to as the *trace based* model, traces mobile hosts in their real world scenarios, by which the observer can gain valuable insight for actual mobility patterns and greatly enhance the mobility models. The third category, referred to as the *detailed simulator*, simulates a certain scenario in great details. for example, student movements on a campus by considering the class schedule and the location of the cafeteria, library and gym and bus movements by considering schedules and stop locations of busses. Both trace based models and detailed simulators are complex and much more accurate, however, they lack diversification convenience.

Concluding the deficiencies of both models above, an ideal mobility model used in a simulator should be both realistic and easy to diversify. In other words, a model should accurately capture the spatial and temporal dependencies of the mobile nodes, as well it can be used to produce a wide range of network scenarios with different work loads, scales and configurations.

1.4.2 Related Work

Due to their importance to the evaluation of the MANET routing protocols, a lot of mobility models have been proposed.

stochastic Models

stochastic models are based on simple assumptions regarding of the nodes' movement. A survey on stochastic models is given in [35], where the stochastic models are further divided into *entity* and *group* mobility models. In the entity models, the

movements of the mobile nodes are independent of each other, while in the group models, the nodes move in groups.

The most commonly used entity model is the random waypoint model (RWP) [36], which is derived from the famous random walk model [37]. In RWP, each node moves from a location to another by a speed uniformly distributed between predefined minimum and maximum speeds. The waypoints are uniformly distributed in the given networking domain. Before moving to another waypoint, the node pauses for a duration that is selected from a uniform distribution with predefined minimum and maximum pausing times.

Although very simple, the RWP model has several major drawbacks which can cause false and invalid evaluations. These drawbacks and the proposed methods (derived from RWP) to remedy them are listed below.

- The long term probability distributions determining the nodes' mobility converge to some steady states along with time. These steady state distributions are different from the initial state of the RWP model: First, the long term spatial distribution of the mobile nodes in RWP tend to cluster at the center of the domain [38], which is inconsistent with the assumption of uniformly distributed waypoints. In order to overcome this problem, random direction model is proposed [39, 40]. In the random direction model, a node's movement does not stop until it reaches the border of the given domain. At the border, the node pauses for a selected duration as RWP. When the pausing time is over, instead of choosing the next waypoint, a new direction is chosen and the node will move along that direction. Therefore, a uniform spatial distribution can be achieved. Second, the average node speed in RWP approaches zeros when the minimum speed is set to zero [41]. This problem can be fixed by set a non-zero minimum speed. In [42], the steady state distributions of node positions, speed and pause time are derived. Both spatial and speed inconsistencies can be solved by selecting them from the steady state distributions instead of the uniform distributions.

- The lack of correlation between two consecutive states of the speed and the

direction. The effect of this drawback is to cause sudden speed changes and sharp turns, which are unrealistic according to the usual moving behavior of a mobile nodes. In order to fix this problem, a *smooth RWP* is proposed in [43], in which it uses two stochastic principles to control the selection of the new speed and direction which are correlated to their previous values. The selection principles assure the smoother nodes' movements than the original RWP. Another model to solve this problem is the *Gauss-Markov* model [44,45]. In this model, a mobile node's speed and direction are updated at discrete time intervals. The new values are randomly chosen from normal distributions with the old values as means.

- The lack of dependency between two consecutive waypoints. Because RWP is a memoryless model, the next position of the node does not depend on the current and the past positions of that node, which is sometimes unrealistic. A *Markovian RWP* is described in [46] for rectification. The model uses a discrete Markovian process with two-dimensional state transition matrix to calculate the next waypoint. The model also allows the Markovian structure for the velocities and the pause times.

All models mentioned above are entity-based. However, in MANETs, in many situations, it is very likely that the mobile devices move in groups such as the passengers in a bus. Among all group stochastic models, the most common one is called the reference point group mobility (RPGM) model [47]. In RPGM, the movements of all individual nodes in a group are based on the movement of a logical center. The trajectory of a logical center is defined by a sequence of check points and the corresponding time intervals. All groups move continuously from one check point to the next. At each check point, the position of a individual node in a group is controlled by a random motion vector varying from check point to check point. Therefore, RPGM defines both the motions of the group and of the individual nodes. The logical centers determine not only the moving paths, but also the speed and pause duration of the groups. Other group mobility models including exponential correlated random mobility model, column mobility model, nomadic community mobility Model and pursue

mobility model are briefly discussed in [35, 48].

Trace Based Models

Compare to stochastic model, trace based mobility model captures the mobility patterns more realistically and accurately. Due to the sparse deployment of MANETs, obtaining real mobility traces and generating detailed simulation is not trivial. Existing real mobility traces can be classified into three categories according to the classes of the mobile the devices experience: persons on foot, vehicles and animals.

When targeting human beings on foot, most traces are obtained from WLAN by recording the traffic (association, disassociation and roaming messages) between the APs and the wireless devices. Owing to the increasing popularity of portable wireless devices (laptops, cell phones and PDAs) and the relative convenience of recording the network traffic at the APs, no extra effort needs to be spent to equip the mobile nodes with the necessary devices. Therefore, WLAN traces are the most popular among all traces. In [49–54], WLAN traces collected on the college campuses of Dartmouth, UNC, ETH Zurich (Switzerland), USC, MIT and UCSD, respectively, are analyzed and used to extract mobility patterns, which are later used to regenerate the mobility traces. WLAN traces for the participators at conferences can also be found in [55,56].

On the other hand, the amount of traces for vehicles and animals are very limited. In [57], each bus running in the city of Seattle, WA is equipped with a global positioning system (GPS). A sequence of time and location data of the buses are recorded. Similarly, the trajectories of the buses running on the campus and the surrounding county of UMASS, Amherst, are recorded, (the original purpose of this project is to construct a Disruption tolerant network (DTN) testbed). The Zebranet project carried at Princeton [58] collects movement traces of the zebras from two real-world deployments at Sweetwaters Game Reserve near Nanyuki, Kenya. However, as far as we know, since it is still under its test deployment stage, the mobility data collected in this trace is very limited.

The existing work most closely related to ours can be found in [52] and [59] at USC. In [52], a comprehensive study of WLAN traces to date collected from different college

campuses is provided. In [59], a time variant mobility model for the WLAN scenarios based on two similar properties observed in WLAN traces. Our work broaden the categories of traces to include both WLAN traces and bus traces. Furthermore, while [59] describes the user behavior at areas centered around access points (APs) or at building granularity, we concentrate on exploring the user behavior in the entire scenario area. Finally, we analyzed the similarities for traces from different scenarios with different configurations and thus our generator can be used to generate more diversified mobility traces.

Detailed Simulator

The models in this category simulates the mobility of the nodes in a very detailed fashion by introducing a series of constraints, such as geographic and time constraints. Two examples of the detailed simulator include the simulated mobility trace for buses running in Zurich, Switzerland can be found in [60] and TRANSIM, which provides a distributed vehicular and pedestrian traffic simulator in the transportation network of a large metropolitan area [61].

1.4.3 Motivation and Contribution

In order to generate both realistic and diversified mobility traces, one feasible solution is to extract statistical properties from real traces; and to generate synthetic mobility traces using these properties. In this way, the generated mobility patterns possess sufficient realism by maintaining the statistical similarity with the real trace. Meanwhile, if the mobility patterns found in real traces exhibit some statistical invariance, then the patterns obtained from one real trace can be used to generate synthetic traces with different configurations.

The contribution of this work can be summarized as following:

- Aiming to discover the statistical invariance in real traces, we studied several real traces including campus wireless local area network (WLAN) traces and

bus traces. We extracted the statistical properties of several features that predominantly determine the behavior of the mobile users. We showed the effects of these parameters on the performance of the routing protocol.

- The analysis showed that some inherent statistical properties of those extracted features are profoundly similar or at least following the same patterns, given different population sizes, time scales and even different scenarios. We provide evidence of such statistical similarities.

- Using the statistical patterns we extracted and the statistical similarity we observed, we further proposed a simple generalized trace based solution, which is able to generate realistic synthetic traces for different scenarios with different configurations.

1.5 Outline

The rest of this dissertation is organized as follows: Chapter 2 introduces a three-phase RSS-based localization approach for the stationary WSNs. Chapter 3 focuses on the basic angle-based localization techniques and our proposed angle-based localization approach for the stationary WSNs. In chapter 4, we present the statistical characteristics of the mobility patterns of the mobile nodes extracted from the real MANET traces, then we propose a novel generalized mobility trace generator. Chapter 5 concludes this dissertation.

Chapter 2

RSS-based Probabilistic Localization in Static WSNs

The proposed RSS-based localization approach belongs to the class of range-based schemes. In our proposed localization scheme, we assume that all sensor nodes in a WSN are stationary. Such assumption is realistic since in many sensor network applications, for example environmental monitoring, the positions of the sensor nodes do not change in their lifetimes once they are deployed. The main difference between our scheme and the existing localization schemes is that the proposed approach assumes inaccurate distance measurements and uses position information propagated multiple hops. The inaccuracies are characterized by modeling the range measurements as a set of probability density functions. These functions are used to compute probabilistic constraints that limit the nodes' positions. Probabilistic negative constraints are introduced in this chapter to further improve estimation accuracy and precision. Therefore, the proposed approach can be implemented in three phases as shown in Fig. 2.1. The detail in each phase is discussed in the following sections. The proposed approach is developed for two-dimensional spaces, but can be easily extended to three dimensions.

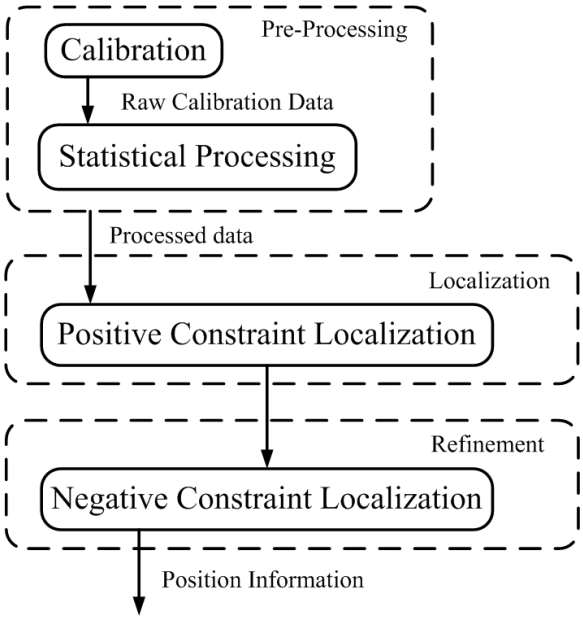


Figure 2.1: Three phase localization.

2.1 A Probabilistic Localization Approach

The inherent randomness and unpredictability properties of the wireless channel [1] lead to inaccurate range measurements. Compared to other RSS-based localization schemes that rely on accurate measurements, the proposed approach specifically assumes inaccurate measurements. Three aspects of the proposed approach are related to probabilities:

- The range measurements are modeled as a set of probability density functions (pdfs) corresponding to different RSSs.
- Each unknown node estimates its own position as a probability distribution function of the two-dimensional coordinate variable (X, Y).
- Each unknown uses both positive and negative constraints to narrow down its position uncertainty. These constraints are probability distribution functions of (X, Y) themselves.

The approach is collaborative and fully distributed. Each unknown calculates its own position relying on position information from both beacons (either neighbor or

multiple hops away) and neighbor unknowns. We separate the localization in three distinct phases in the following subsections.

2.1.1 Phase I. Calibration and Statistical Processing

During the calibration phase, the variation of the RSS is measured as a function of distance between the transmitters and receivers. This phase is usually performed only once at the initial network setup stage in the area where the WSN is deployed. For all measurement-based algorithms, calibration is a critical step for successful localization.

Assume an outdoor environment without major obstructions. Let P and D denote the random variables of RSS (in dB) and distance (in meters), respectively. Using the calibration data, it is easy to calculate the mean $\overline{P}(d)$ and the standard deviation $\sigma_P(d)$ of the RSS measurements at each distance $D = d$; $\overline{P}(d)$ and $\sigma_P(d)$ are used to generate mappings from RSS to pdf of the distance random variable D.

In general, the RSS P at any fixed distance d is normally distributed, and the variance $\sigma_P(d)$ does not vary significantly with the distance [1]. Let $\sigma_P(d) \approx \sigma_P$, $\forall d$, then

$$P[dB] = \overline{P}(d)[dB] + X_{\sigma_P}[dB], \tag{2.1}$$

with

$$\overline{P}(d)[dB] = P_0[dB] - 10n \lg(\frac{d}{d_0}), \tag{2.2}$$

where n is the path loss exponent, and the random variable X_{σ_P} models shadowing effects and has a Gaussian distribution with zero mean and standard deviation σ_P. Both n and σ_P are dependent on the physical environment and can be determined from the calibration data; d_0 is the close-in reference distance, and P_0 is the RSS at d_0. Without loss of generality [1], we assume $d_0 = 1$m. If no randomness exists in (2.1), namely, when $X_{\sigma_P} = 0$, an RSS of $P = p$ can be uniquely mapped to a distance $\overline{d}$ with a probability of 1,

$$p \rightarrow Prob\{\overline{d} = 10^{\frac{P_0 - p}{10n}}\} = 1. \tag{2.3}$$

However, in reality, shadowing effects always exist. Based on the log-normal model (2.1), given a fixed RSS p, the distance D is log-normally distributed. Thereby, each RSS can be mapped uniquely to a pdf of D.

$$p \rightarrow \lg D \sim N(\mu_D(p), \sigma_D(p)), \tag{2.4}$$

$$\mu_D(p) = \lg \overline{d} + 2\sigma_D^2 \ln 10, \tag{2.5}$$

and

$$\sigma_D(p) = \sigma_D = \frac{\sigma_P}{10n}, \tag{2.6}$$

where $\overline{d}$ is the same as shown in (2.3), and $\sigma_D(p)$ is constant. Therefore, the mapping between RSS and pdf of the distance can be generated as shown in (2.4).

For environments with significant obstructions and severe multipath fading, two facts have to be considered:

- The difference between the mean of the RSS $\overline{P}(d)$ (obtained from calibration measurements) at each distance and its theoretical value calculated by (2.2) can be large and random.
- The standard deviation of RSS $\sigma_P(d)$ varies with the distance as well.

The mapping from RSSI to distance pdf obtained above is not accurate enough to precisely characterize the randomness of these two facts. Therefore, a statistical substitution to increase the accuracy is discussed below.

Given a fixed RSS $P = p$, using $\overline{P}(d)$ and $\sigma_P(d)$, the conditional pdf of D can be approximated as:

$$f_D(d|p) = \frac{\xi(d|p)}{\int_0^\infty \xi(d|p)dd}, \tag{2.7}$$

and

$$\xi(d|p) = \frac{1}{\sqrt{2\pi}\sigma_P(d)} e^{-\frac{(p-\overline{P}(d))^2}{2\sigma_P^2(d)}}. \tag{2.8}$$

From our outdoor calibration data, we found that, given an RSS measurement,

the pdf of D can be mapped log-normally in the same format as (2.4) but with the mean $\mu_D(p) = E(\lg d|p)$ and the standard deviation $\sigma_D(p) = \sqrt{Var(\lg d|p)}$ of the variable D's logarithm. In contrast to (2.5) and (2.6), both the mean and the standard deviation are functions of RSS and obtained statistically.

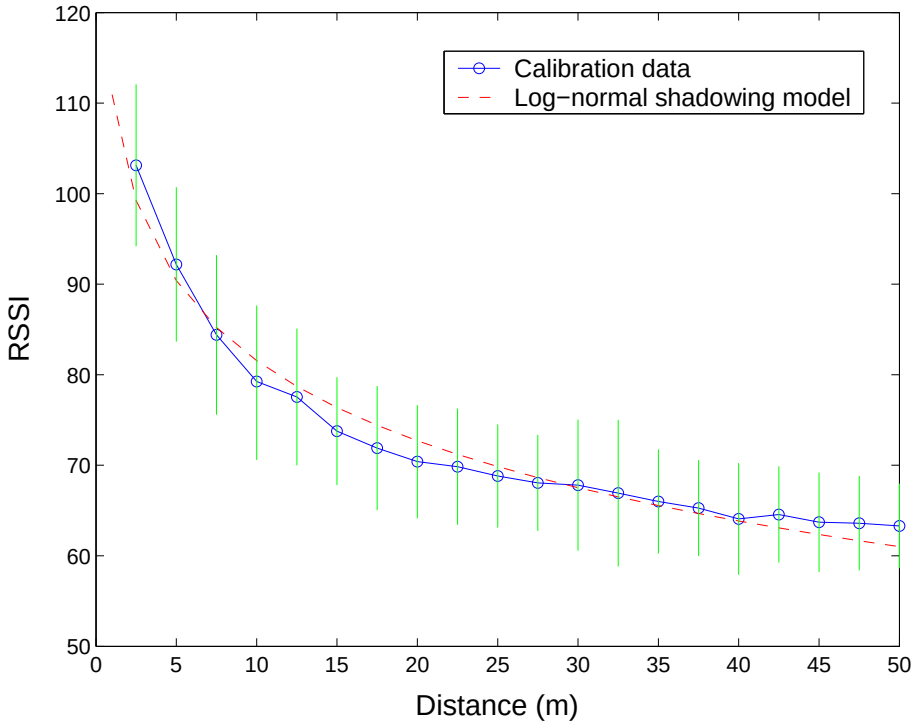


Figure 2.2: Statistical mean and three times the standard deviation of the calibrated data together with a log-normal fit.

Fig. 2.2 shows the results for the calibration data in one of our outdoor experiments [26]. In this experiment, Compaq 3870 iPAQs with Lucent Orinoco cards are used as sensor nodes. Received Signal Strength Indicator (RSSI) data is measured every 2.5m up to 50m. RSSI is an arbitrary integer value corresponding to the power strength of the received packets measured by the wireless card. The mean and three times the standard deviation of the calibrated RSSI at each distance are shown in Fig. 2.2. A log-normal shadowing model curve with $n = 2.94$ is also plotted to fit the calibration data. The exponent n is computed using logarithmic least square fitting. The RSS mean does not follow the fitting curve exactly, and the variance fluctuates

as a function of the distance.

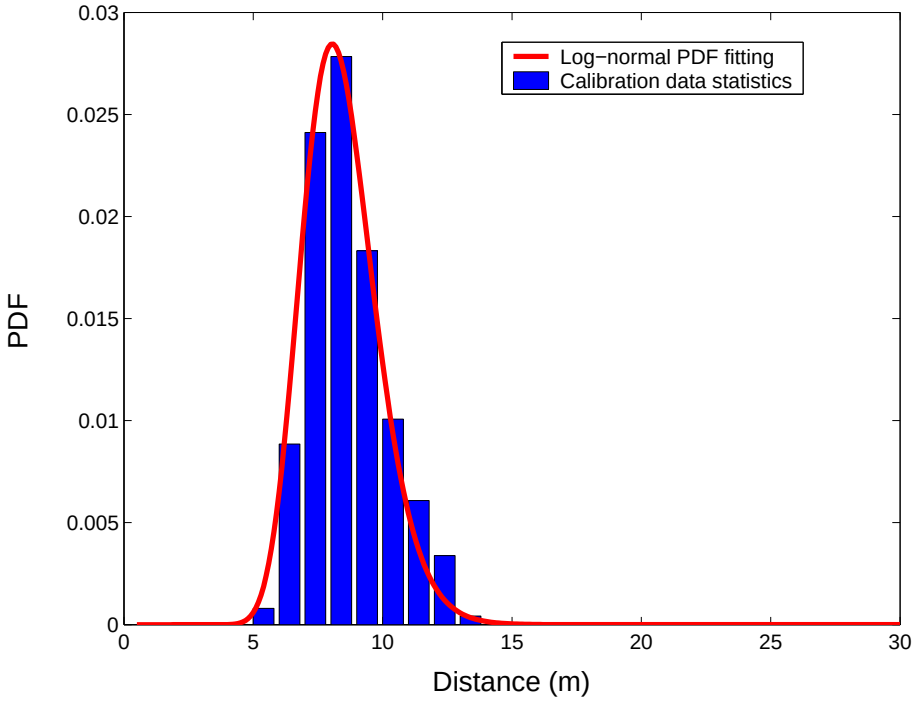


Figure 2.3: Log-normal distribution of distances for packets with RSSI=83.

Figures 2.3 and 2.4 show the mapping results from RSSI to pdfs of the distance using calibration measurements shown in Fig. 2.2. Figure 2.3 depicts the distance distribution for RSSI=83. A log-normal mapping curve with mean $\mu_D(RSSI = 83)$ and standard deviation $\sigma_D(RSSI = 83)$ is also plotted. The mean and standard deviation are computed using the statistical method described above. The log-normal mapping curve fits the bars well, which indicates that the distance for RSSI=83 has indeed a log-normal distribution as predicted. In Fig. 2.4, the log-normal fitting curves for mappings from RSSI=70 to 90 are plotted. During the localization phase, these fitting curves are used to impose constraints on the position estimations of unknown nodes.

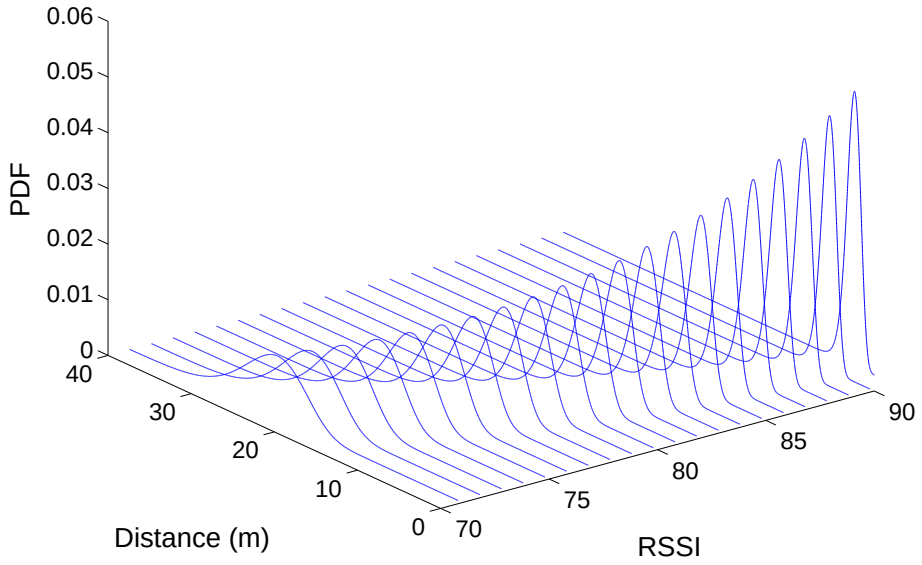


Figure 2.4: The pdfs of distances for different RSSIs.

2.1.2 Phase II. Localization with Positive Constraints

In this phase, each unknown estimates its position probability distribution function, utilizing the log-normal mappings obtained in the first phase. Initially, each unknown sets its initial estimation to a uniform distribution over the entire network area, $f_{X,Y}(x, y) = \frac{1}{A}$ where A is the total area of the network. We define another term, *pseudo-beacon*, to refer to an unknown with an estimated position probability function. Nodes with position information, including both beacons and pseudo-beacons, send out beacon packets to their neighbors. Upon receiving a beacon packet, an unknown executes the following algorithm:

- it measures the RSS of the received beacon packet;
- it maps the RSS to a one-dimensional pdf obtained in phase I and generates a probability constraint, which is a function of the coordinate random variable;
- it updates the old probabilistic distribution and pdf estimates by intersecting it with the generated constraint; and

- finally, the unknown with the updated pdf estimation broadcasts to all its neighbors.

Assume an unknown j is within another node i's transmission range and received a beacon packet with $RSS = p_{ij}$dB. Unknown j maps p_{ij} to a pdf of the distance with mean $\mu_D(p_{ij})$ and standard deviation $\sigma_D(p_{ij})$ as shown in (2.4); both $\mu_D(p_{ij})$ and $\sigma_D(p_{ij})$ are calculated during the first phase. The unknown j then calculates a pdf estimate of its position using the mapped pdf and i's position information as:

$$f_{X_j,Y_j}(x,y) = \int_{y_{min}}^{y_{max}} \int_{x_{min}}^{x_{max}} f_{X_{ij},Y_{ij}}(x-x_i, y-y_i) f_{X_i,Y_i}(x_i,y_i) dx_i dy_i, \tag{2.9}$$

$$f_{X_{ij},Y_{ij}}(x,y) = \frac{1}{\sqrt{2\pi}\sigma_D^2(p_{ij}) d_{ij}^2(x,y)} e^{-\frac{(\lg d_{ij}(x,y) - \mu_D(p_{ij}))^2}{2\sigma_D^2(p_{ij})}}, \tag{2.10}$$

and

$$d_{ij}(x,y) = \sqrt{x^2+y^2}. \tag{2.11}$$

where the constants $x_{min}, x_{max}, y_{min}$ and y_{max} are the bounding coordinates of the network. Node i can be either a beacon or an pseudo-beacon. Function $f_{X_i,Y_i}(x_i,y_i)$ is i's position pdf estimate, which is a function of the coordinate variable (X_i, Y_i). Specifically, if node i is a beacon and placed at (x_b, y_b) (here we assume the position of the beacon is accurate, although the scheme may take into account any inaccuracies of the beacon's position), the pdf estimate of beacon i can be represented as a two-dimensional Dirac's delta function,

$$f_{X_i,Y_i}(x_i,y_i) = \delta^2(x_i - x_b, y_i - y_b). \tag{2.12}$$

After computing $f_{X_j,Y_j}(x,y)$ using i's information, node j calculates its probability distribution of the position $P_j(x,y) = P_j(X_j = x, Y_j = y)$, which represents the probability of unknown j being placed at coordinate (x,y), by:

$$P_j(x,y) = \int_{y-\Delta y}^{y} \int_{x-\Delta x}^{x} f_{X_j,Y_j}(x_j, y_j) dx_j dy_j, \tag{2.13}$$

with $x_{min} \leq x \leq x_{max}$ and $y_{min} \leq y \leq y_{max}$. Both Δx and Δy are arbitrarily small numbers. The double integral can be simplified using numerical approaches.

If unknown j's original probability distribution estimation is $P(x,y) = P(X = x, Y = y)$ (either the initial uniform distribution or an updated one), assuming that (X,Y) and (X_j,Y_j) are independent, unknown j's probability distribution of the position can be updated by intersecting the newly computed and the original probability distribution:

$$P(x,y) = P(X = x, Y = y)P_j(X_j = x, Y_j = y) = P(x,y)P_j(x,y), \tag{2.14}$$

and the updated normalized position pdf is:

$$f_{X,Y}(x,y) = \frac{P(x,y)}{\sum_{y_{min}}^{y_{max}} \sum_{x_{min}}^{x_{max}} P(x,y)} \cdot \frac{1}{\Delta x \Delta y}, \tag{2.15}$$

which is broadcasted to j's neighbors. If there are other neighbors to unknown j in addition to i send beacon packets to j, for each neighbor, one probability distribution constraint is computed and intersected with the current $P(x,y)$ same as (2.14).

Example

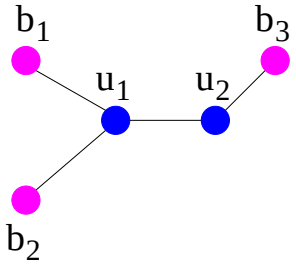


Figure 2.5: Network topology with three beacons (b_1, b_2, b_3) and two unknowns (u_1, u_2).

Figure 3.3 depicts a network with three beacons (b_1, b_2, b_3) and two unknowns (u_1, u_2). Assume the following sequence of events:

- Unknown u_1 uses the packets received from b_1 and b_2 to compute two constraints, which are intersected with each other to generate a new constraint. The new constraint is then intersected with the initial estimation (uniform distribution) to obtain a new position distribution estimation as shown in Fig. 2.6(a).

- Unknown u_2 computes a probability distribution estimate of its position as shown in Fig. 2.6(b) using the beacon packet from b_3.

- Both u_1 and u_2 become pseudo-beacon and send out beacon packets to their neighbors. When receiving the beacon packet from u_2, u_1 calculates a probability constraint as shown in Fig. 2.6(c).

- Node u_1 intersects the probabilistic constraint in Fig. 2.6(c) with its old position estimation in Fig. 2.6(a) to update its estimation. The result of the intersection is shown in Fig. 2.6(e).

- The same sequence is carried for u_2 (i.e., it computes the constraint and intersects it with the old position estimate). Using the beacon packet sent by u_1, u_2 computes the probabilistic constraint shown in Fig. 2.6(d).

- Finally, u_2 intersects the constraint in Fig. 2.6(d) with its old position distribution estimation in Fig. 2.6(b); the new position probability distribution estimation of u_2 is shown in Fig. 2.6(f).

Intersection between the current position distribution estimation and the new constraint is only possible under the assumption of two vector random variables (X_j, Y_j) and (X, Y) in equation (2.14) being mutually independent. We will discuss how to combine the probabilities when dependencies occur (the realistic case) in section 2.2.2.

Therefore, the position of the unknown is determined at the location with the largest probability in a maximum likelihood fashion, Consequently, Node j locates itself at:

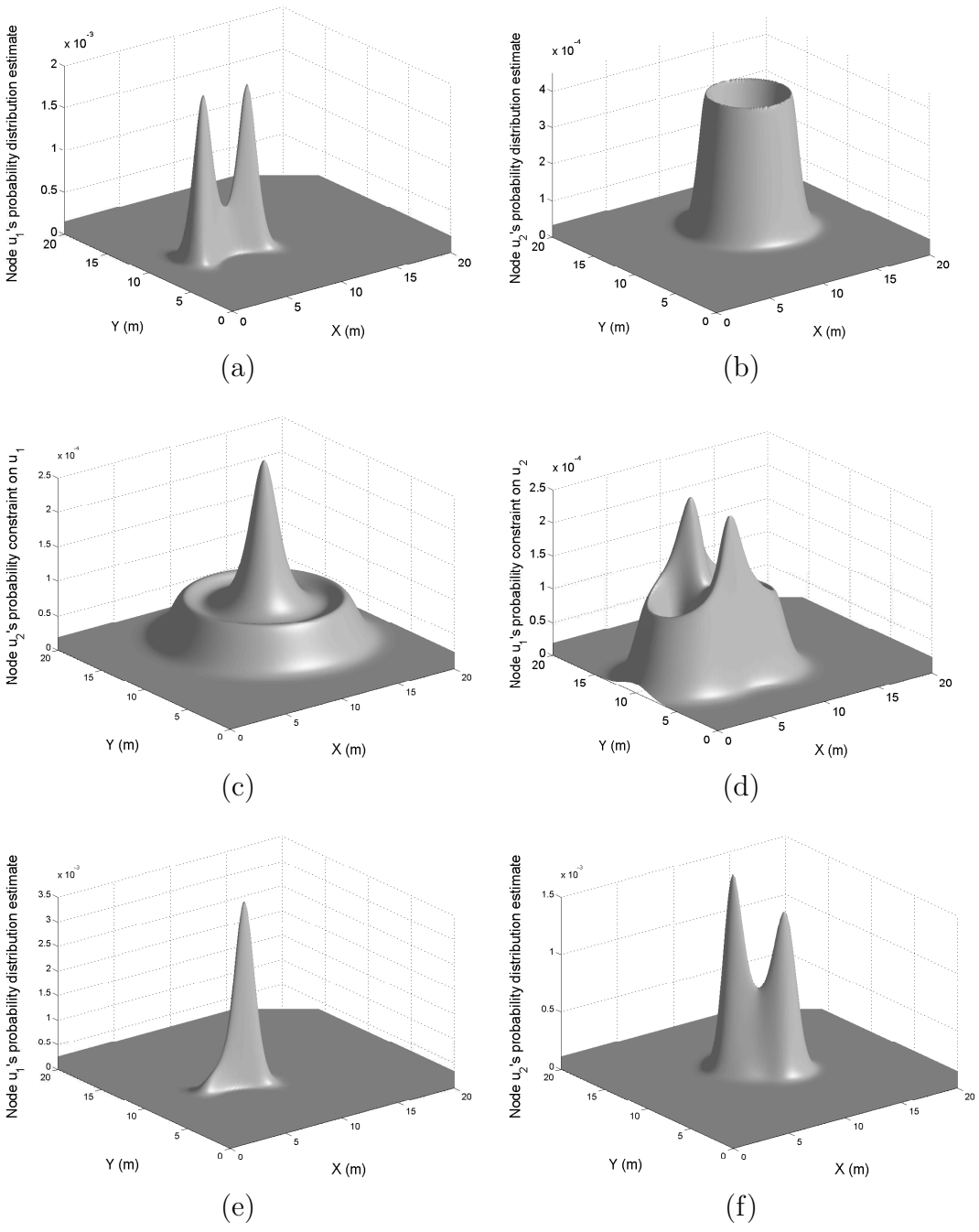


Figure 2.6: Localization evolution for the network topology in Fig. 3.3.

$$(x^*, y^*) = \arg\max_{x,y} P(x, y), \tag{2.16}$$

where $x_{min} \leq x \leq x_{max}$ and $y_{min} \leq y \leq y_{max}$.

In the above example, as shown in Fig. 2.6(e) and Fig. 2.6(f), respectively, while the probability estimation for u_1 has only one peak, which is at the true position of u_1, the probability estimation for u_2 has two peaks, with the true position on the lower peak. To improve estimation accuracy and reduce uncertainty, negative constraints are introduced in the next section.

2.1.3 Phase III. Localization with Negative Constraints

In this phase, we consider probabilistic negative constraints. In this enhancement, each unknown refines its probability estimation using negative constraints. The negative constraint is defined as a function $\rho(x, y)$ representing a constraint on an unknown when it cannot receive any packet from a given beacon. In other words, it is the information inferred by *not* hearing from a beacon.

Assume for the moment there is no shadowing effect and the minimum acceptable RSS is P_{min} (i.e., packets with RSS smaller than P_{min} will not be received); if beacon b is at (x_b, y_b), an unknown j that is aware of the position of this beacon but unable to receive any of its packets can compute a negative constraint

$$\rho(x, y) = \rho(X = x, Y = y) \tag{2.17}$$

$$= \begin{cases} 0 & \text{if } d_{b,j} \leq d_{max} \\ 1 & \text{otherwise} \end{cases}, \tag{2.18}$$

where $d_{b,j}$ is calculated as shown in (2.11) and

$$d_{max} = 10^{\frac{P_0 - P_{min}}{10n}}. \tag{2.19}$$

Considering the shadowing effects (realistic case), the negative constraint can be calculated from calibration measurements. Recall that the RSS (in dB) at each distance is normally distributed as shown in (2.1). The negative constraint is then

$$\rho(x,y) = \int_{-\infty}^{P_{min}} \frac{1}{\sqrt{2\pi}\sigma_P(d_{b,j})} e^{-\frac{(p-\overline{P}(d_{b,j}))^2}{2\sigma_P^2(d_{b,j})}} dp. \tag{2.20}$$

Thus, the probability estimate of unknown j is represented by the following probability:

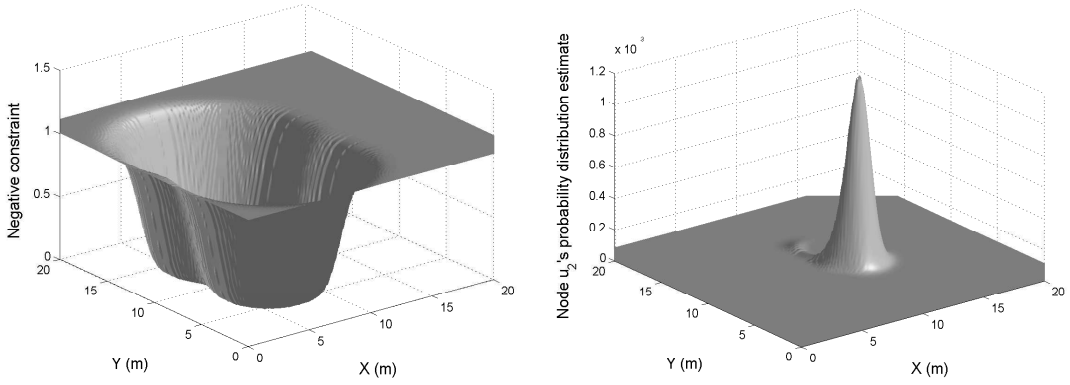
$$\widetilde{P}(x,y) = \frac{P(x,y)\rho(x,y)}{\sum_{x_{min}}^{x_{max}} \sum_{y_{min}}^{y_{max}} P(x,y)\rho(x,y)} \cdot \frac{1}{\Delta x \Delta y}. \tag{2.21}$$

Example

Using the same network topology in Fig. 3.3, node u_2 can neither hear directly from b_1 nor b_2. The intersection of b_1's and b_2's negative constraints is shown in Fig. 2.7(a). The final probability estimation for u_2 after intersecting the old one with the negative constraint is shown in Fig. 2.7(b). As shown in (2.16), the final position estimation of the unknown j corresponds to the coordinates with the maximum probability. For u_2's estimation, only one peak at the true position remains. In this case, the negative constraint refinement clearly helps to reduce the estimation uncertainty and improve the estimation accuracy compared to Fig. 2.6(f). We will evaluate the usefulness of negative constraints for more general settings in Section 2.3.

2.1.4 Reducing the Computational Complexity

In order to implement the proposed approach, there are a number of ways to store and calculate the position estimates. Here, we use a uniform rectangular grid. The pdf estimations in the last section are replaced by probability mass functions (pmfs) estimation as the probability for an unknown being located in each square of the grid. Equation (2.9) become



(a) (b)

Figure 2.7: Refinement using negative constraints. (a) The intersection of b_1's and b_2's negative constraint; (b) Final pdf estimation of u_2 after combining the negative constraints in (a) with the positive constraint estimate in Fig. 2.6(f).

$$f_{X_j,Y_j}(x,y) = \sum_{y_{min}}^{y_{max}} \sum_{x_{min}}^{x_{max}} f_{X_{ij},Y_{ij}}(x - x_i, y - y_i) f_{X_i,Y_i}(x_i, y_i) dx_i dy_i. \tag{2.22}$$

Both (X_{ij}, Y_{ij}) and (X_i, Y_i) are now discrete coordinate random variables. Equation (2.22) indicate that the proposed probabilistic approach can be computationally expensive. Wireless sensors usually have limited computational resources: a mica2 mote can compute and intersect two constraints in 15 seconds for a 100×100 grid [27]. It can be expected that the entire localization approach will take a few minutes; for most applications expecting months to years of service from the sensor network, this is a relatively small price to pay for accurate localization information.

When approximating pdfs using rectangular grids there is a clear tradeoff between precision and computational complexity: the finer the grid, the higher the precision and computational demands. If we assume the network deployment area is square and it is divided into an $N \times N$ uniform grid, the calculation for the pmf estimate for each unknown requires $O(N^4)$ multiplication per operation.

In order to reduce the computational complexity, we propose to use the Fast Fourier Transform (FFT), which is commonly used in digital signal processing. Notice

equation (2.22) is actually a two-dimensional discrete convolution:

$$f_{X_j,Y_j}(x,y) = f_{X_{ij},Y_{ij}}(x,y) * * f_{X_i,Y_i}(x,y). \tag{2.23}$$

Therefore, $f_{X_j,Y_j}(x,y)$ can be calculated using the two-dimensional FFT and inverse FFT (IFFT):

$$f_{X_j,Y_j}(x,y) = \mathbf{IFFT}\{\mathbf{FFT}\{f_{X_{ij},Y_{ij}}(x,y)\} \times \mathbf{FFT}\{f_{X_i,Y_i}(x,y)\}\}. \tag{2.24}$$

where $**$ stands for the two-dimensional convolution. In this way, the number of multiplications required is reduced to $O(N^2 \log_2 N)$.

The processing time can be further reduced if each node only computes constraints for the locations where its current position estimate is not zero - a very simple enhancement that can substantially reduce the computing time.

2.2 Algorithm Design and Implementation

In this section, we provide the details regarding the design of the algorithm that implements the ideas in Section 2.1.

2.2.1 Packet and Log Format

The only nodes providing accurate position information are beacons, and all unknowns estimate their position based on this information. In the proposed algorithm, each unknown stores a log as shown in Table 3.1. All entries start with the length L_i of that row and followed by a beacon identifier (ID_{bi}) and the coordinates (x_{bi}, y_{bi}) of that beacon. Each entry represents a transmission path from the beacon to the unknown node storing the log. ID_{ui}s are the IDs of the intermediate and destination unknowns. A beacon can start multiple entries, as beacon b_1 in the first two entries

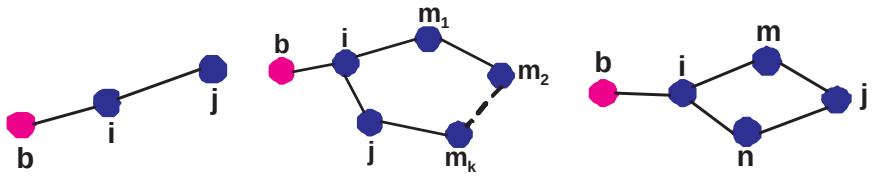
shown in the table, since all of the neighboring unknowns of the beacon relay its position information. Similarly, a single unknown can appear in multiple entries, as unknown u_3 in row 2 and 3. Upon the receipt of a log entry from a neighbor node (beacon or pseudo-beacon), each unknown appends its node ID to the log entry and records the RSS of the received packet. Beacon packets from an unknown contain one or more entries from that unknown's log. Packets from beacon nodes follow the same format (having only the beacon information).

Table 2.1: Beacon packet format for RSS-based localization.

L_1	ID_{b1}	x_{b1}	y_{b1}	ID_{u1}	RSS_{b1u1}	ID_{u2}	RSS_{u1u2}	$\cdots$
L_2	ID_{b1}	x_{b1}	y_{b1}	ID_{u3}	RSS_{b1u3}	ID_{u4}	RSS_{u3u4}	$\cdots$
L_3	ID_{b2}	x_{b2}	y_{b2}	ID_{u3}	RSS_{b2u3}	ID_{u5}	RSS_{u3u5}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

2.2.2 Dependency Elimination

The log processing involves eliminating dependencies among different log file entries. Recall that when vector random variables (X_j, Y_j) and (X, Y) in equation (2.14) are dependent, simple intersection between the position distribution estimate and the constraint cannot be performed. We classify the dependencies in two categories:



(a) (b) (c)

Figure 2.8: Network topologies with dependencies (node b is a beacon, all of the others are unknowns): (a) i to i dependency case 1; (b) i to i dependency case 2; (c) common i constraint dependency.

i to i dependency

Two typical cases of i to i dependency are shown in Figures 2.8(a) and (b).

In Fig. 2.8(a), after computing its position distribution estimation $P(x, y) = P(X = x, Y = y)$ using packets from b, node i sends a beacon packet to unknown j. Upon receiving the beacon packet, unknown j will recalculate its pdf estimate and broadcast the updates back to i. The new probability constraint calculated by i using the beacon packet sent by j is $P_i(x, y) = P_i(X_i = x, Y_i = y)$. It can be shown that (X, Y) is dependent on (X_i, Y_i).

Fig. 2.8(b) shows another case with the same underlying dependency. Assume several unknowns $(m_1, m_2, ..., m_k)$ are on a path between unknowns i and j. Assume that node i propagates a beacon packet on the path $i \rightarrow m_1 \rightarrow m_2 \rightarrow \ldots \rightarrow m_k \rightarrow j \rightarrow i$. Eventually, i will use the beacon packet from j to calculate constraint $P_i(x, y)$. As in the case of Fig. 2.8(a), (X_i, Y_i) and (X, Y) are dependent.

In summary, the i to i dependency occurs in a path (log entry) where one unknown occurs more than once.

Common i constraint dependency

Fig. 2.8(c) shows a common i constraint dependency scenario. Both m and n update their pdfs using beacon packets from unknown i and broadcast their updates to unknown j. Assume the beacon packet from m arrives first, then j updates its position distribution to $P(x, y)$. Next, j will use the beacon packet from n to calculate the probability constraint $P_j(x, y)$. Nevertheless, (X_m, Y_m) and (X_n, Y_n) are dependent, which results in the dependency between (X_j, Y_j) and (X, Y).

In the common i constraint dependency scenario, two or more paths share one or more intermediate unknowns.

From the discussion above, if each node appears only once in a single row of the log file, no i to i dependency exists. Therefore, the first kind of dependency can be eliminated by deleting all of the rows, in each of which a node has been recorded multiple times.

The common i dependency is eliminated by searching for all of the rows that contain the same intermediate unknown with different descendent unknowns. Among all of these rows, the longer paths (in terms of hops) are deleted first. Among all remaining shortest paths, only one (randomly chosen) for every intermediate unknown is kept; others are discarded. In this way, the dependencies are eliminated.

After eliminating the dependency, the position estimate can be computed by intersection. Each unknown uses the following pseudo code LOC(log) to calculate its position distribution and pdf. The input is the stored log and the ID of the estimation unknown.

Algorithm 1 LOC(log,j)

$f(x,y) \leftarrow \frac{1}{A}$
$neighbors \leftarrow j$'s neighbors in the log
for each $i \in neighbors$ **do**
  if $f_{X_i,Y_i}(x,y)$ exists **then**
    $p_{ij} \leftarrow$ RSS from i to j stored in the log
    Map p_{ij} to $\mu_D(p_{ij})$ and $\sigma_D(p_{ij})$
    Calculate $f_{X_{ij},Y_{ij}}(x,y)$
    $f_{X_j,Y_j}(x,y) \leftarrow f_{X_{ij},Y_{ij}}(x,y) * * f_{X_i,Y_i}(x,y)$
    Calculate $P_j(x,y)$
    $P(x,y) \leftarrow P(x,y)P_j(x,y)$
  else
    LOC(log,i)
  end if
end for
for each beacon recorded in log but not directly communicating with j **do**
  Calculating negative constraint $\rho(x,y)$
  $P(x,y) \leftarrow P(x,y)\rho(x,y)$
end for
update $f_{X,Y}(x,y)$ using $P(x,y)$
return $P(x,y)$,$f_{X,Y}(x,y)$;

2.2.3 Waiting Period

In order to reduce the traffic overhead, a waiting time can be inserted between the receiving of a beacon packet and the broadcasting of the update. Consider the

scenario shown in Fig. 2.9 where both m and n are in the transmission range of beacon b and reachable by unknown i. Node i necessarily receives the beacon packets from m and n consecutively. In order for i to avoid sending two packets (one due to the packet received from m and another one for the packet from n), node i should wait for a time T before it broadcasts. Thus, it can *aggregate* the information from both node m and n before it sends the packet to node j. The time T should be chosen sufficiently large in order to allow all intermediate nodes to send their beacons.

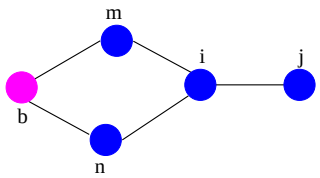


Figure 2.9: Node i aggregates the beacon packets from unknowns m and n before broadcasting to unknown j.

The algorithm is repeated distributively at every unknown; eventually, data from all beacon nodes will reach all of the unknown nodes, and the algorithm will stop. Alternatively, to reduce the number of messages, we can impose a lower threshold on the difference that has to occur in the position of an unknown for that node to forward a beacon packet: information from beacons after it travels many hops is often too diluted (i.e., the constraints do not constrain too much) to make any difference in the position of a node. Therefore, the compute procedure at an unknown will stop after a relatively stable estimation is achieved.

2.3 Simulation Results

Meaningful evaluation of the proposed approach is not trivial. We showed the practical feasibility of other (single hop and not considering negative information) probabilistic approaches [26, 27]. However, a practical experiment is limited with respect to the number of nodes and scenarios that can be considered.

On the other hand, current network simulators (e.g., ns-2, Qualnet, OPNET), do

not consider realistic physical layer effects like the (unintended) directionality of the antennas and variability in the power of each radio.

Therefore, we decided to use a custom simulator, implemented in Matlab and using calibration data based on real world measurements. The results consider the precision (i.e., the uncertainty in the position estimates) and the accuracy (i.e., the difference between the real position of the nodes and the position determined by the localization algorithm) of the proposed approach. We compare the results with other well-known, range-based localization approaches (DV-distance [18] and multilateration [19, 20]). The precision of our approach is also compared with the Cramer-Rao lower bound (CRLB).

2.3.1 Cramer-Rao Lower Bound

The CRLB is a lower bound which can be used to evaluate the variance of any unbiased estimator [62]. Assuming that there are M beacons and N unknowns in the network, let $\theta = [x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N]$ be the coordinate variables to be estimated. The likelihood function is

$$f(\lg \mathbf{d}; \theta) = \prod_{j=1}^{M+N} \prod_{\substack{i \in H(j) \\ i<j}} \frac{e^{-\frac{(\lg d_{ij} - \lg \bar{d}_{ij})^2}{2\sigma_D^2}}}{\sqrt{2\pi}\sigma_D}, \tag{2.25}$$

where $i \in H(j)$ means node i is within j's transmission range. The CRLB can be calculated based on $2N \times 2N$ Fisher Information matrix $I(\theta)$ [63, 64]:

$$[I(\theta)]_{k,j} = \begin{cases} \sum_{i \in H(j)} \beta \frac{(x_i - x_j)^2}{\bar{d}_{ij}^4} & k = j \\ \beta \frac{(x_k - x_j)^2}{\bar{d}_{ij}^4} & k \neq j, k \in H(j) \\ 0 & \text{otherwise} \end{cases} \tag{2.26}$$

if $1 \leq k \leq N$ and $1 \leq j \leq N$,

$$[I(\theta)]_{k,j} = \begin{cases} \sum_{i \in H(j)} \beta \frac{(x_i - x_j)(y_i - y_j)}{\overline{d}_{ij}^4} & k = j \\ \beta \frac{(x_k - x_j)(y_k - y_j)}{\overline{d}_{ij}^4} & k \neq j, k \in H(j) \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

if $1 \leq k \leq N$ and $N+1 \leq j \leq 2N$, or $1 \leq j \leq N$ and $N+1 \leq k \leq 2N$,

$$[I(\theta)]_{k,j} = \begin{cases} \sum_{i \in H(j)} \beta \frac{(y_i - y_j)^2}{\overline{d}_{ij}^4} & k = j \\ \beta \frac{(y_k - y_j)^2}{\overline{d}_{ij}^4} & k \neq j, k \in H(j) \\ 0 & \text{otherwise} \end{cases} \tag{2.28}$$

if $N+1 \leq k \leq 2N$ and $N+1 \leq j \leq 2N$. The constant β is inversely proportional to σ_D, and $\overline{d}_{ij}$ is the distance between node i and node j. Therefore, variance of each estimated parameter is bounded from below:

$$\sigma_{\theta_i}^2 \geq [I^{-1}(\theta)]_{i,i}. \tag{2.29}$$

2.3.2 Performance Evaluation

For the simulation, we deployed the sensor nodes randomly in a square area (of variable size depending on the desired density). We used a rectangular grid with squares of 1m^2 and imposed a threshold on the difference in position of 1m (i.e., unless the current estimated position is different by more than 1m from the previous position, the node does not rebroadcast the new information). The transmission radius was restricted to 13.5m (corresponding to an RSSI measurement of 78). We evaluated the effect of the variation of several parameters (number of hops the information is allowed to travel, density, number of unknowns, fraction of beacons and measurement inaccuracy), on the accuracy and the precision of the proposed ap-

proach. Unless otherwise specified, we used 6 beacons, 50 unknowns, a node density of 0.03 nodes/m^2 and a standard deviation for the RSS of 2.5 dB (corresponding to the value obtained from calibration measurements). For every graph, we present the average of 20 simulations with different network topologies. All of the results are normalized with respect to the transmission range.

The Effect of the Number of Hops between Beacons and Unknowns

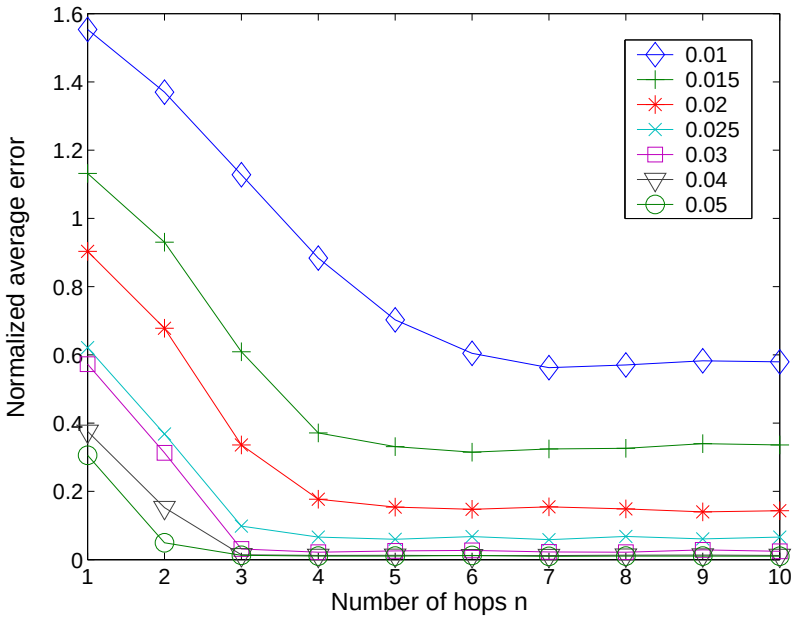


Figure 2.10: Accuracy of the proposed approach as a function of the number of hops from the beacons to the unknown nodes for different node densities.

To evaluate the number of hops after which beacon information becomes irrelevant, we simulated the network and varied the number of hops that the beacon packets are allowed to travel from the beacon nodes (similar to the time-to-live (TTL) field in IP). To eliminate the influence of measurement noise, we just ignored it in this simulation (but included it in all other simulations). The accuracy of the proposed approach as a function of the number of hops that beacon information is allowed to travel is shown in Fig. 2.10. Different lines correspond to different network densities.

For sparse networks, many unknown nodes are unable to hear directly from the

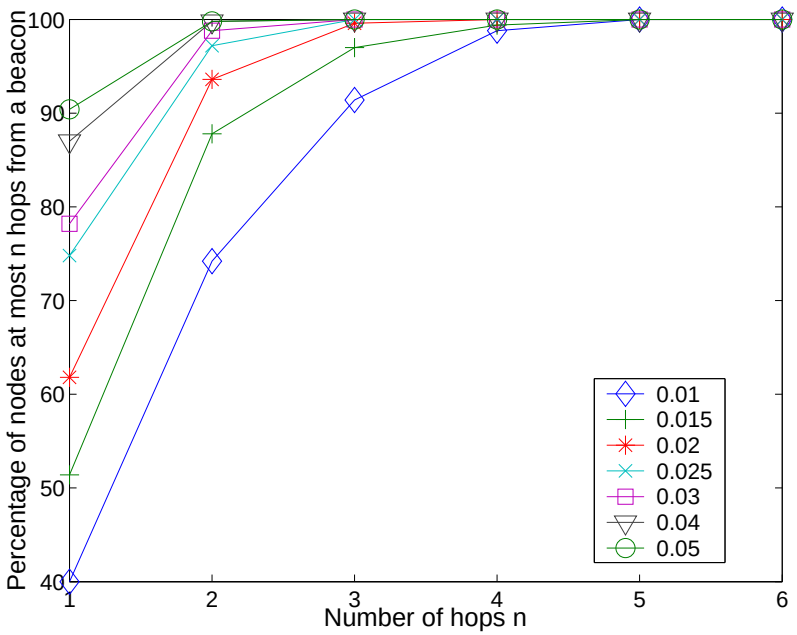


Figure 2.11: The percentage of nodes reachable by at least one beacon in n hops as a function of n for different node densities.

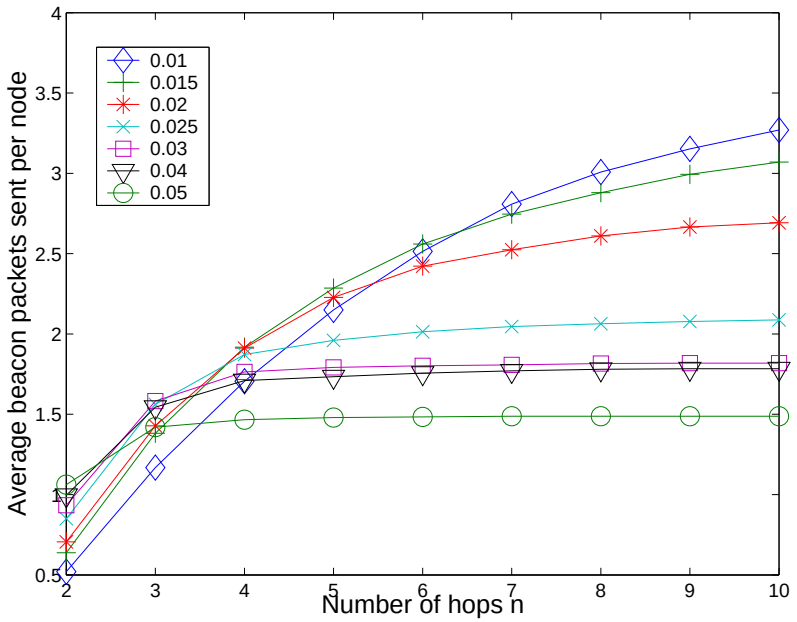


Figure 2.12: The average communication cost as a function of the number of hops from the beacons to the unknown nodes for different node densities.

beacons, and, hence, have to rely on information propagated through multiple hops. Fig. 2.10 shows that, regardless of the network density, information does not have to travel for many hops before the proposed approach converges; information from distant beacons cannot significantly improve the accuracy of the proposed approach. Thus, in the proposed approach, unknown nodes only need local (a few hops away) information to localize themselves.

Fig. 2.11 shows the percentage of nodes that are within n hops range of at least one beacon for different densities.

Fig. 2.12 depicts the average number of packets sent by every unknown node for different network densities. Since each unknown broadcasts a beacon packet only after it has a considerable position improvement (larger than the threshold we set as 1m), the number of beacon packets required does not increase linearly with the limit on the number of hops. Two observations are in order:

- More beacon packets are sent in high density networks when the allowed number of hops is small: more unknowns can hear directly from the beacons and forward the information.

- For dense networks, beacon packets with small TTL are enough for localization (as also shown in Fig. 2.10); however, for low density networks, beacon packets from distant beacons may still affect the position estimate of an unknown. Therefore, more beacon packets are required to broadcast the change in the position estimates when the allowed number of hops is large.

As shown in Fig. 2.10, information from the beacons has to travel only a few hops for the approach to be effective. Thus, while the beacon packets become larger as they travel from the beacons (by two to three bytes per hop), the beacon packets do not have to be more than 15-20 bytes long.

In what follows, all approaches are evaluated through two aspects: accuracy (average estimation error) and precision (standard deviation of the location estimate). For the proposed approach, results both before and after the negative constraint refinement are shown in all of the figures in order to highlight the influence of negative constraints on the performance of the proposed system.

The Effect of Network Density

In this simulation, we studied the effect of network density on the accuracy of the proposed approach. To change the density, we changed the deployment area and kept the number of beacons and unknowns constant. The accuracy and precision results are presented in Figures 2.13 and 2.14 respectively.

Both accuracy and precision improve with the increase of network density. This is not surprising as the CRLB is rather sensitive to the number of neighbors of a node. Similarly, in the proposed approach, the more neighbors, the more constraints will be placed on the unknowns. Another reason for the poor performance at lower densities is that (as shown in Fig. 2.11) the unknowns are relatively too far from the beacon nodes to obtain any estimation improvements. The negative constraints improve the effectiveness of the proposed approach considerably: the performance is very close to the optimum (the CRLB).

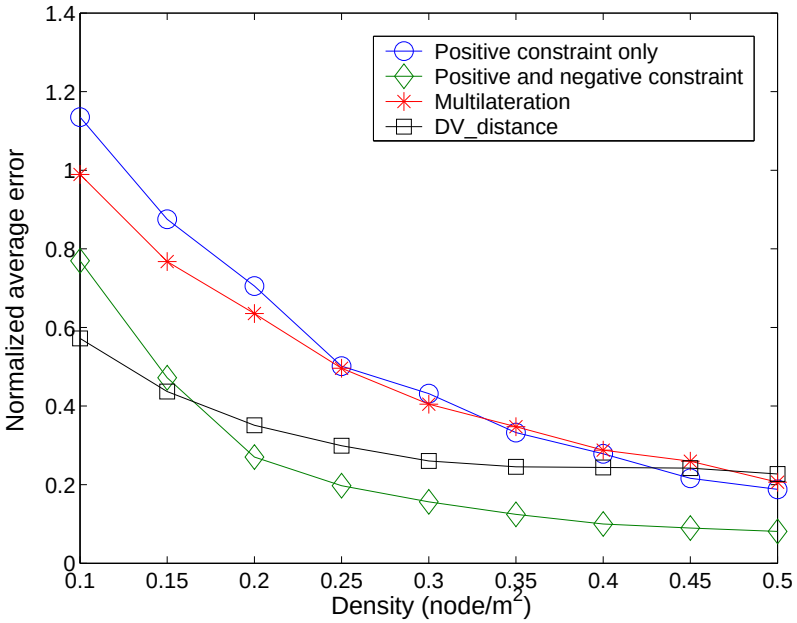


Figure 2.13: The effect of network density on localization accuracy.

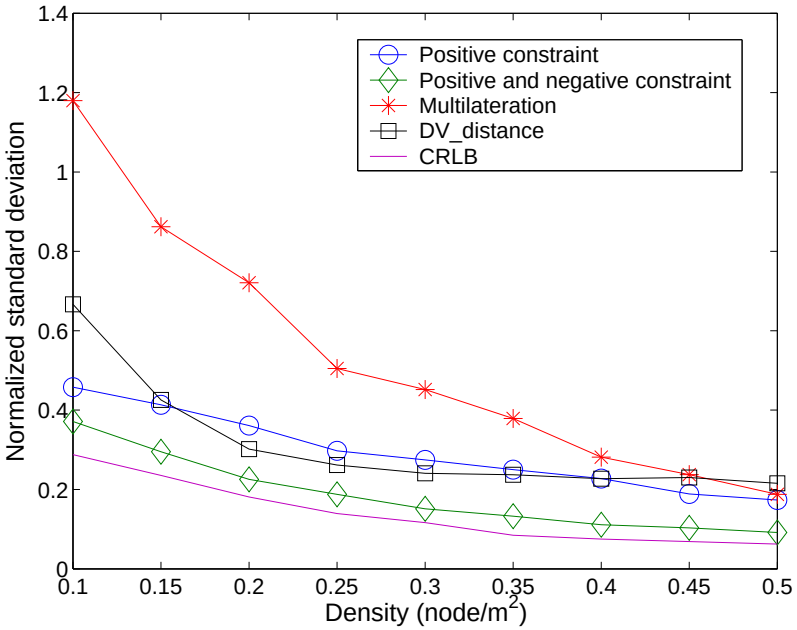


Figure 2.14: The effect of network density on localization precision.

The Effect of the Number of Nodes

In this scenario, we varied the number of beacons and unknowns in three different ways.

First, we varied the number of unknowns from 10 to 100 while maintaining the number of beacons and network density constant. The average errors and the standard deviations of the estimation are shown in Figures 2.15 and 2.16 respectively. The standard deviations for all methods increase as the number of unknowns increases, while the CRLB is almost constant. The intuition behind this result is that only the beacon nodes are aware of their positions and introduce information into the system: the unknown nodes will estimate their positions based on the information from the beacons. As the number of unknowns increases, the number of hops from beacons to most of the unknowns increases as well (recall that the node density is kept constant); thus, the information from the beacons will be diluted by going through multiple hops. The constant number of neighbors and beacons account for the almost constant CRLB.

Second, we varied the number of total nodes in the network from 30 to 100 while

keeping the fraction of the beacons in the network as 10% and constant density. As can be seen from Figures 2.17 and 2.18, as long as the fraction of the beacons is constant, the CRLB decreases slightly. The performances for the three-phase probabilistic approach do not vary much along with the network scale, which obeys the tendency of the CRLB.

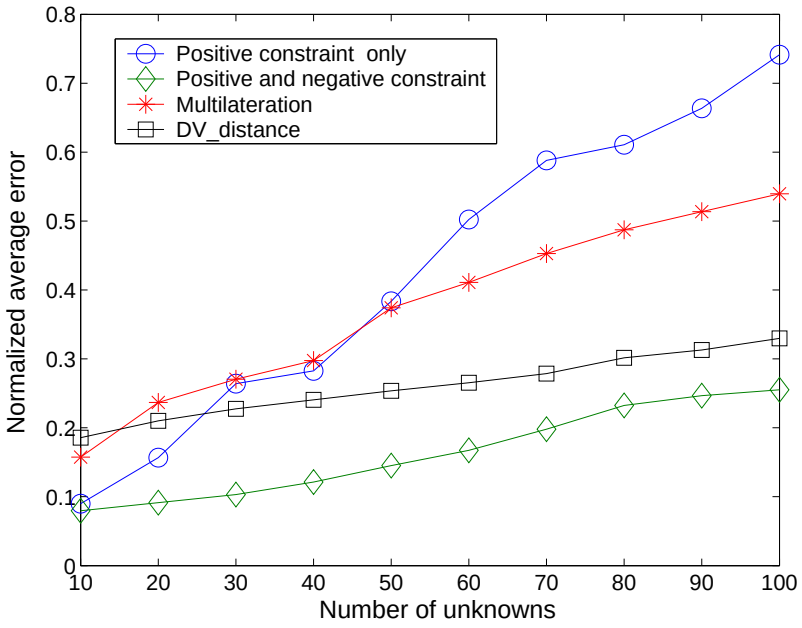


Figure 2.15: The effect of the number of unknowns on localization accuracy (beacon number is constant).

Finally, to evaluate the effect of the beacon density on the proposed approach, we varied the percentage of the beacons from 6% to 20%. The results are shown in Figures 2.19 and 2.20. The CRLB indicates that the localization precision does not increase significantly with the increase of beacon percentage. However, the precision of all approaches decreases for low beacon densities: at low beacon densities the packets from a beacon will take several hops to reach an unknown. In all algorithms, the estimation error is accumulated and propagated in each hop, which causes the relative poor performances for low beacon densities.

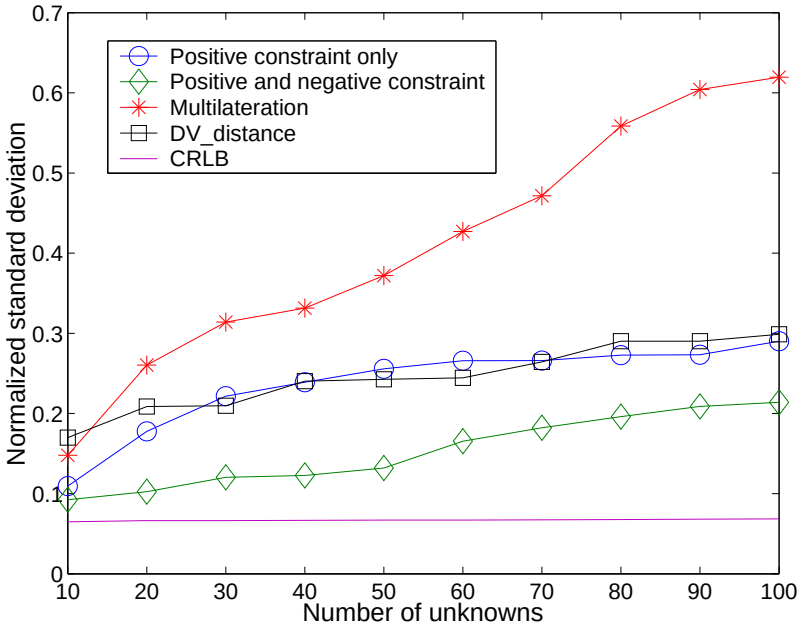


Figure 2.16: The effect of the number of unknowns on localization precision (beacon number is constant).

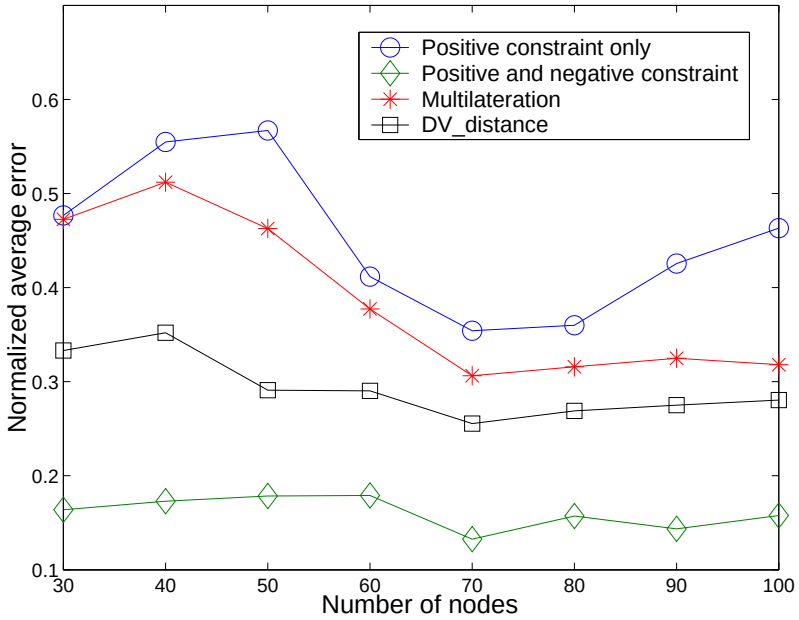


Figure 2.17: The effect of the network scale on localization accuracy (beacon percentage is 10%).

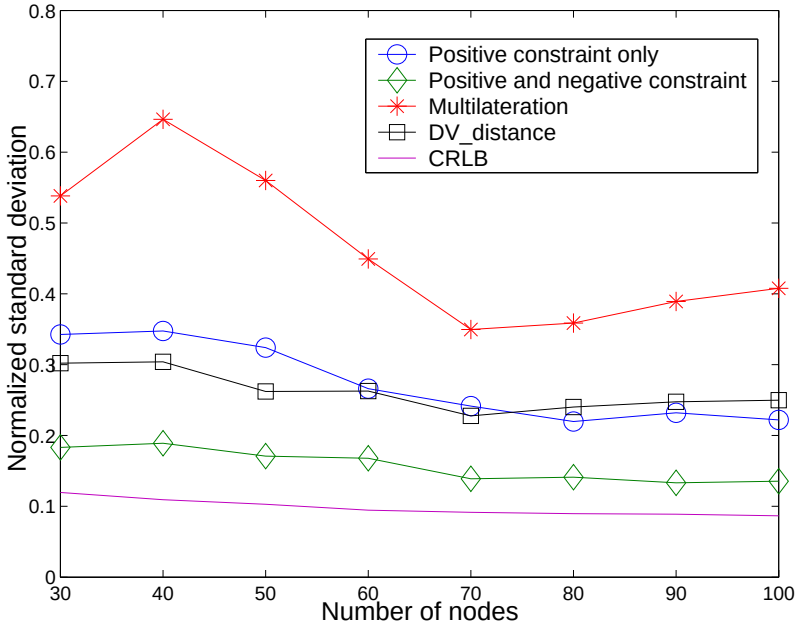


Figure 2.18: The effect of the network scale on localization on localization precision (beacon percentage is 10%).

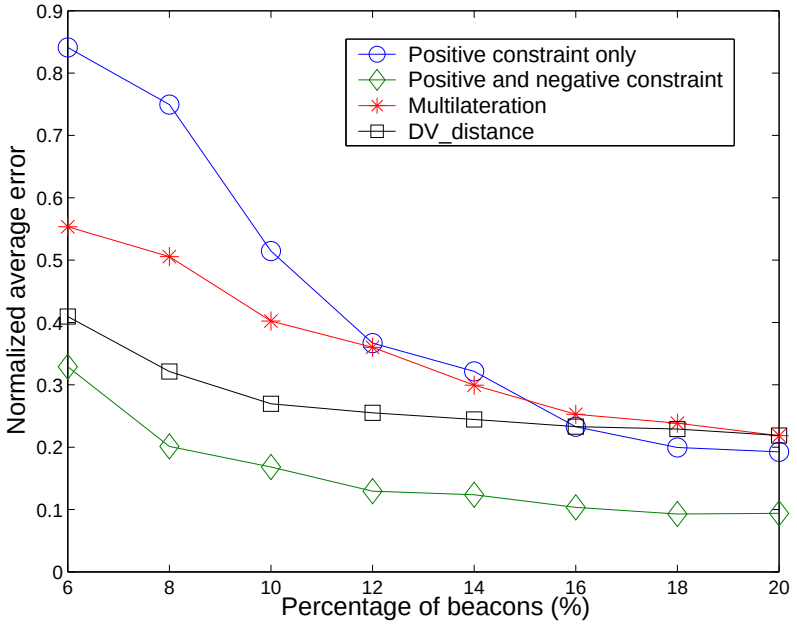


Figure 2.19: The effect of beacon density on localization accuracy.

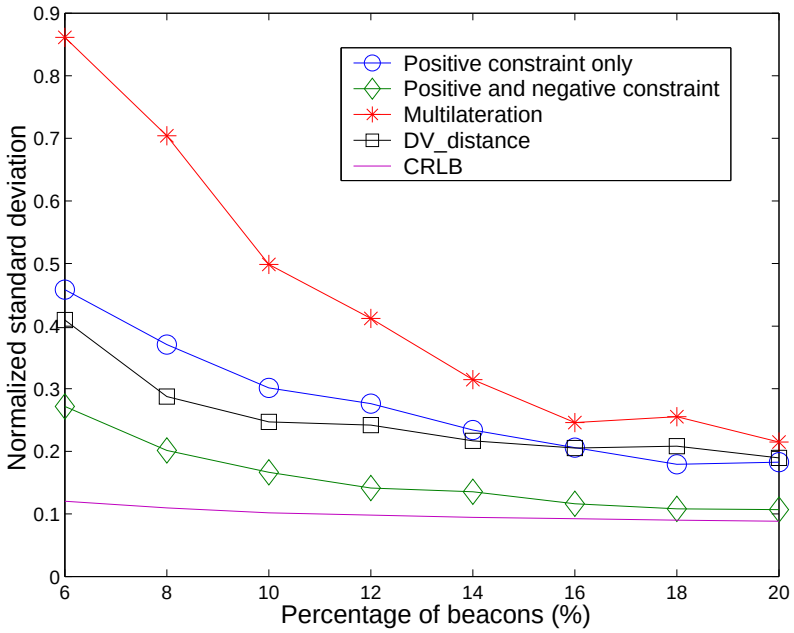


Figure 2.20: The effect of beacon density on localization precision.

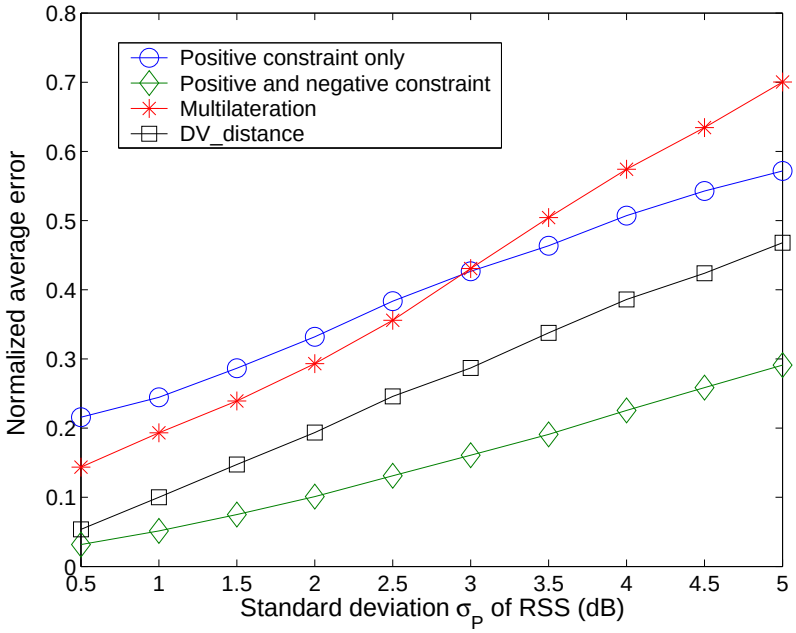


Figure 2.21: The effect of range inaccuracy on localization accuracy.

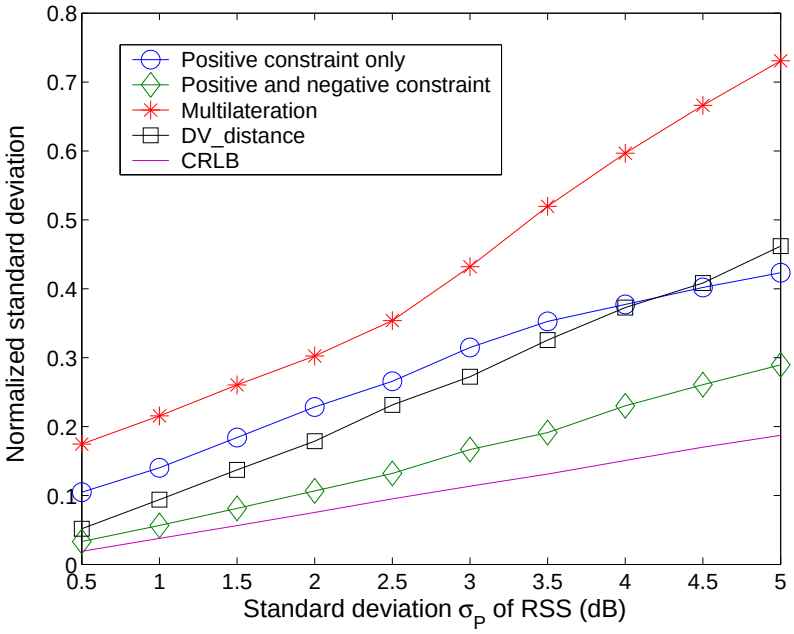


Figure 2.22: The effect of range inaccuracy on localization precision.

The Effect of the Range Measurement Inaccuracy

Finally, we studied the effect of the inaccurate measurements on the accuracy and the precision of the proposed approach, by increasing the standard deviation of the RSS (σ_P). We varied σ_P between 0.5dB and 5dB (recall that the average σ_P from calibration measurements is approximately 2.5dB). The results are shown in Figures 2.21 and 2.22. The CRLB increases proportionally to σ_P. The performance of all algorithms follows this increasing tendency, worsening their performance as the measurements become more inaccurate.

All of the results share some common characteristics:

- the negative constraints considerably improve the accuracy and the precision of the proposed approach;
- the proposed approach consistently outperforms other existing localization approaches; and
- the precision of the proposed approach is, in many situations, very close to the

theoretical optimum (the CRLB).

2.4 Conclusion

In this chapter, we presented a distributed RSS-based probabilistic localization approach for wireless stationary sensor networks. The approach solves two major problems. First, it accounts for inaccurate range measurements by using probability functions for range measurements. Second, it accomplishes localization with only a small number of beacons by using multi-hop position information.

The approach can be divided into three phases. The strength of the received power at an unknown nodes is mapped to the pdf of the distance in the first phase. Positive and negative constraints are used in the second and third phases, respectively, to localize the unknown nodes. The negative constraints originating at the beacons can significantly reduce the uncertainties of the positions estimate of the unknown nodes. In order to reduce the computational complexity for fine estimation and large-scale networks, we also presented an extended probabilistic grid approach based on Fast Fourier Transform (FFT). Further more, solutions to eliminate the dependencies among the beacon packets are described. Last, as shown in the simulation results, the probabilistic approach outperforms other existing range-based localization schemes and approaches the Cramer-Rao lower bound.

Chapter 3

Angle-based Probabilistic Localization in Static WSNs

In this chapter, we focus on localization techniques based on angle information between neighbor nodes. Inspired by the localization method introduced in the last chapter, we propose a new angle-based probabilistic localization approach. The approach includes two scenarios: localization when orientations of the unknowns are available and when they are not available. In the second scenario, both positions and orientations of the unknowns can be determined.

3.1 Overview of Localization Using Angle of Arrival

The angle of arrival (AOA) is defined as the angle between the propagation direction of an incident wave and some reference direction, which is known as orientation. *Orientation*, defined as a fixed direction against which the AOAs are measured, is

represented in degrees in a clockwise direction from the North. When the orientation is 0° or pointing to the North, the AOA is absolute, otherwise, relative. One common approach to obtain AOA measurements is to use an antenna array on each sensor node. Other techniques to detect the angles between nodes are discussed in [28] and [65]. In this chapter, we assume that the beacons have no information about their orientations and the unknowns can detect the AOA information between neighbor nodes by one of the above methods.

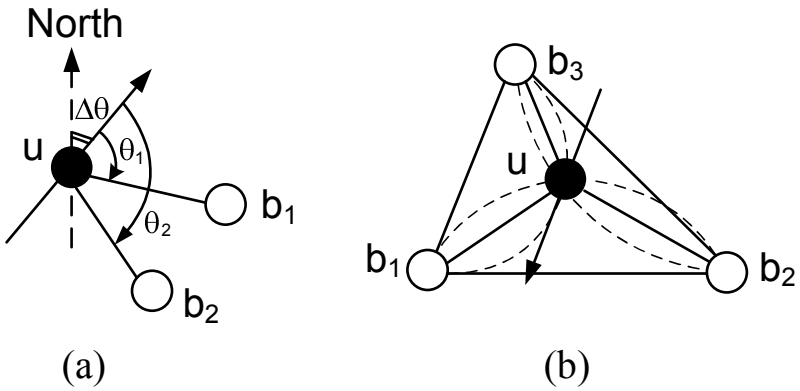


Figure 3.1: Triangulation in AOA localization: (a) Localization with orientation information; (b) Localization without orientation information.

The orientations of the unknowns may or may not be known at the deployment time. Localizations under both scenarios can be solved using triangulation. We first consider the case when the orientations of the unknowns are known. As shown in Fig. 3.1(a), angle θ_1 and θ_2, which are measured at unknown u, are the relative AOAs of the signals sent from beacons b_1 and b_2, respectively. Assume the orientation of an unknown is $\Delta\theta$, the absolute AOAs from b_1 and b_2 can be calculated as $\theta_i + \Delta\theta (\text{mod } 2\pi), i = \{1, 2\}$. Each absolute AOA measurement corresponding to a beacon restricts the location of the unknown along a ray starting at the beacon. The location of the unknown u is located at the intersection of all rays when two or more noncollinear beacons are available. When the orientations of the unknowns are not available, in another words, when the absolute AOAs cannot be obtained, the

AOA differences can be used instead. As shown in Fig. 3.1(b), angle $\angle b_1ub_2$, $\angle b_1ub_3$ and $\angle b_2ub_3$ can be computed using the knowledge of the relative AOAs. According to the properties of the circle, all angles inscribed in a circle and subtended by the same chord are equal. This infers that given two points and the chord joining them, a third point from which the chord subtends a fixed angle can be constrained on an arc of a circle. For example, $\angle b_1ub_2$ and chord b_1b_2 restricts u's position on the arc passing through b_1, u and b_2. Each chord determines one arc; thus the location of an unknown is estimated at the intersection of all arcs when three or more non-collinear beacons are available. In summary, two neighbor noncollinear beacons are the minimum requirement to discover the locations when the orientations are known; and three noncollinear beacons are the minimum requirement to discover both the locations and the orientations.

Similar to localization using other measurements, AOA localization also faces the challenges of measurement noise and additional problems if unknowns cannot hear from a sufficient number of beacons. In [28], the AOA measurements are exchanged between neighbor nodes, the relative AOA with respect to each beacon (even hops away) can be calculated based on geometry relations among the nodes. This enables the use of triangulation for localization. In [30], the problems were solved by using a semidefinite programming relaxation based method.

The propose a probabilistic localization scheme solves both challenges. In contrast to the existing schemes, the proposed scheme is derived under the assumption of AOA measurement inaccuracies. The measurement noise is modeled probabilistically and the algorithm utilizes the model localize the unknowns. The position information of the beacons and the AOA information at each unknown are broadcasted into the network (within a limited number of hops). Therefore, even without hearing from a sufficient number of beacons, the position information of the beacons can be utilized indirectly by the unknowns several hops away.

3.2 Probabilistic Localization

In this section, we present our angle-based localization and orientation approach. Without loss of generality, the following assumptions are made throughout the rest of the chapter:

- All of the angle related variables are in the range of $[0, 2\pi)$.
- The transmission between nodes are bounded by a maximum transmission range d_{max}, any packets received from a node outside of the transmission range is considered too weak to contribute.
- There are no major obstructions between the transmission of any two nodes. Therefore, the AOA measurements can be assumed to have certain distribution centered around the direction of the line-of-sight (LOS).

Each sensor node utilize both the position of the beacons and the AOA measurements to estimate the position. Similar to the RSS-based approach shown in the last chapter, the position of each sensor node is estimated through a probability distribution function of the two-dimensional coordinate random variable (X, Y) in a collaborative and distributive manner.

3.2.1 Distribution of AOA measurements

The AOA measurement inaccuracies can be caused by either the wireless communication channel or the measuring device/method, or both. The spatial properties of the wireless channel have significant impact on the detection of AOA [66]. A considerable effort has been dedicated to finding good models to characterize these properties. Existing statistical models for the distribution of the AOA received at a wireless node include: Laplacian AOA distribution [67], von Mises AOA distribution [68] and Gaussian AOA distribution [69]. The measuring device and method also plays an important role in the accuracy of the AOA measurements [70, 71].

Since the distribution of the AOA measurements highly depends on the communication environment and the AOA detection device/method, it is very difficult and out of the scope of this report to find a single model that can be applied to all situations. Although the proposed localization approach is suitable for all AOA distributions mentioned above, for generalization and analytical convenience, we used the Gaussian distribution to characterize the AOA measurements combining the errors cause by both the channel and the device/method. Given that the direction of the LOS is θ_{LOS}, the AOA measurements can be described by a Gaussian distribution with mean of θ_{LOS} and standard deviation of σ_θ, which can be used to describe the spread of the AOA measurements and varies with the environment. Conversely, given a measured AOA of θ_{AOA}, the distribution of the direction of the LOS is:

$$\theta_{LOS} \sim N(\theta_{AOA}, \sigma_\theta). \tag{3.1}$$

A Gaussian noise model for the AOA measurement is also considered in [28, 30].

3.2.2 Probabilistic Localization with Orientation Information

In this part, we assume that the orientation of each sensor node is *a priori* known. The absolute AOA can be calculated using both the orientation information and the relative AOA as depicted in Fig. 3.1(a). Recall that for nodes with known orientation, the minimum number of beacons required to estimate the position is two. However, since the nodes are usually assumed to be (very) sparse, most of the nodes can hear directly from no or only one beacon, causing the localizations for most of the nodes to fail. In the proposed approach, the position information of beacons multiple hops away is used, such that the estimations can be performed at each node.

Again, we use *pseudo-beacon* to refer an unknown with an estimated position probability distribution function. To propagate the position information of the beacons through the network, both beacons and pseudo-beacons send out beacon packets to

their one-hop neighbors. Initially, each unknown sets its position to be uniformly distributed over the entire network deployment area. An unknown node j receiving a beacon packet from a beacon/pseudo-beacon node i executes the following steps:

- Measures the relative AOAs of the received packets and calculates the absolute AOAs;
- Updates its position distribution and pdf using both the position information of beacons/pseudo-beacons and the computed absolute AOAs;
- All unknowns with updated pdfs become pseudo-beacons and send out their updated pdfs to their one-hop neighbors.

Assume the angle $\widetilde{\theta}_{ij}$ is j's measured absolute AOA for i's beacon packet, according to (3.1), the distribution of i's LOS direction 'seen' at j is:

$$f_{\Theta_{ij}}(\theta) = \frac{1}{\sqrt{2\pi}\sigma_\theta} e^{-\frac{(\theta-\widetilde{\theta}_{ij})^2}{2\sigma_\theta^2}}. \tag{3.2}$$

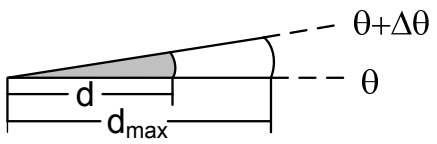


Figure 3.2: An infinitesimal sector for a given angle of θ_j.

Variable Θ_{ij} in (3.2) can be considered as the angular coordinate in the polar coordinate system. In addition, we define random variable D_{ij}, which can be treated as the distance variable in the polar coordinate system, representing the distance from node i to j. Variables Θ_{ij} and D_{ij} are independent. In order to find the joint pdf $f_{\Theta_{ij},D_{ij}}(\theta, d)$, it boils down to find the pdf of D_{ij} . As we use only pure angle measurement, the only information about the distance between nodes i and j is that $D_{ij} \leq d_{max}$. Recall that in calculus, we can use a sector with an infinitesimal angle $\Delta\theta$ to approximate an angle Θ_{ij} within $[\theta, \theta + \Delta\theta]$ as in Fig. 3.2. If the radius of this sector is d_{max}, since node j can be anywhere in the sector, the probability that $D_{ij} \leq d$ is the same as the probability that node j falls in the gray area:

$$P(D_{ij} \leq d) = F(d) = \frac{d^2}{d_{max}^2}, \tag{3.3}$$

where $F(d)$ is the cumulative distribution function (cdf) for variable D_{ij}. The pdf for the distance coordinate can be found simply by taking the first order derivative of (3.3):

$$f_{D_{ij}}(d) = \frac{2d}{d_{max}^2}. \tag{3.4}$$

Thus j's joint pdf for the polar coordinate random variable (D_{ij}, Θ_{ij}) is the product of (3.2) and (3.4). For convenience, this joint pdf is transformed into the joint pdf of the cartesian coordinate random variable (X_{ij}, Y_{ij}) by using the *Jacobian* of the transformation

$$\begin{aligned} f_{X_{ij},Y_{ij}}(x,y) &= \frac{1}{|d|} f_{D_{ij},\Theta_{ij}}(\theta,d)\Big|_{\substack{d=\sqrt{x^2+y^2} \\ \theta=\vartheta(x,y)}} \\ &= \frac{\sqrt{2}}{\sqrt{\pi}\sigma_\theta d_{max}^2} e^{-\frac{(\vartheta(x,y)-\tilde{\theta}_{ij})^2}{2\sigma_\theta^2}}. \end{aligned} \tag{3.5}$$

where $\vartheta(x,y)$ is the polar angular function of x and y:

$$\vartheta(x,y) = \begin{cases} \frac{\pi}{2} - \tan^{-1}\frac{y}{x}, & x \geq 0, \\ \\ \frac{3\pi}{2} - \tan^{-1}\frac{y}{x} \pmod{2\pi}, & x < 0. \end{cases} \tag{3.6}$$

Assume the position pdf of a beacon/pseudo-beacon i is $f_{X_i,Y_i}(x,y)$, the position pdf for unknown j can be calculated by:

$$f_{X_j,Y_j}(x,y) = f_{X_i,Y_i}(x,y) * * f_{X_{ij},Y_{ij}}(x,y). \tag{3.7}$$

Intuitively, the convolution simply shifts the pdf of the LOS from the origin to the location of the beacon i.

After computing $f_{X_j,Y_j}(x,y)$ using i's information, node j calculates its position probability distribution, position pdf and position estimate according to equations (2.13,2.14,2.15,2.16).

Since the intersection combining the original and new probability distribution information is possible only when all coordination random variables are mutual independent random variables. Here we use the approach introduced in section 2.2.2 to eliminate the dependencies among the coordination random variables.

Example

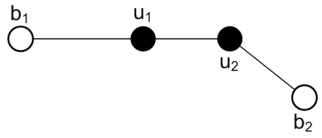


Figure 3.3: Network topology with two beacons (b_1, b_2) and two unknowns (u_1, u_2).

The beacons (b_1 and b_2) and unknowns (u_1 and u_2) are deployed as depicted in Fig. 3.3. Unknowns u_1 and u_2 can hear from b_1 and b_2, respectively, and from each other. Upon receipt of the beacon packets from beacons, nodes u_1 and u_2 update their position probability distribution as shown in Fig. 3.4(a) and 3.4(b). Both unknowns become pseudo-beacons and broadcast their updated pdf. After receiving from u_2, node u_1 computes a new distribution constraint as shown in Fig. 3.4(c) and intersects it with the one in Fig. 3.4(a) to obtain the final estimation in Fig. 3.4(e). Unknown u_2 uses u_1's pdf to calculate a new distribution as shown in Fig. 3.4(d) and obtain the final estimation in Fig. 3.4(f).

3.2.3 Log Format and Dependency Elimination

In order to facilitate the presentation of the localization without orientation information, we first explore how the algorithm operates when the orientations are known.

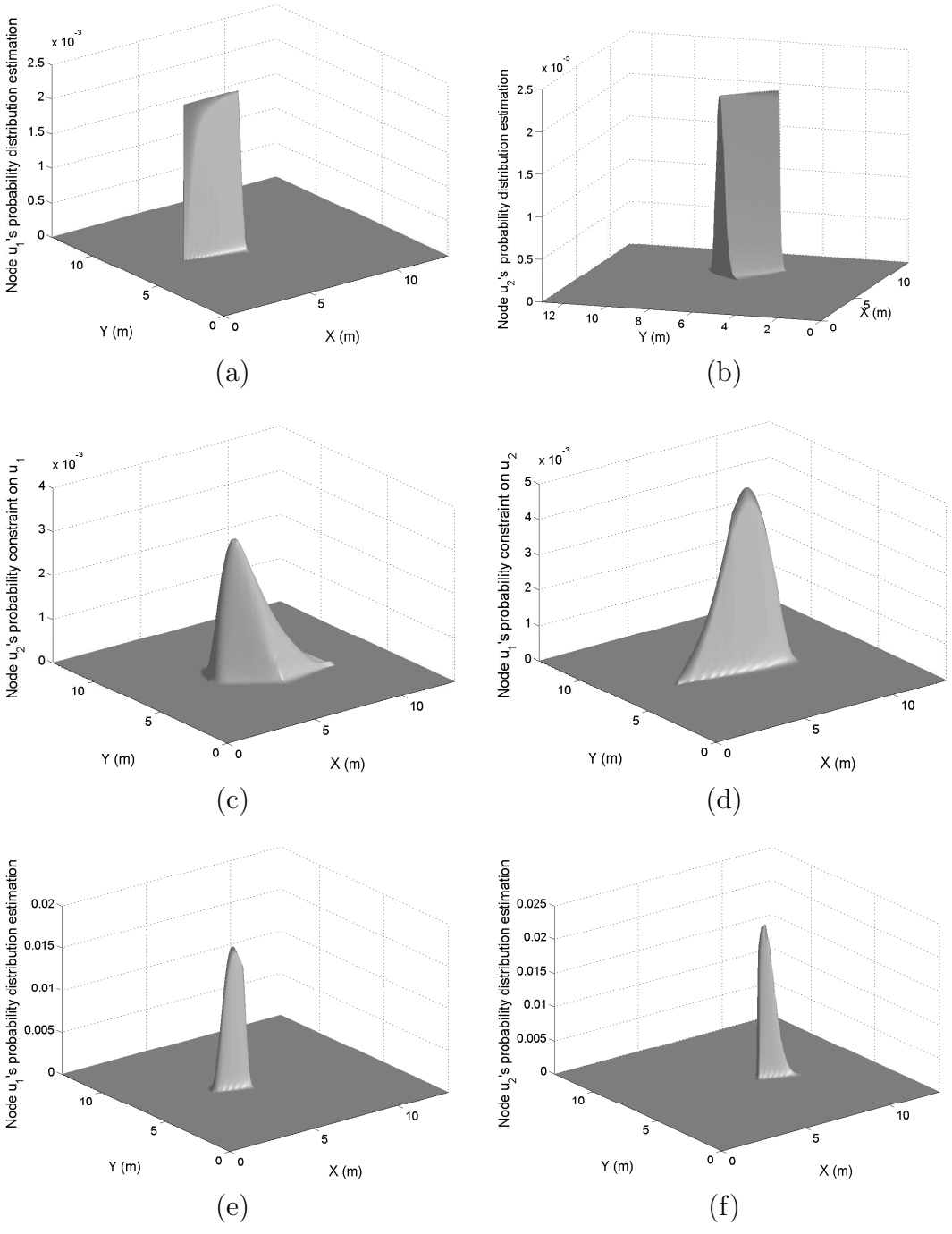


Figure 3.4: Localization evolution for the network topology in Fig. 3.3

As stated, the position information of the beacons and the updated pdf of the unknowns are transmitted in the network. Since the pdf estimation of each unknown is different from the others and too complicated to be expressed analytically, to record such pdfs, each unknown keeps a log, in which only a cascade of measured AOAs are stored as shown in table 3.1.

Table 3.1: Beacon packet format for AOA-based localization.

L_1	ID_{b1}	x_{b1}	y_{b1}	ID_{u1}	θ_{b1u1}	θ_{u2u1}	ID_{u2}	θ_{u1u2}	$\cdots$
L_2	ID_{b1}	x_{b1}	y_{b1}	ID_{u3}	θ_{b1u3}	θ_{u4u3}	ID_{u4}	θ_{u3u4}	$\cdots$
L_3	ID_{b2}	x_{b2}	y_{b2}	ID_{u3}	θ_{b2u3}	θ_{u5u3}	ID_{u5}	θ_{u3u5}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Same as the log in RSS-based approach, each entry in a log starts with the length of that entry and the beacon information, and each entry actually represents a transmission path from the beacon to the unknown node storing the log as in table 3.1. Upon the receipt of a beacon packet from a neighbor node, an intermediate unknown appends its node ID to the packet entry and records two AOAs: one is the measured AOA of the incident packet from its ancestor node; the other is the measured AOA of the next node along the path. If the orientations are known, the recorded AOAs are absolute, otherwise, relative.

Each destination unknown appends its ID and the measured AOAs from its ancestor to the packet, and stores the new packet in its log. The log is then processed using the method introduced in section 2.2.2 to eliminate the dependencies among entries. Summarizing the dependency elimination method, the processed log has following properties to ensure the information in the log is independent:

- A node can appear only once in an entry.
- If an intermediate unknown appears in more than one entry and occupies column C_i in ith entry, the value of all $L_i - C_i$ must be same. In another word, in all the entries, the hops between an specific unknown and the estimation node must be the same.

- If there are more than one entry containing a path starting from intermediate unknown i to intermediate unknown j, the nodes between i and j must be same.

The processed log is then used at each unknown to calculate the position distribution estimation. The estimation node executes the following pseudo code LOC-OK(log) (**Loc**alization when **O**rientation is **K**nown) to calculate its position pdf. The input is the stored log and the ID of the estimation unknown.

Algorithm 2 LOC-OK(log,j)

$f(x,y) = \frac{1}{A}$
$neighbors \leftarrow j$'s neighbors in the log
for each $i \in neighbors$ **do**
  if $f_{X_i,Y_i}(x,y)$ exists **then**
    $\widetilde{\theta}_{ij} \leftarrow$ AOA from i to j stored in the log
    Calculate $f_{X_{ij},Y_{ij}}(x,y)$
    $f_{X_j,Y_j}(x,y) \leftarrow f_{X_i,Y_i}(x,y) * * f_{X_ij,Y_ij}(x,y)$
    Calculate $P_j(x,y)$
    $P(x,y) \leftarrow P(x,y)P_j(x,y)$
  else
    LOC-OK(log,i)
  end if
end for
update $f_{X,Y}(x,y)$ using $P(x,y)$
return $P(x,y)$,$f_{X,Y}(x,y)$;

3.2.4 Probabilistic Localization without orientation information

In some scenarios, orientation information is not available to the sensor nodes. For example, the sensor nodes may be dropped from a helicopter or an Unmanned Aerial Vehicles (UAV) without compass function embedded. For such situation, only relative AOAs can be obtained and the localization scheme discussed above does not apply. In section II, we showed that at least three neighbor beacons are required to realize the estimation of both location and orientation as in Fig 3.1(b). However,

when this requirement is not satisfied, one possible scheme to solve the localization problem is to utilize the angle difference between AOAs from two different neighbors (either beacons or pseudo-beacons). Assume node j hearing from two neighbors i and k with AOAs $\Theta_{ij} \sim N(\widetilde{\theta}_{ij}, \sigma_\theta)$ and $\Theta_{kj} \sim N(\widetilde{\theta}_{kj}, \sigma_\theta)$, respectively. The angle difference $\angle ijk$, which is defined by variable $\Theta_{ijk} = \Theta_{ij} - \Theta_{kj}$, has a pdf of

$$f_{\Theta_{ijk}}(\theta) = \frac{1}{2\sqrt{\pi}\sigma_\theta} e^{-\frac{(\theta-(\widetilde{\theta}_{ij}-\widetilde{\theta}_{kj}))^2}{4\sigma_\theta^2}} \tag{3.8}$$

Variable Θ_{ijk} is a function of both (X_{ij}, Y_{ij}) and (X_{kj}, Y_{kj}), which are the random variables for the distribution constraints calculated using i's and k's information, respectively. Recall that in (3.5), when Θ_{ij} is a function of $(X_i j, Y_i j)$, the two dimensional convolution in (3.7) requires double integral on both X_i and Y_i to obtain the position pdf for node j. Similarly, using the pdf of θ_{ijk} to calculate j's position pdf, 4th-order integral on X_j, Y_j, X_k and Y_k is required. Furthermore, if node j has n neighbors, it leads to a computation of C_n^2 position pdfs. When n is large, the computational cost can be high.

To avoid C_n^2 times computation of 4th-order integrals, we propose a new scheme to estimate the orientations of the unknowns without three neighbor beacons.

First, we define a new term *AOA pair*: An AOA pair consists of two angle measurements between two neighbor nodes i and j: one is j's relative AOA measured at i and the other is i's relative AOA measured at j.

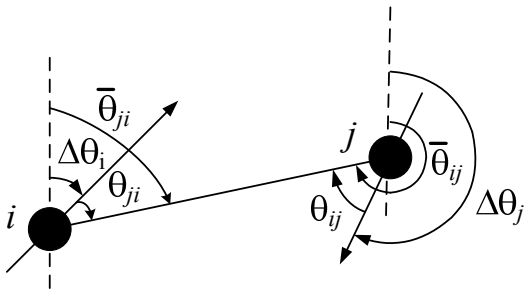


Figure 3.5: θ_1 and θ_2 consist of an AOA pair.

As shown in Fig. 3.5, θ_{ij} and θ_{ji} consist of an AOA pair. In the figure, angles $\Delta\theta_i$ and $\Delta\theta_j$ represent the orientations for i and j. Angles $\overline{\theta}_{ij}$ and $\overline{\theta}_{ij}$ are the absolute AOAs corresponding to the relative ones θ_{ij} and θ_{ji}. When the AOA estimation is accurate, we have

$$|\overline{\theta}_{ij} - \overline{\theta}_{ji}| = \pi. \tag{3.9}$$

All six angles ($\Delta\theta_i$, $\Delta\theta_j$, $\overline{\theta}_{ji}$, $\overline{\theta}_{ij}$, θ_{ji}, θ_{ij}) in the figure are in $[0, 2\pi)$. Since the values of θ_{ji} and θ_{ij} can be measured, given any one of the other four, the rest three angles can be uniquely determined. For example, if $\Delta\theta_i = \theta_0$, then

$$\overline{\theta}_{ji} = \theta_{ji} + \Delta\theta_i \pmod{2\pi}, \tag{3.10}$$

$$\overline{\theta}_{ij} = \begin{cases} \overline{\theta}_{ji} + \pi, & \overline{\theta}_{ji} \leq \pi, \\ \\ \overline{\theta}_{ji} - \pi, & \overline{\theta}_{ji} > \pi, \end{cases} \tag{3.11}$$

$$\Delta\theta_j = \overline{\theta}_{ij} - \theta_{ij} \pmod{2\pi}. \tag{3.12}$$

Recall that when the unknowns are unaware of their orientations, the AOAs recorded in the log, as shown in Fig. 3.1, are relative. Assuming that the estimation node's orientation is $\Delta\theta_0$, using the AOA pair relation discussed above, the orientations of the unknowns that are one hop before the estimation node can be determined. Using these unknowns' newly determined orientations, the unknowns that are one hop before these unknowns can also be determined. Therefore, if we apply the AOA pair relation to all neighboring unknowns in the log, we can obtain a set of orientations for all of the unknowns in the log $\{\Delta\theta_k|\Delta\theta_0\}, ID_k \in \log$, in which each orientation is related to $\Delta\theta_0$. Consequently, the absolute AOAs corresponding to each relative AOA in the log can be calculated. Now if we rotate the estima-

tion node's orientation by an step of $\Delta\theta'$, we will obtain another set of orientations $\{\Delta\theta'_k|\Delta\theta_0 + \Delta\theta'\}, ID_k \in \log$ that are related to $\Delta\theta_0 + \Delta\theta'$.

The localization scheme is based on calculating a position estimate of the estimation unknown for each set of the orientations given an assumed orientation of that unknown. For different assumed orientations, there are different position estimations.

Example

In Fig. 3.6, b_1, b_2, b_3 are beacons and u_1 and u_2 are unknowns. None of the unknowns can hear directly from all three beacons. The true orientations of both unknowns are 0. Unknown u_1 receives beacon packets from b_1, b_2 and u_2; and u_2 receives beacon packets from b_3 and u_1. First, consider the localization for u_1. Assume the AOA measurements are accurate, if we let u_1's orientation $\Delta\theta_1 = 0$ (same as the true orientation), as shown in Fig. 3.6(a), the absolute AOA $\overline{\theta}_{b1}$ and $\overline{\theta}_{b2}$ are same as the relative ones θ_{b1} and θ_{b2}. Two absolute AOAs constrain u_1 at the intersection of two lines $\overrightarrow{b_1u_1}$ and $\overrightarrow{b_2u_1}$ (we use $\overrightarrow{AB}$ to represent a line segment that starts from A , passing through B, and having a length of d_{max}). Using u_1's orientation and the AOA pair relation, u_2's orientation can be calculated as 0. Thus, the AOA from b_3 constrains u_2 on $\overrightarrow{b_3u_2}$. Consequently, u_1 using AOA coming from u_2, it locates itself in an area defined by an infinite number of lines with length of d_{max} and parallel to $\overrightarrow{u_2u_1}$. Each of these lines starts from a point on $\overrightarrow{b_3u_2}$. The intersection of these lines and the original intersection of $\overrightarrow{b_1u_1}$ and $\overrightarrow{b_2u_1}$ results in a unique point, which is the exact position of u_1. Similarly, at u_2, if we let u_2's orientation $\Delta\theta_2 = 0$, u_1's orientation can be computed as 0 and u_1's position can be obtained. The constraints from b_3 and u_1 intersects on u_2's exact position.

Now, rotate u_1's orientation to the direction shown in Fig. 3.6(b), assume $\Delta\theta'$, constraints from b_1 and b_2 intersect at the point of u_1 as shown in 3.6(b) (a different point on the same arc as in the first figure). By AOA pair relation, we can compute that u_2's orientation is also $\Delta\theta'$. By utilizing this newly computed orientation, u_2 is constrained on the dashed line $\overrightarrow{b_3u_2}$. Unknown u_2's constraint constrains u_1 in an area consisted of infinite lines parallel to $\overrightarrow{u_2u'_1}$ and started from a point on $\overrightarrow{b_3u_2}$. It is not hard to see that the intersection of all the constraints is empty for u_1. Similarly,

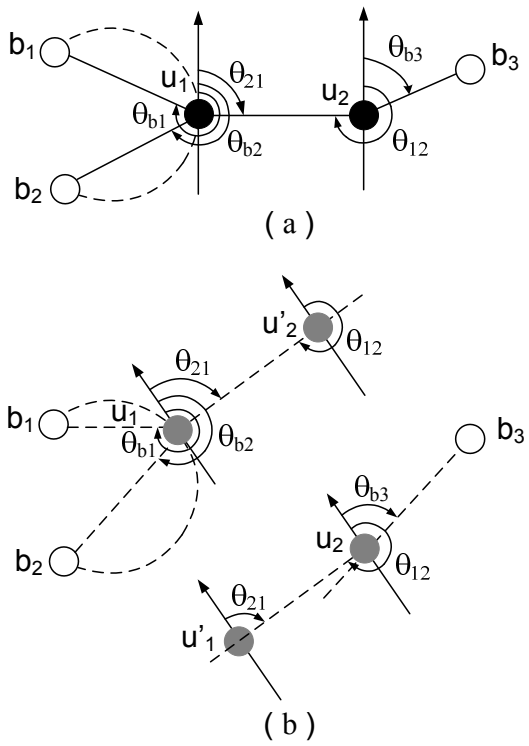


Figure 3.6: Localization when the orientations are unknown.

for u_2, the intersection also results to empty when its orientation is set to $\Delta\theta'$.

As stated in section 3.2.2, when the AOA measurements are not accurate and when the unknowns are aware of their orientations, for each unknown node, a probability distribution as in (2.14) can be generated by intersecting all probabilistic constraints from its neighbors. The location of this node is estimated at (x^*, y^*) as in (2.16). When unknowns are not aware of their orientations, a location estimate $(x^*_{\Delta\theta}, y^*_{\Delta\theta})$ of the estimation node is obtained for each assumed value of orientation $\Delta\theta$. The closer the assumed orientation to the true one, the higher probability of the intersection; reversely, the more different, the more likely that the intersection of all constraints will be empty or void. Therefore, for each assumed orientation of the unknown node, we record the probability of the estimated location $(x^*_{\Delta\theta}, y^*_{\Delta\theta})$:

$$\xi_{\Delta\theta} = P(x^*_{\Delta\theta}, y^*_{\Delta\theta}), \tag{3.13}$$

where $P(x^*_{\Delta\theta}, y^*_{\Delta\theta})$ is 0 when the intersection is empty.

The estimation of an unknown's orientation is

$$\Delta\theta^* = \arg\max_{\Delta\theta}\{\xi_{\Delta\theta}\}. \tag{3.14}$$

In Fig. 3.7, we rotate the orientations of u_1 and u_2 in Fig 3.6 from $-\pi$ to π with a step size of 0.0028π (1°). For each orientation, the estimation of $\xi_{\Delta\theta}$ is obtained. The normalized $\xi_{\Delta\theta}$ is plotted against the orientations in radian (only a small range of the orientations are shown, the rest are empty or zero). Orientations in $[-\pi, 0)$ and in $[\pi, 2\pi)$ have the same estimations, therefore we used orientations of $[-\pi, \pi)$ instead of $[0, 2\pi)$ to have the curve centering at zero.

In order to estimate the orientation, at each unknown, the pseudo code FIND-ORIENT(log,j) is executed.

After the orientation is determined, the position probability and the position estimation can be determined. The rotation step size of the orientation affects the performance of the localization. The smaller the step, the more accurate the orientation estimation. There is a clear tradeoff between the accuracy and the computational

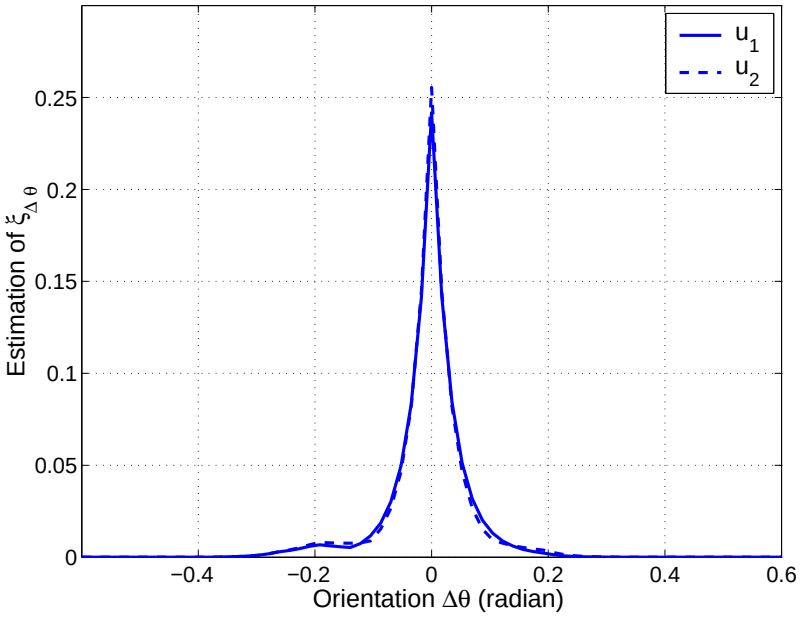


Figure 3.7: Orientation estimation for u_1 and u_2 in Fig. 3.6

Algorithm 3 FIND-ORIENT(log,j)

for $\Delta\theta \leftarrow 0$ **to** 2π **do**
  orientation of the estimation unknown $\leftarrow \Delta\theta$;
  for each unknown in the log **do**
    Calculate orientation using AOA pair relation and $\Delta\theta$;
  end for
  for each relative AOA in the log **do**
    Calculate the corresponding absolute AOA;
    Overwrite the relative AOA by absolute AOA;
  end for
  $P(x, y)$,$f_{X,Y}(x, y) \leftarrow$ Loc-OK(log,j);
  $\xi_{\Delta\theta} \leftarrow \max\{P(x, y)\}$;
  $\Delta\theta \leftarrow \Delta\theta + step$;
end for
orientation $\leftarrow \Delta\theta$ with maximum $\xi_{\Delta\theta}$;

cost.

3.3 Simulation Results

Same as in the RSS-based approach, we used a custom simulator implemented in Matlab. The results considered both the accuracy and precision. In order to evaluate the performance, we compared the results in most simulations with the APS+AOA-based localization approach proposed in [28]. For algorithm APS+AOA, we used the following techniques proposed in [28] to control the estimation error:

- A threshold of 0.35 ($\approx 20°$) is used to avoid the interference caused by small angles or degenerate triangles;
- outliers in the position estimations are eliminated to reduce the clustered errors.

We also compared those results with the CRLB in order to further evaluate the precision of the estimation.

3.3.1 Cramer-Rao Lower Bound

Considering M beacons and N unknowns in the network, let the vector of estimation parameters be

$$\gamma = [x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N, \Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_N], \tag{3.15}$$

in which x_i and y_i are the coordinate parameters and $\Delta\theta_i, i = 1, 2, \ldots, N$ is the orientation parameter. Since we assumed that the angle measurements at each unknown are Gaussian, if we also assume that all angle observations are independent at each unknown, the likelihood function can be simplified to

$$f(\theta;\gamma)=\prod_{j=1}^{N}\prod_{i\in H(j)}\frac{e^{-\frac{(\widetilde{\theta}_{i,j}-\mu(\theta_{i,j}))^2}{2\sigma_\theta^2}}}{\sqrt{2\pi}\sigma_\theta}, \tag{3.16}$$

where $i=1,2,\ldots,M+N$ and $\mu(\theta_{i,j})$ is the true AOA from i to j. If $\widetilde{\theta}_{i,j}$ is relative, $\mu(\theta_{i,j})$ consists of two parts: the absolute angle from i to j and the orientation of unknown j; otherwise, $\mu(\theta_{i,j})$ consists of only the absolute angle. Node $i\in H(j)$ means that i's beacon packet can reach node j. Thus, when the orientations are unknown, the $3N\times 3N$ Fisher Information Matrix (FIM) $I(\gamma)$ and the bound can be obtained as shown in [72]. When the orientations are known, the estimation vector γ consists of the first $2N$ parameters, and the elements in the FIM can be simplified to

$$I(\gamma)=\begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy}^T & I_{yy} \end{bmatrix}, \tag{3.17}$$

in which

$$I_{xx}(k,j)=\begin{cases} \sum_{i\in H(j)}\beta\frac{(y_i-y_j)^2}{d_{i,j}^4}, & k=j \\ -\beta\frac{(y_k-y_j)^2}{d_{i,j}^4}, & k\neq j, k\in H(j) \\ 0, & \text{otherwise} \end{cases} \tag{3.18}$$

$$I_{xy}(k,j)=\begin{cases} -\sum_{i\in H(j)}\beta\frac{(x_i-x_j)(y_i-y_j)}{d_{i,j}^4}, & k=j \\ \beta\frac{(x_k-x_j)(y_k-y_j)}{d_{i,j}^4}, & k\neq j, k\in H(j) \\ 0, & \text{otherwise} \end{cases} \tag{3.19}$$

$$I_{yy}(k,j)=\begin{cases} \sum_{i\in H(j)}\beta\frac{(x_i-x_j)^2}{d_{i,j}^4}, & k=j \\ -\beta\frac{(x_k-x_j)^2}{d_{i,j}^4}, & k\neq j, k\in H(j) \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

where $d_{i,j}$ is the distance between node i and j and constant β is inversely proportional to the variance of the angle measurement. Therefore the lower bound for the variance of each position estimation parameter is

$$\sigma^2_{\gamma(k)} \geq [I^{-1}(\gamma)]_{k,k}. \tag{3.21}$$

3.3.2 Performance Evaluation

For the simulation we considered 5 beacons and 50 unknowns randomly placed in a square area (of variable size depending on the desired density). Unknowns are not aware of their orientations before the estimation. We evaluated the effect of the variation of several parameters (number of hops the beacon packet is allowed to travel, number of beacons, number of unknowns and measurement inaccuracy) on both the estimation of position and orientation. Gaussian noise is added to the AOA measurements. Unless otherwise specified, we used a network density of 0.024 nodes/m^2 (average degree$\approx$10) and a standard deviation for AOA noise of $\frac{\pi}{18}$ (10°). To discover the orientation, we used a orientation rotation step of 4°. For every graph we present the average of 10 simulations with different network topologies. All results are normalized with respect to the maximum transmission range.

The effect of TTL for beacon packets

The maximum number of hops (time to live (TTL) in IP) of the beacon packets has a considerable impact on the performance of the proposed algorithm. To evaluate this impact, we varied the TTL from 1 to 10. The position estimation accuracy is plotted in Fig. 3.8 against the TTL. In the figure, the solid lines stand for the accuracies of the estimations without the orientation information, while the dashed lines stand for the estimations with the orientation information. In addition, different markers represent different network densities including 0.016, 0.02, 0.024 and 0.028, which correspond to average degrees of 6, 8, 10 and 12, respectively. Fig. 3.9 shows the percentage of the nodes that can discover their orientations and positions when the orientations are unaware.

Since most of the nodes in the network cannot hear directly from enough beacons to perform triangulation, they rely on the beacon packets that are generated from

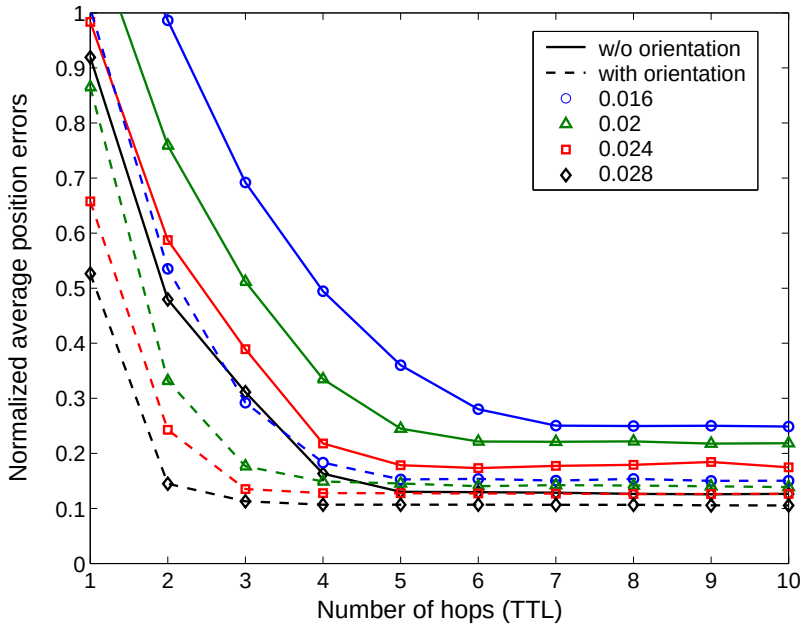


Figure 3.8: Average accuracy as a function of the TTL for beacon packets for different node density.

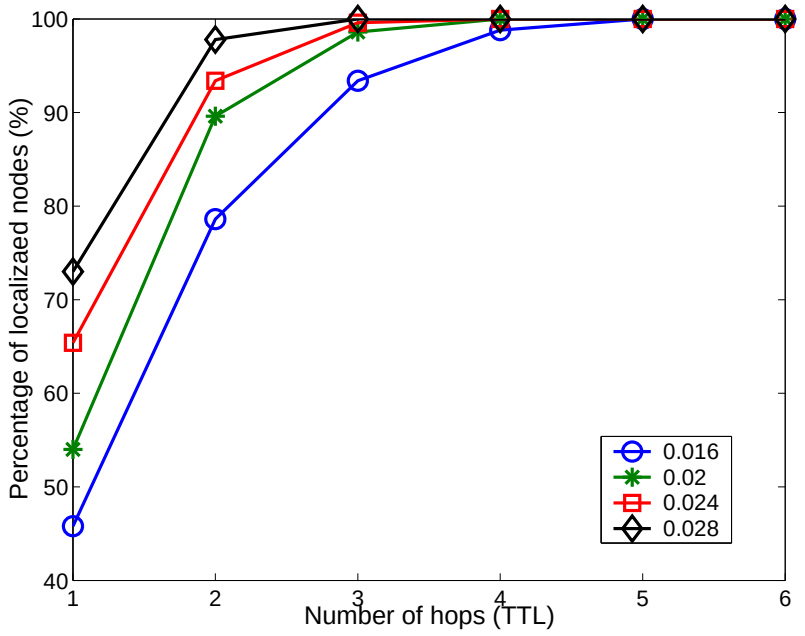


Figure 3.9: Percentage of nodes that can be oriented and localized for different node density.

beacons several hops away and relayed by other unknowns. Figures also show that, regardless of the network density, the accuracy of the localization improves along with the TTL for the first several hops. This is not surprising since the larger the TTL is allowed, the more nodes can be heard from by an unknown and the more angle probability constraints can be placed on this unknown. However, the improvement is not significant after $5-6$ hops. This can be caused by either the unknown has heard from all beacons in the network or the contributions from distant beacons are too weak to make any improvement.

The beacon packets become lager as they travel through the network. Lager packets require more computation resource for processing. Furthermore, larger TTLs cause more traffic in the network. In what follows, we used TTL of 4 (which reaches 100% coverage for density of 0.024 nodes/m^2) for all simulations (both for the probabilistic and APS+AOA approach).

The results shown in Figs. 3.8 and 3.9 also describes the varying of the accuracy along with the average number of neighbors for a node (average node degree). The higher the density, the more neighbors that a node is adjacent to on the average and hence the higher accuracy it can achieve.

The effect of number of beacons

We varied the number of beacons in the network from 3 to 8. The estimations of the positions with and without the orientation information are shown in Figs. 3.10 and 3.11, respectively. From the figures, both the accuracy (average error in solid lines) and the precision (average standard deviation of both x and y coordinates in dashed lines) of the position estimate improve along with the increasing of the beacon number. However, unlike the APS+AOA approach, the improvements for the proposed approach are not significant (especially when number of beacons is greater than 5), which is also indicated by the CRLB.

In Fig. 3.12, the orientation estimate is given. The percentages of the nodes that can be localized and oriented are also marked for each number of beacons. Our probabilistic approach achieves 100% coverage for all numbers of beacons. Again,

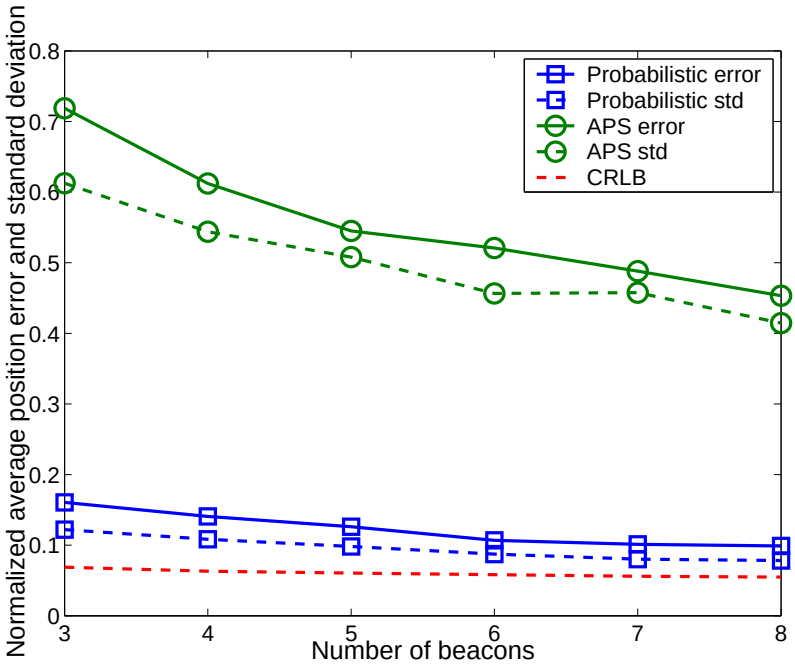


Figure 3.10: Average accuracy and precision as a function of the number of beacons when the orientations are known.

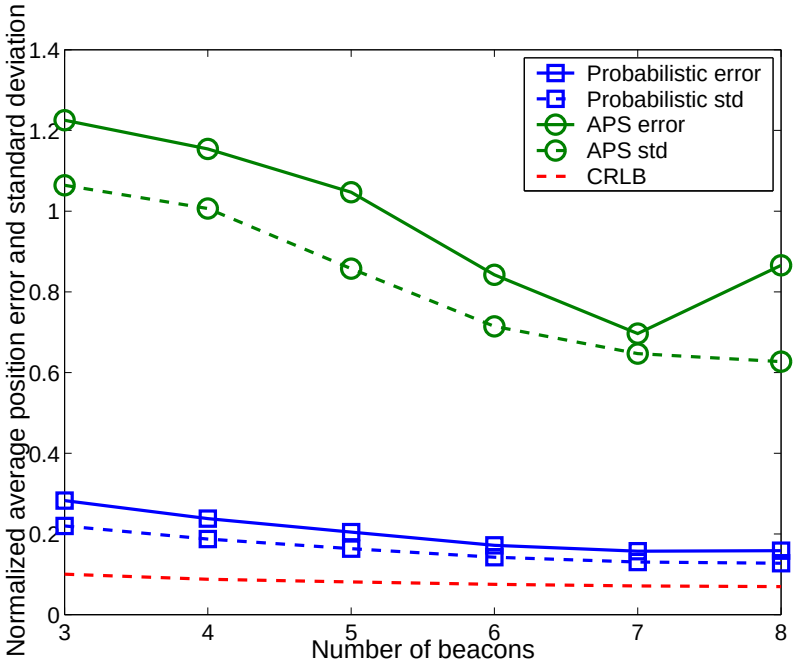


Figure 3.11: Average accuracy and precision as a function of the number of beacons when the orientations are unknown

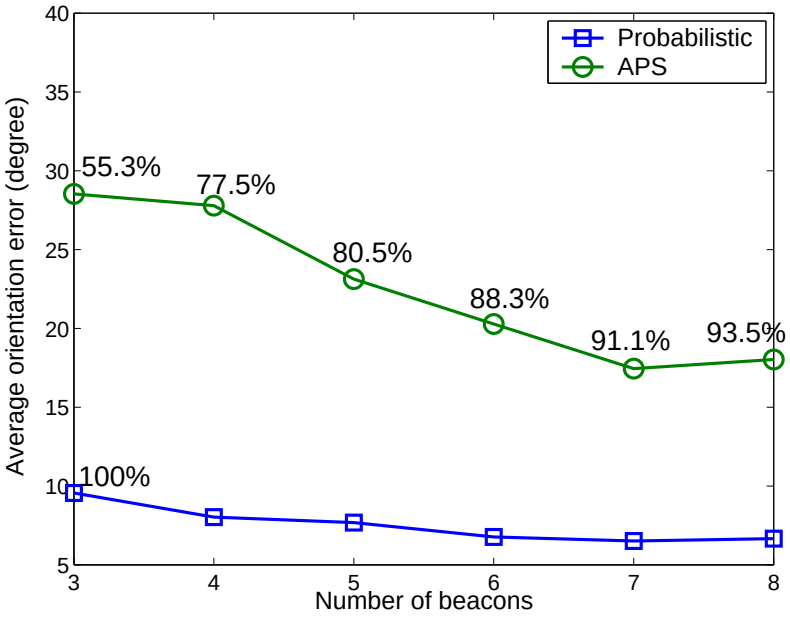


Figure 3.12: Average orientation error (and the number of located node) as a function of the number of beacons.

the improvement on the accuracy of the orientation estimate for our approach is not as significant as APS+AOA. This suggests that even a low number of beacons can achieve good accuracy, precision, and coverage.

The effect of number of unknowns

In this scenario, we changed the number of unknowns from 30 to 80 while keeping the density constant. The simulation results are shown in Figs. 3.13, 3.14 and 3.15. The estimation errors for both position and orientation increase slightly with the increase of the network size. Intuitively, since the network density is constant, the larger the network size, the fewer unknowns can hear directly from the beacons. Thus, localization for most nodes rely on relayed beacon packets with larger TTL. Since we limited the TTL to 4, the average number of beacons that an unknown can hear indirectly decreases for larger networks, which causes the slight increase of the estimation error. For the estimation precision, the CRLB indicates that the increasing of the number of unknowns does not have a dramatic impact on the precision. As

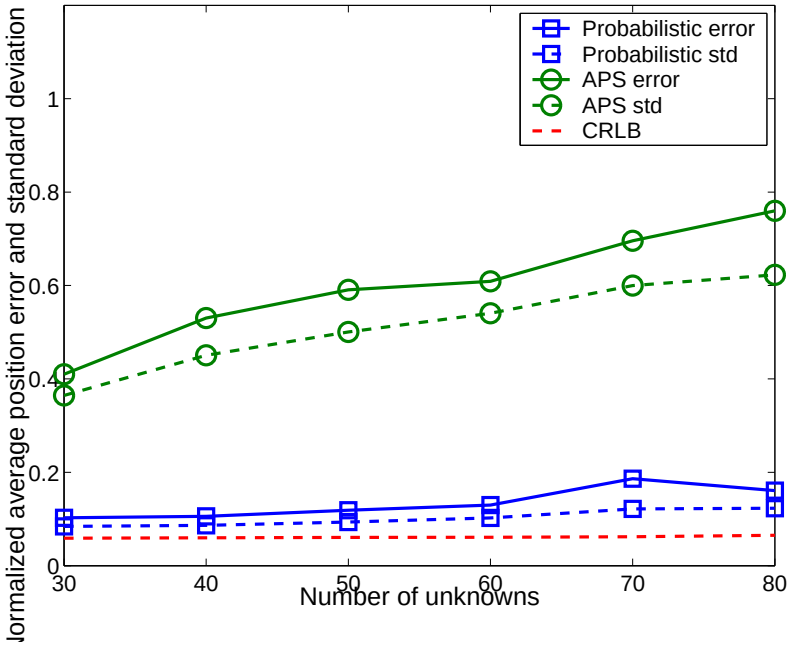


Figure 3.13: Average accuracy and precision as a function of the number of unknowns when the orientations are known.

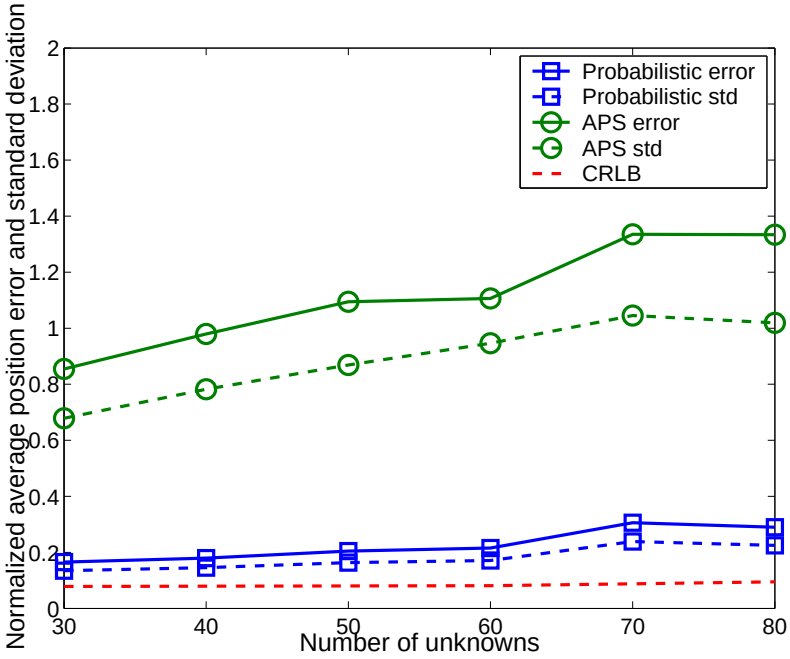


Figure 3.14: Average accuracy and precision as a function of the number of unknowns when the orientations are unknown.

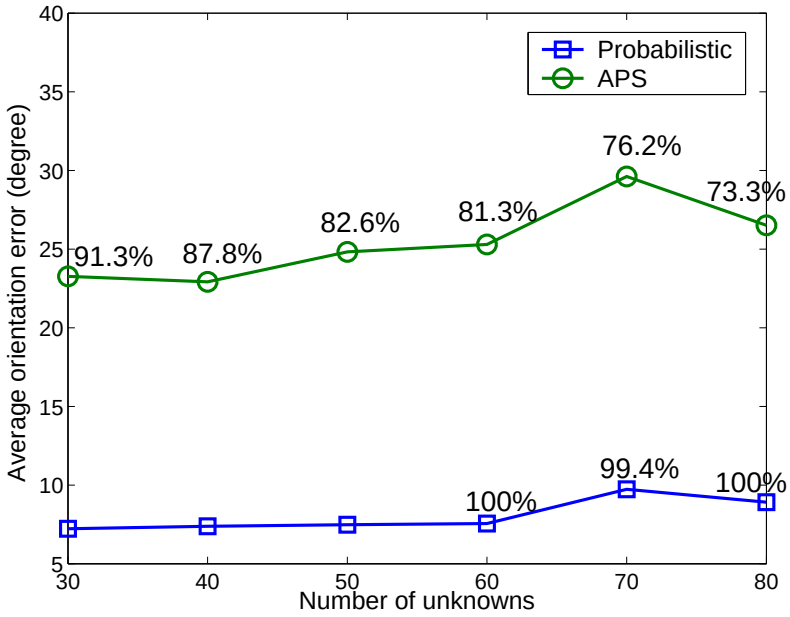


Figure 3.15: Average orientation error (and the number of located node) as a function of the number of unknowns.

shown in equations (3.18,3.19,3.20), the CRLB is more sensitive to the number of neighbors and the noise level compared with the number of beacons and unknowns, therefore the constant number of neighbors and noise level lead to the constancy in the precision.

The effect of measurement noise

A good localization scheme must be robust to measurement errors. We varied the AOA measurement error from 1° to 25°. The localization and the orientation results are shown in Figs. 3.16, 3.17 and 3.18. The presence of AOA measurement noise has a significant impact on localization and orientation. However, the probabilistic approach behaves far more robustly than the APS+AOA scheme in accuracy. In addition, its precision is much closer to the optimum bound; and it achieves much better coverage especially when the noise is large.

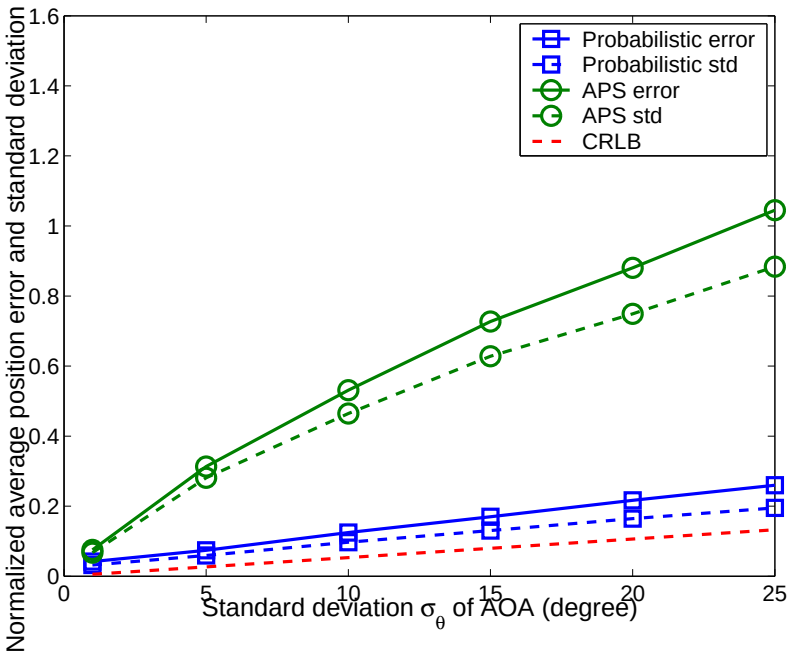


Figure 3.16: Average accuracy and precision as a function of the number of AOA noise when the orientations are known.

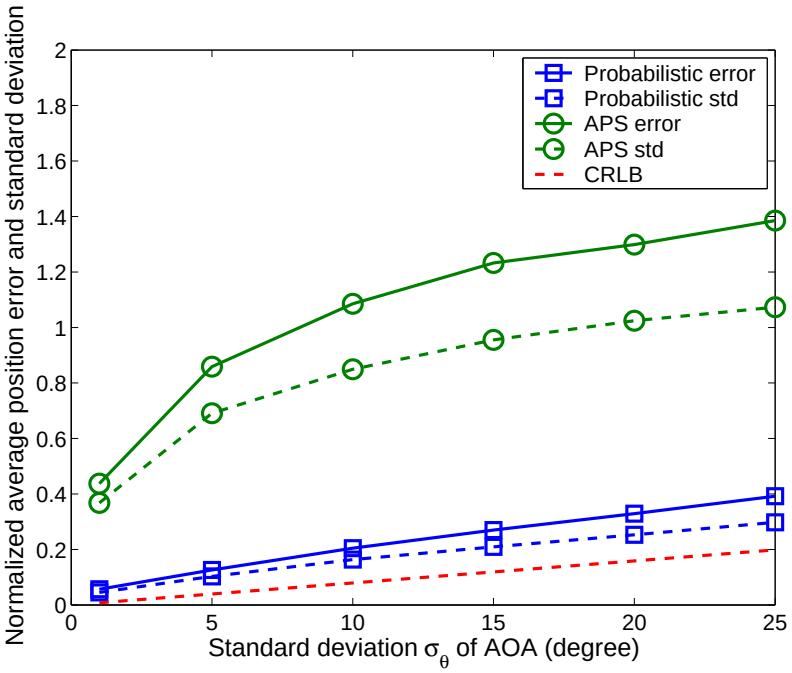


Figure 3.17: Average accuracy and precision as a function of the number of AOA noise when the orientations are unknown.

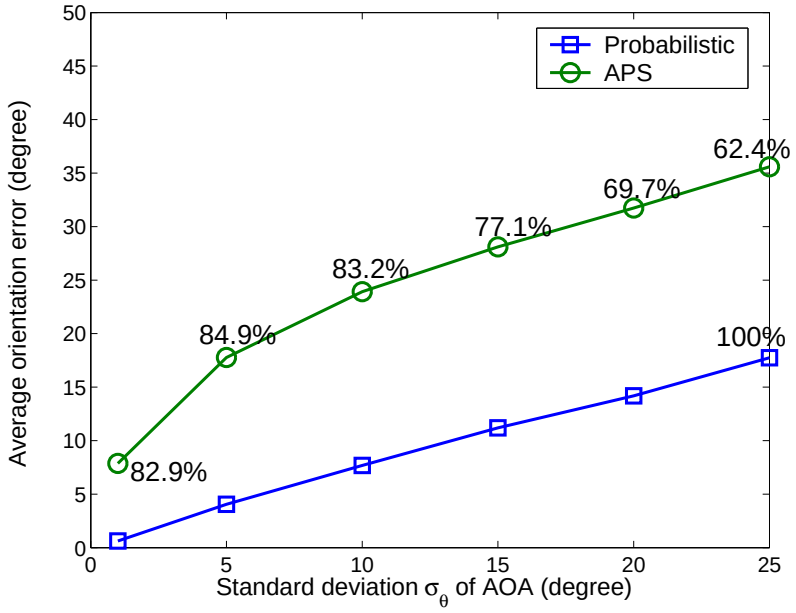


Figure 3.18: Average orientation error (and the number of located node) as a function of the number of AOA nosie.

3.4 Conclusion

In this chapter, we presented a distributed AOA-based localization and orientation approach for wireless stationary sensor networks under the assumption that all sensor nodes are capable of detecting angles of the incident signal from the neighboring nodes. The simulation results show that even with inaccurate AOA measurements and a small number of beacons, the proposed approach exhibits excellent accuracy and precision for the estimation, and achieve much better localization coverage than existing approaches.

Notice that though we did not apply negative constraints to the angle-based localization approach, the degree of the accuracy of the position estimates are similar to the RSS-based approach. This is caused mainly by two reasons. First, the number of beacons required in the angle-based localization is less than RSS-based. Therefore, given the similar number of beacons (6 in RSS-based and 5 in angle-based), the angle-based estimation could achieve higher level of accuracy. Second, compared to the positive constraints generated by the distance pdf, the ones generated by the

angle pdf are usually more constrained (pointing), especially when the RSS is small which results in large distance standard deviation. The most intuitive example is the positive constraint generated by a beacon. While the positive constraint calculated using the distance pdf has a doughnut shape peaking at the measured distance as shown in Fig. 2.6(b), the one calculated by angle pdf is a sector peaking at the measured angle and restrained within the maximum transmission range as shown in Figs. 3.4(a) and (b). The isotropic property of the distance constraint increase the uncertainty of the probabilistic constraint, and thus less accurate position estimate. The negative constraint generated using a beacon's position information can significantly decrease the isotropic effect of the positive constraints for unknowns that are more than one hop away from that beacon. Nevertheless, negative constraint can also be applied to the angle-based localization to further improve the estimation accuracy and precision.

It is also not hard to see that the RSS-based and angle-based localization approaches can be combined to provide better accuracy. This can be done by recording both RSSs and AOAs between neighboring nodes, and using both measurements to calculate the probabilistic position constraint. Better accuracy is achieved because the position distribution is constrained on both the distance and the angle.

Chapter 4

Generalized Trace-based Mobility Generator

In this chapter, we present the mobility modeling and mobility trace generation for MANETs. Since our proposed mobility generator is driven by the real traces (i.e., the statistical properties are extracted from the real world traces and used to regenerate the synthetic traces), the discussion in this chapter is divided into collecting real traces, exacting and analyzing statistical properties in those traces and regenerating synthetic traces.

4.1 Trace Data

We focus on three sets of traces that are publicly available: two campus traces (Dartmouth and UCSD) and one bus trace containing location data collected by GPS devices installed in the buses. Details of theses traces are listed in Table 4.1.

The first set of trace is the Dartmouth campus trace. This is a set of traces collected from over 450 wireless access points (APs) on the Dartmouth campus [73].

Table 4.1: Trace summary.

Traces	Analyzed / Total Nodes	Duration	Analyzed Part
Dartmouth	5000 / 13888	04/01 - 06/04	01/03
USCD	275 / 275	09/05 - 12/05	10/05
Seattle	1293 / 1293	11/01 - 12/01	11/01

The original trace records log files of syslog events, SNMP polls and tcpdump data. The Dartmouth campus is very compact in shape, and the wireless network coverage extends from buildings to almost all campus areas, enabling clients to roam seamlessly between the APs. To recover the actual user movement from these events, we used the triangle centroid method introduced in [49].

The second set of trace is the UCSD campus trace [54], which was collected through 275 HP Jornada PDAs distributed to the first year students in UCSD. Each of these PDAs is equipped with an 802.11 card and special software running in the background that collects all the WiFi access records. Different from most of the laptop users in the Dartmouth trace, the UCSD trace, collecting the activity of PDA users, shows more mobility. However, in this trace there are far fewer nodes than in the other traces, and for some reason, the user activity deteriorates as time progresses, so we primarily use the first half of the trace.

The third set of trace is the Seattle bus GPS trace collected from the Ad-Hoc City project [57]. GPS receivers are deployed on more than 1200 buses running in the Seattle metropolitan area, and for most records, the location update period is 12 minutes. This set of traces is very good for studying the movement of mobile users for three reasons: first it contains a sufficient amount of users (more than 1200); second, it covers long duration (more than 20 days); finally, it provides precise position information (accurate GPS position) with relatively frequent updates.

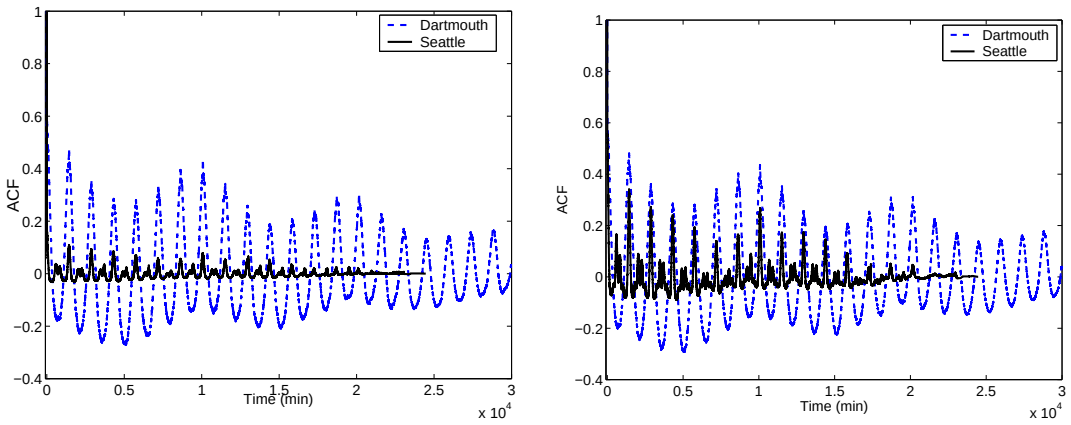
4.2 Mobility Pattern Analysis

The mobility patterns in a given trace can be characterized by many features. According to different types of mobility models used to generate the synthetic trace, a different number and type of features are used. For example, in the RWP model, only several coarse features including the dimensions of the deployment area and the range of the speed and of the pause time are considered. Such a limited model cannot reproduce realistic behaviors of mobile nodes. In contrast, in both trace based mobility models and detailed simulators, the features considered are very detailed (e.g., including such as the daily schedule and particular routes of each node). We believe that the more detailed features of the model considered, the more realistic the generated mobility pattern can be; however, since a lot of the detailed features are specific to a scenario, such models lack the possibility of diversification to other scenarios.

Different features generally have different level of significance on the performance of MANET protocols. In order to generate both a realistic and diversified model (i.e., between the very simple stochastic models and the very detailed trace based models or detailed simulators), we need to identify the features that have higher impacts on the performance of the networking protocols. Therefore, in the following section, we extracted and analyzed several macro-leveled mobility features for two WLAN traces and one bus trace. We also compared the features according to different population sizes and different scenarios, the results of this comparison can be used to provide diversification without compromising realism.

4.2.1 Arrival and Departure Processes

For a fixed simulation area, node density is directly related to the total number of nodes, and for a fixed transmission range, more nodes result in a higher average node degree, which in turn has a great influence on many protocols (including MAC and routing protocols). In contrast to the fixed number of nodes in RWP, in reality, the number of the nodes in the scenario varies with time. In addition, the identity of nodes

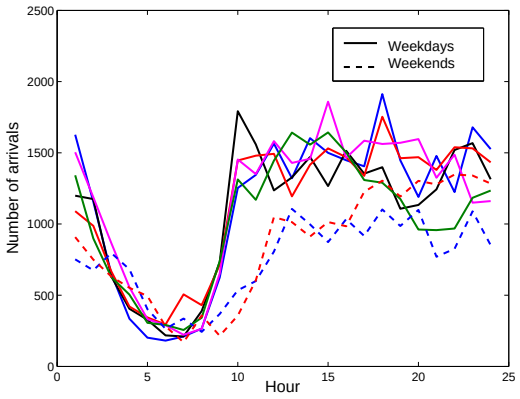


(a) Arrival processes (b) Departure processes

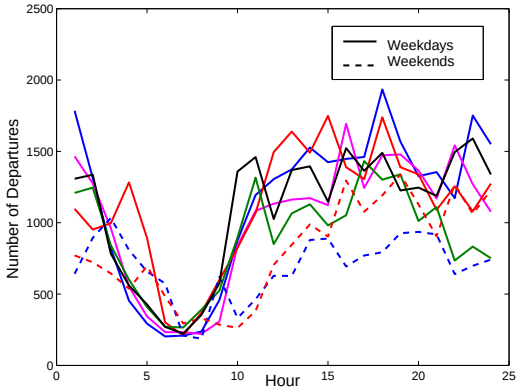
Figure 4.1: ACFs of arrival and departure processes.

in the scenario is also important for some networks, (e.g., for Delay Tolerant Networks (DTNs)). In order to emulate the fluctuation of the number and combination of nodes, it is both easier and more accurate to emulate the arrival and departure patterns of the mobile nodes. As the first step, we counted the hourly number of arrivals/departures at the scenarios for all three traces. If a node left the scenario (for example a user turned off the mobile device) and returned later, or if a node is disconnected and reconnected, we count the re-entering or the reconnection as a new arrival.

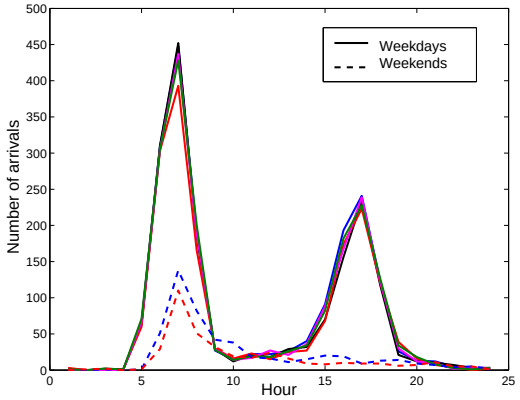
Intuitively, a node's arrival and departure pattern should follow a certain routine, for example, students usually go to school in the morning and leave in the afternoon during weekdays. In order to discover such routine, we calculated the autocorrelation function (ACF) of both arrival and departure processes for all traces. Not surprisingly, all ACFs show that both the arrival and departure processes are periodic. In Fig. 4.1, the ACFs of both arrival and departure processes for the Dartmouth and the Seattle traces are shown. From the figures, both arrival and departure curves reveal two main periodic trends: weekly and daily. If consider the traces for one week and plot the hourly arrivals in each day as in Fig. 4.2, the daily period can be further clarified. However, the daily patterns for weekdays and weekends are different: first, the number of arrivals and departures for both school and city bus scenarios decrease during the weekend; second, the repeating patterns are different between weekdays



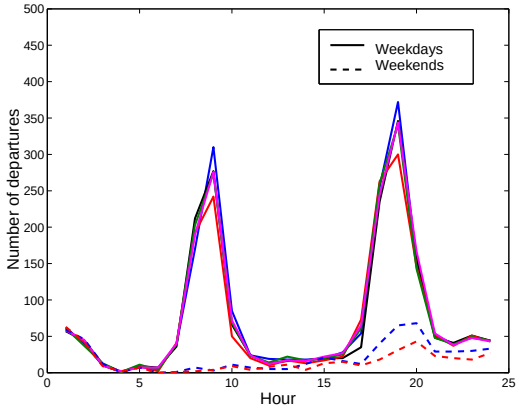
(a) Dartmouth arrival process



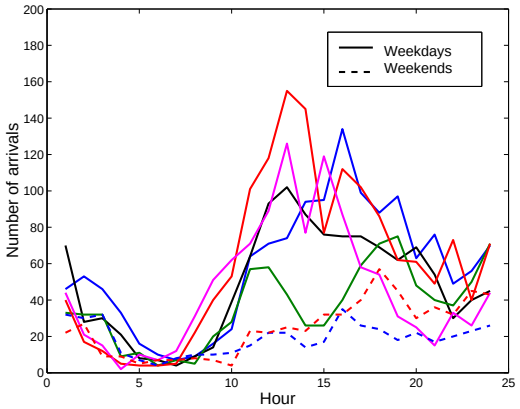
(b) Dartmouth departure process



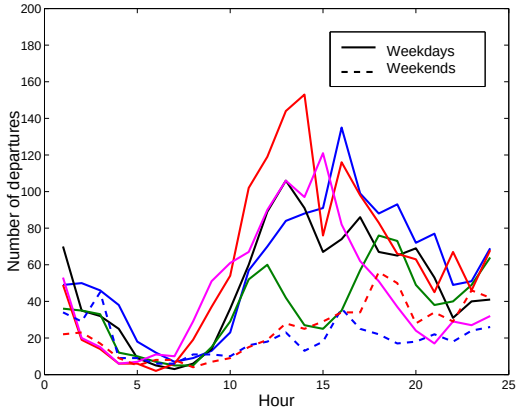
(c) Seattle arrival process



(d) Seattle departure process



(e) UCSD arrival process



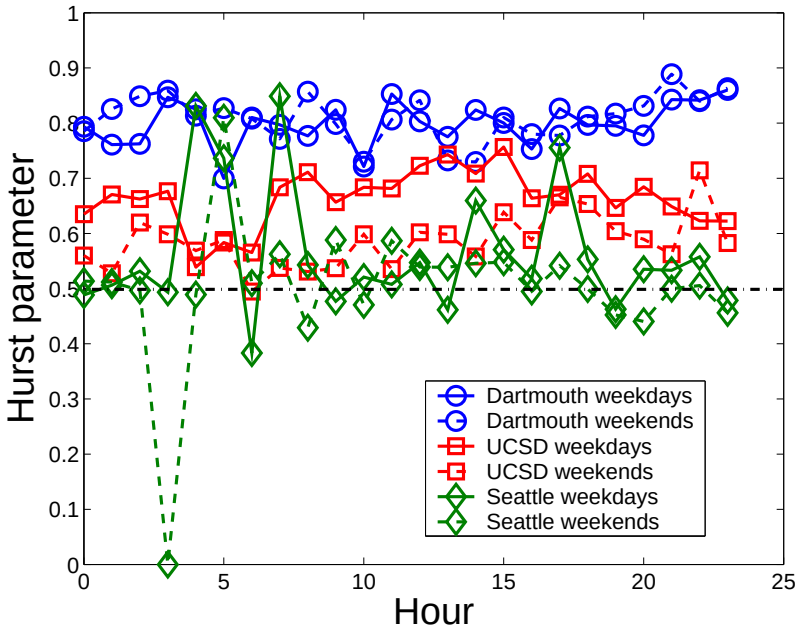
(f) UCSD departure process

Figure 4.2: The number of arrivals and departures for a week as a function of the time of the day.

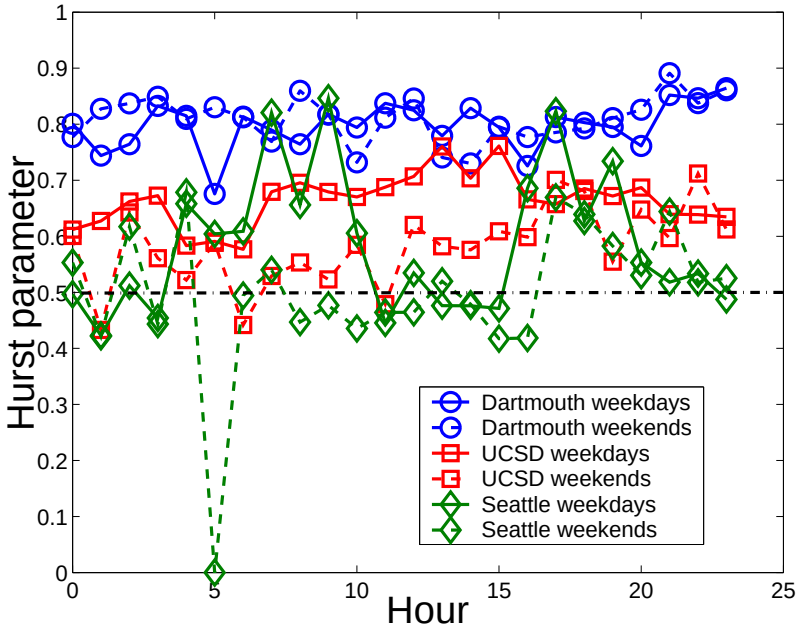
and weekends. For example, in the Dartmouth trace, the start of the "busy" hour is approximately 2.5 hour late for the weekends than for the weekdays; in the Seattle trace, the arrival "peak" around 5pm and the departure "peak" around 9am appearing in the weekdays are missing for the weekends; furthermore, in the UCSD traces, while there exists clear indication of the start of the "busy" hours for weekdays, no such indication exists in weekends. Therefore, for simplicity and accuracy we will study and generate the arrival and departure process daily, but separately for weekdays and weekends.

Considering that the arrival and the departure patterns are repeating daily, we aggregate the arrival and departure processes for weekdays and weekends separately, compute the average arrivals and departure numbers for each hour, and normalize them by the total number of nodes in the scenario. Figure 4.2 indicates that the normalized arrival and departure rates should be time-varying. Therefore, if we assume the arrival and departure processes for each hour are stationary, the generated synthetic arrival and departure processes for all traces should have hourly-varying rates. Another interesting observation in Fig. 4.1 is the different tail behaviors of the ACFs. The Seattle ACF tail decreases much faster than the Dartmouth ACF tail, especially for the arrival processes. Since any process with long-range dependency has a slowly (hyperbolic) decaying ACF [74, 75], which implies the *self-similarity* in the process, an interesting question here is if the arrival and departure processes are self-similar?

In order to investigate the presence of self-similarity, we estimated the Hurst parameter H, (which is a parameter to indicate the degree of self similarity in a process,) for each hour's arrival and departure processes. The Hurst parameter of a process with self similarity takes value in the interval $(0.5, 1)$. The closer the H to 1, the higher the degree of self similarity [74]. In this paper, we used two graphical methods including variance-time plot and rescaled range (R/S) plot to estimate H. The average of the two estimates is used as the self similarity indication and is shown in Fig. 4.3. Clearly, different traces behave differently. In the Dartmouth trace, all processes show strong self similarity consistently. In UCSD trace, while the values of the Hurst parameter during weekdays indicate self similarity, part of the



(a) Arrival processes



(b) Departure processes

Figure 4.3: Hourly Hurst parameter estimation.

weekends' parameter do not show a significant difference from 0.5. In the Seattle trace, no process behaves consistently. Notice that in the Seattle trace, both arrival and departure have a sample of 0, meaning that no node arrived or departed within that hour. However, for all traces, the self similarities are time-varying. Therefore, in order to generate accurate synthetic traces, not only the hourly rate should be different, the processes used to generate arrival/departure in an hour should also be different according to the levels of self similarity.

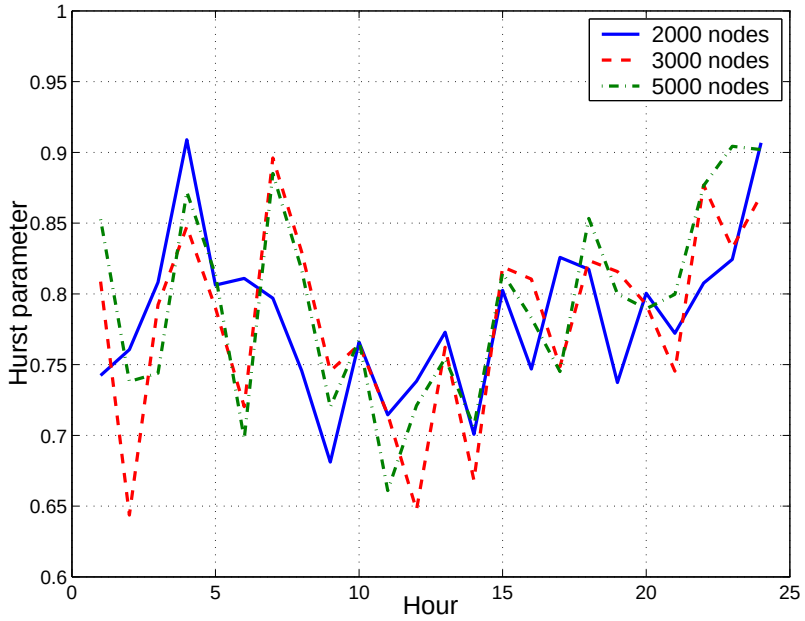


Figure 4.4: Hurst parameter estimations with different population sizes for the Dartmouth trace.

We also analyzed the arrival and departure processes using different population sizes. As mentioned before, to realize diversification, it is important to reuse the statistics extracted for some population size to generate synthetic traces with other population sizes. This requires that the traces with different population sizes possess similar statistical features. We call this similarity (with respect to the number of mobile nodes) *population similarity*. In order to check for the existence of population similarity, we randomly truncated the original traces into smaller traces with different population sizes and estimated their Hurst parameters. The estimation results show

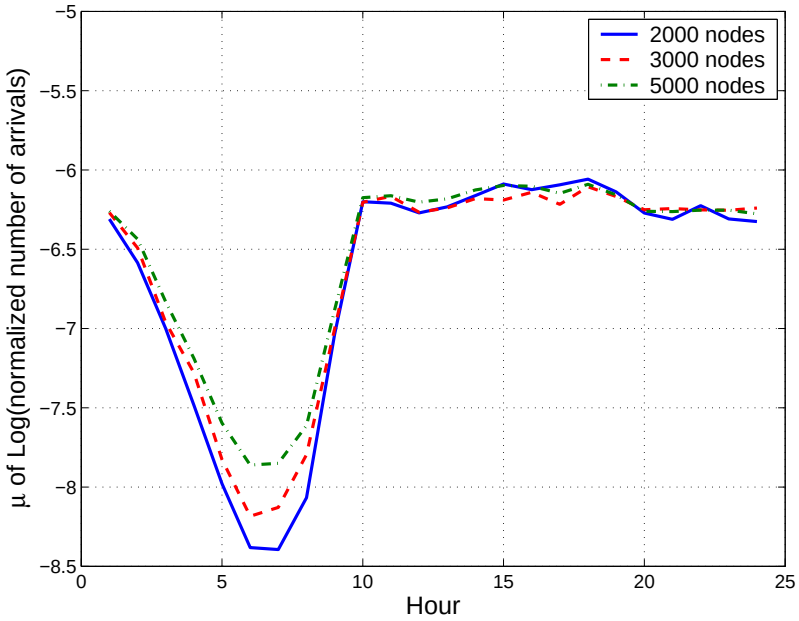


Figure 4.5: Average of normalized number of arrivals for the Dartmouth trace.

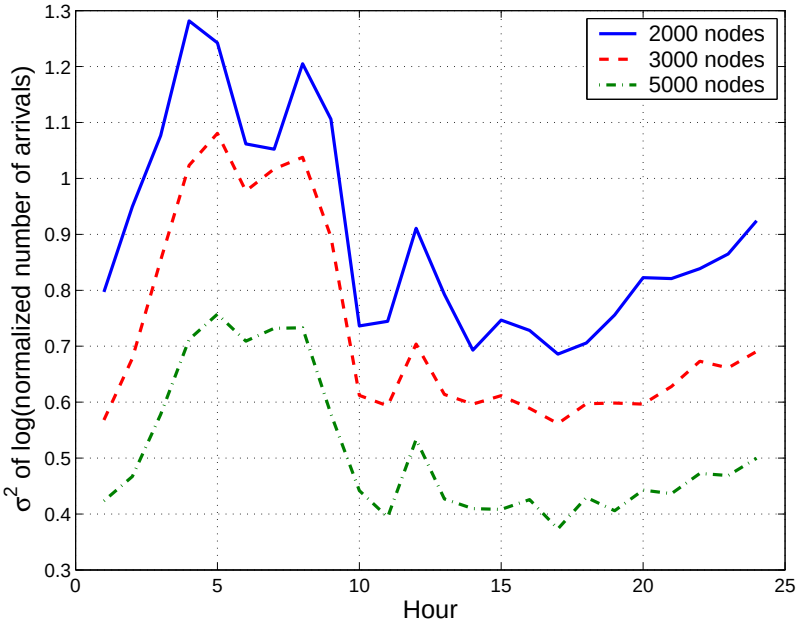


Figure 4.6: Standard deviation of normalized number of arrivals for the Dartmouth trace.

that, while the exact values of Hurst parameter estimations are not exactly the same within each hour, the estimations are very similar and consistent as shown in Fig. 4.4. This means if the Hurst parameter of a population size indicates that a certain hour's process is self-similar, the same result will be obtained with another population size. We estimated the mean μ and the variance σ^2 of the logarithm of the number of arrivals within an hour normalized by the number of nodes and we obtained the following results:

$$\mu_M \approx \mu_N, \tag{4.1}$$

$$\sigma_M^2 \approx \frac{N}{M}\sigma_N^2, \tag{4.2}$$

where both M and N are population sizes. Thus, while the mean value for each hour, given different population sizes, has only a slight change, the variance is inversely proportional to the population size, as shown in Fig. 4.5 and 4.6. The logarithm is taken for better visibility. Intuitively, if we consider the number of arrivals/departures of each single node as one of the independent, identically distributed (IID) random variable $\{X_i, i = 1, \ldots, N\}$, recall that the mean of the normalized sum $\frac{1}{N}\sum_{i=1}^{N} X_i$ is invariant with the number of variables N but the variance of it is inversely proportional to N, which generates the same result as in (4.1) and (4.2). Therefore, with the population self-similarity, we can use equations (4.1) and (4.2) to obtained arrival/departure distributions for different population sizes.

4.2.2 Speed

Speeds at which nodes travel within the network is critical to the performance of many networking protocols. In many scenarios, higher speed implies a higher rate of link changes. In RWP and most stochastic models, constant speed [39, 40, 76] or simple distributions such as uniform [36, 47] and normal [77] are used to characterize the speed values. In [41, 42], it was mathematically shown that, when speeds are selected from a uniform distribution with low minimum speed, the mean speed approaches zero over time, which creates a nearly stable network topology that results

in unrealistically good performance. In our observation of real traces, we concluded that simple distributions lack realism. For example, in the campus traces, within the range of speeds, most mobile users prefer to move with relatively slow speeds, which denies the suitability of the uniform distribution.

A real trace records the nodes' trajectories over time. To measure the nodes' speeds, the trajectories are translated into times and locations, the speed can be simply calculated by dividing the location difference of two consecutive recordings by the time difference between them. However, this method may sometimes causes very small or very large speeds that cannot be realistic. Such erroneous speeds are most likely caused by inaccurate data samples. The very small speeds may also be caused by the pause at intermediate positions while moving from a location to another one, while the large speeds may be caused by immediate changing associations in WLAN. In order to reduce the effects of these factors, we impose range of speeds $[v_{min}, v_{max}]$. For all speeds smaller than v_{min}, we extract the pause time and estimate the real speed in that duration according to the method in [49]; for all speeds larger than v_{max}, we simply ignore that movement.

We assume that the speed distribution is stationary. Such assumption is reasonable because, even though the moving speed of a single node can be variant from time to time, on a larger scale, the distribution of the moving speeds of a single node is stationary. Figure 4.7 depicts the probability distribution functions (pdfs) of the speeds in the three scenarios we consider. As expected, the distributions in the school scenarios are quite similar, with low speeds dominating the distribution. The figure also shows that the distribution of the speeds is invariant with respect to the the number of nodes.

4.2.3 Spatial Distribution

In stochastic models, the long term spatial distribution of nodes over the simulation area is invariant for a given model. In RWP for example, nodes tend to concentrate around the center of the simulation area [42]. On the other hand, although real world node spatial distributions may vary from scenario to scenario, their

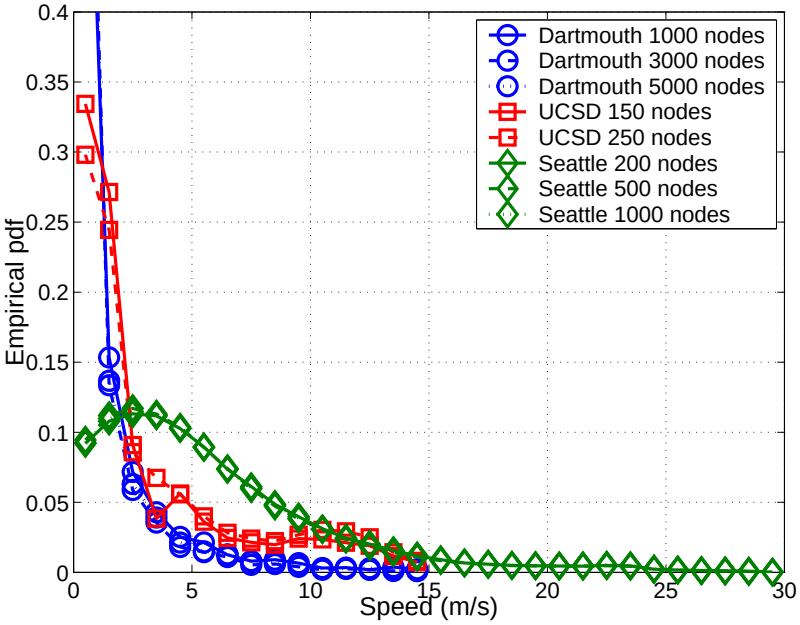


Figure 4.7: Probability distribution function of the speed.

long-term distributions tend to be stationary because nodes in real world scenarios are usually attracted to particular destinations with important semantic meanings, such as a classroom, a library, etc. In this context, assuming a uniform node distribution can lead to unrealistic results. The long term distribution of the nodes over the simulation area affects the connectivity of the topology graph and hence the routing performance, since intuitively, locations with higher density of nodes have heavier traffic.

We define a *hotspot* as a region where many location measurement samples are clustered together in a trace. A hotspot could either be a region which nodes visit frequently or a region where nodes pause for long durations. Clustering methods [78, 79] can be used to classify measurement samples into different groups with a predefined number of groups. However, the number of groups is hard to obtain from a real trace without knowing any topology information of the deployment scenario. Furthermore, the resulting groups usually depend on the initial random assignments in the clustering methods. There are several clustering methods that can define the number of groups automatically, but they are usually computationally expensive [80]

especially for large real traces. Other trace based hotspot extracting methods often require predetermined conditions. For example, in [49] the accuracy of the position estimate is required.

In order to reduce the complexity and increase the generality, we divide the whole scenario into small square grids and count the location measurement samples in each grid. Each sample is weighted by its pause time at that location. In Fig. 4.8, we ordered the grids in a descending sequence according to their density and plotted the affects on number of measurements and the area of the grids as we add each hotspot (in the order of its density starting with the most dense). Not surprisingly, we found that most location measurements are clustered in a small percentage of the total area. Figure 4.8 shows that for all traces 80% of the measurements are distributed in less than 22% of the area, which provides a strong evidence of the existence of highly clustered grids. Therefore, for each trace there are two kinds of grids, highly dense grids occupy only small area (especially for WLAN traces) and low density grids occupy most of the area. To extract the hotspots, we can simply removed the low density grids. In this paper, we used a $slope \leq 1$ criterion. Notice that when $slope > 1$, it identifies the points where adding new grids increase the area more than the population. Therefore, by finding the point where the slope of the curve equals to 1 and discarding all measurements that are on the right side of that point, the hotspots can be found. In a second step we merge all grids that are adjacent into a single cluster and the resulting clusters are the hotspots.

The size of the grid can affect the number and the area of the hotspot. While grids that are too small can separate a large hotspot into several smaller hotspots, grids that are too large can result in hotspots larger than in reality. In this paper, we used a grid size that is related to the scenario area. Assume the bounding coordinates of a scenario is $x_{min}, x_{max}, y_{min}$ and y_{max} and a grid is $l \times l$, the length l of the grid is:

$$l = 10^{\lfloor \log_{10} \alpha \rfloor - 1} \tag{4.3}$$

where $\alpha = \min\{x_{max} - x_{min}, y_{max} - y_{min}\}$. After finding the hotspots, we define the

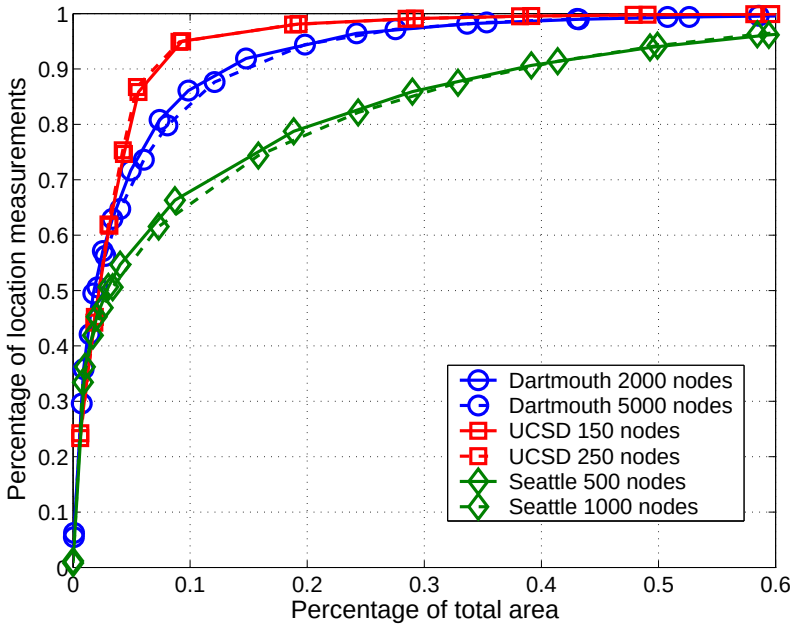
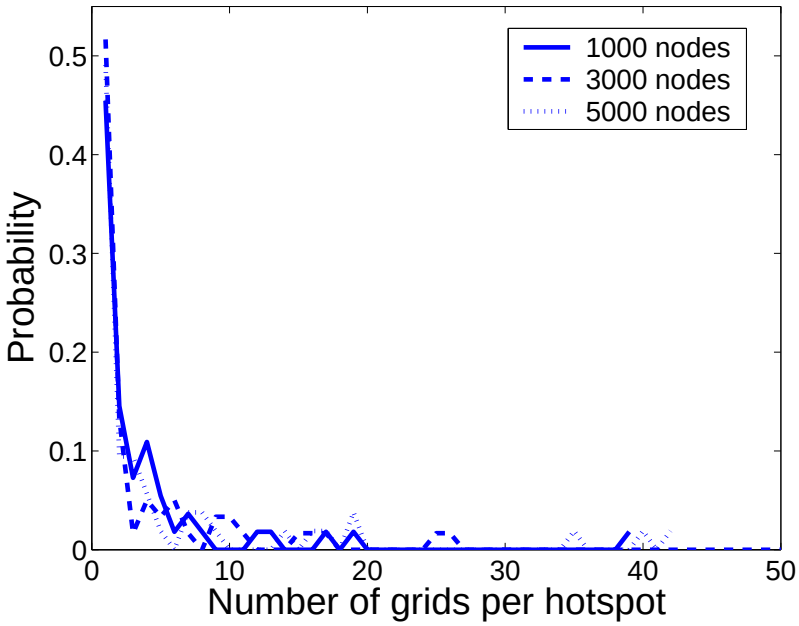


Figure 4.8: Population and area distribution - most traces are clustered in highly popular locations we call hotspots.

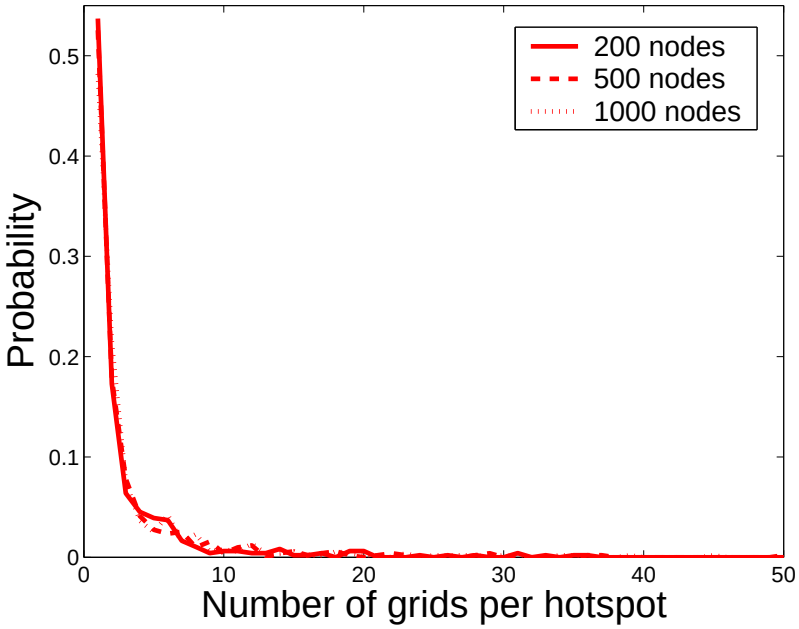
several parameters that characterize these hotspots.

Area

The first such parameter is the area of a hotspot. Using the grid method above, the area can be simply calculated as the number of grids multiplied by the area of a grid. Figure 4.9, shows the distribution of the number of grids per hotspot for the Dartmouth and the Seattle traces for different number of nodes. We did not consider the hotspot area distribution for the UCSD trace because the number of hotspots of that trace is too small to result in meaningful statistics. Two conclusions can be drawn from Fig. 4.9: first, the population size does not significantly affect the area distribution of the hotspots, and, second, for both WLAN and bus scenarios, small area hotspots are more likely than large areas. In the Seattle trace, in addition to the results shown in the figure, there is also a very large hotspot (more than 100 grids) for all population sizes (probably the bus depot?). However, to improve the better visibility of the figure, we truncated the y-axis at about 50 grids without affecting



(a) Dartmouth



(b) Seattle

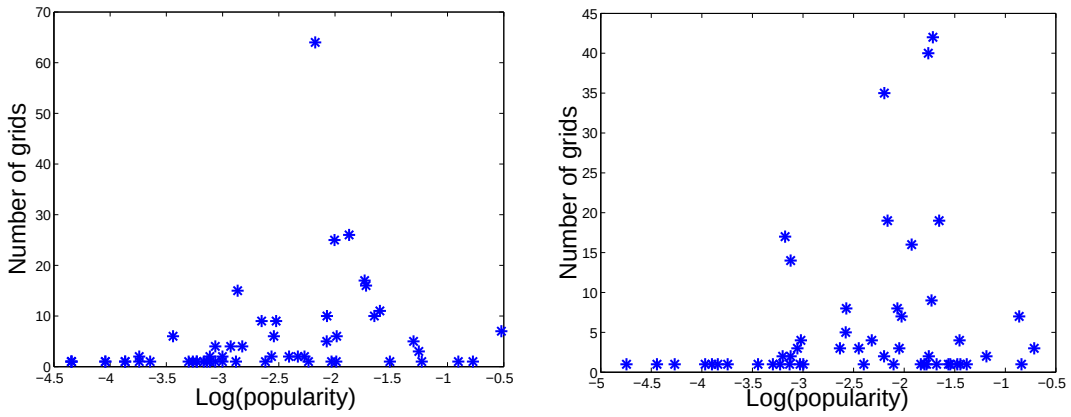
Figure 4.9: Distribution of number of grids per hotspot.

the conclusion on population similarity.

Popularity

We define the *popularity* of a hotspot as the number of arrivals at each hotspot normalized by the total number of arrivals at all hotspots, i.e., the larger the number of arrivals at a hotspot, the more popular it is. We first examine the distribution of area given as a function of popularity; the results are shown in Fig. 4.10. Due to the small number of hotspots, we plot one star for each hotspot according to its size and popularity. The figure shows that the most popular and least popular hotspots are small in Dartmouth, while the most popular hotspot in Seattle is also the largest. For medium popularity hotspots, there is a large variation in size. The population similarity is readily visible with the popular hotspots remaining relatively as popular as they are regardless of the number of mobile nodes.

We also examined the relationship between the the number of transitions between hotspots and the distance between them. We define the distance between two hotspots as the distance between the centers of mass of the two hotspots. We first calculated the pairwise distance between two hotspots and recorded the number of transitions from the first hotspot to the second; this results in a distance between hotspots and a number of transitions. The number of transitions between hotspots A and B is in general different from the transitions between B and A. The results are shown in Fig. 4.11. A star is plotted corresponding to a distance and number of transition pair. The distance in the figure is normalized with respect to the size of the grid (l) and the number of transitions is normalized by the total number of nodes. Since there are hotspot pairs with no transitions between them, we also add 1 to the transition numbers before taking the logarithm. Figure 4.11 shows two facts: first that most transitions occur between neighboring hotspots; second that the number of transitions between hotspots that are close to each other has a large variance. Also, the distributions in Fig. 4.11 indicate population similarity.



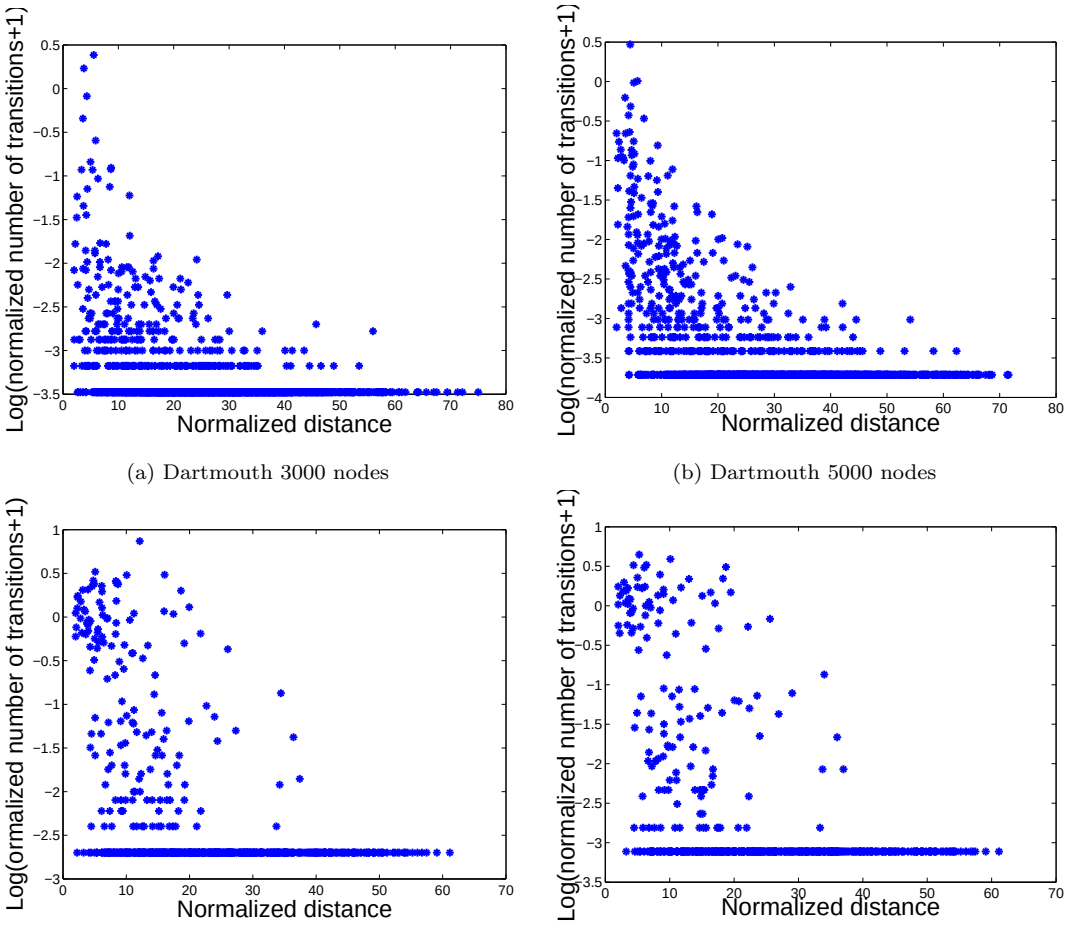
(a) Dartmouth 3000 nodes

(b) Dartmouth 5000 nodes

(c) Seattle 500 nodes

(d) Seattle 1000 nodes

Figure 4.10: The size of the hotspots as a function of their popularity.



(a) Dartmouth 3000 nodes

(b) Dartmouth 5000 nodes

(c) Seattle 500 nodes

(d) Seattle 1000 nodes

Figure 4.11: The number of transitions between hotspots as a function of the distance between the hotspots.

Pause Time

The last parameter we consider for the hotspot is the pause time, i.e., the time period that a user remains in the same location with zero or very low speed. The distribution of pause time is highly related to the stability of the network topology, scenarios with longer pausing times have fewer link changes. In RWP, nodes pause for a duration selected from a uniform distribution with user selected limits. Similar to the case of the speed, such distribution is not realistic. Therefore, to generate synthetic traces with realistic pause times, it is necessary to extract more accurate distributions from the real trace. At the same time, to enable diversity, we study the statistical similarities of the pause times.

For some scenarios, nodes are likely to move within the same hotspot and thus cause longer pause times in larger hotspots, for example, students in a department usually have most classes in the same building. In order to eliminate the affect of the hotspot size on the pause time, we considered the statistical features of pause time in a grid within a hotspot rather than in a hotspot. Through our study, we found that the distribution of the pause time in each grid does not change significantly in different hotspots. Therefore, for pause time, we aggregated samples in all grids within all hotspots; the resulting cumulative distribution function (cdf) is shown in Fig. 4.12. We used cdf here for better visibility. In addition to strong population similarity, the figure shows that the students in Dartmouth tend to pause much longer in a grid than the buses in Seattle.

For all parameters related to hotspots, we considered only the Dartmouth and the Seattle traces, since in the UCSD trace, only three hotspots were found, (preventing any statistical analysis).

4.2.4 Impact on Protocol Performance

In this section we evaluate the impact of the mobility features including arrival and departure process, speed and hotspots on the performance of AODV, a commonly used MANET routing protocol, using the Dartmouth trace. The goal is to determine

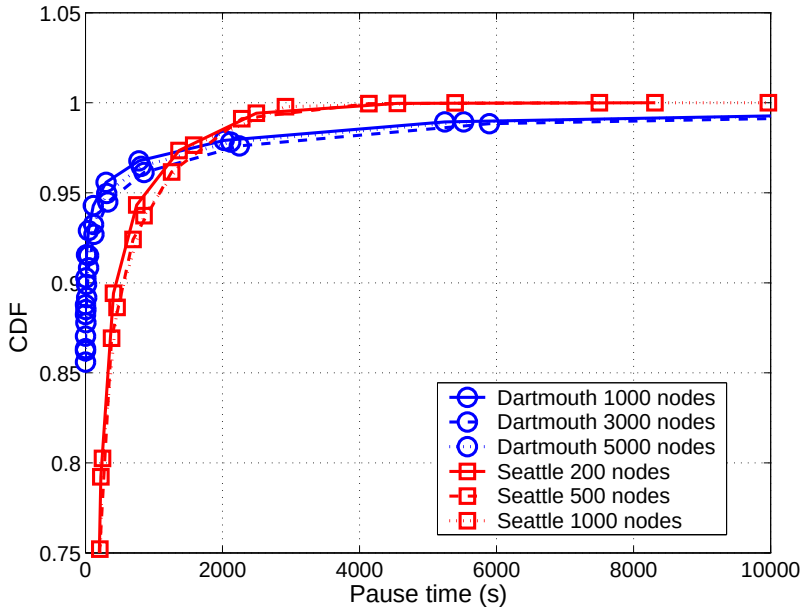


Figure 4.12: The CDF of the pause time in a grid.

how much a single feature can affect the performance. To achieve the goal, we generate four sets of different traces. While the first trace is generated using all features, the other three use only two of the three features. The generated traces are then fed to the NS2 network simulator to check the performance of AODV. We also compare the performance with the one of the real trace.

In all scenarios, we used CBR network traffic patterns with packet size of 512 bytes. We set the transmission range to 200m and the total simulation duration from 0am to 12pm. The MAC protocol used in the evaluation is 802.11 with default settings. Traffic patterns are selected between random source-destination pairs. From last section, it is clear that all features have population similarity, therefore, for our evaluation, we generate a series of traces with different number of nodes from 20 to 100. Each simulation is executed 5 times with different traffic patterns and different sets of nodes for real traces, and each result is averaged over all runs. The details of the trace generation mechanism will be shown in the next section. Briefly, for traces using hotspot features, we use the original hotspot deployment, which means that the position, the area and the popularity for each hotspot are invariant with the

real trace. The pause time in each grid is obtained from the empirical distribution in Fig. 4.12. For traces using speed distribution, the empirical distribution in Fig. 4.7 is used. The ranges of the speed and pause time are consistent with the values observed in the real trace. Last, for traces with arrival/departure processes, we used the Hurst parameters and the mapped distributions (normal and log-normal).

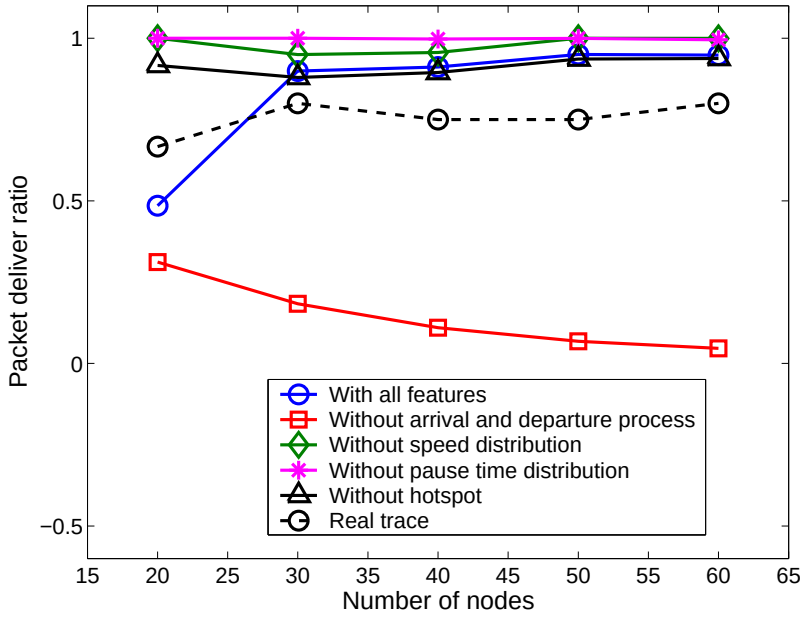


Figure 4.13: The packet delivery ratio for different Dartmouth mobility traces.

In Fig 4.13, we show the packet deliver ratios for the Dartmouth scenario. From the figure, we can see that the performance generated by using all features is the closest to the real trace. Interestingly, the features affects the performance of AODV the most is the arrival and departure process in both scenario. Without such processes, all nodes are in the scenario from the beginning to the end. Intuitively, more nodes causes higher congestion rate, which increase of the failure rate for packet delivery. Another observation is that when the number of node is small, the lack of features including hotspot, speed and pause time cause more dissimilarity among the curves. However, when the number of nodes is large, the effect is minimized. Fig 4.14 shows the average delays for all received packets, normalized by the number of total received packets. For this performance, except for arrival and departure processes and

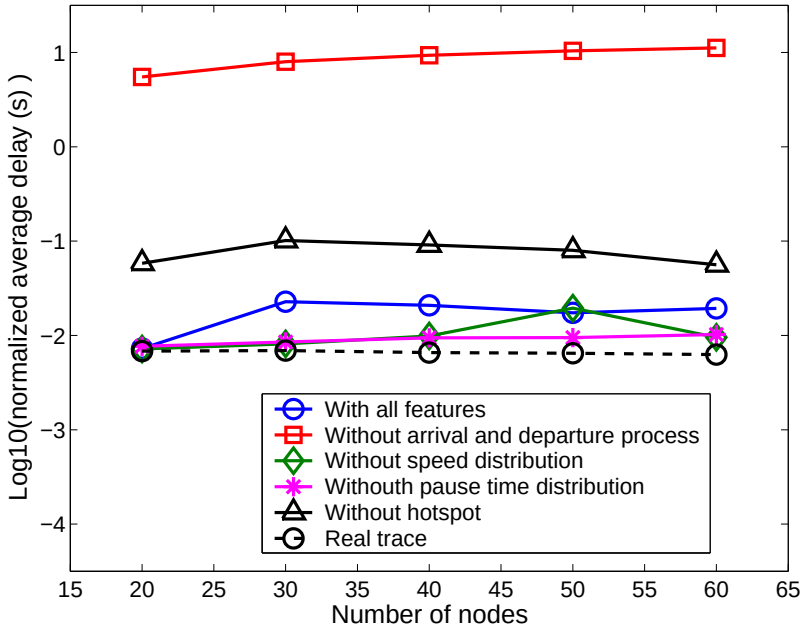


Figure 4.14: The normalized average delay for different Dartmouth mobility traces.

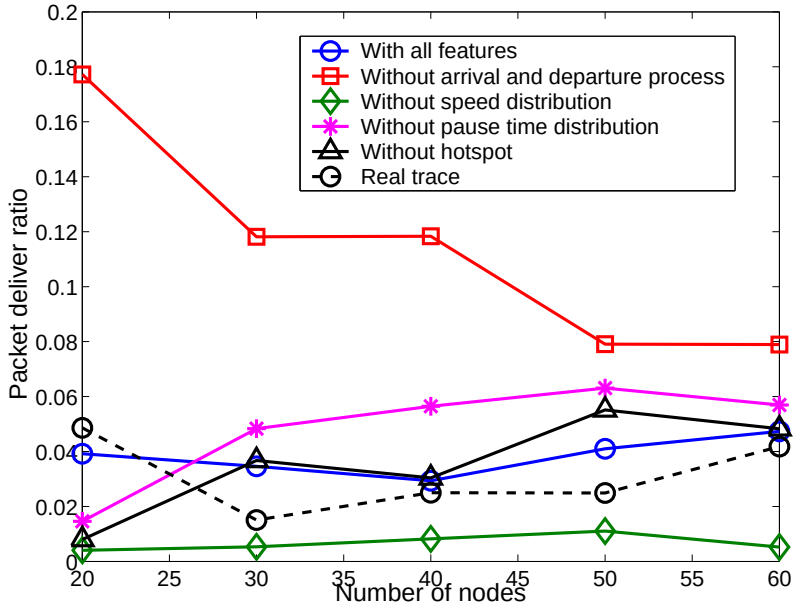


Figure 4.15: The packet delivery ratio for different Seattle mobility traces.

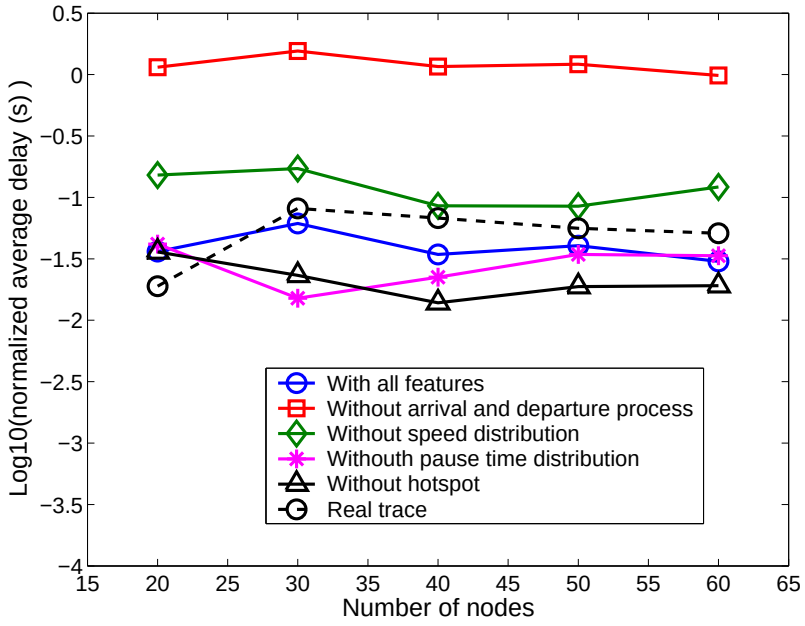


Figure 4.16: The normalized average delay for different Seattle mobility traces.

the hotspots, other features do not have significant impact. In Fig 4.15, we show the packet deliver ratio for the Seattle scenario. Again the curve generated using all features is most similar to the real trace. Notice that though the arrival and departure features are still the most significant ones to affect the performances, the effect caused by this feature is different compared to the Dartmouth trace. In the Dartmouth trace, the packet deliver ratios generated using arrival and departure processes are higher than the ones generated without using the processes. As mentioned before, the reason to cause such difference is that in the Dartmouth scenario, the campus is compact and all mobile nodes sharing the same wireless media. Thus no arrival and departure processes implies more nodes and thus more congestion; on the other hand, in the Seattle scenario, the network covers the whole city. Most nodes has only a few neighbors and therefore there is not much competition for accessing the media. When using the arrival and departure processes, there are less nodes in the field than when not using them, which would cause potentially less neighbors for each nodes, and consequently higher delivery failure rate. Besides, the packet deliver ratios in Seattle case are generally lower than the Dartmouth case. This is also cause by larger

deployment area and less nodes' degree in the Seattle case. As for the average delays in the Seattle case, similarly to Dartmouth case, except for the arrival and departure processes, other features do not have significant impact.

4.3 Mobility Trace Generation

Assuming each node moves independently, using the features we studied in section 4.2 and their similarity properties, here, we illustrate how these properties can be used to generate synthetic traces. The trace generation can be divided in three components: generation of the arrivals and departures, hotspots and mobilities.

4.3.1 Generation of Arrivals/Departures

In section 4.2.1, we showed that in order to be accurate, the arrival and departure processes possess time-varying rates and self-similarity. Therefore, different distributions should be used for each hour. We found that for hours without self-similarity, a Gaussian distribution $f_G(n)$ is a good approximation for the number of arrivals/departures n in the unit time (in this paper we used 1 minute) within an hour. For hours with self-similarity, in order to capture the long range dependency property, we used a log-normal distribution $f_L(n)$. Though log-normal distribution is not heavy-tailed, since the samples from the traces usually have a large but finite variance, the log-normal can be a good model to fit such distribution. Therefore, for each hour, the only recorded parameters are Hurst parameter H, the mean μ and the variance σ for that hour.

Next, for hours without self similarity, we can simply generate the random variable of number of arrivals/departures in unit time using the mapped Gaussian distribution $f_G(n)$. However, for self similar hours, in order to regenerate the self similarity, both the mapped log-normal distribution $f_G(n)$ and the fractional ARIMA$(0, d, 0)$ model are used, where the *fractional differencing* parameter $d = H - 0.5$. Details about this model can be found in [81]. In summary, the arrival and departure processes within

an hour can be implemented as in Fig. 4.17.

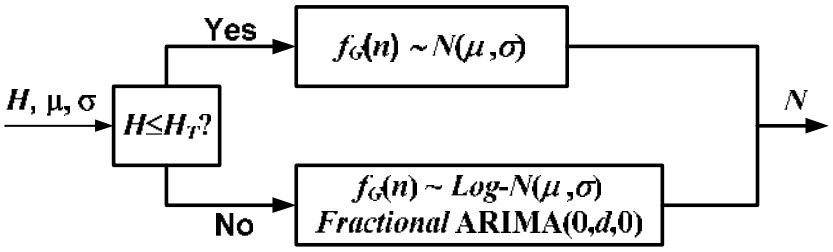


Figure 4.17: Generation process for arrival and departure.

Parameter H_T in Fig. 4.17 is a self similarity threshold used to decide which processes to use. In our simulation, we used $H_T = 0.6$. The result N is a sequence of number of arrivals/departures in each minute of the hour in question.

4.3.2 Hotspot Generation

We can consider two methods of generating hotspots: The first method is to generate a map on which the hotspots are exactly the same as in the original trace, i.e., the position, the area and the popularity for each hotspot are identical with the ones in the real trace.

The second type is to generate a map that is statistically similar to the original trace. Such generation is valuable in terms of diversification, (e.g., when considering a simulation in a larger campus based on real traces taken in a smaller campus). Assume that the total area of the new generated map is S_{new} and the density of the hotspot is ρ_{new}, while the corresponding parameters for the real trace are S_{orig} and ρ_{orig}. If in the original trace there are N_{orig} hotspots, in the generated trace, the number of hotspots to generate is:

$$N_{new} = N_{orig} \cdot \frac{\rho_{new} \cdot S_{new}}{\rho_{orig} \cdot S_{orig}}. \tag{4.4}$$

For each of the N_{new} hotspots, we generate its area according to the area distribution $f_A(a)$ (in which a is the area variable) obtained in Fig. 4.9. We then randomly

drop N_{new} points in the new map and calculate the pairwise distances for all points. Based on the relation between the distance and the number of transitions $f_{D-P}(d)$ (in which d is the distance variable) in Fig. 4.11, for a hotspot i, we can obtain a sequence of transition numbers from other hotspots $j, j \neq i$, to hotspot i using the distance d_{ji}. We then normalize the sequence by the total number of transitions into hotspot i and the resulted sequence can be used as the transition probabilities p_{ji} from hotspots $j, j \neq i$ to hotspot i. By listing all the transition probabilities sequences for each hotspot, we obtain a transition matrix. Thus, the popularity $\pi_i, i = 1, \ldots, n = N_{new}$ of each hotspot can be calculated as the stationary distribution of a Markov chain:

$$\begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix} = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix} \tag{4.5}$$

with $\sum_{i=1}^{n} \pi_i = 1$.

The next step is to map each popularity to an area. Recall that in Fig. 4.10, we recorded the size of a hotspot and its popularity $f_{P-A}(\pi, a)$. Therefore, for a generated popularity π_i for hotspot i, we can choose a corresponding area range (from the records found in $f_{P-A}(\pi, a)$). The area of the hotspot i is thus decided by randomly choosing a generated area that falls into the area range corresponding to popularity π_i. The hotspot generation procedure is summarized in Fig. 4.18.

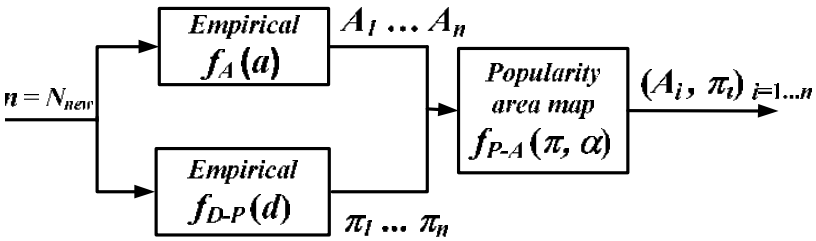


Figure 4.18: Hotspot generation procedure.

4.3.3 Mobility Generation

The mobility generator creates trajectories for each node while using the arrival/departure and the hotspots obtained above. The initial positions of all nodes are in the hotspots and distributed according to the popularity of the hotspots. Next, as long as the time is smaller then the generated time of departure for a node, nodes move between hotspots. In our model, we used a grid by grid movement, which means between the source and destination position, there are several intermediate sources and destinations. For example, if a node is now at position (x_i, y_i) in hotspot i, it chooses its next destination (x_j, y_j) which is located in hotspot j. Then for each grid that it is going to be passed by, we randomly select a position in that grid as one intermediate destination with the position in the previous grid as the intermediate source. The node also chooses a speed, using the speed distribution in Fig. 4.7 as its moving speed from i to j; the speed is constant between i and j. Depending on the position of the node at a hotspot, the node can either directly move out of the hotspot or it will pass through several grids within the hotspot before moving out. In our generator, as long as a node is still in a hotspot, (e.g, it moves to a new grid in the same hotspot), the node selects a pause time in that grid from the pause time distribution in Fig. 4.12, until it moves out of the hotspot. It behaves in the same way at the destination hotspot before reaching the destination grid. After the node reaches its destination, it chooses another pause time before moving to the next hotspot.

4.3.4 Generalized, Trace Driven Mobility Generator

The mobility analysis of the traces can be combined with the mobility generator to produce diversified traces with an increased degree of realism. Figure 4.19 shows the block diagram of the trace generator. The statistical analysis on the real trace is used to generate realistic traces, while user defined parameters (e.g., population size, area, etc.) can generate diversity.

To validate the arrival and departure generation process and the hotspot gener-

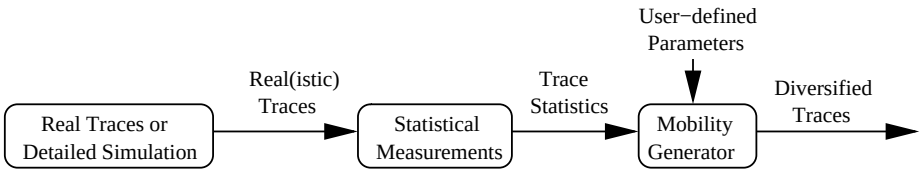
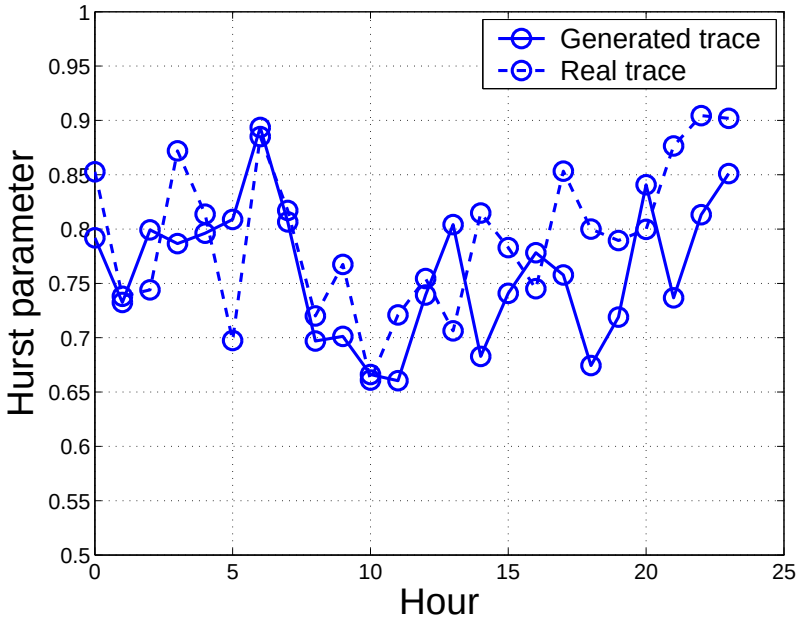


Figure 4.19: Generalized trace driven mobility generator.

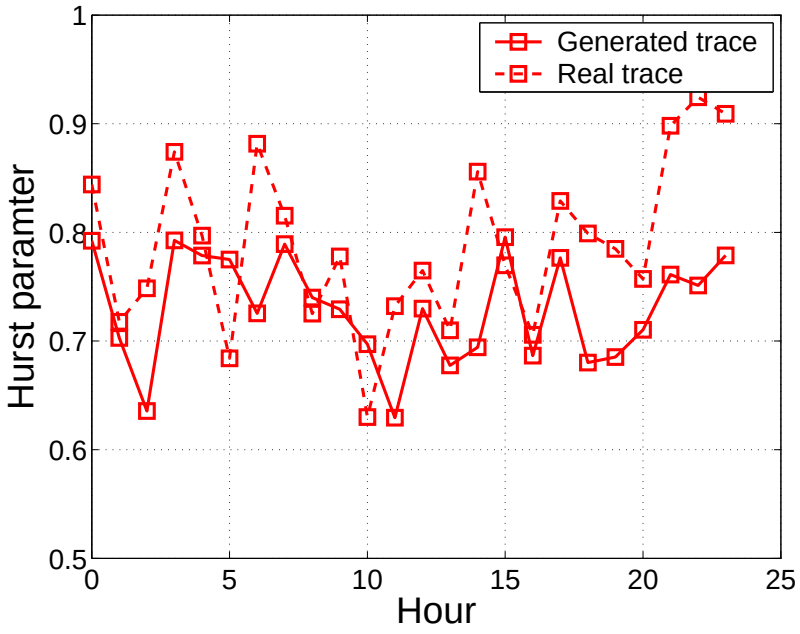
ator, we regenerated arrival and departure processes and the hotspots for the Dartmouth trace using the procedures described above. The Hurst parameters of the generated arrival and departure processes are plotted in Fig. 4.20 and the distribution of the generalized popularity for the generated hotspots is plotted in Fig. 4.21. The popularity is multiplied by the total number of the hotspots for comparison. The figures show that the two generators produce Hurst parameters and the popularity distribution similar to the real trace.

4.4 Conclusion

In this chapter we extracted and analyzed the statistical properties of several features that are important in defining the mobility patterns in the real traces. We verified that the existence of several types of statistical similarities (invariance) in those features, given traces with different population sizes and different scenarios, is essential for generating realistic and diversified synthetic traces using the mobility patterns extracted from real traces. Therefore, using the statistical properties of the extracted features and the observed similarities, we proposed a new mobility generator, which can be used to analyze the real traces and generate the synthetic traces for general scenarios. The main advantage of the proposed generator is that it can accurately capture the statistical properties in the real traces while being capable of generating diversity.



(a) Arrival



(b) Departure

Figure 4.20: Hurst parameter of the generated arrival and departure processes.

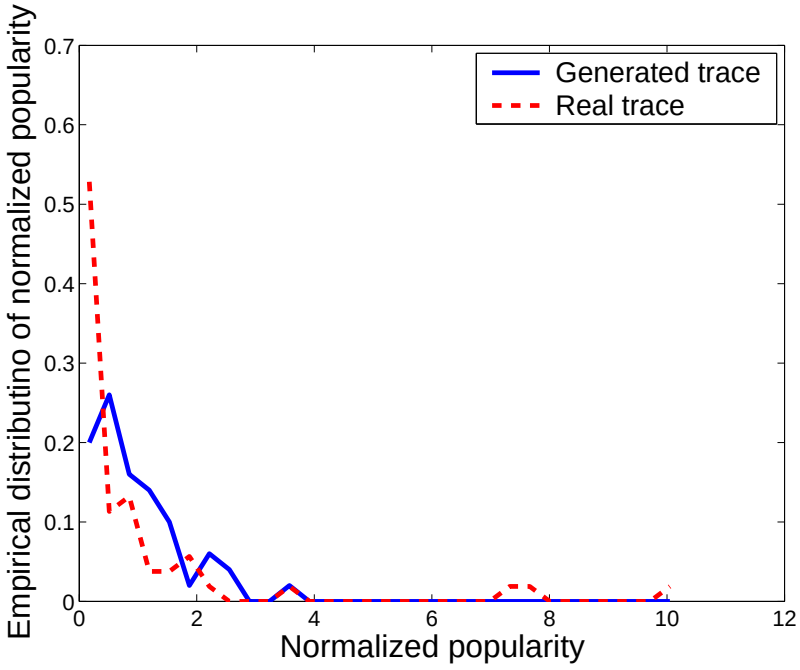


Figure 4.21: Empirical distribution of the generalized popularity.

Chapter 5

Conclusion

In the past decade, a lot of research activity has been devoted to the wireless ad hoc networks due to their great potential in military and commercial applications. With decentralized control, wireless ad hoc networks communicate in a multi-hop manner for nodes that are out of each other's transmission range. Wireless ad hoc networks require advances in a wide range of technical areas such as sensors, distributed computing, data fusion, signal processing, and networking protocols. Despite their inherent technical challenges, wireless ad hoc networks have attracted significant interest due to their advantages of fast and easy deployment in many unfriendly environments, such as vehicle communication and sensing application. In this dissertation, we focus on two technical challenges: localization in the stationary wireless sensor networks and mobility modeling in the mobile ad hoc networks.

Location service for a wireless sensor network is important since many sensor network applications require the awareness of the position of each sensor node. However the task of localizing the sensor nodes is not trivial: localization challenges include inaccurate range measurements, a small number of beacons and limited power and computing resources. In our proposed probabilistic localization algorithms, we address the first two problems. In summary we:

- Derived and proposed efficient models to accurately model the probability dis-

tributions for both RSS and angle range measurements.

- Proposed a fully distributed localization algorithms that uses probabilistic constraints to reduce the uncertainties and to estimate the positions of the sensor nodes.
- Proposed a probabilistic approach with reduced complexity using angle measurements, to estimate the orientation of the sensor nodes.
- Proposed probabilistic negative constraints to further improve the estimation accuracy and precision.
- Used FFT to reduce the computational complexity for the probabilistic approaches.
- Proposed effective methods to eliminate the dependencies (redundancies) between the probabilistic position information at sensor nodes (pseudo-beacons).
- Provided comprehensive performance evaluations.

The simulation results of localization algorithms using both RSS and angle range measurements show that the probabilistic approach outperforms other existing range-based localization approaches and approaches the Cramer-Rao lower bound.

Accurate mobility models are essential to build simulations for valid evaluations of many routing protocols. Mobility models used now falls into three categories, statistical models, trace based models and detailed simulators. The statistical models are simplistic, diversified but unrealistic, while the trace based models and detailed simulators are accurate but lack diversification. Given the drawbacks of both models, we propose a hybrid model that is driven by real traces but also use the strength of the statistical models. The proposed model is capable of generating realistic mobility traces for diverse scenarios with different configurations but without the requirements of the highly detailed mobility information. Our contribution in this area includes:

- Extracted and analyzed the statistical properties of several mobility features that predominantly determine the behavior of the mobile users in different scenarios.

- Defined and discovered the statistical similarities among same traces with different configurations and among different traces.
- Evaluated the effect of the mobility features and their statistical properties on the performance of AODV. The simulation results show that each statistical property extracted has a significant impact on the performance of AODV.
- Proposed a simplistic trace-based generator utilizing the statistical properties and similarities to generate realistic and diversified mobility traces.

Bibliography

[1] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Pearson Education, 2001.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, vol. 40, no. 8, pp. 102–116, Aug. 2002.

[3] ——, "Wireless sensor networks: A survey," *IEEE Computer*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[4] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser, and H. Marcy, "Wireless integrated network sensors: Low power systems on a chip," in *Proc. of the 24th European Solid-State Circuits Conference*, The Hague, Netherlands, 1998. [Online]. Available: citeseer.nj.nec.com/278712.html

[5] N. Bulusu, *Wireless Sensor Networks: A Systems Perspective*, 1st ed. Artech House, 2005.

[6] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. of SenSys 2004, the 2nd international conference on Embedded networked sensor systems*.

[7] "The zebranet wildlife tracker," http://www.princeton.edu/ mrm/zebranet.html.

[8] A. Mainwaring, J. Polastre, R. Szewczyk, , and D. Culler, "Wireless sensor networks for habitat monitoring," in *ACM Workshop on Sensor Networks and Applicationsn*, 2002.

[9] "Habitat monitoring on great duck island," http://www.greatduckisland.net.

[10] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8.

[11] R. Want, A. Hopper, V. Falco, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, pp. 91–102, Jan. 1992.

[12] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, pp. 28–34, Oct. 2000.

[13] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE Infocom 2001*, vol. 3, Anchorage AK, Apr. 2001, pp. 1655–1663.

[14] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," in *ACM MobiCom*, 2003, pp. 81–95.

[15] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. of Infocom'2000*, vol. 2, Tel Aviv, Israel, Mar. 2000, pp. 775–584.

[16] N. Priyantha, A. Chakraborthy, and H. Balakrishnan, "The cricket location-support system," in *Proc. of International Conference on Mobile Computing and Networking*, Boston,MA, Aug. 2000, pp. 32–43.

[17] D. Niculescu and B. Nath, "Ad-hoc positioning system," in *Proc. of IEEE GLOBECOM*, 2001.

[18] ——, "DV Based Positioning in Ad hoc Networks," *Telecommunication Systems*, 2003.

[19] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. of Mobicom'2001*, Rome, Italy, July 2001, pp. 166–179.

[20] 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*, Atlanta, GA, Sept. 2002.

[21] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *Proc. of ICASSP'01*, vol. 4, 2001, pp. 2037–2040.

[22] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *Cluster Computing*, vol. 5, no. 2, April 2002.

[23] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *Second ACM Conference on Embedded Networked Sensor Systems*, Nov. 2004, pp. 50–61.

[24] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling," in *IEEE INFOCOM*, Mar. 2004, pp. 2652–2661.

[25] A. Galstyan, B. Krishnamachari, K. Lerman1, and S. Pattem, "Distributed Online Localization in Sensor Networks Using a Moving Target," in *the third international symposium on information processing in sensor networks*, 2004, pp. 61–70.

[26] V. Ramadurai and M. L. Sichitiu, "Localization in wireless sensor networks: A probabilistic approach," in *Proc. of the 2003 International Conference on Wireless Networks (ICWN 2003)*, Las Vegas, NV, June 2003, pp. 275–281.

[27] M. L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," in *Proc. of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, Fort Lauderdale, FL, Oct. 2004.

[28] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *IEEE INFOCOM*, Apr. 2003.

[29] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, Sept. 2002.

[30] P. Biswas, H. Aghajan, and Y. Ye, "Integration of angle of arrival information for multimodal sensor network localization using semidefinite programming," in *39th Asilomar Conference on Signals, Systems and Computers*, Nov. 2005.

[31] R. Peng and M. L. Sichitiu, "Robust, probabilistic, constraint-based localization for wireless sensor networks," in *Proc. of SECON 2005*, Santa Clara, CA, Sept. 2005.

[32] ——, "Probabilistic localization for wireless sensor networks," *ACM Mobile Computing and Communications Review*, vol. 11, no. 1, Jan. 2007.

[33] V.Ramadurai and M.L.Sichitiu, "Localization in wireless sensor networks: A probabilistic approach," in *Proc. of the 2003 International Conference on Wireless Networks (ICWN 2003)*, Las Vegas, NV, June 2003, pp. 275–281.

[34] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *Proc. of SECON 2006*, Veston, VA, Sept. 2006.

[35] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Comm. and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, no. 5, 2002.

[36] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic Publishers, 1996, vol. 353.

[37] W. Feller, *An Introduction to Probability Theory and its Applications (Volume 1)*, 3rd ed. Wiley, 1968.

[38] C. Bettstetter, G. Resta, and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 257–269, Jul–Sep 2003.

[39] R. A. Guérin, "Channel Occupancy Time Distribution in a Cellular Radio System," *IEEE Trans. on Vehicle Technology*, vol. 35, no. 3, pp. 89 – 108, Aug 1987.

[40] Z. J. Haas and M. R. Pearhnan, "The performance of query control schemes for the Zone Routing Protocol," in *ACM SIGCOMM*, Vancouver, Canada, Sep 1998.

[41] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *IEEE INFOCOM*, Apr 2003.

[42] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Trans. on Mobile Computing*, vol. 3, no. 1, pp. 99 – 108, 2004.

[43] C. Bettstetter, "Smooth is better than sharp: a random mobility model for simulation of wireless networks," in *the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, Rome, Italy, 2001, pp. 19–27.

[44] B. Liang and Z. Haas, "Predictive distance-based mobility management for PCS networks," in *IEEE INFOCOM*, Mar 1999.

[45] Y. C. Hu and D. B. Johnson, "Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks," in *ACM MobiCom*, Boston, MA, Aug 2000.

[46] E. Hyytiä, P. Lassila, and J. Virtamo, "A Markovian Waypoint Mobility Model with Application to Hotspot Modeling," in *IEEE ICC*, June 2006.

[47] X. Hong, M. Gerla, G. Pei, and C. C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," in *the 2nd ACM International Workshop on Modeling, Analysis, Simulation of Wireless and Mobile Systems*, Seattle, 1999.

[48] M. Sanchez and P. Manzoni, "A Java-Based Ad Hoc Networks Simulator," in *the SCS Western Multiconference Web-based Simulation Track*, Jan 1999.

[49] M. Kim, D. Kotz, and S. Kim, "Extracting A Mobility Model from Real User Traces," in *IEEE INFOCOM*, Barcelona, Catalunya, Spain, Apr 2006.

[50] M. Papadopouli, H. Shen, and M. Spanakis, "Modeling Client Arrivals at Access Points in Wireless Campus-Wide Networks," in *14th IEEE Workshop on Local and Metropolitan Area Networks*, Chania, Crete, Greece, Sep 2005.

[51] C. Tuduce and T. Gross, "A Mobility Model Based on WLAN Traces and its Validation," in *IEEE INFOCOM*, Miami, FL, Mar 2005.

[52] W. Hsu and A. Helmy, "Impact: Investigation of mobile-user patterns across university campuses using wlan trace analysis," USC, Los Angeles, CA, Tech. Rep., Jul 2005.

[53] N. Eagle and A. Pentland, "Reality Mining: Sensing Complex Social Systems," *Journal of Personal and Ubiquitous Computing*, 2005.

[54] M. McNett and G. M. Voelker, "Access and Mobility of Wireless PDA Users," *ACM Mobile Computing Communications Review*, Apr 2005.

[55] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing User Behavior and Network Performance in a Public Wireless LAN," in *ACM SIGMETRICS*, Jun 2002.

[56] C. Reis, M. Rodrig, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks," in *ACM SIGCOMM*, Sep 2006.

[57] J. G. Jetcheva, Y. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture," in *the 5th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, Monterey, CA, Oct 2003.

[58] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure Coding Based Routing for Opportunistic Networks," in *the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, Philadelphia, PA, Aug 2005.

[59] W. Hsu, T. Spyropoulosn, and A. Helmy, "Modeling Time-variant User Mobility in Wireless Mobile Networks," in *IEEE INFOCOM*, 2007.

[60] W. Naumov, R. Baumann, and T. R. Gross, "An Evaluation of Inter-vehicle Ad Hoc Networks Based on Realistic Vehicular Traces," in *ACM Mobihoc*, Florence, Italy, May 2006.

[61] [Online]. Available: http://transims.tsasa.lanl.gov/

[62] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, 1st ed. Prentice Hall, 1993.

[63] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'dea, "Relative Localization Estimation in Wireless Sensor Network," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2137–2148, Aug. 2003.

[64] K. Langendoen and N. Reihers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Computer Networks*, vol. 43, no. 4, 2003.

[65] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, Sept. 2002.

[66] R. B. Ertel, P. Cardieri, K. W. Sowerby, T. S. Rappaport, and J. H. Reed, "Overview of spatial channel models for antenna array communication systems," *IEEE Personal Communications*, Feb. 1998.

[67] Q. H. Spencer, B. D. Jeffs, M. A. Jensen, and A. L. Swindlehurst, "Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, Mar. 2000.

[68] A. Abdi, J. Barger, and M. Kaveh, "A parametric model for the distribution of the angle of arrival and the associated correlation function and power spectrum at the mobile station," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 3, May 2002.

[69] K. I. Pedersen, P. E. Mogensen, and B. H. Fleury, "A stochastic model of the temporal and azimuthal dispersion seen at the base station in outdoor propagation environments," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, Mar. 2000.

[70] D. Astly and B. Ottersten, "The effects of local scattering on direction of arrival estimation with music," *IEEE Transaction on Signal Processing*, vol. 47, no. 12, Dec. 1999.

[71] J. B. Andersen and K. I. Pedersen, "Angle-of-arrival statistics for low resolution antennas," in *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 3, Bern, Switzerland, Mar. 2002.

[72] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, Nov. 2005.

[73] J. Yeo, D. Kotz, and T. Henderson, "CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth," *ACM Sigcomm Computer Communication Review*, Apr 2006.

[74] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Trans. on Networking (TON)*, vol. 2, no. 1, Feb 1994.

[75] M. E. Crovella and A. Bestavrosn, "Self-similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Trans. on Networking (TON)*, vol. 5, pp. 833 – 846, Dec 1997.

[76] ESTI, "Universal Mobile Telecommunications System (UMTS); selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03, version 3.2.0)," European Telecommunications Standard Institute (ESTI), European, Tech. Rep., Apr 1998.

[77] C. Hartmozm and H.-J. Vögel, "Teletraffic analysis of SDMA-systems with inhomogeneous MS location distribution and mobility," *Kluwer Wireless Personal Communications*, vol. 11, no. 1, Oct 1999.

[78] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*.

[79] J. C. Bezdec, *Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[80] J.Kang, W.Welbourne, B.Stewart, and G.Borriello, "Extracting places from traces of locations," in *the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, Philadelphia, PA, 2004.

[81] M. W. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," in *ACM SIGCOMM*, 1994, pp. 269–280.